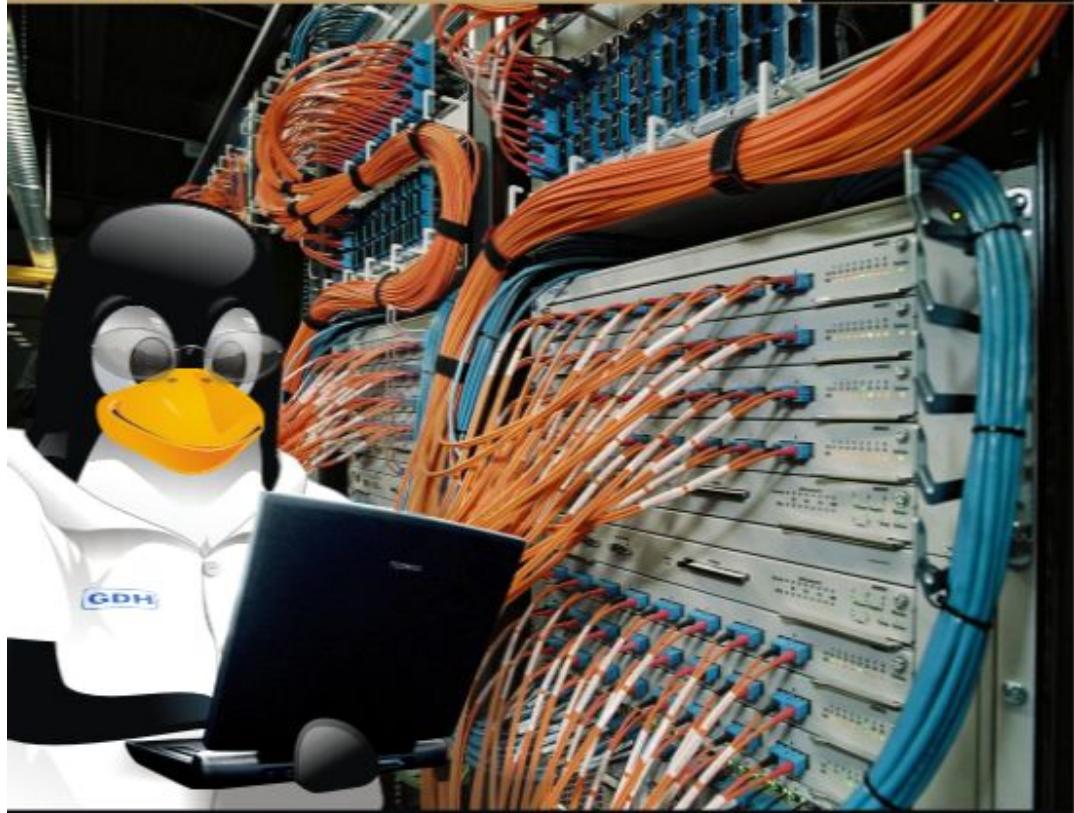


Linux Redes e Servidores

Guia Prático

2ª Edição
atualizada e ampliado



Carlos E. Morimoto

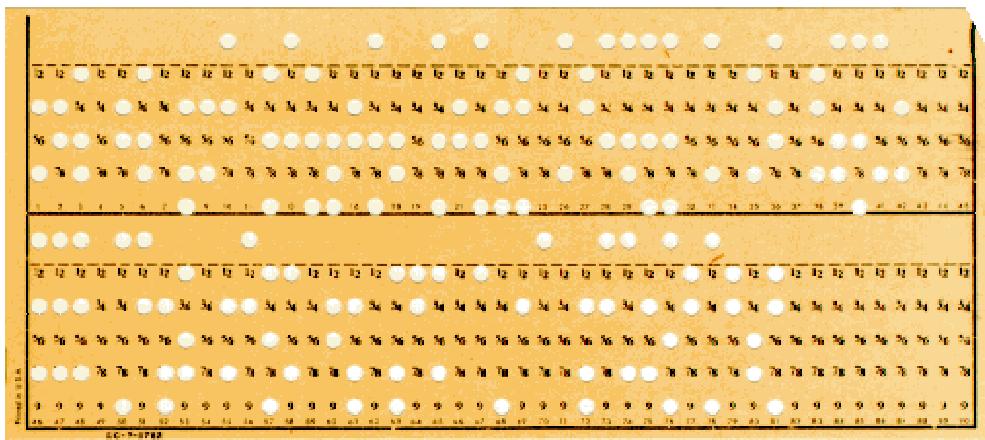
(desenvolvedor do Kurumin Linux)

<http://www.guiadohardware.net>

Inicialmente, as redes eram simplesmente uma forma de transmitir dados de um micro a outro, substituindo o famoso DPL/DPC (disquete pra lá, disquete pra cá), usado até então.

As primeiras redes de computadores foram criadas ainda durante a década de 60, como uma forma de transferir informações de um computador a outro. Na época, o meio mais usado para armazenamento externo de dados e transporte ainda eram os cartões perfurados, que armazenavam poucas dezenas de caracteres cada (o formato usado pela IBM, por exemplo, permitia armazenar 80 caracteres por cartão).

Eles são uma das formas mais lentas, trabalhosas e demoradas de transportar grandes quantidades de informação que se pode imaginar. São, literalmente, cartões de cartolina com furos, que representam os bits um e zero armazenados:



De 1970 a 1973 foi criada a Arpanet, uma rede que interligava várias universidades e diversos órgãos militares. Nesta época surgiu o e-mail e o FTP, recursos que utilizamos até hoje. Ainda em 1973 foi feito o primeiro teste de transmissão de dados usando o padrão Ethernet, dentro do PARC (o laboratório de desenvolvimento da Xerox, em Palo Alto, EUA). Por sinal, foi no PARC onde várias outras tecnologias importantes, incluindo a interface gráfica e o mouse, foram originalmente desenvolvidas.

O padrão Ethernet é utilizado pela maioria das tecnologias de rede local em uso, das placas mais baratas às redes wireless. O padrão Ethernet define a forma como os dados são organizados e transmitidos. É graças a ele que placas de diferentes fabricantes funcionam perfeitamente em conjunto.

A partir de 1995, com a abertura do acesso à internet, tudo ganhou uma nova dimensão e a principal função da maioria das redes passou a ser simplesmente compartilhar a conexão com a web. Estamos agora assistindo a uma segunda mudança, que é o uso da web não apenas para comunicação, mas como uma forma de rodar aplicativos. Inicialmente surgiram os webmails, depois os clientes de MSN/ICQ (como o meebo.com) e agora temos também processadores de texto, planilhas e outros aplicativos, desenvolvidos com base no Ajax ou outras ferramentas similares, que permitem desenvolver aplicativos web complexos, que rodam com um bom desempenho mesmo em conexões via modem.

Pouco a pouco, a internet se torna o verdadeiro computador, e o seu PC passa a ser cada vez mais um simples terminal, cuja única função é mostrar informações processadas por servidores remotos.

Isso se tornou possível devido à popularização do ADSL, wireless e outras formas de acesso rápido e contínuo, às redes locais, que permitem compartilhar a conexão entre vários micros (seja em casa, no escritório ou em uma lan-house) e a servidores como o Apache e o Bind, que formam a espinha dorsal da internet.

Futuramente, a tendência é que mais e mais aplicativos passem a ser usados via web, tornando um PC desconectado cada vez mais limitado e inútil. Eventualmente, é possível que o próprio PC seja substituído por dispositivos mais simples e baratos, que sirvam como terminais de acesso, mas isso já é um exercício de futurologia ;).

Dentro de uma rede local, tudo começa com o cabeamento, onde você pode usar a estrutura tradicional, com hubs e cabos de par trançado, investir em uma rede wireless, ou usar um misto das duas coisas, o que acaba sendo o mais comum hoje em dia.

As redes wireless são mais práticas, pois você pode acessar a rede de qualquer lugar e não precisa ficar espalhando cabos por aí, porém acabam saindo mais caro (embora os preços venham caindo rapidamente), são mais lentas e você precisa tomar um cuidado adicional com a segurança. Para montar uma rede cabeada, por outro lado, você precisa comprar apenas o hub/switch e os cabos, já que quase todas as placas-mãe hoje em dia possuem rede onboard.

Configurar a rede não é complicado, mas existem vários detalhes a abordar (principalmente dentro das redes wireless), por isso os três primeiros capítulos do livro são dedicados a explicar em detalhes como montar e configurar a rede. Temos em seguida o capítulo 4, que fala sobre segurança, incluindo o uso de programas como o Nessus, Ethereal e Kismet. Eles são facas de dois gumes, que podem ser tanto usados "para o bem", verificando a segurança da rede e ajudando a corrigir os problemas, quanto "para o mal", invadindo redes de outras pessoas. É importante que você saiba usá-los para detectar problemas de segurança na sua própria rede, antes que outras pessoas o façam por você.

Com a rede funcionando, o primeiro passo é compartilhar a conexão, o que pode ser feito de forma muito simples. Ao compartilhar a conexão, seu servidor passa a funcionar como um roteador, encaminhando os pacotes da rede local para a internet e vice-versa. As duas redes continuam sendo separadas, de forma que os micros da rede interna podem acessar os servidores disponíveis na internet, mas não podem ser acessados diretamente de fora, a menos que você ative o roteamento de portas no servidor.

Em muitas situações, o acesso à web precisa ser controlado. Em um ambiente de trabalho, não é muito interessante que os funcionários fiquem acessando o Orkut durante o expediente, e nenhum diretor vai querer que os alunos fiquem usando os micros da biblioteca para baixar filmes e música, por exemplo.

Entra em cena, então, a possibilidade de monitorar e limitar o acesso, usando a dupla **Squid** e Sarg. O Squid é um servidor proxy. Você pode usá-lo de duas formas: na forma convencional você pode criar logins de acesso e tem um relatório detalhado, com o que cada usuário acessa, mas, por outro lado, tem o trabalho de configurar cada micro para usar o proxy, manualmente.

Existe ainda a opção de configurar um proxy transparente (a mais usada), onde você perde a possibilidade de usar logins de acesso, mas em troca não precisa fazer nenhuma configuração manual nos micros da rede. Neste caso, o relatório dos acessos é baseado nos endereços IP dos micros. Além de servir de dedo-duro, o Squid mantém um cache das páginas e arquivos acessados, agilizando o acesso quando eles são acessados a partir de várias máquinas. Pense nos casos em que você precisa baixar e instalar uma atualização do Windows, ou instalar um programa grande via apt-get em 20 máquinas, por exemplo.

Completando o time, você pode incluir um servidor **DHCP**, que automatiza a configuração da rede (permitindo inclusive fornecer IPs fixos para os micros da rede), e um servidor DNS local, que permite que você acesse os micros da rede através de nomes, ao invés dos endereços IP. Tudo isso é abordado no capítulo 5.

Temos em seguida outra necessidade comum: compartilhar arquivos e impressoras. Em um grupo onde várias pessoas necessitam trabalhar nos mesmos arquivos (dentro de um escritório de arquitetura, por exemplo, onde normalmente várias pessoas trabalham no mesmo projeto), é muito útil centralizar os arquivos em um só lugar, pois assim temos apenas uma versão do arquivo circulando pela rede e os usuários passam a automaticamente trabalhar com a versão mais recente. Em uma situação mais corriqueira, você pode usar a rede para assistir um filme ou ouvir arquivos de música que estão em outro micro, sem precisar primeiro transferir tudo para o seu micro.

Centralizar e compartilhar arquivos permite também economizar espaço no HD, já que, ao invés de uma cópia do arquivo em cada máquina, passamos a ter uma única cópia localizada no servidor de arquivos. Com todos os arquivos no mesmo local, manter um backup de tudo também torna-se muito mais simples.

O Windows utiliza um protocolo chamado SMB para o compartilhamento de arquivos e impressoras. Você ativa o "Compartilhamento de Arquivos para redes Microsoft" nas configurações da rede e a partir daí surge a opção "Compartilhar" nas propriedades das pastas e impressoras. No Linux, usamos o **Samba**, que permite tanto compartilhar arquivos e impressoras com os micros Windows da rede, quanto acessar os compartilhamentos disponibilizados por eles. O Samba suporta uma série de opções avançadas e incomuns, o que adiciona uma enorme flexibilidade. Por outro lado, todos esses recursos tornam a configuração um pouco mais complicada que no Windows, por isso o capítulo 6 é dedicado a explicar a configuração em detalhes.

Em muitas situações, é mais prático rodar todos os programas a partir de um servidor central do que ter vários PCs separados, onde você precisa instalar e configurar o sistema

em cada um. Isso é útil tanto para facilitar a administração da rede, quanto para aproveitar máquinas antigas, lentas demais para rodar programas como o Firefox e o OpenOffice.

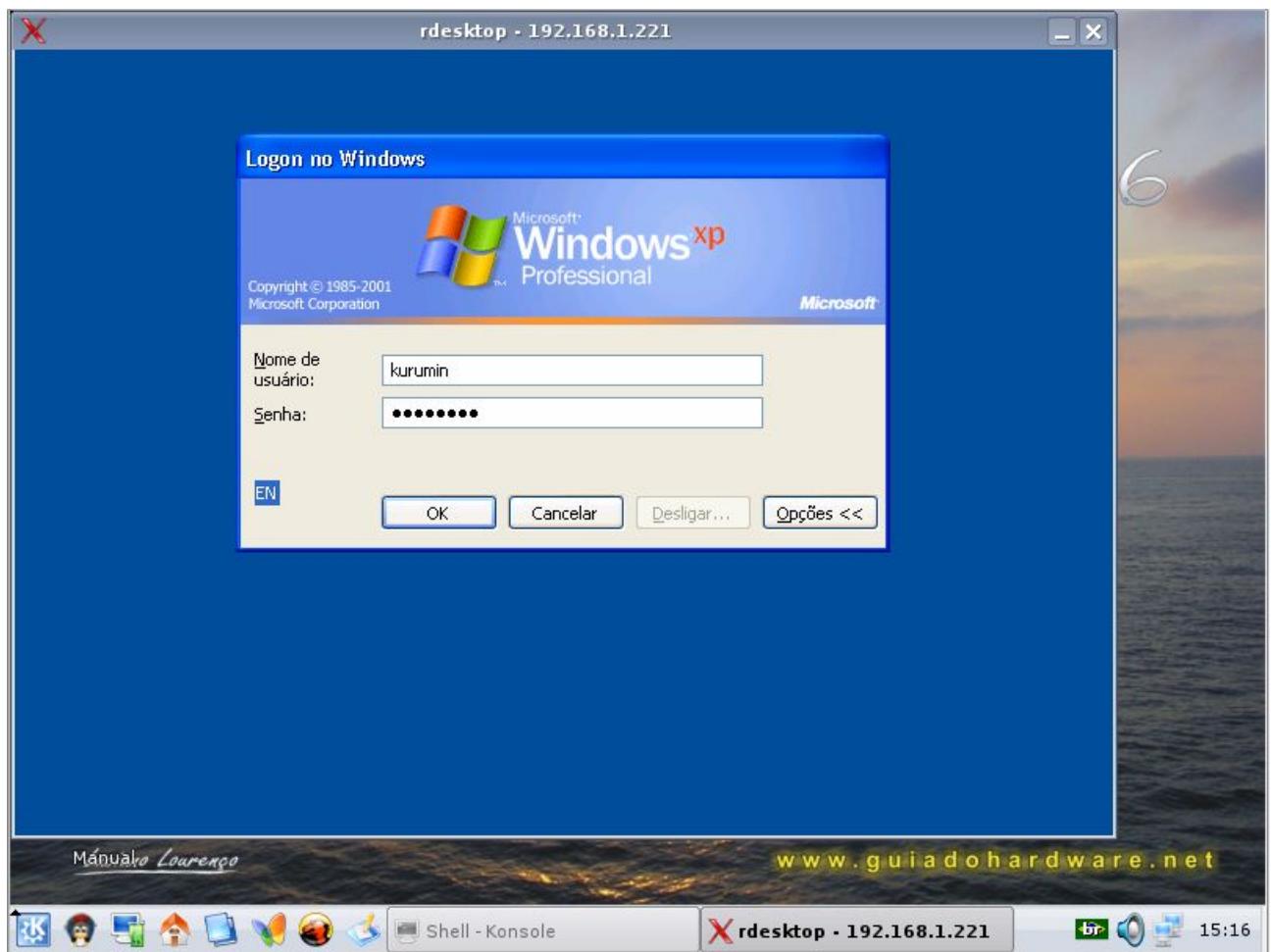
Dar manutenção e instalar qualquer sistema operacional em uma máquina antiga é mais complicado e demorado que numa atual. Em muitos casos, essas máquinas acabam sendo doadas a escolas e outras instituições, mas mesmo assim nem sempre são aproveitadas.

Ao invés de ter o trabalho de instalar o sistema e configurar cada micro individualmente e ainda assim ter que conviver com um sistema limitado e lento, é possível transformar micros a partir de 486 em terminais leves, onde um servidor mais rápido executa os aplicativos e envia apenas a imagem a ser mostrada na tela para os terminais. Isso é feito usando uma combinação de servidor de arquivos e servidor de acesso remoto, instalada através do **LTSP**, que veremos em detalhes no capítulo 9.

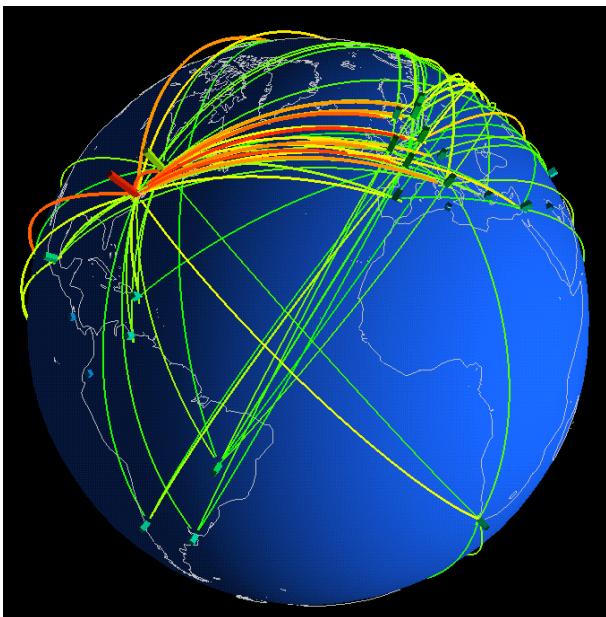
Qualquer PC atual, como um Sempron ou Celeron com 1 GB de RAM, pode servir como servidor LTSP para de 10 a 20 terminais. Um servidor mais robusto pode servir 40 terminais ou até mais, dependendo dos aplicativos usados. Neste caso, os aplicativos rodam com o desempenho máximo do servidor, fazendo com que cada estação rode o OpenOffice, Firefox e outros aplicativos com praticamente o mesmo desempenho que teriam se fossem usadas máquinas novas. Compre teclados novos e troque os gabinetes e os usuários realmente terão dificuldade em descobrir o truque ;).



Em casos onde as licenças de uso não são um problema, é possível usar também os terminais para rodar aplicativos Windows, usando o rdesktop. Neste caso, além do servidor LTSP, você inclui um servidor Windows, com o terminal services habilitado. O acesso ao servidor Windows pode ser simplificado criando um ícone no desktop, assim os usuários precisam de um único clique para abrir a janela do rdesktop, onde podem fazer login no servidor Windows e rodar os aplicativos disponíveis nele. O rdesktop pode ser usado em qualquer máquina Linux, como veremos em detalhes no capítulo 8.



Dentro da rede local, você dá as cartas e pode fazer muitas coisas da forma que achar melhor. Mas, quando configuramos um servidor que ficará disponível na Internet, é necessário seguir diversas regras. A internet é formada por uma malha de cabos de fibra óptica, que interligam praticamente todos os países. Mesmo em locais mais remotos, é possível se conectar via modem, via celular (ou algum tipo de rede sem fio de longa distância) ou mesmo via satélite. A peça central são os roteadores, que interligam diferentes segmentos da rede, formando uma coisa só. Cada roteador conhece os vizinhos, sabe quais redes estão conectadas a eles e sabe escolher o caminho mais curto para cada pacote de dados.



Naturalmente, os dados transmitidos precisam sair de algum lugar. Entram em cena então os servidores, que podem ser desde PCs normais, hospedados em algum data center, até supercomputadores.

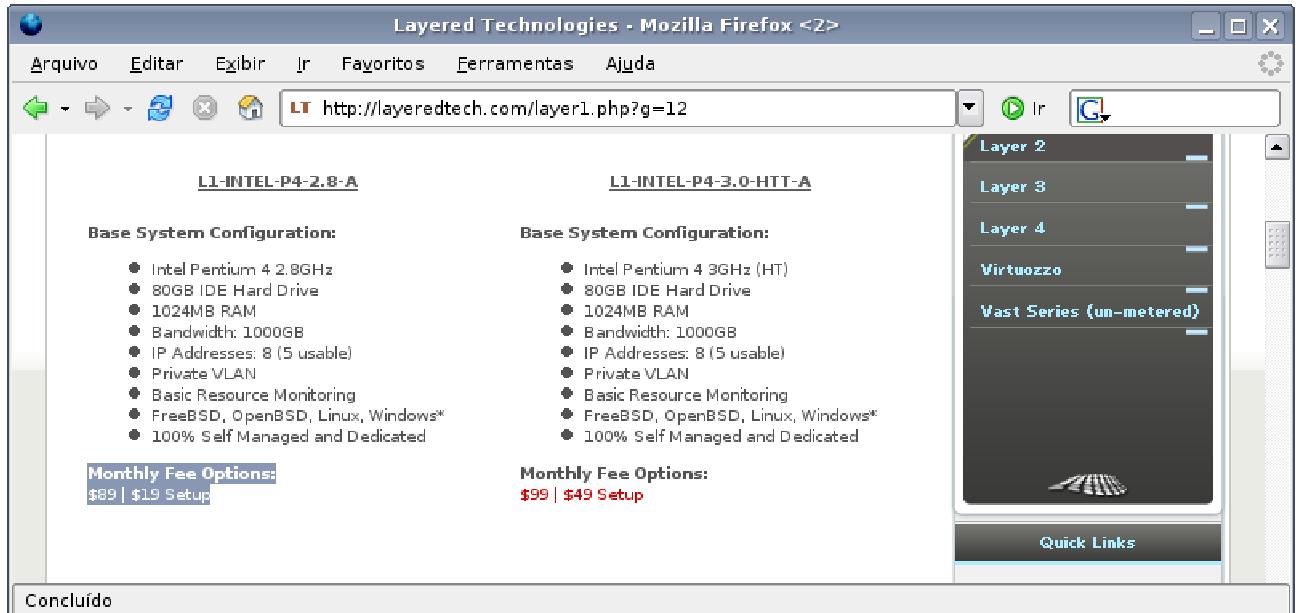
Antigamente, ter um **servidor dedicado** era um luxo reservado para poucos. Além da máquina em si, você precisava de um link dedicado, que custava um braço e duas pernas (por mês ;). A solução para colocar seu site no ar era pagar por um plano de shared hosting (hospedagem compartilhada), onde um mesmo servidor é compartilhado por milhares de sites diferentes e você fica restrito a uma quota de tráfego e espaço em disco, sem poder mexer na configuração do servidor.

Hoje em dia, tudo está muito mais acessível, graças ao barateamento tanto dos links, quanto dos micros. Você pode ter seu próprio servidor, hospedado em um data center dos EUA ou da Alemanha, hospedando o site da sua empresa, ou de seus clientes, compartilhando e baixando arquivos via bittorrent, armazenando backups, entre inúmeras outras funções.

A principal vantagem de alugar um servidor dedicado, hospedado em um datacenter ao invés de simplesmente montar seu próprio servidor e ligá-lo na conexão via ADSL ou cabo que você já tem, é a questão da conectividade. Uma conexão via ADSL funciona bem para acessar a web como cliente e fazer downloads, mas não é uma boa idéia para hospedar servidores, pois o link de upload é muito reduzido (em geral apenas 128 ou 300 kbits, de acordo com o plano), sem falar na questão da confiabilidade.

As grandes empresas de hospedagem trabalham com links absurdamente rápidos, sempre ligados simultaneamente a vários dos principais backbones. Isso garante uma boa velocidade de acesso a partir de qualquer lugar do mundo. A confiabilidade também é melhor, pois os datacenters são ambientes fechados, com geradores próprios, segurança física, links redundantes, etc.

Os dedicados mais baratos chegam a custar menos de US\$ 100 mensais, sempre com um link de pelo menos 10 megabits (geralmente com um limite de transferência mensal de 1 ou 2 terabytes):



Em um servidor dedicado, você precisaria configurar um servidor **DNS** para responder pelo domínio registrado (minhaempresa.com.br, por exemplo), um servidor **Apache**, com suporte a **PHP** e **MySQL** (como vemos no capítulo 7), para rodar as páginas web, servidor de **e-mail**, com um webmail para que os usuários possam acessar as mensagens usando o navegador (capítulo 10) e, naturalmente, um **firewall** (capítulo 11).

Para disponibilizar arquivos de forma pública, você poderia usar o próprio servidor web, instalar um servidor **FTP** (capítulo 7), ou mesmo rodar um tracker para disponibilizá-los via bittorrent. Para acesso seguro ao servidor, você pode usar o **SFTP** ou o **RSSH**, que abordo no capítulo 8.

Em casos onde outras pessoas tenham acesso limitado ao seu servidor (como ao hospedar vários sites, onde cada um tem acesso a seus arquivos), é útil incluir um sistema de quotas (capítulo 7), de forma que cada um tenha sua parcela justa de espaço e assim afastar o risco de alguém entupir o HD do servidor sozinho.

Como nesse caso você não tem acesso físico ao servidor, você usaria o **SSH**, **VNC** ou **FreeNX** (capítulo 8) para acessá-lo remotamente e, assim, poder fazer toda a configuração.

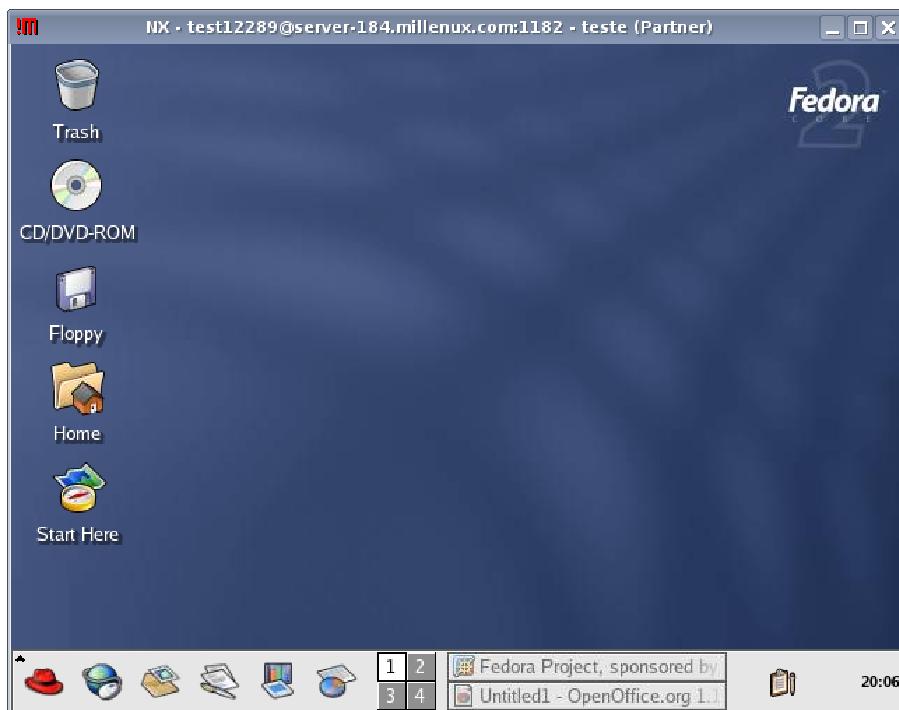
O **SSH** é a ferramenta mais usada para acessar o servidor via linha de comando. Para quem vem do Windows, isso parece uma coisa exótica, mas com o tempo você percebe que é realmente a forma mais rápida de fazer muitas coisas. A maior parte da configuração do servidor consiste em instalar pacotes e editar arquivos de configuração em texto, o que pode ser feito via linha de comando sem maiores problemas.

Existem ainda ferramentas como o Webmin, phpMyadmin e outras, que são acessadas via navegador. Elas são bastante práticas, pois você não precisa instalar o ambiente gráfico no servidor para usá-las, o que é desejável do ponto de vista da performance e até mesmo segurança.

Finalmente, temos o VNC e o FreeNX (o sistema mais atual, que é ao mesmo tempo mais rápido e mais seguro). Em ambos, você pode acessar o desktop de outras máquinas da rede, ou mesmo de servidores remotos, dentro de uma janela (como ao acessar um servidor Windows usando o rdesktop) ou mesmo em tela cheia.

Antigamente, rodar aplicativos remotamente era muito lento (pergunte para alguém das antigas, que já tenha tentado usar o PC Anywhere via modem ;), mas, hoje em dia, existem tecnologias de compressão e cache muito eficientes, que realmente permitem rodar aplicativos gráficos confortavelmente via ADSL ou qualquer outro tipo de acesso rápido.

O FreeNX utiliza um sistema bastante avançado de encriptação, compressão e cache, que torna a sessão remota surpreendentemente responsável e agradável de usar, mesmo ao acessar servidores distantes (ou com um ping alto) ou mesmo ao conectar via modem:



Tudo isso pode parecer complicado de início, mas ao longo do livro você vai descobrir que é bem mais simples do que parece. De certa forma, é bom que as outras pessoas continuem achando tudo muito difícil, pois assim você tem emprego garantido ;).

Estudando sobre a história do Linux e do Unix de uma forma geral, você vai perceber rapidamente que o sistema foi desenvolvido originalmente para uso em servidores, compartilhados por muitos usuários. Mais tarde, o sistema passou a ser usado também em

desktops, palms, celulares e vários outros tipos de dispositivos, mas sem nunca perder suas origens.

Um servidor é uma máquina que fica o tempo todo ligada, sempre fazendo a mesma coisa. Existem vários tipos de servidores, como servidores web, servidores de arquivos, servidores de impressão, etc., sendo que uma única máquina pode rodar simultaneamente vários serviços, dependendo apenas da configuração de hardware e da demanda.

Segundo as estatísticas de Abril (2006) da Netcraft, quase 70% dos servidores web do mundo usam o Apache, contra apenas 20.5% do IIS:
http://news.netcraft.com/archives/2006/04/06/april_2006_web_server_survey.html

Embora o Apache também rode sobre o Windows e outros sistemas, o mais comum é rodá-lo sobre o Linux, o que dá uma idéia da penetração do sistema em servidores Web. Temos diversos outros exemplos, como o caso do Samba, que permite substituir diretamente servidores de arquivos Windows e vem sendo cada vez mais usado. Quase todos os servidores DNS da internet utilizam o Bind, rodando sobre o Linux ou sobre alguma versão do Unix.

O Linux é também um dos sistemas mais utilizados para compartilhar a conexão e em firewalls. Ele é também um dos sistemas mais robustos para rodar bancos de dados, como o MySQL, Postgre SQL ou Oracle. O sistema é feito para ser configurado uma vez e depois ficar ativo durante anos, sem precisar de manutenção (além das atualizações de segurança, naturalmente), o que reduz brutalmente o trabalho necessário para manter um servidor dedicado no ar.

Um exemplo que gosto de comentar é um pequeno servidor, que compartilha arquivos e a conexão ADSL usando um proxy transparente, que instalei no início de 2003, usando uma das primeiras versões do Kurumin, em um Pentium 133 com 32 MB de RAM.

Além de ser um micro muito velho, que já não estava em suas melhores condições, o servidor não tinha no-break e por isso reiniciava com uma certa freqüência por falta de luz. Apesar dos maus-tratos, ele funcionou até o final de 2005, ou seja, por quase 3 anos (quando foi substituído por uma máquina nova), sem nunca ter dado problemas.

Um exemplo de maior porte é o servidor do <http://guiadohardware.net>, onde uma única máquina (um Athlon 64 com 1 GB de RAM) mantém todo o sistema do site, com suas quase 100.000 visitas diárias (cerca de 600.000 pageviews), mais o fórum, que possui cerca de 1.7 milhões de mensagens e 55.000 usuários registrados, além dos sistemas de backup, indexação de conteúdo e e-mail, uma carga de trabalho que seria impensável para um único servidor Windows.

De início, configurar um servidor Linux pode parecer complicado, pois existem muitas opções de distribuições e ferramentas de configuração disponíveis. Praticamente qualquer distribuição Linux pode ser usada como servidor, pois os serviços utilizados, como o Apache, Bind, MySQL, etc. serão os mesmos, mudando apenas o processo de instalação. Contudo, as distribuições mais usadas são o Debian, Fedora (ou Red Hat, para quem precisa de suporte comercial), Ubuntu, SuSE e Mandriva. Cada uma delas oferece um

conjunto diferente de utilitários de configuração, junto com utilitários "genéricos", como o Webmin e o Swat, que podem ser usados em qualquer uma.

O Debian é um caso interessante, pois, além do Debian "original", você pode escolher entre algumas centenas (sem exagero) de distribuições baseadas nele. Elas utilizam a mesma base de pacotes, os mesmos utilitários básicos e a mesma ferramenta de instalação de programas, o apt-get, mas vêm pré-configuradas de uma certa maneira e, em geral, oferecem alguns utilitários adicionais.

Entre as comerciais (pouco utilizadas aqui no Brasil), temos o Xandros e o Linspire. Entre as gratuitas, temos o Ubuntu, Knoppix, Kanotix (entre muitas outras) e o Kurumin, que desenvolvo.

Embora o Kurumin seja voltado para uso em desktops, nada impede que você o utilize também em servidores, já que está, na verdade, utilizando os pacotes do Debian, com a vantagem de contar com um conjunto de scripts que facilitam bastante a configuração. Quase todos os serviços que comento neste livro podem ser instalados no Kurumin de forma semi-automática, com o uso dos ícones mágicos. Você pode começar testando no Kurumin para ver o servidor funcionando e entender melhor sua configuração e depois encarar a instalação manual, descrita neste livro.



Ao estudar, recomendo que você comece procurando entender a configuração de cada serviço diretamente através dos arquivos de configuração (como ensino aqui) e, depois de entender bem o processo, comece a pesquisar e usar utilitários de configuração que facilitem seu trabalho. Quem entende as opções e sabe como configurar manualmente em geral não tem problemas para se adaptar a um dos utilitários de configuração, ao contrário de quem aprende apenas a trabalhar com o Webmin ou outro utilitário e acaba "preso" a ele.

Estes capítulos foram escritos tendo em mente quem já trabalha com Linux há algum tempo e possui familiaridade com o sistema. Se você está começando agora, recomendo que leia também meu livro *Linux, Entendendo o Sistema* (que faz parte desta mesma série).

Capítulo 1: A parte física: placas, cabos, conectores, hubs e switches

Os componentes básicos de uma rede são uma **placa de rede** para cada micro, os **cabos** e um **hub** ou **switch**, que serve como um ponto de encontro, permitindo que todos os micros se enxerguem e conversem entre si. Juntos, esses componentes fornecem a infra-estrutura básica da rede, incluindo o meio físico para a transmissão dos dados, modulação dos sinais e correção de erros.

As placas de rede já foram componentes caros. Mas, como elas são dispositivos relativamente simples e o funcionamento é baseado em padrões abertos, qualquer um com capital suficiente pode abrir uma fábrica de placas de rede. Isso faz com que exista uma concorrência acirrada, que obriga os fabricantes a produzirem placas cada vez mais baratas, trabalhando com margens de lucro cada vez mais estreitas.

As placas de rede mais baratas chegam a ser vendidas no atacado por menos de 3 dólares. O preço final é um pouco mais alto, naturalmente, mas não é difícil achar placas por 20 reais ou até menos, sem falar que, hoje em dia, praticamente todas as placas-mãe vendidas possuem rede onboard, muitas vezes duas, já pensando no público que precisa compartilhar a conexão.

No começo da década de 90, existiam três padrões de rede, as redes Arcnet, Token Ring e Ethernet. As redes Arcnet tinham problemas de desempenho e as Token Ring eram muito caras, o que fez com que as redes **Ethernet** se tornassem o padrão definitivo. Hoje em dia, "Ethernet" é quase um sinônimo de placa de rede. Até mesmo as placas wireless são placas Ethernet.

Lembre-se que Ethernet é o nome de um padrão que diz como os dados são transmitidos. Todas as placas que seguem este padrão são chamadas de placas Ethernet. Não estamos falando de uma marca ou de um fabricante específico.

Temos aqui alguns exemplos de placas de rede. O conector para o cabo é chamado de "RJ45" e o soquete vago permite instalar um chip de boot. Veremos a função destes chips com mais detalhes ao estudar sobre redes de terminais leves.

Existem basicamente 3 tipos diferentes de cabos de rede: os cabos de par trançado (que são, de longe, os mais comuns), os cabos de fibra óptica (usados principalmente em links de longa distância) e os cabos coaxiais, ainda usados em algumas redes antigas.

Existem vários motivos para os cabos coaxiais não serem mais usados hoje em dia: eles são mais propensos a mal contato, os conectores são mais caros e os cabos são menos flexíveis que os de par trançado, o que torna mais difícil passá-los por dentro de tubulações. No

entanto, o principal motivo é o fato de que eles podem ser usados apenas em redes de 10 megabits: a partir do momento em que as redes 10/100 tornaram-se populares, eles entraram definitivamente em desuso, dando lugar aos cabos de par trançado. Entre eles, os que realmente usamos no dia-a-dia são os cabos "cat 5" ou "cat 5e", onde o "cat" é abreviação de "categoria" e o número indica a qualidade do cabo.

Fabricar cabos de rede é mais complicado do que parece. Diferente dos cabos de cobre comuns, usados em instalações elétricas, os cabos de rede precisam suportar freqüências muito altas, causando um mínimo de atenuação do sinal. Para isso, é preciso minimizar ao máximo o aparecimento de bolhas e impurezas durante a fabricação dos cabos. No caso dos cabos de par trançado, é preciso, ainda, cuidar do entrançamento dos pares de cabos, que também é um fator crítico.

Existem cabos de cat 1 até cat 7. Como os cabos cat 5 são suficientes tanto para redes de 100 quanto de 1000 megabits, eles são os mais comuns e mais baratos; geralmente custam em torno de 1 real o metro. Os cabos cat5e (os mais comuns atualmente) seguem um padrão um pouco mais estrito, por isso dê preferência a eles na hora de comprar.

Em todas as categorias, a distância máxima permitida é de 100 metros. O que muda é a freqüência (e, consequentemente, a taxa máxima de transferência de dados suportada pelo cabo) e o nível de imunidade a interferências externas. Esta é uma descrição de todas as categorias de cabos de par trançado existentes:

Categoria 1: Utilizado em instalações telefônicas, porém inadequado para transmissão de dados.

Categoria 2: Outro tipo de cabo obsoleto. Permite transmissão de dados a até 2.5 megabits e era usado nas antigas redes Arcnet.

Categoria 3: Era o cabo de par trançado sem blindagem mais usado em redes há uma década. Pode se estender por até 100 metros e permite transmissão de dados a até 10 Mbps. A principal diferença do cabo de categoria 3 para os obsoletos cabos de categoria 1 e 2 é o entrançamento dos pares de cabos. Enquanto nos cabos 1 e 2 não existe um padrão definido, os cabos de categoria 3 (assim como os de categoria 4 e 5) possuem pelo menos 24 tranças por metro e, por isso, são muito mais resistentes a ruídos externos. Cada par de cabos tem um número diferente de tranças por metro, o que atenua as interferências entre os pares de cabos. Praticamente não existe a possibilidade de dois pares de cabos terem exatamente a mesma disposição de tranças.

Categoria 4: Cabos com uma qualidade um pouco melhor que os cabos de categoria 3. Este tipo de cabo foi muito usado em redes Token Ring de 16 megabits. Em teoria podem ser usados também em redes Ethernet de 100 megabits, mas na prática isso é incomum, simplesmente porque estes cabos não são mais fabricados.

Categoria 5: A grande vantagem desta categoria de cabo sobre as anteriores é a taxa de transferência: eles podem ser usados tanto em redes de 100 megabits, quanto em redes de 1 gigabit.

Categoria 5e: Os cabos de categoria 5e são os mais comuns atualmente, com uma qualidade um pouco superior aos cat 5. Eles oferecem uma taxa de atenuação de sinal mais baixa, o que ajuda em cabos mais longos, perto dos 100 metros permitidos. Estão disponíveis tanto cabos blindados, quantos cabos sem blindagem, os mais baratos e comuns.

Além destes, temos os cabos de categoria 6 e 7, que ainda estão em fase de popularização:

Categoria 6: Utiliza cabos de 4 pares, semelhantes aos cabos de categoria 5 e 5e. Este padrão não está completamente estabelecido, mas o objetivo é usá-lo (assim como os 5e) nas redes Gigabit Ethernet. Já é possível encontrar cabos deste padrão à venda em algumas lojas. Você pode ler um FAQ sobre as características técnicas dos cabos cat 6 no: <http://www.tiaonline.org/standards/category6/faq.cfm>

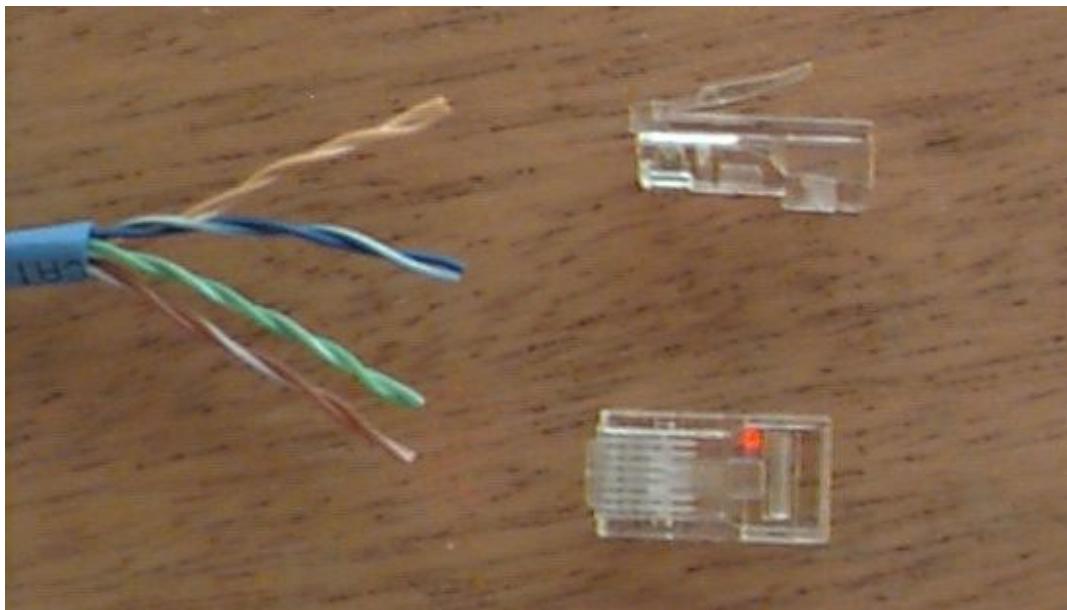
Categoria 7: Os cabos cat 7 também utilizam 4 pares de fios, porém usam conectores mais sofisticados e são muito mais caros. Tanto a freqüência máxima suportada, quanto a atenuação de sinal são melhores do que nos cabos categoria 6. Está em desenvolvimento um padrão de 10 Gigabit Ethernet que utilizará cabos de categoria 6 e 7.

Em caso de dúvida, basta checar as inscrições decalcadas no cabo. Entre elas está a categoria do cabo, como na foto.



Você pode comprar alguns metros de cabo e alguns conectores e crimpar os cabos você mesmo, ou pode comprá-los já prontos. Em ambos os casos, os cabos devem ter no mínimo 30 centímetros e no máximo 100 metros, a distância máxima que o sinal elétrico percorre antes que comece a haver uma degradação que comprometa a comunicação.

Naturalmente, os 100 metros não são um número exato. A distância máxima que é possível atingir varia de acordo com a qualidade dos cabos e conectores e as interferências presentes no ambiente. Já vi casos de cabos de 180 metros que funcionavam perfeitamente, e casos de cabos de 150 que não. Ao trabalhar fora do padrão, os resultados variam muito de acordo com as placas de rede usadas e outros fatores. Ao invés de jogar com a sorte, é mais recomendável seguir o padrão, usando um hub/switch ou um repetidor a cada 100 metros, de forma a reforçar o sinal.



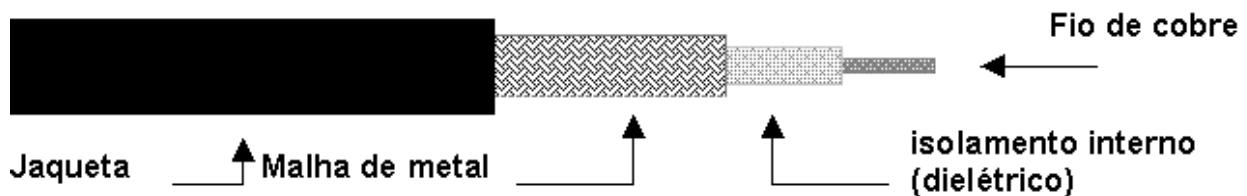
Comprar os cabos já prontos é muito mais prático, mas tem alguns inconvenientes. Muita gente (a maioria, acredito :) não acha muito legal ver cabos espalhados pelo chão da sala. Alguns desavisados chegam a tropeçar neles, derrubando micros, quebrando os conectores das placas de rede, entre outros acidentes desagradáveis.

Para dar um acabamento mais profissional, você precisa passar os cabos por dentro das tubulações das paredes ou pelo teto e é mais fácil passar o cabo primeiro e crimpar o conector depois do que tentar fazer o contrário. Se preferir crimpar o cabo você mesmo, vai precisar comprar também um alicate de crimpagem. Ele "esmaga" os contatos do conector, fazendo com que eles entrem em contato com os fios do cabo de rede.



Os cabos de rede transmitem sinais elétricos a uma freqüência muito alta e a distâncias relativamente grandes, por isso são muito vulneráveis a interferências eletromagnéticas externas.

Nos cabos coaxiais (tanto os de rede quanto os usados em antenas de TV) é usada uma malha de metal que protege o cabo de dados contra interferências externas. Os cabos de par trançado, por sua vez, usam um tipo de proteção mais sutil: o entrelaçamento dos cabos cria um campo eletromagnético que oferece uma razoável proteção contra interferências externas. Cada um dos quatro pares segue um padrão diferente de entrancamento, o que faz com que as transmissões de um não interfiram com as dos vizinhos.

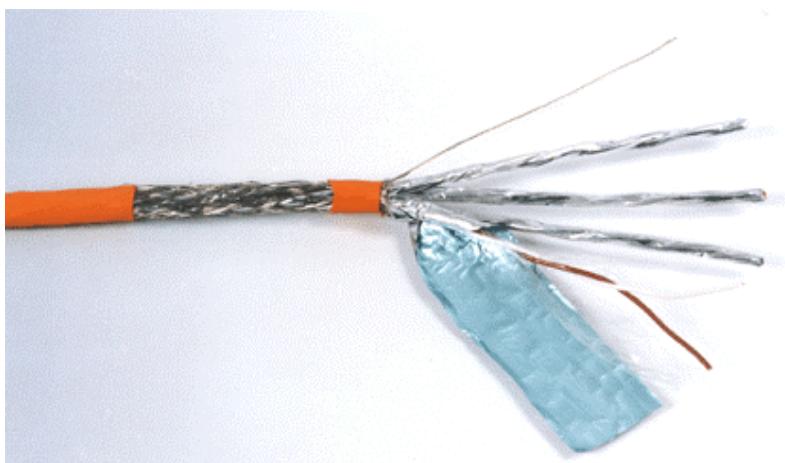


(esquema do antigo cabo de rede coaxial)

Além dos cabos sem blindagem, conhecidos como **UTP** (Unshielded Twisted Pair), existem os cabos blindados conhecidos como **STP** (Shielded Twisted Pair). A única diferença entre eles é que os cabos blindados, além de contarem com a proteção do entrelaçamento dos fios, possuem uma blindagem externa (assim como os cabos coaxiais) e

por isso são mais adequados a ambientes com fortes fontes de interferências, como grandes motores elétricos ou grandes antenas de transmissão muito próximas.

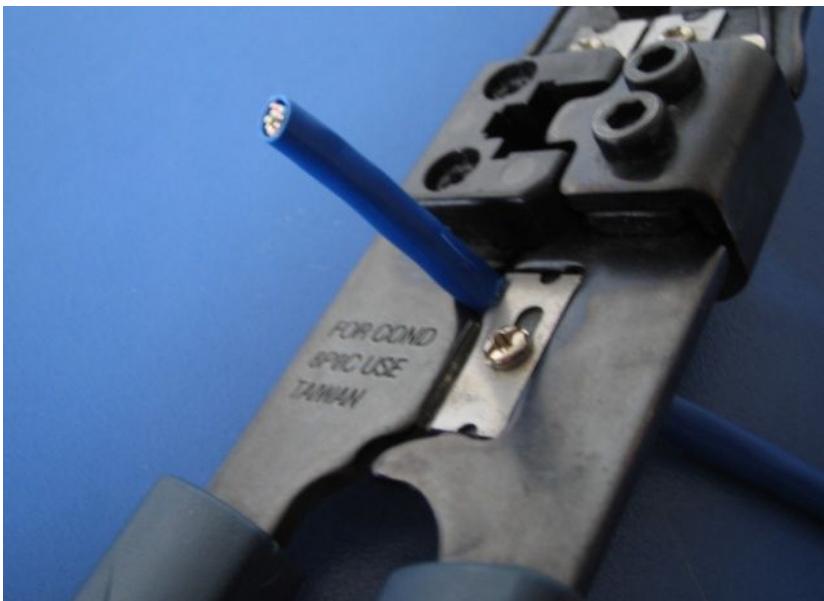
Quanto maior for o nível de interferência, menor será o desempenho da rede, menor será a distância que poderá ser usada entre os micros e mais vantajosa será a instalação de cabos blindados. Em ambientes normais, porém, os cabos sem blindagem funcionam perfeitamente bem. Na ilustração temos um exemplo de cabo com blindagem, com proteção individual para cada par de cabos. Existem também cabos mais "populares", que utilizam apenas uma blindagem externa que envolve todos os cabos.



Outras fontes menores de interferências são as lâmpadas fluorescentes (principalmente lâmpadas cansadas, que ficam piscando), cabos elétricos, quando colocados lado a lado com os cabos de rede, e mesmo telefones celulares muito próximos dos cabos. Este tipo de interferência não chega a interromper o funcionamento da rede, mas pode causar perda de pacotes.

No final de cada pacote TCP são incluídos 32 bits de CRC, que permitem verificar a integridade dos dados. Ao receber cada pacote, a estação verifica se a soma dos bits "bate" com o valor do CRC. Sempre que a soma der errado, ela solicita a retransmissão do pacote, o que é repetido indefinidamente, até que ela receba uma cópia intacta. Graças a esse sistema é possível transmitir dados de forma confiável mesmo através de links ruins (como, por exemplo, uma conexão via modem). Porém, quanto mais intensas forem as interferências, mais pacotes precisam ser retransmitidos e pior é o desempenho da rede.

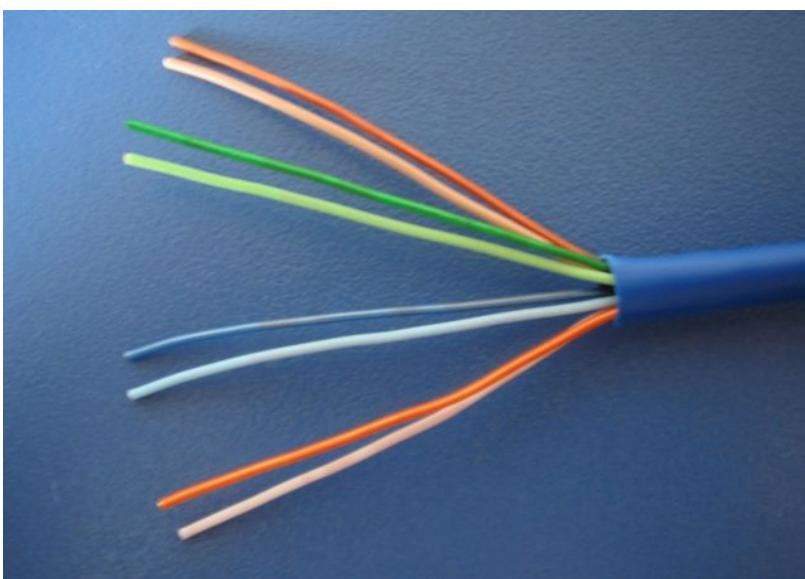
Ao crimpar os cabos de rede, o primeiro passo é descascar os cabos, tomando cuidado para não ferir os fios internos, que são frágeis. Normalmente, o alicate inclui uma saliência no canto da guilhotina, que serve bem para isso. Existem também descascadores de cabos específicos para cabos de rede.



Os quatro pares do cabo são diferenciados por cores. Um par é laranja, outro é azul, outro é verde e o último é marrom. Um dos cabos de cada par tem uma cor sólida e o outro é mais claro ou malhado, misturando a cor e pontos de branco. É pelas cores que diferenciamos os 8 fios.

O segundo passo é destrançar os cabos, deixando-os soltos. É preciso organizá-los em uma certa ordem para colocá-los dentro do conector e é meio complicado fazer isso se eles estiverem grudados entre si :-P.

Eu prefiro descascar um pedaço grande do cabo, uns 6 centímetros, para poder organizar os cabos com mais facilidade e depois cortar o excesso, deixando apenas os 2 centímetros que entrarão dentro do conector. O próprio alicate de crimpagem inclui uma guilhotina para cortar os cabos, mas você pode usar uma tesoura se preferir.



No padrão EIA 568B, a ordem dos fios dentro do conector é a seguinte:

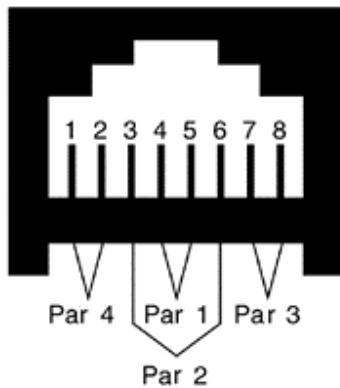
1-	Branco	com	Laranja
2-			Laranja
3-	Branco	com	Verde
4-			Azul
5-	Branco	com	Azul
6-			Verde
7-	Branco	com	Marrom
8- Marrom			

Os cabos são encaixados nesta ordem, com a trava do conector virada para baixo, como no diagrama.



Ou seja, se você olhar o conector "de cima", vendo a trava, o par de fios laranja estará à direita e, se olhar o conector "de baixo", vendo os contatos, eles estarão à esquerda.

Este outro diagrama mostra melhor como fica a posição dos cabos dentro do conector:



No caso de um cabo "reto" (straight), que vai ser usado para ligar o micro ao hub, você usa esta mesma disposição nas duas pontas do cabo. Existe ainda um outro tipo de cabo, chamado de "**cross-over**", que permite ligar diretamente dois micros, sem precisar do hub. Ele é uma opção mais barata quando você tem apenas dois micros. Neste tipo de cabo a posição dos fios é diferente nos dois conectores, de um dos lados a pinagem é a mesma de um cabo de rede normal, enquanto no outro a posição dos pares verde e laranja são trocados. Daí vem o nome cross-over, que significa, literalmente, "cruzado na ponta".

Para fazer um cabo cross-over, você crimpa uma das pontas seguindo o padrão que vimos acima e a outra com este segundo padrão (EIA 568A). Note que trocamos a posição dos pares verde e laranja:

1-	Branco	com	Verde
2-			Verde
3-	Branco	com	Laranja
4-			Azul
5-	Branco	com	Azul
6-			Laranja
7-	Branco	com	Marrom
8- Marrom			

Na hora de crimpas é preciso fazer um pouco de força para que o conector fique firme. A qualidade do alicate é importante: evite comprar alicates muito baratos, pois eles precisam ser resistentes para aplicar a pressão necessária.

A função do alicate é fornecer pressão suficiente para que os pinos do conector RJ-45, que internamente possuem a forma de lâminas, esmaguem os fios do cabo, alcançando o fio de cobre e criando o contato. Você deve retirar apenas a capa externa do cabo e não descascar individualmente os fios, pois isso, ao invés de ajudar, serviria apenas para causar mau contato, deixando frouxo o encaixe com os pinos do conector.



É preciso um pouco de atenção ao cortar e encaixar os fios dentro do conector, pois eles precisam ficar perfeitamente retos. Isso demanda um pouco de prática. No começo, você vai sempre errar algumas vezes antes de conseguir.

Veja que o que protege os cabos contra as interferências externas são justamente as tranças. A parte destrançada que entra no conector é o ponto fraco do cabo, onde ele é mais vulnerável a todo tipo de interferência. Por isso, é recomendável deixar um espaço menor possível sem as tranças. Para crimpas cabos dentro do padrão, você precisa deixar menos de 2,5 centímetros destrançados. Você só vai conseguir isso cortando o excesso de cabo solto antes de encaixar o conector, como na foto:



O primeiro teste para ver se os cabos foram crimpados corretamente é conectar um dos micros (ligado) ao hub e ver se os LEDs da placas de rede e do hub acendem. Isso mostra que os sinais elétricos enviados estão chegando até o hub e que ele foi capaz de abrir um canal de comunicação com a placa.

Se os LEDs nem acenderem, então não existe o que fazer. Corte os conectores e tente de novo. Infelizmente, os conectores são descartáveis: depois de crimpar errado uma vez, você precisa usar outro novo, aproveitando apenas o cabo. Mais um motivo para prestar atenção ;).

Existem também aparelhos testadores de cabos, que oferecem um diagnóstico muito mais sofisticado, dizendo, por exemplo, se os cabos são adequados para transmissões a 100 ou a 1000 megabits e avisando caso algum dos 8 fios do cabo esteja rompido. Os mais sofisticados avisam inclusive em que ponto o cabo está rompido, permitindo que você aproveite a parte boa.



Esses aparelhos serão bastante úteis se você for crimpar muitos cabos, mas são dispensáveis para trabalhos esporádicos, pois é muito raro que os cabos venham com fios rompidos de fábrica. Os cabos de rede apresentam também uma boa resistência mecânica e flexibilidade, para que possam passar por dentro de tubulações. Quase sempre os problemas de transmissão surgem por causa de conectores mal crimpados.

Uma curiosidade é que algumas placas mãe da Asus, com rede Yukon Marvel (e, eventualmente, outros modelos lançados futuramente), incluem um software testador de cabos, que pode ser acessado pelo setup, ou através de uma interface dentro do Windows.

Ele funciona de uma forma bastante engenhosa. Quando o cabo está partido em algum ponto, o sinal elétrico percorre o cabo até o ponto onde ele está rompido e, por não ter para onde ir, retorna na forma de interferência. O software cronometra o tempo que o sinal demora para ir e voltar, apontando com uma certa precisão depois de quantos metros o cabo está rompido.



Existem três padrões de redes Ethernet (com fio): de **10** megabits, **100** megabits e **1000** megabits (também chamadas de Gigabit Ethernet). Já estão disponíveis também as redes de 10 gigabits, mas por enquanto elas ainda são muito caras, pois utilizam placas específicas e cabos de fibra óptica. Esses três padrões são intercompatíveis: você pode perfeitamente misturar placas de 100 e 1000 megabits na mesma rede, mas, ao usar placas de velocidades diferentes, a velocidade é sempre nivelada por baixo, ou seja, as placas Gigabit são obrigadas a respeitar a velocidade das placas mais lentas.

As redes de 10 megabits estão em desuso já a vários anos e tendem a se extinguir com o tempo. As de 100 megabits são o padrão (por enquanto), pois são muito baratas e propiciam uma velocidade suficiente para transmitir grandes arquivos e rodar aplicativos remotamente.

Tudo o que a placa de rede faz é transmitir os uns e zeros enviados pelo processador através do cabo de rede, de forma que a transmissão seja recebida pelos outros micros. Ao transferir um arquivo, o processador lê o arquivo gravado no HD e o envia à placa de rede para ser transmitido.

Os HDs atuais são capazes de ler dados a 30 ou 40 MB por segundo. Lembre-se que um byte tem 8 bits, logo 30 MB (megabytes, com o B maiúsculo) correspondem a 240 megabits (Mb, com o b minúsculo) e assim por diante. Se você dividir 100 megabits por 8, terá 12.5 megabytes por segundo. É bem menos do que um HD atual é capaz, mas já é uma velocidade razoável. No que depender da rede, demora cerca de um minuto para copiar um CD inteiro, por exemplo.

A opção para quem precisa de mais velocidade são as redes Gigabit Ethernet, que transmitem a uma taxa de até 1000 megabits (125 megabytes) por segundo. As placas Gigabit atuais são compatíveis com os mesmos cabos de par trançado cat 5 usados pelas placas de 100 megabits (veja mais detalhes a seguir), por isso a diferença de custo fica por

conta apenas das placas e do switch. Graças a isso elas estão caindo de preço e se popularizando rapidamente.

Existem basicamente 3 tipos diferentes de cabos de rede: os cabos de par trançado (que são, de longe, os mais comuns), os cabos de fibra óptica (usados principalmente em links de longa distância) e os cabos coaxiais, ainda usados em algumas redes antigas.

Existem vários motivos para os cabos coaxiais não serem mais usados hoje em dia: eles são mais propensos a mal contato, os conectores são mais caros e os cabos são menos flexíveis que os de par trançado, o que torna mais difícil passá-los por dentro de tubulações. No entanto, o principal motivo é o fato de que eles podem ser usados apenas em redes de 10 megabits: a partir do momento em que as redes 10/100 tornaram-se populares, eles entraram definitivamente em desuso, dando lugar aos cabos de par trançado. Entre eles, os que realmente usamos no dia-a-dia são os cabos "cat 5" ou "cat 5e", onde o "cat" é abreviação de "categoria" e o número indica a qualidade do cabo.

Fabricar cabos de rede é mais complicado do que parece. Diferente dos cabos de cobre comuns, usados em instalações elétricas, os cabos de rede precisam suportar freqüências muito altas, causando um mínimo de atenuação do sinal. Para isso, é preciso minimizar ao máximo o aparecimento de bolhas e impurezas durante a fabricação dos cabos. No caso dos cabos de par trançado, é preciso, ainda, cuidar do entrancamento dos pares de cabos, que também é um fator crítico.

Existem cabos de cat 1 até cat 7. Como os cabos cat 5 são suficientes tanto para redes de 100 quanto de 1000 megabits, eles são os mais comuns e mais baratos; geralmente custam em torno de 1 real o metro. Os cabos cat5e (os mais comuns atualmente) seguem um padrão um pouco mais estrito, por isso têm preferência a eles na hora de comprar.

Em todas as categorias, a distância máxima permitida é de 100 metros. O que muda é a freqüência (e, consequentemente, a taxa máxima de transferência de dados suportada pelo cabo) e o nível de imunidade a interferências externas. Esta é uma descrição de todas as categorias de cabos de par trançado existentes:

Categoria 1: Utilizado em instalações telefônicas, porém inadequado para transmissão de dados.

Categoria 2: Outro tipo de cabo obsoleto. Permite transmissão de dados a até 2.5 megabits e era usado nas antigas redes Arcnet.

Categoria 3: Era o cabo de par trançado sem blindagem mais usado em redes há uma década. Pode se estender por até 100 metros e permite transmissão de dados a até 10 Mbps. A principal diferença do cabo de categoria 3 para os obsoletos cabos de categoria 1 e 2 é o entrancamento dos pares de cabos. Enquanto nos cabos 1 e 2 não existe um padrão definido, os cabos de categoria 3 (assim como os de categoria 4 e 5) possuem pelo menos 24 tranças por metro e, por isso, são muito mais resistentes a ruídos externos. Cada par de cabos tem um número diferente de tranças por metro, o que atenua as interferências entre os pares

de cabos. Praticamente não existe a possibilidade de dois pares de cabos terem exatamente a mesma disposição de tranças.

Categoria 4: Cabos com uma qualidade um pouco melhor que os cabos de categoria 3. Este tipo de cabo foi muito usado em redes Token Ring de 16 megabits. Em teoria podem ser usados também em redes Ethernet de 100 megabits, mas na prática isso é incomum, simplesmente porque estes cabos não são mais fabricados.

Categoria 5: A grande vantagem desta categoria de cabo sobre as anteriores é a taxa de transferência: eles podem ser usados tanto em redes de 100 megabits, quanto em redes de 1 gigabit.

Categoria 5e: Os cabos de categoria 5e são os mais comuns atualmente, com uma qualidade um pouco superior aos cat 5. Eles oferecem uma taxa de atenuação de sinal mais baixa, o que ajuda em cabos mais longos, perto dos 100 metros permitidos. Estão disponíveis tanto cabos blindados, quantos cabos sem blindagem, os mais baratos e comuns.

Além destes, temos os cabos de categoria 6 e 7, que ainda estão em fase de popularização:

Categoria 6: Utiliza cabos de 4 pares, semelhantes aos cabos de categoria 5 e 5e. Este padrão não está completamente estabelecido, mas o objetivo é usá-lo (assim como os 5e) nas redes Gigabit Ethernet. Já é possível encontrar cabos deste padrão à venda em algumas lojas. Você pode ler um FAQ sobre as características técnicas dos cabos cat 6 no: <http://www.tiaonline.org/standards/category6/faq.cfm>

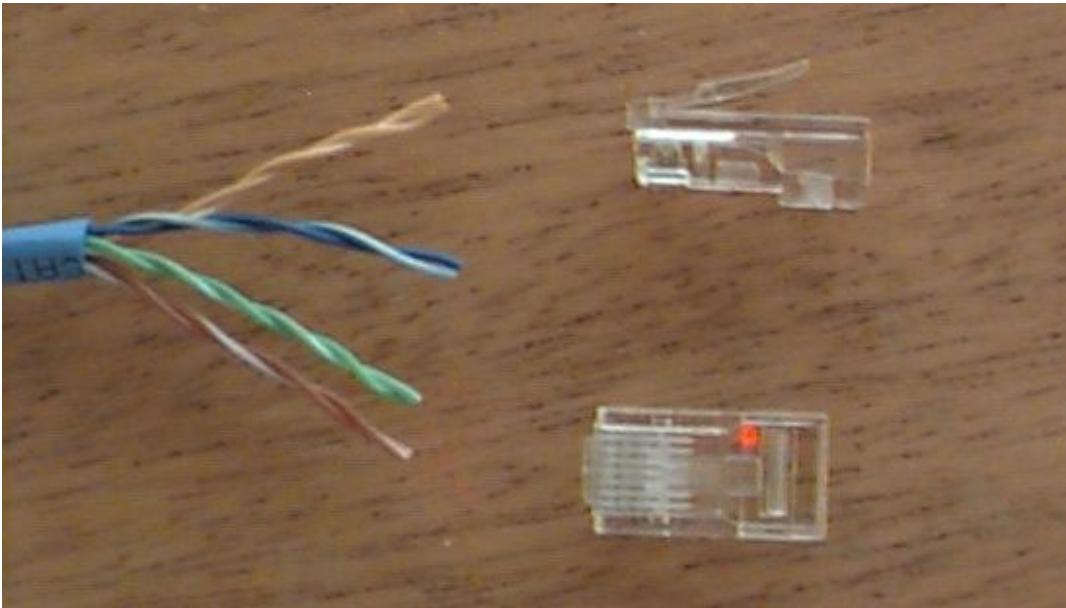
Categoria 7: Os cabos cat 7 também utilizam 4 pares de fios, porém usam conectores mais sofisticados e são muito mais caros. Tanto a freqüência máxima suportada, quanto a atenuação de sinal são melhores do que nos cabos categoria 6. Está em desenvolvimento um padrão de 10 Gigabit Ethernet que utilizará cabos de categoria 6 e 7.

Em caso de dúvida, basta checar as inscrições decalcadas no cabo. Entre elas está a categoria do cabo, como na foto.



Você pode comprar alguns metros de cabo e alguns conectores e crimpá-los você mesmo, ou pode comprá-los já prontos. Em ambos os casos, os cabos devem ter no mínimo 30 centímetros e no máximo 100 metros, a distância máxima que o sinal elétrico percorre antes que comece a haver uma degradação que comprometa a comunicação.

Naturalmente, os 100 metros não são um número exato. A distância máxima que é possível atingir varia de acordo com a qualidade dos cabos e conectores e as interferências presentes no ambiente. Já vi casos de cabos de 180 metros que funcionavam perfeitamente, e casos de cabos de 150 que não. Ao trabalhar fora do padrão, os resultados variam muito de acordo com as placas de rede usadas e outros fatores. Ao invés de jogar com a sorte, é mais recomendável seguir o padrão, usando um hub/switch ou um repetidor a cada 100 metros, de forma a reforçar o sinal.



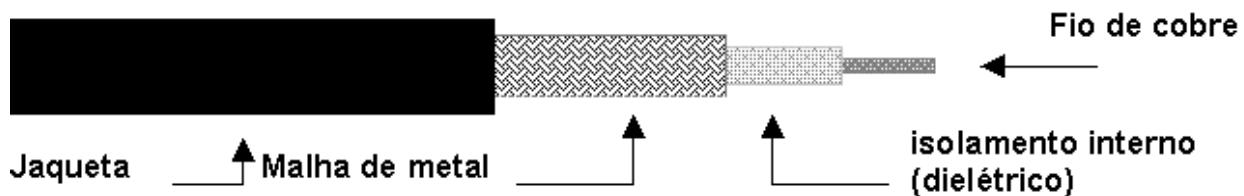
Comprar os cabos já prontos é muito mais prático, mas tem alguns inconvenientes. Muita gente (a maioria, acredito :) não acha muito legal ver cabos espalhados pelo chão da sala. Alguns desavisados chegam a tropeçar neles, derrubando micros, quebrando os conectores das placas de rede, entre outros acidentes desagradáveis.

Para dar um acabamento mais profissional, você precisa passar os cabos por dentro das tubulações das paredes ou pelo teto e é mais fácil passar o cabo primeiro e crimpar o conector depois do que tentar fazer o contrário. Se preferir crimpar o cabo você mesmo, vai precisar comprar também um alicate de crimpagem. Ele "esmagá" os contatos do conector, fazendo com que eles entrem em contato com os fios do cabo de rede.



Os cabos de rede transmitem sinais elétricos a uma freqüência muito alta e a distâncias relativamente grandes, por isso são muito vulneráveis a interferências eletromagnéticas externas.

Nos cabos coaxiais (tanto os de rede quanto os usados em antenas de TV) é usada uma malha de metal que protege o cabo de dados contra interferências externas. Os cabos de par trançado, por sua vez, usam um tipo de proteção mais sutil: o entrelaçamento dos cabos cria um campo eletromagnético que oferece uma razoável proteção contra interferências externas. Cada um dos quatro pares segue um padrão diferente de entrancamento, o que faz com que as transmissões de um não interfiram com as dos vizinhos.

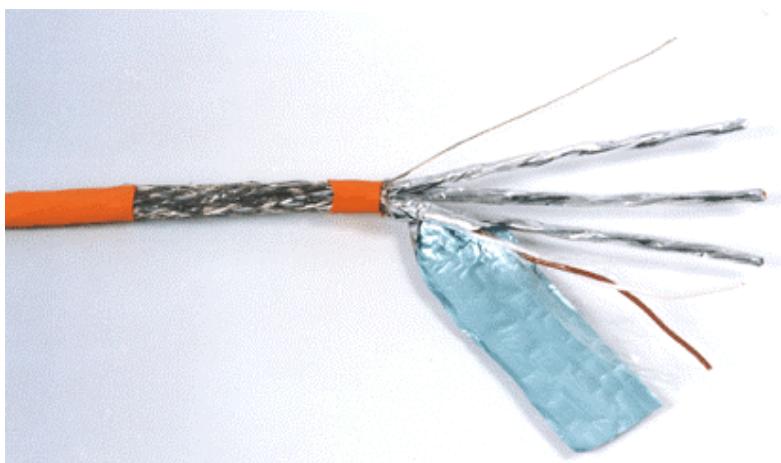


(esquema do antigo cabo de rede coaxial)

Além dos cabos sem blindagem, conhecidos como **UTP** (Unshielded Twisted Pair), existem os cabos blindados conhecidos como **STP** (Shielded Twisted Pair). A única diferença entre eles é que os cabos blindados, além de contarem com a proteção do entrelaçamento dos fios, possuem uma blindagem externa (assim como os cabos coaxiais) e

por isso são mais adequados a ambientes com fortes fontes de interferências, como grandes motores elétricos ou grandes antenas de transmissão muito próximas.

Quanto maior for o nível de interferência, menor será o desempenho da rede, menor será a distância que poderá ser usada entre os micros e mais vantajosa será a instalação de cabos blindados. Em ambientes normais, porém, os cabos sem blindagem funcionam perfeitamente bem. Na ilustração temos um exemplo de cabo com blindagem, com proteção individual para cada par de cabos. Existem também cabos mais "populares", que utilizam apenas uma blindagem externa que envolve todos os cabos.



Outras fontes menores de interferências são as lâmpadas fluorescentes (principalmente lâmpadas cansadas, que ficam piscando), cabos elétricos, quando colocados lado a lado com os cabos de rede, e mesmo telefones celulares muito próximos dos cabos. Este tipo de interferência não chega a interromper o funcionamento da rede, mas pode causar perda de pacotes.

No final de cada pacote TCP são incluídos 32 bits de CRC, que permitem verificar a integridade dos dados. Ao receber cada pacote, a estação verifica se a soma dos bits "bate" com o valor do CRC. Sempre que a soma der errado, ela solicita a retransmissão do pacote, o que é repetido indefinidamente, até que ela receba uma cópia intacta. Graças a esse sistema é possível transmitir dados de forma confiável mesmo através de links ruins (como, por exemplo, uma conexão via modem). Porém, quanto mais intensas forem as interferências, mais pacotes precisam ser retransmitidos e pior é o desempenho da rede.

» Próximo: [Crimpando os cabos](#)

Ao crimpar os cabos de rede, o primeiro passo é descascar os cabos, tomando cuidado para não ferir os fios internos, que são frágeis. Normalmente, o alicate inclui uma saliência no

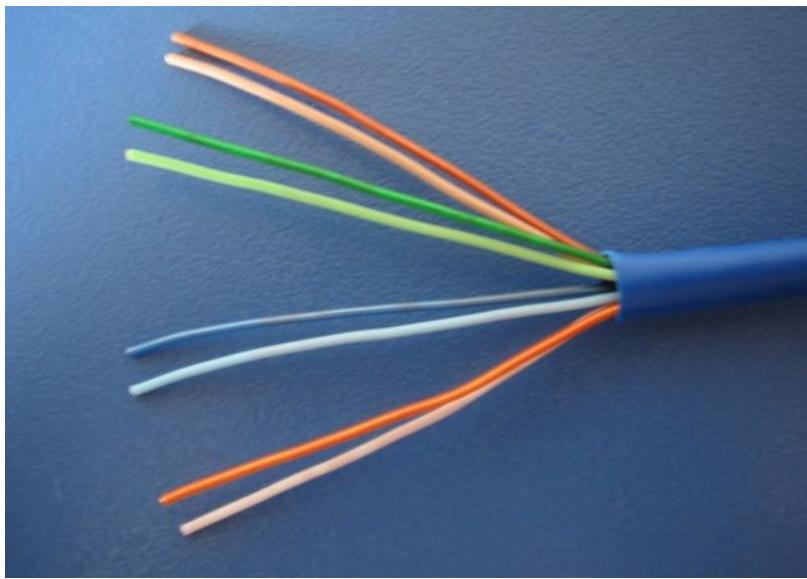
canto da guilhotina, que serve bem para isso. Existem também descascadores de cabos específicos para cabos de rede.



Os quatro pares do cabo são diferenciados por cores. Um par é laranja, outro é azul, outro é verde e o último é marrom. Um dos cabos de cada par tem uma cor sólida e o outro é mais claro ou malhado, misturando a cor e pontos de branco. É pelas cores que diferenciamos os 8 fios.

O segundo passo é destrançar os cabos, deixando-os soltos. É preciso organizá-los em uma certa ordem para colocá-los dentro do conector e é meio complicado fazer isso se eles estiverem grudados entre si :-P.

Eu prefiro descascar um pedaço grande do cabo, uns 6 centímetros, para poder organizar os cabos com mais facilidade e depois cortar o excesso, deixando apenas os 2 centímetros que entrarão dentro do conector. O próprio alicate de crimpagem inclui uma guilhotina para cortar os cabos, mas você pode usar uma tesoura se preferir.



No padrão EIA 568B, a ordem dos fios dentro do conector é a seguinte:

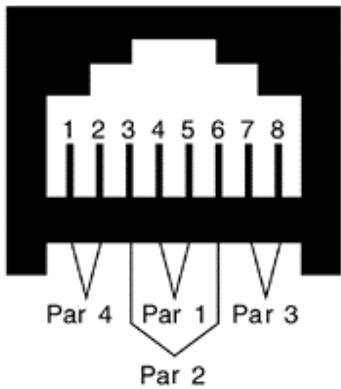
1-	Branco	com	Laranja
2-			Laranja
3-	Branco	com	Verde
4-			Azul
5-	Branco	com	Azul
6-			Verde
7-	Branco	com	Marrom
8- Marrom			

Os cabos são encaixados nesta ordem, com a trava do conector virada para baixo, como no diagrama.



Ou seja, se você olhar o conector "de cima", vendo a trava, o par de fios laranja estará à direita e, se olhar o conector "de baixo", vendo os contatos, eles estarão à esquerda.

Este outro diagrama mostra melhor como fica a posição dos cabos dentro do conector:



No caso de um cabo "reto" (straight), que vai ser usado para ligar o micro ao hub, você usa esta mesma disposição nas duas pontas do cabo. Existe ainda um outro tipo de cabo, chamado de "**cross-over**", que permite ligar diretamente dois micros, sem precisar do hub. Ele é uma opção mais barata quando você tem apenas dois micros. Neste tipo de cabo a posição dos fios é diferente nos dois conectores, de um dos lados a pinagem é a mesma de um cabo de rede normal, enquanto no outro a posição dos pares verde e laranja são trocados. Daí vem o nome cross-over, que significa, literalmente, "cruzado na ponta".

Para fazer um cabo cross-over, você crimpa uma das pontas seguindo o padrão que vimos acima e a outra com este segundo padrão (EIA 568A). Note que trocamos a posição dos pares verde e laranja:

1-	Branco	com	Verde
2-			Verde
3-	Branco	com	Laranja
4-			Azul
5-	Branco	com	Azul
6-			Laranja
7-	Branco	com	Marrom
8- Marrom			

Na hora de crimpar é preciso fazer um pouco de força para que o conector fique firme. A qualidade do alicate é importante: evite comprar alicates muito baratos, pois eles precisam ser resistentes para aplicar a pressão necessária.

A função do alicate é fornecer pressão suficiente para que os pinos do conector RJ-45, que internamente possuem a forma de lâminas, esmaguem os fios do cabo, alcançando o fio de cobre e criando o contato. Você deve retirar apenas a capa externa do cabo e não descascar individualmente os fios, pois isso, ao invés de ajudar, serviria apenas para causar mau contato, deixando frrouxo o encaixe com os pinos do conector.



É preciso um pouco de atenção ao cortar e encaixar os fios dentro do conector, pois eles precisam ficar perfeitamente retos. Isso demanda um pouco de prática. No começo, você vai sempre errar algumas vezes antes de conseguir.

Veja que o que protege os cabos contra as interferências externas são justamente as tranças. A parte destrançada que entra no conector é o ponto fraco do cabo, onde ele é mais vulnerável a todo tipo de interferência. Por isso, é recomendável deixar um espaço menor possível sem as tranças. Para crimpar cabos dentro do padrão, você precisa deixar menos de 2,5 centímetros destrançados. Você só vai conseguir isso cortando o excesso de cabo solto antes de encaixar o conector, como na foto:



O primeiro teste para ver se os cabos foram crimpados corretamente é conectar um dos micros (ligado) ao hub e ver se os LEDs da placas de rede e do hub acendem. Isso mostra

que os sinais elétricos enviados estão chegando até o hub e que ele foi capaz de abrir um canal de comunicação com a placa.

Se os LEDs nem acenderem, então não existe o que fazer. Corte os conectores e tente de novo. Infelizmente, os conectores são descartáveis: depois de crimpado errado uma vez, você precisa usar outro novo, aproveitando apenas o cabo. Mais um motivo para prestar atenção ;).

Existem também aparelhos testadores de cabos, que oferecem um diagnóstico muito mais sofisticado, dizendo, por exemplo, se os cabos são adequados para transmissões a 100 ou a 1000 megabits e avisando caso algum dos 8 fios do cabo esteja rompido. Os mais sofisticados avisam inclusive em que ponto o cabo está rompido, permitindo que você aproveite a parte boa.



Esses aparelhos serão bastante úteis se você for crimpado muitos cabos, mas são dispensáveis para trabalhos esporádicos, pois é muito raro que os cabos venham com fios rompidos de fábrica. Os cabos de rede apresentam também uma boa resistência mecânica e flexibilidade, para que possam passar por dentro de tubulações. Quase sempre os problemas de transmissão surgem por causa de conectores mal crimpados.

Uma curiosidade é que algumas placas mãe da Asus, com rede Yukon Marvel (e, eventualmente, outros modelos lançados futuramente), incluem um software testador de cabos, que pode ser acessado pelo setup, ou através de uma interface dentro do Windows.

Ele funciona de uma forma bastante engenhosa. Quando o cabo está partido em algum ponto, o sinal elétrico percorre o cabo até o ponto onde ele está rompido e, por não ter para onde ir, retorna na forma de interferência. O software cronometra o tempo que o sinal demora para ir e voltar, apontando com uma certa precisão depois de quantos metros o cabo está rompido.

» Próximo: [Hubs e Switches](#)

O **hub** ou **switch** é simplesmente o coração da rede. Ele serve como um ponto central, permitindo que todos os pontos se comuniquem entre si.

Todas as placas de rede são ligadas ao hub ou switch e é possível ligar vários hubs ou switches entre si (até um máximo de 7), caso necessário.



A diferença entre os hubs e switches é que o hub apenas retransmite tudo o que recebe para todos os micros conectados a ele, como se fosse um espelho. Isso significa que apenas um micro pode transmitir dados de cada vez e que todas as placas precisam operar na mesma velocidade, que é sempre nivelada por baixo. Caso você coloque um micro com uma placa de 10 megabits na rede, a rede toda passará a trabalhar a 10 megabits.

Os switches por sua vez são aparelhos muito mais inteligentes. Eles fecham canais exclusivos de comunicação entre o micro que está enviando dados e o que está recebendo, permitindo que vários pares de micros troquem dados entre si ao mesmo tempo. Isso melhora bastante a velocidade em redes congestionadas, com muitos micros. Outra vantagem dos switches é que, em redes onde são misturadas placas 10/10 e 10/100, as comunicações podem ser feitas na velocidade das placas envolvidas, ou seja, quando duas placas 10/100 trocarem dados, a comunicação será feita a 100 megabits e quando uma das placas de 10 megabits estiver envolvida, será feita a 10 megabits.

Hoje em dia, os hubs "burros" caíram em desuso. Quase todos à venda atualmente são "**hub-switches**", modelos de switches mais baratos, que custam quase o mesmo que um hub antigo. Depois destes, temos os switches "de verdade", capazes de gerenciar um número muito maior de portas, sendo, por isso, adequados a redes de maior porte.

Tanto os "hub-switches", quanto os switches "de verdade" são dispositivos que trabalham no nível 2 do modelo OSI. O que muda entre as duas categorias é o número de portas e recursos. Os switches "de verdade" possuem interfaces de gerenciamento, que você acessa através do navegador em um dos micros da rede, que permitem visualizar diversos detalhes

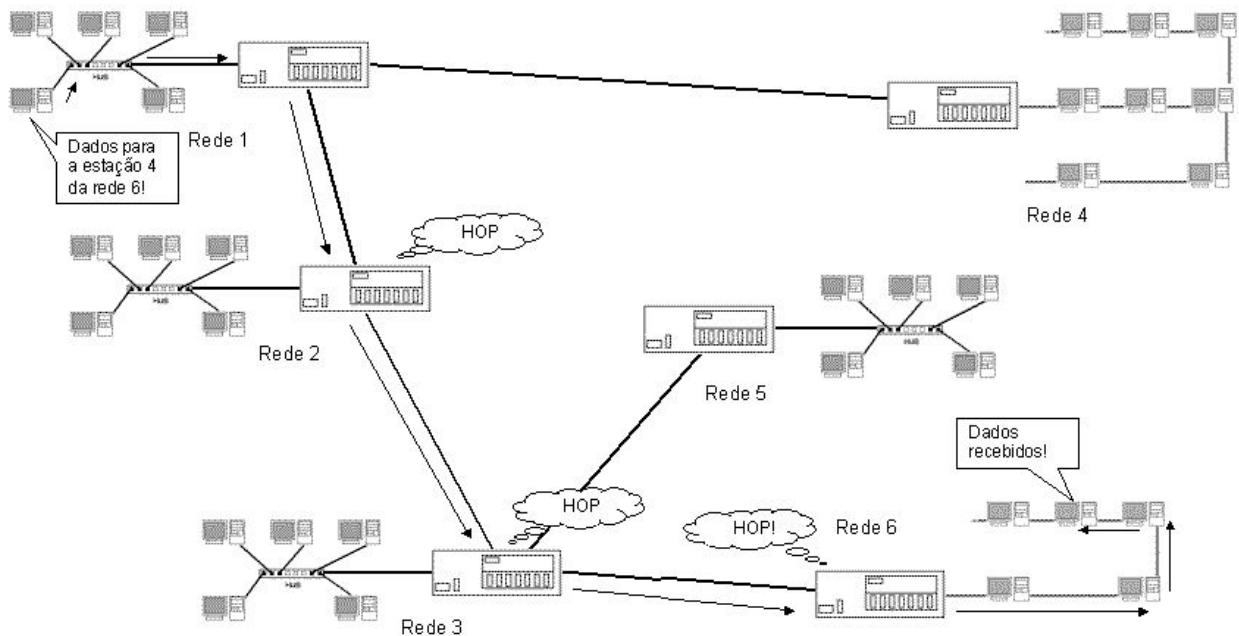
sobre o tráfego, descobrir problemas na rede e alterar diversas configurações, enquanto que os "hub-switches" são dispositivos burros.

Hoje em dia, existem ainda os "level 3 switches", uma categoria ainda mais inteligente de switches, que incorporam algumas características dos roteadores. Eles permitem definir rotas entre os diferentes micross da rede com base no endereço IP, criar "redes virtuais", onde os micross passam a se comportar como se estivessem ligados a dois switches diferentes, e assim por diante.

Finalmente, temos os **roteadores**, que são o topo da cadeia evolutiva. Os roteadores são ainda mais inteligentes, pois são capazes de interligar várias redes diferentes e sempre escolher a rota mais rápida para cada pacote de dados. Os roteadores operam no nível 3 do modelo OSI, procurando por endereços IP, ao invés de endereços MAC.

Usando roteadores, é possível interligar um número enorme de redes diferentes, mesmo que situadas em países ou mesmo continentes diferentes. Note que cada rede possui seu próprio roteador e os vários roteadores são interligados entre si. É possível interligar inúmeras redes diferentes usando roteadores, e não seria de se esperar que todos os roteadores tivessem acesso direto a todos os outros roteadores a que estivesse conectado.

Pode ser que, por exemplo, o roteador 4 esteja ligado apenas ao roteador 1, que esteja ligado ao roteador 2, que por sua vez seja ligado ao roteador 3, que esteja ligado aos roteadores 5 e 6. Se um micro da rede 1 precisar enviar dados para um dos micross da rede 6, então o pacote passará primeiro pelo roteador 2, será encaminhado ao roteador 3 e finalmente ao roteador 6. Cada vez que o dado é transmitido de um roteador para outro, temos um "**hop**".



Os roteadores são inteligentes o suficiente para determinar o melhor caminho a seguir. Inicialmente, o roteador procurará o caminho com o menor número de hops: o caminho mais curto. Mas se por acaso perceber que um dos roteadores desta rota está ocupado demais (o que pode ser medido pelo tempo de resposta), ele procurará caminhos alternativos para desviar deste roteador congestionado, mesmo que para isso o sinal tenha que passar por mais roteadores. No final, apesar do sinal ter percorrido o caminho mais longo, chegará mais rápido, pois não precisará ficar esperando na fila do roteador congestionado.

A internet é, na verdade, uma rede gigantesca, formada por várias sub-redes interligadas por roteadores. Todos os usuários de um pequeno provedor, por exemplo, podem ser conectados à internet por meio do mesmo roteador. Para baixar uma página do Yahoo, por exemplo, o sinal deverá passar por vários roteadores, várias dezenas em alguns casos. Se todos estiverem livres, a página será carregada rapidamente. Porém, se alguns estiverem congestionados, pode ser que a página demore vários segundos antes de começar a carregar.

Você pode medir o tempo que um pedido de conexão demora para ir até o destino e ser respondido usando o comando "**ping**", disponível tanto no Linux quanto no prompt do MS-DOS, no Windows. Para verificar por quantos roteadores o pacote está passando até chegar ao destino, use o comando "**traceroute**" (no Linux) ou "**tracert**" (no Windows).

Os roteadores podem ser desde PCs comuns, com duas ou mais placas de rede, até supercomputadores capazes de gerenciar centenas de links de alta velocidade. Eles formam a espinha dorsal da internet.



Quando você usa um PC com duas placas de rede para compartilhar a conexão com os micros da rede local, você está configurando-o para funcionar como um roteador simples, que liga uma rede (a internet) a outra (a sua rede doméstica). O mesmo acontece ao configurar seu modem ADSL como roteador.

Pense que a diferença entre hubs e switches e os roteadores é justamente esta: os hubs e switches permitem que vários micros sejam ligados formando uma única rede, enquanto que os roteadores permitem interligar várias redes diferentes, criando redes ainda maiores, como a própria internet.

Dentro de uma mesma rede é possível enviar pacotes de broadcast, que são endereçados a todos os integrantes da rede simultaneamente. Ao usar um hub burro, todos os micros recebem todas as transmissões. Um roteador filtra tudo isso, fazendo com que apenas os pacotes especificamente endereçados a endereços de outras redes trafeguem entre elas. Lembre-se de que, ao contrário das redes locais, os links de internet são muito caros (muitas vezes se paga por gigabyte transferido), por isso é essencial que sejam bem aproveitados.

» Próximo: [Conectando hubs](#)

A maioria dos hub-switches possui apenas 8 portas. Os modelos mais caros chegam a ter 24 portas, mas sempre existe um limite. E se este limite não for suficiente para conectar todos os micros de sua rede? Para quebrar esta limitação, existe a possibilidade de conectar dois ou mais hubs ou hub-switches entre si. Quase todos os hubs possuem uma porta chamada "up-link", que se destina justamente a esta função. Basta ligar as portas up-link de ambos os hubs, usando um cabo de rede normal (straight) para que os hubs passem a se enxergar.

Como para toda a regra existe uma exceção, alguns hubs não possuem a porta up-link. Mas nem tudo está perdido, lembra-se do cabo cross-over que serve para ligar diretamente dois micros sem usar um hub? Ele também serve para conectar dois hubs. A única diferença neste caso é que, ao invés de usar as portas up-link, usaremos duas portas comuns.

No caso dos hub-switches, você quase nunca encontrará a porta up-link, pois eles são capazes de detectar automaticamente a ligação de outro switch ou hub em qualquer uma das portas, alterando o sinal caso necessário, recurso chamado de "auto-sense". Este recurso permite usar também um cabo cross-over para ligar um determinado micro da rede, mesmo que todos os demais utilizem cabos normais. Novamente, o próprio hub-switch faz a correção do sinal internamente.



» Próximo: [Modo full-duplex](#)

O modo full-duplex permite que a placa de rede envie e receba dados simultaneamente, permitindo que você baixe um arquivo grande a partir do servidor de arquivos da rede, ao mesmo tempo em que outro micro da rede copia um arquivo a partir do seu. Ambas as transferências são feitas na velocidade máxima permitida pela rede, ou seja, 100 ou 1000 megabits.

No modo antigo, o half-duplex, é possível apenas enviar ou receber dados, uma coisa de cada vez. Como as transmissões são divididas em pacotes e são concluídas num espaço muito pequeno de tempo, também é possível enviar e receber dados "ao mesmo tempo", mas neste caso a velocidade da rede é dividida entre as duas transmissões: na melhor das hipóteses você poderia baixar um arquivo a 50 megabits e enviar outro a 50 megabits.

O full-duplex é um recurso disponível apenas nos switches ou hub-switches. Outro pré-requisito é que sejam usados cabos de rede com 4 pares. Existem cabos de rede com apenas 2 pares, que eram muito usados em redes de 10 megabits. De acordo com a qualidade, eles também funcionam em redes de 100 megabits, mas neste caso trabalhando apenas em modo half-duplex.

Note que só existe um grande ganho ao usar o modo full-duplex quando as duas estações precisarem transmitir grandes quantidades de dados ao mesmo tempo. O cenário mais comum é uma das estações transmitindo dados e a outra apenas confirmando o recebimento dos pacotes, onde o modo full-duplex não faz muita diferença.

No Linux, o full-duplex é ativado sempre que suportado. Você pode verificar a configuração através do comando "mii-tool", como em:

```
#                                     mii-tool
eth0: negotiated 100BaseTx-FD, link ok
```

O "100BaseTx-FD" indica que a placa está operando em modo full-duplex. Caso a placa estivesse trabalhando em modo half-duplex, ela apareceria como "100BaseTx-HD". Note que o "FD" e "HD" são justamente abreviações de full-duplex e half-duplex. Caso você estivesse usando placas antigas (de 10 megabits), seriam usados, respectivamente, os modos 10BaseT-FD e 10BaseT-HD.

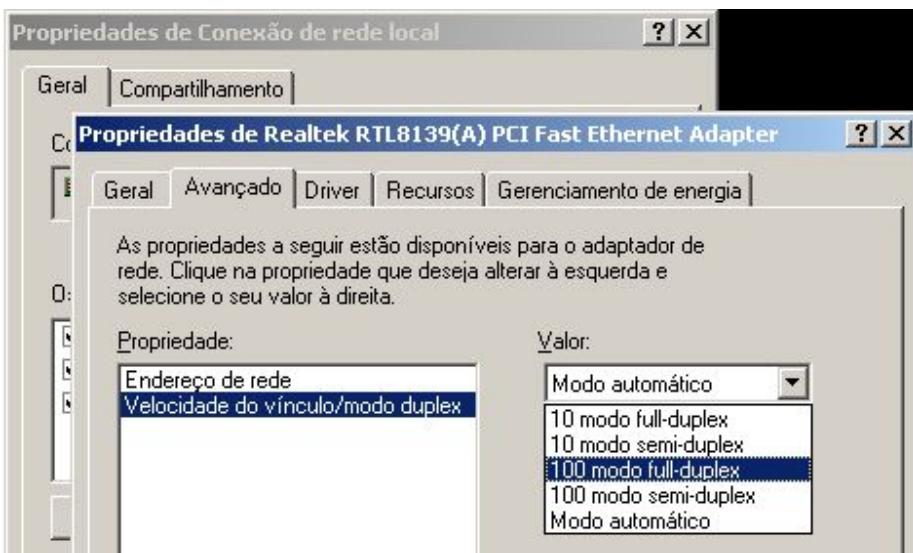
Existe ainda um último modo possível, o "100BaseT4", utilizado ao usar um cabo cross-over para ligar diretamente dois micros.

É possível também usar o mii-tool para forçar um determinado modo de operação usando o parâmetro "-F", seguido do padrão desejado, como em:

```
#                                     mii-tool          -F          100BaseTx-FD
ou:
# mii-tool -F 100BaseTx-HD
```

Note que forçar o modo full-duplex em uma rede onde o hub ou os cabos não suportem este modo de operação, vai fazer com que muitos pacotes comecem a ser perdidos, deixando a rede lenta ou mesmo derrubando a conexão do seu micro.

Para ativar o modo full-duplex no Windows, você precisa apenas acessar as propriedades da conexão de rede e clicar em "configurar" para abrir a janela de opções da placa de rede. Você encontrará a opção de ativar o full-duplex na sessão "Avançado".



» Próximo: [Gigabit Ethernet](#)

Depois dos padrões de 10 e 100 megabits, o passo natural para as redes Ethernet seria novamente multiplicar por 10 a taxa de transmissão, atingindo 1000 megabits. E foi justamente o que aconteceu. O padrão Gigabit Ethernet começou a ser desenvolvido pelo IEEE em 1997 e acabou se ramificando em quatro padrões diferentes.

O **1000BaseLX** é o padrão mais caro, que suporta apenas cabos de fibra óptica e utiliza a tecnologia "long-wave laser", com lasers de 1300 nanômetros. Apesar de, nos quatro padrões, a velocidade de transmissão ser a mesma (1 gigabit), o padrão 1000Base-LX é o que atinge distâncias maiores. Usando cabos de fibra óptica com núcleo de 9 micrônios, o sinal é capaz de percorrer distâncias de até 5 km, enquanto utilizando cabos com núcleo de 50 ou 62.5 micrônios, com freqüências de respectivamente 400 e 500 MHz (que são os cabos mais baratos), o sinal percorre 550 metros.

O segundo padrão é o **1000BaseSX**, que também utiliza cabos de fibra óptica, mas utiliza uma tecnologia de transmissão mais barata, chamada short-wave laser, que é uma derivação da mesma tecnologia usada em CD-ROMs, com feixes de curta distância. Justamente por já

ser utilizado em diversos dispositivos, esta tecnologia é mais barata, mas em compensação o sinal também é capaz de atingir distâncias menores.

Existem quatro padrões de lasers para o 1000BaseSX: com lasers de 50 mícrons e freqüência de 500 MHz (o padrão mais caro), o sinal é capaz de percorrer os mesmos 550 metros dos padrões mais baratos do 1000BaseLX. O segundo padrão também utiliza lasers de 50 mícrons, mas a freqüência cai para 400 MHz e a distância é reduzida para apenas 500 metros. Os outros dois padrões utilizam lasers de 62.5 mícrons e freqüências de 200 e 160 MHz, por isso são capazes de atingir apenas 275 e 220 metros, respectivamente.

Para distâncias mais curtas existe o **1000BaseCX**, que ao invés de fibra óptica utiliza cabos twiaxiais, um tipo de cabo coaxial com dois fios, que tem a aparência de dois cabos coaxiais grudados. Este padrão é mais barato que os dois anteriores, mas em compensação o alcance é de apenas 25 metros. A idéia é que ele sirva para interligar servidores em data centers, que estejam no mesmo rack, ou em racks próximos. Na prática, este padrão é raramente usado.

O padrão que acabou crescendo mais rapidamente, a ponto de quase condenar os demais ao desuso, fora do ramo dos links de longa distância, é o **1000BaseT**, também chamado de GoC ou "Gigabit over Copper", por utilizar os mesmos cabos de par trançado categoria 5 que as redes de 100 megabits atuais.

Isso representa uma enorme economia, não apenas por eliminar a necessidade de trocar os cabos atuais por cabos muito mais caros, mas também nas próprias placas de rede, que passam a ser uma evolução das atuais e não uma tecnologia nova. O alcance continua sendo de 100 metros e os switches compatíveis com o padrão são capazes de combinar nós de 10, 100 e 1000 megabits, sem que os mais lentos atrapalhem os demais. Toda esta flexibilidade torna uma eventual migração para o 1000BaseT relativamente simples, já que você pode aproveitar o cabeamento já existente. Na verdade, pouca coisa muda.

Note que apesar dos cabos serem os mesmos, o 1000BaseT faz um uso muito mais intensivo da capacidade de transmissão e, por isso, detalhes como o comprimento da parte destrançada do cabo para o encaixe do conector, o nível de interferência no ambiente, cabos muito longos, etc. são mais críticos. Com um cabeamento ruim, o índice de pacotes perdidos será muito maior do que em uma rede de 100 megabits, o que vai eliminar parte do ganho de velocidade.

Todos esses padrões de Gigabit Ethernet são intercompatíveis a partir da camada 2 (link de dados) do modelo OSI. Abaixo desse nível está apenas a camada física da rede, que inclui o tipo de cabos e o tipo de modulação usado pela placa de rede para transmitir dados através deles. Os dados transmitidos, incluindo camadas de correção de erro, endereçamento, etc. são idênticos em qualquer um dos padrões.

Assim como muitos hubs antigos permitiam juntar redes que utilizavam cabo de par trançado e cabo coaxial, é muito simples construir dispositivos que interliguem esses diferentes padrões. Isso permite conectar facilmente segmentos de rede com cabeamento e cobre e segmentos com fibra óptica, que podem ser usados para interligar várias redes distantes entre si.

As placas Gigabit Ethernet podem operar tanto no modo full-duplex, quanto no modo half-duplex. Você verá muitas placas anunciadas como capazes de operar a 2 gigabits, o que nada mais é do que uma alusão ao uso do modo full-duplex. Já que temos 1 gigabit em cada sentido, naturalmente a velocidade total é de 2 gigabits.

Mas, como vimos, na prática não funciona bem assim, pois raramente ambas as estações precisarão transmitir grandes quantidades de dados. O mais comum é uma relação assimétrica, com uma falando e a outra apenas enviando os pacotes de confirmação, onde o uso do full-duplex traz um ganho marginal.



Assim como as placas de 100 megabits, as placas gigabit são completamente compatíveis com os padrões anteriores. Você pode até mesmo ligar uma placa Gigabit Ethernet a um hub 10/100 se quiser, mas a velocidade terá de ser nivelada por baixo, respeitando a do ponto mais lento.

A exceção fica por conta de alguns switches nível 3 (os modelos mais inteligentes e caros, que incorporam recursos dos roteadores), que são capazes de "rotear" pacotes de diversas estações operando a 100 megabits, agrupando-os em um único link de 1 gigabit ligado ao servidor. Neste caso, você poderia ter (em teoria) 10 estações baixando arquivos a 100 megabits cada, simultaneamente, a partir de um único servidor com uma placa gigabit.

De qualquer forma, como as placas gigabit estão caindo de preço rapidamente, é importante dar prioridade a placas e switches do novo padrão na hora da compra, de forma a já ir se preparando para uma migração completa no futuro.

» Próximo: [10 Gigabit Ethernet](#)

O padrão para redes 10 Gigabit Ethernet (10G), novamente 10 vezes mais rápido que o anterior, está em desenvolvimento desde 1999. Ele é bastante interessante do ponto de vista técnico, pois além da velocidade, o alcance máximo é de nada menos que 40 km, utilizando cabos de fibra óptica monomodo.



O 10 Gigabit Ethernet também representa o fim dos hubs. O padrão permite apenas o modo de operação full-duplex, onde ambas as estações podem enviar e receber dados simultaneamente, o que só é possível através do uso de switches. Isso encarece mais ainda o novo padrão, mas traz ganhos de desempenho consideráveis, já que além de permitir o uso do modo full-duplex, o uso de um switch inteligente praticamente acaba com as colisões de pacotes.

O 10 Gigabit não se destina a substituir os padrões anteriores, pelo menos a médio prazo. A idéia é complementar os padrões de 100 e 1000 megabits, oferecendo uma solução capaz de interligar redes distantes com uma velocidade comparável ou superior a dos backbones DWDM e SONET, tecnologias muito mais caras, utilizadas atualmente nos backbones da internet.

Suponha, por exemplo, que você precise interligar 5.000 PCs, divididos entre a universidade, o parque industrial e a prefeitura de uma grande cidade. Você poderia utilizar um backbone 10 Gigabit Ethernet para os backbones principais, unindo os servidores dentro dos três blocos e ligando-os à internet, usar uma malha de switches Gigabit Ethernet para levar a rede até as salas de aula e departamentos e, finalmente, usar hubs 10/100 para levar a rede até os alunos e funcionários, complementando com pontos de acesso 802.11bg para oferecer também uma opção de rede sem fio.

Isso estabelece uma pirâmide, onde os usuários individuais possuem conexões relativamente lentas, de 11, 54 ou 100 megabits, interligadas entre si e entre os servidores pelas conexões mais rápidas e caras, formando um sistema capaz de absorver várias chamadas de videoconferência simultâneas, por exemplo.

Outra aplicação em destaque é o próprio uso em backbones de acesso à internet. Usando o 10G, um único cabo de fibra óptica transmite o equivalente a mais de 600 linhas T1 (de 1.5 megabits cada), cada uma suficiente para atender uma empresa de médio porte, um prédio residencial ou mesmo um pequeno provedor de acesso via rádio. Ou seja, com um único link 10G temos banda suficiente para atender com folga a uma cidade de médio porte. O

limite de 40 km pode ser ampliado com o uso de repetidores para completar a distância necessária.

A seqüência de lançamento dos padrões 10G lembra o desenvolvimento dos padrões de Gibabit Ethernet, há alguns anos. Primeiro foram desenvolvidos os padrões utilizando fibra óptica (a mídia de melhor qualidade), seguido por um padrão utilizando cabos twiaxiais (**10GBaseCX-4**), para distâncias curtas.

Existem vários padrões diferentes de redes 10G, voltados para aplicações específicas. O padrão **10GBASE-EX** utiliza fibras monomodo, com laser de 1550 nm e alcance de 40 km (usando fibras de alta qualidade é possível atingir distâncias ainda maiores). O padrão **10GBASE-LX** utiliza um laser mais curto, de 1310 nm (permitindo o aproveitamento de muitas instalações antigas), mas com alcance de "apenas" 15 km. Esses dois padrões são voltados para links de longa distância e backbones.

Depois temos os padrões **10GBASE-SX**, que utilizam fibras multimodo, bem mais baratas. Nele o alcance é de apenas 300 metros, tornando-os mais adequados para interligar roteadores dentro de grandes redes locais, criando uma espinha dorsal de links rápidos.

Um padrão menos usado é o **10GBASE-CX**. Ele utiliza cabos de cobre twiaxiais e por isso é bem mais barato que os anteriores. Porém, ele só funciona em distâncias muito curtas (de até 20 metros), por isso é útil apenas para interligar roteadores dentro de data centers, onde as máquinas estão fisicamente muito próximas.

Está em estágio final de desenvolvimento o padrão **10GBASE-T**, que utiliza cabos de par trançado. Existe a impressão de que o Gigabit Ethernet já levou os cabos de par trançado ao limite, transmitindo 10 vezes mais dados do que os cabos cat 5 foram originalmente desenvolvidos para suportar. O padrão proposto inicialmente para o 10G prevê o uso de cabos categoria 6 e categoria 7, com distância máxima de, respectivamente, 55 metros e 100 metros. Note que o fato de demandar cabos de maior qualidade vai ser um grande obstáculo à popularização do 10G, pois mesmo a partir do ponto em que as placas e switches tornarem-se baratos, ainda haverá o custo de trocar todo o cabeamento.

Existe um grande esforço sendo feito no sentido de criar um padrão que permita utilizar cabos categoria 5e, mesmo que a distâncias mais curtas, possivelmente mantendo os mesmos 55 metros permitidos ao usar cabos categoria 6.

O grande obstáculo é que existe uma grande diferença de qualidade entre os cabos de categoria 5e (os atuais), categoria 6 e categoria 7. Os cabos categoria 5e suportam freqüências de até 100 MHz. Os cabos categoria 6 suportam até 250 MHz, enquanto que os de categoria 7 suportam até 600 MHz.

Os cabos categoria 6 são muito similares aos cabos cat 5e, enquanto nos cabos categoria 7 cada par possui uma blindagem individual, de forma a minimizar a interferência mútua, que torna-se um fator crítico a altas freqüências.

No caso dos cabos categoria 7, está disponível também um novo tipo de conector, desenvolvido pela Siemon, chamado "TERA". Embora muito mais caro e complexo que os

conectores RJ45 atuais, ele oferece a vantagem de ser inteiramente blindado e utilizar um sistema especial de encaixe, que reduz a possibilidade de mal contato:



Atualmente, o 10GBASE-T é ainda um padrão em desenvolvimento. Por isso, muitas novidades ainda podem surgir antes de sua popularização.

» Próximo: [Redes wireless](#)

Usar algum tipo de cabo, seja um cabo de par trançado ou de fibra óptica, é a forma mais rápida e em geral a mais barata de transmitir dados. Os cabos de par trançado cat 5e podem transmitir dados a até 1 gigabit a uma distância de até 100 metros, enquanto os cabos de fibra ótica são usados em links de longa distância, quando é necessário atingir distâncias maiores. Usando 10G, é possível atingir distâncias de até 40 km, sem necessidade de usar repetidores.

Mas, em muitos casos não é viável usar cabos. Imagine que você precise ligar dois escritórios situados em dois prédios diferentes (porém próximos), ou que a sua mãe/esposa/marido não deixa você nem pensar em espalhar cabos pela casa.

A solução nesses casos são as redes sem fio, que estão caindo de preço e, por isso, tornando-se bastante populares. O padrão mais usado é o Wi-Fi (Wireless Fidelity), o nome comercial para os padrões 802.11b, 802.11a e 802.11g. A topologia deste tipo de rede é semelhante a das redes de par trançado, com o hub central substituído pelo ponto de acesso. A diferença no caso é que são usados transmissores e antenas ao invés de cabos. É possível encontrar tanto placas PCMCIA ou mini-PCI, para notebooks, quanto placas PCI, para micros desktop.

Quase todos os notebooks à venda atualmente, muitos modelos de palmtops e até mesmo smartphones já incluem transmissores wireless integrados. Muita gente já acha inconcebível comprar um notebook sem wireless, da mesma forma que ninguém mais

imagina a idéia de um PC sem disco rígido, como os modelos vendidos no início da década de 80.



Na verdade, é bastante raro um notebook que venha com uma placa wireless "onboard". Quase sempre é usada uma placa mini-pci (uma versão miniaturizada de uma placa PCI tradicional, que usa um encaixe próprio), que pode ser substituída como qualquer outro componente. A antena não vai na própria placa, mas é montada na tampa do monitor, atrás do LCD e o sinal vai até a placa através de dois cabos, que correm dentro da carcaça do notebook.

Estas placas mini-pci levam uma vantagem muito grande sobre as placas wireless PCMCIA por causa da antena. As placas PCMCIA precisam ser muito compactas, por isso invariavelmente possuem uma antena muito pequena, com pouca sensibilidade. As antenas incluídas nos notebooks, por sua vez, são invariavelmente muito maiores, o que garante uma conexão muito mais estável, com um alcance muito maior e ajuda até mesmo na autonomia das baterias (já que é possível reduzir a potência do transmissor).

A maioria dos notebooks fabricados a partir do final de 2002 trazem o slot mini-pci e a antena, permitindo que você compre e instale uma placa mini-pci, ao invés de ficar brigando com o alcance reduzido das placas PCMCIA.



Existem vários modelos de placas mini-pci no mercado, mas elas não são um componente comum, de forma que você só vai encontrá-las em lojas especializadas. É possível também substituir a placa que acompanha o notebook por outro modelo, melhor ou mais bem suportado no Linux.



Não se engane pela foto. As placas mini-pci são muito pequenas, quase do tamanho de uma caixa de fósforos e os conectores a antena são quase do tamanho de uma cabeça de alfinete. Eles são frágeis, por isso é preciso ter cuidado ao plugá-los na placa. O fio branco vai sempre no conector no canto da placa e o preto no conector mais ao centro, como na foto.

Quase sempre, o notebook tem uma chave ou um botão que permite ligar e desligar o transmissor wireless. Antes de testar, verifique se ele está ativado.

Embora as placas mini-pci sejam componentes tão padronizados quanto as placas PCMCIA, sempre existe a possibilidade de algumas placas específicas não serem compatíveis com seu notebook. O ideal é sempre testar antes de comprar, ou comprar em uma loja que aceite trocar a placa por outra em caso de problemas.

» Próximo: [O básico](#)

Em uma rede wireless, o hub é substituído pelo **ponto de acesso** (access-point em inglês), que tem a mesma função central que o hub desempenha nas redes com fios: retransmitir os pacotes de dados, de forma que todos os micros da rede os recebam.



Os pontos de acesso possuem uma saída para serem conectados em um hub tradicional, permitindo que você "junte" os micros da rede com fios com os que estão acessando através da rede wireless, formando uma única rede, o que é justamente a configuração mais comum.

Existem poucas vantagens em utilizar uma rede wireless para interligar micros desktops, que afinal não precisam sair do lugar. O mais comum é utilizar uma rede cabeada normal para os desktops e utilizar uma rede wireless complementar para os notebooks, palmtops e outros dispositivos móveis.

Você utiliza um hub/switch tradicional para a parte cabeadas, usando cabo também para interligar o ponto de acesso à rede. O ponto de acesso serve apenas como a "última milha", levando o sinal da rede até os micros com placas wireless. Eles podem acessar os recursos da rede normalmente, acessar arquivos compartilhados, imprimir, acessar a internet, etc. A única limitação fica sendo a velocidade mais baixa e o tempo de acesso mais alto das redes wireless.

Isso é muito parecido com juntar uma rede de 10 megabits, que utiliza um hub "burro" a uma rede de 100 megabits, que utiliza um switch. Os micros da rede de 10 megabits continuam se comunicando entre si a 10 megabits, e os de 100 continuam trabalhando a 100 megabits, sem serem incomodados pelos vizinhos. Quando um dos micros da rede de 10 precisa transmitir para um da rede de 100, a transmissão é feita a 10 megabits, respeitando a velocidade do mais lento.

Para redes mais simples, onde você precise apenas compartilhar o acesso à internet entre poucos micros, todos com placas wireless, você pode ligar o modem ADSL (ou cabo) direto ao ponto de acesso. Alguns pontos de acesso trazem um switch de 4 ou 5 portas embutido, permitindo que você crie uma pequena rede cabeadas sem precisar comprar um hub/switch adicional.



A principal diferença é que em uma rede wireless o meio de transmissão (o ar) é compartilhado por todos os clientes conectados ao ponto de acesso, como se todos estivessem ligados ao mesmo cabo coaxial. Isso significa que apenas uma estação pode transmitir de cada vez, e todas as estações recebem todos os pacotes transmitidos da rede, independentemente do destinatário. Isso faz com que a segurança dentro de uma rede wireless seja uma questão sempre bem mais delicada que em uma rede cabeadas. Outra questão importante é que a velocidade da rede decai conforme aumenta o número de micros conectados, principalmente quando vários deles transmitem dados ao mesmo tempo.

Dependendo da potência dos transmissores nas placas e no pontos de acesso e do tipo de antenas usadas, é possível propagar o sinal da rede por 200, 300 ou até 500 metros de

distância (desde que não existam obstáculos importantes pelo caminho). Usando antenas Yagi (que geram um sinal mais focalizado) e amplificadores é possível interligar dois pontos distantes a 2 km ou mais.

Isso traz mais um problema, que é a questão da interferência entre diferentes redes instaladas na mesma área. Imagine um grande prédio comercial, com muitos escritórios de empresas diferentes e cada uma com sua própria rede wireless. Os pontos de acesso podem ser configurados para utilizarem freqüências diferentes, divididas em 16 canais. Devido à legislação de cada país, apenas 11, 13 ou 14 destes canais podem ser usados e destes, apenas 4 podem ser usados simultaneamente, sem que realmente não exista interferência. Ou seja, com várias redes instaladas próximas umas das outras, os canais disponíveis são rapidamente saturados, fazendo com que o tráfego de uma efetivamente reduza o desempenho da outra.

Existe ainda a questão das interferências e de materiais que atenuam o sinal. Em primeiro lugar temos as superfícies de metal em geral, como janelas, portas metálicas, lajes, vigas e até mesmo tintas com pigmentos metálicos. Depois temos concentrações de líquido, como aquários, piscinas, caixas d'água e até mesmo pessoas passeando pelo local (nossa corpo é composto de 70% de água).

Fornos de microondas operam na mesma freqüência das redes wireless, fazendo com que, quando ligados, eles se transformem em uma forte fonte de interferência, prejudicando as transmissões num raio de alguns metros. Telefones sem fio, que operam na faixa dos 2.4 GHz, também interferem, embora em menor grau.

Os fabricantes falam em 150 ou até 300 metros de alcance máximo, mas essas distâncias são atingidas apenas em campo aberto, em condições ideais. Na prática, o alcance varia muito de acordo com o ambiente. Você pode conseguir pegar o sinal de um ponto de acesso instalado na janela de um prédio vizinho, distante 100 metros do seu (campo aberto), mas não conseguir acessar a rede do andar de cima (a armação de ferro e cimento da laje é um obstáculo difícil de transpor). Para compensar grandes distâncias, obstáculos ou interferências, o ponto de acesso reduz a velocidade de transmissão da rede, como um modem discado tentando se adaptar a uma linha ruidosa. Os 54 megabits originais podem se transformar rapidamente em 11, 5.5, 2 ou até mesmo 1 megabit.

Temos ainda a questão da segurança: se você morar em um sobrado e colocar o ponto de acesso próximo da janela da frente do quarto no primeiro andar, provavelmente um vizinho do quarteirão seguinte ainda vai conseguir se conectar à sua rede, desde que substitua a antena da placa por uma mais potente. Existe até uma velha receita que circula pela internet de como fazer uma antena caseira razoável usando um tubo de batata Pringles. Não é brincadeira: o tubo é forrado de papel alumínio e tem um formato adequado para atuar como uma antena.

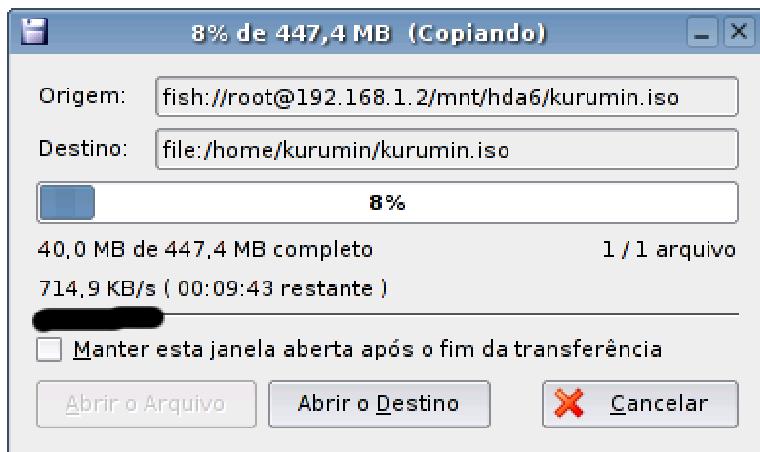
Caímos, então, em um outro problema. Você simplesmente não tem como controlar o alcance do sinal da rede. Qualquer vizinho próximo, com uma antena potente (ou um tubo de batata), pode conseguir captar o sinal da sua rede e se conectar a ela, tendo acesso à sua conexão com a web, além de arquivos e outros recursos que você tenha compartilhado entre os micros da rede, o que não é muito interessante.

Eis que surge o **WEP**, abreviação de "Wired-Equivalent Privacy", que, como o nome sugere, traz como promessa um nível de segurança equivalente ao das redes cabeadas. Na prática, o WEP tem muitas falhas e é relativamente simples de quebrar, mas não deixa de ser uma camada de proteção básica que você sempre deve manter ativa. A opção de ativar o WEP aparece no painel de configuração do ponto de acesso.

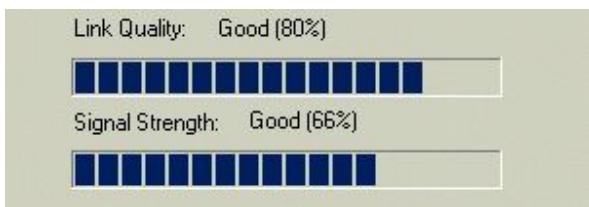
O WEP se encarrega de encriptar os dados transmitidos através da rede. Existem dois padrões WEP: de 64 e de 128 bits. O padrão de 64 bits é suportado por qualquer ponto de acesso ou interface que siga o padrão WI-FI, o que engloba todos os produtos comercializados atualmente. O padrão de 128 bits, por sua vez, não é suportado por todos os produtos, mas em compensação é bem menos inseguro. Para habilitá-lo será preciso que todos os componentes usados na sua rede suportem o padrão, caso contrário os nós que suportarem apenas o padrão de 64 bits ficarão fora da rede.

Existe ainda o **WPA**, um padrão mais seguro, que já é suportado pela grande maioria das placas e dos pontos de acesso. Existem várias variações do WPA, que utilizam diversos sistemas de encriptação diferentes, com a opção de usar um servidor Radius para centralizar os logins da rede, opção muito usada em empresas. No entanto, o mais comum em pequenas redes é usar o WPA-PSK (o padrão mais simples), onde é definida uma chave (uma espécie de senha), usada para autenticar os clientes da rede. PSK é abreviação de "Pre-Shared Key", ou "chave previamente compartilhada".

Temos, em seguida, a questão da velocidade. Nas redes **802.11b**, o padrão original, a velocidade teórica é de apenas 11 megabits (ou 1.35 MB/s). Como as redes wireless possuem um overhead muito grande, por causa da modulação do sinal, checagem e retransmissão dos dados, as taxas de transferências, na prática, ficam em torno de 750 KB/s, menos de dois terços do máximo.



Conforme o cliente se distancia do ponto de acesso, a taxa de transmissão cai para 5 megabits, 2 megabits e 1 megabit, até que o sinal se perca definitivamente. No Windows você pode usar o utilitário que acompanha a placa de rede para verificar a qualidade do sinal em cada parte do ambiente onde a rede deverá estar disponível. No Linux isso é feito por programas como o Kwifimanager, que veremos a seguir.



Veja que tanto na questão da segurança, quanto na questão do desempenho, as redes wireless perdem para as redes cabeadas. A maior arma das redes wireless é a versatilidade. O simples fato de poder interligar os PCs sem precisar passar cabos pelas paredes já é o suficiente para convencer muitas pessoas, mas existem mais alguns recursos interessantes que podem ser explorados.

Sem dúvida, a possibilidade mais interessante é a mobilidade para os portáteis. Tanto os notebooks, quanto handhelds e as webpads podem ser movidos livremente dentro da área coberta pelos pontos de acesso sem que seja perdido o acesso à rede. Essa possibilidade lhe dá mobilidade dentro de casa para levar o notebook para onde quiser, sem perder o acesso à web, mas é ainda mais interessante para empresas e escolas. No caso das empresas, a rede permite que os funcionários possam se deslocar pela empresa sem perder a conectividade com a rede (entrando e saindo de reuniões ou caminhando pela linha de produção, por exemplo), e basta se aproximar do prédio para que seja possível se conectar à rede e ter acesso aos recursos necessários.

No caso das escolas, a principal utilidade seria fornecer acesso à web aos alunos. Muitas lojas e a maior parte dos aeroportos pelo mundo já oferecem acesso à web através de redes sem fio como uma forma de serviço para seus clientes. Um exemplo famoso é o da rede de cafés Starbuks nos EUA e Europa, onde todas as lojas oferecem acesso gratuito à web para os clientes que possuem um notebook ou outro portátil com placa wireless.

» Próximo: [Padrões](#)

O **802.11b** foi o primeiro padrão wireless usado em grande escala. Ele marcou a popularização da tecnologia. Naturalmente, existiram vários padrões anteriores, como o 802.11 (que trabalhava a 1 ou 2 megabits) e também alguns padrões proprietários, incompatíveis entre si, como o Arlan da Aironet e o WaveLAN, da NCR, que trabalhavam na faixa dos 900 MHz e transmitiam a respectivamente 860 kbits e 2 megabits.

O 802.11b permitiu que placas de diferentes fabricantes se tornassem compatíveis e os custos caíssem, graças ao aumento na demanda e à concorrência. O padrão seguinte foi o **802.11a** (que na verdade começou a ser desenvolvido antes do 802.11b, mas foi finalizado depois), que utiliza uma faixa de freqüência mais alta: 5 GHz e oferece uma velocidade teórica de 54 megabits, porém a distâncias menores, cerca de metade da distância atingida por uma placa 802.11b usando o mesmo tipo de antena.

Embora os dois padrões sejam incompatíveis, a maior parte das placas 802.11a incorporam chips capazes de trabalhar nas duas faixas de freqüência, permitindo que sejam usadas nos

dois tipos de redes. Uma observação importante é que, ao misturar placas 802.11a e 802.11b, a velocidade é nivelada por baixo e toda a rede passa a operar a 11 megabits. Lembre-se que uma rede wireless opera de forma similar às redes antigas, com cabos coaxiais: todos compartilham o mesmo "cabo".

Finalmente, temos o padrão atual, o **802.11g**. Ele utiliza a mesma faixa de freqüência do 802.11b: 2.4 GHz. Isso permite que os dois padrões sejam intercompatíveis. A idéia é que você possa adicionar placas e pontos de acesso 802.11g a uma rede 802.11b já existente, mantendo os componentes antigos, do mesmo modo como hoje em dia temos liberdade para adicionar placas e switches Gigabit Ethernet a uma rede já existente de 100 megabits.

Apesar disso, a velocidade de transmissão no 802.11g é de 54 megabits, como nas redes 802.11a. Na prática, em redes 802.11a é possível atingir taxas de transmissão (reais) em torno de 3,4 MB/s, enquanto que as redes 802.11g são um pouco mais lentas, atingindo cerca de 3,0 MB/s em condições ideais. Mas, fora esta pequena desvantagem no desempenho, as redes 802.11g juntam o melhor dos dois mundos.

Note que, para que a rede efetivamente trabalhe a 54 megabits, é necessário que o ponto de acesso e todas as placas sejam 802.11g. Ao incluir uma única placa 802.11b na rede (mesmo que seja seu vizinho roubando sinal), toda a rede passa a operar a 11 megabits. As placas 802.11g não são compatíveis com o padrão 802.11a, mas os dois tipos de placas podem conversar a 11 megabits, utilizando o padrão 802.11b, que vira um denominador comum.

Temos ainda as placas dual-band, que transmitem simultaneamente em dois canais diferentes, dobrando a taxa de transmissão (e também o nível de interferência com outras redes próximas). Chegamos então às placas de 22 megabits (802.11b) e 108 megabits (802.11g). Lembre-se de que, como de praxe, você só atinge a velocidade máxima usando apenas placas dual-band.

Ou seja, sem um bom controle sobre quem se conecta à rede, você corre o risco de ver sua rede operando a 11 megabits na maior parte do tempo.

» Próximo: [Aumentando o alcance](#)

Assim como em outras tecnologias de transmissão via rádio, a distância que o sinal é capaz de percorrer depende também da qualidade da antena usada. As antenas padrão utilizadas nos pontos de acesso (geralmente de 2 dBi) são pequenas e práticas, além de relativamente baratas, mas existe a opção de utilizar antenas mais sofisticadas para aumentar o alcance da rede.

Alguns fabricantes chegam a dizer que o alcance dos seus pontos de acesso chega a 300 metros, usando as pequenas antenas padrão. Isso está um pouco longe da realidade, pois só pode ser obtido em campos abertos, livres de qualquer obstáculo e, mesmo assim, com o

sinal chegando muito fraco ao final dos 300 metros, já com a rede trabalhando na velocidade mínima, a 1 megabit e com um lag muito grande.

Apesar disso, a distância máxima e a qualidade do sinal (e, consequentemente, a velocidade de transmissão) podem variar bastante de um modelo de ponto de acesso para outro, de acordo com a qualidade e potência do transmissor e da antena usada pelo fabricante. Existem basicamente três tipos de antenas que podem ser utilizadas para aumentar o alcance da rede:

As antenas **Yagi** são as que oferecem um maior alcance, mas em compensação são capazes de cobrir apenas uma pequena área, para onde são apontadas. Estas antenas são mais úteis para cobrir alguma área específica, longe do ponto de acesso, ou interligar duas redes distantes.

Em ambos os casos, o alcance ao usar uma antena Yagi pode passar facilmente ultrapassar os 1000 metros. Usando uma antena de alto ganho em cada ponto, uma delas com um amplificador de 1 watt (o máximo permitido pela legislação), é possível atingir 5 km ou mais. As Yagi são também o melhor tipo de antena a usar quando é preciso concentrar o sinal para "furar" um obstáculo entre as duas redes, como, por exemplo, um prédio bem no meio do caminho. Nestes casos a distância atingida será sempre mais curta, naturalmente.

Uma solução muito adotada nestes casos é usar um repetidor instalado num ponto intermediário, permitindo que o sinal desvie do obstáculo. Existem até mesmo pontos de acesso extremamente robustos, desenvolvidos para uso industrial, que além de uma gabinete reforçado, utilizam placas solares e baterias, que permitem a eles funcionar de forma inteiramente autônoma.



Outra solução comum é usar dois pares de cabo de rede (a rede funciona perfeitamente apenas com dois cabos) para enviar energia ao ponto de acesso, eliminando o uso de um cabo de força separado. Esta solução é chamada de "Power Over Ethernet" (POE), veja mais detalhes no: <http://www.poweroverethernet.com/>.

Voltando ao tema principal, a instalação das antenas Yagi é complicada, pois uma antena deve ficar apontada exatamente para a outra, cada uma no topo de um prédio ou morro, de forma que não exista nenhum obstáculo entre as duas. No final da instalação é usado um laser para fazer um ajuste fino "mirando" as duas antenas.

As antenas feitas com tubos de batatas Pringles são justamente um tipo de antena Yagi de baixo ganho. Outra dica é que os pontos de acesso quase sempre possuem duas saídas de antena. Você pode usar uma antena convencional em uma delas, para manter o sinal em um raio circular, atendendo aos micros próximos e usar uma antena Yagi na segunda, para criar um link com um local específico, distante do ponto de acesso.



A segunda opção são as antenas **omnidirecionais**, que, assim como as antenas padrão dos pontos de acesso, cobrem uma área circular em torno da antena. Elas são boas irradiando o sinal na horizontal, mas não na vertical, por isso devem ser sempre instaladas "de pé", a menos que a intenção seja pegar sinal no andar de cima. As antenas nos clientes devem sempre estar alinhadas (também de pé) com a antena do ponto de acesso, para uma melhor recepção. Caso o cliente use algum tipo de antena mini-yagi, então a antena deve ficar apontada para o ponto de acesso.

A vantagem de usar uma omnidirecional externa é a possibilidade de utilizar uma antena de maior ganho. Existem modelos de antenas omnidirecionais de 3 dBi, 5 dBi, 10 dBi ou até mesmo 15 dBi, um grande avanço sobre as antenas de 2 ou 3 dBi que acompanham a maioria dos pontos de acesso.



Assim como as Yagi, as antenas omnidirecionais podem ser usadas tanto para aumentar a área de cobertura do ponto de acesso, quanto serem instaladas em placas de rede wireless com antenas destacáveis, permitindo captar o sinal do ponto de acesso de uma distância maior.

Uma terceira opção de antena são as **parabólicas** ou **miniparabólicas**, que também captam o sinal em apenas uma direção, de forma ainda mais concentrada que as Yagi, permitindo que sejam atingidas distâncias maiores. As miniparabólicas mais "populares" possuem, geralmente, 24 ou 28 dbi de potência, enquanto as maiores e mais caras podem chegar a 124 dBi (ou mais).



Estas antenas podem custar de 30 a mais de 200 dólares, dependendo da potência. As antenas Yagi estão entre as mais caras, vendidas por 150 dólares ou mais. Além do problema do preço, existe um aumento no risco de uso indevido na rede, já que o sinal irá se propagar por uma distância maior, mais uma razão para reforçar a segurança.

Para ligar as antenas ao ponto de acesso ou à placa é usado um cabo especial chamado **pigtail**, um cabo fino, sempre relativamente curto, usado como um adaptador entre a minúscula saída usada nas placas e a entrada do cabo ou antena. Os pigtails invariavelmente causam uma pequena perda de sinal, pois para ser flexível o cabo possui apenas uma fina camada de blindagem. Justamente por isso, eles devem ser o mais curto possíveis, tendo apenas o comprimento necessário para realizar a conexão.



Ao cobrir distâncias maiores, o ideal é que o ponto de acesso seja instalado próximo à antena, com um cabo de rede ligando-o ao servidor ou switch. As redes 802.11x trabalham com sinais de baixa potência (em geral menos de 0.25 watt); por isso, qualquer tipo de cabo longo causa uma grande perda de sinal.

Para casos em que a antena do ponto de acesso não é suficiente, mas também não existe necessidade de uma antena cara, existe a opção de fazer um concentrador caseiro, um tipo de "antena" que concentra o sinal recebido pela antena padrão do ponto de acesso, fazendo com que ela cubra uma área mais focalizada, porém com um ganho maior. Além de melhorar a qualidade do sinal na área desejada, ela reduz o alcance nas demais direções, fazendo com que seja muito mais difícil captar o sinal da sua rede de fora.

Esta é uma receita muito simples. Você precisa de alguma folha de metal ou fio (como uma malha de fios, pedaço de lata ou papel laminado) e um pedaço de isopor ou papelão, recortado em formato de lua e com um orifício no centro, usado para encaixar na antena. O papel laminado é colado em volta do molde e o conjunto é encaixado em uma das antenas do ponto de acesso.



Assim como em uma antena miniparabólica, os sinais recebidos em determinada direção (para onde a antena é apontada) são refletidos de forma concentrada em direção à antena do ponto de acesso ou placa, aumentando o ganho. Por outro lado, o sinal torna-se muito mais fraco nas outras direções, dificultando as coisas para seu vizinho interessado em roubar sinal.

Você pode baixar o modelo com os ângulos corretos no:
<http://www.freeantennas.com/projects/template/parabolic.pdf>

Várias fotos com exemplos estão disponíveis no:
<http://www.freeantennas.com/projects/template/gallery/>

Existe ainda a popular "cantenna", um tipo de antena Yagi feita usando uma lata de batata Pringles. Você encontra a receita no:
<http://www.oreillynet.com/cs/weblog/view/wlg/448>

» Próximo: [**Capítulo 2: Configurando a rede**](#)

O desenvolvimento das diferentes arquiteturas de redes começou bem antes do que se imagina e, como a maioria das grandes invenções, o propósito inicial era o uso militar, ainda na época da Guerra Fria. Uma das principais prioridades dentro de uma força militar é a comunicação, certo? No final da década de 60, esta era uma grande preocupação do DOD, Departamento de Defesa do Exército Americano: como interligar computadores de arquiteturas completamente diferentes, e que ainda por cima estavam muito distantes um do outro, ou mesmo em alto-mar, dentro de um porta aviões ou submarino?

Após alguns anos de pesquisa, surgiu o TCP/IP, abreviação de "Transmission Control Protocol/Internet Protocol", ou protocolo de controle de transmissão/protocolo internet. O TPC/IP permitiu que as várias pequenas redes de computadores do exército Americano fossem interligadas, formando uma grande rede, embrião do que hoje conhecemos como Internet. Como vimos, o TCP/IP é composto de dois protocolos, o IP cuida do endereçamento, enquanto o TCP cuida da transmissão dos dados e correção de erros.

O segredo do TCP/IP é dividir a grande rede em pequenas redes independentes, interligadas por roteadores. Como (apesar de interligadas) cada rede é independente da outra, caso uma das redes pare, apenas aquele segmento fica fora do ar, sem afetar a rede como um todo.

No caso do DOD, este era um recurso fundamental, pois durante uma guerra ou durante um ataque nuclear, vários dos segmentos da rede seriam destruídos, junto com suas respectivas bases, navios, submarinos, etc. Era crucial que o que sobrasse da rede continuasse no ar, permitindo ao comando coordenar um contra-ataque. Veja que mesmo atualmente este recurso continua sendo fundamental na Internet: se os roteadores de um provedor de acesso ficam fora do ar, apenas os clientes dele são prejudicados.

Apesar de inicialmente o uso do TPC/IP ter sido restrito a aplicações militares, com o passar do tempo o protocolo acabou tornando-se de domínio público, o que permitiu aos fabricantes de software adicionar suporte ao TCP/IP aos seus sistemas operacionais de rede.

Atualmente, o TPC/IP é suportado por todos os principais sistemas operacionais, não apenas os destinados a PCs, mas a praticamente todas as arquiteturas, incluindo até mesmo celulares e handhelds. Qualquer sistema com um mínimo de poder de processamento pode conectar-se à Internet, desde que alguém desenvolva uma implementação do TCP/IP para ele, juntamente com alguns aplicativos.

Até mesmo o MSX já ganhou um sistema operacional com suporte a TCP/IP e navegador que, embora de forma bastante limitada, permite que um jurássico MSX com 128k de memória (ligado na TV e equipado com um modem serial) acesse a web. Se duvida, veja com seus próprios olhos no: <http://uzix.sourceforge.net/uzix2.0/> ;).



Voltando à história da Internet, pouco depois de conseguir interligar seus computadores com sucesso, o DOD interligou alguns de seus computadores às redes de algumas universidades e centros de pesquisa, formando uma inter-rede, ou Internet. Logo a seguir, no início dos anos 80, a NSF (National Science Foundation) construiu uma rede de fibra óptica de alta velocidade, conectando centros de supercomputação localizados em pontos-chave nos EUA e interligando-os também à rede do DOD.

Essa rede da NSF teve um papel fundamental no desenvolvimento da Internet, por reduzir substancialmente o custo da comunicação de dados para as redes de computadores existentes, que foram amplamente estimuladas a se conectar ao backbone da NSF e, consequentemente, à Internet. A partir de abril de 1995, o controle do backbone (que já havia se tornado muito maior, abrangendo quase todo o planeta através de cabos submarinos e satélites) foi passado para o controle privado. Além do uso acadêmico, o interesse comercial pela Internet impulsionou seu crescimento, chegando ao que temos hoje.

Tudo o que vimos até agora, sobre placas e cabos, representa a parte física da rede, os componentes necessários para fazer os uns e zeros enviados por um computador chegarem ao outro. O protocolo de rede é o conjunto de regras e padrões que permite que eles realmente falem a mesma língua.

Pense nas placas, hubs e cabos como o sistema telefônico e no TCP/IP como a língua falada, que você realmente usa para se comunicar. Não adianta ligar para alguém na China que não saiba falar português. Sua voz vai chegar até lá, mas a pessoa do outro lado não vai entender nada. Além da língua em si, existe a necessidade de ter assuntos em comum para poder manter a conversa.

Ligar os cabos e ver se os leds do hub e das placas estão acesos é o primeiro passo. O segundo é configurar os endereços da rede para que os micros possam conversar entre si e o terceiro é finalmente compartilhar a internet, arquivos, impressoras e o que mais você quer que os outros micros da rede tenham acesso (dentro da rede interna), ou mesmo alugar seu próprio servidor dedicado, hospedado em um datacenter.

Graças ao TCP/IP, tanto o Linux quanto o Windows e outros sistemas operacionais em uso são intercompatíveis dentro da rede. Não existe problema para as máquinas com o Windows acessarem a Internet através da conexão compartilhada no Linux, por exemplo. O TCP/IP é a língua mãe que permite que todos se comuniquem.

» Próximo: [Endereços e compartilhamentos](#)

Independentemente do sistema operacional usado, os parâmetros necessários para configurar a rede e acessar a web através de uma conexão compartilhada são os mesmos. Muda apenas a ferramenta de configuração usada.

- **Endereço IP:** Os endereços IP identificam cada micro na rede. A regra básica é que cada

micro deve ter um endereço IP diferente e devem ser utilizados endereços dentro da mesma faixa.

Um endereço IP é composto de uma seqüência de 32 bits, divididos em 4 grupos de 8 bits cada. Cada grupo de 8 bits recebe o nome de octeto. Veja que 8 bits permitem 256 combinações diferentes (para comprovar, é só calcular quanto é dois elevado à oitava potência). Para facilitar a configuração dos endereços, usamos números de 0 a 255 para representar cada octeto, formando endereços como 220.45.100.222, 131.175.34.7 etc. Muito mais fácil do que ficar decorando seqüências de números binários.

O endereço IP é dividido em duas partes. A primeira identifica a rede à qual o computador está conectado (necessário, pois, em uma rede TCP/IP, podemos ter várias redes conectadas entre si, como no caso da internet) e a segunda identifica o computador (chamado de host) dentro da rede.

Obrigatoriamente, os primeiros octetos servirão para identificar a rede e os últimos servirão para identificar o computador em si. Como temos apenas 4 octetos, esta divisão limitaria bastante o número de endereços possíveis, o que seria uma grande limitação no caso da internet, onde existe um número muito grande de redes diferentes, muitas delas com um número muito grande de micros conectados, como no caso dos grandes provedores de acesso.

Se fosse reservado apenas o primeiro octeto do endereço, por exemplo, teríamos um grande número de hosts (micros conectados a cada rede), mas em compensação poderíamos ter apenas 256 redes diferentes, o que seria muito complicado, considerando o tamanho do mundo ;).

Mesmo se reservássemos dois octetos para a identificação da rede e dois para a identificação do host, os endereços possíveis seriam insuficientes, pois existem muito mais de 65 mil redes diferentes no mundo, conectadas entre si através da internet, e existem algumas redes com mais de 65 mil micros.

Para permitir uma gama maior de endereços, os desenvolvedores do TPC/IP dividiram o endereçamento IP em cinco classes, denominadas A, B, C, D e E, sendo que as classes D e E estão reservadas para expansões futuras. Cada classe reserva um número diferente de octetos para o endereçamento da rede.

Na **classe A**, apenas o primeiro octeto identifica a rede, na **classe B** são usados os dois primeiros octetos e na **classe C** (a mais comum) temos os três primeiros octetos reservados para a rede e apenas o último reservado para a identificação dos hosts.

O que diferencia uma classe de endereços da outra é o valor do primeiro octeto. Se for um número entre **1 e 126** (como em 113.221.34.57), temos um endereço de classe **A**. Se o valor do primeiro octeto for um número entre **128 e 191**, então temos um endereço de classe **B** (como em 167.27.135.203) e, finalmente, caso o primeiro octeto seja um número entre **192 e 223**, teremos um endereço de classe **C**, como em 212.23.187.98.

Isso permite que existam ao mesmo tempo redes pequenas, com até 254 micros, usadas, por exemplo, por pequenas empresas e provedores de acesso, e redes muito grandes, usadas por grandes empresas, datacenters ou grandes provedores de acesso.

Todos os endereços IP válidos na internet possuem dono. Seja alguma empresa ou alguma entidade certificadora que os fornece junto com novos links. Por isso, não podemos utilizar nenhum deles a esmo. Quando você se conecta na internet, você recebe um (e apenas um) endereço IP válido, emprestado pelo provedor de acesso, algo como, por exemplo, "200.220.231.34". É através deste número que outros computadores na internet podem enviar informações e arquivos para o seu.

Quando quiser configurar uma rede local, você deve usar um dos **endereços reservados**, endereços que não existem na internet e que, por isso, podemos utilizar à vontade em nossas redes particulares. As faixas reservadas de endereços são:

10.x.x.x,	com máscara	de sub-rede	255.0.0.0
172.16.x.x	até 172.31.x.x,	com máscara	de sub-rede
192.168.0.x até 192.168.255.x,	com máscara de sub-rede 255.255.255.0		255.255.0.0

Você pode usar qualquer uma dessas faixas de endereços na sua rede. Uma faixa de endereços das mais usadas é a 192.168.0.x, onde o "192.168.0." vai ser igual em todos os micros da rede e muda apenas o último número, que pode ser de 1 até 254 (o 0 e o 255 são reservados para o endereço da rede e o sinal de broadcast). Se você tiver 4 micros na rede, os endereços deles podem ser, por exemplo, 192.168.0.1, 192.168.0.2, 192.168.0.3 e 192.168.0.4.

Micros configurados para usar faixas de endereços diferentes entendem que fazem parte de redes diferentes e não conseguem se enxergar mutuamente. Uma configuração muito comum em grandes redes é dividir os micros em diversas faixas de IPs diferentes (como 192.168.0.x, 192.168.1.x, 192.168.2.x, etc.) e usar um roteador (que pode ser um servidor com várias placas de rede) para interligá-las.

- Máscara de sub-rede: Ao contrário do endereço IP, que é formado por valores entre 0 e 255, a máscara de sub-rede é formada por apenas dois valores: 0 e 255, como em 255.255.0.0 ou 255.0.0.0, onde um valor 255 indica a parte do endereço IP referente à rede e um valor 0 indica a parte do endereço referente ao host, o endereço particular de cada computador que faz parte dela.

A máscara de rede padrão acompanha a classe do endereço IP: em um endereço de classe A, a máscara será 255.0.0.0, indicando que o primeiro octeto se refere à rede e os três últimos ao host; em um endereço classe B, a máscara padrão será 255.255.0.0, onde os dois primeiros octetos referem-se à rede e os dois últimos ao host, enquanto em um endereço classe C, a máscara padrão será 255.255.255.0, onde apenas o último octeto refere-se ao host.

Mas, afinal, para que servem as máscaras de sub-rede então? Apesar das máscaras padrão acompanharem a classe do endereço IP, é possível "mascarar" um endereço IP, mudando as faixas do endereço que serão usadas para endereçar a rede e o host.

Veja, por exemplo, o endereço "192.168.0.1". Por ser um endereço de classe C, sua máscara padrão seria 255.255.255.0, indicando que o último octeto se refere ao host, e os demais à rede. Porém, se mantivéssemos o mesmo endereço, mas alterássemos a máscara para 255.255.0.0, apenas os dois primeiros octetos (192.168) continuariam representando a rede, enquanto o host passaria a ser representado pelos dois últimos (e não apenas pelo último).

O endereço "192.168.0.1" com máscara 255.255.255.0 é diferente de "192.168.0.1" com máscara 255.255.0.0. Enquanto no primeiro caso temos o host "1" dentro da rede "192.168.0", no segundo caso temos o host "0.1" dentro da rede "192.168".

A moral da história é que dentro da rede você deve configurar sempre todos os micros para usarem a mesma máscara de sub-rede, seguindo a faixa de endereços escolhida. Se você está usando a faixa 192.168.0.x, então a máscara de sub-rede vai ser 255.255.255.0 para todos os micros.

- **Default Gateway** (gateway padrão): Lembra que disse que quando você se conecta à internet através de um provedor de acesso qualquer você recebe apenas um endereço IP válido? Quando você compartilha a conexão entre vários micros, apenas o servidor que está compartilhando a conexão possui um endereço IP válido, só ele "existe" na internet. Todos os demais acessam através dele.

O default gateway ou gateway padrão é justamente o micro da rede que tem a conexão, que os outros consultarão quando precisarem acessar qualquer coisa fora da rede local. Por exemplo, se você montar uma rede doméstica com 4 PCs, usando os endereços 192.168.0.1, 192.168.0.2, 192.168.0.3 e 192.168.0.4, e o PC 192.168.0.1 estiver compartilhando o acesso à internet, as outras três estações deverão ser configuradas para utilizar o endereço "192.168.0.1" como gateway padrão.

Servidor DNS: O DNS (domain name system) permite usar nomes amigáveis ao invés de endereços IP para acessar servidores. Quando você se conecta à internet e acessa o endereço <http://www.guiadohardware.net>, é um servidor DNS que converte o "nome fantasia" no endereço IP real do servidor, permitindo que seu micro possa acessá-lo.

Para tanto, o servidor DNS mantém uma tabela com todos os nomes fantasia, relacionados com os respectivos endereços IP. A maior dificuldade em manter um servidor DNS é justamente manter esta tabela atualizada, pois o serviço tem que ser feito manualmente.

Dentro da internet, temos várias instituições que cuidam desta tarefa. No Brasil, por exemplo, temos a FAPESP. Para registrar um domínio, é preciso fornecer a eles o endereço IP real do servidor onde a página ficará hospedada. A FAPESP cobra uma taxa de manutenção anual de R\$ 30 por este serviço. Servidores DNS também são muito usados em intranets, para tornar os endereços mais amigáveis e fáceis de guardar.

Faz parte da configuração da rede informar os endereços DNS do provedor (ou qualquer outro servidor que você tenha acesso), que é para quem seu micro irá perguntar sempre que

você tentar acessar qualquer coisa usando um nome de domínio e não um endereço IP. O jeito mais fácil de conseguir os endereços do provedor é simplesmente ligar para o suporte e perguntar.

O ideal é informar dois endereços. Assim, se o primeiro estiver fora do ar, você continua acessando através do segundo. Também funciona com um endereço só, mas você perde a redundância. Exemplos de endereços de servidores DNS são **200.204.0.10** e **200.204.0.138**.

DHCP: O DHCP ("Dynamic Host Configuration Protocol" ou "protocolo de configuração dinâmica de endereços de rede") permite que todos os micros da rede recebam suas configurações de rede automaticamente a partir de um servidor central, sem que você precise ficar configurando os endereços manualmente em cada um.

O protocolo DHCP trabalha de uma forma bastante interessante. Inicialmente, a estação não sabe quem é, não possui um endereço IP e não sabe sequer qual é o endereço do servidor DHCP da rede. Ela manda, então, um pacote de broadcast endereçado ao IP "255.255.255.255", que é transmitido pelo switch para todos os micros da rede. O servidor DHCP recebe este pacote e responde com um pacote endereçado ao endereço IP "0.0.0.0", que também é transmitido para todas as estações.

Apesar disso, apenas a estação que enviou a solicitação lerá o pacote, pois ele é endereçado ao endereço MAC da placa de rede. Como vimos na introdução, quando uma estação recebe um pacote destinado a um endereço MAC diferente do seu, ela ignora a transmissão.

Dentro do pacote enviado pelo servidor DHCP estão especificados o endereço IP, máscara, gateway e servidores DNS que serão usados pela estação. Veremos como configurar um servidor DHCP em detalhes no capítulo 5 do livro, com mais algumas dicas no capítulo 9, onde falo sobre a configuração de servidores de boot remoto.

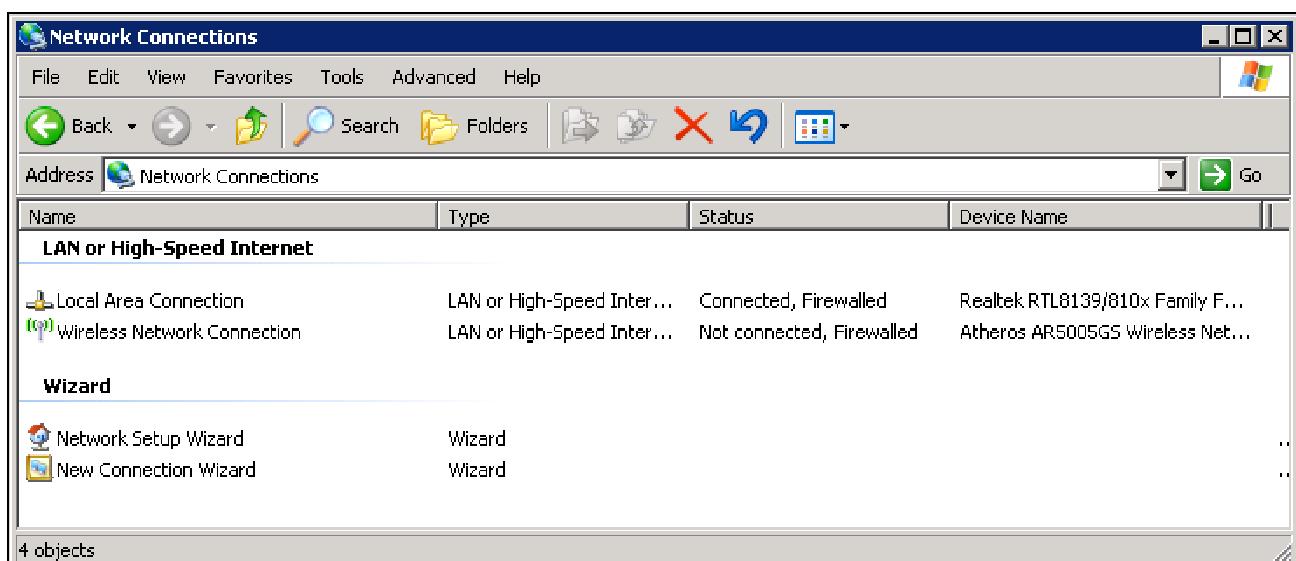
Este endereço é temporário, não é da estação, mas simplesmente é "emprestado" pelo servidor DHCP para que seja usado durante um certo tempo (lease time), definido na configuração do servidor. Depois de decorrido metade do tempo de empréstimo, a estação tentará contatar o servidor DHCP para renovar o empréstimo. Se o servidor DHCP estiver fora do ar, ou não puder ser contatado por qualquer outro motivo, a estação esperará até que tenha se passado 87.5% do tempo total, tentando várias vezes em seguida. Se, terminado o tempo do empréstimo, o servidor DHCP ainda não estiver disponível, a estação abandonará o endereço e ficará tentando contatar qualquer servidor DHCP disponível, repetindo a tentativa a cada 5 minutos. Porém, por não ter mais um endereço IP, a estação ficará fora da rede até que o servidor DHCP volte a responder.

Veja que uma vez instalado, o servidor DHCP passa a ser essencial para o funcionamento da rede. Se ele estiver travado ou desligado, as estações não terão como obter seus endereços IP e não conseguirão entrar na rede. Todos os provedores de acesso discado usam servidores DHCP para fornecer dinamicamente endereços IP aos usuários. No caso deles, esta é uma necessidade, pois o provedor possui uma quantidade de endereços IP válidos, assim como um número de linhas bem menor do que a quantidade total de

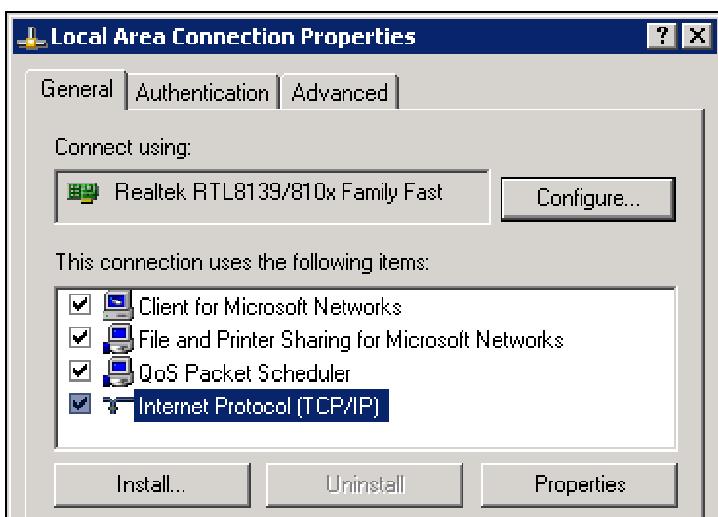
assinantes, pois trabalham sobre a perspectiva de que nem todos acessarão ao mesmo tempo.

Não é necessário ter um servidor DHCP dedicado. Muito pelo contrário, o DHCP é um serviço que consome poucos recursos do sistema, por isso o mais comum é deixá-lo ativo no próprio servidor que compartilha a conexão. Freqüentemente, o mesmo servidor incorpora também o firewall e um proxy transparente. Embora não ofereçam os mesmos recursos que um servidor Linux, os modems ADSL que podem ser configurados como roteadores quase sempre incluem a opção de ativar o servidor DHCP.

No **Windows**, a configuração de rede vai dentro do Painel de Controle > Conexões de rede, onde são listadas todas as placas instaladas. O objetivo do livro é falar sobre a configuração de servidores Linux, mas não faz mal fazer uma revisão rápida da configuração de máquinas Windows, usadas como clientes.

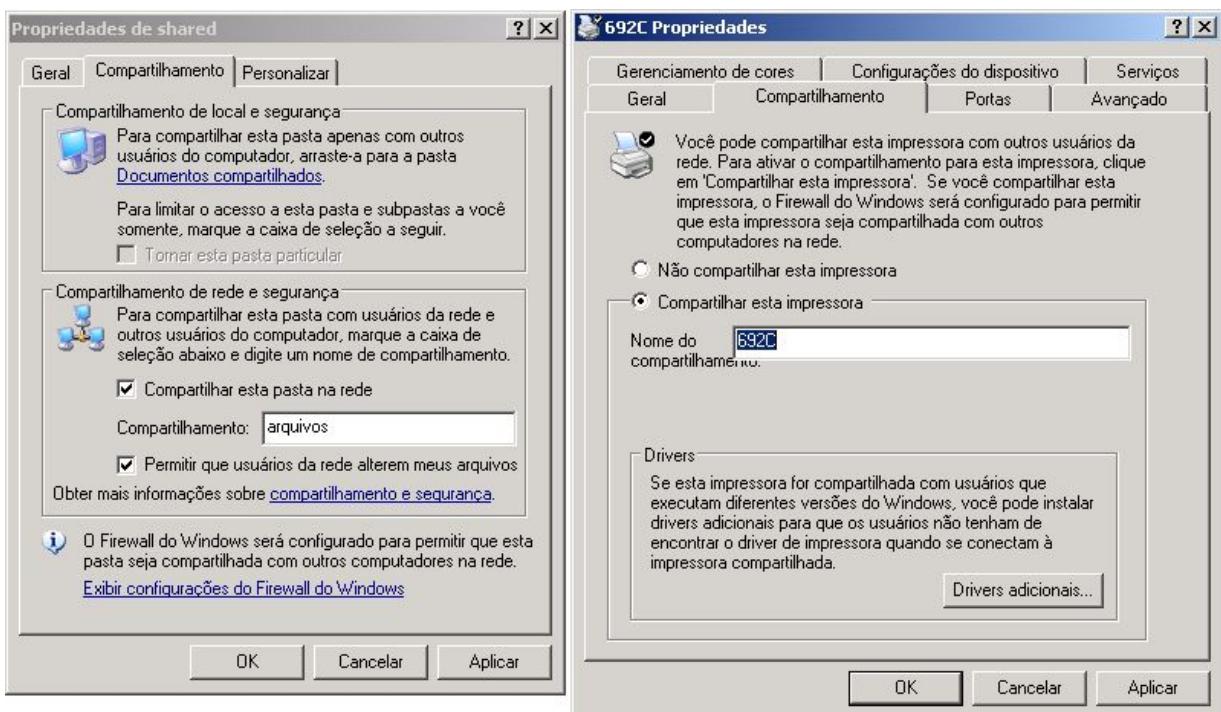


Dentro das propriedades de cada interface, vai uma lista dos protocolos disponíveis. As opções "Cliente para redes Microsoft" e "Compartilhamento de arquivos e impressoras para redes Microsoft" ativam o compartilhamento e acesso a compartilhamentos de redes em outros micros Windows, usando o protocolo SMB. As máquinas Linux também podem participar, usando o Samba, que aprenderemos a configurar no capítulo 6. Além de compartilhar arquivos e acessar compartilhamentos em outros micros, o servidor Samba pode servir como servidor de autenticação para as máquinas Windows, facilitando o gerenciamento dos logins de acesso nas máquinas Windows.

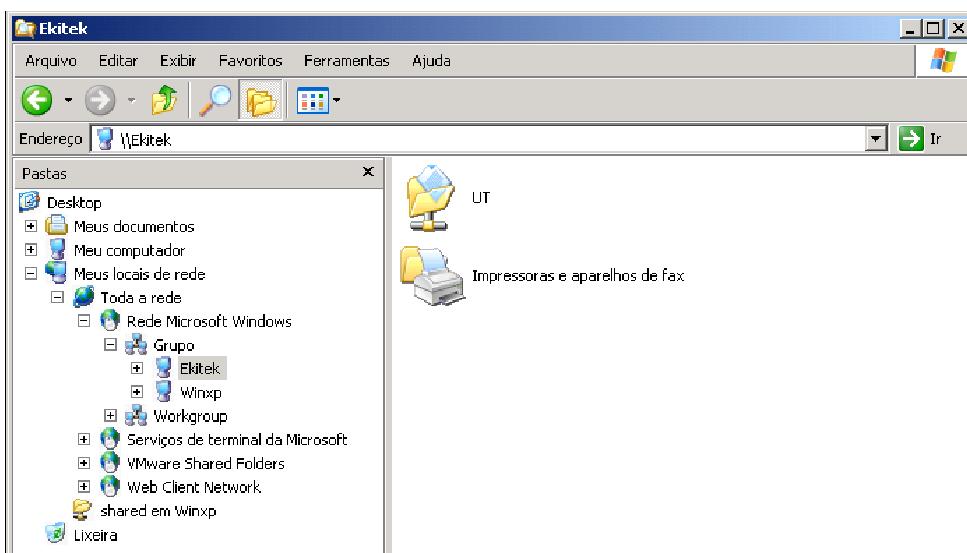


Mantendo o "Compartilhamento de arquivos e impressoras para redes Microsoft" ativo, você pode compartilhar pastas clicando com o botão direito sobre elas e acessando a opção "Compartilhamento e segurança". Marque a opção "Compartilhar esta pasta na rede" e, opcionalmente, a opção "Permitir que usuários da rede alterem meus arquivos" para tornar o compartilhamento leitura e escrita. Por padrão, o Windows XP utiliza uma pasta chamada "Arquivos compartilhados", que é a única compartilhada por padrão. Para compartilhar outras pastas, você precisa primeiro clicar sobre o link "Se você entende os riscos de segurança, mas deseja compartilhar arquivos sem executar o assistente, clique aqui", dentro da aba de compartilhamento.

O mesmo vale para as impressoras instaladas, que você pode compartilhar através do "Painel de Controle > Impressoras". Clique com o botão direito sobre ela e accesse a opção "Compartilhamento":

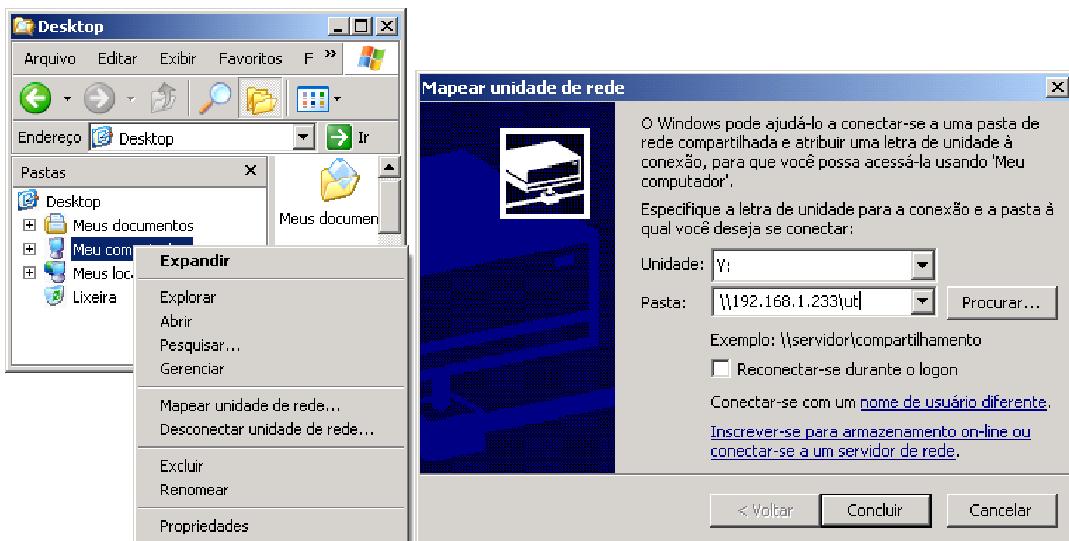


Você pode visualizar e acessar os compartilhamentos disponíveis nas outras máquinas da rede através do menu "Meus locais de rede", dentro do próprio Windows Explorer, o famoso "Ambiente de rede":

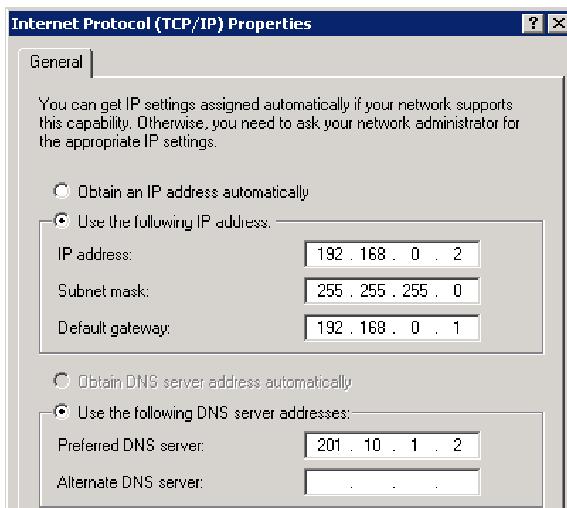


A navegação em redes Windows é um recurso que depende fortemente do envio de pacotes de broadcast e da figura do "Master Browser", uma das máquinas da rede, eleita com a função de colocar "ordem na casa", localizando os compartilhamentos e entregando a lista para as demais. Em resumo, existem muitas coisas que podem dar errado, fazendo com que novos compartilhamentos demorem para aparecer, ou que micros configurados para usar diferentes grupos de trabalho (porém na mesma rede) não se enxerguem.

Nesses casos, você pode mapear o compartilhamento manualmente. Ainda dentro do Windows Explorer, clique com o botão direito sobre o "Meu Computador" e accesse a opção "Mapear unidade de rede". Na tela seguinte, escolha uma letra para a unidade e indique o endereço IP, ou nome do servidor, seguido pelo nome do compartilhamento, como em "\\\192.168.1.233\ut". Note que você usa duas barras invertidas antes do nome do servidor e mais uma barra antes do nome do compartilhamento. Ao acessar um servidor que fica ligado continuamente, você pode marcar a opção "Reconectar-se durante o logon", o que torna o mapeamento permanente:



Voltando às propriedades da conexão, a configuração da rede vai dentro das propriedades do protocolo TCP/IP, onde você pode escolher entre ativar o cliente DHCP ou configurar manualmente os endereços. O segundo servidor DNS é desejável pela questão da redundância, mas não é obrigatório dentro da configuração:



Ao configurar a rede via DHCP, você pode checar rapidamente qual endereço IP está sendo usado por cada micro usando o comando "ipconfig" dentro do prompt do MS-DOS:

```
C:\> ipconfig
Windows IP Configuration

Ethernet adapter Local Area Connection:
  Connection-specific DNS Suffix . :
  IP Address. . . . . : 192.168.1.45
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.1.1

Ethernet adapter Wireless Network Connection:
  Connection-specific DNS Suffix . :
  IP Address. . . . . : 192.168.1.77
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.1.254
```

Uma curiosidade é que, no Windows XP, você pode também configurar a rede via linha de comando usando o comando "netsh". Na prática, não existe nenhuma grande vantagem sobre configurar pelo Painel de controle, mas não deixa de ser um truque interessante.

Para configurar a rede, especificando manualmente os endereços, você usaria:

```
C:\> netsh int ip set address name="Conexão Local" source=static 192.168.0.22
255.255.255.0 192.168.0.1 1
```

... onde o "Conexão Local" é o nome da conexão de rede (da forma como aparece no painel de Conexões de rede do Painel de controle), seguido pelo endereço IP, máscara e gateway da rede. Não se esqueça do número "1" no final, que é um parâmetro para a configuração do gateway.

Para configurar o DNS, você usaria:

```
C:\> netsh int ip set dns "Conexão Local" static 200.204.0.10
```

Para configurar os endereços e DNS via DHCP, você pode usar os comandos:

```
C:\> netsh int ip set address name="Conexão Local" source=dhcp
C:\> netsh int ip set dns "Conexão Local" dhcp
```

Como vimos, o endereço obtido via DHCP precisa ser renovado periodicamente, o que é feito de forma automática. Mas, em algumas situações, o sistema pode falhar em renovar o endereço (o que é relativamente comum ao acessar via cabo, por exemplo) fazendo com que seu micro seja desconectado da rede. Nestes casos, você pode forçar a renovação do endereço IP clicando com o botão direito sobre o ícone da conexão, dentro do painel de controle e acessando a opção "Reparar", ou usando os dois comandos abaixo no prompt do MS-DOS:

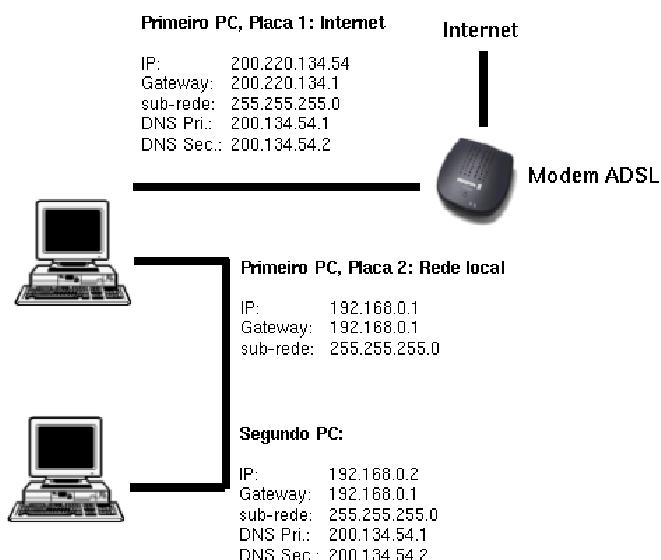
```
C:\> ipconfig /release
C:\> ipconfig /renew
```

Um exemplo de configuração de rede completa para um dos micros da rede, que vai acessar a internet através do micro que está compartilhando a conexão, seria:

IP:	192.168.0.2
Máscara:	255.255.255.0
Gateway:	192.168.0.1 (o endereço do micro compartilhando a conexão)
DNS:	200.204.0.10, 200.204.0.138

O micro que está compartilhando a conexão, por sua vez, vai ter duas placas de rede, uma para a internet e outra para a rede local, por isso vai ter uma configuração separada para cada uma. A configuração da internet é feita da forma normal, de acordo com o tipo de conexão que você usa, enquanto a configuração da rede interna segue o padrão que vimos até aqui.

Neste exemplo, estou usando dois endereços de servidores DNS externos na configuração do cliente, mas é possível instalar um servidor DNS na máquina que está compartilhando a conexão, incluindo inclusive nomes para as máquinas da rede local (como veremos no capítulo 7). Neste caso, você pode usar o endereço do gateway também como DNS.



Note que, neste caso, os micros da rede local utilizam uma faixa de endereços privada (192.168.0.x no exemplo), uma faixa de endereços que não existe na internet. O único que possui um endereço IP válido na internet é o roteador, que por isso é o único que pode ser acessado diretamente de fora. Ele fica responsável por interligar as duas redes, permitindo que os micros da rede interna acessem a internet.

Este método de compartilhamento de conexão é chamado de "NAT" (Network Address Translation). Ao receber um pacote de um dos micros da rede local endereçado à internet, o servidor substitui o endereço da estação (192.168.0.2, por exemplo) pelo seu endereço de internet (200.220.134.54, por exemplo) e o envia ao destinatário. Ao receber resposta, o servidor novamente troca o endereço de internet do destinatário pelo seu (do servidor) IP de

rede local. A estação acha que está conversando diretamente com o servidor e não enxerga os demais hosts da internet, enquanto eles (os demais hosts) enxergam apenas seu servidor e não os demais micros da rede local, que permanecem invisíveis.

Ao usar uma máquina XP com duas ou mais conexões de rede, é possível ainda criar uma ponte (bridge connection) dentre elas, permitindo que os micros conectados a cada uma das duas interfaces se enxerguem mutuamente.

Imagine uma situação onde você tenha três micros e precisa configurar rapidamente uma rede entre eles para jogar uma rodada de Doom 3, sem usar um switch. Se um dos micros tiver duas placas de rede (mesmo que seja uma placa cabeada e uma placa wireless), você pode usar cabos cross-over ou conexões wireless add-hoc para ligar os outros dois micros a ele. Inicialmente, o micro com as duas placas enxergaria os outros dois, mas os dois não se enxergariam mutuamente. A ponte resolve este problema, permitindo que os três se enxerguem e façam parte da mesma rede.

Para ativá-la, selecione as duas placas com o mouse, clique com o botão direito e acesse a opção "Conexões em ponte".



No caso das máquinas **Linux**, o utilitário de configuração de rede muda de acordo com a distribuição usada:

Um dos mais populares é o "**network-config**", que é usado por padrão no Ubuntu e diversas outras distribuições que utilizam o Gnome como interface padrão, onde ele fica disponível no menu "Sistema > Administração > Rede". Ele está disponível também no Kurumin (Iniciar > Sistema > Gnome System Tools > Configuração da rede) e pode ser usado em outras distribuições derivadas do Debian, que não o incluem por padrão, através da instalação do pacote "gnome-system-tools".

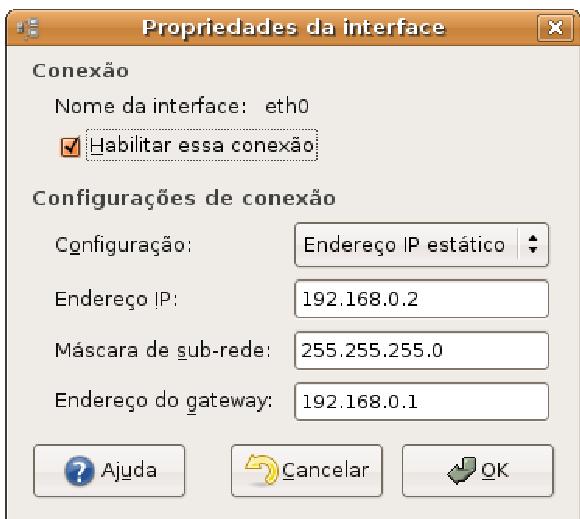
Ao ser aberto, ele mostra as interfaces disponíveis na sua máquina. Na aba "geral" você configura o nome da máquina e na aba "DNS" você define os endereços dos servidores DNS, que são "de uso comum", usados por qual seja a interface onde está a conexão com a web.

Em caso de um micro com duas ou mais placas, como no caso de um notebook com uma placa cabeada e uma placa wireless, ou no caso de um servidor compartilhando a conexão, você precisa definir qual delas é a interface com a conexão com a internet, através da opção "Dispositivo padrão de gateway":

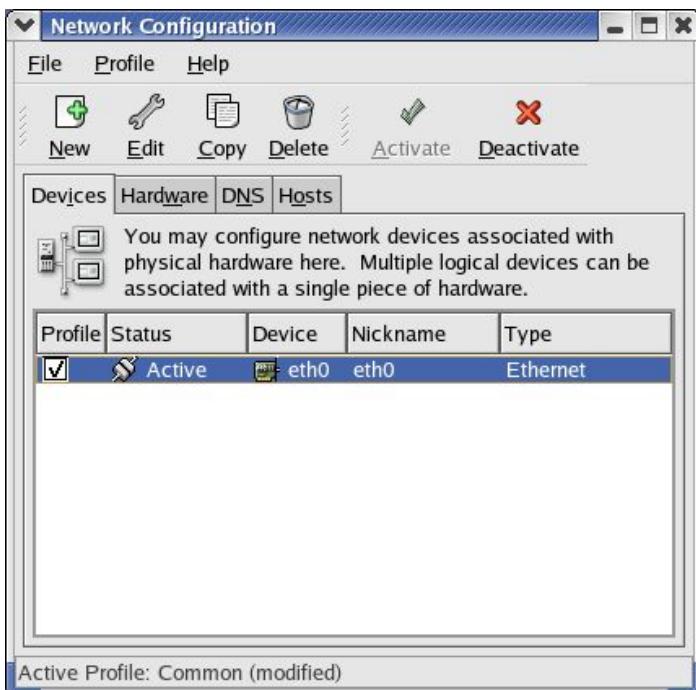


Na aba "Máquinas" você pode definir "apelidos" para as outras máquinas da rede, relacionando seus nomes a endereços IP. Isso permite que você digite algo como "ssh server" ao invés de "ssh 192.168.0.1" para acessar a máquina via SSH, por exemplo. Esta opção equivale à edição do arquivo "/etc/hosts", que você encontra em qualquer distribuição.

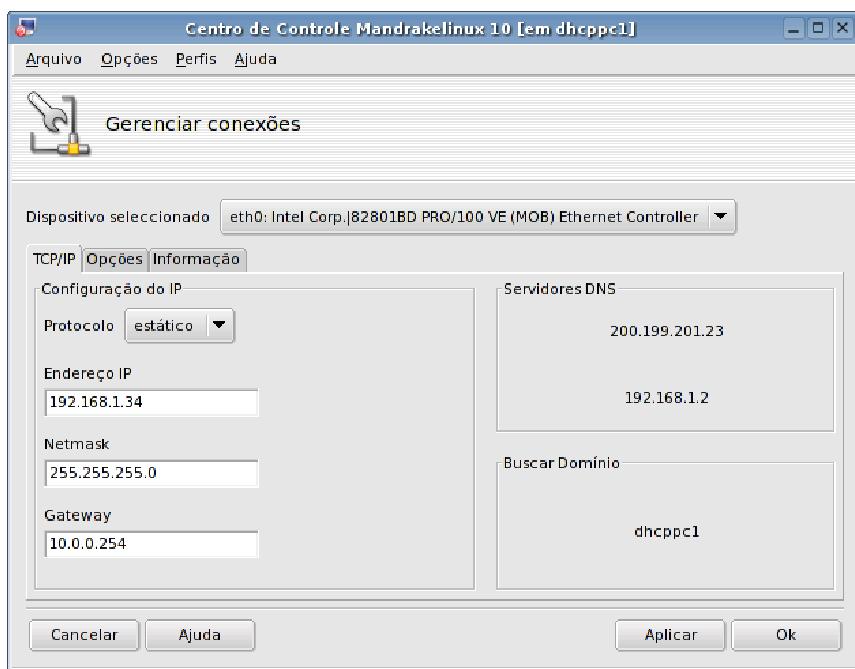
Clicando nas propriedades de cada interface, você cai no menu de configuração, onde pode definir os endereços ou ativar a configuração via DHCP:



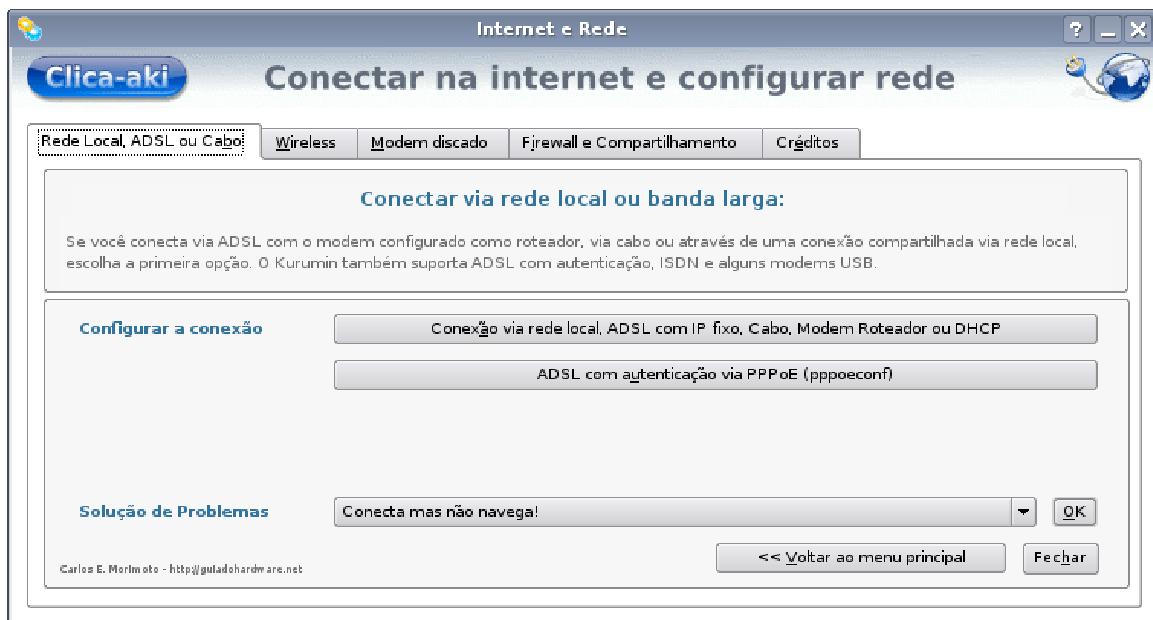
O Fedora inclui o "**system-config-network**", que pode ser chamado através do menu, ou diretamente via linha de comando. Ele é um "primo" do network-admin, que utiliza uma interface um pouco diferente, mas oferece as mesmas opções:



No Mandriva use os utilitários disponíveis na seção "Rede e internet" do Painel de Controle. Clique no "Gerenciar conexões" para definir os endereços da rede ou configurar o sistema para usar DHCP (na opção "protocolo"). Na opção "Acesso à internet" vão os endereços dos servidores DNS do provedor, caso esteja configurando a rede manualmente.



No caso do Kurumin, além do "network-config", você pode contar com um conjunto de scripts adicionais, disponíveis na seção "Conectar na Internet ou configurar a rede", dentro do painel de controle:

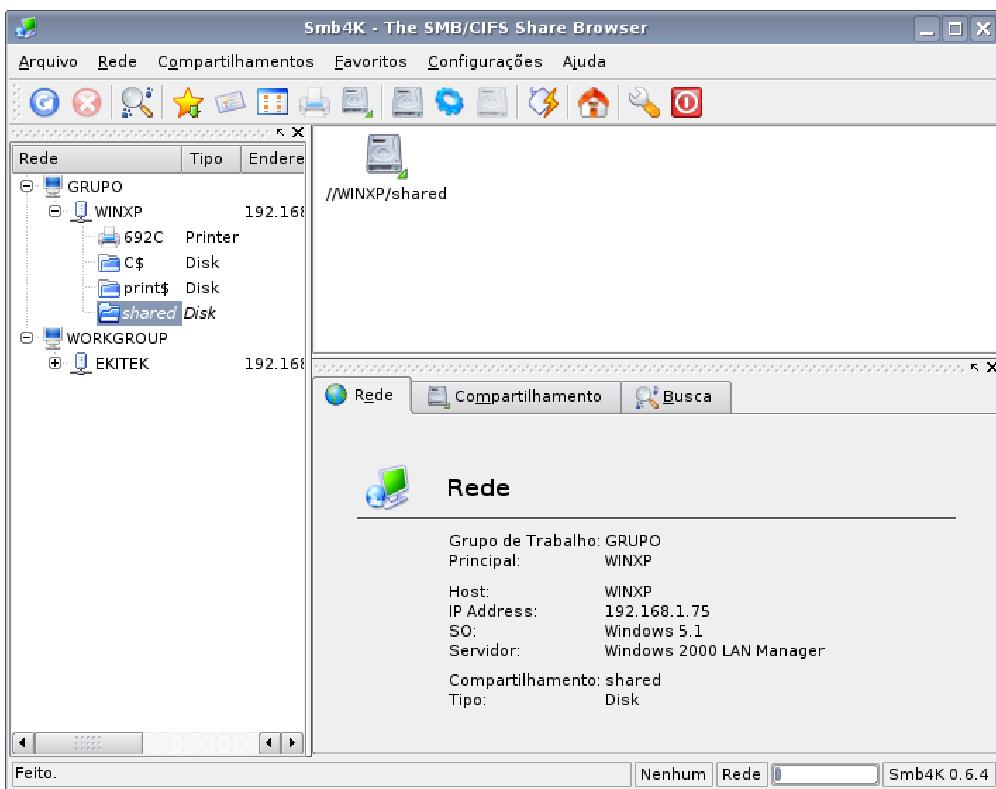


O script responsável pela configuração da rede é o "netcardconfig", um assistente que reúne as informações necessárias e atualiza a configuração do sistema:

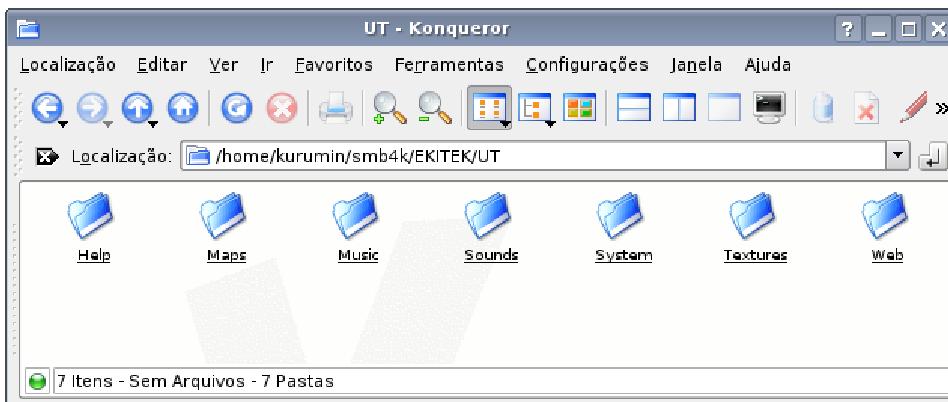


Esta configuração de IP, máscara, gateway e DNS vale tanto para redes cabeadas, quanto para redes wireless. A diferença é que as redes wireless possuem alguns parâmetros adicionais, que são necessários para estabelecer a conexão com o ponto de acesso. Só depois que a conexão é estabelecida, passamos para a configuração dos endereços.

Temos ainda a questão do acesso a compartilhamentos da rede a partir das máquinas Linux. Naturalmente, você pode também acessar compartilhamentos de rede, tanto em máquinas Windows, quanto em outras máquinas Linux com o servidor Samba ativo. Um dos programas mais usados é o "**Smb4k**", que vem instalado por padrão em diversas distribuições. Ao ser aberto, ele mostra os grupos de trabalho disponíveis na rede e, dentro de cada um, os servidores e compartilhamentos. Ao clicar sobre um compartilhamento que exige autenticação, ele abre um prompt de login. Os compartilhamentos acessados aparecem no menu da direita. Clicando sobre eles você abre uma tela do gerenciador de arquivos.

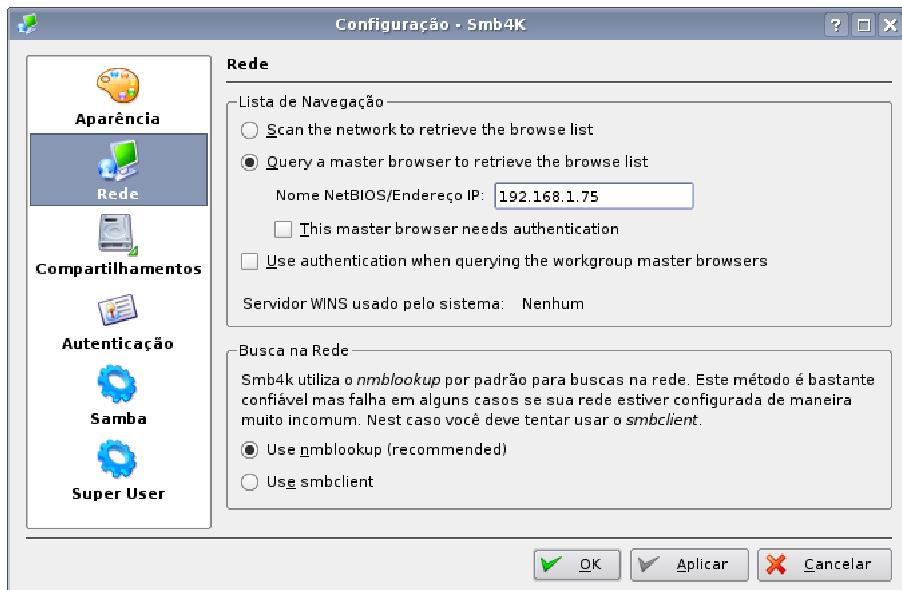


Os compartilhamentos acessados através do Smb4k são, na verdade, montados dentro da pasta "smb4k", dentro do seu diretório home. Ele são organizados em uma estrutura de pastas, com uma pasta separada para os compartilhamentos de cada servidor. Note que quando falo em "servidor" me refiro a qualquer máquina da rede que esteja compartilhando arquivos.



Em casos onde o Smb4k não consiga mostrar corretamente os compartilhamentos, ou a navegação fique instável, você pode indicar manualmente o endereço IP de uma máquina Windows, ou servidor Samba de onde ele obterá a lista dos compartilhamentos. Acesse o "Configurações > Configurar Smb4k > Rede" e indique o servidor na opção "Query a master browser to retrieve the browse list". Na opção "Compartilhamentos", você pode marcar a opção "Remount recently used shares on program start", que faz com que ele

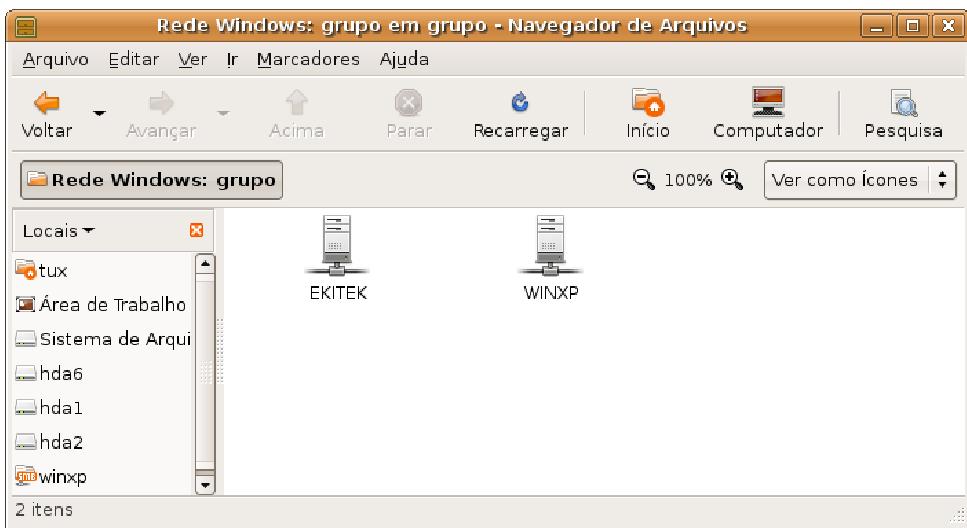
"lembre" dos compartilhamentos acessados e restaure o acesso a eles a cada abertura do programa, mesmo depois de reiniciar a máquina.



Outra opção para acessar os compartilhamentos é usar o módulo "smb://" do Konqueror. Abra uma janela do gerenciador de arquivos e digite "smb://servidor" (onde o "servidor" pode ser o endereço IP, ou o nome do servidor Windows ou Samba dentro da rede) para ver os compartilhamentos disponíveis. Você pode digitar também "smb://grupo" (onde "grupo" é o nome do grupo de trabalho) ou mesmo apenas "smb://" (neste caso com uma única barra) para que ele tente mostrar toda a rede:



No Ubuntu você pode usar o módulo de acesso a compartilhamentos do Nautilus, disponível no menu "Locais > Servidores de Rede" (no menu do topo da tela). Ao ser aberto, ele mostra os servidores e compartilhamentos disponíveis dentro do ícone "Rede Windows". Você pode também especificar compartilhamentos manualmente usando a opção "Locais > Conectar ao Servidor":



» Próximo: [Configurando manualmente no Linux](#)

Como vimos, existem diversas ferramentas gráficas de configuração da rede, que você pode usar de acordo com a distribuição. Mesmo assim, nenhuma ferramenta é completamente à prova de falhas. Quanto mais automática é a ferramenta de detecção, maior é a possibilidade de erros.

Outro dia, por exemplo, tive um problema com uma instalação do Slackware. Apesar da placa de rede PCMCIA ter sido automaticamente detectada pelo hotplug, o **netconfig** (o script de configuração de rede incluído no Slackware) não estava conseguindo configurar a rede corretamente. Em casos como este, você pode apelar para a configuração manual da rede, um método que funciona em qualquer distribuição.

A configuração da rede envolve, basicamente, três passos:

- 1-** Carregar o módulo correto para a placa de rede e certificar-se de que o sistema o utilizou para habilitar a interface de rede.
- 2-** Configurar o IP, máscara e as demais configurações da rede, usando o **ifconfig**.
- 3-** Configurar a rota padrão e colocar o DNS do provedor no arquivo "**/etc/resolv.conf**".

Tudo isso pode ser feito diretamente através de comandos de terminal. Depois de testar a configuração você pode torná-la definitiva, adicionando os mesmos comandos a um dos arquivos de inicialização do sistema. Esta dica pode ser usada em qualquer distribuição, sempre que as ferramentas de configuração falharem ou você estiver em busca de aventura.

Em primeiro lugar, verifique se o módulo que habilita o suporte à placa de rede está carregado. Use o comando **lsmod**:

Module	Size	Used by	Tainted:	PF
				(unused)
snd	27716	0		
i830		69248		1
agpgart		38296		11
i810_audio		25064		0
ac97_codec	11884	0	[i810_audio]	
soundcore	3428	2 [snd]	i810_audio]	
8139too		27500		1
serial		51972		0
mousedev		3832		1
ds		6536		1
yenta_socket		9408		1
pcmcia_core	39712	0 [ds]	yenta_socket]	
rtc	6908	0		

No meu caso a placa é uma Encore, com o chipset **Realtek 8139**, o módulo que habilita suporte a ela (o **8139too**) está carregado, mas ainda assim a rede não está funcionando. Outros módulos usados por placas comuns são o "via-rhine", "e100" e o "sis900".

O próximo passo é configurar o arquivo **"/etc/modules.conf"**, para ter certeza de que o módulo está sendo usado para habilitar a interface de rede. Se você tem apenas uma placa de rede (cabecada), ela será sempre a "eth0". Placas wireless podem receber outros nomes, de acordo com o driver usado.

Abra o arquivo **"/etc/modules.conf"** e adicione a linha:

alias eth0 8139too

... trocando o "8139too" pelo módulo usado pela sua placa. Caso você tenha duas placas de rede que utilizem módulos diferentes, você pode usar o mesmo arquivo para indicar manualmente como cada uma será vista pelo sistema, como em:

alias	eth0	8139too
alias eth1 sis900		

Isso pode ser usado em casos em que o sistema troca a posição das placas de rede (a eth0 passa a ser a eth1 e vice-versa) a cada boot. Caso o módulo da placa não estivesse carregado, você poderia ativá-lo manualmente usando o comando **"modprobe"**, como em:

modprobe 8139too

Em seguida, falta fazer a configuração da rede. A melhor opção para fazer a configuração manualmente é usar o **ifconfig**, como em:

ifconfig eth0 192.168.0.10 netmask 255.255.255.0 up

Este comando configura o endereço IP e a máscara de sub-rede. O "up" serve para ativar a interface de rede especificada, a "eth0", no exemplo. O passo seguinte é definir a rota padrão, ou seja, o gateway da rede e a interface que será usada para contatá-lo. Por

segurança, rodamos primeiro o comando "route del default", que desativa qualquer configuração anterior:

```
#          route      del      default
# route add default gw 192.168.0.1 dev eth0
```

... onde o "192.168.0.1" é o gateway da rede e a "eth0" é a placa conectada a ele. Estes mesmos dois comandos resolvem casos em que o micro tem duas placas de rede, ou uma placa de rede e um modem e o sistema tenta acessar a internet usando a placa errada.

Verifique também se o arquivo "/etc/resolv.conf" contém os endereços DNS do provedor, como em:

```
nameserver      200.204.0.10
nameserver 200.219.150.5
```

A falta dos endereços no "/etc/resolv.conf" é, provavelmente, a causa mais comum de problemas com a navegação.

Para que estes comandos sejam executados durante o boot, restaurando a configuração automaticamente, coloque-os no final do arquivo "/etc/init.d/bootmisc.sh", no caso do Kurumin ou outros derivados do Debian; ou no arquivo "/etc/rc.d/rc.local", no caso das distribuições derivadas do Red Hat, como em:

```
modprobe
ifconfig      eth0      192.168.0.10      netmask      255.255.255.0      8139too
route        del
route add default gw 192.168.0.1 dev eth0      up
                                         default
```

Caso você esteja configurando um servidor com várias placas de rede (cada uma ligada a um hub diferente, com a rede dividida em várias sub-redes com faixas de endereços IP diferentes) e esteja tendo problemas para explicar para o sistema qual placa usar para cada faixa, você pode novamente usar o comando **route**, especificando as faixas de endereço usadas e a placa responsável por cada uma, como em:

```
#   route   add   -net   192.168.1.0   netmask   255.255.255.0   eth1
#   route   add   -net   192.168.2.0   netmask   255.255.255.0   eth2
#   route   add   -net   192.168.3.0   netmask   255.255.255.0   eth3
# route add default eth0
```

Neste caso estamos dizendo que o sistema tem 4 placas de rede instaladas: eth0, eth1, eth2 e eth3, sendo que a conexão com a web está ligada na eth0 e as outras 3 são placas ligadas a três redes diferentes (um hub ou switch separado para cada placa, formando três redes locais separadas) que usam as faixas de IP's 192.168.1.x, 192.168.2.x e 192.168.3.x, as três com a máscara de sub-rede: 255.255.255.0. Estas linhas também podem ser incluídas no script de configuração da rede.

Basicamente, estamos dizendo:

Quando mandar alguma coisa para um micro na rede 192.168.1.x use a interface eth1.
Quando mandar alguma coisa para um micro na rede 192.168.2.x use a interface eth2.
Quando mandar alguma coisa para um micro na rede 192.168.3.x use a interface eth3.
Quando mandar alguma coisa para qualquer outra faixa de endereços, ou para a internet, use a interface eth0.

» Próximo: [Configurando uma rede wireless](#)

Em uma rede wireless, o dispositivo central é o access point (ponto de acesso). Ele é ligado no hub da rede, ou diretamente no modem ADSL ou cable modem, e se encarrega de distribuir o sinal para os clientes.

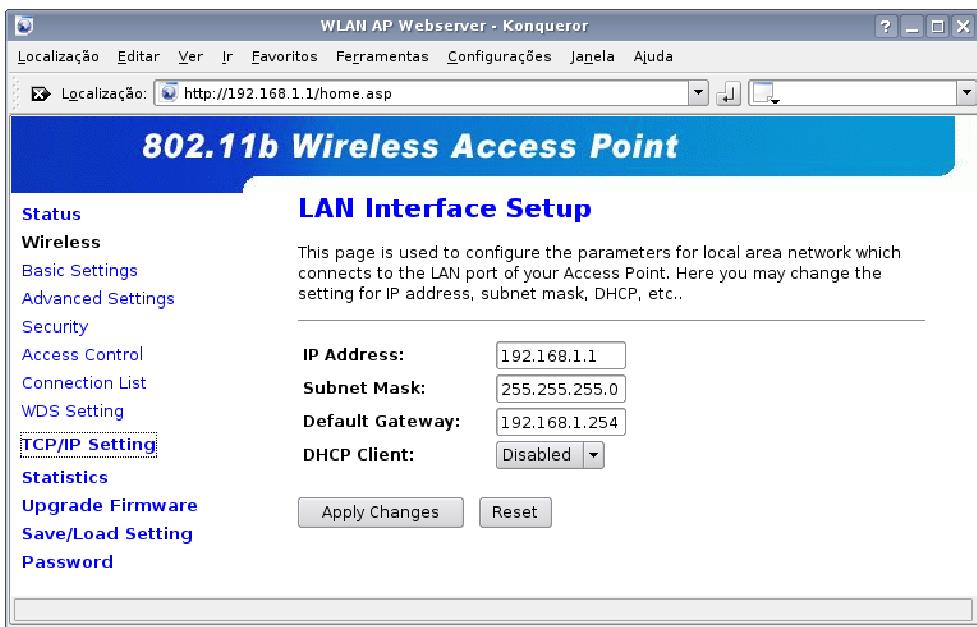
Ao contrário de um hub, que é um dispositivo "burro", que trabalha apenas no nível físico e dispensa configuração, o access point possui sempre uma interface de configuração (similar à dos modems ADSL), que pode ser acessada via navegador, a partir de qualquer um dos micros da rede. Verifique no manual qual é o endereço e a senha padrão do seu.

Se o endereço padrão do ponto de acesso usar uma faixa diferente da usada na sua rede, é necessário configurar seu micro para usar temporariamente um endereço qualquer dentro da mesma faixa que ele, de forma que os dois passam enxergar.

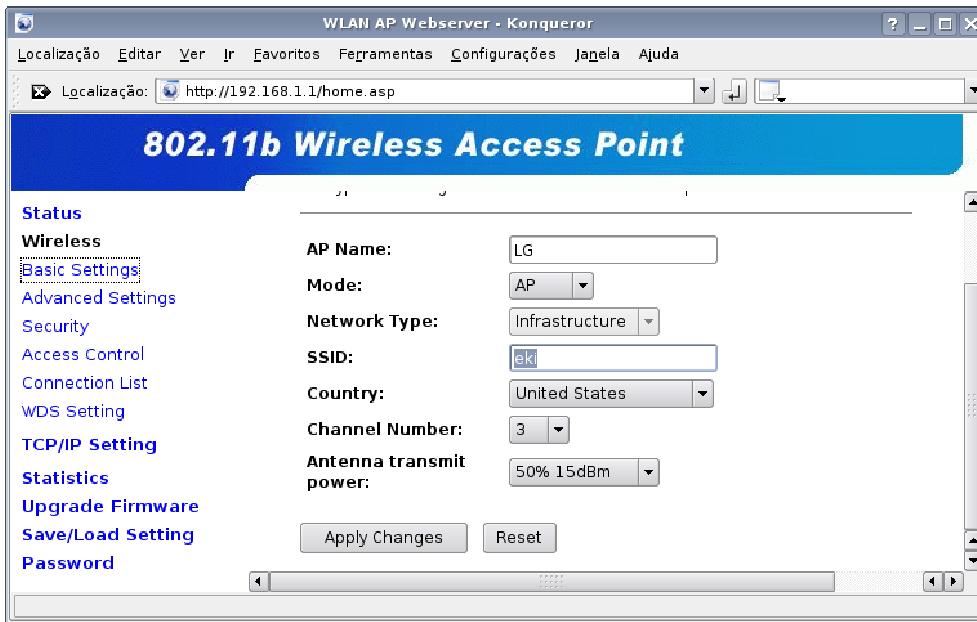
Ao usar uma estação Linux, uma solução simples é criar um alias para a placa de rede. Isso permite que você acesse a configuração do modem sem precisar alterar sua configuração de rede. Se o endereço default do AP for 192.168.1.100, por exemplo (como o padrão de alguns dos modelos da LG), configure seu micro para usar um endereço dentro da faixa "192.168.1.x", como em "192.168.1.2". Se a sua placa é a "eth0", o comando seria:

ifconfig eth0:1 192.168.1.2

Depois de acessar da primeira vez, aproveite para definir uma senha de acesso e alterar o endereço padrão por um dentro da faixa de endereços IP usada na sua rede. A partir daí, ele se integra à sua rede local e você não precisa mais usar o truque de criar o alias para acessá-lo.



O próximo passo é configurar os parâmetros da rede wireless, incluindo o SSID da rede e o canal usado. Neste AP da LG, elas estão dentro do menu "Basic Settings":



O SSID (ou ESSID) é o "nome" da rede wireless. Nomes diferentes fazem com que diferentes pontos de acesso dentro da mesma área de cobertura entendam que fazem parte de redes diferentes. Todo ponto de acesso vem com um SSID padrão, mas é importante modificá-lo, pois o nome default é geralmente o mesmo em todos os APs do fabricante.

O ponto de acesso pode trabalhar em dois modos diferentes. No modo "AP", ele desempenha suas funções normais de ponto de acesso, dando acesso aos micros e notebooks com placas wireless. Já no modo "client", ele se comporta como se fosse uma

placa de rede, conectando-se a outro ponto de acesso. Como os pontos de acesso mais baratos custam muitas vezes quase o mesmo que uma (boa) placa wireless, muita gente prefere usar um ponto de acesso configurado como cliente em seu desktop, ao invés de comprar uma placa PCI. A vantagem, no caso, é que a instalação é mais simples (já que basta ligar na placa de rede) e a qualidade de recepção é melhor, pois o ponto de acesso cliente pode ficar sobre a mesa ou próximo da janela, em uma posição com menos obstáculos atenuando o sinal.

A lista dos canais disponíveis varia de acordo com a configuração de país no ponto de acesso. Em teoria, podem ser usados 17 canais, de 0 a 16. Porém, apenas 14 deles (de 1 a 14), são licenciados pelo FCC e a lista diminui mais um pouco de acordo com o país escolhido. Nos EUA é permitido o uso dos canais de 1 a 11; na Europa, de 1 a 13 e no Japão, de 1 a 14. Enquanto escrevo ainda não existe legislação sobre isso no Brasil, mas é provável que seja seguido o padrão dos EUA.

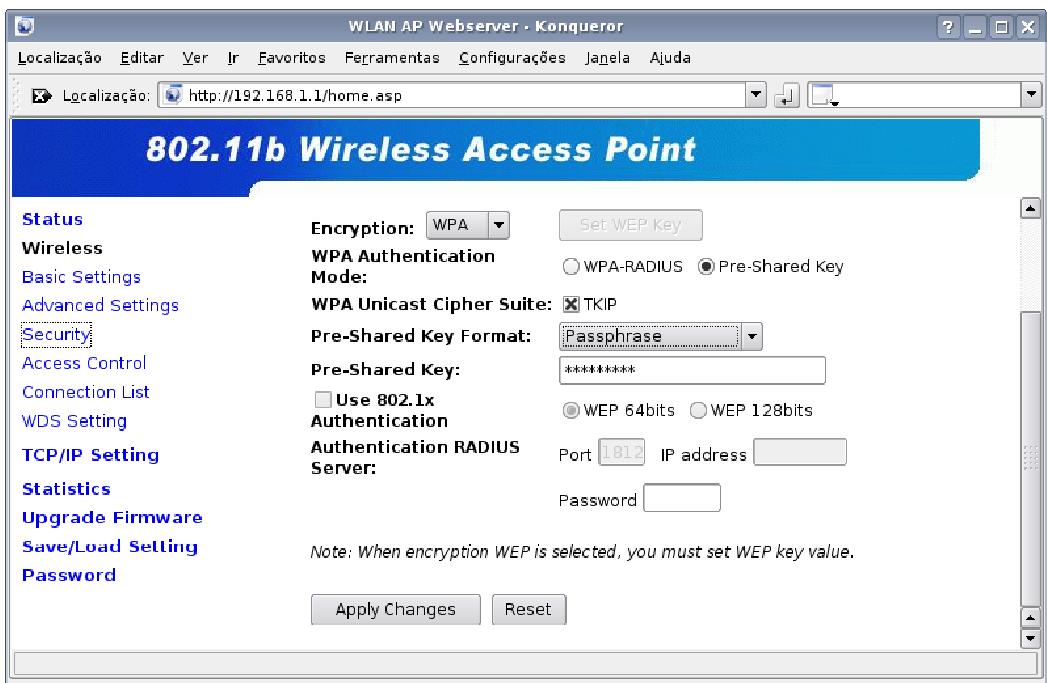
Usar canais diferentes é uma forma de minimizar interferências caso você esteja colocando vários pontos de acesso dentro da mesma área, ou perceba que existem pontos de acesso de vizinhos, muito próximos do seu. No capítulo 4 veremos como é possível escanear as redes próximas, usando o Kismet, e descobrir quais canais estão ocupados.

Existe uma diferença de freqüência de apenas 5 MHz entre cada canal, porém o sinal das placas 802.11b ocupa uma faixa de 30 MHz. Por isso, para que realmente não exista possibilidade de interferência entre dois pontos de acesso próximos, é preciso usar canais distantes, como, por exemplo, 1, 6 e 11 ou 1, 7 e 14. De qualquer forma, a moral da história é que, independentemente do canal usado, é preciso usar o mesmo tanto na configuração do ponto de acesso quanto na configuração dos clientes para que a rede funcione.

Uma última configuração, disponível na maioria dos pontos de acesso, é o ajuste de potência da antena. Em situações em que o alcance da rede é importante, você naturalmente deixaria a potência de transmissão no máximo ou, quem sabe, até substituisse a antena padrão, por uma de maior ganho. Mas, em situações onde você quer que a rede fique disponível apenas em uma área restrita, reduzir a potência é uma boa opção de segurança.

Hoje em dia, cada vez mais gente utiliza placas wireless e nada melhor do que usá-la para acessar a web de graça. Qualquer rede aberta acaba sendo rapidamente descoberta e usada pelos freeloaders. Pode ser que você seja uma pessoa magnânima e não se importe, mas, na maioria dos casos, a idéia é proteger a rede dos intrusos.

Entra aí a necessidade de ativar um sistema de encriptação. A grande maioria dos pontos de acesso permite que você escolha entre usar o WEP de 64 bits, WEP de 128 e WPA. O WEP, mesmo de 128 bits, é fácil de quebrar (como veremos em detalhes no capítulo 4). Por isso, a única opção que realmente adiciona uma boa camada de segurança é mesmo o **WPA**, que no meu caso está disponível dentro da seção "Security":



Para uma rede doméstica, escolha a opção "Pre-Shared Key" (também chamada de WPA-PSK) com encriptação TKIP. Você deve, então, definir uma passphrase, uma espécie de senha, que serve como chave de autenticação, que deve ser fornecida na configuração dos clientes. Qualquer pessoa dentro da área de acesso que saiba a passphrase poderá se conectar à sua rede, por isso é importante que ela seja mantida em segredo e revelada apenas às pessoas autorizadas. Também é recomendável trocá-la periodicamente.

Ao usar a encriptação via WEP, temos as chaves de encriptação, que possuem a mesma função da passphrase do WPA. Você tem a opção de criar uma chave usando caracteres hexadecimais (onde temos 16 dígitos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F e cada dígito equivale a 4 bits) ou usar caracteres ASCII, onde é possível misturar letras, números e caracteres especiais.

Ao usar caracteres hexadecimais, a chave terá 10 dígitos ("123456789A", exemplo). Se a chave for em ASCII, onde cada caracter equivale a 8 bits, a chave terá apenas 5 dígitos ("qwerty", exemplo). Ao usar o WEP de 128 bits, a chave terá 26 dígitos em hexa ou 13 em ACSII.

Veja que 10 caracteres hexadecimais formam uma chave de apenas 40 bits (4 por caracter). Este é justamente o problema fundamental das chaves de 64 bits: na verdade, são duas chaves separadas, uma de 40 bits e outra de 24 bits (chamada de vetor de inicialização), que é, justamente, a parte vulnerável da chave, que permite quebrar a chave principal.

Uma chave de 64 bits sem problemas óbvios poderia oferecer uma segurança aceitável, mas uma chave de 40 bits é fraca em todos os aspectos. No caso das chaves de 128 bits, a chave de encriptação tem 104 bits, mas continua sendo usado o vetor de 24 bits, o que torna a chave vulnerável.

Outra forma de reforçar a segurança é ativar o controle de acesso baseado nos endereços MAC da placa de rede. Através desta opção ("Access Control", no meu caso), você pode definir uma lista de placas "autorizadas", declarando o endereço MAC de cada uma. Lembre-se de que você pode checar o endereço MAC de cada placa usando o comando "ifconfig" no Linux, como em:

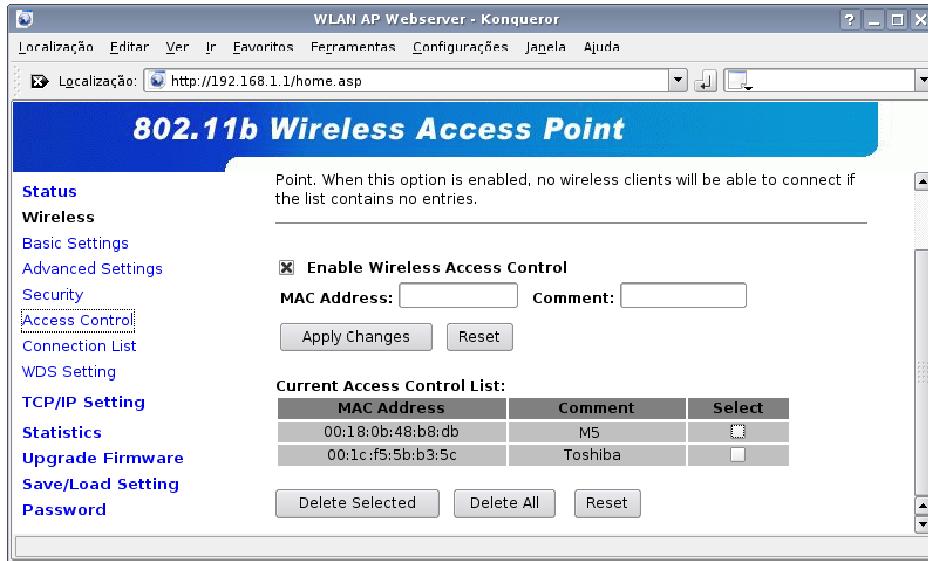
ifconfig wlan0

```
wlan0    Encapsulamento    do    Link: Ethernet      Endereço de HW    00:15:00:4B:68:DB
inet     end.:          192.168.1.12   Bcast:192.168.1.255 Masc:255.255.255.0
         endereço          inet6: fe80::215:ff:fe4b:68db/64 Escopo:Link
         UP                BROADCASTMULTICAST   MTU:1500           Métrica:1
         RX    packets:38    errors:5        dropped:12523       overruns:0        frame:0
         TX    packets:23    errors:0        dropped:0          overruns:0        carrier:0
         colisões:0
         RX    bytes:529611   (517.1      KiB)    TX    bytes:61025   (59.5      KiB)
         IRQ:3 Endereço de E/S:0x8000 Memória:fe8ff000-fe8fffff
```

O ifconfig só mostra as propriedades das redes ativas. Caso necessário, você pode ativar a placa wireless (sem precisar configurar a rede), usando o comando "ifconfig placa up", como em:

ifconfig wlan0 up

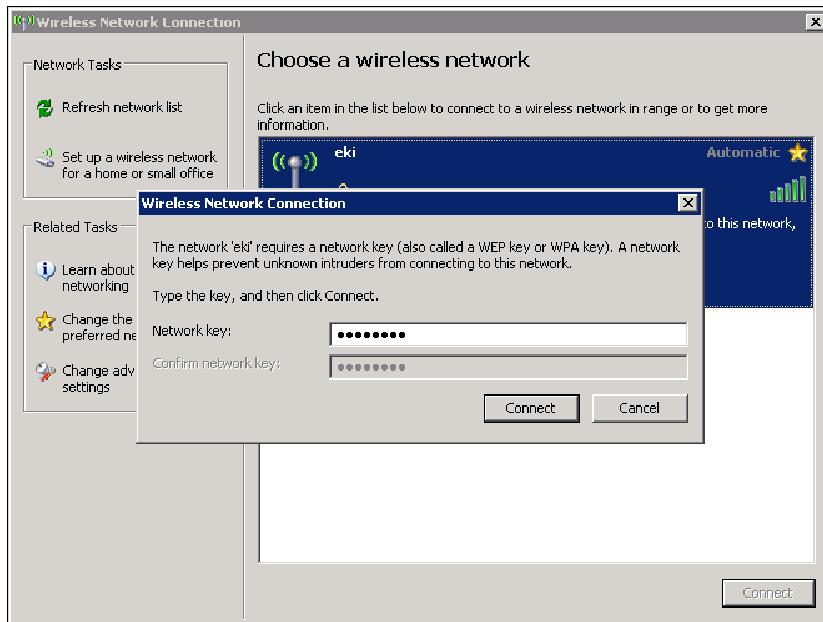
O controle de acesso dificulta o acesso à rede, mas não é uma solução por si só, pois é fácil falsear o endereço MAC da placa de rede. Ele deve ser sempre usado em conjunto com um dos sistemas de encriptação.



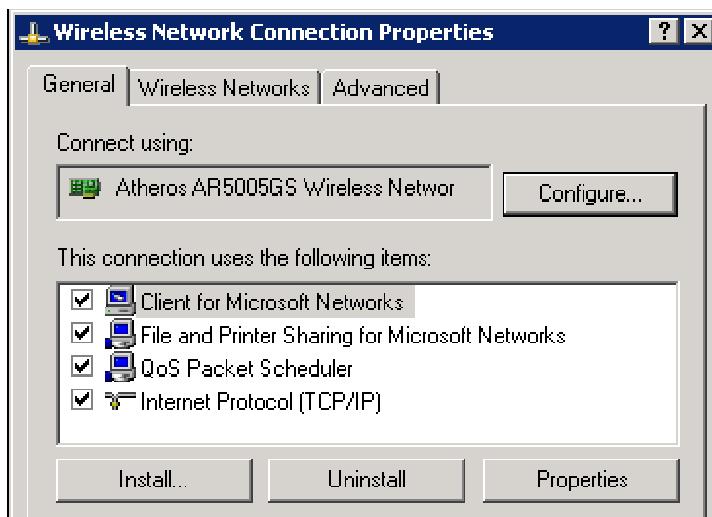
Depois da configuração do ponto de acesso, vem a configuração dos clientes que se conectarão a ele.

Nos clientes **Windows**, clique com o botão direito sobre o ícone da placa wireless dentro do "Painel de Controle > Conexões de rede". Ele mostra uma lista das redes disponíveis e a qualidade do sinal de cada uma. No caso das redes com WEP ou WPA ativado, você

precisa fornecer a chave ou passphrase para se conectar (o suporte a WPA está disponível apenas a partir do Windows XP SP2, nas versões anteriores você depende dos utilitários fornecidos pelos fabricantes):



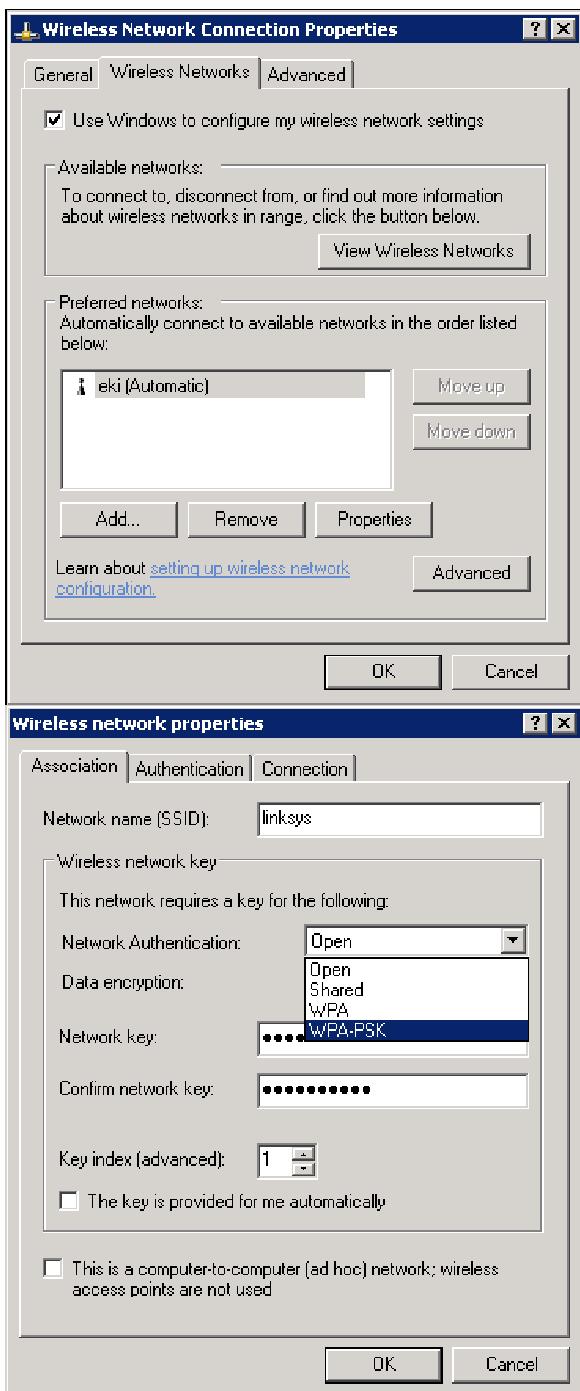
A configuração é feita em dois níveis. Primeiro é necessário se "associar" ao ponto de acesso, o que estabelece a conexão de rede e em seguida fazer a configuração normal de endereços. Por default, o Windows tenta configurar a rede via DHCP, mas você pode definir os endereços manualmente dentro da configuração do protocolo TCP/IP, nas propriedades da conexão wireless:



Na aba "Wireless Networks" você pode ajustar manualmente os parâmetros do ponto de acesso e também se conectar a redes "ocultas" (com o "SSID Broadcast" desativado no ponto de acesso), que não aparecem na varredura do assistente. Esta é outra configuração

comum em redes projetadas para serem seguras, pois muitas ferramentas de varredura (como o Netstumbler) não detectam a rede caso o ponto de acesso não divulgue o SSID.

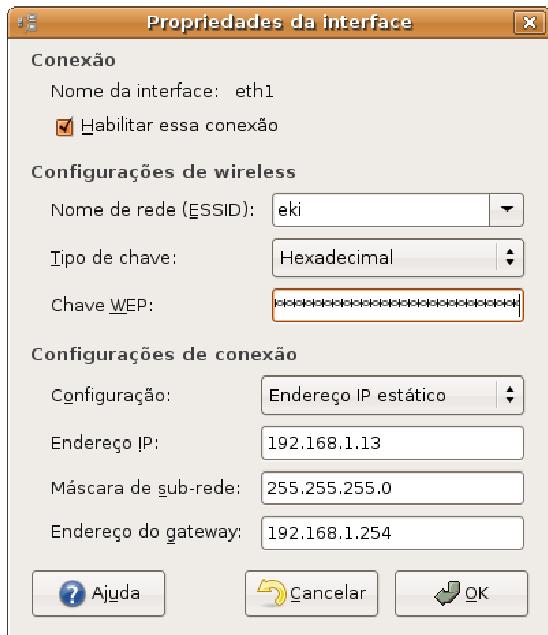
Caso você use um notebook e precise se conectar a várias redes diferentes, de acordo com o local, adicione cada uma dentro da configuração, especificando o nome, o tipo de encriptação e a chave/passphrase de acesso e estabeleça uma ordem de prioridade. Ao ligar o note, o Windows tenta se conectar à primeira rede e, quando ela não estiver disponível, tenta as outras da lista, até conseguir estabelecer a conexão com sucesso:



Para que este sistema de conexão automática funcione, é necessário que o serviço "Configuração zero sem fio" (ou "Wireless zero configuration", na versão em inglês) esteja ativo no "Painel de Controle > Ferramentas administrativas > Serviços".

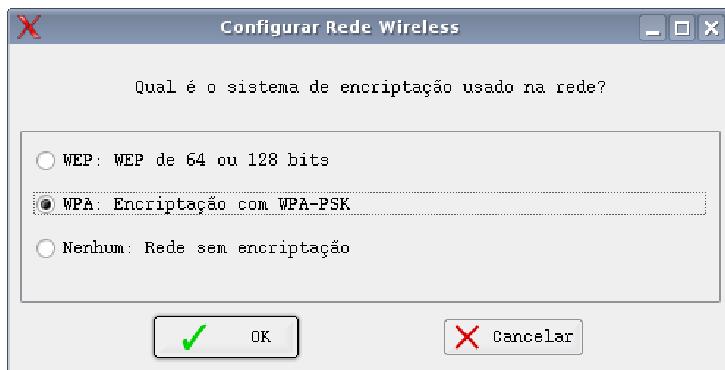
Temos em seguida a configuração dos clientes **Linux**. Se a placa já tiver sido detectada corretamente, não existem grandes dificuldades. No Ubuntu e em outras distribuições que

incluem o "network-admin", acesse as propriedades da placa wireless, onde você deve indicar o SSID da rede, o tipo de encriptação e a chave de acesso. Na mesma tela vai a configuração de IP, máscara e gateway, ou DHCP:



Uma observação importante é que o network-admin (pelo menos até o Ubuntu 6.6) ainda não oferece suporte ao WPA. Neste caso, você deve usar o `wpa_supplicant`, que aprenderemos a configurar a seguir. Ele não é particularmente difícil de usar e permite especificar diversas redes diferentes e definir uma ordem de prioridade. Ele passa, então, a testar cada uma das redes e a se conectar na primeira disponível. A principal falha é que não existe nenhuma interface gráfica utilizável para ele, de forma que a configuração ainda precisa ser feita manualmente.

Ao usar o **Kurumin**, você pode utilizar o "**wireless-config**" (Painel de Controle > Conectar na Internet e configurar rede > Wireless > Configurar os parâmetros da rede wireless), meu script de configuração que, a partir da versão 6.1, oferece suporte a WPA, servindo como uma interface para o `wpa_supplicant`:



Outra boa opção é o **Kwifimanager**, um pequeno utilitário, baseado na biblioteca QT (do KDE), que permite configurar vários perfis de redes wireless e alternar entre eles rapidamente. Ele está disponível em várias distribuições e é uma ferramenta essencial para quem carrega o notebook para cima e para baixo e utiliza redes diferentes ao longo do dia.

Hoje em dia, com exceção das redes públicas ou mantidas pelos provedores de acesso, quase todas redes wireless são protegidas, usando uma combinação de uma chave de encriptação (WEP ou WPA), controle de acesso baseado no endereço MAC das placas de rede autorizadas, uso de um canal específico e nome da rede.

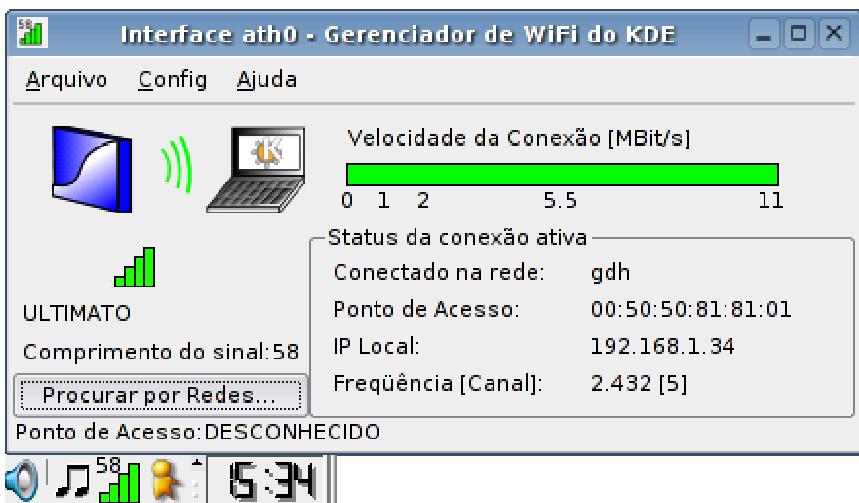
Como a questão da (ou falta de) segurança em redes wireless é muito divulgada, apenas os muito desleixados deixam suas redes desprotegidas hoje em dia. Nenhum sistema de segurança para redes wireless em uso é inquebrável, mas a combinação de várias dificuldades pode tornar sua rede difícil de invadir o suficiente para que procurem outro alvo mais fácil.

É a mesma questão do roubo de carros: você não precisa tornar seu carro impossível de roubar; precisa apenas fazer com que ele seja mais difícil de levar que os que estão estacionados ao lado. O problema é que quanto mais camadas de proteção são adicionadas, mais trabalhosa se torna a configuração da rede. O Kwifimanager facilita a configuração, permitindo juntar o melhor dos dois mundos.

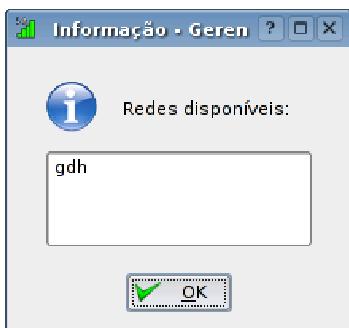
Para instalar, procure pelo pacote "**kwifimanager**" no gerenciador de pacotes da distribuição que está utilizando. Ele é atualmente bastante popular e, por isso, encontrado nas principais distribuições. Nas baseadas no Debian, por exemplo, você pode instalá-lo com um "**apt-get install kwifimanager**". Em muitas distribuições, o Kwifimanager é incluído dentro do pacote "**kdenetwork**", geralmente instalado por padrão junto com o KDE. Na pior das hipóteses, use o Google para procurar um pacote compilado.

A página oficial é a: <http://kwifimanager.sourceforge.net/>.

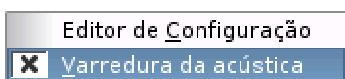
Ao ser aberto, ele automaticamente procura por redes disponíveis. Caso você já tenha configurado o sistema para se conectar a uma rede, usando outro utilitário, ele mostra os detalhes da conexão e fica residente ao lado do relógio, mostrando um gráfico com a potência do sinal.



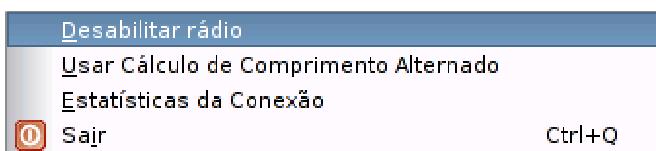
Para se conectar a uma rede, clique no "**Procurar por Redes**". Ele tem a mesma função do comando "iwlist wlan0 scan", ou seja, detectar redes públicas, onde a encriptação está desativada e o ponto de acesso divulga o ESSID da rede, ou mostrar quais das redes cujo perfil você já tenha configurado estão acessíveis no momento.



Quando o sinal da rede estiver fraco e você estiver procurando um lugar melhor para se conectar, marque a opção "Varredura acústica" no menu "Config". Ao ativar esta opção, o Kwfimanager começa a emitir um bit periódico que indica a potência do sinal: quanto mais alto o bip, melhor é a qualidade do sinal, o que permite que você se guie pelo som para sair da área com interferência.



Quase todos os notebooks possuem uma chave que permite desativar o rádio da placa wireless quando ela não estiver sendo utilizada. Isso serve tanto para economizar as baterias, quanto como uma precaução de segurança. Clicando no menu "Arquivo" do Kwfimanager você tem a opção de fazer isso via software.



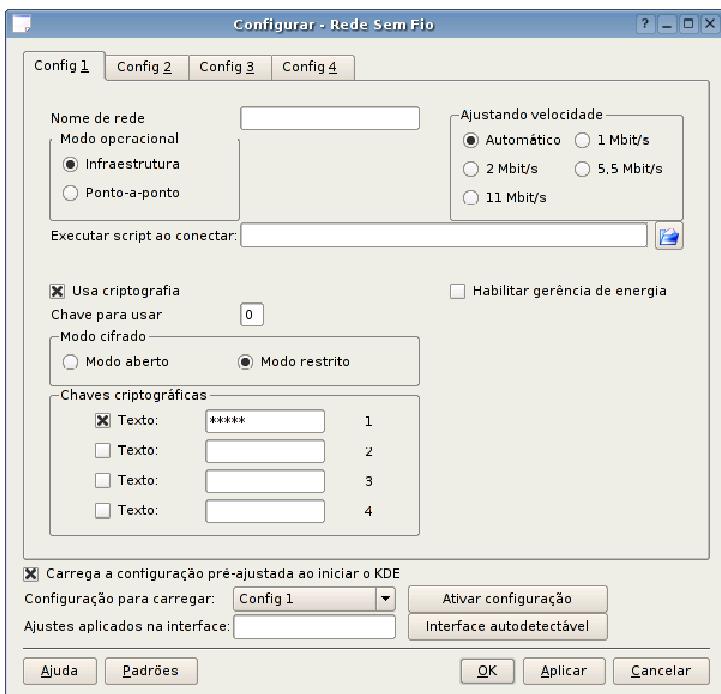
Outro recurso interessante é a janela de estatísticas da conexão, que permite monitorar a qualidade do sinal da rede, detectando em que áreas do local o sinal é melhor.



Até aqui, apenas estamos usando o Kwifimanager como monitor para uma conexão de rede já configurada. Mas, ele oferece a opção de configurar a rede diretamente, dentro da opção "Config > Editor de Configuração". Você pode criar até 4 configurações diferentes, incluindo o ESSID (que vai no campo "nome da rede"), incluir a chave de encriptação, caso exista, e configurar as opções de economia de energia da placa.

Na opção "Ajustando velocidade", prefira usar sempre a opção "Automático", que permite que o utilitário baixe a velocidade de transmissão caso o sinal esteja fraco. Lembre-se de que as redes 802.11b podem transmitir a 11, 5.5, 2 ou 1 megabit, onde quanto mais baixa a velocidade de transmissão, maior é o alcance do sinal. Se você forçar o uso de 11 megabits, não vai conseguir captar o sinal da rede quando estiver distante do ponto de acesso.

Ao usar um ponto de acesso, você deve marcar sempre a opção "Infraestrutura". O modo "Ponto-a-ponto" (ad-Hoc) é usado apenas ao conectar diretamente dois micros, sem ponto de acesso.



No campo "Executar script ao conectar", você especifica o comando usado para configurar os endereços da rede (IP, máscara, etc.), que será executado ao conectar ou ao trocar de profile.

Para configurar a rede via DHCP, preencha o campo com o comando "dhclient" (no Fedora ou Mandriva), "dhcpcd" (outras distribuições) ou "pump -i wlan0" (no Debian). Você pode também escrever um script com a configuração completa da rede (como vimos anteriormente), marcar a permissão de execução para o arquivo e indicá-lo aqui.

Depois de configurar os profiles referentes a todas as redes que utiliza, marque a opção "Carrega a configuração pré-ajustada ao iniciar o KDE" para que a configuração mais usada seja ativada no boot. Ao mudar para outra rede, acesse novamente o menu e troque o perfil.

» Próximo: [A questão dos drivers](#)

Um dos principais fatores que você deve levar em conta na hora de escolher uma placa wireless é se o modelo escolhido é bem suportado no Linux. Caso a placa tenha um driver disponível, a configuração será simples, mas sem o driver você fica trancado do lado de fora do carro. Lembre-se: o driver é a chave e você nunca deve comprar um carro sem a chave :). No caso do Windows, a situação é muito mais tranquila, já que os fabricantes incluem os drivers no CD que acompanha a placa, embora, em muitos casos, estejam disponíveis apenas drivers para Windows XP, o que deixa de fora quem ainda usa o Windows 98 ou ME.

Vamos, então, a um resumo dos drivers Linux, que você pode usar como ponto de partida na hora de resolver problemas de detecção da placa wireless. Você vai encontrar uma descrição mais detalhada dos drivers para placas wireless for Linux disponíveis, juntamente com informações sobre placas suportadas e instruções de instalação de cada um no meu livro *Linux, ferramentas técnicas*.

1- Você pode verificar o chipset utilizado na sua placa usando o comando "**lspci**" (em algumas distribuições ele está disponível apenas para o root). Os fabricantes muitas vezes utilizam chipsets diferentes em variações do mesmo modelo, por isso é sempre mais seguro se orientar pela saída do comando lspci do que pelo modelo da placa.

O lspci lista todos os periféricos plug-and-play instalados no micro. Entre eles, você verá uma linha sobre a placa wireless, como em:

02:00.0 Network controller: **Texas Instruments ACX 111** 54Mbps Wireless Interface
(veja que neste exemplo temos uma placa ACX 111) ou:

Network controller: **Intel Corporation PRO/Wireless 2200BG** Network Connection
(rev 05)
(que indica que a placa é uma Intel IPW2200)

2- A maior parte das placas possui drivers for Linux, mas muitos destes drivers são de código fechado ou precisam do firmware da placa (um componente proprietário, que faz parte dos drivers for Windows), por isso muitas distribuições não os incluem. No Mandriva, por exemplo, os drivers de código fechado são incluídos apenas na versão power pack, que não está disponível para download público.

Nesses casos é preciso instalar o driver manualmente, baixando o código fonte e compilando. Para isso você precisa ter instalados os compiladores (a opção "Development" (desenvolvimento) geralmente disponível durante a instalação) e também os pacotes "kernel-source" (o código fonte do Kernel) e "kernel-headers" (os headers do Kernel, um conjunto de símbolos utilizados na compilação do driver). Sem esses componentes corretamente instalados, você não conseguirá compilar nenhum driver.

Para as placas **ACX100** e **ACX111**, use o driver disponível no <http://rhlx01.fht-esslingen.de/~andi/acx100/>.

Para instalar, você deve usar o procedimento padrão para arquivos .tar.gz com código fonte, ou seja, baixar o arquivo, descompactá-lo e executar (dentro da pasta criada) os comandos:

```
$ (este usando seu login de make usuário) # make install
# (este como root)
```

Este é um dos drivers que precisam do firmware da placa para funcionar. Depois de instalar o driver propriamente dito, execute o arquivo "*scripts/fetch_firmware*" (ainda dentro da pasta onde o arquivo foi descompactado), use a opção "C" para baixar os firmwares para os dois chipsets de uma vez.

Ele baixa os arquivos necessários e os salva na pasta "firmware/". Para instalar, acesse a pasta e copie todo o conteúdo para a pasta "/usr/share/acx/" (crie a pasta manualmente, caso necessário). A partir daí, a placa passa a ser detectada durante o boot.

Para as placas com chipset **Atheros**, use o driver disponível no: <http://madwifi.otaku42.de/> ou <http://madwifi.sourceforge.net/>.

Para as placas **ADMteck** com chipset ADM8122, use o driver disponível no: <http://aluminum.sourmilk.net/adm8211/>.

As placas **Intel**, com chipset IPW2100 ou IPW2200 (usadas nos notebooks com tecnologia **Centrino**), possuem um driver nativo muito estável, desenvolvido pela própria Intel. O driver já vem pré-instalado em praticamente todas as distribuições atuais, mas muitas não incluem o firmware, que é proprietário.

Para baixá-lo manualmente, acesse o <http://ipw2200.sourceforge.net/> (ou <http://ipw2100.sourceforge.net/>, se você possui uma placa IPW2100 antiga) e baixe o arquivo compactado na seção "firmware". Copie o arquivo para a pasta "/lib/firmware" e descompacte-o, como em:

```
# cp -a ipw2200-fw-2.4.tgz /lib/firmware/  
# cd /lib/firmware/  
# tar -zxf ipw2200-fw-2.4.tgz
```

Uma pegadinha é que existem várias versões do firmware disponíveis no <http://ipw2200.sourceforge.net/firmware.php>, acompanhando as diferentes versões do driver. A versão 2.4 funciona em conjunto com o driver de versão 1.07 até 1.10 (usado no Kurumin 6.0, por exemplo), enquanto o firmware versão 3.0 funciona em conjunto com o 1.11 em diante. Ao instalar uma nova versão do driver, lembre-se também de checar e, se necessário, atualizar também o firmware. Você pode checar qual é a versão instalada usando o comando:

```
# modinfo ipw2200  
(ou modinfo ipw2100)
```

Com os arquivos do firmware no local correto, o driver passa a funcionar corretamente. Você pode verificar as mensagens de inicialização usando o comando "**dmesg | grep ipw**", como em:

```
# dmesg | grep ipw
```

ipw2200:	Intel(R)	PRO/Wireless	2200/2915	Network	Driver,	1.0.8	
ipw2200:	Copyright(c)		2003-2005	Intel	Corporation		
ipw2200:	Detected Intel PRO/Wireless 2200BG Network Connection						

Se, por outro lado, o comando exibir algo como:

```
ipw2200:      Intel(R)      PRO/Wireless      2200/2915      Network      Driver,      1.0.8
ipw2200:          Copyright(c)           2003-2005      Intel
ipw2200:      Detected      Intel      PRO/Wireless      2200BG      Network      Corporation
ipw2200:          Unable      to      load      ucode:      Connection
ipw2200:          Unable      to      load      firmware:      -62
ipw2200:          failed      to      register      network      -62
ipw2200: probe of 0000:01:05.0 failed with error -5      device
```

... rode os comandos abaixo para recarregar o driver:

```
#       echo      100      >      /sys/class/firmware/timeout
#       modprobe      ipw2200
# modprobe ipw2200
```

A partir daí a placa passará a funcionar normalmente. Você pode incluir os comandos no final do arquivo "/etc/init.d/bootmisc.sh" (se você usa uma distribuição derivada do Debian) ou "/etc/rc.d/rc.local" (se você usa o Fedora ou outra distribuição derivada do Red Hat), para que eles sejam executados automaticamente durante o boot.

O primeiro comando aumenta o tempo de espera do Kernel na hora de carregar o firmware. O default são 10 milissegundos, o que não é suficiente em algumas versões do driver. Aumentando o tempo de espera, o driver passa a carregar corretamente.

As placas com chipset **Ralink** também são muito bem suportadas. A Ralink é uma pequena fabricante taiwanesa, que fabrica placas de baixo custo. Além de divulgar as especificações das placas, eles apóiam o desenvolvimento dos drivers Linux e fornecem hardware e outros recursos aos desenvolvedores.

As distribuições atuais incluem os drivers para elas, mas, em casos de problemas, você pode baixa-los no: <http://prdownloads.sourceforge.net/rt2400/>.

Existe também um driver nativo para as placas baseadas no chipset **Realtek rtl8180**, que pode ser baixado no: <http://rtl8180-sa2400.sourceforge.net/>. Ele também é open-source e incluído por padrão nas distribuições atuais.

3- Para as placas que não possuem um driver nativo, você pode utilizar o **Ndiswrapper**, que permite ativar a placa utilizando o driver do Windows XP. Ele utiliza parte do código do Wine, adaptado para trabalhar com drivers de placas wireless, ao invés de executáveis de programas. A página oficial é a <http://sourceforge.net/projects/ndiswrapper/>.

Ele vem pré-instalado na maioria das distribuições modernas. No Ubuntu (Breezy ou Dapper), ele não é instalado por padrão, mas faz parte do CD de instalação, de forma que você não precisa estar conectado para instalar:

```
$ sudo apt-get install ndiswrapper-utils
```

Para usar o Ndiswrapper, você precisa ter em mãos o driver da placa para Windows XP, que pode ser encontrado no CD de instalação ou no site do fabricante. Comece

descompactando o arquivo do driver (caso necessário); para carregar o arquivo do driver, rode o comando "**ndiswrapper -i**" (como root), seguido do caminho completo para o arquivo, como em:

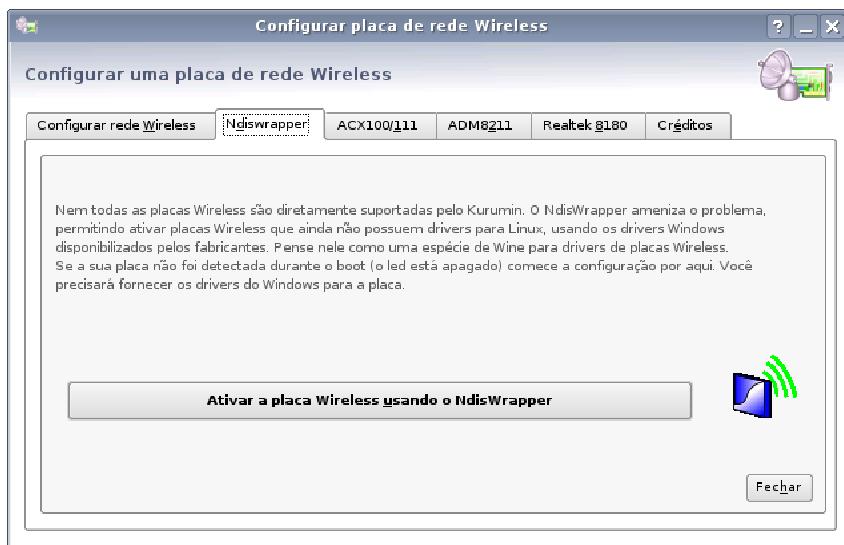
```
# ndiswrapper -i /mnt/hda6/Driver/WinXP/GPLUS.inf
```

Com o driver carregado, ative o módulo com o comando:

```
# modprobe ndiswrapper
```

Se tudo estiver ok, o led da placa acenderá, indicando que ela está ativa. As placas ativadas através do Ndiswrapper são sempre detectadas como "wlan0", independentemente do driver usado. Com a placa ativa, falta apenas configurar os parâmetros da rede wireless, usando os mesmos passos que vimos anteriormente.

No Kurumin, você pode usar o script de configuração disponível na aba "Ndiswrapper", do painel de configuração de placas wireless. Ele automatiza o processo de configuração, pedindo que você indique a localização do driver Windows e, em seguida, forneça a configuração da rede:



No Ubuntu, a melhor opção de interface gráfica de configuração é o "ndisgtk", que você pode instalar via apt-get:

```
$ sudo apt-get install ndisgtk
```

Depois de instalado, será incluído o ícone "Windows Wireless Drivers" no menu "Sistema > Administração". Ele é bem simples de usar: clique no "install new driver", indique o driver Windows que será carregado. Clicando no "Configure Network" você abre o network-admin, onde pode configurar os parâmetros da rede:



Muitos drivers que funcionam através do Ndiswrapper trabalham com um conjunto limitado de recursos. Em alguns casos, recursos como o monitoramento da qualidade do sinal, configuração da potência do transmissor, WPA ou mesmo o WEP de 128 bits não são suportados, embora os recursos básicos da placa funcionem perfeitamente. As placas ativadas através do Ndiswrapper também não funcionam em conjunto com o Kismet ou outros softwares similares, que colocam a placa em modo monitor. Sempre que for comprar, procure diretamente uma placa com drivers nativos, deixe para utilizar o Ndiswrapper como último recurso.

» Próximo: [Configurando a rede wireless manualmente](#)

Assim como no caso das redes cabeadas, você também pode configurar a rede wireless manualmente nos clientes Linux. Conhecer estes comandos é interessante não apenas para solucionar problemas, mas também para poder desenvolver pequenos scripts de configuração.

Para detectar os pontos de acesso disponíveis (a partir do cliente), use o comando:

```
# iwlist wlan0 scan
```

Lembre-se de que, dependendo do driver usado, o dispositivo de rede usado será diferente. Ao utilizar o driver para placas Intel IPW2200 ou ADM8211, por exemplo, a placa wireless será vista pelo sistema como "eth0" (ou eth1) e não como "wlan0", que seria o mais comum.

Para que o comando funcione, é preciso que a placa esteja ativada. Caso necessário, antes de executá-lo, use o comando:

```
# ifconfig wlan0 up
```

Se você estiver dentro do alcance de algum ponto de acesso, o iwlist lhe retorna um relatório como este:

```
wlan0           Scan completed :  
Cell 01 - Address: 00:51:56:81:81:01  
          ESSID:""  
          Mode:Master  
          Channel:11  
          Quality:22   Signal level:0  
          Encryption key:on  
          Bit Rate:11Mb/s
```

Nesse caso, temos um ponto de acesso dentro da área de alcance. Falta apenas configurar a placa para se conectar a ele. Veja que este ponto de acesso está com a encriptação ativa (Encryption key:on) e não está divulgando seu ESSID (ESSID:""). Este é um exemplo de configuração de um ponto de acesso não público, onde é necessário saber ambas as informações para se conectar à rede.

Na ilustração, temos um exemplo de resultado ao escanear uma rede pública. Neste caso, o serviço de acesso oferecido em um aeroporto, onde o objetivo é permitir que os clientes se conectem da forma mais simples possível.

```
Sessão Editar Ver Favoritos Configurações Ajuda  
ath0      Scan completed :  
Cell 01 - Address: 00:02:6F:36:3D:6D  
          ESSID:"vex"  
          Mode:Master  
          Frequency:2.437GHz  
          Quality:0/0  Signal level:-95 dBm  Noise level:-95 dBm  
          Encryption key:off  
          Bit Rate:1Mb/s  
          Bit Rate:2Mb/s  
          Bit Rate:5Mb/s  
          Bit Rate:11Mb/s  
          Extra:bcn_int=100  
Cell 02 - Address: 00:02:6F:36:3B:8E  
          ESSID:"vex"  
          Mode:Master  
          Frequency:2.437GHz  
          Quality:0/0  Signal level:-95 dBm  Noise level:-95 dBm  
          Encryption key:off  
          Bit Rate:1Mb/s  
          Bit Rate:2Mb/s  
          Bit Rate:5Mb/s  
          Bit Rate:11Mb/s  
          Extra:bcn_int=100
```

Veja que neste caso estão disponíveis dois pontos de acesso, ambos usam o ESSID "vex" e ambos estão com a encriptação de dados desativada (Encryption key:off). Por usarem o mesmo ESSID, eles fazem parte da mesma rede, por isso você não precisa especificar em qual deles quer se conectar. Basta configurar a rede wireless e, em seguida, obter a configuração da rede via DHCP.

Em outros casos, pode haver mais de uma operadora oferecendo acesso no mesmo local, ou mesmo outros pontos de acesso de particulares, que intencionalmente ou não estejam com a

encriptação desativada, oferecendo acesso público. Nesse caso, você escolhe em qual rede quer se conectar especificando o ESSID correto na configuração da rede.

É comum também que os pontos de acesso sejam configurados para usar um canal específico. Neste caso, ao rodar o "iwlist wlan0 scan" você verá também uma linha "channel=x", onde o x indica o número do canal, que também precisa ser especificado na configuração da rede.

Tome cuidado ao se conectar a pontos de acesso público. Com a encriptação desativada, todos os dados transmitidos através da rede podem ser capturados com muita facilidade por qualquer um dentro da área de alcance. Lembre-se de que o alcance de uma rede wireless cresce de acordo com a potência da antena usada no cliente. Com uma antena de alto ganho, é possível se conectar a um ponto de acesso a 500 metros de distância, ou até mais, caso não exista nenhum tipo de obstáculo pelo caminho.

Sempre que precisar transferir arquivos, use um protocolo que transmita os dados de forma encriptada (como o SSH). Jamais dê upload de arquivos para o servidor do seu site via FTP. Acesse e-mails apenas em servidores que oferecem suporte a pop3 com SSL. Não acesse páginas de bancos, pois a encriptação usada nos navegadores pode ser quebrada com uma relativa facilidade e obter senhas bancárias é o tipo de situação em que o trabalho necessário vale a pena.

Voltando à configuração, o primeiro passo é definir o SSID da rede, usando o comando "**iwconfig**", como em:

```
# iwconfig wlan0 essid casa
```

Lembre-se sempre de verificar qual é o dispositivo usado pela sua placa de rede wireless, ele varia de acordo com o driver usado (a placa pode ser vista pelo sistema como wlan0, ath0 ou mesmo eth0). Você pode verificar isso rapidamente rodando o comando ifconfig.

Caso você tenha configurado o ponto de acesso para utilizar um canal específico, configure a placa para utilizá-lo com o comando:

```
# iwconfig wlan0 channel 10
```

Caso você tenha ativado a encriptação via **WEP** no ponto de acesso, é necessário especificar também a chave. Ao usar caracteres hexadecimais, a chave terá 10 dígitos (123456789A no exemplo) e o comando será:

```
# iwconfig wlan0 key restricted 123456789A
```

Se a chave for em ASCII, onde cada caracter equivale a 8 bits, a chave terá apenas 5 dígitos (qwerty no exemplo) e o comando será:

```
# iwconfig wlan0 key restricted s:qwerty
```

Veja que ao usar uma chave em ASCII você precisa adicionar o "s:" antes da chave. Ao configurar o ponto de acesso para usar uma chave de 128 bits, a chave terá 26 dígitos em

hexa ou 13 em ACSII. Depois de terminar a configuração inicial, você pode ativar a interface com o comando:

```
# ifconfig wlan0 up
```

O último passo é configurar os endereços da rede, como você faria em uma placa convencional. Se você se empolgou e quer ir adiante, configurando também a rede manualmente, use os comandos que vimos há pouco:

```
#      ifconfig      wlan0      192.168.0.2      netmask      255.255.255.0
#          route           del           default
# route add default gw 192.168.0.1 dev wlan0
```

Se preferir configurar a rede via DHCP, rode o comando:

```
# dhcpcd wlan0
```

Algumas distribuições (como o Knoppix e o Kurumin) usam o pump no lugar do dhcpcd. Neste caso, o comando fica:

```
# pump -i wlan0
```

Não se esqueça de configurar também os endereços dos servidores DNS no arquivo "/etc/resolv.conf".

Com a rede funcionando, você pode monitorar a qualidade do link, a taxa de transmissão de dados, o tipo de encriptação, as informações sobre o ponto de acesso, entre outros detalhes da conexão usando o "wavemon", um pequeno utilitário incluído na maioria das distribuições.

Depois de fazer a configuração inicial, você pode criar um pequeno script, contendo os comandos de configuração, de forma que não precise ficar executando-os cada vez que desligar o micro ou precisar se reassociar ao ponto de acesso. Crie um arquivo de texto, como, por exemplo "/home/kurumin/wireless". Dentro dele vão os comandos, um por linha, como em:

```
#!/bin/sh
iwconfig
iwconfig      wlan0
iwconfig      wlan0      essid      casa
iwconfig      wlan0      key       channel    10
ifconfig
ifconfig      wlan0      wlan0
ifconfig      wlan0      192.168.0.2   restricted  1234567890
route
route        wlan0      del       netmask    up
route        add           default   255.255.255.0
route add default gw 192.168.0.1   default   wlan0
```

Transforme o arquivo em um executável com o comando "chmod +x /home/kurumin/wireless" e execute-o (como root) sempre que quiser ativar a rede wireless. Você pode ter vários arquivos diferentes caso precise se conectar a várias redes com configurações diferentes. Outra opção é usar o wpa_supplicant, que oferece a vantagem de

se associar automaticamente ao ponto de acesso, ao entrar dentro do alcance de qualquer uma das redes configuradas.

Se você usar ao mesmo tempo uma placa de rede cabeada e uma placa wireless e o acesso pela placa wireless ficar intermitente, com a configuração caindo poucos minutos depois de configurada a rede, experimente começar a desativar a placa cabeada ao configurar a rede wireless.

Esse é um problema freqüente, principalmente ao utilizar o ndiswrapper, mas felizmente fácil de resolver. Antes de configurar a placa wireless, desative a placa cabeada. Se a placa cabeada é a eth0, por exemplo, rode o comando:

```
# ifconfig eth0 down
```

Você pode adicionar o comando no seu script de configuração da rede, para que ele seja executado sempre antes dos comandos que configuram a placa wireless.

» Próximo: [Usando o wpa_supplicant](#)

Embora as versões mais recentes do WEP, usadas nos pontos de acesso e placas atuais, sejam mais seguras que as primeiras versões, elas ainda podem ser facilmente quebradas, como veremos com mais detalhes no capítulo 4.

Embora não seja infalível, o WPA é um padrão mais seguro, por isso é o preferido em redes onde a segurança é um fator importante. Nem todas as placas são compatíveis com o WPA e, no Linux, existe um complicador adicional, que é o fato de alguns dos drivers não oferecerem suporte a ele, ainda que a placa originalmente suporte. Para usar o WPA em uma rede de médio ou grande porte, é preciso escolher com um certo cuidado quais placas e pontos de acesso usar.

Para conectar os clientes Linux à rede, usamos o **wpa_supplicant**. Algumas distribuições já incluem ferramentas de configuração para ele, mas vou descrever aqui o processo manual de configuração, que você pode usar para corrigir problemas ou quando não houver nenhuma ferramenta mais simples disponível.

Comece instalando o pacote `wpa_supplicant` ou `wpasupplicant`. Nas distribuições derivadas do Debian, você pode instalá-lo via apt-get:

```
# apt-get install wpasupplicant
```

Uma dica é que, ao usar o **Ubuntu** você deve primeiro abrir o arquivo `/etc/apt/sources.list` e descomentar a linha referente ao repositório Universe, como em:

```
deb http://br.archive.ubuntu.com/ubuntu dapper universe
```

Depois de salvar o arquivo, rode o "apt-get update" e você poderá instalar o wpa_supplicant via apt-get, usando o mesmo comando do Debian. A partir do Ubuntu 6.6, ele já vem instalado por padrão.

No **Fedora**, instale-o usando o yum:

```
# yum install wpa_supplicant
```

O passo seguinte é criar o arquivo de configuração do wpa_supplicant, contendo o SSID e a senha da sua rede. É possível também criar uma configuração que permita conectar em várias redes diferentes, como veremos a seguir.

Rode o comando "**wpa_passphrase**" seguido do SSID da rede e a passphrase (a senha), como em:

```
$ wpa_passphrase minharede minhapassphrase
```

Ele retorna a configuração que deve ser incluída no arquivo, como em:

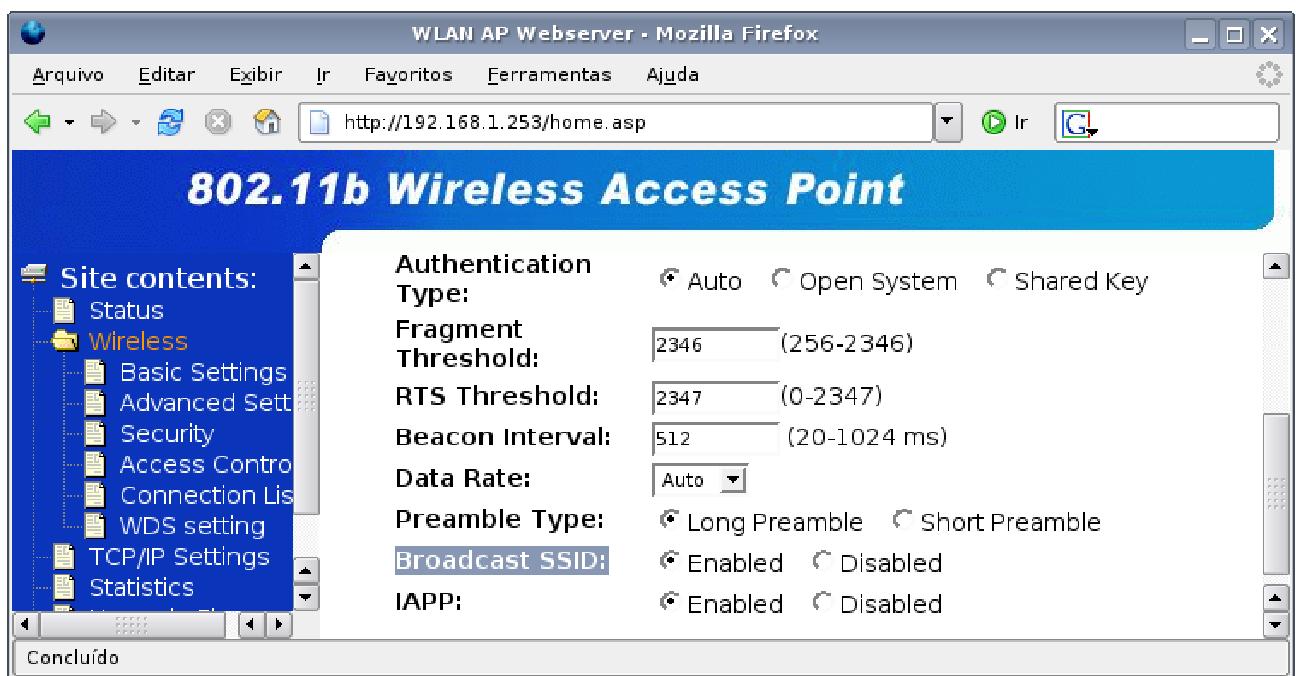
```
network={  
    ssid="minharede"  
    #psk="minhapassphrase"  
    psk=24b0d83ee1506019e87fcf1705525ca60abbd9b24ac5bedf183620d0a22ab924  
}
```

Note que ele inclui duas linhas "psk", onde vai a passphrase. A linha que está comentada contém sua passphrase real, enquanto a segunda contém um "hash" (verificador), que funciona da mesma forma, mas evita que você precise deixá-la disponível dentro do arquivo para qualquer um ver. Apague a linha comentada, deixando apenas a segunda linha, com o hash.

Agora edite (ou crie) o arquivo "**/etc/wpa_supplicant.conf**", de forma que ele contenha apenas as linhas retornadas pelo comando. Você pode usar também o comando abaixo (como root), que já modifica automaticamente o arquivo, matando os dois coelhos com uma cajadada só:

```
# wpa_passphrase minharede minhapassphrase > /etc/wpa_supplicant.conf
```

Falta agora se conectar ao ponto de acesso. Uma ressalva importante é que, para usar o WPA em conjunto com o wpa_supplicant, o ponto de acesso deve estar configurado com a opção "Broadcast SSID" ativada. Caso contrário, o wpa_supplicant não consegue encontrar o ponto de acesso e não estabelece a conexão. Verifique e, caso necessário, altere a configuração do AP:



Agora vem o comando que ativa o `wpa_supplicant`, especificando a placa de rede que será usada, o arquivo de configuração que acabamos de criar e o driver que será usado:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf -d -D wext
```

O "wlan0" indica a sua placa wireless, lembre-se de verificar qual é o dispositivo correto no seu caso. O "wext", por sua vez, indica o driver que será usado pelo `wpa_supplicant`. As possibilidades aqui são as seguintes:

wext: Este é um driver genérico, que dá suporte à maioria das placas. Ele está se tornando a opção "default", com a incorporação de cada vez mais drivers, que antes eram separados. Nas versões recentes ele incorpora também suporte às placas ativadas através do ndiswrapper (usando o driver do Windows) e também às placas IPW2200.

ndiswrapper: Este é o driver para placas ativadas através do ndiswrapper, que, nas versões recentes, foi incorporado ao wext. Note que muitas placas funcionam perfeitamente no Ndiswrapper em redes sem encriptação ou WEP, mas ficam instáveis ao usar o WPA, justamente porque ele utiliza mais camadas e por isso tem uma possibilidade maior de apresentar problemas diversos.

ipw: Driver para as placas com os chipsets Intel IPW2100 e IPW2200, usadas nos notebooks Intel Centrino. Assim como no caso do ndiswrapper, o driver foi incorporado ao wext nas versões recentes. De qualquer forma, o driver antigo continua disponível e você pode experimentar ambos caso esteja tendo problemas para ativar a placa.

madwifi: Este é o driver para placas com chipset Atheros, utilizadas (por exemplo) em alguns notebooks Toshiba.

broadcom: Este é o driver nativo para as placas com chipset Broadcom, desenvolvido via engenharia reversa. Estas placas podem ser configuradas também através do Ndiswrapper. Cheque a forma como a placa está configurada no seu micro.

prism54, hermes e atmel: Estes três drivers são os mais incomuns, usados (respectivamente) pelas placas com chipset Prism (em suas várias versões), Hermes, Hermes II e Atmel.

Se você estivesse usando uma placa Atheros, reconhecida pelo sistema como ath0, por exemplo, o comando seria:

```
# wpa_supplicant -i ath0 -c /etc/wpa_supplicant.conf -d -D madwifi
```

Por causa da opção "**-d**" que incluímos no comando, ele roda em modo verbose, onde são mostrados detalhes sobre a conexão com o ponto de acesso. Este modo é interessante para descobrir problemas.

Se a conexão for bem-sucedida, você terá (depois de uma rápida sucessão de mensagens), algo como:

```
State: GROUP_HANDSHAKE      -> COMPLETED
CTRL-EVENT-CONNECTED - Connection to 00:50:50:81:81:01 completed (auth)
EAPOL: External notification - portValid=1
EAPOL: External notification - success=1
EAPOL: SUPP_PAE entering state AUTHENTICATING
EAPOL: SUPP_BE  entering state SUCCESS
EAP:   EAP    entering state DISABLED
EAPOL: SUPP_PAE entering state AUTHENTICATED
EAPOL: SUPP_BE  entering state IDLE
EAPOL: startWhen --> 0
```

Estas mensagens indicam que ele se conectou ao ponto de acesso com o endereço MAC "00:50:50:81:81:01" e que a conexão está disponível para transmitir dados.

A partir daí, você precisa apenas configurar os parâmetros da rede (IP, máscara, gateway e DNS) usando a ferramenta adequada para que a conexão fique disponível. No Kurumin e em outras distribuições derivadas do Knoppix você pode usar o "netcardconfig", no Fedora pode usar o "system-config-network", no Ubuntu pode usar o "network-admin" (o "Configurar rede no Menu"), no Slackware pode usar o "netconfig" e assim por diante.

Se, por outro lado, você receber mensagens como:

Scan	BSS	from	results:	priority	group	0
Selecting						0
No	suitable			AP		found.
Setting	scan	request:		5	sec	usec
Starting	AP	scan			(broadcast)	SSID)
Wireless event: cmd=0x8b1a len=8						

ou:

```
Setting      scan    request:      1      sec      0      usec
Starting     AP      scan          sec      (broadcast
Wireless    event:      scan          cmd=0x8b19
ioctl[SIOCGIWSCAN]:   Resource      temporarily
Scan          results:      -1
Failed       to      get           scan
Failed to get scan results - try scanning again
```

... significa que a conexão não foi estabelecida. Pode ser que o ponto de acesso esteja configurado para não divulgar o SSID, que o seu notebook está muito longe do ponto de acesso, fora da área de alcance ou mesmo que a antena do notebook está desativada.

No caso da segunda mensagem, é provável que o driver indicado na linha de comando esteja incorreto ou que não suporte o WPA. Este é o caso de muitas placas configuradas através do Ndiswrapper (por exemplo) ou casos em que você tenta usar o driver "wext" em uma placa que possui um driver específico, como as placas Atheros (madwifi).

Depois de testar e ver que a conexão está funcionando corretamente, você pode passar a usar o comando abaixo, trocando o "-d" por "-B". Isso faz com que o wpa_supplicant rode em modo daemon, sem bloquear o terminal nem mostrar mensagens na tela:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf -B -D wext
```

» Próximo: [Ativando o wpa_supplicant no boot](#)

Falta agora automatizar as coisas, fazendo com que o comando seja executado automaticamente durante o boot. Existem várias formas de fazer isso. Você poderia desde criar um ícone no desktop até adicionar o comando no final do arquivo "/etc/init.d/bootmisc.sh" ou "/etc/init.d/rc.local". Mas a solução mais correta é fazer com que o sistema estabeleça a conexão ao ativar as interfaces de rede.

Se você usa o Ubuntu, Kurumin ou qualquer outra distribuição derivada do **Debian**, abra o arquivo **/etc/network/interfaces**. Adicione as duas linhas abaixo, contendo os parâmetros do wpa_supplicant, **no final do arquivo**. Note que agora adicionamos também a opção "w", específica para uso em scripts de inicialização:

```
up wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf -wB -D wext
down killall wpa_supplicant
```

Lembre-se de substituir o "wlan0" e "wext" pelos parâmetros corretos no seu caso. Para que fique tudo certo, o arquivo deve conter também uma seção com os endereços usados pela placa. Veja um exemplo de arquivo completo:

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
```

```

auto                                lo                               wlan0
iface lo inet loopback

iface          wlan0      inet      static
address        192.168.1.56
netmask        255.255.255.0
network        192.168.1.0
broadcast      192.168.1.255
gateway        192.168.1.254

up      wpa_supplicant -i     wlan0      -c      /etc/wpa_supplicant.conf      -wB      -D      weext
down killall wpa_supplicant

```

Como de praxe, ao usar esse exemplo como modelo, preste atenção para substituir todos os "wlan0" pela interface correta no seu caso e ajustar os endereços de acordo com a configuração da sua rede.

Caso você use o **Fedora**, Mandriva ou outra distribuição originária do Red Hat, abra o arquivo **"/etc/sysconfig/network-scripts/ifup-wireless"** e adicione a linha:

```
wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf -wB -D weext
```

Crie o arquivo **"/etc/sysconfig/network-scripts/ifdown-wireless"**, contendo a linha "killall wpa_supplicant" e transforme-o em executável:

```
# echo 'killall wpa_supplicant' > /etc/sysconfig/network-scripts/ifdown-wireless
# chmod +x /etc/sysconfig/network-scripts/ifdown-wireless
```

O arquivo ifup-wireless é executado pelo Fedora ao ativar a placa de rede, enquanto o ifdown-wireless é executado ao desativá-la. Ao editar os dois arquivos, estamos fazendo com que os comandos do wpa_supplicant sejam executados corretamente nas duas situações.

Uma observação importante é que, ao ser configurado para rodar em background, o wpa_supplicant ficará o tempo todo tentando se conectar às redes wireless configuradas. Se você quiser se conectar a uma rede cabeadas, deverá, além de configurar a rede, desativar o wpa_supplicant, usando o comando:

```
# killall wpa_supplicant
```

» Próximo: [Opções avançadas](#)

Um dos recursos mais interessantes do wpa_supplicant é a possibilidade de definir várias redes diferentes no arquivo de configuração. O wpa_supplicant passa, então, a testar cada uma delas periodicamente, conectando-se a que estiver disponível. Daí que surgiu o nome: supplicant significa, literalmente, "pedinte".

Neste caso, você deve especificar cada uma das redes no arquivo **"/etc/wpa_supplicant.conf"**, juntamente com um "peso" ou prioridade para cada uma. O

arquivo poderia conter uma entrada para a rede da sua casa, que usa WPA, a rede do escritório, que usa WEP, e uma entrada para redes públicas, sem encriptação. O restante da configuração continua igual ao que já vimos.

A rede de casa pode ter peso 5, a do escritório, peso 3 e as redes públicas, peso 1, de forma que ele dá prioridade às duas redes e tenta se conectar a qualquer rede pública disponível caso nenhuma das duas seja encontrada.

Neste caso o arquivo ficaria:

```
# Rede de casa, com WPA (esta é a entrada gerada pelo wpa_passphrase)

network={
ssid="casa"
key_mgmt=WPA-PSK
psk=2ceaa0388fa863213f5f527055846101dc449c9a569c1e43ea535a3344d3dc32
priority=5
}

# Rede do escritório, com WEP:

network={
ssid="escritorio"
key_mgmt=NONE
wep_key0=ADADA54321
wep_tx_keyidx=0
priority=3
}

# Redes públicas, sem encriptação

network={
ssid=""
key_mgmt=NONE
priority=1
}
```

Note que incluí a linha "priority", dentro de cada uma das entradas, especificando a prioridade de cada uma. As redes com prioridade mais alta são testadas primeiro, deixando a entrada para redes públicas como último recurso.

No caso da entrada para redes WEP, você substituiria apenas o SSID e a chave de encriptação pelos valores corretos, mantendo as linhas "key_mgmt=NOME" e "wep_tx_keyidx=0", que fazem parte da configuração.

Assim como existem dois padrões WEP (de 64 e 128 bits), existem dois padrões WPA, chamados "WPA Personal" e "WPA Enterprise".

O WPA Personal (também chamado de WPA-PSK) é o padrão mais comum, que abordei até agora. Nele você define uma passphrase (pre-shared key) na configuração do ponto de acesso e a fornece na configuração de cada cliente. É um sistema relativamente simples, mas que garante uma boa segurança, desde que você não use uma passphrase fácil de adivinhar.

O WPA Enterprise, por sua vez, utiliza uma estrutura muito mais complexa, em que o ponto de acesso é ligado a um servidor Radius, que controla a autenticação. Além da senha de acesso, o cliente precisa possuir um certificado digital, que pode ser desde um arquivo no HD, até um smartcard. Quando o cliente se conecta, é criado um túnel encriptado (uma espécie de VPN) entre ele e o servidor, garantindo a segurança dos dados.

Por ser bem mais trabalhoso, este é um sistema geralmente usado apenas em empresas e ambientes onde a segurança é essencial, mas mesmo assim é perfeitamente suportado pelo wpa_supplicant. Este é um exemplo de configuração, que você usaria no arquivo "/etc/wpa_supplicant.conf":

```
network={  
    ssid="RedeSuperSegura"  
    scan_ssids=1  
    key_mgmt=WPA-EAP  
    pairwise=CCMP  
    group=CCMP  
    eap=TLS  
    identity="seunome@empresa.com"  
    ca_cert="/etc/cert/ca.pem"  
    client_cert="/etc/cert/user.pem"  
    private_key="/etc/cert/user.prv"  
    private_key_passwd="suasenha"  
}
```

TKIP
TKIP

Note que nesse caso é necessário indicar a localização dos certificados, que devem ser previamente instalados no HD (do cliente), além de fornecer o login e senha.

» Próximo: [Placas Ralink](#)

Toda regra tem sua exceção e no caso do wpa_supplicant não é diferente. As placas **Ralink rt2400**, **rt2500**, **rt2570** e **RT61** (que funcionam com os drivers disponíveis no <http://rt2x00.serialmonkey.com>) não funcionam bem em conjunto com o wpa_supplicant, pois o driver implementa o suporte a WPA nativamente.

Nelas, você pode configurar o suporte a WPA diretamente via linha de comando, usando os comandos abaixo:

```
# iwconfig      iwconfig      ra0          essid       rede  
#             iwconfig      ra0          mode        managed  
#             iwpriv       ra0          set          set  
#             iwpriv       ra0          set          Channel=11  
#             iwpriv       ra0          set          AuthMode=WPAPSK  
#             iwpriv       ra0          set          EncrypType=TKIP  
#             iwpriv       ra0          set          WPAPSK="passphrase"  
# iwpriv ra0 set TxRate=0
```

Substitua o "rede" pelo SSID correto, o "11" pelo canal usado e o "passphrase" pela passphrase definida no ponto de acesso. A linha com o "TKIP" define o algoritmo de encriptação usado, também definido no ponto de acesso.

O TKIP é o padrão "oficial" de encriptação, usado por default. Muitos pontos de acesso suportam também o padrão "AES". Ao utilizá-lo, substitua a linha por "iwpriv ra0 set EncrypType=AES".

Estes comandos fazem com que a placa se associe ao ponto de acesso. Depois, falta apenas fazer a configuração normal da rede, definindo os endereços da rede. Você pode aproveitar o embalo e completar a configuração via linha de comando, como em:

```
# ifconfig     ra0      192.168.0.2      netmask      255.255.255.0      up
# route add default gw 192.168.0.1 dev ra0
```

Para configurar via DHCP, use o comando:

```
# dhclient ra0
```

Como de praxe, estes comandos devem ser executados a cada boot. Se você se conecta sempre à mesma rede, pode colocá-los diretamente no final do arquivo "/etc/init.d/bootmisc.sh" ou "/etc/rc.d/rc.local" (no Fedora), para que sejam executados durante a inicialização. Caso use mais de uma rede diferente, a melhor solução é criar vários scripts dentro do seu diretório home, cada um contendo os comandos para uma rede diferente e ir executando-os conforme a necessidade.

» Próximo: [Configurando em modo Ad-Hoc](#)

Assim como é possível ligar dois micros diretamente usando duas placas Ethernet e um cabo cross-over, sem usar hub, também é possível criar uma rede wireless entre dois PCs sem usar um ponto de acesso. Basta configurar ambas as placas para operar em modo Ad-Hoc. A velocidade de transmissão é a mesma, mas o alcance do sinal é bem menor, já que os transmissores e as antenas das interfaces não possuem a mesma potência do ponto de acesso.

Esse modo pode servir para pequenas redes domésticas, com dois PCs próximos, embora mesmo nesse caso seja mais recomendável utilizar um ponto de acesso, interligado ao primeiro PC através de uma placa Ethernet, e uma placa wireless no segundo PC ou notebook, já que as diferenças entre o custo das placas e pontos de acesso não é tão grande assim.

Um uso comum para o modo Ad-Hoc é quando você tem em mãos dois notebooks com placas wireless. Um deles pode ser ligado ao modem ADSL (com fio) para acessar a internet e compartilhar a conexão com o segundo usando a placa wireless, que fica livre dos fios.

Depois de configurada, a placa wireless é vista pelo sistema como um dispositivo de rede normal. Você pode compartilhar a conexão da mesma forma que faria em um micro com duas placas de rede.

Para ativar a rede em modo Ad-Hoc no Linux você utiliza a mesma configuração que já aprendemos, adicionando o comando "iwconfig wlan0 mode Ad-Hoc". É bem mais fácil (embora inseguro) configurar uma rede Ad-Hoc sem encriptação, pois isso evita diversos casos de incompatibilidade, sobretudo com os clientes Windows. Para ativar a placa wlan0, configurando a rede com o IP "10.0.0.1", os comandos seriam:

```
#      ifconfig      wlan0      10.0.0.1      netmask      255.0.0.0      up
#      iwconfig      wlan0      mode          Ad-Hoc
#      iwconfig      wlan0      essid         casa
#      iwconfig      wlan0      channel       8
# iwconfig wlan0 key off
```

Para ativar a encriptação, substitua o comando "iwconfig wlan0 key off" pelo comando abaixo, especificando a chave WEP que será usada, como em:

```
# iwconfig wlan0 key restricted 12345678AF
```

Se preferir especificar a chave usando caracteres alfanuméricos, adicione o parâmetro "s:", como em:

```
# iwconfig wlan0 key restricted s:asdfg
```

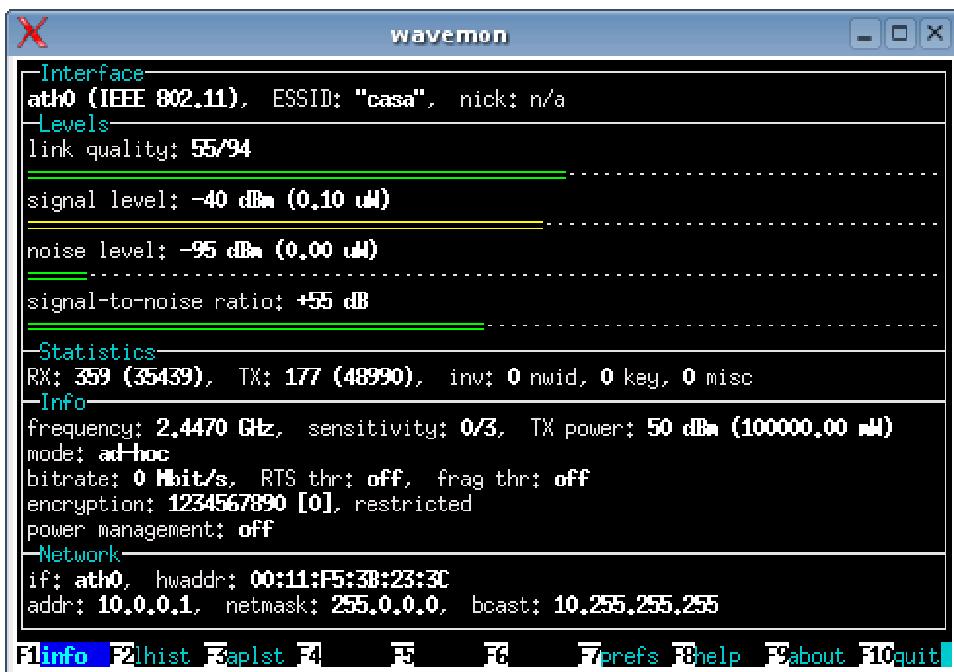
Os mesmos devem ser executados no segundo micro, mudando apenas o endereço IP da placa de rede.

Depois de estabelecida a conexão Ad-Hoc entre os dois micros, você pode compartilhar a conexão no micro com o ADSL, permitindo que o segundo navegue. Neste caso, rode os comandos abaixo no servidor, substituindo o "eth0" pela interface de rede onde está configurada a conexão com a Internet:

```
#                      modprobe      iptable_nat
#  iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Configure o cliente para utilizar o IP do servidor como gateway e adicione os endereços DNS do provedor para que ele navegue através dele.

Em uma rede Ad-Hoc, a qualidade do sinal tende a se degradar muito mais rapidamente conforme aumenta a distância. Você pode monitorar a qualidade do sinal utilizando o **wavemon**. Uma observação é que ele não funciona bem ao utilizar o Ndiswrapper, pois nele as extensões que permitem acessar as estatísticas de sinal fornecidas pela placa não funcionam em conjunto com a maior parte dos drivers.



Conforme o sinal fica fraco, as placas automaticamente negociam taxas mais baixas de transmissão. Essas negociações ocorrem apenas quando o número de retransmissões torna-se muito alto. Em muitas situações, onde você precisa apenas de um link lento para acessar a Web, configurar de uma vez a placa para usar uma taxa mais baixa pode ser a melhor forma de obter uma conexão mais estável e, em alguns casos, até mesmo mais rápida, já que é eliminado o overhead das negociações.

As taxas disponíveis em uma rede 802.11b são 11M, 5.5M, 2M e 1M. No caso de uma rede 802.11g, temos também 22M e 54M. Para que a rede passe a trabalhar a 2 megabits, por exemplo, use:

```
# iwconfig eth0 rate 2M
```

Outra opção importante é a **txpower**, que permite ativar ou desativar o transmissor da placa, além de configurar sua potência. Em alguns drivers, o padrão é (sabe-se lá porque) deixar o transmissor desativado, fazendo com que nada funcione, mesmo que você configure os parâmetros da rede corretamente. Neste caso, você pode ativar o transmissor usando o comando "iwconfig interface txpower on", como em:

```
# iwconfig wlan0 txpower on
```

Se você estiver usando um notebook e quiser desativar o transmissor temporariamente, de forma a economizar bateria enquanto não estiver usando o wireless, use o comando "iwconfig wlan0 txpower off".

Outra questão é a potência de transmissão, que determina o alcance e a qualidade do sinal. Muitas placas permitem que você configure a potência do sinal, de forma a ajustar o alcance desejado, assim como na configuração do ponto de acesso. Novamente, em alguns

drivers o default é usar uma potência bem abaixo do máximo, fazendo com que o alcance seja bem menor do que seria normalmente. Você pode ver as opções disponíveis na sua placa (e qual está sendo usada atualmente) usando o comando:

```
# iwlist wlan0 txpower
```

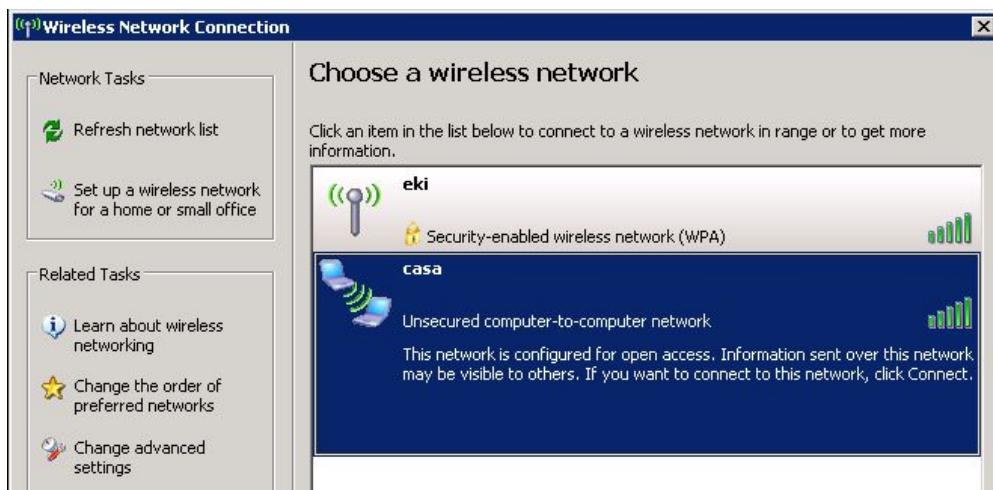
A maioria das placas suporta até um máximo de 20 dBm (100 mW) ou 25 dBm (250 mW). Para configurar a placa "wlan0" para usar 25 dBm, por exemplo, use:

```
# iwconfig wlan0 txpower 25
```

A maioria dos drivers retorna um erro caso você tente usar uma potência maior do que o máximo suportado, mas outros aceitam qualquer número, ajustando qualquer número alto para o máximo suportado pela placa. Isso, às vezes, resulta em resultados estranhos.

Ao usar uma placa Atheros, por exemplo, posso digitar "iwconfig ath0 txpower 80" e o comando "iwlist wlan0 txpower" me diz que a placa está operando com uma potência de 100.000.000 mW (milliwatts). Se isso fosse verdade, ou eu cozinharia instantâneamente (a potência média de um aparelho de microondas são 2.500.000 milliwatts) ou seria preso (o máximo permitido pela legislação são 1.000 milliwatts). Felizmente, isso é apenas um bug no driver, na verdade a placa está estando operando a 250 mW, que é o máximo que o transmissor suporta.

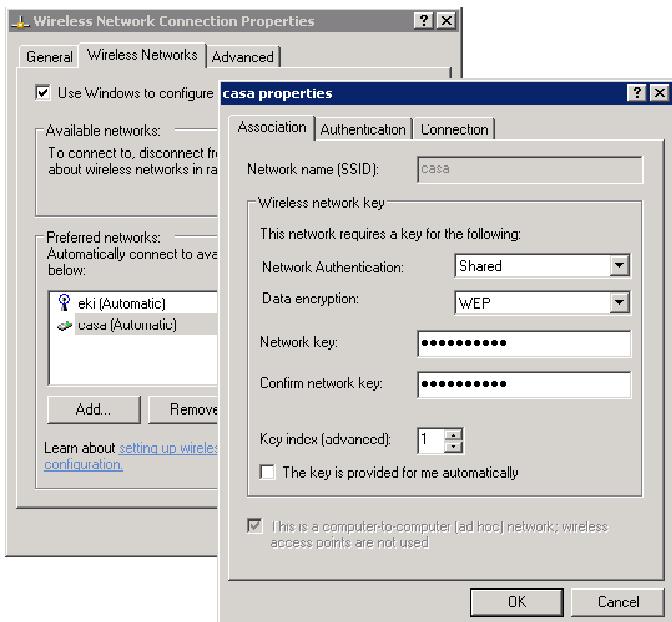
As redes Ad-Hoc aparecem normalmente na lista de redes disponíveis, nos clientes Windows, permitindo que eles se conectem diretamente:



Ao contrário de uma rede em modo de infra-estrutura (ao usar um ponto de acesso), em uma rede Ad-Hoc todos os micros estão no mesmo nível hierárquico, sem uma autoridade central. Todas as estações configuradas para usarem o mesmo SSID e as mesmas configurações de encriptação, estabelecem contato e criam uma rede ponto a ponto.

A menos que algum dos micros esteja com um servidor DHCP ativo, você precisará configurar os endereços de cada um nas propriedades do protocolo TCP/IP. Coloque sempre o micro que está compartilhando a conexão como gateway. Caso você esteja usando

encriptação, é necessário definir as chaves manualmente dentro das propriedades da conexão Ad-hoc, como neste exemplo:



» Próximo: [Acesso discado no Linux](#)

No Linux, todos os dispositivos da máquina, incluindo o HD, memória, placa de rede, etc. são acessados através de um módulo de Kernel, análogo ao driver de dispositivo que usaríamos no Windows. Este módulo recebe as informações que devem ser enviadas ao dispositivo através de um arquivo especial, criado dentro da pasta "/dev". Ao "salvar" alguma informação dentro do arquivo "/dev/modem", por exemplo, ela é enviada para o modem.

No caso das placas de rede, o procedimento é um pouco mais complexo, pois é preciso utilizar o TCP/IP ou outro protocolo de rede, mas a idéia central continua sendo a mesma. Os programas se comunicam com a placa de rede utilizando um "device", ou seja, um arquivo especial dentro da pasta "/dev".

A placa de rede é vista pelo sistema com o dispositivo "**eth0**". Caso você tenha mais de uma, a segunda torna-se a "**eth1**", a terceira passa a ser a "**eth2**" e assim por diante. Ao conectar via acesso discado, seja via modem, seja via ADSL PPPoE (com autenticação), é criada a interface virtual "**ppp0**". O "eth" vem de "Ethernet", enquanto o "ppp" vem de "Point to Point Protocol".

No caso do acesso via ADSL, a interface "ppp0" substitui temporariamente a interface "eth0" onde o modem está conectado. É importante entender que neste ponto o sistema não

utiliza mais a eth0 para enviar dados, mas sim a ppp0. As duas passam a ser vistas como dispositivos diferentes. Você pode ver a configuração atual das interfaces de rede rodando o comando:

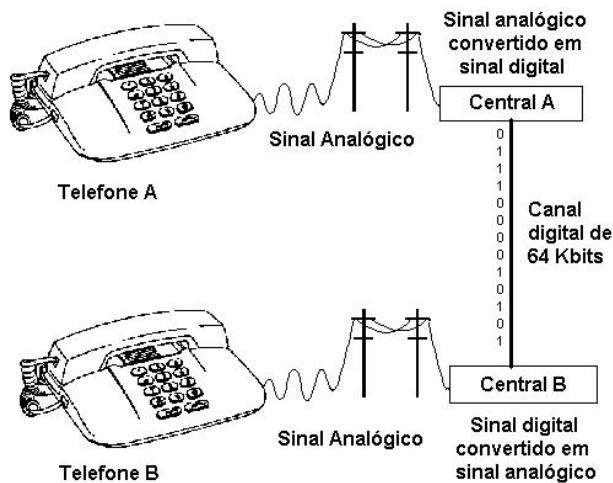
ifconfig

Vamos precisar lidar com os nomes das interfaces corretamente ao configurar as regras de firewall, compartilhar a conexão, criar tabelas de roteamento e muitas outras coisas.

Existem três formas de acesso discado. O mais tradicional são os **modems**, que realizam uma chamada de voz comum, utilizando a linha telefônica. Em seguida temos o ISDN, onde o modem cria um link digital de 64kbits com a central telefônica.

Na verdade, todas as chamadas de voz (incluindo aí as conexões via modem) são transmitidas de forma analógica apenas da sua casa até a central telefônica. Chegando na central, o sinal é digitalizado e transmitido através de um link de fibra óptica. Cada chamada de voz tem reservado para si um link de 64 kbits.

No caso dos modems, são feitas duas conversões: uma analógica/digital (ao chegar na central do seu bairro) e mais uma, digital/analogica (ao chegar na central usada pelo provedor de acesso). Esta dupla conversão atenua o sinal, fazendo com que a conexão fique limitada a 33.6 kbits.



Os modems de 56k são a exceção à regra. Para possibilitar o aumento na taxa de download, o modem instalado no provedor de acesso passa a ser ligado digitalmente à central telefônica, eliminando a conversão analógico/digital para as informações transmitidas do provedor para você (ou seja, o download). Mas, como continua existindo a conversão analógico/digital do seu micro até o provedor, a taxa de upload continua limitada a 33.6k.

No caso dos modems analógicos não existe muito o que fazer para aumentar a velocidade. Os 56k para download e 33.6 para upload parecem mesmo ser o limite final da tecnologia. O **ISDN** é o próximo passo. Nele o modem instalado na sua casa cria um link digital com a

central (na verdade um sinal digital transmitido dentro de um portador analógico), eliminando a conversão e permitindo aproveitar os 64k reservados para a chamada de voz.

Para tornar o serviço mais atrativo, as operadoras instalam duas linhas, de forma que você pode usar uma para conectar e outra para receber chamadas de voz, ou usar as duas simultaneamente para se conectar a 128k. O grande problema do ISDN é o custo: além do custo do modem e das taxas para habilitar o serviço, você continua pagando pulsos (em dobro se resolver conectar a 128k), o que explica por que o ISDN sempre foi tão pouco usado no Brasil.

Finalmente, temos o **ADSL**. Nele não é mais usado o sistema telefônico comutado, mas sim um link de fibra óptica, que liga a central telefônica diretamente aos roteadores do provedor de acesso. Como sairia muito caro puxar um cabo de fibra óptica até a casa de cada assinante, o modem ADSL estabelece um link digital com o modem instalado na central. A distâncias curtas (menos de 500 metros), o link é de 8 megabits, para até 3 km o link é de 2 megabits e, para até 5 km, o link é de apenas 1 megabit.

Na prática, a distância máxima varia muito, de acordo com a qualidade dos cabos e fontes de interferência pelo caminho, mas, de qualquer forma, as distâncias atingidas vão muito além do que seria possível com um sinal puramente digital. Lembre-se de que, para uma rede Ethernet, temos apenas 100 metros de alcance, mesmo utilizando um cabo de 4 pares, com uma qualidade muito superior à de um simples cabo telefônico. O sinal do modem ADSL vai tão longe porque na verdade o sinal digital é transmitido dentro de um portador analógico. Justamente por isso, o modem ADSL continua sendo um "modem", ou seja, **Modulador/Demodulador**.

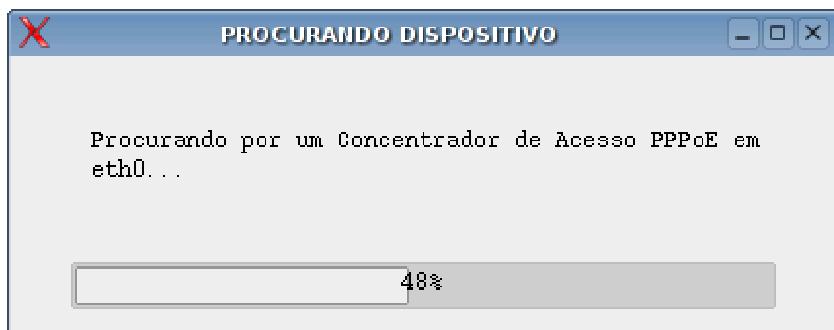
Este link "real" de 1 a 8 megabits é limitado a 128, 256, 300, 512, 600, 1024 ou 2048k, de acordo com o plano de acesso escolhido. A limitação é feita na própria central, por isso não existe como modificar o modem cliente para liberar mais banda.

Originalmente, o ADSL utilizava o sistema **ATM**, onde o cliente recebe um IP fixo e a conexão é contínua, como se fosse uma conexão de rede local. No ATM o modem funciona apenas como um bridge, um meio de ligação entre o equipamento da central e a placa de rede do seu micro. Basta configurar a rede usando a grade de configuração dada pelo provedor e você já está conectado.

Hoje em dia, o sistema ATM é usado apenas (e, mesmo assim, nem sempre) nos planos para empresas. Para o acesso residencial foi implantado o **PPPoE** (PPP sobre Ethernet), onde é simulado um acesso discado, em que é preciso "discar" e fornecer login e senha. No PPPoE a conexão não é necessariamente contínua e o IP muda periodicamente, ou cada vez que a conexão é estabelecida. Isso faz com que ele seja uma modalidade mais barata para os provedores, pois não é preciso mais ter um IP reservado para cada cliente. Outra vantagem (para eles) é que no PPPoE é possível contabilizar o tempo de conexão, permitindo que sejam criados planos com limitação de horas de acesso ou de dados transmitidos.

Para conectar via PPPoE no Linux é necessário usar o **pppoeconf** (usado no Debian e derivados) ou o **rp-pppoe** (usado na maioria das outras distribuições). Muitos utilitários gráficos incluídos nas distribuições servem como interface para um deles.

Ao usar o **pppoeconf**, chame o utilitário como root. Ele primeiro procura o modem, testando todas as placas de rede disponíveis, depois contata o modem na central, pede o login e senha, e depois estabelece a conexão.



Depois de conectar pela primeira vez, você pode terminar a conexão usando o comando "**poff -a**" e reconectar usando o "**pon dsl-provider**", mesmo depois de reiniciar o micro.

Caso ele não consiga detectar o modem (mesmo que ele esteja ativado e funcionando), pode ser que exista uma conexão anteriormente configurada ativa. Desative-a com o comando "**poff -a**". Em seguida, desative a interface de rede ligada ao modem ADSL usando o comando ifconfig; se o modem estiver ligado à interface eth0, por exemplo, o comando será "**ifconfig eth0 down**".

Nas distribuições que utilizam o **rp-pppoe**, use o comando "**adsl-setup**" para configurar a conexão e "**adsl-start**" para conectar. Para parar a conexão manualmente, use o comando "**adsl-stop**".

Nas versões mais recentes, existem dois utilitários, chamados "go" e "go-GUI" que automatizam a configuração. Caso tenha problemas com o rp-pppoe incluído na sua distribuição, experimente baixar a versão mais recente no: <http://www.roaringpenguin.com/penguin/pppoe/>.

Ao conectar via pppoe, é criada a interface de rede "ppp0", da mesma forma que ao conectar através de um modem discado. Do ponto de vista do sistema operacional, as duas conexões são similares.

Nos modems ADSL modernos existe a opção de manter o modem configurado como **bridge** (ponte), onde é necessário conectar manualmente, da forma como descrevi acima, ou configurar o modem como **router** (roteador), onde o próprio modem passa a fazer a autenticação e compartilhar a conexão, enviando a configuração de rede via DHCP para os micros conectados a ele. Ao configurar o modem como router, sua vida se torna muito mais simples e você não precisa se dar ao trabalho de configurar um dos micros da rede para compartilhar a conexão, pois o modem já faz isso sozinho. Basta ligar todos ao hub.

No caso do acesso discado, a principal dificuldade é instalar o driver necessário para ativar o modem. Até por volta de 1998, a maioria dos modems vendidos eram **hardmodems**. Eles são fáceis de configurar, pois o modem executa via hardware todas as funções necessárias. Você precisa apenas configurar o discador para procurar o modem na porta serial correta. As possibilidades vão da porta /dev/ttyS0 (com 1) à porta /dev/ttyS4 (com 5).

A forma mais rápida é simplesmente abrir o **kppp** e testar na base da tentativa e erro na opção "Dispositivo de Modem".



Infelizmente, os **hardmodems** são raridade hoje em dia, pois são muito mais caros. Os **softmodems** dominaram rapidamente o mercado, pois são mais simples e, por isso, custam uma fração do preço. Neles, a maior parte das funções são executadas por um software incluído no driver do modem. O sistema envia os comandos AT para o driver (e não mais diretamente ao modem) e ele (driver) se encarrega de modular o sinal, fazer a correção de erros e os demais passos necessários. Ao usar um softmodem, não adianta indicar a porta do modem no kppp: você precisa ter o driver instalado, ou nada feito.

Você pode encontrar uma explicação detalhada sobre os drivers disponíveis e como instalar cada um no meu livro *Linux, ferramentas técnicas*. Aqui vai um pequeno resumo dos drivers disponíveis para as distribuições Linux recentes, que utilizam o Kernel 2.6.14 em diante. Eles já vem pré-instalados no Kurumin, onde você pode ativá-los através da aba "Acesso discado", dentro do painel de controle. As instruções abaixo podem ser usadas quando você precisar usar o modem em outras distribuições ou solucionar problemas.

Você pode identificar o modem usando o comando "**lspci**", que lista todas as placas PCI instaladas na placa-mãe. Procure pela linha iniciada com "Modem" ou "Communication controller", como em:

```
00:05.0 Communication controller: Conexant HSF 56k HSFi Modem (rev 01)
```

A marca do modem, na maioria dos casos, não diz muito, pois o chipset do modem é projetado por um fabricante, produzido por vários outros e comprado por um número maior ainda de pequenos fabricantes que montam as placas, que são finalmente vendidas por um sem número de empresas que apenas colocam sua própria marca. Independentemente do seu modem ser LG, Kayomi, Clone ou o que quer que seja, o que importa mesmo é o chipset usado.

Os modems discados são provavelmente a categoria de dispositivos com suporte mais precário no Linux. É uma espécie de problema cultural. Quase todos os desenvolvedores e usuários avançados (que são os que podem desenvolver drivers e dar suporte) migraram rapidamente para as conexões via ADSL, cabo, rádio e outras modalidades de banda larga assim que elas se tornaram acessíveis. Quem continua usando os modems é, na sua maioria, o público mais leigo, que acessa pouco e, por isso, chega à conclusão de que não vale a pena pagar um ADSL.

Os fabricantes vêem os modems como uma forma de commodity, um tipo de dispositivo barato, vendido com margem reduzida de lucro, no qual não vale a pena fazer grandes investimentos. Muitas vezes não investem sequer em resolver os problemas do driver for Windows, quanto mais em desenvolver e dar suporte para uma versão Linux.

Desenvolver um driver para um softmodem é uma tarefa complexa, pois é preciso implementar via software todas as funções que o modem propriamente dito não executa, como a modulação do sinal, correção de erros e muito mais. Para você ter uma idéia, o módulo para uma placa de rede SiS900 tem 22k, enquanto que o módulo que dá suporte ao modem Intel 537EP tem 1.5 MB, mais de 60 vezes maior.

No final, temos uma situação em que os fabricantes têm pouco interesse em desenvolver um driver e não divulgam as especificações. Poucos desenvolvedores têm interesse em encarar o herculeo trabalho de desenvolver um driver fazendo engenharia reversa simplesmente porque não acessam via modem e, para completar, cada vez mais gente acessa via banda larga, fazendo com que a demanda por drivers seja cada vez menor.

Mesmo assim, vários modems possuem suporte no Linux. Pesquisando um pouco, você pode comprar diretamente um modem compatível e assim evitar muitas dores de cabeça. Se você tem em mãos um modem que não possui driver, não perca tempo com ele: venda ou troque com alguém que usa Windows e compre um modem suportado. Se puder, resolva o problema definitivamente, comprando um hardmodem externo (ligado ao PC através da porta serial), como alguns modelos comercializados pela Dlink, Trendnet e US Robotics, ou migrando para algum plano de banda larga.

Voltando aos drivers, a partir do Kernel **2.6.14** passou a vir incluído um pequeno conjunto de drivers open-source, desenvolvidos pela equipe do Alsa, responsável pelo

desenvolvimento dos drivers para placas de som. Até certo ponto, um modem tem uma função similar à de uma placa de som: ele transforma sinais digitais em sinais analógicos e vice-versa. Um modem inclui muitas funções adicionais, como modulação de dados, compressão e correção de erros, mas os desenvolvedores têm conseguido superar as dificuldades.

Os drivers disponíveis são:

snd-intel8x0m: Este driver dá suporte aos modems Intel AC97, encontrados em muitos notebooks (incluindo a maioria dos Centrinos), aos modems onboard encontrados em placas com chipset nVidia nForce e também a alguns modems PCI com chipset Intel ou PC-Tel. Uma observação é que, em alguns casos, carregar o driver do modem faz com que a placa de som pare de funcionar.

snd-atiixp-modem: Ele dá suporte aos modems onboard encontrados em notebooks com o chipset ATI IXP, como o Toshiba A70. Apesar da ATI ser uma novata no ramo de chipsets, o modem é bem suportado e mantém conexões estáveis.

snd-via82xxx-modem: Este é um driver ainda em estágio inicial de desenvolvimento, que dá suporte aos modems onboard encontrados em placas-mãe recentes, com chipset Via. Note que muitas placas, sobretudo as PC-Chips, incluem modems AMR, que funcionam com o driver para modems Intel AC'97 ou com o slamr (que veremos a seguir).

snd-alii5451-modem: Também em estágio inicial de desenvolvimento, dá suporte aos modems onboard de placas com o chipset ALI 5451, encontrado em algumas placas de baixo custo.

Para usar qualquer um dos quatro, comece carregando o driver usando o comando "modprobe" (como root), como em:

```
# modprobe snd-intel8x0m
```

Em seguida, você precisa instalar o "**slmodemd**", o utilitário que faz a interface entre o driver e o sistema, criando o dispositivo de comunicação. Se você está usando uma distribuição derivada do Debian, pode instalá-lo via apt-get:

```
# apt-get install sl-modem-daemon
```

Em outras distribuições, procure pelo pacote "slmodem" ou "sl-modem". Caso ele não esteja disponível, resta a opção de instalá-lo a partir do código fonte. Nesse caso, você vai precisar ter instalados os pacotes de desenvolvimento. No Ubuntu, instale os pacotes "build-essential", "gcc", "g++" e "libasound2-dev":

```
# sudo apt-get install build-essential gcc g++ libasound2-dev
```

O próximo passo é baixar o pacote "slmodem-2.9.9d-alsa.tar.gz" (ou a versão mais recente no momento em que estiver lendo este texto) no

<http://linmodems.technion.ac.il/packages/smartlink/>. Note que você precisa baixar um dos arquivos com "alsa" no nome.

Descompacte o arquivo, acesse a pasta que será criada e rode os comandos:

```
$                               cd                               modem/  
$                               make                            SUPPORT_ALSA=1  
$                               su                                <senha>  
(no           Ubuntu          use          "sudo          su")  
# make install
```

Depois de instalado, execute-o incluindo o parâmetro "--alsa", que especifica que ele deve usar o driver do alsas, como em:

```
#                           killall                         slmodemd  
# slmodemd --country=BRAZIL --alsa modem:1
```

O "killall slmodemd" é importante, pois se houver outra instância ativa, ele não conseguirá acessar o modem e retornará um erro. O "**modem:1**" especifica o dispositivo do modem (da forma como é referenciado pelo driver). Dependendo da versão do driver usada, o modem pode ser visto como "**modem:1**", "**hw:1**", "**modem:0**" (atribuído geralmente ao ATI IXP) ou "**hw:0**" (comum em notebooks Centrino). Você pode testar as 4 possibilidades até encontrar o correto no seu caso.

Ao executar o comando, incluindo o parâmetro correto, você verá uma mensagem como:

```
SmartLink      Soft      Modem:      version      2.9.9d      Sep      27      2005      00:00:18  
symbolic      link      `'/dev/ttysL0'          ->          `/dev/pts/4'      created.  
modem `modem:1' created. TTY is `/dev/pts/4'
```

Use `/dev/ttysL0` as modem device, Ctrl+C for termination.

Como pode ver, o slmodemd é um programa que fica residente. Ao fechá-lo, o acesso ao modem é desativado. Se não quiser que ele obstrua o terminal, use o "&" no final do comando. O `/dev/ttysL0` é o dispositivo por onde o modem é acessado. Crie o link `/dev/modem` apontando para ele, assim fica muito mais fácil localizar o modem dentro do programa de discagem:

```
# ln -sf /dev/ttysL0 /dev/modem
```

A partir daí, você pode discar usando o KPPP. Este é o resultado do relatório gerado pelo "perguntar ao modem" do KPPP de um Intel AC'97 usado no HP NX6110. Como pode ver, ele é detectado como se fosse um modem Smartlink, por causa do uso do slmodemd. A pista para o driver que está realmente sendo usado é a linha "modem:1 alsas modem driver".



Outro driver muito usado é o "**slamr**", uma espécie de "curinga", um driver desenvolvido pela Smartlink que funciona com os modems PC-Tel onboard (ele consegue ativar simultaneamente o modem e o som onboard, ao contrário do driver antigo) e também em modems PCI LG Netodragon e até mesmo com alguns modelos de modems Intel. Você pode baixar a versão mais recente do driver no mesmo link do pacote do slmodemd: <http://linmodems.technion.ac.il/packages/smartlink/>.

Neste caso, você baixa o arquivo sem "alsa" no nome, como em "slmodem-2.9.11.tar.gz". Você precisa da versão 2.9.11 ou mais recente, pois as antigas não compilam em distribuições com o Kernel 2.6.13 ou mais recente.

Para instalar o driver, você precisa ter instalado (além dos compiladores que vimos há pouco) o pacote com os headers (ou o código fonte completo, dependendo da distribuição) do Kernel em uso. No Kurumin este é um item "de série". No Ubuntu, use o comando "uname -a" para verificar a versão do Kernel carregado na memória e instale os headers via apt-get, especificando a versão, como em:

```
$ sudo apt-get install linux-headers-2.6.15-23-386
```

Com tudo nos lugares, descompacte o arquivo, acesse a pasta que será criada e rode os comandos "make" e "make install" (este último como root), como em:

```
$ tar -zxvf slmodem-2.9.11.tar.gz
$ cd slmodem-2.9.11/
$ make
<senha>
# make install
```

Antes de discar, você precisa carregar o driver e executar o slmodemd. Crie um script (como vimos no tópico sobre configuração de placas Wireless) para não precisar ficar digitando-os manualmente a cada conexão.

```
# modprobe
# slmodemd --country=BRAZIL /dev/slamr0 &
```

Ao abrir o slmodemd é criado o dispositivo "/dev/ttySL0". Crie o link "/dev/modem" apontando para ele:

```
# ln -sf /dev/ttySL0 /dev/modem
```

Temos ainda os drivers comerciais para modems **Conexant HSF** e **HCF**, desenvolvidos pela Linuxant. Os drivers funcionam, mas custam US\$ 19, o que acaba sendo mais caro que comprar outro modem. No site está disponível uma versão demo, onde a conexão fica limitada a 14.4k: <http://www.linuxant.com/drivers>.

A Intel chegou a desenvolver drivers para os modems 537 e 537EP, os famosos "Intel Ambient", muito comuns há algum tempo atrás. Os drivers ainda estão disponíveis no <http://linmodems.technion.ac.il/packages/Intel>, mas não possuem muita utilidade hoje em dia, pois não dão suporte aos modems Intel atuais (os mais baratos, que usam um chip DSP pequenininho) e não compilam nas versões recentes do Kernel, da 2.6.13 em diante.

Outro exemplo de driver obsoleto é o driver para modems Lucent/Agere, que não funciona com os modelos atuais (os V92, com chipset "sv92") e que, por isso, não tem mais tanta utilidade hoje em dia. De qualquer forma, caso você tenha um modem antigo, fabricado entre 2000 e 2002, pode baixar o driver disponível no: <http://linmodems.technion.ac.il/packages/lmodem/kernel-2.6/>.

» Próximo: [Configurando o modem ADSL](#)

Quase todos os modems ADSL vendidos atualmente podem ser configurados como roteador, compartilhando a conexão entre os micros da rede local, sem a necessidade de usar um micro com duas placas de rede para isso.

Em geral os modems ADSL fazem um bom trabalho, eles não oferecem opções mais avançadas, como, por exemplo, incluir um proxy transparente, para fazer cache das páginas e arquivos acessados e, assim, melhorar a velocidade de acesso, mas são capazes de fazer o arroz com feijão, como bloquear tentativas de acesso vindas da internet e redirecionar portas para micros da rede local.

As **vantagens** de usar o modem configurado como roteador são:

1- Não é preciso usar o pppoeconf para se conectar, nem configurar o compartilhamento da conexão. A conexão é estabelecida pelo próprio modem, basta ligá-lo no hub e configurar os demais PCs para obterem a configuração da rede via DHCP.

2- O modem fica com as portas de entrada, de forma que qualquer tipo de ataque proveniente da internet é bloqueado pelo próprio modem, antes de chegar nos micros da rede local. O modem serve como uma camada adicional de proteção.

As **desvantagens** são:

1- Como as portas de entrada ficam com o modem, é preciso configurar o redirecionamento de portas para que você possa usar qualquer servidor ou programa que precise de portas de entrada. Um exemplo clássico é o bittorrent, que precisa que pelo menos uma das portas entre a 6881 e a 6889 esteja aberta.

2- Ao contrário dos servidores Linux, os modems ADSL não costumam receber atualizações de segurança. Não é impossível que uma brecha de segurança no próprio modem permita que alguém de fora altere a configuração de redirecionamento de portas (por exemplo) e assim consiga ter acesso aos micros da sua rede local. Alguns modems permitem inclusive a instalação de programas adicionais. Do ponto de vista da segurança, um servidor Linux atualizado e bem configurado é mais seguro.

O modem ADSL pode ser configurado através de uma interface de configuração, que fica acessível apenas a partir da rede local. Em primeiro lugar, crie a comunicação física entre o PC e o modem, ligando-os através do cabo cross que acompanha o modem ou usando um hub.

O modem vem de fábrica com um endereço IP padrão, como, por exemplo, 10.0.0.138 ou 192.168.1.1 e uma senha de acesso simples, como "1234" ou "admin". A configuração padrão varia de modem para modem, por isso é importante ter em mãos o manual do seu.

Geralmente, as operadoras alteram as senhas dos modems, para impedir que o usuário o reconfigure para trabalhar como roteador. Nesse caso, você vai ter o trabalho de pesquisar na web quais as senhas usadas pela operadora e testar uma a uma até achar a usada no seu modem, uma dor de cabeça a mais.

Dependendo do modem, o utilitário de configuração pode ser acessado de duas formas: via telnet (como no Parks Prestige) ou por meio de uma página web acessível pelo navegador (como na maioria dos modelos novos).

Os modems vêm sempre configurados de fábrica em modo bridge, em que a configuração da conexão (login, senha, etc.) é feita no PC. Isso facilita a vida dos fabricantes em termos de suporte. Basta instalar o programa de discagem e conectar.

Ao configurar o modem roteador você precisa fornecer a configuração da rede local, o endereço IP usado pelo modem e a faixa de endereços que serão atribuídos via DHCP para os clientes da rede local, os endereços dos servidores DNS do provedor, seu login e senha de acesso e os códigos VPI/VCI usados pela operadora.

Os valores VPI/VCI usados atualmente no Brasil são:

Telefonica	(Speedy):	VCI	35,	VPI	8
Telemar	(Velox):	VPI	0,	VCI	33
Brasil	Telecom:	VPI	0,	VCI	35

Brasil Telecom (no RS): VPI 1, VCI 32
GVT: VPI 0, VCI 35

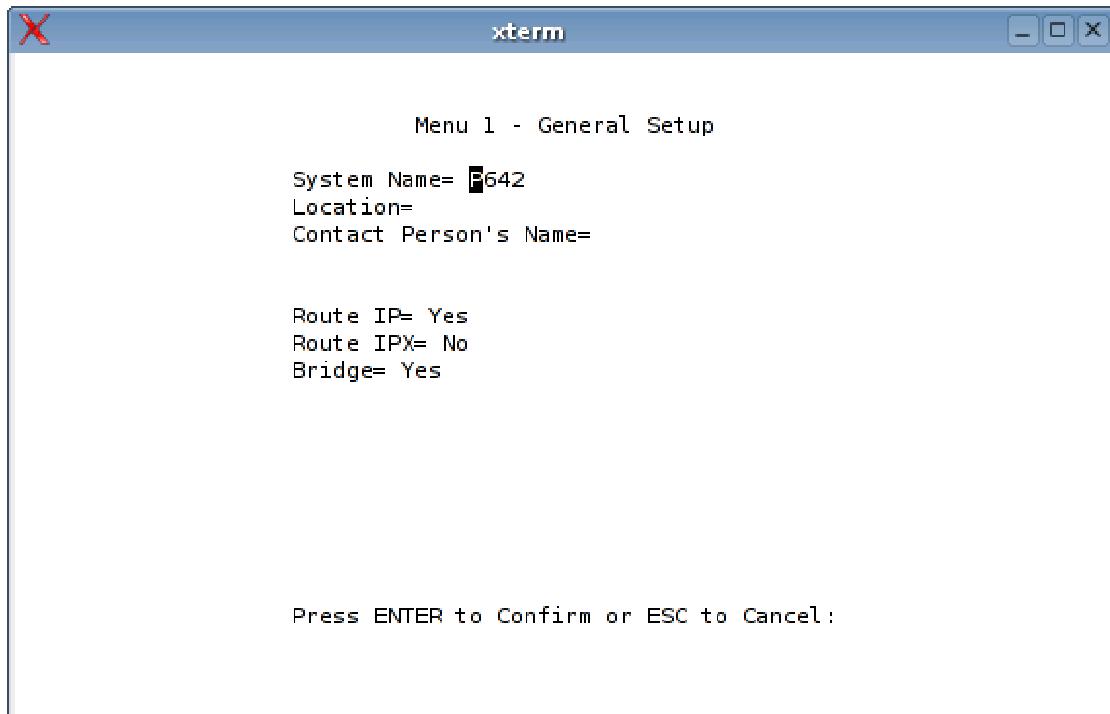
Você pode confirmar esses valores ligando para o suporte técnico. É fácil obter estas informações.

Vamos a um exemplo de configuração, usando um modem **ZyXEL Parks Prestige**. Este é um modelo da velha guarda, que é acessível via Telnet. O IP padrão de fábrica é 192.168.1.1, de forma que para ter acesso a ele você deve configurar seu PC para usar um IP qualquer na faixa 192.168.1.x. A senha padrão é "1234" ou "adminttd". O telnet pode ser usado tanto a partir do terminal, no Linux, quanto a partir do prompt do MS-DOS, no Windows. Para isso use o comando:

\$ telnet 192.168.1.1

Em primeiro lugar, antes de conectar na internet ou configurar qualquer coisa, altere a senha de acesso na opção "**23. System Password**".

De volta à tela inicial, comece acessando a opção "**1. General Setup**" e mude a opção "**Route IP**" para "**Yes**". É aqui que configuramos o modem para trabalhar como roteador. Mantenha a opção "Bridge" como "Yes". Pressione Enter para confirmar a alteração e voltar ao menu inicial.



Acesse agora a opção "**3. Ethernet Setup**" e, dentro dela, a opção "**2. TCP/IP and DHCP Setup**". Aqui vai toda a configuração relacionada à rede local.

Na opção "DHCP Setup" configuramos o servidor DHCP incluído no modem, que vai atribuir os endereços para os PCs da rede interna. A opção "Client IP Pool Starting Address=" indica o primeiro endereço IP que será atribuído (os números abaixo deste ficam reservados para micros com IP fixo) e o número máximo de clientes que receberão endereços IP (Size of Client IP Pool), que deve ser compatível com o número de micros na sua rede. Não faz mal usar um valor alto aqui, só não vale colocar cinco endereços em uma rede com 20 micros ;).

Em seguida são incluídos os endereços dos servidores DNS que serão fornecidos aos clientes. Normalmente você usa os servidores DNS do provedor, mas, caso você tenha configurado um servidor DNS local (veremos como fazer isso mais adiante), você pode usá-lo aqui.

Nas opções "IP Address" e "IP Subnet Mask" vai o endereço IP e a máscara que serão usados pelo próprio modem. Lembre-se que, ao configurar o modem para compartilhar a conexão, o endereço IP do modem passa a ser o gateway padrão da rede.

```
Menu 3.2 - TCP/IP and DHCP Ethernet Setup

DHCP Setup
  DHCP= Server
  Client IP Pool Starting Address= 192.168.1.33
  Size of Client IP Pool= 6
  Primary DNS Server= 200.199.201.23
  Secondary DNS Server= 192.168.1.2
  Remote DHCP Server= N/A

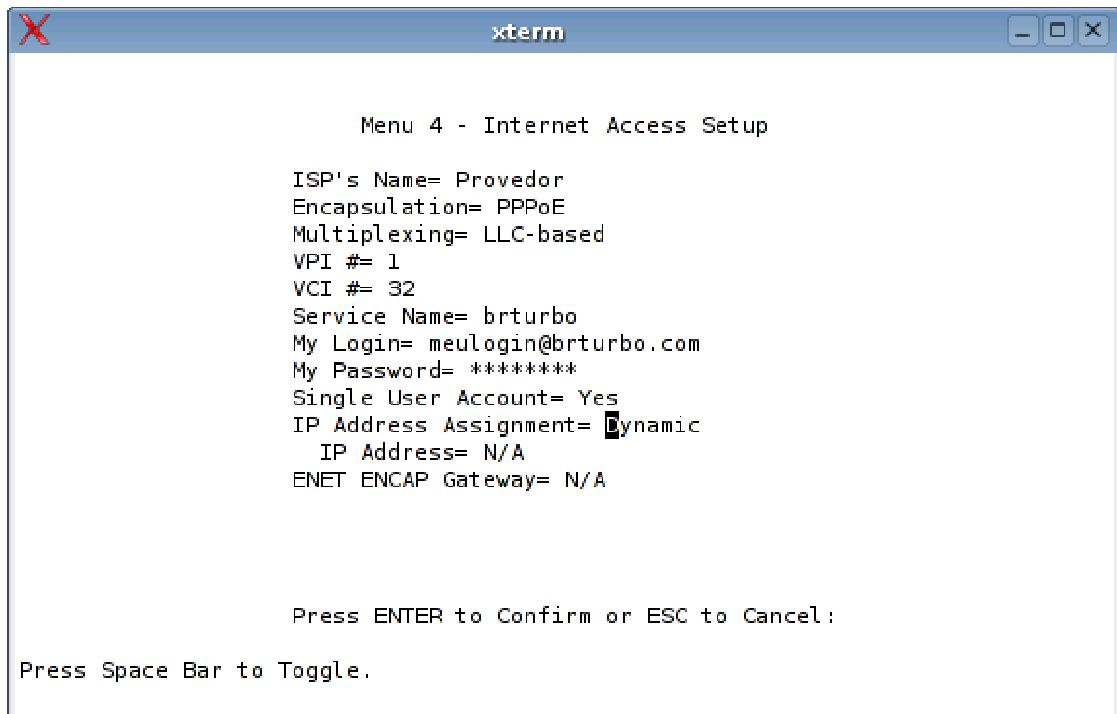
TCP/IP Setup:
  IP Address= 192.168.1.1
  IP Subnet Mask= 255.255.255.0
  RIP Direction= None
  Version= N/A
  Multicast= None
  IP Policies=
  Edit IP Alias= No

Press ENTER to Confirm or ESC to Cancel:
```

De volta ao menu principal, acesse agora a opção "**4. Internet Access Setup**". Aqui vão as informações sobre o provedor de acesso, que o modem usará para estabelecer a conexão. As opções "ISP's Name" e "Service Name" são apenas para seu controle, caso tenha configurada mais de uma conexão. Pode usar os nomes que quiser.

O importante mesmo são as opções "Encapsulation" (no Brasil é quase sempre usado PPPoE; a opção RCF é usada em algumas instalações antigas, com IP fixo) e "Multiplexing", que no Brasil fica sempre como "LLC-based". Na opção "IP Address Assignment" use a opção "Dynamic" se você usa um plano residencial, com IP dinâmico ou "Static" se usa um plano empresarial, com IP fixo. Neste segundo caso, forneça também o

IP usado. Não se esqueça de incluir também os códigos VPI e VCI usados pela operadora, além do login e senha de acesso.



The screenshot shows an xterm window titled "xterm" with the title bar "Menu 4 - Internet Access Setup". The window displays the following configuration parameters:

```
ISP's Name= Provedor
Encapsulation= PPPoE
Multiplexing= LLC-based
VPI #= 1
VCI #= 32
Service Name= brturbo
My Login= meulogin@brturbo.com
My Password= *****
Single User Account= Yes
IP Address Assignment= Dynamic
IP Address= N/A
ENET ENCAP Gateway= N/A
```

At the bottom of the window, there are two lines of instructions:

Press ENTER to Confirm or ESC to Cancel:
Press Space Bar to Toggle.

Para configurar o redirecionamento de portas para os micros da rede local (port forwarding), acesse a opção "**15. SUA Server Setup**", onde você define a porta e o endereço da rede local para onde ela será redirecionada.

Por exemplo, para permitir que o micro 192.168.1.30 fique acessível via SSH, basta redirecionar a porta 22 para ele. Quando alguém tentar acessar a porta 22 do endereço IP de internet, atribuído ao modem, cairá no servidor aberto no micro da rede local.

Você pode direcionar portas diferentes para endereço IP diferentes, mas nunca direcionar a mesma porta para mais de um endereço ao mesmo tempo. Uma limitação deste modem é que ele permite configurar o redirecionamento de apenas 8 portas simultaneamente.

Port #	IP Address
1.Default	0.0.0.0
2. 6881	192.168.1.2
3. 8080	192.168.1.3
4. 2121	192.168.1.25
5. 22	192.168.1.30
6. 1720	192.168.1.233
7. 6886	192.168.1.35
8. 25	192.168.1.34

Press ENTER to Confirm or ESC to Cancel:

HTTP:80 FTP:21 Telnet:23 SMTP:25 POP3:110 PPTP:1723

As operadoras quase sempre bloqueiam as portas 21 e 80 (ftp e http), para impedir que os assinantes dos planos residenciais mantenham servidores. Mas, você pode alterar a porta usada na configuração do Apache ou do Proftpd para 8080 e 2121, por exemplo, para burlar esta limitação. Para verificar seu endereço IP atual, acesse o <http://myipaddress.com> ou o <http://www.whatismyip.com>.

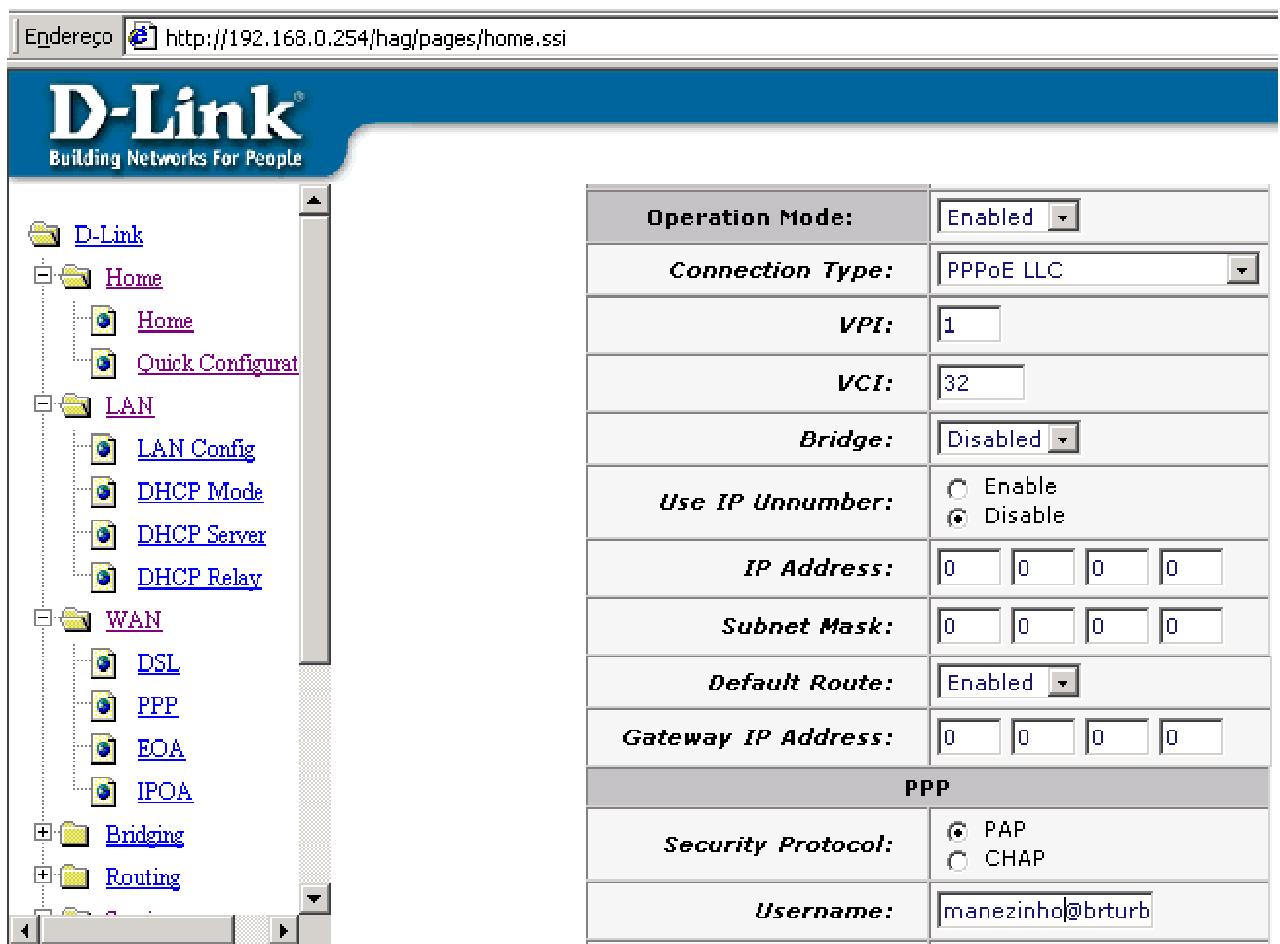
Você pode ver uma lista de portas de entrada usadas por vários programas e jogos no: <http://www.portforward.com/cports.htm>.

Um segundo exemplo é o modem **D-Link DSL 500G**, que utiliza uma interface de configuração via navegador.

O endereço padrão do modem é 10.1.1.1, login "admin", senha "admin". Para acessá-lo, configure seu micro para utilizar o endereço 10.1.1.2, com máscara 255.0.0.0 (ou outro dentro da mesma faixa) e acesse a url: <http://10.1.1.1>.

Para compartilhar a conexão, acesse a opção "Home > Quick Configuration", onde vão as informações de acesso. Em "Connection Type" escolha "PPPoE LCC", deixe a opção "Bridge" como "Disabled", a opção "Default route" como "Enabled" e mantenha a opção "use DNS" como "Enable", preenchendo os dois campos com os endereços DNS do provedor. Não se esqueça de fornecer seu login e senha de acesso dentro da seção "PPP", além dos códigos VPI e VCI do seu provedor.

Acesse agora a seção "Services > NAT", onde é ativado o compartilhamento da conexão. No campo "NAT Options" selecione a opção "Nat Global Info" e marque a opção "Enable". Para que a configuração entre em vigor, acesse a opção "Admin > Commit & Reboot".



Para configurar o roteamento de portas, acesse a opção "Services > NAT". Dentro da página, selecione a opção "NAT Rule Entry" na caixa de seleção e clique no botão "Add". Ao contrário do Parks 600, este modem já permite direcionar um número indefinido de portas, basta usar um número de identificação diferente para cada regra.

Por padrão já existe uma regra, com o número "1". Ao adicionar uma nova regra, você deve usar o número "2", depois "3" e assim por diante. O número de identificação é informado na opção "Rule ID".

Para fazer redirecionamento de portas, use a opção RDR em "Rule Flavor". Indique o endereço da estação que receberá a porta nas opções "Local Address From" e "Local Address To" (o mesmo endereço nas duas opções) e a porta que será redirecionada nas opções "Destination Port From", "Destination Port To" e "Local Port".

Este modem permite também redirecionar uma porta para outra porta, de número diferente na estação. Você pode, por exemplo, direcionar a porta 2222 no roteador para a porta 22 da estação 192.168.0.10 e a porta 2223 do roteador para a porta 22 da estação 192.168.0.11. Ou seja, você poderia acessar (via Internet) os servidores SSH habilitados nas duas estações (e na mesma porta em ambas) através de diferentes portas do roteador.

Neste caso, você informa a porta do roteador nas opções "Destination Port From" e "Destination Port To", além da porta da estação que receberá o redirecionamento na opção "Local Port".

NAT Rule - Add

NAT Rule Information			
Rule Flavor:	<input type="button" value="RDR"/>		
Rule ID:	2		
IF Name:	<input type="button" value="ALL"/>		
Protocol:	<input type="button" value="ANY"/>		
Local Address From:	192	168	0
			10
Local Address To:	192	168	0
			10
Global Address From:	0	0	0
			0
Global Address To:	0	0	0
			0
Destination Port From:	<input type="button" value="Any other port"/>		2222
Destination Port To:	<input type="button" value="Any other port"/>		2222
Local Port:	<input type="button" value="Any other port"/>		22
<input type="button" value="Submit"/> <input type="button" value="Cancel"/>			

Para que as regras entrem em vigor, é necessário reiniciar o modem, através da opção "Admin > Commit & Reboot".

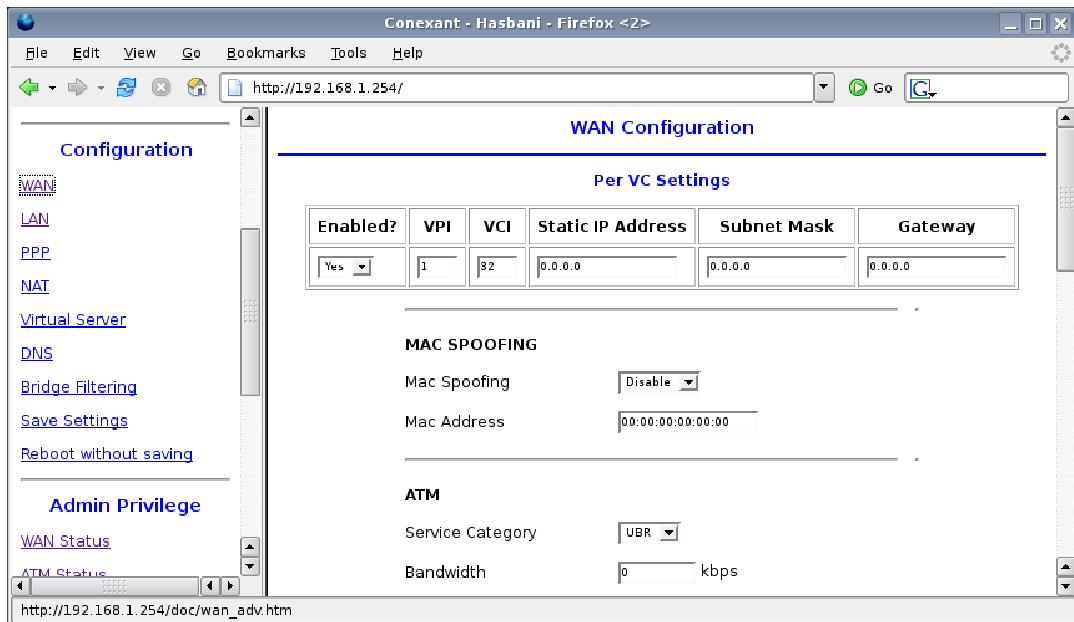
Uma observação importante sobre a interface de administração do 500G e alguns outros modelos da D-Link é que, embora não utilize ActiveX nem nenhum outro recurso especial, a interface de administração não funciona corretamente em todos os navegadores. Para ser mais exato, ela funciona corretamente apenas em algumas versões do IE e do Opera. No Firefox, Mozilla, Konqueror e também nas versões mais recentes do IE, aparecem erros diversos.

No Firefox, por exemplo, a Interface demora absurdamente para abrir e, ainda assim, não é carregada completamente. Uma configuração que ajuda é acessar o "about:config" (esta opção dá acesso às opções avançadas de configuração do navegador, digite na barra de endereços) e alterar a opção "network.http.version" de "1.1" para "1.0". Depois da modificação, a interface passa a funcionar normalmente, mas a opção de redirecionar portas ainda continua retornando um erro de "função não suportada".

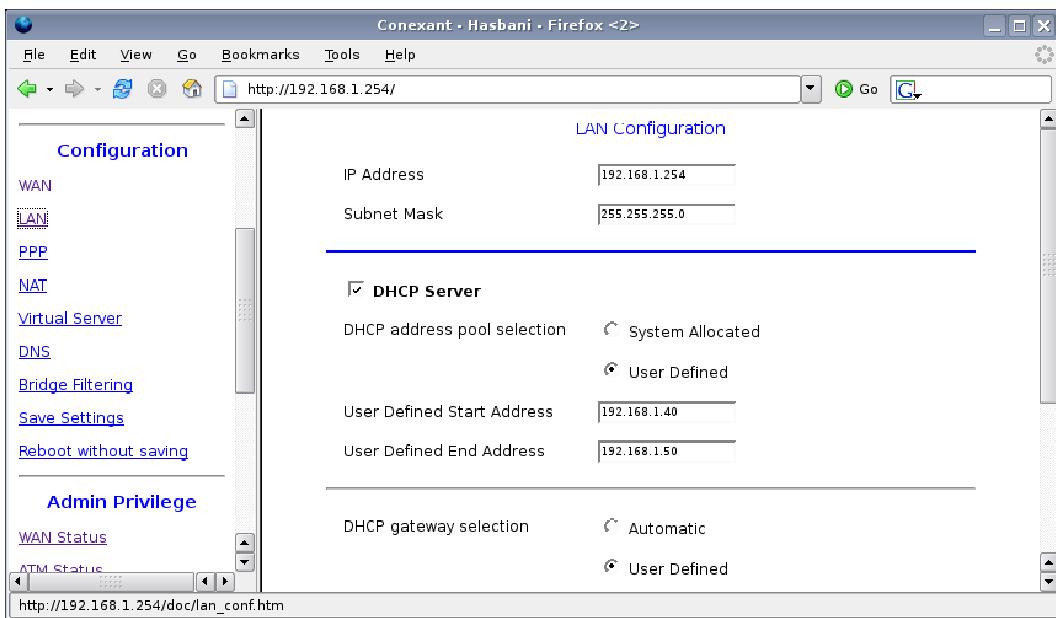
Um atenuante para o problema é que a configuração do modem pode ser acessada a partir de qualquer micro ligado ao hub, incluindo máquinas virtuais do VMware. Isso permite

testar diferentes navegadores e sistemas operacionais que você tenha em mãos, até encontrar um navegador que funcione perfeitamente. Eu, por exemplo, usei o IE 5 de uma instalação do Windows 2000 dentro de uma máquina virtual do VMware. Felizmente, estes problemas crônicos do Dlink 500G não são comuns em outros modems.

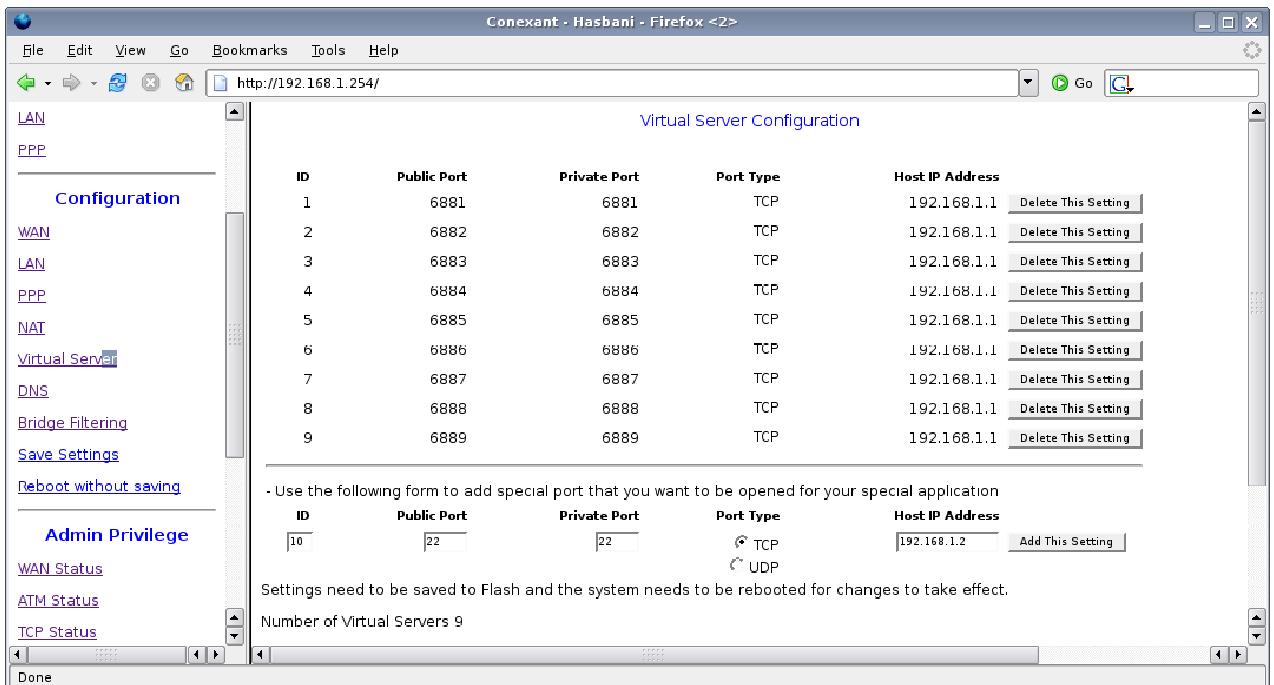
Um terceiro exemplo é um modem Kayomi LP-AL2011P (que é, na verdade, um Conexant Hasbani). Nele, a configuração básica vai dentro da seção "WAN", onde você define os valores VPI e VCI do provedor, o tipo de encapsulamento (escolha "PPPoE LCC), além do login e senha de acesso:



Na seção "LAN" você define as configurações para a rede local, incluindo o endereço IP usado pelo modem e a faixa de endereços que será usada pelo servidor DHCP:



A configuração do forwarding de portas vai na seção "Virtual Server". A interface é bastante simples, você só precisa indicar a porta externa que será redirecionada e a porta e o IP do micro local que a receberá. Neste exemplo, estou redirecionando as portas do bittorrent para o micro "192.168.1.1":



Note que a porta do micro na rede local, para onde é feito o forwarding, não precisa necessariamente ser a mesma que a porta externa. Você pode fazer com que a porta 22 externa seja direcionada para a porta 2222 do micro 192.168.1.2, por exemplo.

Depois de terminar, use a opção "Save Settings" no menu. Este modem precisa de um soft-reset para ativar as alterações.

» Próximo: [Capítulo 3: Entendendo o endereçamento IP](#)

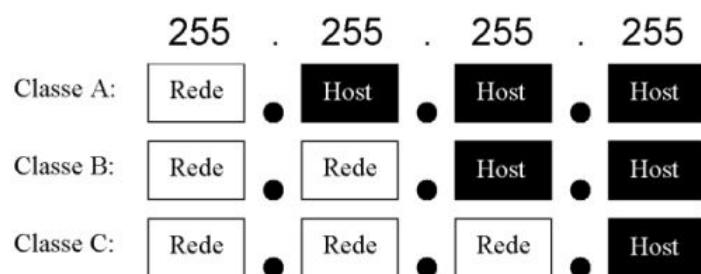
Como já vimos, dentro de uma rede TCP/IP, cada micro recebe um endereço IP único que o identifica na rede. Um endereço IP é composto de uma seqüência de 32 bits, divididos em 4 grupos de 8 bits cada. Cada grupo de 8 bits recebe o nome de **octeto**.

O endereço IP é dividido em duas partes. A primeira identifica a rede à qual o computador está conectado e a segunda identifica o host dentro da rede. Para melhorar o aproveitamento dos endereços disponíveis, os desenvolvedores do TPC/IP dividiram o endereçamento IP em cinco classes, denominadas A, B, C, D, e E, sendo as três primeiras são usadas para fins de endereçamento e as duas últimas são reservadas para expansões futuras. Cada classe reserva um número diferente de octetos para o endereçamento da rede.

Na **classe A**, apenas o primeiro octeto identifica a rede, na **classe B** são usados os dois primeiros octetos e na **classe C** temos os três primeiros octetos reservados para a rede e apenas o último reservado para a identificação dos hosts dentro da rede.

O que diferencia uma classe de endereços da outra é o valor do primeiro octeto. Se for um número entre 1 e 126 temos um endereço de classe A. Se o valor do primeiro octeto for um número entre 128 e 191, então temos um endereço de classe B e, finalmente, caso o primeiro octeto seja um número entre 192 e 223, teremos um endereço de classe C.

Octetos:

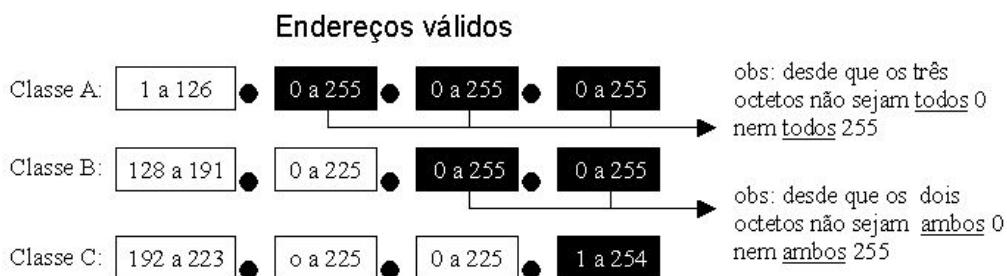


Ao configurar uma rede local, você pode escolher a classe de endereços mais adequada. Para uma pequena rede, uma faixa de endereços de classe C é a mais apropriada, pois você precisa se preocupar em configurar apenas o último octeto do endereço ao atribuir os endereços. Em uma rede de maior porte, com mais de 254 micros, passa a ser necessário usar um endereço de classe B, onde podemos usar diferentes combinações de números nos dois últimos octetos, permitindo um total de 65.534 endereços.

É muito difícil encontrar uma situação onde seja necessário usar uma faixa de endereços de classe A, pois redes muito grandes acabam sendo divididas em vários segmentos diferentes, interligados por roteadores. Neste caso, cada segmento é endereçado como se fosse uma rede separada, usando faixas de classe C ou B.

Na internet, todos os endereços IP disponíveis já possuem dono. Ao contratar algum tipo de conexão você recebe um único endereço (como numa linha ADSL) ou uma faixa de classe C inteira (ao alugar um backbone por exemplo). Os endereços de classe B são reservados às grandes empresas e provedores de acesso, enquanto os endereços de classe A são praticamente impossíveis de se conseguir, mesmo para grandes corporações.

Ao alugar um backbone vinculado a uma faixa de endereços classe C, por exemplo, você recebe uma faixa de endereços como "203.107.171.x", onde o "203.107.171" é o endereço de sua rede dentro da internet, e o "x" é a faixa de 254 endereços que você pode usar para identificar seus servidores. Na ilustração temos um resumo das regras para endereços TCP/IP válidos:



Como você pode notar no diagrama, nem todas as combinações de endereços são permitidas, pois o primeiro endereço (0) é reservado à identificação da rede, enquanto o último (255) é reservado ao endereço de broadcast, que é usado quando alguma estação precisa enviar um pacote simultaneamente para todos os demais micros da rede.

Os pacotes de broadcast são usados para, por exemplo, configurar a rede via DHCP e localizar os compartilhamentos de arquivos dentro de uma rede Windows. Mesmo os switches e hub-switches detectam os pacotes de broadcast e os transmitem simultaneamente para todas as portas. A desvantagem é que, se usados extensivamente, eles prejudicam a velocidade da rede.

Veja alguns exemplos de endereços **inválidos**:

0.xxx.xxx.xxx: Nenhum endereço IP pode começar com zero, pois ele é usado para o endereço da rede. A única situação em que um endereço começado com zero é usado, é quando um servidor DHCP responde à requisição da estação. Como ela ainda não possui um endereço definido, o pacote do servidor é endereçado ao endereço MAC da estação e ao endereço IP "0.0.0.0", o que faz com que o switch o envie para todos os micros da rede.

127.xxx.xxx.xxx: Nenhum endereço IP pode começar com o número 127, pois este número é reservado para a interface de loopback, ou seja, são destinados à própria máquina que enviou o pacote. Se por exemplo você tiver um servidor de SMTP e configurar seu programa de e-mail para usar o servidor 127.0.0.1, ele acabará usando o servidor instalado na sua própria máquina. O mesmo acontece ao tentar acessar o endereço 127.0.0.1 no navegador: você vai cair em um servidor web habilitado na sua máquina. Além de testes em geral, a interface de loopback é usada

para comunicação entre diversos programas, sobretudo no Linux e outros sistemas Unix.

255.xxx.xxx.xxx, xxx.255.255.255, xxx.xxx.255.255: Nenhum identificador de rede pode ser 255 e nenhum identificador de host pode ser composto apenas de endereços 255, seja qual for a classe do endereço, pois estes endereços são usados para enviar pacotes de broadcast. Outras combinações são permitidas, como em 65.34.255.197 (em um endereço de classe A) ou em 165.32.255.78 (endereço de classe B).

xxx.0.0.0, xxx.xxx.0.0: Nenhum identificador de host pode ser composto apenas de zeros, seja qual for a classe do endereço, pois estes endereços são reservados para o endereço da rede. Como no exemplo anterior, são permitidas outras combinações como 69.89.0.129 (classe A) ou 149.34.0.95 (classe B).

xxx.xxx.xxx.255, xxx.xxx.xxx.0: Nenhum endereço de classe C pode terminar com 0 ou com 255, pois, como já vimos, um host não pode ser representado apenas por valores 0 ou 255, já que eles são usados para o envio de pacotes de broadcast.

Se você não pretende conectar sua rede à internet, pode utilizar qualquer faixa de endereços IP válidos e tudo irá funcionar sem problemas. Mas, a partir do momento em que você resolver conectá-los à web, os endereços da sua rede poderão entrar em conflito com endereços já usados na web.

Na prática isto não acontece, pois os roteadores do provedor de acesso perceberão que estão sendo usados endereços inválidos e se recusarão a rotear pacotes provenientes da sua rede, mas, de qualquer forma, não é elegante depender dos outros para corrigir seus erros de configuração.

Para resolver este problema, basta utilizar uma das faixas de endereços reservados que vimos no capítulo 2. Estas faixas são reservadas justamente ao uso em redes internas, por isso não são roteadas na internet. As faixas de endereços reservados mais comuns são **10.x.x.x** e **192.168.x.x**, onde respectivamente o 10 e o 192.168 indicam o endereço da rede e o endereço do host pode ser configurado da forma que você desejar.

O ICS (o recurso de compartilhamento de conexão, presente no Windows 98 SE em diante) usa a faixa de endereços 192.168.0.x. Ao compartilhar a conexão com a web utilizando este recurso, você simplesmente não terá escolha. O servidor de conexão passa a usar o endereço 192.168.0.1, e todos os demais micros que forem ter acesso à web devem usar endereços de 192.168.0.2 a 192.168.0.254, já que o ICS permite compartilhar a conexão entre apenas 254 PCs.

Ao compartilhar a conexão usando um servidor Linux (como veremos no capítulo 5), você pode escolher qualquer faixa de endereços e também configurar uma "zona" para os endereços do servidor DHCP, permitindo que você tenha micros com IPs fixos e IPs dinâmicos, fornecidos pelo servidor DHCP, na mesma rede.

Veja que usar uma destas faixas de endereços reservados não impede que os PCs da sua rede possam acessar a internet, todos podem acessar através de uma conexão compartilhada via NAT ou de um servidor proxy.

O uso dos endereços de rede local tem aliviado muito o problema da falta de endereços IP válidos, pois uma quantidade enorme de empresas e usuários domésticos, que originalmente precisariam de uma faixa de endereços de classe C para colocar todos os seus micros na internet, pode sobreviver com um único IP válido, compartilhado via NAT entre todos. Em muitos casos, mesmo provedores de acesso chegam a vender conexões com endereços de rede interna nos planos mais baratos, como, por exemplo, alguns planos de acesso via rádio, onde um roteador com um IP válido distribui endereço de rede interna (conexão compartilhada) para os assinantes.

Embora seja possível, pelo menos em teoria, ter redes com até 24 milhões de PCs, usando a faixa de endereços 10.x.x.x, na prática é raro encontrar segmentos de rede com mais de 100 ou 200 micros. Conforme a rede cresce, o desempenho acaba caindo, pois, mesmo ao utilizar um switch, sempre são transmitidos alguns pacotes de broadcast (que são retransmitidos a todos os micros da rede), sem falar nas colisões.

A solução nesse caso é dividir a rede em diversos segmentos, interligados entre si por um roteador. Imagine o caso de uma escola com 5 laboratórios, cada um com 40 micros. Não seria muito prático, nem eficiente, tentar interligar todos os micros diretamente. Ao invés disso, você poderia dividir a rede em pequenos segmentos, onde os 40 micros de cada laboratório são ligados a um pequeno servidor e estes são ligados a um roteador central, que compartilha a conexão com a web e pode acumular funções de firewall, proxy, servidor de arquivos, etc.

Quando falo em "roteador", tenha em mente que você pode perfeitamente usar um servidor Linux com diversas placas de rede, configurado com as dicas do restante do livro.

» Próximo: [Entendendo as máscaras de sub-rede](#)

Além do endereço IP propriamente dito, é necessário fornecer também a máscara de sub-rede, ou "subnet mask" na configuração da rede. Ao contrário do endereço IP, que é formado por valores entre 0 e 255, a máscara de sub-rede é normalmente formada por apenas dois valores: 0 e 255, como em 255.255.0.0 ou 255.0.0.0, onde o valor 255 indica a parte endereço IP referente à rede, e o valor 0 indica a parte endereço IP referente ao host.

A máscara de rede padrão acompanha a classe do endereço IP: em um endereço de classe A, a máscara será 255.0.0.0, indicando que o primeiro octeto se refere à rede e os três últimos ao host. Em um endereço classe B, a máscara padrão será 255.255.0.0, onde os dois primeiros octetos referem-se à rede e os dois últimos ao host e, em um endereço classe C, a máscara padrão será 255.255.255.0, onde apenas o último octeto refere-se ao host.

Ex. de endereço IP	Classe do endereço	Parte referente à rede	Parte referente ao host	Máscara de sub-rede padrão
98.158.201.128	Classe A	98.	158.201.128	255.0.0.0 (rede.host.host.host)
158.208.189.45	Classe B	158.208.	189.45	255.255.0.0 (rede.rede.host.host)
208.183.34.89	Classe C	208.183.34.	89	255.255.255.0 (rede.rede.rede.host)

Mas, é possível usar máscaras diferentes para utilizar os endereços IP disponíveis de formas diferentes das padrão. O importante, neste caso, é que todos os micros da rede sejam configurados com a mesma máscara, caso contrário poderão não conseguir comunicar-se, pois pensarão estar conectados a redes diferentes.

Um exemplo comum é o uso da faixa de endereços 192.168.0.x para redes locais. Originalmente, esta é uma faixa de endereços classe C e por isso a máscara padrão é 255.255.255.0. Mesmo assim, muita gente prefere usar a máscara 255.255.0.0, o que permite mudar os dois últimos octetos (192.168.x.x). Neste caso, você poderia ter dois micros, um com o IP "192.168.2.45" e o outro com o IP "192.168.34.65" e ambos se enxergariam perfeitamente, pois entenderiam que fazem parte da mesma rede. Não existe problema em fazer isso, desde que você use a mesma máscara em todos os micros da rede.

Até agora vimos apenas máscaras de sub-rede simples. Porém, o recurso mais refinado das máscaras de sub-rede é quebrar um octeto do endereço IP em duas partes, fazendo com que tenhamos dentro de um mesmo octeto uma parte que representa a rede e outra que representa o host. Chegamos às máscaras de tamanho variável (VLSM).

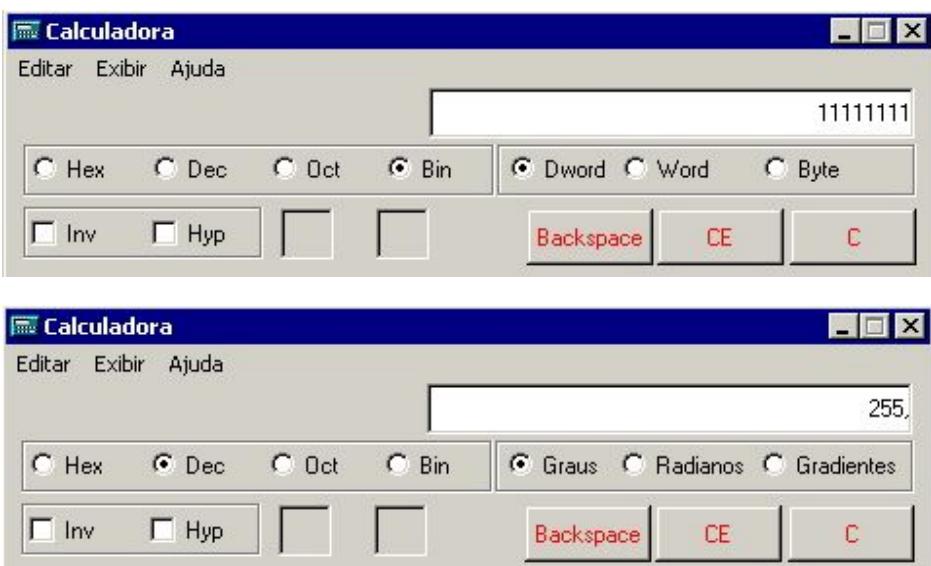
Este conceito é um pouco complicado, mas, em compensação, pouca gente sabe usar este recurso, por isso vale à pena fazer um certo esforço para aprender.

Configurando uma máscara complexa, precisaremos configurar o endereço IP usando números binários e não decimais. Para converter um número decimal em um número binário, você pode usar a calculadora do Windows ou o Kcalc no Linux. Configure a calculadora para o modo científico (exibir/científica) e verá que do lado esquerdo aparecerá um menu de seleção permitindo (entre outros) escolher entre decimal (dec) e binário (bin).



Configure a calculadora para binário e digite o número 11111111, mude a opção da calculadora para decimal (dec) e a calculadora mostrará o número 255, que é o seu

correspondente em decimal. Tente de novo agora com o binário 00000000 e terá o número decimal 0.



Veja que 0 e 255 são exatamente os números que usamos nas máscaras de sub-rede simples. O número decimal 255 (equivalente a 11111111) indica que todos os 8 números binários do octeto se referem ao host, enquanto o decimal 0 (correspondente a 00000000) indica que todos os 8 binários do octeto se referem ao host. Numa rede com máscara 255.255.255.0 temos:

Decimal:	255	255	255	0
Binário:	11111111	11111111	11111111	00000000
	rede	rede	rede	host

As máscaras de tamanho variável permitem dividir uma única faixa de endereços (seja de classe A, B ou C) em duas ou mais redes distintas, cada uma recebendo parte dos endereços disponíveis. Imagine o caso de um pequeno provedor de acesso, que possui um backbone com uma faixa de endereços de classe C e precisa dividi-lo entre dois clientes, onde cada um deles deve ter uma faixa completa de endereços.

O backbone do provedor utiliza a faixa de endereços 203.107.171.x onde o 203.107.171 é o endereço da rede e o "x" é a faixa de endereços de que eles dispõem para endereçar os micros das duas empresas. Como endereçar ambas as redes, se não é possível alterar o "203.107.171" que é a parte do seu endereço que se refere à rede?

Este problema poderia ser resolvido usando uma máscara de sub-rede complexa. Veja que podemos alterar apenas dos últimos 8 bits do endereço IP:

Decimal:	203	107	171	x
Binário:	11001011	11010110	10101011	????????

Usando uma máscara 255.255.255.0, são reservados todos os 8 bits para o endereçamento dos hosts, e não sobra nada para diferenciar as duas redes. Usando uma máscara complexa, é possível "quebrar" os 8 bits do octeto em duas partes, usando a primeira para diferenciar as duas redes e a segunda para endereçar os hosts:

Decimal:	203	107	171	x
Binário:	11001011	11010110	10101011	???? ?????
	rede	rede	rede	rede host

Para tanto, ao invés de usar a máscara de sub-rede 255.255.255.0 que, como vimos, reservaria todos os 8 bits para o endereçamento do host, usaremos uma máscara 255.255.255.240 (corresponde ao binário 11111111.11111111.11111111.11110000). Veja que numa máscara de sub-rede os números binários "1" referem-se à rede e os números "0" referem-se ao host. Na máscara 255.255.255.240 temos exatamente esta divisão: os 4 primeiros binários do último octeto são positivos e os quatro últimos são negativos:

Decimal:	255	255	255	240
Binário:	11111111	11111111	11111111	1111 0000
	rede	rede	rede	rede host

Temos agora o último octeto dividido em dois endereços binários de 4 bits cada. Cada um dos dois grupos representa agora um endereço distinto, e deve ser configurado independentemente. Como fazer isso? Veja que 4 bits permitem 16 combinações diferentes. Se você converter o número 15 em binário terá "1111" e, se converter o decimal 0, terá "0000". Se converter o decimal 11 terá "1011" e assim por diante.

Neste caso, é possível usar endereços de 1 a 14 para identificar os hosts e as redes separadas. Note que os endereços 0 e 15 não podem ser usados, pois assim como os endereços 0 e 255, eles são reservados para pacotes de broadcast:

Decimal:	203	107	171	12 _ 14
Binário:	11111111	11111111	11111111	1100 1110
	rede	rede	rede	rede host

Estabeleça um endereço de rede para cada uma das duas sub-redes disponíveis e um endereço diferente para cada micro da rede, mantendo a formatação do exemplo anterior. Por enquanto, apenas anote em um papel os endereços escolhidos, junto com seu correspondente em binários.

Na hora de configurar o endereço IP nas estações, configure primeiro a máscara de sub-rede como 255.255.255.240 e, em seguida, converta os endereços binários em decimais, para ter o endereço IP de cada estação. No exemplo da ilustração anterior, havíamos estabelecido o endereço 12 para a rede e o endereço 14 para a estação; 12 corresponde a "1100" e 14 corresponde a "1110". Juntando os dois temos "11001110", que corresponde ao

decimal "206". O endereço IP da estação será então 203.107.171.206, com máscara 255.255.255.240.

Se tivesse escolhido o endereço 10 para a rede e o endereço 8 para a estação, teríamos "10101000" que corresponde ao decimal 168. Neste caso, o endereço IP da estação seria 203.107.171.168.

Neste primeiro exemplo dividimos a faixa de endereços em 14 redes distintas, cada uma com 14 endereços. Isso permitiria que o provedor de acesso do exemplo fornecesse links para até 14 empresas diferentes, desde que cada uma não precisasse de mais de 14 endereços. É possível criar diferentes combinações, reservando números diferentes de bits para a rede e o host:

Máscara	Bits da rede	Bits do host	Número de redes		Número de hosts
255.255.255.240	1111	0000	14 (de 1 a 14)	endereços	14 (de 1 a 14) endereços
255.255.255.192	11	000000	2 (2 e 3)	endereços	62 (de 1 a 62) endereços
255.255.255.224	111	00000	6 (de 1 a 6)	endereços	30 (de 1 a 30) endereços
255.255.255.248	11111	000	30 (de 1 a 30)	endereços	6 (de 1 a 6) endereços
255.255.255.252	111111	00	62 (de 1 a 62)	endereços	2 (de 2 e 3) endereços

Em qualquer um dos casos, para obter o endereço IP basta converter os dois endereços (rede e estação) para binário, "juntar" os bits e converter o octeto para decimal.

Usando uma máscara de sub-rede 192, por exemplo, e estabelecendo o endereço 2 (ou "10" em binário) para a rede e 47 (ou "101111" em binário) para o host, juntaríamos ambos os binários obtendo o octeto "10101111" que corresponde ao decimal "175".

Se usássemos a máscara de sub-rede 248, estabelecendo o endereço 17 (binário "10001") para a rede e o endereço 5 (binário "101") para o host, obteríamos o octeto "10001101" que corresponde ao decimal "141".

Claro que as instruções acima valem apenas para quando você quiser conectar vários micros à web, usando uma faixa de endereços válidos, como no caso de uma empresa que precisa colocar no ar vários servidores, ou de uma empresa de hospedagem que aluga servidores dedicados. Caso você queira apenas compartilhar a conexão entre vários PCs, você precisará de apenas um endereço IP válido.

» Próximo: [Portas TCP e UDP](#)

Ao conectar na internet, seu micro recebe um endereço IP válido. Mas, normalmente mantemos vários programas ou serviços abertos simultaneamente. Em um desktop é normal ter um programa de e-mail, um cliente de FTP ou SSH, o navegador, um cliente de ICQ ou MSN, dois ou três downloads via bittorrent e vários outros programas que enviam e recebem informações, enquanto um único servidor pode manter ativos servidores web, FTP, SSH, DNS, LDAP e muitos outros serviços.

Se temos apenas um endereço IP, como todos estes serviços podem funcionar ao mesmo tempo sem entrar em conflito?

Imagine que as duas partes do endereço IP (a parte referente à rede e a parte referente ao host) correspondem ao CEP da rua e ao número do prédio. Um carteiro só precisa destas duas informações para entregar uma carta. Mas, dentro do prédio moram várias pessoas. O CEP e número do prédio só vão fazer a carta chegar até a portaria. Daí em diante é preciso saber o número do apartamento. É aqui que entram as famosas **portas TCP**.

Existem 65.536 portas TCP, numeradas de 0 a 65535. Cada porta pode ser usada por um programa ou serviço diferente, de forma que em teoria poderíamos ter até 65536 serviços diferentes ativos simultaneamente em um mesmo servidor, com um único endereço IP válido. O endereço IP contém o CEP da rua e o número do prédio, enquanto a porta TCP determina a que sala dentro do prédio a carta se destina.



As portas TCP mais usadas são as portas de 1 a 1024, que são reservadas para serviços mais conhecidos e utilizados, como servidores web, FTP, servidores de e-mail, compartilhamento de arquivos, etc. A porta 80, por exemplo, é reservada para uso de servidores web, enquanto a porta 21 é a porta padrão para servidores FTP.

Além do endereço IP, qualquer pacote que circula na internet precisa conter também a porta TCP a que se destina. É isso que faz com que um pacote chegue até o servidor web e não ao servidor FTP instalado na mesma máquina.

Além das 65.536 portas TCP, temos o mesmo número de portas **UDP**, seu protocolo irmão. Embora seja um protocolo menos usado que o TCP, o UDP continua presente nas redes atuais pois oferece uma forma alternativa de envio de dados, onde ao invés da confiabilidade é privilegiada velocidade e simplicidade. Vale lembrar que, tanto o TCP, quanto o UDP, trabalham na camada 4 do modelo OSI. Ambos trabalham em conjunto com o IP, que cuida do endereçamento.

No TCP, os dados são transmitidos através de conexões. Tudo começa com o cliente enviando o pacote "SYN", que solicita a abertura da conexão. Caso a porta esteja fechada, o servidor responde com um pacote "RST" e a conversa para por aí. Caso, por outro lado, exista algum servidor disponível na porta solicitada (um servidor apache, por exemplo), então ele responde com outro pacote "SYN", seguido de um pacote "ACK", avisando que a porta está disponível e prosseguindo com a abertura da conexão.

O cliente responde então com outro pacote "ACK", o que abre oficialmente a conexão. Começa então a transferência dos dados, que são organizados em pacotes com até 1550 bytes cada um. Para cada pacote recebido, a estação envia um pacote de confirmação e, caso algum pacote se perca, ela solicita a retransmissão. Cada pacote inclui 4 bytes adicionais com um código de CRC, que permite verificar a integridade do pacote. É através dele que o cliente sabe quais pacotes chegaram danificados.

Depois que todos os dados são transmitidos, o servidor envia um pacote "FYN" que avisa que não tem mais nada a transmitir. O cliente responde com outro pacote "FYN" e a conexão é oficialmente encerrada.

Graças a tudo isso, a confiabilidade é muito boa. Quando a conexão está ruim, é normal ocorrerem mais perdas de pacotes e retransmissões, mas as corrupções são geralmente causadas pelo próprio programa que está baixando o arquivo e não pelo protocolo. O problema é que toda esta formalidade torna as transferências um pouco mais lentas. Imagine que, para transmitir uma mensagem de texto com 300 bytes, via TCP, seria necessário transmitir um total de 9 pacotes!

Veja um exemplo de como a transmissão funcionaria:

Estação:	SYN	(solicita a abertura da conexão)
Servidor:	SYN	(confirma o recebimento e avisa que a porta está disponível)
Servidor:	ACK	(inicia a conexão)
Estação:	ACK	(confirma)
Estação:	DATA	(é enviado o pacote com a mensagem de texto)
Servidor:	OK	(a confirmação, depois de verificar a integridade do pacote)
Estação:	FYN	(solicita o fechamento da conexão)
Servidor:	FYN	(confirma)
Estação:	FYN	(confirma que recebeu a confirmação)

No UDP, as coisas são mais simples. Nele não existe abertura de conexão, os pacotes são transmitidos diretamente. A estação solicita alguma informação e o servidor envia a resposta. Assim como no TCP, são usados pacotes de até 1550 bytes, contendo os bits adicionais de verificação. A estação pode verificar a integridade dos pacotes, mas não tem como perceber se algum pacote se perdeu, ou solicitar a retransmissão de um pacote corrompido. Se um pacote se perde, fica por isso mesmo.

Um exemplo típico do uso do UDP é o streaming de vídeo e audio via web, uma situação onde o que vale é a velocidade e não a confiabilidade. Você não gostaria nada se o navegador parasse a exibição do vídeo para solicitar uma retransmissão cada vez que um pacote se perdesse ou chegasse corrompido. É preferível que ele pule o quadro e continue exibindo o restante do vídeo.

Outra aplicação comum são os servidores DNS. Sempre que você acessa um site, a solicitação do endereço IP referente ao domínio do site e a resposta do servidor são enviadas via UDP, para ganhar tempo.

Na prática, é bem raro encontrar algum programa que utilize unicamente pacotes UDP para qualquer coisa além do envio de mensagens curtas. Mesmo no caso do streaming de vídeo, é quase sempre usada uma porta TCP para estabelecer a conexão e enviar informações de controle, deixando o UDP apenas para o envio dos dados.

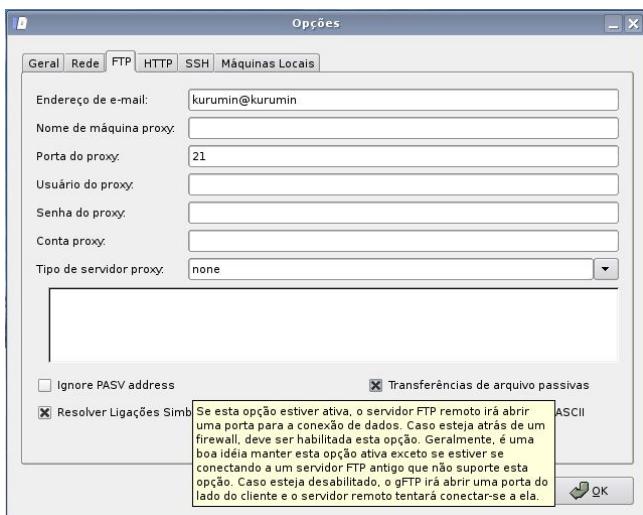
As portas TCP mais usadas são:

21: **FTP** – O FTP é um dos protocolos de transferência de arquivos mais antigos e ainda assim um dos mais usados. O ponto fraco do FTP é a questão da segurança: todas as informações, incluindo as senhas trafegam em texto puro e podem ser capturadas por qualquer um que tenha acesso à transmissão.

O FTP possui dois modos de operação: passivo e ativo. No modo ativo, o cliente contata o servidor usando uma porta vaga aleatória, como, por exemplo, a porta 1026, endereçando o pacote à porta 21 do servidor. O servidor imediatamente contata o cliente de volta, usando a porta seguinte (do cliente) para enviar os dados. Se o cliente usou a porta 1026 para abrir a conexão, então o servidor enviará os dados na porta 1027. O problema é que o modo ativo não funciona quando o cliente acessa através de uma conexão compartilhada. Ao tentar responder, o servidor cairia na porta 1027 do gateway da rede, sem conseguir chegar ao cliente.

No modo passivo, o cliente também abre a conexão contatando a porta 21 do servidor; entretanto, ao invés de iniciar a conexão imediatamente, o servidor responde avisando que o cliente pode contatá-lo numa segunda porta, escolhida aleatoriamente (a 2026, por exemplo). O cliente inicia, então, uma nova conexão na porta especificada e o servidor responde enviando os dados. Esta porta fica reservada ao cliente durante o tempo que durar a transferência. Em teoria, isto seria um limite ao número de clientes que poderiam se conectar simultaneamente, mas, na prática, seriam necessárias mais de 64.000 conexões simultâneas ao mesmo servidor FTP para esgotar as portas disponíveis.

Praticamente todos os clientes de FTP atuais utilizam o modo passivo por padrão, mas isso pode ser modificado dentro da configuração. Alguns poucos servidores de FTP não podem ser acessados em modo passivo, pois para isso é necessário que o administrador faça uma configuração de firewall mais cuidadosa, mantendo abertas um conjunto de portas altas.



Em resumo, no modo ativo o servidor precisa ter aberta apenas a porta 21, mas em compensação o cliente precisa acessar a web diretamente e ter um conjunto de portas altas abertas no firewall. No modo passivo, os papéis se invertem: o cliente não precisa ter portas abertas, mas o servidor sim.

22: SSH – O SSH é o canivete suíço da administração remota em servidores Linux. Inicialmente o SSH permitia executar apenas comandos de texto remotamente; depois passou a permitir executar também aplicativos gráficos e, em seguida, ganhou também um módulo para transferência de arquivos, o SFTP. A vantagem do SSH sobre o Telnet e o FTP é que tudo é feito através de um canal encriptado, com uma excelente segurança.

O SSH pode ser usado também para encapsular outros protocolos, criando um túnel seguro para a passagem dos dados. Criando túneis, é possível acessar servidores de FTP, proxy, e-mail, rsync, etc. de forma segura. Graças a isso, o SSH é usado como meio de transporte por diversos programas, como o FreeNX. Veremos tudo isso em detalhes no capítulo sobre acesso remoto.

O sistema de encriptação utilizado pelo SSH, assim como os túneis encriptados trabalham no nível 6 do modelo OSI, acima da camada de sessão, do protocolo TCP/IP e de toda a parte física da rede. Ao contrário do FTP, o SSH não precisa de portas adicionais: tudo é feito através da porta 22, que é a única que precisa ficar aberta no firewall do servidor. O cliente não precisa ter porta alguma aberta e pode acessar através de uma conexão compartilhada.

23: Telnet – O Telnet é provavelmente o protocolo de acesso remoto mais antigo. A primeira demonstração foi feita em 1969, com o acesso de um servidor Unix remoto, ainda através da antiga Arpanet, muito antes de ser inventado o padrão Ethernet e, antes mesmo da primeira versão do TCP/IP. O Telnet foi muito usado durante a década de 80 e 90, mas depois caiu em desuso, sendo rapidamente substituído pelo SSH. Além de não possuir nenhum dos recursos mais sofisticados suportados pelo SSH, o Telnet é um protocolo completamente aberto (no sentido pejorativo), que transmite login, senha e todos os comandos em texto puro. Isso torna ridiculamente simples capturar a transmissão (usando,

por exemplo, o Ethereal, que veremos no capítulo 4) e assim "invadir" o servidor, usando a senha roubada.

Uma curiosidade, é que o sistema usado pelo Telnet para a transmissão de comandos é usado como base para diversos outros protocolos, como o SMTP e o HTTP. De fato, você pode usar um cliente Telnet para mandar um e-mail (como veremos no capítulo 10), ou mesmo acessar um servidor web, desde que consiga simular uma conexão HTTP válida, como faria um navegador.

25: SMTP – O SMTP é o protocolo padrão para o envio de e-mails. Ele é usado tanto para o envio da mensagem original, do seu micro até o servidor SMTP do provedor, quanto para transferir a mensagem para outros servidores, até que ela chegue ao servidor destino. Tradicionalmente, o Sendmail é o servidor de e-mails mais usado, mas, devido aos problemas de segurança, ele vem perdendo espaço para o Qmail e o Postfix, que abordo no capítulo 10.

53 (UDP): DNS – Os servidores DNS são contatados pelos clientes através da porta 53, UDP. Eles são responsáveis por converter nomes de domínios como "guiadohardware.net" nos endereços IP reais dos servidores. Existem no mundo 13 servidores DNS principais, chamados "root servers". Cada um deles armazena uma cópia completa de toda a base de endereços. Estes servidores estão instalados em países diferentes e ligados a links independentes. A maior parte deles roda o Bind, mas pelo menos um deles roda um servidor diferente, de forma que, mesmo no caso de um gigantesco cyberataque, pelo menos um dos servidores continue no ar, mantendo a internet operacional.

Para acessar qualquer endereço, é preciso primeiro consultar um servidor DNS e obter o endereço IP real do servidor. Em geral, uma consulta a um dos root servers demora alguns segundos, por isso os provedores de acesso e responsáveis por grandes redes sempre configuram servidores DNS locais, que criam um cache das consultas anteriores, de forma a agilizar o acesso. Você mesmo pode configurar um servidor DNS para a sua rede usando o Bind, que aprenderemos a configurar mais adiante.

67: Bootps, 68: Bootpc – Estes dois protocolos são usados em sistemas de boot remoto (como no LTSP, que aprenderemos a configurar mais adiante), onde os clientes não possuem HD nem CD-ROM e acessam todos os arquivos de que precisam a partir do servidor.

69 (UDP): TFTP – O TFTP é uma versão simplificada do FTP, que utiliza portas UDP para a transferência dos dados e não inclui suporte à correção de erros. Ele pode ser usado para transferência de arquivos em geral, mas é mais freqüentemente usado em sistemas de boot remoto.

80: HTTP – O HTTP é o principal protocolo da internet, por onde acessamos as páginas. Embora a porta 80 seja a porta padrão dos servidores web, é possível configurar um servidor web para usar qualquer outra porta TCP. Neste caso, você precisa especificar a porta ao acessar o site, como em: <http://200.234.34.12:8080>.

110: POP3 – Servidores de e-mail, como o Postfix, armazenam os e-mails recebidos numa pasta local. Se você tiver acesso ao servidor via SSH, pode ler estes e-mails localmente, usando Mutt. Entretanto, para transferir os e-mails para a sua máquina, é necessário um servidor adicional. É aí que entra o protocolo POP3, representado pelo courier-pop e outros servidores.

Programas como o Thunderbird e o Outlook contatam o servidor POP3 através da porta 110 e baixam as mensagens utilizando um conjunto de comandos de texto, derivados do Telnet. Originalmente, o POP3 é um protocolo tão inseguro quanto o Telnet, mas os servidores atuais suportam encriptação via SSL (o mesmo sistema de encriptação usado para acessar páginas seguras, via HTTPPs), o que garante um bom nível de segurança.

137, 138 e 139: Netbios – Estas três portas são usadas pelo protocolo de compartilhamento de arquivos em redes Microsoft. Cada uma das portas tem uma função específica (nome, datagrama e sessão), mas é necessário que as três estejam abertas no firewall para que a visualização dos compartilhamentos e acesso aos arquivos funcione corretamente.

143: IMAP – O IMAP é mais um protocolo para recebimento de e-mails, assim como o POP3. A diferença entre os dois é que, ao receber os e-mails via POP3, eles são apagados do servidor assim que baixados, liberando o espaço usado na caixa postal. No IMAP, os e-mails continuam no servidor até serem deletados manualmente.

Embora oferecer contas de e-mail com acesso via IMAP seja muito mais oneroso do que via POP3 (já que o número de requisições é maior, e os usuários podem conservar mensagens antigas por muito tempo), ele vem "roubando a cena" com a popularização dos webmails, que são justamente clientes IMAP, que rodam no próprio servidor (através do Apache ou outro servidor web), e são acessados no cliente usando o navegador. No capítulo 10 veremos um exemplo de instalação, com o Squirrelmail.

177: XDMCP – O XDMCP é um protocolo de acesso remoto, suportado nativamente pelo X. Ele permite rodar aplicativos remotamente e é a base para o LTSP e outros sistemas onde é usado um servidor central e terminais leves. Pode ser também usado no dia-a-dia, para simplesmente rodar programas instalados em outra máquina da rede.

A vantagem do XDMCP é que ele é um protocolo bastante simples e rápido, que oferece um bom desempenho via rede local e consome poucos recursos, tanto no servidor, quanto no cliente. Ele é também um recurso nativo do X, de forma que você não precisa instalar nenhum software adicional, basta ativar o recurso na configuração do KDM ou GDM (os gerenciadores de login usados nas distribuições atuais).

A desvantagem é que o XDMCP é um protocolo "da velha guarda", que não utiliza encriptação, e utiliza um conjunto de portas altas para enviar dados aos clientes. Além da porta 177, onde o servidor recebe conexões, é necessário que estejam abertas as portas de 6010 à 6099 (no servidor) e as portas de 5000 a 5200 nos clientes, o que complica um pouco as coisas ao manter um firewall ativo.

389: LDAP – O LDAP é muito usado atualmente para criar servidores de autenticação e definir permissões de acesso para os diferentes usuários da rede. Existem vários padrões de

LDAP, um dos mais usados é o OpenLDAP, suportado pela maioria das distribuições Linux atualmente em uso.

443: **HTTPS** – O HTTPS permite transmitir dados de forma segura, encriptados em SSL. Ele é usado por bancos e todo tipo de site de comércio eletrônico ou que armazene informações confidenciais. No capítulo 7 aprenderemos a configurar um servidor Apache com suporte a SSL.

Naturalmente, esta é uma lista rápida, contendo apenas as portas mais usadas. Você pode ver uma lista longa e completa, com todos os serviços conhecidos e as portas utilizadas por cada um no: <http://www.iana.org/assignments/port-numbers>.

» Próximo: [ICMP](#)

Além do TCP e do UDP, temos o **ICMP** (Internet Control Message Protocol), um protocolo de controle, que opera no nível 3 do modelo OSI (junto com o protocolo IP). Ao contrário do TCP e UDP, o ICMP não é usado para a transmissão de dados, mas nem por isso deixa de desempenhar diversas funções importantes. A mais trivial delas é o **ping**, que usamos para verificar se uma determinada máquina está online, como em:

```
$ ping -c 3 guiahardware.net
```

```
PING      guiahardware.net      (64.246.6.25)      56(84)      bytes      of      data.
64  bytes  from  gdhs.guiahardware.net  (64.246.6.25): icmp_seq=1  ttl=53  time=8.72  ms
64  bytes  from  gdhs.guiahardware.net  (64.246.6.25): icmp_seq=2  ttl=53  time=8.62  ms
64  bytes  from  gdhs.guiahardware.net  (64.246.6.25): icmp_seq=3  ttl=53  time=8.37  ms
---          guiahardware.net          ping          statistics  ---
3  packets  transmitted,  3  received,  0%  packet  loss,    time   2000ms
rtt min/avg/max/mdev = 8.373/8.576/8.728/0.183 ms
```

O "-c" indica o número de repetições, neste caso 3. Sem ele, o ping fica enviando pacotes indefinidamente (no Linux), até que você aborre o programa pressionando Ctrl+C. Assim como outros comandos básicos, o ping também está disponível no Windows, através do prompt do MS-DOS.

Normalmente, os pings para qualquer servidor na Internet (com exceção dos servidores do seu provedor de acesso, ou outros servidores muito próximos), voltam com pelo menos 100 milissegundos de atraso. Quanto mais distante geograficamente estiver o servidor, ou quanto mais saturado estiverem os roteadores e links até ele, maior será o tempo de resposta. Um ping muito alto faz com que o carregamento de páginas seja mais demorado (pois o ping determina o tempo necessário para cada requisição do navegador chegar até o servidor) e atrapalha principalmente quem joga online, ou usa programas de administração remota, como o SSH.

No meu caso, consegui pings de apenas 8 ms até o servidor do Guia do Hardware, pois "trapaceei", acessando via SSH um outro servidor ligado ao mesmo backbone que ele e rodando o ping a partir dele :).

A resposta a pings pode ser desativada na configuração do sistema. No Linux, você pode usar o comando abaixo:

```
# echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

É possível também desativar a resposta a pings na configuração do firewall, de forma que, o fato de um micro da internet, ou mesmo dentro da sua rede local não responder a pings não significa muita coisa. Se ele responder, significa que está online; se não responder, significa que pode estar online também, porém configurado para não responder aos seus chamados :P.

Outra função importante do ICMP é o controle do **TTL** (time to live) de cada pacote TCP ou UDP. Os pacotes tem vida curta e sua única função é carregar os dados até o destino. Eles são transmitidos de um roteador a outro e, uma vez que chegam ao destino, são desmontados e destruídos. Mas, o que acontece em casos onde não existe uma rota possível até o destino, seja por que a máquina está desligada, por erro no endereçamento, ou por um problema em algum dos links?

Existem duas possibilidades. A primeira é um roteador próximo perceber que a máquina está fora do ar e destruir o pacote. Neste caso, ele responde ao emissor com um pacote ICMP "Destination Unreachable", avisando do ocorrido. Caso isso não aconteça, o pacote fica circulando pela rede, passando de um roteador a outro, sem que consiga chegar ao destino final.

O TTL existe para evitar que estes pacotes fiquem em loop eterno, sendo retransmitidos indefinidamente e assim consumindo banda de forma desnecessária. Graças a ele, os pacotes têm "vida útil".

O pacote é criado com um TTL de 64 hops (o default nas versões atuais do Linux). Cada vez que o pacote passa por um roteador, o número é reduzido em um. Se o número chegar a zero, o roteador destrói o pacote e avisa o emissor enviando um pacote ICMP "Time Exceeded".

No Linux, o TTL padrão é configurável através do arquivo "/proc/sys/net/ipv4/ip_default_ttl". Você pode brincar com isso alterando o valor padrão por um número mais baixo, como em:

```
# echo 8 > /proc/sys/net/ipv4/ip_default_ttl
```

Com um valor tão baixo, os pacotes gerados pela sua máquina terão vida curta, e não conseguirão atingir hosts muito distantes. Você vai continuar conseguindo acessar a página do seu provedor, por exemplo, mas não conseguirá acessar servidores geograficamente distantes. Para retornar o valor padrão, use o comando:

```
# echo 64 > /proc/sys/net/ipv4/ip_default_ttl
```

Os pacotes ICMP "Time Exceeded" são usados pelo comando "**traceroute**" (no Linux) para criar um mapa dos caminho percorrido pelos pacotes até chegarem a um determinado endereço. Ele começa enviando um pacote com um TTL de apenas 1 hop, o que faz com que ele seja descartado logo pelo primeiro roteador. Ao receber a mensagem de erro, o traceroute envia um segundo pacote, desta vez com TTL de 2 hops, que é descartado no roteador seguinte. Ele continua, enviando vários pacotes, aumentando o TTL em 1 hop a cada tentativa. Isso permite mapear cada roteador por onde o pacote passa até chegar ao destino, como em:

\$ traceroute kurumin.com.br

```

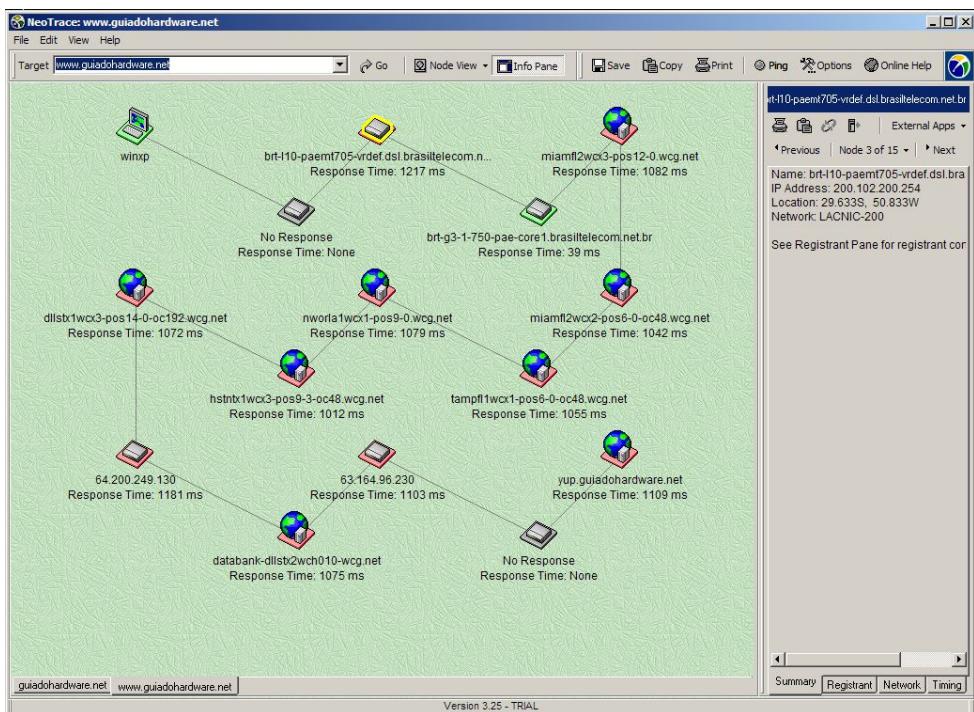
1 10.62.0.1          (10.62.0.1)      7.812      ms 6.262      ms 9.966      ms
2 poaguswh01.poa.virtua.com.br   (200.213.50.90)  9.738      ms 8.206      ms 7.761      ms
3 poagu-ebt-01.poa.virtua.com.br (200.213.50.31)  7.893      ms 7.318      ms 8.033      ms
4 embratel-F3-7-gacc07.rjo.embratel.net.br (200.248.95.253)  8.590      ms 8.315      ms 7.960      ms
5 embratel-G2-3-gacc01.pae.embratel.net.br (200.248.175.1)   7.788      ms 8.602      ms 7.934      ms
6 ebt-G1-0-dist04.pae.embratel.net.br (200.230.221.8)  31.656      ms 31.444      ms 31.783      ms
7 ebt-P12-2-core01.spo.embratel.net.br (200.244.40.162) 32.034      ms 30.805      ms 32.053      ms
8 ebt-P6-0-intl03.spo.embratel.net.br (200.230.0.13)   32.061      ms 32.436      ms 34.022      ms
9 ebt-SO-2-0-1-intl02.mia6.embratel.net.br (200.230.3.10) 298.051 ms 151.195 ms 306.732 ms 10 peer-SO-2-1-0-
intl02.mia6.embratel.net.br (200.167.0.22) 269.818 ms peer-SO-1-1-0-intl02.mia6.embratel.net.br (200.167.0.18) 144.997 ms
*
11 0.so-1-0-0.XL1.MIA4.ALTER.NET    (152.63.7.190) 240.564      ms 147.723      ms 150.322      ms
12 0.so-1-3-0.XL1.ATL5.ALTER.NET   (152.63.86.190) 438.603      ms 162.790      ms 172.188      ms
13 POS6-0.BR2.ATL5.ALTER.NET     (152.63.82.197) 164.539      ms 337.959      ms 162.612      ms
14 204.255.168.106        (204.255.168.106) 337.589      ms 337.358      ms 164.038      ms
15 dcr2-so-2-0-0.dallas.savvis.net (204.70.192.70) 212.376      ms 366.212      ms 211.948      ms
16 dcr1-so-6-0-0.dallas.savvis.net (204.70.192.49) 396.090 ms bhr2-pos-4-0.fortworthdal1.savvis.net (208.172.131.86)
189.068      ms dcr1-so-6-0-0.dallas.savvis.net (204.70.192.49) 186.161      ms
17 216.39.64.26 (216.39.64.26) 185.749 ms 191.218 ms bhr1-pos-12-0.fortworthdal1.savvis.net (208.172.131.82) 361.970 ms
18 216.39.81.34 (216.39.81.34) 186.453 ms 216.39.64.3 (216.39.64.3) 245.389 ms 216.39.81.34 (216.39.81.34) 184.444 ms
19 216.39.81.34        (216.39.81.34) 182.473      ms * 182.424      ms
20 * kurumin.com.br (72.232.35.167) 185.689 ms *

```

Neste exemplo, o pacote começa passando pelos links da Net (Virtua), passa em seguida por vários roteadores da Embratel, passando por São Paulo e Miami (já nos EUA), para então passar por roteadores da Alter.net e Savvis, até chegar ao destino final.

O Windows inclui o comando "**tracert**", que atua de forma similar, porém enviando um ping para cada host. O resultado acaba sendo similar, com exceção de casos em que o servidor é configurado para não responder a pings. Existem ainda vários exemplos de programas gráficos, como o Neotrace (para Windows), que você encontra em qualquer site de downloads e o Xtraceroute (para Linux).

Eles exibem as mesmas informações, porém de uma forma bem mais agradável. Este é um exemplo do Neotrace mostrando uma conexão especialmente ruim com um servidor hospedado no exterior, a partir de um link ADSL da Brasil Telecom. Veja que o problema começa num roteador congestionado, da própria operadora (com tempo de resposta de mais de 1200 ms!) e continua numa seqüência de links lentos da wcg.net:



Na internet, os roteadores são espertos o suficiente para conhecerem os roteadores vizinhos e escolher a melhor rota para cada destino. Sempre que um roteador fica congestionado, os demais passam a evitá-lo, escolhendo rotas alternativas. Esta comunicação é feita através de pacotes ICMP "Redirect", que avisam o emissor que uma rota mais rápida está disponível e os pacotes seguintes devem ser encaminhados através dela.

Durante as transferências de dados, os pacotes ICMP são usados também para regular a velocidade da transmissão, fazendo com que o servidor envie pacotes na maior velocidade possível permitida pelo link, sem entretanto sobrecarregar o link do cliente. Sempre que um dos roteadores pelo caminho, percebe que o link está saturado, envia um pacote ICMP "Source Quench", que faz o servidor reduzir a velocidade da transmissão. Sem isso, os pacotes excedentes seriam descartados, causando um grande desperdício de banda.

» Próximo: [ARP](#)

Dentro da rede local, os pacotes são transformados em frames, onde são endereçados ao endereço MAC da placa de rede destino e não ao endereço IP. Acontece que, inicialmente, o sistema não sabe quais são os endereços MAC das placas dos outros micros da rede local, sabe apenas os endereços IP que deve acessar.

O ARP (Address Resolution Protocol) faz companhia ao IP e ao ICMP na camada 3 do modelo OSI, oferecendo justamente uma forma simples de descobrir o endereço MAC de um determinado host, a partir do seu endereço IP. A estação manda um pacote de broadcast

(chamado "ARP Request"), contendo o endereço IP do host destino e ele responde com seu endereço MAC. Como os pacotes de broadcast são custosos em termos de banda da rede, cada estação mantém um cache com os endereços conhecidos.

Naturalmente, isso é feito de forma transparente. É mais um detalhe técnico com o qual você não precisa se preocupar se quer apenas usar a rede, mas que é interessante estudar quando está interessado em entender seu funcionamento. Você pode verificar o cache de endereços ARP do seu micro (no Linux) usando o comando "arp":

```
$ arp
```

Address	HWtype	HWaddress	Flags	Mask	Iface
192.168.1.254	ether	00:30:CD:03:CD:D2	C	eth0	
192.168.1.23	ether	00:11:D8:56:62:76	C	eth0	
192.168.1.56	ether	00:11:D8:57:45:C3	C	eth0	

Existe também o "RARP" (reverse ARP), que tem a função oposta: contatar um host da rede quando o endereço MAC é conhecido, mas o endereço IP não. Embora menos usado, o RARP também é importante, pois ele é usado quando uma estação precisa obter sua configuração de rede via DHCP.

Ao receber o pacote de broadcast enviado pela estação, o servidor DHCP sabe apenas o endereço MAC da estação e não seu endereço IP (que afinal ainda não foi definido). Ele é capaz de responder à solicitação graças ao RARP. Sem ele, não teríamos DHCP :).

Muitas distribuições Linux incluem o "**arping**", um pequeno utilitário que utiliza o ARP ao invés do ping para descobrir se outras máquinas da rede local estão online. A vantagem é que mesmo máquinas protegidas por firewall, ou configuradas para não responder pings respondem a pacotes ARP, fazendo com que ele seja mais uma ferramenta interessante na hora de diagnosticar problemas na rede.

Nas distribuições derivadas do Debian, você pode instalá-lo via apt-get (apt-get install arping). Para usar, basta informar o endereço IP ou endereço MAC da máquina alvo, como em:

```
$ arping 192.168.1.110
```

```
ARPING                                                 192.168.1.110
60 bytes from 00:11:d8:21:52:76 (192.168.1.110): index=0   time=112.057   usec
60 bytes from 00:11:d8:21:52:76 (192.168.1.110): index=1   time=101.089   usec
60 bytes from 00:11:d8:21:52:76 (192.168.1.110): index=2 time=99.897 usec
```

Uma observação importante é que o ARP é usado apenas dentro da rede local, o único local onde são usados endereços MAC. Quando o pacote passa pelo gateway e é encaminhado para a internet, os campos com os endereços MAC são removidos, o que faz com que o arping e outros utilitários baseados em pacotes ARP deixem de funcionar.

Se você tiver a curiosidade de disparar o arping contra um host da internet, vai perceber que, embora o comando seja executado sem erros, ele fica parado indefinidamente aguardando por uma resposta que nunca vem:

\$ arping google.com

ARPING 64.233.167.99

(espera infinita...)

Isso acontece pois o pacote de broadcast enviado pelo arping não é encaminhado pelo gateway da rede, ele só seria respondido se, por acaso, existisse um micro dentro da rede local utilizando o endereço "64.233.167.99". Mesmo que o pacote fosse incorretamente encaminhado para a internet, ele não iria muito longe, pois seria descartado no primeiro roteador por onde passasse.

» Próximo: [**Outros protocolos de rede**](#)

O TCP/IP tornou-se um protocolo onipresente. Ele é usado desde servidores de grande porte até palmtops e celulares, permitindo que dispositivos de plataformas completamente diferentes possam conversar entre si.

Mas, antes do TCP/IP, os protocolos mais usados eram o NetBEUI e o IPX/SPX. Eles ainda são utilizados em algumas redes, por isso é importante saber um pouco sobre eles:

NetBEUI: O NetBEUI é uma espécie de "vovô protocolo", pois foi lançado pela IBM no início da década de 80 para ser usado junto com o IBM PC Network, um micro com configuração semelhante à do PC XT, mas que podia ser ligado em rede. Naquela época, o protocolo possuía bem menos recursos e era chamado de NetBIOS. O nome NetBEUI passou a ser usado quando a IBM estendeu os recursos do NetBIOS, formando a versão final do protocolo.

No jargão técnico atual, usamos o termo "NetBEUI" quando nos referimos ao protocolo de rede em si e o termo "NetBIOS" quando queremos nos referir aos comandos deste mesmo protocolo usado pelos programas para acessar a rede.

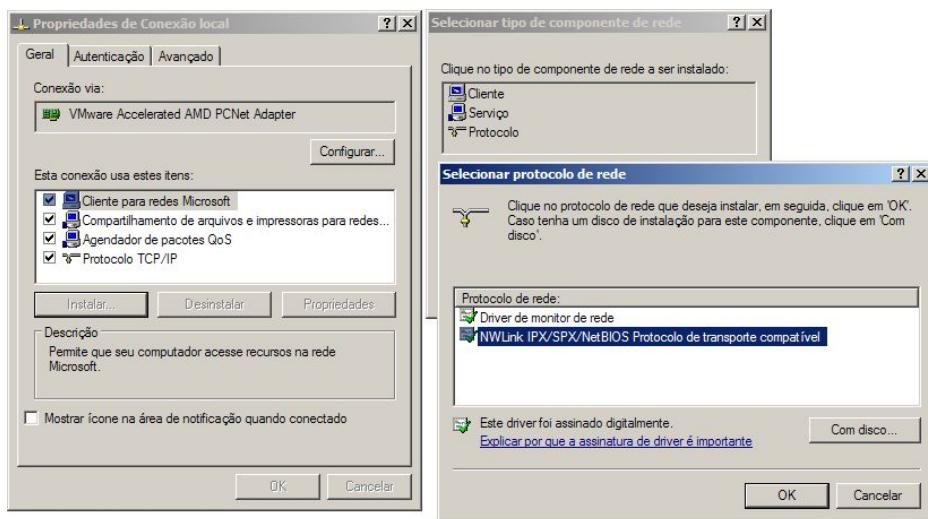
Ao contrário do IPX/SPX e do TPC/IP, o NetBEUI foi concebido para ser usado apenas em pequenas redes e por isso sempre foi um protocolo extremamente simples. Por um lado, isto fez que ele se tornasse bastante rápido e fosse considerado o mais rápido protocolo de rede durante muito tempo. Para você ter uma idéia, apenas as versões mais recentes do IPX/SPX e TCP/IP conseguiram superar o NetBEUI em velocidade.

Mas, esta simplicidade toda tem um custo: devido ao método simples de endereçamento usado pelo NetBEUI, podemos usá-lo em redes de no máximo 255 micros. Além disso, o NetBEUI não suporta enumeração de redes (para ele todos os micros estão ligados na mesma rede). Isto significa que, se você tiver uma grande Intranet, composta por várias redes interligadas por roteadores, os micros que usarem o NetBEUI simplesmente não serão

capazes de enxergar micros conectados às outras redes, enxergarão apenas os micros a que estiverem conectados diretamente. Devido a esta limitação, dizemos que o NetBEUI é um protocolo "não-roteável".

Apesar de suas limitações, o NetBEUI ainda é usado em algumas redes Windows, por ser rápido, fácil de instalar e usar. Você não pode usá-lo para acessar a internet, acessar outras máquinas da rede via SSH nem nenhum outro dos serviços que vimos até aqui, mas ele permite que as máquinas Windows compartilhem arquivos entre si.

De qualquer forma, para instalá-lo, no Windows XP, acesse o menu de configuração da rede e acesse a opção Adicionar > Protocolo > NWLink/IPX/SPX/NetBIOS Protocolo de transporte compatível.



Embora não seja recomendável utilizá-lo nos dias de hoje, não existe problema em mantê-lo ativo junto com o TCP/IP. No NetBEUI também não existe configuração de endereços, pois os micros conversam diretamente usando os endereços MAC.

Ao instalar uma estação de trabalho com o XP numa rede antiga, baseada em micros com o Windows 95/98, pode ser necessário ativar o NetBEUI para que ele consiga conversar com as outras máquinas, já que antigamente, antes da popularização do acesso à internet, era comum configurar redes locais usando apenas o NetBEUI, sem TCP/IP.

IPX/SPX: Este protocolo foi desenvolvido pela Novell, para ser usado em seu Novell Netware. Como o Netware acabou tornando-se muito popular, outros sistemas operacionais de rede (incluindo o Windows), passaram a suportar este protocolo. O IPX/SPX é tão rápido quanto o TPC/IP (apesar de não ser tão versátil) e suporta roteamento, o que permite seu uso em redes de médio ou grande porte.

As versões recentes do Novell Netware oferecem a opção de usar o IPX/SPX ou o TCP/IP, sendo o uso do TCP/IP mais comum, já que é mais fácil interligar máquinas de várias plataformas à rede.

No Netware, além do módulo principal (instalado no servidor), é fornecido um módulo cliente, que deve ser instalado em todas as estações de trabalho. Além da versão principal do Netware, existe a versão Personal, um sistema de rede ponto a ponto, que novamente roda sobre o sistema operacional. Esta versão do Netware é bem fácil de usar, porém nunca foi muito popular, pois o Windows sozinho já permite a criação de redes ponto a ponto muito facilmente, desde o 3.11.

Atualmente é muito comum utilizar servidores Linux, rodando o Samba, substituindo servidores Windows NT, 2000 ou 2003 Server. No início de 2003, a Novell comprou a SuSE, uma das maiores distribuições Linux na Europa e, em seguida, a Ximian, que entre outras coisas desenvolve soluções de interoperabilidade entre servidores Linux e Windows. Isso mostra que as futuras soluções da Novell devem ser baseadas em Linux.

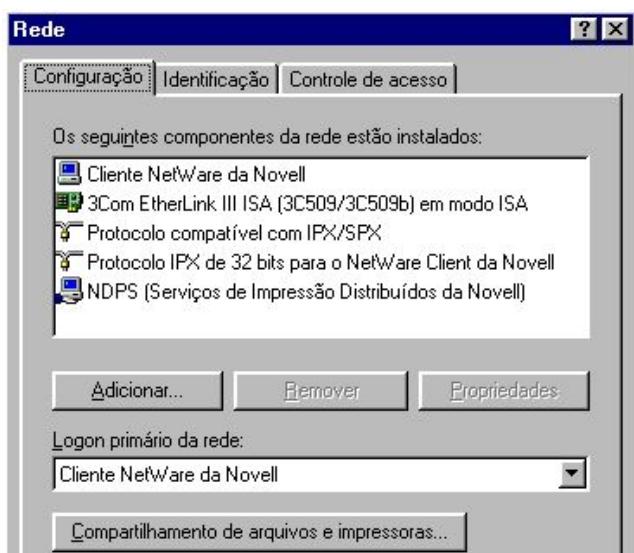
Mas, voltando ao assunto principal, é possível usar estações Windows e Linux como clientes de um servidor Novell. No caso do Windows, é necessário ter instalado o protocolo IPX/SPX e também um cliente para redes Netware. No Windows XP, a compatibilidade com o IPX é fornecida pelo protocolo " NWLink/IPX/SPX/NetBIOS", o mesmo que instalamos para ativar o suporte ao NetBEUI.

Para instalar o protocolo IPX/SPX no Windows 95/98, abra o ícone de configuração da rede e use a opção "Adicionar > Protocolo > Microsoft > Protocolo compatível com IPX/SPX". Para instalar o cliente para redes Novell no Windows 98, clique em "Adicionar > Cliente > Microsoft > Cliente para redes NetWare".

Apesar do cliente fornecido com o Windows 98 não ficar devendo muito em termos de recursos, é preferível usar o cliente da própria Novell, que traz alguns recursos únicos, além de ser mais rápido. O programa cliente da Novell acompanha o módulo servidor, mas você também pode baixá-lo gratuitamente (12 MB) do site da Novell: <http://www.novell.com.br>.

Após baixar o arquivo, execute-o para que ele se descompacte automaticamente e, em seguida, execute o arquivo "setup.exe" para instalar o cliente. O programa de instalação adicionará o "Cliente NetWare da Novell" e o "Protocolo IPX de 32 Bits para o NetWare Client da Novell", que aparecerão na janela de configuração da rede.

O cliente ficará residente na forma de um ícone ao lado do relógio, já que você depende do programa para ter acesso ao servidor. Como no caso dos servidores NT, você deverá criar uma conta de usuário no servidor Novell e logar-se na rede informando o nome de usuário e senha estabelecidos.



Ao usar clientes Linux, você pode utilizar o NovelClient (com um L só), que pode ser baixado no:
<http://novelclient.sourceforge.net/>.



» Próximo: [Capítulo 4: Segurança](#)

A questão da segurança tem se tornado cada vez mais importante à medida que a internet torna-se um ambiente cada vez mais hostil e as ferramentas para capturar tráfego, quebrar sistemas de encriptação, capturar senhas e explorar vulnerabilidades diversas tornam-se cada vez mais sofisticadas.

Outra questão importante é que usamos cada vez mais tecnologias diferentes de acesso e transmissão de dados, o que torna manter sua rede segura uma tarefa mais complicada. Por exemplo, sua rede pode ser bastante segura contra invasões "diretas", via internet, graças ao

firewall ativado no gateway da rede, mas, ao mesmo tempo, ser muito fácil de invadir através da rede wireless sem encriptação que você utiliza.

Ao usar clientes Windows, existe ainda o problema dos vírus, trojans e worms. Os vírus se espalham através de arquivos infectados, páginas que exploram vulnerabilidades no navegador, e-mails e assim por diante, geralmente utilizando alguma técnica de engenharia social que leve o usuário a clicar em um link ou executar um arquivo.

Assim como na vida real, os vírus variam muito em termos de potencial nocivo. Existem desde vírus extremamente perigosos, que destroem os dados do HD, subscrevendo os arquivos com dados aleatórios (de forma que seja impossível recuperá-los) e, algumas vezes até mesmo danificando o BIOS da placa mãe; até vírus relativamente inofensivos, que não fazem muita coisa além de se replicarem por diversos meios, tentando infectar o maior número de PCs possíveis.

Os trojans são muito similares aos vírus, mas o objetivo principal é abrir portas e oferecer alguma forma de acesso remoto à máquina infectada. Ao invés de deletar arquivos ou mostrar pop-ups (como os vírus), os trojans são quase sempre muito discretos, de forma que o usuário não perceba que sua máquina está infectada. Isso permite que o invasor roube senhas, use a conexão para enviar spam, procure por informações valiosas nos arquivos do HD, ou mesmo use as máquinas sob seu controle para lançar ataques diversos contra outras máquinas.

Os worms se diferenciam dos vírus e trojans pela forma como infectam as máquinas. Ao invés de depender do usuário para executar o arquivo infectado, os worms se replicam diretamente, explorando vulnerabilidades de segurança nas máquinas da rede. Os mais complexos são capazes de explorar diversas brechas diferentes, de acordo com a situação. Um worm poderia começar invadindo um servidor web com uma versão vulnerável do IIS, infectar outras máquinas da rede local a partir dele, acessando compartilhamentos de rede com permissão de escrita e, a partir delas, se replicar via e-mail, enviando mensagens infectadas para e-mails encontrados no catálogo de endereços; tudo isso sem intervenção humana.

Os worms podem ser bloqueados por um firewall bem configurado, que bloquee as portas de entrada (e, se possível, também portas de saída) usadas por ele. É possível também bloquear parte dos vírus e trojans adicionando restrições com base em extensão de arquivos no Squid, ou adicionando o Clamav no servidor de e-mails (como veremos ao longo do livro), mas, a principal linha de defesa acaba sempre sendo o antivírus ativo em cada máquina Windows.

No Linux, as coisas são um pouco mais tranquilas neste ponto. Os vírus são quase que inexistentes e as vulnerabilidades em servidores muito utilizados, como o Apache, SSH, etc. são muito menos comuns. O problema é que todos estes prognósticos favoráveis dão uma falsa sensação de segurança, que acabam levando muitos usuários a assumirem um comportamento de risco, deixando vários serviços ativados, usando senhas fracas ou usando a conta de root no dia-a-dia.

Também é muito comum que os novos usuários fiquem impressionados com os recursos de conectividade disponíveis no Linux e acabem abrindo brechas de segurança ao deixar servidores XDMCP, NFS, Squid, etc. abertos para a internet. Muitos usuários do Windows sequer sabem que é possível manter um servidor FTP aberto no micro de casa, enquanto muitas distribuições Linux instalaram servidores Apache ou SSH por default. Muitos usuários Linux mantêm servidores diversos habilitados em suas máquinas, algo muito menos comum no mundo Windows.

No final das contas, a segurança do sistema depende muito mais do comportamento do usuário do que do sistema operacional em si. Um usuário iniciante que use o Windows XP, sem nenhum firewall ou qualquer cuidado especial, mas que tenha o cuidado de manter o sistema atualizado e não executar qualquer porcaria que chegue por mail provavelmente estará mais seguro do que um usuário Linux sem noções de segurança que use o sistema como root e mantém um batalhão de servidores desatualizados ativos na máquina.

Você poderia perguntar porque alguém teria interesse em invadir máquinas de usuários domésticos, que não possuem arquivos valiosos, ou mesmo estações de trabalho que são usadas apenas para editar textos e enviar e-mails.

A questão principal não é o que está armazenado do HD, mas sim a banda. Ter vários PCs sob seu controle, principalmente se eles possuírem conexões de alta velocidade, significa poder. É possível usá-los para alimentar redes P2P como o Kazaa e outros, fundar uma rede de distribuição de warez ou moviez, usá-los como servidores complementares para um site pornô qualquer, enviar spam, usá-los para rodar portscans e lançar ataques contra outras máquinas ou até mesmo usá-los em um ataque coordenado para tirar um grande portal do ar.

» Próximo: [As dicas gerais](#)

Segurança envolve mais do que simplesmente instalar um firewall ou substituir o Outlook por um leitor de e-mails mais seguro, envolve um conjunto de atitudes. A idéia básica é em primeiro lugar evitar que outras pessoas tenham acesso a dados pessoais, como senhas, versões dos servidores que você mantém abertos na sua máquina e se possível até mesmo restringir o acesso ao seu endereço de e-mail ou número de ICQ e dificultar a obtenção do endereço IP da sua máquina. A idéia é que, a partir de qualquer uma destas informações, alguém pode conseguir obter mais dados até conseguir acesso.

Nenhum programa é livre de bugs. Com o seu número de ICQ alguém poderia tentar se aproveitar de alguma brecha descoberta no protocolo ou no cliente que você utiliza. Com o seu e-mail é possível se aproveitar de uma vulnerabilidade recém-descoberta no leitor de e-mails, ou simplesmente tentar convencê-lo a abrir um arquivo anexado, que contenha um trojan.

A questão das senhas é um pouco mais delicada, pois envolve não só os logins e senhas de e-mail, mas também senhas de banco, números de cartão de crédito, etc. A forma mais

comum de conseguir estes dados é através de um programa de keytrap que captura tudo que é digitado no teclado, gerando um arquivo de texto que pode ser recuperado depois ou enviado automaticamente por e-mail.

Existem várias formas de conseguir instalar um keytrap no PC de alguém: as possibilidades vão desde enviar um programa qualquer por e-mail (que, ao ser executado, instala o programa), invadir a máquina usando um trojan que dê acesso remoto e a partir daí instalar o keytrap, entre outras possibilidades.

Mesmo que o seu micro esteja limpo, ainda existe a possibilidade de que os seus dados sejam capturados ao utilizar o micro de alguém ou, principalmente, ao utilizar um Cybercafé. Evite digitar qualquer tipo de senha ou dados confidenciais em qualquer micro que não seja seu ou, se isso realmente for necessário, pelo menos tenha o cuidado de usar algum tipo de teclado virtual (como os que os sistemas de acesso online dos bancos oferecem, o teclado virtual incluído no Windows ou o Xvkbd no Linux). Outra boa solução é dar boot usando um CD do Kurumin ou outro live-cd, permitindo que você tenha um sistema "limpo".

Os ambientes mais críticos são os Cybercafés, onde muitas pessoas utilizam os mesmos PCs. Não utilize nenhum serviço onde não exista uma boa política de segurança, baseada em logins separados para cada cliente e de preferência com estações Linux que oferecem um suporte multiusuário mais desenvolvido.

Com senhas em mãos, qualquer um poderá ler seus e-mails, acessar sua máquina remotamente caso você mantenha um servidor de FTP ou SSH ativo e, assim por diante. As senhas são o ponto fraco de qualquer sistema de segurança, por isso devem ser uma preocupação constante. Utilize sempre boas senhas, misturando letras e números e com pelo menos 8 (de preferência 12) caracteres, jamais utilize palavras como senha e troque suas senhas, sobretudo a senha de root constantemente.

O ideal é que ninguém além de você tenha acesso físico ao seu PC. Mesmo que você deixe o micro desligado, ou protegido por uma proteção de tela, é possível instalar programas dando boot através de um CD-ROM ou disquete. A partir do momento em que uma pessoa mal intencionada senta na frente do seu servidor e dá boot através de um live-CD, o servidor não é mais seu, e sim dele.

Se você administra um servidor ou permite que outros usuários accessem sua máquina remotamente, exija que todos utilizem boas senhas. Muitas brechas de segurança permitem obter acesso de root partindo de um simples login de usuário. Por isso, além de exigir o uso de boas senhas, você deve dar logins de usuário apenas à pessoas de confiança.

Outra boa idéia é "esconder" seus servidores, alterando suas portas default. Por exemplo, um servidor de FTP escutando na porta 21 (a default) seria facilmente descoberto pelo atacante, que, a partir daí, poderia tentar explorar algum tipo de vulnerabilidade no programa para obter acesso. Mas, se você configurá-lo para operar na porta 44756, por exemplo, já seria muito mais complicado que alguém o descobrisse. Seria preciso fazer uma varredura de portas completa, que demora várias horas para perceber que a porta

44756 está aberta e mais algum tempo para descobrir que ela está sendo usada por um servidor de FTP. Quanto mais dificuldade melhor, não é mesmo?

Caso você esteja usando um programa de detecção de intrusões, como o Snort, a varredura de portas iria disparar o alarme, fazendo com que você tivesse conhecimento do ataque antes mesmo do atacante descobrir quais portas estão abertas para tentar fazer qualquer coisa.

Mais um erro comum é deixar servidores de FTP, web, SSH, etc. disponíveis para toda a internet enquanto você só precisa deles dentro da sua rede interna. Se você tem duas placas de rede, ou mesmo uma placa de rede e um modem, é fácil filtrar o tráfego permitindo que apenas os acessos vindos dos clientes locais sejam aceitos. Isto pode ser feito na configuração do servidor (como no caso do Samba e do Apache) quanto na configuração do firewall (que abordo em detalhes no capítulo 11).

O ideal em termos de segurança é não acessar a web diretamente nos desktops. Sempre que possível, acesse por trás de uma conexão compartilhada, através de um servidor Linux com o firewall ativo, ou através de um modem ADSL configurado como roteador. Direcione apenas as portas realmente necessárias para os clientes.

Todas estas medidas representam a chamada segurança passiva. As brechas de segurança são como balas perdidas, ninguém pode dizer onde surgirá a próxima. Mesmo um sistema com um excelente histórico de segurança pode revelar um bug monstruoso a qualquer momento. A idéia é impedir ou pelo menos dificultar a exploração de qualquer eventual brecha.

Imagine que amanhã alguém descubra uma brecha grave no SSH, por exemplo. Se você deixa o serviço ativo no seu servidor e ainda por cima aberto ao mundo, você estaria com sérios problemas. Mas, se você mantém o serviço desativado, ou disponível apenas para a sua rede interna, a brecha não afetaria diretamente o seu sistema, pois seria preciso passar primeiro pelo firewall para ter acesso a ele.

» Próximo: [Usando o Nmap](#)

O Nmap é um portscan de uso geral. Ele é um dos componentes-base usados pelo Nessus (que veremos a seguir), mas pode também ser usado diretamente, sempre que você precisar verificar rapidamente as portas abertas em determinado host, seja na sua rede local, seja na Internet.

O Nmap é um pacote muito usado e por isso está disponível em todas as principais distribuições. Você pode instalá-lo usando o yast (SuSE), yum (Fedora), urpmi (Mandriva), ou outro gerenciador de pacotes disponível. Nas distribuições derivadas do Debian, você pode instalá-lo via apt-get:

```
# apt-get install nmap
```

Para usar todos os recursos do Nmap, você deve executá-lo como root. O uso mais simples é escanear diretamente uma máquina da rede, como em:

```
# nmap 192.168.0.3
```

```
Starting nmap 3.81 ( http://www.insecure.org/nmap/ )

Interesting ports on 192.168.0.3:
(The 1661 ports scanned but not shown below are in state: closed)

PORT      STATE          SERVICE
68/tcp    open           dhcpclient
631/tcp   open           ipp
MAC Address: 00:0F:B0:55:EE:16 (Compal Electronics)
Nmap finished: 1 IP address (1 host up) scanned in 0.339 seconds
```

Neste exemplo, o teste foi disparado contra uma máquina Linux, rodando uma instalação personalizada do Debian Sarge. As duas portas abertas são o cliente DHCP (é normal que a porta 68 fique aberta em clientes configurados via DHCP) e o servidor Cups, que escuta na porta 631. O Cups mantém esta porta aberta sempre que é ativado (você precisa dele para imprimir, tanto em compartilhamentos da rede, quanto em impressoras locais). Por padrão, ele permite que apenas que o localhost imprima e accesse a interface de administração, mas é possível configurá-lo para compartilhar as impressoras com a rede de forma bem simples através do arquivo "/etc/cups/cupsd.conf", como veremos em detalhes no capítulo 6.

Nem o cliente DHCP, nem o Cups permitem acesso via shell, por isso, salvo eventuais graves brechas de segurança, os ataques mais graves que poderiam ser lançados neste caso seriam tentar modificar a configuração de rede, tentando responder ao cliente DHCP no lugar do servidor DHCP da rede, ou tentar usar impressoras compartilhadas no Cups.

O simples fato de uma determinada porta estar aberta, não significa que a máquina está vulnerável, mas apenas que existem serviços ativos e as portas não estão sendo bloqueadas por nenhum firewall.

Você pode obter mais informações sobre as portas abertas, incluindo a versão de cada serviço ativo usando a opção "-sV", como em:

```
# nmap -sV 192.168.0.3
```

Esta opção é muito mais demorada, no lugar dela você pode preferir fazer logo um scan completo usando o Nessus.

É possível também escanear de uma vez toda uma faixa de endereços, como em:

```
# nmap 192.168.0.1-254
```

Outro parâmetro interessante é a opção "-O", que faz com que o Nmap tente identificar qual é o sistema operacional usado em cada máquina. Esta identificação permite diferenciar máquinas rodando diferentes versões do Windows de máquinas rodando Linux ou MacOS,

por exemplo, mas não é muito eficiente em identificar diferentes distribuições Linux. Veja um exemplo:

nmap -O 192.168.0.4

```
Starting           nmap      3.81      (           http://www.insecure.org/nmap/      )
Interesting        ports      on          192.168.1.35:
(The 1658 ports scanned but not shown below are in state: closed)

PORT              STATE
135/tcp           open
139/tcp           open
445/tcp           open
1025/tcp          open
5000/tcp          open  UPnP

MAC Address:      02:0F:B0:55:EE:16      (Unknown)
Device type:      general             purpose
Running:          Microsoft           Windows          95/98/ME|NT/2K/XP
OS details:       Microsoft           Windows          Millennium
Windows 2000      Pro or Advanced   Server,          or
                  Edition (Me), Windows XP

Nmap finished: 1 IP address (1 host up) scanned in 1.145 seconds
```

Neste caso temos uma instalação limpa do Windows XP, sem o firewall ativo. Note que a identificação do sistema não é exata, o Nmap indicou corretamente que é uma máquina Windows, mas não soube identificar precisamente a versão.

Os scans do Nmap podem ser facilmente detectados caso alguma das máquinas alvo esteja com o Snort, ou outro detector de intrusões ativo, o que vai lhe render no mínimo um puxão de orelha do administrador da rede. Para dificultar isso o Nmap oferece a opção de fazer um half-open scan, especificando a opção "-sS", como em:

nmap -sS 192.168.0.1-254

Operando neste modo, o Nmap apenas envia um pacote SYN para cada porta alvo e espera para ver se recebe um pacote ACK de confirmação. Se a confirmação é recebida, ele sabe que a porta está aberta. Mas, de qualquer forma, a conexão para por aí, ele não estabelece uma conexão completa como faria normalmente. Embora seja mais complicado, estes scans não são impossíveis de detectar. Versões recentes do Snort, por exemplo, são capazes de detectá-los e logá-los da mesma forma que os scans tradicionais.

Apesar de menos comum, é possível fazer também uma varredura de portas UDP abertas. Embora poucos serviços possam ser diretamente conectados através de portas UDP, muitos as utilizam para transferir dados e, em geral, os firewalls são configurados para bloquear apenas as portas TCP. Escanear as portas UDP é uma forma alternativa de detectar serviços abertos em uma máquina, mesmo que todas as portas TCP estejam fechadas no firewall. Existem também casos de backdoors acessíveis via UDP, como o Back Orifice (no Windows) e até mesmo (raras) brechas de segurança em serviços do Linux ou outros sistemas Unix, como uma brecha em certas versões do rpcbind do Solaris, que podia ser explorada através de uma porta UDP alta, a partir da 32770 (variando de acordo com a versão).

Os scans de UDP são rápidos se direcionados a máquinas Windows, mas são absurdamente lentos se feitos contra máquinas Linux ou BSD, onde o sistema limita o número de erros de ICMP (dos quais o scan do Nmap depende) a uma taxa de aproximadamente 20 por segundo. No Windows não existe limite.

Para usar o scan UDP, usamos a opção "-sU", como em:

```
# nmap -sU 192.168.0.4
```

Por padrão, o Nmap escaneia apenas um conjunto de 1661 portas, que incluem as usadas pelos serviços mais comuns. Uma medida de segurança comum é esconder serviços como o SSH em portas altas, de forma que eles sejam mais difíceis de detectar. Nesses casos, você pode fazer um scan completo, incluindo todas as portas TCP (ou UDP) usando a opção "-p 1-65535", como em:

```
# nmap -sS -p 1-65535 192.168.0.4
```

A opção "-p" pode ser usada para escanear apenas uma porta específica, ou uma faixa de portas em que esteja interessado. Se executado via rede local, o scan é sempre relativamente rápido (a menos que a máquina alvo esteja com um firewall ativo, configurado em modo "DROP"), mas, via internet, as coisas tornam-se bem mais demoradas. Ao tentar localizar vulnerabilidades em uma determinada faixa de endereços IP, você começaria lançando o teste rápido contra toda a faixa, reservando as opções mais demoradas para algumas máquinas específicas.

A opção "-sS", combinada com a "-p 1-65535", permite localizar serviços escondidos em portas altas, mas não é capaz de dizer muito sobre eles. Ele sempre retorna algo como:

```
22543/tcp open unknown
```

Você pode escanear esta porta específica usando a opção "-sV" para descobrir mais sobre ela, como em:

```
# nmap -sV -p 22 192.168.0.4
```

PORt	STATE	SERVICE	VERSION
22543/tcp	open	ssh	OpenSSH 3.8.1p1 Debian-8.sarge.4 (protocol 1.99)

Nmap finished: 1 IP address (1 host up) scanned in 0.284 seconds

Agora você sabe que a máquina tem ativo um servidor OpenSSH (versão 3.8.1, do Debian Sarge), escondido na porta 22543.

Tudo é muito simples quando a máquina alvo não possui nenhum firewall ativo. O scan é rápido e você pode lançar toda sorte de ataques sobre os serviços ativos. Mas, com um firewall ativo, as coisas tornam-se um pouco mais complicadas e demoradas. Um firewall configurado para descartar (DROP) todos os pacotes recebidos, faz com que o scan torne-se extremamente lento.

Versões antigas do Nmap não chegavam sequer a concluir o teste quando o alvo estava configurado desta forma, retornando uma mensagem como:

```
Starting      nmap      3.50      (      http://www.insecure.org/nmap/      )
Host      192.168.0.33      appears      to      be      down,      skipping      it.
Note:          Host      seems      down.
Nmap run completed -- 1 IP address (0 hosts up) scanned in 12.053 seconds
```

Nesses casos, você pode forçar o Nmap a concluir o teste, a fim de detectar serviços escondidos em portas altas, usando o parâmetro "-P0", como em:

```
# nmap -sS -P0 -p 1-65535 192.168.0.4
```

O problema neste caso é que o scan demora muito mais que o normal, já que, por não receber respostas, ele precisa aguardar um tempo muito maior antes de passar para a porta seguinte. Um teste executado contra um micro na Internet, através de uma conexão lenta, pode literalmente demorar dias. Apesar de não responder, o micro remoto pode ser configurado para logar suas tentativas, permitindo que o administrador tome conhecimento e aja de acordo, bloqueando seu endereço IP ou contatando seu provedor de acesso. Um firewall bem configurado é realmente uma grande vantagem de segurança para qualquer servidor.

» Próximo: [Usando o Nessus](#)

O Nessus é uma ferramenta de auditoria muito usada para detectar e corrigir vulnerabilidades nos PCs da rede local. Ele realiza uma varredura de portas, detectando servidores ativos e simulando invasões para detectar vulnerabilidades. Uma característica importante é que o Nessus procura por servidores ativos não apenas nas portas padrão, mas em todas as portas TCP. Ele será capaz de detectar uma vulnerabilidade em um servidor Apache escondido na porta 46580, por exemplo.

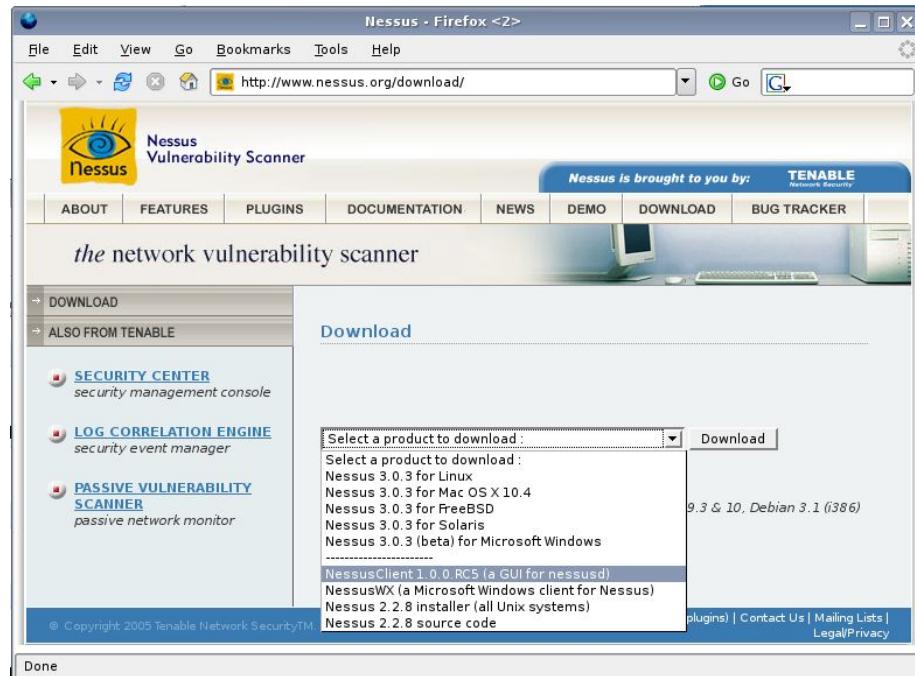
Até a versão 2.2.8, o Nessus era um aplicativo open-source. Os desenvolvedores trabalham na área de segurança, prestando consultoria e vendendo versões personalizadas do Nessus, com plugins e recursos adicionais. O problema é que outras empresas de segurança passaram a se aproveitar disso para incorporar recursos do Nessus em seus produtos proprietários e a desenvolver versões modificadas, que competiam diretamente com as soluções oferecidas por eles.

Isso criou um clima crescente de tensão até que os desenvolvedores decidiram mudar a licença a partir da versão 3.0. O Nessus continua sendo de uso gratuito, mas o código fonte passou a ser fechado, para evitar a concorrência predatória de outras empresas.

Você pode baixar a versão mais recente na seção de downloads do <http://www.nessus.org>.

Para baixar, você precisa fornecer um endereço de e-mail válido, para onde é enviado um código de ativação. Estão disponíveis pacotes para diversas distribuições, entre eles um pacote .deb para as distribuições derivadas do Debian e pacotes .rpm para o Fedora, Red

Há e SuSE. Você precisa baixar tanto o Nessus propriamente dito, quanto o "NessusClient", disponível na mesma página.



Instale o pacote baixado usando o comando "dpkg -i" (no caso do pacote .deb), ou "rpm -Uvh" (para os pacotes .rpm), como em:

```
# dpkg -i Nessus-3.0.3-debian3_i386.deb
```

O Nessus utiliza o Nmap como portscan, por isso é necessário que ele também esteja instalado. O Nmap faz a primeira rodada de testes, detectando as portas abertas e o Nessus usa as informações fornecidas por ele como ponto de partida para executar uma rodada adicional de testes, que permitem devolver um relatório bastante detalhado das vulnerabilidades encontradas.

Depois de instalar, você precisa criar um login de usuário para utilizar o Nessus. Isso é necessário pois ele é dividido em dois componentes: um servidor (que é quem faz todo o trabalho pesado) e um cliente, que funciona como uma interface segura para ele. Isso permite que você instale o servidor em uma máquina que faça parte da rede que vai ser escaneada e use seu notebook para apenas rodar o cliente, o que pode ser feito até mesmo remotamente.

Com isso, seu notebook fica livre durante o teste, permitindo que você execute testes adicionais ou pesquise sobre as vulnerabilidades na web enquanto o teste é realizado. Naturalmente, você pode rodar ambos os componentes na mesma máquina, o único pré-requisito é usar uma máquina relativamente rápida, com pelo menos 256 MB de RAM livres (ou seja, descontando a memória usada pelo sistema e outros programas).

Este login é válido apenas para o Nessus, onde é usado para fazer a autenticação no módulo servidor, ele não é um login de sistema. Para criá-lo, use o comando **"/opt/nessus/sbin/nessus-add-first-user"**. Ele pedirá o login e senha, o tipo de autenticação (escolha "pass") e permitirá que você adicione regras para o usuário (User Rules). Se você quiser apenas criar o usuário usando as regras default, basta pressionar **"Ctrl+D"**. Ele pedirá uma última confirmação, basta responder "y":

```
# /opt/nessus/sbin/nessus-add-first-user
```

```
Using          /var/tmp      as      a      temporary      file      holder
Add           a             new     nessusd       user

-----
Login          :                   tux
Authentication (pass/cert) : [pass] : pass
Login          password : *****

Login password (again) : *****

User          rules
-----
nessusd has a rules system which allows you to restrict the hosts
that tux has the right to test. For instance, you may want
him to be able to scan his own host only.
Please see the nessus-adduser(8) man page for the rules syntax
Enter the rules for this user, and hit ctrl-D once you are done:
(the user can have an empty rules set)
^D

Login          :                   tux
Password      : ***** : :
DN            : : :
Rules          : : :

Is that ok? (y/n) [y] y
```

Uma vez instalado, você pode iniciar o servidor Nessus usando o comando:

```
#                               /etc/init.d/nessusd      start
ou:
# /opt/nessus/sbin/nessusd -D
```

Em ambos os casos, ele roda em background, sem obstruir o terminal. Para fechá-lo, use o comando **"killall nessusd"**.

Isto conclui a instalação do servidor. O próximo passo é instalar o pacote do cliente. No site você pode baixar tanto o cliente Linux, quanto o NessusWx, que roda em máquinas Windows.

No meu caso, tive um pouquinho de trabalho para instalar o cliente Linux, pois, no momento em que escrevi este tópico, ainda não estava disponível uma versão do cliente para o Debian, de forma que precisei baixar o pacote para o Fedora 5, convertê-lo usando o alien e criar dois links simbólicos para bibliotecas com nomes diferentes nos dois sistemas.

O primeiro passo foi instalar o alien via apt-get e usá-lo para converter o pacote baixado do site:

```
# apt-get install alien
# alien NessusClient-1.0.0.RC5-fc5.i386.rpm
```

O alien gera um pacote .deb com o mesmo nome do pacote original, que pode ser instalado usando o dpkg, como em:

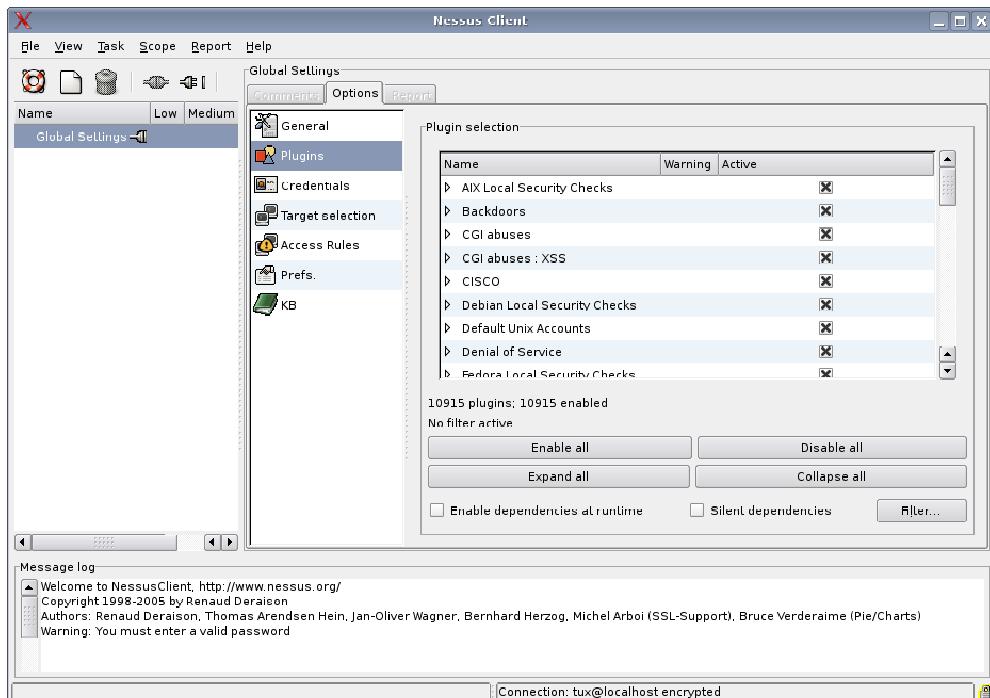
```
# dpkg -i nessusclient_1.0.0.RC5-1_i386.deb
```

O NessusClient é aberto usando o comando "NessusClient" (que você executa usando sua conta de usuário e não como root). Entretanto, por instalar uma versão para outra distribuição, ele reclamou da falta das bibliotecas "libssl.so.6" e "libcrypto.so.6". Na verdade, ambas estavam disponíveis, porém com nomes diferentes. Acessando o diretório "/usr/lib" vi que existiam os " libssl.so.0.9.8" e " libcrypto.so.0.9.8", de forma que precisei apenas criar dois links, apontando para eles:

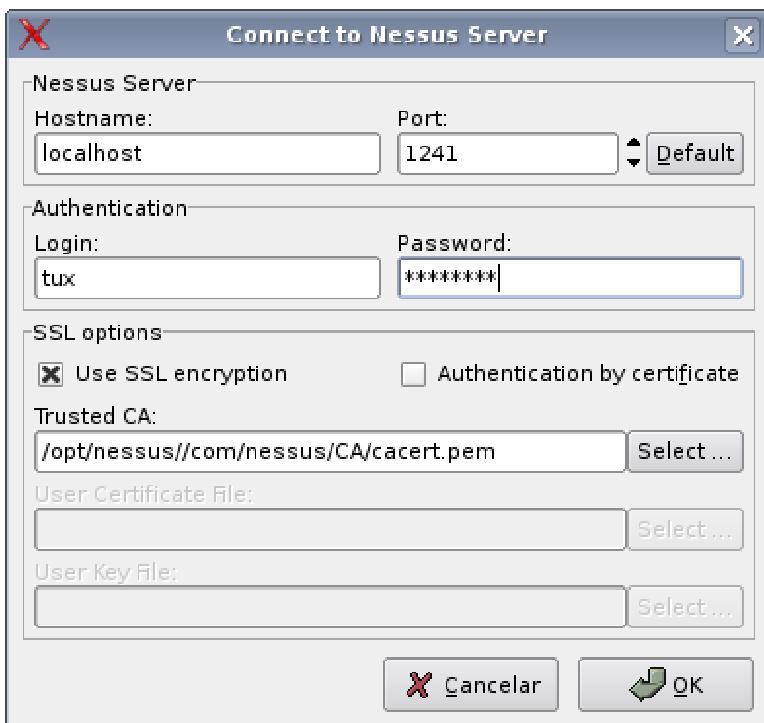
```
# cd /usr/lib
# ln -s libcrypto.so.0.9.8 libcrypto.so.6
# ln -s libssl.so.0.9.8 libssl.so.6
```

A partir daí, o NessusClient passou a abrir corretamente:

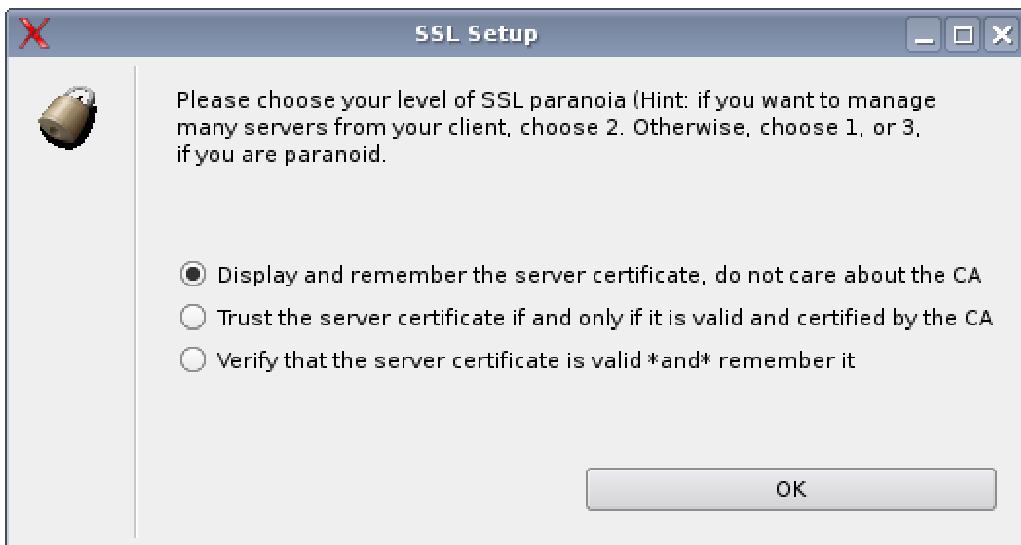
```
$ NessusClient
```



A interface desta versão é bem diferente da usada no cliente que acompanhava o Nessus 2.x, mas o funcionamento é basicamente o mesmo. Comece clicando no botão "Connect" para abrir a conexão com o servidor. Se estiver rodando-o na mesma máquina, use "localhost" como endereço, justamente com o login e senha criados. Caso contrário, forneça o IP correto da máquina onde o servidor está ativo:



Da primeira vez que se conectar, ele perguntará sobre o certificado do servidor. Ele (certificado) permite que você verifique a autenticidade do servidor onde está se conectando, evitando a possibilidade de que alguém o tenha substituído por outra máquina.



Ao usar servidores Nessus remotos, você pode usar certificados de autenticidade para melhorar a segurança. Nesse caso, use o comando nessus-mkcert-client (no servidor):

```
# /opt/nessus/bin/nessus-mkcert-client
```

Ao terminar o processo, ele salva o certificado gerado em uma pasta temporária, como em:

```
Your client certificates are in /tmp/nessus-mkcert.9904
You will have to copy them by hand
```

Dentro da pasta você encontra um arquivo ".pem" com o nome do usuário criado, como em "cert_nessuswx_joao.pem". Para usar este certificado, você deve copiá-lo para a pasta "/opt/nessus/com/nessus/CA/" (do cliente). Na hora de se conectar usando o NessusClient, marque a opção "Autentication by certificate" e indique a localização do arquivo.

Note que o uso do certificado apenas melhora a segurança da comunicação entre o servidor Nessus e o cliente. É uma medida saudável para os paranóicos de plantão :).

Uma vez conectado ao servidor Nessus, você pode definir diversas opções dentro da aba "Global Settings". Algumas opções interessantes dentro da aba "**General**" são:

Port range: O default é escanear apenas as portas de 1 a 1024, o que resulta em testes relativamente rápidos, mas que deixam passar serviços escondidos em portas altas. Para que ele escaneie todas as portas, mude para "1-65535". Note que isso torna o teste muito mais demorado, pois ele precisa enviar um pacote TCP e outro UDP para cada uma das portas, para então executar os testes adicionais nas portas abertas.

Number of hosts to test at the same time: Esta opção determina o número de hosts que serão verificados simultaneamente durante o teste. O default para esta opção são 20 hosts, o que é adequado para situações onde você use um micro de configuração modesta para executar o teste dentro da rede local. Aumentar o número faz com que o Nessus consuma mais recursos do servidor e mais banda da rede, o que não é muito interessante caso o teste seja realizado durante o expediente.

Number of checks to perform at the same time: Esta opção determina o número de testes simultâneos em cada um dos micros escaneados. Muitos dos testes do Nessus são demorados, porém geram pouco tráfego de rede. Aumentar o número de testes simultâneos é uma boa forma de acelerar o teste caso sua rede tenha poucos micros.

Note que é aberta uma instância do scanner para cada host e para cada teste. Ou seja, com 100 hosts e 4 testes simultâneos em cada um, podem ser abertas até 400 instâncias, o que consumirá quase 500 MB de memória do servidor. Se você está rodando o servidor em uma máquina de configuração modesta, ou está usando sua máquina de trabalho e não deseja que ela fique muito lenta durante o teste, reduza os números.

Optimize the test: Esta opção torna o teste do Nessus mais "inteligente". Baseado em informações de testes anteriores, o scanner evita usar testes demorados, que tenham baixa probabilidade de revelar novas vulnerabilidades. Ativar esta opção, torna o teste muito mais rápido, mas abre uma pequena possibilidade de algumas vulnerabilidades mais incomuns não serem descobertas.

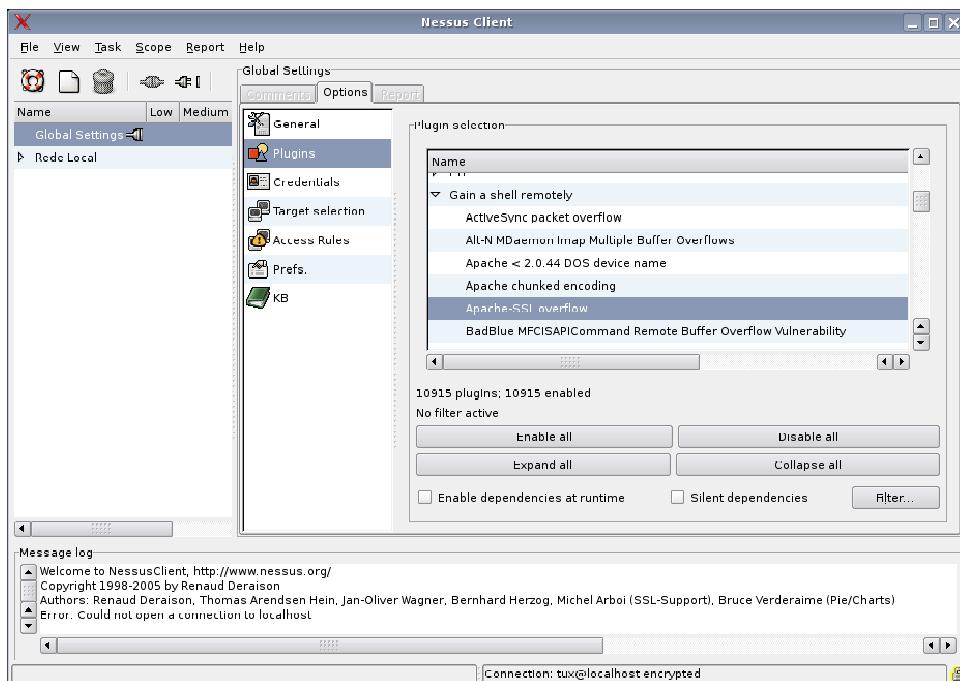
Safe checks: Alguns sistemas抗igos possuem brechas de segurança que podem causar travamentos. Máquinas com o Windows 95, sem atualizações de segurança, por exemplo, são vulneráveis ao famoso "ping da morte", um pacote ICMP defeituoso, que trava o sistema. Ativando esta opção, o Nessus deixa de realizar os testes que podem levar a

travamentos das máquinas, ou de outros dispositivos da rede, como roteadores e modems ADSL.

Designate hosts by their MAC address: Ativando esta opção, os hosts são identificados pelo endereço MAC no relatório do teste, ao invés de pelo endereço IP. Isto pode ser útil em redes onde os clientes são configurados via DHCP.

Na aba "Plugins", você tem acesso à configuração dos plugins, que são scripts responsáveis por detectar vulnerabilidades específicas. Por exemplo, ao detectar que a porta "45234" está aberta, o Nessus primeiro tenta identificar qual servidor está ativo, executando um conjunto de testes. Se for detectado um servidor Apache, por exemplo, serão usados os plugins que detectam vulnerabilidades em servidores web.

O Nessus inclui um número assustador de plugins, divididos em categorias. Ao marcar a opção "Safe checks" (na aba general), são automaticamente desativados os plugins potencialmente perigosos, mas você pode reativar todos clicando no "Enable all".



Naturalmente, novas brechas de segurança são descobertas todos os dias, por isso é necessário atualizar os plugins periodicamente. Para isso, use o comando "/opt/nessus/bin/nessus-fetch", informando o código fornecido no e-mail de confirmação do registro, como em:

```
# /opt/nessus/bin/nessus-fetch --register FFCB-382E-3990-D3DA-2BFC
```

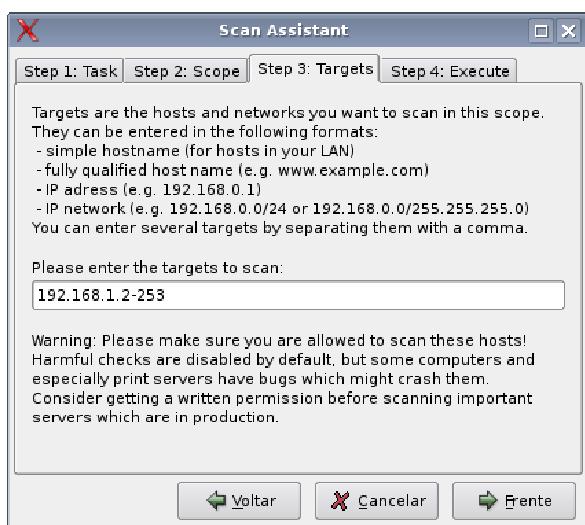
A partir daí, os plugins são atualizados automaticamente, uma vez por dia. Se preferir desativar a atualização automática, edite o arquivo "/opt/nessus/etc/nessus/nessusd.conf", substituindo a linha "auto_update = yes" por "auto_update = no".

Os plugins são os componentes mais importantes do Nessus. São eles que o diferenciam de um portscan genérico, como o Nmap. O portscan detecta que uma determinada porta está aberta e qual servidor está sendo usado, mas são os plugins que informam que está sendo usada uma versão com a vulnerabilidade X, que pode ser corrigida com a atualização Y.

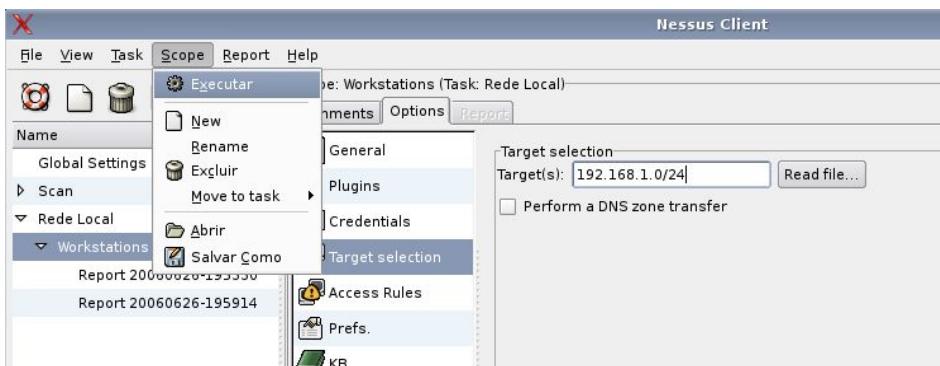
As atualizações gratuitas são fornecidas com um delay de 7 dias, o que dificulta seu uso profissional. Para ter acesso às atualizações em tempo real, você precisa assinar o plano comercial.

Depois de ajustar as opções gerais, clique na opção "File > Scan Assistant" para definir a faixa de endereços que será escaneada. A primeira tarefa é definir nomes de identificação do teste nas abas "Task" e "Scope".

Depois vem o que interessa, a definição dos alvos na aba "Targets". Você pode tanto lançar o teste contra um IP isolado, quanto contra uma faixa de endereços. Neste caso, você pode indicar uma faixa de endereços, como em "192.168.1.0/24" (o 24 indica a máscara de sub-rede), ou um intervalo de endereços, como em "192.168.1.2-253". Clicando no "Executar" o teste é finalmente lançado:



Uma novidade sobre as versões anteriores é que agora você pode definir várias faixas diferentes e criar uma configuração independente para cada uma. Você pode, por exemplo, ter uma configuração para o teste contra hosts da rede local e outra já engatilhada para testar periodicamente o servidor que hospeda seu site, por exemplo. Cada uma permite definir faixas de portas e configurações diferentes. Para lançar um teste já pré-configurado, selecione-o na lista da esquerda e clique no "Scope > Executar".



O teste do Nessus é feito em duas partes. A primeira é o portscan, onde ele utiliza o Nmap, combinado com alguns testes adicionais para descobrir quais portas estão abertas em cada host. A partir daí, entram em ação os plugins, que testam cada porta em busca de vulnerabilidades conhecidas.

Concluído o teste, ele exibe uma lista com as vulnerabilidades encontradas em cada PC. Existem três níveis de alerta, o primeiro e mais grave tem o símbolo de uma luz vermelha e indica uma brecha de segurança em um servidor ativo na máquina. No screenshot, por exemplo, temos uma instalação bastante desatualizada do Windows XP, com diversas vulnerabilidades, entre elas uma vulnerabilidade no protocolo SMB (responsável pelo compartilhamento de arquivos), que permite travar a máquina remotamente e duas vulnerabilidades graves, que permitem executar código e obter acesso à máquina.

The screenshot shows the Nessus Client report interface. The top navigation bar includes 'File', 'View', 'Task', 'Scope', 'Report', and 'Help'. The 'Report' tab is active. The main window displays a report for the scope 'Workstations (Task: Rede Local)'. On the left, there's a tree view of hosts under 'Hosts' (192.168.1.10, 192.168.1.77, 192.168.1.78, 192.168.1.221, 192.168.1.11, 192.168.1.12, 192.168.1.13, 192.168.1.33). The central pane shows a table with columns 'Port', 'Severity', and 'Description'. The 'Port' column lists various ports and services (e.g., microsoft-ds (445/tcp), epmap (135/udp), blackjack (1025/tcp), complex-main (5000/tcp), ntp (123/udp), netbios-ssn (139/tcp), netbios-ns (137/tcp)). The 'Severity' column indicates three levels: 'Security Hole' (red), 'Security Warning' (orange), and 'Security Note' (green). A specific entry for host 192.168.1.10 is expanded, detailing a 'Security Hole' in the SMB stack. The description notes that the remote host is vulnerable to a denial of service attack in its SMB stack, and an attacker may exploit this flaw to crash the remote host remotely, without any kind of authentication. It provides a solution URL (<http://www.microsoft.com/technet/security/bulletin/ms02-045.mspx>), risk factor (High), CVE number (CVE-2002-0724), and BID (5556). The synopsis and description sections provide further details about the vulnerability.

Veja que, além de apontar o problema, o Nessus oferece uma descrição detalhada da vulnerabilidade e aponta uma solução. Na maioria dos casos, o problema é corrigido simplesmente instalando as atualizações de segurança ou atualizando para a versão mais recente. Em casos onde o problema é gerado por erros de configuração, ele quase sempre fornece dicas de como corrigi-lo.

O teste do Nessus permite também identificar serviços indesejados, que podem ser desativados ou ter suas portas bloqueadas no firewall, além de avisar sobre qualquer backdoor que tenha sido instalado sem seu conhecimento.

Continuando, o segundo nível é um alerta de que um serviço potencialmente inseguro está ativo em uma determinada porta do sistema, como, por exemplo, um servidor Telnet ou XDMCP. Neste caso, não foi encontrada nenhuma vulnerabilidade específica, mas o fato de o serviço ser fundamentalmente inseguro já representa uma brecha de segurança. Tanto o Telnet quanto o XDMCP transmitem dados de forma não encriptada, o que permite que alguém mal intencionado possa sniffar a rede, capturando os dados transmitidos, incluindo as senhas dos usuários. Ambos devem ser usados apenas dentro da rede local.

The remote host is running XDMCP.

This protocol is used to provide X display connections for X terminals. XDMCP is completely insecure, since the traffic and passwords are not encrypted.



An attacker may use this flaw to capture all the keystrokes of the users using this host through their X terminal, including passwords.

Risk factor : Medium

Solution : Disable XDMCP

O terceiro nível de alerta tem o símbolo de uma luz. Estes são apenas lembretes de que existe um servidor ativo na porta indicada, mas sem que fosse detectada qualquer brecha de segurança.

Remote FTP server banner :

💡 220 ProFTPD 1.2.5rc1 Server (ProFTPD Default Installation) [spartacus]::

Como disse, em muitos casos, o Nessus ajuda também a detectar erros de configuração, que podem ser perigosos. Por exemplo, este é um aviso de segurança, que mostra um servidor dedicado com o servidor DNS aberto para consultas a outros domínios:

The screenshot shows the Nessus application interface with a report titled 'Report for scope: cani (Task: Scan)'. The main pane displays a table with columns for Subnet, Port, and Severity. A single row is highlighted in blue, indicating a 'Security Warning' for port 53 (DNS). The 'Port' column lists various services: domain (53/udp), sunrpc (111/tcp), ssh (22/tcp), http (80/tcp), general/udp, general/tcp, general/icmp, and domain (53/tcp). The 'Severity' column shows a warning icon. Below the table, a 'Host' section shows the IP address 73.232.32.206 with a warning icon. To the right, a 'Solution' panel provides guidance on how to restrict recursive queries to hosts within the LAN by modifying the 'named.conf' file. It includes code snippets for 'allow-recursion' and 'hosts_definition_in_acl'. It also notes that this applies to Bind 9 and suggests consulting documentation for other name servers. A 'Risk factor' section indicates a medium risk with a CVSS base score of 4. At the bottom, the file path '/etc/bind/named.conf' is mentioned.

Todo servidor web trabalha em conjunto com um servidor DNS, que responde pelos domínios dos sites hospedados. Embora não seja por si só uma brecha de segurança, esta configuração faz com que o DNS se transforme em um servidor "público", que faz a resolução de qualquer domínio solicitado, assim como os servidores DNS dos provedores de acesso. Isso abre brecha para ataques de "DNS poisoning", onde um cliente externo insere uma entrada inválida no cache do DNS, fazendo com que ele responda a algumas das consultas com endereços IPs incorretos, além de abrir uma pequena possibilidade de que o servidor seja usado como auxiliar em ataques DoS contra outros servidores.

O próprio Nessus dá a dica de como corrigir o problema. Pesquisando no Google sobre a opção "allow-recursion" que ele sugere, você chega facilmente à artigos que sugerem a inclusão das quatro linhas abaixo no final do arquivo **"/etc/bind/named.conf"**:

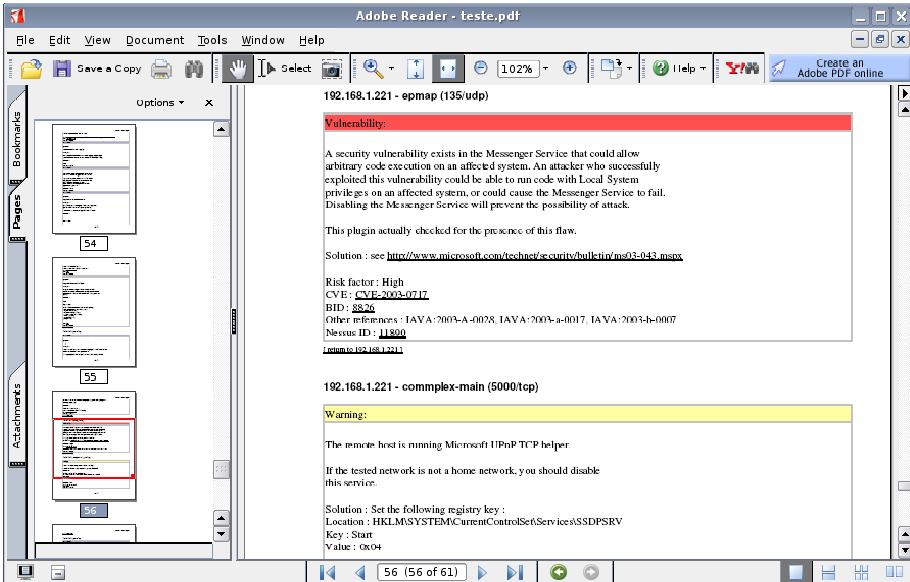
```
options
  directory
  recursion
}; {  
  "/var/named";
  no;
```

Elas fazem com que o servidor DNS responda apenas pelos domínios sobre os quais tem autoridade (ou seja, os domínios dos sites hospedados no servidor), corrigindo o problema. Executando o teste novamente, o Nessus continua detectando que a porta 53 está aberta, mas não acusa mais a falha.

Os relatórios gerados pelo Nessus podem ser salvos em diversos formatos, incluindo html, latex e PDF, um recurso extremamente interessante na hora de apresentar os resultados e explicar os problemas para outras pessoas.

Para isso, selecione o relatório que será exportado no menu da esquerda e use a opção "Report > Export". Para que o suporte à geração de arquivos PDF funcione, você deve ter instalado o pacote "htmldoc", disponível via apt-get:

```
# apt-get install htmldoc
```



Naturalmente, assim como você pode utilizar o Nessus para detectar e tapar brechas de segurança, outras pessoas podem utilizá-lo para detectar vulnerabilidades na sua rede e lançar ataques. Hoje em dia, a variedade de scripts e ferramentas gráficas prontas que exploram vulnerabilidades é tão grande que você pode encontrar algum exploit fácil de usar para praticamente qualquer vulnerabilidade que você possa encontrar. Basta saber fazer pesquisas no Google.

Estes exploits prontos são o grande perigo, pois não requerem nenhum tipo de prática ou habilidade para serem usados. Basta indicar o IP a ser atacado e pronto. Ou seja, aquele garoto com quem você brigou no chat pode muito bem fazer um estrago na sua rede caso algum serviço ativo no seu servidor possua alguma vulnerabilidade grave. É importante resolver o problema antes que alguém o faça por você.

O Nessus é só o primeiro passo. Caso você rode qualquer tipo de servidor na sua máquina, é importante acompanhar sites especializados em notícias relacionadas à segurança, como o <http://lwn.net> e o <http://www.linuxsecurity.com>. A maioria das distribuições oferecem boletins por e-mail que avisam quando novas atualizações de segurança estão disponíveis.

Lembre-se de que, apesar das notícias de brechas e atualizações serem sempre muito freqüentes, você só precisa se preocupar com os servidores que você mantém ativos na sua máquina. Se você mantém apenas o SSH e o FreeNX, por exemplo, não precisa se preocupar com as atualizações do Apache e do Sendmail.

Além dos servidores, clientes de e-mail (Evolution, Kmail, etc.) e navegadores (Firefox, Konqueror, etc.) também costumam receber atualizações de segurança com uma certa

freqüência. Estes programas clientes não podem ser atacados diretamente, ou seja, ninguém poderá explorar um buffer overflow no Firefox (por exemplo) apenas por ele estar instalado, seria necessário que você acessasse alguma página contendo o script malicioso. É aí que entram os ataques de engenharia social, como no caso dos e-mails com textos que tentam levá-lo a clicar em um link ou ao executar um arquivo anexado.

» Próximo: [Usando o Wireshark \(Ethereal\)](#)

Além do Nessus, outro aliado importante é o Wireshark, o bom e velho Ethereal, que mudou de nome em Junho de 2006. Ele é um poderoso sniffer, que permite capturar o tráfego da rede, fornecendo uma ferramenta poderosa para detectar problemas e entender melhor o funcionamento de cada protocolo.

Assim como o Nessus, ele pode ser usado tanto para proteger seu sistema quanto para roubar dados dos vizinhos, uma faca de dois gumes. Devido a isso, ele é às vezes visto como uma "ferramenta hacker" quando na verdade o objetivo do programa é dar a você o controle sobre o que entra e sai da sua máquina e a possibilidade de detectar rapidamente qualquer tipo de trojan, spyware ou acesso não autorizado.

Embora ele geralmente não venha instalado por padrão, a maioria das distribuições disponibilizam o pacote "wireshark" ou "ethereal", de acordo com o nível de atualização. Nas distribuições derivadas do Debian, você pode usar o apt-get, como de praxe.

É possível instalar também a partir do pacote com o código fonte, disponível no <http://www.wireshark.org/>, que deve ser instalada com os conhecidos "./configure", "make" e "make install". Como ele depende de um número relativamente grande de compiladores e de bibliotecas, muitas delas pouco comuns, você quase sempre vai precisar instalar alguns componentes adicionais manualmente.

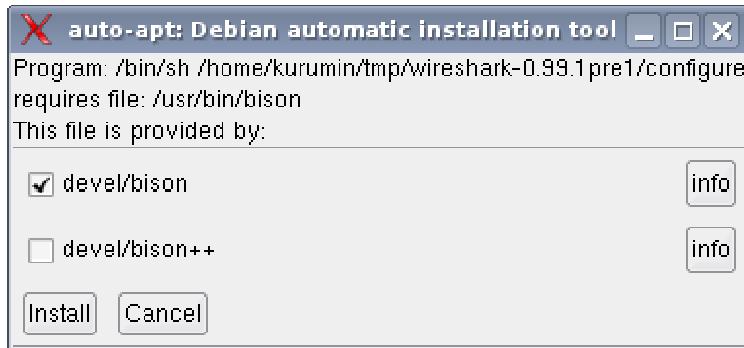
Uma forma simples de instalar todos os componentes necessários para a compilação é usar o "**auto-apt**", disponível através do apt-get. Para usá-lo, instale o pacote via apt-get e rode o comando "auto-apt update":

```
# apt-get install auto-apt
# auto-apt update
```

A partir daí, você pode rodar os comandos de compilação através dele, como em:

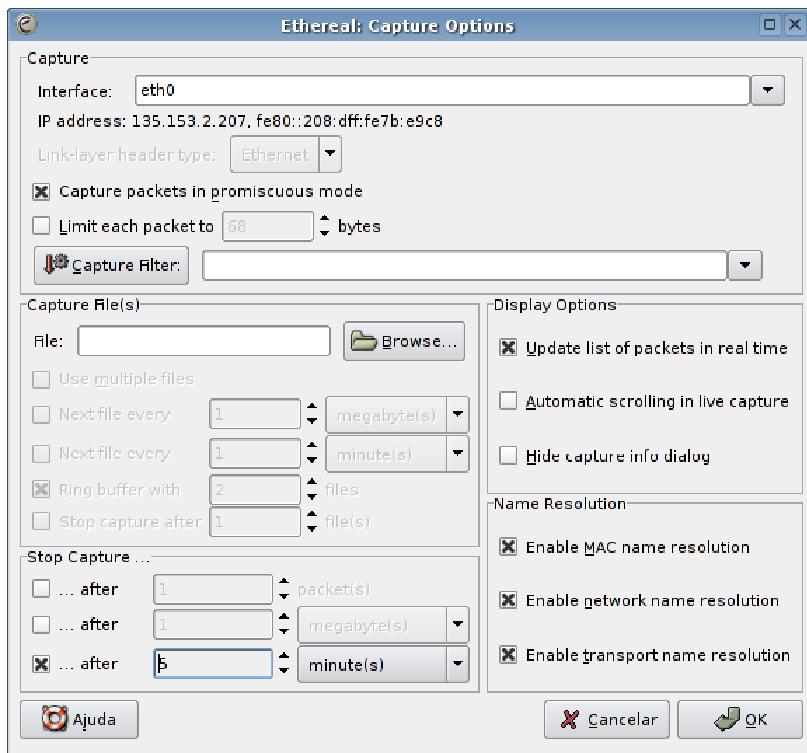
```
$ tar -zxvf wireshark-0.99.1pre1
$ cd wireshark-0.99.1pre1
$ ./configure
$ make
<senha>
# make install
```

Durante a instalação, o auto-apt usa o apt-get para instalar os componentes necessários, como neste screenshot:



Depois de instalado, abra o programa usando o comando "**wireshark**" ou "**ethereal**", de acordo com a versão instalada.

O Wireshark é um daqueles programas com tantas funções que você só consegue aprender realmente usando. Para começar, nada melhor do que capturar alguns pacotes. Clique em "Capture > Start".



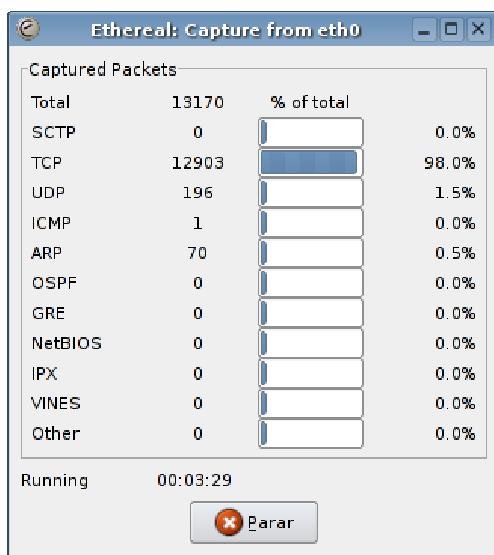
Aqui estão as opções de captura. A primeira opção importante é a "Capture packets in promiscuous mode", onde você decide se quer capturar apenas os pacotes endereçados à sua própria máquina, ou se quer tentar capturar também pacotes de outras máquinas da rede.

Isto é possível pois os hubs tradicionais apenas espelham as transmissões, enviando todos os pacotes para todas as estações. No início de cada pacote, vai o endereço MAC do destino. Normalmente, a placa escuta apenas os pacotes destinados a ela, ignorando os demais, mas, no promiscuous mode ela passa a receber todos os pacotes, independentemente de a qual endereço MAC ele se destine.

Em seguida, você tem a opção "Update list of packets in real time". Ativando esta opção, os pacotes vão aparecendo na tela conforme são capturados, em tempo real. Caso contrário, você precisa capturar um certo número de pacotes para só depois visualizar todo o bolo.

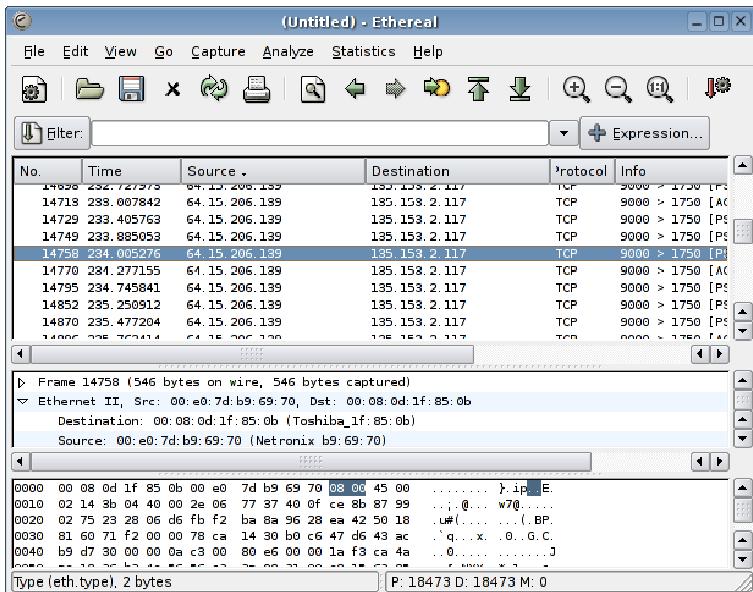
Mais abaixo estão também algumas opções para interromper a captura depois de um certo tempo ou depois de capturar uma certa quantidade de dados. O problema aqui é que o Ethereal captura todos os dados transmitidos na rede, o que (em uma rede local) pode rapidamente consumir toda a memória RAM disponível, até que você interrompa a captura e salve o dump com os pacotes capturados em um arquivo.

Dando o OK, será aberta a tela de captura de pacotes, onde você poderá acompanhar o número de pacotes capturados:

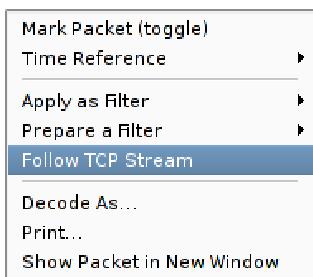


Na tela principal, temos a lista dos pacotes, com várias informações, como o remetente e o destinatário de cada pacote, o protocolo utilizado (TCP, FTP, HHTP, AIM, NetBIOS, etc.) e uma coluna com mais informações, que incluem a porta TCP à qual o pacote foi destinado.

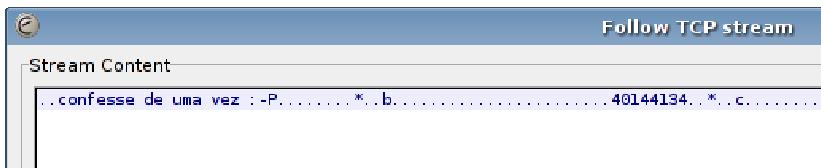
Os pacotes que aparecem com um micro da rede local como emissor e um domínio ou IP da internet como destinatário incluem requisições, upload de arquivos, e-mails enviados, mensagens de ICQ e MSN e, em muitos casos, também senhas de acesso. Os pacotes provenientes de microsserviços da internet são respostas à estas requisições, incluindo páginas web, e-mails lidos, arquivos baixados e, assim por diante. Através do sniffer, é possível capturar todo tipo de informação que trafegue de forma não encriptada pela rede.



Clicando sobre um dos pacotes e, em seguida, no "Follow TCP Stream", o Ethereal mostrará uma janela com toda a conversão, exibida em modo texto.



A maior parte do que você vai ver serão dados binários, incluindo imagens de páginas web e arquivos diversos. Mesmo o html das páginas chega muitas vezes de forma compactada (para economizar banda), novamente em um formato ilegível. Mas, garimpando, você vai encontrar muitas coisas interessantes, como, por exemplo, mensagens (MSN e ICQ) e e-mails, que, por padrão, são transmitidos em texto puro. Usando a opção "Follow TCP Stream", é possível rastrear toda a conversa.



Como disse, o Wireshark pode ser usado também pelo lado negro da força. Se você estiver em uma rede local, com micros ligados através de um hub ou através de uma rede wireless, outro usuário pode usá-lo para capturar todas as suas transmissões.

Isto é extremamente perigoso. Qualquer um, que tenha a chance de plugar um notebook na rede ou colocá-lo dentro da área de cobertura de sua rede wireless, poderá capturar dados e

senhas suficientes para comprometer boa parte do sistema de segurança da sua empresa. Apenas conexões feitas através do SSH e outros programas que utilizam encriptação forte estariam a salvo.

Naturalmente, além de alguém de fora, existe a possibilidade de um dos seus próprios funcionários resolver começar a brincar de script kiddie, pregando peças nos outros e causando danos. Como vimos, isso não requer muita prática. Enfim, a menos que você esteja em uma simples rede doméstica, onde exista uma certa confiança mútua, utilizar um hub burro é simplesmente um risco grande demais a correr.

Ao utilizar um hub-switch, o risco é um pouco menor, já que, por default, os pacotes são enviados apenas às portas corretas. Entretanto, muitos sistemas são vulneráveis a ataques de ARP poisoning, sem falar dos ataques de MAC flooding, que permitem burlar a proteção.

No ARP poisoning, o micro do atacante envia pacotes com respostas forjadas para requisições ARP de outros micros da rede. Como vimos no capítulo 3, o ARP é utilizado para descobrir os endereços MAC dos demais micros da rede, já que os switchs não entendem endereços IP. Estes pacotes forjados fazem com que os outros micros passem a enviar seus pacotes para o micro do atacante, que é configurado para retransmiti-los para os destinatários corretos.

A rede continua funcionando normalmente, mas agora o atacante tem chance de logar todo o tráfego, usando o Wireshark ou outro sniffer. Felizmente, o Wireshark também pode ser usado para perceber as anormalidades na rede e chegar até o espertinho.

Os ataques de MAC flooding, por sua vez, tem como alvo o switch da rede e trabalham dentro de um princípio bastante simples. O switch possui uma área limitada de memória para armazenar a tabela com os endereços MAC dos micros da rede, de forma que, ao receber um grande número de pacotes com endereços MAC forjados, a tabela é completamente preenchida com os endereços falsos, não deixando espaço para os verdadeiros. Para não travar, o switch passa a trabalhar em modo hub, desativando a tabela e passando a simplesmente encaminhar todos os pacotes para todas as portas (até ser reiniciado), permitindo ao atacante capturar todos os pacotes da rede.

Em outras situações, pode ser que você mesmo, como administrador da rede, precise policiar o que os usuários estão fazendo durante o expediente na conexão da empresa. Neste caso, eu sugiro que você mantenha um servidor SSH ativo nas estações de trabalho. Assim, você pode se logar em cada uma das máquinas, sempre que necessário, e rodar o Wireshark para acompanhar o tráfego de dados de cada uma, sem que o usuário tome conhecimento.

Outra possibilidade seria rodar o Wireshark na máquina que compartilha a conexão, assim você poderá observar os pacotes vindos de todas as máquinas da rede. Alguns modelos de switchs mais caros podem ser programados para direcionar todo o tráfego da rede para uma determinada porta, onde você poderia plugar o seu micro para "ver tudo".

No caso das redes wireless, a situação é um pouco mais complicada, pois o meio de transmissão é sempre compartilhado. Os pacotes trafegam pelo ar, por isso não é possível

impedir que sejam capturados. Mas, você pode dificultar bastante as coisas ativando a encriptação, se possível usando o WPA e diminuindo a potência de transmissão do ponto de acesso, de forma a cobrir apenas a área necessária.

Lembre-se de que apenas informações não encriptadas podem ser capturadas. Utilizando protocolos seguros, como o SSH, as informações capturadas não terão utilidade alguma, pois estarão encriptadas.

Além do lado negativo, pessoas sniffarem sua rede e assim descobrirem senhas e outros dados sigilosos, existe um lado positivo: monitorando sua conexão durante algum tempo, você vai logo perceber vários tipos de abusos, como sites que enviam requisições para várias portas da sua máquina ao serem acessados, banners de propaganda que enviam informações sobre seus hábitos de navegação para seus sites de origem, gente escaneando suas portas usando o Nessus ou outros programas similares, programas que ficam continuamente baixando banners de propaganda e, assim por diante.

Estas informações são úteis não apenas para decidir quais sites e serviços evitar, mas também para ajudar na configuração do seu firewall. Pode ser que no início você não entenda muito bem os dados fornecidos pelo Wireshark, mas, depois de alguns dias observando, você vai começar a entender muito melhor como as conexões TCP funcionam.

» Próximo: [Entendendo \(e quebrando\) a segurança em redes Wireless](#)

Um dos grandes problemas em uma redes wireless é que os sinais são transmitidos pelo ar. Os pontos de acesso e placas utilizam por padrão antenas baratas, que proporcionam um alcance reduzido, fazendo com que o sinal da sua rede possa ser capturado de muito mais longe por alguém com uma antena de alto ganho.

Não existe como impedir que o sinal se propague livremente pelas redondezas (a menos que você pretenda ir morar em um bunker, com paredes reforçadas com placas de aço), de forma que a única forma eficaz de proteção é encriptar toda a transmissão, fazendo com que as informações capturadas não tenham serventia.

Como esta questão da segurança em redes wireless é muito divulgada, a maior parte das redes já utiliza algum tipo de proteção, seja ativando o WEP ou WPA, seja bloqueando os micros que podem se conectar ao ponto de acesso com base no endereço MAC.

Este tópico se destina a mostrar como é fácil burlar a maioria destas proteções e quebrar a encriptação do WEP (inclusive do WEP de 128 bits), usando ferramentas simples.

É melhor que você conheça os ataques mais usados e veja você mesmo como é possível derrubar cada uma das proteções que utilizamos em uma rede típica, do que ficar com um falso senso de segurança, achando que o WEP de 128 é inquebrável, que não é possível

detectar um ponto de acesso com o SSID broadcast desativado, ou que não é possível burlar a restrição de acesso baseada no endereço MAC usada em muitas redes.

» Próximo: [Usando o Kismet](#)

O Kismet é uma ferramenta poderosa, que pode ser usada tanto para checar a segurança de sua própria rede wireless, quanto para checar a presença de outras redes próximas e assim descobrir os canais que estão mais congestionados (e assim configurar sua rede para usar um que esteja livre) ou, até mesmo, invadir redes. O Kismet em si não impõe restrições ao que você pode fazer. Assim como qualquer outra ferramenta, ele pode ser usado de forma produtiva ou destrutiva, de acordo com a índole de quem usa. A página do projeto é a: <http://www.kismetwireless.net/>.

A principal característica do Kismet é que ele é uma ferramenta passiva. Ao ser ativado, ele coloca a placa wireless em modo de monitoramento (rfmon) e passa a escutar todos os sinais que chegam até sua antena. Mesmo pontos de acesso configurados para não divulgar o ESSID ou com a encriptação ativa são detectados.

Como ele não transmite pacotes, apenas escuta as transmissões, todo o processo é feito sem prejudicar as redes vizinhas, de forma praticamente indetectável. A principal limitação é que, enquanto está em modo de monitoramento, a placa não pode ser usada para outros fins. Para conectar-se a uma rede, você precisa primeiro parar a varredura.

Esta questão da detecção dos pontos de acesso com o ESSID desativado é interessante. Não é possível detectá-los diretamente, pois eles não respondem a pacotes de broadcast (por isso eles não são detectados por programas como o Netstumbler), mas o Kismet é capaz de detectá-los quando um cliente qualquer se associa a eles, pois o ESSID da rede é transmitido de forma não encriptada durante o processo de associação do cliente.

A partir daí, o Kismet passa a capturar todos os pacotes transmitidos. Caso a rede esteja encriptada, é possível descobrir a chave de encriptação usando o aircrack (que veremos a seguir), permitindo tanto escutar as conexões, quanto ingressar na rede.

Como o Kismet é uma das ferramentas mais usadas pelos crackers, é sempre interessante usá-lo para verificar a segurança da sua própria rede. Tente agir como algum vizinho obstinado agiria, capturando os pacotes ao longo de alguns dias. Verifique a distância de onde consegue pegar o sinal de sua rede e quais informações consegue descobrir. Depois, procure meios de reforçar a segurança da rede e anular o ataque.

Por ser uma ferramenta popular, ele está disponível na maioria das distribuições. Algumas, como o Knoppix (a partir da versão 3.7), já o trazem instalado por padrão.

Nas distribuições derivadas do Debian, você pode instalá-lo via apt-get:

```
# apt-get install kismet
```

Antes de ser usar, é preciso configurar o arquivo "/etc/kismet/kismet.conf", especificando a placa wireless e o driver usado por ela, substituindo a linha:

```
source=none,none,addme
```

Por algo como:

```
source=madwifi_ag,ath0,atheros
```

... onde o "madwifi_ag" é o driver usado pela placa (você pode verificar o chipset da placa instada usando o comando **lspci**). Na documentação do Kismet, o driver é chamado de "capture source", pois é a partir dele que o Kismet obtém os pacotes recebidos.

o "ath0" é a interface (que você vê através do comando **ifconfig**) e o "atheros" é um apelido para a placa (que você escolhe), com o qual ela será identificada dentro da tela de varredura.

Isso é necessário, pois o Kismet precisa de acesso de baixo nível ao hardware, mas, por outro lado, faz com que a compatibilidade esteja longe de ser perfeita. Diversas placas não funcionam em conjunto com o Kismet, com destaque para as placas que não possuem drivers nativos e precisam ser configurados através do ndiswrapper. Se você pretende usar o Kismet, o ideal é pesquisar antes de comprar a placa.

Naturalmente, para que possa ser usada no Kismet, a placa precisa ter sido detectada pelo sistema, com a ativação dos módulos de Kernel necessários. Por isso, prefira sempre usar uma distribuição recente, que traga um conjunto atualizado de drivers. O Kurumin e o Kanotix estão entre os melhores neste caso, pois trazem muitos drivers que não vem pré instalados em muitas distribuições.

Você pode ver uma lista detalhada dos drivers de placas wireless disponíveis e como instalar manualmente cada um deles no meu livro *Linux Ferramentas Técnicas*.

Veja uma pequena lista dos drivers e placas suportados no Kismet 2006-04-R1:

acx100: O chipset ACX100 foi utilizado em placas de diversos fabricantes, entre eles a DLink, sendo depois substituído pelo ACX111. O ACX100 original é bem suportado pelo Kismet, o problema é que ele trabalha a 11 megabits, de forma que não é possível testar redes 802.11g.

admtek: O ADM8211 é um chipset de baixo custo, encontrado em muitas placas baratas. Ele é suportado no Kismet, mas possui alguns problemas. O principal é que ele envia pacotes de broadcast quando em modo monitor, fazendo com que sua varredura seja detectável em toda a área de alcance do sinal. Qualquer administrador esperto vai perceber que você está capturando pacotes.

bcm43xx: As placas com chipset Broadcom podiam até recentemente ser usadas apenas no ndiswrapper. Recentemente, surgiu um driver nativo (<http://bcm43xx.berlios.de>) que passou a ser suportado no Kismet. O driver vem incluído por padrão a partir do Kernel 2.6.17, mas a compatibilidade no Kismet ainda está em estágio experimental.

ipw2100, ipw2200, ipw2915 e ipw3945: Estes são os drivers para as placas Intel, encontrados nos notebooks Intel Centrino. O Kismet suporta toda a turma, mas você precisa indicar o driver correto para a sua placa, entre os quatro. O ipw2100 é o chipset mais antigo, que opera a 11 megabits; o ipw2200 é a segunda versão, que suporta tanto o 802.11b, quanto o 802.11g; o ipw2915 é quase idêntico ao ipw2200, mas suporta também o 802.11a, enquanto o ipw3945 é uma versão atualizada, que é encontrada nos notebooks com processadores Core Duo.

madwifi_a, madwifi_b, madwifi_g, madwifi_ab e madwifi_ag: Estes drivers representam diferentes modos de operação suportados pelo driver madwifi (<http://sourceforge.net/projects/madwifi/>), usado nas placas com chipset Atheros. Suportam tanto o driver madwifi antigo, quanto o madwifi-ng. Usando os drivers madwifi_a, madwifi_b ou madwifi_g, a placa captura pacotes apenas dentro do padrão selecionado (o madwifi_a captura apenas pacotes de redes 802.11a, por exemplo). O madwifi_g é o mais usado, pois captura simultaneamente os pacotes de redes 802.11b e 802.11g. O madwifi_ag, por sua vez, chaveia entre os modos "a" e "g", permitindo capturar pacotes de redes que operam em qualquer um dos três padrões, apesar de em um ritmo mais lento, devido ao chaveamento.

rt2400 e rt2500: Estes dois drivers dão suporte às placas com chipset Ralink, outro exemplo de chipset de baixo custo que está se tornando bastante comum. Apesar de não serem exatamente "placas de alta qualidade", as Ralink possuem um bom suporte no Linux, graças em parte aos esforços do próprio fabricante, que abriu as especificações e fornece placas de teste para os desenvolvedores. Isto contrasta com a atitude hostil de alguns fabricantes, como a Broadcom e a Texas (que fabrica os chipsets ACX).

rt8180: Este é o driver que oferece suporte às placas Realtek 8180. Muita gente usa estas placas em conjunto com o ndiswrapper, mas elas possuem um driver nativo, disponível no <http://rtl8180-sa2400.sourceforge.net/>. Naturalmente, o Kismet só funciona caso seja usado o driver nativo.

prism54g: Este driver dá suporte às placas com o chipset Prism54, encontradas tanto em versão PCI ou PCMCIA, quanto em versão USB. Estas placas são caras e por isso relativamente incomuns no Brasil, mas são muito procuradas entre os grupos que fazem wardriving, pois as placas PCMCIA são geralmente de boa qualidade e quase sempre possuem conectores para antenas externas, um pré-requisito para usar uma antena de alto ganho e assim conseguir detectar redes distantes.

orinoco: Os drivers para as placas com chipset Orinoco (como as antigas Orinoco Gold e Orinoco Silver) precisam de um conjunto de patches para funcionar em conjunto com o Kismet, por isso acabam não sendo placas recomendáveis. Você pode ver detalhes sobre a instalação dos patches no http://www.kismetwireless.net/HOWTO-26_Orinoco_Rfmon.txt.

Depois de definir o driver, a interface e o nome no `"/etc/kismet/kismet.conf"`, você pode abrir o Kismet chamando-o como root:

```
# kismet
```

Name	T	W	Ch	Packts	Flags	IP Range
! <r3d3m3taann35d1a5>	A	Y	001	9865	A4	10.1.1.1
! <no ssid>	A	Y	011	33274		0.0.0.0
+ <Data Networks>	G	N	---	8		0.0.0.0

Status

```
Found SSID "p3l0ta5" for cloaked network BSSID 00:02:2D:A9:EE:24
Associated probe network "00:90:4B:AB:92:37" with "00:50:50:81:81:01" via da
Saving data files.
Saving data files.
```

Battery: AC charging 44%

Inicialmente, o Kismet mostra as redes sem uma ordem definida, atualizando a lista conforme vai descobrindo novas informações. Pressione a tecla "s" para abrir o menu de organização, onde você pode definir a forma como a lista é organizada, de acordo com a qualidade do canal, volume de dados capturados, nome, etc. Uma opção comum (dentro do menu sort) é a "c", que organiza a lista baseado no canal usado por cada rede.

Por padrão, o Kismet chaveia entre todos os canais, tentando detectar todas as redes disponíveis. Neste modo, ele captura apenas uma pequena parte do tráfego de cada rede, assim como você só assiste parte de cada programa ao ficar zapiando entre vários canais da TV.

Selecione a rede que quer testar usando as setas e pressione "shift + L" (L maiúsculo) para travá-lo no canal da rede especificada. A partir daí, ele passa a concentrar a atenção em uma única rede, capturando todos os pacotes transmitidos:

Name	T	W	Ch	Packts	Flags	IP Range
<Data Networks>	G	N	---	13		0.0.0.0
<r3d3m3taann35d1a5>	A	Y	001	11511	A4	10.1.1.1
! <minharede>	A	Y	011	336706		0.0.0.0

Info

- Ntwrks 14
- Pckets 658773
- Cryptd 336899
- Weak 593
- Noise 0
- Discrd 0
- Pkts/s 1004
- madwif Ch: 11
- Elapsed 01:40:03

Status

Saving data files.
Saving data files.
Saving data files.
Saving data files.

Battery: AC charging 95%

Você pode também ver informações detalhadas sobre cada rede selecionando-a na lista e pressionando **enter**. Pressione "**q**" para sair do menu de detalhes e voltar à tela principal.

Outro recurso interessante é que o Kismet avisa sobre "clientes suspeitos", micros que enviam pacotes de conexão para os pontos de acesso, mas nunca se conectam a nenhuma rede, indício de que provavelmente são pessoas fazendo wardriving ou tentando invadir redes. Este é o comportamento de programas como o Netstumbler (do Windows). Micros rodando o Kismet não disparam este alerta, pois fazem o scan de forma passiva:

ALERT: Suspicious client 00:12:F0:99:71:D1 - probing networks but never participating.

O Kismet gera um dump contendo todos os pacotes capturados, que vai por padrão para a pasta **"/var/log/kismet/"**. A idéia é que você possa examinar o tráfego capturado posteriormente usando o Wireshark (Ethereal), que permite abrir o arquivo e examinar os dados capturados. O problema é que, ao sniffar uma rede movimentada, o dump pode se transformar rapidamente em um arquivo com vários GB, exigindo que você reserve bastante espaço no HD.

Um dos maiores perigos em uma rede wireless é que qualquer pessoa pode capturar o tráfego da sua rede e depois examiná-lo calmamente em busca de senhas e outros dados confidenciais transmitidos de forma não encriptada. O uso do WEP ou outro sistema de encriptação minimiza este risco, pois antes de chegar aos dados, é necessário quebrar a encriptação.

Evite usar chaves WEP de 64 bits, pois ele pode ser quebrado via força bruta caso seja possível capturar uma quantidade razoável de pacotes da rede. As chaves de 128 bits são um pouco mais seguras, embora também estejam longe de ser inquebráveis. Em termos de segurança, o WPA está bem à frente, mas usá-lo traz problemas de compatibilidade com algumas placas e drivers.

Sempre que possível, use o SSH, SSL ou outro sistema de encriptação na hora de acessar outras máquinas da rede ou baixar seus e-mails. No capítulo sobre acesso remoto, veremos como é possível criar um túnel seguro entre seu micro e o gateway da rede (usando o SSH), permitindo assim encriptar todo o tráfego.

» Próximo: [Quebrando chaves WEP](#)

Para você entender a importância de usar o SSH ou outros protocolos seguros ao usar uma rede wireless, vou falar um pouco mais sobre como quebrar chaves de encriptação, para que você possa entender os ataques usados pelos que estão do outro lado.

Alguns pontos de acesso utilizam versões vulneráveis do WEP, que são muito rápidas de quebrar (em muitos casos você pode corrigir através de uma atualização de firmware) mas, mesmo as versões "não vulneráveis" do WEP podem ser quebradas via força bruta, sempre que seja possível capturar um volume suficiente de tráfego da rede.

Você pode simular uma invasão na sua própria rede, para verificar qual seria o volume de trabalho necessário para invadi-la. Para isso, você vai precisar de pelo menos dois micros ou notebooks. Um deles vai ser usado como um cliente de rede normal e pode usar qualquer placa de rede, enquanto o segundo (que usaremos para simular o ataque) precisa ter uma placa compatível com o Kismet.

Configure o seu ponto de acesso ativando o WEP e desativando o Broadcast do SSID. Ou seja, faça uma configuração relativamente segura, mas depois faça de conta que esqueceu tudo :).

Comece abrindo o Kismet no notebook "invasor". Deixe que ele detecte as redes próximas, pressione "s" para ajustar a ordem dos nomes na lista, selecione sua rede e pressione "**shift + L**" para que ele trave a varredura na sua rede e deixe de bisbilhotar nas redes dos vizinhos.

Inicialmente, sua rede será detectada como "no ssid", já que o broadcast do SSID foi desativado no ponto de acesso. Mas, assim que qualquer micro se conecta ao ponto de acesso, o Kismet descobre o SSID correto. Pressione "**i**" para ver os detalhes da rede e anote o endereço MAC do ponto de acesso (BSSID), que precisaremos para iniciar o passo seguinte.

```

Network List - (Channel)
Name          T W Ch Packts Flags IP Range      Info      Networks
+ Network Details
  Name        : minharede
  SSID       : minharede
    SSID Cloaking on/Closed Network
  Server     : localhost:2501
  BSSID      : 00:50:50:81:81:01
  Manuf      : Unknown
  Max Rate   : 11.0
  BSS Timet  : bad001c0
  First      : Mon Apr 24 11:38:29 2006
  Latest     : Mon Apr 24 11:52:36 2006
  Clients    : 4
  Type       : Access Point (infrastructure)
  Info       :
  Channel   : 11
  Privacy   : Yes
  Encrypt   : WEP
  Decryptd  : No
  Beacon    : 4 (0.004096 sec)
  Packets   : 1104
  Data      : 92

```

Battery: AC 100% 65% (+) Down

Agora que já sabemos o SSID e o MAC do ponto de acesso, falta quebrar o WEP. Para isso precisaremos do **Aircrack**, uma suíte de aplicativos para verificação de redes wireless, que pode ser encontrada no <http://freshmeat.net/projects/aircrack/>. Nos derivados do Debian, ele pode ser instalado via apt-get:

```
# apt-get install aircrack
```

Outra opção é baixar o Back Track, um Live-CD baseado no Slax, que já vem com o Aircrack, Kismet e outras ferramentas úteis pré-instaladas.

O primeiro passo é capturar pacotes da rede usando o **airodump** (que faz parte da suíte aircrack). A sintaxe do comando é "airodump interface arquivo-de-log mac-do-ponto-de-acesso", como em:

```
# airodump ath0 logrede 00:50:50:81:41:56
```

Você pode também indicar um canal (neste caso, ele escuta todas as redes que estejam transmitindo no canal indicado), como em:

```
# airodump ath0 logrede 14
```

Isto gerará o arquivo "logrede.cap", que contém um dump de todos os pacotes capturados. Neste ponto você precisa esperar algum tempo até conseguir um volume razoável de pacotes. Para acelerar isso, faça com que o micro isca baixe alguns arquivos grandes a partir de outro micro da rede.

Abra outro terminal e use o **aircrack** para tentar quebrar a chave de encriptação. Você pode fazer isso sem interromper a captura do airodump, daí a idéia de usar dois terminais separados.

Ao usar o aircrack, é preciso especificar o comprimento da chave WEP (64 ou 128 bits) e o arquivo gerado pelo airodump, como em:

```
# aircrack-n 64 logrede.cap  
ou:  
# aircrack -n 128 logrede.cap
```

Caso o arquivo contenha pacotes destinados a mais de um ponto de acesso, ele pergunta qual verificar. No caso, indique sua rede.

O aircrack usa um ataque de força bruta para tentar descobrir a chave de encriptação da rede. A base do ataque são os IV's (vetores de inicialização), a parte de 24 bits da chave de encriptação, que é trocada periodicamente. O volume de IV's gerados varia de acordo com a rede (por isso existem redes mais vulneráveis que outras) mas, em teoria, é possível quebrar a encriptação de uma rede de 128 bits caso você consiga capturar de 500 mil a um milhão de IV's, enquanto uma chave de 64 bits pode ser quebrada com pouco mais de 200 mil. Caso seja usada uma chave fácil de adivinhar, os números são drasticamente reduzidos, permitindo em muitos casos que a chave seja quebrada com a captura de alguns poucos milhares de IV's.

Como todo processo de força bruta, o tempo necessário é aleatório. O aircrack pode descobrir a chave correta tanto logo no início da captura, quanto só depois de capturar mais de um milhão de IV's. Por isso é interessante deixar o terminal de captura do airodump aberto e ir executando o aircrack periodicamente, até que ele descubra a chave. Quanto maior o volume de dados capturados, maior a possibilidade de descobrir a chave.

Uma forma de aumentar a eficiência do ataque, ou seja, aumentar a chance de descobrir a chave, usando o mesmo número de IV's, é aumentar o "fudge factor", o que faz com que o aircrack teste um número maior de combinações. Isso, naturalmente, aumenta proporcionalmente o tempo necessário para o teste.

O default do aircrack é um fudge factor 2. Você pode alterar o valor usando a opção "-f", como em:

```
# aircrack -f 4 -n 128 logrede.cap
```

É comum começar fazendo um teste com o valor default, depois com fudge 4 (como no exemplo) e a partir daí ir dobrando até descobrir a chave, ou a demora se tornar inviável.

Com um grande volume de IV's, uma chave WEP de 64 bits é um alvo fácil. Neste caso a quebra demorou apenas 21 segundos:

KB	depth	byte(vote)
0	0/ 7	AB(51) 3F(15) A1(15) 0B(13) 96(12) D5(12)
1	1/ 6	AB(30) 59(26) 3D(13) 4C(13) 51(12) 92(7)
2	0/ 7	A1(27) 16(15) B7(13) 14(5) 63(5) CA(5)

KEY FOUND! [AB:AB:A1:23:45]

Como é necessário capturar um grande volume de dados e muitas redes são usadas apenas para acessar a Internet e outras tarefas leves, capturar o volume de pacotes necessário poderia demorar dias.

Um invasor com um nível mediano de conhecimento, provavelmente não se contentaria em esperar tanto tempo. Ao invés disso, ele poderia usar um ataque de flood para induzir tráfego na sua rede, de forma a acelerar o processo, transformando os muitos dias em apenas alguns minutos.

Um exemplo de ferramenta usada para este tipo de ataque é o **aireplay**, mais um integrante da equipe do aircrack. O comando abaixo lança um "chopchop attack" (o tipo de ataque mais eficiente para quebrar chaves WEP) contra o ponto de acesso referente ao endereço MAC especificado, através da interface ath0:

```
# aireplay -b 00:50:50:81:81:01 -x 512 ath0 -4
```

Neste ataque, o aireplay captura um weak packet emitido por algum dos outros micros conectados ao ponto de acesso e o repete indefinidamente, obrigando o ponto de acesso a responder e assim aumentar rapidamente a contagem de IV's, permitindo quebrar a chave WEP muito mais rapidamente. Este é o tipo de ataque mais efetivo, pois derruba a última grande barreira contra a quebra do WEP, que era justamente a demora em capturar um grande volume de pacotes.

O "-x 512" especifica o número de pacotes que serão enviados por segundo. Aumentar o número permite quebrar a chave mais rapidamente, mas, por outro lado, vai reduzir o desempenho da rede, o que pode levar o administrador a perceber o ataque (a menos que ele seja feito em um momento de ociosidade da rede).

Como pode ver, o WEP dificulta o acesso à rede, mas quebrá-lo é apenas questão de tempo. Para melhorar a segurança da sua rede, o ideal é combinar várias camadas de segurança e monitorar os acessos, fazendo com que o tempo e trabalho necessário para invadir a rede seja maior (o que vai afastar os curiosos e invasores casuais) e vai lhe dar tempo para detectar e investigar casos mais graves.

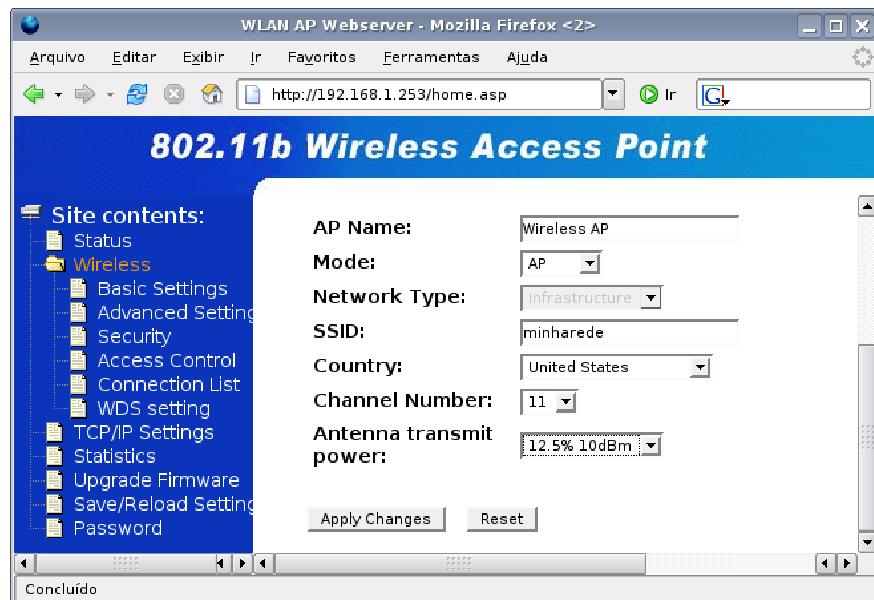
Ao usar o WEP, o ideal é trocar a chave de encriptação regularmente, de forma que, mesmo que alguém consiga descobrir a chave, não consiga usar a rede por muito tempo antes que ela seja trocada. Se possível, utilize o WPA, que, apesar dos problemas de compatibilidade, é muito mais seguro.

O WPA também usa 128 bits de encriptação. A principal diferença é que ele usa vetores de inicialização de 48 bits (o WEP usa vetores de 24 bits), juntamente com um conjunto de proteções contra possíveis ataques. É possível descobrir a chave de encriptação em uma rede WPA-PSK via força bruta, depois de capturar uma pequena quantidade de dados da rede, mas isso só é viável para chaves com poucos caracteres, ou que são baseadas em palavras encontradas no dicionário.

Por exemplo, uma chave como "supercampeao" pode ser descoberta rapidamente, enquanto outra baseada em caracteres aleatórios, como "ErGCDtv8i9S3" demoraria muito tempo para ser quebrada, fazendo com que o invasor desistisse. A dica geral ao usar WPA é escolher uma senha de pelo menos 12 caracteres, misturando letras, números, e caracteres maiúsculos e minúsculos.

Caso a rede seja usada apenas dentro de um pequeno espaço, como uma única sala, apartamento ou escritório, você pode também reduzir a potência do transmissor no ponto de acesso, o que acaba sendo uma medida muito efetiva, pois realmente impede que o sinal da rede seja captado de longe.

Procure pela opção "Antenna transmit power" ou similar (dentro da configuração do ponto de acesso) e veja qual é o menor valor com que a rede funciona corretamente dentro da área necessária:



Outra dica que dificulta, é habilitar a restrição de acesso à rede com base no endereço MAC, geralmente disponível através da opção "Access Control" do ponto de acesso. Ao

ativar esta opção, você cria uma lista com os endereços MAC das placas autorizadas e o ponto de acesso restringe o acesso de qualquer outra.

Programas como o airodump e o próprio Kismet permitem descobrir o endereço MAC dos micros que estão acessando determinada rede muito facilmente, e o endereço MAC da placa de rede pode ser forjado (no Linux, por exemplo, você pode falsear usando o comando "ifconfig wlan0 hw ether 00:11:D8:76:59:2E", onde você substitui o "wlan0" pela interface e o "00:11:D8:76:59:2E" pelo endereço MAC desejado). A questão é que, ao forjar o endereço, o invasor vai derrubar o micro com a placa que foi clonada, de forma que você perceba que algo está errado.

O próximo passo seria isolar sua rede wireless do restante da rede, fazendo com que o invasor possa acessar a Internet, mas não tenha como acessar compartilhamentos e outros recursos da rede.

O mais simples neste caso é instalar uma placa de rede adicional no servidor da rede (ou em qualquer outro micro na ausência dele), onde é conectado o ponto de acesso. Compartilhe a conexão com a placa do AP, mas utilize duas faixas de IP's separados, com um firewall ativo, configurado para bloquear tentativas de conexão provenientes dos micros dentro da rede wireless.

» Próximo: [Usando o HFNetChk](#)

Embora muitos torçam o nariz para os Service Packs e outras atualizações, que por vezes causam problemas diversos ao serem instalados, no mundo Windows as atualizações de segurança para as máquinas Windows da rede são um remédio inevitável. Uma instalação limpa do Windows XP, sem atualizações, conectada à Internet é infectada em questão de minutos, primeiro por worms diversos e depois por crackers.

Mesmo dentro da rede local, estas máquinas sem atualizações são um risco, pois uma máquina infectada indiretamente (infectada ao acessar uma página maliciosa através do IE, ou ao executar um arquivo recebido por e-mail, por exemplo) pode infectar rapidamente todas as demais, fazendo com que você perca o dia atualizando os antivírus e reinstalando o sistema.

Apesar disso, manter todas as máquinas da rede atualizadas pode não ser uma tarefa simples. O Windows não dispõe de uma interface robusta de linha de comando, ou outra ferramenta que permita automatizar tarefas de forma eficiente. Devido a isso, mesmo os administradores mais experientes acabam perdendo suas tardes atualizando cada máquina individualmente. Outro problema que surge com o tempo é saber quais atualizações cada máquina já recebeu, a fim de não precisar ficar repetindo o trabalho.

Uma ferramenta que pode ajudar no segundo problema é o HFNetChk. Para baixa-lo, faça uma busca por "HFNetChk" no <http://download.microsoft.com/>. Você baixará o arquivo "mbsasetup.msi".

O mais interessante nesta ferramenta é que ele não se limita a escanear apenas a sua máquina local, ele pode escanear toda a rede local, dando o status de atualização de todas as máquinas Windows na rede. Este report remoto funciona apenas em máquinas que não estão protegidas por firewall. O report gerado por ele inclui também informações sobre os compartilhamentos e serviços ativos em cada máquina, o que o torna útil também como uma ferramenta preventiva.

A função dele é apenas avisar das atualizações que precisam ser aplicadas em cada máquina, as atualizações propriamente ditas precisam ser aplicadas manualmente por você. Para funcionar, ele precisa que você tenha privilégios de administrador, tanto localmente, quanto nas outras máquinas da rede que forem ser escaneadas.

View security report

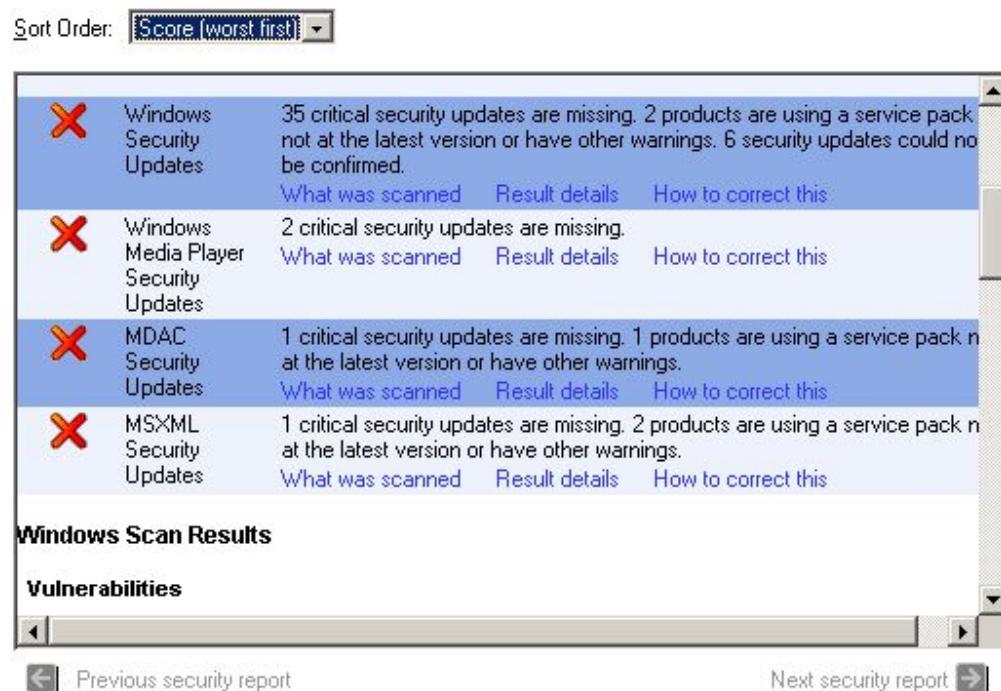
Sort Order: [Score \(worst first\)](#) ▾

	Windows Security Updates	35 critical security updates are missing. 2 products are using a service pack not at the latest version or have other warnings. 6 security updates could not be confirmed.	What was scanned	Result details	How to correct this
	Windows Media Player Security Updates	2 critical security updates are missing.	What was scanned	Result details	How to correct this
	MDAC Security Updates	1 critical security updates are missing. 1 products are using a service pack not at the latest version or have other warnings.	What was scanned	Result details	How to correct this
	MSXML Security Updates	1 critical security updates are missing. 2 products are using a service pack not at the latest version or have other warnings.	What was scanned	Result details	How to correct this

Windows Scan Results

Vulnerabilities

◀ [Previous security report](#) [Next security report](#) ▶



Note que na terceira coluna de cada report consta um link para o artigo correspondente dentro da Knowledge Base da Microsoft, que permite ver mais detalhes sobre o problema.

» Próximo: [Capítulo 5: Compartilhando a conexão](#)

Um dos usos mais comuns e mais simples para um servidor Linux é simplesmente compartilhar a conexão com a internet. A vantagem de usar um servidor dedicado ao invés de simplesmente compartilhar usando o próprio modem ADSL é que você pode incluir outros serviços, como um cache de páginas (Squid), filtro de conteúdo (DansGuardian), firewall, servidor Samba (compartilhando arquivos com a rede interna), servidor de impressão (usando o próprio Cups) e assim por diante.

Em uma rede pequena ou média, com de 10 a 50 micros, é possível usar um único servidor, de configuração razoável para todas estas funções. Em redes maiores, com 100 micros ou mais, isso passa a depender muito do nível de utilização do servidor.

Por exemplo, um simples Pentium 100, com 32 MB de RAM pode compartilhar a conexão com um link de 1 a 8 megabits para um número indefinido de clientes. O mesmo servidor pode compartilhar uma impressora e compartilhar arquivos (serviços mais pesados que simplesmente compartilhar a conexão), desde que estes serviços não sejam utilizados de forma intensiva. Porém, uma configuração modesta como esta já não é adequada para rodar um servidor proxy para uma rede de 50 micros, por exemplo.

Uma máquina mais atual, um Sempron 3000+ com 512 MB de RAM e um HD de 7200 RPM, já pode rodar o mesmo proxy para 100 ou 200 micros com folga. Adicione 2 GB de RAM e ele poderá rodar também um servidor de arquivos para os mesmos 200 micros.

» Próximo: [Uma revisão sobre gerenciamento de pacotes e serviços](#)

Graças aos sistemas de gerenciamento de pacotes usados nas distribuições, instalar e remover cada servidor, assim como inicializar e reinicializar cada um, ativando mudanças na configuração, tornou-se uma tarefa bastante simples. Isso permite que você se concentre na configuração propriamente dita, deixando as tarefas mecânicas a cargo dos mantenedores.

Mas, para isso, é necessário ter uma boa noção de como usar as ferramentas de gerenciamento de pacotes usadas em cada distribuição. Se você já é um usuário antigo, já deve estar careca de saber como fazer tudo isso e pode pular para o próximo tópico. Caso contrário, continue lendo ;).

As distribuições derivadas do **Debian**, incluindo o Ubuntu, Kurumin, Knoppix e muitos outros, utilizam o apt-get como gerenciador de pacotes. O uso do apt-get é bastante simples. Ele trabalha baixando pacotes a partir dos repositórios oficiais do Debian, de forma que você obtém sempre a versão mais atualizada. Para que o sistema funcione, é preciso baixar periodicamente uma lista com os pacotes disponíveis em cada servidor, permitindo que o apt-get mantenha seu banco de dados local. Isso é feito usando o comando:

```
# apt-get update
```

Ele deve ser executado regularmente, de preferência antes de fazer cada instalação, já que num servidor é crítico trabalhar com as versões mais atuais dos pacotes, devido à questão das atualizações de segurança. Para instalar qualquer pacote pelo apt-get, use o comando "apt-get install", como em:

```
# apt-get install apache2
```

O apt-get instala automaticamente todas dependências do pacote, pedindo sua confirmação. Leia tudo antes de confirmar, pois em alguns casos raros o apt-get pode propor soluções um tanto quanto desastrosas para conflitos de pacotes. Pode acontecer dele propor remover todo o KDE para instalar uma biblioteca antiga, que conflite ou substitua algum dos pacotes base, por exemplo. Mesmo em casos extremos como este, você sempre tem a chance de abortar a instalação, basta ficar atento às mensagens ;).

Para remover um pacote, use o comando "apt-get remove", como em:

```
# apt-get remove apache2
```

Para atualizar um pacote, rode o comando "apt-get update" e em seguida use o "apt-get install", como se estivesse instalando-o novamente. O apt-get sabe que o pacote já está instalado, por isso se limita a atualizá-lo, mantendo as configurações. É importante atualizar periodicamente os pacotes relacionados a servidores, por causa das atualizações de segurança.

É possível também atualizar de uma vez todos os pacotes do sistema. Isso só é recomendável se você está utilizando uma distribuição baseada em uma versão estável do Debian. Por exemplo, o Kurumin 5.0 é baseado no Sarge, que foi finalizado em 2005, enquanto o Kurumin 6.2 será baseado no Etch, cujo lançamento está planejado para o final de 2006.

Para isso, use o comando:

```
# apt-get upgrade
```

Muitas distribuições (como o Kanotix e o Knoppix) são baseadas no Debian Sid (a versão instável) e a maioria das versões do Kurumin são baseadas no Debian testing. Nestes casos, você precisa tomar bastante cuidado ao rodar o "apt-get upgrade", pois as atualizações viram um alvo em movimento.

Para instalar um pacote .deb que você baixou manualmente, use o comando "dpkg -i", como em:

```
# dpkg -i sarg_2.0.5.dfsg-1_i386.deb
```

Os pacotes instalados manualmente através do dpkg não passam pela verificação de dependências do apt-get, abrindo margem para problemas diversos. Para que o apt-get verifique a instalação e coloque ordem na casa, é recomendável rodar logo em seguida o comando:

```
# apt-get -f install
```

O apt-get trabalha com dois arquivos de configuração. O principal é o arquivo "**/etc/apt/sources.list**", onde vai a lista dos servidores utilizados. Imagine que na sua máquina o arquivo contém a linha:

```
deb http://ftp.br.debian.org/debian etch main contrib non-free
```

Por algum motivo, o servidor está inacessível no momento, o que faz com que você receba mensagens de erro ao rodar o "apt-get update" ou ao instalar pacotes. Para resolver o problema, você precisaria apenas alterar o arquivo, especificando outro servidor, como em:

```
deb http://ftp.us.debian.org/debian etch main contrib non-free
```

Aqui substituímos o "br" por "us", fazendo com que passe a ser utilizado o mirror dos EUA. Os servidores disponíveis são identificados pelos códigos de país, como "de", "uk", "es", "it", etc.

No caso do **Ubuntu**, existe uma pegadinha. A maioria dos pacotes, incluindo quase todos os pacotes para servidores estão concentrados no "**Universe**", um repositório extra que vem desabilitado por padrão. Antes de mais nada, você precisa editar o arquivo "**/etc/apt/sources.list**", descomentando a linha referente ao Universe, e rodar o "apt-get update".

No **Fedora** é usado por padrão o "**yum**", que funciona de forma bem similar ao apt-get, baixando os pacotes da Internet, junto com as dependências. Existem muitas diferenças entre o Fedora e o Debian, uma delas é o formato dos pacotes utilizados: o Fedora utiliza pacotes .rpm, enquanto o debian utiliza pacotes .deb.

Ambos também utilizam repositórios separados, com pacotes construídos especificamente para cada uma das duas distribuições, de forma que existem algumas diferenças nos nomes dos pacotes e arquivos de configuração usados, que aponto ao longo do livro. Voltando ao uso do yum, comece baixando as listas dos pacotes disponíveis, usando o comando:

```
# yum check-update
```

Ele funciona de forma análoga ao "apt-get update", com a diferença que de a lista dos servidores usados não vai no arquivo sources.list, mas é dividida em diversos arquivos (um para cada servidor usado), organizados na pasta "**/etc/yum.repos.d/**".

Para instalar um pacote, use o comando "yum install", como em:

```
# yum install wpa_supplicant
```

Para removê-lo posteriormente, use:

```
# yum remove wpa_supplicant
```

O yum possui também um recurso de busca, que é bastante útil quando você está procurando por um pacote, mas não sabe o nome exato, ou em casos de pacotes que possuem nomes diferentes em relação a outras distribuições. Use o comando "yum search", seguido por alguma palavra ou expressão, que faça parte do nome do pacote ou descrição, como em:

```
# yum search apache
```

Ele retorna um relatório contendo todos os pacotes relacionados, incluindo o texto de descrição de cada um. Para atualizar um pacote, use o comando "yum update", como em:

```
# yum update wpa_supplicant
```

Se usado sem especificar um pacote, o "update" vai atualizar de uma vez só todos os pacotes do sistema, de forma similar ao "apt-get upgrade" do Debian:

```
# yum update
```

Existe ainda o comando "yum upgrade", que é um pouco mais incisivo, incluindo também pacotes marcados como obsoletos (que não existem mais na versão atual). Ele é útil em casos em que é necessário atualizar uma versão antiga do Fedora:

```
# yum upgrade
```

O yum é utilizado também em distribuições derivadas do Fedora, ou do Red Hat Enterprise, como o CentOS. Embora também seja derivado do Red Hat, o **Mandriva** utiliza um gerenciador próprio, o "**urpmi**", que é também bastante simples de usar. Para instalar um pacote, você usa o comando "urpmi pacote", como em:

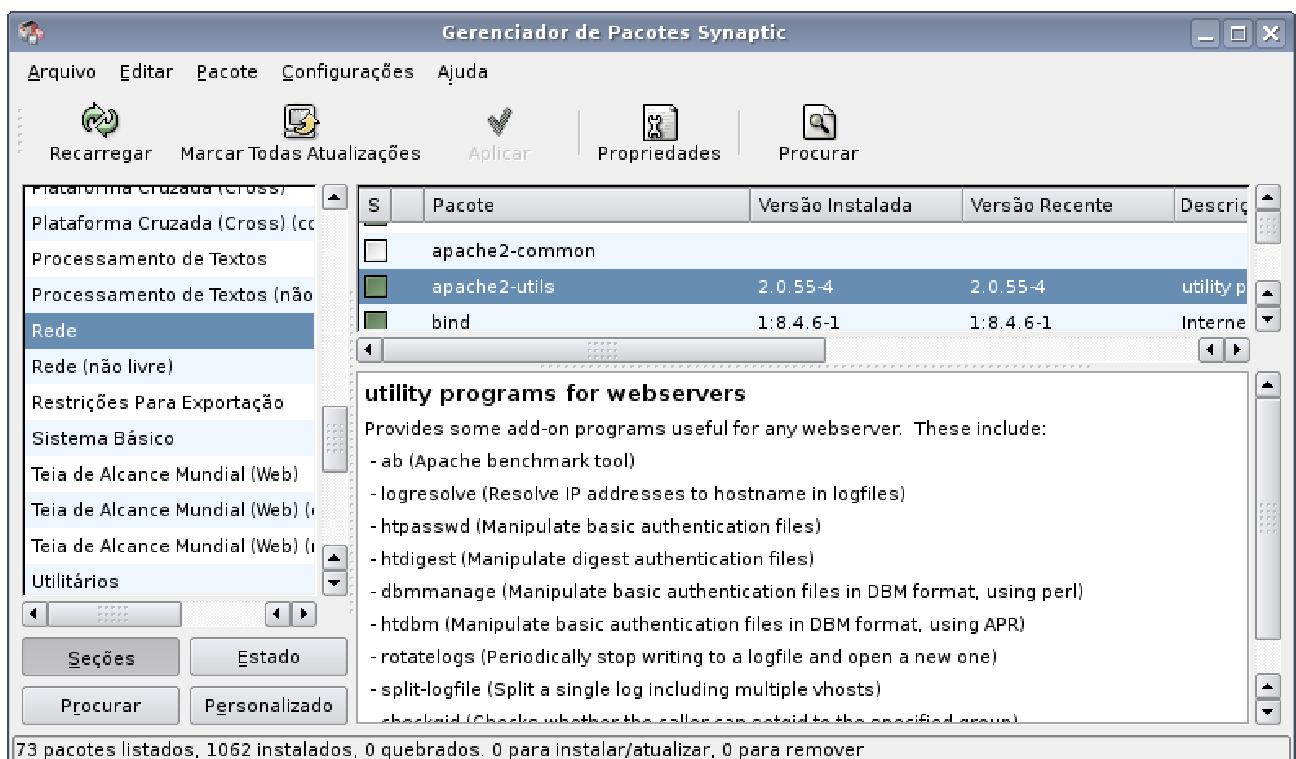
```
# urpmi samba
```

Para removê-lo, use o comando "urpme", como em:

```
# urpme samba
```

Um dos pontos fortes do Mandriva é que todo o gerenciamento de pacotes pode ser feito usando as opções dentro do Mandriva Control Center, o que permite dispensar a linha de comando para a maioria das operações.

Nas distribuições derivadas do Debian, você tem algo similar usando o **Synaptic**, que (quando não vem instalado por padrão) pode ser instalado via apt-get:



Para instalar um pacote baixado manualmente, é usado (tanto no Fedora, quanto no Mandriva) o comando "rpm -U", que tem uma função análoga ao "dpkg -i" do Debian. Para usá-lo, você deve acessar a pasta onde foi baixado o arquivo, ou incluir o caminho completo até ele (lembre-se de que você pode usar a tecla Tab para completar o nome do arquivo):

```
# rpm -U webmin-1.270-1.noarch.rpm
```

Se quiser ver uma barra de progresso e detalhes da instalação, adicione os parâmetros "vh", como em:

```
# rpm -Uvh webmin-1.270-1.noarch.rpm
```

Temos, em seguida, a questão dos **serviços**:

Servidores como o Apache também são chamados de "daemons" ou "serviços de sistema", tradução do termo "system services". Ambos os termos indicam um programa que fica residente, respondendo a requisições de outros micros ou executando tarefas de forma automatizada. É o caso não apenas de servidores como o Squid, Samba, Postfix e tantos outros, mas também de programas, como o Cron, e de ferramentas de detecção de hardware, como o Udev e o Hotplug.

A tarefa de iniciar, parar e reiniciar estes serviços é automatizada por um conjunto de scripts localizados na pasta "/etc/init.d" (ou "/etc/rc.d/init.d", dependendo da distribuição usada). Nas distribuições derivadas do **Debian**, você inicia um serviço usando o comando "/etc/init.d/nome start", como em:

```
# /etc/init.d/apache2 start
```

Para parar, é usado o mesmo comando, com o parâmetro "stop":

```
# /etc/init.d/apache2 stop
```

Muitos serviços suportam também os parâmetros "**restart**" e "**reload**" (ou "**force-reload**"), usados para ativar mudanças na configuração. A diferença entre os três é que o "restart" é simplesmente uma combinação do "stop" e "start", enquanto o ""reload ou "force-reload" (quando disponíveis) fazem com que o serviço releia o arquivo de configuração, porém sem interromper sua atividade. Isso é importante no caso de um servidor web com muitos acessos, por exemplo.

Veremos os nomes de cada serviço ao longo do livro. Por enquanto, preocupe-se apenas com os comandos de gerenciamento.

Temos ainda o comando "**update-rc.d**", que permite definir se um serviço vai ser ativado ou não durante o boot. Imagine o caso de um servidor onde você utiliza um servidor FTP para disponibilizar arquivos esporadicamente, ativando-o e desativando-o manualmente. Ao instalar o Proftpd (o servidor FTP que aprenderemos a configurar mais adiante) ele é configurado para subir automaticamente durante o boot. Para desativar a inicialização automática, você usaria o comando:

```
# update-rc.d -f proftpd remove
```

Se mudar de idéia e quiser que ele volte a ser inicializado durante o boot, use:

```
# update rc.d -f proftpd defaults
```

No caso das distribuições derivadas do **Fedora** (e também no Mandriva), usamos os comandos "**service**" e "**chkconfig**". O primeiro permite ativar, desativar e reiniciar um serviço, como em:

#	service	sshd	start
#	service	ssh	stop
# service ssh restart			

... enquanto o segundo permite definir se ele vai ser inicializado ou não durante o boot, de forma análoga ao "update-rc.d" do Debian. Para fazer com que o SSH seja ativado durante o boot, você usaria:

```
# chkconfig sshd on
```

E para desativá-lo, usaria:

```
# chkconfig sshd off
```

A maioria das distribuições se enquadra em um desses dois sistemas. A principal exceção é o **Slackware**. Ele usa uma estrutura simplificada, na qual, para ativar ou desativar a

inicialização de um serviço, você simplesmente adiciona ou retira a permissão de execução do script correspondente, dentro da pasta "**/etc/rc.d**".

Para ativar o SSH você usaria:

```
# chmod +x /etc/rc.d/rc.ssh
```

E para desativá-lo, usaria:

```
# chmod -x /etc/rc.d/rc.ssh
```

Para iniciar ou parar o serviço, você executa o mesmo script, usando os parâmetros "start" ou "stop", como em:

```
# /etc/rc.d/rc.ssh start
```

```
# /etc/rc.d/rc.ssh stop
```

» Próximo: [Compartilhando](#)

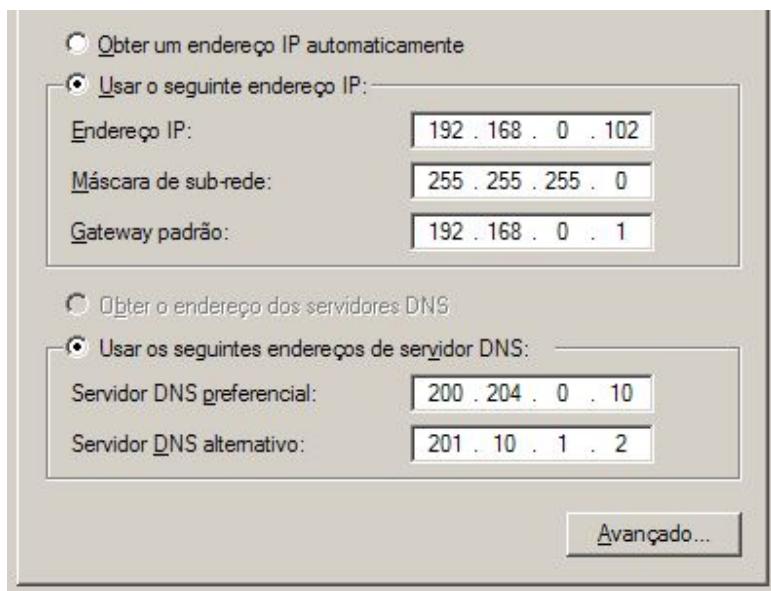
Do ponto de vista da segurança e até mesmo da facilidade de configuração, é sempre recomendável usar um servidor com duas placas de rede, separando o tráfego proveniente da internet do tráfego da rede local. Com duas placas separadas, fica mais fácil criar as regras de firewall adequadas para bloquear acessos provenientes da internet e, ao mesmo tempo, permitir o tráfego vindo da rede local.

Se você acessa via ADSL, é recomendável manter o modem configurado como bridge ao invés de configurá-lo como roteador. Dessa forma, o servidor recebe todas as portas de entrada, permitindo que você acesse o servidor remotamente via SSH (muito útil para prestar suporte remoto em servidores instalados por você) ou disponibilize um servidor web ou FTP.

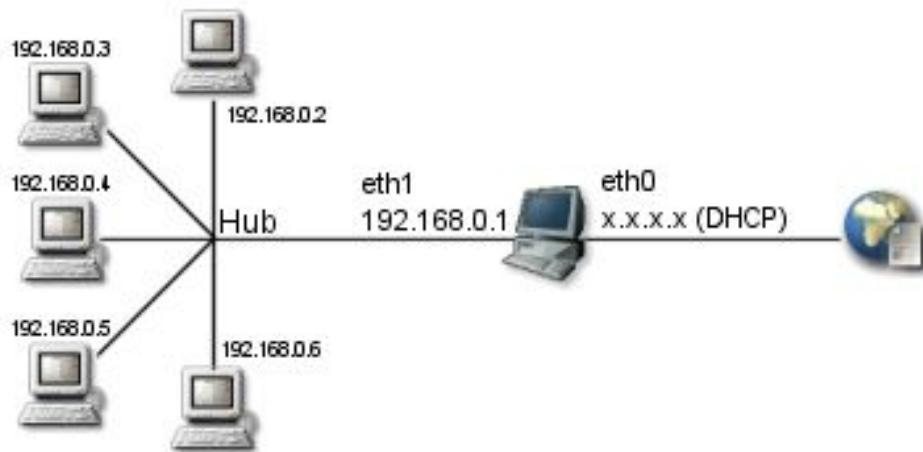
Embora acabe sendo mais trabalhoso, nada impede que você configure o modem como roteador e use o servidor para novamente compartilhar a conexão recebida do modem, acrescentando os demais serviços. Nestes casos, você vai precisar configurar o modem para direcionar ao servidor as portas que devem ficar abertas para a internet, como a porta 22, usada pelo SSH (caso você pretenda administrar o servidor remotamente), por exemplo, o que é configurado na interface de administração do modem.

Comece configurando a rede local, usando uma das faixas de endereços IP reservadas a redes locais, como, por exemplo, 192.168.0.x, onde o servidor fica com o IP 192.168.0.1 e os micros da rede interna recebem endereços IP seqüenciais, de 192.168.0.2 em diante. Não se esqueça de colocar o endereço IP do servidor (192.168.0.1 no exemplo) como gateway padrão na configuração dos clientes. A configuração de gateway padrão no servidor fica em

branco, pois o gateway padrão do servidor vai ser definido ao configurar a conexão com a internet.



Verifique se é possível dar ping nos PCs da rede interna (ex: ping 192.168.0.2) a partir do servidor. Se estiver usando Linux nas estações, experimente ativar o servidor SSH em uma das estações e tentar se conectar a ela a partir do servidor.



Da primeira vez que configurar a rede, use endereços IP estáticos para todas as estações, pois assim é mais fácil detectar problemas diversos. Depois de tudo funcionando, mude para configuração via DHCP se preferir. Se alguma das estações estiver inacessível, verifique se não existe um firewall ativo, verifique o cabo, troque a porta usada no hub, etc.

Com a rede local funcionando, o próximo passo é conectar o servidor à internet. É preferível deixar para configurar a placa da internet depois da placa da rede local, pois isso evita alguns erros comuns. Por exemplo, se você configurar a conexão com a web e depois configurar a rede local, colocando um endereço qualquer no campo "default gateway", o

gateway informado na configuração da rede local vai substituir o gateway do provedor (definido ao conectar na internet), fazendo com que a conexão deixe de funcionar.

Rode o comando "**ifconfig**" para verificar qual é a Interface ligada à internet. Lembre-se de que ao conectar via ADSL com autenticação ou acesso discado, a interface **eth0** ou **eth1** é substituída pela interface virtual **ppp0**.

Em seguida, temos os comandos que compartilham a conexão. No Linux, o compartilhamento é feito usando o Iptables, o firewall integrado ao Kernel. Na verdade, o Iptables é expandido através de módulos, por isso suas funções vão muito além das de um firewall tradicional. Para ativar o compartilhamento, são necessários apenas três comandos:

```
# modprobe iptable_nat
# echo 1 > /proc/sys/net/ipv4/ip_forward
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Substitua o "eth0" pela placa da internet. Este comando simplesmente compartilha a conexão proveniente da placa da internet com todas as demais placas de rede espalhadas no servidor, por isso não é necessário especificar a placa de rede local.

O primeiro comando ativa o "**iptable_nat**", o módulo do Iptables responsável por oferecer suporte ao roteamento de pacotes via NAT. O segundo ativa o "**ip_forward**", o módulo responsável pelo encaminhamento de pacotes, utilizado pelo módulo `iptable_nat`.

Finalmente, o terceiro cria uma regra de roteamento, que orienta o servidor a direcionar para a internet todos os pacotes (recebidos dos clientes) que se destinarem a endereços que não façam parte da rede local (ou seja, qualquer coisa fora da faixa 192.168.0.x). A partir daí, o servidor passa a ser o gateway da rede.

Nem todas as distribuições instalam o executável do Iptables por padrão. No **Mandriva**, por exemplo, ele é instalado ao marcar a categoria "firewall" durante a instalação. Para instalá-lo posteriormente, use o comando "**urpmi iptables**".

Em muitas distribuições com o Kernel 2.6, é necessário usar um quarto comando ao compartilhar uma conexão ADSL. Este comando ajusta os tamanhos dos pacotes recebidos do modem ao MTU usado na rede local. Note que, apesar da diagramação do livro quebrar o comando em duas linhas, trata-se de um único comando:

```
# iptables -A FORWARD -p tcp --tcp-flags SYN,RST SYN -m \
tcpmss --mss 1400:1536 -j TCPMSS --clamp-mss-to-pmtu
```

A barra invertida ("\\") faz com que o shell não interprete o caractere seguinte (no caso, a quebra de linha), permitindo quebrar o comando em duas linhas, sem causar um erro. Ele é um truque que permite incluir comandos longos demais para caberem na página, divididos em duas linhas ou mais. Na verdade, o comando forma uma única linha.

Adicione os comandos em um dos scripts de inicialização do sistema, para que eles sejam executados automaticamente durante o boot. No Debian e derivados, coloque-os no final do

arquivo "/etc/init.d/bootmisc.sh". No Fedora, Mandriva e outros derivados do Red Hat, use o arquivo "/etc/rc.d/rc.local".

Esta receita é genérica, deve funcionar em qualquer distribuição. Lembre-se de substituir "eth0" no comando por "**ppp0**", caso você conecte via ADSL com autenticação (PPPoE) e precise usar o pppoeconf ou o adsl-setup para estabelecer a conexão. Em caso de dúvida, cheque qual é a interface usando o comando "ifconfig".

Uma segunda opção, mais elegante, porém mais complicada, é criar um serviço de sistema, que pode ser ativado e desativado. Neste caso, crie o arquivo de texto **"/etc/init.d/compartilhar"**. Dentro dele vão as linhas abaixo. Observe que, novamente, o comando para ajustar o MTU dos pacotes em conexões via ADSL forma uma única linha:

```

#!/bin/bash

iniciar(){
modprobe
echo
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
iptables -A FORWARD -p tcp --tcp-flags SYN,RST -m tcpmss --mss \
1400:1536 -j TCPMSS
}

parar(){
iptables -F -t nat
}

case
"start")
"iniciar
;;
"stop")
"parar
;;
"restart")
*) echo "Use os parâmetros start ou stop"
esac

```

Este é um shell script que aceita três funções: start, stop e restart, executando dentro de cada uma os comandos que compartilham e param o compartilhamento da conexão. Esta estrutura é similar à usada nos demais scripts de inicialização do sistema, como os do Apache, Samba, Squid e outros serviços.

Transforme-o em um arquivo executável, usando o comando:

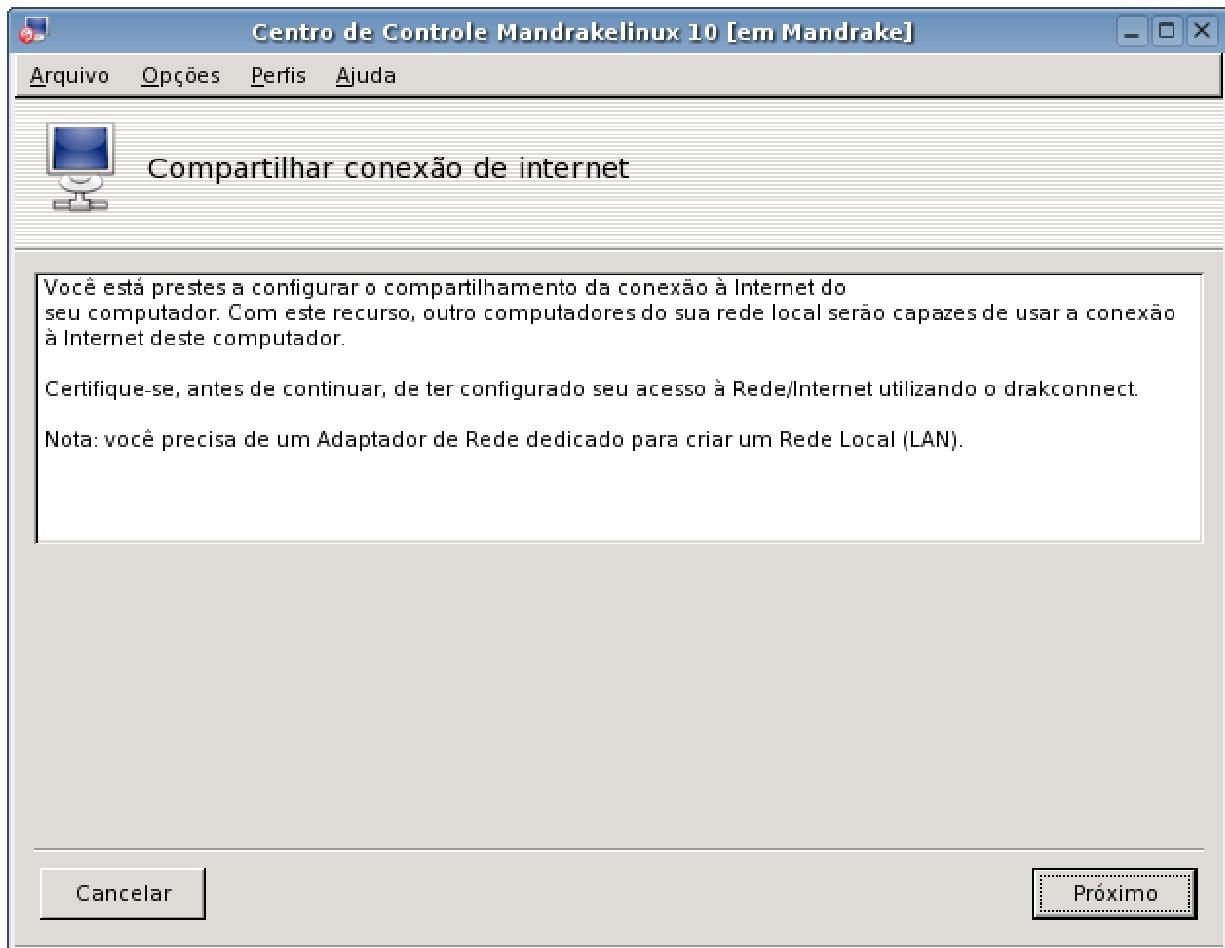
```
# chmod +x /etc/init.d/compartilhar
```

A partir daí, você pode iniciar e parar o compartilhamento usando os comandos:

```
# /etc/init.d/compartilhar start  
# /etc/init.d/compartilhar stop
```

Em muitas distribuições você pode usar o comando "**service**", que simplifica o comando. Neste caso, você poderia digitar apenas "service compartilhar start". Para que a conexão seja ativada durante o boot, adicione o comando "/etc/init.d/compartilhar start" dentro do arquivo "/etc/init.d/bootmisc.sh" ou "/etc/rc.d/rc.local", de acordo com a distribuição usada.

Se preferir, você pode compartilhar usando uma das ferramentas disponíveis nas distribuições. No Kurumin use o "Iniciar > Internet > Compartilhar conexão e firewall > Compartilhar conexão via modem ou ADSL PPPoE". No Mandriva você pode usar o DrakGw, encontrado na seção "Rede e Internet" do Mandriva Control Center.



» Próximo: [Configurando um servidor DHCP](#)

Hoje em dia quase todas as redes utilizam algum tipo de servidor DHCP. Em geral, eles são ativados automaticamente ao compartilhar a conexão ou junto com algum outro serviço, de forma que você acaba não aprendendo muita coisa sobre a sua configuração. De um modo geral, o trabalho de um servidor DHCP é bastante simples. Ele responde aos pacotes de broadcast das estações, enviando um pacote com um dos endereços IP disponíveis e os demais dados da rede. Os pacotes de broadcast são enviados para o último endereço da rede, como em 192.168.1.255 ou 255.255.255.255, e são recebidos por todos os micros da rede.

Periodicamente o servidor DHCP verifica se as estações ainda estão lá, exigindo uma renovação do "aluguel" do endereço IP (opção "lease time"). Isso permite que os endereços IP sejam gastos apenas com quem realmente estiver online, evitando que os endereços disponíveis se esgotem. No Linux o serviço de DHCP é exercido pelo **dhcp3-server**, que nas distribuições baseadas no **Debian** pode ser instalado através do comando:

```
# apt-get install dhcp3-server
```

Os comandos "/etc/init.d/dhcp3-server start" e "/etc/init.d/dhcp3-server stop" gerenciam a atividade do serviço. No **Fedora** o pacote com o servidor dhcp se chama simplesmente "**dhcp**" e pode ser instalado através do yum. Uma vez instalado, use os comandos "service dhcpcd start" e "service dhcpcd stop". No caso do **Mandriva**, o pacote se chama "dhcpcd".

Em qualquer um dos três casos, o arquivo de configuração é o **dhcpcd.conf**. No Debian, o caminho completo para ele é "**/etc/dhcp3/dhcpcd.conf**" e no Mandriva e Fedora é apenas "**/etc/dhcpcd.conf**". Apesar dessas diferenças nos nomes, o que interessa mesmo é a configuração do arquivo e esta sim é igual, independentemente da distribuição.

Este é um exemplo de arquivo de configuração básico:

```
ddns-update-style                                     none;
default-lease-time                                    600;
max-lease-time 7200;

authoritative;

subnet      192.168.0.0          netmask    255.255.255.0  {
range        192.168.0.100        routers
option       routers
option       domain-name-servers 200.177.250.10,200.204.0.10;
option       broadcast-address   192.168.0.255;
}
```

A opção " default-lease-time" controla o tempo de renovação dos endereços IP. O "600" indica que o servidor verifica a cada dez minutos se as estações ainda estão ativas. Se você tiver mais endereços IP do que máquinas, os endereços IP das estações raramente vão precisar mudar. Mas, no caso de uma rede congestionada, o "max-lease-time" determina o tempo máximo que uma estação pode usar um determinado endereço IP. Isso foi planejado para ambientes onde haja escassez de endereços IP, como, por exemplo, em um provedor de acesso, onde sempre existem mais clientes do que endereços IP disponíveis e se trabalha contando que nem todos vão ficar conectados simultaneamente.

Em condições normais, essas duas opções não são muito importantes. O que interessa mesmo é o bloco que vai abaixo, onde ficam as configurações da rede.

A opção "**range**" determina a faixa de endereços IP que será usada pelo servidor. Se você utiliza a faixa de endereços 192.168.0.1 até 192.168.0.254, por exemplo, pode reservar os endereços de 192.168.0.1 a 192.168.0.100 para estações configuradas com IP fixo e usar os demais para o DHCP, ou então reservar uma faixa específica para ele, de 192.168.0.100 a 192.168.0.201, por exemplo. O importante é usar faixas separadas para o DHCP e os micros configurados com IP fixo.

Na "**option routers**" vai o endereço do default gateway da rede, ou seja, o endereço do servidor que está compartilhando a conexão. Não é necessário que o mesmo micro que está compartilhando a conexão rode também o servidor DHCP. Pode ser, por exemplo, que na sua rede o gateway seja o próprio modem ADSL que está compartilhando a conexão e o DHCP seja um dos PCs.

A opção "**option domain-name-servers**" contém os servidores DNS que serão usados pelas estações. Ao usar dois ou mais endereços, eles devem ser separados por vírgula, sem espaços. Em geral você vai usar os próprios endereços DNS do provedor, a menos que você configure um servidor DNS interno na sua rede, que pode ser instalado no próprio micro que está compartilhando a conexão e rodando o DHCP. Estes serviços consomem poucos recursos da máquina.

O servidor DNS mais usado no Linux é o Bind. No Kurumin ou Debian em geral, você mata o coelho com um "**apt-get install bind**". Este servidor DNS pode ser configurado para implementar um sistema de domínios e subdomínios na sua rede, mas o uso mais comum é simplesmente fazer um "cache", onde o servidor DNS simplesmente repassa as requisições para um dos 13 root servers da internet e vai armazenando os endereços que já foram acessados.

Você pode substituir o arquivo de configuração padrão por este modelo, ou editá-lo conforme a necessidade. Ao fazer qualquer alteração no arquivo, você deve reiniciar o servidor DHCP usando o comando:

```
#          /etc/init.d/dhcp3-server      restart  
(ou "service dhcp restart" no Fedora)
```

Sempre que configurar um servidor com duas placas de rede, é importante que o servidor DHCP seja configurado para escutar apenas na placa da rede local. No Debian, esta configuração vai no arquivo "**/etc/default/dhcp3-server**". Procure pela linha:

```
INTERFACES=""
```

... e adicione a placa que o servidor DHCP deve escutar, como em:

```
INTERFACES="eth0"
```

Para que a configuração entre em vigor, basta reiniciar o serviço novamente.

» Próximo: [DHCP com IP fixo](#)

Mais uma opção interessante no servidor DHCP é a possibilidade de relacionar um determinado endereço IP com o endereço MAC de certo micro da rede. Isso faz com que

ele sempre obtenha o mesmo endereço a partir do servidor DHCP, como se tivesse sido configurado para usar IP fixo.

Esse recurso é usado em redes de terminais leves, para que o servidor "reconheça" os terminais e possa enviar a configuração adequada a cada um, mas pode ser usado em outras situações, como, por exemplo, em uma pequena rede, onde alguns micros compartilham impressoras e arquivos e, por isso, não podem ficar mudando de endereço IP a cada reboot.

Configurar o servidor DHCP para dar a eles sempre o mesmo IP pode ser mais prático que configurá-los para usar IP fixo manualmente, pois eles continuarão recebendo o mesmo IP mesmo que você reinstale o sistema (pois, apesar da mudança de sistema operacional, a placa de rede continuará a mesma). Veja o caso de quem usa live-CDs como o Kurumin, por exemplo.

Para usar este recurso, adicione uma seção como esta para cada host, no final do arquivo **dhcpd.conf**, depois de todas as linhas de configuração, mas antes de fechar a chave (>):

```
host                               kurumin          {  
hardware                         ethernet          00:0F:B0:55:EA:13;  
fixed-address                     192.168.0.202;  
}  
}
```

Veja que a seção começa com o nome da máquina, "kurumin" no exemplo. Em seguida vêm, entre chaves, o endereço MAC da placa de rede (que você pode verificar através do comando "ifconfig") e o endereço IP que a estação deve usar. Um exemplo de arquivo completo, incluindo a configuração de IP fixo para duas máquinas seria:

```
ddns-update-style               none;  
default-lease-time              600;  
max-lease-time                  7200;  
authoritative;  
  
subnet    192.168.0.0      netmask   255.255.255.0 {  
range        192.168.0.100           192.168.0.201;  
option       routers             192.168.0.10;  
option       domain-name-servers 200.177.250.10,200.204.0.10;  
option broadcast-address 192.168.0.255;  
  
host                               kurumin          {  
hardware                         ethernet          00:0F:B0:55:EA:13;  
fixed-address                     192.168.0.202;  
}  
  
host                               mandriva          {  
hardware                         ethernet          00:0F:B0:45:BC:17;  
fixed-address                     192.168.0.203;  
}  
}
```

Em situações normais, você nunca deve manter mais de um servidor DHCP ativo ao mesmo tempo, principalmente se ambos estiverem configurados para dar endereços dentro da mesma faixa. Caso contrário, começam a surgir problemas com micros configurados com o mesmo IP (cada um dado por um DHCP diferente) e assim por diante. Mas, em algumas situações, uma configuração com dois servidores DHCP pode funcionar, naturalmente depois de bem testada.

O dhcp3-server usado no Linux é bastante rápido, por isso (desde que a configuração não seja muito complexa) costuma responder antes dos servidores DHCP usados nos servidores Windows e na maioria dos modems ADSL, o que pode ser usado a seu favor.

Imagine um caso comum: uma rede de 10 ou 20 micros, com um ADSL de 1 megabit, compartilhado pelo próprio modem. Para melhorar o desempenho da rede, você resolve implantar um servidor com o Squid configurado para trabalhar como um proxy transparente, além de um servidor DNS próprio e DHCP.

Como este "servidor" é o seu próprio micro, que precisa ser desligado de vez em quando, você decide manter a rede da forma que está, com o modem compartilhando a conexão e o seu micro funcionando como um segundo gateway, dentro da rede local. Você quer que a rede continue funcionando mesmo quando seu micro precisar ser desligado por um certo tempo, por isso mantém o servidor DHCP do modem ativo, junto com o servidor DHCP instalado no seu micro.

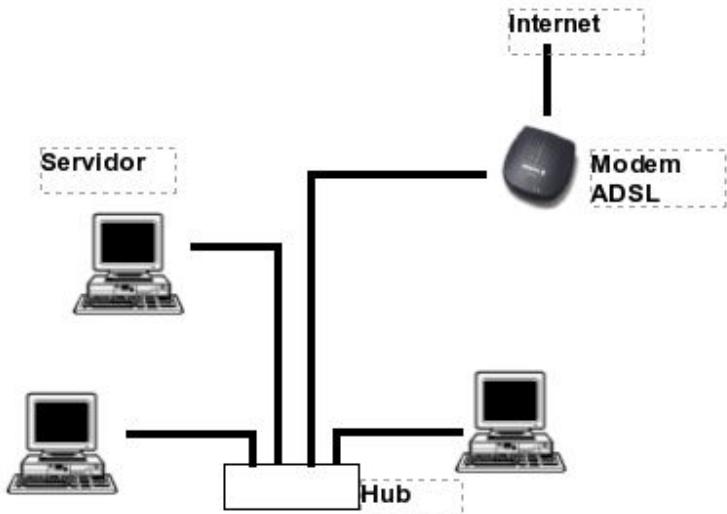
No seu caso, o Bind é mais rápido que o DHCP do modem. Por isso, enquanto ele está ligado, os micros da rede local são configurados para acessar através dele, passando pelo proxy transparente. Quando ele é desligado, o modem ADSL passa a responder as chamadas e os micros passam a ser configurados para acessar diretamente através dele (é preciso reconfigurar os clientes via DHCP para que eles obtenham a configuração a partir do modem e passem a utilizá-lo como gateway). A rede continua funcionando mesmo que seu micro seja desconectado definitivamente.

Note que isso é tecnicamente errado e só funciona em redes pequenas, onde todos os micros são ligados ao mesmo hub ou switch. Quanto maior a rede, mais imprevisível se torna o comportamento dos servidores DHCP e mais importante torna-se manter apenas um ativo.

» Próximo: [Compartilhar a conexão usando uma única placa de rede](#)

Se você está usando um notebook ou barebone, com uma placa onboard e sem slots de expansão, existe a possibilidade de compartilhar a conexão usando uma única placa de rede, utilizando uma placa de rede virtual.

Normalmente, a topologia para compartilhar a conexão é ligar o modem ADSL/Cabo na placa eth0 do servidor, conectar a placa eth1 do mesmo servidor no hub/switch, juntamente com as demais estações. Ao compartilhar usando uma única placa, todo mundo passa a ser conectado diretamente ao hub/switch, inclusive o modem. O servidor é configurado para ter duas placas de rede "lógicas", uma para se conectar na internet e outra para a rede local.



Uma dica é que os modems ADSL geralmente utilizam um cabo de rede cross-over, já que são feitos para serem conectados diretamente a um PC e não ao hub. Nestes casos, você precisa ligar o modem na porta up-link do hub. Isso normalmente não é necessário ao usar um switch, pois eles são capazes de detectar o cabo cruzado e corrigir o sinal via software.

O primeiro passo é se conectar normalmente à internet no servidor, usando as configurações de sempre. A partir do momento em que ele estiver acessando, crie o alias para a placa de rede "lógica" que o conectará aos micros da rede local, usando o comando:

```
# ifconfig eth0:1 192.168.0.1/24
```

Isso fará com que o servidor passe a se comportar como se tivesse duas placas de rede, uma ligada ao modem ADSL e outra ligada à rede local, respondendo no endereço 192.168.0.1 (você pode trocar por outro se preferir). O "/24" indica a configuração da máscara de sub-rede, equivale a digitar "255.255.255.0". Para que isso funcione no Debian e algumas distribuições derivadas dele, é preciso instalar o pacote "net-tools" (apt-get install net-tools).

Compartilhe a conexão da forma usual, configure os clientes da rede e eles já serão capazes de navegar. Lembre-se de que um alias para a placa de rede não é o mesmo que uma placa de rede física espetada na placa-mãe. Por isso, o utilitário para compartilhar a conexão incluído na sua distribuição pode ter problemas para trabalhar desta forma. Se por acaso ele falhar, use os quatro comandos para compartilhar diretamente através do Iptables que já vimos.

Compartilhar a conexão usando uma única placa de rede relaxa um pouco a segurança da rede. Embora o modem ADSL fique conectado diretamente ao hub, ninguém na internet será capaz de enxergar os micros da rede local, pois eles utilizarão uma faixa de IPs inválida, como 192.168.0.x ou 10.0.0.x. Você ainda pode adicionar um firewall "fecha tudo" no servidor, para que ele não responda a pings, feche todas as portas, etc.

O problema é que com o modem ADSL ligado diretamente ao hub, alguém que consiga obter acesso à configuração do modem poderia ganhar acesso aos micros da rede local através dele. Os modems ADSL não são apenas dispositivos burros que fazem a conversão analógico/digital, eles possuem vários recursos para rotear pacotes, criar vários tipos de filtros e, em muitos casos, até túneis VPN.

As empresas de telefonia e provedores geralmente protegem as configurações do modem com uma senha para que o usuário não possa ficar brincando com elas, mas em geral usam a mesma senha em milhares de modems. Em alguns casos, o modem vem aberto para aceitar conexões da web, protegido apenas pela senha, sem falar que por terem tantos recursos sempre existe a possibilidade de surgirem bugs diversos de segurança. Pense no modem ADSL como um PC vulnerável, que nunca recebe atualizações de segurança.

Se alguém consegue obter acesso à configuração do modem, pode ganhar acesso aos micros da rede local que estarão conectados diretamente a ele. Este é o grande problema. Usando duas placas de rede ainda seria preciso passar pelo servidor de compartilhamento, que pode ser protegido com um bom firewall. Ao conectar o modem diretamente ao hub, esta linha de proteção é perdida.

» Próximo: [Configurando um servidor proxy com o Squid](#)

O Squid permite compartilhar a conexão entre vários micros, servindo como um intermediário entre eles e a internet. Usar um proxy é diferente de simplesmente compartilhar a conexão diretamente, via NAT.

Ao compartilhar via NAT, os micros da rede acessam a internet diretamente, sem restrições. O servidor apenas repassa as requisições recebidas, como um garoto de recados. O proxy é como um burocrata que não se limita a repassar as requisições: ele analisa todo o tráfego de dados, separando o que pode ou não pode passar e guardando informações para uso posterior.

Compartilhar a conexão via NAT é mais simples do que usar um proxy como o Squid sob vários aspectos. Você compartilha a conexão no servidor, configura os clientes para o utilizarem como gateway e pronto. Ao usar um proxy, além da configuração da rede, é necessário configurar o navegador e cada outro programa que for acessar a internet em cada cliente para usar o proxy. Esta é uma tarefa tediosa e que acaba dando bastante dor de cabeça a longo prazo, pois toda vez que um micro novo for colocado na rede ou for preciso reinstalar o sistema, será preciso fazer a configuração novamente.

A configuração do proxy muda de navegador para navegador. No Firefox, por exemplo, você a encontra em "Editar > Preferências > Geral > Proxy". No IE, a configuração está em "Opções da Internet > Opções > Configurações da Lan > Usar um servidor Proxy".



Além do navegador, outros programas podem ser configurados para trabalhar através do proxy: clientes de ICQ e MSN e até programas P2P. As vantagens de usar um proxy são basicamente três:

1- É possível impor restrições de acesso com base no horário, login, endereço IP da máquina e outras informações e bloquear páginas com conteúdo indesejado.

2- O proxy funciona como um cache de páginas e arquivos, armazenando informações já acessadas. Quando alguém acessa uma página que já foi carregada, o proxy envia os dados que guardou no cache, sem precisar acessar a mesma página repetidamente. Isso acaba economizando bastante banda, tornando o acesso mais rápido, sem precisar investir em uma conexão mais rápida.

Hoje em dia os sites costumam usar páginas dinâmicas, onde o conteúdo muda a cada visita, mas, mesmo nestes casos, o proxy dá uma ajuda, pois embora o html seja diferente a cada visita, e realmente precise ser baixado de novo, muitos componentes da página, como ilustrações, banners e animações em flash, podem ser aproveitados do cache, diminuindo o tempo total de carregamento.

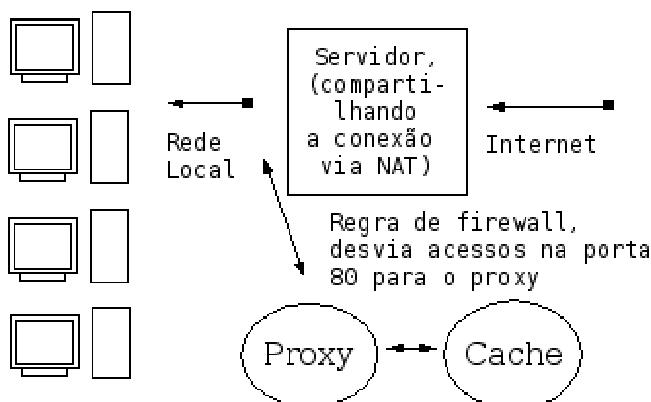
Dependendo da configuração, o proxy pode apenas acelerar o acesso às páginas ou servir como um verdadeiro cache de arquivos, armazenando atualizações do Windows Update, downloads diversos e pacotes instalados através do apt-get, por exemplo. Ao invés de ter que baixar o Service Pack XYZ do Windows XP ou o OpenOffice nos 10 micros da rede,

você vai precisar baixar apenas no primeiro, pois os outros 9 vão baixar a partir do cache do Squid.

3- Uma terceira vantagem de usar um proxy é que ele loga todos os acessos. Você pode visualizar os acessos posteriormente usando o **Sarg**, assim você sabe quem acessou quais páginas e em que horários. Além de tudo, o Squid é "dedo-duro" ;).

Mesmo assim, você pode estar achando que as vantagens não vão compensar o trabalho de sair configurando micro por micro, programa por programa para usar o proxy, e que é mais fácil simplesmente compartilhar via NAT. Mas existe a possibilidade de juntar as vantagens das duas formas de compartilhamento, configurando um **proxy transparente** como veremos adiante.

Ao usar um proxy transparente, você tem basicamente uma conexão compartilhada via NAT, com a mesma configuração básica nos clientes. O proxy entra na história como um adicional. Uma regra de firewall envia as requisições recebidas na porta 80 do servidor para o proxy, que se encarrega de responder aos clientes. Toda a navegação passa a ser feita automaticamente através do proxy (incluindo o cache dos arquivos do Windows update, downloads diversos e os pacotes instalados através do apt-get), sem que você precise fazer nenhuma configuração adicional nos clientes :).



O Kurumin inclui um script de instalação e configuração de proxy, disponível juntamente com meus outros scripts no painel de configuração de servidores. O script permite instalar o proxy, estabelecer algumas restrições de acesso e ativar o proxy transparente de forma simples. Neste tópico veremos como configurar o Squid "no muque" e criar regras elaboradas de restrição de acesso em qualquer distribuição.

» Próximo: [Instalando o Squid](#)

O Squid é composto de um único pacote, por isso a instalação é simples. Instale o pacote "**squid**" usando o apt-get, yum ou urpmi, como em:

```
# apt-get install squid
```

Toda a configuração do Squid é feita em um único arquivo, o "**/etc/squid/squid.conf**". Caso você esteja usando uma versão antiga do Squid, como a incluída no Debian Woody, por exemplo, o arquivo pode ser o "**/etc/squid.conf**". Apesar da mudança na localização do arquivo de configuração, as opções descritas aqui vão funcionar sem maiores problemas.

O arquivo original, instalado junto com o pacote, é realmente enorme, contém comentários e exemplos para quase todas as opções disponíveis. Ele pode ser uma leitura interessante se você já tem uma boa familiaridade com o Squid e quer aprender mais sobre cada opção. Mas, de início, é melhor começar com um arquivo de configuração mais simples, apenas com as opções mais usadas.

Em geral, cada distribuição inclui uma ferramenta diferente para a configuração do proxy, como o ícone mágico que inclui no Kurumin. Uma das mais usadas é o **Webmin**, disponível em várias distribuições. A função destas ferramentas é disponibilizar as opções através de uma interface gráfica e gerar o arquivo de configuração com base nas opções escolhidas.

Em alguns casos estas ferramentas ajudam bastante. Mas como elas mudam de distribuição para distribuição, acaba sendo mais produtivo aprender a trabalhar direto no arquivo de configuração, que, afinal, não é tão complicado assim. Comece renomeando o arquivo padrão:

```
# mv /etc/squid/squid.conf /etc/squid/squid.conf.velho
```

... e crie um novo arquivo "**/etc/squid/squid.conf**", com apenas as quatro linhas abaixo:

http_port		3128
visible_hostname		kurumin
acl	all	src
http_access	allow all	0.0.0.0/0.0.0.0

Estas linhas são o suficiente para que o Squid "funcione". Como você percebeu, aquele arquivo de configuração gigante tem mais uma função informativa, citando e explicando as centenas de opções disponíveis. Não se esqueça de substituir o "kurumin" na opção "visible_hostname" pelo nome correto do seu servidor, como informado pelo comando "hostname".

As quatro linhas dizem o seguinte:

http_port 3128: A porta onde o servidor Squid vai ficar disponível. A porta 3128 é o default.

visible_hostname kurumin: O nome do servidor, o mesmo que foi definido na configuração da rede.

acl all src 0.0.0.0/0.0.0.0 e http_access allow all: Estas duas linhas criam uma acl (uma política de acesso) chamada "all" (todos), incluindo todos os endereços IP possíveis. Ela permite que qualquer um dentro desta lista use o proxy, ou seja, permite que qualquer um use o proxy, sem limitações.

Para testar a configuração, reinicie o servidor Squid com o comando:

```
# /etc/init.d/squid restart
```

Se estiver no **Slackware**, o comando será:

```
# /etc/rc.d/rc.squid restart
```

Configure um navegador (no próprio servidor) para usar o proxy, através do endereço 127.0.0.1 (o localhost), porta 3128, e teste a conexão. Se tudo estiver ok, você conseguirá acessar o proxy também através dos outros micros da rede local, basta configurar os navegadores para usarem o proxy, fornecendo o endereço do servidor na rede local.

» Próximo: [Criando uma configuração básica](#)

O problema é que com apenas estas quatro linhas o proxy está muito aberto. Se você deixar o servidor proxy ativo no próprio servidor que compartilha a conexão e não houver nenhum firewall ativo, qualquer um na internet poderia usar o seu proxy, o que naturalmente não é desejado. O proxy deve ficar ativo apenas para a rede local.

Vamos gerar, então, um arquivo mais completo, permitindo que apenas os micros da rede local possam usar o proxy e definindo mais algumas políticas de segurança. Neste segundo exemplo já aproveitei algumas linhas do arquivo original, criando regras que permitem o acesso a apenas algumas portas e não a qualquer coisa, como na configuração anterior:

```
http_port 3128
visible_hostname kurumin

acl all src 0.0.0.0/0.0.0.0
acl manager src proto cache_object
acl localhost src port 127.0.0.1/255.255.255.255
acl SSL_ports port 443
acl Safe_ports port 80
acl Safe_ports port 21
acl Safe_ports port 443
acl Safe_ports port 563
acl Safe_ports port 70
acl Safe_ports port 210
acl Safe_ports port 280
acl Safe_ports port 488
acl Safe_ports port 591
acl Safe_ports port 777
acl Safe_ports port 901
acl Safe_ports port 1025-65535
acl purge method PURGE
acl CONNECT method CONNECT
```

```

http_access          allow           manager      localhost
http_access          deny            manager      manager
http_access          allow           purge        localhost
http_access          deny            purge        purge
http_access          deny           !Safe_ports
http_access deny CONNECT !SSL_ports

acl                 redelocal       src         192.168.1.0/24
http_access          allow           localhost
http_access allow redelocal

http_access deny all

```

Veja que agora criei duas novas acl's. A acl "localhost" contém o endereço 127.0.0.1, que você utiliza ao usar o proxy localmente (ao navegar usando o próprio servidor), e a acl "rede local", que inclui os demais micros da rede local. Substitua o "**192.168.1.0/24**" pela a faixa de endereços IP e a máscara de sub-rede usada na sua rede local (o 24 equivale à máscara 255.255.255.0).

Depois de criadas as duas políticas de acesso, vão duas linhas no final do arquivo que especificam que os micros que se enquadram nelas poderão usar o proxy:

```

http_access          allow           localhost
http_access allow redelocal

```

Lembra-se da acl "all", que contém todo mundo? Vamos usá-la para especificar que quem não se enquadra nas duas regras acima (ou seja, micros não-autorizados, da internet) não poderá usar o proxy:

```
http_access deny all
```

Esta linha deve ir no final do arquivo, depois das outras duas. A ordem é importante, pois o Squid interpreta as regras na ordem em que são colocadas no arquivo. Se você permite que o micro X acesse o proxy, ele acessa, mesmo que uma regra mais abaixo diga que não.

Se você adicionasse algo como:

```

acl                 redelocal       src         192.168.1.0/24
http_access          allow           redelocal
http_access deny redelocal

```

... os micros da rede local continuariam acessando, pois a regra que permite vem antes da que proíbe.

» Próximo: [Configurando o cache de páginas e arquivos](#)

Outra coisa importante é configurar o cache do proxy. O Squid trabalha com dois tipos de cache:

- 1- Cache rápido, feito usando parte da memória RAM do servidor.
- 2- Cache um pouco mais lento porém maior, feito no HD.

O cache na memória RAM é ideal para armazenar arquivos pequenos, como páginas .html e imagens, que serão entregues instantaneamente para os clientes. O cache no HD é usado para armazenar arquivos maiores, como downloads, arquivos do Windows update e pacotes baixados pelo apt-get.

O cache na memória RAM é sempre relativamente pequeno. Em um servidor não-dedicado (ou seja, uma máquina que é usada para fazer outras coisas, mas roda também o proxy), você vai reservar algo como 32 ou 64 MB de RAM para o cache, a fim de evitar que o cache do Squid ocupe toda a memória RAM, deixando o micro lento. Se você tiver uma rede maior e preferir deixar um micro dedicado apenas para o Squid, então o cache pode ter até 1/3 da memória RAM do servidor. Não caia no erro de reservar quase toda a RAM para o cache, pois além do cache o sistema vai precisar de memória para fazer outras coisas. Em um servidor com 1 GB de RAM você pode reservar uma percentagem um pouco maior, como 1/2 da memória total.

O cache no HD pode ser mais generoso, afinal a idéia é que ele guarde todo tipo de arquivos, principalmente os downloads grandes, que demoram para ser baixados. A única limitação neste caso é o espaço livre no HD. A configuração do cache é feita adicionando mais algumas linhas no arquivo de configuração:

1- A configuração da quantidade de memória RAM dedicada ao cache é feita adicionando a opção "cache_mem", que contém a quantidade de memória que será dedicada ao cache. Para reservar 64 MB, por exemplo, a linha ficaria:

cache_mem 64 MB

2- Abaixo vai mais uma linha, que determina o tamanho máximo dos arquivos que serão guardados no cache feito na memória RAM (o resto vai para o cache feito no HD). O cache na memória é muito mais rápido, mas como a quantidade de RAM é muito limitada, é melhor deixá-la disponível para páginas web, figuras e arquivos pequenos em geral. Para que o cache na memória armazene arquivos de até 64 KB, por exemplo, adicione a linha:

maximum_object_size_in_memory 64 KB

3- Em seguida vem a configuração do cache em disco, que armazenará o grosso dos arquivos. Por default, o máximo são downloads de 16 MB e o mínimo é zero, o que faz com que mesmo imagens e arquivos pequenos sejam armazenados no cache. Quase sempre é mais rápido ler a partir do cache do que baixar de novo da web, mesmo que o arquivo seja pequeno.

Se você faz download de arquivos grandes e deseja que eles fiquem armazenados no cache, aumente o valor da opção "maximum_object_size". Isso é especialmente útil para quem precisa baixar muitos arquivos através do apt-get ou Windows update em muitos micros da rede. Se você quiser que o cache armazene arquivos de até 512 MB, por exemplo, as linhas ficariam:

maximum_object_size	512	MB
minimum_object_size	0 KB	

Você pode definir ainda a percentagem de uso do cache que fará o Squid começar a descartar os arquivos mais antigos. Por padrão, sempre que o cache atingir 95% de uso, serão descartados arquivos antigos até que a percentagem volte para um número abaixo de 90%:

cache_swap_low	90
cache_swap_high	95

4- Depois vem a configuração do tamanho do cache em disco propriamente dita, que é composta por quatro valores. O primeiro, (**/var/spool/squid**) indica a pasta onde o Squid armazena os arquivos do cache. Você pode querer alterar para uma pasta em uma partição separada, por exemplo. O "**2048**" indica a quantidade de espaço no HD (em MB) que será usada para o cache. Aumente o valor se você tem muito espaço no HD do servidor e quer que o Squid guarde os downloads por muito tempo.

Finalmente, os números **16 256** indicam a quantidade de subpastas que serão criadas dentro do diretório. Por padrão, temos 16 pastas com 256 subpastas cada uma. O número "ideal" de pastas e subpastas para um melhor desempenho varia de acordo com o sistema de arquivos usado, mas esta configuração padrão é adequada para a maioria das situações:

```
cache_dir ufs /var/spool/squid 2048 16 256
```

5- Você pode definir ainda o arquivo onde são guardados os logs de acesso do Squid. Por padrão, o Squid guarda o log de acesso no arquivo "**/var/log/squid/access.log**". Este arquivo é usado pelo Sarg para gerar as páginas com as estatísticas de acesso.

```
cache_access_log /var/log/squid/access.log
```

6- Mais uma configuração que você pode querer alterar é o padrão de atualização do cache. Estas três linhas precisam sempre ser usadas em conjunto, ou seja, você pode alterá-las, mas sempre as três precisam estar presentes no arquivo. Eliminando um, o Squid ignora as outras duas e usa o default.

Os números indicam o intervalo (em minutos) que o Squid irá aguardar antes de verificar se um item do cache (uma página, por exemplo) foi atualizado, para cada um dos três protocolos. O primeiro número (o 15) indica que o Squid verificará (a cada acesso) se as páginas e arquivos com mais de 15 minutos foram atualizados. Ele faz uma verificação rápida, checando o tamanho do arquivo, o que é rápido. Se o arquivo não mudou, ele continua fornecendo aos clientes o arquivo que está no cache, economizando banda da conexão

O terceiro número (o 2280, equivalente a dois dias) indica o tempo máximo, depois do qual o objeto é sempre verificado. Além do http e ftp, o Squid suporta o protocolo gopher, que era muito usado nos primórdios da internet para localizar documentos de texto, mas perdeu a relevância hoje em dia:

```

refresh_pattern          ^ftp:               15      20%      2280
refresh_pattern          ^gopher:            15      0%       2280
refresh_pattern . 15 20% 2280

```

Depois de adicionar estas configurações todas, o nosso arquivo de configuração já ficará bem maior:

```

http_port                3128
visible_hostname kurumin

cache_mem                32      MB
maximum_object_size_in_memory   KB
maximum_object_size           MB
minimum_object_size           KB
cache_swap_low              90
cache_swap_high             95
cache_dir      ufs        /var/spool/squid    2048      16      256
cache_access_log             /var/log/squid/access.log
refresh_pattern          ^ftp:               15      20%      2280
refresh_pattern          ^gopher:            15      0%       2280
refresh_pattern . 15 20% 2280

acl          all          src      0.0.0.0/0.0.0.0
acl          manager      proto   cache_object
acl          localhost    src     127.0.0.1/255.255.255.255
acl          SSL_ports    port    443      563
acl          Safe_ports   port    80       #
acl          Safe_ports   port    21       #
acl          Safe_ports   port    443      #
acl          Safe_ports   port    70       #
acl          Safe_ports   port    210      #
acl          Safe_ports   port    1025-65535 #
acl          Safe_ports   port    280      #
acl          Safe_ports   port    488      #
acl          Safe_ports   port    591      #
acl          Safe_ports   port    777      #
acl          Safe_ports   port    901      #
acl          purge        method   PURGE
acl CONNECT method CONNECT

http_access   allow      manager   localhost
http_access   deny       manager   manager
http_access   allow      purge     localhost
http_access   deny      purge     purge
http_access   deny      !Safe_ports
http_access deny CONNECT !SSL_ports

acl          redelocal   src      192.168.1.0/24
http_access   allow      redelocal  localhost
http_access allow redelocal

http_access deny all

```

Aqui já temos uma configuração mais completa, incluindo um conjunto de regras de segurança (para que o proxy seja usado apenas a partir da rede local) e a configuração do cache. Esta é uma configuração adequada para uso em uma rede doméstica ou pequeno escritório.

Em uma rede maior, você provavelmente iria querer adicionar algumas limitações de acesso, limitando o acesso a algumas páginas, criando um sistema de autenticação ou limitando o uso com base no horário, entre outras possibilidades.

» Próximo: [Adicionando restrições de acesso](#)

Em um ambiente de trabalho, a idéia é que os funcionários usem a internet para comunicação, pesquisa e outras funções relacionadas ao que estão fazendo. Muitas empresas permitem que acessem os e-mails pessoais e coisas do gênero, mas sempre até um certo limite. Seu chefe não vai gostar se começarem a passar a maior parte do tempo no Orkut, por exemplo.

» Próximo: [Bloqueando por domínio ou palavras](#)

O Squid permite bloquear sites indesejados de forma relativamente simples, onde você inclui na configuração uma acl contendo os sites não permitidos e cria uma política de acesso que bloqueia o acesso a eles.

Isso é feito usando o parâmetro "dstdomain" (destination domain). Veja um exemplo:

```
acl bloqueados dstdomain orkut.com playboy.abril.com.br
http_access deny bloqueados
```

Aqui eu criei uma acl chamada "**bloqueados**", que contém os endereços "orkut.com" e "playboy.abril.com.br" e, em seguida, incluí a regra "http_access deny bloqueados", que bloqueia o acesso a eles. Ao aplicar a regra, o Squid faz a resolução do domínio e passa a bloquear todas sub-páginas.

Existe uma ressalva: muitos sites podem ser acessados tanto com o "www" quanto sem. Para o Squid, "www.orkut.com" e "orkut.com" são duas coisas diferentes. Bloqueando o "orkut.com" os usuários ainda conseguirão acessar o site através do "www.orkut.com" e vice-versa. Para bloquear ambos, é preciso incluir as duas possibilidades dentro da regra, como em:

```
acl bloqueados dstdomain orkut.com www.orkut.com playboy.abril.com.br
http_access deny bloqueados
```

Você pode incluir quantos domínios quiser dentro da regra, basta separá-los por espaço e deixar tudo na mesma linha. Se a regra começar a ficar muito grande, você tem a opção de transferir as entradas para um arquivo. Neste caso, crie um arquivo de texto simples, com

todos os domínios desejados (um por linha) e use a regra abaixo na configuração do Squid. No exemplo, estou usando o arquivo "/etc/squid/bloqueados":

```
acl      bloqueados      url_regex      -i      "/etc/squid/bloqueados"  
http_access deny bloqueados
```

Naturalmente, não seria viável tentar bloquear manualmente todos os sites pornográficos, chats, comunidades online, e todos os outros tipos de sites que não são úteis num ambiente de trabalho. A idéia seria logar os acessos (com a ajuda do Sarg, que veremos mais adiante) e bloquear os sites mais acessados, conforme tomar conhecimento deles. É sempre uma corrida de gato e rato, mas, em se tratando de pessoas adultas, não há nada que uma boa conversa com o chefe não possa resolver ;).

De qualquer forma, em alguns ambientes, pode ser mais fácil bloquear inicialmente o acesso a todos os sites e ir abrindo o acesso a apenas alguns sites específicos, conforme a necessidade. Neste caso, invertemos a lógica da regra. Criamos um arquivo com sites permitidos, adicionamos a regra que permite o acesso a eles e em seguida bloqueamos o acesso a todos os demais, como neste exemplo:

```
acl      permitidos      url_regex      -i      "/etc/squid/permitidos"  
http_access      allow      permitidos  
http_access deny all
```

Nas versões recentes do Squid, ao bloquear um domínio, é automaticamente bloqueado também o endereço IP do servidor correspondente. Isso evita que os usuários da rede consigam burlar o proxy, acessando os sites diretamente pelo IP. De qualquer forma, você pode criar diretamente regras que bloqueiem determinados endereços IP, o que é útil em casos de servidores sem domínio registrado, ou que respondam por vários domínios. Neste caso, a regra ficaria:

```
acl      ips-bloqueados      dst      200.234.21.23      200.212.15.45  
http_access deny ips-bloqueados
```

Você pode descobrir rapidamente o endereço IP de um determinado domínio usando o comando "host", como em:

\$	host	google.com
google.com	A	216.239.57.99
google.com A 216.239.37.99		

Depois de adicionar as novas regras, nosso arquivo de configuração ficaria assim:

http_port	3128
visible_hostname kurumin	
cache_mem	32
maximum_object_size_in_memory	64
maximum_object_size	512
minimum_object_size	0
cache_swap_low	90
cache_swap_high	95

```

cache_dir          ufs      /var/spool/squid      2048      16      256
cache_access_log
refresh_pattern   ^ftp:    15      20%      2280
refresh_pattern   ^gopher:  15      0%       2280
refresh_pattern . 15 20% 2280

acl               all      src      proto      0.0.0.0/0.0.0.0
acl               manager  src      proto      cache_object
acl               localhost src      port      127.0.0.1/255.255.255.255
acl               SSL_ports src      port      443      563
acl               Safe_ports port     80      #
acl               Safe_ports port     21      #
acl               Safe_ports port     443      563      #
acl               Safe_ports port     70      #
acl               Safe_ports port     210      #
acl               Safe_ports port    1025-65535 #
acl               Safe_ports port     280      #
acl               Safe_ports port     488      #
acl               Safe_ports port     591      #
acl               Safe_ports port     777      #
acl               Safe_ports port     901      #
acl               purge      method
acl CONNECT method CONNECT

http_access allow  manager
http_access deny   manager
http_access allow  purge
http_access deny   purge
http_access deny   !Safe_ports
http_access deny CONNECT !SSL_ports

acl      bloqueados      url_regex      -i      "/etc/squid/bloqueados"
http_access deny bloqueados

acl      redelocal
http_access allow redelocal
http_access allow redelocal

http_access deny all

```

Veja que coloquei as duas regras antes do "http_access allow redelocal", que abre tudo para a rede local. Como o Squid processa as regras seqüencialmente, as páginas que forem bloqueadas pelas duas regras não chegarão a passar pela seguinte.

Uma segunda possibilidade é usar o parâmetro "dstdom_regex", que permite bloquear sites de uma forma mais geral, com base em palavras incluídas na URL de acesso. Você pode bloquear todas as páginas cujo endereço inclua a palavra "sexo", por exemplo. Ao usar esta regra, o Squid verifica a existência das palavras na URL do site e não no conteúdo da página.

Crie mais um arquivo de texto, contendo as palavras que devem ser bloqueadas (uma por linha) e adicione a regra abaixo, contendo a localização do arquivo:

```

acl      nomesproibidos      dstdom_regex      "/etc/squid/nomesproibidos"
http_access deny nomesproibidos

```

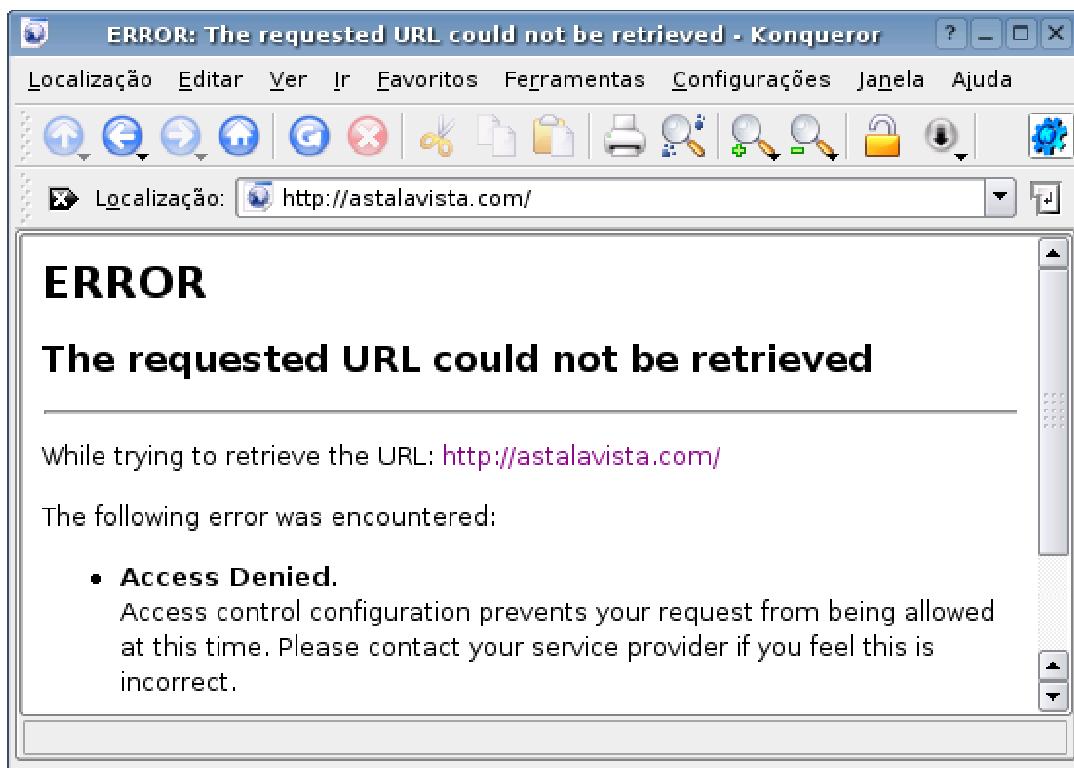
O uso desta regra é um pouco mais problemática, pois bloqueará todas páginas que contenham alguma das palavras listadas na URL. Esta opção sempre levará a alguns falsos positivos e por isso deve ser usada com mais cuidado.

Não existe problema em combinar o bloqueio de domínios e de palavras dentro da URL, você pode lançar mão de uma combinação das duas coisas, de acordo com a situação. Basta usar as duas regras simultaneamente, como em:

```
acl          bloqueados      url_regex      -i          "/etc/squid/bloqueados"
http_access
acl          nomesproibidos  dstdom_regex    "etc/squid/nomesproibidos"
http_access deny nomesproibidos

acl          redelocal       src           192.168.1.0/24
http_access
http_access allow          localhost
http_access allow          redelocal
http_access deny all
```

Incluídas as regras, os clientes passam a ver uma mensagem de erro ao tentar acessar páginas que se enquadrem nos bloqueios:



Você pode personalizar as páginas de erro editando os arquivos dentro da pasta "/usr/share/squid/errors/English" ou "/usr/share/squid/errors/Portuguese" (de acordo com a língua definida na configuração). São várias páginas html, uma para cada tipo de erro indicado.

» Próximo: [Gerenciando o uso da banda](#)

O Squid oferece uma forma simples de limitar o uso da banda disponível e definir o quanto cada usuário pode usar (mantendo parte do link livre para os demais), utilizando um recurso chamado "delay pools". Por exemplo, imagine que você tem um link de 1 megabit para uma rede com 20 usuários. Se cada um puder ficar baixando o que quiser, é provável que a rede fique saturada em determinados horários, deixando a navegação lenta para todo mundo.

Você pode evitar isso limitando a banda que cada usuário pode usar e a banda total, que todos os usuários somados poderão usar simultaneamente. É recomendável, neste caso, que o servidor proxy (que combina todos os acessos via http) consuma um pouco menos que o total de banda disponível, de forma a sempre deixar um pouco reservado para outros protocolos.

Um link de 1 megabit (1024 kbytes) corresponde a 131.072 bytes por segundo. Nas regras do Squid, sempre usamos bytes, por isso lembre-se de fazer a conversão, dividindo tudo por 8 e multiplicando por 1024 para ter o número em bytes.

Podemos limitar a banda total usada pelo Squid a 114.688 bytes por segundo, deixando 128 kbytes do link livres para outros protocolos e limitar cada usuário a no máximo 16.384 bytes por segundo, que correspondem a 128 kbytes. Nem todos os usuários vão ficar baixando arquivos a todo momento, por isso o valor ideal reservado a cada usuário vai variar muito de acordo com a rede. Você pode acompanhar o uso do link e ir ajustando o valor conforme a utilização.

Neste caso, a parte final do arquivo de configuração ficaria:

acl	redelocal	src	192.168.1.0/24
delay_pools			1
delay_class			2
delay_parameters	1	114688/114688	16384/16384
delay_access	1	allow	redelocal
http_access		allow	localhost
http_access		allow	redelocal
http_access deny all			

A acl "redelocal" agora está condicionada a três novas regras, que aplicam o uso do limite de banda. O acesso continua sendo permitido, mas agora dentro das condições especificadas na linha "**delay_parameters 1 114688/114688 16384/16384**", onde vão os valores com a banda total disponível para o Squid e a banda disponível para cada usuário.

Veja que nesta regra limitamos a banda apenas para a acl "redelocal" e não para o "localhost". Isso significa que você continua conseguindo fazer downloads na velocidade máxima permitida pelo link a partir do servidor; a regra se aplica apenas às estações. É

possível também criar regras para endereços IP específicos, que poderão fazer downloads sem passar pelo filtro.

Concluindo, mais um tipo de bloqueio que é útil em muitas situações é com relação a formatos de arquivos. Você pode querer bloquear o download de arquivos .exe ou .sh para dificultar a instalação de programas nas estações, ou bloquear arquivo .avi ou .wmf para economizar banda da rede, por exemplo. Neste caso, você pode usar a regra a seguir, especificando as extensões de arquivo desejadas:

```
acl          video      url_regex      -i          \.avi  
http_access deny video
```

» Próximo: [Bloqueando por horário](#)

As regras a seguir fazem com que o proxy recuse conexões feitas dentro de determinados horários. Você pode definir regras para períodos específicos e combiná-las para bloquear todos os horários em que você não quer que o proxy seja usado. Para que o proxy bloquee acessos feitos entre meia-noite e 6:00 da manhã e no horário de almoço, por exemplo, você usaria as regras:

```
acl          madrugada    time        00:00-06:00  
http_access deny madrugada  
  
acl          almoco       time        12:00-14:00  
http_access deny almoco
```

Estas regras iriam novamente antes da regra "http_access allow redelocal" no arquivo de configuração.

Agora imagine que você quer fazer diferente. Ao invés de bloquear o acesso na hora de almoço, você quer deixar o proxy aberto, para que aqueles que queiram acessar o Orkut ou acessar os e-mails possam fazer isso fora do horário de trabalho. Neste caso você usaria uma regra como:

```
acl          almoco       time        12:00-14:00  
http_access allow almoco
```

Esta regra entraria no arquivo de configuração antes das regras "http_access deny bloqueados" e "http_access deny nomesproibidos". Assim, os acessos que forem aceitos pela regra do almoço não passarão pelas regras que fazem o bloqueio.

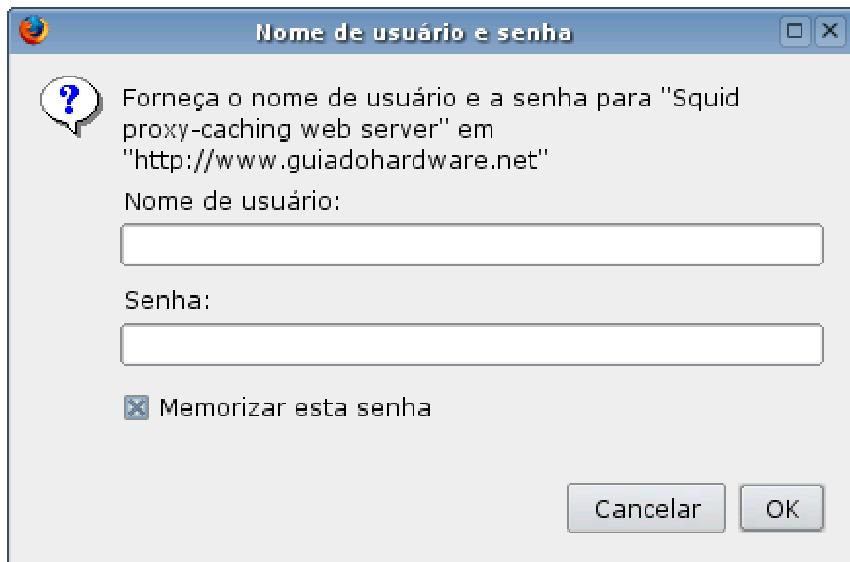
Você pode também combinar o bloqueio de palavras ou domínio com as regras de bloqueio por horário, permitindo que os usuários acessem um determinado site apenas no horário de almoço, por exemplo. A regra, neste caso, seria:

```
acl          almoco          time          12:00-14:00
acl      orkut      dstdomain      orkut.com      www.orkut.com
http_access allow orkut almoco
```

Assim, o acesso ao site (que normalmente estaria bloqueado em uma acl mais adiante) é permitido dentro do horário de almoço.

» Próximo: [Proxy com autenticação](#)

Você pode adicionar uma camada extra de segurança exigindo autenticação no proxy. Este recurso pode ser usado para controlar quem tem acesso à internet e auditar os acessos em caso de necessidade. Quase todos os navegadores oferecem a opção de salvar a senha, de modo que o usuário precisa digitá-la apenas uma vez a cada sessão:



Para ativar a autenticação, você vai precisar de um programa chamado "**htpasswd**". Se ele não estiver presente, instale o pacote **apache-utils**:

```
# apt-get install apache-utils
```

Em seguida crie o arquivo que será usado para armazenar as senhas:

```
# touch /etc/squid/squid_passwd
```

Cadastre os logins usando o comando:

```
#          htpasswd      /etc/squid/squid_passwd      kurumin
(onde o "kurumin" é o usuário que está sendo adicionado)
```

Depois de terminar de cadastrar os usuários, adicione as linhas que ativam a autenticação no "/etc/squid/squid.conf":

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/squid_passwd  
acl autenticados proxy_auth REQUIRED  
http_access allow autenticados
```

O "/usr/lib/squid/ncsa_auth" é a localização da biblioteca responsável pela autenticação. Eventualmente, ela pode estar em uma pasta diferente dentro da distribuição que estiver usando. Neste caso, use o comando "**locate**" ou a busca do KDE para encontrar o arquivo e altere a linha indicando a localização correta.

Estas três linhas criam uma acl chamada "autenticados" (poderia ser outro nome), que contém os usuários que se autenticarem usando um login válido. Ao implementar a autenticação, você pode criar regras de acesso com base nos logins dos usuários, não apenas com base nos endereços IP.

Por exemplo, imagine que você queria que apenas dois usuários da rede tenham acesso irrestrito ao proxy. Os demais (mesmo se autenticando), poderão acessar apenas no horário do almoço, e quem não tiver login e senha válidos não acessa em horário nenhum. Neste caso você poderia usar esta configuração:

```
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/squid_passwd  
acl autenticados proxy_auth REQUIRED  
  
acl permitidos proxy_auth kurumin tux  
acl almoco time 12:00-13:00  
  
http_access allow permitidos  
http_access allow autenticados almoco
```

Aqui temos os usuários que passaram pela autenticação divididos em duas regras. A acl "autenticados" inclui todos os usuários, enquanto a acl "permitidos" contém apenas o kurumin e o tux.

Graças à regra "http_access allow permitidos", os dois podem acessar em qualquer horário, enquanto os demais caem na regra "http_access allow autenticados almoco", que cruza o conteúdo das acls "autenticados" e "almoco", permitindo que eles acessem, mas apenas das 12:00 às 13:00.

» Próximo: [Configurando um proxy transparente](#)

Uma garantia de que os usuários realmente vão usar o proxy e, ao mesmo tempo, uma grande economia de trabalho e dor de cabeça para você é o recurso de proxy transparente.

Ele permite configurar o Squid e o firewall de forma que o servidor proxy fique escutando todas as conexões na porta 80. Mesmo que alguém tente desabilitar o proxy manualmente nas configurações do navegador, ele continuará sendo usado.

Outra vantagem é que este recurso permite usar o proxy sem precisar configurar manualmente o endereço em cada estação. Basta usar o endereço IP do servidor rodando o proxy como gateway da rede.

Lembre-se de que, para usar o proxy transparente, você já deve estar compartilhando a conexão no servidor via NAT, como vimos anteriormente. O proxy transparente apenas fará com que o proxy intercepte os acessos na porta 80, obrigando tudo a passar pelas suas regras de controle de acesso, log, autenticação e cache.

Para ativar o proxy transparente, rode o comando abaixo. Ele direciona as requisições recebidas na porta 80 para o Squid:

```
# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j \
REDIRECT --to-port 3128
```

O "eth0" no comando indica a placa da rede local, onde o proxy recebe as requisições dos outros micros da rede e o "3128" indica a porta usada pelo Squid. Adicione o comando junto com os 4 comandos que compartilham a conexão no final do arquivo "/etc/rc.local" ou ao seu script de firewall para que ele seja executado durante o boot.

Finalmente, você precisa ativar o suporte ao modo transparente dentro do arquivo "/etc/squid/squid.conf" e reiniciar o serviço.

Se você está usando uma versão recente, do **Squid 2.6** em diante, a configuração é mais simples. Basta substituir a linha "**http_port 3128**" no início do arquivo por:

```
http_port 3128 transparent
```

Ou seja, na verdade você precisa apenas adicionar o "transparent", para que o Squid passe a entender as requisições redirecionadas pela regra do firewall.

No caso das versões mais antigas, anteriores à 2.6 (como a usada no Debian Sarge e no Ubuntu 5.10), é necessário adicionar as quatro linhas abaixo, no final do arquivo "/etc/squid/squid.conf" (neste caso, sem alterar a linha "http_port 3128"):

httpd_accel_host	virtual
httpd_accel_port	80
httpd_accel_with_proxy	on
httpd_accel_uses_host_header on	

Em qualquer um dos dois casos, você precisa reiniciar o serviço para que a alteração entre em vigor:

```
# /etc/init.d/squid restart
```

Em caso de dúvida sobre qual versão do Squid está instalada, use o comando "squid -v", que além de reportar a versão, informa todas as opções que foram usadas durante a compilação:

```
# squid -v
```

Squid Cache: Version 2.6.STABLE2

```
configure options: '--prefix=/usr' '--exec_prefix=/usr' '--bindir=/usr/sbin' '--sbindir=/usr/sbin' '--libexecdir=/usr/lib/squid' '--sysconfdir=/etc/squid' '--localstatedir=/var/spool/squid' '--datadir=/usr/share/squid' '--enable-async-io' '--with-pthreads' '--enable-storeio=ufs,aufs,diskd,null' '--enable-linux-netfilter' '--enable-linux-proxy' '--enable-arp-acl' '--enable-epoll' '--enable-removal-policies=lru,heap' '--enable-snmp' '--enable-delay-pools' '--enable-htcp' '--enable-cache-digests' '--enable-underscores' '--enable-referer-log' '--enable-useragent-log' '--enable-auth=basic,digest,ntlm' '--enable-carp' '--with-large-files' 'i386-debian-linux' 'build_alias=i386-debian-linux' 'host_alias=i386-debian-linux' 'target_alias=i386-debian-linux'
```

Em resumo, você vai ter a conexão compartilhada via NAT no servidor e configurará os clientes para acessar através dela, colocando o servidor como gateway da rede. Ao ativar o proxy transparente, a configuração dos clientes continua igual, a única diferença é que agora (graças à nova regra do Iptables) todo o tráfego da porta 80 passará, obrigatoriamente, pelo servidor Squid.

Isso permite que você se beneficie do log dos acessos e do cache feito pelo proxy, sem ter que se sujeitar às desvantagens de usar um proxy, como ter que configurar manualmente cada estação.

Uma observação importante é que esta configuração de proxy transparente **não** funciona em conjunto com o sistema de autenticação incluso no Squid. Ao usar o proxy transparente a autenticação deixa de funcionar, fazendo com que você precise escolher entre as duas coisas.

Outra limitação importante do uso do proxy transparente é que ele atende apenas ao tráfego da porta 80. Como a conexão é compartilhada via NAT, todo o tráfego de outros protocolos (incluindo páginas em HTTPS, que são acessadas através da porta 443) é encaminhado diretamente, sem passar pelo proxy. Ou seja, embora seja uma forma simples de implementar um sistema de cache e algumas restrições de acesso, o uso do proxy transparente está longe de ser uma solução ideal.

Em situações onde você realmente precisa ter controle sobre o tráfego da rede, a única opção acaba sendo utilizar um proxy "normal", sem NAT. Uma solução para reduzir seu trabalho de administração nesse caso é implantar um sistema de configuração automática de proxy nos clientes.

» Próximo: [Usando o Sarg para monitorar o acesso](#)

O Sarg é um interpretador de logs para o Squid, assim como o Webalizer é para o Apache. Sempre que executado, ele cria um conjunto de páginas, divididas por dia, com uma lista de todas as páginas que foram acessadas e a partir de que máquina da rede veio cada acesso. Caso você tenha configurado o Squid para exigir autenticação, ele organiza os acessos com base nos logins dos usuários. Caso contrário, ele mostra os endereços IP das máquinas.

A partir daí você pode acompanhar as páginas que estão sendo acessadas, mesmo que não exista nenhum filtro de conteúdo, e tomar as medidas cabíveis em casos de abuso. Todos sabemos que os filtros de conteúdo nunca são completamente eficazes, eles sempre bloqueiam algumas páginas úteis e deixam passar muitas páginas impróprias. Se você tiver algum tempo para ir acompanhando os logs, a inspeção manual é sempre o método mais eficiente. Você pode ir fazendo um trabalho incremental, ir bloqueando uma a uma as páginas onde os usuários perdem muito tempo, ou fazer algum trabalho educativo, explicando que os acessos estão sendo monitorados e estabelecendo algum tipo de punição para quem abusar.

Aqui está um exemplo do relatório gerado pelo Sarg. Por padrão, ele gera um conjunto de páginas html dentro da pasta `/var/www/squid-reports/` (ou `/var/www/html/squid/`, em muitas distribuições), que você pode visualizar através de qualquer navegador. Os acessos são organizados por usuário (caso esteja sendo usada autenticação) ou por IP, mostrando as páginas acessadas por cada um, quantidade de dados transmitidos, tempo gasto em cada acesso, tentativas de acesso bloqueadas pelos filtros de conteúdo e outras informações.

Kurumin, log de acessos através do proxy

Periodo: 2004Oct02-2004Oct02
Ordem: BYTES, reverse
Topuser Relatorio

NUM	USUÁRIO	CONEXÃO	BYTES	%BYTES IN-CACHE-OUT	TEMPO GASTO	MILISEG	%TEMPO
1	data/hora192.168.0.2	266	1.314.660	81.81% 1.18% 98.82%	00:09:19	559.519	76.63%
2	data/horakurumin	46	287.118	17.87% 14.85% 85.15%	00:02:39	159.650	21.87%
3	data/hora127.0.0.1	11	5.199	0.32% 0.00% 100.00%	00:00:10	10.985	1.50%
TOTAL			3231,606,977	3.62% 96.39%	00:12:10	730,154	
MÉDIA			107	535,659		00:04:03	243,384

Gerado por [sarg-1.4.1 25Apr2003](#) em Oct/02/2004 09:04

Página carregada.

O Sarg é incluído na maioria das distribuições atuais, em alguns casos instalado por padrão junto com o Squid. No Debian e derivados ele pode ser instalado com um:

```
# apt-get install sarg
```

No Mandriva, ele é instalado através do "**urpmi sarg**".

Depois de instalado, chame o comando "**sarg**" (como root) para que os relatórios sejam gerados automaticamente a partir do log do Squid.

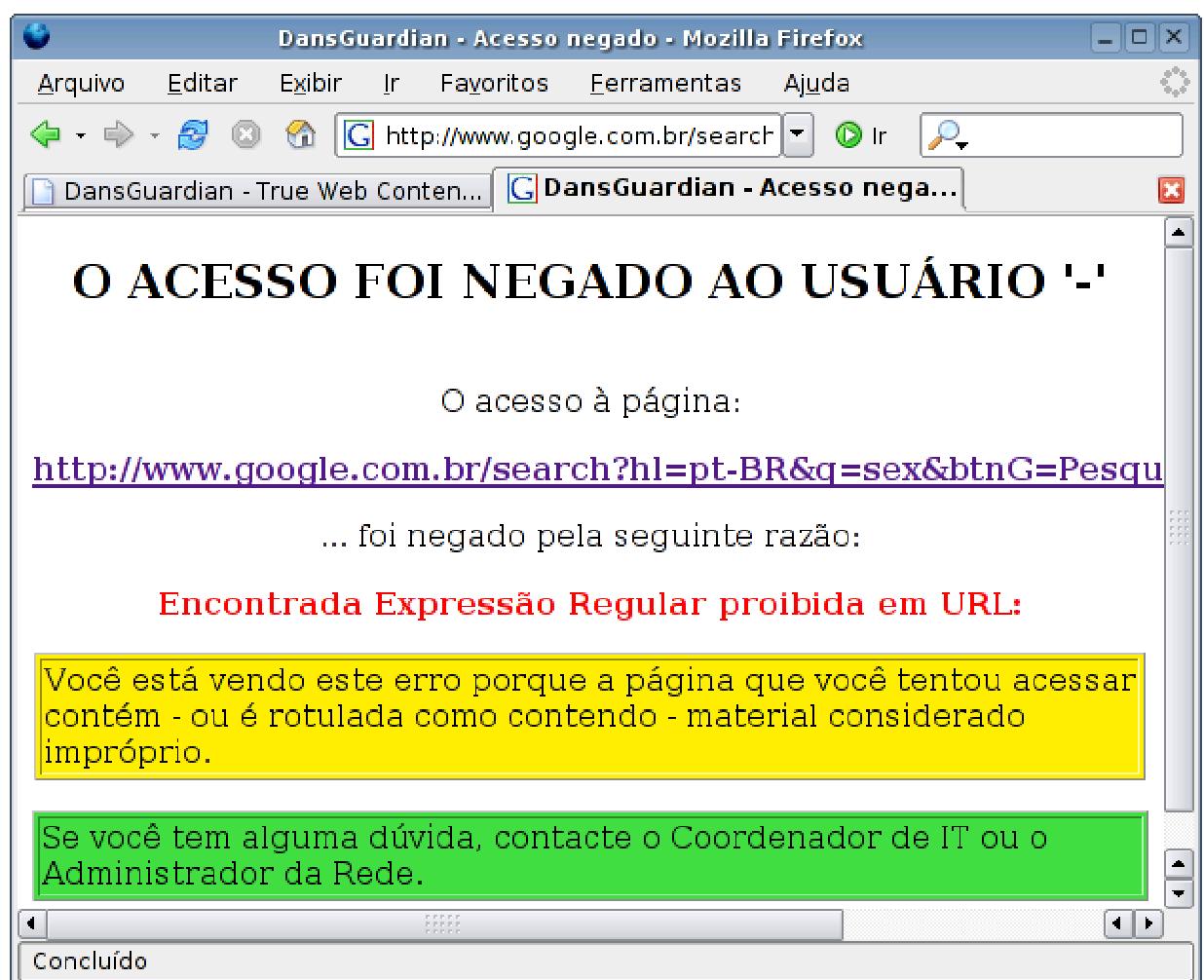
O Sarg não é um daemon que fica residente, você precisa apenas chamá-lo quando quiser atualizar o relatório. Se quiser automatizar esta tarefa, você pode usar o cron para que ele seja executado automaticamente todos os dias ou uma vez por hora, por exemplo.

Você pode alterar a pasta onde são salvos os relatórios, limitar o acesso às estatísticas e alterar várias opções cosméticas no arquivo de configuração do Sarg, que é o "**/etc/sarg/sarg.conf**" (no Mandriva) ou "**/etc/squid/sarg.conf**" (no Debian). O arquivo é auto-explicativo, nele você pode alterar os diretórios-padrão, alterar o layout da página, cores e títulos, etc. Outro recurso interessante é o envio de uma cópia do relatório por e-mail sempre que o Sarg for executado.

» Próximo: [Usando o DansGuardian para barrar páginas impróprias](#)

Bloquear domínios e endereços IP individuais funciona bem para bloquear páginas específicas, mas não funciona para bloquear páginas pornográficas, por exemplo, simplesmente porque existem muitas delas e você iria morrer louco se tentasse bloquear todas manualmente.

O DansGuardian é um filtro de conteúdo destinado a trabalhar junto com o Squid, filtrando páginas pornográficas e com outros tipos de conteúdo indesejado. Ele inclui um conjunto de regras prontas, que contém palavras, frases e tipos de arquivos freqüentemente usados neste tipo de página e endereços de páginas conhecidas. Cruzando estas informações, ele consegue fazer um excelente trabalho, realmente bloqueando quase todas as páginas indesejadas, em várias línguas, com relativamente poucos falsos-positivos.



Ele é ideal para uso em micros de trabalho e, principalmente, para quem tem crianças em casa e não quer que elas acessem páginas pornográficas.

Atualmente, o DansGuardian é um produto "semicomercial", que tem o código aberto e é gratuito para uso pessoal ou para qualquer fim não comercial (pode ser usado em uma escola ou escritório, por exemplo, desde que implementado internamente), mas é pago para uso comercial (quando você cobra pelo serviço de implantação, ou o fornece como parte de uma solução comercial). Você pode ver mais detalhes sobre a licença de uso no: <http://dansguardian.org/?page=copyright2>

Ao instalar, comece verificando se já não existe um pacote disponível na distribuição que está usando. O DansGuardian é um pacote de uso muito comum; por isso, a maioria das distribuições o inclui nos CDs de instalação. No Debian, por exemplo, você pode instalá-lo com um:

```
# apt-get install dansguardian
```

Você pode também encontrar pacotes para várias distribuições, junto com o tradicional pacote com código fonte no <http://dansguardian.org/?page=download2>.

Depois de instalar o pacote, inicie-o com o comando:

```
#                               /etc/init.d/dansguardian      start  
ou:  
# dansguardian &
```

Para que o DansGuardian funcione, é preciso que o Squid esteja instalado e ativo. Ele trabalha sobre o Squid, implementando suas políticas de acesso, mas deixando que o próprio Squid faça o acesso à web, cache e autenticação.

O principal arquivo de configuração é o "**/etc/dansguardian/dansguardian.conf**". Ao editá-lo pela primeira vez, é importante verificar algumas opções:

```
# UNCONFIGURED
```

Esta linha deve ficar comentada, indicando que o arquivo já foi configurado por você.

```
language = 'portuguese'
```

Esta opção configura a língua em que as mensagens de acesso bloqueado serão mostradas aos clientes.

```
loglocation = '/var/log/dansguardian/access.log'
```

Aqui vai a localização do arquivo de log do dansguardian, onde ficam armazenados os endereços das páginas cujo acesso foi bloqueado. Serve tanto para verificar a eficiência do filtro, quanto para identificar falsos-positivos, ou seja, páginas legítimas que estão sendo bloqueadas por engano. Estas exceções podem ser especificadas individualmente no arquivo "**/etc/dansguardian/exceptionsitelist**", que funciona como uma white list, contendo

uma lista de páginas que sempre são permitidas, mesmo que sejam encontradas palavras proibidas dentro do texto.

filterport = 8080

A porta onde o DansGuardian fica ativo. Ele sempre deve utilizar uma porta diferente do Squid, pois são duas coisas separadas. O padrão é a porta 8080.

proxyip = 127.0.0.1

O endereço IP do servidor proxy que será usado. Por padrão, ele vai utilizar uma cópia do Squid ativa na mesma máquina, mas é possível utilizar outro servidor Squid disponível na rede.

proxyport = 3128

A porta TCP onde o servidor Squid especificado na opção acima está ativo. Lembre-se de que, por padrão, o Squid usa a porta 3128.

A filtragem de páginas funciona em dois níveis. Ao receber a requisição do cliente, o DansGuardian verifica se o endereço a ser acesso está em uma das listas de domínios ou IPs proibidos. Caso esteja, o cliente recebe a mensagem de erro e o acesso sequer é feito, economizando banda.

Se não existir nenhum bloqueio relacionado ao domínio, a requisição é enviada ao Squid e o acesso é realizado. Ao receber os arquivos da página, o DansGuardian verifica o conteúdo da página, em busca de expressões e palavras "ruins", freqüentemente encontradas em páginas indesejadas, e também palavras "boas", normalmente encontradas em páginas de bom conteúdo.

Cada palavra ruim soma um certo número de pontos. Por exemplo, a palavra "s3xy" soma apenas 5 pontos, enquanto a expressão "s3x org1es" soma 80 pontos (estou trocando as vogais por números para que o meu próprio texto não caia nos filtros ;). Palavras "boas", por outro lado, subtraem pontos, fazendo com que a página tenha uma possibilidade menor de ser bloqueada. A palavra "education" subtrai 20 pontos, enquanto "medical problem" subtrai 50. As listas com palavras boas e ruins, juntamente com o peso positivo ou negativo de cada uma, vão na pasta "**/etc/dansguardian/phraselist**".

No final, o site recebe uma nota, apelidada pelos desenvolvedores de "naughtynesslimit", ou "índice de sem-vergonhice", resultado da soma de todas as palavras boas e ruins. Você define um índice máximo a ser tolerado no arquivo "**/etc/dansguardian/dansguardianf1.conf**", na opção:

naughtynesslimit = 160

Quanto mais baixo o número, mais severa é a censura, porém mais páginas boas acabam sendo bloqueadas por engano. Os valores recomendados pelos desenvolvedores são "60" para crianças pequenas, "100" para pré-adolescentes e "160" para adolescentes. Para um público adulto, onde a principal preocupação seja não bloquear páginas boas, mesmo que

isso faça com que uma ou outra página inadequada passe pelo filtro de vez em quando, você pode arriscar "200" ou mesmo "240".

Como você pode notar dando uma olhada no conteúdo dos arquivos das listas de palavras, o DansGuardian vem configurado com listas em inglês, que deixam passar muitos sites nacionais. Você pode baixar um arquivo com listas em outras línguas, incluindo português no:

<http://dansguardian.org/downloads/grosvenor/languages.tar.gz>

Para instalar, descompacte o arquivo "languages.tar.gz" e copie os arquivos de dentro da pasta "languages", que será criada para a pasta "**/etc/dansguardian/phraselist/**". Falta agora configurar o DansGuardian para utilizar os novos arquivos. Para isso, abra o arquivo "**/etc/dansguardian/weightedphraselist**" e adicione as linhas:

```
.Include</etc/squid/dansguardian/languages/weightedphraselist.pornsites.portuguese>
.Include</etc/squid/dansguardian/languages/weightedphraselist.pornwords.portuguese>
```

Antes de usar estas listas de palavras, verifique o conteúdo dos arquivos. As listas de palavras em português são excessivamente rigorosas, o que faz com que seja bloqueado o acesso a um número muito grande de sites "bons", mesmo ao usar um naughtynesslimit alto. Use-os com cautela.

Aparentemente, os arquivos disponíveis no site foram escritos por um estrangeiro. Por isso, não se adaptam bem à nossa realidade. Se decidir corrigir os arquivos, não deixe de enviá-los para os mantenedores, para que sejam incluídos no pacote.

Note que neste arquivo são especificados todos os arquivos de palavras que são utilizados. Na pasta existem várias categorias diferentes. Em algumas situações, você pode querer desabilitar algumas das categorias, a fim de flexibilizar o filtro. Você pode adicionar novas palavras ou editar o peso de cada uma, editando diretamente os arquivos.

Concluindo, abra também o arquivo "**/etc/dansguardian/bannedphraselist**" e inclua a linha:

```
.Include</etc/squid/dansguardian/languages/bannedphraselist.portuguese>
```

Lembre-se de que é necessário reiniciar o DansGuardian para que qualquer uma das alterações tenha efeito:

/etc/init.d/dansguardian restart

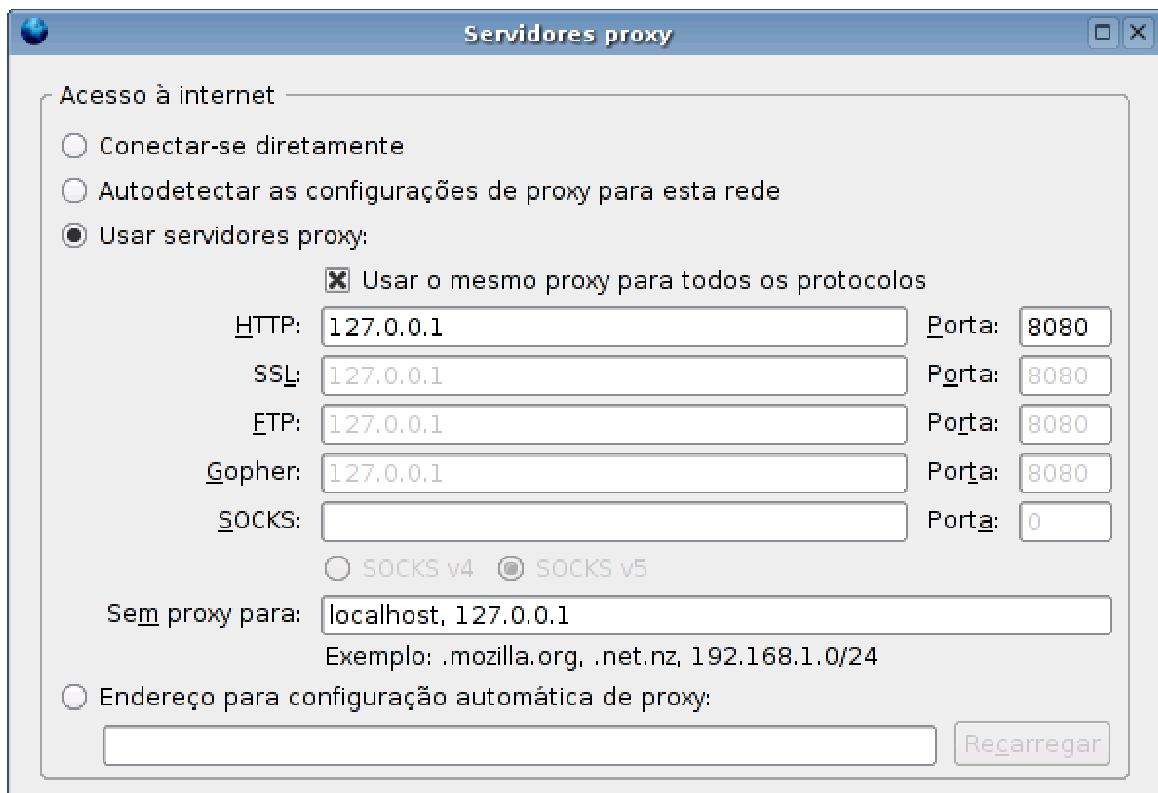
» Próximo: [Configurando os clientes](#)

O DansGuardian pode ser usado tanto dentro da rede, quanto localmente. Além de ser

utilizado em redes de todos os tamanhos, muita gente com crianças em casa se dá ao trabalho de instalá-lo avulso, no micro de casa.

Para usá-lo, você precisa configurar os navegadores, tanto os instalados no próprio servidor onde ele está sendo executado, quanto em outras máquinas da rede para acessarem via proxy. Nas configurações do proxy, coloque o endereço IP do servidor (como por exemplo 192.168.0.1) e a porta do DansGuardian, definida no arquivo de configuração. Lembre-se de que, por padrão, ele usa a porta 8080.

No Firefox, a opção de configurar um proxy está em "Editar > Preferências > Proxy". Ou seja, com exceção da porta diferente, a configuração para usar o DansGuardian é a mesma que para um proxy tradicional. Em casos onde ele é usado num micro doméstico, com o objetivo de servir como um simples filtro de conteúdo, você pode até mesmo rodá-lo localmente. Neste caso, use o endereço "127.0.0.1" como proxy:



» Próximo: [Atualizando as blacklists](#)

Além do filtro com base em palavras, o DansGuardian utiliza uma lista de sites proibidos, que sequer chegam a ser acessados. Por padrão, o DansGuardian vem com uma lista muito

pequena e desatualizada, apenas como exemplo. Para efetivamente usar este recurso, é preciso baixar uma lista de palavras mais elaborada.

Você pode baixar uma lista longa e atualizada no: <http://urlblacklist.com/>.

O link completo para a versão mais recente é:
<http://urlblacklist.com/cgi-bin/commercialdownload.pl?type=download&file=bigblacklist>

Para instalar, basta descompactar o arquivo e mover o conteúdo para dentro da pasta "/etc/dansguardian/", substituindo a pasta "/etc/dansguardian/blacklists" existente:

```
$ tar -zvxf bigblacklist.tar.gz  
# cp -a --reply=yes blacklists/ /etc/dansguardian/
```

Depois de instalar o arquivo completo, você pode usar o script de atualização, disponível no site para baixar atualizações de forma automática. Baixe-o em:
<http://urlblacklist.com/downloads/UpdateBL>

Basta ativar a permissão de execução e executá-lo. Em algumas distribuições é preciso criar a pasta "/var/lib/lrpkg/", onde ele guarda os logs. Sem esta pasta, ele exibe um erro e não conclui a atualização.

```
# mkdir /var/lib/lrpkg  
# chmod +x /var/lib/lrpkg/UpdateBL  
# ./UpdateBL
```

O pacote inclui várias listas diferentes, separadas por assunto. As listas incluem muitos assuntos inocentes como, "cellphones", "sports" e "childcare" (saúde infantil). Ele não é uma "blacklist" no sentido estrito da palavra, mas sim um conjunto de listas que inclui também sites sobre conteúdos diversos. A idéia aqui é que você pode bloquear todos os assuntos desejados.

```

kurumin@kurumin:/etc/dansguardian/blacklists$ ls
ads          dangerous_material  jewelry        religion
adult        dating             jobsearch      ringtones
aggressive   dialers           kidstimestwasting searchengines
antispware   domains           mail           sportnews
artnudes     drugs             mobile-phone  sports
audio-video  ecommerce         entertainment spyware
beerliquorinfo  expressions    onlineauctions strong_redirector
beerliquorsale forums           onlinegames   updatesites
blacklists.info forums          onlinepayment urls
CATEGORIES   frencheducation   personalfinance vacation
cellphones   gambling          pets            violence
chat         gardening         porn            virusinfected
childcare    government       proxy           warez
cleaning     hacking           publicite      weapons
clothing    homerepair       radio           webmail
culinary    hygiene           redirector    whitelist
kurumin@kurumin:/etc/dansguardian/blacklists$ █

```

Dentro de cada uma das subpastas, você encontra três arquivos: domains (sites completamente bloqueados), expressions e urls (páginas específicas, dentro de sites permitidos). Para ativar o uso das blacklists, edite os arquivos `"/etc/dansguardian/bannedsitelist"` e `"/etc/dansguardian/bannedurllist"`, adicionando (ou descomentando) as linhas referentes às categorias que devem ser ativadas.

Para bloquear páginas de conteúdo adulto (adult), drogas (drugs), páginas pornográficas (porn) e warez, adicione (ou descomente) no arquivo `"/etc/dansguardian/bannedurllist"` as linhas:

```

.Include</etc/dansguardian/blacklists/adult/urls>
.Include</etc/dansguardian/blacklists/drugs/urls>
.Include</etc/dansguardian/blacklists/porn/urls>
.Include</etc/dansguardian/blacklists/warez/urls>

```

No arquivo `"/etc/dansguardian/bannedsitelist"` vão as linhas:

```

.Include</etc/dansguardian/blacklists/adult/domains>
.Include</etc/dansguardian/blacklists/drugs/domains>
.Include</etc/dansguardian/blacklists/porn/domains>
.Include</etc/dansguardian/blacklists/warez/domains>

```

Você pode usar também os arquivos com expressões proibidas, incluídos no pacote para reforçar a lista adicional, com os termos em português, que já ativamos anteriormente. Para isso, abra novamente o arquivo `"/etc/dansguardian/bannedphraselist"` e adicione as linhas:

```

.Include</etc/dansguardian/blacklists/adult/expressions>
.Include</etc/dansguardian/blacklists/drugs/expressions>

```

```
.Include</etc/dansguardian/blacklists/porn/expressions>
.Include</etc/dansguardian/blacklists/warez/expressions>
```

Faça o mesmo com outras categorias que quiser adicionar.

» Próximo: [Proxy transparente com o DansGuardian](#)

Como vimos até agora, o DansGuardian funciona como uma camada extra, uma espécie de "pedágio", por onde as requisições passam antes de chegarem ao Squid e por onde as respostas passam antes de serem enviadas ao cliente.

Normalmente, os clientes precisam ser configurados manualmente para utilizar o DansGuardian como proxy, acessando-o através da porta 8080. Isso traz de volta o problema de configurar manualmente cada um dos micros e evitar que os usuários removam a configuração para acessar diretamente, sem passar pelo filtro.

Contudo, é possível configurar o DansGuardian para trabalhar como proxy transparente, da mesma forma que fizemos anteriormente com o Squid. Neste caso, o firewall redireciona as requisições recebidas na porta 80 para o DansGuardian e ele as repassa para o Squid, que finalmente faz o acesso. Os clientes precisam apenas ser configurados para acessar a internet usando o servidor onde estão instalados o Squid e DansGuardian como gateway.

Comece adicionando as quatro linhas que ativam o proxy transparente no "/etc/squid/squid.conf":

```
httpd_accel_host                                     virtual
httpd_accel_port                                    80
httpd_accel_with_proxy                             on
httpd_accel_uses_host_header on
```

Depois vêm as regras de firewall para habilitar o compartilhamento da conexão e direcionar as requisições recebidas na porta 80 para a porta usada pelo DansGuardian. Novamente, é a mesma configuração usada para fazer um proxy transparente no Squid, mudando apenas a porta. Lembre-se que o "eth0" deve ser substituído pela interface ligada na rede local:

```
modprobe
iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT \
--to-port 8080
iptables -A INPUT -m tcp -p tcp -s ! 127.0.0.1 --dport 3128 -j DROP
```

A última regra bloqueia a porta 3128 usada pelo Squid, para impedir que algum espertinho configure o navegador para acessar diretamente através do Squid, sem passar pelo DansGuardian. A única exceção é o endereço 127.0.0.1, ou seja, o próprio servidor. Lembre-se de colocar estes comandos no arquivo "/etc/rc.d/rc.local" ou "/etc/init.d/bootmisc.sh" para não precisar ficar digitando tudo a cada boot.

» Próximo: [Capítulo 6: Configurando um servidor de arquivos e impressoras](#)

A necessidade de compartilhar arquivos e impressoras motivou o aparecimento das primeiras redes (ainda na década de 70) e continua sendo uma necessidade comum. Mesmo para fazer um simples backup armazenado remotamente, é necessário configurar algum tipo de compartilhamento de arquivos.

Hoje em dia, as duas soluções mais usadas são o Samba e o NFS. O Samba é a solução mais completa, pois inclui várias opções de segurança e permite que os compartilhamentos sejam acessados tanto a partir de clientes Windows, quanto de clientes Linux. O NFS é um sistema mais limitado, porém mais simples de usar, que permite compartilhar arquivos entre máquinas Linux.

Também é possível transferir arquivos via FTP, SFTP ou até mesmo via HTTP, mas estas soluções são mais apropriadas para uso via internet. Numa rede local, um compartilhamento do Samba acaba sendo mais prático de usar.

O Samba é o servidor que permite compartilhar arquivos e acessar compartilhamentos em máquinas Windows. Ele é dividido em dois módulos, o servidor Samba propriamente dito e o "smbclient", o cliente que permite acessar compartilhamentos em outras máquinas. Usando o Samba, o servidor Linux se comporta exatamente da mesma forma que uma máquina Windows, compartilhando arquivos e impressoras e executando outras funções, como autenticação de usuários. Você pode configurar o Samba até mesmo para tornar-se um controlador de domínio.

A primeira versão do Samba, disponibilizada em 1992, foi escrita por Andrew Tridgell, um australiano então estudante de ciências da computação. Como na época a especificação do SMB utilizada pela Microsoft ainda era fechada, Andrew desenvolveu um pequeno programa, batizado de *clockspy*, para examinar os pacotes de dados enviados por uma máquina Windows e, assim, ir implementando uma a uma as chamadas de sistema utilizadas, um trabalho bastante complexo.

O resultado foi um programa que rodava no Solaris (o sistema Unix desenvolvido pela Sun) e era capaz de responder às chamadas SMB como se fosse um servidor Windows. Este arquivo ainda pode ser encontrado em alguns dos FTPs do <http://samba.org>, com o nome "server-0.5".

O objetivo desta primeira versão era apenas resolver um problema doméstico: interligar um PC rodando o Windows 3.1 à workstation Sun que ele tinha em casa. Na época isso já era possível utilizando um dos clientes NFS comerciais para DOS, mas Andrew precisava de suporte a NetBIOS para um aplicativo que pretendia utilizar, o WindX, um servidor X para Windows, que permitia rodar aplicativos via rede a partir do servidor Unix.

Até aí o objetivo era apenas fazer o programa funcionar, não criar um sistema de compartilhamento de arquivos. Depois de algum tempo, Andrew recebeu um e-mail contando que o programa também funcionava com o LanManager da Microsoft, permitindo compartilhar arquivos de um servidor Unix com máquinas rodando o DOS. Andrew só acreditou depois de testar, mas ficou tão maravilhado com o que havia conseguido que criou o projeto "NetBios for Unix" e começou a recrutar voluntários através da Usenet. Mais tarde o projeto passou a usar o nome Samba, que foi adotado não em apologia ao Carnaval, mas apenas porque é uma das poucas palavras do dicionário do Aspell que possui as letras S, M e B, de "Server Message Blocks".

Em 94 a Microsoft liberou as especificações do SMB e do NetBios, o que permitiu que o desenvolvimento do Samba desse um grande salto, tanto em recursos quanto em compatibilidade, passando a acompanhar os novos recursos adicionados ao protocolo da Microsoft, que mais tarde novamente deixou de ser aberto.

Hoje, além de ser quase 100% compatível com os recursos de rede do Windows 98, NT e 2000, o Samba é reconhecido por ser mais rápido que o próprio Windows na tarefa de servidor de arquivos.

Um dos pontos fortes do Samba é que o projeto foi todo desenvolvido sem precisar apelar para qualquer violação de patentes. Todas as chamadas (com exceção das que a Microsoft tornou públicas em 94) foram implementadas monitorando as transmissões de dados através da rede, uma espécie de engenharia reversa que não tem nada de ilegal. É como se você descobrisse como funciona um código de encriptação apenas examinando arquivos encriptados por ele. Matemáticos fazem isso a todo instante e muitas vezes são bem pagos para isso. Graças a este "detalhe", o Samba não corre o perigo de sofrer restrições devido a ações judiciais.

De qualquer forma, não existem sinais de que a Microsoft pretenda declarar guerra ao Samba. Pelo contrário, foi a existência do Samba que permitiu que a Microsoft conseguisse colocar PCs rodando o Windows em muitos nichos onde só entravam Workstations Unix, já que com o Samba os servidores Unix existentes passaram a ser compatíveis com as máquinas Windows. Ou seja: de certa forma, o Samba foi vantajoso até mesmo para a Microsoft.

» Próximo: [Instalando](#)

O Samba é dividido em dois módulos. O servidor propriamente dito e o cliente, que permite acessar compartilhamentos em outras máquinas (tanto Linux quanto Windows). Os dois são independentes, permitindo que você mantenha apenas o cliente instalado num desktop e instale o servidor apenas nas máquinas que realmente forem compartilhar arquivos. Isso permite melhorar a segurança da rede de uma forma geral.

Os pacotes do Samba recebem nomes um pouco diferentes nas distribuições derivadas do Debian e no Fedora e outras distribuições derivadas do Red Hat. Veja:

Pacote Debian Fedora

Servidor:	samba	samba
Cliente:	smbclient	samba-client
Documentação	samba-doc	samba-doc
Swat: swat samba-swat		

Lembre-se de que você deve instalar todos os pacotes apenas no servidor e em outras máquinas que forem compartilhar arquivos. O Swat ajuda bastante na etapa de configuração, mas ele é opcional, pois você pode tanto editar manualmente o arquivo smb.conf, quanto usar um arquivo pronto, gerado em outra instalação. Nos clientes que forem apenas acessar compartilhamentos de outras máquinas, instale apenas o cliente.

O Fedora inclui mais um pacote, o "system-config-samba", um utilitário de configuração rápida, que permite criar e desativar compartilhamentos de forma bem prática. Outro configurador rápido é o módulo "Internet & Rede > Samba", disponível no Painel de Controle do KDE. Neste livro abordo apenas o swat, que é o configurador mais completo, mas você pode lançar mão destes dois utilitários para realizar configurações rápidas.

Com os pacotes instalados, use os comandos:

```
#          /etc/init.d/samba      start
# /etc/init.d/samba stop
```

... para iniciar e parar o serviço. Por padrão, ao instalar o pacote é criado um link na pasta "/etc/rc5.d", que ativa o servidor automaticamente durante o boot. Para desativar a inicialização automática, use o comando:

```
# update-rc.d -f samba remove
```

Para reativá-lo mais tarde, use:

```
# update-rc.d -f samba defaults
```

No **Fedora** e **Mandriva**, os comandos para iniciar e parar o serviço são:

```
#          service      smb      start
# service smb stop
```

Para desabilitar o carregamento durante o boot, use o "**chkconfig smb off**" e, para reativar, use o "**chkconfig smb on**". Note que, em ambos, o pacote de instalação se chama "samba", mas o serviço de sistema chama-se apenas "smb".

» Próximo: [Cadastrando os usuários](#)

Depois de instalado, o próximo passo é cadastrar os logins e senhas dos usuários que terão acesso ao servidor. Esta é uma peculiaridade do Samba: ele roda como um programa sobre

o sistema e está subordinado às permissões de acesso deste. Por isso, ele só pode dar acesso para usuários que, além de estarem cadastrados no Samba, também estão cadastrados no sistema.

Existem duas abordagens possíveis. Você pode criar usuários "reais", usando o comando **adduser** ou um utilitário como o "**user-admin**" (disponível no Fedora e no Debian, através do pacote `gnome-system-tools`). Ao usar o adduser, o comando fica:

```
# adduser maria
```

Uma segunda opção é criar usuários "castrados", que terão acesso apenas ao Samba. Esta abordagem é mais segura, pois os usuários não poderão acessar o servidor via SSH ou Telnet, por exemplo, o que abriria brecha para vários tipos de ataques. Neste caso, você cria os usuários adicionando os parâmetros que orientam o adduser a não criar o diretório home e a manter a conta desativada até segunda ordem:

```
# adduser --disabled-login --no-create-home maria
```

Isso cria uma espécie de usuário fantasma que, para todos os fins, existe e pode acessar arquivos do sistema (de acordo com as permissões de acesso), mas que, por outro lado, não pode fazer login (nem localmente, nem remotamente via SSH), nem possui diretório home.

Uma dica é que no **Fedora** (e outras distribuições derivadas do Red Hat), você só consegue usar o comando caso logue-se como root usando o comando "**su -**" ao invés de simplesmente "**su**". A diferença entre os dois é que o "**su -**" ajusta as variáveis de ambiente, incluindo o PATH, ou seja, as pastas onde o sistema procura pelos executáveis usados nos comandos. Sem isso, o Fedora não encontra o executável do adduser, que vai na pasta "/usr/sbin".

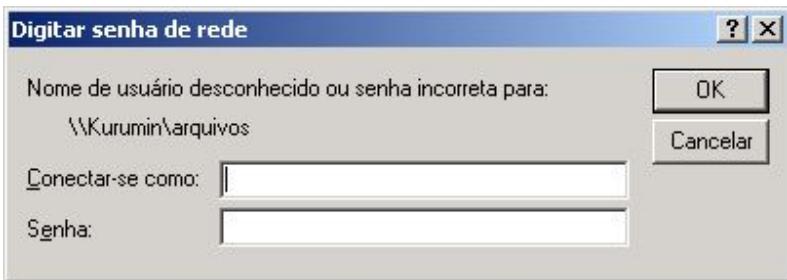
Os parâmetros suportados pelo adduser também são um pouco diferentes. O padrão já é criar um login desabilitado (você usa o comando "passwd usuário" para ativar) e, ao invés do "--no-create-home", usa a opção "-M". O comando (no Fedora) fica, então:

```
# adduser -M maria
```

De qualquer uma das duas formas, depois de criar os usuários no sistema você deve cadastrá-los no Samba, usando o comando "**smbpasswd -a**", como em:

```
# smbpasswd -a maria
```

Se você mantiver os logins e senhas sincronizados com os usados pelos usuários nos clientes Windows, o acesso aos compartilhamentos é automático. Caso os logins ou senhas no servidor sejam diferentes, o usuário precisará fazer login ao acessar.



Um detalhe importante é que, ao usar clientes Windows 95 ou 98, você deve marcar a opção de login como "**Login do Windows**" e não como "Cliente para redes Microsoft" (que é o default) na configuração de rede (Painel de controle > Redes).

Depois de criados os logins de acesso, falta agora apenas configurar o Samba para se integrar à rede e compartilhar as pastas desejadas, trabalho facilitado pelo **Swat**. A segunda opção é editar manualmente o arquivo de configuração do Samba, o "**/etc/samba/smb.conf**", como veremos mais adiante. Neste caso, o ideal é começar a partir de um arquivo pré-configurado, alterando apenas as opções necessárias. Você pode baixar o arquivo modelo, que é utilizado na instalação do Samba no Kurumin no <http://www.guiadohardware.net/kurumin/modelos/>.

» Próximo: [Configurando usando o Swat](#)

O Samba pode ser configurado através do Swat, um utilitário de configuração via web, similar ao encontrado nos modems ADSL. Isso permite que ele seja acessado remotamente e facilita a instalação em servidores sem o X instalado. Esta mesma abordagem é utilizada por muitos outros utilitários, como o Webmin e o Pagode.

Manter o X instalado e ativo em um servidor dedicado é considerado um desperdício de recursos, por isso os desenvolvedores de utilitários de configuração evitam depender de bibliotecas gráficas. Desse modo, mesmo distribuições minimalistas podem incluí-los.

Nas distribuições derivadas do Red Hat, o Swat é inicializado através do xinetd. Para ativá-lo depois da instalação, use os comandos:

```
# chkconfig swat on
# service xinetd restart
```

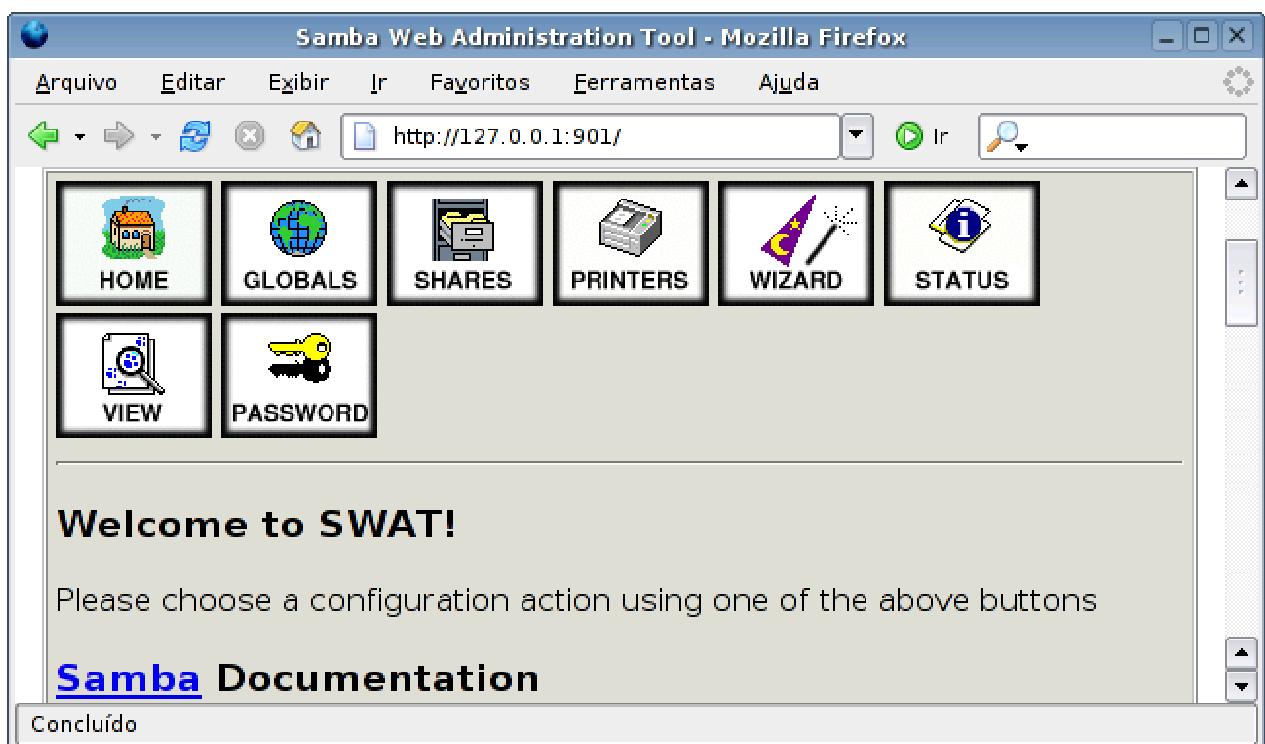
No Debian, Slackware e também no Gentoo, o Swat é inicializado através do inetd. A função do inetd e xinetd é parecida, eles monitoram determinadas portas TCP e carregam serviços sob demanda. Isto evita que utilitários que são acessados esporadicamente (como o Swat) precisem ficar ativos o tempo todo, consumindo recursos do sistema. Apesar disso, a configuração dos dois é diferente: no caso das distribuições que usam o inetd, você ainda precisa adicionar (ou descomentar) a linha abaixo no arquivo de configuração do inetd, o "**/etc/inetd.conf**":

```
swat stream tcp nowait.400 root /usr/sbin/tcpd /usr/sbin/swat
```

Para que a alteração entre em vigor, reinicie o inetd com o comando:

```
# /etc/init.d/inetd restart
```

Para acessar o Swat, basta abrir o Konqueror ou outro Browser disponível e acessar o endereço <http://localhost:901>. No prompt de login, forneça a senha de root (do sistema) para acessar. Ao abrir o Swat, você verá um menu como o do screenshot abaixo, com vários links para a documentação disponível sobre o Samba, que você pode consultar para se aprofundar no sistema. Na parte de cima, estão os links para as seções da configuração, que é o que nos interessa.



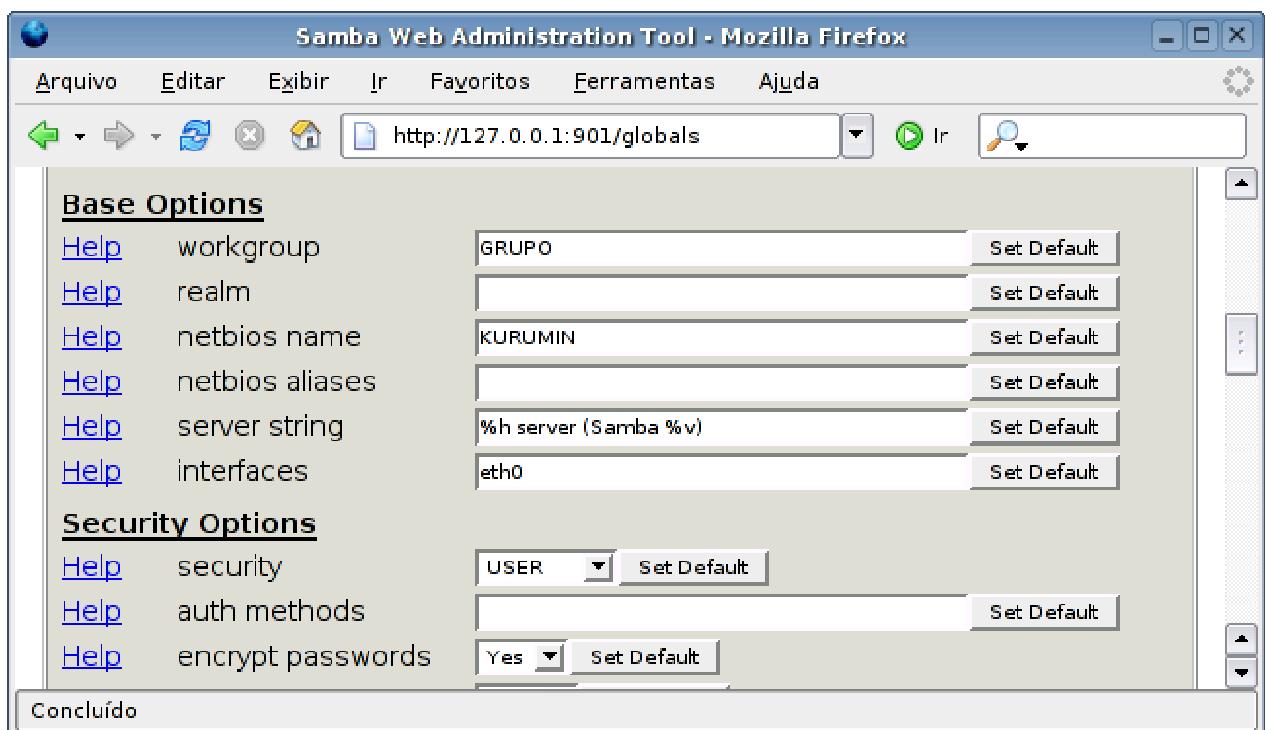
Na seção **Password**, você pode cadastrar usuários, substituindo o uso manual do comando "smbpasswd -a". Neste caso, você precisará primeiro cadastrar os usuários, utilizando o comando adduser. O Swat apenas cadastra os usuários no Samba.



Em seguida, accese a seção "**Globals**", que engloba todas as configurações de rede e acesso.

Nas opções "**workgroup**" e "**netbios name**", você deve colocar o nome do computador e o grupo de trabalho a que ele pertence, como faria em uma máquina Windows. Você pode tanto utilizar o mesmo grupo de trabalho em todas as máquinas da rede, quanto agrupar suas máquinas em grupos distintos como "diretoria", "vendas", etc.

A opção "**netbios aliases**" permite criar "apelidos" para o servidor, de modo de que ele possa ser acessado por mais de um nome. Usando um alias, o servidor realmente aparece duas vezes no ambiente de rede, como se existissem duas máquinas. Em geral isso acaba confundindo mais do que ajudando, mas pode ser útil em algumas situações, quando, por exemplo, um servidor é desativado e os compartilhamentos são movidos para outro. O novo servidor pode responder pelo nome do servidor antigo, permitindo que os desavisados continuem acessando os compartilhamentos.



A seguir temos a opção "**interfaces**", que permite limitar os acessos ao servidor se você tiver mais de uma placa de rede. É o caso, por exemplo, de quem acessa via ADSL ou cabo e possui uma segunda placa de rede para compartilhar a conexão com os micros da rede local. Nesses casos, a placa da web será reconhecida como **eth0**, enquanto a placa da rede local será reconhecida como **eth1**, por exemplo.

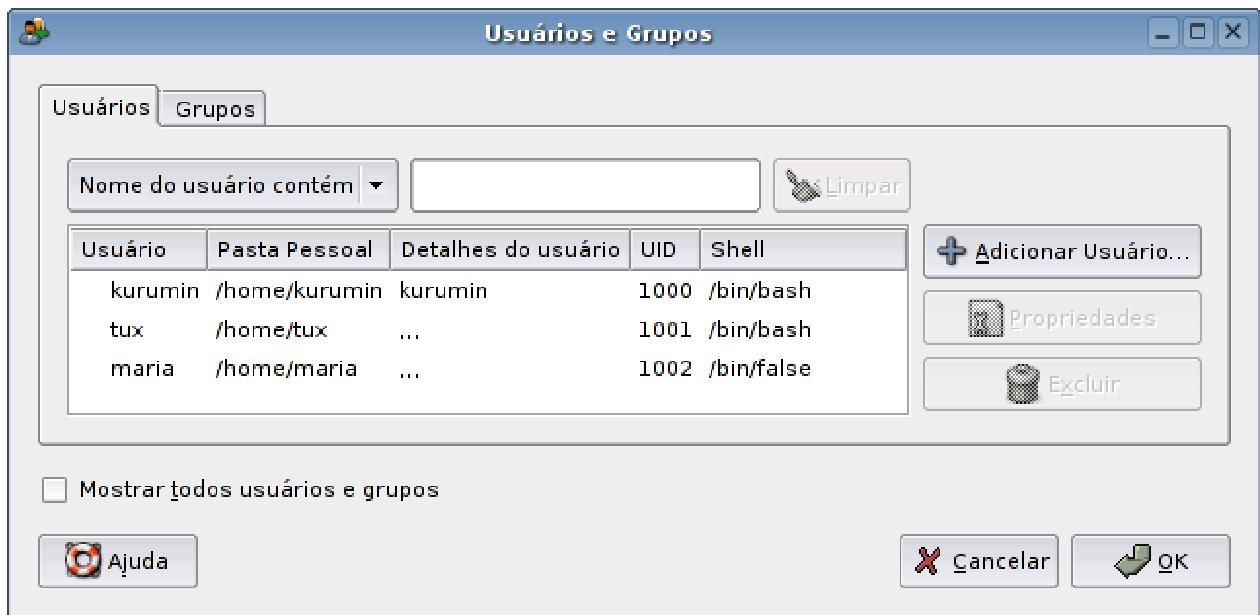
Você pode, então, preencher o campo com o endereço da placa da rede local (eth1). Assim, o Samba só aceitará conexões vindas dos micros da rede local, descartando automaticamente todas as tentativas de acesso vindas da internet. Caso o campo permaneça vazio, o Samba permite acessos vindos de todas as placas de rede, e é necessário bloquear os acessos provenientes da internet usando o firewall.

Na seção **Security Options** chegamos a uma das decisões mais importantes, decidir entre utilizar segurança com base no login do usuário (**user**) ou com base no compartilhamento (**share**).

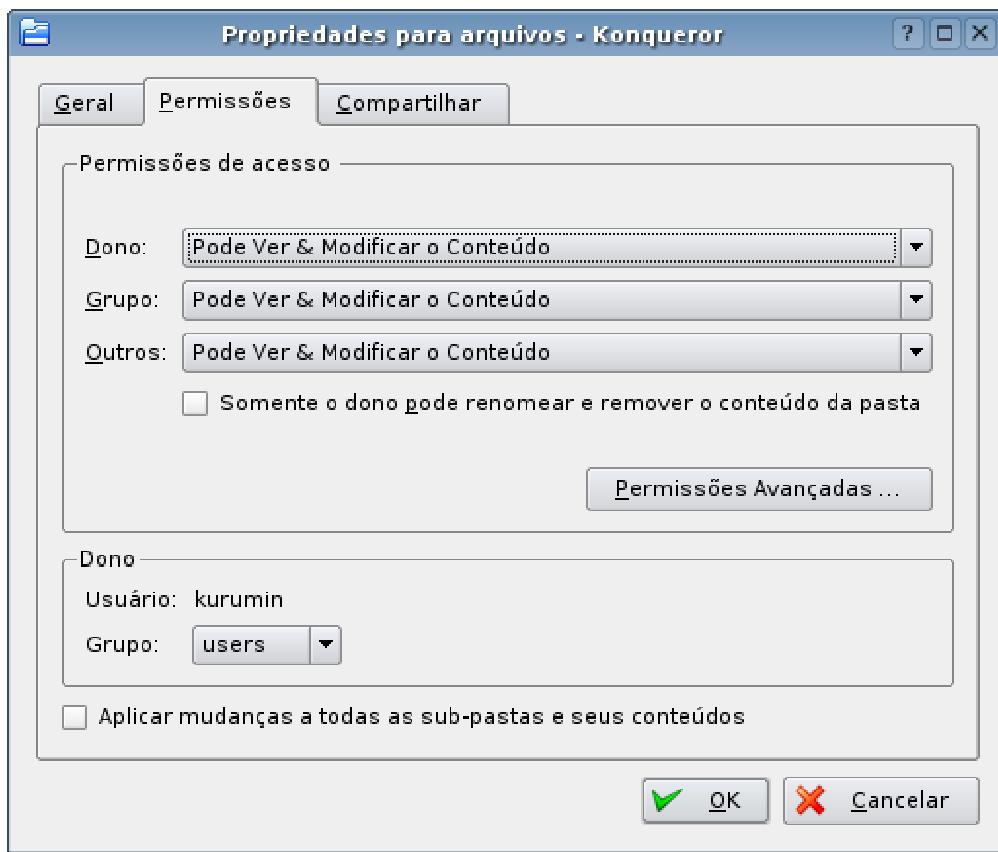
A opção **share** oferece um nível de segurança semelhante ao de uma máquina Windows 98. Os compartilhamentos podem ser acessados por todos os usuários, através da conta guest. Em compensação, esta opção é a mais simples de configurar e pode ser útil em pequenas redes onde não há necessidade de segurança. A opção **user** é a mais recomendável, pois permite especificar exatamente quais usuários terão acesso a cada compartilhamento, como em um servidor NT ou Windows 2003. Naturalmente, para que isso funcione, é necessário que você tenha registrado todos os usuários no Linux e no Samba (como vimos anteriormente), e que os clientes Windows efetuem login na rede usando estes mesmos logins e senhas, ou os forneçam na hora de acessar os compartilhamentos.

Utilizando o modo user, as permissões de acesso aos compartilhamentos do samba ficam condicionadas às permissões de acesso de cada usuário. Por exemplo, se você compartilhar a pasta **/home/maria/arquivos**, por default apenas a usuária maria terá permissão para gravar novos arquivos e alterar o conteúdo da pasta.

Para que outros usuários tenham acesso à pasta, você deve dar permissão a eles, criando um novo grupo e dando permissão de escrita para os integrantes do mesmo. Outra opção é adicionar os demais usuários no grupo "maria" (cada usuário possui um grupo com o mesmo nome do login, criado no momento em que é cadastrado) e configurar as permissões de acesso de forma que o grupo possa escrever na pasta. Você pode fazer a administração de grupos usando o "**users-admin**", que facilita bastante as coisas ao trabalhar com um grande número de usuários. Lembre-se que no Debian ele é instalado através do pacote "gnome-system-tools". No Fedora ele se chama "system-config-users".



Se você não está tão preocupado com a segurança, pode fazer do jeito "fácil", alterando a opção "outros" nas permissões de acesso da pasta, que dá acesso a todo mundo. Isso faz com que qualquer usuário local do sistema (ou logado via SSH) tenha acesso aos arquivos da pasta, mas não permite necessariamente que outros usuários do Samba possam acessar, pois neste caso ainda são usadas as permissões de acesso no Samba. A alteração das permissões da pasta é feita usando o Konqueror ou outro gerenciador de arquivos e não através do Samba.



Ou seja, é necessário fazer com que os usuários do grupo, ou todos os usuários do sistema, possam escrever na pasta, evitando que as permissões do sistema conflitem com as permissões configuradas no Samba. Se configuro o Samba para permitir que o usuário "joao" possa escrever no compartilhamento, mas a configuração das permissões da pasta compartilhada não permitem isso, o joao vai continuar sem conseguir escrever. Ao criar compartilhamentos no Samba, é preciso se preocupar com as duas coisas.

Mais abaixo, temos a opção **Encrypt Password**. Ela também é importantíssima, e deve ficar sempre ativada (Encrypt Password = Yes). O Windows 95 original não suporta encriptação de senhas, por isso só poderá se conectar ao servidor caso a opção seja configurada com o valor "No". Porém, o Windows 95 OSR/2, Windows 98/SE/ME, Windows NT, Windows 2000, XP e Vista utilizam senhas encriptadas. Ao utilizar máquinas com qualquer um destes sistemas (99.9% dos casos), a opção deve ser configurada como "Yes", caso contrário o Samba simplesmente não conseguirá conversar com as máquinas Windows e você vai ficar quebrando a cabeça até se lembrar deste parágrafo ;).

A partir do Samba 3 existe a opção de fazer com que o próprio Samba mantenha as senhas dos usuários sincronizadas em relação às senhas dos mesmos no sistema. Antigamente, sempre que você alterava a senha de um usuário no Samba, usando o "smbpasswd", precisava alterar também a senha do sistema, usando o comando "passwd". As duas senhas precisam ficar em sincronismo, do contrário caímos no problema das permissões, onde o

Samba permite que o usuário acesse o compartilhamento, mas o sistema não permite que o Samba acesse os arquivos no disco.

Para ativar este recurso, ative a opção "**unix password sync**" no Swat. Originalmente, esta opção fica desativada e aparece apenas dentro das opções avançadas. Para chegar até ela você deve clicar no botão "Change View To: Advanced" no topo da tela. Depois de alterar, clique no Commit Changes".

Para que tudo funcione, é necessário que as opções "passwd program" e "passwd chat" estejam configuradas com (respectivamente) os valores: "/usr/bin/passwd %u" e "*Enter\snew\ssUNIX\spassword:*\n%u\n*Retype\snew\ssUNIX\spassword:*\n%u\n.". Estes já são os valores padrão no Swat, mas não custa verificar.

The screenshot shows the "Samba Web Administration Tool - Mozilla Firefox" interface. The title bar reads "Samba Web Administration Tool - Mozilla Firefox". The menu bar includes "Arquivo", "Editar", "Exibir", "Ir", "Favoritos", "Ferramentas", and "Ajuda". The address bar shows the URL "http://127.0.0.1:901/globals". The main content area is titled "Security Options". It lists various configuration options with their current values and "Set Default" buttons:

Opção	Valor Atual	Ação
security	USER	Set Default
auth methods		Set Default
encrypt passwords	Yes	Set Default
client schannel	Auto	Set Default
server schannel	Auto	Set Default
obey pam restrictions	Yes	Set Default
passdb backend	smbpasswd	Set Default
guest account	nobody	Set Default
passwd program	/usr/bin/passwd %u	Set Default
passwd chat	*Enter\snew\ssUNIX\spassword:*\n%u\n*Retype\snew\ssUNIX\spassword:*\n%u\n.	Set Default
unix password sync	Yes	Set Default

At the bottom left, there is a "Concluído" button.

A opção "**Hosts Allow**" deve incluir os endereços IP de todos os computadores que terão permissão para acessar o servidor. Se quiser que todos os micros da rede tenham acesso, basta escrever apenas a primeira parte do endereço IP, como em "**192.168.0.**", onde todos os endereços dentro do escopo serão permitidos. Se for incluir mais de um endereço ou mais de um escopo de endereços, separe-os usando vírgula e espaço, como em: "192.168.0., 10.0.0., 123.73.45.167". Caso o campo permaneça vazio, a opção fica desativada e todos os micros que tiverem acesso ao servidor Samba poderão acessar.

A opção "**Hosts Deny**", por sua vez, permite especificar máquinas que **não** terão permissão para acessar o servidor. É importante notar que as opções "Hosts Allow" e "Hosts Deny" possuem algumas peculiaridades, sobretudo quando usadas em conjunto. Veremos mais detalhes sobre o uso das duas mais adiante.

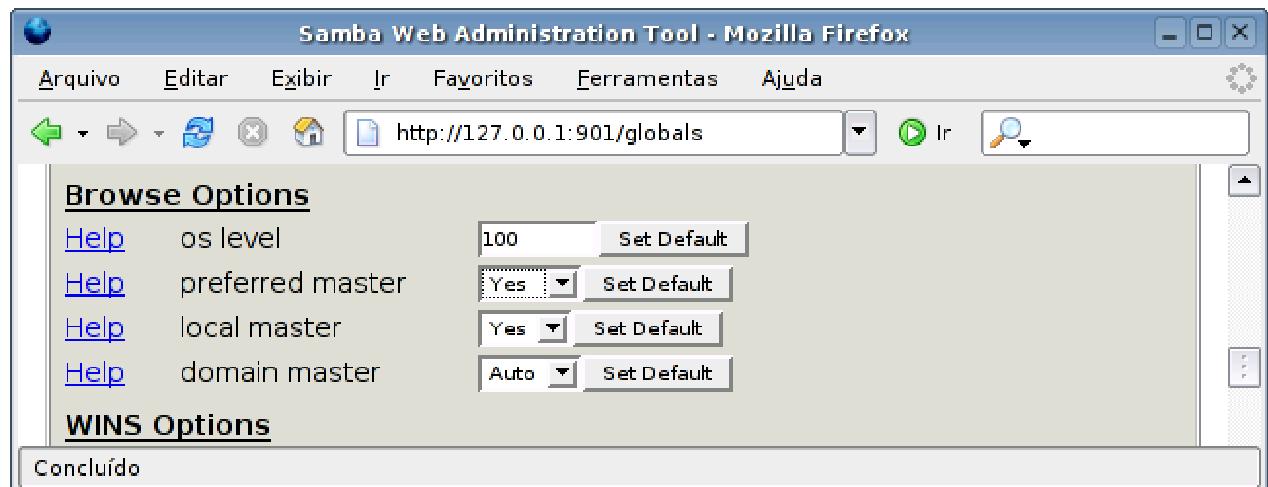
Ao combinar o uso das opções "hosts allow" e "hosts deny", a opção "hosts allow" tem precedência (não importa a ordem em que elas sejam colocadas), de forma que as máquinas listadas terão acesso, mesmo que ele seja negado pela opção "hosts deny". Por exemplo, ao combinar a opção "hosts allow = 192.168.1." com a "hosts deny = 192.168.1.43" o host "192.168.1.43" continuará tendo acesso ao compartilhamento, pois faz parte da faixa de endereços cujo acesso é autorizado pela opção "hosts allow".

Neste caso, o Samba não considera a opção "hosts deny = 192.168.1.43" como uma exceção, mas sim como um erro de configuração. Para bloquear a máquina, você deveria usar: hosts allow = 192.168.1. EXCEPT 192.168.1.43.

Numa rede Windows, uma das máquinas fica sempre responsável por montar e atualizar uma lista dos compartilhamentos disponíveis e enviá-la aos demais, conforme solicitado. O host que executa esta função é chamado de "**Master Browser**".

Na seção Browse Options, a opção "**OS Level**" permite especificar qual chance o servidor Linux terá de ser o Master Browser do grupo de trabalho ou domínio. Sempre que você estiver configurando o Samba para ser o servidor principal, é desejável que ele seja o master browser.

Para isso, configure esta opção com um valor alto, 100 por exemplo, para que ele sempre ganhe as eleições. O default dessa opção é 20, que faz com que ele perca para qualquer máquina Windows NT, Windows 2000 ou XP. Para completar, deixe a opção "**Local Master**" e "**Preferred Master**" como "Yes".



A configuração do OS Level é muito importante. Caso não seja o Master Browser, você poderá ter problemas para acessar seu servidor Linux a partir de outras máquinas Windows, principalmente rodando o NT/2000/XP. Com o valor 100, sempre que uma das máquinas Windows tentar ser o Master Browser da rede, o Samba convocará uma nova eleição e a máquina Linux sempre ganhará :-).

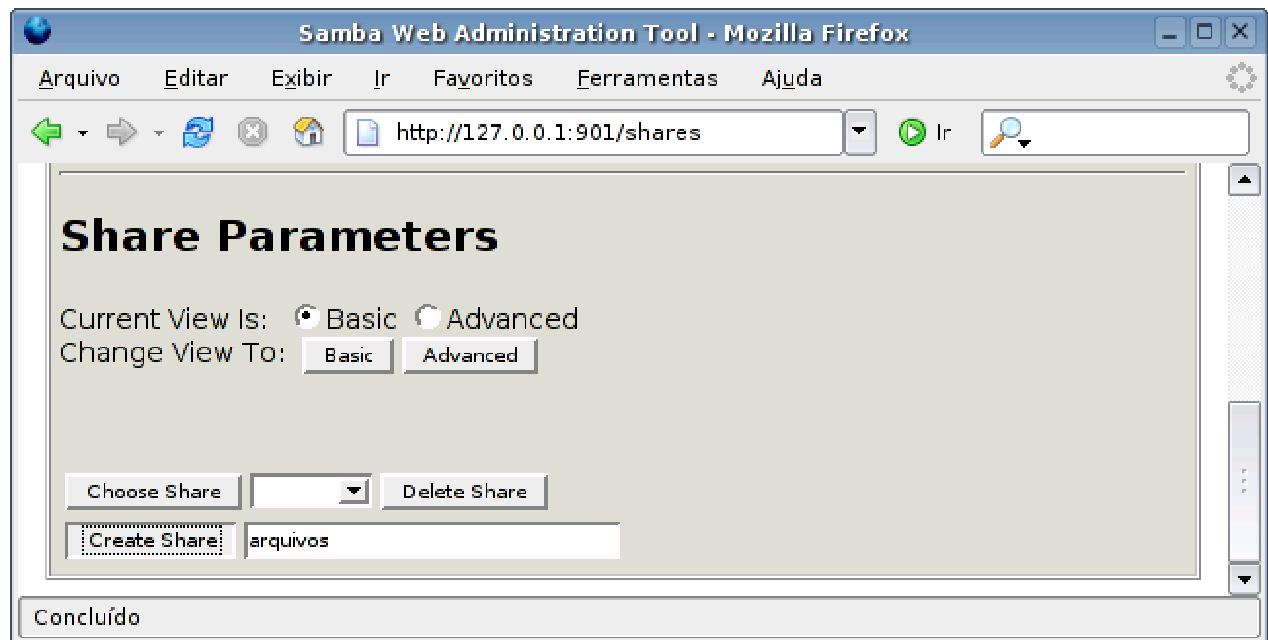
Abaixo, deixe a opção **Wins Support** ativada (**Yes**). A opção **Wins Server** deve ser deixada em branco, a menos que exista na rede algum servidor Wins (rodando o NT server ou o 2K server) ao qual o servidor Linux esteja subordinado.

Caso o único servidor seja a máquina Linux, você pode configurar as máquinas Windows para utilizá-la como servidor Wins, para isto basta colocar o seu endereço IP no campo "Servidor Wins" na configuração de rede das estações. Terminando, pressione o botão "**Commit Changes**" no topo da tela para que as alterações sejam salvas no arquivo "`/etc/samba/smb.conf`".

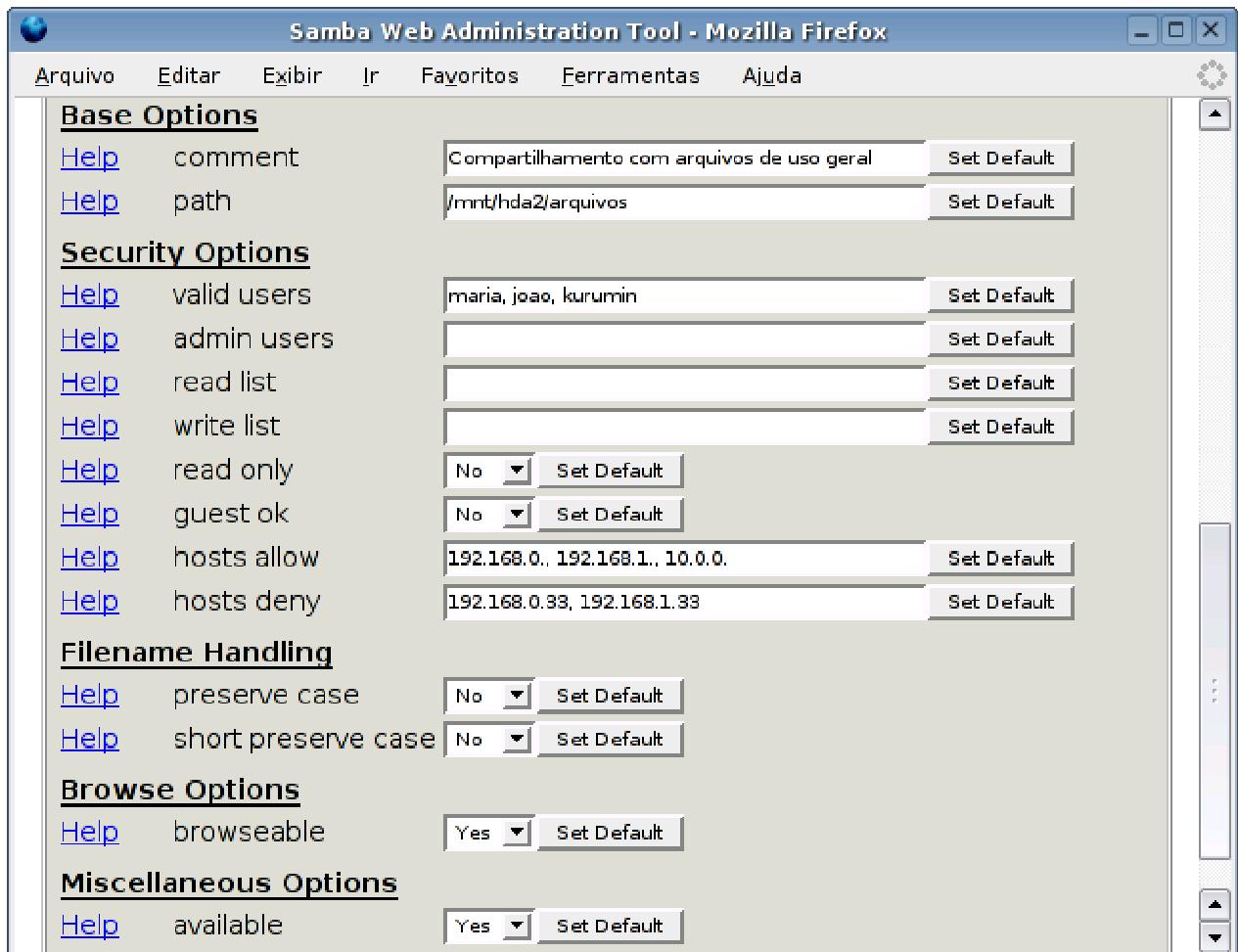
Uma observação importante é que o Swat lê o arquivo smb.conf ao ser aberto, lendo as opções configuradas e mostrando-as na interface, mas gera um novo arquivo sempre que você clica no "Commit Changes". Ao ler o arquivo, ele procura por trechos específicos de texto, ignorando tudo que for diferente. Isso faz com que ele remova qualquer tipo de comentário incluído manualmente no arquivo. Em geral, quem tem o hábito de editar manualmente o smb.conf, acaba nunca usando o Swat e vive-versa.

Depois de cadastrar os usuários no sistema e no Samba e configurar a seção Globals, falta apenas configurar as pastas que serão compartilhadas com as estações, através da seção "**Shares**".

Cada usuário válido cadastrado no sistema possui automaticamente um diretório home. Estas pastas ficam dentro do diretório /home e podem ser usadas para guardar arquivos pessoais, já que, a menos que seja estabelecido o contrário, um usuário não terá acesso à pasta pessoal do outro. Além dos diretórios home, você pode compartilhar mais pastas de uso geral. Para criar um compartilhamento, basta escrever seu nome no campo no topo da tela e clicar no botão "**Create Share**".



Depois de criado um compartilhamento, escolha-o na lista e clique no botão "**Choose Share**" para configurá-la. Você verá uma lista de opções, contendo campos para especificar usuários válidos e inválidos, usuários que podem ou não escrever no compartilhamento, nomes ou endereços de máquinas, entre outras opções.



O campo "**path**" é o mais importante, pois indica justamente qual pasta do sistema será compartilhada. O nome do compartilhamento diz apenas com que nome ele aparecerá no ambiente de rede, que não precisa necessariamente ser o mesmo nome da pasta. A opção "**comment**" permite que você escreva um breve comentário sobre a pasta que também poderá ser visualizado pelos usuários no ambiente de rede. Este comentário é apenas para orientação, não tem efeito algum sobre o compartilhamento.

A opção "**read only**" determina se a pasta ficará disponível apenas para leitura (opção **Yes**) ou se os usuários poderão também gravar arquivos (opção **No**). Você pode também determinar quais máquinas terão acesso ao compartilhamento através das opções "**Hosts Allow**" e "**Hosts Deny**". As configurações feitas aqui subscrevem as feitas na seção global. Se, por exemplo, a máquina 192.168.0.5 possui permissão para acessar o sistema, mas foi incluída na campo Hosts Deny do compartilhamento **programas**, ela poderá acessar outros compartilhamentos do sistema, mas não o compartilhamento **programas** especificamente.

A opção "**browsable**" permite configurar se o compartilhamento aparecerá entre os outros compartilhamentos do servidor no ambiente de rede, ou se será um compartilhamento oculto, que poderá ser acessado apenas por quem souber que ele existe. Isso tem uma função semelhante a colocar um "\$" em uma pasta compartilhada no Windows 98. Ela fica compartilhada, mas não aparece no ambiente de rede. Apenas usuários que saibam que o compartilhamento existe conseguirão acessá-lo. Esta opção tem efeito apenas sobre os clientes Windows, pois no Linux a maior parte dos programas clientes (como o Smb4k) mostra os compartilhamentos ocultos por padrão.

Finalmente, a opção "**available**" especifica se o compartilhamento está ativado ou não. Você pode desativar temporariamente um compartilhamento configurando esta opção como "**No**". Fazendo isso, ele continuará no sistema e você poderá torná-lo disponível quando quiser, alterando a opção para "**Yes**".

Um detalhe importante é que os usuários só terão permissão para acessar pastas que o login permite acessar. Por exemplo, no Linux o único usuário que pode acessar a pasta **/root** é o próprio root, ou outro autorizado por ele. Mesmo que você compartilhe a pasta root através do Samba, os demais usuários não poderão acessá-la.

Para editar as permissões de uma pasta, basta abrir o gerenciador de arquivos e, nas propriedades da pasta, acessar a guia "Permissões". As permissões podem ser dadas apenas ao usuário, para todos os usuários pertencentes ao grupo do usuário dono da pasta ou para todos os usuários. A opção "Aplicar mudanças a todas as subpastas e seus conteúdos" deve ficar marcada para que as permissões sejam aplicadas também às subpastas.

Terminadas as configurações, o servidor já irá aparecer no ambiente de rede, como se fosse um servidor Windows. Os compartilhamentos podem ser acessados de acordo com as permissões que tiverem sido configuradas, mapeados como unidades de rede, entre outros recursos.



Para compartilhar uma impressora já instalada na máquina Linux, o procedimento é o mesmo. Dentro do Swat, accese a seção **printers**, escolha a impressora a ser compartilhada (a lista mostrará todas as instaladas no sistema), configure a opção **available** como "yes" e ajuste as permissões de acesso, como vimos anteriormente.

No **Mandriva**, você pode instalar impressoras através do Control Center. No **Fedora** está disponível o "**system-config-printer**", que contém basicamente as mesmas funções. Em outras distribuições, você pode usar o **kaddprintewizard** ou a própria interface de

administração do Cups, que você acessa (via navegador) através da URL: <http://127.0.01:631>.

Se você não gostou do Swat, pode experimentar o **Pagode**, outra opção de configurador gráfico para o Samba, que pode ser baixado no: http://www.anahuac.biz/lesp/index.php?id_menu=24&tipo=3

O Pagode é um sistema desenvolvido em **PHP**, que roda sobre o **Apache**. Ele utiliza o **sudo** para permitir que o Apache execute o script como root, de forma a conseguir alterar os arquivos de configuração do Samba e reiniciar os serviços quando necessário.

Para instalá-lo você vai precisar das três coisas. Comece instalando um servidor Apache com suporte a PHP. Verifique em seguida se o pacote "**sudo**" está instalado. Ele vem instalado por padrão no Knoppix, Kurumin e outros live-CDs. Nas demais distribuições ele pode ser instalado usando o gerenciador de pacotes.

Depois de tudo pronto, baixe o arquivo de instalação do Pagode para dentro do diretório raiz do Apache (/var/www/), o que criará a pasta "pagode". Dentro dela existe um script de instalação que pode ser acessado através do navegador: http://127.0.0.1/pagode/install/check_instalation.php

Depois de instalado, você pode acessar o Pagode através do endereço <http://127.0.0.1/pagode/>

» Próximo: [Permitindo que os usuários compartilhem pastas](#)

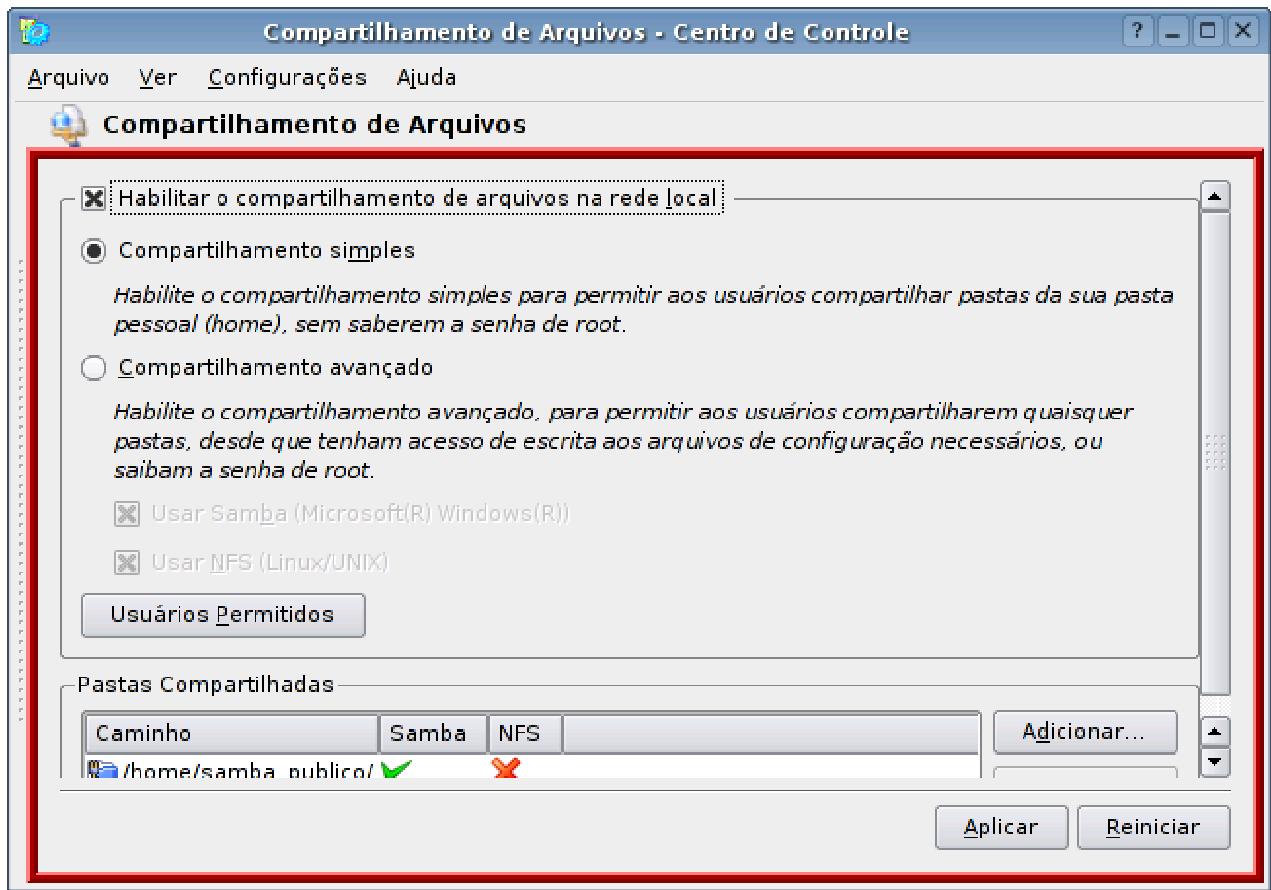
A configuração do Samba através do Swat é bem simples para configurar um servidor de arquivos, por exemplo, mas, e se você quiser permitir que os usuários também criem compartilhamentos, assim como no Windows? Não seria muito prático ter que ensiná-los a usar o Swat, sem falar que em muitos casos seria como dar uma arma na mão de uma criança.

O KDE possui um módulo que resolve este último problema, permitindo que os usuários compartilhem arquivos dentro dos seus respectivos diretórios de usuário de uma forma bastante simples, algo parecido com o que temos no Windows 98. Para que este recurso funcione, você deve instalar o módulo de compartilhamento de arquivos do Konqueror. No Debian, ele é fornecido pelo pacote "**kdenetwork-filesharing**", que pode ser instalado pelo apt-get. Em outras distribuições ele é incluído diretamente no pacote "**kdenetwork**", que precisa estar instalado.

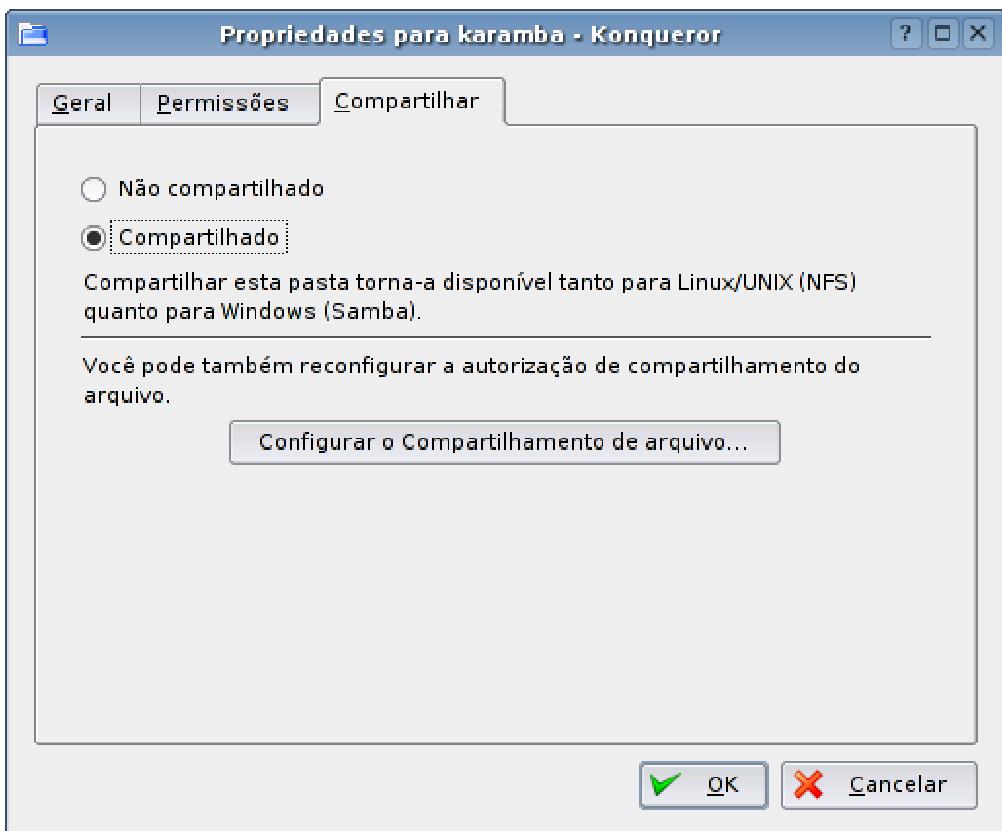
Como os usuários podem apenas compartilhar seus próprios arquivos, a possibilidade de danos ao sistema é pequena. Se você tiver um firewall isolando a sua rede local da internet, você poderá conviver com isso sem muitos sustos :-).

Dentro do Centro de Controle do KDE, acesse a seção "Internet & Rede > Compartilhamento de arquivos". Clique no "Modo administrador", forneça a senha de root e marque a opção "Compartilhamento simples (habilite o compartilhamento simples, para permitir que os usuários compartilhem pastas de sua pasta pessoal (home), sem saberem a senha de root.)".

No botão "Usuários permitidos" você tem a opção de autorizar todos os usuários (permitir a todos os usuários compartilhar pastas) ou autorizar apenas os usuários de um determinado grupo. Neste caso, use o "users-admin" ou outro programa de configuração de usuários e grupos para criar um novo grupo e adicionar os usuários desejados a ele.



A partir daí os usuários poderão compartilhar pastas simplesmente acessando a aba "Compartilhar", dentro das propriedades de cada uma.



Este compartilhamento do KDE faz, na verdade, um duplo compartilhamento. Além do Samba, os compartilhamentos ficam disponíveis na rede através do NFS, permitindo que você possa escolher qual protocolo prefere usar em cada caso. Lembre-se de que se você não quiser o compartilhamento via NFS, basta desativar (ou desinstalar) o serviço "nfs-kernel-server" (ou "nfs", nas distribuições derivadas do Red Hat). Naturalmente, para que o compartilhamento funcione, você deverá ter o servidor e o cliente Samba instalados no sistema e manter o serviço SMB ativo.

» Próximo: [Configurando manualmente o /etc/samba/smb.conf](#)

Toda a configuração do Samba, incluindo as configurações gerais do servidor, impressoras e todos os compartilhamentos, é feita em um único arquivo de configuração, o "["/etc/samba/smb.conf"](#)". Programas de configuração, como o Swat, simplesmente lêem este arquivo, "absorvem" as configurações atuais e depois geram o arquivo novamente com as alterações feitas. Isso permite que o Swat coexista com a edição manual do arquivo. Como o formato é bastante simples e conciso, muitas vezes é mais rápido e até mais simples editar diretamente o arquivo do que através do Swat. O único porém é que o Swat remove todos os seus comentários e formatação, deixando apenas as opções.

O smb.conf possui as mesmas seções mostradas no swat: global, homes, printers, etc. Ao instalar o Samba através do ícone mágico, é instalado um smb.conf já pré-configurado com

uma configuração de exemplo. A idéia é que o servidor já fique acessível imediatamente depois da instalação e você possa se concentrar em adicionar os usuários e compartilhamentos.

Para abri-lo, com privilégios de root, você pode digitar simplesmente "**kdesu kedit /etc/samba/smb.conf**" no terminal. Veja um exemplo do conteúdo do arquivo. Lembre-se de que as linhas iniciadas com # são comentários, não interferem na configuração:

```
# Arquivo de Configuração do Samba escrito para o Kurumin
# Por Carlos E. Morimoto

# Seção Globals:
# Aqui vão parâmetros gerais, como o nome da máquina e grupo de trabalho.

[global]
workgroup = KURUMIN
netbios name = %h server = lmhosts, host, (Samba wins, %v)
server string = %h
name resolve order = lmhosts, host, (Samba wins, %v)
printcap name = lpstat
encrypt passwords = Yes
wins support = yes
preferred master = yes
panic action = /usr/share/samba/panic-action %d
invalid users = root
preserve case = no
short preserve case = no
default case = lower
os level = 100

[homes]
comment = Home Directories
create mask = 0700
directory mask = 0700
browseable = No

[printers]
comment = Todas as Impressoras
path = /var/spool/samba
guest ok = yes
public = yes
printable = yes
browseable = yes
use client driver = yes

#
# Aqui vai a configuração das pastas compartilhadas. Você pode criar mais
# compartilhamentos usando o Swat ou editando diretamente este arquivo.
# Veja como funciona a configuração:
#
# [publico] : O nome do Compartilhamento, como aparecerá no ambiente de redes.
# path = /home/samba_publico : A pasta local que está sendo compartilhada.
# available = yes : O compartilhamento está disponível?
# Mudando para "available = no" ele ficará "congelado" e ninguém poderá acessar.
# browseable = yes : O compartilhamento aparecerá na rede?
# Mudando para "browseable = no" ele virará um compartilhamento oculto
# writable = yes : O compartilhamento fica disponível para leitura e escrita.
# writable = no : o compartilhamento fica disponível para somente leitura.

Compartilhamentos:
```

Agora é a sua vez:

```
#[compartilhamento]
# path = /pasta/pasta
# available = yes
# browseable = yes
# writable = yes
```

Se você quiser criar um novo compartilhamento, chamado "**arquivos**", que dá acesso à pasta "**/home/arquivos**" e pode ser acessado em modo somente-leitura por todos os usuários cadastrados no Samba, bastaria adicionar as linhas:

```
[arquivos]
path = /home/arquivos
available = yes
writable = no
```

Se você quiser permitir que o compartilhamento fique com permissão de escrita e leitura, mas fique acessível apenas pelos usuários "maria" e "joao" (os outros usuários não acessam nem para leitura), adicione a linha: "**valid users = joao maria**". A entrada ficaria:

```
[arquivos]
path = /home/arquivos
available = yes
writable = yes
valid users = maria, joao
```

Se preferir, você pode continuar permitindo que os outros acessem o compartilhamento para leitura e criar uma lista de escrita, contendo a maria e o joao:

```
[arquivos]
path = /home/arquivos
available = yes
writable = yes
write list = maria, joao
```

Outra forma de limitar o acesso é usar a opção "hosts allow" para permitir que apenas alguns endereços IP possam acessar os compartilhamentos, como em:

```
[arquivos]
path = /home/arquivos
available = yes
writable = yes
hosts allow = 192.168.0.2, 192.168.0.5
```

É possível ainda combinar as duas coisas, permitindo que apenas a maria e o joao acessem o compartilhamento e, ainda assim, só se estiverem usando uma das duas máquinas permitidas, como em:

```
[arquivos]
path = /home/arquivos
available = yes
writable = yes
write list = maria, joao
hosts allow = 192.168.0.2, 192.168.0.5
```

O Swat serve apenas como uma interface para a edição deste arquivo. Seja qual for o modo de configuração escolhido, basta fazer backups regulares deste arquivo para restaurar as configurações do servidor em caso de problemas. Além do arquivo `smb.conf`, salve também o arquivo `/etc/samba/smbpasswd`, que contém os usuários e senhas.

Sempre que alterar manualmente `smb.conf`, ou mesmo alterar algumas opções pelo Swat e quiser verificar se as configurações estão corretas, rode o `testparm` (basta chamá-lo no terminal). Ele funciona como uma espécie de debug, indicando erros grosseiros no arquivo. Depois de fazer qualquer alteração, reinicie o Samba usando o comando `/etc/init.d/samba restart` ou `"service smb restart"`. O comando `smbstatus` também é muito útil, pois permite verificar quais estações estão conectadas ao servidor e quais recursos estão sendo acessados no momento.

» Próximo: [Usando o Samba como controlador de domínio \(PDC\)](#)

Em uma pequena rede, manter as senhas dos usuários sincronizadas entre as estações Windows e o servidor Samba não chega a ser um grande problema. No entanto, em redes de maior porte, isso pode se tornar uma grande dor de cabeça e passar a consumir uma boa parte do seu tempo.

Para solucionar o problema, existe a opção de usar o servidor Samba como um controlador primário de domínio (PDC), onde ele passa a funcionar como um servidor de autenticação para os clientes Windows e (opcionalmente) armazena os perfis de cada usuário, permitindo que eles tenham acesso a seus arquivos e configurações a partir de qualquer máquina onde façam logon.

Nota: A Microsoft usa o termo "logon" (logar em) em toda documentação relacionada a redes Microsoft. Por isto adoto este termo dentro da configuração do PDC, substituindo o tempo "login" (logar no) que uso no restante do livro.

Ao cadastrar um novo usuário no servidor Samba, ele automaticamente pode fazer logon em qualquer uma das estações configuradas. Ao remover ou bloquear uma conta de acesso, o usuário é automaticamente bloqueado em todas as estações. Isso elimina o problema de sincronismo entre as senhas no servidor e nas estações e centraliza a administração de usuários e permissões de acesso no servidor, simplificando bastante seu trabalho de administração.

O primeiro passo é modificar o arquivo de configuração do Samba. Existem algumas regras adicionais para transformar o Samba em um controlador de domínio. A seção "global" deve conter as linhas `"domain master = yes"`, `"domain logons = yes"` e `"logon script = netlogon.bat"` e (importante) **não** deve conter a linha `"invalid users = root"`, pois precisaremos usar a conta de root no Samba ao configurar os clientes. É preciso ainda adicionar um compartilhamento chamado `"netlogon"`, que conterá o script de logon que será executado pelas estações.

Este é um exemplo de arquivo de configuração do Samba para um controlador de domínio. Ele não contém as configurações para compartilhamento de impressoras, que você pode

adicionar (juntamente com os compartilhamentos desejados) depois de testar a configuração básica:

```
[global]
workgroup = Dominio
netbios = GDH
server string = Samba PDC

domain master = yes
preferred master = yes
local master = yes
domain logons = yes
logon script = netlogon.bat

security = user
encrypt passwords = yes
os level = 100

[netlogon]
comment = Servico de Logon
path = /var/samba/netlogon
guest ok = Yes
browseable = No

[homes]
comment = Diretorio Home
valid users = %S
guest ok = Yes
browseable = No
```

Acostume-se a sempre rodar o comando "**testparm**" depois de fazer alterações no arquivo, pois ele verifica a sintaxe e indica erros de configuração. Ao configurar o Samba como PDC, ele deve exibir a mensagem: "Server role: ROLE_DOMAIN_PDC".

Depois de configurar o arquivo, verifique se a conta root do sistema foi cadastrada no Samba e se as senhas estão iguais. Caso necessário, use o comando "**smbpasswd -a root**" para cadastrar o root. Aproveite para criar a pasta "/var/samba/netlogon" e configurar corretamente as permissões:

```
# mkdir -p /var/samba/netlogon
# chmod 775 /var/samba/netlogon
```

Com o "775" estamos permitindo que, além do root, outros usuários que você adicionar no grupo possam alterar o conteúdo da pasta. Isso pode ser útil caso existam outros administradores de rede além de você.

Cadastre agora os logins dos usuários, com as senhas que eles utilizarão para fazer logon a partir das máquinas Windows. Neste caso, não é preciso se preocupar em manter as senhas em sincronismo entre o servidor e as estações. Na verdade, as contas que criamos aqui não precisam sequer existir nas estações, pois o login será feito no servidor. Para adicionar um usuário de teste "joao", use os comandos:

```
# adduser joao
# smbpasswd -a joao
```

É importante criar também a pasta "profile.pds" dentro do diretório home do usuário, onde o cliente Windows armazena as informações da sessão cada vez que o usuário faz logon no domínio:

```
# mkdir /home/joao/profile.pds
```

Ao rodar este comando como root, não se esqueça de ajustar as permissões da pasta, de forma que o usuário seja o dono:

```
# chown -R joao.joao /home/joao/profile.pds
```

Além das contas para cada usuário, é preciso cadastrar também uma conta (bloqueada, e por isso sem senha), para cada máquina. Você deve usar aqui os mesmos nomes usados na configuração de rede em cada cliente. Se a máquina se chama "athenas" por exemplo, é preciso criar um login de máquina com o mesmo nome:

```
# useradd -d /dev/null -s /bin/false athenas$  
# passwd -l athenas$  
# smbpasswd -a -m athenas
```

Note que nos dois primeiros comandos é adicionado um "\$" depois do nome, que indica que estamos criando uma conta de máquina, que não tem diretório home (-d /dev/null), não possui um shell válido (-s /bin/false) e está travada (passwd -l). Esta conta é válida apenas no Samba, onde é cadastrada com a opção "-m" (machine). Estas contas de máquina são chamadas de "trusted accounts" ou "trustee".

Lembre-se que para usar este comando o arquivo "/etc/shells" deve conter a linha "/bin/false". Se preferir, você pode adicionar as contas de máquina dentro de um grupo do sistema ("maquinas" ou "machines" por exemplo). Neste caso, crie o grupo usando o comando "groupadd" e use o comando abaixo para criar as contas de máquina já incluindo-as no grupo:

```
# useradd -g maquinas -d /dev/null -s /bin/false athenas$
```

Por último, é necessário criar o arquivo "/var/samba/netlogon/netlogon.bat", um script que é lido e executado pelos clientes ao fazer logon. Você pode fazer muitas coisas através dele, mas um exemplo de arquivo funcional é:

```
net use h: /HOME  
net use x: \\gdh\arquivos /yes
```

Este script faz com que a pasta home de cada usuário (compartilhada pelo Samba através da seção "homes") seja automaticamente mapeada como a unidade "H:" no cliente, o que pode ser bastante útil para backups, por exemplo. Naturalmente, cada usuário tem acesso apenas a seu próprio home.

A segunda linha é um exemplo de como fazer com que determinados compartilhamentos do servidor sejam mapeados no cliente. O "net use x: \\gdh\arquivos /yes" faz com que o compartilhamento "arquivos" (que precisaria ser configurado no smb.conf) seja mapeado

como o drive "X:" nos clientes. Lembre-se que o "gdh" dentro do netlogon.bat deve ser substituído pelo nome do seu servidor Samba, configurado na opção "netbios name =" do smb.conf.

Mais um detalhe importante é que o arquivo do script de logon deve usar quebras de linhas no padrão MS-DOS e não no padrão Unix (que é o padrão da maioria dos editores de texto do Linux). Você pode criá-lo usando um editor de texto do Windows ou usar algum editor do Linux que ofereça esta opção. No Kwrite por exemplo, a opção está em: "Configurar > Configurar Editor > Abrir/Salvar > Fim de linha > DOS/Windows".

Mais uma configuração útil (porém opcional) é fazer com que o servidor armazene os arquivos e configurações do usuário (recurso chamado **Roaming Profiles**, ou perfis móveis), fornecendo-os à estação no momento em que o usuário faz logon. Isso permite que o usuário possa trabalhar em outras máquinas da rede e faz com que seus arquivos de trabalho sejam armazenados no servidor, diminuindo a possibilidade de perda de dados.

Por outro lado, ativar os perfis móveis faz com que seja consumido mais espaço de armazenamento do servidor e aumenta o tráfego da rede, já que os arquivos precisam ser transferidos para a estação a cada logon. Isso pode tornar-se um problema caso os usuários da rede tenham o hábito de salvar muitos arquivos grandes na área de trabalho.

Note que o servidor não armazena todos os arquivos do usuário, apenas as configurações dos aplicativos, entradas do menu iniciar, cookies, bookmarks e arquivos temporários do IE e o conteúdo das pastas Desktop, Modelos e Meus Documentos.

Para ativar o suporte no Samba, adicione as duas linhas abaixo no final da seção "global" do smb.conf (abaixo da linha "logon script = netlogon.bat"):

```
logon          home           =      \\\\%L\\\\%U\\\\profiles  
logon path = \\\\%L\\\\profiles\\\\%U
```

A variável "%L" neste caso indica o nome do servidor e o "%U" o nome do usuário que está fazendo logon. Quando, por exemplo, o "joao" faz logon é montado o compartilhamento "\\\gdh\\profiles\\joao". Adicione também um novo compartilhamento, adicionando as linhas abaixo no final do arquivo:

```
[profiles]  
path          =      /var/profiles  
writeable     =      Yes  
browseable    =      No  
create        mask    =      0600  
directory mask = 0700
```

Crie a pasta "/var/profiles", com permissão de escrita para todos os usuários:

```
#          mkdir      /var/profiles  
# chmod 1777 /var/profiles
```

Cada usuário passa a ter uma pasta pessoal dentro da pasta ("/var/profiles/joao", por exemplo) onde as configurações são salvas. Apesar das permissões locais da pasta permitirem que qualquer usuário a acesse, o Samba se encarrega de permitir que cada usuário remoto tenha acesso apenas ao seu próprio profile.

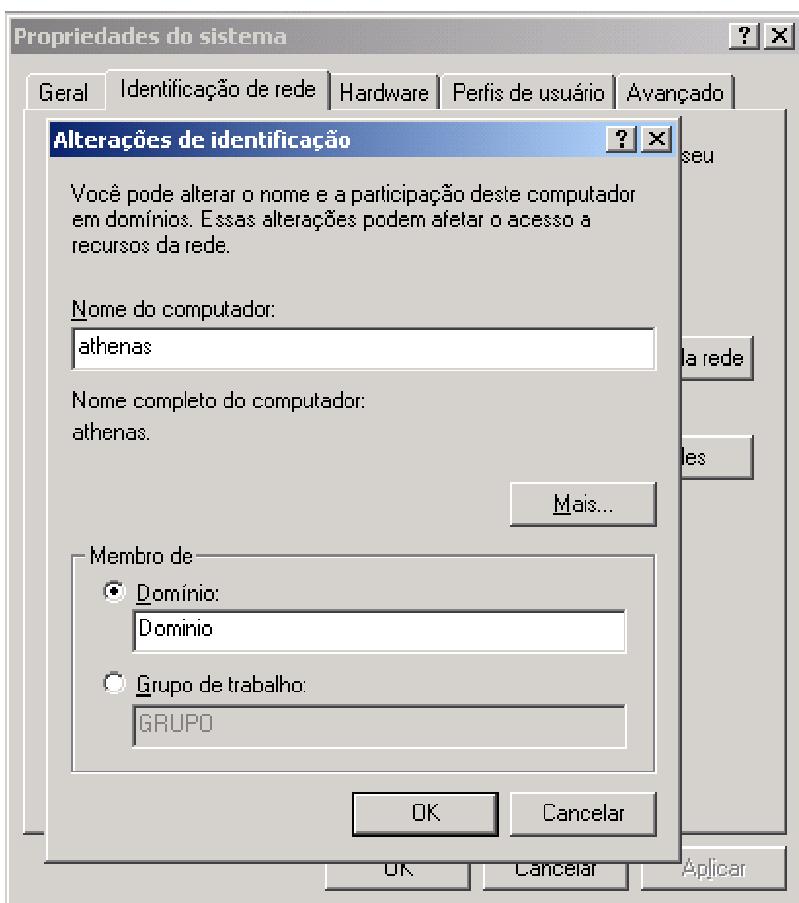
As estações Windows 2000 e Windows XP utilizam os perfis móveis automaticamente, quando o recurso está disponível no servidor Samba. Você pode verificar a configuração e, caso desejado, desativar o uso do perfil móvel no cliente no "Meu Computador > Propriedades > Perfis de Usuário > Alterar tipo".

» Próximo: [Logando clientes Windows](#)

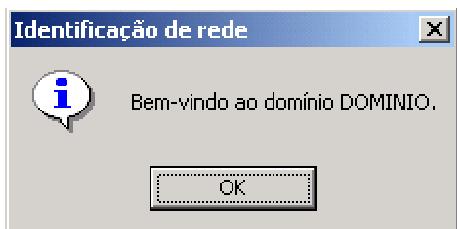
Neste ponto, a configuração do servidor Samba está pronta. Falta apenas configurar os clientes Windows para efetuarem logon no domínio. Nem todas as versões do Windows suportam este recurso. Como controladores de domínio são usados principalmente em redes de médio ou grande porte em empresas, a Microsoft não inclui suporte no Windows XP Home e no XP Starter (também chamado jocosamente de "Miserable Edition"), de forma a pressionar as empresas a comprarem o XP Professional, que é mais caro.

A configuração muda de acordo com a versão do Windows:

- No **Windows 2000**, acesse o "Meu Computador > Propriedades > Identificação de rede > Propriedades". Coloque aqui o nome do computador (que precisa ser um dos logins de máquinas adicionados na configuração do Samba) e o nome do Domínio, que é definido na opção "workgroup =" do smb.conf. Para ter acesso a esta opção você deve estar logado como administrador.



Na tela de identificação que será aberta a seguir, logue-se como "root", com a senha definida no Samba. É normal que a conexão inicial demore dois ou três minutos. Se tudo der certo, você é saudado com a mensagem "Bem-vindo ao domínio DOMINIO".



É necessário identificar-se como root ao fazer a configuração inicial, para que seja criada a relação de confiança entre o servidor e o cliente. A partir daí aparece a opção opção "Efetuar logon em: DOMINIO" na tela de login, permitindo que o usuário faça logon usando qualquer uma das contas cadastradas no servidor. Continua disponível também a opção de fazer um login local.

- No **Windows 98 ou ME**: Comece logando-se na rede (na tela de login aberta na inicialização) com o mesmo usuário e senha que será usado para fazer logon no domínio. Acesse agora o "Painel de Controle > Redes > Cliente para redes Microsoft > Propriedades". Marque a opção "Efetuar Logon num domínio NT", informe o nome do

domínio e marque a opção "Efetuar logon e restaurar conexões". Ao terminar, é preciso fornecer o CD de instalação e reiniciar a máquina.

Note que as máquinas com o Windows 98/ME não são compatíveis com todos os recursos do domínio, elas acessam o domínio dentro de uma espécie de modo de compatibilidade, onde podem acessar os compartilhamentos, mas não têm acesso ao recurso de perfis móveis, por exemplo.

- No **Windows XP Professional** o procedimento varia de acordo com a versão do Samba usada. Se você está usando uma versão recente do Samba, da versão 3.0 em diante, a configuração é muito simples, basta seguir os mesmos passos da configuração no Windows 2000.

Se, por outro lado, você ainda está usando o Samba 2.x, a configuração é um pouco mais complicada. Comece copiando o arquivo "/usr/share/doc/samba-doc/registry/WinXP_SignOrSeal.reg" (do servidor), que fica disponível como parte da instalação do pacote "samba-doc". Esta é uma chave de registro que precisa ser instalada no cliente.

Acesse agora as propriedades do "Meu Computador" e na aba "Nome do Computador" clique no botão "ID de rede". Será aberto um Wizard que coleta o nome do domínio, nome da máquina e login de usuário. Lembre-se que é necessário efetuar o primeiro logon como root.

Se não der certo da primeira vez, acesse o "Painel de controle > Ferramentas administrativas > Diretiva de segurança local > Diretivas locais > Opções de segurança" e desative as seguintes opções:

- * Membro do domínio: criptografar ou assinar digitalmente os dados de canal seguro (sempre)
- * Membro do domínio: desativar alterações de senha de conta da máquina
- * Membro do domínio: requer uma chave de sessão de alta segurança (Windows 2000 ou posterior)

Para confirmar se os clientes estão realmente efetuando logon no servidor, use o comando "**smbstatus**" (no servidor). Ele retorna uma lista dos usuários e máquinas logadas, como em:

Samba	version	3.0.14a-Debian	
PID	Username	Group	Machine
<hr/>			
4363	joao joao athenas (192.168.0.34)		
<hr/>			
Service	pid	machine	Connected at
<hr/>			
joao	4363	athenas	Sat Jul 9 10:37:09 2005

» Próximo: [Logando clientes Linux](#)

Além de autenticar as máquinas Windows, o servidor Samba PDC pode ser usado para logar também os clientes Linux, centralizando a autenticação de toda a rede. Fazer uma máquina Linux se logar no PDC é mais complicado do que uma máquina Windows, pois temos que fazer várias alterações que alteram a forma como sistema autentica os usuários. Ao invés de verificar os arquivos "/etc/passwd" e "/etc/shadow", onde ficam armazenadas as contas locais, o cliente passa a utilizar o Samba e o Winbind (que permite que uma máquina Linux ingresse no domínio) para buscar os logins no servidor.

Esta configuração é indicada para distribuições derivadas do Debian que utilizam o KDM, com destaque para o Kurumin, ideal para situações em que você usa o Kurumin nos desktops da empresa e quer usar a lista de logins de um servidor Samba, ao invés de logins locais. Ela funciona em outras distribuições, mas eventualmente podem ser necessárias pequenas mudanças, de acordo com as peculiaridades de cada uma.

O primeiro passo é instalar os pacotes "**samba**" (ou samba-server), "**winbind**" (ou samba-winbind) e "**libpam-modules**" em cada cliente. Nas distribuições derivadas do Debian, instale diretamente os três pacotes:

```
# apt-get install samba winbind libpam-modules
```

No Fedora, o winbind está incluído no pacote principal do Samba e os módulos do PAM são instalados através do pacote "pam_smb":

```
# yum install samba pam_smb
```

A configuração no servidor não muda em relação ao que já vimos. Toda a configuração que vemos aqui é feita nos clientes. Abra agora o arquivo "**/etc/samba/smb.conf**" (no cliente Linux) e faça com que a seção Global fique como o exemplo. Você pode tanto adicionar compartilhamentos, quanto ficar apenas com esta configuração básica:

[global]						
workgroup			=			
netbios				=		
winbind	use	name	default		domain	
obey	pam			restrictions		=
security			=			
encrypt		passwords				
wins		server				true
winbind		uid		=		192.168.1.1
winbind		gid		=		10000-20000
template		shell		=		10000-20000
template		homedir		=		/bin/bash
winbind		separator				/home/%U
printing						+
invalid users = root						cups

Não se esqueça de substituir o "Dominio" pelo nome do domínio usado na rede, o "cliente1" pelo nome do cliente e o 192.168.1.1" pelo endereço IP do servidor Samba PDC.

Abra agora o arquivo "**/etc/nsswitch.conf**" e substitua as linhas:

passwd:		compat
group:		compat
shadow: compat		

... no início do arquivo, por:

passwd:		compat	winbind
group:		compat	winbind
shadow: compat	winbind		

Um exemplo do arquivo completo é:

passwd:		compat	winbind
group:		compat	winbind
shadow: compat	winbind		
hosts:	files	dns	mdns
networks:	files		
protocols:		db	files
services:		db	files
ethers:		db	files
rpclists:		db	files
netgroup: nis			

Depois de modificar os dois arquivos, reinicie o Samba e o Winbind e teste a configuração, ingressando no domínio. Para isso, use o comando "net rpc join":

net rpc join member -U root

Password:
Joined domain DOMINIO.

A senha solicitada é a senha de root do servidor PDC, cadastrada no Samba, assim como fazemos ao cadastrar as máquinas Windows. Em caso de problemas, você pode usar também o comando abaixo, que especifica o nome do servidor (-S) e o nome do domínio (-w):

net rpc join -S gdh -w dominio -U root

Se você receber uma mensagem de erro, como:

Creation	of	workstation	account	failed
Unable to join domain DOMINIO.				

... provavelmente você esqueceu de cadastrar a máquina cliente no servidor. O nome da máquina (que você verifica através do comando "hostname") deve ser o mesmo que o incluído no arquivo smb.conf. Para criar a conta de máquina para o cliente, use (no servidor) os comandos que vimos anteriormente:

```
#      useradd      -d      /dev/null      -s      /bin/false      cliente1$  
#          passwd      -l      cliente1$  
# smbpasswd -a -m cliente1
```

Neste ponto o cliente já está logado no domínio. Esta configuração é permanente, de forma que você não precisa se preocupar em refazer a configuração a cada boot.

Falta agora a parte mais problemática, que é configurar o PAM, o sistema de autenticação do sistema, para buscar os logins no servidor. Isso é feito modificando os arquivos "/etc/pam.d/login" e "/etc/pam.d/kdm".

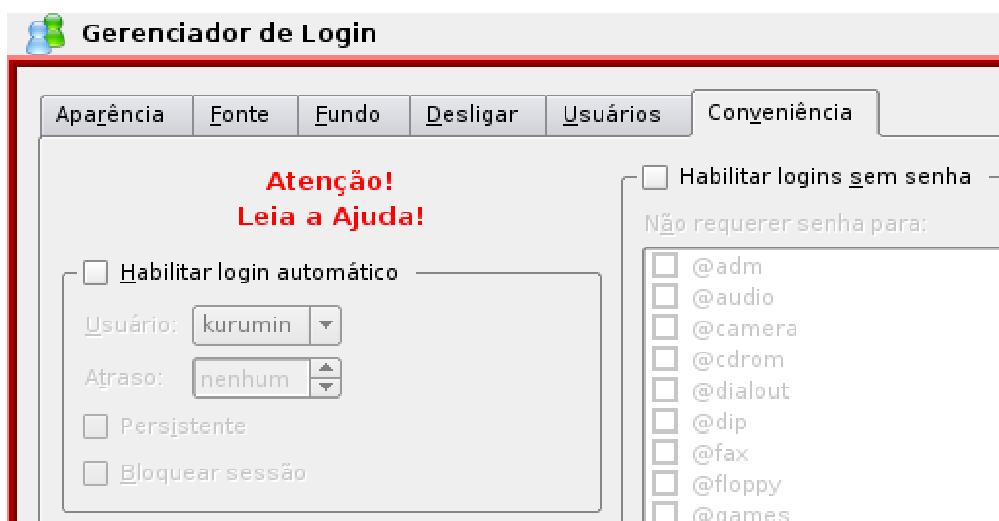
Comece adicionando as linhas abaixo no **íncio** do arquivo "**/etc/pam.d/login**" (responsável pela autenticação dos usuários no sistema), sem apagar as demais:

session	required	pam_mkhomedir.so	skel=/etc/skel	umask=0022
session		optional	pam_mount.so	
auth		sufficient	pam_winbind.so	
account		sufficient	pam_winbind.so	
session required pam_winbind.so				

Abra agora o arquivo "**/etc/pam.d/kdm**", deixando o arquivo com o seguinte conteúdo (apague ou comente as demais linhas). A mesma configuração pode ser usada no arquivo "**/etc/pam.d/gdm**", usado por distribuições que trazem o Gnome por padrão:

auth	required		/lib/security/pam_securetty.so
auth	required		/lib/security/pam_nologin.so
auth	sufficient		/lib/security/pam_winbind.so
auth required /lib/security/pam_pwdb.so		use_first_pass	shadow nullok
account required			/lib/security/pam_winbind.so
session required /lib/security/pam_mkhomedir.so skel=/etc/skel umask=0022			

Esta configuração faz com que o KDM exiba a lista de usuários cadastrados no servidor e permita que você faça login diretamente no domínio, sem passar pela autenticação local. É importante também desativar o autologin do KDE (ainda no cliente), no Centro de Controle do KDE > Administração do Sistema > Gerenciador de login.



Se você apenas adicionar as linhas acima no "/etc/pam.d/kdm", mas não apagar as linhas que já existem no arquivo (que permitem a autenticação local), a tela do KDM vai exibir a lista de logins do servidor, mas vai recusar o login, dizendo que a senha está incorreta. Este é um dos erros de configuração mais comuns :).

Se você deixar disponível a opção "Bloquear sessão" do KDE, vai precisar editar também o arquivo "**/etc/pam.d/kscreensaver**", para que ele também use as contas do servidor. Caso contrário, o usuário vai acabar tendo que reiniciar o X, cada vez que clicar por engano no ícone.



Adicione as duas linhas abaixo no início do arquivo (*/etc/pam.d/kscreensaver*), sem apagar as demais:

```
auth           sufficient          pam_winbind.so
auth required pam_unix.so shadow nullok
```

Para que esta configuração funcione, é importante que os usuários sejam cadastrados no servidor como usuários reais, usando o comando "adduser", e não o "adduser --disabled-login --no-create-home" ou similar. Basicamente, é preciso que o usuário possa se logar no servidor, caso contrário também não vai conseguir se logar nas estações.

No cliente, acesse a pasta "**/etc/rc5.d**" e verifique se os links responsáveis por inicializar os serviços **samba**, **winbind** e **kdm** foram criados corretamente. Eles precisam ser carregados nessa ordem. No caso de distribuições que inicializam o KDM primeiro (como no caso do Kurumin), renomeie o link, de forma que ele seja inicializado por último, como em:

```
# mv /etc/rc5.d/S02kdm /etc/rc5.d/S99kdm
```

Reinic peace o cliente, para que os módulos do PAM sejam atualizados e os serviços inicializados na ordem correta. Você notará que a tela de login do KDM passará a exibir os usuários cadastrados no servidor, ao invés dos usuários locais, sinalizando de que está tudo funcionando.



Configurando desta forma, os usuários locais que forem eventualmente criados no terminal chegam a aparecer na lista, mas não é possível fazer login neles através do KDM (essa é justamente a idéia). Apesar disso, você pode se logar nos terminais remotamente (usando o root e outros logins locais) via SSH, quando precisar alterar as configurações.

No arquivo "/etc/pam.d/login", incluímos a linha "session required pam_mkhomedir.so skel=/etc/skel umask=0022". Ela faz com que a pasta "/etc/skel" (da estação) seja usada como um template para a criação dos diretórios home dos usuários que só existem no servidor. A pasta "/home" (na estação) armazena apenas os arquivos que forem alterados em relação à pasta "/etc/skel", simplificando os backups. Você pode configurar o servidor Samba instalado em cada estação para compartilhar o diretório home, com permissões de acesso apenas para o administrador da rede, de forma que você possa acessar o home de cada estação a partir do servidor e fazer backup periodicamente.

O "/etc/skel" é justamente uma pasta modelo, cujo conteúdo é copiado para o diretório home, sempre que um novo usuário é criado. As configurações padrão mudam muito de distribuição para distribuição. Esta configuração privilegia o uso das configurações padrão de cada distribuição, permitindo que você use diversas distribuições diferentes nos clientes, independentemente de qual esteja usando no servidor. O Fedora continua com cara de Fedora, o Slackware de Slackware, e assim por diante.

» Próximo: [Usando o NFS](#)

Enquanto o Samba permite solucionar sem muita dor de cabeça o desafio de interligar máquinas Linux e Windows na mesma rede, o NFS é uma opção para compartilhar sistemas de arquivos entre máquinas Linux, de uma forma prática e estável. Na verdade,

você pode perfeitamente usar o Samba para compartilhar arquivos entre máquinas Linux, mas o NFS não deixa de ser um recurso importante, que você não deve deixar de estudar.

Assim como o Samba, o NFS é um servidor que precisa ser habilitado manualmente na maior parte das distribuições. No Mandriva, Fedora e outras distribuições derivadas do Red Hat, procure pelo serviço "**nfs**". Nas distribuições derivadas do Debian, procure pelo serviço "**nfs-kernel-server**".

O NFS utiliza um outro serviço, o portmap, para gerenciar as requisições dos clientes. Este serviço precisa estar ativo para que o NFS funcione, ou seja, para inicializar o servidor NFS, você precisa ativar os dois:

```
#           /etc/init.d/portmap          start
#           /etc/init.d/nfs-common       start
#           /etc/init.d/nfs-kernel-server start
(ou simplesmente "service portmap start; service nfs start", no Fedora)
```

A configuração do NFS é feita em um único arquivo, o "**/etc/exports**", onde vai a configuração dos diretórios compartilhados, um por linha. Originalmente, este arquivo fica vazio, ou contém apenas um comentário. Você precisa apenas abri-lo num editor de textos e adicionar as pastas que deseja compartilhar. Por exemplo, para compartilhar a pasta "/home/arquivos" como somente leitura, para todos os micros da sua rede local, adicione a linha:

```
/home/arquivos 192.168.0.*(ro)
```

Para compartilhar a pasta "/home/trabalhos" com permissão de leitura e escrita, adicione a linha:

```
/home/trabalhos 192.168.0.*(rw)
```

Para compartilhar a pasta "/arquivos", de forma que apenas o micro 192.168.0.3 possa acessar:

```
/arquivos 192.168.0.3(rw)
```

Outra opção, útil em redes locais, é a "async", que permite que o NFS transfira arquivos de forma assíncrona, sem precisar esperar pela resposta do cliente a cada pacote enviado. Sem a opção async, a taxa de transmissão em uma rede de 100 megabits fica, em geral, em torno de 6 a 7 MB/s, enquanto que, ao ativá-la, sobe para até 11 MB/s, ficando limitada apenas à velocidade da rede e dos HDs no servidor e cliente.

Ao adicioná-la, a linha de compartilhamento ficaria:

```
/home/trabalhos 192.168.0.*(rw,async)
```

Você pode usar, ainda, o parâmetro "**noaccess**", que permite que você compartilhe apenas os arquivos dentro do diretório, mas não subdiretórios que eventualmente estejam presentes.

Depois de incluir todos os diretórios que deseja compartilhar, com suas respectivas permissões de acesso, salve o arquivo e reinicie o serviço **nfs** para que as alterações surtam efeito. Para isso, use o comando:

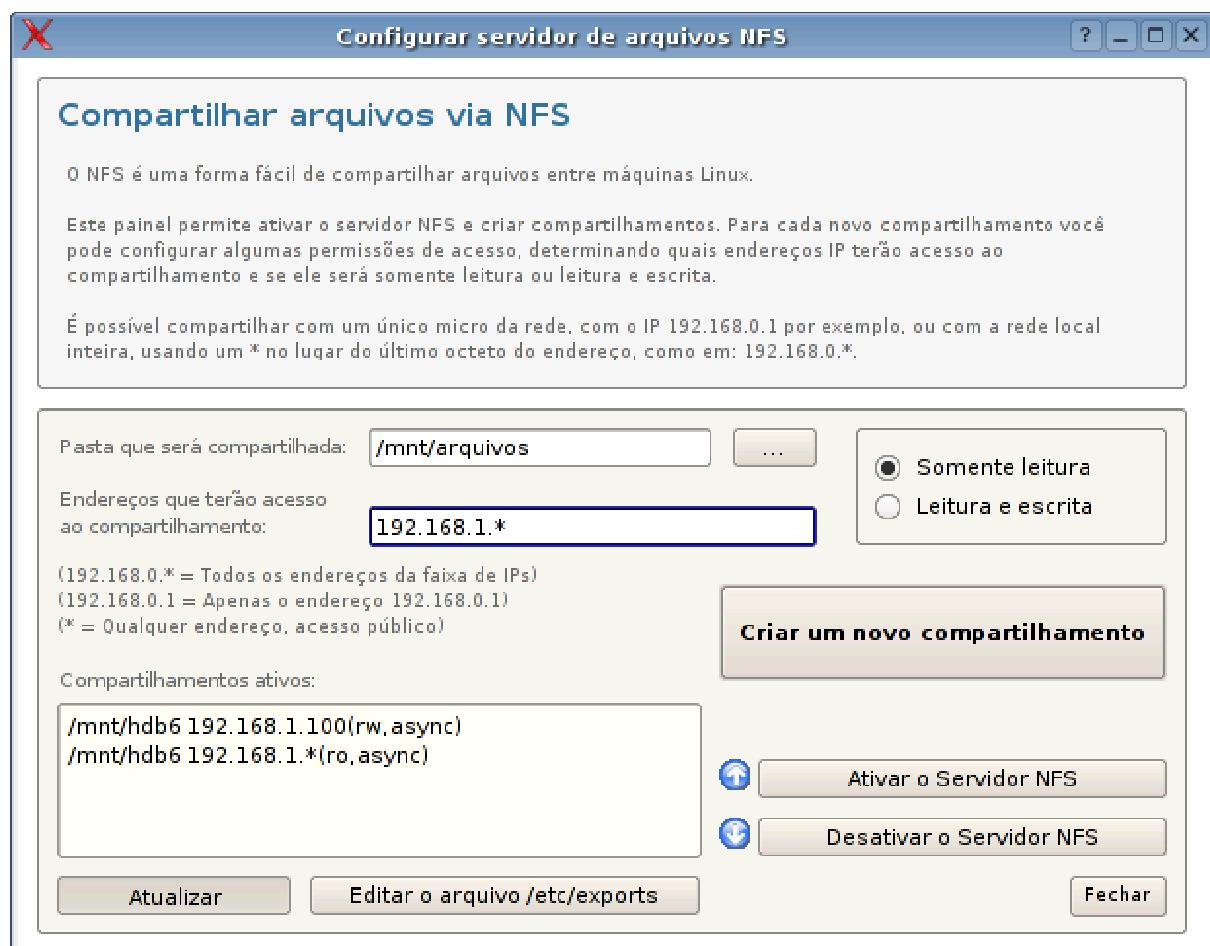
```
# /etc/init.d/nfs-kernel-server restart
```

Sempre que desejar parar o serviço, você pode usar os comandos abaixo, que respectivamente param e inicializam o serviço:

#	/etc/init.d/nfs-kernel-server	stop
#	/etc/init.d/nfs-kernel-server	start

(note que no Fedora o serviço não se chama "nfs-kernel-server", mas apenas "nfs").

Embora seja fácil editar diretamente o arquivo "/etc/exports", muitas distribuições incluem ferramentas gráficas para gerenciar os compartilhamentos NFS. O Fedora, por exemplo, inclui o "**system-config-nfs**" (que se chama "redhat-config-nfs" no Red Hat). O Mandriva inclui um utilitário similar dentro do Mandriva Control Center, enquanto que no Kurumin você encontra um painel de configuração no Iniciar > Redes e Acesso Remoto > NFS.



Ao compartilhar os diretórios, resolvemos apenas metade do problema. Ainda falta acessá-los a partir dos clientes. Assim como no caso das partições, você pode montar os compartilhamentos NFS em qualquer pasta vazia. Muitas empresas utilizam compartilhamentos montados no diretório /home (das estações) para que os arquivos gerados pelos usuários (e armazenados no home) sejam armazenados no compartilhamento do servidor, facilitando os backups, por exemplo.

Caso você monte o compartilhamento em uma pasta que contenha arquivos, estes ficarão momentaneamente inacessíveis, dando lugar aos do compartilhamento. Contudo, depois que o compartilhamento é desativado, eles reaparecem. Nada é perdido.

Para montar o compartilhamento manualmente, use (como root) o comando:

```
#           mkdir          /mnt/arquivos  
# mount -t nfs 192.168.0.1:/arquivos /mnt/arquivos
```

Aqui eu comecei criando a pasta "/mnt/arquivos", onde vou montar o compartilhamento. A linha de montagem propriamente dita inclui o sistema de arquivos usado, neste caso o nfs (-t nfs), o endereço IP do servidor, seguido da pasta que ele está compartilhando e, finalmente, a pasta local onde os arquivos ficarão acessíveis.

Ao terminar de acessar o compartilhamento, ou caso precise desligar o servidor, use o comando "**umount /mnt/arquivos**" (no cliente) para desmontá-lo. É importante desmontar o compartilhamento antes de desligar o servidor, do contrário, o cliente continua tentando acessar o compartilhamento sempre que você acessa a pasta onde ele está montado, o que faz com que os gerenciadores de arquivos e outros programas "parem" ao passar pela pasta, aguardando a resposta do servidor que não está mais lá.

Se você acessa o compartilhamento freqüentemente, pode ganhar tempo inserindo uma entrada referente a ele no arquivo "**/etc/fstab**". Assim você pode montar o compartilhamento usando o comando simplificado, ou configurar o sistema para montá-lo automaticamente durante o boot. Basta incluir a linha no final do arquivo, deixando sempre uma linha em branco após ela. A linha para o compartilhamento que acabamos de montar seria:

```
192.168.0.1:/arquivos /mnt/arquivos nfs noauto,users,exec 0 0
```

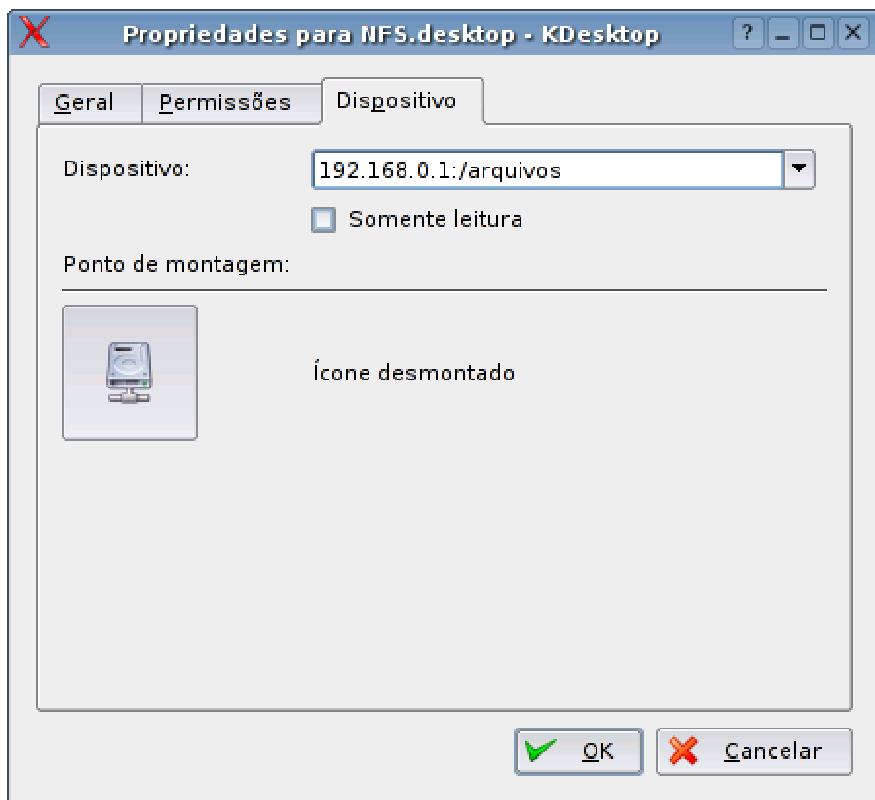
Neste exemplo, o "192.168.0.1:/arquivos" é o IP do servidor, seguido pela pasta compartilhada, o "/mnt/arquivos" é a pasta local onde este compartilhamento ficará acessível e o "nfs" é o sistema de arquivos; os mesmos parâmetros que usamos no comando manual.

O "noauto" faz com que o compartilhamento não seja montado automaticamente durante o boot. Você pode montá-lo e desmontá-lo conforme for utilizá-lo, usando os comandos "**mount /mnt/arquivos**" e "**umount /mnt/arquivos**". Note que graças à entrada no fstab, você agora precisa especificar apenas a pasta, pois o sistema lê os outros parâmetros a partir da entrada no arquivo.

O parâmetro "**users**" permite que você monte e desmonte o compartilhamento usando seu login normal, sem precisar usar o root e o "**exec**", que permite executar programas dentro do compartilhamento. Caso você esteja preocupado com a segurança, pode remover as duas opções.

Você pode facilitar o acesso ao compartilhamento adicionando um ícone para ele no desktop do KDE. Para isso, clique com o botão direito sobre uma área vazia e acesse a opção: "Criar novo > Dispositivo > NFS".

Na janela que se abre, acesse a aba "Dispositivo" e aponte a entrada que foi adicionada ao fstab. A partir daí você monta o compartilhamento clicando sobre o ícone, e pode desmontá-lo clicando com o botão direito e usando a opção "desmontar".



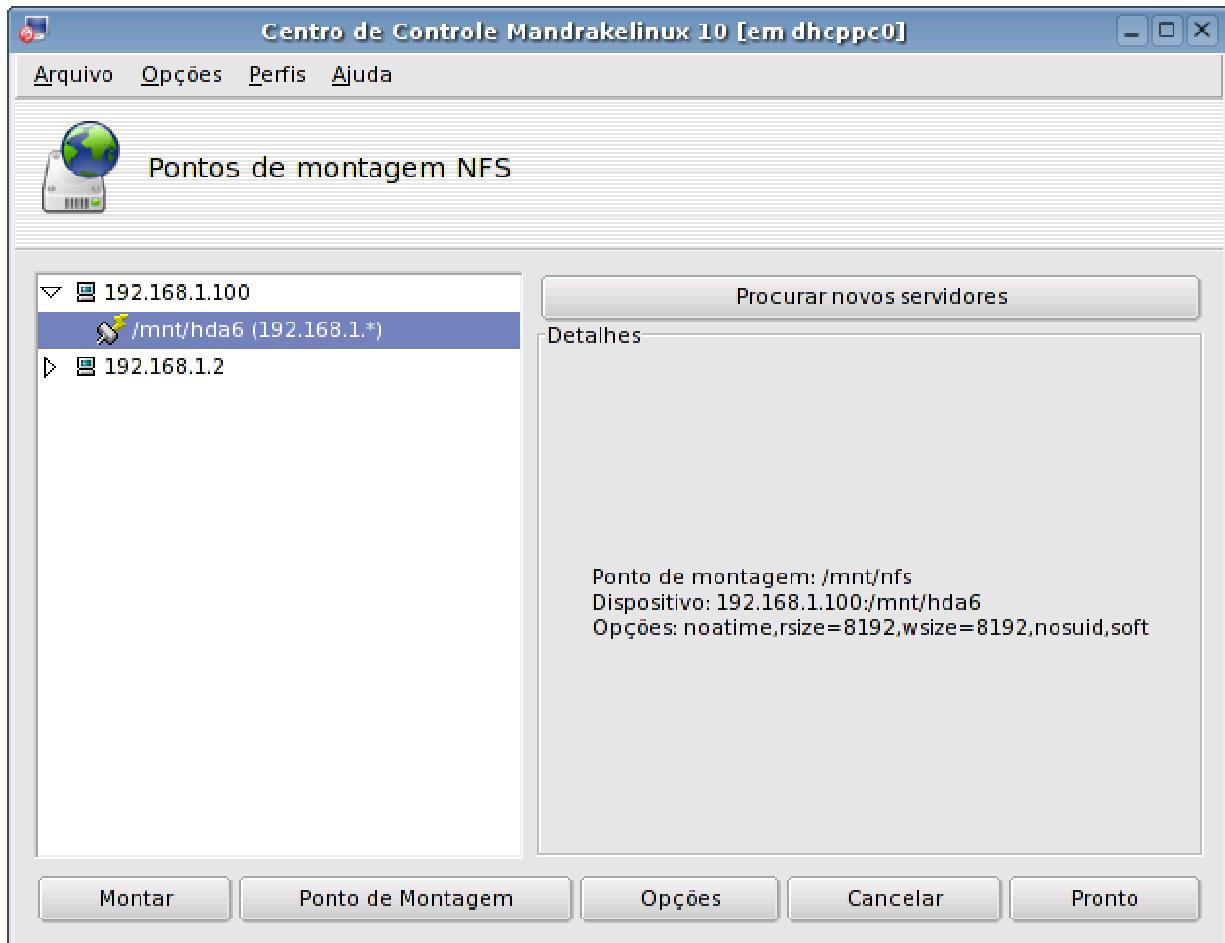
Você pode incluir várias linhas, se desejar montar vários compartilhamentos. Caso o servidor fique sempre ligado e você queira que o compartilhamento seja montado automaticamente durante o boot, retire o "noauto". Neste caso, a linha ficaria:

```
192.168.0.1:/arquivos /mnt/arquivos nfs users,exec 0 0
```

Novamente, este é o procedimento manual, muitas distribuições incluem utilitários gráficos para facilitar isso. No Mandriva, por exemplo, você encontra um utilitário de montagem no Centro de Controle (mcc), em "Pontos de Montagem > Pontos de montagem NFS".

Nele você clica no "Servidores de busca" para ver uma lista dos compartilhamentos disponíveis na rede. Ao escolher um compartilhamento, clique no "Ponto de montagem"

para definir a pasta local onde ele será acessado e configure as opções adicionais (como o "noauto" e "user") em "Opções". Depois de terminar, clique no "Pronto" e ele pergunta se você quer salvar a configuração no "/etc/fstab".



Mais um comando útil ao utilizar o NFS é o "**showmount -a**" (só funciona se dado pelo root) que mostra uma lista com os diretórios NFS compartilhados na sua máquina que foram acessados e quais máquinas os acessaram desde o último reboot. Não é muito específico, pois não mostra datas nem horários, mas pelo menos permite descobrir se alguém não autorizado está acessando os compartilhamentos.

» Próximo: [Mais opções](#)

Por padrão, os compartilhamentos do NFS são montados com a opção "**hard**". Isso causa um certo transtorno quando o servidor é desligado ou desconectado da rede, pois os clientes ficam tentando se reconectar ao servidor indefinidamente, fazendo que programas travem ao tentar acessar ou salvar arquivos no compartilhamento e você não consiga desmontá-lo por vias normais até que o servidor volte.

Para prevenir este problema, você pode montar os compartilhamentos (nos clientes) usando a opção "soft". Neste caso, o compartilhamento é escondido caso o servidor seja desconectado e programas tentando acessá-lo passam a exibir mensagens de "não é possível ler o arquivo", ao invés de travarem. Para usar esta opção, adicione a opção "-o soft" no comando de montagem:

```
# mount -t nfs -o soft 192.168.0.1:/home/morimoto/arquivos /mnt/arquivos
```

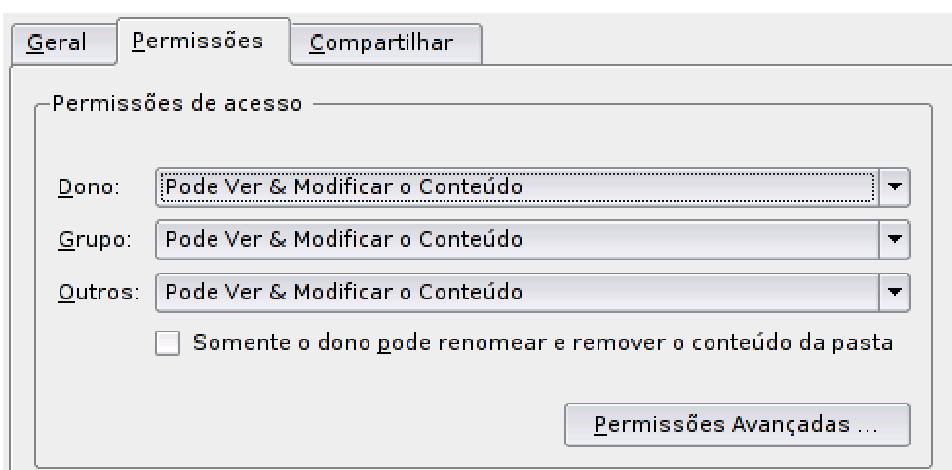
A linha no "/etc/fstab", com a opção, ficaria:

```
192.168.0.1:/home/morimoto/arquivos /mnt/arquivos nfs users,exec,soft 0 0
```

Outra questão importante ao usar o NFS é a questão das permissões de acesso. O servidor NFS "confia" na estação e permite que o usuário logado nela accesse os arquivos no compartilhamento, com as mesmas permissões que teria um usuário local de mesmo nome. Por exemplo, imagine que compartilhei a pasta "/home/morimoto/arquivos". Por estar dentro da pasta "/home/morimoto", os arquivos da pasta podem ser lidos e alterados pelo usuário "morimoto", mas apenas lidos pelos demais usuários do sistema.

O compartilhamento é, então, montado no host 192.168.0.4, usado pela usuária "maria". Não existe na minha máquina nenhuma conta de usuário chamada "maria", apenas o "morimoto" e o "root". Por isso, a "maria" accessa o compartilhamento na minha máquina restrita às permissões de acesso da pasta para outros usuários (que podem apenas ler, mas não alterar os arquivos). Neste caso, mesmo que a pasta seja compartilhada com a opção "rw", a usuária "maria" não consegue fazer alterações, pois ainda está restrita às permissões do sistema.

Existem duas soluções neste caso. A primeira seria criar uma conta "morimoto" (com qualquer senha), também no host 192.168.0.4, e usá-la para alterar os arquivos. A segunda (mais insegura) seria abrir as permissões de acesso da pasta (e arquivos dentro dela), de forma que todos os usuários possam fazer alterações. Neste caso, você usaria o comando "chmod -R 777 /home/morimoto/arquivos".



A exceção para esta regra é o usuário root. Por padrão, o NFS não permite que o usuário root de outra máquina acesse arquivos nos compartilhamentos (root_squash). Assim como a maria, o root do 192.168.0.4 acessa o compartilhamento restrito às permissões de acesso para outros usuários, já que vira um usuário inválido.

Para que o root remoto possa alterar arquivos no compartilhamento, com as mesmas permissões do root local, use a opção "no_root_squash" ao criar o compartilhamento. Note que isso bipassa apenas as permissões de acesso do sistema, não as permissões de acesso do compartilhamento (ro ou rw). Neste caso, a linha que ativa o compartilhamento, dentro do arquivo "/etc/exports" (no servidor), ficaria:

```
/home/morimoto/arquivos 192.168.0.*(rw,async,no_root_squash)
```

Ao adicionar novos compartilhamentos no arquivo "/etc/exports", você pode ativá-los usando o comando "**exportfs -a -v**". Isso ativa os novos compartilhamentos sem precisar reiniciar o servidor NFS e sem causar interrupções nos acessos dos clientes.

» Próximo: [Compartilhando impressoras com clientes Linux e Windows](#)

O **Cups**, o servidor de impressão padrão no Linux, possui um recurso nativo de compartilhamento de impressoras. Ele permite não apenas compartilhar impressoras com outras máquinas Linux, mas também com máquinas Windows da rede, através de um servidor unificado. Para habilitar o compartilhamento, edite o arquivo "/etc/cups/cupsd.conf", deixando-o com o seguinte conteúdo:

```
AccessLog /var/log/cups/access_log
ErrorLog /var/log/cups/error_log
LogLevel info
PageLog /var/log/cups/page_log
Printcap /etc/printcap.cups
User lp
Group sys
Port 631
Browsing On
BrowseAllow All
BrowseInterval 30

<Location />
Order Deny,Allow
Deny None
Allow All
</Location>

<Location /admin>
Order Deny,Allow
Deny All
Allow 127.0.0.1
</Location>
```

<pre><Location Order Deny Allow </Location></pre>	From From	/printers> Deny,Allow None All
--	----------------------------	--

Veja que a seção "/printers", que contém as impressoras, fica com permissão de acesso para todo mundo, enquanto o utilitário de administração do Cups (seção /admin) continua acessível apenas localmente, através do endereço **http://127.0.0.1:631**.

Aqui não estamos impondo nenhum tipo de restrição, por isso contamos com o firewall para bloquear qualquer tentativa de impressão proveniente de micros da Internet. Você pode também fazer o compartilhamento de uma forma mais segura, especificando manualmente a faixa de endereços da rede local, ou mesmo especificando individualmente os endereços IP que poderão imprimir. Neste caso, as seções <Location /> (onde vai a configuração que permite aos clientes verem as impressoras disponíveis) e <Location /printers> ficaria:

<pre><Location Order Deny Allow Allow </Location></pre>	From From From	/> Deny,Allow All 127.0.0.1 192.168.0.*
 <pre><Location Order Deny Allow Allow </Location></pre>		/printers> Deny,Allow All 127.0.0.1 192.168.0.*

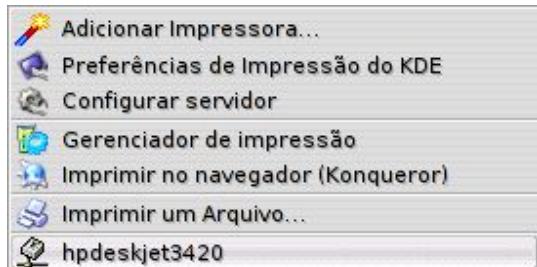
Não se esqueça de incluir o endereço "127.0.0.1" na lista. Caso contrário, todo mundo vai imprimir na impressora, menos você mesmo :).

Compartilhar impressoras através do Cups é mais simples do que fazê-lo através do Samba e oferece a vantagem adicional de permitir o uso do recurso de autodiscover do Cups nos clientes Linux. O autodiscover permite que os clientes Linux da rede reconheçam automaticamente a impressora compartilhada e a configurem automaticamente durante o boot, sem necessidade de nenhuma intervenção manual. É um recurso bastante interessante: você dá boot com o CD do Kurumin, por exemplo, manda imprimir qualquer coisa e o trabalho é direcionado de forma automática para a impressora compartilhada no servidor.

Funciona mais ou menos assim: durante o boot, o cliente manda um broadcast para a rede, perguntando se alguém está compartilhando impressoras. O servidor responde que está compartilhando a "hp" e aproveita para transmitir detalhes, como o modelo e driver usado pela impressora, configuração de impressão, etc. Como ambos estão rodando o Cups, significa que o cliente usa o mesmo conjunto de drivers de impressão do servidor; isso permite que ele simplesmente configure a impressora usando as informações recebidas, sem

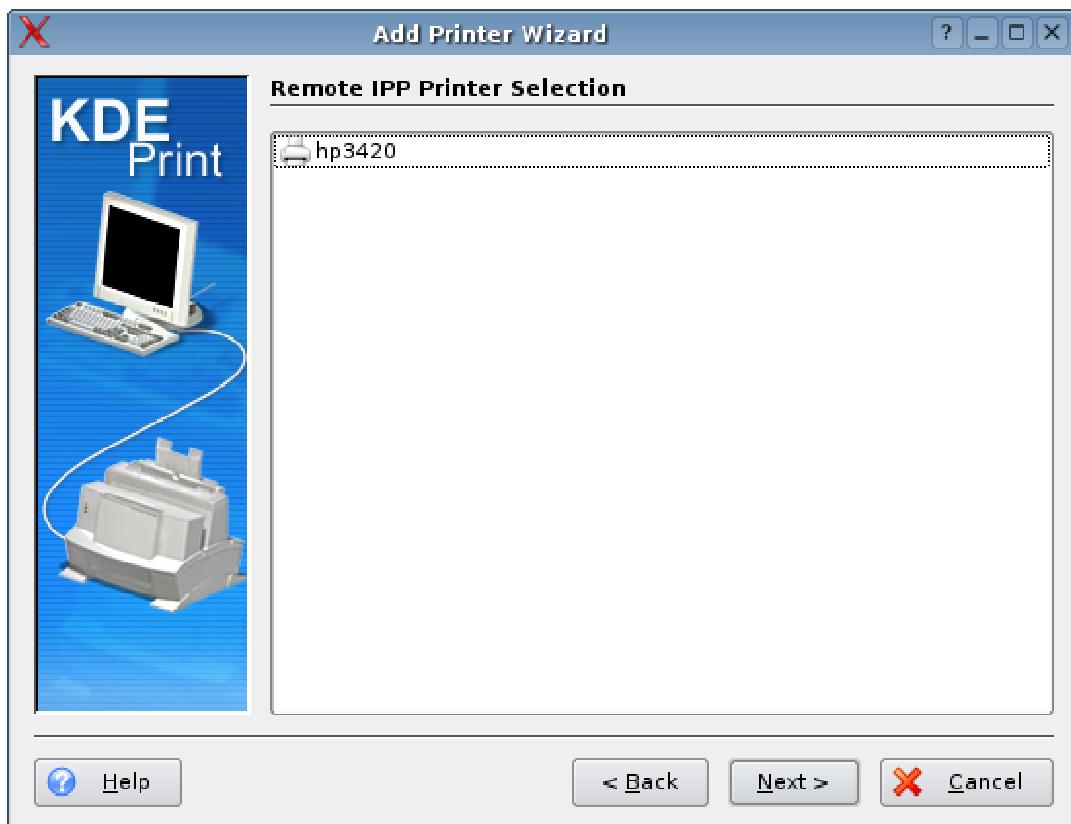
precisar perguntar nada ao usuário. O pacote de broadcast é reenviado periodicamente pelo cliente, permitindo que impressoras recentemente compartilhadas sejam descobertas.

Caso existam mais impressoras na rede, você pode escolher qual usar nas preferências de impressão. É um recurso que funciona surpreendentemente bem.



Caso você precise adicionar a impressora manualmente, abra o **kaddprintewizard** e selecione a opção Remote CUPS Server. Forneça o endereço IP do servidor na rede local (ex: 192.168.0.10) e a porta onde o Cups está escutando, que por padrão é a **631**.

Isso mostrará uma lista das impressoras disponíveis no servidor. Basta escolher a que será usada, apontar o driver que será usado e configurar as opções da impressora (papel, qualidade de impressão, etc.).



Nos clientes **Windows**, a configuração é semelhante. Eles não suportam o autodiscover, por isso é preciso adicionar a impressora manualmente pelo Painel de Controle > Impressoras e fornecer o CD com os drivers.

Vamos por passos. Comece abrindo o navegador e tentando acessar a página de administração do Cups no servidor. Acesse o <http://192.168.0.10:631> substituindo o "192.168.0.10" pelo endereço IP correto do servidor. Acesse a opção "**Manage Printers**" e clique no link da impressora que será usada. Você verá um endereço, como "<http://192.168.0.10:631/printers/hp>", na barra do navegador. Este é o endereço "completo" da sua impressora, que vamos usar na instalação.

De volta ao "Painel de Controle > Impressora", clique no "**Adicionar Impressora**" e marque a opção "**Impressora de rede**". Selecione a opção "Conectar-se a uma impressora na internet ou na intranet" e preencha o campo "URL" com o endereço completo da impressora (o "<http://192.168.0.10:631/printers/hp>" que anotamos no passo acima).

Se você estiver usando o Windows 2000 sem o Service Pack 2 ou o XP sem atualizações, ele vai dar um erro estúpido, dizendo que não é possível se conectar à impressora, mas isso é esperado. Dê ok e volte à tela inicial. Marque agora a opção "**Impressora local**" e deixe marcado o "Detectar e instalar automaticamente impressora Plug and Play". Ele dará outro erro, simplesmente confirme e diga que quer indicar a impressora manualmente. Você verá que, apesar dos erros, a impressora aparecerá disponível no final da lista. Basta selecioná-la e continuar com o processo normal de instalação da impressora, fornecendo o CD de drivers, etc.

Se você tem um servidor de impressão problemático na sua rede, que precisa ser reiniciado várias vezes ao dia, etc., recomendo que experimente substituí-lo por um servidor de impressão Linux. O Cups é um servidor de impressão muito sólido, ele raramente dá problemas. Uso na minha rede interna e até hoje não precisei reiniciar os micros por problemas na impressão uma única vez.

Se você estiver rodando o Windows em uma janela do VMware, o procedimento de instalação da impressora é o mesmo. Basta compartilhar a impressora no Linux e instalá-la no Windows do VMware seguindo os passos que mostrei acima, como se fosse uma impressora de rede.

Lembre-se de que qualquer tipo de compartilhamento de rede é sempre um risco potencial de segurança. Se você for ativá-lo em um micro simultaneamente conectado à internet e à rede local, não se esqueça de habilitar o firewall, abrindo apenas para os endereços da rede local.

O suporte a impressoras de rede compartilhadas no Cups foi incluído apenas a partir do Windows 2000. Para usar este recurso no Windows 95, 98 ou ME, você deve instalar o "Internet Printer Services", uma atualização disponibilizada pela Microsoft, que você pode baixar em:

<http://www.microsoft.com/windows98/downloads/contents/WUPreviews/IPP/Default.asp>

Depois de reiniciar, acesse o Painel de Controle > Impressora, clique no "Adicionar Impressora" e marque a opção "Impressora de rede". Coloque o endereço da impressora (<http://192.168.0.10:631/printers/hp>, por exemplo) no lugar do caminho para a impressora e forneça o driver.

» Próximo: [Mantendo o horário sincronizado](#)

Ao compartilhar arquivos na rede, manter os relógios das máquinas sincronizados passa a ser uma necessidade, afinal, se cada máquina está usando um horário diferente, fica impossível acompanhar as datas de modificações dos arquivos. Achar a versão mais recente de um determinado arquivo torna-se uma tarefa impossível e o trabalho de ferramentas diversas de backup fica prejudicado, sem falar nos logs do sistema e outros recursos que dependem do horário.

Felizmente, é muito simples manter os horários das máquinas sincronizados, graças a vários servidores NTP públicos, disponíveis pelo mundo. Os servidores principais, chamados de stratum 1, sincronizam seus relógios a partir de relógios atômicos ou um sistema de GPS e, por isso, são extremamente precisos. A seguir, temos os servidores stratum 2, servidores menores sincronizados a partir dos primeiros.

Você pode sincronizar o relógio da sua máquina rapidamente usando o comando "**ntpdate -u**", seguido pelo servidor desejado. O comando faz parte do pacote "ntp" ou "ntpd", instalado por padrão na maioria das distribuições. A opção "-u" faz com que seja usada uma porta alta, necessário se você acessa usando uma conexão compartilhada ou tem um firewall ativo.

Para facilitar as coisas, existe o servidor "pool.ntp.org", que serve como um load balancer, encaminhando as requisições para um servidor geograficamente próximo de você. Ao invés de ficar caçando servidores públicos no Google, você pode sincronizar diretamente a partir dele:

```
# ntpdate -u pool.ntp.org
```

```
8 Sep 14:12:29 ntpdate[20592]: step time server 128.208.109.7 offset -9.091791 sec
```

O Linux utiliza um sistema relativamente complexo para manter o horário do sistema. Ao invés de simplesmente confiar no horário informado pelo relógio da placa mãe, ele utiliza um sistema mais complexo, baseado no clock da placa mãe para calcular a passagem do tempo. Sempre que o sistema é desligado corretamente, diferenças no horário do sistema e no horário informado pelo relógio da placa mãe são salvas em um arquivo e recuperadas na hora do boot.

Em geral, este sistema é bem mais preciso e permite que o horário mantenha-se correto (desde que o micro não seja desligado) mesmo nos casos em que a bateria do setup está fraca e o relógio da placa mãe está atrasando.

No entanto, existem casos onde o sistema calcula o clock de forma incorreta, fazendo com que o relógio comece a adiantar ou atrasar, mesmo que o relógio da placa mãe esteja indicando o horário corretamente.

A solução, nestes casos, é rodar o comando ntpdate periodicamente, de forma que o horário seja sempre corrigido antes que as diferenças se acumulem. Neste caso, a melhor solução é fazer com que o cron execute o comando de hora em hora.

O jeito mais simples de fazer isso é criar um pequeno script dentro da pasta "/etc/cron.hourly/", cujo conteúdo é executado de hora em hora pelo cron. Crie o arquivo "/etc/cron.hourly/ntpdate", contendo as duas linhas a seguir:

```
#!/bin/sh  
ntpdate -u pool.ntp.org
```

Transforme-o em executável:

```
# chmod +x /etc/cron.hourly/ntpdate
```

O cron detecta mudanças nos arquivos automaticamente. Mas, se preferir, você pode forçar a atualização usando o comando:

```
# /etc/init.d/cron restart
```

Os servidores NTP atendem clientes de todo o mundo, independentemente do fuso horário, pois são configurados para utilizar um horário comum o UTC (Universal Time Zone). Os clientes ajustam o horário de acordo com o fuso horário local.

Naturalmente, para que isso funcione, é necessário que o fuso horário esteja configurado corretamente. A maioria das distribuições ajusta isso logo durante a instalação, mas você pode configurar o fuso horário do sistema através de vários utilitários, como o "tzconfig" ou o configurador do KDE (kcmshell clock), que aparece ao clicar com o botão direito sobre o relógio e acessar a opção "Mudar data e hora". Existem também vários clientes Windows que utilizam o protocolo NTP. Você pode baixar as versões oficiais no: <http://ntp.isc.org/bin/view/Main/ExternalTimeRelatedLinks>

O protocolo NTP leva em conta o ping entre as máquinas e outros fatores para fazer as atualizações de forma extremamente precisa. Diferenças de sincronismo entre os servidores são sempre da ordem de poucos milésimos de segundo.

» Próximo: [Capítulo 7: Web, FTP e Quota](#)

Nos primórdios da internet, usávamos páginas html estáticas e scripts CGI. O Apache em si continua oferecendo suporte apenas a esses recursos básicos, mas ele pode ser expandido através de módulos, passando a suportar scripts em PHP e acessar bancos de dados MySQL, entre inúmeros outros recursos.

Sempre que é solicitada uma página em PHP ou outra linguagem, entra em ação o módulo apropriado, que faz o processamento necessário e devolve ao Apache a página html que será exibida. Entram em ação, então, os gestores de conteúdo e fóruns, que combinam os recursos do PHP com um banco de dados como o MySQL, acessado através dele. A combinação de tudo isso forma a solução que é popularmente chamada de "**LAMP**" (Linux + Apache + MySQL + PHP).

» Próximo: [**Instalando um servidor LAMP**](#)

Atualmente quase 70% dos servidores web do mundo rodam o Apache, a maior parte deles sobre o Linux. O Apache é um dos servidores web mais antigos, seguro e com inúmeros módulos, que adicionam suporte aos mais exóticos recursos.

A maioria das páginas atuais utiliza uma estrutura em **PHP**, freqüentemente com um banco de dados **MySQL** ou PostgreSQL. Existem, inclusive, muitos sistemas prontos, como o **phpBB** (fórum) e o **PHP Nuke** (e derivados) para gerenciamento de conteúdo, que podem ser instalados sem muita dificuldade depois que o servidor web já estiver rodando. Outro recurso muito usado é a encriptação de páginas em **SSL**, necessário para a criação de páginas seguras (usadas em lojas virtuais, por exemplo) e um sistema de estatísticas de acesso como o **Webalizer**.

Além do servidor web em si, você quase sempre vai precisar configurar também um servidor **DNS**, que responde pelo domínio do seu site ou empresa. Aprender a configurar o DNS corretamente é importante, caso contrário você pode ter problemas ao enviar e-mails (pela falta do DNS reverso), ou mesmo ter problemas mais graves com o registro do domínio.

A Apache permite hospedar vários sites no mesmo servidor, recurso que chamamos de **virtual hosts**. Apenas os sites mais acessados são capazes de saturar os recursos de um servidor dedicado de configuração razoável. Por isso, hospedar vários sites no mesmo servidor é uma forma de economizar recursos e trabalho.

Ao hospedar vários sites, passamos a ter dois novos problemas: precisamos oferecer alguma forma segura de acesso aos arquivos, para que os responsáveis possam atualizar suas páginas sem alterar arquivos dos vizinhos, e precisamos de um sistema de quotas, para que cada um tenha sua fatia justa de espaço em disco.

Criamos, então, vários logins de acesso e configuramos um servidor **FTP**, para que cada um tenha acesso a seus próprios arquivos, mas sem ter como alterar os demais. Apesar de muito usado, o FTP é inseguro. No capítulo 8 veremos como usar o **SFTP** e o **SHFS**, que permitem acesso seguro, através de um túnel encriptado.

Os três são, na verdade, servidores de arquivos de uso geral, que não estão limitados a uso em conjunto com um servidor web. É muito comum usar um servidor FTP para disponibilizar arquivos para download público, por exemplo.

Completando o time, temos o **Quota**, que permite limitar o espaço em disco usado por cada usuário, garantindo uma divisão justa dos recursos disponíveis. Ao atingir seu limite, o usuário recebe uma mensagem de "disco cheio", mesmo que ainda existam vários GB's livres no HD do servidor.

» Próximo: [Instalando o Apache](#)

A primeira escolha é entre instalar o Apache 2, ou o Apache 1.3, que ainda é usado por muita gente. O Apache 2 traz muitas vantagens, sobretudo do ponto de vista do desempenho, mas, por outro lado, ele é incompatível com os módulos compilados para o Apache 1.3 e muitas opções de configuração são diferentes.

A questão dos módulos não chega a ser um grande problema hoje em dia, pois todos os principais módulos já foram portados, mas muita gente que aprendeu a configurar o Apache 1.3 se sente mais confortável com ele e, por isso, continua usando-o até hoje, apesar das vantagens da nova versão. Muitas distribuições continuam oferecendo as duas versões, de forma a satisfazer os dois públicos. No Debian, por exemplo, o Apache 1.3 é instalado através do pacote "**apache**", enquanto o Apache 2 (a versão recomendada) é instalado através do "**apache2**".

Ao instalar o Apache 2, o suporte a SSL é instalado automaticamente junto com o pacote principal (mas ainda é preciso ativá-lo na configuração, como veremos a seguir). Instale também o pacote **apache2-utils**, que contém diversos utilitários de gerenciamento que usaremos a seguir:

```
# apt-get install apache2  
# apt-get install apache2-utils
```

Você vai precisar também do pacote "**ssl-cert**", necessário para ativar o suporte a SSL e gerar os certificados. Ele não é instalado por padrão ao fazer uma instalação enxuta do Debian ou Ubuntu:

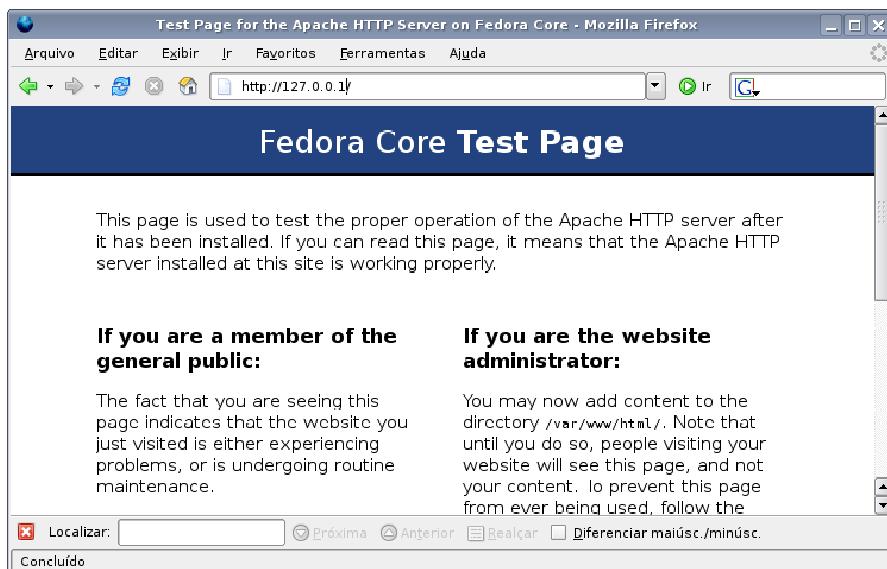
```
# apt-get install ssl-cert
```

No Fedora, instale o pacote "**httpd**", que contém o Apache 2 e utilitários:

```
# yum install httpd
```

Seguindo os nomes dos pacotes, no Debian o serviço se chama "apache2", enquanto no Fedora se chama "httpd". Para reiniciar o servidor, você usaria respectivamente "/etc/init.d/apache2 restart" e "service httpd restart".

Acessando o endereço "**http://127.0.0.1**", você verá uma página de boas-vindas, que indica que o servidor está funcionando. Se não houver nenhum firewall no caminho, ele já estará acessível a partir de outros micros da rede local ou da internet.

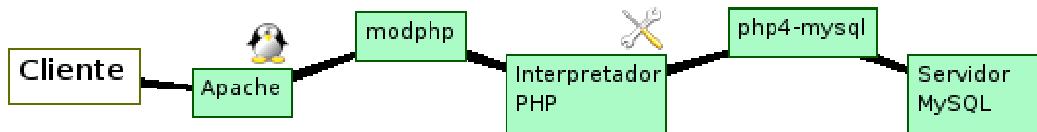


Por enquanto temos apenas uma versão básica do Apache, que simplesmente exibe arquivos html. Por padrão, o diretório raiz do servidor Web é "**/var/www**" (no Debian) ou "**/var/www/html**" (no Fedora). A página "http://seu.servidor/index.html" é, na verdade, o arquivo "**/var/www/index.html**".

» Próximo: [Entendendo a organização dos arquivos Apache 2](#)

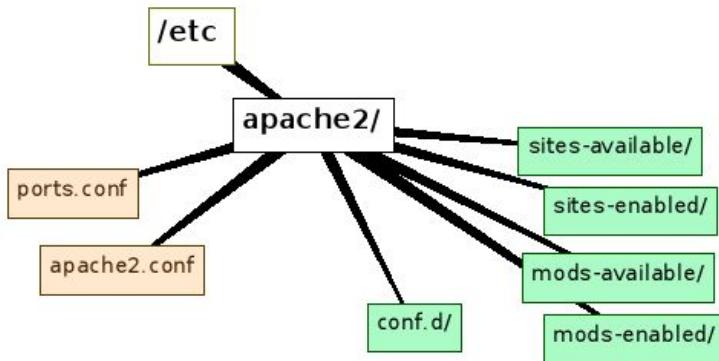
A principal característica do Apache é a modularidade. Ao invés de ser um aplicativo grande e complexo, que tenta desempenhar sozinho todas as funções, o Apache se limita a executar uma única tarefa: entregar páginas html e outros tipos de arquivos aos clientes. Qualquer outra coisa é invariavelmente feita por um módulo externo.

Por exemplo, quando você acessa uma página em PHP em um site que roda sobre um servidor Apache, ele (Apache) lê o arquivo no disco e repassa a requisição para o modphp, o módulo encarregado de processar arquivos PHP. Ele, por sua vez, aciona o interpretador PHP, que processa a página e a entrega, já processada, ao Apache, que, finalmente, a entrega ao cliente. Caso seja necessário acessar um banco de dados (como no caso de um fórum), entra em ação outro módulo, como o php4-mysql, que permite que o interpretador PHP acesse o banco de dados:



Pode parecer estranho que depois de toda essa volta, o Apache ainda consiga entregar a página processada em tempo hábil, mas é justamente essa divisão de tarefas que permite que o Apache seja tão rápido e seguro. O trabalho é dividido em várias partes e cada módulo é mantido separadamente por uma equipe que entende do assunto e zela pelo desempenho e confiabilidade do código. Graças a isso, é bastante raro que sejam descobertos problemas graves de segurança no Apache ou no interpretador PHP, por exemplo. Quase sempre, os problemas de segurança não estão no servidor Web em si, mas sim no gestor de conteúdo (phpNuke, Xoops, phpBB, etc.) usado.

No Apache 2, esta arquitetura modular é extendida também aos arquivos de configuração. No Apache 1.3, a configuração era centralizada no arquivo "/etc/apache/httpd.conf", enquanto no Apache 2 ela é dividida em vários arquivos. À primeira vista, a organização do Apache 2 parece muito mais complicada, mas depois de entender a coisa se revela muito mais simples e lógica:



Todos os arquivos de configuração estão organizados dentro do diretório "/etc/apache2". Dentro dele, temos as pastas "sites-available" e "sites-enabled", que contém a configuração dos sites hospedados; as pastas "mods-available" e "mods-enabled", que armazenam a configuração dos módulos; o arquivo "ports.conf", onde vai a configuração das portas TCP que o servidor vai escutar; o arquivo "apache2.conf", que armazena configurações diversas relacionadas ao funcionamento do servidor e a pasta "conf.d", que armazena arquivos com configurações adicionais.

O Apache é capaz de hospedar simultaneamente vários sites, cada um representado por um arquivo de configuração diferente. Imagine o caso de uma empresa de hosting que mantém um servidor com 2.000 pequenos sites. Quando cada cliente registra seu site e assina o plano de hospedagem, você cria um novo arquivo dentro da pasta "sites-available" com as configurações necessárias e um link para ele na pasta "sites-enabled".

Como os nomes sugerem, a primeira pasta armazena a configuração de todos os sites hospedados no servidor, mas apenas os sites que estiverem presentes na pasta "sites-enabled" ficam disponíveis. Quando é necessário suspender temporariamente um site por falta de pagamento, você simplesmente remove o link na pasta "sites-enabled", sem precisar mexer na configuração.

Ao invés de criar e remover os links manualmente, você pode usar os comandos "**a2ensite**" e "**a2dissite**", que fazem isso para você. Para ativar e desativar um site configurado no arquivo "/etc/apache2/sites-available/kurumin", por exemplo, os comandos seriam:

# (ativa)	a2ensite	kurumin
# (desativa)	a2dissite	kurumin

Quando o Apache é instalado, é criado por padrão o arquivo "/etc/apache2/sites-available/default", que contém a configuração de um site "raiz", que usa como diretório de páginas a pasta "/var/www". Se o seu servidor web vai hospedar um único site, então essa configuração é suficiente. Mas, caso você queira hospedar vários sites no mesmo servidor, é necessário criar uma pasta e um arquivo de configuração para cada site adicional.

Seu servidor pode, por exemplo, hospedar o "joao.com.br" e o "maria.com.br". Um servidor DNS, mantido por você, é configurado para responder pelos dois domínios, em ambos os casos dando o IP do seu servidor web. Na configuração do apache, criamos os arquivos "/etc/apache2/sites-available/joao" e "/etc/apache2/sites-available/maria", um utilizando a pasta "/var/www/joao" e "/var/www/maria".

Quando um visitante digita "http://joao.com.br", o servidor da Fapesp (que responde pelos domínios .br) vai passar a requisição para seu servidor DNS, que responde com o endereço do seu servidor web. Ao acessar o servidor, o navegador solicita o site "joao.com.br" e o servidor responde enviando o arquivo "/var/www/joao/index.html" ou "index.php" ao cliente.

Esta configuração parece bem complicada à primeira vista, mas na prática é relativamente simples. Veremos mais detalhes sobre a configuração de servidores Apache com vários domínios mais adiante.

Continuando, a mesma idéia das duas pastas separadas se aplica aos módulos. A pasta "mods-available" contém a configuração e scripts de inicialização para todos os módulos disponíveis, mas apenas os módulos referenciados (através de um link) na pasta "mods-enabled" são realmente carregados.

Muita gente simplesmente cria e deleta os links manualmente, mas isso pode ser feito mais rapidamente usando os comandos "a2enmod" e "a2dismod", que ativam e desativam módulos específicos. Para desativar o suporte a PHP, por exemplo, você usaria o comando:

```
# a2dismod php4
```

Para ativá-lo novamente, usaria:

```
# a2enmod php4
```

Uma vez que um determinado módulo é ativado, ele fica automaticamente disponível para todos os sites hospedados no servidor. Lembre-se de que, ao mexer na configuração dos módulos ou sites, é sempre necessário recarregar a configuração, para que a alteração entre em vigor:

```
# /etc/init.d/apache2 force-reload
```

Você tem o mesmo efeito se simplesmente reiniciar o Apache, mas isso não é aconselhável em um sistema de produção, pois vai derrubar temporariamente todos os sites hospedados no servidor ;). Note que, no Apache 1.3, toda a configuração de módulos e sites ia diretamente no arquivo httpd.conf, que o tornava muito mais complexo.

Outra configuração que foi desmembrada é a configuração de portas, que foi para o arquivo "**ports.conf**". Originalmente o arquivo vem com uma única linha:

```
Listen 80
```

É aqui que você altera a porta padrão do seu servidor ao disponibilizá-lo em uma conexão via ADSL (onde a operadora bloqueia a porta 80) ou adicionar novas portas, como faremos mais adiante ao ativar o SSL, por exemplo.

Alguns serviços de banda larga, como, por exemplo, o Speedy da Telefonica, bloqueiam a porta 80, obrigando os usuários a manter seus servidores em portas alternativas. Você pode também alterar a porta para manter o seu servidor um pouco mais secreto, principalmente se for utilizada uma porta acima de 1024, já que, além do endereço IP ou domínio, os visitantes precisariam saber também a porta do servidor. Ao usar uma porta diferente da padrão, é preciso incluir a porta usada na URL, como em: <http://seu-site.com.br:8080>.

Para fazer com que seu servidor escute também a porta 8080, você adicionaria uma nova linha, como em:

```
Listen 80  
Listen 8080
```

No caso do Apache 1.3, a configuração da porta vai dentro do arquivo "/etc/apache/httpd.conf", na linha "Port 80". Basta alterar o 80 pela porta desejada, salvar o arquivo e reiniciar o Apache para que a alteração entre em vigor. Se você quiser que o servidor escute em várias portas simultaneamente, use a diretiva "Listen" para adicionar as portas desejadas, como em:

```
Port 80  
Listen 1080
```

Finalmente, chegamos ao arquivo "**apache2.conf**", que agrupa o "resto" das configurações. É ele que você vai alterar quando, por exemplo, precisar ajustar o número de processos

usados pelo Apache ou aumentar o número de conexões simultâneas permitidas pelo servidor.

» Próximo: [Ativando o SSL](#)

A configuração do SSL no **Apache 2** é um pouco complicada, pois envolve a modificação de vários arquivos. Vou fazer um apanhado geral, sem explicar muito sobre a configuração de cada arquivo, já que eles são explicados individualmente mais adiante. Sugiro que leia este tópico novamente depois de terminar de ler todo o capítulo.

O primeiro passo é obter um certificado SSL. Você pode gerar seu próprio certificado, o que é rápido, grátis e indolor, ou adquirir um certificado reconhecido na Verisign ou outra entidade certificadora. O problema de usar um certificado caseiro é que os clientes receberão um aviso de "certificado não reconhecido" ao acessarem a página, emitido pelo próprio navegador. Um certificado reconhecido é caro, mas muitos provedores permitem que você utilize um certificado compartilhado pagando uma taxa anual.

Você pode obter também um certificado gratuito no: <http://www.cacert.org/>. Ele é reconhecido pela CAcert, mas o certificado raiz deles não vem pré-instalado na maioria dos navegadores, o que faz com que os clientes continuem recebendo a mensagem de certificado não válido ao acessar o servidor.

Para gerar um certificado caseiro, use o comando:

apache2-ssl-certificate

No Debian Etch, o script apache2-ssl-certificate não está mais disponível. Nele, você usaria o comando "make-ssl-cert", especificando o arquivo com o template (/usr/share/ssl-cert/ssleay.cnf) e o arquivo onde o certificado será salvo (/etc/apache2/ssl/apache.pem, para gerar um certificado padrão para o servidor), como em:

```
#                         mkdir          /etc/apache2/ssl/
# make-ssl-cert /usr/share/ssl-cert/ssleay.cnf /etc/apache2/ssl/apache.pem
```

Ele pergunta várias informações sobre a empresa e sua localização, que os clientes podem verificar quando acessam o site. Se, por acaso, o comando não estiver disponível, verifique a instalação do pacote "ssl-cert".

O próximo passo é ativar o módulo "ssl" dentro do Apache 2, o que pode ser feito rapidamente usando o comando "a2enmod". Atualize a configuração do servidor para que a alteração entre em vigor:

```
#                                     a2enmod          ssl
# /etc/init.d/apache2 force-reload
```

Abra agora o arquivo "**/etc/apache2/ports.conf**" e adicione a linha "Listen 443" (a porta usada pelo https), como em:

Port Listen 443	80
---------------------------	----

Com isso, o Apache 2 já está configurado. Falta apenas ativar o uso do SSL dentro da configuração de cada host virtual, ou seja, cada página hospedada no servidor Apache 2. Para testar, vamos ativá-lo na página padrão que usamos para testar o servidor.

Abra o arquivo "**/etc/apache2/sites-available/default**". No início do arquivo, substitua a linha "NameVirtualHost *:", por:

NameVirtualHost	*:443
NameVirtualHost *:80	

Isso explica que o Apache deve escutar tanto a porta 80 padrão, quanto a porta 443, usada pelo SSL. Logo em seguida, substitua a linha "<VirtualHost *>", por:

<VirtualHost *:80>	
--------------------	--

Até aqui, dividimos a configuração em duas seções, uma para a porta 80, outra para a porta 443, usada pelo SSL. Falta agora adicionar a seção referente à configuração do SSL no final do arquivo:

<VirtualHost	*:443>
DocumentRoot	/var/www/
ErrorLog	/var/log/apache2/error.log
CustomLog	combined
SSLEngine	on
SSLCertificateFile	/etc/apache2/ssl/apache.pem
</VirtualHost>	

Reinic peace o servidor (/etc/init.d/apache2 restart) e acesse o endereço "<https://127.0.0.1>" para testar a configuração. Ao conectar, o navegador exibe um aviso "O certificado do servidor falhou no teste de autenticidade" ou similar, o que é normal ao usar um certificado caseiro.

» Próximo: [Instalando o suporte a PHP](#)

No início, existiam apenas páginas html estáticas, com links atualizados manualmente. Depois surgiram os scripts CGI, geralmente escritos em Perl, que permitiram criar vários tipos de formulários e automatizar funções. Finalmente, surgiu o PHP, adotado rapidamente como a linguagem padrão para criação de todo tipo de página dinâmica, fórum ou gerenciador de conteúdo.

Além da linguagem ser bastante flexível, um script em PHP chega a ser mais de 100 vezes mais rápido que um script CGI equivalente, além de mais seguro. Em resumo, um script CGI é um executável, que precisa ser carregado na memória, executado e descarregado cada vez que é feita uma requisição. No caso do PHP, o interpretador fica carregado continuamente e simplesmente vai executando de forma contínua os comandos recebidos dos scripts incluídos nas páginas.

Para quem programa em Perl, existe a possibilidade de utilizar o mod-perl, instalável através do pacote "apache-mod-perl" ou "libapache2-mod-perl2". Assim como o PHP, o mod-perl é um módulo do Apache que fica continuamente carregado na memória, executando os scripts Perl de uma forma bem mais rápida e segura que os scripts CGI.

Mas, voltando ao assunto principal, no **Debian** o suporte a PHP é instalado através do pacote "**php5**" (ou "php4", de acordo com a versão escolhida). Para instalá-lo, basta usar o gerenciador de pacotes da distribuição em uso, como em:

```
# apt-get install php5
```

No caso do **Fedora**, é usado um pacote unificado, o "php", que inclui a versão mais recente do interpretador, eliminando a confusão:

```
# yum install php
```

Com o interpretador PHP instalado, falta instalar o módulo do Apache 2, que no **Debian** está disponível através do pacote "libapache2-mod-php4" ou "libapache2-mod-php5", como em:

```
# apt-get install libapache2-mod-php5
```

O módulo "libapache2-mod-php5" é instalado dentro da pasta "**/usr/lib/apache2/modules/**" e é ativado de uma forma diferente que no Apache 1.3. Ao invés de adicionar as linhas que ativam o módulo e criam as associações de arquivos no final do arquivo httpd.conf, são criados dois arquivos dentro da pasta "**/etc/apache2/mods-available/**", com, respectivamente, a ativação do módulo e as associações de arquivos. Para ativar o suporte a PHP, é preciso copiar ambos para a pasta "**/etc/apache2/mods-enabled/**":

```
# cd /etc/apache2/mods-available/  
# cp -a php4.conf php4.load ..mods-enabled/
```

Como vimos, você pode automatizar esta etapa ativando o módulo através do a2enmod, que cria os links automaticamente:

```
# a2enmod php4  
ou  
# a2enmod php5
```

Não se esqueça de atualizar a configuração do Apache:

```
# /etc/init.d/apache2 force-reload
```

O **Fedora** não utiliza as pastas "modules-available" e "modules-enabled" como no Debian. O módulo do Apache é instalado junto com o pacote "php", mas para ativá-lo é necessário adicionar a linha abaixo no final do arquivo "/etc/httpd/conf/httpd.conf":

```
AddType application/x-httdp-php .php .phps .php3 .phtml .html .htm .shtml .fds
```

O serviço "httpd" do Fedora também não suporta o parâmetro "force-reload". Ao invés disso, use simplesmente "service httpd reload".

A partir daí, o Apache continua exibindo diretamente páginas com extensão .htm ou .html, mas passa a entregar as páginas .php ou .phps ao interpretador php, que faz o processamento necessário e devolve uma página html simples ao Apache, que se encarrega de enviá-la ao cliente.

Estas páginas processadas são "descartáveis": cada vez que um cliente acessa a página, ela é processada novamente, mesmo que as informações não tenham sido alteradas. Dependendo do número de funções usadas e da complexidade do código, as páginas em PHP podem ser bastante pesadas. Não é incomum que um site com 300.000 pageviews diários (o que significa umas 20 a 30 requisições por segundo nos horários de pico) precise de um servidor dedicado, de configuração razoável.

Quase sempre, os sistemas desenvolvidos em PHP utilizam também um banco de dados MySQL ou Postgre SQL. Para utilizá-los, você precisa ter instalados (além do MySQL ou Postgre propriamente ditos) os módulos "php4-mysql" e "php4-pgsql" (ou respectivamente "php5-mysql" e "php5-pgsql" ao usar o PHP 5), que permitem aos scripts em PHP acessarem o banco de dados:

```
# apt-get install php5-mysql  
ou  
# apt-get install php5-pgsql
```

Não se esqueça de reiniciar o Apache, para que as alterações entrem em vigor:

```
# /etc/init.d/apache force-reload
```

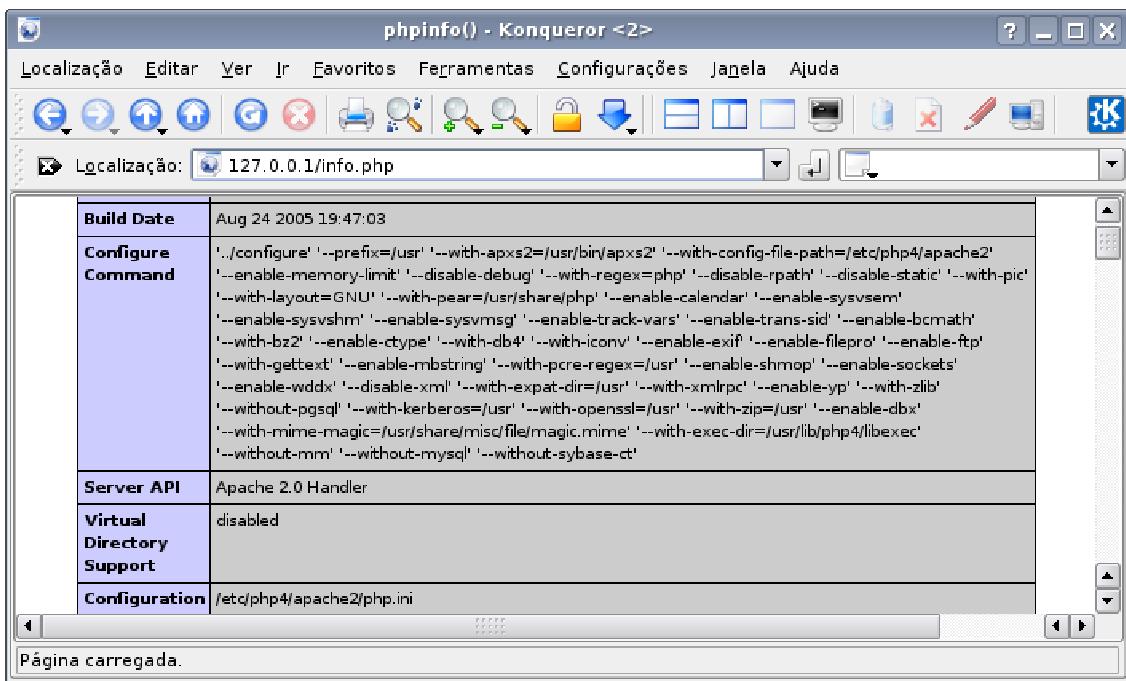
No caso do **Fedora**, instale o pacote "php-mysql":

```
# yum install php-mysql
```

Para verificar se o PHP está realmente ativo (em qualquer versão do Apache), crie um arquivo de texto chamado "info.php" (ou outro nome qualquer, seguido da extensão .php) dentro da pasta do servidor web, contendo apenas a linha abaixo:

```
<?php phpinfo(); ?>
```

Salve o arquivo e abra a página através do navegador. A função "phpinfo()", que usamos no arquivo, faz com que o servidor exiba uma página com detalhes do configuração do PHP e módulos ativos:



Depois de verificar, remova o arquivo, pois não é interessante que essas informações fiquem disponíveis ao público.

» Próximo: [Instalando o MySQL](#)

O MySQL é um banco de dados extremamente versátil, usado para os mais diversos fins. Você pode acessar o banco de dados a partir de um script em PHP, através de um aplicativo desenvolvido em C ou C++, ou praticamente qualquer outra linguagem (até mesmo através de um Shell Script! :).

Existem vários livros publicados sobre ele, por isso vou me limitar a falar sobre a instalação e a configuração necessária para utilizá-lo em um servidor LAMP, em conjunto com o Apache e o PHP.

O primeiro passo é instalar o servidor MySQL propriamente dito:

```
# apt-get install mysql-server
ou
# yum install mysql-server
```

Você pode instalar também os pacotes "mysql-client" (o cliente que permite acessar os dados e fazer modificações no banco de dados) e o "mysql-navigator" (uma interface gráfica para ele).

Antes de iniciar o serviço, rode o comando "mysql_install_db", que cria a base de dados "mysql", usada para armazenar informações sobre todas as outras criadas posteriormente, e uma base chamada "test", que pode ser usada para testar o servidor:

```
# mysql_install_db
```

O passo seguinte é ativar o servidor:

```
# /etc/init.d/mysql start
```

No caso do **Fedora**, o serviço se chama "mysqld", ao invés de simplesmente "mysql", como no Debian:

```
# service mysqld start
```

O MySQL possui um usuário padrão chamado "root", que, assim como o root do sistema, tem acesso completo a todas as bases de dados e é usado para fazer a configuração inicial do sistema, assim como tarefas de manutenção. Esta conta inicialmente não tem senha, por isso você deve definir uma logo depois de iniciar o serviço, usando o comando "mysqladmin -u root password senha", incluindo a senha desejada diretamente no comando, como em:

```
# mysqladmin -u root password 123456
```

Se você precisar trocar a senha posteriormente, é necessário acrescentar o parâmetro "-p" antes do "password" e especificar a nova senha, como em:

```
# mysqladmin -u root -p password asdfg
```

Enter password:

Note que nesse caso o comando solicita a senha antiga antes de continuar, já que do contrário teríamos uma brecha óbvia de segurança.

Continuando, depois de definir a senha, o próximo passo é criar uma base de dados. Você pode instalar vários scripts diferentes (um fórum, um chat e um gestor de conteúdo, por exemplo) no mesmo servidor e, inclusive, várias cópias de cada um. Isso é cada vez mais utilizado, tanto dentro de sites que oferecem diversos serviços, quanto em servidores compartilhados, onde os responsáveis por cada site têm a liberdade de instalar os sistemas de sua preferência.

» Próximo: [Administração básica do banco de dados](#)

Existem muitas interfaces de administração para o MySQL, mas a forma mais elementar é usar o prompt de comando. Para acessar o prompt do MySQL, use o comando :

```
# mysql -u root -p <enter>
```

```
Enter password: <senha>
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 43 to server version: 4.0.15-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql>
```

Veja que o cabeçalho normal do bash foi substituído por um "mysql>", que lembra onde você está ;). Para sair, pressione "Ctrl+C" ou execute o comando "Bye".

Dentro do prompt, use o comando "create database" (criar base de dados), seguido pelo nome desejado. Neste exemplo, estou criando uma base de dados para usar na instalação do phpBB, que veremos a seguir. Um detalhe importante é que todos os comandos dados dentro do prompt do MySQL devem terminar com ponto-e-vírgula:

```
mysql> CREATE DATABASE phpbb;
Query OK, 1 row affected (0.04 sec)
```

Para confirmar, use o comando "SHOW DATABASES;", que lista as bases de dados criadas no servidor, como em:

```
mysql> SHOW DATABASES;
```

Database
information_schema
mysql
phpbb
test

Note que além da base "phpbb" que criamos, existem mais três bases de dados, criadas durante a instalação. As bases "mysql" e "information_schema" são para uso interno do MySQL, incluindo o armazenamento das configurações (sendo um banco de dados, o MySQL usa a si mesmo para armazenar suas configurações :), enquanto a base "test" é uma DB vazia, que pode ser usada para fins de teste.

Temos em seguida a questão das permissões de acesso. Nada impede que você sempre utilize a conta "root" do MySQL e inclusive configure os scripts instalados para o utilizarem. Entretanto, isso é extremamente inseguro, principalmente se você pretende instalar vários scripts e aplicativos no mesmo servidor, ou se as bases de dados serão acessadas por vários usuários.

O ideal é que cada base de dados tenha um usuário próprio e seja acessível apenas por ele. Se você vai instalar o phpBB (fórum) e o Xoops (gerenciador de conteúdo), por exemplo,

crie duas bases de dados ("phpbb" e "xoops", por exemplo) e dois usuários separados, cada um com permissão para acessar uma das duas bases.

Na configuração de cada um, informe a base de dados a ser usada e o usuário e senha correspondente. Isso evita que eventuais problemas de segurança em um coloquem em risco também os dados referentes ao outro.

Outra situação comum é ao configurar um servidor com vários virtual hosts. Nesse caso, o webmaster de cada site vai precisar de uma ou mais bases de dados e, naturalmente, cada um vai precisar de um login próprio, com acesso apenas às suas próprias bases de dados.

Para criar um usuário "phpbb", com senha "12345" e dar a ele acesso à base de dados "phpbb" que criamos, use (dentro do prompt do MySQL) o comando:

```
mysql> GRANT ALL ON phpbb.* TO phpbb IDENTIFIED BY '12345';  
(permite tudo na base phpbb para o usuário phpbb, identificado pela senha 12345)
```

Para trocar a senha posteriormente, use o comando:

```
mysql> SET PASSWORD FOR phpbb = PASSWORD('123456');  
(defina senha para o usuário phpbb, onde a senha é 123456)
```

Este mesmo comando pode ser usado para trocar a senha do root, como em:

```
mysql> SET PASSWORD FOR root = PASSWORD('asdfgh');
```

Se mais tarde você precisar remover as permissões de acesso de um usuário anteriormente criado (em um site com vários webmasters, onde um se desligou da equipe, por exemplo) use o comando:

```
mysql> REVOKE ALL ON phpbb.* FROM phpbb;  
(remova todos os direitos para a base phpbb, para o usuário phpbb)
```

Para remover uma base de dados, use o comando "DROP DATABASE", como em:

```
mysql> DROP DATABASE phpbb;
```

Veja que os comandos usados dentro do prompt do MySQL seguem uma linguagem literal, usando palavras do inglês. Quem tem uma boa familiaridade com a língua tem bem mais facilidade em dominar os comandos.

Outra observação é que os comandos não são case sensitive. Tanto faz escrever "CREATE DATABASE phpbb;" ou "create database phpbb;". Escrever os comandos em maiúsculas é apenas uma forma de dar mais destaque a eles.

Depois dessa configuração inicial, você pode experimentar instalar um gerenciador gráfico para facilitar a manutenção do seu servidor MySQL. Uma boa opção neste caso é o **phpMyAdmin**.

Para instalá-lo, basta instalar o pacote "phpmyadmin", como em:

```
# apt-get install phpmyadmin
```

O pacote para instalação em outras distribuições, que não incluem o pacote por padrão, pode ser encontrado no: <http://www.phpmyadmin.net/>.

O phpMyAdmin é um script em PHP, que trabalha em conjunto com o Apache. O script de pós-instalação incluído no pacote do Debian faz a configuração inicial para você, perguntando se ele deve ser configurado para trabalhar em conjunto com o Apache (1.3), Apache-ssl (ainda na versão 1.3) ou Apache 2 (a opção correta em nosso caso). O SSL permite que a interface de administração seja acessada via https, onde os dados são transmitidos de forma encriptada, melhorando a segurança. Ao usar o Apache 2, o acesso via SSL fica automaticamente habilitado, desde que você tenha habilitado o SSL no servidor, como vimos há pouco.



Depois de instalado, acesse o endereço "<http://127.0.0.1/phpmyadmin/>" ou "<https://127.0.0.1/phpmyadmin/>" e você cairá na tela de administração do phpMyAdmin, onde você pode logar-se usando qualquer uma das contas registradas no MySQL. Use o root para tarefas administrativas, quando for necessário ter acesso a todas as bases ou fazer backup de tudo, e uma das contas restritas para acessar uma base específica. Por questões de segurança, a configuração padrão permite que ele seja acessado apenas localmente.

The screenshot shows the phpMyAdmin interface running in Mozilla Firefox. The title bar reads "127.0.0.1 >> localhost >> test | phpMyAdmin 2.6.2 - Mozilla Firefox". The menu bar includes Arquivo, Editar, Exibir, Ir, Favoritos, Ferramentas, and Ajuda. The top right has a search icon. The main area shows the "Banco de Dados: test" selected. Below it is a table titled "Tabela" with columns "Ações" and "Re". The table lists several tables from the "test" database, each with a checkbox and a set of icons for management. The tables listed are: phpbb_auth_access, phpbb_banlist, phpbb_categories, phpbb_config, phpbb_confirm, phpbb_disallow, phpbb_forum_prune, and phpbb_forums.

Uma observação importante é que ao ser usado em conjunto com o Apache, instalado no mesmo servidor que ele, o MySQL é acessado apenas localmente, através da interface de loopback. O Apache envia a requisição ao módulo PHP que faz o acesso ao banco de dados, tudo localmente. Nessa configuração o servidor MySQL **não** deve ficar disponível para a Internet. Configure o firewall para bloquear a porta 3306 usada pelo servidor MySQL, além de todas as outras portas que não forem explicitamente necessárias.

Caso o servidor MySQL vá ser utilizado por outros servidores (você pode configurar o phpBB e outros scripts para utilizarem um servidor MySQL externo), deixe a porta aberta apenas para os endereços IP dos servidores que forem ter acesso. Como os servidores dedicados sempre utilizam endereços fixos (ao contrário dos servidores domésticos), esta configuração fica mais simples. Para administrar seu servidor MySQL remotamente, o ideal é que se conecte ao servidor via SSH e faça todo o trabalho através dele. Se precisar acessar diretamente alguma ferramenta de configuração, como o Webmin ou o PhPMyAdmin, você pode criar um túnel (novamente usando o SSH) ligando a porta correspondente do servidor a uma porta da sua máquina e fazer o acesso através dela. Veremos em detalhes como usar o SSH e criar túneis encriptados mais adiante.

» Próximo: [Usando o phpBB](#)

Com o php4, php4-mysql e o mysql-server instalados você tem pronta a estrutura necessária para instalar os diversos scripts de fórum, chat, gestores de conteúdo e outros.

A maioria destes scripts é simples de instalar. Você precisa apenas criar uma base de dados no MySQL ou Postgre, copiar os arquivos para uma pasta dentro do servidor web e editar um arquivo (ou acessar uma página de configuração através do navegador) para incluir as informações sobre o servidor (base de dados a ser usada, login e senha, etc.) e concluir a configuração.

Note que embora o Apache e o MySQL sejam bastante seguros, nada garante que os scripts desenvolvidos por terceiros também serão. De nada adianta ter um servidor web extremamente seguro, se o script de gerenciamento de conteúdo que você instalou tem um buffer overflow no campo de login, que permite executar comandos arbitrários, o que pode ser usado para obter a senha do servidor MySQL (que o script usa para fazer seu trabalho) e, de posse dela, fazer alterações no conteúdo do site.

O ponto fraco na segurança de qualquer site ou fórum é quase sempre a segurança do script usado. Não escolha qual usar pensando apenas na facilidade de uso. Investigue o histórico de segurança e, uma vez escolhido qual usar, fique de olho nas atualizações de segurança.

Vou usar como exemplo a instalação do phpBB, um script de fórum bastante usado e com um bom histórico de segurança e desempenho. Ele é o script usado no fórum do Guia do Hardware (<http://www.guiadohardware.net/comunidade/>), que tem quase dois milhões de mensagens postadas, 60 mil usuários registrados e chega a ter mais de 1.000 usuários simultâneos.

O phpBB tem código aberto e é gratuito, você pode baixá-lo no <http://www.phpbb.com/downloads.php>.

Comece baixando o pacote principal. Enquanto escrevo, ele está na versão 2.0.15 e o arquivo é o "**phpBB-2.0.15.tar.gz**".

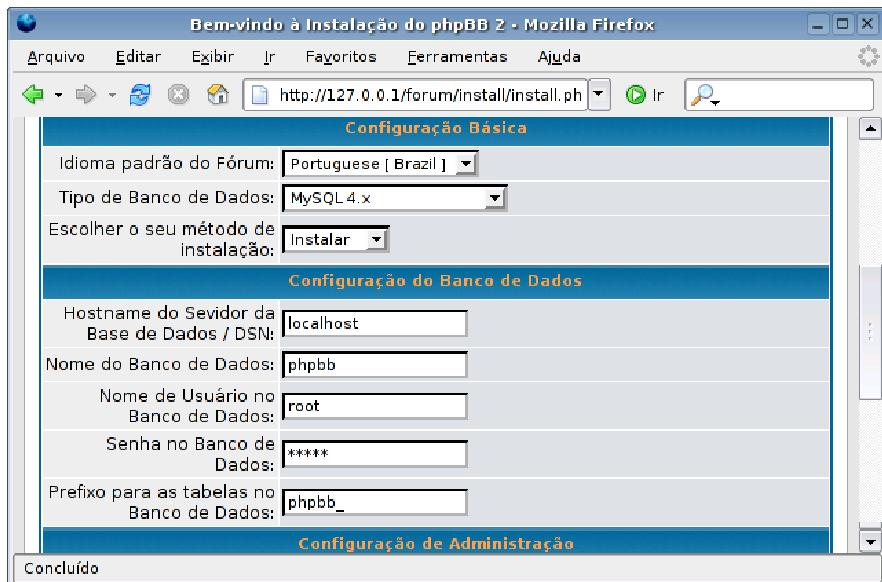
Para instalar, salve-o dentro da pasta "**/var/www**" (ou a pasta de dados do seu servidor Apache, caso esteja usando outro diretório) e renomeie a pasta criada para o diretório onde o fórum deve ficar acessível. No meu caso, estou instalando-o na pasta "**forum/**". Delete o arquivo original, pois não vamos mais precisar dele:

```
#                               cd          /var/www
#           tar      -zxvf   phpBB-2.0.15.tar.gz
#           rm      -f        phpBB-2.0.15.tar.gz
# mv phpBB2/ forum
```

Aproveite para instalar também o suporte à internacionalização. O phpBB já foi traduzido para vários idiomas, incluindo português do Brasil. Comece baixando o arquivo "**lang_portuguese.tar.gz**" (que contém a tradução propriamente dita) e descompacte-o dentro da pasta "**/var/www/forum/language**". Baixe, em seguida, o arquivo "**subSilver_portuguese_brazil.tar.gz**" (que contém botões e ícones com o texto de legenda traduzido) e descompacte-o na pasta "**/var/www/forum/templates**".

Veja que tudo isso pode ser feito via ftp ou sftp, mesmo que você não tenha acesso via shell no servidor. Tudo o que é preciso fazer é copiar os arquivos para as pastas apropriadas. Esse sistema de instalação foi desenvolvido pensando em quem utiliza planos de hospedagem em servidores compartilhados.

Depois de copiar os arquivos, acesse a página "**/forum/install/install.php**" dentro da árvore do seu site. O acesso pode ser feito tanto localmente (<http://127.0.0.1/forum/install/install.php>) quanto via internet. Esta é a página usada para concluir a instalação. É importante que você acesse a página assim que os arquivos forem copiados, pois ela fica acessível para qualquer um.



Preencha os campos com as informações do seu servidor:

- Database Type: Escolha o MySQL 4.x ou MySQL 3.x, de acordo com a versão instalada. Note que o phpBB também oferece suporte ao PostgreSQL e até mesmo ao MS SQL Server, caso o Apache esteja rodando sobre o Windows.

- Database Server Hostname / DSN: O phpBB pode acessar um servidor MySQL instalado em outra máquina da rede, não é necessário que o Apache e o MySQL estejam instalados na mesma máquina. Separar os dois é interessante do ponto de vista da performance e também da segurança, mas, por outro lado, é mais caro e trabalhoso. Caso o MySQL esteja instalado na mesma máquina, mantenha o "**localhost**", do contrário, informe o endereço IP ou domínio do servidor a ser utilizado.

- Your Database Name: Aqui você indica a base de dados que será usada pelo fórum. No tópico anterior, criamos a base de dados "**phpbb**". Caso você esteja instalando só para testar, pode usar também a base de dados "test", criada por padrão.

- Database Username: Ao criar a base "phpbb" criamos também o usuário "**phpbb**", com acesso a ela. Ao instalar para teste, você pode usar a conta "root", que tem acesso a tudo,

mas isso não é recomendável do ponto de vista da segurança. Nunca use o root em uma instalação que vá ficar disponível na internet.

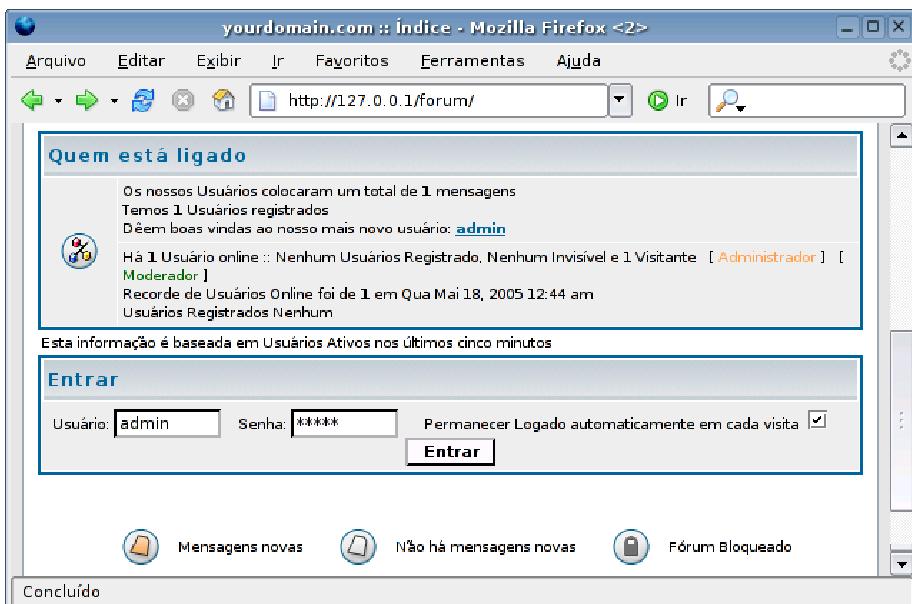
- **Database Password:** A senha do usuário indicado na opção acima.
- **Prefix for tables in database:** phpbb_ (mantenha o default).
- **Admin Email Address:** seu@email.com (um e-mail válido para o envio de mensagens de erro e alertas).
- **Domain Name:** Aqui vai o domínio do seu site, como "meunome.com.br". Se você está fazendo uma instalação de teste, fora do servidor real, deixe o valor padrão.
- **Server Port:** 80 (porta onde o servidor Apache está disponível. Informe a porta correta caso tenha alterado a configuração do Apache)
- **Script path:** /forum/ (pasta no servidor onde está instalado o fórum).
- **Administrator Username:** admin (o login que você usará para administrar o fórum).

Ao terminar, clique no "**Start Install**" e feche a janela do navegador. Caso apareçam mensagens de erro, significa que o suporte a PHP, o banco de dados MySQL ou o módulo phpmysql não estão instalados corretamente. Verifique se todos os pacotes foram instalados sem erros, se o servidor MySQL está ativo e se você não se esqueceu de reiniciar o Apache depois de ter instalado o suporte a PHP.

É importante deletar as pastas "install" e "contrib" dentro da pasta do fórum, que contém arquivos necessários apenas durante a instalação:

```
#                                     cd                               /var/www/forum/  
# rm -rf install contrib
```

Terminados esses passos, seu fórum já estará funcionando. Assim como qualquer gerenciador que se preze, o phpBB oferece um painel de administração, que fica disponível ao se logar usando a conta administrativa criada durante a instalação. Através do painel, você pode criar novas salas, alterar as configurações do fórum, moderar e assim por diante.



» Próximo: [Virtual Hosts](#)

Outro recurso suportado pelo Apache, e muito usado, é a possibilidade de hospedar vários sites no mesmo servidor (shared hosting). Mais de 80% dos sites da internet são hospedados dessa forma econômica.

Nesse caso, os arquivos de cada site ficam guardados em uma pasta diferente e o servidor se encarrega de direcionar cada visitante ao site correto. Servidores como os dos serviços de hospedagem gratuita exploram esse recurso ao extremo, com um número assustador de sites hospedados em cada servidor.

Em um host compartilhado, os recursos do servidor (HD, memória, processamento e link) são divididos entre os sites hospedados, assim como vários programas abertos simultaneamente disputam os recursos da máquina. Isso faz muito sentido no caso de sites pequenos ou médios, que não possuem um número suficiente de visitas para saturarem, sozinhos, o servidor.

Ao hospedar seu site em um host compartilhado, você tem direito a uma determinada quota de espaço em disco e pode acessar os arquivos do site via FTP ou SFTP. Porém, você não pode instalar novos pacotes, nem alterar a configuração do servidor: fica limitado aos recursos disponíveis.

A segunda opção é alugar um host dedicado, onde você tem um servidor completo à sua disposição, inclusive com a possibilidade de criar virtualhosts e hospedar outros sites.

Invariavelmente, ao hospedar vários domínios, você precisa configurar um servidor DNS (o mais usado é o Bind) para responder a todos os domínios hospedados no servidor, entregando o endereço IP do seu servidor Apache. Não é necessário que o DNS esteja instalado no mesmo servidor que o Apache, a função dele será unicamente responder às requisições dos clientes, fornecendo o IP correto.

Em muitos casos, a própria empresa que lhe aluga o servidor oferece esse serviço, mas vamos aprender como fazer isso manualmente logo a seguir.

Configurar um servidor DNS pode ser um pouco indigesto, mas a configuração do Apache é bastante simples. Em primeiro lugar, você deve criar uma pasta separada para cada site que será hospedado. Você pode usar a própria pasta "/var/www", como em:

```
#                               mkdir          /var/www/joao
# mkdir /var/www/maria
```

Além de configurar o servidor Web, precisaremos fazer com que os responsáveis possam acessar os arquivos nas suas respectivas pastas, a fim de atualizar seus sites. A forma mais simples de fazer isso é criar um usuário para cada um e dar acesso a eles via FTP. Mais adiante veremos como configurar o servidor FTP com o Proftpd. Veja que as três coisas acabam se integrando: o Bind resolve os nomes de domínio, o Apache fornece as páginas e o Proftpd permite que os webmasters atualizem os sites.

Acesse agora a pasta "**/etc/apache2/sites-available**", onde vai a configuração dos sites disponíveis. Comece editando o arquivo "**/etc/apache2/sites-available/default**", substituindo as linhas:

```
NameVirtualHost *
<VirtualHost *>
```

Por:

```
NameVirtualHost *:80
<VirtualHost *:80>
```

Essa configuração é necessária para ativar o suporte a SSL para os virtual hosts. Sem ela, além do SSL não funcionar para os virtual hosts (funcionaria apenas para o site default), você precisaria modificar a configuração de cada um, usando sempre "<VirtualHost *>" ao invés de "<VirtualHost *:80>".

Imagine que vamos hospedar os sites "www.joao.com.br" e "www.marria.com.br", usando as duas pastas criadas anteriormente. Criaríamos, então, um arquivo para cada site, contendo o seguinte:

- **/etc/apache2/sites-available/joao:**

```
<VirtualHost *:80>
ServerAdmin joao@joao.com.br
ServerName www.joao.com.br
```

```

ServerAlias          joao.com.br
DocumentRoot        /var/www/joao
</VirtualHost>

```

- **/etc/apache2/sites-available/maria:**

```

<VirtualHost           *:80>
  ServerAdmin          maria@gmail.com
  ServerName            www.marria.com.br
  ServerAlias           marria.com.br
  DocumentRoot          /var/www/maria
  </VirtualHost>

```

Note que adicionei uma nova diretiva, a "ServerAlias", que permite que o site seja acessado tanto com, quanto sem o "www". A linha "ServerAdmin" é, na verdade, opcional, contém apenas o e-mail de contato do administrador do site.

Fazendo dessa forma, os logs de acessos serão misturados no log principal do Apache, o "/var/log/apache2/access.log". Para que cada site tenha seus logs separados, você deve adicionar duas linhas adicionais, especificando a localização. Não é recomendável deixar os logs disponíveis ao público, por isso seria interessante usar diretórios diferentes, como em:

```

<VirtualHost           *:80>
  ServerAdmin          joao@joao.com.br
  ServerName            www.joao.com.br
  ServerAlias           joao.com.br
  DocumentRoot          /var/www/joao/html
  ErrorLog              /var/www/joao/logs/error.log
  CustomLog             /var/www/joao/logs/access.log
  </VirtualHost>

```

Caso queira ativar o suporte a SSL para os virtual hosts, adicione (depois de fazer a configuração que vimos no tópico sobre virtual hosts) a sessão referente ao SSL dentro da configuração de cada site, indicando corretamente a pasta do site e os arquivos de log. O SSL pode ser tanto ativado para o raiz do site, permitindo que os visitantes visualizem qualquer parte do site usando o "https://", ou utilizar uma pasta separada, onde está a parte de comércio eletrônico do site, por exemplo, como em:

```

<VirtualHost           *:443>
  DocumentRoot          /var/www/joao/ssl
  ErrorLog              /var/www/joao/logs/error.log
  CustomLog              combined
  SSLEngine               on
  SSLCertificateFile    /etc/apache2/ssl/apache.pem
  </VirtualHost>

```

Neste caso ao acessar o "<http://joao.com.br>" o visitante visualizará o conteúdo da pasta "/var/www/joao/html", enquanto ao acessar o "<https://joao.com.br>", visualizará a "/var/www/joao/ssl".

Depois de feita a configuração, ative ambos os sites usando o comando a2ensite, o que criará links para eles na pasta "/etc/apache2/sites-enabled":

```
#                                         a2ensite          joao
# a2ensite maria
```

Você pode adicionar quantos sites quiser usando esses mesmos passos. Sempre que alterar a configuração, é necessário dar um force-reload, para que as alterações entrem em vigor:

```
# /etc/init.d/apache2 force-reload
```

Note que, como todos os sites ficam hospedados no mesmo servidor, a única forma de chegar ao site desejado é fazendo o acesso através do domínio. Se você tentar acessar diretamente o IP do servidor, vai cair no site padrão (configurado através do arquivo "/etc/apache2/sites-available/default"), que, por padrão, usa o raiz da pasta "/var/www".

Esta página default pode ser usada para mostrar alguma publicidade da empresa responsável pelo servidor, ou uma lista dos sites hospedados.

» Próximo: [Configurando no Fedora](#)

O **Fedorá** não usa uma estrutura modular para a configuração dos virtual hosts como no Debian. Ao invés disso, a configuração do Apache 2 vai toda no arquivo "/etc/httpd/conf/httpd.conf".

Procure pela seção "Virtual Hosts", perto do final do arquivo, e descomente a linha:

```
NameVirtualHost *:80
```

A partir daí, você pode adicionar cada um dos sites hospedados, como em:

```
<VirtualHost *:80>
  ServerName www.joao.com.br
  ServerAlias joao.com.br
  DocumentRoot /var/www/joao

<VirtualHost *:80>
  ServerName www.maría.com.br
  ServerAlias maría.com.br
  DocumentRoot /var/www/maría
```

Essa configuração manual funciona para pequenos servidores, que hospedam algumas dezenas ou centenas de páginas. Grandes serviços de hospedagem geralmente acabam desenvolvendo algum tipo de sistema para automatizar a tarefa. Nos serviços de hospedagem gratuita, por exemplo, onde o número de clientes é assustadoramente grande,

as alterações são feitas de forma automática quando o visitante faz seu cadastro, geralmente através de um sistema escrito em PHP ou Java.

Conforme o número de usuários cresce e o espaço em disco no servidor começa a ficar escasso, você começará a sentir falta de um sistema de quotas que limite o espaço que cada usuário pode usar. Para isso, consulte o tópico sobre quotas de disco, mais adiante.

» Próximo: [Gerando estatísticas](#)

O Webalizer é um gerador de estatísticas de acesso para o servidor web. O Apache, por si só, loga todos os acessos feitos ao servidor, incluindo as páginas acessadas, o tráfego gerado, os navegadores e os sistemas operacionais usados pelos clientes, entre outras informações úteis para entender os hábitos e interesses de seus visitantes.

Com o Apache funcionando, é simples instalar o Webalizer: procure pelo pacote "webalizer" dentro do gerenciador de pacotes. Ele é incluído em todas as principais distribuições. Nas derivadas do Debian, você pode instalá-lo via apt-get:

```
# apt-get install webalizer
```

Ao contrário do Apache, o Webalizer não é um serviço que fica residente, mas sim um executável que precisa ser chamado cada vez que quiser ver a página de estatísticas atualizada (assim como o Sarg). Basta chamá-lo como root:

```
# webalizer
```

Por padrão, a página de estatísticas é armazenada na pasta "webalizer/", dentro do seu servidor web. Se o Apache estiver configurado para armazenar as páginas dentro do diretório "/var/www", então as estatísticas vão para a pasta local "/var/www/webalizer".

O arquivo de configuração do Webalizer é o "**/etc/webalizer.conf**". É importante que você revise o arquivo de configuração, indicando pelo menos a localização correta do arquivo de log do Apache e altere a pasta onde as estatísticas ficarão armazenadas, caso não queira que elas fiquem disponíveis ao público. Você pode armazená-las em uma pasta isolada no servidor web, como, por exemplo, "/var/webalizer", de forma que elas fiquem disponíveis apenas localmente ou através de um script. As duas opções "essenciais" dentro do arquivo são:

LogFile	/var/log/apache/access.log
OutputDir	/var/www/webalizer

Para não precisar executar o comando "webalizer" manualmente sempre que precisar atualizar as estatísticas, você pode configurar o cron para executá-lo automaticamente uma vez por dia ou uma vez por hora. Para isso, basta criar um script dentro da pasta "**/etc/cron.daily/**" ou "**/etc/cron.hourly/**", contendo o comando "**webalizer**".

Todos os scripts colocados dentro dessas pastas são respectivamente executados todos os dias de manhã, ou uma vez por hora. Para que funcione, é importante verificar se o serviço "**cron**" ou "**crond**" está ativo. No caso do Debian, o script do Cron é criado automaticamente e configurado para ser executado diariamente.

Em um servidor Apache com vários virtual hosts, é possível fazer com que o Webalizer gere estatísticas separadas para cada um, com uma configuração um pouco mais cuidadosa. Em primeiro lugar, você deve configurar o Apache para gerar arquivos de log separados para cada site hospedado, como no exemplo que vimos há pouco:

```
<VirtualHost *:80>
  ServerAdmin joao@joao.com.br
  ServerName www.joao.com.br
  ServerAlias joao.com.br
  DocumentRoot /var/www/joao/html
  ErrorLog      /var/www/joao/logs/error.log
  CustomLog     /var/www/joao/logs/access.log  combined
</VirtualHost>
```

Configurando dessa forma, os logs do site "joao.com.br" ficarão armazenados no arquivo "/var/www/joao/logs/access.log", os do "maria.com.br" no "/var/www/maria/logs/access.log" e assim por diante, sempre separados dos arquivos disponíveis ao público, que vão na pasta "html".

O próximo passo é criar um arquivo de configuração do Webalizer separado para cada um. Para manter as coisas organizadas, crie o diretório "**/etc/webalizer**" e crie uma cópia do arquivo webalizer.conf original para cada site, dentro da pasta, como em:

```
#           mkdir          /etc/webalizer
#           cp            /etc/webalizer.conf        /etc/webalizer/joao.conf
# cp /etc/webalizer.conf /etc/webalizer/maria.conf
```

Precisamos agora editar cada um dos arquivos, informando o arquivo de log, domínio e diretório onde ficarão armazenados os relatórios de cada site. No arquivo "/etc/webalizer/joao.conf", por exemplo, você teria (além das outras linhas do arquivo padrão) as linhas:

```
LogFile          /var/www/joao/logs/access.log
OutputDir        /var/www/joao/html/webalizer
HostName joao.com.br
```

Depois de gerar os arquivos de configuração para todos os sites, falta fazer com que o Webalizer processe todos automaticamente. Uma forma rápida de fazer isso (dica do próprio FAQ do webalizer) é usar o comando:

```
# for i in /etc/webalizer/*.conf; do webalizer -c $i -q; done
```

Esse é, na verdade, um mini-script, que vai executar o Webalizer uma vez para cada arquivo ".conf" encontrado no diretório "/etc/webalizer", gerando todas as estatísticas de uma tacada só. Para que o comando seja executado todos os dias automaticamente, coloque-o dentro de um script no diretório **"/etc/cron.daily"**.

No **Debian**, é criado o script **"/etc/cron.daily/webalizer"**, que se encarrega de gerar as estatísticas diariamente, lendo o arquivo **"/etc/webalizer.conf"**. Podemos modificá-lo para que leia também os arquivos dentro do diretório **"/etc/webalizer"**. Para isso, edite o arquivo e substitua as linhas:

```
# Run webalizer quietly
${WEBALIZER_BIN} -c ${WEBALIZER_CONF} -q ${nonrotatedlog}
```

Por:

```
# Run webalizer quietly
${WEBALIZER_BIN} -c ${WEBALIZER_CONF} -q ${nonrotatedlog}

for i in /etc/webalizer/*.conf; do ${WEBALIZER_BIN} -c $i -q; done
for i in /etc/webalizer/*.conf; do ${WEBALIZER_BIN} -c $i -q ${nonrotatedlog}; done
```

Um problema comum é como restringir o acesso às estatísticas, afinal, em muitos casos, elas não devem ficar disponíveis ao público. A solução mais simples nesse caso é usar um arquivo **.htaccess**, que permite restringir o acesso ao diretório, exigindo login e senha.

O primeiro passo é criar um arquivo de senhas, usando o comando **"htpasswd"**, que faz parte do pacote "apache-utils" (o mesmo que utilizamos para gerar as senhas do Squid). O arquivo de senhas deve ser armazenado num diretório fora do diretório com os arquivos do site. De preferência, use um arquivo separado para cada site hospedado.

Vou usar como exemplo a pasta **"/etc/apache2/auth"**:

```
# mkdir /etc/apache2/auth
# cd /etc/apache2/auth
# touch joao.auth
# htpasswd joao.auth joao
```

```
New password: password:
Re-type new password:
```

Aqui criamos o arquivo **"/etc/apache2/auth/joao.auth"**, contendo o usuário "joao" e a senha digitada, armazenada num formato encriptado. Você pode armazenar vários logins no mesmo arquivo, executando o comando uma vez para cada usuário.

Com o arquivo de senhas criado, crie um arquivo de texto chamando **".htaccess"** no raiz do diretório das estatísticas, contendo o seguinte:

AuthName	"Acesso	Restrito"
AuthType		Basic

```
AuthUserFile /etc/apache2/auth/joao.auth
require valid-user
```

A linha "AuthName" contém o texto que será mostrado na tela de login exibida para o cliente, enquanto o "AuthUserFile" contém o arquivo de senhas gerado.

» Próximo: [Performance do Apache](#)

O número de instâncias do servidor é uma das configurações mais diretamente relacionadas à performance do servidor e ao consumo de memória. O Apache é capaz de responder a um número indefinido de acessos simultâneos, de acordo com o link e recursos da máquina. Para cada requisição simultânea, é necessário que exista uma instância do Apache carregada na memória.

Quando o cliente acessa uma página, ele monopoliza uma dessas instâncias abertas até que a requisição seja concluída, ou seja, até que a página seja carregada ou o arquivo baixado. Em horários de alta demanda, são abertas mais instâncias do servidor Apache, que vão sendo fechadas (para economizar memória) conforme os acessos diminuem.

O número de instâncias abertas é determinada pelas quatro linhas abaixo, dentro do arquivo "/etc/apache2/apache2.conf":

StartServers	5
MinSpareServers	5
MaxSpareServers	10
MaxClients 20	

A opção "StartServers" determina o número padrão de servidores que ficarão carregados na memória, respondendo a requisições. Cada instância ocupa cerca de 2 MB de memória (um pouco mais de acordo com as opções de compilação usadas), de forma que 5 instâncias consomem cerca de 10 MB de memória do servidor.

Além das instâncias principais, temos instâncias "reservas", que ficam disponíveis para absorver rapidamente picos de acesso, sem que o Apache tenha que perder tempo carregando mais instâncias para só depois começar a responder às requisições. As opções "MinSpareServers" e "MaxSpareServers" determinam o número mínimo e máximo de "reservas", sendo que o número real flutua entre os dois parâmetros, de acordo com a demanda.

A opção "MaxClients" é a parede de concreto, o número máximo de conexões simultâneas que o Apache aceita manter abertas, independentemente da demanda. Quando esse número é atingido, o acesso ao site fica cada vez mais lento, pois cada novo visitante "entra na fila" e precisa esperar que uma das instâncias do Apache fique livre, antes de conseguir carregar cada página.

Essa configuração default do Apache é adequada a um site de baixa demanda. Para um servidor dedicado, que hospede um site com muitas visitas, é necessário ajustar estes valores de acordo com a demanda. Uma forma fácil de verificar o status do servidor é ativar a diretiva "server-status" dentro do arquivo "/etc/apache2/apache2.conf". Adicione (ou descomente) as linhas abaixo:

```
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 200.234.23.233
  # (onde o 200.234.23.233 é o IP do micro do onde o relatório será acessado)
</Location>
```

Ative a configuração usando o comando "/etc/init.d/apache2 force-reload". A partir daí, você pode ver um instantâneo do status do servidor acessando a pasta "server-status", como em "<http://kurumin.com.br/server-status>", a partir do navegador.

A oitava linha indica o número de instâncias abertas, como em:

```
8 requests currently being processed, 5 idle workers
```

Nesse exemplo temos 8 conexões abertas e 5 instâncias reservas do Apache abertas, prontas para receber novas conexões. A velocidade do acesso ao site está normal.

Mas, o que acontece no caso de um pico de acesso, com 50 acessos simultâneos? Na configuração padrão você teria:

```
20 requests currently being processed, 0 idle workers
```

Ou seja, o Apache responde às primeiras 20 conexões e coloca as demais na fila, fazendo com que os visitantes tenham que esperar vários segundos. Em casos mais extremos, o tempo de espera poderia ser tamanho que o site ficaria virtualmente fora do ar. É o que muitas vezes acontece com links publicados em sites muito acessados, como o slashdot.org.

Isso pode ser minimizado configurando o Apache corretamente. Se você tem um servidor dedicado, com 256 MB de RAM, por exemplo, onde cada instância do Apache consome 2 MB, você poderia deixar a opção "MaxClients" com o valor "80" ou "100", de forma que o Apache aceite conexões até esgotar o limite da memória disponível (lembre-se de que além do Apache, temos o MySQL, PHP e outros recursos, que também consomem memória). Acima disso não adianta, pois o servidor passaria a usar memória swap, o que comprometeria o desempenho.

A partir daí, você pode ajustar as opções "StartServers" e "MinSpareServers" de acordo com o número médio de acessos simultâneos. Em um site com, em média, 10 acessos simultâneos e picos de 20 ou 30 acessos, uma boa configuração seria:

StartServers	10
MinSpareServers	10

MaxSpareServers
MaxClients 80

20

Outra opção que afeta negativamente o desempenho é a "HostNameLookups", onde o Apache verifica a origem de cada acesso, fazendo uma busca de domínio. Ativar essa opção permite criar estatísticas de acesso mais detalhadas, incluindo os países e provedores de acesso usados pelos visitantes, mas tem um impacto negativo muito grande na performance de um servidor congestionado. No Apache 2 ela já vem desativada por padrão, mas em versões antigas era necessário desativá-la manualmente:

HostNameLookups Off

Você pode simular tráfego no seu servidor, verificando como ele se comporta com níveis variados de tráfego usando o comando "ab", que faz parte do pacote "apache-utils". Chame-o indicando o número de requisições simultâneas e a página que será carregada, como em:

```
$ ab -n 1000 -c 20 http://www.meusite.com/teste.php
```

Nesse exemplo, estamos fazendo 1000 acessos à página, mantendo 20 requisições simultâneas. Se possível, lance o teste a partir de outro servidor com link rápido, ou peça para vários amigos rodarem o comando simultaneamente, cada um usando sua conexão. Isso transformará o teste em algo mais parecido com uma situação normal.

Instalando o Servidor FTP

O servidor de FTP mais usado no Linux é o Proftpd, incluído em quase todas as distribuições. O funcionamento do FTP é bem mais simples que o do Samba ou SSH, por isso ele é usado como uma forma simples de disponibilizar arquivos na internet ou mesmo dentro da rede local, sem muita segurança.

A principal limitação do protocolo FTP é que todas as informações são transmitidas pela rede de forma não encriptada, como texto puro, incluindo os logins e senhas. Ou seja, alguém capaz de sniffar a conexão, usando um programa como o Ethereal, veria tudo que está sendo transmitido.

Para aplicações onde é necessário ter segurança na transmissão dos arquivos, é recomendável usar o **SFTP**, o módulo do SSH que permite transferir arquivos de forma encriptada. Apesar disso, se você quiser apenas criar um repositório com alguns arquivos para download ou manter um servidor público como o Ibiblio.org, então o FTP é mais interessante, por ser mais simples de usar.

O servidor aceita conexões remotas usando os logins dos usuários cadastrados na máquina. Lembre-se de que para adicionar novos usuários você pode usar o comando **adduser** ou algum utilitário de administração incluído na distribuição, como o users-admin (que faz parte do pacote "gnome-system-tools", encontrado no Debian e outras distribuições), kuser, drakuser (Mandriva) ou system-config-users (Fedora).

Não é difícil instalar o Proftpd, basta procurar pelo pacote "proftpd" na distribuição usada, como em:

```
#           apt-get      install      proftpd
#           yum         install      proftpd
ou:
# urpmi proftpd
```

No Debian, durante a instalação do pacote do Proftpd, geralmente serão feitas algumas perguntas. A primeira é se você deseja deixar o servidor FTP ativo em modo **standalone** ou em modo **inetd**.

O standalone é mais seguro e mais rápido, enquanto o inetd faz com que ele fique ativo apenas quando acessado, economizando cerca de 400 KB de memória RAM (que fazem pouca diferença hoje em dia). O modo standalone é a opção recomendada.

Você terá também a opção de ativar o **acesso anônimo**, que permite acessos anônimos (somente leitura) na pasta "/home/ftp", onde você pode disponibilizar alguns arquivos para acesso público. Nesse caso, os usuários se logam no seu servidor usando a conta "**anonymous**" e um endereço de e-mail como senha. Caso prefira desativar o acesso anônimo, apenas usuários com login válido na máquina poderão acessar o FTP.

Depois de concluída a instalação, o servidor fica ativo por default, é inicializado automaticamente durante o boot e pode ser controlado manualmente através do serviço "proftpd", como em "/etc/init.d/proftpd start".

Você pode acessar outras máquinas da rede com servidores FTP ativos usando o GFTP, o Konqueror ou o próprio navegador. O FTP pode ser usado também como opção para transferência de arquivos na rede local. Uma das vantagens do FTP é que existem clientes para todas as plataformas, você pode baixar o FileZilla, um servidor e cliente de FTP for Windows, no <http://filezilla.sourceforge.net/>.

A configuração manual do servidor FTP é feita através do arquivo **"/etc/proftpd.conf"**. Um arquivo configurado no Kurumin pode ser usado no Mandriva (por exemplo), ou vice-versa; afinal, independentemente de estar usando o Debian, Fedora ou o Mandriva, o proftpd será sempre o mesmo.

Sempre que fizer alterações no arquivo, reinicie o servidor para que elas entrem em vigor. Para isso, use o comando **"/etc/init.d/proftpd restart"**.

No caso do Debian Etch e do Ubuntu 6.10 em diante, o servidor é configurado para utilizar endereços IPV6 por padrão, o que faz com que ele exiba uma mensagem de erro e aborte a inicialização caso você não tenha configurado a rede IPV6.

Para solucionar o problema, abra o arquivo **"/etc/proftpd.conf"** e substitua a linha:

UseIPv6 on

por:

UseIPv6 off

Uma das primeiras opções do arquivo é a opção **Port**, que permite alterar a porta usada pelo FTP. O padrão é usar a porta 21, mas muitos serviços de banda larga bloqueiam as portas 21 e 80 para que os usuários não rodem servidores. Nesse caso, você pode mudar para a porta 2121 por exemplo:

```
#      Port      21      is      the      standard      FTP      port.  
Port 2121
```

Ao mudar a porta padrão do servidor, os usuários precisarão indicar manualmente a porta no cliente de ftp ou navegador, como em: `ftp://200.234.213.23:2100`.

Em seguida vem a opção **MaxInstances**, que limita o número de conexões simultâneas do servidor FTP. Esta opção trabalha em conjunto com a limitação de banda (veja a seguir). Você pode limitar os downloads de cada usuário a um máximo de 10 KB/s e limitar o servidor a 3 usuários simultâneos. Assim, o FTP consumirá um máximo de 30 KB/s do link do servidor.

MaxInstances 30

Se você quiser limitar o acesso dos usuários a seus diretórios home, adicione a linha "**DefaultRoot ~**" no final do arquivo. Lembre-se de que no Linux o "~" é um curinga, que é automaticamente substituído pela pasta home do usuário que está logado:

```
DefaultRoot ~
```

Para ativar a limitação de banda, adicione a linha "TransferRate RETR 8:10", onde o "8" pode ser substituído pela taxa desejada, em KB/s, por usuário:

```
TransferRate RETR 8:10
```

A princípio, apenas os usuários que tiverem logins válidos no servidor poderão acessar o FTP. Caso você queira abrir um FTP público, adicione estas linhas no arquivo de configuração. Elas ficam comentadas no arquivo original:

```
<Anonymous  
User          ~ftp>  
Group         ftp  
UserAlias    anonymous  
DirFakeUser  on  
DirFakeGroup on  
RequireValidShell off  
MaxClients 20  
DisplayLogin welcome.msg  
DisplayFirstChdir .message  
<Directory *:>  
<Limit WRITE>  
DenyAll  
</Limit>  
</Directory>  
<Directory incoming>  
Umask        022  
<Limit READ 022  
WRITE>
```

```

DenyAll
</Limit>

<Limit
AllowAll
</Limit>
<Directory>
</Anonymous>
```

STOR>

A linha "**MaxClients**" determina o número máximo de anônimos que poderão se logar simultaneamente no servidor. Essa opção é separada da MaxClients principal, que limita o número de usuários com login válido. Você pode permitir 30 usuários válidos e mais 20 anônimos, por exemplo.

A opção "**DisplayLogin welcome.msg**" indica a mensagem de boas-vindas que é mostrada quando os usuários fazem login no FTP. Por padrão, é exibido o conteúdo do arquivo "**/home/ftp/welcome.msg**".

Os usuários anônimos têm acesso apenas aos arquivos dentro da pasta "**/home/ftp**". Esse é o diretório raiz para eles, eles não têm como ver, muito menos alterar outros arquivos do sistema.

A seção "**Directory incoming**" mais abaixo cria uma pasta de upload (por padrão a "**/home/ftp/incoming**") onde os anônimos poderão dar upload de arquivos. A idéia é que você veja periodicamente o conteúdo da pasta e mova o que for útil para a pasta "**/home/ftp**" para que o arquivo fique disponível para download.

Por padrão, os anônimos não podem ver o conteúdo da pasta incoming, podem apenas dar upload. Se necessário, crie a pasta incoming usando os comandos:

```
#                         mkdir                               /home/ftp/incoming
# chown nobody.nogroup /home/ftp/incoming
```

Para acessar o seu servidor, os clientes devem usar o login "anonymous" ou "ftp", usando um endereço de e-mail qualquer como senha.

Uma medida comum ao ativar o upload dos usuários anônimos é usar uma partição separada para o FTP, para evitar que algum engraçadinho fique dando upload durante a madrugada até lotar o HD do servidor. Nesse caso, você precisa apenas adicionar uma linha no arquivo "**/etc/fstab**" para que a partição desejada seja montada durante o boot. Esta linha de exemplo montaria a partição /dev/hda3, formatada em ReiserFS na pasta /home/ftp:

```
/dev/hda3 /home/ftp reiserfs defaults 0 2
```

» Próximo: [Criando usuários](#)

Imagine agora que você quer uma configuração um pouco mais complexa, com vários usuários, cada um tendo acesso a apenas uma pasta específica. Esta configuração pode ser usada em conjunto com os virtual hosts do Apache (permitindo que os responsáveis possam atualizar os arquivos do site), ou em situações em que seu servidor hospeda arquivos de diversos usuários ou projetos diferentes.

O responsável pelo **projeto1** pode dar upload para a pasta "**/home/ftp/projeto1**" (por exemplo), mas não deve ter acesso a outras pastas nem a outros arquivos do sistema. Os usuários anônimos terão acesso às pastas de todos os projetos, mas naturalmente apenas para leitura.

A forma mais simples de fazer isso é criar os usuários que terão acesso ao FTP, colocando a pasta a que terão acesso como seu diretório home e bloqueando o uso do shell, para que eles não possam acessar o servidor remotamente através de outros meios (via ssh, por exemplo).

Vamos começar adicionando no arquivo a opção que prende os usuários nos seus diretórios home. Abra o arquivo "**/etc/proftpd.conf**" e adicione (no final do arquivo) a linha:

DefaultRoot ~

Você vai precisar adicionar também a seção para liberar o acesso anônimo ao ftp, que vimos acima. Como queremos apenas que os mantenedores dos projetos possam dar upload de arquivos, remova a seção "**<Directory incoming>**". A seção vai ficar:

```
<Anonymous
User                                ~ftp>
Group                               ftp
UserAlias                           nogroup
DirFakeUser                         ftp
DirFakeGroup                         ftp
RequireValidShell                   off
MaxClients                         20
DisplayLogin                      welcome.msg
DisplayFirstChdir                   .messag
<Directory
<Limit
DenyAll
</Limit>
</Directory>
</Anonymous>
```

O diretório padrão do FTP, onde os visitantes terão acesso aos arquivos, é a "**/home/ftp**". Em outras distribuições pode ser a pasta "**/var/ftp**"; dê uma olhada em como o arquivo vem configurado por padrão.

Vamos começar criando subpastas para cada projeto:

#	mkdir	/home/ftp/projeto1
#	mkdir	/home/ftp/projeto2

```
#                               mkdir                         /home/ftp/projeto3  
etc...
```

O próximo passo é ir adicionando os usuários no sistema, tendo o cuidado de fazer as alterações no diretório home e no shell padrão, para que eles tenham acesso somente via FTP e apenas à pasta desejada.

Para adicionar os usuários, use o comando "**adduser**", como se estivesse criando uma conta normal:

```
# adduser projeto1
```

```
Acrescentando novo usuário projeto1 (1005) projeto1...  
Acrescentando novo usuário projeto1 com grupo projeto1...  
Acrescentando novo usuário projeto1 pessoal /home/projeto1.  
Criando diretório  
Copiando arquivos de /etc/skel  
  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully
```

Veja que por padrão ele cria a pasta "/home/projeto1", que passa a ser o diretório home do usuário criado. Mas, neste caso, queremos que o home seja a pasta "**/home/ftp/projeto1**", onde ele irá dar upload dos arquivos.

Para alterar isso, vamos no arquivo "**/etc/passwd**", onde ficam guardadas as informações dos usuários:

```
# kedit /etc/passwd
```

Na última linha do arquivo você verá:

```
projeto1:x:1005:1005:,:/home/projeto1:/bin/bash
```

Vamos alterar o "**/home/projeto1**" para "**/home/ftp/projeto1**", para trocar o home e o "**/bin/bash**" para "**/bin/false**", para travar o usuário e impedir que ele fique fazendo o que não deve no servidor. Se você preferir que, além do acesso via ftp, os usuários tenham acesso via ssh, então mantenha o "**/bin/bash**". Depois das alterações, a linha ficará:

```
projeto1:x:1005:1005:,:/home/ftp/projeto1:/bin/false
```

Você pode aproveitar para remover a pasta /home/projeto1, já que não precisaremos mais dela:

```
# rm -rf /home/projeto1/
```

Na verdade, esse processo serve para que você entenda melhor o procedimento de criação destes usuários "falsos" no Linux. Estas alterações podem ser especificadas ao criar o usuário. Não é preciso sair editando todos os arquivos manualmente.

O comando para criar o usuário "projeto1", usando a pasta "/home/ftp/projeto1" como home e o "/bin/false" como shell, seria:

```
# adduser --home /home/ftp/projeto1 --shell /bin/false --no-create-home projeto1
```

Não se esqueça de acertar as permissões da pasta /home/ftp/projeto1:

```
# chown -R projeto1.projeto1 /home/ftp/projeto1/
```

Depois de concluir a configuração, falta só reiniciar o servidor FTP para que as configurações entrem em vigor:

```
# /etc/init.d/proftpd restart
```

Em distribuições derivadas do Debian, você vai precisar adicionar a linha "/bin/false" no final do arquivo /etc/shells para que ele possa ser usado:

```
# echo "/bin/false" >> /etc/shells
```

Feito isso, você já conseguirá logar-se no servidor usando o login criado. O usuário não enxerga nada fora da pasta "/home/ftp/projeto1" e todos os arquivos que ele der upload vão para lá.

A senha de acesso ao FTP é a mesma definida na hora de criar o usuário. O proftpd simplesmente aproveita o sistema de autenticação do sistema. Se você precisar alterar a senha do usuário, use o comando "**passwd projeto1**".

Para usar o Proftpd em conjunto com os virtual hosts do Apache, a configuração é a mesma, com exceção de que você não precisaria habilitar o acesso anônimo, já que o FTP seria usado apenas pelos responsáveis pelos sites.

Nesse caso, ao criar os usuários, use a pasta com os arquivos do site como home, como em:

```
# adduser --home /var/www/joao --shell /bin/false --no-create-home joao
```

Não se esqueça de alterar as permissões da pasta, de forma que o usuário possa escrever nela:

```
# chown -R joao.joao /var/www/joao
```

» Próximo: [Quotas de disco](#)

O Quota é um recurso muito útil em servidores de terminais, servidores web e servidores de arquivos com muitos usuários. Imagine, por exemplo, um servidor LTSP com 20 terminais,

usado por 300 usuários diferentes. Como impedir que alguns poucos usuários comecem a baixar um monte de filmes e músicas em MP3, entupindo o HD do servidor?

Através do Quota é possível limitar a quantidade de espaço em disco disponível para cada usuário, reservando 50 MB para cada aluno, por exemplo. O uso mais comum do Quota é utilizar uma partição /home separada e ativar o Quota para ela. Isso faz todo o sentido, pois, por default, os usuários podem gravar arquivos apenas dentro da pasta /home. Mas, é possível usar o Quota em qualquer partição, como, por exemplo, em um servidor web compartilhado entre vários virtual hosts, onde a partição de dados está montada no diretório /var/www.

Ao ser ativado, o Quota procura por todos os arquivos de cada usuário dentro da partição, não por uma pasta específica. O sistema de Quotas funciona mesmo que os arquivos de um determinado usuário estejam espalhados por várias pastas.

Originalmente, apenas os sistemas de arquivos EXT2 e EXT3 ofereciam suporte ao Quota nativamente, mas, a partir do Kernel 2.6 (usado nas distribuições atuais), foi incluído também suporte para o ReiserFS. É possível instalar o suporte a Quota no ReiserFS em distribuições antigas, baseadas no Kernel 2.4 através de um patch para o Kernel.

Para usar o Quota, o ideal é dividir o HD em três partições: uma partição menor (de 5 ou 10 GB), formatada em ReiserFS para a instalação do sistema, a partição swap e outra partição maior (englobando o restante do espaço do HD), formatada em ReiserFS ou Ext3 para o diretório /home (ou a pasta onde o quota será ativado) onde ficarão armazenados os arquivos dos usuários.

No Quota existem dois limites que podem ser estabelecidos, o **soft limit** e o **hard limit**. O hard limit é o limite de espaço em si, digamos, 1000 MB para cada usuário. O sistema não permitirá que seja gravado nenhum byte acima do limite. O soft limit é um limite de advertência, digamos, 800 MB. Sempre que superar o soft limit, o usuário receberá uma mensagem de alerta, mas ainda poderá gravar mais dados até que atinja o hard limit. Você pode especificar também um **grace period**, que será o tempo máximo em que o usuário poderá ficar acima do soft limit (uma semana, por exemplo).

Passado o período, o usuário será obrigado a apagar alguma coisa e voltar a ocupar menos de 800 MB antes de poder gravar qualquer novo arquivo (nenhum arquivo do usuário é deletado pelo Quota). Um problema comum relacionado ao uso do Quota em servidores de terminais é que o KDE deixa de abrir quando o limite de espaço do usuário é atingido: ele precisa sempre de algum espaço disponível para criar os arquivos temporários que armazenam as informações da sessão. Nesses casos, você (administrador) vai precisar deletar manualmente alguns arquivos ou aumentar a quota do usuário para que ele possa voltar a usar a conta.

Você pode estabelecer os mesmos limites também para os grupos e inclusive combinar as duas limitações. Você pode, por exemplo, permitir que cada usuário do grupo "alunos" use 5 GB de disco, desde que o grupo todo não use mais do que 50 GB.

Para que o Quota funcione, é necessário instalar os pacotes "**quota**" e "**quotatool**", que contém um conjunto de utilitários usados para configurar e verificar as quotas de disco. No Debian, os dois podem ser instalados via apt-get:

```
# apt-get install quota  
# apt-get install quotatool
```

No Fedora, você precisa apenas instalar o pacote "quota" usando o Yum:

```
# yum install quota
```

Em seguida, é necessário carregar o módulo "**quota_v2**", que ativa o suporte necessário no Kernel:

```
# modprobe quota_v2
```

Para que ele seja carregado automaticamente durante o boot, adicione a linha "**quota_v2**" no final do arquivo "**/etc/modules**", ou adicione o próprio comando "modprobe quota_v2" no final do arquivo "**/etc/rc.d/rc.local**" ou "**/etc/init.d/bootmisc.sh**" (esse passo não é necessário no Fedora Core 5, onde o suporte a Quota vem compilado no executável principal do Kernel).

Com o módulo carregado, o primeiro passo da configuração é alterar a entrada no fstab que monta a partição, de modo que o suporte a quotas de disco seja ativado. Abra o arquivo "**/etc/fstab**", localize a linha referente à partição e adicione os parâmetros "**usrquota,grpquota**" logo após o "defaults". Se você está ativando o Quota para a partição "**/home**", a linha seria parecida com:

```
/dev/hda2 /home ext3 defaults 0 2
```

Depois da alteração, a linha ficaria:

```
/dev/hda2 /home ext3 defaults,usrquota,grpquota 0 2
```

Ao usar uma partição formatada em ReiserFs, a linha ficaria:

```
/dev/hda2 /home reiserfs defaults,usrquota,grpquota 0 2
```

Em seguida você deve criar os arquivos "**aquota.user**" e "**aquota.group**", onde ficam armazenadas as configurações no diretório em que a partição é montada. Se você está ativando o Quota para a partição montada no **/home**, então os dois arquivos serão "**/home/aquota.user**" e "**/home/aquota.group**".

Por enquanto, vamos apenas criar dois arquivos vazios, usando o comando touch. É importante que ambos fiquem com permissão de acesso "600", de modo que apenas o root possa acessá-los ou fazer modificações:

```
# touch /home/aquota.user  
# chmod 600 /home/aquota.user
```

```
# touch /home/aquota.group  
# chmod 600 /home/aquota.group
```

Depois da configuração inicial, falta apenas definir as quotas. A forma mais prática é utilizar o Webmin, que oferece módulos para configurar os mais diversos servidores. Alguns são desnecessariamente complicados, mas outros (como no caso do Quota) são simples de usar e realmente facilitam o trabalho de configuração.

Procure pelo pacote "**webmin**" no gerenciador de pacotes da distribuição usada. Em algumas, o próprio pacote webmin instala os módulos disponíveis, enquanto que em outras (como no Debian), o Webmin é desmembrado em vários pacotes diferentes, permitindo que você instale apenas os módulos que for usar. Para instalar o Webmin e o módulo de configuração do Quota pelo apt-get, os comandos seriam:

```
# apt-get install webmin  
# apt-get install webmin-quota
```

Depois de instalado, inicie o Webmin com o comando "**/etc/init.d/webmin start**".

O **Fedora** não inclui o pacote do Webmin em seus repositórios, por isso você não pode instalá-lo usando o Yum. Ao invés disso, baixe o pacote .rpm no <http://webmin.com> (procure pelo link Download: RPM) e instale-o usando o "rpm -i", como em:

```
# rpm -i webmin-1.270-1.noarch.rpm
```

Não se esqueça de iniciar o serviço, usando o comando "service webmin start".

O Webmin não é um programa gráfico no sentido tradicional, mas sim uma interface de configuração que você acessa usando o navegador. Isso permite que ele seja usado remotamente ou mesmo usado em servidores sem o ambiente gráfico instalado.

Para acessá-lo, abra o navegador e acesse o endereço "**https://127.0.0.1:10000**".

O navegador exibe um aviso sobre autenticidade do certificado. Isso é normal, pois o Webmin utiliza uma conexão criptografada (https) e o certificado de segurança é gerado durante a instalação. Este certificado "caseiro" não é reconhecido pelas entidades certificadoras (você não pagou nada por ele), por isso o aviso. De qualquer forma, a segurança é a mesma.

Na tela de login do Webmin, logue-se usando o login "root" e a senha de root da máquina. Se preferir, você pode trocar a senha de root do Webmin, de forma que as duas senhas sejam diferentes, na opção "Webmin > Webmin Users".

Por padrão, o Webmin só pode ser acessado localmente. Para acessar a partir de outras máquinas da rede, inclua os endereços autorizados dentro da opção "Webmin > Webmin Configuration > IP Access Control". Você pode também fazer com que a interface fique em português no "Webmin > Webmin Configuration > Language".

Depois de logado, acesse a seção "**System > Disk Quotas**". Comece clicando no "Enable Quotas", isso faz com que ele realize uma varredura inicial, calculando o espaço ocupado por cada usuário, o que pode demorar alguns minutos em partições com muitos arquivos. Depois de tudo ativado, chegamos finalmente à tela inicial de configuração:

The screenshot shows the 'Edit Quota' interface. At the top, there's a navigation bar with links to 'Arquivo', 'Editor', 'Exibir', 'Ir', 'Favoritos', 'Ferramentas', and 'Ajuda'. Below that is a sidebar with links to 'Webmin Index', 'Module Index', and 'Help..'. The main title is 'Edit Quota'. The central area displays 'Quota for kurumin ON /home'. It shows usage statistics: 'Blocks used' (12836611), 'Files used' (3845). It also shows limit settings: 'Soft block limit' (radio button selected) with a dropdown menu showing 'Unlimited' and '1000000'; 'Hard block limit' (radio button selected) with a dropdown menu showing 'Unlimited' and '1000000'. Similar sections are provided for 'Soft file limit' and 'Hard file limit'. Below these are 'Available blocks on disk' (28034012 total / 6431892 free) and 'Available files on disk' (0 total / 0 free). At the bottom left is an 'Update' button, and at the bottom right is a 'List All Quotas' button. A 'Return to user list' link is also present.

Aqui você tem a opção de configurar quotas individuais para cada usuário (/home users) ou quotas para grupos de usuários (/home groups). Os valores são informados em blocos. O mais comum é cada bloco ter 1 KB, mas o tamanho pode variar de 512 a 4096 bytes, de acordo com o tamanho da partição e do sistema de arquivos usado. Do lado direito você tem a opção de limitar também o número de arquivos que o usuário pode criar, opção menos usada.

The screenshot shows the 'Disk Quotas' interface. At the top, there's a navigation bar with links to 'Arquivo', 'Editor', 'Exibir', 'Ir', 'Favoritos', 'Ferramentas', and 'Ajuda'. Below that is a sidebar with links to 'Webmin Index', 'Help..', 'Module', and 'Config'. The main title is 'Disk Quotas'. On the right, there's a 'Search Docs..' link. The central area displays a table of mounted filesystems:

Filesystem	Type	Mounted From	Status	Action
/home (users)	Reiser Filesystem	IDE device A partition 6	User and Group Quotas Active	Disable Quotas
/home (groups)				

Below the table are buttons for 'Edit User Quotas' and 'Edit Group Quotas'. A 'Return to index' link is located at the bottom left.

O espaço ocupado por cada usuário é recalculado periodicamente, mas você pode atualizar as informações a qualquer momento clicando no "Check Quotas". Uma vez configurado, o Quota fica residente e é reativado automaticamente durante o boot, no momento em que a partição é montada. Não estranhe caso o sistema fique alguns minutos parado durante o

"Checking quotas" a cada boot; isso é normal, pois é necessário refazer a busca de arquivos.

Depois de configurar as Quotas de disco, você pode ativar a configuração (sem precisar reiniciar) remontando a partição:

```
# mount -o remount /home
```

No caso do **Fedora**, o SSL não é habilitado por padrão. Por isso, você deve acessar usando http, como em: <http://127.0.0.1:10000>.

Não existem muitos problemas em acessar o Webmin sem encriptação localmente, mas se você pretende acessá-lo a partir de outros micros, crie um túnel usando o SSL (capítulo 8). Se preferir ativar o SSL no Webmin (de forma a poder acessar via https), você pode seguir as instruções do link: <http://webmin.com/ssl.html>.

» Próximo: [Configurando o Bind](#)

Na configuração do Apache, especificamos o domínio e o diretório local correspondente para cada site. A idéia aqui é que o visitante digita o nome de domínio do site no navegador e o Apache se encarrega de enviá-lo ao diretório correto. Mas, para que o cliente chegue até o servidor, faltam mais duas peças importantes.

A primeira é o registro do **domínio**, que pode ser feito no Registro.br, Internic ou outro órgão responsável. Ao registrar, você precisa fornecer dois endereços de DNS. Na maioria dos casos, o segundo DNS não é obrigatório, ele é apenas uma segurança para o caso do primeiro sair fora do ar. Uma opção muito usada para o segundo DNS é pedir para que algum amigo que também possua um servidor dedicado seja seu DNS secundário. Ele precisará apenas adicionar a configuração do seu domínio na configuração do DNS, o que é rápido e indolor.

Em casos onde o sistema não permite continuar sem fornecer o segundo endereço, existe um pequeno truque: conecte-se via modem na hora de fazer o registro, assim você terá dois endereços (o do link e o do modem) e conseguirá concluir o registro. Naturalmente, neste caso, você perde a redundância: se o seu DNS principal cair seu site ficará fora do ar.

Note que nem sempre esta questão da redundância é realmente um problema, pois se o servidor DNS está hospedado no mesmo servidor que seu site, não faz muita diferença ter dois servidores DNS, pois se o servidor principal cair, o site ficará fora do ar de qualquer forma. Sites maiores possuem sistemas de redundância e muitas vezes servidores DNS separados, o que cria uma malha de segurança. É por isso que é muito raro a página de um portal ficar fora do ar, por exemplo.

É aqui que acaba o trabalho deles e começa o seu. Ao acessar o domínio, o visitante é direcionado para o endereço de DNS fornecido no registro. Isto significa que... bingo! além do Apache você vai precisar de um servidor DNS :).

Quando alguém tenta acessar um dos domínios registrados, a requisição vai do DNS do provedor para um dos 14 root servers disponíveis na internet, que são responsáveis por todos os domínios.

Na verdade, os root servers não armazenam uma tabela local. Ao invés disso eles redirecionam a requisição para o Registro.br, ou outra entidade responsável, que por sua vez redireciona para o seu servidor. O ciclo se fecha e o cliente consegue finalmente acessar a página.

Resolver um nome de domínio é uma operação que pode demorar alguns segundos, por isso os servidores DNS armazenam um cache de domínios já resolvidos, minimizando o número de requisições. É por isso que quando você faz alguma mudança na configuração do domínio, demoram algumas horas para que ela se replique.

Se seu servidor estiver hospedando subdomínios, ou seja, endereços como "www.fulano.guiadohardware.net", "www.ciclano.guiadohardware.net", etc., como fazem serviços como o hpg, a configuração continua basicamente a mesma. Você especifica o sub-domínio do cliente na configuração do VirtualHost do Apache e também no servidor de DNS.

Uma observação importante é que para o Apache, o domínio "www.fulano.guiadohardware.net" é diferente de apenas "fulano.guiadohardware.net". Para que o site possa ser acessado tanto com o www quanto sem ele, é necessário incluir um "ServerAlias" na configuração de cada site, como vimos a pouco, na configuração do Apache.

Como no caso anterior, você deve informar o endereço do seu servidor de DNS no registro do domínio. Como os servidores de registro de domínio lêem as URLs de trás para a frente, todos os acessos a subdomínios dentro do guiadohardware.net serão enviados para o seu servidor DNS e daí para o servidor Apache.

A **Internic** cuida dos registros dos domínios raiz (.com, .org, .info e outros), enquanto o **Registro.br** responde pelos domínios com extensão .br (.com.br, .org.br, etc.).

O registro de domínios na Internic é menos burocrático, pois você não precisa ter uma empresa registrada. De qualquer forma, registrando seu domínio no Registro.br ou na Internic, você precisará fornecer dois endereços de DNS, para onde serão enviadas as consultas referentes ao seu domínio.

O servidor DNS mais usado no Linux é o **Bind**, que aprenderemos a configurar aqui. Não existe problema em instalá-lo no mesmo servidor onde foi instalado o Apache e o Proftpd, embora do ponto de vista da segurança o ideal seja utilizar servidores separados ou um chroot.

Para instalar o Bind, procure pelo pacote "**bind**" ou "**bind9**" no gerenciador de pacotes da distribuição usada. No Debian ou Kurumin você instala com um "**apt-get install bind**" e no Mandriva com um "**urpmi bind**". No Slackware você encontra o pacote dentro da pasta "**n**"

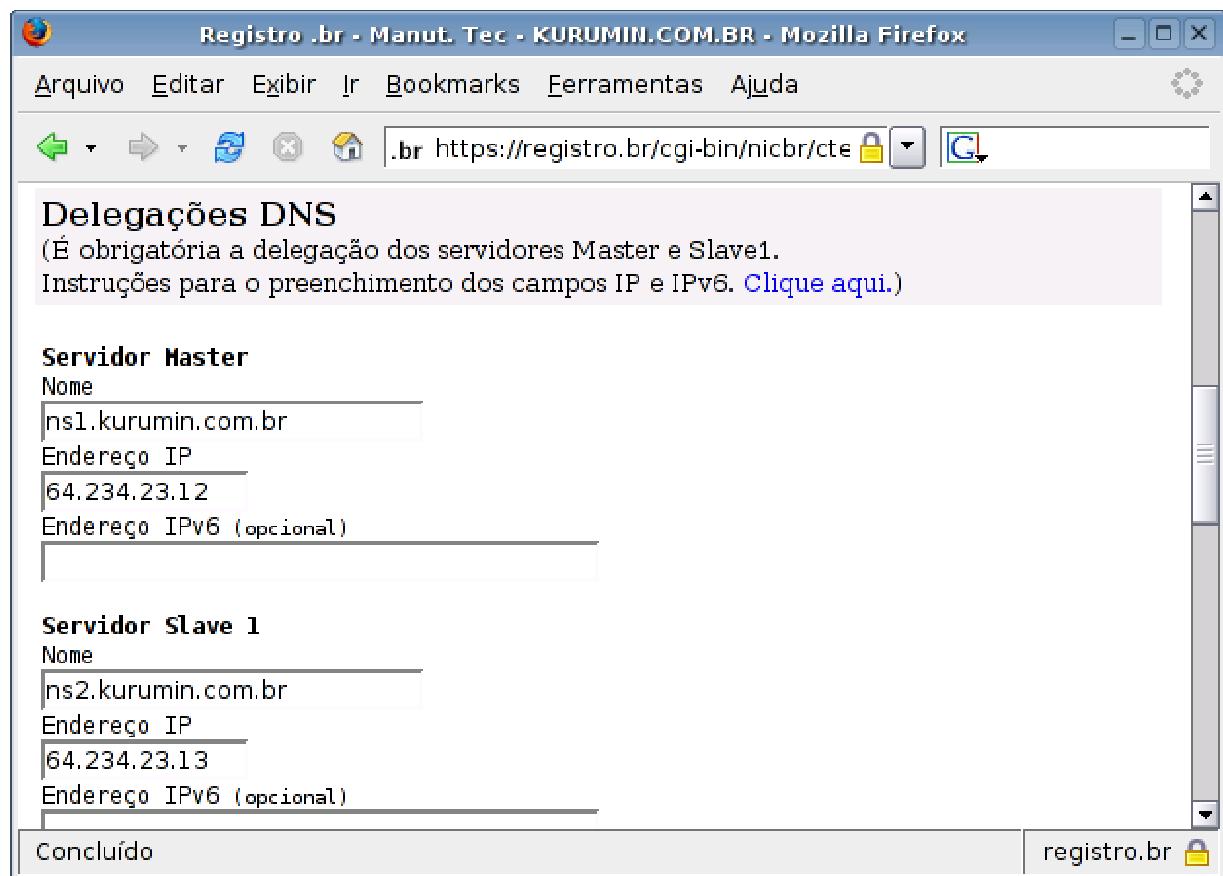
do primeiro CD. Ao instalar, verifique a versão incluída na distribuição. Use sempre o Bind 8 ou 9; nunca o Bind 4, que está em desuso.

O arquivo de configuração principal é o "**/etc/bind/named.conf**". Em versões antigas, o arquivo pode ser simplesmente "**/etc/named.conf**".

Por padrão, o Bind já vem configurado para trabalhar como um servidor DNS de cache para a rede local. Inicie o serviço com o comando "**/etc/init.d/bind start**" ou "**service bind start**" e configure os micros da rede interna para utilizarem o endereço IP do servidor onde foi instalado o bind como DNS (192.168.0.1 por exemplo), e você verá que eles já serão capazes de navegar normalmente, sem precisar mais do DNS do provedor.

O próximo passo é configurar o Bind para responder pelos domínios que você registrou. Vamos usar como exemplo o domínio "**kurumin.com.br**".

Como é um domínio .br, ele é registrado no Registro.br, através do <http://registro.br>. Depois de pagar e fornecer os dados da empresa e do responsável pelo domínio, é necessário fornecer os endereços dos dois endereços de DNS do seu servidor.



Naturalmente você vai precisar de um endereço IP válido e fixo. O ideal é usar um link dedicado, mas como eles são muito caros no Brasil, muitas empresas optam por usar

conexões ADSL. Em geral as operadoras fornecem endereços IP fixos nos planos empresariais.

Outra opção (mais recomendável, tanto do ponto de vista da confiabilidade, quanto do ponto de vista do custo) é alugar um servidor dedicado, hospedado em um datacenter estrangeiro. Os preços variam de US\$ 50 a US\$ 200 mensais, dependendo da configuração e link. Alguns exemplos de empresas de hospedagem são o <http://ev1servers.net/>, <http://www.serverbeach.com> e <http://www.layeredtech.com>.

Os servidores hospedados no Brasil acabam saindo sempre muito mais caro, devido ao custo dos links, que são muito mais caros aqui do que no exterior. Existem ainda algumas empresas brasileiras, como a <http://braslink.com>, que locam servidores hospedados no exterior.

Depois de se decidir sobre onde hospedar e concluir o registro do domínio, falta configurar o Bind para responder pelo domínio registrado. Vou usar como exemplo o domínio "kurumin.com.br".

Comece adicionando as seguintes linhas no final do arquivo "**/etc/bind/named.conf**" (sem modificar as demais):

```
zone          "kurumin.com.br"      IN      {  
type           master;  
file          "/etc/bind/db.kurumin";  
};
```

Ao usar um servidor DNS secundário, inclua a linha "allow-transfer", especificando o endereço IP do segundo servidor, como em:

```
zone          "kurumin.com.br"      IN      {  
type           master;  
file          "/etc/bind/db.kurumin";  
allow-transfer    {  
                    64.234.23.13;      };  
};
```

No caso do Debian, é recomendado que você use o arquivo "**/etc/bind/named.conf.local**" (que é processado como se fosse parte do named.conf principal), mas na verdade o efeito é o mesmo.

O **zone "kurumin.com.br"** na primeira linha indica o domínio que estamos configurando, como registrado no Registro.br.

O **"file "/etc/bind/db.kurumin"** especifica o arquivo onde vai a configuração deste domínio. Na verdade você pode salvar este arquivo em qualquer lugar, muita gente usa a pasta "**/var/named**". Aqui estou seguindo o padrão do Debian, colocando os arquivos dentro da pasta "**/etc/bind**", junto com os demais arquivos de configuração do Bind.

Em seguida você precisa adicionar a configuração do domínio no arquivo **"/etc/bind/db.kurumin"** que foi citado na configuração. O conteúdo do arquivo fica:

```

@      IN      SOA      servidor.kurumin.com.br.      hostmaster.kurumin.com.br.      (
2006040645          3H           15M           1W           1D      )
NS
IN      MX           10
kurumin.com.br.          A
www          A
ftp          A
smtp A 64.234.23.12

```

Neste arquivo, a formatação é importante. Você pode usar espaços e tabs (ambos tem o mesmo efeito) para organizar as opções, mas existem algumas regras. As linhas "IN SOA" até "IN MX" precisam ficar justificadas (como no exemplo), e você não pode esquecer dos espaços entre as opções.

Pesquisando no Google, você pode encontrar inúmeros templates como este, mas é difícil encontrar alguma explicação clara de cada uma das opções. Isso faz com que configurar um servidor DNS pareça muito mais complicado do que realmente é.

Vamos então a uma descrição detalhada de cada um dos campos:

```
@ IN SOA servidor.kurumin.com.br. hostmaster.kurumin.com.br. (
```

O "@" na primeira linha indica a origem do domínio e, ao mesmo tempo, o início da configuração. Ele é sempre usado, assim como num endereço de e-mail, por isso não é preciso se preocupar muito com ele. O "IN" é abreviação de "internet" e o "SOA" de "Start of authority". Em seguida vem o nome do servidor (que você checa usando o comando "hostname"), seguido do e-mail de contato do administrador.

Note que, no caso do e-mail, temos a conta separada do domínio por um ponto, e não por uma @. O mais comum é criar uma conta chamada "hostmaster", mas isso não é uma regra. Você poderia usar "fulaninho.meudomonio.com.br", por exemplo.

Note também que existe um ponto depois do "servidor.kurumin.com.br" e do "hostmaster.kurumin.com.br", que faz parte da configuração.

A linha diz algo como "Na internet, o servidor "servidor" responde pelo domínio kurumin.com.br e o e-mail do responsável pelo domínio é "hostmaster.kurumin.com.br".

A primeira linha termina com um parênteses, que indica o início da configuração do domínio. Temos então:

```
2006061645 8H 2H 1W 1D )
```

O "**2006061645**" é o valor de sincronismo, que permite que o servidor DNS secundário mantenha-se sincronizado com o principal, detectando alterações na configuração. Este número é composto da data da última alteração (como em: 20060616), e um número de dois dígitos qualquer que você escolhe. Sempre que editar a configuração, ou sempre que configurar um servidor DNS a partir de um template qualquer, lembre-se de atualizar a data e mudar os dois dígitos.

Os quatro campos seguintes orientam o servidor DNS secundário (caso você tenha um). O primeiro campo indica o tempo que o servidor aguarda entre as atualizações (8 horas). Caso ele perceba que o servidor principal está fora do ar, ele tenta fazer uma transferência de zona, ou seja, tenta assumir a responsabilidade sob o domínio. Caso a transferência falhe e o servidor principal continue fora do ar, ele aguarda o tempo especificado no segundo campo (2 horas) e tenta novamente.

O terceiro campo indica o tempo máximo que ele pode responder pelo domínio, antes que as informações expirem (1 semana, tempo mais do que suficiente para você arrumar o servidor principal ;) e o tempo mínimo antes de devolver o domínio para o servidor principal quando ele retornar (1 dia).

Estes valores são padrão, por isso não existem muitos motivos para alterá-los. A transferência do domínio para o DNS secundária é sempre uma operação demorada, por isso a principal prioridade deve ser evitar que o servidor principal fique indisponível em primeiro lugar.

Muita gente prefere especificar estes valores em segundos. Uma configuração muito comum é separar os valores por linha, como em:

2006061645;		serial
28800;	refresh,	seconds
7200;	retry,	seconds
604800;	expire,	seconds
86400); minimum, seconds		

O resultado é exatamente o mesmo. A única diferença é que você vai acabar digitando várias linhas a mais ;).

As duas linhas que vem a seguir concluem a seção inicial:

NS		servidor.kurumin.com.br.
IN MX 10 servidor.kurumin.com.br.		

A primeira, diz quem são as máquinas responsáveis pelo domínio. Ao usar apenas um servidor DNS, você simplesmente repete o nome do servidor, seguido pelo domínio, como adicionamos na primeira linha. Caso você esteja usando dois servidores, então você precisa declarar ambos, como em:

NS		servidor.kurumin.com.br.
NS ns2.kurumin.com.br.		

A linha "IN MX" é necessária sempre que você pretende usar um servidor de e-mails (você pode escolher entre usar o Postfix, Qmail, Sendmail ou outro MTA). Aqui estou simplesmente usando a mesma máquina para tudo, por isso novamente citei o "servidor.kurumin.com.br", que acumula mais esta função. Assim como no caso do DNS, você pode especificar um servidor de e-mails secundário, que passa a receber os e-mails caso seu servidor principal saia fora do ar. Neste caso você adiciona uma segunda linha, como em:

IN	MX	10	servidor.kurumin.com.br.
	IN MX 20 outroservidor.outrodominio.com.br.		

Os números indicam a prioridade de cada servidor. O servidor da primeira linha tem prioridade 10, por isso é o primário. O segundo tem prioridade 20 e por isso só assume em casos de problemas com o primário. Usar um segundo servidor de e-mails, num domínio separado, adiciona uma camada extra de redundância e evita que você perca e-mails caso seu servidor fique temporariamente fora do ar.

Depois destas linhas iniciais, temos a parte mais importante, onde você especifica o IP do servidor e pode cadastrar subdomínios, como em:

kurumin.com.br.	A	64.234.23.12
www	A	64.234.23.12
ftp	A	64.234.23.12
smtp	A 64.234.23.12	

Neste exemplo, inclui também dois subdomínios, o "www" e "ftp", ambos relacionados ao IP do servidor. Isso permite que os visitantes digitem "www.kurumin.com.br" ou "ftp.kurumin.com.br" no navegador. Ao trabalhar com subdomínios, você pode relacioná-los com IP's ou domínios diferentes. Por exemplo, muitos portais possuem vários subdomínios, como "www1", "www2", "www3" e assim por diante, onde cada um é um servidor diferente.

Sites como o <http://no-ip.com>, que oferecem "domínios virtuais", que apontam para seu endereço IP atuais são na verdade grandes bases de dados, atualizadas automaticamente, onde cada subdomínio aponta para o IP atual do dono.

Ao trabalhar com dois servidores DNS, adicione também uma entrada para ele, especificando o nome do segundo servidor (ns2 no exemplo) e o IP, como em:

kurumin.com.br.	A	64.234.23.12
www	A	64.234.23.12
ftp	A	64.234.23.12
smtp	A	64.234.23.12
ns2 A 64.234.23.13		

Note o segundo DNS usa o IP .13, enquanto o servidor principal usa o .12, mas ambos estão dentro da mesma faixa de IP's. É comum que ao locar um servidor dedicado, você receba dois endereços IP (quase sempre dentro da mesma faixa), permitindo que você configure o segundo DNS.

Você pode testar a configuração do seu servidor DNS usando o comando **dig**. No Debian ele é instalado juntamente com o pacote "dnsutils".

Faça uma busca pelo domínio, especificando o endereço IP do DNS que acabou de configurar, como em:

```
$ dig kurumin.com.br @64.234.23.12
```

Isso faz com que ele pergunte diretamente ao seu servidor, o que permite testar a configuração imediatamente, sem precisar esperar pela propagação do registro do domínio. Se tudo estiver correto, você verá algo como:

```
;;
ANSWER
kurumin.com.br. 86400 IN A 64.234.23.12 SECTION:

;; AUTHORITY SECTION:

gdhpress.com.br. 86400 IN NS servidor.kurumin.com.br.
gdhpress.com.br. 86400 IN NS ns2.kurumin.com.br.
```

Faça o mesmo com o IP do DNS secundário, como em:

```
$ dig kurumin.com.br @64.234.23.13
```

» Próximo: [DNS Reverso](#)

O DNS reverso é um recurso que permite que outros servidores verifiquem a autenticidade do seu servidor, checando se o endereço IP atual bate com o endereço IP informado pelo servidor DNS. Isso evita que alguém utilize um domínio que não lhe pertence para enviar spam, por exemplo.

Qualquer um pode enviar e-mails colocando no campo do remetente o servidor do seu domínio, mas um servidor configurado para checar o DNS reverso vai descobrir a farsa e classificar os e-mails forjados como spam.

O problema é que os mesmos servidores vão recusar seus e-mails, ou classificá-los como spam caso você não configure seu servidor DNS corretamente para responder às checagens de DNS reverso. Chegamos a um ponto em que o problema do spam é tão severo, que quase todos os servidores importantes fazem esta checagem, fazendo com que, sem a configuração, literalmente metade dos seus e-mails não sejam entregues.

O primeiro passo é checar os arquivos "**/etc/hostname**" e "**/etc/hosts**" (no servidor), que devem conter o nome da máquina e o domínio registrado.

O arquivo "**/etc/hostname**" deve conter apenas o nome da máquina, como em:

```
servidor
```

No Fedora e em algumas outras distribuições, o nome da máquina vai dentro do arquivo "**/etc/sysconfig/network**".

No arquivo "**/etc/hosts**" deve conter duas entradas, uma para a interface de loopback, o 127.0.0.1, e outra para o IP de internet do seu servidor, que está vinculado ao domínio, como em:

```
127.0.0.1           localhost.localdomain      localhost
64.234.23.12        servidor.kurumin.com.br    servidor
```

A partir daí, falta adicionar a zona reversa no bind complementando a configuração do domínio, que já fizemos. Começamos adicionando a entrada no "/etc/bind/named.conf" ou "/etc/bind/named.conf.local":

```
zone "23.234.64.in-addr.arpa" {
    type master;
    file "/etc/bind/db.kurumin.rev";
};
```

O endereço IP do servidor é 64.234.23.12. Se retiramos o último octeto e escrevemos o restante do endereço de trás pra frente, temos justamente o "23.234.64" que usamos no registro reverso. A terceira linha indica o arquivo onde a configuração do domínio reverso será salva. Neste caso indiquei o arquivo "db.kurumin.rev", mas você pode usar qualquer nome de arquivo.

Este arquivo "db.kurumin.rev" é bem similar ao arquivo com a configuração do domínio, que acabamos de configurar. As três linhas iniciais são idênticas (incluindo o número de sincronismo), mas ao invés de usar o "A" para relacionar o domínio e cada subdomínio ao IP correspondente, usamos a diretiva "PTR" para relacionar o endereço IP de cada servidor ao domínio (é justamente por isso que chamamos de DNS reverso ;).

No primeiro arquivo, usamos apenas os três primeiros octetos do endereço (a parte referente à rede), removendo o octeto final (o endereço do servidor dentro da rede). Agora, usamos apenas o número omitido da primeira vez.

O IP do servidor é "64.234.23.12", removendo os três primeiros octetos ficamos apenas com o "12". Temos também o endereço do DNS secundário, que é 64.234.23.13, de onde usamos apenas o "13". Relacionando os dois a seus respectivos domínios, o arquivo fica:

```
@ IN SOA servidor.kurumin.com.br. hostmaster.kurumin.com.br. (
2006040645 3H 15M 1W 1D )
NS servidor.kurumin.com.br.
12 PTR kurumin.com.br.
13 PTR ns1.kurumin.com.br.
```

Caso você não esteja usando um DNS secundário, é só omitir a linha referente a ele, como em:

```
@ IN SOA servidor.kurumin.com.br. hostmaster.kurumin.com.br. (
2006040645 3H 15M 1W 1D )
NS servidor.kurumin.com.br.
12 PTR kurumin.com.br.
```

Depois de terminar, reinicie o Bind e verifique usando o **dig**. Comece checando o domínio, como em:

```
# dig kurumin.com.br
```

Na resposta, procure pela seção "ANSWER SECTION", que deverá conter o IP do servidor, como configurado no bind:

```
;; ANSWER SECTION:
kurumin.com.br. 86400 IN A 64.234.23.12
```

Faça agora uma busca reversa pelo endereço IP, adicionando o parâmetro "-x", como em:

```
# dig -x 64.234.23.12
```

Na resposta você verá:

;;	ANSWER	SECTION:
	12.23.234.64.in-addr.arpa. 86400 IN PTR	kurumin.com.br.

Ou seja, com o DNS reverso funcionando, o domínio aponta para o IP do servidor e o IP aponta para o domínio, permitindo que os outros servidores verifiquem a autenticidade do seu na hora de receber e-mails provenientes do seu domínio.

Lembre-se que seus e-mails podem ser classificados como spam também se seu IP estiver marcado em alguma blacklist. Você pode verificar isso rapidamente no <http://rbls.org/>.

Você vai notar, por exemplo, que praticamente endereço IP de uma conexão via ADSL ou modem vai estar listado, muitas vezes "preventivamente", já que é muito comum que conexões domésticas sejam usadas para enviar spam. É recomendável verificar periodicamente os IP's usados pelo seu servidor, além de verificar qualquer novo IP ou link antes de contratar o serviço.

» Próximo: [Capítulo 8: Acesso remoto](#)

Uma vantagem no uso do Linux, apontada por muitos administradores de rede, é a facilidade de administrar o sistema remotamente, tanto via linha de comando (usando o SSH) quanto com acesso à interface gráfica (usando o VNC, FreeNX, ou o próprio SSH). Essa é uma necessidade para qualquer um que mantém diversos servidores, em diferentes locais.

Hoje em dia poucas empresas hospedam seus websites "in house", ou seja, em servidores instalados dentro da própria empresa. Quase sempre os servidores ficam hospedados em data centers, complexos que oferecem toda a estrutura necessária para que os servidores fiquem no ar de forma confiável, incluindo links redundantes (se o link principal cai, existe um segundo de reserva), no-breaks de grande porte, geradores, refrigeração (a temperatura ambiente mais baixa ajuda os componentes a trabalharem de forma mais estável) e assim por diante.

Isso significa que apesar do servidor ser "seu", você não tem nenhum tipo de acesso físico a ele. Não pode usar o teclado ou mouse por exemplo, tudo precisa ser feito a distância. No Linux, toda a configuração do sistema, instalação de novos programas, etc. pode ser feita a partir do modo texto, o que permite configurar o servidor e mantê-lo atualizado remotamente, via SSH. Outro ponto interessante é que, apesar de ser nativo do Unix, existem clientes SSH também para Windows e outras plataformas, permitindo que o responsável administre o servidor a partir de uma estação Windows, por exemplo.

Outra possibilidade interessante para o SSH é o suporte a distância. Você pode se conectar no micro de um amigo ou cliente para corrigir algum problema. Praticamente tudo pode ser feito remotamente. Desde a instalação de um novo Kernel (você instala, configura o lilo ou grub e em seguida reinicia a máquina torcendo para que tudo esteja correto e ela volte depois de dois minutos), até uma reinstalação completa do sistema, onde você instala a nova cópia em uma partição separada, usando um chroot, e configura o gerenciador de boot para iniciá-la por default depois do reboot.

Outro uso comum, desta vez dentro das redes locais, é o uso remoto de aplicativos. Em muitas situações faz sentido instalar determinados aplicativos em um servidor central e abrir sessões remotas nos clientes. Isso permite centralizar as informações no servidor (facilitando os backups) e, ao mesmo tempo, usar menos recursos nos clientes, permitindo o uso de micros mais antigos.

O exemplo mais desenvolvido é o LTSP, que permite usar micros antigos, de praticamente qualquer configuração, como clientes de um servidor rápido. Um único servidor pode atender a 20 ou até mesmo 30 clientes. Este capítulo trata das diferentes formas de acesso remoto disponíveis e das aplicações para cada uma.

» Próximo: [Usando o VNCserver](#)

O VNC permite acessar remotamente uma máquina rodando o Windows, Linux, MacOS ou praticamente qualquer outro sistema a partir de outro PC, ou até mesmo de um palmtop. Ele é uma ferramenta essencial que ajuda a resolver a falta de conectividade entre os vários sistemas operacionais que temos em uso.

Um dos problemas mais comuns, que qualquer um se depara ao tentar ajudar um cliente, ou amigo pelo telefone, a resolver problemas do tipo "meu PC está travando", é que nem sempre o usuário saberá lhe dizer exatamente o que está se passando. Frases do tipo "apareceu uma janelinha piscando" nem sempre ajudam muito :-). Outro caso comum é alguém que trabalha em trânsito, ou viaja muito, e precisa acessar arquivos ou programas que estão no PC de casa.

O VNC é até semelhante a programas como o PC Anyware, mas traz a vantagem de ser aberto e gratuito. Além disso, ele é bem simples de usar e tem versões para Linux, Windows, Solaris, BeOS, Amiga e até mesmo para palmtops Pocket PC ou Palm. Ele pode ser usado tanto para acessar PCs ligados em uma rede local, quanto via internet.

O VNC se divide em dois módulos: o módulo servidor e o módulo cliente. O módulo servidor deve ser instalado no micro que ficará acessível, bastando usar o módulo cliente para acessá-lo de qualquer outro. O mais interessante é que os módulos são intercompatíveis, ou seja, você pode acessar uma máquina rodando Linux a partir de outra que roda Windows, ou mesmo acessar ambas a partir de um Pocket PC com rede wireless.

O programa exibe uma janela com o mesmo conteúdo da área de trabalho do micro que está sendo acessado, permitindo que você o utilize como se estivesse de frente para ele. Isso é

perfeito para quem trabalha com suporte, pois basta pedir para o usuário abrir o programa ao invés de ficar perguntando pelo telefone, o que torna o atendimento bem mais rápido.

O servidor VNC pode ser encontrado na maior parte das distribuições na forma do pacote "**vncserver**" ou "**tightvncserver**". Você pode também baixar o pacote disponível no <http://www.tightvnc.com/download.html> que, embora antigo e compilado para o Red Hat 7, ainda funciona na maioria das distribuições.

Para abrir o servidor VNC, basta usar (como usuário, não como root) o comando:

```
$ vncserver  
(ou tightvncserver)
```

Da primeira vez que é executado, ele pede para definir uma senha de acesso, que deve ser fornecida pelos clientes. O VNCserver abre uma sessão separada do servidor X, independente da tela :0 que você vê no monitor. Isso permite que outras pessoas acessem o mesmo PC ou servidor sem atrapalhar quem está trabalhando nele.

O servidor VNC abre o desktop do usuário que o chamou, com as mesmas permissões de acesso. Justamente por isso, você nunca deve abri-lo como root, salvo tarefas de manutenção em que isso seja realmente necessário. Ao dar acesso à sua máquina para outra pessoa, o ideal é criar um novo usuário e usá-lo para abrir o VNC. Assim, você não corre o risco do seu amigo sair deletando seus arquivos e alterando suas configurações.

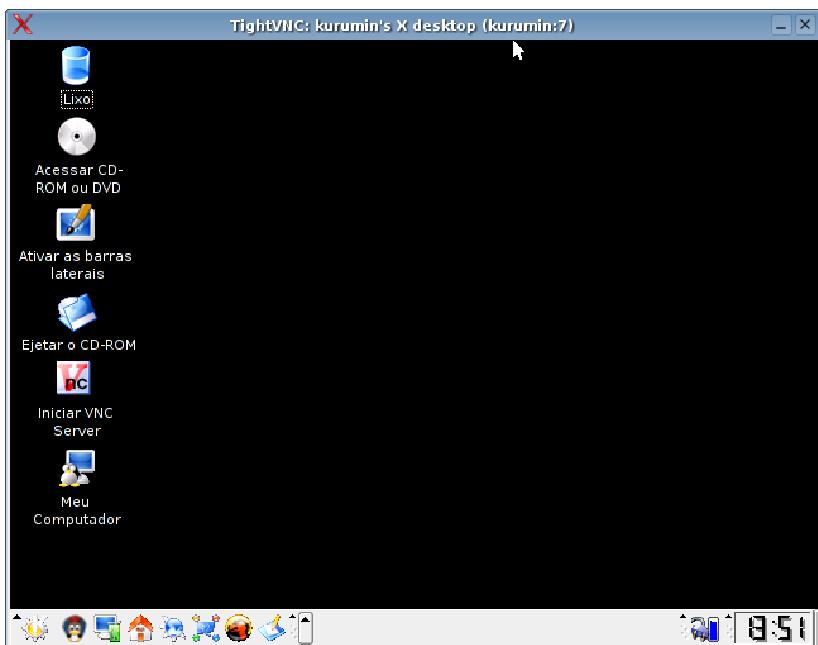
É possível abrir várias sessões independentes executando o comando VNCserver várias vezes. As sessões do VNC são numeradas de :1 em diante. Cada vez que o comando VNCserver é executado, você recebe uma mensagem avisando sobre o número da sessão aberta:

```
New 'X' desktop is kurumin:1  
Starting applications specified in /home/kurumin/.vnc/xstartup  
Log file is /home/kurumin/.vnc/kurumin:1.log
```

Para acessar a partir de outra máquina Linux, use o comando:

```
$ vncviewer 192.168.0.1:1
```

... onde o "192.168.0.1" é o IP da máquina que está sendo acessada e o ":1" é o número da sessão. Você pode utilizar também o **krdc** (Conexões com Ambiente de Trabalho Remoto), um aplicativo que faz parte do KDE. Ele pode ser utilizado tanto como cliente do VNC, quanto fazer par com o **krfb**, que permite compartilhar a área de trabalho (a tela :0, local) através de uma sessão do VNC, uma possibilidade interessante para fins de suporte.



O VNCserver do Debian vem configurado para abrir uma tela de 1024x768 com 8 bits de cor (no Kurumin o padrão é 1012x704 com 16 bits de cor, para ocupar quase toda a tela, mas sem cobrir a barra de tarefas do KDE ou do Windows). Para alterar a configuração padrão, abra o arquivo `/etc/vnc.conf` e procure pelas linhas:

```
$geometry = "1012x704";
$depth = "16";
```

Você pode alterá-las para o valor desejado. O VNC pode ser aberto em qualquer resolução (como os 1012x704 do Kurumin), não é necessário usar uma das resoluções padrão. Uma opção para abrir diversas sessões do VNC, com resoluções diferentes, é usar o comando VNCserver com os parâmetros **`-depth`** e **`-geometry`**, como em:

```
$ vncserver -depth 16 -geometry 800x600
```

Em geral, o VNCserver vem configurado nas distribuições para usar o KDE ou Gnome como gerenciador de janelas padrão nas sessões abertas. Dependendo da versão do VNC, você pode alterar esta configuração dentro do arquivo `.xsession` ou `.vnc/xstartup`, dentro do diretório home. Substitua o "startkde" ou "gnome-desktop" pelo comando que inicializa o gerenciador de janelas desejado, como, por exemplo, "wmaker" ou "fluxbox".

Uma vez abertas, as sessões do VNC ficam abertas até que o servidor seja reiniciado. Isso é bom em muitas situações, pois você pode se conectar a partir de diferentes micros da rede e sempre continuar exatamente de onde parou, sem precisar ficar abrindo e fechando a sessão. Mas, por outro lado, isso é ruim, pois sempre acabam sobrando sessões "esquecidas". Para fechar as sessões, use a opção `-kill`, seguida pelo número da sessão, como em:

```
$ vncserver -kill :1
```

Outra possibilidade interessante é acessar o cliente usando um browser com suporte a Java. O VNC inclui um mini-servidor http, que permite que as sessões compartilhadas sejam acessadas a partir de qualquer navegador com o suporte a Java instalado. Na maioria das distribuições, o módulo vem incluído no pacote do VNC ou TightVNC, enquanto em outras é necessário instalar o pacote "vnc-java". Com o módulo instalado, não é necessário fazer nenhuma configuração adicional para ativá-lo; ele fica automaticamente disponível sempre que uma sessão do VNC for aberta.

No cliente, basta abrir o navegador e acessar o endereço "<http://192.168.0.1:5801>", onde o "192.168.0.1" é o IP do servidor e o "5801" indica o número da sessão. A sessão :1 é acessada pela porta 5801, a sessão :2 pela 5802, e assim por diante. No VNC Server for Windows, onde é compartilhada a tela local, acesse a porta 5800.

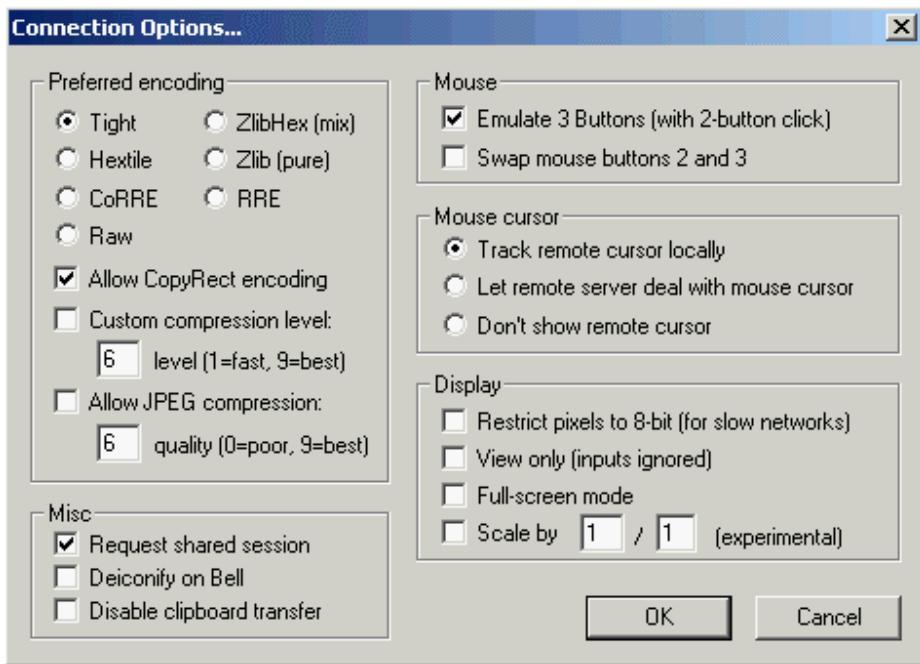
O applet Java é uma versão simplificada do cliente VNC e pode ser acessado tanto a partir do Firefox e IE, quanto a partir de outros navegadores com suporte a Java. Ele é uma opção útil em máquinas onde o cliente VNC não esteja instalado, mas tem a desvantagem de ser perceptivelmente mais lento.

Note que a porta usada pelo Applet Java (5800 + o número de tela) é diferente da porta usada pelo servidor VNC propriamente dito (acessado pelo cliente VNC), que precisa ser aberta no firewall do servidor. O servidor regular utiliza a porta 5900 em diante: 5901 para a sessão :1, 5902 para a sessão :2, e assim por diante.

» Próximo: [TightVNC](#)

Como vimos, na hora de instalar existem freqüentemente duas opções, o VNCserver "normal" e o TightVNCserver, uma versão mais recente, que oferece um algoritmo de compressão mais eficiente. Ele garante tempos de atualização de tela mais baixos (à custa de um pouco mais de processamento no cliente) e também suporte à compressão via JPG, que degrada a qualidade da imagem, em troca de uma redução considerável na banda utilizada, permitindo executar o VNC de forma aceitável mesmo em uma conexão via modem.

O TightVNC também oferece algumas melhorias secundárias, como o suporte a scroll de tela usando a roda do mouse e processamento local do cursor do mouse. Para que esses recursos funcionem, é necessário que seja utilizado o TightVNC tanto no servidor quanto no cliente. No screenshot temos o menu de opções oferecidas pelo cliente for Windows.



As opções são as seguintes:

Emulate 3 Buttons: Pressionar os dois botões simultaneamente equivale a pressionar o terceiro botão do mouse. Esta opção é útil para quem usa um mouse de dois botões e está acessando um servidor Linux.

Swap mouse buttons 2 and 3: Troca os dois botões do mouse dentro da tela do VNC, para canhotos ou caso os dois micros estejam configurados de forma diferente.

Track remote cursor locally: Processa o movimento do mouse no cliente e não no servidor. Uma novidade do Tight que faz o movimento do mouse ficar muito mais uniforme.

Restric pixels to 8-bit: Usa apenas 8 bits de profundidade de cor. Na prática, não faz tanta diferença, graças aos algoritmos de compactação. O mais útil para aumentar o desempenho no VNC é diminuir o tamanho da tela.

Full-screen mode: Inicia direto em tela cheia (no cliente).

Request shared session: Permite que dois ou mais clientes se conectem ao mesmo tempo à mesma sessão do servidor. Nesse caso, os movimentos do mouse e o input do teclado são misturados, de forma que só um pode usá-la de cada vez. Esta opção é útil para treinamentos.

Opções de encriptação (Preferred encoding):

Esta é a configuração mais importante, que vai definir o desempenho do VNC. Cada um dos algoritmos diferentes apresenta um certo balanço entre uso da banda da rede e carga de processamento. Por isso, a melhor escolha varia de acordo com a situação:

Tight: Este é o algoritmo exclusivo do Tight, que pode ser usado apenas quando tanto o cliente quanto o servidor utilizam a versão. O Tight oferece uma dupla compressão de dados, uma semelhante ao PNG, buscando pixels repetidos e substituindo-os por um único código, e uma segunda camada, baseada em um algoritmo de compressão desenvolvido pela equipe. É possível ativar, ainda, a compressão via JPG, estipulando um nível de perda.

O Tight é o ideal para redes lentas, sobretudo conexões via modem, mas não é uma boa escolha para redes locais ou micros muito lentos, pois a carga extra de processamento faz com que a atualização de tela fique lenta ao usar dois micros relativamente antigos (dois Pentium III 600, por exemplo), mesmo via rede local.

Hextile: Este algoritmo é o usado pela versão tradicional do VNC. A imagem da tela é dividida em áreas de 16x16 pixels e apenas as áreas atualizadas vão sendo enviadas aos cliente de forma compactada. Este algoritmo é o que oferece o melhor balanço entre uso da rede e do processador. É recomendável para PCs acima de 233 MHz e redes locais de 10 ou 100 megabits. Este é o algoritmo que oferece respostas mais rápidas ao utilizar uma rede de 100 megabits e dois PCs relativamente atuais, a melhor opção se você deseja algo rápido o suficiente para rodar aplicativos de trabalho confortavelmente.

RRE: É um algoritmo mais simples de compactação, que se resume a substituir seqüências de pixels da mesma cor por um único valor, indicando apenas que o cliente deve repetir o pixel x vezes. É eficiente para visualizar menus, textos, etc., mas não para imagens. Não se esqueça de desativar o wallpaper :).

CoRRE: Esta é uma variação do RRE que limita o número de repetições de pixels a 255, permitindo enviar um único bit de dados. Combina um uso relativamente baixo da banda da rede com pouco processamento. É o algoritmo que oferece melhores resultados em micros antigos e rede de 10 megabits.

Zlib (pure): Usa o algoritmo Zlib para compactar as imagens, sem perda de qualidade. É o segundo mais eficiente em nível de compressão, perdendo apenas para o Tight. Apesar disso, a carga de processamento no Zlib é consideravelmente maior que a do Tight, mais que o dobro, em muitas situações. O Zlib continua disponível no Tight apenas para manter compatibilidade com o VNC tradicional, que não suporta o algoritmo Tight.

ZlibHex (mix): Combina o Zlib com o Hexlite para quebrar a tela em pequenos pedaços, mantendo a compressão com o Zlib. O uso do processador é semelhante ao Zlib pure, mas existe um ganho perceptível de velocidade quando pequenos pedaços da tela são atualizados (abertura de menus por exemplo), mas nas atualizações de tela inteira, ao abrir uma nova janela ou dar scroll em uma página aberta no browser, por exemplo, o Zlib puro se sai melhor.

Raw: É o oposto do Tight. As imagens são enviadas via rede sem compressão alguma, minimizando a carga sobre o processador. Pode ser útil em redes de 100 megabits, mas com micros muito lentos, abaixo de 133 MHz. A quantidade de dados enviada através da rede é de 50 a 100 vezes maior que a do Tight (num dos testes publicados, uma sessão de 6:30 min em Raw totalizou um tráfego de 217 MB, contra apenas 3.3 MB usando o Tight), mas, em compensação, a carga de processamento é quase nula.

Você pode ver o comparativo entre a eficiência dos algoritmos de compressão, feito pela equipe do Tight no: <http://www.tightvnc.com/compare.html>. Usando o cliente Windows, basta selecionar as opções desejadas ao fazer a conexão, como no screenshot que vimos acima. No cliente Linux é preciso passar os parâmetros via linha de comando ao conectar. A sintaxe é:

```
$ vncviewer opções ip_do_servidor:sessão
```

As opções podem incluir:

-encodings: Para especificar um dos algoritmos de compactação acima.
Ex: vncviewer -encodings CoRRE 192.168.0.1:1

-fullscreen: Para iniciar o VNC em modo de tela cheia (o default é abrir em uma janela, o que, muitas vezes, faz com que apareçam barras de rolagem).
Ex: vncviewer -encodings Raw -fullscreen 192.168.0.1:1

-compresslevel 9: Esta opção permite especificar o nível de compressão para os algoritmos Tight e Zlib (a opção não tem efeito algum com os demais), permitindo dosar o uso da rede e do processador. O número vai de 1 (pouca compressão, menos processamento) a 9 (máxima compressão). O número 0 equivale ao modo Raw, sem compressão
alguma.

Ex: vncviewer -encodings Zlib -compresslevel 9 220.200.125.67:3

-quality 2: Aqui é possível especificar o nível de compressão via JPG para o algoritmo Tight, especificando um número de 0 (péssima qualidade, menor uso da rede) a 9 (compressão sem perda, o default). Esta opção pode ser combinada com a opção -compresslevel. A opção mais rápida possível no VNC para uma conexão via modem seria:

```
$ vncviewer -encodings Tight -compresslevel 9 -quality 0 220.200.125.23:2
```

Usando a opção -quality 0, a qualidade da imagem fica realmente sofrível, mas as áreas por onde o mouse passa são atualizadas usando a qualidade máxima, permitindo que você consiga ver os detalhes. Apesar disso, é o melhor meio de conseguir ter uma velocidade utilizável através de uma conexão via modem.

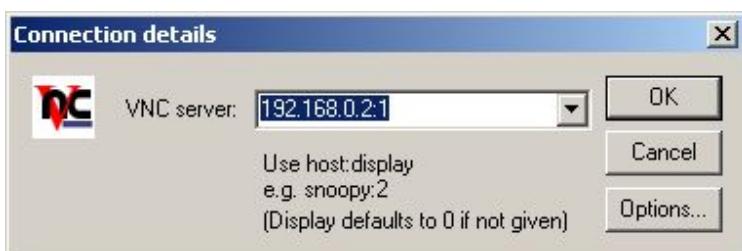
-viewOnly: Para apenas visualizar o host remoto, sem input do teclado ou mouse. É uma opção útil para apresentações, aulas, etc.
Ex: vncviewer -encodings Raw -fullscreen -viewOnly 192.168.0.1:1

» Próximo: [VNC no Windows](#)

Você pode baixar o VNC for Windows no mesmo link que citei anteriormente: <http://www.tightvnc.com/download.html>.

Para instalar o servidor, basta executar o programa, concordar com as licenças, etc. Para ativar o compartilhamento, clique em "Run WinVNC", dentro da pasta VNC do menu iniciar. Forneça uma senha de acesso, preferencialmente uma bem elaborada (afinal, é só você quem deve ter acesso ao micro, não a internet Inteira ;), e mantenha o programa residente.

Para acessar, abra o módulo cliente, digite o endereço IP do servidor e forneça a senha. Se você não souber o endereço IP, digite "netstat -r" ou "ipconfig" no prompt do DOS.



O VNC serve apenas como um terminal, gráfico, não permite transferir arquivos diretamente. Caso seja necessário, o melhor é complementá-lo usando um outro programa, como um servidor de FTP, por exemplo. Outra dica importante ao acessar uma máquina Windows via VNC é sempre desabilitar a aceleração de vídeo, o que pode ser feito em Painel de controle > Video > Configurações > Avançado > Soluções de problemas.



A aceleração de vídeo faz com que a própria placa de vídeo faça parte do trabalho de atualização da tela, o que melhora bastante a velocidade da atualização da tela no monitor local, mas é um tormento para o servidor VNC, que não tem como acompanhar as atualizações feitas pela placa de vídeo, apenas as feitas via software. O resultado é que a imagem via VNC fica bastante falhada, desagradável de usar. Ao desativar a aceleração de vídeo temporariamente, tudo volta ao normal.

Ao contrário do servidor Linux, no Windows o VNC sempre compartilha a tela local. Remotamente você enxerga exatamente a mesma imagem exibida no monitor, fazendo com que, no Windows, o VNC seja mais uma ferramenta de administração remota do que um sistema para rodar aplicativos remotamente.

» Próximo: [Usando o SSH](#)

O SSH é a minha ferramenta preferida. Ele permite administrar máquinas remotamente, executando inclusive aplicativos gráficos, permite transferir arquivos de várias formas diferentes e, como se não bastasse, permite também encapsular outros protocolos, permitindo, por exemplo, acessar uma sessão do VNC através de um túnel seguro.

A grande vantagem do SSH sobre outras ferramentas de acesso remoto é a grande ênfase na segurança. Um servidor SSH bem configurado é virtualmente impenetrável e você pode acessá-lo de forma segura, mesmo que a sua rede local esteja comprometida. Ele utiliza um conjunto de técnicas de criptografia para assegurar que apenas as pessoas autorizadas terão acesso ao servidor, que todos os dados transmitidos sejam impossíveis de decifrar e que a integridade da conexão seja mantida.

São previstas respostas para diversos tipos de ataques conhecidos. O SSH detecta casos em que o servidor tenha sido substituído por outra máquina, situações nas quais se tenta injetar dados na conexão, ou seja, tirar proveito de uma conexão aberta para incluir pacotes com comandos adicionais e inclui até mesmo técnicas de "despiste", que tornam muito mais complicado descobrir em qual pacote encriptado foi transmitida a senha de acesso, por exemplo, dificultando a vida de quem pretende descobrir a senha usando um ataque de força-bruta.

A idéia central é que, mesmo em situações onde seja fácil interceptar a transmissão (como no caso de uma rede wireless pública), seja impossível descobrir o conteúdo dos pacotes, devido à encriptação. É possível, ainda, utilizar um par de chaves ao invés de uma simples senha como forma de autenticação. Nesse caso, além da chave (um arquivo salvo no HD, pendrive ou smartcard), é preciso saber a passphrase, que pode ser uma senha especialmente longa e difícil de adivinhar.

Qualquer algoritmo de encriptação pode ser quebrado via força bruta, onde simplesmente são testadas todas as possibilidades possíveis, até encontrar a combinação correta. Porém, isso só é realmente possível para chaves de 40 ou no máximo 64 bits; acima disso é inviável, pois a cada bit adicionado, o processo torna-se exponencialmente mais demorado.

O WEP de 64 bits (que na verdade utiliza uma chave de 40 bits), usado em redes wireless pouco protegidas, pode ser quebrado em pouco tempo, caso você consiga capturar um volume considerável de transmissões usando um sniffer. O DES, um dos algoritmos mais tradicionais, que usa chaves de 64 bits (reais), pode ser quebrado em alguns dias, caso você tenha acesso a um cluster de 100 máquinas Athlon 64.

Uma chave de 64 bits é cerca de 16 milhões de vezes mais difícil de quebrar via força bruta do que uma de 40 bits, como as que eram utilizadas no SSL dos navegadores a até poucos anos atrás. Uma chave de 128 bits por sua vez, é (arredondando)

18.447.000.000.000.000 vezes mais demorada de quebrar que uma de 64 bits, de forma que, uma chave de 64 bits pode ser quebrada caso você tenha o tempo e os recursos necessários à disposição, mas uma de 128 (sem brechas conhecidas) é impossível de quebrar com tecnologia atual.

O perigo no caso dos algoritmos de encriptação é quando são descobertas falhas que permitam descobrir a chave usada em menos tempo. As versões originais do WEP, por exemplo, podiam ser quebradas rapidamente devido a um conjunto de falhas no algoritmo usado, o que levou os fabricantes a atualizarem rapidamente todos os seus produtos. Outro exemplo é o sistema usado na encriptação dos DVDs, que é quebrado em poucos segundos por uma máquina atual, utilizando um algoritmo de poucas linhas.

Felizmente, este não é o caso dos algoritmos usados no SSH. Por serem abertos, qualquer falha similar que pudesse eventualmente existir já teria sido descoberta e corrigida. O SSH é usado em tantos servidores importantes que uma brecha grave poderia (literalmente) parar o mundo. Por isso, todo o código é exaustivamente auditado por uma variedade de empresas e órgãos governamentais.

O SSH utiliza chaves assimétricas para fazer a autenticação. As chaves assimétricas são um sistema muito interessante, onde temos um par de chaves. Uma (a chave pública), permite apenas encriptar dados, enquanto a segunda (a chave privada) permite desencriptar as informações embaralhadas pela primeira.

Quando você se conecta a um servidor SSH, seu micro e o servidor trocam suas chaves públicas, permitindo que um envie informações para o outro de forma segura. Através deste canal inicial é feita a autenticação, seja utilizando login e senha, seja utilizando chave e passphrase (como veremos a seguir).

Até aqui, tudo é feito utilizando chaves de 512 bits ou mais (de acordo com a configuração). O problema é que, embora impossível de quebrar, este nível de encriptação demanda uma quantidade muito grande de processamento. Se todas as informações fossem transmitidas desta forma, o SSH seria muito lento.

Para solucionar este problema, depois de fazer a autenticação, o SSH passa a utilizar um algoritmo mais simples, que demanda muito menos processamento, para transmitir os dados. Por padrão é utilizado o 3DES (triple-DES), que utiliza uma combinação de três chaves DES, de 64 bits cada. As chaves são trocadas periodicamente durante a conexão, o que torna o sistema quase impossível de quebrar. Na configuração do servidor e/ou cliente, é possível especificar outro algoritmo, como o Blowfish. Isso garante uma boa relação entre segurança e desempenho.

O SSH é dividido em dois módulos. O **sshd** é o módulo servidor, um serviço que fica residente na máquina que será acessada, enquanto o **ssh** é o módulo cliente, um utilitário que você utiliza para acessá-lo.

Nas distribuições derivadas do Red Hat, o servidor SSH é instalado através do pacote "**openssh-server**" e o cliente, através do "**openssh-clients**". No Debian, ambos são instalados através do pacote "**ssh**". Com o pacote instalado, você inicia o servidor usando o

comando "**service sshd start**" (nos derivados do Red Hat), ou "**/etc/init.d/ssh start**", no caso do Debian. Para que ele seja inicializado durante o boot, use respectivamente o "**chkconfig sshd on**" ou "**update-rc.d -f ssh defaults**".

A partir daí as coisas se unificam. A configuração do servidor, independentemente da distribuição usada, vai no arquivo "**/etc/ssh/sshd_config**", enquanto a configuração do cliente vai no "**/etc/ssh/ssh_config**". Note que muda apenas um "**d**" entre os dois, cuidado para não confundir cará com inhame ;).

Note que além do OpenSSH, que abordo aqui, existem outras versões do SSH, como o Tectia (uma versão comercial, disponível no <http://ssh.com>) e o SunSSH que, embora conservem diferenças no funcionamento e na configuração, são compatíveis entre si. O SSH é, na verdade, um protocolo aberto e não o nome de uma solução específica.

» Próximo: [Configuração do cliente](#)

Ao ser habilitado, o padrão do servidor SSH é permitir acesso usando qualquer uma das contas de usuário cadastradas no sistema, pedindo apenas a senha de acesso. Para acessar o servidor "192.168.0.2", usando o login "morimoto", por exemplo, o comando seria:

```
$ ssh morimoto@192.168.0.2
```

Ao invés de usar a arroba, você pode também especificar o login usando o parâmetro "-l" (de login), como em:

```
$ ssh -l morimoto 192.168.0.2
```

Você pode também acessar o servidor usando o nome ou domínio, como em:

```
$ ssh morimoto@web.kurumin.com.br
```

Caso você omita o nome do usuário, o SSH presume que você quer acessar usando o mesmo nome de usuário que está usando na máquina local. Se você está logado como "tux", ele tentará fazer login usando uma conta "tux" no servidor remoto. Naturalmente, só funciona caso você use o mesmo login em ambas as máquinas.

Ao acessar micros dentro da rede local, você pode também chamá-los pelo nome, como em "ssh morimoto@servidor". Neste caso, você precisará primeiro editar o arquivo **/etc/hosts** (no cliente), incluindo os números de IP das máquinas e os nomes correspondentes. O formato deste arquivo é bem simples, basta fornecer o IP e o nome da máquina correspondente, um por linha, como em:

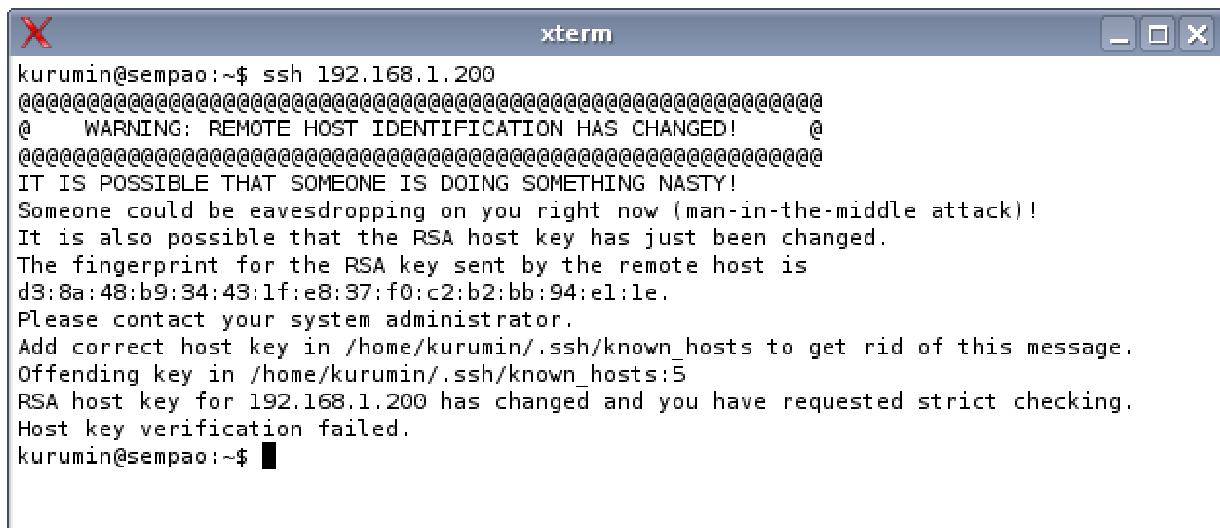
```
127.0.0.1  
192.168.0.2  
192.168.0.6 athenas
```

localhost
servidor

- **Verificação do servidor:** Como parte das verificações de segurança, o SSH utiliza também um sistema baseado em chaves assimétricas para verificar a identidade do servidor. O servidor tem uma chave pública, que envia ao cliente na primeira conexão. As identificações de todos os servidores conhecidos ficam armazenadas no arquivo ".ssh/known_hosts" dentro do seu diretório home. Sempre que você se conecta daí em diante, o cliente SSH envia um "desafio" ao servidor, uma frase encriptada usando a chave pública, que só pode ser descoberta usando a chave privada.

Isso previne um tipo de ataque muito comum chamado "man in the middle" (que poderia ser traduzido para "intermediário", ou "impostor"), em que alguém simplesmente substitui o servidor por outra máquina, usando o mesmo IP, ou sabota o servidor DNS da rede (ou do provedor) de forma que ele entregue um IP forjado quando você tenta acessar seu servidor baseado no domínio.

O servidor falso pode ser configurado para gravar sua senha e responder com uma mensagem do tipo "O servidor está em manutenção, tente novamente daqui a algumas horas". Dessa forma, ele vai ter não apenas acesso à sua senha, mas tempo para usá-la para acessar o servidor verdadeiro sem que você desconfie. Por sorte, o SSH percebe que a identificação do servidor mudou e lhe avisa do problema:



```
xterm
kurumin@sempao:~$ ssh 192.168.1.200
@@@@@@@@@@@WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack) !
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is
d3:8a:48:b9:34:43:1f:e8:37:f0:c2:b2:bb:94:e1:1e.
Please contact your system administrator.
Add correct host key in /home/kurumin/.ssh/known_hosts to get rid of this message.
Offending key in /home/kurumin/.ssh/known_hosts:5
RSA host key for 192.168.1.200 has changed and you have requested strict checking.
Host key verification failed.
kurumin@sempao:~$
```

Para continuar é preciso que você edite manualmente o arquivo ".ssh/known_hosts", dentro do home e remova a linha com a antiga identificação do servidor, deixando as demais. Da próxima vez que tentar se conectar, o SSH exibe uma mensagem mais simpática, perguntando se você quer adicionar a nova chave:

```
The authenticity of host '192.168.1.200' (192.168.1.200) can't be established.
RSA key fingerprint is f1:0f:ae:c6:01:d3:23:37:34:e9:29:20:f2:74:a4:2a.
Are you sure you want to continue connecting (yes/no)?
```

Não existe forma de fazer com que o cliente SSH adicione as novas chaves automaticamente, isso seria uma brecha de segurança. É sempre preciso primeiro remover a chave antiga no arquivo "~/.known_hosts" manualmente.

As chaves são geradas durante a instalação do SSH e salvas nos arquivos "/etc/ssh/ssh_host_rsa_key" e "/etc/ssh/ssh_host_dsa_key" (no servidor). Para não disparar o alarme nos clientes quando precisar reinstalar o servidor, salve os dois arquivos em um pendrive e restaure-os depois da instalação. Você pode fazer isso mesmo ao migrar para outra distribuição, pois as localizações dos dois arquivos não mudam.

Uma opção, seria desabilitar a checagem das chaves, adicionando a linha "StrictHostKeyChecking no" na configuração dos clientes. Contudo, isso não é recomendável, pois desabilita completamente a checagem, abrindo brechas para ataques.

- **Compressão:** No caso de servidores acessíveis via internet, você pode reduzir um pouco o consumo de banda ativando a compressão de dados via gzip, o que é feito adicionado a linha:

```
Compression = yes
```

Você pode também ativar a compressão adicionando a opção "-p" na hora de se conectar. Quase todas as opções do cliente SSH podem ser especificadas tanto no arquivo, quanto via linha de comando.

- **Aplicativos gráficos:** Além de oferecer acesso via linha de comando, o SSH permite rodar aplicativos gráficos remotamente (X11 forwarding). Algumas distribuições, como o Slackware, trazem o recurso desabilitado por padrão. Nestes casos, edite o arquivo "/etc/ssh/ssh_config" (a configuração do cliente) e substitua a linha "ForwardX11 no" por:

```
ForwardX11 yes
```

Outra opção é adicionar o parâmetro "-X" ao se conectar, como em "ssh -X tux@192.168.0.1". A partir daí, você pode chamar os aplicativos gráficos normalmente, como se estivesse num terminal local.

O maior problema com o uso de aplicativos remotos via SSH é que ele só funciona satisfatoriamente via rede local. Via internet os aplicativos gráficos ficam realmente muito lentos (mesmo em uma conexão de 1 ou 2 megabits), pois o protocolo do X é otimizado para uso local, com uso intensivo de pacotes de retorno e sem nenhum tipo de cache. Isso faz com que muitos administradores desabilitem o X11 forwarding no próprio servidor.

Para rodar aplicativos gráficos de forma segura via internet, a melhor solução é usar o FreeNX (que veremos em detalhes mais adiante). Ele é um sistema de acesso remoto baseado no SSH, que utiliza um protocolo bastante otimizado. Nele você tem um desktop completo (similar ao VNC), mas com um desempenho muito superior, mesmo em conexões via modem.

- **Keep Alive:** Concluindo a configuração do cliente, outro problema comum é a conexão ser fechada pelo servidor depois de alguns minutos de inatividade. Em muitas situações você quer manter a conexão aberta por longos períodos, sem precisar ficar dando um "ls" a cada dois minutos para manter a conexão aberta. Você pode evitar o problema fazendo com

que o próprio cliente mande pacotes periodicamente a fim de manter a conexão aberta. Para ativar isso, adicione a linha abaixo no "/etc/ssh/ssh_config":

```
ServerAliveInterval 120
```

Este é um exemplo de arquivo "/etc/ssh/ssh_config" configurado com as opções que vimos até aqui (excluindo os comentários):

ForwardX11	=	yes
Compression	=	yes
Port	=	22
ServerAliveInterval 120		

» Próximo: [Configuração do servidor](#)

Você pode configurar várias opções relacionadas ao servidor SSH, incluindo a porta TCP a ser usada editando o arquivo **"/etc/ssh/sshd_config"**. A maior parte das opções dentro do arquivo podem ser omitidas, pois o servidor simplesmente utiliza valores defaults para as opções que não constarem no arquivo. Mas, de qualquer forma, é saudável especificar todas as opções que conhece: além de evitar enganos, é uma forma de praticar e memorizar as opções.

- **Porta:** Uma das primeiras linhas é a:

```
Port 22
```

Esta é a porta que será usada pelo servidor SSH. O padrão é usar a porta 22. Ao mudar a porta do servidor aqui, você deverá usar a opção "-p" ao conectar a partir dos clientes, para indicar a porta usada, como em:

```
# ssh -p 2222 morimoto@192.168.0.1
```

Outra opção é editar o arquivo "/etc/ssh/ssh_config" (nos clientes) e alterar a porta padrão usada também por eles. Mudar a porta padrão do SSH é uma boa idéia se você está preocupado com a segurança. Muitos dos ataques "casuais", quando não existe um alvo definido, começam com um portscan genérico, onde é feita uma varredura em faixas inteiras de endereços IP, porém apenas em algumas portas conhecidas, como a 21, 22 e 80 (a fim de tornar o teste mais rápido, embora menos preciso).

A partir daí, os ataques vão sendo refinados e direcionados apenas para os servidores vulneráveis encontrados na primeira varredura. Colocar seu servidor para escutar uma porta mais escondida, algo improvável como a porta 32456 ou 54232, já dificulta um pouco as coisas.

- Controle de acesso: Logo abaixo vem a opção "ListenAddress", que permite limitar o SSH a uma única placa de rede (mesmo sem usar firewall), útil em casos de micros com duas ou mais placas. O típico caso onde você quer que o SSH fique acessível apenas na rede local, mas não na internet, por exemplo. Digamos que o servidor use o endereço "192.168.0.1" na rede local e você queira que o servidor SSH não fique disponível na internet. Você adicionaria a linha:

```
ListenAddress 192.168.0.1
```

Note que especificamos nesta opção o próprio IP do servidor na interface escolhida, não a faixa de IP's da rede local ou os endereços que terão acesso a ele.

- Protocolo: Atualmente utilizamos o SSH 2, mas ainda existem alguns poucos clientes que utilizam a primeira versão do protocolo. Por padrão, o servidor SSH aceita conexões de clientes que utilizam qualquer um dos dois protocolos, o que é indicado na linha:

```
Protocol 2,1
```

O protocolo SSH 1 tem alguns problemas fundamentais de segurança, por isso alguns administradores preferem desabilitar a compatibilidade com ele, aceitando apenas clientes que usam o SSH 2. Neste caso, a linha fica apenas "Protocol 2"

- Usuários e senhas: Outra opção interessante, logo abaixo é a:

```
PermitRootLogin yes
```

Esta opção determina se o servidor aceitará que usuários se loguem como root. Do ponto de vista da segurança, é melhor deixar esta opção como "no", pois assim o usuário precisará primeiro se logar usando um login normal e depois virar root usando o "su" ou "su -". Desta forma, será preciso saber duas senhas, ao invés de saber apenas a senha do root.

Por padrão, o SSH permite que qualquer usuário cadastrado no sistema se logue remotamente, mas você pode refinar isso através da opção "AllowUsers", que especifica uma lista de usuários que podem usar o SSH. Quem não estiver na lista, continua usando o sistema localmente, mas não consegue se logar via SSH. Isso evita que contas com senhas fracas, usadas por usuários que não têm necessidade de acessar o servidor remotamente coloquem a segurança do sistema em risco. Para permitir que apenas os usuários joao e maria possam usar o SSH, adicione a linha:

```
AllowUsers joao maria
```

Você pode ainda inverter a lógica, usando a opção "DenyUsers". Nesse caso, todos os usuários cadastrados no sistema podem fazer login, com exceção dos especificados na linha, como em:

```
DenyUsers ricardo manoel
```

Outra opção relacionada à segurança é a:

```
PermitEmptyPasswords no
```

Esta opção faz com que qualquer conta sem senha fique automaticamente desativada no SSH, evitando que alguém consiga se conectar ao servidor "por acaso" ao descobrir a conta desprotegida. Lembre-se que a senha é justamente o ponto fraco do SSH. De nada adianta usar 2048 bits de encriptação se o usuário escreve a senha num post-it colado no monitor, ou deixa a senha em branco.

- **Banner:** Alguns servidores exibem mensagens de advertência antes do prompt de login, avisando que todas as tentativas de acesso estão sendo monitoradas ou coisas do gênero. A mensagem é especificada através da opção "Banner", onde você indica um arquivo de texto com o conteúdo a ser mostrado, como em:

```
Banner = /etc/ssh/banner.txt
```

- **X11 Forwarding:** Um pouco depois temos a opção:

```
X11Forwarding yes
```

Esta opção determina se o servidor permitirá que os clientes executem aplicativos gráficos remotamente. Se o servidor será acessado via internet ou se possui um link lento, você pode deixar esta opção como "no" para economizar banda. Desta forma, os clientes poderão executar apenas comandos e aplicativos de modo texto.

- **Módulos:** O SSH inclui um módulo de transferência de arquivos (o SFTP), que veremos em detalhes a seguir. Ele é ativado através da linha:

```
Subsystem sftp /usr/lib/sftp-server
```

É realmente necessário que esta linha esteja presente para que o SFTP funcione. Comente esta linha apenas se você realmente deseja desativá-lo.

» Próximo: [Usando chaves](#)

Ao invés de depender unicamente da senha como forma de autenticação, o SSH permite o uso de um par de chaves, onde a chave pública é instalada nos servidores que serão acessados e a chave privada (que nunca sai da sua máquina) é protegida por uma passphrase.

Nesse caso, temos uma segurança de dois níveis, em que é preciso saber a passphrase e, além dela, ter a chave privada, um arquivo salvo no HD ou em um pendrive, algo similar ao sistema bancário, onde você precisa ter o cartão e saber a senha. Para gerar o par de chaves, use o comando:

```
$ ssh-keygen -t rsa
```

Ele é sempre executado usando seu login de usuário, não como root:

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/morimoto/.ssh/id_rsa):
Created directory /home/morimoto/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/morimoto/.ssh/id_rsa.
Your public key has been saved in /home/morimoto/.ssh/id_rsa.pub.
The key fingerprint is:
ff:28:26:f6:87:67:9f:4c:9a:c8:0a:3b:21:81:b4 morimoto@athenas
```

A passphrase pode ser desde uma senha "normal", de 8 ou 12 caracteres, até uma frase complexa, sem limite de tamanho. O importante é que não seja algo fácil de adivinhar. A passphrase é, na verdade, um componente da chave de encriptação. Sem a passphrase é impossível usar a chave.

Isso vai gerar os arquivos ".ssh/id_rsa" e ".ssh/id_rsa.pub" dentro do seu diretório home, que são respectivamente sua chave privada e a chave pública. O ".ssh/id_rsa" é um arquivo secreto, que deve usar obrigatoriamente o modo de acesso "600" (que você define usando o chmod), para evitar que outros usuários da máquina possam lê-lo. Muito servidores recusam a conexão caso os arquivos estejam com as permissões abertas.

Agora vem o comando final, que grava a chave no servidor:

\$ ssh-copy-id login@servidor

Substitua o "login" pelo seu login de usuário, e o "servidor" pelo endereço IP ou domínio do servidor. Isso abre uma conexão via SFTP, ainda utilizando seu login e senha de acesso, que é usada pelo ssh-copy-id para instalar a chave pública (o arquivo .ssh/id_rsa.pub, dentro do seu home) no servidor. Caso você trabalhe com várias chaves diferentes, pode escolher qual instalar especificando o arquivo a ser instalado, como em:

\$ ssh-copy-id -i ~/.ssh/id_rsa-123432.pub login@servidor

A partir daí, ao invés de pedir sua senha, o servidor verifica a chave privada, instalada na sua máquina e em seguida pede a passphrase. Mesmo que alguém consiga roubar sua chave privada, não conseguirá conectar sem saber a passphrase e vice-versa.

É possível ainda deixar a passphrase em branco na hora de gerar as chaves, o que faz com que o login passe a ser automático. Isso torna as coisas muito práticas, pois você pode escrever até mesmo scripts para copiar arquivos via SFTP, sem precisar se preocupar com senhas, mas não é necessariamente uma boa idéia, pois alguém que consiga copiar sua chave privada poderia ganhar acesso irrestrito a seu servidor.

Não é algo tão corriqueiro quanto pode parecer, pois a chave privada nunca sai do seu micro. O servidor remoto envia um "desafio" para o cliente na sua máquina e a chave é apenas usada para processar a resposta. Para roubar sua chave privada, seria necessário que alguém efetivamente invadisse o sistema, ou tivesse acesso físico ao seu micro, para dar boot com o live-CD e copiar o arquivo para um pendrive. De qualquer forma, não é bom dar sopa para o azar.

A melhor forma de usar chaves sem precisar ficar digitando a passphrase toda hora é usar o "**ssh-agent**". Ele funciona como uma espécie de "cache", onde você digita a passphrase apenas uma vez, depois de inicializar o sistema, e ela fica gravada na memória até que a sessão seja encerrada.

A segurança não é prejudicada, pois a passphrase não é salva em lugar algum, fica apenas armazenada (de forma encriptada) em uma área protegida de memória, acessível apenas ao ssh-agent. Ao desligar o micro, tudo é perdido.

Para usar o ssh-agent, abra um terminal e use os comandos:

```
$ ssh-add ssh-agent
```

Ele vai solicitar sua passphrase, como neste exemplo:

```
Enter passphrase for /home/morimoto/.ssh/id_rsa:  
Identity added: /home/morimoto/.ssh/id_rsa (/home/morimoto/.ssh/id_rsa)
```

A partir daí ela fica carregada na memória e você não precisa mais se preocupar até o próximo reboot. Uma forma prática de fazer com que os dois comandos sejam executados automaticamente durante a abertura do sistema, é adicioná-los em um ícone dentro da pasta ".kde/Autostart", dentro do seu diretório home. Note que eles não devem ser adicionados no bootmisc.sh, rc.local ou outro arquivo de inicialização, pois precisamos que os comandos sejam executados dentro do seu login de usuário e não pelo root.

Até aqui, aprendemos como utilizar uma única chave. É comum que seja usada uma única chave para acessar vários micros. Isso não representa um risco de segurança, desde que você escolha uma boa passphrase.

Porém, muitos administradores preferem trabalhar com várias chaves distintas, uma para cada servidor que será acessado, por exemplo. Isso é perfeitamente possível, embora bem mais trabalhoso. Para gerar novas chaves, rode o comando "**ssh-keygen -t rsa**", prestando atenção para informar um nome de arquivo alternativo na hora que ele pergunta:

```
Enter file in which to save the key (/home/morimoto/.ssh/id_rsa):
```

Se você salvou a segunda chave como "id_rsa2", por exemplo, o comando para instalá-la no servidor seria "ssh-copy-id -i ~/.ssh/id_rsa2.pub seu_login@servidor". Na hora de adicionar a segunda chave no ssh-agent, você deve também especificar o nome do arquivo, como em: "ssh-add /root/.ssh/id_rsa2".

Este procedimento pode ser repetido para adicionar quantas chaves diferentes quiser, mas as coisas ficam mais trabalhosas a cada nova chave adicionada :).

Ao usar o ssh-agent para guardar suas passphrases, você pode ativar a opção **ForwardAgent** (no cliente) para permitir que o agente disponível na sua máquina possa ser usado para abrir novas sessões SSH quando estiver logado em outras máquinas.

Imagine que você administra dois servidores remotos: servidor A e servidor B. Você instalou a sua chave pública em ambos e armazenou sua passphrase no ssh-agent, de forma que você pode logar em ambos, a partir da sua máquina sem digitar senha. Porém, se você estiver logado no servidor A, e precisar copiar um arquivo via sftp para o servidor B, você precisaria fornecer a senha ou passphrase, pois o servidor A não tem acesso à sua chave privada, que está no seu micro.

O ForwardAgent resolve isso, permitindo que a partir da sessão aberta no servidor A, você possa se conectar no servidor B. Isso é feito de forma segura, criando um túnel temporário, diretamente entre a sua máquina e o servidor B e fazendo a verificação da chave através dele, sem passar pelo servidor A. Desta forma, não existe a possibilidade de um keytrap, ou qualquer armadilha instalada no servidor A, ser usado para capturar sua chave ou passphrase.

Para ativar este recurso, abra o arquivo "**/etc/ssh/ssh_config**" (na sua máquina) e adicione a opção:

```
ForwardAgent yes
```

Depois de gerar a chave e conseguir se conectar através dela, você pode desativar a possibilidade de fazer logins normais, usando senha. Nesse caso, apenas você, que possui a chave gerada, conseguirá se conectar ao servidor.

Outras pessoas, mesmo que descubram a senha de alguma das contas, não terão como se conectar e nem como instalar uma nova chave para fazê-lo, a menos que tenham acesso físico ao servidor, a fim de copiar a chave manualmente.

Isso significa que, mesmo alguém com a senha de root do seu servidor em mãos não conseguirá fazer nada remotamente (o sonho de todo administrador ;). Isso pode ser usado para incrementar a segurança.

Para isso, mude as opções "ChallengeResponseAuthentication", "PasswordAuthentication" e "UsePAM" para "no" no arquivo "**/etc/ssh/sshd_config**" do servidor:

ChallengeResponseAuthentication	no
PasswordAuthentication	no
UsePAM	no

Para que as alterações entrem em vigor, reinicie o servidor SSH:

#	/etc/init.d/ssh	restart
ou:		
# service sshd restart		

» Próximo: [Transferindo arquivos](#)

O SSH é um verdadeiro canivete suíço. Além de permitir rodar aplicativos e fazer toda a administração de um servidor remotamente, ele também pode ser usado para transferir arquivos. A forma mais básica de fazer isso é usar o **sftp**, um comando que faz parte do pacote padrão.

Ele oferece uma interface similar à dos antigos programas de FTP de modo texto, mas todos os arquivos transferidos através dele trafegam através de um túnel encriptado, criado através do SSH. Na prática, temos uma espécie de VPN temporária, criada no momento em que é efetuada a conexão. A melhor parte é que o próprio SSH cuida de tudo, não é necessário instalar nenhum programa adicional.

Para se conectar a um servidor usando o sftp, o comando é:

```
$ sftp usuario@192.168.0.1
```

Se o servidor ssh na outra ponta estiver configurado para escutar em uma porta diferente da 22, é preciso indicar a porta no comando, incluindo o parâmetro **-o port=**, como em:

```
$ sftp -o port=22 morimoto@10.0.0.1
```

A partir daí você tem um prompt do sftp. Use o comando "**put**" para dar upload de um arquivo e "**get**" para baixar um arquivo do servidor para a pasta local. Para navegar entre as pastas do servidor, use os comandos "**cd pasta/**" (para acessar a pasta), "**cd ..**" (para subir um diretório), "**ls**" (para listar os arquivos) e "**pwd**" (para ver em qual diretório está). Veja um exemplo:

```
morimoto@athenas:~$ sftp -o port=2222 morimoto@10.0.0.1
Connecting to 10.0.0.1...
Password:
sftp> ls
Desktop Meu Computador OpenOffice.org1.1.1a icones-magicos.deb

sftp> get icones-magicos.deb
Fetching /home/kurumin/icones-magicos.deb to icones-magicos.deb
/home/kurumin/icones-magicos.deb 100% 825KB 825.1KB/s 00:01

sftp> put RealPlayer10GOLD.bin
Uploading RealPlayer10GOLD.bin to /home/kurumin/RealPlayer10GOLD.bin
RealPlayer10GOLD.bin 100% 6726KB 3.3MB/s 00:02

sftp> pwd
Remote working directory: /home/morimoto
```

Existem ainda os comandos "**lcd**" (local cd), "**lls**" (local ls), "**lmdir**" (local mkdir) e "**lpwd**" (local pwd), que permitem mudar o diretório local.

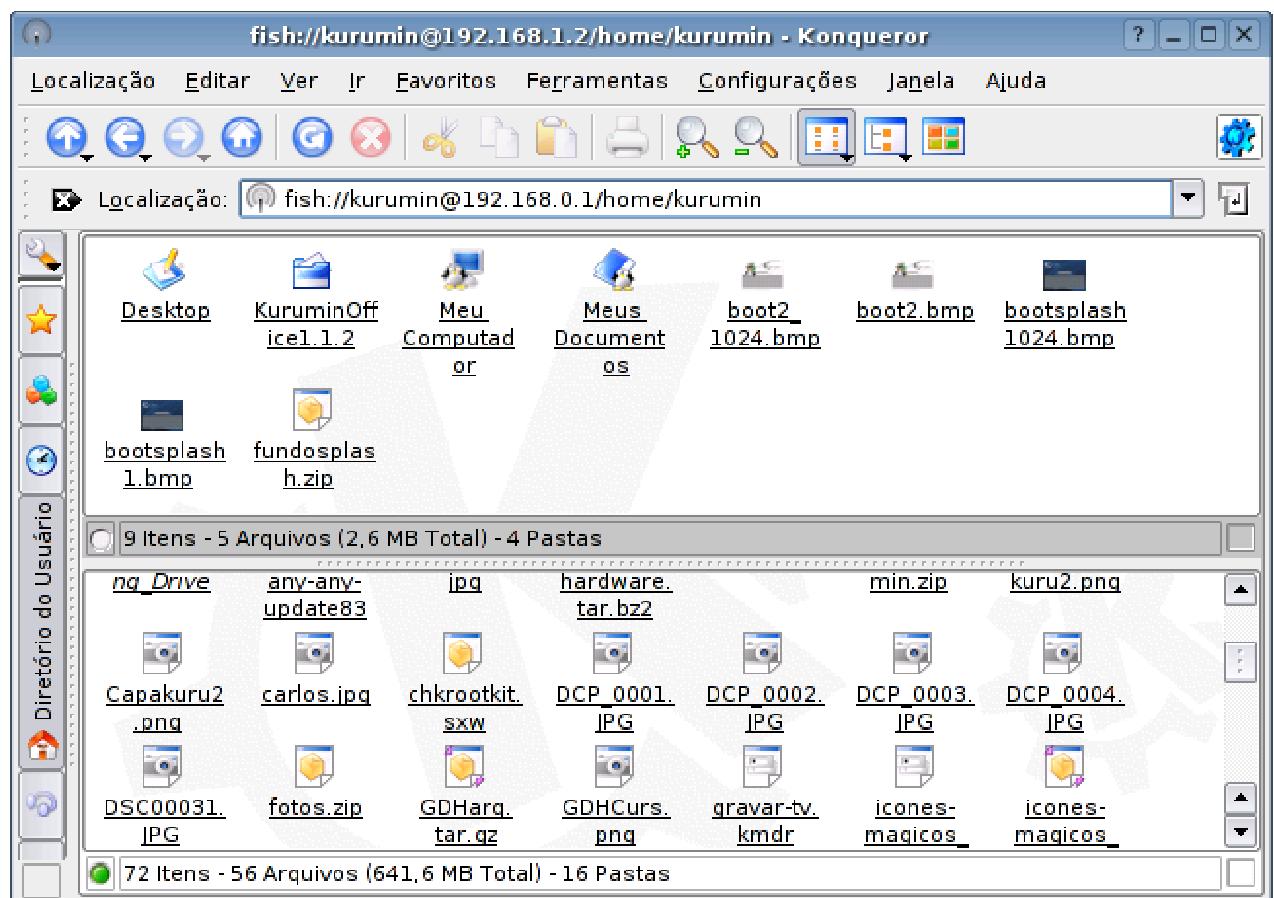
Por exemplo, digamos que você está atualmente no diretório "/mnt/arquivos". Ao abrir a conexão via sftp, tudo que você baixar será colocado automaticamente neste diretório. Mas, digamos que você queira baixar um determinado arquivo para o diretório "/home/joao". Você usaria, então, o comando "lcd /home/joao" para mudar o diretório local e depois o

"get arquivo" para baixá-lo já na pasta correta. Na hora de dar upload de um arquivo é a mesma coisa. Você pode usar o "lpwd" para listar os arquivos no diretório local e depois o "put arquivo" para dar upload.

Naturalmente, existem meios mais práticos de fazer isso, usando programas gráficos que suportam o sftp. O mais usado, neste caso, é o **konqueror**. Ele possui um módulo chamado "fish://", que permite acessar servidores remotos e transferir arquivos simplesmente arrastando-os para outra janela.

Acesse o endereço desejado através da própria barra de endereços, incluindo o login de acesso, como em "**fish://kurumin@192.168.0.1**". Você pode também especificar diretamente uma pasta no servidor remoto que quer acessar (por padrão você cai na pasta home), como em: **fish://kurumin@192.168.0.1/mnt/arquivos/**.

Para tornar as coisas mais práticas, eu uso o recurso de dividir a janela em duas, que você encontra no Janela > Separar visão em topo/base. Assim, fico com uma janela mostrando os arquivos locais e outra mostrando os arquivos do servidor, e posso simplesmente arrastar os arquivos que devem ser transferidos.



Uma forma mais primitiva de transferir arquivos via SSH é usar o "scp", que permite especificar em uma única linha o login e endereço do servidor, junto com o arquivo que

será transferido. Graças a isso, ele é muito usado em scripts. A sintaxe do scp é: "scp arquivo_local login@servidor:pasta_remota", como em:

```
$ scp /home/arquivo.tar usuario@empresa.com.br:/var/www/download
```

Você pode adicionar também as opções "-p" (que preserva as permissões de acesso além das datas de criação e modificação do arquivo original), "-r" (que permite copiar pastas, recursivamente), "-v" (verbose, onde são mostradas todas as mensagens) e "-C" (que ativa a compressão dos dados, ajuda muito na hora de transferir grandes arquivos via internet). Nesse caso, o comando ficaria:

```
$ scp -prvC /home/arquivo.tar usuario@empresa.com.br:/var/www/download
```

Ao incluir a opção "-r", você pode especificar diretamente uma pasta no primeiro parâmetro. Esta opção é interessante para backups.

O SSH pode ser ainda usado como "meio de transporte" por outros programas. Por exemplo, o **rsync** é um comando que permite sincronizar uma pasta local com uma pasta do servidor (para fazer backup, por exemplo). Ele é capaz inclusive de consertar arquivos danificados e dar upload de atualizações, enviando apenas as partes dos arquivos que forem diferentes, o que torna a transferência muito mais rápida.

Para instalar o rsync no Debian, use o comando "apt-get install rsync". Não vou falar muito sobre o rsync em si, pois a idéia é só dar mais um exemplo de como ele poderia ser usado em conjunto com o SSH.

O uso básico do rsync, para sincronizar duas pastas locais seria "rsync -a origem/ destino/". A pasta destino poderia ser um segundo HD, um cartão de memória ou um compartilhamento de rede, por exemplo.

Para usar o rsync via SSH, o comando acaba sendo bem mais complexo, mas o resultado é bem interessante. Ele vai apenas atualizar as partes dos arquivos que forem modificadas, sem dar upload dos arquivos inteiros novamente, como muitos programas de backup fariam.

Para sincronizar a pasta local "**/home/joao**" com a pasta remota "**/backup**", no servidor 64.246.47.76 (onde seria feito um backup dos arquivos locais), usando o login "joao", por exemplo, tudo via SSH, o comando seria:

```
$ rsync -av --rsh="ssh -l joao" /home/joao/ joao@64.246.47.76:/backup
```

Para recuperar posteriormente o backup no caso de um desastre, baixando os arquivos salvos no servidor bastaria inverter a ordem dos diretórios no comando:

```
$ rsync -av --rsh="ssh -l joao" joao@64.246.47.76:/backup /home/joao/
```

No primeiro comando os arquivos da pasta "/home/joao" vão para a pasta /backup do servidor e no segundo eles são recuperados, subscrevendo os arquivos locais. A parte mais

significativa deste comando é o parâmetro "--rsh="ssh -l joao", que diz para o rsync usar um programa externo (o SSH) para fazer o trabalho.

Uma observação é que usando apenas os parâmetros "-av", o rsync apenas atualiza e grava novos arquivos na pasta do servidor, sem remover arquivos que tenham sido deletados na pasta local. Por um lado isto é bom, pois permite recuperar arquivos deletados acidentalmente, mas por outro pode causar confusão. Se você preferir que os arquivos que não existem mais sejam deletados ao atualizar o backup, adicione o parâmetro "--delete", como em:

```
$ rsync -av --delete --rsh="ssh -l joao" /home/joao/ joao@64.246.47.76:/backup
```

» Próximo: [Usando o shfs](#)

Mesmo usando o "fish://" do Konqueror, o acesso aos arquivos do servidor remoto não é tão transparente quanto ao montar um compartilhamento NFS ou Samba, pois, por baixo dos panos, ele ainda precisa transferir o arquivo inteiro antes de abri-los ou salvar. Se você tentar abrir um vídeo, por exemplo, ele vai primeiro transferir todo o arquivo para um diretório temporário e só então abri-lo.

O shfs derruba esta limitação, permitindo montar diretórios do servidor remoto, como se fossem compartilhamentos de rede, permitindo que você acesse os arquivos de forma transparente, como se fossem arquivos locais. Tudo é feito via ssh, de forma que você não precisa manter nenhum serviço adicional ativado no servidor. Toda a configuração abaixo é feita no cliente.

Para usar o shfs, é necessário ter instalado o pacote "**shfs-utils**", junto com o módulo de Kernel "**shfs**". Para usar algumas das opções que veremos a seguir, você vai precisar também do pacote "**ssh-askpass**", por isso é importante instalá-lo também.

Vamos por partes. A página do projeto é a <http://shfs.sourceforge.net/>, onde você pode baixar um pacote contendo o código fonte tanto do módulo, quanto dos executáveis shfsmount e shfsumount. Comece descompactando o arquivo baixado, como em:

```
$ tar -zvxf shfs-0.35.tar.gz
```

Acesse a pasta que será criada. Para compilar o módulo "shfs", acesse a pasta "**shfs/Linux-2.6/**" e rode o comando "**make**". Note que para compilar o módulo, você deve ter instalados os pacotes kernel-headers e (em algumas distribuições) também o pacote "kernel-source", além dos compiladores básicos. Carregue o módulo gerado usando o comando "**insmod shfs.ko**".

Para compilar e instalar os dois executáveis, concluindo a instalação, acesse a pasta "**shfsmount/**" e rode os comandos "**make**" e "**make install**".

Nas distribuições derivadas do Debian, a instalação é mais simples, pois você pode instalar tudo via apt-get. Comece instalando os pacotes "shfs-utils" e "ssh-askpass":

```
# apt-get install shfs-utils ssh-askpass
```

Para instalar o módulo, instale o pacote "module-assistant" e o "shfs-source" e compile/install o módulo, usando os comandos:

```
# apt-get install module-assistant shfs-source  
# module-assistant install shfs
```

Algumas distribuições, como o Kanotix e versões recentes do Kurumin (a partir do 6.0), já trazem todos estes componentes pré-instalados, dispensando todos estes passos manuais.

Quando corretamente instalado, o shfs é bastante estável. Se você estiver tendo problemas de instabilidade, conexões interrompidas com erros estranhos, etc., atualize para a última versão. Pode ser que a distribuição em uso inclua uma versão antiga ou de desenvolvimento.

Com tudo nos devidos lugares, comece carregando o módulo "shfs":

```
# modprobe shfs
```

Se preferir que ele seja carregado automaticamente durante o boot, adicione a linha "shfs" no final do arquivo "/etc/modules". A partir daí, você pode usar o "shfsmount" para montar pastas do servidor remoto, como em:

```
# shfsmount morimoto@192.168.0.1:/mnt/arquivos /mnt/servidor
```

Veja que você precisa indicar o login e o endereço IP do servidor remoto, seguido da pasta que será acessada e do diretório local onde os arquivos ficarão disponíveis. No exemplo estou montando a pasta "/mnt/arquivos" do servidor remoto, mas você pode montar qualquer pasta (que o usuário usado tenha acesso), inclusive o diretório raiz. Para desmontar a pasta, use o comando "shfsumount", ao invés do "umount", como em:

```
# shfsumount /mnt/servidor
```

Se o sshd no servidor estiver configurado para usar uma porta diferente da padrão, indique a porta usada com o parâmetro "-P" (maiúsculo), como em:

```
# shfsmount -P 2222 morimoto@192.168.0.1:/mnt/arquivos /mnt/servidor
```

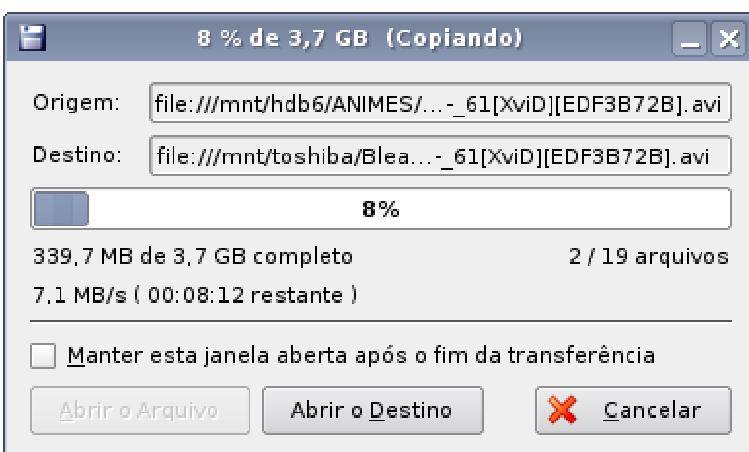
Originalmente, os arquivos ficam acessíveis apenas para o root. Se quiser acessar a pasta montada usando seu login de usuário, indique-o no comando, usando a opção "uid=", como em:

```
# shfsmount morimoto@192.168.0.1:/mnt/arquivos /mnt/servidor -o uid=tux
```

Se quiser abrir as permissões de acesso para todos os usuários (o que não é uma boa idéia do ponto de vista da segurança mas, enfim...), use a opção "-o rmode=777". Se quiser que os demais usuários tenham apenas permissão de leitura, use "-o rmode=755".

No caso de conexões instáveis ou ao acessar servidores remotos via internet, você pode adicionar a opção "**-p**" (minúsculo), que torna a conexão persistente, restabelecendo o acesso caso o servidor fique temporariamente indisponível. O "-p" torna o comportamento do shfsmount bastante interessante. Mesmo que o servidor remoto seja desligado, ele continua periodicamente tentando reabrir a conexão, durante dias, caso necessário. Quando o servidor fica novamente disponível ele abre um prompt gráfico pedindo a senha novamente e remonta a pasta.

Outra vantagem do shfs é o desempenho. Por implementar um sistema de cache, que reduz o número de requisições enviadas ao servidor e maximiza o throughput, ele acaba obtendo taxas de transferência muito mais altas, sobretudo via rede local. Apenas para efeito de comparação, tendo um Sempron 2800+ como cliente, um Pentium 4 3.06 como servidor e uma rede de 100 megabits, obtenho, em média, 4.8 MB/s em transferências de arquivos grandes usando o "fish://" do Konqueror e pouco mais de 7.0 MB/s usando o shfs. Apesar do algoritmo de encriptação continuar o mesmo, o shfs consegue ser quase 50% mais rápido.



É possível fazer com que determinados diretórios sejam montados automaticamente durante o boot, via shfs. Basta colocar o comando de montagem em algum arquivo de inicialização. O problema neste caso é que ele vai abrir a tela de autenticação, pedindo a senha a cada boot, o que pode ser irritante, sobretudo se você precisar montar diretórios de vários servidores diferentes.

Uma solução é usar a dica do SSH com login automático, usando um par de chaves sem passphrase. Neste caso, gere o par de chaves como root e adicione os comandos para montar os diretórios via shfs no arquivo "/etc/init.d/bootmisc.sh" ou "/etc/init.d/rc.local". Mesmo usando uma chave sem passphrase, a segurança ainda é bem melhor do que ao usar um protocolo sem encriptação, como o NFS ou SMB.

Se preferir fazer tudo usando seu login de usuário (o que é melhor do ponto de vista da segurança), coloque os comandos em um script dentro da pasta ".kde/Autostart".

» Próximo: [Criando túneis seguros](#)

Uma forma simples de encriptar protocolos que em condições normais não suportam encriptação é usar o SSH para criar túneis seguros, ligando uma das portas da sua máquina à porta do servidor onde o serviço em questão está ativo. Nesse caso, é criada uma espécie de VPN temporária, através da qual é possível acessar o serviço de forma segura. Todas as informações transmitidas são encriptadas pelo SSH, tornando seguros mesmo protocolos "escancarados", como o FTP.

Um dos usos mais comuns para este recurso é encriptar sessões do VNC, evitando que pessoas mal intencionadas tenham acesso ao que foi feito dentro da sessão, mesmo que ela seja interceptada.

O VNC utiliza uma chave de encriptação de mão única durante a autenticação, de forma que a senha não circula abertamente pela rede. Isso impede que alguém sniffando a rede consiga capturar sua senha do VNC, como acontece no caso do Telnet, por exemplo. Apesar disso, o algoritmo de encriptação de senha usada pelo VNC não é muito seguro e, depois que a conexão é iniciada, os dados são enviados de forma não-encriptada, abrindo a possibilidade de que alguém capaz de capturar os pacotes transmitidos possa ver o que você está fazendo e até mesmo capturar as teclas digitadas no teclado.

Se você utiliza o VNC para tarefas sensíveis, como administrar servidores, acessar sistemas bancários, etc., pode implantar uma camada extra de segurança, utilizando o VNC em conjunto com o SSH.

Neste caso, a segurança é quase total, pois além de ser necessária uma dupla autenticação, primeiro no SSH e depois no VNC, todos os dados são transmitidos através da rede de forma encriptada, utilizando um algoritmo reconhecidamente seguro.

Para utilizar o SSH em conjunto com o VNC, utilizamos a opção "-L", que permite redirecionar uma determinada porta local para uma porta no servidor. A sintaxe do SSH, neste caso, seria "ssh -L porta_local:servidor:porta_do_servidor servidor" (parece complicado, mas vai melhorar... :).

O servidor VNC escuta na porta 5900 + o número do display (5901, 5902, 5903, etc.). Note que a porta é diferente do servidor Java, acessível utilizando o browser, que utiliza as portas de 5800 em diante. Se você vai acessar o display 1 (porta 5901), na máquina 220.132.54.78, precisamos orientar o SSH a redirecionar esta porta para uma outra porta acessível pelo cliente VNC (a 5902, por exemplo) no PC local. Para isso, é necessário que o

servidor SSH esteja aberto no servidor remoto e que você tenha uma conta nele. O comando seria:

```
$ ssh -f -N -L5902:220.132.54.78:5901 -l login 220.132.54.78
```

Substitua o "login" pela sua conta de usuário na máquina remota. O SSH pedirá a senha e, pronto, você está conectado.

Tudo o que você precisa fazer agora é abrir o cliente VNC e acessar o endereço "localhost:2". Isso fará com que o cliente acesse a porta 5902 na máquina local, que por sua vez será redirecionada (através do túnel) para a porta 5901 do servidor remoto. Você usará o VNC da mesma forma, mas desta vez usando um túnel seguro.

Se você fosse acessar o display 4 (porta 5904) no servidor 192.168.0.4, redirecionando-o para a porta 5905 (display 5) da máquina local, logando-se usando o usuário "tux", o comando seria:

```
$ ssh -f -N -L5905:192.168.0.4:5904 -l tux 192.168.0.4
```

Neste caso, você acessaria o endereço "localhost:5" no cliente VNC.

A desvantagem de utilizar o SSH é que a atualização de tela ficará um pouco mais lenta, pois o servidor terá dois trabalhos, o de compactar os dados usando um dos algoritmos de VNC e, em seguida, encriptar os pacotes usando a chave do SSH, uma dupla jornada.

Uma observação é que este comando pode ser usado para criar túneis para outras portas, criando uma espécie de VPN entre os dois micros. Para redirecionar portas privilegiadas, da 1 a 1024, você precisa executar o comando como root. Para as portas altas (como as usadas pelo VNC), você pode usar um login normal de usuário.

O parâmetro "**-f**" dentro do comando faz com que o comando seja executado em background, liberando o terminal depois que a conexão é estabelecida. O "**-N**" faz com que o SSH apenas crie o redirecionamento da porta, sem abrir um terminal do servidor remoto. O "**-L**" é a opção principal, que especifica que queremos fazer um redirecionamento de portas. Ele é seguido (sem espaços) pela porta local que receberá a porta remota, o endereço do servidor e a porta do servidor que será redirecionada ("**-L2121:192.168.0.4:21**" redireciona a porta 21 do servidor remoto para a porta 2121 da máquina local). O "**-l**" em seguida especifica o login que será usado para estabelecer a conexão, seguido pelo endereço IP do servidor.

Em resumo, a sintaxe deste longo comando é: **ssh -f -N -Lporta-local:servidor:porta-do-servidor -l login servidor** (veja que é necessário especificar o endereço do servidor remoto duas vezes).

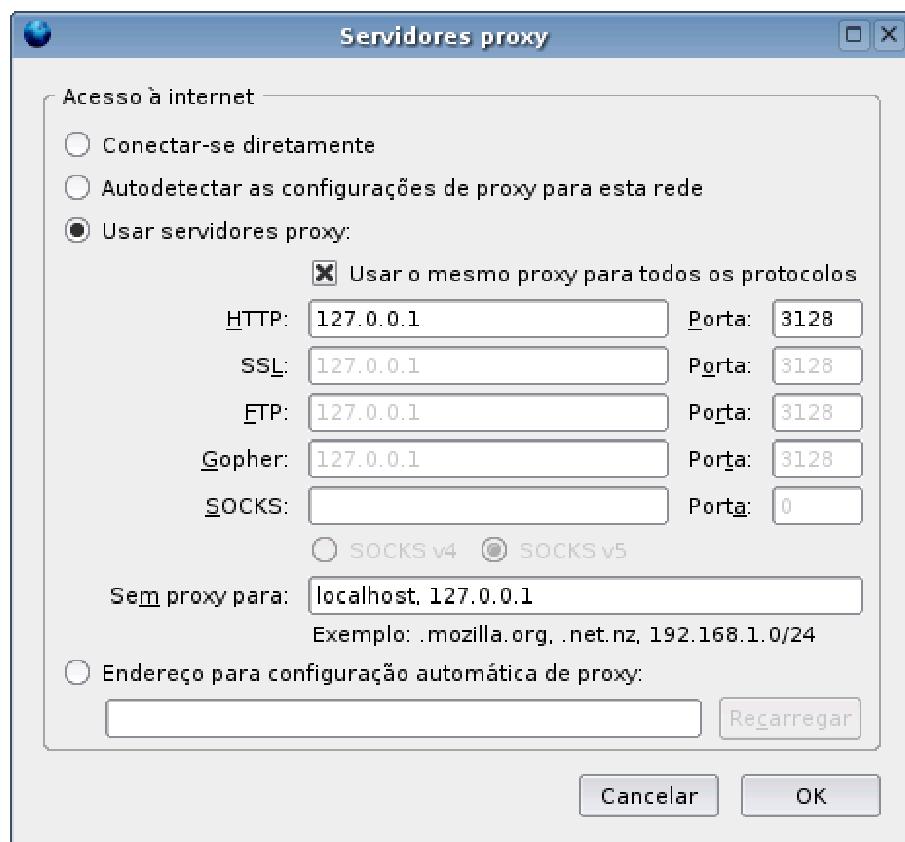
Além do VNC, podemos usar estes túneis seguros para encriptar praticamente qualquer outro protocolo. Um segundo exemplo interessante seria usar um túnel para encriptar todo o tráfego Web, de forma que você possa navegar de forma segura, ler seus e-mails, etc. mesmo ao acessar através de uma conexão wireless sem qualquer tipo de encriptação.

Para isso, é preciso que o gateway da rede (ou alguma máquina na Internet, que seja acessível por você) esteja com um servidor proxy aberto. Se você estiver usando o Squid, por exemplo, o proxy ficará aberto na porta 3128 do servidor.

Podemos usar o SSH para criar um túnel, ligando a porta 3128 do servidor à porta 3128 (ou qualquer outra) do seu micro. Para que isso funcione, é necessário que você tenha um login de acesso válido no servidor:

```
$ ssh -f -N -L3128:200.231.23.21:3128 -l tux 200.231.23.21
```

O próximo passo é configurar o navegador na sua máquina para acessar usando o proxy. Mas, ao invés de configurar o navegador para acessar o proxy diretamente, vamos configurá-lo para procurar o proxy na porta 3128 do localhost. Isso faz com que todos os acessos sejam direcionados ao túnel criado através do SSH e cheguem até o proxy de forma encriptada:



Embora inseguro, o FTP ainda é muito usado para tarefas sensíveis, como atualizar o conteúdo de websites. O perigo é óbvio: qualquer um em condições de sniffar o tráfego da rede pode capturar sua senha e usá-la para alterar o conteúdo do seu site, fazendo um acesso normal via FTP.

Para evitar isso, você pode novamente usar um túnel SSH. Se você tem acesso ao servidor via SSH, pode simplesmente criar o túnel diretamente, ligando a porta 21 do servidor a uma

porta da sua máquina e configurando o cliente FTP para acessar através dela. Mas, mesmo que isso não seja possível, ainda existe a possibilidade de usar qualquer outro servidor disponível na Internet, ao qual você tenha acesso via SSH para criar o túnel.

Se, por exemplo, você quer acessar o servidor FTP que está escutando a porta 21 do host "meu-site.com.br", criando um túnel através do host "meu-amigo.com.br" (ao qual você tem acesso via SSH), através da porta 2121 do localhost, o comando ficaria:

```
$ ssh -f -N -L2121:meu-site.com.br:21 -l login meu-amigo.com.br
```

Nesse caso, é criado um túnel entre a porta 2121 da sua máquina e o host "meu-amigo.com.br", que encaminha os pacotes para a porta 21 do host "meu-site.com.br". Essa configuração é menos segura que um túnel direto, pois o túnel encriptado existe apenas entre a sua máquina e o "meu-amigo.com.br". Dele até o servidor "meu-site.com.br" é feita uma conexão normal, sem encriptação.

Em teoria, os dados ainda poderiam ser capturados, mas pelo menos eles passam ilesos através da rede local, que normalmente é o ponto mais vulnerável a interceptações, sobretudo se você está acessando através de uma rede wireless sem encriptação.

Para usar os túneis SSH é necessário que o servidor esteja apenas com a porta do SSH aberta no firewall. Seja qual for a porta destino, todo o tráfego é transportado através da porta do SSH e encaminhada localmente para a porta final. Graças a essa peculiaridade, os túneis são uma forma muito usada para acessar ferramentas como o Webmin, PhpMyAdmin ou Swat em servidores remotos, sem precisar manter as respectivas portas abertas para toda a Internet.

Basta que a porta 22 (ou outra em que o servidor SSH esteja escutando) esteja aberta para que você consiga acessar qualquer outra usando túneis. Em casos em que o servidor remoto esteja configurado para usar uma porta diferente da padrão para o SSH, adicione a opção "-p porta" no início do comando, como em:

```
$ ssh -p 2222 -f -N -L2121:meu-site.com.br:21 -l login meu-amigo.com.br
```

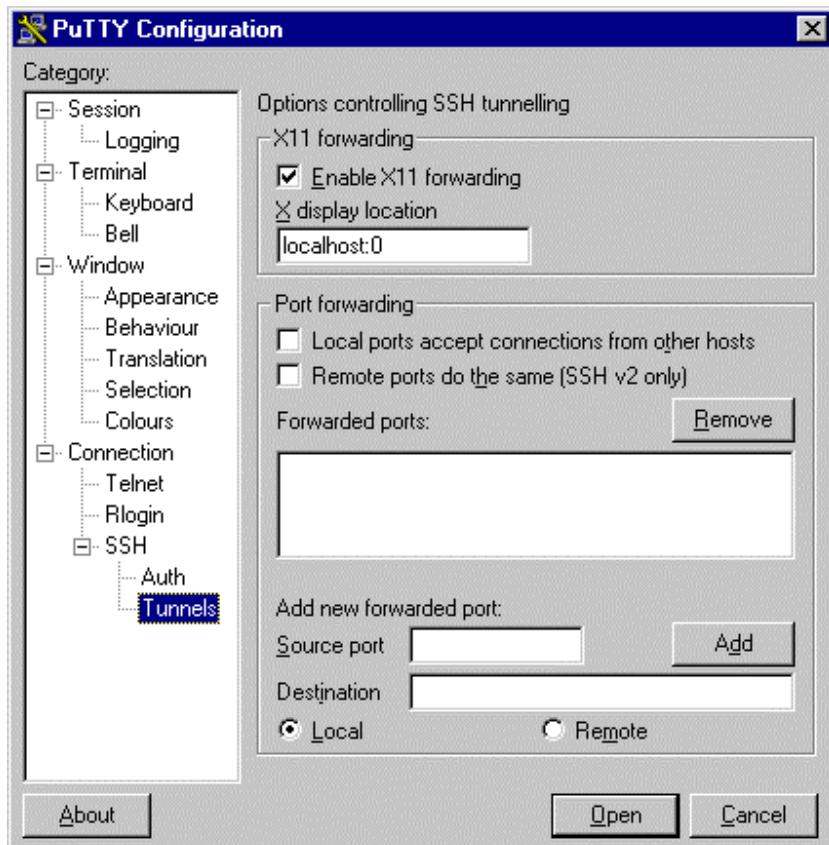
» Próximo: [SSH no Windows](#)

Existem diversas versões do SSH. A maioria das distribuições Linux inclui o OpenSSH, que não possui um cliente for Windows. No entanto, isso não chega a ser um problema, pois o SSH é um protocolo aberto, o que permite o desenvolvimento de clientes para várias plataformas, inclusive Windows. Eles são usados por muita gente para administrar servidores Linux remotamente.

Um exemplo de cliente simples e gratuito é o Putty, que inclui também o PSFTP, um cliente de SFTP, que permite também transferir arquivos. Ambos podem ser baixados no: <http://www.putty.nl/>. Outro exemplo é a versão da SSH Security, que tem vários recursos mas é gratuita apenas para universidades e usuários domésticos. O link é: <http://www.ssh.com>.

O Putty, o SSH da SSH Security e o OpenSSH são intercompatíveis. A grande limitação é que esses dois clientes são limitados ao modo texto, pois, para exibir aplicativos gráficos via SSH, é necessário que o sistema cliente possua um servidor X. O Windows, entretanto, não oferece nada parecido nativamente.

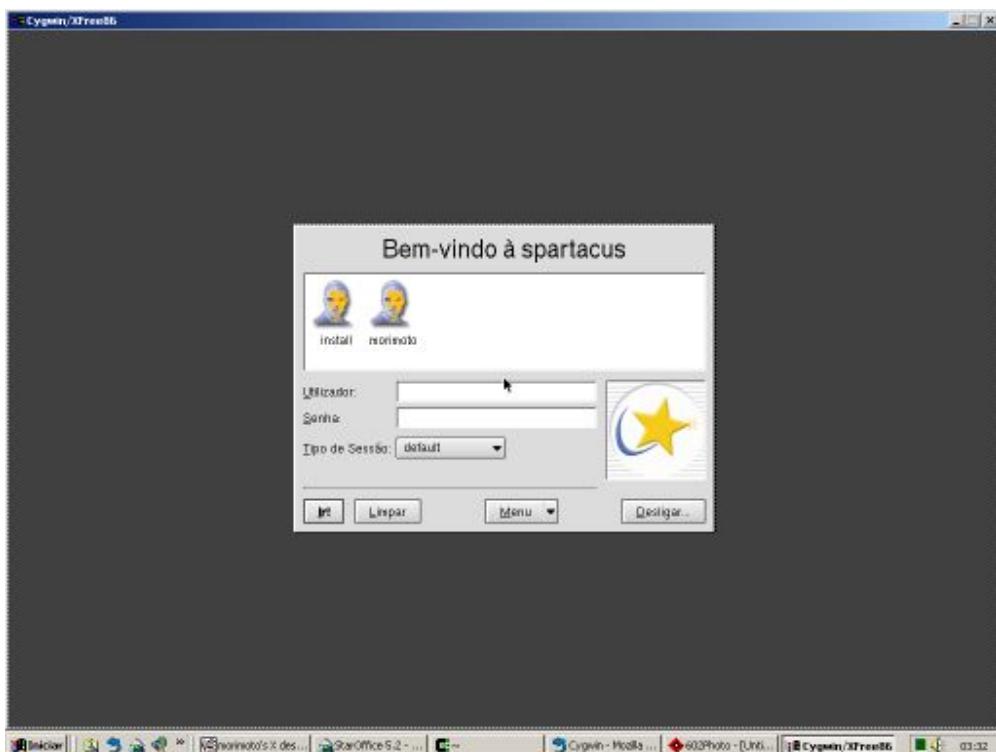
Existem alguns servidores X para Windows, que abrem uma sessão do X dentro de uma janela, como o X-Win32 (<http://xwin32.dk>) e o WinaXe (um Shareware de 90 dólares que pode ser baixado no <http://labf.com>). O Putty pode ser configurado para utilizar um desses servidores, marcando a opção "Enable X11 forwarding", em "Connection > SSH > Tunnels".



Contudo, a melhor opção para quem precisa rodar aplicativos gráficos é o **Cygwin**, uma implementação da API do Linux que roda sobre o Windows. Você pode compilar e rodar programas Linux no Windows através dele, incluindo gerenciadores de janelas. Até mesmo o KDE já foi portado :).

O grande apelo do Cygwin é a facilidade de uso. Tudo bem que recompilar programas Linux para rodar sobre ele pode não ser tão transparente assim, mas você não precisa se preocupar em configurar nada além dos programas, pois todo o acesso a hardware continua sendo feito pelo Windows.

O mais importante, no nosso caso, é que o Cygwin inclui um servidor X, que pode ser chamado pelo comando "**XWin**" (ou "xinit" nas versões antigas). Com o servidor X aberto, você pode usar o cliente SSH incluído para rodar aplicativos gráficos. Outra possibilidade é capturar a tela de login de uma máquina Linux com o XDMCP habilitado e rodar todos os programas a partir dela. Nesse caso, você chamaria o X com o comando "**XWin -query ip_do_servidor**", como em "**XWin -query 192.168.0.5**".



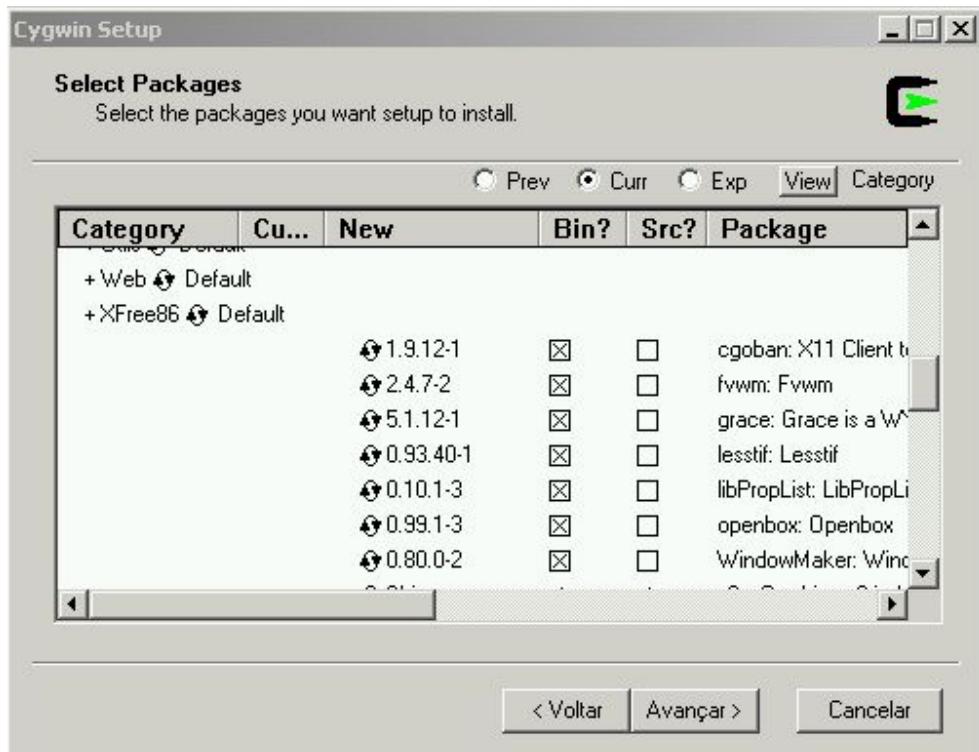
A partir daí você tem todos os aplicativos da máquina Linux em uma janela, semelhante ao VNC, só que mais rápido e com uma utilização muito mais baixa do processador. Ideal, por exemplo, para transformar máquinas com o Windows 95/98 em clientes de um servidor Linux, sem precisar reinstalar o sistema.

Você pode baixar o Cygwin no <http://cygwin.com>. O pacote de instalação e os aplicativos podem ser baixados gratuitamente. Para instalar, basta baixar o "**setup.exe**" e marcar os pacotes que deseja instalar. A instalação é feita via web, similar à do antigo Netscape. Tudo é bem simples. Depois de instalado, é criado um atalho para o terminal no desktop, basta abri-lo e chamar os programas desejados, como no Linux.

O método mais seguro de instalar é escolher primeiro a opção "Download from Internet", que baixa todos os pacotes em uma pasta sem instalar e depois rodar novamente o instalador, escolhendo a opção "Install from local directory" para instalar a partir dela.

Fazendo isso, você poderá instalar o Cygwin outras vezes ou em outras máquinas, sem precisar ficar baixando tudo de novo a cada instalação. Apesar de ser pequeno se comparado a uma distribuição Linux completa, uma instalação típica do Cygwin geralmente fica com 100 ou até 150 MB.

Para que tudo funcione, não se esqueça de marcar os pacotes do Xfree na tela de instalação do Cygwin, na seção "XFree86".



O Cygwin inclui, também, um cliente SSH, que pode ser marcado na seção "Network" durante a instalação. Você pode usar este cliente para se conectar em outras máquinas Linux e até mesmo rodar aplicativos gráficos. Para isso faça o seguinte:

- 1- Abra o Cygwin (ele vai abrir uma janela com um prompt de comando);
- 2- Abra o X usando o comando "**/usr/X11R6/bin/startxwin.bat**";
- 3- Abra o Windowmaker com o comando "**wmaker**". Se você não instalou o Windowmaker, pode usar o TWM que é instalado por default. Neste caso o comando é simplesmente "**twm**";
- 4- Abra um terminal e conecte-se ao servidor, usando o parâmetro -X, como em "**ssh -X joao@192.168.0.1**";
- 5- Pronto, agora você pode executar os aplicativos gráficos normalmente e eles serão abertos na janela com o Windowmaker, quase da mesma forma que no Linux.

Veja que utilizar o Windows para se conectar em outras máquinas Linux via SSH, NFS ou outros protocolos que ele não suporte nativamente é mais complicado do que utilizar uma máquina Linux para a mesma tarefa. Você pode experimentar utilizar um CD do Kurumin que contém estas ferramentas e voltar para o Windows depois que terminar. Outra opção é utilizar o VMware para instalar uma distribuição Linux "dentro" do Windows.

» Próximo: [Usando o rssh](#)

Uma das limitações do ssh, shfs e do sftp é que, ao criar uma conta de usuário, ele tem acesso não apenas aos arquivos que deve modificar, mas acesso via shell ao servidor, que pode ser usado para rodar comandos diversos e até mesmo explorar brechas de segurança locais (onde um usuário restrito do sistema pode obter privilégios adicionais).

Você pode dar um espaço de armazenamento para um amigo, onde espera que ele guarde apenas alguns backups e descobrir mais tarde que ele andou saturando a banda do servidor baixando filmes e músicas via bittorrent.

O rssh é uma resposta para esses casos. Ele permite que o usuário tenha acesso ao servidor apenas via sftp ou scp, sem ter como executar comandos adicionais. A página do projeto é <http://www.pizzashack.org/rssh/>.

Comece instalando o pacote "rssh", que é encontrado na maioria das distribuições. Você pode também instalar baixando o pacote .tar.gz com os fontes, disponível na página. No Debian ele está disponível via apt-get:

apt-get install rssh

Abra agora o arquivo "/etc/rssh.conf" (ou "/usr/local/etc/rssh.conf", ao instalar a partir dos fontes) e descomente as linhas:

```
allowscp  
allowsftp
```

Elas especificam que os usuários remotos poderão usar o scp e sftp para transferir arquivos, mas nenhum outro comando. Verifique também se o arquivo "**/etc/shells**" contém a linha "**/usr/bin/rssh**" e, caso necessário, adicione-a manualmente. Crie agora o usuário que terá acesso, usando os passos de sempre:

adduser manuel

Originalmente, o usuário criado teria acesso completo, via SSH e SFTP. Para limitá-lo ao SFTP, abra o arquivo "**/etc/passwd**", onde vai a configuração dos usuários do sistema, e procure a linha referente ao usuário criado (que normalmente será última). Originalmente você verá algo como:

```
manuel:x:1005:1005:,,,:/home/manuel:/bin/bash
```

O "/bin/bash" indica o shell ao qual o usuário terá acesso. O pulo do gato é substituir o "/bin/bash" pelo "**/usr/bin/rssh**", fazendo com que ele fique restrito aos comandos scp e sftp que indicamos no arquivo "/etc/rssh.conf". Depois da alteração, a linha ficará assim:

```
manuel:x:1005:1005:,:/home/manuel:/usr/bin/rssh
```

Em algumas distribuições (e ao instalar a partir dos fontes), o rssh será instalado dentro da pasta "/usr/local/bin" e não "/usr/bin". Preste atenção para sempre indicar a localização correta.

Você pode alterar também o "/home/manuel", colocando o diretório onde ficam os arquivos que o usuário pode alterar. Se ele vai apenas alterar os arquivos de um site colocado na pasta "/var/www/manuel", por exemplo, você poderia usar:

```
manuel:x:1005:1005:,:/var/www/manuel:/usr/bin/rssh
```

Desta forma, ao conectar ele cai automaticamente na pasta correta, o que facilita as coisas. Depois de verificar tudo, teste tentando acessar localmente, usando o usuário criado:

```
$ sftp manuel@127.0.0.1
```

Você notará que, via SFTP você conseguirá acessar os arquivos normalmente. Mas, ao tentar acessar via SSH, você recebe um erro, como:

```
This account is restricted by rssh.  
Allowed commands: scp sftp  
  
If you believe this is in error, please contact your system administrator.  
Connection to 127.0.0.1 closed.
```

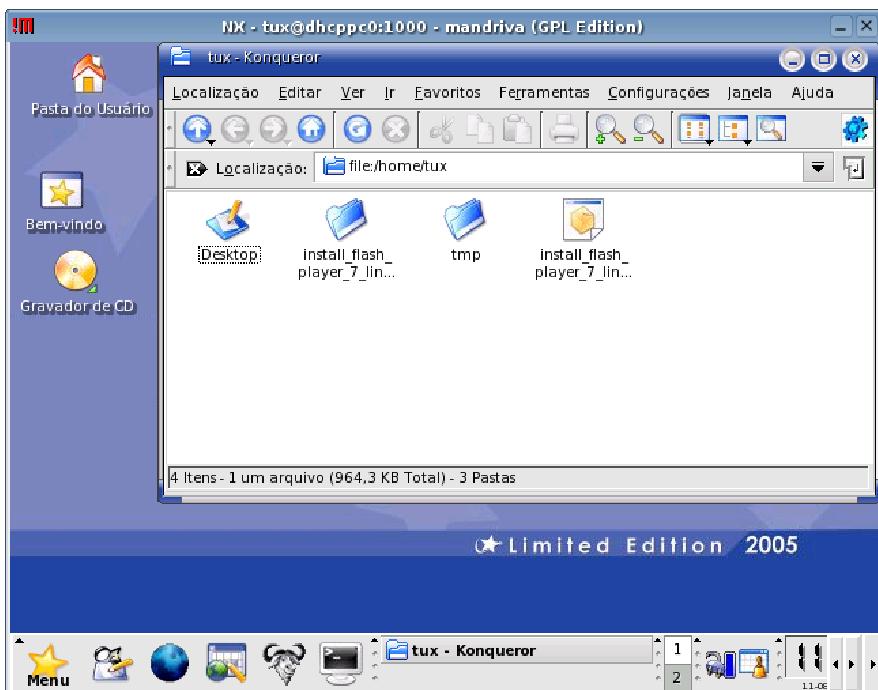
Uma observação é que usando o rssh, você não conseguirá conectar usando o "fish://" do Konqueror, precisará conectar através de algum programa que use o SFTP "puro". Dois exemplos são o GFTP (no Linux) e o Filezilla (no Windows). Em ambos, procure pela opção que indica o protocolo usado e troque de "FTP" para "SSH2". Indique também a porta usada pelo servidor, que no SFTP é 22 e não 21.



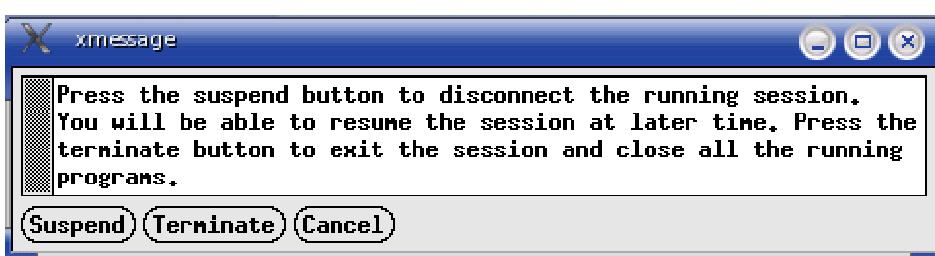
» Próximo: [Usando o FreeNX Server](#)

O FreeNX Server é uma espécie de sucessor do VNC. Ele é mais prático de usar e utiliza um sistema mais inteligente de compressão dos dados. Ao invés de simplesmente tirar screenshots da tela e comprimir as imagens, como faz o VNC, ele abre uma sessão remota do X (como ao usar o XDMCP, que veremos a seguir), onde são transmitidas as instruções e os pixmaps usados para montar a tela que será exibida no cliente. Esses dados são compactados usando um algoritmo próprio (mais eficiente que sistemas tradicionais de compressão de dados, como o Zlib) e encriptados usando o SSH, o que torna o FreeNX mais rápido e mais seguro que o VNC, tanto em links lentos (sobretudo conexões via ADSL ou modem) quanto em redes locais, onde banda não é problema.

Assim como no VNC, o FreeNX exibe uma janela contendo um desktop do servidor. O tamanho da janela é ajustável e cada sessão é independente, permitindo que dezenas de clientes (Linux ou Windows) se conectem ao mesmo servidor.



Ao encerrar a sessão, você tem a opção de suspender-la, o que permite reconectar mais tarde (a partir do mesmo cliente), sem perder as janelas e trabalhos abertos.



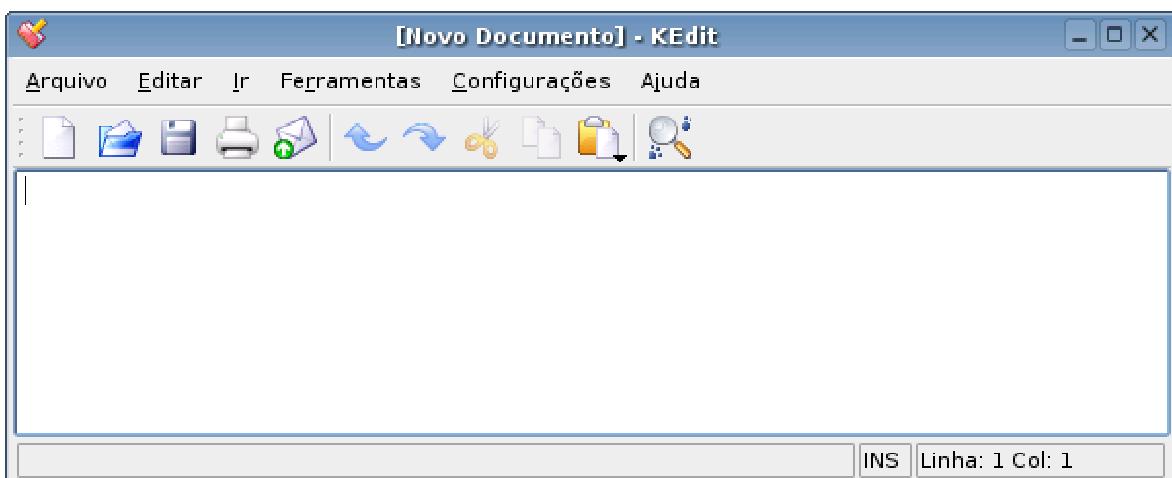
O FreeNX consome menos processamento e menos banda que o VNC. Em uma rede local isso permite abrir mais sessões simultâneas a partir do mesmo servidor, usar micros mais antigos como clientes e, ainda assim, executar os aplicativos de forma transparente, com tempos de resposta muito baixos. Os ganhos são ainda mais notáveis ao acessar máquinas remotamente através de conexões lentas. Um ADSL de 256k já é suficiente para trabalhar confortavelmente, acessando seu micro do trabalho, por exemplo. Surpreendentemente, mesmo uma conexão via modem se revela bastante utilizável.

No site da NoMachine (<http://www.nomachine.com/>) você pode fazer um testdrive, acessando um dos servidores NX da empresa. Os servidores estão localizados na Europa e ficam sobrecarregados em muitos horários, por isso são uma boa amostra do desempenho ao acessar servidores congestionados e distantes geograficamente.

» Próximo: [O sistema de compressão](#)

O próprio X permite executar aplicativos remotamente de forma bastante completa e transparente. Ao obter a tela de login de outra máquina da rede via XDMCP, ou usar um terminal LTSP, você está justamente utilizando este protocolo nativo do X.

Originalmente, os dados são transmitidos da forma mais simples possível, sem nenhum tipo de encriptação. O servidor X, que roda na máquina cliente, recebe as informações enviadas pelos programas executados no servidor e as usa para montar as imagens que serão mostradas na tela. Cada janela é formada por um conjunto de instruções e coordenadas, texto e pixmaps, que são as imagens e os componentes gráficos usadas nela. Todos os componentes são posicionados de forma a criar a tela do aplicativo, como esta janela do kedit que você vê abaixo:



O protocolo nativo do X é interessante apenas para uso em redes locais, onde existe muita banda disponível e a questão da segurança não é um fator crítico. Para conexões via internet, é possível encriptar as informações usando o SSH, basta que o arquivo "/etc/ssh/sshd_config" no servidor contenha a linha:

```
X11Forwarding yes
```

Com o X11Forwarding ativo (no servidor), você precisa apenas fazer uma conexão normal via SSH e chamar os aplicativos gráficos desejados através do terminal. Assim como ao usar o XDMCP, os aplicativos são executados no servidor e as instruções necessárias para exibir a imagem na tela são enviadas para o servidor X ativo no cliente. A diferença é que, neste caso, tudo é feito através do canal encriptado estabelecido pelo SSH.

Via rede local, este sistema funciona muito bem, os aplicativos rodam com um bom desempenho e de forma transparente. Mas, ao tentar usar o SSH para tentar rodar aplicativos gráficos via Internet (mesmo que através de uma conexão de banda larga) o desempenho se torna muito ruim, com os aplicativos demorando muito tempo só para abrir e continuando a apresentar um desempenho pífio depois disso. Via modem a situação é ainda pior.

O SSH permite encriptar a transmissão usando o zlib, o mesmo algoritmo que usamos para gerar arquivos .gz. Para ativar a compressão, você adiciona a opção "-C", como em:

```
$ ssh -C morimoto@kurumin.com.br
```

Usando um monitor de conexão, você pode notar que, ao ativar compressão, o SSH transmite uma quantidade de dados muito menor que o VNC (mesmo com a compressão via JPG ativa), mas, mesmo assim, o desempenho ao rodar aplicativos gráficos via Internet, usando o SSH, ainda é bem inferior. Os aplicativos continuam demorando pra abrir e demorando pra responder, muito diferente do que acontece ao usá-lo via rede local.

Se o X + SSH + Zlib é mais eficiente que o VNC na compressão dos dados, por que o desempenho via Internet é tão ruim?

Entra aí uma das principais desvantagens do protocolo X, os famosos "roundtrips", ou pacotes de resposta. O X foi desenvolvido para trabalhar localmente, ou via rede local, onde o tempo de latência do link é muito baixo. Em uma rede local, o ping é muito baixo, pois o impulso elétrico viaja quase à velocidade da luz e a distância entre um micro e outro é muito pequena.

Para assegurar a integridade da transmissão, para cada instrução recebida é enviado um pacote de resposta. O servidor transmite a instrução seguinte apenas depois de receber a resposta para a primeira. Como disse, via rede local o ping é muito baixo, por isso estes pacotes de resposta não fazem muita diferença. Via Internet a coisa muda de figura, pois, ao invés de poucos nanosegundos, passamos a trabalhar com pings de 30 milissegundos ou mais.

Um aplicativo como o Firefox pode exigir a transmissão de mais de 2.000 pacotes de resposta durante seu carregamento inicial. Alguns aplicativos específicos chegam a transmitir mais de 6.000, o que pode fazer com que demorem vários minutos só para abrir. Nesse caso, o limitante não é a velocidade do link, mas sim o tempo de resposta.

O FreeNX elimina este problema através de um duplo proxy. O primeiro proxy faz parte do próprio servidor NX, ele recebe os movimentos do mouse e texto digitado no teclado do cliente e os transmite para os aplicativos executados no servidor. O próprio servidor NX monta as telas e transmite todos os pacotes de resposta localmente (no próprio servidor) e depois transmite as atualizações de imagem "prontas" para o cliente, usando um protocolo próprio.

O proxy no servidor (chamado de "nxagent") elimina quase que inteiramente os pacotes de resposta, acabando com o principal gargalo. O proxy sozinho seria suficiente para trazer o desempenho do FreeNX a um nível similar ao obtido pelo VNC. Apesar disso, o FreeNX vai além, implementando o sistema de compressão, um segundo proxy (desta vez no cliente) e um sistema de cache.

Ao contrário do VNC, o servidor NX não envia screenshots da tela, mas sim instruções, texto e componentes gráficos (ícones, imagens, etc.) que são usados pelo cliente para montar a tela. O algoritmo de compressão usado pelo NX é otimizado para esta situação

específica, tratando as imagens, texto e instruções de forma diferente e aplicando seletivamente o tipo de compressão mais adequado para cada caso. As imagens, os ícones e outros elementos gráficos são compactados em JPG ou PNG, o texto é comprimido usando o Zlib e assim por diante.

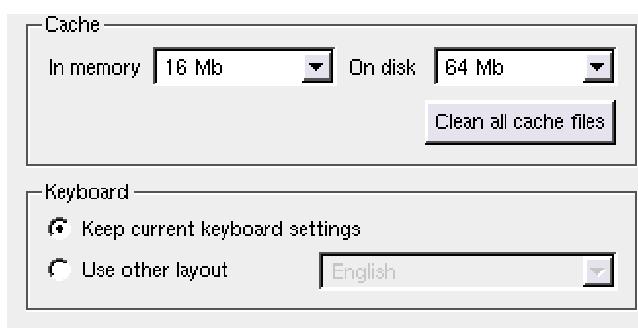
No caso de instruções, imagens ou blocos de texto com poucas modificações em relação a outros já recebidos, é utilizado um sistema de transferência diferencial, similar ao rsync, que transmite apenas as partes que foram modificadas, reduzindo brutalmente o número de bits transmitidos.

As instruções usadas para montar a tela são transmitidas com uma prioridade mais alta do que imagens, o que faz com que a interface continue respondendo enquanto uma grande imagem (um papel de parede, por exemplo) é transmitida. A própria imagem do papel de parede é quebrada em vários pequenos pacotes, o que permite que as partes já transmitidas sejam mostradas na tela assim que recebidas.

O segundo proxy, que fica ativo no cliente (chamado de "nxproxy"), cria um cache que armazena as informações já recebidas e as reutiliza sempre que possível. O papel de parede pode demorar um pouco para ser transmitido se você está conectado via modem, mas a imagem é transmitida uma única vez. Depois de carregada, ela é utilizada indefinidamente, sem degradar a performance da conexão. O mesmo se aplica a ícones e outros elementos gráficos, instruções e até mesmo texto exibido dentro das janelas.

Este cache pode ser reutilizado em futuras conexões ao mesmo servidor, fazendo com que, a partir da segunda conexão, o volume de dados transmitidos seja reduzido substancialmente. Você pode ver mais detalhes sobre o funcionamento interno destes componentes no: <http://www.nomachine.com/documentation.php>

A configuração do cache está disponível na seção "Advanced" na janela de configuração (no cliente). Existem dois caches, um mais rápido, feito na memória RAM e outro maior, armazenado no HD. Quanto maiores os caches, menos informações precisarão ser retransmitidas, melhorando o tempo de resposta, sobretudo em links lentos.



Na mesma tela está disponível a configuração do teclado. Por padrão, o FreeNX mantém a mesma configuração de teclado usada no cliente, ou seja, se você usa uma teclado ABNT2, as teclas de acentuação continuarão funcionando corretamente, mesmo ao acessar um servidor com um teclado Americano, configurado com o layout Turco. Você pode também

especificar manualmente um determinado layout de teclado, opção usada geralmente para solucionar problemas.

Outro detalhe que conta pontos para o FreeNX é que o cliente possui uma interface fácil de usar (bem voltada para o usuário final), e existem clientes tanto para Linux quanto para Windows. A principal limitação é que o servidor em si existe apenas em versão Linux, já que o objetivo é acessar máquinas Linux remotamente. Se você precisar acessar uma máquina Windows, o VNC ainda é a melhor opção.

O NXserver nasceu como um produto comercial, distribuído pelo <http://www.nomachine.com> como uma solução para redes com terminais leves. O servidor é pago e o cliente está disponível para download gratuito. Apesar disso, todas as bibliotecas usadas são open-source, o que possibilitou o desenvolvimento do FreeNX Server, que é gratuito. Usamos então o servidor FreeNX em conjunto com o cliente da NoMachine, chegando a uma solução completamente gratuita.

Por incrível que possa parecer, o FreeNX é desenvolvido com o apoio da NoMachine, que inclusive contribui com o projeto. Embora tenham funcionalidade similar, o servidor comercial possui uma interface muito melhor acabada, é mais fácil de instalar e conta com o suporte oficial. Isso faz com que o público corporativo (que são os principais clientes da NoMachine) pague pelo cliente comercial, enquanto o FreeNX permite atrair uma massa de usuários e colaboradores, que ajudam no desenvolvimento das bibliotecas base.

Este é, na minha opinião, um dos melhores exemplos de como uma empresa pode manter um modelo de negócios viável e ao mesmo tempo desenvolver código aberto, que beneficia um grande número de usuários.

» Próximo: [Instalando no Debian e derivados](#)

Originalmente, o FreeNX estava disponível apenas na forma de código fonte e pacotes .deb, hospedados no <http://archive.kalyxo.org>. Este servidor acabou sendo desativado e pacotes pré-compilados para outras distribuições foram desenvolvidos. Os pacotes originais, com o código fonte, estão atualmente disponíveis no <http://download.berlios.de/freenx/> e <http://debian.tu-bs.de/knoppix/nx/>.

Estes pacotes com o código fonte do FreeNX são particularmente complicados de compilar. Uma solução muito mais simples se você utiliza o Debian, Kurumin ou outra distribuição derivada do Debian Sid ou Etch é usar os pacotes do Kanotix, que estão disponíveis no:

<http://debian.tu-bs.de/project/kanotix/unstable/>

Como são vários pacotes (freenx, expectk, nxdesktop, nxlibs, nxagent e nxviewer), a forma mais simples e menos propensa a problemas é adicionar o endereço como fonte de atualização do apt. Assim você pode instalar tudo de uma vez, incluindo as dependências,

usando o apt-get. Abra o arquivo **"/etc/apt/sources.list"** e adicione a linha abaixo no final do arquivo:

```
deb http://debian.tu-bs.de/project/kanotix/unstable/ sid nx
```

Depois, rode o **"apt-get update"** para atualizar a lista de pacotes do apt e instale o FreeNX server com o comando:

```
# apt-get install freenx
```

Uma observação importante é que estes pacotes do Kanotix são compilados para o Debian unstable. Caso você esteja utilizando o stable ou testing será necessário baixar alguns pacotes atualizados a partir do unstable para utilizá-los. Para que o apt consiga baixar os pacotes necessários de forma automática, adicione linhas referentes ao unstable no final do arquivo **"/etc/apt/sources.list"**, como em:

```
deb http://ftp.de.debian.org/debian unstable main contrib
```

Depois da alteração, rode novamente o comando **"apt-get update"** e instale o FreeNX, usando o comando:

```
# apt-get install -t unstable freenx
```

Aqui estamos utilizando a opção **"-t unstable"**, que permite ao apt-get baixar as dependências necessárias a partir do unstable e não a partir do stable ou testing, que seria o padrão do sistema. Verifique com cuidado a lista de dependências antes de autorizar a instalação, para ter certeza de que não sejam feitas alterações indesejadas no sistema.

Se você está usando o **Ubuntu**, use o repositório Seveas, que inclui pacotes específicos para cada versão. Se você usa o Dapper Drake, por exemplo, adicione a linha abaixo no **"/etc/apt/sources.list"**:

```
deb http://seveas.ubuntulinux.nl/ dapper-seveas freenx
```

Se usa o Breezy Badger, adicione a linha:

```
deb http://seveas.ubuntulinux.nl/ breezy-seveas freenx
```

Você pode ver uma lista das versões e dos mirrors disponíveis no:

<https://wiki.ubuntu.com/SeveasPackages>

Depois de adicionado o repositório, rode o **"apt-get update"** e o **"apt-get install freenx"**, como no Debian.

Em qualquer um dos dois casos, é executado um script de instalação que pergunta:

Which NoMachine	key	type	should	freenx	use? Keys
--------------------	-----	------	--------	--------	--------------

Custom		keys
Remove	freenx	keys
Manual setup		

O FreeNX utiliza o SSH como meio de login, transporte e encriptação, por isso utiliza um par de chaves de encriptação para logar os clientes. O cliente NX, distribuído pela NoMachine, já vem com uma chave pré-instalada, por isso é mais simples (embora um pouco menos seguro) que o servidor utilize a mesma, o que elimina a necessidade de configuração manual, tanto ao conectar a partir de um cliente Linux, quanto a partir do Windows. Para isso, use a primeira opção, "NoMachine Keys" e sua vida será muito mais longa e feliz ;).

Caso você faça questão de usar uma chave exclusiva (opção "Custom keys"), vai precisar instalar a chave do servidor manualmente nos clientes. Se o servidor SSH estiver habilitado (no cliente), você pode copiar a chave diretamente a partir do servidor, usando o comando:

```
# scp /var/lib/nxserver/home/.ssh/client.id_dsa.key root@ip/usr/NX/share/
```

Neste comando, o "usuario" é o login que será usado na máquina cliente para se conectar ao servidor NX, seguido pelo IP da máquina (cliente). Você pode também transportar manualmente o arquivo "/var/lib/nxserver/home/.ssh/client.id_dsa.key" (do servidor) para dentro da pasta "/usr/NX/share/" nas máquinas clientes. Ao usar clientes Windows, o arquivo vai para a pasta "C:\Program Files\NX Client for Windows\share". Como pode perceber, é bem mais simples usar a primeira opção.

Caso necessário, você pode repetir esta configuração inicial, feita durante a instalação do pacote, usando o comando:

```
# nxsetup --install --setup-nomachine-key
```

Com isso, o FreeNX server já fica habilitado, mas você precisa adicionar, ainda, os usuários que terão acesso remotamente. O FreeNX usa um sistema parecido com o Samba, no qual você precisa primeiro criar os usuários no sistema, usando o comando "**adduser**" e, em seguida, adicioná-lo no FreeNX Server, usando os comandos "nxserver --adduser" e "nxserver --passwd". Para criar um usuário chamado "tux", por exemplo, os comandos seriam:

#	adduser	tux
#	nxserver	
# nxserver --passwd tux	--adduser	tux

Uma vez instalado, o FreeNX fica habilitado continuamente, acessível através do SSH. O fato de manter o FreeNX ativo, não reduz por si só a segurança do servidor, pois continua aberta apenas a porta do SSH. É necessário que o cliente primeiro se autentique no servidor SSH, para depois abrir a conexão com o servidor NX.

Desativando o SSH, você automaticamente desativa também o servidor NX. Você pode, ainda, desativar apenas o NX, usando o comando:

```
# nxserver --stop
```

» Próximo: [FreeNX no Mandriva](#)

Como disse a pouco, o FreeNX usa o SSH como meio de conexão. Por isso, antes de mais nada, você deve ter o servidor SSH instalado e ativo.

Para instalar o servidor SSH no Mandriva, use o comando:

```
# urpmi openssh-server
```

Para ativar o servidor, depois de instalado, use o comando:

```
# service sshd start
```

O pacote do FreeNX não faz parte dos CDs de instalação, mas está disponível nos repositórios "contrib", que contém justamente pacotes extra-oficiais, mantidos por colaboradores. Por padrão, o Mandriva procura por pacotes apenas nos CDs de instalação, por isso é necessário configurar o sistema para usar o repositório adicional. Para isso, acesse o: <http://easyurpmi.zarb.org/>.

Selecione a versão do Mandriva que está usando e marque a opção para adicionar a mídia "contrib". Um mirror que é rápido, no meu caso, é o "USA (mirrors.usc.edu)". O Easyurpmi lhe devolve o comando que deve ser executado como root para adicionar a mídia escolhida, como em:

```
# urpmi.addmedia contrib ftp://mirrors.usc.edu/pub/linux/distributions/\\
mandrakelinux/official/2005/i586/media/contrib with media_info/hdlist.cz
```

Você pode colar o comando no terminal usando o botão do meio do mouse. Ele vai baixar o pacote hdlist.cz, que contém os índices dos pacotes disponíveis no endereço. Depois de terminado, instale os pacotes usando os comandos:

```
# urpmi freenx
```

Para concluir a instalação, o urpmi baixará um conjunto de pacotes (dependências do FreeNX), alguns baixados a partir do contrib e outros copiados dos CDs de instalação. Para concluir a instalação, ajuste as chaves de autenticação, usando os comandos:

```
# chmod 640 /var/lib/nxserver/nxhome/.ssh/client.id_dsa.key
# mv /var/lib/nxserver/nxhome/.ssh/authorized_keys2 \
/var/lib/nxserver/nxhome/.ssh/authorized_keys
```

Falta, agora, adicionar os usuários que terão acesso remoto ao servidor NX, usando (para cada um) os comandos:

```
# nxserver --adduser tux  
# nxserver --passwd tux
```

Só para ter certeza, verifique se o servidor SSH e o FreeNX Server estão inicializados:

```
# service sshd start  
# nxserver --start
```

O pacote com o FreeNX do Mandriva não oferece a opção de usar a chave de encriptação do cliente da NoMachine durante a instalação. Por isso, depois de configurar o servidor você ainda precisará copiar o arquivo com a chave gerada durante a instalação do servidor para cada cliente que irá acessá-lo. Você pode usar o próprio SSH para transferir os arquivos (usando o scp, sftp ou o fish:// do Konqueror), usar um pendrive ou qualquer outra forma que ache prática.

Copie o arquivo "/var/lib/nxserver/nxhome/.ssh/client.id_dsa.key" para a pasta "/usr/NX/share/" de cada cliente. Depois de copiar o arquivo, acerte as permissões de acesso (em cada cliente) usando o comando "chmod 644 /usr/NX/share/client.id_dsa.key". No caso dos clientes Windows, copie a chave para dentro da pasta "C:\Program Files\NX Client for Windows\share" e verifique se o arquivo está com permissão de leitura para todos os usuários.

Se você não quiser ter este trabalho todo para copiar e gerenciar as chaves (o que é problemático quando o mesmo cliente precisa se conectar em diferentes servidores), é possível forçar o servidor FreeNX no Mandriva a usar a chave padrão da NoMachine, como nos pacotes para o Debian e Ubuntu, usando o comando:

```
# nxsetup --install --setup-nomachine-key --force
```

» Próximo: [FreeNX no Fedora](#)

A instalação do FreeNX no Fedora é bem simples, graças ao trabalho do Rick Stout, que há várias versões mantém os pacotes necessários. Você pode baixar os pacotes no:

http://fedoranews.org/contributors/rick_stout/freenx/

Na página estão disponíveis pacotes para várias versões do Fedora. Você vai precisar de dois pacotes: o nx e o freenx. Para o Fedora 5, por exemplo, você baixaria os pacotes "freenx-0.4.9-050test7.FC5.1.noarch.rpm" e "nx-1.5.0-7.FC5.1.i386.rpm".

Comece usando o yun para baixar os pacotes expect e nc (as dependências) e, em seguida, use o comando "rpm -Uvh" para instalar os dois pacotes previamente baixados:

```
#           yum      install      expect      nc
#           rpm      -Uvh      freenx-0.4.9-050test7.FC5.1.noarch.rpm
# rpm -Uvh nx-1.5.0-7.FC5.1.i386.rpm
```

O segundo pacote inclui um script de instalação, que se encarrega de rodar o comando nxsetup e cadastrar um login de usuário no NX.

Assim como no Mandriva, o pacote do Fedora usa uma chave personalizada ao invés da chave da NoMachine, por isso é necessário fazer a instalação manual das chaves, copiando a chave de encriptação gerada pelo servidor para cada cliente. Copie o arquivo "/var/lib/nxserver/home/.ssh/client.id_dsa.key" do servidor para a pasta "/usr/NX/share/" ou "C:\Program Files\NX Client for Windows\share" de cada cliente e certifique-se que todos os usuários têm permissão de leitura para o arquivo.

Ao invés disso, você pode simplesmente reconfigurar o servidor NX, de forma que ele use as chaves da NoMachine, usando comando:

```
# nxsetup --install --setup-nomachine-key
```

» Próximo: [FreeNX no SuSE](#)

A partir do SuSE 10, o FreeNX vem incluído nos CDs de instalação, permitindo que você instale diretamente através do Yast. Acesse a aba de gerenciamento de pacotes e faça uma busca por "freenx".

Caso esteja com o SuSE Firewall ativo, não se esqueça de alterar a configuração, liberando os acessos na porta 22 (SSH). Com isso, o servidor FreeNX estará instalado, falta apenas fazer a configuração inicial, usando o comando:

```
# nxsetup --install --setup-nomachine-key --clean --purge
```

O "--clean --purge" força o uso da chave da NoMachine, removendo a configuração padrão do pacote. Ela tem uma função similar ao "--force" que usamos na configuração para o Mandriva.

» Próximo: [Usando o cliente NX](#)

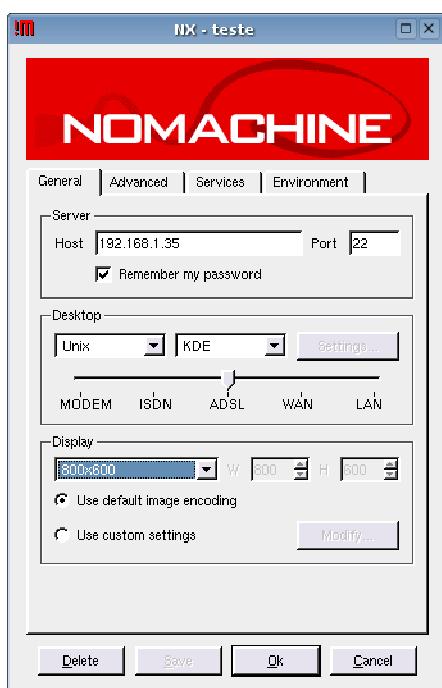
Para se conectar ao servidor a partir dos clientes, você precisa primeiro baixar o NXclient

no: <http://www.nomachine.com/download.php>. Existem versões para várias distribuições Linux, Windows e Solaris, todas de download gratuito. Na maioria das distribuições, ao instalar o cliente NX, são criados ícones no "Iniciar > Internet".

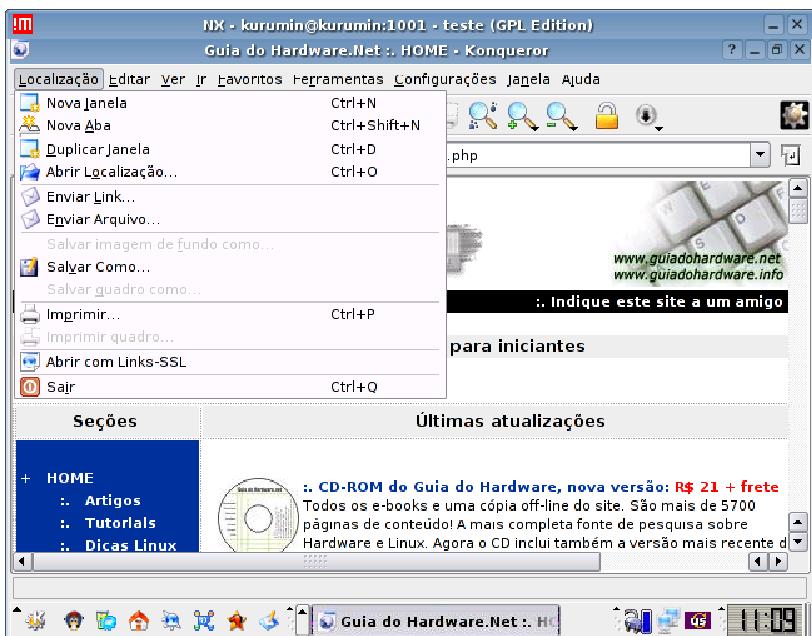
Além do IP do servidor, você precisa informar a porta em que ele está escutando. O FreeNX usa o SSH para estabelecer a conexão, por isso a porta padrão é a 22. Caso você queira mudar a porta usada pelo servidor, é necessário editar dois arquivos.

Comece editando o arquivo "**/etc/ssh/sshd.conf**" (no servidor), alterando a opção "port 22" para a porta desejada. Em seguida, abra o arquivo "**/usr/lib/nx/nxloadconfig**" e altere a opção "SSHD_PORT=22", de forma que seja informada a mesma porta nos dois arquivos. Para aplicar a alteração, reinicie o servidor SSH e, em seguida, também o servidor NX, usando o comando "nxserver --restart".

De volta à configuração do cliente, estão disponíveis ainda opções com o nível de compressão dos dados e do tamanho da janela. Usando a opção "LAN", que é destinada a conexões via rede local, não existe perda de qualidade de imagem, mas, ao usar as demais opções, destinadas a conexões mais lentas, as imagens são comprimidas via JPG, o que garante uma atualização mais rápida, porém com uma certa perda de qualidade. Conforme você vai usando, os pixmaps vão sendo retransmitidos e substituídos por cópias sem compressão, fazendo com que a imagem vá ficando mais nítida.



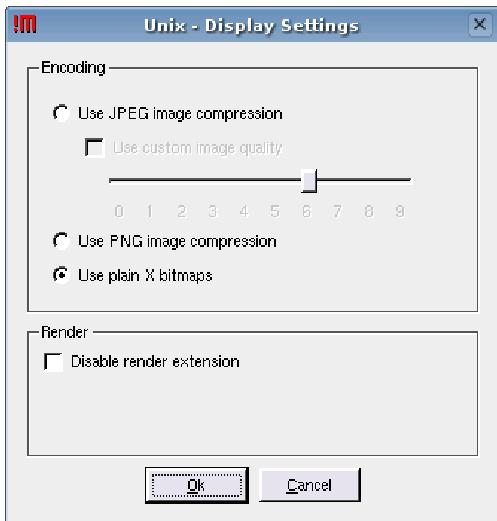
Na ilustração abaixo, estou me conectando a um servidor com o Kurumin via ADSL (256k), abrindo uma tela de 640x480. Você pode notar que alguns pontos da imagem estão um pouco embacados por causa da compressão via JPG, mas no geral a imagem está nítida e os tempos de resposta são bem superiores aos obtidos com o VNC na mesma conexão.



Ao acessar via rede local, você pode desativar a compressão em JPG, evitando a perda de qualidade da imagem. Para isso, na configuração do cliente NX, acesse a opção "General > Display > Use custom settings > Modify".

Aqui você tem a opção de ajustar o nível de compressão do JPG (os níveis mais altos são úteis em conexões via modem), usar compressão via PNG (um formato sem perda) ou desabilitar completamente a compressão usando a opção "Use plain X bitmaps". Esta última opção consome mais banda da rede (o que não chega a ser um problema em uma rede de 100 megabits), mas em troca usa menos processamento, tanto no servidor quanto no cliente.

Em versões antigas do FreeNX, a compressão em PNG não funciona corretamente, fazendo com que, ao escolher a opção, continue sendo usada a compressão em JPG. Nesses casos, a única opção é usar os bitmaps do X, sem compressão. Uma observação importante é que essa opção reduz a eficiência do cache local do FreeNX, já que com arquivos maiores ele passa a armazenar um número muito menor de arquivos. A solução, nesse caso, é aumentar o tamanho do cache em memória de 8 para 32 MB na aba "Advanced".

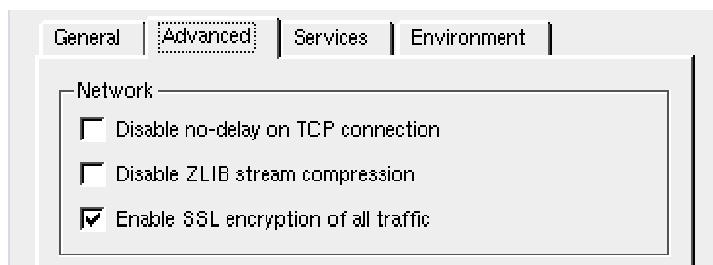


Para que você consiga se conectar a um servidor FreeNX com firewall ativo, é necessário que o firewall no servidor esteja com a porta 22 e as portas de 5000 a 5200 abertas. Você pode abrir as portas necessárias no Iptables, incluindo as duas regras a seguir no início do script de firewall do servidor:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 5000:5200 -j ACCEPT
```

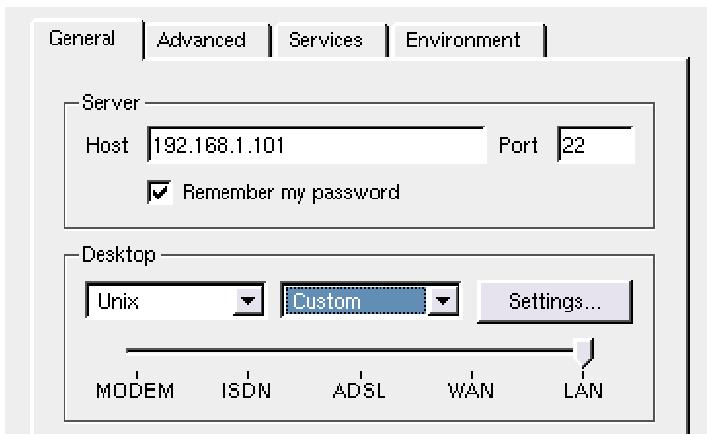
As portas de 5000 a 5200 são necessárias por que por padrão o cliente NX utiliza o SSH para fazer autenticação e estabelecer a conexão inicial, mas depois utiliza outra porta para a transmissão dos dados. É possível fazer com que todo o tráfego seja transmitido através do SSH. Isso faz com que a sessão fique com tempos de resposta um pouco mais altos (pois o sistema tem o trabalho de descriptar as informações antes de fazer cada atualização), mas, por outro lado, melhora a segurança, pois tudo passa a ser encriptado.

Ativando a encriptação passa a ser utilizada apenas a porta do SSH (22), o que torna desnecessária a segunda regra de firewall. A opção está na tela de configuração da conexão, em "Advanced > Enable SSL encryption of all traffic".

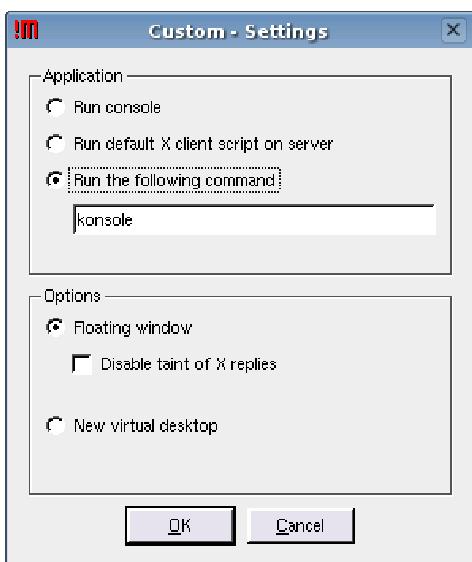


No cliente, não é necessária nenhuma configuração em especial. Você continua conseguindo se conectar ao servidor mesmo que o firewall bloqueeie completamente as conexões de entrada. Basta ter acesso (de saída) ao servidor através da porta 22.

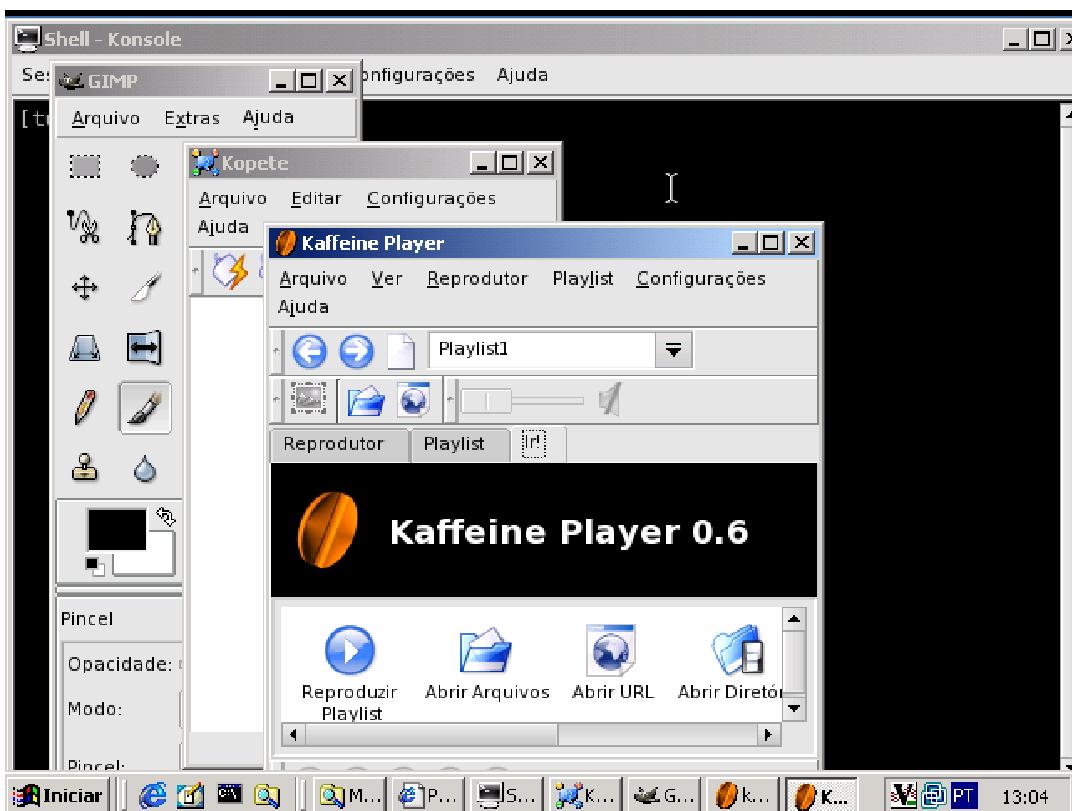
Além de abrir um desktop completo (como no VNC), o FreeNX pode ser configurado para abrir um aplicativo específico. Isso é especialmente útil quando você tem clientes Windows que precisam rodar um aplicativo específico a partir de um servidor Linux. Para isso, abra a janela de configuração e accese a opção "General > Desktop". Selecione a opção "Custom"



Clique agora no botão "Settings". Na janela que será aberta, marque a opção "Run the following command" e especifique o comando do aplicativo que você deseja que seja executado, como, por exemplo, "gimp" ou "konqueror". Marque também a opção "Floating window" ao invés de "New virtual desktop".



Um detalhe interessante é que se você abrir um terminal, usando o comando "konsole" ou "xterm", você poderá chamar qualquer outro aplicativo instalado no servidor a partir dele. No screenshot a baixo, por exemplo, configurei o cliente NX para abrir uma janela do Konsole, que usei para abrir o Gimp, o Kopete e o Kaffeine. Como pode ver, todos estão rodando integrados ao desktop do Windows, com janelas próprias e botões na barra de tarefas.



Ao criar uma nova conexão, o cliente NX oferece a opção de criar um ícone no desktop. Ao clicar sobre o ícone criado, é aberto o aplicativo configurado, que é exibido no cliente como se fosse um aplicativo nativo, aparecendo na barra de tarefas e tudo mais. É uma boa forma de "rodar" aplicativos Linux no Windows e impressionar seus amigos, ou "quebrar o gelo" dentro de projetos de migração, implantando os aplicativos Linux que serão futuramente usados, inicialmente dentro do Windows (através do FreeNX) e, depois, nativamente.

» Próximo: [FreeNX e VNC](#)

Existem muitas diferenças na forma como o FreeNX e o VNC trabalham. No VNC, a sessão fica aberta mesmo depois de fechar a janela, o que permite que você continue trabalhando na mesma sessão, mesmo depois de ir para outro micro. No FreeNX, é possível se reconectar a uma sessão suspensa apenas a partir do mesmo cliente.

No VNC é preciso rodar o comando "vncserver" no servidor (o que remotamente é geralmente feito conectando primeiro via SSH), para abrir cada sessão, e o comando "vncserver --kill :1" (onde o :1 é o número da sessão), para fechá-la, tudo feito manualmente. No FreeNX, o servidor principal fica residente e as sessões vão sendo abertas automaticamente, conforme os usuários se conectam. Cada usuário loga-se usando seu próprio login e senha.

O VNC não oferece suporte a antialiasing de fontes e outros recursos gráficos, o que faz com que a qualidade da imagem não seja a mesma que ao usar a máquina localmente. No FreeNX esses recursos são melhor suportados.

Versões antigas no VNC tinham problemas com a acentuação em teclados ABNT, isso foi resolvido nas versões recentes. No FreeNX a configuração do teclado é independente do servidor, você pode configurar o teclado no servidor da forma que for necessária, pois vale a configuração do cliente.

O VNC trabalha tirando screenshots da tela e compactando as imagens antes de enviar via rede. Ao chegar no cliente, as imagens precisam ser descompactadas, montadas e só então exibidas na tela. É preciso que tanto o servidor quanto o cliente sejam relativamente rápidos (600 MHz ou mais) para obter uma boa velocidade. O FreeNX usa o sistema de comunicação do X, combinado com compactação e encriptação via SSH, uma combinação que consome bem menos processamento, tanto no cliente quanto no servidor. O requisito mínimo nos clientes seria algo como um Pentium 100.

No VNC não existe encriptação dos dados, você pode usar um túnel via SSH para obter uma conexão segura (como vimos a pouco), mas isso torna a conexão mais lenta e a configuração mais trabalhosa. O FreeNX já oferece encriptação nativamente, embora seja possível desabilitá-la para diminuir um pouco o uso de recursos do sistema.

O VNC não oferece nenhum sistema para transferir arquivos entre o servidor e o cliente, é preciso manter um servidor FTP, SSH, Samba ou NFS aberto para isso. No FreeNX é possível usar o próprio SSH para transferir arquivos, usando o **sftp**. Se você está usando um cliente Linux, a forma mais fácil de fazer isso é usar o módulo "**fish://**" do Konqueror. Abra o gerenciador de arquivos e escreva na barra de endereços: "**fish://morimoto@192.168.1.35**", onde o "morimoto" é o login e o "192.168.1.35" é o IP do servidor. Ele pedirá a senha e em seguida mostrará os arquivos do servidor. Você pode transferir arquivos simplesmente arrastando-os para uma janela do gerenciador de arquivos local.

» Próximo: [Acessando máquinas Windows via RDP \(WTS\)](#)

Embora você possa acessar máquinas Windows remotamente usando o VNC, o Windows possui um protocolo próprio de acesso remoto, o RDP, que é mais eficiente que o VNC (sobretudo via Internet) e permite que vários clientes abram sessões independentes no mesmo servidor, o que é impossível ao usar o VNC for Windows.

O maior obstáculo é a questão do licenciamento, pois além da licença do servidor, você precisa de licenças para os clientes. As máquinas Windows XP também podem ser acessadas remotamente, mas sem suporte a várias conexões simultâneas (quando você se loga remotamente, ele coloca a sessão local em espera e ao se logar localmente ele fecha a

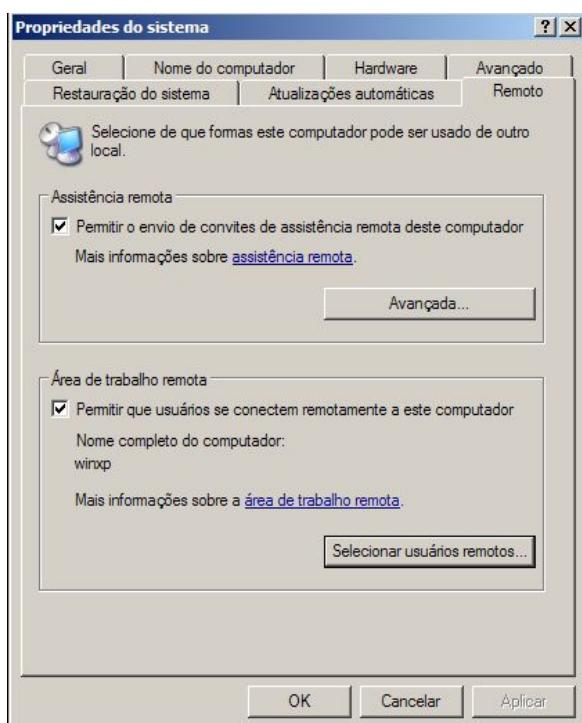
conexão remota), enquanto nas versões server o número de sessões simultâneas é limitado apenas ao hardware do servidor e ao número de licenças.

Uma solução para permitir mais conexões simultâneas em máquinas com o Windows XP é usar o XP Unlimited, que remove a barreira técnica, permitindo abrir um número indefinido de conexões, como no Windows 2003 Server.

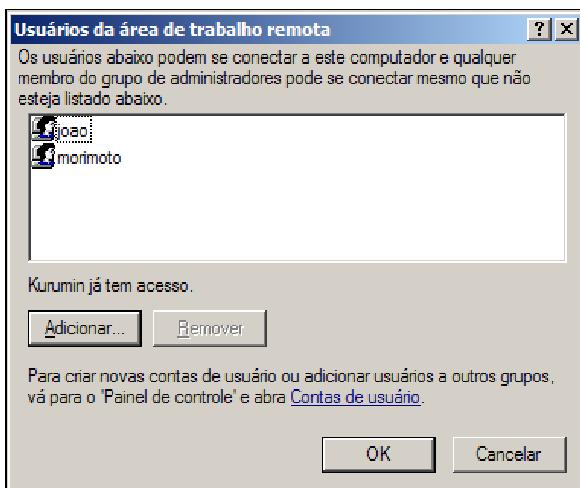
A versão demo disponível no <http://www.xpunlimited.com/demo.html> (apenas para uso não-comercial) permite três conexões simultâneas, enquanto a versão completa custa US\$ 85. Note que embora o XP Unlimited remova a limitação técnica, a questão do licenciamento fica nebulosa, já que em tese o Windows XP não poderia ser usado simultaneamente por mais de um usuário, sem que cada um tivesse uma licença. Verifique essa questão antes de considerar o uso em ambientes de produção.

Para ativar o acesso remoto em uma máquina Windows, clique com o botão direito no "Meu Computador" e, no menu "Propriedades do Sistema", acesse a aba "Remoto" e marque a opção "Área de trabalho remota".

Clique no botão "Selecionar usuários remotos" e indique quais logins de acesso poderão ser usados remotamente. Por padrão, apenas o Administrador e o usuário logado atualmente podem acessar.



É importante enfatizar que apenas os usuários com senhas definidas podem acessar as máquinas remotamente. Todos os logins sem senha são automaticamente recusados. Você pode definir as senhas na seção "Contas de usuário" do Painel de Controle.



Em caso de problemas na ativação, acesse a opção "Ferramentas administrativas > Serviços" do Painel de Controle e verifique se os serviços "Alocador Remote Procedure Call (RPC)" e "Serviços de terminal" estão ativados.

Com o acesso remoto ativado na máquina Windows, vamos ao tema central deste tópico, que é justamente como acessá-la remotamente a partir de clientes Linux. Esta solução é muito usada por empresas que migram as estações de trabalho para Linux, mas precisam manter algumas cópias do Windows para rodar alguns aplicativos específicos. Ao invés de manter máquinas com o Windows, ou rodá-lo via VMware, pode fazer mais sentido manter um servidor Windows na rede, com o acesso remoto ativado e permitir que os usuários abram sessões remotas quando necessário.

Nos clientes Linux, usamos o **rdesktop**, que pode ser tanto utilizado via linha de comando, quanto através do **TScient**, **Krde** ou outra das interfaces de acesso remoto que oferecem suporte a ele. O uso mais simples para o rdesktop é simplesmente passar o endereço IP ou domínio da máquina remota como argumento, como em:

\$ rdesktop 192.168.0.1

O problema é que ele vai utilizar todas as opções default, abrindo uma tela de 800x600 com 256 cores. O protocolo RDP v5 usado no XP e 2003 server, suporta o uso de 16 bits de cor. Para ativar o recurso, inclua as opções "-5 -a 16" (o -5 é a versão do protocolo e o -a 16 especifica os bits de cor), como em:

\$ rdesktop -5 -a 16 192.168.0.1

Para especificar a resolução, use a opção "-g", seguida pela resolução desejada, como em:

\$ rdesktop -5 -a 16 -g 1000x700 192.168.0.1

Ao especificar a resolução, você pode usar qualquer número que adapte a janela ao seu desktop. Não é necessário se limitar às resoluções padrão. Para abrir a sessão em tela cheia, use a opção "-f", como em:

```
$ rdesktop -5 -a 16 -f 192.168.0.1  
(pressione "Ctrl+Alt+Enter" para chavear entre o modo fullscreen e janela)
```

Ao acessar uma máquina XP ou 2003 server, você pode também redirecionar o som para o cliente, de forma que os sons dos aplicativos sejam tocados usando a placa de som e caixas do seu micro, ao invés de no servidor. Funciona mesmo que o servidor não possua placa de som.

Esse é um recurso que deve ser usado com cautela em redes com muitos clientes, ou via Internet, pois gera um fluxo de aproximadamente 800 kbits para cada cliente usando o som. Para ativar, adicione a opção "-r sound:local=/dev/dsp", como em:

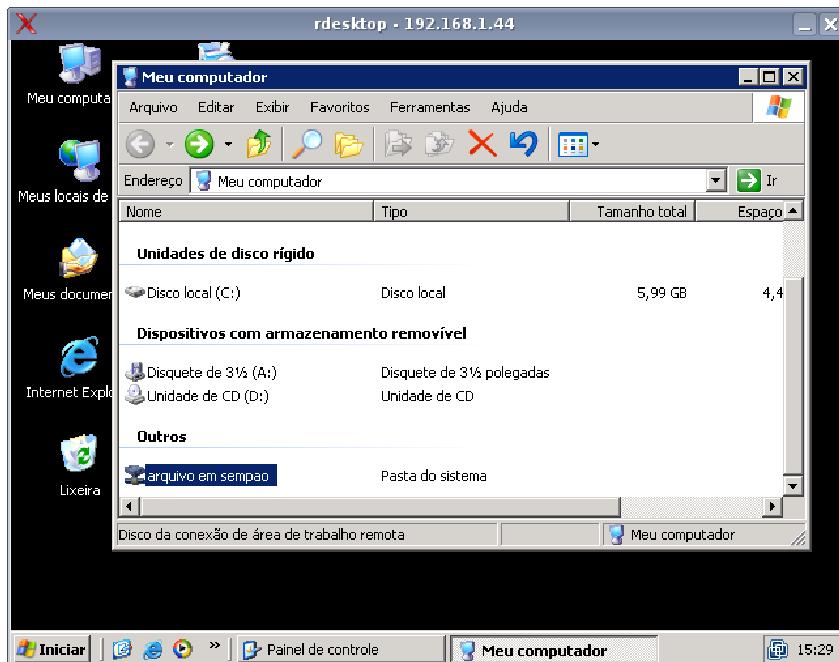
```
$ rdesktop -5 -a 16 -r sound:local=/dev/dsp 192.168.0.1
```

Note que o "/dev/dsp" indica o dispositivo da placa de som no cliente. Se não funcionar da primeira vez, verifique as permissões de acesso (no cliente). Caso necessário, abra as permissões usando o comando "chmod 666 /dev/dsp" (como root, no cliente).

É possível também "compartilhar" pastas no cliente, de forma que os arquivos sejam acessados dentro da sessão remota. Você pode, por exemplo, editar documentos em uma pasta dentro do seu home, usando os programas instalados no servidor. Para isso, adicione a opção "-r disk:nome=pasta", onde o "nome" indica como ele será visto dentro da sessão e a "pasta" é a pasta no cliente que está sendo "compartilhada". Esta opção pode ser usada em combinação com as anteriores, como em:

```
$ rdesktop -5 -a 16 -r sound:local=/dev/dsp -r disk:arquivo=/home/joao 192.168.0.1
```

As pastas compartilhadas aparecem dentro do "Meu Computador > Outros", como se fossem compartilhamentos de rede montados:



Para compartilhar o CD-ROM, pendrive ou disquete, basta indicar a pasta onde eles ficam acessíveis, como em "-r disk:cdrom=/mnt/cdrom" ou "-r disk:pendrive=/mnt/pendrive". A observação, nesse caso, é que você vai sempre precisar montar o CD-ROM ou pendrive no cliente para acessá-lo dentro da sessão remota. O comando simplesmente compartilha os arquivos acessíveis dentro da pasta.

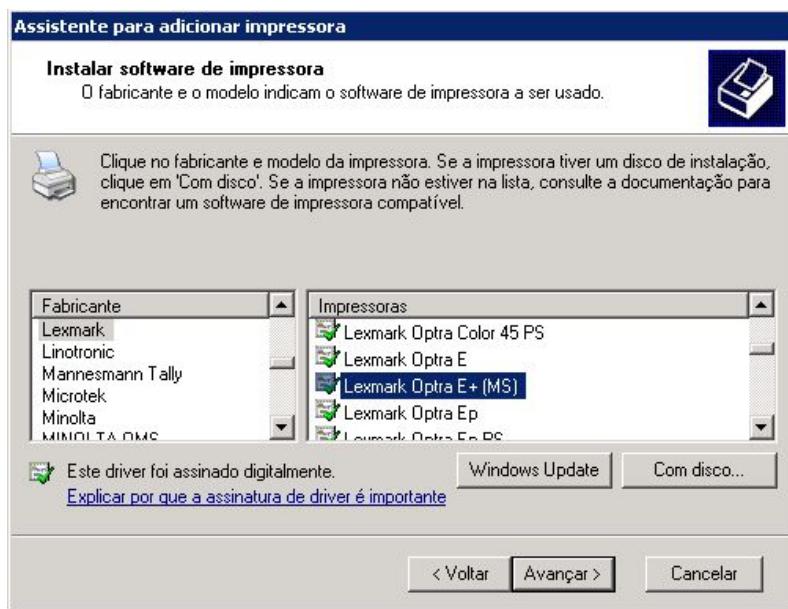
É possível, ainda, mapear a **impressora**, de forma que você consiga imprimir na impressora instalada no seu cliente Linux de dentro dos aplicativos na sessão remota. Se os clientes e o servidor estão na mesma rede local, é mais simples compartilhar a impressora via Cups ou Samba e instalá-la no servidor. O mapeamento de impressoras do RPD, por sua vez, permite usar as impressoras quando isso não é uma opção, como ao acessar um servidor via Internet.

Em primeiro lugar, a impressora deve estar instalada no cliente e você deve conseguir imprimir nela usando o lpr. Nas distribuições derivadas do Debian, instale o pacote "**cupsys-bsd**" (que substitui o lpr); caso contrário, nada vai funcionar.

Ao conectar no servidor, é preciso especificar o nome da impressora, da forma como é vista pelos aplicativos no cliente e também o driver Windows (esta é a parte mais complicada...) que o servidor vai usar na hora de enviar trabalhos para ela, como em:

```
$ rdesktop -5 -a 16 -r printer:e230="Lexmark Optra E+ (MS)" 192.168.0.1
```

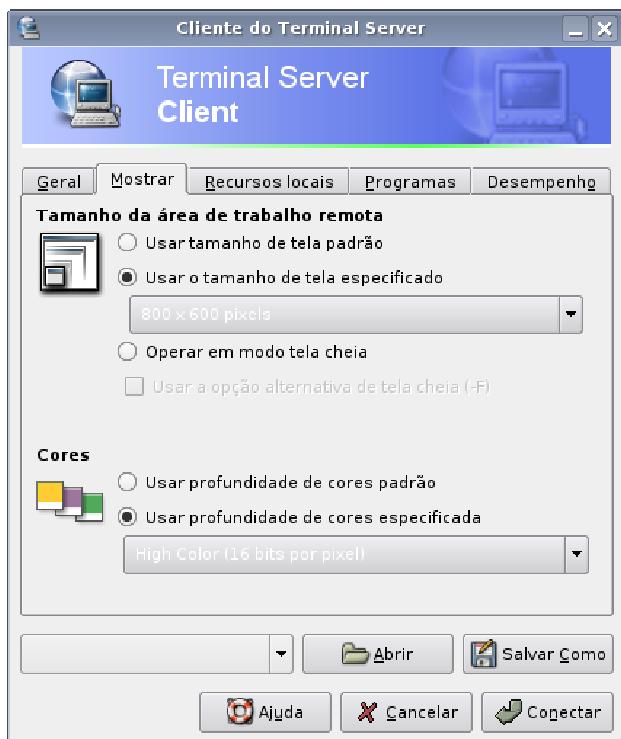
Para descobrir o driver da Impressora no Windows, abra o menu de instalação de impressora, indique o fabricante e copie o nome que aparece no menu da esquerda:



No caso de impressoras paralelas, você pode também redirecionar a porta "/dev/lp0". Nesse caso, você poderia instalar a impressora dentro da sessão remota, como se ela estivesse instalada no próprio servidor, adicionando o parâmetro "**-r lptport:LPT1=/dev/lp0**" ao

comando de conexão. É possível, ainda, redirecionar portas seriais, usando a opção "**-r comport:COM1=/dev/ttyS0**".

Como viu, o rdesktop suporta um grande número de opções, o que torna os comandos de acesso bastante longos. É aí que entra o TSclient, que permite especificar as opções através de uma interface muito mais amigável. Ele está disponível em várias distribuições; nas derivadas do Debian, você pode instalá-lo via apt-get. A página oficial é a <http://gnomepro.com/tsclient/>.



» Próximo: [Configurando um servidor XDMCP](#)

Apesar de ter lá suas falhas, como um baixo desempenho de vídeo em conjunto com alguns modelos de placas de vídeo e uma arquitetura considerada "ultrapassada", por uns, e "inchada", por outros, o bom e velho X continua seguindo firme como o servidor gráfico mais usado no Linux.

Parte do sucesso se deve à fartura de recursos de rede disponíveis no X. De fato, ele foi desenvolvido inicialmente para servir terminais burros, com aplicativos gráficos executados num servidor central. No início da década de 80, esta era praticamente a única forma viável de rodar aplicativos gráficos em estações de trabalho, já que os computadores com o poder de processamento necessário eram muito caros. Era possível, então, diluir o custo entre

vários clientes, assim como hoje em dia os serviços de hospedagem utilizam um único servidor para hospedar vários sites.

Isso, claro, era feito usando alguma das versões do Unix disponíveis então. Por rodar em vários sabores de Unix, não foi difícil portar o X também para o Linux. Com o crescimento do sistema, houve um grande aumento no número de usuários e desenvolvedores trabalhando no X, fazendo com que o ritmo de desenvolvimento também se acelerasse bastante. Basta comparar o suporte a placas de vídeo do Xfree 3.3.6 e do 4.4 ou X.org, por exemplo. Embora o 3.3.6 suporte um número maior de placas antigas, o suporte às placas recentes não se compara nas duas versões.

Felizmente, apesar de toda a evolução, o X não abandonou suas raízes. Continuamos tendo o mesmo sistema de cliente e servidor usado no início da década de 80. Isso permite que o X ofereça algumas vantagens-chave sobre o Windows e outros sistemas:

1- É possível abrir vários servidores X e rodar não apenas aplicativos, mas também gerenciadores de janelas diferentes em cada um. Sim, você pode rodar o KDE e o Gnome, junto com o Window Maker, Blackbox, etc. todos ao mesmo tempo ;).

2- Os servidores X não estão limitados a rodar aplicativos locais, eles podem rodar aplicativos a partir de qualquer micro da rede, ou de qualquer ponto da internet. Este sistema de compartilhamento do X é chamado de XDMCP.

3- Ao rodar aplicativos remotamente a carga fica toda com o servidor. O cliente utiliza um mínimo de processamento, já que basicamente se limita a enviar os dados recebidos via rede para a tela. Usando um 486 com 8 MB de RAM já é possível ter um terminal X funcional.

4- Ao contrário do VNC, o X consegue uma velocidade de atualização de tela muito boa, mesmo em uma rede de 10 megabits. A principal diferença é que enquanto o VNC transmite a tela na forma de uma imagem, o X transmite apenas texto, com as instruções necessárias para o cliente montar as janelas. Apenas figuras e ícones são transmitidos na forma de imagem.

Lembre-se de que, nesse aspecto, o FreeNX ganha de ambos, pois ele utiliza o protocolo do X combinado com compressão e encriptação dos dados. A desvantagem do FreeNX é que ele utiliza mais processamento, por isso não é tão interessante ao usar micros antigos como clientes.

» Próximo: [Abrindo diversos terminais gráficos](#)

Pressionando Ctrl+Alt+F2 dentro da interface gráfica, você irá para um terminal independente, onde poderá inclusive logar-se como outro usuário. Você tem 6 destes

terminais, que podem ser acessados pressionando Alt+F1, F2, F3, F4, F5 ou F6. A partir dos terminais de texto, você pode voltar ao modo gráfico pressionando Alt+F7.

Mas qual é a função das teclas F8 a F12? Elas servem para alternar entre servidores X. Assim como é possível alternar entre os terminais, é possível alternar entre vários terminais gráficos diferentes, teclando Ctrl+Alt+F8, F9, etc. Você pode logar-se como um usuário diferente em cada um, rodar aplicativos diferentes, ou até mesmo usar interfaces gráficas diferentes.

Para abrir mais servidores X, basta mudar para um terminal de texto (Ctrl+Alt+F2 ou outra tecla até F6) e rodar o comando "**startx -- :1**", onde o 1 pode ser substituído por outro número, caso você pretenda abrir vários servidores X, como em:

```
#               startx          --          :2
# startx -- :3
```

Se você quiser abrir vários servidores X com interfaces gráficas diferentes, substitua o "**startx**" por "**xinit**", como em: "**xinit -- :2**". Isso abrirá um servidor X sem gerenciador de janelas algum, apenas com uma janela de terminal, que você utilizará para abrir a interface gráfica desejada. Basta dar o comando adequado:

startkde	:	para	abrir	o	KDE
gnome-session	:	usar	o	Gnome	
wmaker	:	Window		Maker	
fluxbox : usar o Fluxbox					

Você pode chamar qualquer interface gráfica que tenha instalada. Ao sair, você voltará para o servidor X "puro" e poderá inicializar outra interface. Se preferir encerrar a sessão, digite "**exit**" no terminal.

Para abrir diretamente a interface desejada, sem precisar passar pela tela cinza do X, adicione o parâmetro "**-e**" seguido pelo comando que inicia a interface desejada no comando do xinit, como em:

```
# xinit -- :2 -e fluxbox
```

Note que cada servidor X consome uma quantidade considerável de memória, principalmente se você utilizar uma interface diferente em cada um. Use esse recurso com parcimônia em micros antigos, com pouca memória RAM.

» Próximo: [Ativando o compartilhamento](#)

Naturalmente, o suporte a XDMCP, que permite às outras máquinas da rede rodar aplicativos instalados no servidor, vem desativado por padrão. O XDMCP não utiliza

nenhum tipo de encriptação ou compressão, simplesmente transmite os dados da forma mais simples e rápida possível. Você se loga no servidor, carrega o KDE ou outra interface da sua preferência e roda todos os aplicativos instalados no servidor, de uma forma completamente transparente e um desempenho muito bom, mesmo em uma rede de 10 megabits.

Por um lado isso é bom, pois o overhead é muito pequeno, mas por outro lado é ruim, pois não existe muita segurança. O XDMCP deve ser usado apenas dentro da rede local; use sempre um firewall para bloquear conexões provenientes da internet.

Para configurar o servidor para aceitar as conexões, é preciso alterar dois arquivos. O primeiro faz com que o X aceite conexões remotas e o segundo configura o gerenciador de login (KDM ou GDM) para escutar as conexões e fornecer a tela de login aos clientes.

Nas distribuições que utilizam o **KDM** (como o Kurumin, Kubuntu e Mandriva), procure pelos arquivos **kdmcrc** e **Xaccess** (que sempre ficam na mesma pasta). Em algumas distribuições (como no Mandriva) eles ficam na pasta **"/usr/share/config/kdm/"** e, em outras (como no Kurumin e outras distribuições derivadas do Debian), ficam na pasta **"/etc/kde3/kdm/"**. Você pode usar o comando "locate" para encontrá-los.

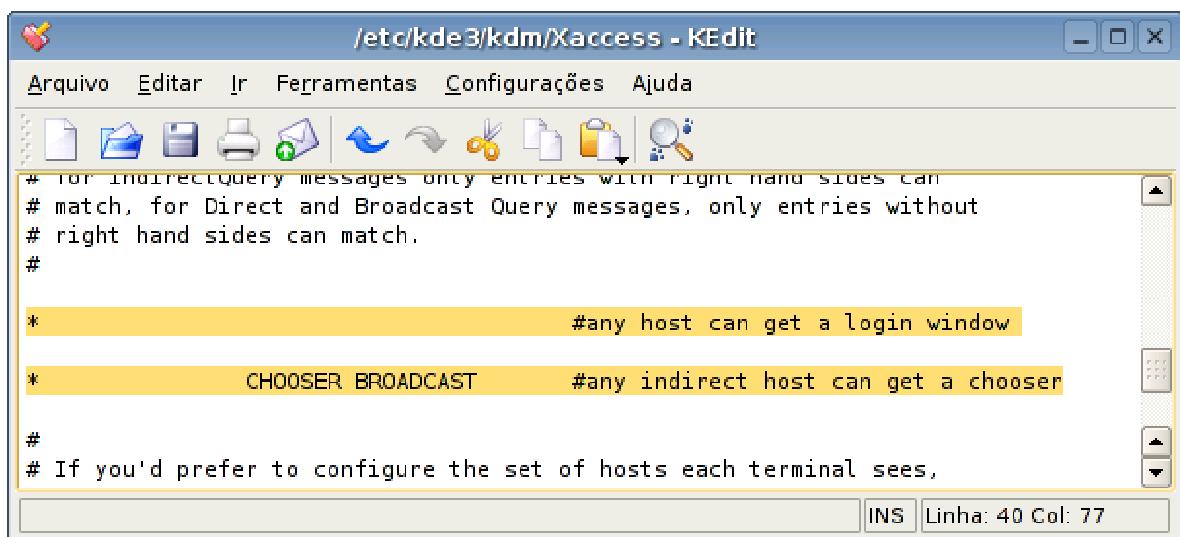
Dentro do arquivo "**Xaccess**", descomente a linha:

```
# * #any host can get a login window
```

Basta retirar a tralha (#), fazendo com que o asterisco seja o primeiro caractere. Esta linha faz com que o servidor passe a aceitar conexões de todos os hosts da rede. Caso você prefira limitar o acesso a apenas alguns endereços (mais seguro), basta substituir o asterisco pelos endereços desejados.

Um pouco mais abaixo, no mesmo arquivo, descomente também a linha abaixo, novamente retirando a tralha:

```
# * CHOSER BROADCAST #any indirect host can get a chooser
```



```
* for indirect query messages only entries with right hand sides can
# match, for Direct and Broadcast Query messages, only entries without
# right hand sides can match.
#
*                      #any host can get a login window
*          CHOOSE BROADCAST      #any indirect host can get a chooser
#
# If you'd prefer to configure the set of hosts each terminal sees,
```

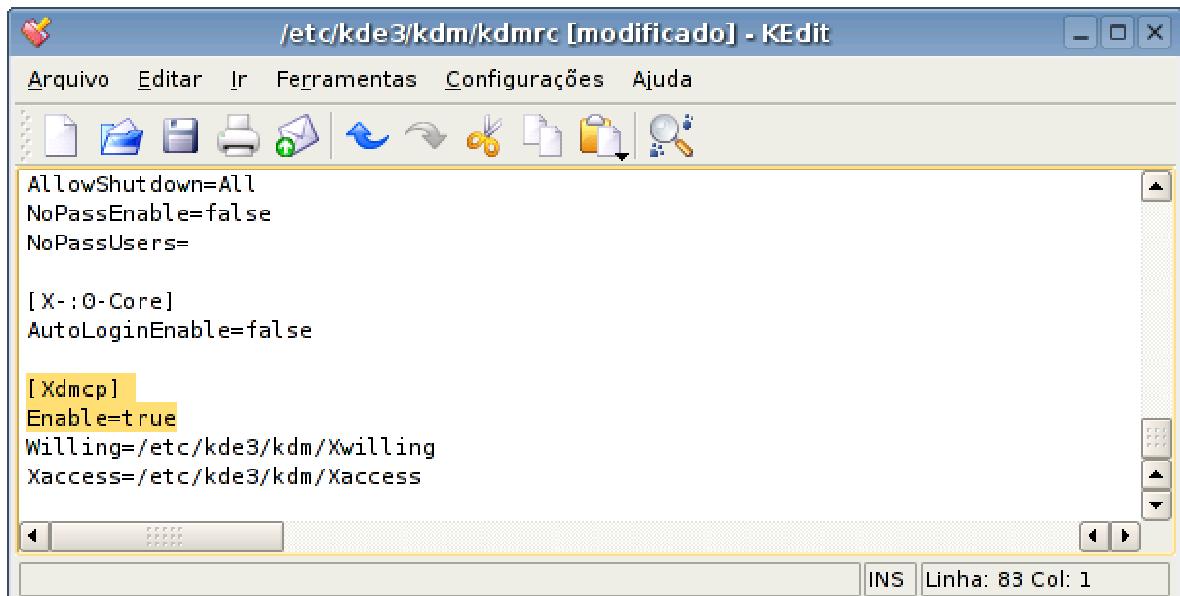
Esta linha é opcional. O Chooser Broadcast permite que os clientes contatem o servidor para obter uma lista de todos os servidores XDM disponíveis na rede (você pode ter mais de um, como veremos a seguir). Isso é feito usando o comando "**X -indirect**".

Em seguida, edite também o arquivo **kdmcrc**. Quase no final do arquivo você encontrará a linha:

```
[Xdmcp]
Enable=false
```

Basta alterá-la para:

```
[Xdmcp]
Enable=true
```

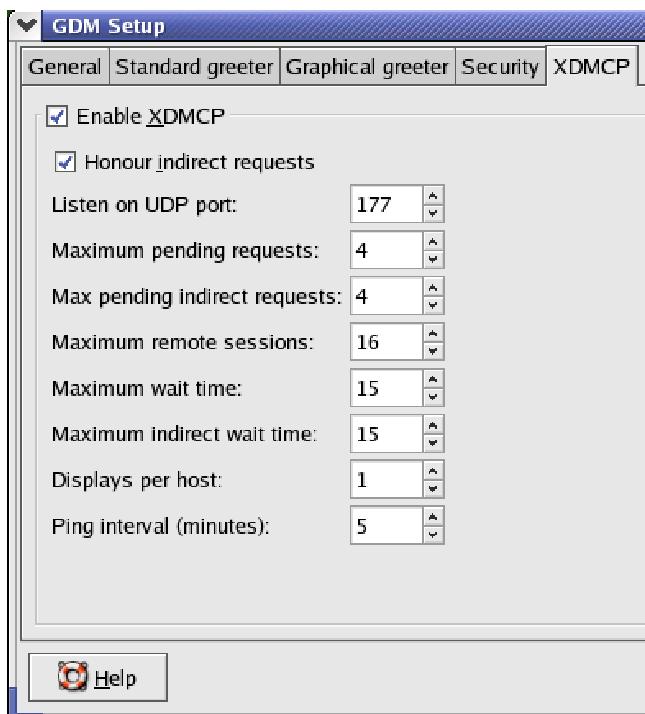


```
AllowShutdown=All
NoPassEnable=false
NoPassUsers=

[X-:0-Core]
AutoLoginEnable=false

[Xdmcp]
Enable=true
Willing=/etc/kde3/kdm/Xwilling
Xaccess=/etc/kde3/kdm/Xaccess
```

Nas distribuições que utilizam o **GDM** (o gerenciador de login do Gnome), como, por exemplo, o Fedora, você pode ativar o compartilhamento e configurar as opções da tela de login usando um utilitário gráfico disponível no "**Iniciar > System Settings > Login Screen**". Acesse a aba "XDMCP" e marque a opção "Enable XDMCP" e, se desejado, também a "Honour indirect requests", que permite que os clientes se conectem usando o "X-indirect".



Na aba "**General**", opção "Greeter", mude a opção "Remote" de "Standard greeter" para "Graphical greeter". Dessa forma, os clientes farão login usando o gerenciador de login gráfico, que é especialmente bonito graficamente, ideal para passar uma boa impressão.

Para que as alterações entrem em vigor (tanto no KDM quanto no GDM), é necessário reiniciar o gerenciador de login. Para isso, mude para um terminal de texto (Ctrl+Alt+F2) e rode o comando **"/etc/init.d/kdm restart"** ou **"/etc/init.d/gdm restart"**. No Mandriva é usado o comando **"service dm restart"**.

A partir daí, os terminais já poderão abrir a tela de login do servidor através do comando **"X -query IP_do_servidor"**, como em:

X -query 192.168.0.1

O comando deve ser dado com o terminal em modo texto. Se o cliente já estiver com uma sessão do X aberta, ou você desejar abrir mais de uma tela do servidor ao mesmo tempo, basta adicionar o parâmetro ":2", como em:

X :2 -query 192.168.0.1

O comando abrirá um segundo terminal gráfico, independente do primeiro, exibindo a tela de login do servidor. Você pode alternar entre os dois usando as teclas Ctrl+Alt+F7 e Ctrl+Alt+F8. Para abrir mais terminais, basta substituir o ":2" por um número de 3 em diante.

Para automatizar o processo, fazendo com que o terminal abra automaticamente a tela de login do servidor no final do boot, sem passar pelo login local e sem a necessidade de digitar este comando a cada boot, edite o arquivo "/etc/inittab" (no terminal, como root) e altere a linha "**x:5:respawn:/etc/X11/prefdm -nodaemon**", que estará no final do arquivo para "x:5:respawn:/etc/X11/X -query IP_do_servidor", como em:

x:5:respawn:/etc/X11/X -query 192.168.0.1

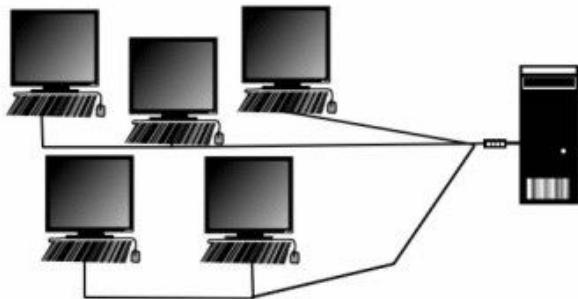
Uma segunda opção é utilizar o comando "**X -broadcast**" em substituição ao "X -query". A diferença é que enquanto o X -query exige que você especifique o endereço IP do servidor, o X -broadcast é automático, ele se encarrega de emitir um pacote de broadcast na rede e contatar o primeiro servidor X que responder ao chamado. O "**X -broadcast**" é sempre usado sem argumentos, como em:

X :2 -broadcast

Se você tiver mais de um servidor XDM na rede, uma terceira opção é usar o comando: "**X -indirect IP_do_servidor**". Nesse caso, você se conectará a um servidor X configurado, mas, ao invés de obter a tela de login automaticamente, terá um menu com todos os servidores X disponíveis na rede e poderá escolher qual usar a cada sessão. A partir daí o cliente escolhe a qual servidor deseja se conectar a cada boot.

» Próximo: [Capítulo 9: Terminais leves com o LTSP](#)

O **LTSP**, Linux Terminal Server Project, é uma solução mais usada para a criação de terminais leves com o Linux. Ele utiliza uma combinação de DHCP, TFTP, NFS e XDMCP para permitir que as estações não apenas rodem aplicativos instalados no servidor, mas realmente dêem boot via rede, baixando todos os softwares de que precisam diretamente do servidor. Não é preciso ter HD nem CD-ROM nas estações, apenas um disquete (ou CD) de boot ou ainda um chip de boot espetado na placa de rede.



O boot é dado com a ajuda do **Etherboot**, um software minúsculo que permite que as estações dêem boot através da rede, obtendo todo o software a partir do servidor. O software é muito pequeno, apenas 35 ou 40 KB (dependendo do driver usado pela placa de rede), e é lido apenas uma vez, no início do boot. Por isso, mesmo ao usar um disquete, o boot é rápido e não existem muitos problemas com desgaste do leitor ou erros de leitura na mídia.

A maior parte das placas de rede, mesmo as Encore de 15 reais, traz um soquete vago para o encaixe de um chip de boot. Os chips são relativamente baratos, de 10 a 20 reais em média, mas você ainda precisará gravá-las com o Etherboot. Em geral, o aplicativo de configuração que vem no disquete oferece a opção de gravar a ROM, algo parecido com quando você atualiza o BIOS da placa-mãe. Caso isso não seja possível, ainda existe a possibilidade de procurar alguém que tenha um gravador de EPROMs. Como a maior parte dos gravadores de BIOS também grava ROMs de placas de rede, isso não é um grande problema hoje em dia.

A maioria das placas-mãe novas, com rede onboard, suporta boot via rede utilizando o protocolo **PXE**. O LTSP pode ser configurado para suportar este protocolo, facilitando muito a configuração. Nesses casos, você vai precisar apenas configurar o cliente para dar boot através da rede no setup, sem se preocupar com disquetes ou chips de boot.

Ao utilizar o LTSP, você deve ter instalados no seu servidor os pacotes relacionados ao KDE (ou Gnome), além de aplicativos como o OpenOffice, Firefox e outros, que ficarão disponíveis aos usuários. Ou seja, você deve fazer uma instalação "Desktop" da distribuição escolhida, ao invés de seguir o modelo minimalista que utilizamos ao configurar um servidor web.

Ao utilizar o Debian, marque as categorias "Desktop", "Print server" (se você for utilizar impressoras) e "Standard System". Se você for utilizar o Ubuntu, utilize a imagem padrão de instalação, ao invés da versão server.

É preciso também manter o ambiente gráfico ativo no servidor (mesmo que ele for ficar trancado, sem monitor), pois o KDM ou GDM (os gerenciadores responsáveis por exibir a tela de login) são necessários para que os terminais obtenham a tela de login do servidor. Ou seja, você deve instalar o sistema no servidor como se ele fosse um desktop que vá ser utilizado por alguém.

De início, deixe o firewall completamente desativado, pois o LTSP utiliza um conjunto de vários serviços, o que torna complicado ir abrindo as portas necessárias durante a configuração. É mais simples começar com um servidor desprotegido e adicionar as camadas de segurança desejadas depois que o LTSP estiver funcionando.

Verifique as regras de firewall ativas usando o comando:

```
# iptables -L
```

Caso perceba que existem regras ativas, use o comando que limpa a tabela do iptables:

```
# iptables -F
```

» Próximo: [Entendendo o LTSP](#)

O LTSP é, na verdade, uma espécie de distribuição Linux destinada a ser carregada pelos terminais. Ele é composto por um conjunto de pacotes, que criam um sistema de arquivos dentro da pasta `/opt/Ltsp/i386/`, que é compartilhada com a rede e acessada via NFS pelos clientes como se fosse uma partição local.

Dentro do diretório vai um sistema simplificado, destinado apenas a detectar o hardware do cliente e permitir que ele abra uma sessão do X. Terminado o boot, o cliente obtém a tela de login do servidor via **XMCP**. A partir daí, o servidor roda os aplicativos e o cliente apenas mostra as imagens geradas na tela, atuando como uma espécie de terminal burro.

Veja que o LTSP é carregado nos clientes usando uma série de serviços. Tudo começa com o cliente dando boot usando a imagem de boot gravada no chip de boot, disquete ou CD-ROM. Essa imagem contém um software muito simples, que ativa a placa de rede e envia um pacote de broadcast, pedindo a configuração da rede.

Um servidor **DHCP** instalado no servidor LTSP é configurado para responder ao chamado, enviando a configuração da rede, juntamente com informações do Kernel, que o cliente deve carregar via TFTP, e a pasta no servidor com a instalação do LTSP, que deve ser acessada via NFS.

O **TFTP** é um protocolo bem simples de transferência de arquivos dentro de redes locais. Tão simples que a imagem de boot, com seus poucos kbytes, é grande o suficiente para incluir um cliente TFTP, usado na etapa inicial do boot.

Depois que o Kernel é carregado via TFTP, começa o boot "real" da estação. O TFTP é substituído, então, por um cliente **NFS** (um protocolo muito mais robusto), que é usado para montar a pasta `/opt/Ltsp/i386` do servidor (em modo somente leitura) como diretório

raiz. A estação pode então carregar o sistema do LTSP, que se encarrega de detectar o hardware da estação e abrir o X.

Todos os arquivos de configuração e alterações gerados nesta fase são salvos em um ramdisk, já que a estação não tem permissão para alterar os arquivos do servidor.

Opcionalmente, é possível especificar também uma configuração específica para cada estação, especificando o tipo de mouse e a resolução de tela, por exemplo. Esta configuração fica armazenada em um arquivo de configuração central, o "/opt/Ltsp/i386/etc/lts.conf" (armazenado no servidor), que é lido pelas estações durante o processo de boot.

Ou seja, além do LTSP propriamente dito, é necessário ter instalado um conjunto de serviços, cuidadosamente configurados, para que tudo funcione em conjunto. Isso faz com que o LTSP seja um sistema um pouco trabalhoso de instalar, onde é necessário prestar muita atenção em cada passo da configuração, já que qualquer erro pode fazer com que tudo deixe de funcionar.

O LTSP inclui dois utilitários de configuração: o **ltspadmin**, que automatiza partes da instalação e configuração inicial do sistema, e o **ltspcfg**, que é utilizado para alterar a configuração depois de instalado. Entretanto, como são desenvolvidas para trabalhar em conjunto com muitas distribuições, essas duas ferramentas nem sempre funcionam corretamente; por isso, seguindo a idéia inicial deste livro, vou explicar aqui como fazer uma instalação manual do LTSP, para que você entenda os componentes envolvidos e aprenda a solucionar problemas. Depois de instalar manualmente da primeira vez, experimente as duas ferramentas e veja até que ponto elas podem facilitar seu trabalho em futuras instalações.

» Próximo: [O servidor](#)

Em um servidor LTSP, os aplicativos usados por todos os clientes rodam no mesmo servidor, o que garante o compartilhamento de recursos. Por outro lado, em um desktop tradicional, o processador fica ocioso na grande maioria do tempo.

Você pode verificar isso no seu próprio micro, utilizando o comando "**top**". Ele mostra uma longa lista dos programas abertos e um instantâneo da utilização do processador perto do topo da tela, atualizado freqüentemente. No meu caso, mesmo com vários programas abertos e com um mp3 tocando, a utilização do processador fica na maior parte do tempo entre 3 e 6%, com picos rápidos para 10 ou 15%. Aqui, por exemplo, estão sendo usados 2.6% dos ciclos de processamento para os programas que estão abertos e mais 1.7% para tarefas relacionadas ao sistema:

```

xterm
top - 15:35:05 up 2 days, 6:44, 1 user, load average: 0.20, 0.22, 0.26
Tasks: 97 total, 1 running, 96 sleeping, 0 stopped, 0 zombie
Cpu(s): 2.6% us, 1.7% sy, 0.0% ni, 95.4% id, 0.0% wa, 0.0% hi, 0.3% si
Mem: 1034976k total, 1013816k used, 21160k free, 126304k buffers
Swap: 979924k total, 0k used, 979924k free, 457016k cached

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
2278 root 15 0 158m 89m 4380 S 2.3 8.8 56:08.22 Xorg
19852 kurumin 15 0 28408 14m 12m S 1.0 1.5 0:01.12 ksnapshot
18930 kurumin 15 0 56592 15m 6180 S 0.7 1.5 2:14.66 xmms
15369 kurumin 15 0 202m 102m 57m S 0.3 10.1 7:50.70 soffice.bin
  1 root 16 0 156 76 52 S 0.0 0.0 0:00.91 init
  2 root 34 19 0 0 0 S 0.0 0.0 0:00.01 ksoftirqd/0
  3 root 10 -5 0 0 0 S 0.0 0.0 0:09.48 events/0
  4 root 11 -5 0 0 0 S 0.0 0.0 0:00.03 khelper
  9 root 10 -5 0 0 0 S 0.0 0.0 0:00.00 kthread
  24 root 20 -5 0 0 0 S 0.0 0.0 0:00.00 kacpid
  139 root 10 -5 0 0 0 S 0.0 0.0 0:00.64 kblockd/0
  195 root 15 0 0 0 0 S 0.0 0.0 0:01.98 pdflush
  196 root 15 0 0 0 0 S 0.0 0.0 0:02.77 pdflush
  198 root 15 -5 0 0 0 S 0.0 0.0 0:00.00 aio/0

```

A menos que você passe a maior parte do tempo compilando programas, rodando games 3D ou editando vídeo, na maioria do tempo, a utilização do processador ficará sempre abaixo de 5 ou 10%.

É justamente isso que os processadores da maioria dos desktops do mundo ficam fazendo na maior parte do tempo: nada. Em um servidor de terminais, a utilização média do processador em geral continua sendo baixa, mesmo com 20 ou 30 terminais pendurados nele, já que temos uma máquina relativamente rápida e muitos dos usuários conectados ficam fazendo tarefas simples, como ler e-mails ou escrever textos.

Isso faz com que quase sempre que um usuário precisa executar um programa, ou realizar uma tarefa intensiva, encontra o processador livre, como se ele estivesse sozinho no servidor. O desempenho (subjetivo) ao utilizar um terminal ligado a um servidor com um processador de 3.0 GHz, compartilhado entre 20 terminais, é quase sempre melhor que utilizar um desktop com um processador (de desempenho por clock similar) de 1.5 GHz.

A memória RAM também é compartilhada de uma maneira bastante interessante. Os aplicativos são carregados na memória do servidor apenas uma vez, independentemente do número de usuários que o utilizarem simultaneamente. O sistema carrega o aplicativo uma vez, e depois passa a abrir diferentes sessões do mesmo programa (como ao abrir uma segunda janela do navegador, por exemplo), o que faz com que o carregamento passe a ser mais rápido (afinal, o aplicativo já está carregado) e o uso de memória seja otimizado.

Um servidor com 1 GB de memória RAM, dividido entre 20 terminais, executa, em geral, os aplicativos com um desempenho muito melhor que um desktop com 256 MB usado por um único usuário.

A configuração mínima para atender a 10 terminais seria um Pentium III ou Athlon com 512 MB de RAM. Mas, como o servidor é um só, é recomendável investir um pouco nele,

principalmente hoje em dia, quando os preços dos pentes de memória estão cada vez mais baixos. O ideal é começar com um processador razoavelmente rápido e 1 GB de RAM.

Monitore a utilização do processador e a memória RAM livre durante algum tempo. Conforme for necessário, você pode adicionar mais 1 GB de RAM ou um processador dual core. Um servidor dual oferece uma grande vantagem ao utilizar muitos terminais, pois ele pode executar aplicativos separados em cada processador, executando mais tarefas simultaneamente e eliminando o gargalo em momentos em que vários usuários resolvem utilizar aplicativos pesados simultaneamente.

Ao contrário de um desktop regular, no qual em geral apenas um aplicativo pesado é executado por vez, fazendo com que o segundo processador seja pouco usado, um servidor de terminais está sempre executando muitos aplicativos diferentes e fazendo muitas coisas ao mesmo tempo. Isso faz com que realmente exista uma divisão de trabalho entre os dois processadores, fazendo com que o desempenho ao utilizar dois processadores seja, em muitos casos, próximo do dobro de utilizar apenas um.

A terceira característica mais importante, rivalizando com o desempenho do processador, é o desempenho e capacidade dos HDs. Lembre-se de que o servidor será quem armazenará todos os arquivos, por isso é importante que o HD tenha muito espaço livre. Um sistema RAID IDE (seja usando uma controladora dedicada, seja via software) é uma opção interessante, pois permite combinar vários HDs de forma a criar um único disco lógico com a capacidade e desempenho somados (RAID 0). Isso acaba sendo muito mais interessante do que simplesmente adicionar vários HDs separados.

O desempenho do RAID fará com que os aplicativos carreguem mais rapidamente e as operações de cópias de arquivo sejam concluídas muito mais depressa, evitando a saturação do servidor em momentos de pico. Em contrapartida, ao usar RAID 0, o risco de perda de dados é maior do que ao utilizar um único HD, pois uma pena de hardware em qualquer um dos discos faz com que todos os dados sejam perdidos. Por isso, um sistema de backup é essencial.

As dicas que dei até aqui são voltadas a redes de grande porte, com 20 a 50 terminais. Em uma rede pequena, com de 4 ou 6 terminais, você pode começar instalando o LTSP em um desktop comum, com 512 MB de RAM, e pensar em atualizar o servidor apenas se notar problemas de estabilidade, ou caso precise adicionar mais terminais.

Dessa maneira, o servidor não precisa sequer ser dedicado. Nada impede que você o utilize junto com os terminais, apenas tome o cuidado de não ficar apertando o botão de reset nem ficar dando tapas na CPU... :) Outra configuração importante é desabilitar a opção de desligamento local e remoto no Centro de Controle do KDE > Administração do Sistema > Gerenciador de Login > Desligar > Permitir desligamento (acesse como root).

Fazendo isso, nenhum usuário vai conseguir desligar o servidor por engano. Lembre-se: "quem tem HD, tem medo", como as estações não têm HD, então não existe necessidade de "desligar o sistema corretamente", é só dar um logout e depois desligar a estação no botão. Apenas o servidor precisa passar pelo processo normal de desligamento.

» Próximo: [Os terminais](#)

A configuração mínima para os terminais é um 486 com 8 MB, e a configuração ideal é um Pentium 100 com 32 MB. Em teoria, você pode utilizar até mesmo um 386 como terminal, mas, nesse caso, você vai começar a sentir uma certa demora na atualização da tela.

O servidor fica com o grosso do trabalho, que é executar os programas e armazenar todos os dados. Ele envia para os clientes apenas instruções para montar as janelas que serão exibidas, e estes enviam de volta os movimentos do mouse e as teclas digitadas no teclado.

O ping numa rede local, mesmo que seja uma rede de 10 megabits, é muito baixo, em torno de 10 milissegundos, na pior das hipóteses. Isso significa que o tempo necessário para um click do mouse ir da estação até o servidor e este enviar de volta a resposta é mínimo, quase imperceptível. Apesar disso, a estação precisa rodar uma versão compacta do Linux com um servidor X e tem o trabalho de montar as janelas baseado nas instruções recebidas do servidor, daí a necessidade de um mínimo de poder de processamento.

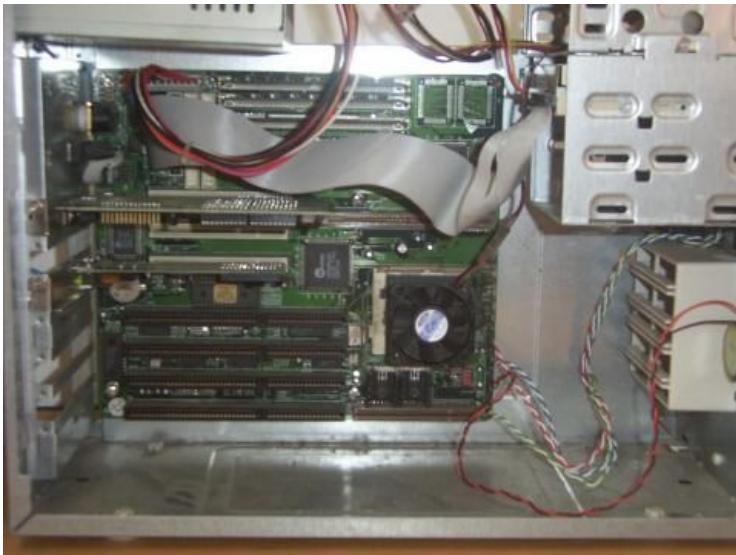
Se o processador for muito lento, a estação tem dificuldades para fazer a atualização de tela e as respostas começam a ficar muito lentas. Um 486 DX-100 demora cerca de 0.5 segundo para redimensionar uma janela (usando o X.org padrão do LTSP), o que é ainda relativamente rápido. No entanto, um 386 demoraria 2 ou 3 segundos para fazer a mesma tarefa, o que já seria bastante incômodo.

O ideal é utilizar, no mínimo, micros 486 DX-100 com uma placa de vídeo PCI. Se você utilizar micros um pouco mais rápidos, a partir de um Pentium 100, a atualização de tela já passará a ser instantânea, como se os aplicativos estivessem rodando localmente.

O próximo gargalo é a velocidade da rede, que precisa transportar as informações destinadas a todas as estações. Graças à eficiência do protocolo utilizado pelo X (se comparado a sistemas como o VNC), é possível pendurar 20 ou até mesmo 30 terminais em uma rede de 100 megabits, antes que a velocidade da rede comece a se tornar um gargalo.

Em redes maiores, ou ao usar equipamentos novos, existe a opção de investir em uma rede gigabit, o que evita a saturação da rede nos momentos de pico e permite usar um número ainda maior de terminais.

Aqui estão as entranhas de um dos micros que uso nos meus testes. Ele é um 486 de 133 MHz da AMD com 12 MB de RAM. Ele tem um desempenho mais parecido com um DX-80, pois usa uma placa-mãe sem cache L2. Como você pode ver, ele tem espelhos apenas a placa de rede PCI, uma placa de vídeo Trident 9440 de 1 MB e um drive de disquetes.



Aqui ele já está rodando o KDE a partir de um Celeron 700 com 256 MB que estou usando como servidor para três clientes, a fim de testar o desempenho ao usar um servidor lento. Como disse, ao usar poucos clientes, o servidor não precisa ser nenhum topo de linha.



O boot é bem rápido, demora menos de 30 segundos (no 486) para cair na tela de login do servidor e, a partir daí, o tempo de carregamento do KDE e dos programas depende apenas do desempenho deste. Se você usar como servidor um Athlon X2, com HDs em RAID e muita RAM, por exemplo, todos os clientes terão a impressão de estarem usando uma super máquina que abre qualquer coisa quase instantaneamente, mesmo que na verdade estejam usando um monte de 486 velhos. Essa é justamente a parte mais interessante: você pode continuar usando os micros que tem em mãos e mesmo assim obter um bom desempenho, investindo apenas em um servidor mais rápido.

» Próximo: [Usando os terminais](#)

Tenha em mente que, ao utilizar o LTSP, todos os aplicativos rodam no servidor e, por isso, os arquivos gerados também são salvos nele. Por isso, o ideal é criar uma conta de usuário para cada usuário do sistema, de modo que ele possa salvar seus arquivos e configurações, sem ser perturbado pelos demais usuários do sistema.

É importante que cada usuário tenha permissão de acesso apenas ao seu próprio diretório home, sem ter como xeretar nos arquivos nos demais. Para isso, depois de adicionar cada usuário, use o comando "chmod -R o-rwx", que retira todas as permissões de acesso para os demais usuários, deixando apenas o próprio usuário.

chmod -R go-rwx /home/usuario

Para modificar de uma vez as permissões do diretório home de todos os usuários do sistema, use:

chmod -R go-rwx /home/*

Para fazer com que todos os arquivos criados daí em diante, por todos os usuários, já fiquem com as permissões corretas, adicione a linha "**umask 077**" no final do arquivo "**/etc/bash.bashrc**".

O comando umask configura as permissões default para novos arquivos, subtraindo as permissões indicadas. Se o umask é "077", significa que são retiradas todas as permissões de acesso para todos os usuários, com exceção do dono do arquivo. Na maioria das distribuições, o padrão é "022", que retira apenas a permissão de escrita.

Outra questão interessante ao usar terminais LTSP são os backups, que se tornam bem mais simples, já que os arquivos ficam centralizados no servidor. Você pode ter, por exemplo, um segundo HD e uma gaveta para fazer o backup sempre que necessário e guardá-lo em um local seguro. Uma dica importante é sempre usar um sistema com suporte a journaling no servidor, como o ReiserFS (preferencialmente) ou o EXT3. Ambos são muito mais seguros que o antigo EXT2, que é muito suscetível à perda de dados depois de desligamentos incorretos.

A manutenção do servidor pode ser feita a partir de qualquer terminal, ou até mesmo via internet (se você configurar o firewall para liberar o acesso via SSH). Se precisar instalar novos programas, basta instalá-los no servidor.

Os problemas com vírus e cavalos de Tróia são muito menores no Linux. Um programa executado pelo usuário não tem mais permissões do que ele mesmo, ou seja, se um usuário não tem permissão para alterar arquivos fora da sua pasta, qualquer programa executado por ele também não terá. Na pior das hipóteses, ele pode acabar com seus próprios arquivos pessoais, mas não afetará os arquivos dos demais usuários ou as configurações do sistema.

Nas estações, a única preocupação é com problemas de hardware, que provavelmente serão relativamente freqüentes, já que estamos falando de máquinas com, em muitos casos, 6, 8 ou até 10 anos de uso. Mas pelo menos você não precisará se preocupar com perda de dados, já que estará tudo no servidor. Se possível, mantenha uma ou duas torres já montadas de reserva, assim você poderá trocar rapidamente qualquer terminal com problemas de hardware.

Existem naturalmente algumas limitações no uso dos terminais, como os jogos, por exemplo. Jogos de cartas, ou de tabuleiro, ou até mesmo títulos como o Freeciv (um clone do Civilization 2), onde existe pouca movimentação, rodam sem problemas, mas jogos de movimentação rápida, em tela cheia, não vão rodar satisfatoriamente. Naturalmente, os terminais leves também não são um ambiente adequado para rodar jogos 3D.

O CD-ROM e o drive de disquetes do servidor poderão ser usados normalmente pelos usuários, inclusive com vários usuários acessando o CD que está na bandeja, por exemplo. Você pode também criar imagens dos CDs usados, utilizando o comando dd, e montar estas imagens como pastas do sistema (mantendo assim vários CDs disponíveis simultaneamente) através do comando "mount -o loop nome_do_cd.iso /mnt/nome_do_cd" como em:

```
# dd if=/dev/cdrom of=cdrom1.iso  
# mount -o loop cdrom1.iso /mnt/cdrom1
```

No LTSP 4.2 ficou bem mais fácil de ativar o suporte a dispositivos locais, permitindo usar também CD-ROM, pendrives e outros dispositivos de armazenamento instalados nos terminais. Ao plugar um pendrive, um ícone aparece no desktop permitindo acessar os arquivos. Com o suporte a dispositivos locais ativado, os terminais podem ser usados de forma transparente, como se cada um fosse um desktop completo.

A princípio, pode parecer que rodar aplicativos de 10 clientes no servidor ao mesmo tempo irá deixá-lo bastante lento, mas na prática isso funciona da mesma forma que as linhas dos provedores de acesso discados. Nenhum provedor tem o mesmo número de linhas e de assinantes, geralmente utilizam uma proporção de 10 ou 20 para um, presumindo que jamais todos os assinantes vão resolver conectar ao mesmo tempo.

Mesmo com 10 clientes, raramente todos vão resolver rodar ao mesmo tempo algo que consuma todos os recursos do servidor por um longo período. Normalmente temos apenas tarefas rápidas, como abrir um programa, carregar uma página web, etc., feitas de forma intercalada. Ou seja, na maior parte do tempo, os clientes vão se sentir como se estivessem sozinhos no servidor.

Em um ambiente típico, um servidor de configuração razoável, com 1 GB de RAM, pode manter de 20 a 30 terminais, enquanto um servidor dual, com 2 GB de RAM e HDs em RAID, pode atender a 50 terminais. Em ambientes em que as estações rodem alguns poucos aplicativos específicos (um navegador e um aplicativo de gerenciamento, por exemplo), é possível manter 40 ou 50 terminais com um servidor de configuração mais modesta.

Outro ponto interessante diz respeito às suas estratégias de upgrade. Ao invés de gastar dinheiro com upgrades de memória e processador para os clientes, você deve investir os recursos disponíveis em melhorar o servidor e a rede, além de trocar monitores, teclados e mouses nas estações. Um monitor de 17" e um teclado novo em algumas das estações vão fazer muito mais efeito que um upgrade na torre.

» Próximo: [Instalando os serviços-base](#)

Antes de começar a instalação do LTSP propriamente dita, é importante instalar e testar os serviços-base utilizados por ele. Ao longo do livro já aprendemos a trabalhar com a maioria deles, então vou fazer apenas uma revisão rápida:

Os serviços usados pelo LTSP, que precisam estar instalados e ativos no servidor são:

```
tftpd  
dhcp3-server  
portmap  
nfs-kernel-server  
xdmcp
```

O "**tftpd**" é o servidor TFTP, um pacote bem pequeno utilizado para transferir o Kernel usado pelas estações. Instale o pacote "tftpd" através do gerenciador de pacotes da distribuição usada. Em algumas distribuições o tftpd roda como um serviço, controlado pelo script "/etc/init.d/tftpd" e, em outras roda através do inetd. Não se preocupe com a configuração do tftpd por enquanto, pois vamos revisar sua configuração mais adiante.

O segundo pacote necessário é o servidor **DHCP**, que aprendemos a configurar no capítulo sobre compartilhamento da conexão. Por enquanto, verifique apenas se ele está instalado e ativo. Vamos revisar a configuração mais adiante. Lembre-se de que no Debian o pacote do servidor DHCP se chama "dhcp3-server" e que, em outras distribuições, ele chama-se apenas "dhcp" ou "dhcpcd".

O LTSP utiliza o servidor **NFS** para compartilhar a pasta "/opt/ltsp/i386/", que é montada como o diretório raiz (/) pelos terminais na hora do boot. Instale o **Portmap** e o servidor **NFS**, como vimos no capítulo sobre servidores de arquivos, e crie um compartilhamento de teste para verificar se ele está mesmo funcionando.

Finalmente, precisamos habilitar o **XDMCP**, que permitirá que os terminais obtenham a tela de login do servidor e executem os aplicativos remotamente. O XDMCP é o componente mais importante ao instalar o LTSP. Habilite o compartilhamento editando os arquivos "/etc/kde3/kdm/kdmrc" e "/etc/kde3/kdm/Xaccess" como vimos anteriormente, reinicie o gerenciador de login (no servidor) e verifique se consegue acessá-lo a partir de outro micro da rede usando o comando "X :2 -query 192.168.0.1".

Com tudo funcionando, podemos passar para a instalação do LTSP propriamente dito.

» Próximo: [Instalando os pacotes do LTSP](#)

Até o LTSP 3.0, estavam disponíveis para download um conjunto de pacotes para várias distribuições, incluindo pacotes .rpm para o Fedora, Mandrake, etc., pacotes .deb para as distribuições derivadas do Debian e pacotes .tgz para o Slackware.

Estes pacotes ainda estão disponíveis. Mas, a partir do LTSP 4.0, foi desenvolvido um sistema unificado de instalação, onde você baixa o instalador e ele se encarrega de baixar os pacotes e fazer a instalação.

Enquanto escrevo este texto, a versão mais recente é a **4.2**. A estrutura do LTSP não muda muito de uma versão a outra; por isso, mesmo que ao ler você esteja instalando uma versão mais recente, é provável que os passos que descrevi aqui funcionem sem modificações.

Comece baixando o pacote "**ltsp-utils**", disponível no: <http://ltsp.mirrors.tds.net/pub/ltsp/utils/>.

Existem pacotes para várias distribuições. Ao instalar em distribuições derivadas do Debian, baixe o arquivo "ltsp-utils_0.25_all.deb" e instale-o usando o dpkg:

```
# dpkg -i ltsp-utils_0.25_all.deb
```

Depois de instalado, chame o comando "**ltspadmin**" para abrir o instalador. Ele é escrito em Perl e precisa que o pacote "**libwww-perl**" esteja instalado. Caso ele retorne um erro, procure-o no gerenciador de pacotes da distribuição usada.

```
# apt-get install libwww-perl
# ltspadmin
```

Ao abrir o programa, selecione a segunda opção, "**Configure the installer options**". Aqui vamos definir a localização dos pacotes e quais serão instalados.

Por padrão, o instalador se oferece para baixar os pacotes via web. Esse modo de instalação baixa pouco mais de 100 MB de pacotes, não é problema se você tem banda larga. Nesse caso, basta manter o valor default quando ele pergunta "Where to retrieve packages from?".

Se, por outro lado, você prefere baixar todos os pacotes antes da instalação, pode baixar uma imagem ISO, contendo todos os pacotes no: <http://ltsp.mirrors.tds.net/pub/ltsp/isos/>

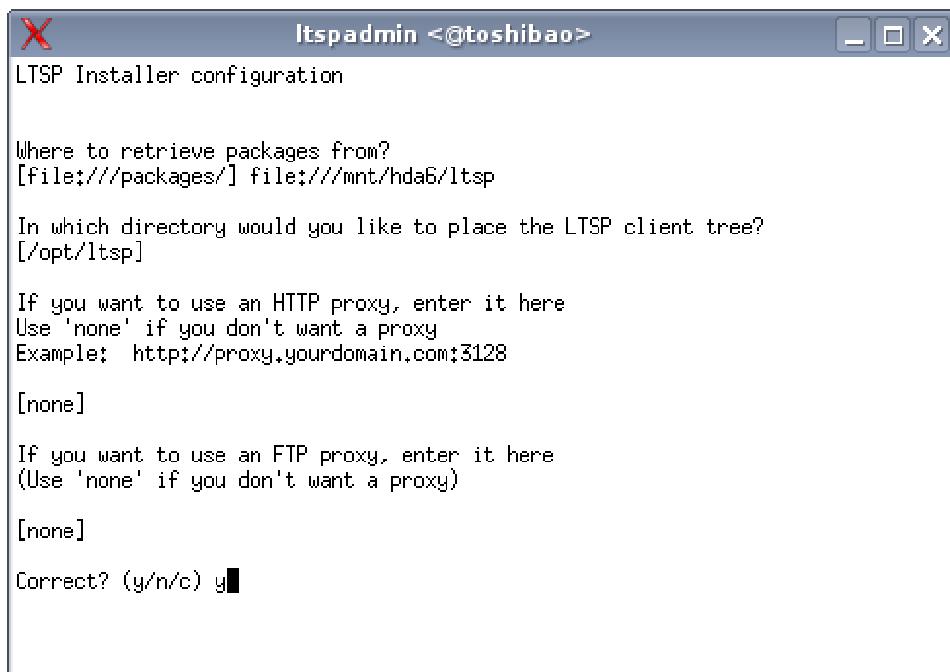
Depois de baixar o ISO, você pode montá-lo diretamente numa pasta, assim você não tem o trabalho de queimar o CD. Para montar, use o comando abaixo, incluindo a pasta (vazia) onde a imagem será montada:

```
# mount -o loop ltsp-4.2u2-0.iso /mnt/hda6/ltsp
```

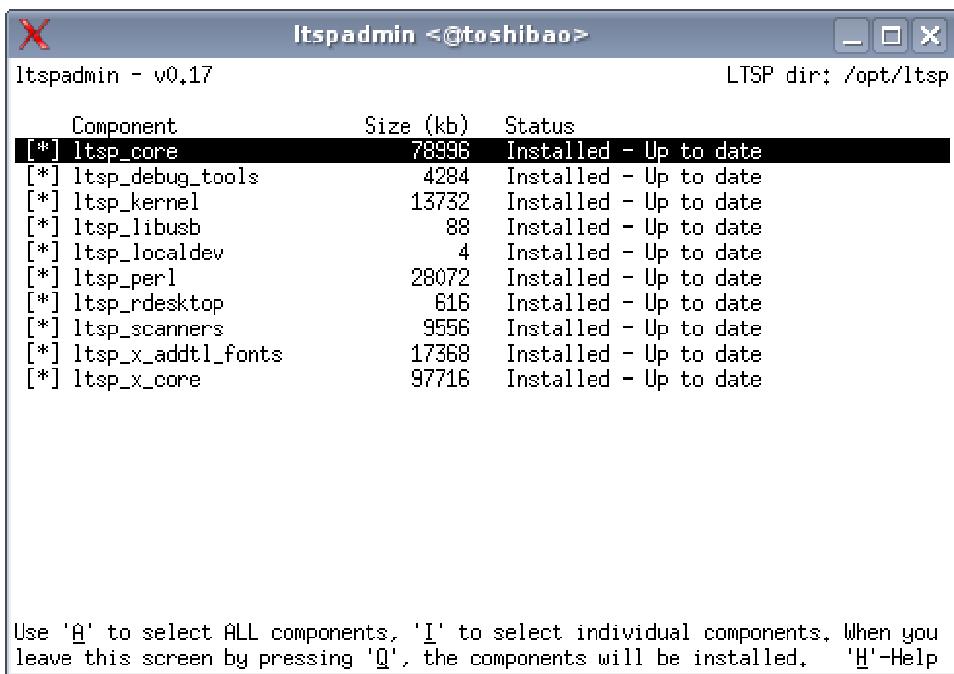
Para que o instalador use os pacotes do ISO, informe o diretório quanto ele pergunta "Where to retrieve packages from?". Se você o montou no diretório "/mnt/hda6/ltsp", por exemplo, responda "file:///mnt/hda6/ltsp"; se você gravou o CD e ele está montado na pasta "/mnt/cdrom", responda "/mnt/cdrom" e assim por diante.

Note que a localização contém três barras, pois é a junção de "file://" e a pasta onde o ISO ou CD-ROM está acessível (/mnt/hda6/ltsp, no meu exemplo). Mantenha o diretório de instalação como "**/opt/Ltsp**".

O instalador se oferece para configurar um proxy, mas isso só é necessário ao instalar via web e, mesmo assim, apenas se você realmente acessa via proxy. No final responda "y" para salvar a configuração.

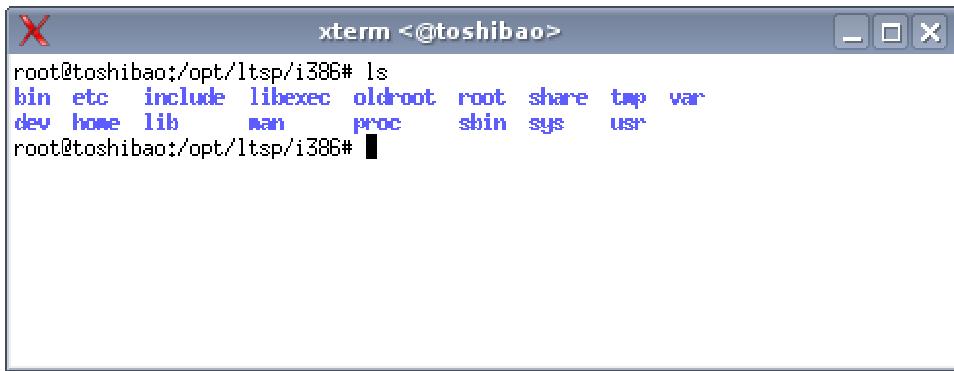


De volta à tela inicial, escolha agora a primeira opção "**Install/Update LTSP Packages**". Você cairá na tela de seleção dos pacotes a instalar. Pressione a tecla "A" para marcar todos os pacotes e "Q" para iniciar a instalação.



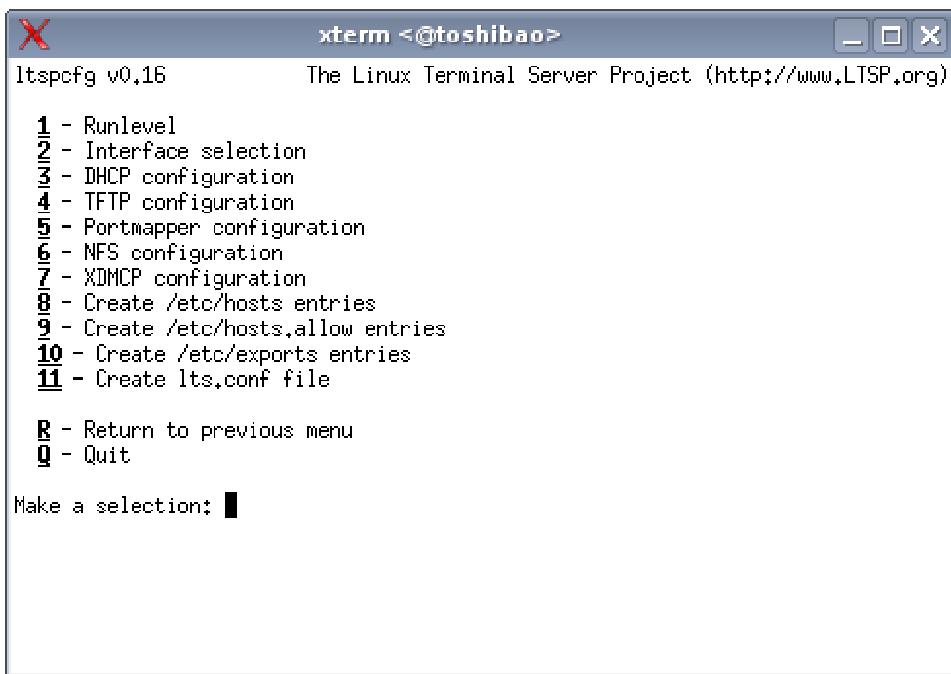
Note que esta tela se parece um pouco com o programa de instalação de algumas distribuições antigas. Como disse, o LTSP é na verdade uma distribuição Linux, que é instalada em uma pasta do servidor (sem interferir com o sistema principal). A pasta é compartilhada com os clientes via NFS e o sistema é carregado por eles durante o boot.

Depois de concluída a instalação, acesse a pasta "**/opt/Ltsp/i386**". Você verá que ela foi "populada" com o conjunto de pastas padrão encontrado em uma distribuição Linux.



A instalação do LTSP é bastante simples, principalmente ao utilizar o novo instalador. A parte mais complicada é a configuração dos serviços-base e a configuração das estações, que vai no arquivo "**/opt/Ltsp/i386/etc/lts.conf**", que veremos daqui em diante.

Concluída a instalação, você tem a opção de utilizar a terceira opção do instalador: "**Configure LTSP**", que verifica se os serviços necessários estão habilitados e ajuda dando a opção de criar um arquivo de configuração padrão para cada um. De qualquer forma, a maior parte da configuração precisa ser feita manualmente, ele só ajuda criando os modelos e verificando alguns problemas comuns.



Para facilitar a configuração, escrevi um conjunto de arquivos de configuração, incluindo exemplos e comentários auto-explicativos para uso no script do Kurumin-terminal-server. Estes arquivos de configuração estão disponíveis no:

<http://www.guiadohardware.net/kurumin/modelos/kurumin-terminal-server/4.2/>

Na pasta você encontra os arquivos **dhcpd.conf** (a configuração do servidor DHCP), **exports**, (a configuração do servidor NFS), **hosts**, (onde vão os endereços IP e nomes das estações), **hosts.allow** (permissões de acesso), **inetd.conf** (a configuração do inetd, responsável por carregar o tftpd) e o arquivo **lts.conf** (onde vai a configuração de cada estação).

Recomendo que comece utilizando estes arquivos como modelo para a sua configuração. Eles são auto-explicativos e já incluem uma configuração semifuncional, com 8 estações pré-configuradas.

Para instalá-los no Debian, baixe todos para uma pasta local e use os comandos abaixo para copiá-los para os diretórios apropriados:

```

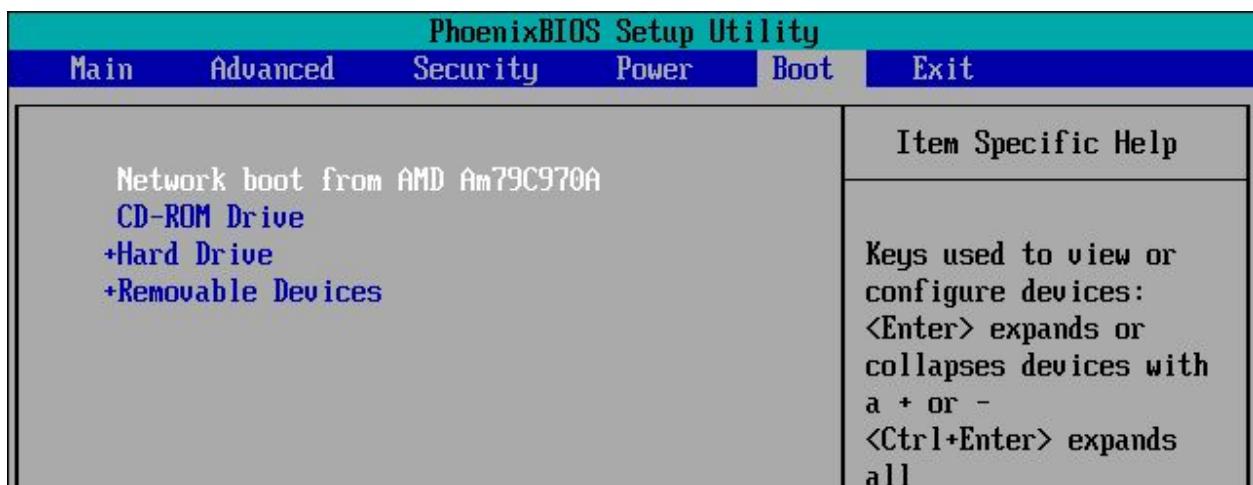
# cp --reply=yes dhcpd.conf /etc/dhcp3/dhcpd.conf
# cp --reply=yes exports /etc(exports
# cp --reply=yes hosts.allow /etc/hosts.allow
# cp --reply=yes lts.conf /opt/ltsp/i386/etc/lts.conf
# cp --reply=yes hosts /etc/hosts

```

» Próximo: [Configurando os terminais](#)

O primeiro passo é configurar os clientes para darem boot via rede, seja via PXE, seja usando o Etherboot.

O **PXE** é um protocolo de boot criado pela Intel, que é suportado pela grande maioria das placas mãe com rede onboard. Para dar boot via rede, acesse o Setup e, dentro da seção com a configuração de boot, deixe a opção "Network Boot" em primeiro na lista, antes das opções, para dar boot pelo HD ou CD-ROM. Na maioria das placas é também possível dar boot via rede (independentemente da configuração do Setup), pressionando a tecla F12 durante o boot.



Dar boot diretamente pela placa de rede oferece uma praticidade muito grande, pois você precisa apenas mudar uma opção no Setup ou pressionar uma tecla durante o boot ao invés de ter que manter um drive de disquete em cada micro ou sair atrás de alguém que venda ROMs para as placas de rede.

Muita gente prefere montar terminais usando placas novas, já que micros novos dão menos problemas de hardware e podem ter uma configuração padronizada. Lembre-se de que os clientes não precisam ter HD nem CD-ROM e podem ser montados usando processadores baratos, já que performance não é um problema. Em muitos casos, placas-mãe com problemas de estabilidade e pentes de memória com endereços queimados, que não podem ser utilizados em outras situações, funcionam perfeitamente como terminais, já que os requisitos, nesse caso, são muito mais baixos.

Outro uso comum é deixar o sistema de boot via rede como um sistema operacional "de reserva", para ser usado em casos de problemas com o sistema principal, instalado no HD. Isso permite que o usuário continue trabalhando até que o problema seja resolvido.

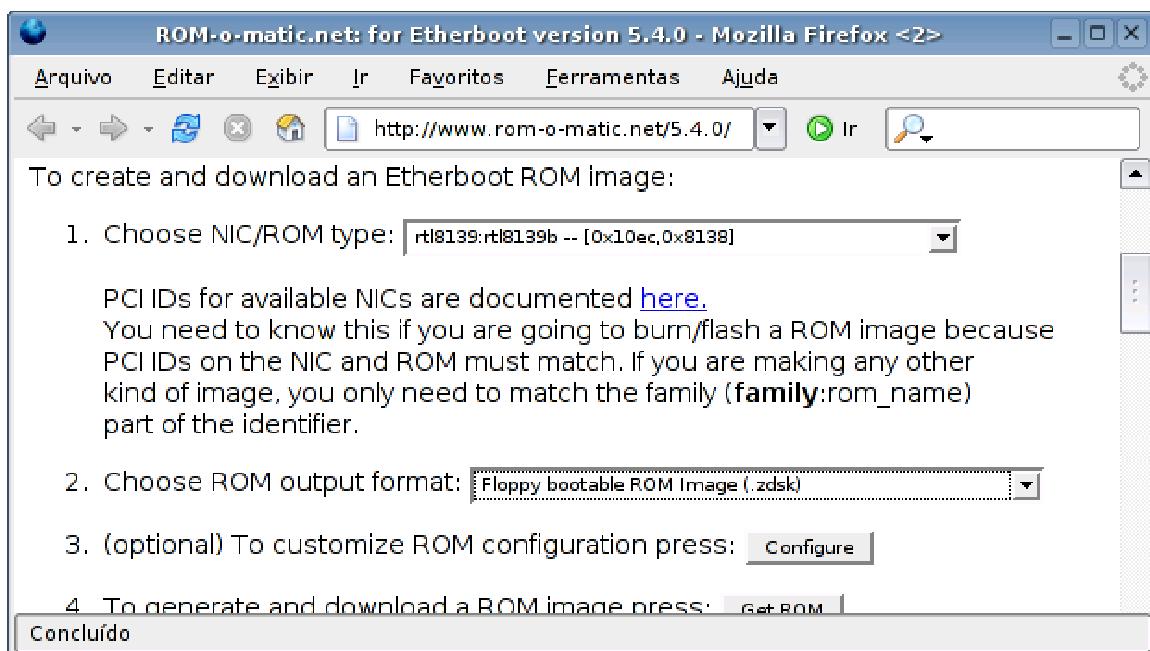
Mais uma aplicação interessante é em treinamentos de Linux em geral. Você pode utilizar os próprios micros de um escritório ou laboratório de informática, trazendo um servidor

LTSP pré-configurado e dando boot via rede nos clientes, sem necessidade de sair instalando Linux em todas as máquinas.

No caso de micros antigos, ou placas que não suportam boot via rede, você pode usar a segunda opção, que é criar os discos de boot do **Etherboot**, através da página do rom-o-matic, onde você indica o modelo da placa de rede e o formato desejado e ele lhe devolve a imagem de boot a usar. Acesse o site no <http://www.rom-o-matic.org/>.

No primeiro campo, indique o modelo da placa de rede e, no segundo, indique o tipo de imagem que será gerada. Estão disponíveis módulos para várias placas de rede, incluindo as 3com, Intel, sis900 (usada em muitas placas onboard) e via-rhine-6105, usado nas placas Encore novas. As antigas placas com chipset Realtek 8139 trabalham em conjunto com o módulo rtl8139.

Use a opção "Floppy Bootable ROM Image (.zdisk)" para gerar a imagem de um disquete de boot ou a opção "ISO bootable image with legacy floppy emulation (.iso)" para gerar um CD de boot. Nesse caso, renomeie o arquivo gerado, de ".iso" para ".iso".



Se estiver usando uma máquina virtual do VMware como cliente para testar, escolha a opção "pcnet32:lancepc" como placa de rede e gere um CD de boot. Em caso de dúvidas sobre os módulos usados pelas placas que tem em mãos, você pode consultar esta tabela: <http://www.etherboot.org/db/>.

Para gravar um CD de boot, basta gravar a imagem da forma como gravaria um arquivo .iso tradicional. Depois, configure o cliente para dar boot através do CD-ROM. Intencionalmente, escolhemos a opção "with legacy floppy emulation", que torna a imagem compatível com os BIOS usados em micros antigos.

Para gravar os disquetes, use o comando:

```
$ dd if=eb-5.0.10-rtl8139.lzdsk of=/dev/fd0  
(onde o eb-5.0.10-rtl8139.lzdsk é o nome do arquivo)
```

Em seguida, dê um boot em cada estação, seja via PXE ou usando o disquete ou CD de boot e anote o número do endereço MAC de cada placa de rede, que é mostrado no início do boot. Você precisará fornecer os endereços MAC de cada estação nos arquivos de configuração do LTSP que editaremos em seguida.

O endereço MAC é um número de 12 dígitos (como: 00:0C:29:6F:F4:AB), diferente em cada placa, que você pode localizar facilmente entre as mensagens exibidas. Ele aparece na penúltima linha (lancepc: 00:0C:29:6F:F4:AB).

```
SYSLINUX 2.08 2003-12-12 Copyright (C) 1994-2003 H. Peter Anvin  
Etherboot ISO boot image generated by genliso  
Loading pcnet32.zli....Ready.  
Etherboot 5.4.0 (GPL) http://etherboot.org  
Drivers: PCNET32/PCI Images: NBI ELF PXE Exports: PXE  
Protocols: DHCP TFTP  
Relocating _text from: [00010220,00024370) to [0fedbeb0,0fef0000)  
Boot from (N)etwork or (Q)uit?  
  
Probing pci nic...  
[lancepc]pcnet32.c: Found lancepc, Vendor=0x1022 Device=0x2000  
lancepc: 00:0C:29:6F:F4:AB at ioaddr 1080, No MII transceiver found!  
  
Searching for server (DHCP)....
```

Enquanto o servidor não estiver configurado, os terminais vão ficar indefinidamente no "Searching for DHCP Server...", um sintoma de que o boot via rede está ativo. Agora falta apenas o principal: o servidor :).

» Próximo: [Configurando o servidor](#)

Agora vem a parte mais complicada, que é a configuração do servidor propriamente dita, feita em seis arquivos separados. Os arquivos-padrão que criei vêm com entradas para cinco clientes usando o Etherboot e mais três usando o PXE, mas você pode adicionar mais entradas caso necessário. Lembre-se de que em qualquer arquivo de configuração as linhas começadas por um "#" são comentários que não possuem efeito algum.

» Próximo: [DHCP](#)

O servidor DHCP é o primeiro a ser acessado pela estação. Ela "acorda" sem saber quem é, e o DHCP responde entregando as configurações da rede e dizendo qual Kernel ou cliente PXE a estação deve carregar e em qual compartilhamento de rede (no servidor) onde está o sistema a ser carregado por ela. Antes de mais nada, verifique se o pacote "**dhcp3-server**" está instalado:

```
# apt-get install dhcp3-server
```

Em seguida, vamos à configuração do arquivo "**/etc/dhcp3/dhcpd.conf**", onde vai a configuração do servidor DHCP.

A configuração do DHCP para o LTSP é mais complexa do que vimos no capítulo sobre compartilhamento da conexão, por isso é importante prestar atenção. O arquivo é dividido em duas sessões, a primeira é a "shared-network WORKSTATIONS", onde vão as configurações gerais do servidor, enquanto a sessão "group" contém a configuração de cada estação.

O arquivo possui uma formatação bastante estrita, onde cada linha de configuração deve terminar com um ";" e cada sessão começa com um "{" e termina com um "}". Se o servidor DHCP se recusa a iniciar com o comando "/etc/init.d/dhcp3-server restart", reportando um erro no arquivo de configuração, provavelmente você esqueceu algum ponto e vírgula ou esqueceu de fechar alguma sessão. Como em outros arquivos, você pode usar tabs, espaços e quebras de linha para organizar o arquivo da forma que achar melhor.

Este é um exemplo de configuração funcional. Lembre-se de que você pode baixar os modelos comentados no: <http://www.guiadohardware.net/kurumin/modelos/kurumin-terminal-server/4.2/>

```
shared-network WORKSTATIONS {  
    subnet 192.168.0.0 netmask 255.255.255.0 {  
        default-lease-time 21600;  
        max-lease-time 21600;  
        option subnet-mask 255.255.255.0;  
        option broadcast-address 192.168.0.255;  
        option routers 192.168.0.1;  
        option domain-name-servers 192.168.0.1;  
  
        deny unknown-clients;  
        # range 192.168.0.100 192.168.0.201;  
  
        option root-path "192.168.0.10:/opt/ltsp/i386";  
        next-server 192.168.0.10;  
    }  
}  
  
group {  
    use-host-decl-names on;
```

```

#                                     terminal          1:
host                               ws001           {
hardware                           ethernet        00:E0:7D:B2:E5:83;
fixed-address                      filename       192.168.0.11;
filename                           }             "lts/2.6.17.3-ltsp-1/pixelinux.0";

#                                     terminal          2:
host                               ws002           {
hardware                           ethernet        00:D0:09:A2:9B:8D;
fixed-address                      filename       192.168.0.12;
filename                           }             "lts/2.6.17.3-ltsp-1/pixelinux.0";

}

```

Na primeira parte do arquivo, você deve fornecer as configurações da rede, como a máscara de sub-rede, o endereço do default gateway e o DNS do provedor. O default dos arquivos de configuração que criei é usar a faixa de IP's 192.168.0.x, onde o servidor de terminais é configurado para usar o endereço 192.168.0.10.

Se você preferir usar esses endereços, seu trabalho será bem menor. Caso contrário, preste atenção para substituir todas as referências ao servidor (192.168.0.10) pelo endereço IP correto, modificando também os endereços dos terminais (192.168.0.11 a 192.168.0.18) por endereços dentro da mesma faixa de endereços usada pelo servidor.

Logo abaixo vem a opção onde você deve fornecer o endereço IP usado pelo servidor LTSP. Ela diz que o cliente deve usar a pasta "/opt/ltspl386" do servidor "192.168.0.10" como diretório raiz.

Note que o "**/opt/ltspl386**" representa a pasta de instalação do LTSP, que é montada pelos clientes como diretório-raiz durante o boot. Não se esqueça de verificar e alterar esta configuração se tiver instalado o LTSP em outra pasta ou estiver utilizando outro endereço IP no servidor.

A opção "deny unknown-clients" faz com que o servidor DHCP aceite apenas os clientes do terminal server, sem conflitar com um servidor DHCP já existente. Caso prefira que o servidor DHCP atribua endereços também para os demais micros da rede (que não estão cadastrados como terminais), comente a linha "deny unknown-clients" e descomente a linha abaixo, informando a faixa de endereços que será usada pelos clientes que não estejam cadastrados como terminais. Você pode usar os endereços de .11 a .50 para os terminais e de .100 a .200 para os demais micros, por exemplo.

range 192.168.0.100 192.168.0.201;

Lembre-se que a linha "range" conflita com a "deny unknown-clients", você deve sempre usar uma ou outra, nunca ambas ao mesmo tempo.

A seguir vem a configuração dos terminais, onde você deve fornecer o endereço MAC de cada um. O "fixed-address 192.168.0.11;" é o endereço IP que o servidor DHCP dará para

cada terminal, vinculado ao endereço MAC da placa de rede e o arquivo que ele carregará durante o boot, como em:

```
host ws001 {  
hardware ethernet 00:E0:7D:B2:E5:83;  
fixed-address 192.168.0.11;  
filename "lts/2.6.17.3-ltsp-1/pixelinux.0";  
}
```

Esta configuração pode ser repetida ad-infinitum, uma vez para cada terminal que adicionar, mudando apenas o nome do terminal (ws001), o MAC da placa e o IP que será usado por ele.

O cliente PXE é capaz de carregar apenas arquivos pequenos, de no máximo 32k. Por isso, antes de carregar o Kernel é necessário carregar um bootstrap, o arquivo **pixelinux.0**, que se responsabiliza por obter a configuração via DHCP e carregar o Kernel, dando início ao boot.

Os arquivos de boot são instalados por padrão dentro da pasta "**/tftpboot**". Você verá uma pasta separada para cada Kernel disponível, como em: "**2.6.17.3-ltsp-1**".

A pasta contém um conjunto completo, como respectivo Kernel, um arquivo initrd, o arquivo pixelinux.0 e um arquivo de configuração para ele, o "pixelinux.cfg/default". Esse arquivo contém instruções que serão executadas pela estação ao carregar o arquivo pixelinux.0, incluindo a localização do Kernel e do arquivo initrd correspondente.

No LTSP 4.1 estavam disponíveis dois Kernels diferentes, um da série 2.4 (mais leve) e outro da série 2.6. No LTSP 4.2 voltou a ser usado um único Kernel unificado (o 2.6.17.3-ltsp-1), que, além de mais atualizado, é extremamente otimizado, a ponto de consumir menos memória que o Kernel da série 2.4 usado pelo LTSP 4.1.

Note que a versão do Kernel usada e consequentemente o nome da pasta mudam a cada versão do LTSP. Lembre-se de sempre verificar a versão incluída na sua instalação e alterar a configuração de forma apropriada.

Esta configuração para clientes PXE funciona também para clientes que dão boot usando os discos do Etherboot. Isso permite que você unifique a configuração dos clientes, facilitando as coisas.

Em versões antigas do LTSP era necessário trabalhar com dois tipos de configuração diferentes, uma para os clientes PXE e outra para os clientes Etherboot. Como disse, isso não é mais necessário nas versões atuais, mas, apenas a título de desencargo, aqui vai um exemplo da configuração para clientes Etherboot:

```
host ws005 {  
hardware ethernet 00:E0:7D:AB:E3:11;  
fixed-address 192.168.0.15;  
filename "lts/vmlinuz-2.6.17.3-ltsp-1";  
}
```

Veja que a mudança é o arquivo de Kernel que será carregado. Ao invés de carregar o bootstrap pxelinux.0, a estação passa a carregar o Kernel diretamente.

Depois de configurar o arquivo, reinicie o servidor DHCP:

```
# /etc/init.d/dhcp3-server restart
```

Dê boot em algum dos clientes para testar. Com o DHCP funcionando, eles devem receber a configuração da rede e parar no ponto em que tentam carregar a imagem de boot via TFTP:

```
Network boot from AMD Am79C970A
Copyright (C) 2003-2005 VMware, Inc.
Copyright (C) 1997-2000 Intel Corporation

CLIENT MAC ADDR: 00 0C 29 F4 34 19  GUID: 564D61B9-C816-D0B6-485E-01EB03F43419
CLIENT IP: 192.168.0.18  MASK: 255.255.255.0  DHCP IP: 192.168.0.10
GATEWAY IP: 192.168.0.1
TFTP...._
```

Se houver algum erro com o DHCP, revise a configuração antes de continuar. Não deixe que os erros se acumulem, caso contrário você vai acabar perdendo bem mais tempo.

Uma observação importante relacionada à configuração da rede: nunca use um alias (criado através do comando ifconfig eth0:1) para criar o endereço IP indicado nos arquivos de configuração do LTSP. Por exemplo, se nos arquivos o endereço IP do servidor é "192.168.0.10", é preciso que este seja o endereço IP real da placa de rede, seja a eth0 ou eth1. Se precisar criar uma segunda placa de rede virtual (para acessar a internet, por exemplo), configure a placa de rede principal para usar o endereço IP indicado nos arquivos de configuração e use o alias para criar o outro endereço. Nesse caso, a eth0 (para os clientes LTSP) ficaria com o IP "192.168.0.10" e a eth0:1 (para o resto da rede) ficaria com, por exemplo, "10.0.0.1".

» Próximo: [TFTP](#)

Com o DHCP funcionando, o próximo passo é ativar o servidor TFTP, para que as estações possam carregar a imagem de boot.

No Debian existem duas opções de servidor TFTP. O "tftpd" é uma versão obsoleta, que não suporta boot via PXE e que, por isso, é fortemente não-recomendado. A versão atual é instalada através do pacote "[tftpd-hpa](#)":

```
# apt-get install tftpd-hpa
```

Quando o script de instalação perguntar "Deverá o servidor ser iniciado pelo inetd?", responda que **não**.

Depois de instalar o pacote, edite o arquivo **/etc/default/tftpd-hpa**. Por padrão ele vem com uma linha que mantém o servidor desativado, mesmo que as demais configurações estejam corretas. Para que o serviço fique ativo, modifique a linha "RUN_DAEMON="no" para "RUN_DAEMON="yes". Altere também a linha "OPTIONS="-l -s /var/lib/tftpboot" (que indica a pasta que será compartilhada pelo servidor tftpd), substituindo o "/var/lib/tftpboot" por "/tftpboot".

Depois da alteração o arquivo fica:

```
#Defaults for tftpd-hpa
RUN_DAEMON="yes"
OPTIONS="-l -s /tftpboot"
```

Para que a alteração entre em vigor, reinicie o serviço tftpd-hpa:

```
# /etc/init.d/tftpd-hpa stop
# /etc/init.d/tftpd-hpa start
```

Por desencargo, usamos também os comandos abaixo, para ter certeza de que ele está configurado para ser executado durante o boot:

```
# update-rc.d -f tftpd-hpa defaults
```

No Debian Etch, o servidor tftpd-hpa pode conflitar com o inetd. Se mesmo depois de instalar e fazer toda a configuração, o servidor tftpd continuar a não responder às requisições das estações (elas continuarão parando no "TFTP...."), este é provavelmente o seu caso. Para solucionar o problema, desative o inetd, utilizando os comandos abaixo e, em seguida, reinicie o servidor:

```
# /etc/init.d/inetd stop
# update-rc.d -f inetd remove
# reboot
```

Ao desativar o inetd, o acesso remoto ao vmware-server e o swat (caso instalados no servidor) deixarão de funcionar. Caso precise utilizar estes serviços, reactive o inetd manualmente quando necessário, utilizando o comando:

```
# /etc/init.d/inetd start
```

Para concluir a configuração, abra o arquivo **/etc/hosts.allow** e substitua todo conteúdo do arquivo por:

```
# /etc/hosts.allow para o LTSP 4.2
#
# Esta configuração permite que todos os serviços usados na rede local utilizem
# os micros da rede pelo LTSP.
```

```
# Altere o "192.168.0." caso você esteja utilizando outra faixa de endereços na sua rede:
```

```
ALL : 127.0.0.1 192.168.0.0/24
```

É essencial que a linha "ALL : 127.0.0.1 192.168.0.0/24" esteja presente, caso contrário o sistema recusa as conexões dos clientes mesmo que os serviços estejam corretamente ativados. Ao editar este arquivo, não é necessário reiniciar nenhum serviço. Não se esqueça de substituir o "192.168.0.0" pela faixa de endereços da sua rede, caso diferente.

Com o TFTP funcionando, as estações conseguem carregar a imagem inicial de boot e o Kernel e prosseguem até o ponto em que tentam montar o diretório "/opt/ltsp/i386" do servidor via NFS:

```
pcnet32: PCnet/PCI II 79C970A at 0x1080, 00 0c 29 f4 34 19 assigned IRQ 11.
eth0: registered as PCnet/PCI II 79C970A
pcnet32: 1 cards_found.
Running dhcpcd on port 67
Creating new ramdisk to hold our root fs...
Mounting root filesystem: /opt/ltsp/i386 from: 192.168.0.10
Mount: RPC: Unable to receive; errno = Connection refused
Mount: nfsmount failed: Bad file descriptor
Mount: Mounting 192.168.0.10:/opt/ltsp/i386 on /newroot/nfsroot failed: Invalid argument

ERROR! Failed to mount the root directory via NFS!
Possible reasons include:
    1) NFS services may not be running on the server
    2) Workstation IP does not map to a hostname, either
       in /etc/hosts, or in DNS
    3) Wrong address for NFS server in the DHCP config file
    4) Wrong pathname for root directory in the DHCP config file

Kernel panic - not syncing: Attempted to kill init!
-
```

Se você está vendo esta mensagem, significa que está tudo funcionando. Esta mensagem de erro é, na verdade, uma boa notícia. O próximo passo é configurar o NFS, para que as estações possam concluir o boot.

» Próximo: [NFS](#)

O próximo arquivo é o **/etc(exports**, onde vai a configuração do servidor NFS. O LTSP precisa que o diretório "/opt/ltsp/i386/" esteja disponível (como somente leitura), para toda a faixa de endereços usada pelas estações. Para isso, adicione a linha:

```
/opt/ltsp/i386/ 192.168.0.0/255.255.255.0(ro,no_root_squash)
```

Note que a configuração dos compartilhamentos inclui a faixa de endereços e a máscara usada na rede (192.168.0.0/255.255.255.0 no exemplo, configuração usada para evitar que eles sejam acessados de fora da rede). Não se esqueça de alterar esses valores ao utilizar uma faixa diferente, caso contrário, o servidor passa a recusar os acessos dos terminais e eles não conseguirão mais carregar o sistema de boot.

Vamos a uma revisão da configuração do NFS:

Para ativar o servidor nas distribuições derivadas do **Debian**, você precisa ter instalados os pacotes "**portmap**", "**nfs-common**" e "**nfs-kernel-server**":

```
# apt-get install portmap nfs-common nfs-kernel-server
```

O portmap deve sempre ser inicializado antes dos outros dois serviços, já que ambos dependem dele. O default é que o portmap seja iniciado através do link "/etc/init.d/rcS.d/S43portmap" (carregado no início do boot), enquanto os outros dois serviços são carregados depois, através de links na pasta "/etc/rc5.d" ou "/etc/rc3.d".

Use os comandos abaixo para corrigir a posição do serviço portmap e verificar se os outros dois estão ativos e configurados para serem inicializados na hora do boot:

```
#      update-rc.d           -f          portmap        remove
#      update-rc.d           -f          nfs-common     remove
#      update-rc.d           -f          nfs-kernel-server remove
#      update-rc.d   -f      portmap       start    43      S   .
#      update-rc.d   -f      nfs-common    start    20      5   .
# update-rc.d -f nfs-kernel-server start 20 5 .
```

Para que o servidor NFS funcione, é necessário que o arquivo "**/etc/hosts**" esteja configurado corretamente. Este arquivo relaciona o nome do servidor e de cada uma das estações a seus respectivos endereços IP. Se você copiou meu modelo a partir do site, o arquivo estará assim:

```
# /etc/hosts, configurado para o LTSP 4.2

127.0.0.1 servidor localhost

# Você pode adicionar aqui os endereços IP e os nomes correspondentes
# de cada terminal, caso queira utilizar mais de 8 terminais.

# IMPORTANTE: A primeira linha deve conter o endereço IP e o nome
# (definido durante a configuração da rede) do servidor, ou seja,
# desta máquina. Se o nome for diferente do definido na configuração
# da rede, as estações não seguirão montar o sistema de arquivos do
# LTSP via NFS e travarão no boot.

192.168.0.10                                servidor
192.168.0.11                                ws001
192.168.0.12                                ws002
192.168.0.13                                ws003
192.168.0.14                                ws004
192.168.0.15                                ws005
192.168.0.16                                ws006
```

192.168.0.17
192.168.0.18 ws008

ws007

É importante que você substitua o "servidor" pelo nome correto do seu servidor (que você checa usando o comando "**hostname**") e substitua os endereços IP's usados pelo servidor e pelas estações caso esteja usando outra faixa de endereços. Sem isso, as estações vão continuar parando no início do boot, com uma mensagem de erro relacionada a permissões do NFS.

Com o NFS funcionando, a estação avança mais um pouco no boot. Agora ela consegue montar o diretório raiz e começar o carregamento do sistema, mas para no ponto em que procura pelo arquivo "**lts.conf**", que é justamente o principal arquivo de configuração do LTSP, carregado pelas estações no início do boot:

```
pcnet32: PCnet/PCI II 79C970A at 0x1080, 00 0c 29 f4 34 19 assigned IRQ 11.
eth0: registered as PCnet/PCI II 79C970A
pcnet32: 1 cards_found.
Running dhcpcd on port 67
Creating new ramdisk to hold our root fs...
Mounting root filesystem: /opt/ltsp/i386 from: 192.168.0.10
Setting up the new root ramdisk area...
Doing the switchroot
SwitchRoot v0.1 - Copyright (c) 2005 Linux Based Systems Design
Freeing ram used by initramfs
input: IMPS/2 Generic Wheel Mouse as /class/input/input1
Mounting /sys...
Mounting /proc...
Starting udevd...
Running udevstart to create initial device nodes...
Mounting devpts...
Building /etc/inittab
Error retrieving file stats for file [/etc/lts.conf]: No such file or directory
Done with early_sysinit

Could not find lts.conf file!

Perhaps you haven't run ltspcfg yet
```

» Próximo: [O arquivo principal: Its.conf](#)

Finalmente, chegamos à parte mais importante da configuração, que fica a cargo do arquivo "**/opt/ltsp/i386/etc/lts.conf**". É aqui que você diz qual a resolução de vídeo e que tipo de mouse será usado em cada estação e tem a opção de ativar ou não o swap via rede do LTSP.

O lts.conf pode ser dividido em duas partes. A primeira contém as configurações default, que são usadas por todas as estações até que dito o contrário. Em seguida temos uma minisessão que especifica opções adicionais para cada estação. Isso permite que uma

estaçao seja configurada para usar mouse serial e teclado US Internacional, mesmo que todas as demais usem mouses PS/2 e teclados ABNT2.

Este é um exemplo de sessão default, similar ao que incluí no modelo disponível no site:

```
[Default]
SERVER = 192.168.0.10

XSERVER = auto

X_MOUSE_PROTOCOL          = "PS/2"
X_MOUSE_DEVICE             = "/dev/psaux"
X_MOUSE_RESOLUTION         = 400
X_MOUSE_BUTTONS = 3

XkbModel                  = ABNT2
XkbLayout = br

SCREEN_01                  = startx
RUNLEVEL = 5
```

Logo no início do arquivo você deve prestar atenção para substituir o "192.168.0.10" pelo IP correto do seu servidor, senão os clientes não conseguirão dar boot nem por decreto :).

Abaixo, a partir da opção "XSERVER", vai a configuração default do LTSP. Não existe necessidade de alterar nada aqui, pois você pode especificar configurações diferentes para cada estação mais abaixo, especificando diferentes resoluções de vídeo, tipos de mouse e taxas de atualização de monitor. Lembre-se de que na verdade as estações executam localmente uma cópia do Kernel, utilitários básicos e uma instância do X. Graças a isso, a configuração de vídeo de cada estação é completamente independente do servidor. Nada impede que uma estação use um monitor de 17" a 1280x1024, enquanto outra usa um VGA Mono a 640x480.

Este é o principal motivo de relacionarmos os endereços MAC de cada placa de rede com um nome de terminal e endereço IP específico na configuração do DHCP. Graças a isso, o servidor consegue diferenciar os terminais e enviar a configuração correta para cada um.

Desde o 4.1, o LTSP utiliza o X.org, que possui um sistema de detecção automática para o vídeo em cada estação (a opção "XSERVER = auto"). No final do boot ele tentará detectar a placa de vídeo e detectar as taxas de atualização suportadas pelo monitor via DDC. Este sistema funciona direto em uns dois terços dos micros, mas em um grande número de casos você precisará especificar algumas configurações manualmente para que tudo funcione adequadamente.

Veja também que o default do LTSP é utilizar um mouse PS/2 (sem roda) em todas as estações. Naturalmente você terá alguns micros com mouses seriais ou PS/2 com roda, o que também precisaremos arrumar. Esta configuração individual das estações é feita logo abaixo, relacionando o nome de cada estação com as opções desejadas, como em:

```
[ws001]
XSERVER = sis
X_MOUSE_PROTOCOL = "IMPS/2"
X_MOUSE_DEVICE = "/dev/input/mice"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 5
X_ZAxisMapping = "4 5"
```

Aqui estou especificando que o ws001 usa o driver de vídeo "sis" e um mouse PS/2 com roda. Você pode criar uma seção extra para cada estação. Esta configuração não é obrigatória, pois as estações que não possuírem sessões exclusivas simplesmente seguirão os valores incluídos na sessão "Default".

Uma curiosidade é que o arquivo lts.conf é lido pelas próprias estações durante o boot. Como elas não possuem HD, nada mais justo do que armazenar suas configurações diretamente no servidor.

Além dessas, existem várias outras opções que podem ser usadas. Se a detecção automática do vídeo não funcionar (a tela vai piscar algumas vezes e depois voltar ao modo texto) você pode indicar manualmente um driver de vídeo, substituindo o "**auto**" por "**vesa**" (um driver genérico, um pouco mais lento mas que funciona na maioria das placas).

Outros drivers disponíveis são: **cirrus** (placas da Cirrus Logic), **i810** (placas com vídeo onboard Intel), **nv** (driver 2D para placas nVidia), **r128** (placas Riva 128 da ATI), **radeon** (ATI Radeon), **rendition**, **s3virge**, **sis** (driver genérico para placas onboard e offboard da SiS), **tdfx** (placas Voodoo Banshee, Voodoo 3 e 4), **trident** e **via** (que dá suporte às placas-mãe com vídeo onboard Via Unichrome, comuns hoje em dia).

Você pode encontrar detalhes sobre as placas suportadas por cada um no <http://www.x.org/X11R6.8.2/doc/>.

Outra configuração importante é o tipo de mouse usado nos terminais. Afinal, não é sempre que você utilizará mouses PS/2. Basta incluir algumas opções, como nos exemplos abaixo.

Exemplo para usar um **mouse serial** na estação:

```
[ws001]
XSERVER = auto
X_MOUSE_PROTOCOL = "Microsoft"
X_MOUSE_DEVICE = "/dev/ttyS0"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 2
X_MOUSE_EMULATE3BTN = Y
```

Exemplo para usar um **mouse PS/2 com roda** (esta configuração também funciona para mouses USB) na estação:

```
[ws001]
XSERVER = auto
X_MOUSE_PROTOCOL = "IMPS/2"
X_MOUSE_DEVICE = "/dev/input/mice"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 5
X_ZAxisMapping = "4 5"
```

O exemplo abaixo força a estação a usar uma configuração de resolução e taxa de atualização específica para o monitor. Ela é útil em casos em que o X chega a abrir, mas o monitor fica fora de sintonia. Isso acontece em muitos micros antigos, em que o monitor ou a placa de vídeo não são compatíveis com o protocolo DDC:

```
[ws001]
XSERVER = auto
X_MODE_0 = 1024x768 #(Resolução de vídeo)
X_VERTREFRESH = 60 #(Refresh rate do monitor)
X_COLOR_DEPTH = 16 #(Bits de Cor)
```

Outra configuração que pode ser importante é o **teclado**. Por padrão, o LTSP vem configurado para usar um teclado padrão americano, sem acentuação.

Ao carregar o KDE, passam a valer as configurações do Kxkb, o gerenciador de teclado do KDE, da forma como configuradas no Painel de Controle (do KDE). O problema é que o Kxkb só funciona se coincidir da distribuição instalada no servidor usar a mesma versão do X.org usada pelo LTSP. Caso contrário, ele mostra um "err" e não funciona:



Devido a isso, é mais simples desativar os layouts de teclado do KDE e definir a configuração do teclado diretamente no arquivo lts.conf.

Para um teclado **ABNT2**, inclua as linhas abaixo, dentro da sessão "Default", ou dentro da configuração de cada estação:

```
XkbModel = ABNT2
XkbLayout = br
```

Dependendo da versão do X usada no servidor, você pode encontrar um problema estranho, onde as teclas "\|" e "]}" nas estações ficam trocadas por "<>" e "\|". A solução, nesse caso, é abrir o arquivo **".xmodmap"** dentro do diretório **"/etc/skel"** e dentro do home de cada usuário, adicionando as linhas:

```
keycode          94          =          backslash          bar
keycode 51 = bracketright braceright
```

Você pode também usar o script abaixo para adicionar as duas linhas nos arquivos .xmodmap dentro dos homes de todos os usuários do sistema automaticamente (útil se você já tiver um servidor com vários usuários configurado):

```
cd /home
for i in *; do echo '
keycode 94 = backslash
keycode 51 = bracketright
' >> $i/.xmodmap; done
```

Para um teclado **US Internacional** a configuração é mais simples. Use as duas linhas abaixo na configuração das estações:

```
XkbModel          =          pc105
XkbLayout         =          us_intl
XkbRules = xorg
```

A linha "**RUNLEVEL = 5**" (que adicionamos na sessão default), faz com que as estações dêem boot direto em modo gráfico, que é o que queremos. Se por acaso você quiser ter alguma estação trabalhando em modo texto (para tentar descobrir o motivo de algum problema, por exemplo), inclua a linha "RUNLEVEL = 3" na configuração da estação.

» Próximo: [Swap](#)

O LTSP inclui um recurso de swap via rede, destinado a estações com 32 MB de RAM ou menos. Ela permite que a estação (que não possui um HD local) faça swap usando o HD do servidor, o que permite usar micros com a partir de 8 MB de RAM como terminais.

Até o LTSP 4.1 era usando um sistema de swap via NFS, que era relativamente lento. No LTSP 4.2 passou a ser usado um novo sistema de swap via NBD (Network Block Device), que, além de mais rápido, é mais estável.

Para usá-lo, você deve primeiro baixar e instalar o pacote "ltsp-localdev". A versão para o Debian Sarge ou Etch está disponível no: <http://ltsp.mirrors.tds.net/pub/ltsp/utils/>.

A versão para o Debian é o arquivo "ltsp-server-pkg-debian_0.1_i386.deb". Na mesma pasta estão disponíveis as versões para outras distribuições.

Depois de baixar o arquivo, instale-o usando o dpkg:

```
# dpkg -i ltsp-server-pkg-debian_0.1_i386.deb
# apt-get -f install
```

Uma observação importante é que esta versão do pacote foi criada para ser instalada no Debian Sarge e distribuições baseadas nele. Ele tem marcado o pacote "fuse-source" como dependência, o que causa um grande problema nas distribuições atuais, já que o fuse passou a vir incluído diretamente no Kernel, fazendo com que o pacote deixe de estar disponível.

Se ao instalar o pacote você receber um erro relacionado à falta do pacote "fuse-source" e o "apt-get -f install" não for capaz de resolver o problema, baixe este pacote:

<http://www.guiadohardware.net/kurumin/download/fuse-source.deb>

Ele é um pacote vazio, que serve apenas para suprir a dependência do pacote enquanto uma versão atualizada não é disponibilizada. Instale-o usando o dpkg e rode o "apt-get -f install":

```
#                               dpkg           -i           fuse-source.deb
# apt-get -f install
```

O pacote ltsp-server inclui o serviço "**ltspswapd**", responsável pelo swap via rede no LTSP 4.2. Ative o serviço e configure-o para subir durante o boot:

```
#                               /etc/init.d/ltspswapd      start
# update-rc.d -f ltspswapd defaults
```

Agora falta apenas adicionar a linha abaixo na configuração das estações que forem utilizar swap, dentro do arquivo **lts.conf**:

USE_NBD_SWAP = Y

O default é armazenar os arquivos de swap dentro da pasta "/var/spool/ltspwap" do servidor, permitindo que cada cliente use um máximo de 64 MB. Esta configuração é mais do que adequada, mas, caso precise alterar, crie o arquivo "/etc/sysconfig/ltspswapd" contendo a linha:

ARGS="[-p 9210] [-s /var/spool/ltspwap] [-z 64mb] [-d]"

Desta forma, você pode trocar o "/var/spool/ltspwap" e o "64mb" pelos valores desejados.

Com o swap ativo, você notará que as estações passarão a exibir uma mensagem "Formating Swap" rapidamente durante o boot, e os arquivos de cada estação serão criados dentro do diretório "/var/spool/ltspwap", como em:

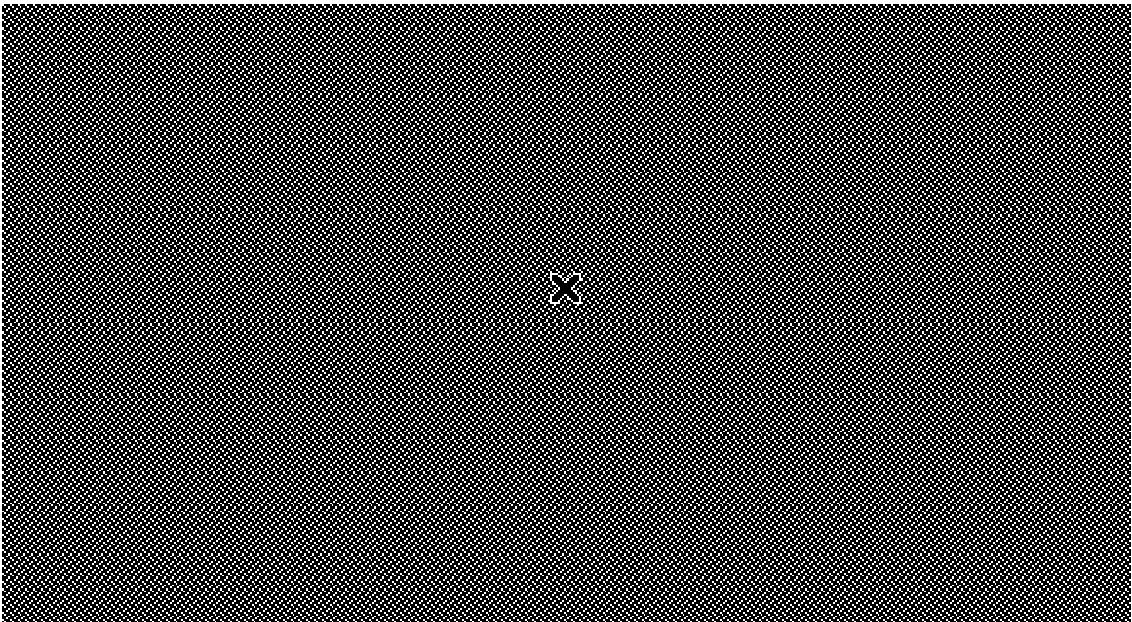
```
# ls -lh /var/spool/ltspswap/
```

total	129M
-rw----- 1 root root 64M 2006-05-23 19:09	192.168.0.11.swap
-rw----- 1 root root 64M 2006-05-24 10:08	192.168.0.12.swap

» Próximo: [Testando](#)

Depois de configurar o arquivo "/opt/lts/i386/etc/lts.conf", os clientes devem conseguir completar o boot, carregando o ambiente gráfico. Se os clientes continuarem parando em algum dos pontos anteriores, verifique a configuração dos serviços que já vimos, tente localizar qual serviço não está funcionando e, a partir daí, corrigir o problema.

Um problema comum nesta etapa é o terminal concluir o boot e carregar o X, mas não conseguir abrir a tela de login do servidor, exibindo em seu lugar a "tela cinza da morte" ;)



Isso acontece se você se esqueceu de ativar o **XDMCP** no servidor. Toda configuração que vimos até aqui permite que a estação dê boot pela rede, mas é o XDMCP que permite que ela rode aplicativos.

Revisando, em distribuições que usam o KDM (Kurumin, Mandriva, SuSE, etc.) você deve editar dois arquivos, o "**kmrc**" e o "**Xaccess**". Nas distribuições derivadas do Debian, ambos os arquivos vão na pasta "**/etc/kde3/kdm/**", enquanto outras distribuições usam a pasta "**/usr/share/config/kdm**".

No **kmrc**, procure pela linha "[Xdmcp]" e substitua o "Enable=false" por "Enable=true":

```
[Xdmcp]
Enable=true
```

No arquivo **Xaccess**, descomente as linhas que dão acesso às estações:

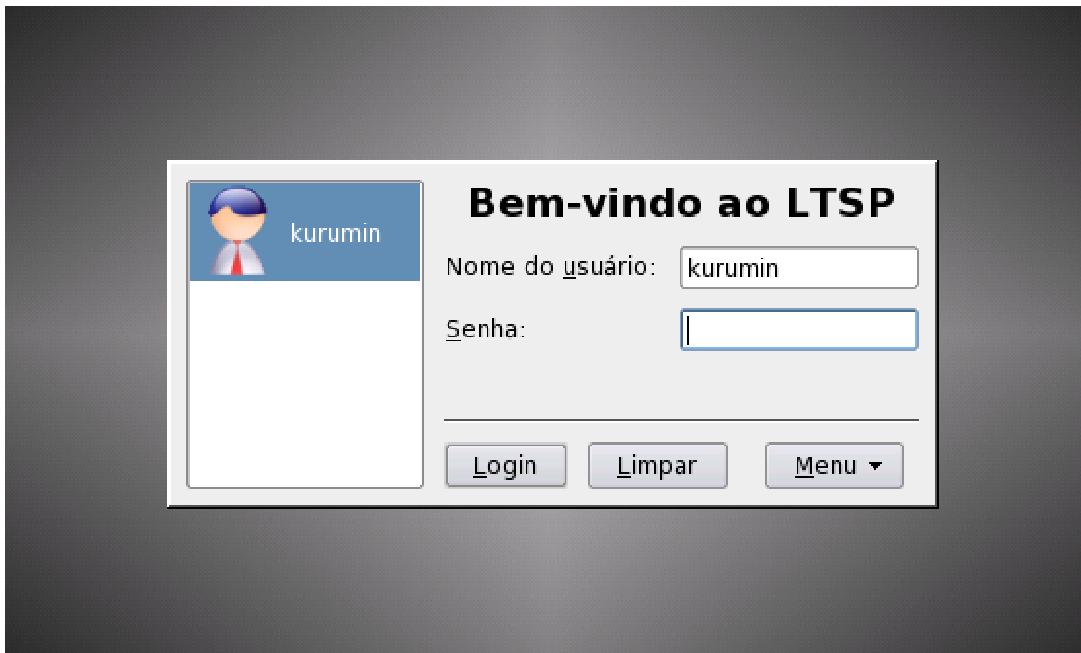
```
* #any host can get a login window
* CHOSER BROADCAST #any indirect host can get a chooser
```

Feito isso, reinicie o serviço:

```
# /etc/init.d/kdm restart
```

O **Ubuntu** utiliza uma estrutura de pacotes e arquivos muito semelhante à do Debian padrão e do Kurumin. Você pode instalar o LTSP no Ubuntu seguindo estas mesmas dicas, a única grande diferença é que ele usa o GDM no lugar do KDM.

Para ativar o XMDCP, procure pelo "gdmconfig", disponível no "System Settings > Login Screen". Acesse a aba "XDMCP" e marque a opção "Enable XDMCP". A partir daí as estações passarão a exibir a tela de login do servidor e rodar os aplicativos normalmente. Agora é só correr pro abraço :-).



Com tudo funcionando, você pode ir criando os logins dos usuários que irão utilizar os terminais. O ideal é que cada pessoa tenha seu login (e não um login para cada terminal), pois assim cada um tem acesso às suas configurações e arquivos em qualquer um dos terminais, o que é uma das grandes vantagens do uso do LTSP.

A conexão com a web, impressora, disquete e gravador instalados no servidor podem ser usados em qualquer um dos terminais, pois na verdade os programas nunca saem do servidor: os terminais funcionam apenas como se fossem vários monitores e teclados ligados a ele. Tenha à mão também a configuração da sua rede, como o endereço deste servidor, máscara de sub-rede, servidores DNS do seu provedor, gateway padrão, etc.

Uma questão importante é que o servidor não deve estar usando nenhum tipo de firewall, caso contrário você precisará fazer uma configuração muito cuidadosa, mantendo abertas cada uma das portas usadas pelos serviços relacionados ao LTSP. Por envolver tantos serviços diferentes, que precisam ficar disponíveis, um firewall em um servidor LTSP é praticamente inútil de qualquer forma.

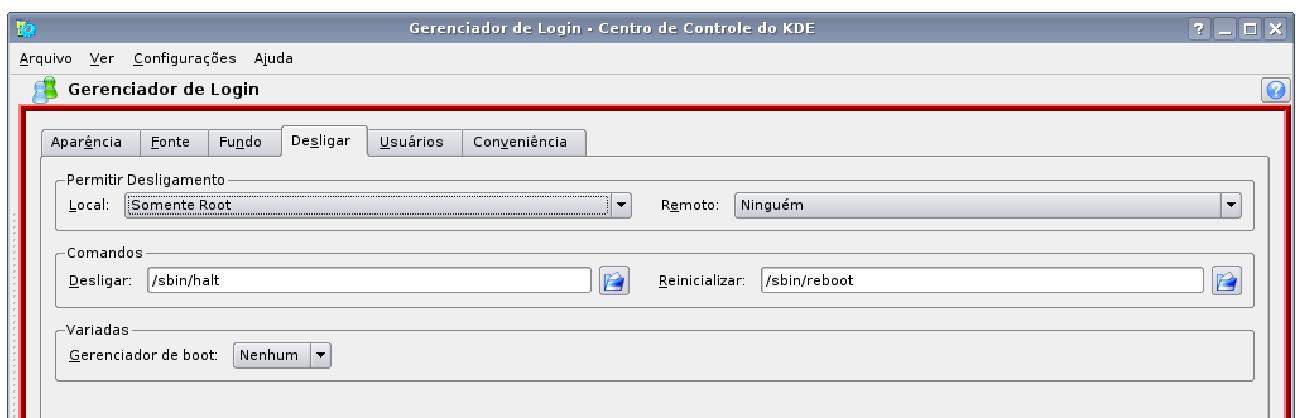
Ao invés de perder tempo com isso, o melhor é que você isole o servidor LTSP da internet, usando uma máquina separada para o compartilhamento da conexão e firewall. Faça com que o servidor LTSP acesse via NAT, por trás do firewall.

» Próximo: [Personalizando as configurações](#)

Com o servidor funcionando corretamente, o próximo passo é a personalização do ambiente e da tela de login, o que inclui personalizar a parte visual, instalar e remover programas, personalizar o menu iniciar e ícones da área de trabalho, etc., criando um ambiente adequado ao ambiente em que os terminais serão utilizados.

A primeira parada é a tela de login, que acaba sendo o cartão de visitas do sistema, já que é a primeira coisa que os usuários vêem. Acesse a opção "**Administração do Sistema > Gerenciador de Login**" dentro do Centro de Controle do KDE. Clique no "Modo administrador" e forneça a senha de root para ter acesso às ferramentas de administração.

É importante que na aba "Desligar", ninguém, ou apenas o root, possa desligar o sistema remotamente. O default, em muitas instalações do KDE, é "Todos", o que pode causar pequenos desastres em um servidor de terminais :).



O ideal em um servidor LTSP é que você crie uma conta para cada usuário e não uma conta por máquina. Isso permite que cada usuário tenha seu próprio espaço e veja sempre o seu desktop, com as mesmas configurações e arquivos. Além de melhorar o nível de satisfação, isso reduz bastante as dúvidas e os problemas de suporte, já que, se todos usam o mesmo login, a tendência é que o desktop vire uma bagunça com o tempo.

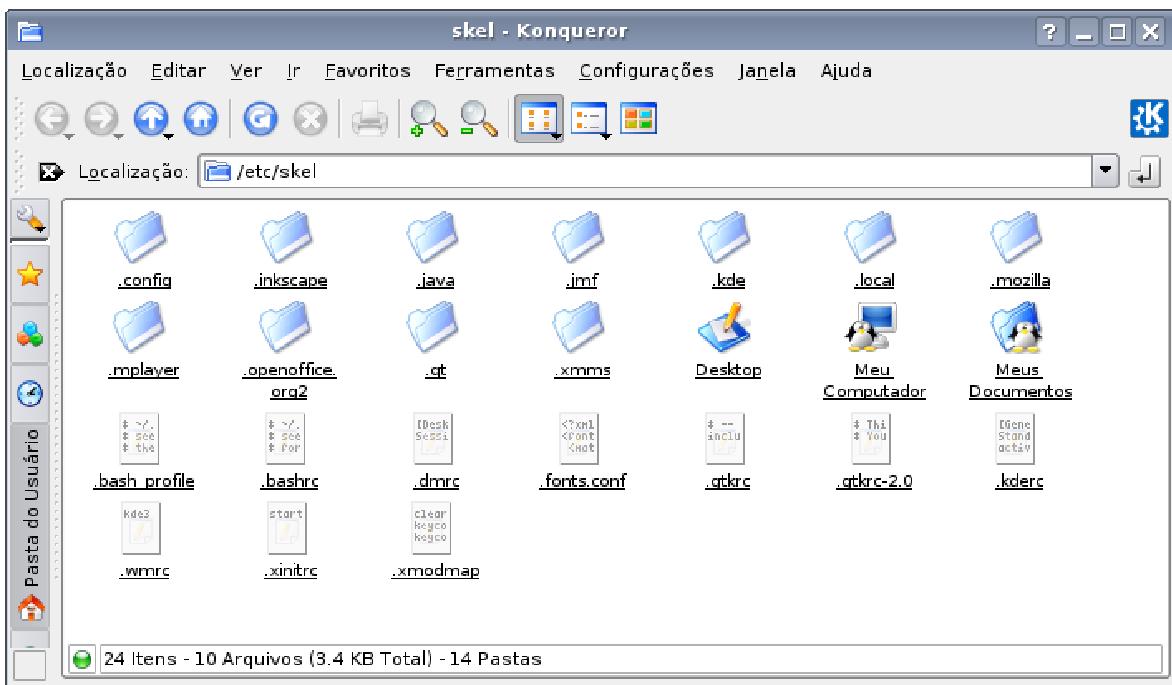
Na aba "Usuários", desmarque a opção "Mostrar lista". Isso torna a tela de login muito mais elegante, sem aquela lista gigante dos logins disponíveis e ajuda um pouco na segurança, pois evita que fiquem testando as contas em busca de logins sem senha ou com senhas fáceis.



Nas abas "Aparência", "Fonte" e "Fundo, você pode personalizar a exibição da tela de login, adicionando o logotipo da empresa e coisas do gênero.

A tela de login do KDM permite fazer login usando qualquer um dos gerenciadores disponíveis (Menu > Tipo de Sessão). Ao instalar o Gnome, XFCE, IceWM, etc. eles aparecerão automaticamente na lista de opções da tela de login. Se, por outro lado, você prefere que um único ambiente fique disponível, o melhor é desinstalar todos os outros, para evitar confusão.

Temos em seguida a personalização do desktop e menu iniciar. No Linux, todas as configurações relacionadas ao usuário são armazenadas em arquivos e pastas ocultos (cujo nome começa com ponto) dentro do seu diretório home. A maioria das configurações do KDE, por exemplo, vai para a pasta ".kde/share/config".



Quando você cria um novo usuário, o sistema usa o conteúdo da pasta "/etc/skel" como um modelo para o home, alterando apenas as permissões dos arquivos. Ou seja, se você modifica as configurações dentro do "/etc/skel", faz com que todos os usuários que criar daí em diante sejam criados já com as configurações desejadas.

A forma mais prática de fazer isso é criar um login normal de usuário e se logar através dele na tela do KDM. Remova os ícones indesejados do desktop, personalize os ícones da barra de tarefas, remova o monitor de bateria, monitor de rede e outros widgets desnecessários para o terminal e assim por diante.

Personalize as configurações usando o Centro de Controle do KDE e personalize o menu iniciar usando o Kmnededit (que você acessa clicando com o botão direito sobre o botão K). É importante que você remova os utilitários de administração do sistema e outros aplicativos que os usuários não devem usar.

Outra configuração importante é desativar o protetor de tela, pois protetores muito animados ativos nos terminais geram um grande tráfego na rede, prejudicando o uso dos demais terminais.

Todas essas configurações são específicas do usuário, salvas em arquivos espalhados pelo home. Precisamos agora mover tudo para o "/etc/skel", fazendo com que as configurações se tornem padrão. Não esqueça de ajustar as permissões. Se o usuário usado se chama "manuel", por exemplo, os comandos seriam:

```
# rm -rf /home/manuel
# cp -a /etc/skel /home/manuel
# chown -R root.root /etc/skel
```

É recomendável também que você use o Kfind para procurar por referências "hardcoded" ao nome do usuário, substituindo qualquer eventual citação ao nome do usuário por "\$USER", uma variável neutra, que indica o nome do usuário atualmente logado.

Quando precisar alterar algo nas configurações padrão, basta repetir o procedimento. O maior problema é que as alterações afetam apenas os usuários criados daí em diante. Por isso, o ideal é que você faça a personalização logo depois de instalar o servidor.

» Próximo: [Uma palavra sobre segurança](#)

O mais problemático em relação à segurança no LTSP são justamente os ataques locais, feitos pelos próprios usuários. Originalmente, um usuário comum não deve conseguir alterar arquivos fora do seu diretório home, nem alterar as configurações do sistema, mas eventualmente podem existir vulnerabilidades locais, que podem ser exploradas para obter privilégios adicionais.

Essas vulnerabilidades locais são muito mais comuns que vulnerabilidades remotas, já que é muito mais difícil proteger o sistema de um usuário que está logado e tem acesso a vários aplicativos, do que de um usuário remoto que precisa passar pelo firewall e encontrar algum serviço vulnerável escutando conexões.

Um exemplo rápido de como isso funciona: imagine que, por descuido, você usou no terminal (como root) o comando "chmod +s /usr/bin/mcedit". O "chmod +s" ativa o SUID para o mcedit, o que faz com que ele seja sempre executado pelo root. Um usuário que percebesse isso, poderia usar o mcedit para editar o arquivo "/etc/passwd" e modificar a linha referente a seu login para:

```
joaozinho:x:0:0:joaozinho:/home/joaozinho:/bin/bash
```

Com o SUID ativado, o medit passaria a ser sempre executado com permissão de root. Isso permitiria que o joaozinho o usasse para editar arquivos que originalmente apenas o root poderia editar. Modificando os campos do UID e GID dentro do "/etc/passwd" para "0", joaozinho faz com que seu login ganhe poderes de root. A partir daí ele pode fazer o que quiser no sistema.

Esse é um exemplo exagerado, que mostra como pequenos erros podem abrir brechas graves, que não podem ser exploradas remotamente, mas que podem ser facilmente exploradas por usuários locais.

Um tipo de ataque grave e relativamente comum são os famosos **rootkits**, softwares que exploram um conjunto de vulnerabilidades conhecidas para tentar obter privilégios de root na máquina afetada. Existem vários rootkits que podem ser baixados da Net, variando em nível de eficiência e atualização.

Os rootkits podem ser instalados tanto localmente (quando alguém tem acesso físico à sua máquina) quanto remotamente, caso o intruso tenha acesso via SSH, VNC, XDMCP (usado

pelo LTSP) ou qualquer outra forma de acesso remoto. Nesse caso, ele precisará primeiro descobrir a senha de algum dos usuários do sistema para poder fazer login e instalar o programa. A partir do momento que é possível logar na máquina, o atacante executa o rootkit para tentar obter privilégios de root.

Uma vez instalado, o rootkit vai alterar binários do sistema, instalar novos módulos no Kernel e alterar o comportamento do sistema de várias formas para que não seja facilmente detectável. O processo do rootkit não aparecerá ao rodar o "ps -aux", o módulo que ele inseriu no Kernel para alterar o comportamento do sistema não vai aparecer ao rodar o "lsmod" e assim por diante.

Aparentemente vai estar tudo normal, você vai poder continuar usando a máquina normalmente, mas existirão outras pessoas com acesso irrestrito a ela, que poderão usá-la remotamente da forma que quiserem.

Naturalmente também existem programas capazes de detectar rootkits. Um dos mais populares é o **chkrootkit**, que pode ser encontrado no: <http://www.chkrootkit.org/>.

No site está disponível apenas o pacote com o código fonte, que você precisa compilar manualmente, mas ele é um programa bastante popular e vem incluso na maior parte das distribuições. No Debian, Kurumin ou derivados, você pode instalá-lo pelo apt-get:

```
# apt-get install chkrootkit
```

Ele pergunta se deve ser executado automaticamente todos os dias, através do cron. Isso garante uma proteção adicional, pois ele avisa caso futuramente a máquina seja comprometida.

Para executar o chkrootkit, basta chamá-lo no terminal:

```
# chkrootkit
```

Ele exibe um longo relatório, mostrando um por um os arquivos checados. Em uma máquina saudável, todos retornarão um "nothing found":

```
Searching      for      Ramen      Worm      files      and      dirs...      nothing      found
Searching      for      Maniac     files      and      dirs...      nothing      found
Searching      for      RK17       files      and      dirs...      nothing      found
Searching      for      Ducoci     rootkit...   nothing      found
Searching      for      Adore      Worm...    nothing      found
Searching      for      ShitC      Worm...    nothing      found
Searching      for      Omega      Worm...    nothing      found
...
...
```

Uma parte importante é a checagem das interfaces de rede, que aparece no final do relatório:

```
Checking      `sniffer'...      lo:      not      promisc      and      no      packet      sniffer      sockets
eth0: not promisc and no packet sniffer sockets
```

Os **sniffers** são usados para monitorar o tráfego da rede e, assim, obter senhas e outras informações não apenas do servidor infectado, mas também de outras máquinas da rede local. Um dos sintomas de que existe algum sniffer ativo é a placa da rede estar em modo promíscuo, onde são recebidos também pacotes destinados a outros micros da rede local.

Alguns programas, como o VMware, o Ethereal e o Nessus colocam a rede em modo promíscuo ao serem abertos, mas caso isso aconteça sem que você tenha instalado nenhum destes programas, é possível que outra pessoa o tenha feito.

Instale o chkrootkit logo depois de configurar o sistema e execute o teste regularmente. Ele é capaz de detectar a maioria das intrusões de usuários locais, permitindo que você tenha uma certa segurança. Caso seja detectada uma intrusão, o ideal é desconectar o servidor da rede, fazer um backup dos dados de todos os usuários e reinstalar o sistema do zero, depois de enforcar e esquartejar o usuário "esperto" naturalmente ;).

Essas precauções não são necessárias se seus usuários são todas pessoas cultas e integrais, interessadas no bem comum (sarcástico). Mas, como não vivemos em um mundo ideal, o melhor é tomar as precauções necessárias e tentar estar sempre um passo a frente.

» Próximo: [Mais configurações](#)

Uma das dúvidas mais comuns com relação ao uso do LTSP é com relação aos dispositivos locais. Como acessar o CD-ROM, disquete ou pendrive conectado à estação? Como fazer com que o som dos aplicativos saia pelas caixas de som da estação e como fazer com que um documento seja impresso na impressora conectada à estação, ao invés de na impressora do servidor?

Isso é mais complicado do que parece à primeira vista, pois ao usar o LTSP as estações exibem na verdade uma sessão remota do servidor. Para que o usuário consiga acessar um CD-ROM colocado no drive da estação de dentro dessa sessão remota, é necessário que a estação compartilhe o CD-ROM com a rede e o servidor monte este compartilhamento, mostrando os arquivos ao usuário de uma forma transparente.

Até o LTSP 4.1, configurar o acesso a dispositivos locais nos clientes era possível, porém bastante trabalhoso. Era usada uma combinação de servidor Samba nas estações (para compartilhar o CD-ROM e disquete com o servidor) e autofs junto com diversos scripts no servidor. Uma salada onde muita coisa podia dar errado.

No LTSP 4.2 passou a ser usado um sistema novo e quase revolucionário de acesso a dispositivos locais, muito mais simples, funcional e robusto que o anterior. Ao inserir um CD-ROM no drive ou plugar um pendrive, são criados ícones no desktop, que desaparecem automaticamente ao ejectar o CD ou remover o pendrive. Existe suporte também a drive de disquetes, caso você seja um dos pobres coitados que ainda é obrigado a conviver com eles ;).



O suporte a impressora também funciona de forma robusta. Apenas o suporte a som ainda é um pouco problemático e trabalhoso de configurar, mas com um pouco de dedicação é possível usar todos os recursos da estação, da mesma forma que usaria um PC local, porém tirando vantagem da melhor velocidade, administração centralizada e outras vantagens do LTSP.

» Próximo: [Usando dispositivos de armazenamento locais](#)

Quando falo em "dispositivos de armazenamento", estou falando em CD-ROMs (tanto CD-ROMs IDE, quanto USB), pendrives (e HDs ligados na porta USB) e disquetes. Por enquanto ainda não são suportados gravadores de CD nas estações, mas nada impede que os usuários nas estações gravem CDs usando o gravador instalado no servidor. Em alguns ambientes, isso pode ser até desejável, pela questão do controle.

O LTSP 4.2 utiliza o módulo fuse e o udev para permitir acesso aos dispositivos nas estações. O fuse é um módulo que permite montar sistemas de arquivos usando um login normal de usuário, ao invés do root, enquanto o udev cuida da detecção de pendrives e outros dispositivos conectados na porta USB.

Comece instalando os pacotes "**fuse-utils**", "**libfuse2**" e "**libx11-protocol-perl**", que contém os utilitários usados:

```
# apt-get install fuse-utils libfuse2 libx11-protocol-perl
```

O passo seguinte é verificar se o módulo fuse está disponível. Ele vem incluído por padrão a partir do Kernel 2.6.14, de forma que muitas distribuições atuais (incluindo o Kubuntu 6.0 e o Ubuntu 5.10 em diante) já o trazem instalado:

```
# modprobe fuse
```

Caso você esteja usando uma distribuição antiga, ainda baseada no Debian Sarge (a versão anterior ao Etch) pode instalá-lo usando o module-assistant, disponível via apt-get:

```
# apt-get install module-assistant fuse-source auto-install
# module-assistant fuse
# modprobe fuse
```

(note que o pacote "fuse-source" não está disponível no Etch e nas versões atuais do Ubuntu, esta receita é apenas para distribuições baseadas no Sarge)

Em qualquer um dos dois casos, adicione a linha "fuse" no final do arquivo **"/etc/modules"**, de forma que ele seja carregado durante o boot:

```
# echo 'fuse' >> /etc/modules
```

Crie em seguida o arquivo **"/etc/fuse.conf"**, contendo a linha "user_allow_other":

```
# echo 'user_allow_other' > /etc/fuse.conf
```

Para que os usuários tenham acesso aos dispositivos, é necessário adicionar cada um ao grupo "fuse", de modo que eles tenham permissão para usá-lo. Normalmente você faria isso usando o comando "adduser", como em:

```
# adduser joao fuse
```

Fazer isso manualmente para cada usuário não é viável em um servidor com muitos usuários já cadastrados. Você pode usar o script abaixo para cadastrar todos os usuários de uma vez:

```
#                                     cd                               /home
# for i in *; do adduser $i fuse; done
```

Falta agora apenas instalar o pacote "ltsp-localdev". Ele é o mesmo pacote necessário para ativar o suporte a swap. Se ainda não está com ele instalado, acesse o <http://ltsp.mirrors.tds.net/pub/ltsp/utils/>, baixe o pacote "ltsp-server-pkg-debian_0.1_i386.deb" (a versão para distribuições derivadas do Debian) e instale-o via apt-get:

```
#          dpkg      -i      ltsp-server-pkg-debian_0.1_i386.deb
(veja a observação sobre erros na instalação no tópico sobre swap)
```

Isso conclui a configuração do servidor. Abra agora o arquivo **"/opt/ltsp/i386/etc/lts.conf"**, onde vamos adicionar a configuração dos clientes, que consiste em duas linhas. A primeira é a genérica "LOCAL_STORAGE = Y", enquanto a segunda indica o módulo que será carregado (no cliente) a fim de ativar o suporte a USB.

Existem três opções possíveis. Em placas-mãe recentes, com portas USB 2.0, é usado o módulo "ehci-hcd". Em placas antigas, é usado o módulo "ohci-hdc" ou "uhci-hcd". Teste os três até encontrar o que funciona. O "ohci-hcd" é o que funciona na maioria das placas.

As duas linhas vão dentro da sessão referente a cada estação, como em:

[ws001]			
XSERVER	=		via
X_MOUSE_PROTOCOL	=		"IMPS/2"
X_MOUSE_DEVICE	=		"/dev/input/mice"

X_MOUSE_RESOLUTION	=	400
X_MOUSE_BUTTONS	=	5
X_ZAxisMapping	=	"4
LOCAL_STORAGE	=	5"
MODULE_01 = ohci-hcd		Y

Depois de fazer todas as alterações, reinicie as estações e faça o teste. Originalmente é usado um ícone em .svg do Gnome para os ícones dos dispositivos no desktop, o que faz com que eles apareçam com ícone fa folha em branco. Para trocar o ícone por outro, abra o arquivo "/usr/sbin/lbus_event_handler.sh" e substitua a linha:

```
ICON=${FOLDER_ICON:-gnome-fs-directory.svg}
```

Por outra com um ícone de sua preferência (os ícones disponíveis vão na pasta /usr/share/icons), como em:

```
ICON="hdd_unmount.png
```

» Próximo: [Usando o som nas estações](#)

Existem duas formas de permitir que os clientes nas estações utilizem aplicativos com som. A primeira solução é simplesmente permitir que utilizem a placa de som do servidor, o que pode ser útil em ambientes pequenos e com poucas estações. Nesse caso, apenas o servidor tem caixas de som e os sons são reproduzidos de forma "pública". Todo mundo ouve.

Para isso, você não precisa mudar a configuração do LTSP, nem dos aplicativos, apenas verificar as permissões de acesso dos dispositivos de som. O default na maioria das distribuições é que apenas o root tem permissão de utilizar a placa de som através de uma sessão remota. Os usuários normais por default podem usar apenas localmente. Isso é compreensível, imagine a bagunça que seria os 40 usuários remotos de um certo servidor querendo usar a placa de som ao mesmo tempo?

No entanto, nada impede que você dê permissão para alguns usuários utilizarem a placa de som, ou mesmo dar permissão para todos os usuários (recomendável apenas para servidores com poucas estações). Para isso, basta editar as permissões de acesso dos arquivos **"/dev/dsp"** (a placa de som propriamente dita) e **"/dev/mixer"** (para ajustar o volume).

Você pode, por exemplo, criar um grupo "som", incluir o root, junto com os demais usuários autorizados no grupo e dar permissão de acesso de leitura e escrita no arquivo para o grupo.

Se você preferir que todo mundo tenha acesso, então basta usar os comandos:

#	chmod	666	/dev/dsp
# chmod 666 /dev/mixer			

A segunda opção é ativar o compartilhamento do som no LTSP, o que permite usar a placa de som e as caixinhas instaladas localmente em cada estação. O LTSP 4.2 traz um conjunto completo de drivers de som, dentro da pasta "/opt/Ltsp/i386/lib/modules/2.6.17.3-ltsp-1/kernel/sound/oss". Você pode experimentar também os drivers alsas, mas, dentro da minha experiência, os drivers OSS oferecem melhores resultados em conjunto com o LTSP.

Em seguida vem o servidor de som, que permite que o servidor da rede envie o fluxo de áudio que será reproduzido pela placa instalada na estação. Aqui temos duas opções, usar o **ESD** (eSound) ou o **NAS**. Infelizmente, o LTSP ainda não suporta o Arts, que é usado por padrão pelos programas do KDE e pode ser utilizado pela maioria dos demais aplicativos via configuração.

Como nem todos os aplicativos funcionam corretamente em conjunto com o ESD ou o NAS, dois servidores bem mais antigos e limitados, muitos aplicativos realmente não vão conseguir reproduzir som nas estações, mesmo que tudo esteja corretamente configurado. Comece com o XMMS, que suporta bem ambos os servidores e, depois de verificar que o som está funcionando, comece a testar os demais programas.

Para ativar o compartilhamento do som, adicione as linhas abaixo na configuração de cada estação, dentro do arquivo "**/opt/Ltsp/i386/etc/lts.conf**":

[ws001]		
SOUND	=	Y
SOUND_DAEMON	=	esd
VOLUME	=	80
SMODULE_01	=	sound
SMODULE_02 = auto		

A opção "SMODULE_01 = auto" faz com que o LTSP tente detectar a placa de som na estação durante o boot. A detecção funciona em boa parte das placas de som PCI, mas em muitos casos é necessário especificar os módulos necessários manualmente.

Este é um exemplo que ativa uma placa Sound Blaster ISA na estação:

SMODULE_01	=	sound
SMODULE_02	=	uart401
SMODULE_03 = sb io=0x220 irq=5 dma=1		

Este ativa uma placa de som ISA com chip Cristal cs423x, outro modelo comum em micros抗igos:

SMODULE_01	=	sound
SMODULE_02 = cs4232		

Este exemplo ativa o som onboard das placas baseadas no chipset nForce, um exemplo de placa mais atual que não é detectada automaticamente. A mesma configuração pode ser usada também em placas e notebooks com chipset Intel, que também utilizam o módulo "i810_audio":

```
SMODULE_01 = sound
SMODULE_02 = i810_audio
```

Para uma placa Sound Blaster Live PCI:

```
SMODULE_01 = sound
SMODULE_02 = emu10k1
```

A maioria das placas de som onboard usam os módulos "ac97", "sis7019" ou "via82cxx_audio", de acordo com o chipset usado. As placas Creative Ensoniq (off-board) usam o módulo "es1371".

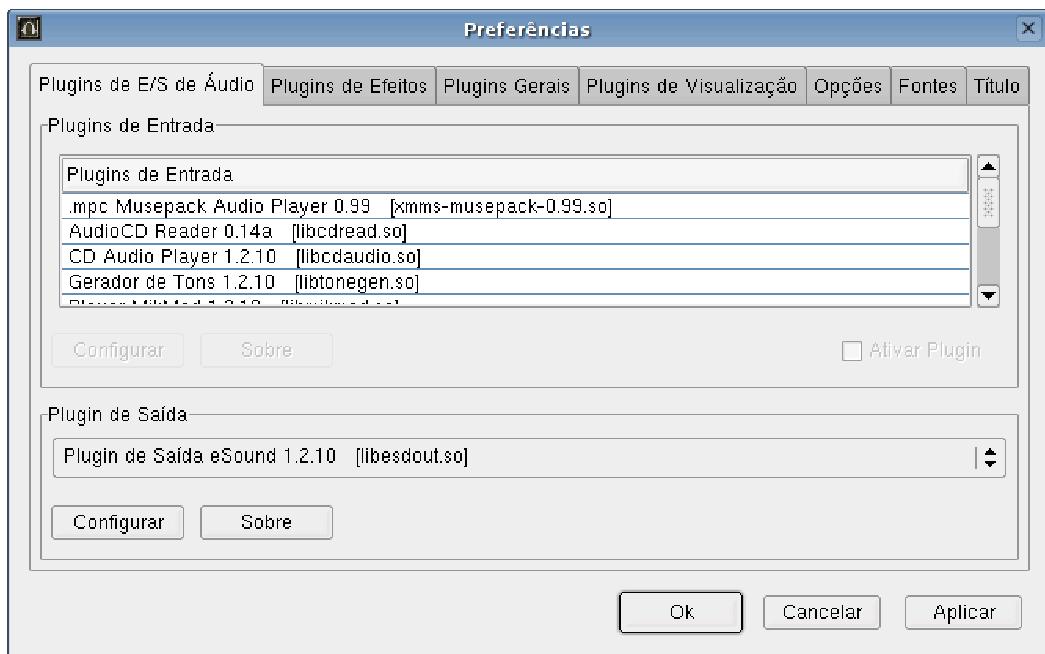
Veja que a configuração é simples. Você precisa apenas indicar o módulo "sound", que habilita o subsistema de som do Kernel, seguido pelo módulo que ativa a placa de som instalada, dentro da configuração de cada estação no arquivo lts.conf.

Um script executado pelo cliente LTSP durante o boot, o "rc.sound" configura as variáveis de ambiente, de forma que os aplicativos dentro da sessão gráfica enviem o fluxo de áudio para a estação correta, mesmo ao utilizar o som em várias estações simultaneamente.

A opção "SOUND_DAEMON = esd" determina qual dos dois servidores de som será usado. O **ESD** é o que funciona melhor na maioria dos casos, pois ele é o servidor de som padrão do Gnome e por isso muitos programas oferecem suporte a ele. Note que o simples fato de um programa oferecer a opção de usar o ESD não significa que ele realmente vá funcionar no LTSP. Muitos programas utilizam um conjunto limitado de funções do ESD, que permitem apenas o uso local e, em outros, podem simplesmente existir bugs diversos que impeçam seu uso ou façam o som ficar falhado. É preciso testar e configurar manualmente cada programa.

Três exemplos de programas que funcionam bem em conjunto com o ESD são o XMMS, o Kaffeine e o Gaim, enquanto outros não funcionam nem com reza brava, insistindo em tentar usar a placa de som do servidor (como muitos dos programas do KDE, que utilizam o Arts) ou travando ao tentar usar o som (como o Mplayer).

Procure nas configurações do programa uma opção relacionada ao servidor de som. No XMMS, por exemplo, vá em "Opções > Preferências > Plugins de E/S de Áudio > Plugin de saída > Plugin de Saída eSound".



O Kaffeine pode ser configurado para usar o ESD em "Configurações > Parâmetros do Motor Xine > Audio"; troque a opção "Auto" por ESD. No caso do Gaim, a opção está em "Ferramentas > Preferências > Sons > Método > ESD".



A segunda opção é usar o **NAS**, que é suportado por menos programas que o ESD, mas ainda assim uma opção. Na verdade, o único programa que encontrei que funciona perfeitamente com ele foi o Mplayer, depois de configurado para usá-lo como servidor de som. Para usá-lo, mude a opção "SOUND_DAEMON = esd" para "SOUND_DAEMON = nasd" na configuração das estações.

Para que ele funcione, você ainda precisará instalar os pacotes "nas", "nas-bin" e "audiooss" no servidor. Eles podem ser instalados via apt-get:

```
# apt-get install nas
# apt-get install nas-bin
# apt-get install audiooss
```

Na verdade, precisamos do pacote "nas" apenas para copiar o executável para dentro da árvore do LTSP, já que ele não foi incluído no pacote ltsp_nas:

```
# cp -a /usr/bin/nasd /opt/ltsp/i386/usr/X11R6/bin/
```

O **Arts**, o servidor de som do KDE, oferece a opção de trabalhar em conjunto com o NAS ou com o ESD em sua aba de configuração, dentro do Painel de Controle do KDE, seção Som & Multimídia > Sistema de som > Hardware > Dispositivo de áudio". Porém, o suporte deve ser ativado durante a compilação, o que é incomum entre as distribuições.

Ao invés de perder tempo com isso, o melhor é que você simplesmente desabilite o Arts, desmarcando a opção "Habilitar o sistema de som" na aba "Geral". Isso evita que ele fique travando ao tentar ativar o ESD ou NAS, bloqueando a placa de som nas estações.

O grande problema em utilizar a placa de som dos terminais é a utilização da banda da rede. Ao ouvir um mp3, por exemplo, o fluxo de áudio é processado no servidor e enviado de forma já decodificada para a estação. Isso significa que o MP3 é transmitido pela rede na forma de um fluxo de som descomprimido, similar a um arquivo .wav.

Cada estação tocando um MP3 consome cerca de 150 KB/s (ou 1.2 megabits) da banda da rede. Em uma rede de 100 megabits, a banda consumida não chega a ser um grande problema, mas faz com que exista um tráfego constante na rede, que aumenta a latência, fazendo com que a atualização de vídeo nos terminais (que é a aplicação prioritária) torne-se cada vez mais lenta. Se o som for usado simultaneamente em 10 estações, isso começa a tornar-se um problema.

Se algum dos usuários baixar um filme em boa resolução e resolver assisti-lo no terminal, as coisas começam a ficar realmente feias, pois novamente o vídeo é decodificado no servidor e transmitido de forma descomprimida através da rede. Em um vídeo de 640x480, com 16 bits de cor e 24 quadros por segundo, temos um fluxo de dados de 14 MB/s (ou 112 megabits), mais do que uma rede de 100 megabits pode fornecer. Isso significa que um único cliente que resolver assistir vídeos, pode consumir sozinho toda a banda disponível na rede.

Ao ativar o som nos terminais, escolha cuidadosamente alguns aplicativos que sejam realmente necessários e que funcionem em conjunto com o NAS ou o ESD, deixe-os pré-configurados e remova os demais aplicativos que não funcionem corretamente, ou que possam utilizar muita banda da rede, com destaque para os players de vídeo, como o Kaffeine, Totem e Gmplayer.

» Próximo: [Usando a impressora nos terminais](#)

Assim como no caso da placa de som, qualquer impressora instalada no servidor fica disponível para uso nas estações. Basta configurar a impressora da maneira usual, de forma que todos os usuários possam imprimir nela. Isso atende à maioria dos casos, já que em qualquer rede de médio porte é normal ter um servidor de impressão, com uma ou duas impressoras instaladas e compartilhadas com a rede.

Contudo, se necessário, é possível configurar o LTSP para permitir o uso de impressoras conectadas às estações. Nesse caso, entra em ação o módulo "lp_server", que faz a estação desempenhar a função de um JetDirect, como se fosse um pequeno servidor de impressão, compartilhando a impressora conectada a ele com a rede. O overhead, nesse caso, é pequeno, o servidor de impressão não atrapalha as funções normais da estação.

Para ativar o módulo de impressão, adicione as linhas abaixo na seção referente à estação, dentro do arquivo "`/opt/ltsp/i386/etc/lts.conf`".

Para uma impressora paralela:

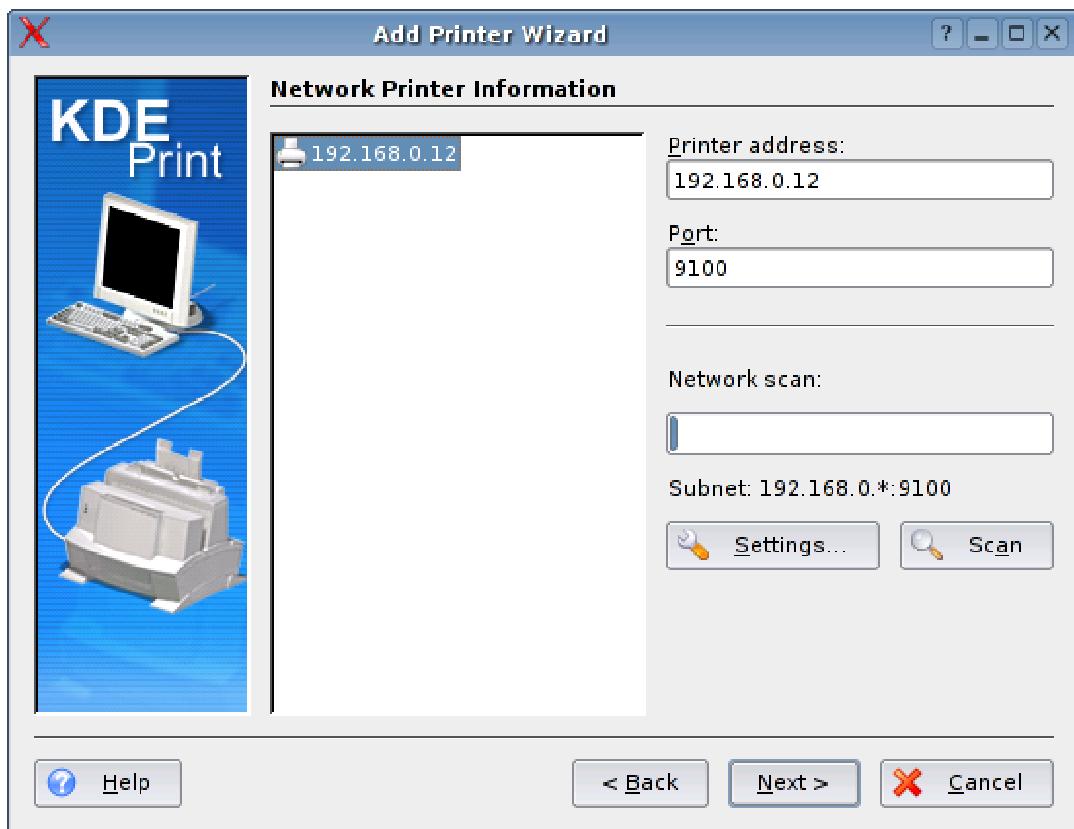
```
[ws001]
PRINTER_0_DEVICE          =
PRINTER_0_TYPE = P          = /dev/lp0
```

Para uma impressora USB (ao usar o Kernel 2.6 do LTSP 4.2):

```
[ws001]
MODULE_01          =
MODULE_02          =
PRINTER_0_DEVICE          =
PRINTER_0_TYPE = S          = ohci-hcd
                           = usblp
                           = /dev/usb/lp0
```

A configuração no LTSP se resume a carregar os módulos de Kernel necessários para ativar a impressora e indicar a porta a que ela está conectada. Isso faz com que a porta da impressora fique disponível para acesso a partir do servidor.

O próximo passo é usar o **Kaddprinterwizard** ou outro utilitário de configuração de impressora, para instalar a impressora no servidor. Procure por uma opção "Network Printer" (como no Kaddprinterwizard) ou "JetDirect Printer". Os compartilhamentos do LTSP aparecem na rede exatamente da forma como a impressora apareceria caso estivesse ligada a um JetDirect da HP, usando inclusive a mesma porta, a 9100.



O restante é a instalação normal de uma impressora de rede, em que você precisa indicar o driver e as configurações da impressora. Note que a impressora é realmente configurada apenas no servidor. A instância do LTSP rodando na estação não se preocupa com isso, ele simplesmente se limita a criar um spool remoto e enviar para a impressora os dados e instruções já formatados pelo servidor.

» Próximo: [Placas de rede ISA](#)

Ao utilizar micros antigos, você vai em muitos casos se deparar com placas de rede ISA. Lembre-se de que todas as placas ISA trabalham a 10 megabits, por isso o ideal é trocá-las por placas PCI sempre que possível. De qualquer forma, quando não for possível substituir as placas, você pode configurá-las no LTSP adicionando algumas linhas extras na configuração do dhcp, especificando o módulo usado pela placa (você já pesquisou sobre isso para gerar o disquete do rom-o-matic, lembra? :).

Comece abrindo o arquivo **"/etc/dhcp3/dhcpd.conf"**. Antes de mais nada, descomente (ou inclua) estas duas linhas, que serão as duas primeiras linhas do arquivo:

```
option          option-128        code      128      =      string;
option option-129 code 129 = text;
```

Mais abaixo, dentro da seção referente à estação, você deverá adicionar mais duas linhas, mantendo as anteriores:

```
host                               ws001          {  
hardware                         ethernet       00:E0:06:E8:00:84;  
fixed-address                     192.168.0.11;  
filename                          "/tftpboot/lts/2.6.17.3-ltsp-1/pixelinux.0";  
option                            option-128     e4:45:74:68:00:00;  
option  option-129  "NIC=3c509  MOPTS=nolock,ro,wsize=2048,rsize=2048";  
}
```

Substitua o "**3c509**" pelo módulo da placa de rede usada (o 3c509 é o módulo para a 3COM 509, uma das placas ISA mais comuns). Não altere o "e4:45:74:68:00:00"; este não é um endereço MAC, mas sim uma string que ativa a linha com o módulo da placa.

O "MOPTS=nolock,ro,wsize=2048,rsize=2048" (dica do Jorge L.) é uma configuração, transmitida ao driver da placa. Ela é necessária nas versões recentes do LTSP, pois força o sistema a utilizar buffers de dados de 2 KB para a placa de rede. Sem ela, diversas placas de rede ISA, como a própria 3C509 não funcionam.

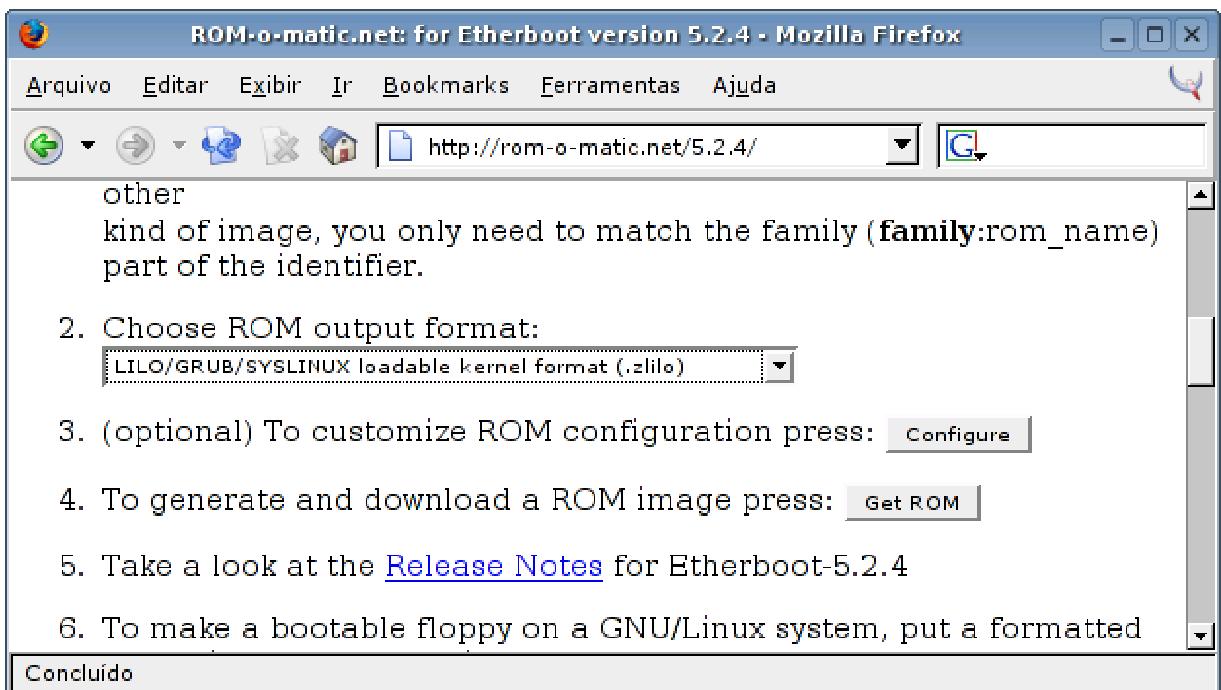
Se você estiver usando uma daquelas placas NE 2000 antigas (que no Linux são ativadas pelo módulo "ne"), onde ainda é preciso especificar o endereço de I/O usado pela placa, você deve incluí-lo na linha logo depois do módulo, como em:

```
option          option-129      "NIC=ne        IO=0x300  
MOPTS=nolock,ro,wsize=2048,rsize=2048";
```

» Próximo: [Usando um HD para boot dos clientes](#)

Em muitos casos pode ser que você queira usar um HD antigo nos clientes, ao invés do drive de disquetes ou chip de boot na placa de rede. Como os HDs costumam ser mais confiáveis que os drives de disquetes, pode ser mais interessante usá-los se você já tiver alguns à mão.

O primeiro passo é ir no <http://rom-o-matic.net> para baixar as imagens de boot. A diferença é que ao baixar as imagens para gravar os disquetes você escolhe a opção "Floppy Bootable ROM image" e ao gravar uma imagem no HD você usa a "**Lilo/Grub/Syslinux Loadable Kernel Format**".



A forma mais fácil de gravar os arquivos é instalar o HD no servidor ou outra máquina com alguma distribuição Linux instalada, que utilize o lilo como gerenciador de boot. Se o HD da estação for instalado como "**hdc**" (master da segunda porta IDE), por exemplo, os passos seriam os seguintes:

1- Usar o **fdisk** ou outro particionador para limpar o HD e criar uma única partição Linux, formatada em **ext2** com 10 MB. Na verdade, o tamanho não importa muito, pois a imagem de boot que iremos gravar tem apenas alguns kbytes.

2- Formate a partição criada com o comando:

```
# mkfs /dev/hdc1
```

3- Crie um diretório qualquer e use-o para montar a partição criada:

```
# mkdir /mnt/hdc1  
# mount /dev/hdc1 /mnt/hdc1
```

4- Copie o arquivo baixado do rom-o-matic.net para dentro da partição montada:

```
# cp eb-5.2.4-rtl8139.zlilo /mnt/hdc1
```

5- Copie os arquivos /boot/boot.b e /boot/map da instalação atual para dentro da partição. Você pode também copiar estes dois arquivos de um disquete bootável como o tomsrbd ou outra distribuição que preferir. O importante é que eles estejam dentro da partição:

```
# cp /boot/boot.b /mnt/hdc1  
# cp /boot/map /mnt/hdc1
```

6- Agora vem a etapa final, que é a gravação do lilo no HD da estação (dica do howto "booting LTSP workstations from a hard drive"):

```
# echo image=/mnt/hdc1/eb-5.2.4-rtl8139.zlilo label=ltsp | lilo -C -b \
/dev/hdc -i /mnt/hdc1/boot.b -c -m /mnt/hdc1/map
```

Preste atenção ao digitar as opções "**image=/mnt/hdc1/eb-5.2.4-rtl8139.zlilo**", "**-b /dev/hdc**", "**/mnt/hdc1/boot.b**" e "**/mnt/hdc1/map**". Elas devem ser substituídas pelas localizações corretas, no seu caso.

» Próximo: [Curiosidade: usando um terminal realmente antigo](#)

Juntando algumas peças velhas que estavam jogadas, acabei montando um velho 486 SX 25 com 8 pentes de 1 MB e uma placa de vídeo VESA tão antiga quanto o resto. Como não tinha mais um gabinete, ele acabou virando um amontoado de peças.



O mais interessante é que, apesar de tudo, esse monte de sucata funcionou como terminal. Só precisei gravar o disquete com o boot para a placa 3com509 no rom-o-matic.net.

Como ele utiliza uma placa de rede ISA, precisei adicionar as duas linhas dentro da configuração da estação no arquivo "/etc/dhcp3/dhcpd.conf", como vimos há pouco:

```
option option-128 e4:45:74:68:00:00;
option option-129 "NIC=3c509";
```

A configuração da placa de vídeo foi a parte mais complicada, pois ela não funciona com a detecção automática do vídeo (acontece com a maioria das placas ISA ou VLB). A melhor

configuração que encontrei foi usar o driver "vesa" com 8 bits de cor (funciona tanto a 800x600 quanto a 1024x768). Existe também a opção de usar o driver "vga", mas não é muito agradável de trabalhar a 640x480 com 16 cores.

Segundo a página de compatibilidade do X, ela talvez funcionasse com o driver "trident" (aparece como não-testado) que me daria um melhor desempenho, mas não funcionou.

A placa também funciona usando 16 bits de cor com o driver "vesa", mas as cores ficam trocadas, talvez por defeito na placa. Precisei também configurar o mouse serial, ligado na COM1. Como estou usando um micro com apenas 8 MB de RAM, é necessário ativar também o swap. No LTSP 4.2 a configuração seria:

```
[ws002]
XSERVER = vesa
X_MODE_0 = 800x600
X_VERTREFRESH = 60 #(Refresh rate)
X_COLOR_DEPTH = 8 #(Bits de Cor)
X_MOUSE_PROTOCOL = "Microsoft"
X_MOUSE_DEVICE = "/dev/ttyS0"
X_MOUSE_RESOLUTION = 400
X_MOUSE_BUTTONS = 2
X_MOUSE_EMULATE3BTN = Y
USE_NBD_SWAP = Y
```

Os últimos segredos estavam no próprio setup da placa-mãe. Tive que ativar o cache L1 e L2 (o padrão nesta placa é eles ficarem desativados!) e ativar o Video BIOS Shadow. Essa opção não tem efeito se você estiver usando um driver adequado para a placa de vídeo, mas ao utilizar o driver VESA genérico a própria placa fica responsável por processar as instruções, fazendo com que a ativação do Video BIOS Shadow chegue a representar um desempenho de mais de 100% para a velocidade do vídeo.

Sem o cache e sem o Video BIOS Shadow, o desempenho desse micro era ridículo, ele demorava mais de 5 segundos pra montar uma tela, mas depois das alterações ele ficou brutalmente mais rápido, o suficiente para fazer algo útil.

Em geral, vale bem mais a pena usar placas um pouco mais novas, que já tenham pelo menos slots PCI. Mas colocar essas velharias para funcionar não deixa de ser um passatempo, que mostra a versatilidade do LTSP.

» Próximo: [Capítulo 10: Configurando um servidor de e-mails](#)

adicionalmente, o **Sendmail** é o servidor de e-mails mais conhecido, não apenas no Linux, mas nos sistemas Unix em geral. Ele é um dos mais antigos (disponível desde 1982, mais

de uma década antes da popularização da Internet) e foi a opção padrão de 9 em cada 10 administradores de sistemas durante muito tempo.

Apesar disso, o uso do Sendmail vem decaindo de forma estável de uma década para cá. As queixas podem ser resumidas a duas questões fundamentais. A primeira é o brutal número de opções e recursos disponíveis, que tornam a configuração bastante complexa e trabalhosa. Muitos administradores da velha guarda gostam da complexidade, mas a menos que você pretenda dedicar sua via à arte de manter servidores Sendmail, ela acaba sendo um grande problema.

A segunda questão é o histórico de vulnerabilidades do Sendmail que, na melhor das hipóteses, pode ser definido como "muito ruim". É verdade que nos últimos anos as coisas melhoraram bastante, mas as cicatrizes do passado ainda incomodam.

O concorrente mais antigo do Sendmail é o **Exim**, que oferece um conjunto bastante equilibrado de recursos, boa performance e um bom histórico de segurança. O EXIM é o MTA usado por padrão no Debian, ele é instalado automaticamente como dependência ao instalar pacotes que necessitem de um servidor de e-mails, mas pode ser rapidamente substituído pelo Postfix ou o Sendmail via apt-get, caso desejado.

O **Qmail** é uma escolha mais complicada. Quando foi lançado, em 1997, o Qmail trouxe várias inovações e um design bastante simples e limpo, com ênfase na segurança, o que o tornou rapidamente uma opção bastante popular.

Entretanto, o Qmail possui dois graves problemas. Ele foi abandonado pelo autor em 1998, depois do lançamento da versão 1.03 e, embora o código fonte seja aberto, a licença de uso impede a redistribuição de versões modificadas, embora seja permitido disponibilizar patches.

Ao longo dos anos, surgiram várias iniciativas de atualizações do Qmail, onde o código original é distribuído junto com um conjunto de patches com atualizações. Para instalar, você precisa primeiro aplicar cada um dos patches, para em seguida poder compilar e instalar o Qmail. Dois dos projetos mais populares são o <http://qmail.org/netqmail/> e o <http://www.qmailrocks.org/>.

Embora o Qmail ainda possua uma legião de seguidores fiéis, a limitação imposta pela licença acaba sendo um grande empecilho para quem deseja utilizá-lo e representa uma grande ameaça à manteneabilidade dos patches a longo prazo, já que as alterações em relação ao código original tornam-se cada vez mais complexas e difíceis de aplicar, com a disponibilização de patches para patches que já são patches para outros patches.. ;).

Finalmente, temos o **Postfix**. Ele é uma espécie de meio termo entre a simplicidade do Qmail e a fartura de recursos do Exim. Entre os três, ele é o mais rápido e o mais simples de configurar, o que faz com que ele seja atualmente o mais popular e o que possui mais documentação disponível. O Postfix também possui um excelente histórico de segurança, sendo considerado por muitos tão seguro quanto o Qmail.

Existem fortes motivos para não usar o Sendmail ou o Qmail em novas instalações, mas temos uma boa briga entre o Postfix e o Exim. Escolhi abordar o Postfix aqui simplesmente por que, entre os dois, ele é mais popular, o que torna mais simples encontrar documentação e conseguir ajuda quando tiver dúvidas.

Apesar disso, a maior parte dos conceitos podem ser usados também na configuração do Sendmail e outros servidores; afinal, a configuração de todos eles reserva mais semelhanças que diferenças.

» Próximo: [Instalando o Postfix](#)

O pacote do Postfix pode ser encontrado em todas as principais distribuições. Nas distribuições derivadas do Debian, você pode instalá-lo usando o apt-get:

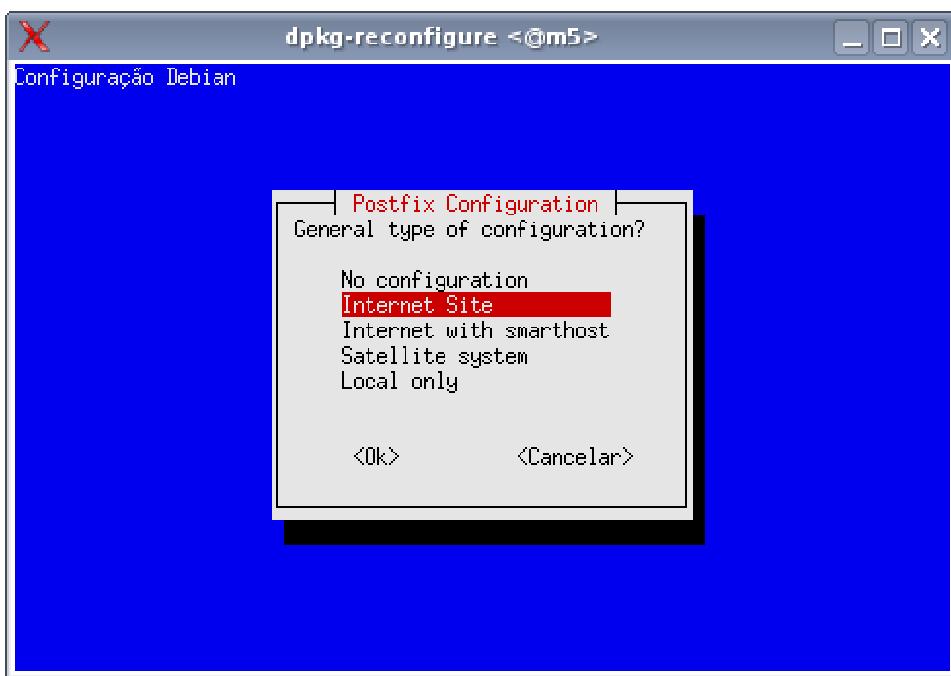
apt-get install postfix

Mais três pacotes que adicionam algumas funcionalidades importantes são:

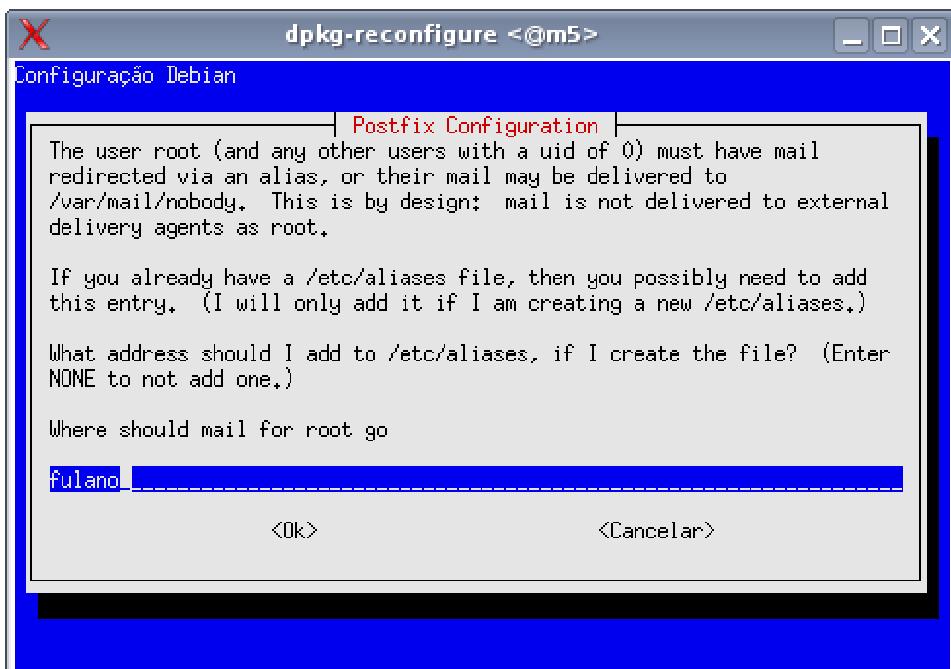
```
#           apt-get      install      postfix-ldap
(permite configurar o servidor para obter a lista de logins e senhas a partir de um servidor
LDAP)
#           apt-get      install      postfix-mysql
#           apt-get      install      postfix-pgsql
(para usar um servidor MySQL ou Postgree para armazenar a lista de logins e senhas)
```

O pacote do Debian possui um wizard configuração, exibido durante a instalação do pacote. Ele faz algumas perguntas e se encarrega de gerar uma configuração básica, suficiente para colocar o servidor para funcionar. Ele não faz nada de sobrenatural, apenas ajusta o "/etc/postfix/main.cf" de acordo com as opções definidas. Por enquanto, vou apenas explicar rapidamente as opções, pois as veremos com mais detalhes ao estudar a configuração manual do postfix.

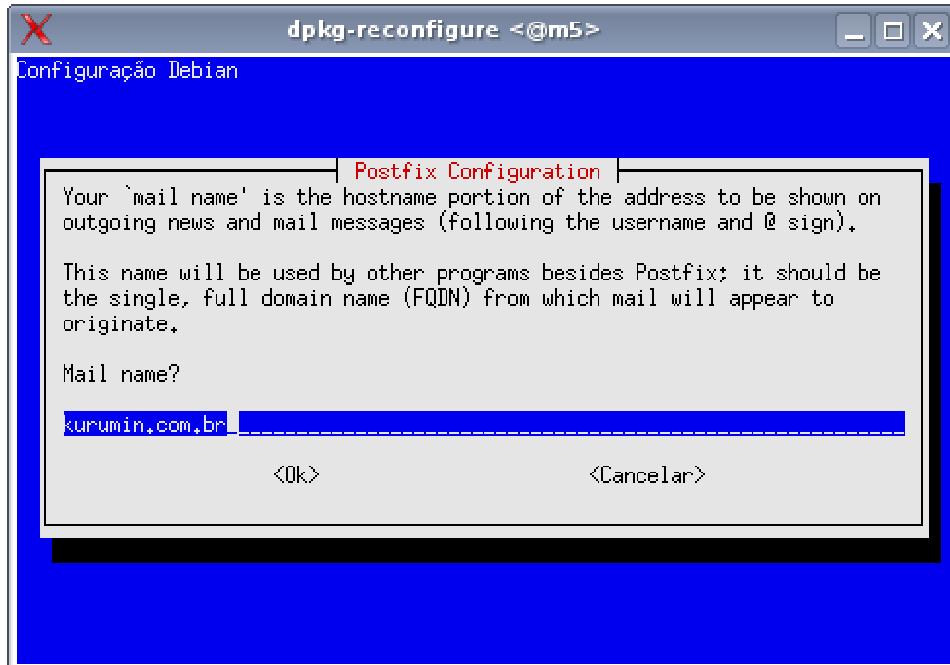
A primeira pergunta é sobre a função do servidor de e-mails que você está configurando. A opção mais usada é "Internet Site", onde você cria um servidor "de verdade", que envia e recebe os e-mails diretamente. Na segunda opção "with smarthost" seu servidor recebe mensagens, mas o envio fica a cargo de outra máquina, enquanto na terceira ("Satellite system", a mais limitada) seu servidor envia através de outra máquina e não recebe mensagens. A opção "Local only" é usada apenas em redes de terminais leves (poderia ter alguma utilidade num servidor LTSP, por exemplo), pois permite apenas que os usuários logados no servidor troquem e-mails entre si.



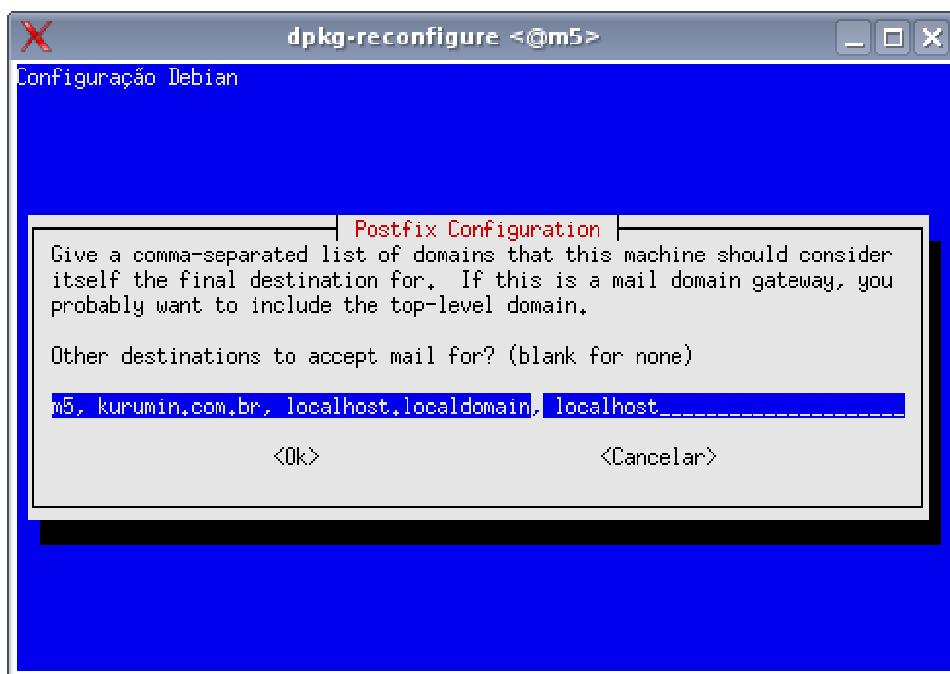
Nos sistemas Linux, é recomendado que você use a conta root apenas para a manutenção do sistema. Mesmo sendo o administrador, você usa uma conta normal de usuário, utilizando o su ou sudo para ganhar privilégios de root apenas quando necessário. Se você quase nunca usa a conta root, significaria que os e-mails enviados para o "root@seu-servidor" nunca seriam lidos. A segunda pergunta mata a questão, permitindo que os e-mails sejam encaminhados para sua conta de usuário:



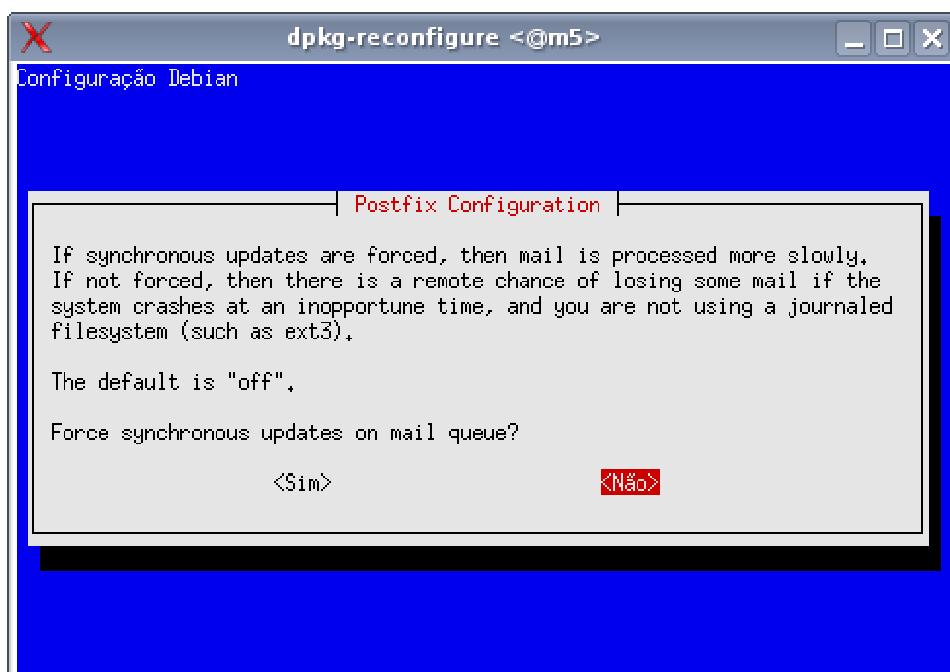
A terceira pergunta é sobre o domínio do servidor, que será incluído nas mensagens enviadas. Se o você está configurando um servidor dedicado use seu domínio registrado. Se está apenas configurando um servidor de testes, pode usar o nome da máquina:



A questão seguinte já é um pouco mais complexa. Você deve definir os destinos que serão aceitos pelo seu servidor, ou seja, os endereços que colocados no destinatário da mensagem fazem ele entender que o e-mail é para ele. Aqui você usa o nome da máquina, o domínio registrado (no caso de um servidor real), "localhost.localdomain" e "localhost", todos separados por vírgula e espaço. Esta forma, qualquer e-mail destinado ao "fulano@kurumin.com.br", "fulano@m5" (o nome da máquina) ou "fulano@localhost", que chegue até seu servidor, será encaminhado para a caixa postal do usuário "fulano". Em compensação, um e-mail destinado ao "ciclano@guiadohardware.net", por exemplo, será repassado ao servidor responsável pelo domínio correspondente.



A opção "synchronous updates" permite desativar as otimizações no envio das mensagens, fazendo com que os e-mails sejam enviados conforme são recebidos e em ordem. Esta opção aumenta um pouco a confiabilidade do servidor, pois reduz a possibilidade de perda de mensagens ainda não enviadas, em casos de travamentos ou quedas de energia. Por outro lado, ela reduz substancialmente o desempenho do servidor, por isso nunca deve ser ativada em servidores de grande volume.



Depois de concluída a instalação, o servidor já estará iniciado e configurado para subir automaticamente durante o boot. Em algumas distribuições, como no Mandriva, o servidor é configurado para subir durante o boot, mas não fica ativado depois da instalação, para que você tenha a chance de revisar o arquivo de configuração antes de ativá-lo. Neste caso, você precisa iniciar o servidor manualmente usando o comando "service postfix start", ou "/etc/init.d postfix start".

O servidor SMTP escuta, por padrão, na porta 25. Os e-mails são transmitidos de uma forma bem simples, com comandos de texto. Uma forma de entender como isso funciona é mandar um e-mail interno para o root do sistema, usando o telnet.

Sim, os servidores SMTP podem ser acessados via telnet, basta mandar o cliente se conectar na porta 25. Isso permitirá enviar o e-mail de testes conversando direto com o servidor Postfix. Se o IP do servidor na rede interna for 192.168.1.33, por exemplo, o comando seria:

```
$ telnet 192.168.1.33 25
```

```
Trying                                         192.168.1.33...
Connected                                     to                               192.168.1.33.
Escape                                           character                         is                               '^]'.
220     kurumin                           ESMTP                            Postfix          (Debian/GNU)
HELO
250
MAIL                                         From:                           eu@eu-mesmo.com
250                                         Ok
RCPT                                         to:                             joao@localhost
250                                         Ok
DATA
354     End                               data                            with               <CR><LF>.<CR><LF>
Vai      ver                                se                               estou              na                   esquina!
.
250     Ok:                                 queued                           as                               8CEDB2215
QUIT
221
Connection closed by foreign host.
```

As linhas em negrito são os comandos executados no terminal, seguidos pelas respostas do servidor. O comando "HELO" serve para iniciar a conversa, onde o emissor se identifica. Os passos seguintes são dizer o emissor do e-mail (MAIL From:) e o destinatário (RCPT to:), seguido pelo texto do e-mail (DATA). Note que depois do texto vem uma linha com um ponto, que indica o final da mensagem.

No caso, enviei um mail com remetente falso para o usuário "joao" da máquina (joao@localhost). Este e-mail local pode ser lido usando um cliente de modo texto, como o mutt:

```
# apt-get install mutt
```

Da próxima vez que você se logar com o usuário especificado, verá uma mensagem avisando da polida mensagem que foi enviada:

You have new mail.

Chamando o mutt, você verá que e-mail realmente está lá.

Antigamente, antes da popularização da internet, esses e-mails locais eram comuns, pois geralmente várias pessoas usavam o mesmo servidor (e cada servidor possuía vários terminais burros ligados a ele). As mensagens eram trocadas diretamente entre os servidores e armazenadas no spool. Quando o usuário se logava, tinha acesso à sua caixa postal.

Hoje em dia, pouca gente ainda utiliza o mutt. Em geral usamos servidores POP3 ou IMAP para armazenar as mensagens e as baixamos de vez em quando usando algum cliente de e-mails gráfico. A idéia continua sendo basicamente a mesma, mas agora em escala muito maior. Cada e-mail enviado passa por vários servidores antes de chegar ao destinatário, as mensagens são armazenadas no servidor POP3 ou IMAP do servidor e, quando o destinatário se conecta, baixa todas as mensagens de uma vez.

O Postfix (ou Qmail ou Sendmail) armazena as mensagens em uma pasta local, por padrão a pasta "Maildir", dentro do diretório home de cada usuário. Programas como o Mutt acessam diretamente as mensagens dentro da pasta, mas, para baixar as mensagens remotamente, via pop3 ou imap, você precisa instalar um componente adicional.

Existem vários servidores pop3, como o Cyrus e o Courier. O Courier é o mais usado, pois inclui vários componentes adicionais. Ele é, na verdade, uma suíte, que inclui até mesmo webmail.

Para instalar o módulo pop3, instale o pacote:

apt-get install courier-pop

Aproveite para instalar também a encriptação via ssl. Este recurso é importante hoje em dia, pois sem encriptação, seus e-mails (incluindo o login e senha) são transmitidos em texto plano pela rede e podem ser interceptados. Uma vez ativo o recurso no servidor, basta marcar a opção no cliente de e-mails.

apt-get install courier-pop-ssl

Para instalar o servidor imap, instale os pacotes:

```
#           apt-get           install           courier-imap
# apt-get install courier-imap-ssl
```

Com esta configuração básica você já conseguirá enviar e receber e-mails. Inicialmente, você pode testar pedindo para alguém enviar um e-mail para seu endereço IP, como em: fulano@200.220.123.32. Se tudo estiver funcionando, o próximo passo é configurar o

servidor DNS (como vimos anteriormente) para que você possa receber e-mails através do seu domínio registrado.

Não é uma boa idéia receber e-mails usando uma conexão ADSL, pois uma conexão instável fará com que alguns e-mails sejam perdidos. Outro problema é que quase todas as faixas de endereços de conexões via ADSL nacionais fazem parte de listas negras de spam (justamente por já terem sido exaustivamente usadas para envio de spam no passado). Nesse caso, é melhor configurar seu servidor como um sistema satélite, onde é usado um servidor SMTP externo para envio de e-mails. Você pode usar o próprio SMTP do provedor, ou o servidor de uma empresa de hospedagem, que tenha o nome "limpo" na praça.

De qualquer forma, nada impede que você registre uma conta em um serviço de DNS dinâmico, como o <http://no-ip.com> e experimente manter seu servidor de e-mails para fins didáticos.

Ao usar os pacotes **courier-pop-ssl** ou **courier-imap-ssl**, é necessário gerar um certificado. Empresas como a Verisign vendem certificados reconhecidos, que são necessários caso você queira abrir um site de comércio eletrônico, por exemplo. Mas, para um servidor particular, não existe nada errado em gerar seu próprio certificado. Ele vai funcionar da mesma forma e, se corretamente gerado, com a mesma segurança. O único efeito desagradável é que os clientes receberão uma mensagem "Não é possível comprovar a autenticidade do certificado..." ao se conectar.

Para criar uma chave para o servidor **IMAP**, comece renomeando a chave de testes, criada durante a instalação:

```
#                                     cd                               /etc/courier
# mv imapd.pem imapd.pem.old
```

Edita agora o arquivo "**imap.conf**" (na mesma pasta), colocando as informações sobre o país, cidade, nome do servidor, etc. Depois, basta gerar o novo certificado com o comando:

```
# mkimapdcert
```

Para gerar a chave para o servidor **POP3**, o procedimento é quase o mesmo. Dentro da pasta "/etc/courier" remova ou renomeie o arquivo "pop3d.pem", edite o arquivo "**pop3d.cnf**", colocando as informações do servidor e gere o certificado com o comando "**mkpop3dcert**".

» Próximo: [Cadastrando usuários](#)

Lendo a documentação, parece que cadastrar usuários no servidor de e-mails é muito complicado, pois existem muitas formas de fazer isso. A forma mais simples é simplesmente criar um novo usuário no sistema, usando o comando adduser, como em:

adduser joao

Desde que o servidor de e-mails esteja instalado, será criada a conta joao@servidor, acessível tanto localmente (usando o mutt e outros clientes), quanto remotamente, via pop3 ou imap.

O problema é que o usuário joão passa a poder logar-se na máquina de outras formas, via ssh, telnet, acessar compartilhamentos de rede e assim por diante. Essa abordagem serve para servidores internos, onde os usuários são conhecidos ou funcionários da mesma empresa, mas não é um sistema adequado para um grande servidor web, com inúmeras contas de e-mails de usuários desconhecidos ou que hospeda um servidor Apache com vários subdomínios.

Hoje em dia existem várias outras opções para cadastrar contas no servidor de e-mails, sem precisar necessariamente criar logins válidos no sistema. Você pode armazenar as contas em um servidor MySQL, Postgree SQL ou até mesmo em um servidor LDAP. Para isso, usamos os pacotes **postfix-ldap**, **postfix-mysql** ou **postfix-pgsql**, que vimos anteriormente.

» Próximo: [Configurando](#)

Antes de começar a falar sobre a configuração do Postfix, é importante que você entenda alguns termos usados com freqüência nos arquivos de configuração e tutoriais:

MTA (Mail Transport Agent): É o servidor de e-mails propriamente dito, com o Postfix, Qmail, Sendmail e o Exim. Um MTA obrigatoriamente suporta enviar e receber e-mails via SMTP, o protocolo utilizado para transportar as mensagens entre os servidores. O servidor pode ser configurado para enviar e receber os e-mails diretamente (internet site) ou se limitar a receber mensagens, usando um servidor SMTP externo (smarthost) na hora de enviar. Normalmente, você configura seu servidor como "internet site" apenas ao utilizar um servidor dedicado, ou caso sua empresa possua um link dedicado, com um IP "limpo", fora dos cadastros das listas negras de spam (você pode checar através do <http://rbls.org/>).

Mua (Mail user agent): Este é o nome técnico do cliente de e-mail, como o Thunderbird, Evolution, Kmail, etc. usados diretamente pelo usuário final.

MDA (Mail Delivery Agent): O MDA funciona como um intermediário entre o MTA e o Mua. Ele não é obrigatório, mas pode fazer algumas coisas úteis, como aplicar filtros antispam, remover vírus anexados nas mensagens, encaminhar para outros endereços e assim por diante. Dois exemplos usados no Linux são o Fetchmail e o Procmail. Você os utiliza quando precisa baixar as mensagens do provedor e aplicar filtros diversos antes de encaminhá-las aos usuários.

O principal arquivo de configuração do Postfix é o `"/etc/postfix/main.cf"`. Este é um exemplo de arquivo de configuração funcional. Veja que, apesar da complexidade da tarefa, a configuração do Postfix é relativamente simples:

```
# /etc/postfix/main.cf

myhostname          =      etch.kurumin.com.br
mydomain            =      kurumin.com.br
append_dot_mydomain =      no
alias_maps          =      hash:/etc/aliases
alias_database       =      hash:/etc/aliases
myorigin             =      /etc/mailname
mydestination        =      etch.kurumin.com.br, kurumin.com.br, localhost
relayhost            =
mynetworks          =      127.0.0.0/8
home_mailbox         =      Maildir/
mailbox_command      =
recipient_delimiter =      +
inet_interfaces      =      all
inet_protocols       =      all
message_size_limit   =      20000000
mailbox_size_limit   =      0
```

Vamos a uma explicação mais detalhada de cada uma das opções:

Embora não seja citado no arquivo, o postfix roda utilizando uma conta com privilégios limitados, de forma a limitar o dano no caso de qualquer problema de segurança relacionado ao servidor de e-mails). Estas opções já vem configuradas por padrão ao instalar o pacote.

As primeiras linhas do arquivo indicam o nome da máquina e o domínio. Caso seu servidor não tenha um domínio registrado, ou é usado apenas dentro da rede local), você pode usar o "localdomain" como domínio. Note que muitos servidores rejeitam mensagens enviadas por servidores sem domínio registrado, para dificultar o envio de spans. É por isso que é tão importante configurar corretamente o DNS reverso no Bind, já que é através dele que os servidores remotos podem verificar se os e-mails realmente vêm do seu domínio.

A opção "myhostname" deve conter o nome completo do servidor, incluindo o domínio, enquanto que a opção "mydomain" contém apenas o domínio, sem o nome da máquina, como em:

```
myhostname          =      etch.kurumin.com.br
mydomain            =      kurumin.com.br
append_dot_mydomain =      no
```

A linha "mydestination" Esta linha indica quais nomes e domínios serão considerados endereços locais pelo servidor. Se o nome do servidor é "kurumin.kurumin.com.br" e o "domínio "kurumin.com.br", o servidor entenderia que tanto e-mails endereçados a "usuario@etch.kurumin.com.br", quanto "usuario@kurumin.com.br" e "usuario@localhost" são endereçados a ele mesmo.

```
mydestination = etch.kurumin.com.br, kurumin.com.br, localhost
```

A linha "mynetworks" especifica os endereços ou faixas de endereços a partir de onde o servidor aceitará o envio de mensagens.

É preciso configurar esta opção com muito cuidado, caso contrário um spammer poderá usar seu servidor para enviar mensagens não solicitadas, consumindo sua banda e possivelmente fazendo seu servidor ser incluído em várias blacklists, o que vai lhe causar muita dor de cabeça.

A opção 'mynetworks = 127.0.0.0/8' permite apenas e-mails enviados localmente. Você pode especificar várias faixas de endereços separando-os com vírgula, como em: "mynetworks = 200.221.149.0/24, 127.0.0.0/8".

```
mynetworks = 127.0.0.0/8  
inet_interfaces = all
```

Na opção "relayhost" você pode indicar um servidor SMTP externo, através do qual as mensagens serão enviadas. Deixando esta opção em branco, todos os e-mails serão enviados diretamente pelo seu servidor.

Hoje em dia é bem mais simples usar um servidor externo por causa da questão do spam. Usar o smtp de um provedor conhecido fará com que menos mensagens se percam nos filtros dos destinatários.

Para usar um relayhost aqui, é preciso indicar um servidor que aceite mensagens enviadas por este servidor sem pedir autenticação. Em geral, as empresas que oferecem serviços de hospedagem oferecem esta opção em troca de uma taxa adicional. É possível também configurar seu provedor para se autenticar, com um pouco mais de trabalho.
Ex: relayhost = smtp.meuprovedor.com

Opcionalmente, você pode configurar os clientes de e-mail nas estações para usarem diretamente o smtp do provedor, deixando seu servidor postfix apenas para receber. Nesse caso, você pode usar qualquer smtp a que tenha acesso.

```
relayhost =
```

A linha "home_mailbox" indica a pasta local, dentro do home de cada usuário, onde os e-mails ficarão armazenados. A pasta Maildir/ é o padrão usado por diversos MTA's. Caso necessário, crie a pasta manualmente, usando o comando "maildirmake ~/Maildir" (executado como o usuário para o qual a pasta será criada). Em seguida, execute o comando "maildirmake /etc/skel/Maildir" como root, para que todos os novos usuários criados a partir daí já venham com a pasta criada. Normalmente, os pacotes instalados pelas distribuições automatizam esta etapa.

```
home_mailbox = Maildir/  
recipient_delimiter = +  
inet_interfaces = all
```

Na maioria dos casos, é desejável limitar o tamanho das mensagens recebidas, para evitar que algum espertinho envie um ISO de CD anexado à mensagem, consumindo toda a banda e acabando com o espaço em disco do servidor. O padrão do postfix é limitar as mensagens a 10 MB. Qualquer anexo maior que isso é recusado. Esta configuração pode ser alterada através da opção "message_size_limit", onde você especifica o valor desejado, em bytes. Note que por causa do uso do MIME, o tamanho dos anexos cresce substancialmente ao serem enviados via e-mail. Um arquivo de 5 MB, transforma-se numa mensagem de quase 7. Leve isto em conta ao definir o limite. Aqui estou usando um limite de 20 MB decimais:

```
message_size_limit = 20000000
```

A opção "mailbox_size_limit" serviria para definir o limite de armazenamento para a caixa postal do usuário. Entretanto, ao usar o formato Maildir para as caixas postais, cada mensagem é salva num arquivo separado, de forma que a opção não funciona. Por isso, usamos o valor "0" para desativá-la. A melhor forma de limitar o espaço dos usuários é simplesmente definir quotas de espaço em disco, usando o Quota.

```
mailbox_size_limit = 0
```

Em geral, os arquivos de configuração padrão, incluídos nas distribuições, são suficientes para ter um servidor de e-mails básico funcional. Mas, depois de feito o primeiro teste, nunca deixe de editar o arquivo, verificando todas as opções. Você pode tanto usar como ponto de partida o arquivo original, quanto usar este modelo.

Com o tempo, o ideal é que você desenvolva um arquivo próprio, com as opções que você usa mais freqüentemente e comentários que lhe ajudem a lembrar como e em quais situações usar cada uma. Lembre-se de que, salvo eventuais diferenças entre as versões instaladas, um arquivo de configuração usado no Fedora ou Mandriva vai funcionar perfeitamente no Debian, Slackware, ou em qualquer outra distribuição que siga um nível mínimo de padrões. O software em si, o Postfix, será o mesmo, independentemente da distribuição usada.

» Próximo: [Instalando um webmail](#)

O Squirrelmail é um script de webmail escrito em php, que permite acessar as mensagens de um servidor imap via web. Ele é bem leve, tanto do ponto de vista dos recursos utilizados no servidor, quanto do ponto de vista dos clientes. As páginas geradas pelo webmail são simples páginas html, sem javascript nem nenhum outro recurso especial. Isso o torna um campeão de compatibilidade, principalmente com os navegadores usados em PDAs e browsers antigos.

Para instalar o Squirrelmail você vai precisar do seguinte:

- 1- Um servidor Postfix (ou outro MTA suportado), com suporte a IMAP, o que inclui basicamente os pacotes "postfix" e "courier-imap". Siga as instruções anteriores para instalar o servidor de e-mails e criar as contas de usuários.

2- Um servidor Apache, com suporte a PHP4 instalado. Tanto faz usar o Apache 1.3 ou o Apache 2, o Squirrelmail roda em ambos, verifique apenas se o suporte a PHP está realmente instalado e funcionando.

Satisfeitos esses dois pré-requisitos, o Squirrelmail em si é bem simples de instalar. Você tem duas opções. Pode instalá-lo usando o gerenciador de pacotes da sua distribuição ou baixar o arquivo manualmente. A principal vantagem em usar o pacote incluído na distribuição é que a instalação é feita com checagem de dependências, o que é uma segurança a mais contra eventuais barbeiragens na configuração do Postfix ou do Apache.

No Debian, por exemplo, você pode instalá-lo usando o apt-get:

```
# apt-get install squirrelmail
```

O Squirrelmail é instalado por padrão dentro da pasta "**/usr/share/squirrelmail**", que fica fora da jurisdição do servidor web. Existem várias formas de fazer com que o webmail fique acessível apesar disso. Você pode, por exemplo, criar um link dentro da pasta "**/var/www/**" apontando para a pasta "**/usr/share/squirrelmail**". Mas, uma forma mais elegante de ter o mesmo resultado, é adicionar as duas linhas abaixo no arquivo "**/etc/apache2/httpd.conf**":

```
Alias /webmail "/usr/share/squirrelmail/"  
DirectoryIndex index.php
```

Aqui estamos criando uma pasta virtual "webmail/" no servidor web, que aponta para o arquivo index.php dentro da pasta real.

Ao baixar manualmente, pegue o arquivo no <http://www.squirrelmail.org/download.php> e copie o conteúdo do arquivo para uma pasta dentro do seu servidor web, como, por exemplo, "**/var/www/webmail**"; basta descompactar o arquivo, como no caso do phpBB. Não é necessário compilar nada. Opcionalmente, você pode usar a pasta "**/usr/share/squirrelmail**" e adicionar a entrada do alias no arquivo de configuração do Apache.

Depois de instalar, é preciso fazer a configuração básica do Squirrelmail, usando o utilitário "**squirrelmail-configure**". Se você instalou a partir do pacote, pode chamá-lo diretamente (como root) a partir do terminal. Se tiver instalado manualmente, execute o script "**configure**", dentro da pasta do Squirrelmail.

The screenshot shows a terminal window titled "xterm". The title bar also displays "SquirrelMail Configuration : Read: config.php (1.4.0)". The main menu is displayed as follows:

```
SquirrelMail Configuration : Read: config.php (1.4.0)
-----
Main Menu --
1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages
D. Set pre-defined settings for specific IMAP servers
C Turn color on
S Save data
Q Quit

Command >> █
```

A configuração mínima inclui:

- a) Defina o nome da empresa, logotipo, URL, etc. na opção **1**.
- b) Defina o domínio do servidor (ex: minhaempresa.com) na opção **2**. Se você está configurando um servidor local, sem usar um domínio registrado, mantenha o default.
- c) Ainda na opção **2**, verifique se as configurações de servidor estão corretas. Elas devem ser:
 - 3. Sendmail or SMTP : SMTP
 - A. Update IMAP Settings : localhost:143 (other)
 - B. Update SMTP Settings : localhost:25

O Squirrelmail pode ser configurado para utilizar outros servidores. Você pode usar o Sendmail (ou Qmail) no lugar do Postfix ou utilizar outros servidores IMAP além do Courier. As configurações acima são as que se aplicam no nosso caso, usando o postfix e o courier-imap.

- d) Acesse a opção **D** no menu e indique qual servidor IMAP está utilizando. Lembre-se de que no exemplo estamos usando o Courier. Definir corretamente o servidor usado aqui permite que o Squirrelmail ative uma série de otimizações para cada servidor específico, que melhoraram consideravelmente o desempenho. Ao terminar, use a opção **S** para salvar e **Q** para sair.

O Squirrelmail não é um serviço, ele é apenas uma aplicação que roda dentro do Apache. Depois de instalar os arquivos, acesse o endereço "http://127.0.0.1/webmail" e você verá a tela de login do Squirrelmail.



Se algo der errado neste ponto, verifique a instalação do Apache, se o suporte a PHP está realmente funcionando (lembre-se de que além de instalar o pacote, é necessário incluir a linha "LoadModule php4_module /usr/lib/apache/1.3/libphp4.so" ou similar no arquivo de configuração do Apache) e se a pasta de instalação, o link ou a entrada no arquivo de configuração para vincular a pasta webmail/ com a pasta onde estão os arquivos estão corretos.

Depois de logado, faça alguns testes, verificando se consegue mandar e-mails para contas em outros servidores e se consegue mandar e-mails de um usuário local para outro. Se o servidor já estiver disponível na internet, experimente enviar um e-mail (a partir de outra conta) para ele, usando inicialmente o endereço IP (ex: joao@200.220.123.32) e depois o nome de domínio registrado (joao@minhaempresa.com).

Caso a tela de login funcione, mas você tenha problemas ao logar, verifique as permissões de acesso da pasta de instalação do Squirrelmail, veja se ela não está com permissão de leitura apenas para o root. Lembre-se de que na maioria das distribuições o Apache roda sob o usuário "apache" ou "daemon" e não sob o usuário root, o que é inseguro.

Verifique também a configuração do Squirrelmail, veja se o serviço "**courier-imap**" está realmente inicializado. Observe que no Squirrelmail o login de acesso é apenas o nome do usuário (ex: joao) e não o endereço de e-mail completo. Ao configurar um servidor simples, onde as contas de acesso do sistema são usadas no servidor de e-mail, as senhas também são as mesmas.

Em algumas distribuições, depois de instalar o pacote courier-imap, é necessário rodar o comando "pw2userdb" para que as contas de usuários do sistema sejam corretamente

incluídas como logins de acesso no servidor de e-mails. Verifique isso e reinicie o courier-imap novamente.

Uma última pegadinha é que, para que o servidor IMAP funcione, é necessário que exista um diretório chamado "Maildir" dentro do home de cada usuário, onde são armazenadas as mensagens. Este diretório contém uma estrutura própria e é criado usando o comando "maildirmake". Normalmente ele é criado automaticamente ao instalar os pacotes usados nas distribuições. Mas, em algumas, este procedimento precisa ser feito manualmente. É mais uma coisa que pode dar errado.

Se isso for necessário no seu caso, comece criando o diretório para o seu próprio usuário, ou o que for usar para testar o webmail:

\$ maildirmake ~/Maildir

Execute agora o comando que cria a pasta dentro do diretório /etc/skel, de forma que os diretórios home de todos os novos usuários criados daqui em diante já sejam criados com ele:

```
# maildirmake /etc/skel/Maildir
```

» Próximo: [Autenticando os clientes](#)

Originalmente, o Postfix determina os clientes que estão autorizados a enviar e-mails através do seu servidor de acordo com a configuração da linha "mynetworks", dentro do arquivo main.cf. Usando a linha "mynetworks = 127.0.0.0/8" ou "mynetworks = 127.0.0.1" o Postfix aceita apenas e-mails enviados a partir do próprio servidor, uma configuração ideal se os usuários enviam os e-mails através de um webmail instalado no próprio servidor, sem SMTP externo.

Você pode também permitir o envio a partir de qualquer micro da rede local, usando algo como "mynetworks = 192.168.0.0/24". O problema surge quando você precisa permitir o envio de e-mails para usuários espalhados pela web, conectados via ADSL, modem ou outras modalidades de conexão com IP dinâmico.

Imagine, por exemplo, o caso de um provedor de acesso que precisa permitir que seus usuários enviem e-mails usando seu SMTP, mesmo quando eles estiverem acessando através de outro provedor.

Você não pode simplesmente permitir o envio a partir de qualquer endereço, caso contrário seu servidor vai ser rapidamente descoberto pelos spammers, que começarão a utilizar toda a sua banda para enviar suas tentadoras ofertas de produtos. Pior, depois de algum tempo, seu servidor vai acabar caindo nas listas negras de endereços usados para envio de spam, fazendo com que seus próprios e-mails válidos passem a ser recusados por outros servidores.

A solução, nesse caso, é passar a autenticar os usuários, como faz a maioria dos provedores. Usamos então o SASL, que no Debian (Etch ou Sid) pode ser instalado via apt-get:

```
# apt-get install libsasl2 sasl2-bin libsasl2-modules libdb3-util procmail
```

Depois de instalar os pacotes, abra o arquivo "**/etc/default/saslauthd**", onde vão as opções de inicialização do autenticador. O primeiro passo é substituir a linha "START=no" por:

```
START=yes
```

Adicione (ou modifique) também a linha:

```
MECHANISMS="pam"
```

Isso faz com que ele seja inicializado durante o boot e aceite a autenticação dos usuários.

Continuando, crie (ou edite) o arquivo "**/etc/postfix/sasl/smtpd.conf**" de forma que ele contenha apenas as linhas:

pwcheck_method: mech_list: plain login	saslauthd
---	-----------

O pacote do Postfix usado no Debian Etch e no Ubuntu 6.10 (ou mais recente) e em outras distribuições derivadas deles, roda dentro de um chroot (ou jaula), o que melhora bastante a segurança, impedindo que qualquer eventual problema de segurança fique restrito ao servidor de e-mails, sem afetar o resto do sistema. Você notará que dentro da pasta "/var/spool/postfix" estão não apenas os diretórios com as filas de mensagens, mas também binários e bibliotecas de que o postfix precisa para funcionar.

O problema é que de dentro do seu chroot, o Postfix não tem acesso ao saslauthd, fazendo com que a autenticação dos usuários não funcione. O próprio saslauthd é necessário por que o Postfix (mesmo ao rodar fora do chroot) não tem acesso aos arquivos de senha do sistema e por isso não é capaz de autenticar os usuários por si só.

Para resolver este problema, precisamos criar a pasta "/var/spool/postfix/var/run/saslauthd", utilizada pelo Postfix dentro do chroot e configurar o sasl para utilizá-la no lugar da pasta padrão. Desta forma, o Postfix consegue se comunicar com ele.

Este tipo de precaução de segurança parece algo complicado e desnecessário à primeira vista, mas é justamente por causa de truques como este que os servidores Linux acabam sendo tão seguros. Para começo de conversa, o Postfix é por si só bastante seguro. Mas, como os servidores de e-mail são um ponto comum de ataque, ele fica isolado dentro do chroot de forma que, mesmo na remota possibilidade de um cracker conseguir obter controle sobre o Postfix através um exploit remoto, ele não poderia fazer muita coisa. Para completar, o Postfix roda dentro de privilégios muito limitados, de forma que, mesmo que o cracker tenha muita sorte e a improvável falha de segurança no Postfix seja combinada com uma falha no sistema que o permita escapar do chroot, ele ainda assim não conseguaria fazer muita coisa. ;)

Comece criando o diretório:

```
# mkdir -p /var/spool/postfix/var/run/saslauthd
```

Abra agora o arquivo "/etc/default/saslauthd" (o mesmo onde substituímos o "START=no" por "START=yes") e substitua a linha

```
OPTIONS="-c"
```

por:

```
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"
```

Reinic peace o serviço para que as alterações entrem em vigor:

```
# /etc/init.d/saslauthd restart
```

Isso faz com que o SASL passe a utilizar o diretório dentro do chroot e o Postfix tenha acesso ao saslauthd e possa assim autenticar os usuários através dele. Note que o "/var/spool/postfix" é o diretório onde está o chroot. Esta é a localização padrão no Debian; ao usar outra distribuição, verifique se não está sendo usado outro diretório.

Só para garantir, adicione o postfix ao grupo sasl:

```
# adduser postfix sasl
```

Isso completa a configuração do SASL.

O passo seguinte é a configuração do Postfix. Abra o arquivo "/etc/postfix/main.cf" e adicione as linhas abaixo no final do arquivo. Ao reciclar um arquivo de configuração anterior, verifique se esta configuração já não foi incluída em outro ponto do arquivo:

smtpd_sasl_local_domain	=	
smtpd_sasl_auth_enable	=	yes
smtpd_sasl_security_options	=	noanonymous
broken_sasl_auth_clients	=	yes
smtpd_recipient_restrictions	=	permit_sasl_authenticated,
permit_mynetworks,		
reject_unauth_destination		
smtpd_tls_auth_only = no		

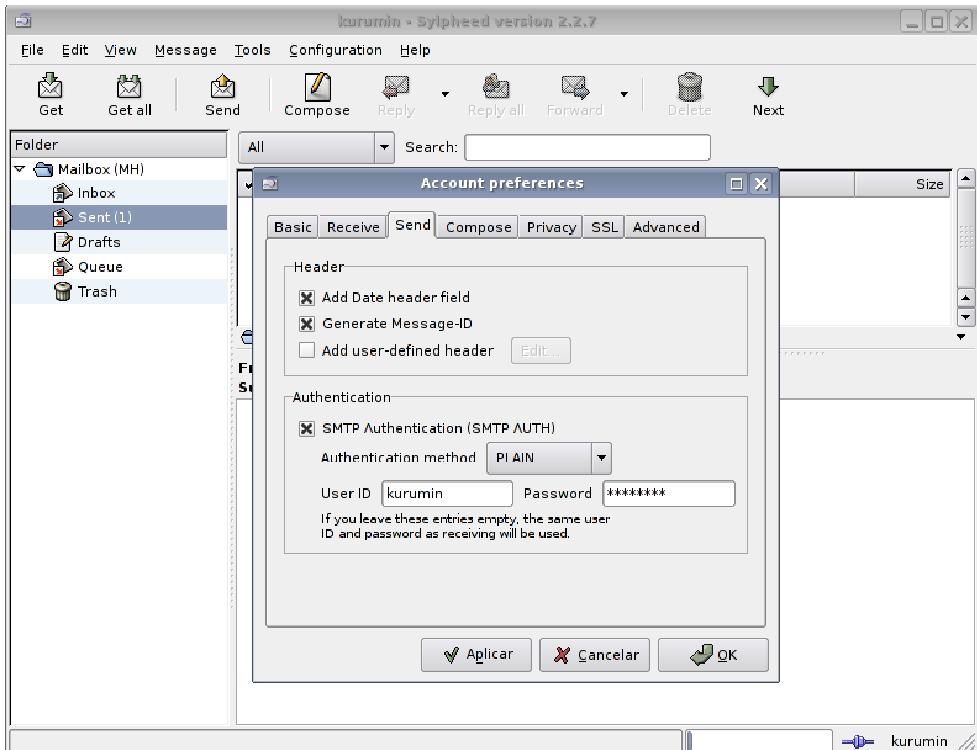
Feito isso, reinicie o Postfix para que as alterações entrem em vigor:

```
# /etc/init.d/postfix restart
```

Por enquanto, o servidor suporta apenas autenticação em texto puro, sem encriptação. Este é o sistema "clássico", ainda usado por muitos provedores de acesso, mas que possui problemas óbvios de segurança, já que alguém que consiga sniffar a rede local, poderia capturar as senhas dos usuários no momento em que eles tentassem baixar os e-mails.

Para testar, configure um cliente de e-mails qualquer para utilizar o endereço IP do servidor como SMTP (aqui estou usando o Sylpeed) e, nas configurações, ative a opção de

autenticação para o servidor SMTP e escolha a opção "PLAIN" (login em texto puro). Envie um e-mail de teste para confirmar se tudo está funcionando.



» Próximo: [Ativando o TLS](#)

O TLS (Transport Layer Security) adiciona segurança ao nosso sistema de autenticação, permitindo que os usuários possam baixar os e-mails sem medo, mesmo ao acessar a partir de redes públicas.

Em algumas distribuições (como no Debian Sarge), você precisa instalar o pacote "postfix-tls". Nas demais (incluindo o Debian Etch, que é a versão atual), ele já vem integrado ao pacote principal do Postfix.

O TLS trabalha utilizando um conjunto de chaves de encriptação e certificados, usados para criar o túnel encriptado e garantir a segurança da seção. O primeiro passo é criar este conjunto de arquivos.

Acesse o diretório "/etc/postfix/ssl" (crie-o se não existir) e rode os comandos abaixo, um de cada vez e nesta ordem. Durante a geração das chaves, será solicitado que você informe uma passphrase, uma senha que pode conter entre 4 e 8191 caracteres. Administradores

paranóicos costumam usar passphrases bem grandes, mas não exagere, pois você precisará confirmá-la algumas vezes durante o processo. Os comandos são:

```
#                               mkdir          /etc/postfix/ssl
#                               cd           /etc/postfix/ssl/
# openssl      genrsa      -des3      -rand      /etc/hosts      -out      smtpd.key    1024
#                               chmod        600
#                               req         -key      smtpd.key      -out      smtpd.key
# openssl      x509       -req      -days     730      -in      smtpd.csr      -signkey  smtpd.key      -out      smtpd.crt
# openssl      rsa        -in      smtpd.key      -out      smtpd.key.unencrypted
#                               mv         -f      smtpd.key.unencrypted      smtpd.key
# openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days 730
```

O "730" usado nas linhas determina a validade dos certificados, em dias. No caso, estou criando certificados válidos por dois anos. Depois deste prazo, os clientes começarão a receber um aviso ao se autenticarem, avisando que o certificado expirou e precisarei repetir o processo para atualizá-los. Se preferir, você pode usar um número mais alto, para gerar certificados válidos por mais tempo. Para gerar certificados válidos por 10 anos, por exemplo, substitua o "730" por "3650".

Continuando, abra novamente o arquivo "/etc/postfix/main.cf" e adicione as linhas abaixo (sem mexer nas linhas referentes ao SASL que adicionamos anteriormente):

smtp_use_tls	=	yes
smtp_tls_note_starttls_offer	=	yes
smtpd_tls_CAfile	=	/etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel	=	1
smtpd_tls_received_header	=	yes
smtpd_tls_session_cache_timeout	=	3600s
smtpd_tls_cert_file	=	/etc/postfix/ssl/smtpd.crt
smtpd_tls_key_file	=	/etc/postfix/ssl/smtpd.key
smtpd_tls_session_cache_database	=	btree:\${queue_directory}/smtpd_scache
smtp_tls_session_cache_database	=	btree:\${queue_directory}/smtp_scache
tls_random_source = dev:/dev/urandom		

Reinic peace o Postfix para que as alterações entrem em vigor:

```
# /etc/init.d/postfix restart
```

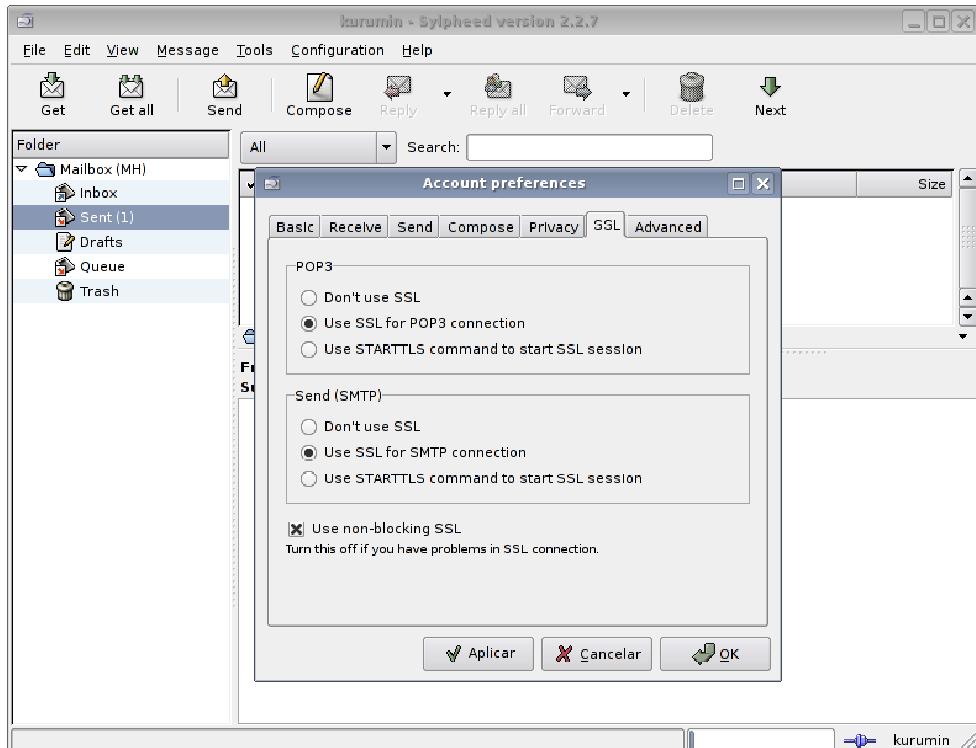
Para que os clientes consigam se autenticar no servidor, é necessário instalar o pacote "courier-authdaemon" e o "courier-ssl", além dos pacotes courier-pop, courier-pop-ssl, courier-imap, courier-imap-ssl que vimos anteriormente. Você pode usar o comando abaixo para instalar de uma vez todos os pacotes necessários:

```
# apt-get install courier-authdaemon courier-base courier-imap courier-imap-ssl \
courier-pop courier-pop-ssl courier-ssl gamin libgamin0 libglib2.0-0
```

Para testar, ative o uso do SSL para o servidor SMTP dentro das preferências do seu cliente de e-mail. No caso do Thunderbird, por exemplo, marque a opção "Usar Conexão Segura > TLS" dentro do menu "Enviar", nas configurações da conta. O cliente de e-mail exibirá alguns avisos sobre a validade do certificado, o que é normal, já que estamos utilizando um certificado "self-signed", ou seja, um certificado "caseiro", que não é reconhecido por

nenhuma autoridade certificadora. Empresas como a Verisign vendem certificados reconhecidos, mas os preços são proibitivos fora de grandes instalações.

Com o TLS, A autenticação continua funcionando da mesma forma, mas agora todos os dados são transmitidos de forma segura. Lembre-se de que ao instalar o courier, já ativamos também o suporte a SSL para o IMAP e POP3, de forma que você pode ativar ambas as opções no cliente de e-mail:



Aqui está um exemplo de arquivo **/etc/postfix/main.cf** completo, incluindo a configuração do SASL e do TLS:

```
# /etc/postfix/main.cf

myhostname          = etch.kurumin.com.br
mydomain            = kurumin.com.br
append_dot_mydomain = no
alias_maps          = hash:/etc/aliases
alias_database      = hash:/etc/aliases
myorigin             = /etc/mailname
mydestination       = kurumin.com.br, localhost
myrelayhost          =
mynetworks           = 127.0.0.0/8
myhome_mailbox       = Maildir/
mailbox_command      =
recipient_delimiter = +
inet_interfaces      = all
inet_protocols       = all
message_size_limit   = 20000000
mailbox_size_limit   = 0
```

```

smtpd_sasl_local_domain = yes
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_sasl_authenticated,
permit_mynetworks,
reject_unauth_destination
smtpd_tls_auth_only = no

smtp_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_CAfile = /etc/postfix/ssl/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
smtpd_tls_cert_file = /etc/postfix/ssl/smtpd.crt
smtpd_tls_key_file = /etc/postfix/ssl/smtpd.key
smtpd_tls_session_cache_database = btree:${queue_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${queue_directory}/smtp_scache
tls_random_source = dev:/dev/urandom

```

O arquivo **/etc/default/saslauthd** (depois de removidos os comentários), ficaria:

```

START=yes
MECHANISMS="pam"
MECH_OPTIONS=""
THREADS=5
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd -r"

```

O arquivo **/etc/postfix/sasl/smtpd.conf**, que vimos no início, continua com apenas as duas linhas:

```

pwcheck_method: saslauthd
mech_list: plain login

```

» Próximo: [Adicionando um antivírus](#)

Depois que o servidor de e-mails estiver funcionando, é interessante instalar um antivírus para proteger as estações Windows. Este é um detalhe interessante: existem vários bons antivírus para Linux, mas todos são destinados a justamente encontrar vírus for Windows em compartilhamentos de rede, páginas web e arquivos, e-mails, etc. Até hoje, o Linux (tanto como servidor, quanto desktop) tem se mantido como uma plataforma livre de vírus dignos de nota, daí a ausência de soluções neste sentido. A demanda simplesmente não existe.

Uma das melhores opções é o **Clamav**, que possui uma lista de definições atualizada com uma freqüência muito grande e oferece um recurso de atualização automática. O Clamav escaneia as mensagens que passam pelo servidor, removendo as mensagens com arquivos infectados. Ele serve tanto para proteger clientes Windows da rede, quanto para reduzir o tráfego de mensagens inúteis.

Para instalar o antivírus, basta instalar o pacote "**clamav**". Ele não costuma mudar de nome entre as distribuições. No Debian, você precisa instalar também o pacote "**clamav-daemon**", em outras distribuições este componente faz parte do pacote principal.

Para utilizar o Clamav em conjunto com o Postfix, de forma que todos os e-mails passem primeiro pelo antivírus, e só depois sejam encaminhados para as caixas postais dos usuários, é necessário instalar também o pacote "**amavisd-new**".

O Amavisd "intercepta" as novas mensagens, entregando-as ao executável do Clamav. De acordo com a configuração, as mensagens com arquivos infectados podem ser simplesmente deletadas, ou colocadas em uma pasta de quarentena. Lembre-se de que quase todas as mensagens com arquivos infectados são enviadas automaticamente pelos vírus da moda, como uma forma de se espalharem, por isso não existe muito sentido em preservá-las. O Amavisd é um software complicado de instalar manualmente, é necessário alterar vários scripts e arquivos de inicialização e configurar corretamente as permissões de várias pastas. Além de trabalhoso, o processo é muito sujeito a erros, por isso é sempre recomendável utilizar os pacotes incluídos nas distribuições, onde o trabalho já está feito.

A comunicação entre o Amavisd e o Clamav é feita automaticamente, mas é necessário configurar o Postfix para direcionar os novos e-mails para o Amavisd, para que o trio comece a trabalhar em conjunto. Para isso, é necessário adicionar as linhas abaixo no final do arquivo **"/etc/postfix/master.cf"** (note que este arquivo é diferente do main.cf que configuramos anteriormente). Estas linhas estão incluídas no arquivo **" /usr/share/doc/amavisd-new/README.postfix"**; você pode copiá-las a partir do arquivo, ao invés de escrever tudo:

```
smtp-amavis unix - - n - 2 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes

127.0.0.1:10025 inet n - n - - smtpd
-o content_filter=
-o local_recipient_maps=
-o relay_recipient_maps=
-o smtpd_restriction_classes=
-o smtpd_client_restrictions=
-o smtpd_helo_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks=127.0.0.0/8
-o strict_rfc821_envelopes=yes
-o smtpd_error_sleep_time=0
-o smtpd_soft_error_limit=1001
-o smtpd_hard_error_limit=1000
```

Adicione também a linha abaixo ao "/etc/postfix/main.cf":

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

É preciso verificar também os arquivos "/etc/clamav/clamd.conf" e "/etc/amavis/amavisd.conf". Em muitas distribuições eles são configurados corretamente ao instalar os pacotes, mas em outras é preciso fazer as alterações manualmente.

No arquivo "/etc/clamav/clamd.conf", verifique se a linha abaixo está presente e descomentada:

```
LocalSocket /var/run/clamav/clamd.ctl
```

No arquivo "/etc/amavis/amavisd.conf", verifique se as linhas abaixo estão descomentadas. Este arquivo inclui vários exemplos que permitem usar diferentes antivírus, por isso é um pouco extenso. Use a função de pesquisar do editor de textos para ir direto ao ponto:

```
### http://www.clamav.net/
['Clam Antivirus-clamd',
 \&ask_daemon,      ["CONTSCAN     {}\\n",      "/var/run/clamav/clamd.ctl"],
 qr/^bOK$/,
 qr/^bFOUND$/,
 qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

O último passo é fazer com que o Amavis tenha acesso aos arquivos de log e diretórios de trabalho do Clamav. No Debian e em muitas outras distribuições, basta adicionar o usuário "clamav" no grupo "amavis", a solução mais rápida e limpa:

adduser clamav amavis

Caso você esteja usando outra distribuição, onde essa primeira alteração não funcione, resta fazer do jeito sujo, alterando manualmente as permissões de acesso dos diretórios:

```
#      chown      -R      amavis:amavis      /var/clamav
#      chown      -R      amavis:amavis      /var/log/clamav
# chown -R amavis:clamav /var/run/clamav
```

Depois de terminar a configuração, reinicie todos os serviços, de forma que tudo entre em vigor:

```
#          /etc/init.d/postfix      restart
#          /etc/init.d/clamav-daemon      restart
("etc/init.d/clamd restart" no Fedora e Mandriva)

#          /etc/init.d/clamav-freshclam      restart
("etc/init.d/freshclam restart" no Fedora e Mandriva)

#          /etc/init.d/amavis      restart
```

Experimente tentar enviar agora um e-mail contendo um arquivo infectado qualquer para uma conta qualquer do seu servidor. O e-mail simplesmente não vai chegar ao destino. O

arquivo "/usr/share/doc/amavisd-new/README.postfix" contém uma string de texto que dispara o antivírus e pode ser usada para simular um e-mail infectado.

Checando o conteúdo do arquivo "/var/log/clamav/clamav.log", você verá uma entrada indicando que um e-mail infectado foi encontrado:

```
Thu Ago 26 19:55:49 2005 -> /var/lib/amavis/amavis-20050526T195549-  
07338/parts/part-00001: Eicar-Test-Signature FOUND
```

O próximo passo é instalar o **Spamassassin** que funciona como um filtro antispam automático, que utiliza uma blacklist com endereços IP e conteúdos de mensagem catalogados. Esta lista funciona de forma semelhante à de um programa antivírus, é atualizada pela equipe de desenvolvimento e atualizada de forma automática. Para instalar:

```
# apt-get install spamassassin
```

No Debian, por padrão, ele fica inativo depois de instalado, talvez como uma precaução para evitar consumir muitos recursos em micros onde ele é instalado acidentalmente junto com outros servidores. Para ativá-lo, edite o arquivo "**/etc/default/spamassassin**", mudando a opção "ENABLED=0" para "ENABLED=1":

```
#      Change      to      one      to      enable      spamd  
ENABLED=1
```

Para finalizar, inicie o serviço "**spamassassin**" (ou "**spamd**" em muitas distribuições):

```
# /etc/init.d/spamassassin start
```

Falta agora configurar o Amavis para utilizar também o Spamassassin ao receber novas mensagens. Agora os e-mails passarão pelos dois filtros, seqüencialmente.

Abra novamente o arquivo: "**/etc/amavis/amavisd.conf**". Na seção 1, por volta da linha 160, **comente (#)** a linha:

```
@bypass_spam_checks_acl = qw( . );
```

Essa linha desativa a checagem de spam. Ela vem descomentada por padrão, pois nem todo mundo utiliza o Amavis em conjunto com o Spamassassin. Ao comentá-la, a checagem é ativada.

Na seção 4, por volta da linha 400, procure pela linha: "**\$final_spam_destiny** =".

Essa linha configura o que será feito com as mensagens marcadas como spam. Ela tem três valores possíveis:

D_PASS – Entrega a mensagem normalmente, incluindo apenas a palavra "SPAM" no subject. Isso permite que os próprios usuários configurem o filtro local do leitor de e-mails para remover as mensagens caso estejam incomodando. Nenhum filtro

antispam é perfeito, sempre algumas mensagens legítimas acabam sendo marcadas como spam. Esta opção minimiza o problema, deixando a remoção das mensagens por conta dos usuários.

D_DISCARD – Descarta a mensagem. Esta é a opção mais usada, mas ao mesmo tempo a mais perigosa, pois mensagens "boas" marcadas como spam vão simplesmente sumir, sem deixar nenhum aviso ao remetente ou destinatário.

D_REJECT – Esta terceira opção também descarta a mensagem, mas envia uma notificação ao emissor. Isso permite que o emissor de uma mensagem "boa", acidentalmente classificada como spam, receba um aviso de que ela foi descartada e tenha a oportunidade de reenviá-la novamente de outra forma. Mas, por outro lado, como a maioria dos spams são enviados a partir de endereços falsos, isso acaba sendo um desperdício de banda.

Ao usar a opção que descarta as mensagens, a opção fica:

```
$final_spam_destiny = D_DISCARD
```

Mais adiante, na seção 7, por volta da linha 1100, você encontrará mais opções relacionadas ao Spamassassin.

Finalmente, é preciso transferir o ownership dos arquivos do Spamassassin para o Amavis (assim como no caso do Clamav), para que ele possa executar corretamente suas funções:

```
# chown -R amavis:amavis /usr/share/spamassassin
```

Não se esqueça de reiniciar os serviços com o comando:

```
# /etc/init.d/spamassassin restart  
("etc/init.d/spamd restart" em muitas distribuições)
```



```
# /etc/init.d/amavis restart
```

» Próximo: [Capítulo 11: Firewall](#)

O Linux, de uma forma geral, é relativamente imune a vírus, worms e trojans, que são a principal causa de invasões e dores de cabeça em geral no Windows. Isso não ocorre apenas porque o Windows é usado em mais máquinas e por isso um alvo maior, mas também porque os aplicativos disponíveis no Linux são, pela média, bem mais seguros.

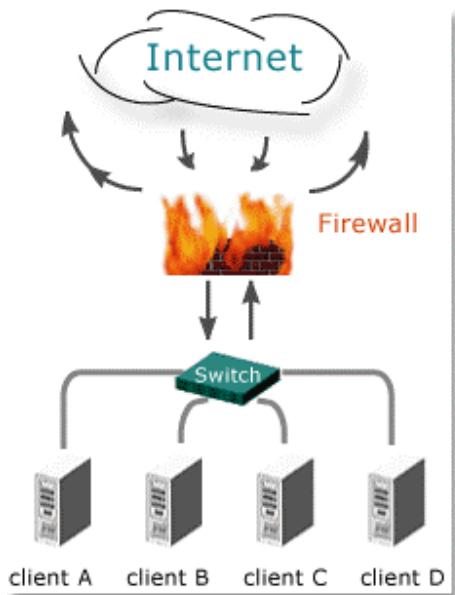
Veja o caso do Apache, por exemplo. Ele é usado em uma percentagem muito maior de servidores que o IIS. Mesmo assim, o número de falhas críticas de segurança e invasões bem-sucedidas registradas contra servidores web rodando o IIS é bem maior do que nos mais numerosos servidores Apache.

Mesmo assim, brechas de segurança podem surgir onde menos se espera. Por exemplo, em 2004 foi descoberto um buffer overflow no servidor SSH, que poderia ser usado para desenvolver um exploit. Esta brecha não chegou a ser explorada, pois, assim que a possível vulnerabilidade foi descoberta, uma correção foi rapidamente disponibilizada e a notícia se espalhou pela web. Antes que alguém tivesse tempo de escrever um exploit, a maior parte dos servidores do mundo já estavam seguros.

A moral da história: é sempre muito melhor prevenir do que remediar, e a melhor forma de se proteger contra brechas deste tipo é manter um firewall ativo, permitindo apenas acesso aos serviços que você realmente deseja disponibilizar. Reduzindo os pontos vulneráveis, fica mais fácil cuidar da atualização dos serviços expostos e, assim, manter seu servidor seguro.

Imagine o firewall como a muralha que cercava muitas cidades na idade média. Mesmo que as casas não sejam muito seguras, uma muralha forte em torno da cidade garante a segurança. Se ninguém consegue passar pela muralha, não é possível chegar até as casas vulneráveis. Se, por acaso, as casas já são seguras, então a muralha aumenta ainda mais a segurança.

A idéia mais comum de firewall é como um dispositivo que fica entre o switch (ou hub) em que estão ligados os micros da rede e a internet. Nesta posição é usado um PC com duas placas de rede (eth0 e eth1, por exemplo), onde uma é ligada à internet e outra à rede local.



O firewall aceita as conexões vindas dos micros da rede local e roteia os acessos à internet. De dentro da rede você consegue acessar quase tudo, mas todas as tentativas de conexão vindas de fora são bloqueadas antes de chegarem aos clientes.

Imagine um micro com o Windows XP, onde o sistema acabou de ser instalado, sem nenhuma atualização de segurança e com o firewall inativo. Conectando este micro na

internet diretamente, será questão de minutos até que ele comece a ser infectado por worms e se transforme em um zumbi a atacar outras máquinas ligadas a ele.

Entretanto, se houver um firewall no caminho, os pacotes nocivos não chegam até ele, de forma que ele fica em uma posição relativamente segura. Ele ainda pode ser infectado de formas indiretas, como ao acessar uma página que explore uma vulnerabilidade do IE ou ao receber um e-mail infectado através do Outlook, mas não mais diretamente, simplesmente por estar conectado à internet.

Opcionalmente, o servidor rodando o firewall pode ser equipado com um servidor Squid configurado para remover arquivos executáveis das páginas acessadas e um servidor Postfix, encarregado de bloquear mensagens infectadas, o que adiciona mais um nível de proteção.

Note que o firewall em si não protege contra vírus e trojans, mas apenas contra tentativas diretas de conexão. Ele cria uma barreira entre os micros da rede local e a internet, fazendo com que os recursos compartilhados na rede não sejam acessíveis de fora. No Linux, o firewall é incluído no próprio Kernel do sistema, na forma do Iptables, encontrado no Kernel 2.4 em diante. Isso garante um excelente desempenho e segurança em relação à maioria dos firewalls for Windows, que rodam em nível de aplicação.

Embora seja sempre mais seguro ter um servidor dedicado, você pode ter um nível de segurança muito bom simplesmente habilitando o firewall localmente. Todos os pacotes provenientes da internet passam primeiro pelo Iptables antes de serem encaminhados para os aplicativos. Por isso, um firewall local, bem configurado, garante uma segurança muito próxima à de um firewall dedicado.

» Próximo: [Escrevendo um script de firewall](#)

Existem muitos firewalls gráficos for Linux, como o GuardDog, o Shorewall e o Firestarter (que comento adiante). Eles variam em nível de facilidade e recursos, oferecendo uma interface amigável e gerando as regras do Iptables de acordo com a configuração feita. Você pode escolher entre usar o programa que melhor atenda suas necessidades ou configurar diretamente o Iptables com as regras desejadas. Neste caso, você pode formular as regras diretamente, definindo condições onde os pacotes serão aceitos ou recusados, como em:

```
# iptables -A INPUT -s 192.168.0.0/255.255.255.0 -j ACCEPT
```

Estes comandos seguem uma sintaxe comum: tudo começa com o comando "**iptables**", que é quem executará as opções incluídas no comando. Em seguida vem uma condição, indicada pela opção "-A". Neste exemplo usei "INPUT -p tcp -s 192.168.0.0/255.255.255.0", que se aplica a qualquer pacote de entrada (INPUT), utilizando o protocolo TCP (-p tcp), proveniente dos micros da rede local (192.168.0.0/255.255.255.0). Note que aqui estou especificando uma faixa de endereços e a máscara de sub-rede. No final, é preciso dizer o que fazer com os pacotes que se

enquadrem nesta situação, indicando uma ação. O "-j ACCEPT" diz que estes pacotes devem ser aceitos.

À primeira vista, isso parece bem complicado, assim como o arquivo de configuração original do Squid, com suas 3.000 e tantas linhas. Mas, as coisas ficam bem mais simples se começarmos com um script simples e formos incluindo novas regras aos poucos.

Este é um exemplo de script que pode ser usado em um desktop que simplesmente acessa a internet como cliente, sem rodar nenhum servidor, nem compartilhar a conexão com outros micros:

```
#      iptables      -A      INPUT      -i      lo      -j      ACCEPT
# iptables -A INPUT -p tcp --syn -j DROP
```

A idéia aqui é que o micro possa acessar a internet sem que ninguém de fora possa invadi-lo de forma alguma. Esses dois comandos fazem isso da forma mais simples possível.

A primeira linha orienta o firewall a deixar passar os pacotes enviados através da interface de loopback (-i lo -j ACCEPT). É importante que esta linha (ou outra com o mesmo efeito) sempre seja usada, em qualquer script de firewall que termine bloqueando todas as conexões, pois no Linux a interface de loopback é usada para comunicação entre diversos programas. Para ter uma idéia, todos os programas gráficos a utilizam para se comunicarem com o X, os programas do KDE a utilizam para trocar mensagens entre si. Sem esta regra, muita coisa deixa de funcionar corretamente.

Depois de abrir o firewall para as mensagens locais, usamos a segunda regra para bloquear todas as novas conexões vindas de fora. O "--syn" faz com que o firewall aplique a regra apenas para tentativas de abrir novas conexões (alguém tentando acessar o servidor SSH que você esqueceu aberto, por exemplo), mas sem impedir que servidores remotos respondam a conexões iniciadas por você. Isso permite que você continue navegando e acessando compartilhamentos em outros micros da rede local, com poucas limitações.

Para não precisar ficar digitando os comandos cada vez que precisar reiniciar o micro, você pode incluí-los em um dos arquivos de inicialização do sistema. Nas distribuições derivadas do Debian, você pode colocá-los no final do arquivo `/etc/init.d/bootmisc.sh` e, nas derivadas do Red Hat, no arquivo `/etc/rc.d/rc.local`.

Essas duas regras podem ser usadas como base para criar o que chamo de firewall de bloqueio. Ele segue uma idéia bastante simples: você diz as portas que gostaria de abrir e ele fecha todas as demais. Ou seja, o firewall fecha por padrão todas as portas, com exceção das que você disser explicitamente que deseja manter abertas. Isso garante uma configuração de firewall bastante segura com um mínimo de dor de cabeça.

Você pode adicionar novas regras, abrindo portas, direcionando faixas de portas para micros da rede interna, fechando portas de saída, de forma a bloquear o uso de programas como o ICQ e o MSN e assim por diante.

Imagine que você está configurando o firewall do servidor da rede. Ele tem duas placas de rede, uma para a rede local e outra para a internet. Você precisa que ele fique acessível sem limitações dentro da rede local, mas quer manter tudo fechado para quem vem da internet.

Nesse caso, você poderia usar a regra que mostrei há pouco no seu script de firewall:

```
# Abre para uma faixa de endereços da rede local  
iptables -A INPUT -s 192.168.0.0/255.255.255.0 -j ACCEPT
```

O "192.168.0.0" indica a faixa de endereços da rede local. A máscara "255.255.255.0" indica que a última parte do endereço muda, ou seja, os micros da rede local usam endereços entre 192.168.0.1 e 192.168.0.254. Tudo o que vier deles (tanto TCP, quanto UDP, já que não indicamos o protocolo) é aceito.

Note que esta faixa de endereços não é roteável, ela simplesmente não existe na internet. Não existe a possibilidade de algum engraçadinho de outro estado tentar configurar seu micro para usar esta faixa de endereços e enganar a regra do firewall.

Como uma proteção adicional, as versões recentes do Iptables são capazes de ignorar pacotes aparentemente destinados a uma interface quando eles chegam em outra. Com duas placas, onde uma está ligada à rede local (usando a faixa 192.168.0.x) e outra à Internet, o firewall não aceitará que um pacote falseado, proveniente da Internet, com endereço de emissor "192.168.0.3" (por exemplo), seja encaminhado a um micro da rede local, pois ele sabe que pacotes com este endereço de emissor devem chegar apenas pela placa ligada à rede local.

Essa mesma regra pode ser usada também para abrir o firewall para endereços ou faixas de endereços da internet. Imagine que você queira dar acesso aos micros da filial da sua empresa em Macapá, onde usam um link com o IP fixo 200.220.234.12. Você poderia abrir a faixa 200.220.234.0 ou apenas o IP 200.220.234.12, de forma que o firewall permitisse acessos vindos de lá, mas continuasse bloqueando o restante. Você pode abrir para várias faixas de endereços distintas, basta repetir a linha adicionando cada uma das faixas desejadas.

Imagine agora que este servidor foi instalado na sede de uma empresa para a qual você presta serviços. Você precisa acessá-lo de vez em quando para corrigir problemas, mas naturalmente quer fazer isso via internet, sem precisar se deslocar até lá. Você pode configurar o firewall para abrir a porta 22 usada pelo SSH adicionando a regra:

```
# Abre uma porta (inclusive para a internet)  
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Note que esta regra abre a porta 22 para todo mundo. Lembre-se do exemplo do SSH: todo servidor disponível para a internet é um risco potencial de segurança, por isso só abra as portas para os servidores que você realmente for utilizar. O ideal seria usar um par de chaves, protegidas por uma passphrase para acessar o servidor e configurá-lo para não aceitar logins com senha (apenas com chaves), como vimos no capítulo sobre SSH.

Ao abrir várias portas, você pode utilizar o parâmetro "-m multiport" para especificar todas de uma vez, separadas por vírgula, sem precisar colocar uma em cada linha. Para abrir as portas 21, 22 e 6881 (bittorrent), por exemplo, você usaria a regra abaixo:

```
#          Abre      um      conjunto      de      portas
iptables -A INPUT -m multiport -p tcp --dport 21,22,6881 -j ACCEPT
```

Se você presta suporte a partir de uma empresa que possui um link dedicado, com IP fixo, você pode tornar a regra mais específica, permitindo apenas o IP de onde você acessa:

```
#          Abre      uma      porta      para      um      IP      específico
iptables -A INPUT -p tcp -s 200.231.14.16 --dport 22 -j ACCEPT
```

Em um micro doméstico, você pode abrir também as portas usadas pelo bittorrent (6881 a 6889) ou portas usadas por jogos multiplayer, por exemplo. Para abrir um intervalo de portas, use a regra:

```
#          Abre      um      intervalo      de      portas
iptables -A INPUT -p tcp --dport 6881:6889 -j ACCEPT
```

Além de trabalhar com endereços IP, é possível criar regras baseadas também em endereços MAC. Isso permite adicionar uma camada extra de proteção ao criar regras para a rede local. Para isso, usamos o parâmetro "-m mac --mac-source", seguido pelo endereço MAC da placa do host desejado. Para permitir que o host "192.168.1.100" tenha acesso ao servidor, mas apenas se o endereço MAC da interface bater, você usaria uma regra como:

```
iptables -A INPUT -s 192.168.1.100 -m mac --mac-source 00:11:D8:76:59:2E -j ACCEPT
```

Note que agora, além do IP, especificamos o endereço MAC da placa. As duas regras são usadas em conjunto, de forma que o acesso é permitido apenas caso as duas informações estejam corretas. Isso dificulta as coisas para alguém que queira acessar o servidor trocando o IP de sua máquina. Você pode descobrir o MAC das máquinas da rede usando o próprio ifconfig ou o comando "arp -a".

Note que limitar o acesso com base no endereço MAC adiciona uma camada extra de proteção, mas não é infalível. O endereço MAC pode ser trocado de forma quase tão simples quanto o endereço IP e, sniffando a rede, é possível descobrir os endereços IP e MAC dos micros com uma certa facilidade.

No Linux, você pode trocar o endereço MAC da placa de rede usando os comandos:

```
#          ifconfig      eth0      down
#      ifconfig      eth0      hw      ether      00:11:D8:76:59:2E
# ifconfig eth0 up
```

Como vê, basta especificar o endereço desejado. O Iptables não é capaz de diferenciar máquinas com os endereços MAC falseados das reais, pois, se alguém desconectasse o micro 192.168.1.100 da rede e configurasse o seu para usar o mesmo IP e MAC, poderia acessar o servidor bipassando a regra de firewall. A única forma de ter uma segurança completa seria utilizar o SSH ou outro protocolo que utilize um algoritmo robusto de encriptação para o login e a transmissão dos dados.

Lembre-se de que o firewall é uma primeira barreira de proteção, mas não é uma garantia por si só. É preciso combiná-lo com outras camadas de segurança para ter um servidor completamente seguro.

Outra limitação é que as regras baseadas em endereços MAC podem ser usadas apenas dentro da rede local. O endereço MAC é descartado do pacote quando ele é roteado para a Internet, ficando apenas o endereço IP. Ao acessar através de uma conexão compartilhada, todos os pacotes provenientes da Internet chegam com o endereço MAC do gateway da rede.

Este é um exemplo de script completo, incluindo algumas regras adicionais para evitar ataques comuns:

```
#!/bin/bash

iniciar(){

#     Abre      para      uma      faixa      de      endereços      da      rede      local
iptables -A INPUT -s 192.168.0.0/255.255.255.0 -j ACCEPT

#     Abre      uma      porta      (inclusive      para      a      internet)
iptables -A INPUT -p tcp --dport 22 -j ACCEPT

#                                     Ignora
#                                     pings
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP

#     Protege      contra      IP      spoofing      (esta      opção      já      vem      ativada      por      padrão      na
#     maioria      das      distribuições      atuais,      mas      não      custa      ter      certeza)
echo 1 > /proc/sys/net/ipv4/conf/default/rp_filter

#     Descarta      pacotes      malformados,      protegendo      contra      ataques      diversos
iptables -A INPUT -m state --state INVALID -j DROP

#     Abre      para      a      interface      de      loopback.      Esta      regra      é      essencial      para      que
#     o      KDE      e      outros      programas      gráficos      funcionem      adequadamente.
iptables -A INPUT -i lo -j ACCEPT

#     Impede      a      abertura      de      novas      conexões,      efetivamente      bloqueando      o      acesso
#     externo      ao      seu      servidor,      com      exceção      das      portas      e      faixas      de      endereços
#     manualmente      especificadas      anteriormente.      Bloqueia      tudo.
iptables -A INPUT -p tcp --syn -j DROP

}

parar(){

iptables
echo      "Regras      de      firewall      -F
}
      desativadas"
}

case
"start")
"stop")
"restart")
*)      echo      "Use      os      parâmetros      start      ou      stop"
esac
```

A receber qualquer conexão, vinda de qualquer endereço, o firewall primeiro verifica todas estas regras, seqüencialmente, para decidir se o pacote passa ou não. Usando esse script de exemplo, teríamos o seguinte:

- Se o pacote vier da rede local, ele é aceito.
- Se o pacote for para porta 22 (do SSH), ele é aceito.
- Se for um ping, ele é recusado (de forma a dificultar um pouco para outros descobrirem que você está online).
- Pacotes danificados ou forjados (unclean) são recusados, protegendo os micros da rede interna.
- Se o pacote vier da sua própria máquina (um programa tentando mostrar alguma coisa na tela, por exemplo), ele é aceito.
- Se o pacote for uma resposta a uma conexão que você iniciou, como, por exemplo, o servidor do guiahardware.net enviando a página do site que você está acessando, ele é aceito.
- Tentativas de conexão (toda conexão TCP é iniciada por um pacote syn) fora das condições especificadas acima são descartadas pelo firewall. A conexão nem sequer chega a ser estabelecida e o emissor não recebe qualquer resposta (DROP). Ele não sabe se o pacote foi recebido ou não, fica no vácuo, o que dá a impressão de que o seu micro nem está online.

Da forma como escrevi, o script suporta as funções "start", "stop" e "restart", e pode ser usado como um serviço de sistema. Salve-o dentro da pasta "/etc/init.d", como em "/etc/init.d/firewall", e marque a permissão de execução:

```
# chmod +x /etc/init.d/firewall
```

A partir daí, você pode ativar as regras usando o comando "/etc/init.d/firewall start" e fazer com que alterações dentro do script entrem em vigor com um "/etc/init.d/firewall restart".

Se você está configurando um servidor dedicado remotamente, é importante que você teste o script antes de configurar o sistema para executá-lo automaticamente durante o boot. O motivo é simples: se houver alguma regra incorreta no script, que bloqueie seu acesso ao servidor, você poderá solicitar um reboot do servidor para que a configuração seja removida e você recupere o acesso. Entretanto, se o sistema for configurado para carregar o script durante o boot, o reboot não resolverá e você precisará abrir uma chamada de suporte, solicitando que um técnico se logue localmente no servidor e desative seu script (o que provavelmente resultará em uma taxa adicional).

Uma opção mais relaxada seria simplesmente colocar os comandos com as regras desejadas no final do arquivo "/etc/init.d/bootmisc.sh" ou "/etc/rc.d/rc.local", mas isso não é tão recomendável, pois você perde a possibilidade de reiniciar o firewall rapidamente depois de alterar as regras.

Assim como nas regras do Squid, cada pacote que chega pela rede precisa passar por todas as regras, para que o firewall possa decidir o que fazer com ele. Quando aceito por uma das regras, ele é imediatamente encaminhado ao aplicativo, sem passar pelas demais. Por isso é necessário sempre colocar as regras mais restritivas por último, de preferência concluindo o script com uma regra que bloqueia todos os pacotes de entrada.

Outra dica é que você pode incluir os comandos para compartilhar a conexão e ativar o proxy transparente (que também são regras de firewall) no script, fazendo que ele desempenhe simultaneamente as duas funções. Nesse caso, tome o cuidado de sempre colocar as regras que compartilham a conexão e ativam o proxy transparente antes das regras que bloqueiam conexões.

Este é um exemplo de script de firewall que inclui as regras para compartilhar a conexão e ativar o proxy transparente. Ao usá-lo, comente as linhas que não se aplicam à sua instalação:

Outra dica importante são os comandos usados para limpar as regras do Iptables. É necessário executá-los sempre que você fizer alterações no seu script de firewall e quiser executá-lo novamente para que as novas regras entrem em vigor. No primeiro script de exemplo, por exemplo, uso o comando "iptables -F" como parte da função "stop", que desativa o firewall. No segundo script, incluí também o "iptables -t nat -F".

iptables -F: Limpa a tabela principal do iptables, onde vão os comandos para abrir e fechar portas, que vimos até aqui.

iptables -t nat -F: Limpa a tabela nat, que é usada por regras que compartilham a conexão e fazem forwarding de portas, como por exemplo:

```
iptables -t nat -A PREROUTING -i eth0 --dport 22 -j DNAT --to-dest 192.168.1.2
```

Todas as regras do Iptables que levam "-t nat" são armazenadas nesta segunda tabela, que precisa ser zerada separadamente. A idéia é que você pode limpar as regras principais do firewall sem desabilitar o compartilhamento da conexão e vice-versa.

iptables -L: Este comando lista a configuração atual, sem alterar nada. É interessante executá-lo depois de fazer alterações na configuração do firewall, para ter certeza que as regras surtiram o efeito esperado. Para ver as regras de forwarding e compartilhamento, use também o "**iptables -t nat -L**"

» Próximo: [Forwarding de portas](#)

Você deve lembrar que, ao compartilhar uma conexão entre vários micros, apenas o servidor que está com a conexão recebe conexões vindas da internet. Os micros da rede local acessam via NAT e apenas recebem respostas para conexões iniciadas por eles.

Mas, imagine que você queira que um servidor web, escutando na porta 80 do micro 192.168.0.3 da rede local, fique disponível para a internet. Como o servidor é o único com um IP válido na internet, a única forma de fazer com que o 192.168.0.3 fique acessível é fazer com que o servidor "passe a bola" para ele ao receber conexões na porta 80. É justamente isso que fazemos ao configurar o forwarding de portas.

Uma vez feita a configuração, sempre que o servidor receber uma conexão qualquer na porta 80 (ou qualquer outra definida por você), ele a repassará para o micro 192.168.0.3. Isso é feito de forma completamente transparente, forma que o emissor nem percebe que quem respondeu à solicitação foi outro servidor.

Essa opção pode ser usada também para permitir que os micros da rede local fiquem com as portas do bittorrent abertas (de forma a baixar arquivos com um melhor desempenho), rodem servidores de games online ou qualquer outra tarefa onde seja necessária manter determinadas portas TCP ou UDP abertas. A limitação é que continua existindo uma única porta 80, uma única porta 21, etc. de forma que apenas um micro da rede interna pode receber cada porta de cada vez.

Veja um exemplo de como redirecionar as portas 6881 a 6889 usadas pelo Bittorrent para o micro 192.168.0.10 da rede local:

```
# Redireciona uma faixa de portas para um micro da rede local.
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 6881:6889 -j DNAT \
--to 192.168.0.10
iptables -t nat -A POSTROUTING -d 192.168.0.10 -j SNAT --to 192.168.0.1
```

Esta regra é um pouco mais complexa, pois trabalha em duas fases. A primeira faz com que o servidor encaminhe todas as conexões que receber na interface e porta especificada para o micro da rede local e a segunda faz com que os pacotes de resposta enviados por ele posam ser encaminhados de volta. Para que ambas funcionem, é necessário usar o comando "echo

1 > /proc/sys/net/ipv4/ip_forward", que ativa o forwarding de portas. É o mesmo comando que usamos ao compartilhar a conexão.

Nos parâmetros que coloquei em negrito, a "eth0" é a placa de internet, onde chegam os pacotes, a "6881:6889" é a faixa de portas que estão sendo redirecionadas e o "192.168.0.10" é o IP do micro dentro da rede local que passa a receber as conexões destinadas a ela. Na segunda regra, temos repetido o IP do micro na rede local e, em seguida, o "192.168.0.1" que indica o IP do servidor, dentro da rede local.

Para redirecionar uma única porta, ao invés de uma faixa, basta citar a porta sem usar os ":";, como em:

```
# Redireciona uma única porta para um micro da rede local.
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 22 -j DNAT --to 192.168.0.10
iptables -t nat -A POSTROUTING -d 192.168.0.10 -j SNAT --to 192.168.0.1
```

É possível ainda indicar uma lista de portas (usando a opção -m multiport), como em:

```
# Redireciona um conjunto de portas
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p tcp -i eth0 -m multiport --dport 21,22,80 -j DNAT \
--to-dest 192.168.0.10
iptables -t nat -A POSTROUTING -d 192.168.0.10 -j SNAT --to 192.168.0.1
```

Note que nos três exemplos usei o parâmetro "-p tcp". Ele é necessário, mas faz com que a regra se aplique apenas a portas TCP. Caso você precise fazer forwarding de portas UDP, deve alterar o protocolo dentro da regra, como em:

```
# Redireciona uma porta UDP
echo 1 > /proc/sys/net/ipv4/ip_forward
iptables -t nat -A PREROUTING -p udp -i eth0 --dport 53 -j DNAT --to 192.168.0.10
iptables -t nat -A POSTROUTING -d 192.168.0.10 -j SNAT --to 192.168.0.1
```

» Próximo: [Bloqueando portas de saída](#)

Mais um uso importante para o firewall é bloquear portas de saída, ou seja, bloquear portas no sentido rede local > internet. Isso permite bloquear o uso de determinados programas que utilizem estas portas.

O MSN, por exemplo, utiliza originalmente a porta 1863. Nas versões recentes ele é capaz de se conectar também através da porta 80 (ou através de sites como o meebo.com, que permitem acessar o MSN diretamente através do navegador). Por isso, ao bloquear a porta 1863, os clientes podem continuar conseguindo se conectar, porém, você obriga o tráfego a passar pela porta 80, onde tem a chance de fazê-lo passar por um servidor Squid, configurado como proxy transparente. Isso permite logar os acessos ou sabotar o sistema de

autenticação do MSN, bloqueando os domínios "messenger.hotmail.com" e "webmessenger.msn.com", além de outros sites que ofereçam clientes via web.

Hoje em dia, cada vez mais programas são capazes de acessar a Web através da porta 80, 443 (https) ou via proxy, o que torna difícil bloqueá-los. Em muitos casos, é preciso usar uma combinação de portas fechadas no firewall, bloqueio a endereços IPs específicos e bloqueio de determinados domínios no Squid.

Ao criar as regras do Iptables, existem duas opções. Bloqueando a porta para "FORWARD", você impede o acesso a partir dos micros da rede local, que acessam através da conexão compartilhada pelo servidor. Bloqueando para "OUTPUT", a porta é bloqueada no próprio micro onde o firewall está ativo. Você pode bloquear as duas situações, duplicando a regra:

```
iptables -A OUTPUT -p tcp --dport 1863 -j REJECT
iptables -A FORWARD -p tcp --dport 1863 -j REJECT
```

Você pode ainda bloquear intervalos de portas, separando-as por ":", como em:

```
iptables -A FORWARD -p tcp --dport 1025:65536 -j REJECT
```

Como estamos criando regras para os micros da rede local e não para possíveis invasores provenientes da Internet, é aconselhável usar a regra "REJECT" ao invés de "DROP". Caso contrário, os programas nos clientes sempre ficarão muito tempo parados ao tentar acessar portas bloqueadas, o que vai gerar reclamações e um certo overhead de suporte.

Você pode descobrir facilmente quais portas de saída são utilizados por cada programa fazendo buscas no Google, mas tentar bloquear um a um todos os programas indesejados acaba sendo tedioso. Ao invés disso, você pode experimentar um solução mais radical: inverter a lógica da regra, bloqueando todas as portas de saída e abrindo apenas algumas portas "permitidas".

O mínimo que você precisa abrir neste caso são as portas 80 e 53 (dns). A partir daí, você pode abrir mais portas, como a 21 (ftp), 25 (smtp), 110 (pop3) e assim por diante. Um exemplo de configuração neste caso seria:

```
iptables -A FORWARD -p udp -i eth1 --dport 53 -j ACCEPT
iptables -A FORWARD -p tcp -i eth1 --dport 80 -j ACCEPT
iptables -A FORWARD -p tcp -i eth1 --dport 21 -j ACCEPT
iptables -A FORWARD -p tcp -i eth1 -j LOG
iptables -A FORWARD -p tcp -i eth1 -j REJECT
```

Veja que todas as regras especificam a interface da rede local (eth1 no exemplo), de onde serão recebidas as conexões dos clientes. Note que não inclui nenhum bloqueio para forwarding de pacotes provenientes da interface eth0 (da internet), pois a idéia é bloquear diretamente as requisições dos clientes e não as respostas. Em uma conexão TCP típica, o cliente envia a requisição na porta TCP usada pelo serviço, mas recebe a resposta em uma porta aleatória. Este é um exemplo de entrada no log do Iptables que mostra a resposta a uma conexão http normal. Veja que ela está endereçada à porta 45159 do cliente:

```
IN=eth0 OUT=eth1 SRC=64.233.169.99 DST=192.168.0.10 LEN=40 TOS=0x00 PREC=0x00 TTL=239
ID=36813 PROTO=TCP SPT=80 DPT=45159 WINDOW=8190 RES=0x00 ACK FIN URGP=0
```

No caso da porta 53 (DNS) estou especificando o protocolo UDP, ao invés de TCP, pois as requisições são feitas usando portas UDP para ganhar tempo. Embora os servidores DNS escutem tanto na porta 53 TCP, quanto UDP, na prática quase sempre é usada a porta 53 UDP, pois o tempo de resposta é menor. No UDP a requisição é simplesmente respondida da forma mais rápida possível, enquanto que no TCP é necessário abrir e encerrar a conexão.

A regra "iptables -A FORWARD -j LOG" é uma boa opção durante a fase de testes, pois ela faz com que o Iptables logue todos os pacotes que forem encaminhados (tanto envio, quanto resposta), permitindo que você verifique o que está ocorrendo quando algo não estiver funcionando. Você pode acompanhar o log usando o comando "dmesg".

Colocado nesta posição (depois das regras que autorizam as conexões nas portas 53 e 80), ele vai mostrar apenas as requisições bloqueadas pelo firewall, dando-lhe a chance de acompanhar os acessos dos clientes e permitir portas adicionais sempre que necessário.

Por exemplo, esta estrada (no log) mostra uma tentativa de conexão de um cliente MSN rodando no micro "192.168.0.10" que foi bloqueada pelo firewall:

```
IN=eth1 OUT=eth0 SRC=192.168.0.10 DST=207.46.28.77 LEN=60 TOS=0x00 PREC=0x00 TTL=63
ID=21328 DF PROTO=TCP SPT=38119 DPT=1863 WINDOW=5840 RES=0x00 SYN URGP=0
```

A opção "DPT" indica a porta usada. Se quisesse autorizar o uso do programa, você adicionaria a regra "iptables -A FORWARD -p tcp -i eth1 --dport 1863 -j ACCEPT" em seu script.

Outra opção, para não precisar abrir tantas portas e ter um melhor controle sobre o tráfego de saída é usar um servidor Squid configurado como proxy transparente (interceptando o tráfego da porta 80) e rodar servidores locais para DNS e e-mail (você pode configurar um servidor Postfix como sistema satélite, de forma que ele envie os e-mails dos usuários da rede usando o SMTP do provedor), de forma que qualquer acesso precise necessariamente passar por algum dos serviços ativos no servidor, sujeito a log e aos bloqueios que configurar.

Neste caso, desabilite o compartilhamento da conexão (ou bloquee o forward de todas as portas) e configure os clientes para utilizarem o IP do servidor como DNS, servidor SMTP, POP e outros serviços que tenha ativado. Mesmo ao ser configurado como proxy transparente, o Squid continua funcionando como um proxy tradicional, através da porta 3128. Você pode configurar clientes de FTP e outros programas com suporte a proxy para acessarem através dele. A vantagem sobre o acesso direto é que ao passar pelo proxy, tudo fica registrado e todo acesso precisa passar pelos filtros de domínios, formatos de arquivos, limitação de banda, etc. definidos por você.

Complementando o bloqueio de portas, você pode também bloquear o acesso de determinados endereços IP, como em:

```
# Bloqueia o acesso à web a partir de um determinado IP
iptables -A FORWARD -p tcp -s 192.168.0.67 -j REJECT
```

Esta regra deve ir logo no início do script, antes das regras que abrem portas de saída, caso contrário não surtirá efeito. Lembre-se de que o Iptables processa as regras seqüencialmente: se uma compartilha a conexão com todos os micros da rede, não adianta tentar bloquear para determinados endereços depois. As regras com as exceções devem sempre vir antes da regra mais geral.

» Próximo: [Bloqueando domínios](#)

É possível ainda bloquear ou permitir com base no domínio, tanto para entrada quanto saída. Isso permite bloquear sites e programas diretamente a partir do firewall, sem precisar instalar um servidor Squid e configurá-lo como proxy transparente. Nesse caso, usamos o parâmetro "-d" (destiny) do Iptables, seguido do domínio desejado.

Para bloquear os acessos ao Orkut, por exemplo, você usaria as regras:

```
iptables -A OUTPUT -d orkut.com -j DROP
iptables -A FORWARD -d orkut.com -j DROP
```

A primeira linha bloqueia pacotes de saída destinados ao domínio, ou seja, impede que ele seja acessado a partir da própria máquina local. A segunda linha bloqueia o forward de pacotes destinados a ele (domínio), ou seja, impede que outras máquinas da rede local, que acessam através de uma conexão compartilhada, accessem-no.

Se for paranóico, você pode usar também a regra:

```
iptables -A INPUT -s orkut.com -j DROP
```

Esta regra impede também que qualquer pacote proveniente do orkut.com chegue até a sua máquina. Como disse, é apenas para paranoides ;).

Originalmente, o Iptables sabia trabalhar apenas com endereços IP. A possibilidade de criar regras baseadas em domínios são um recurso um pouco mais recente, onde o firewall faz um lookup do domínio, para descobrir qual é o IP atual e assim poder bloqueá-lo. Você pode verificar o IP usado pelo servidor de um determinado domínio usando o comando "**dig**" (que no Debian faz parte do pacote "dnsutils"), como em:

```
$ dig orkut.com
```

A vantagem de criar as regras do firewall baseado em domínios é que as regras são automaticamente atualizadas caso o servidor do site mude de endereço.

Ao bloquear o "orkut.com" no Iptables, você automaticamente bloqueia o "www.orkut.com" ou qualquer outra variante ou outros domínios que levem ao mesmo servidor. A principal limitação é que a regra não se aplica a subdomínios hospedados em diferentes servidores. Por exemplo, você pode bloquear o domínio "uol.com.br", mas isso não bloqueará o "tvuol.uol.com.br", que é hospedado em um servidor separado. Em casos como este, a única solução é bloquear ambos.

» Próximo: [Resumo das regras do Iptables](#)

Depois desta rodada inicial de exemplos, nada melhor do que um guia mais detalhado dos parâmetros suportados pelo Iptables. Escrever regras de firewall é quase como aprender um novo dialeto. Existem muitas combinações possíveis entre os parâmetros disponíveis e "regras de concordância" que determinam o que funciona e o que não. Imagine que ao escrever uma nova regra, você está explicando uma idéia. Tente ser claro para que seja entendido ;).

Parâmetros do Iptables:

-A INPUT: Especifica que a regra se aplica a pacotes de entrada, ou seja, pacotes recebidos pelo servidor, em qualquer interface.

-A OUTPUT: A regra se aplica a pacotes de saída, transmitidos pelo próprio servidor.

-A FORWARD: Este parâmetro é usado ao compartilhar a conexão com a internet, permitindo que os micros da rede local acessem através do servidor. Os pacotes de outros micros, encaminhados pelo servidor, são tratados como "FORWARD", diferentemente dos pacotes transmitidos pelo próprio servidor, que são tratados como "OUTPUT". Você pode definir regras diferentes para cada situação.

-p tcp: Especifica que a regra se aplica a pacotes TCP, o que é o mais comum.

-p udp: Alguns serviços usam também portas UDP. Um bom exemplo são os servidores DNS, que escutam tanto na porta 53 TCP, quanto na 53 UDP. Este parâmetro permite definir regras que se aplicam a estes casos, abrindo ou fechando as portas UDP, como em:

```
iptables -A INPUT -p udp --dport 53 -j ACCEPT
```

A maioria das regras do Iptables exigem que você especifique o protocolo, fazendo com que você tenha que repetir a regra caso queira abrir uma porta simultaneamente para TCP e UDP. Ao executar algo como "iptables -A INPUT --dport 53 -j ACCEPT" (sem especificar o protocolo), você receberá um erro como:

```
iptables v1.3.3: Unknown arg
Try `iptables -h' or 'iptables --help' for more information.
`--dport'
```

Como as portas UDP também são usadas por alguns serviços, é muito comum bloquear as portas de 1 a 1024 UDP, autorizando apenas as portas que realmente devem ficar abertas, como em:

```
iptables -A INPUT -p udp --dport 1:1024 -j DROP
```

Note que você nunca deve fechar todas as portas UDP, pois as portas altas são usadas aleatoriamente para pacotes de resposta para DNS e diversos outros protocolos. Você pode fazer o teste: use a regra "iptables -A INPUT -p udp -j DROP" e você não conseguirá mais navegar até desativar o firewall usando o comando "iptables -F".

Alguns administradores mais paranóicos fecham todas as portas UDP até a 32000, por exemplo. Não existem muitos problemas em fechar uma faixa maior de portas, desde que você sempre deixe uma larga faixa de portas altas abertas, de forma a receber os pacotes de resposta.

Ao contrário do TCP, não é possível criar uma regra genérica para permitir todos os pacotes de resposta UDP (como a "iptables -A INPUT -p tcp --syn -j DROP"), pois no UDP não são abertas conexões. Os pacotes são simplesmente transmitidos diretamente, sem aviso prévio.

-p icmp: Além do TCP e UDP, existe o protocolo ICMP, usado para pacotes de controle, pings e envio de mensagens. Um exemplo de uso é a regra para desativar a resposta a pings que vimos há pouco. Na verdade ela atua bloqueando o pedido de ping antes que ele seja repassado ao sistema:

```
iptables -A INPUT -p icmp --icmp-type echo-request -j DROP
```

-i eth0: A opção "-i" permite definir a interface onde os pacotes devem ser recebidos ou enviados. Por exemplo, usar uma regra como:

```
iptables -A INPUT -p tcp -j REJECT
```

... simplesmente desconectaría seu micro da rede, pois bloquearia comunicações em qualquer interface. Porém, se você especificasse a interface, ele bloquearia apenas pacotes recebidos através dela, como em:

```
iptables -A INPUT -i eth2 -p tcp -j REJECT
```

O mesmo se aplica quando você quer permitir conexões em determinadas portas, mas apenas a partir da placa de rede local. Para permitir conexões via SSH apenas a partir da placa eth1, você poderia usar:

```
iptables -A INPUT -i eth1 -p tcp --dport 22 -j ACCEPT
```

-o eth0: É similar ao parâmetro "-i", mas especifica uma interface de saída. Este parâmetro é menos usado, pois normalmente nos preocupamos em impedir que o firewall aceite conexões em determinadas portas, ao invés de tentar interceptar as respostas. Mas esta opção pode ser útil em casos em que você precisa fechar uma porta de saída apenas para

determinada interface. Este é um exemplo de uso, onde bloqueio pacotes de saída na porta 1863 apenas para a placa eth1:

```
iptables -A OUTPUT -p tcp -o eth1 --dport 1863 -j DROP
```

--dport ou --destination-port: Especifica uma porta. O uso mais comum para esta opção é para abrir portas de entrada (e depois aplicar uma regra que fecha as demais), como na regra que abre para conexões na porta 22, que mostrei no exemplo anterior.

-s (source): O parâmetro "-s" permite especificar um endereço IP ou domínio de origem, de forma a aceitar ou recusar as conexões. Embora seja muito fácil forjar endereços IP dentro da rede local, as coisas são muito mais complicadas na Internet. Permitir o acesso a determinadas portas (como a do SSH) apenas para determinados endereços, ou faixas de endereços, é uma medida de segurança interessante em muitas situações.

Este é um exemplo de regra, que abre a porta 631 apenas para hosts dentro da faixa e máscara especificada:

```
iptables -A INPUT -p tcp -s 72.232.35.0/255.255.255.248 -j ACCEPT
```

-d (destiny): Destinado ao endereço IP ou domínio citado. É muito usado ao bloquear o acesso a determinados sites a partir dos micros da rede local, como, por exemplo:

```
iptables -A FORWARD -d torrentreactor.net -j REJECT
```

-m mac --mac-source 00:11:D8:76:59:2E: Esta é a regra que permite especificar endereços MAC dentro de regras do Iptables que vimos há pouco. Ela é uma forma de dificultar o uso de endereços IP falseados para ganhar acesso ao servidor, pois permite relacionar o IP ao endereço MAC da placa instalada. Lembre-se, porém, que ela só pode ser usada em rede local e que os endereços MAC são quase tão fáceis de falsear quanto os endereços IP. Um exemplo de uso seria:

```
iptables -A INPUT --dport 22 -m mac --mac-source 00:11:D8:76:59:2E -j ACCEPT
```

--syn: Cria uma regra válida apenas para novas conexões, não impedindo que o outro micro responda a conexões iniciadas pelo servidor, como em:

```
iptables -A INPUT -p tcp --syn -j DROP
```

-j: É usado no final de cada regra, especificando uma ação, que pode ser:

-j ACCEPT : Aceita o pacote. Ele é encaminhado ao destino sem passar pelas demais regras.

-j REJECT : Rejeita educadamente o pacote, enviando um pacote de resposta ao emissor. Quando uma porta está fechada em modo reject, o emissor recebe rapidamente uma resposta como "connect to host 192.168.1.1 port 22: Connection refused".

-j DROP: O DROP é mais enfático. O pacote é simplesmente descartado, sem aviso. O emissor fica um longo tempo esperando, até que eventualmente recebe um erro de time-out.

-j LOG: Este último parâmetro permite logar conexões. É interessante usar esta regra principalmente em portas muito visadas, como a do SSH, pois assim você tem uma lista de todos os endereços que acessaram seu servidor na porta especificada. Para ativar o log, você deve duplicar a regra que abre a porta, usando a opção "-j LOG" na primeira e "-j ACCEPT" na segunda, como em:

```
iptables -A INPUT -p tcp --dport 22 -j LOG  
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

As mensagens são gravadas no arquivo "/var/log/messages" de forma bastante detalhada, incluindo a data e hora da conexão, o IP e MAC do micro que fez a conexão (SRC), além da porta (DPT). Você pode ver o mesmo log, porém com as entradas escritas de forma resumida, usando o comando "dmesg".

```
Jun 29 15:49:46 spartacus kernel: IN=eth0 OUT= MAC=00:e0:7d:9b:f8:01:00:15:00:4b:68:db:08:00  
SRC=192.168.0.2 DST=192.168.0.1 LEN=52 TOS=0x00 PREC=0x00 TTL=64 ID=32704 DF PROTO=TCP  
SPT=56848 DPT=22 WINDOW=2164 RES=0x00 ACK URGP=0
```

Não se assuste com o volume, pois o log inclui todas as tentativas de conexão. O fato de um determinado IP ter aberto uma conexão com a porta 22 do seu servidor, não significa que o usuário realmente obteve acesso ao SSH. Ele pode ter recebido o prompt para digitar a senha, mas isso não significa que ele realmente conseguiu fazer login.

Note que alterar a ordem das regras altera o resultado. Em caso de duas regras conflitantes, vale a que vem primeiro. Por exemplo, ao usar:

```
iptables -A INPUT -p tcp --dport 22 -j ACCEPT  
iptables -A INPUT -p tcp --dport 22 -j REJECT
```

... a porta 22 permanecerá fechada, pois os pacotes serão descartados pela primeira regra, sem terem chance de serem autorizados pela segunda. É justamente por isso que é sempre necessário colocar as regras menos restritivas primeiro, declarando as portas autorizadas, para só então fechar as demais.

O mesmo se aplica ao logar transmissões. Se você usar algo como:

```
iptables -A INPUT -p tcp --dport 22 -j LOG  
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

... o log não funcionará, pois os pacotes destinados à porta 22 serão aceitos pela primeira regra e não passarão pela segunda, que faz o log. As regras que fazem log devem sempre vir antes das regras que autorizam as conexões.

» Próximo: [Testando com o Nmap](#)

Depois de terminar, você pode testar o firewall usando o Nmap, a partir de outro micro da rede local ou da internet, para procurar vulnerabilidades. O Nmap é um pacote bastante popular. No Debian você pode instalá-lo pelo apt-get.

Caso a regra que bloqueia tudo esteja ativa, você vai ter o seguinte como resultado:

```
# nmap -sS -v 192.168.0.33
```

```
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-04-03 10:12 BRT
Host 192.168.0.33 appears to be down, skipping it.
Note: Host seems down. If it is really up, but blocking our ping probes, try -P0
Nmap run completed -- 1 IP address (0 hosts up) scanned in 12.053 seconds
```

Ou seja, o Nmap não consegue sequer perceber que o PC está realmente lá, e avisa: "Se você realmente tem certeza que ele está online, experimente usar a opção -P0", o que não vai mudar muita coisa:

```
# nmap -P0 -v 192.168.0.33
```

```
Starting nmap 3.50 ( http://www.insecure.org/nmap/ ) at 2004-04-03 10:14 BRT
Host 192.168.0.33 appears to be up ... good.
Initiating SYN Stealth Scan against 192.168.0.33 at 10:14
The SYN Stealth Scan took 1361 seconds to scan 1659 ports.
All 1659 scanned ports on 192.168.0.33 are: filtered
Nmap run completed -- 1 IP address (1 host up) scanned in 1360.579 seconds
```

Como todas as portas estão em modo drop, onde o firewall simplesmente descarta os pacotes sem confirmar o recebimento, o teste demora muito tempo, quase 27 minutos para escanear apenas as primeiras 1659 portas. Uma varredura completa, em todas as 65 mil portas, levaria 17 horas e meia, isso executando o teste via rede local. Via internet, a varredura levaria vários dias e, mesmo assim, só apareceriam as portas manualmente abertas no seu script.

» Próximo: [Executando com um usuário separado](#)

O firewall protege contra worms e invasões, ataques "de fora para dentro", mas não protege contra vírus e trojans executados localmente.

No primeiro tipo, o atacante procura uma porta aberta, tenta identificar o servidor ou programa que está ativo na porta (o SSH ou Apache, por exemplo), pesquisa por alguma vulnerabilidade conhecida ou erro de configuração e, caso encontre alguma coisa, lança um ataque, tentando utilizar a falha para obter acesso ao sistema.

Com o firewall ativo, o ataque é frustrado logo no início. Com todas as portas fechadas, simplesmente não existe por onde entrar, a menos que exista alguma falha de segurança no próprio Iptables, o que seria bastante improvável.

Imagine que um desktop que acessa a web apenas como cliente é uma casa, enquanto que um servidor é uma loja de porta aberta. Uma casa pode ser segura, pois apenas o dono entra e sai.

Você pode fechar e reforçar todas as portas, transformando sua casa em um bunker (habilitar o firewall, fechando todas as portas), onde ninguém terá como entrar, a menos que consiga convencer você a abrir a porta para ele.

Em um servidor a questão é mais complicada. Assim como em uma loja, é preciso deixar a porta bem aberta para que os clientes possam entrar e sair. A segurança, nesse caso, precisa ser feita em vários níveis. Em primeiro vem o firewall, que bloqueia todas as portas que não são usadas, mantendo abertas apenas as portas realmente utilizadas, como, por exemplo, a porta 53 (UDP) do DNS, a porta 80 do Apache e a porta 22 do SSH.

Em segundo lugar vem a questão das atualizações de segurança. Ninguém invade um servidor simplesmente porque o SSH está habilitado. Invade caso esteja em uso uma versão desatualizada, com alguma vulnerabilidade conhecida, ou caso exista alguma conta de usuário ativa, com uma senha fácil. Mantendo o servidor atualizado e seguindo regras básicas de segurança, o risco é muito pequeno.

Na hora de escolher uma distribuição para ser usada em um servidor, um dos quesitos mais importantes a analisar é justamente a questão das atualizações de segurança: em quanto tempo os pacotes são atualizados ao ser descoberta uma vulnerabilidade e por quanto tempo são disponibilizadas atualizações para a versão em uso.

A maior parte das distribuições comerciais oferece atualizações de segurança por 12 ou 18 meses depois de lançada uma nova versão. Neste quesito, as distribuições baseadas no Debian levam vantagem, pois no Debian as atualizações são oferecidas de forma contínua. Sempre que uma nova versão é lançada, você pode atualizar os pacotes utilizando o apt-get e, assim, continuar instalando as atualizações indefinidamente.

O segundo tipo de ataque, que engloba vírus, trojans e afins, exige interação do usuário. Um vírus nunca se instala sozinho (caso contrário não seria um vírus, mas sim um worm), se instala porque alguém executou o arquivo que chegou por e-mail ou por um programa P2P, por exemplo.

No Windows, esta tarefa ficaria por conta do antivírus, antispyware, antitrojan & cia. Mas, como ainda não temos uma quantidade expressiva destas pragas no Linux, apenas o firewall em geral já é suficiente. Digo por enquanto, pois conforme o uso do sistema em desktops cresça, é natural que o número de vírus e pragas em geral para a plataforma também cresça, obrigando-nos a tomar mais cuidados.

Caso eventualmente os vírus e trojans tornem-se um problema no Linux, com certeza surgirão várias opções de antivírus. Mas, mesmo antes que isso aconteça, existe um conjunto de cuidados simples que pode manter seu micro seguro desde já.

O Linux é reconhecidamente um sistema multiusuário, onde as permissões de arquivos e executáveis impedem que um usuário danifique arquivos ou configurações de outro, ou modifique as configurações do sistema. Embora isso torne as coisas mais complicadas em diversas situações, também cria uma barreira de segurança adicional bastante interessante.

Ao invés de rodar todos os programas e executar todo tipo de arquivo com seu usuário principal, crie um segundo login (ou até mais de um) e o utilize para executar arquivos suspeitos ou programas que possam ter problemas de segurança, como, por exemplo, clientes de IRC e navegadores.

Para isso, abra um terminal e use o comando "**su**" para logar-se usando o segundo usuário, como em "**su joao**". Depois de fornecer a senha, todos os programas executados dentro do terminal serão executados pelo outro usuário. Se por acaso você executar qualquer programa malicioso, apenas ele é afetado, sem comprometer seus arquivos pessoais, muito menos os arquivos do sistema. Em caso de problemas, basta deletá-lo e criar outro.

Em distribuições baseadas no Debian, o sistema vem configurado para não permitir que outros usuários executem programas gráficos dentro de uma sessão gráfica já aberta. Ao tentar rodar qualquer programa gráfico, você recebe uma mensagem como:

```
Xlib:      connection      to      ":0.0"      refused      by      server
Xlib:          No           protocol
konqueror: cannot connect to X server :0.0
```

Isso é solucionado por um utilitário chamado "**sux**", que substitui o **su**, transferindo também as credenciais do X. Basta instalar o "**sux**" usando o apt-get e usá-lo para trocar o usuário, como em: "**sux joao**".

» Próximo: [Usando o Firestarter](#)

O Firestarter é um firewall gráfico, que é ao mesmo tempo bastante poderoso e fácil de usar. Ele é adequado para uso em desktops, onde é necessária uma forma simples de monitorar tentativas de conexão e abrir portas.

Ele tornou-se rapidamente uma opção bastante popular e passou a ser incluído nas principais distribuições. Você pode instalá-lo usando o gerenciador de pacotes (opção recomendada) ou baixar os pacotes disponíveis no <http://www.fs-security.com/download.php>.

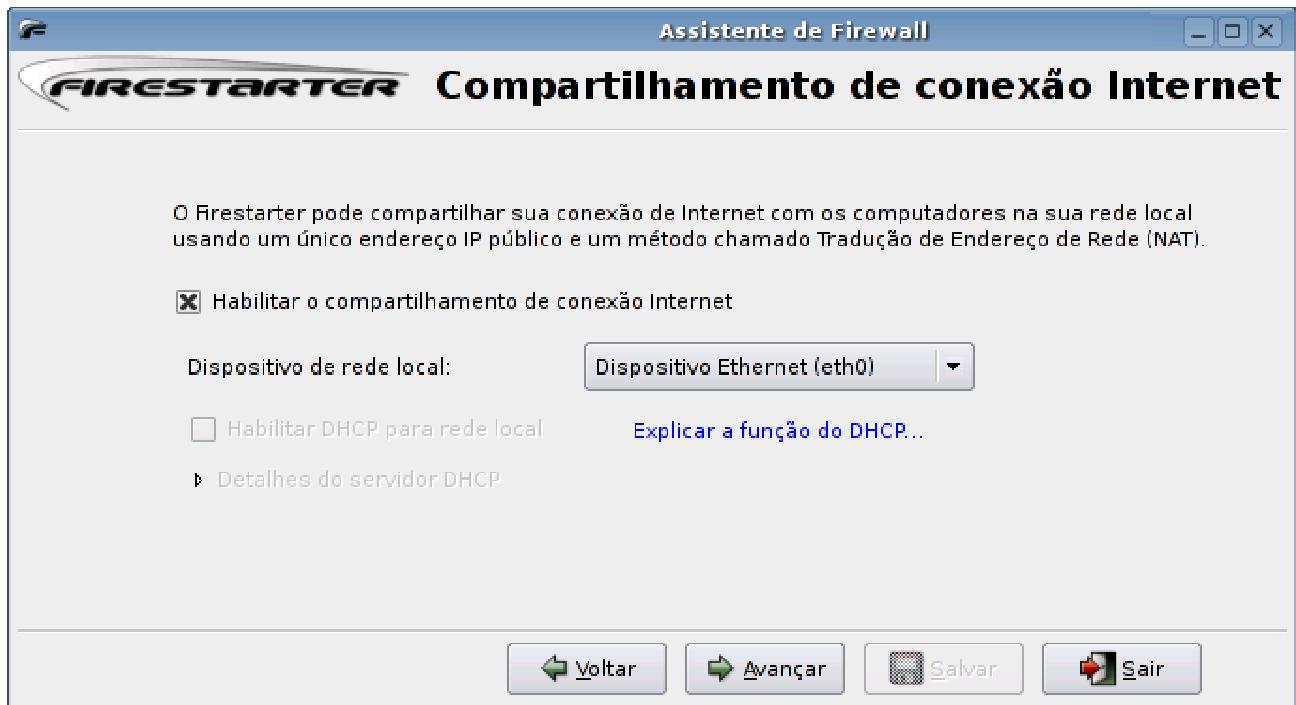
Uma última opção é baixar o pacote com o código-fonte e compilá-lo manualmente. A instalação neste caso é feita descompactando o arquivo e rodando os comandos "**./configure**", "**make**" e "**make install**". A dificuldade é que você precisa ter os compiladores e a biblioteca de desenvolvimento do GTK instalados.

Ao abrir o Firestarter pela primeira vez, é aberto um assistente que pede algumas informações básicas sobre a configuração da rede e oferece opções para compartilhar a conexão e ativar o firewall sob demanda, ao conectar via modem ou ADSL PPPoE.

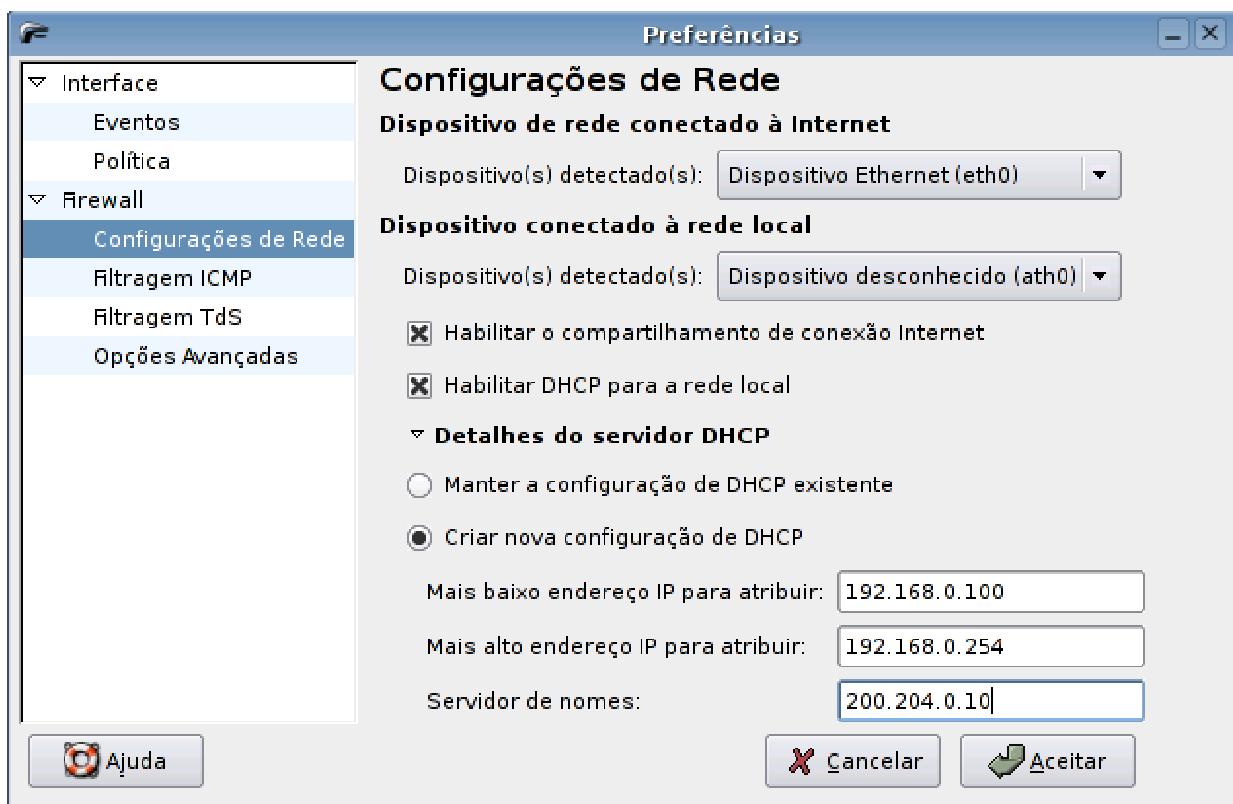
O compartilhamento de conexão cria um compartilhamento simples, via NAT, equivalente a usar os três comandos que vimos anteriormente. A única limitação é que o Firestarter não permite compartilhar usando uma única placa de rede (usando uma interface virtual, como

vimos no capítulo 5). Nele é realmente necessário "fazer do jeito certo", usando duas placas de rede.

Ao compartilhar a conexão, é necessário apenas indicar qual é a placa ligada à rede local:



Estas configurações podem ser alteradas posteriormente no menu "Editar > Preferências". Se a opção de Habilitar o servidor DHCP aparecer desativada na sua configuração, verifique se o pacote com o servidor DHCP (dhcp3-server, dhcp-server ou dhcp, dependendo da distribuição) está instalado. O Firestarter apenas altera a configuração de um servidor DHCP já instalado, ele não faz a instalação para você.



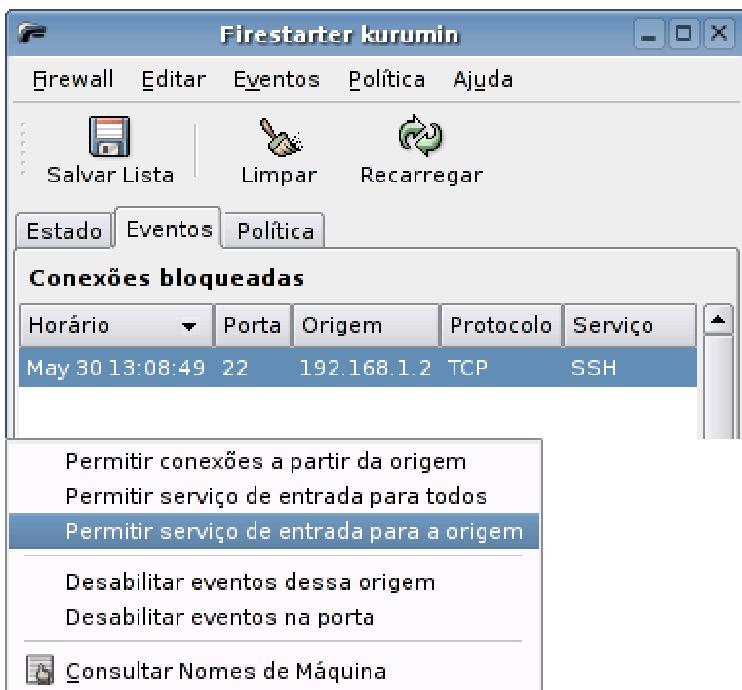
Como o Firestarter precisa manipular as regras do Iptables e configurar outros componentes do sistema, ele só pode ser executado como root. Em muitas distribuições, é adicionado um ícone no menu que executa o Firestarter através do gksu ou kdesu, pedindo a senha de root. Por padrão, uma vez aberto, o Firestarter bloqueia todas as portas e loga todas as tentativas de conexão, uma configuração bastante segura.

Ainda na janela de configurações, verifique se a opção "Método de rejeição de pacotes preferido" está configurada como "Descartar silenciosamente", em que é usada a política "DROP" do Iptables, ao invés de "REJECT", onde o emissor recebe resposta.

A opção "Tráfego de broadcast" se refere a todos os pacotes direcionados à rede, como, por exemplo, os pacotes usados por servidores Windows (e Samba) para mapear os compartilhamentos disponíveis na rede. Deixe sempre a opção "Block broadcasts from external network" (pacotes vindos da internet) habilitada. Caso esteja usando uma rede wireless, ou acessando através de uma rede de terceiros, marque também a opção para a rede local.



Um dos recursos mais interessantes, e o principal diferencial com relação a outros projetos, é que ele transforma os logs de tentativas de acesso gerado pelo Iptables em avisos dentro da aba "eventos". Quando uma nova tentativa de acesso é registrada, o ícone ao lado do relógio fica vermelho e você tem a opção de aceitar ou recusar a conexão. Na ilustração, temos uma tentativa de acesso ao servidor SSH, que está habilitado na porta 22 a partir do host 192.168.1.2.



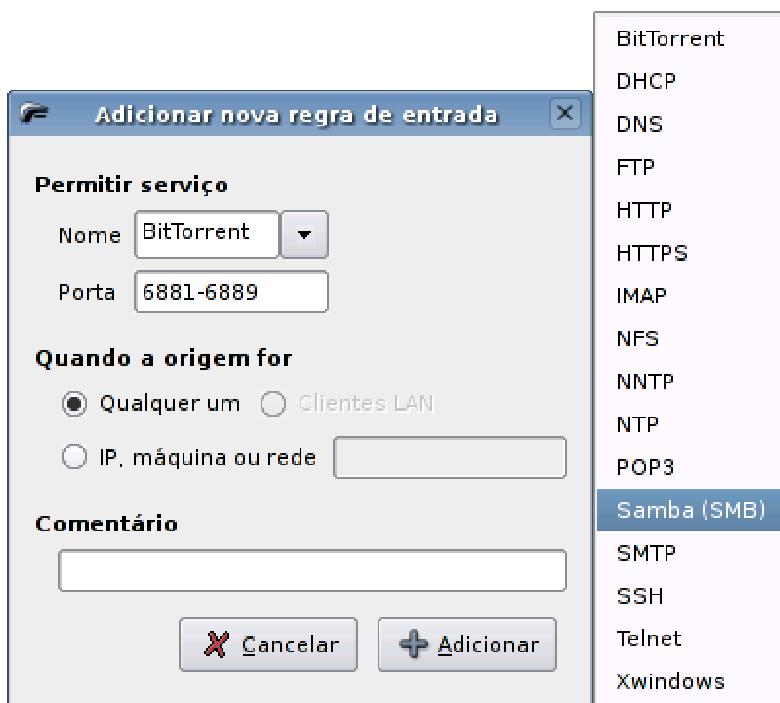
A opção "Permitir serviço de entrada para a origem" faz com que, daí em diante, o host 192.168.1.2 possa acessar o SSH, sem disparar novamente o alarme, enquanto a opção "Permitir conexões a partir da origem" faz com que o 192.168.1.2 possa acessar qualquer serviço, em qualquer porta, sem disparar o alarme. Esta segunda opção é interessante para micros da rede local.

Finalmente, a opção "Permitir serviço de entrada para todos" abre a porta do SSH para todo mundo, incluindo micros da internet. É uma opção que deve ser usada com mais cautela.

Todas as regras adicionadas entram em vigor imediatamente e ficam acessíveis para modificação ou consulta na aba "Política". Você pode ir também direto ao ponto, abrindo as portas utilizadas por algum serviço em especial antes que o firewall bloquee a conexão. Na interface principal, acesse a aba "Política", clique com o botão direito sobre o quadro "Permitir serviço", "Adicionar Regra".

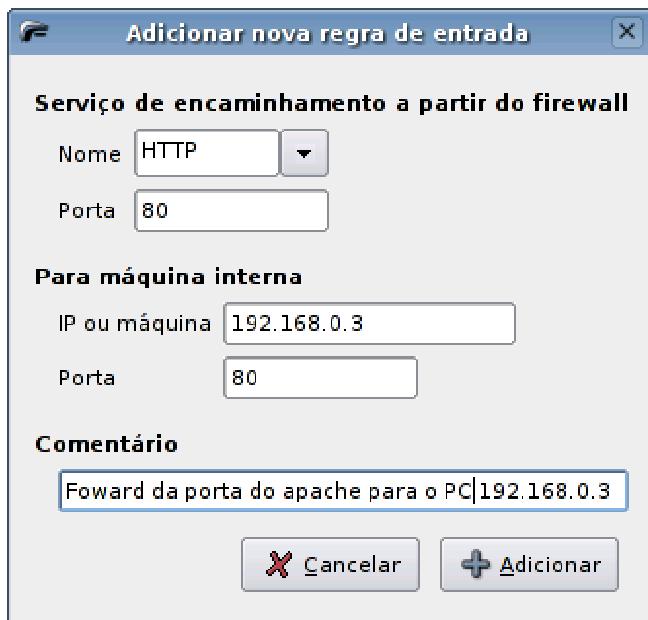
Já estão disponíveis regras prontas para vários serviços. Lembre-se de que é necessário abrir portas apenas quando você está rodando um servidor Samba, Postfix, SSH, etc., não é preciso abrir portas para acessar estes mesmos serviços como cliente. A única exceção importante para esta regra de ouro é o NFS, onde é preciso manter a porta 11 aberta (no cliente) para conseguir montar os compartilhamentos.

Note que além da opção para abrir para todo mundo, você pode abrir apenas para um endereço IP específico ou para uma faixa de IPs, como em "192.168.1.0".



Caso o compartilhamento da conexão esteja ativado, aparecerá mais uma seção dentro da aba "Política", a "Serviço de encaminhamento", que permite redirecionar portas de entrada para micros da rede local.

A configuração é similar à abertura de portas, mas agora, ao invés de especificar quais endereços terão acesso à porta aberta, você especifica qual micro da rede local receberá as conexões direcionadas a ela:



Você pode acompanhar as conexões em uso através do campo "Conexões ativas", na tela principal. Note que a lista inclui todas as conexões, tanto as conexões como cliente, contatando outros micros da rede ou internet, quanto as conexões como servidor, recebendo uma conexão a partir de fora.

Note que muitos programas abrem diversas conexões simultâneas, o Gaim (ou outro cliente de ICQ/MSN), por exemplo, abre uma conexão com o servidor principal, quando você fica online, e mais uma conexão para cada janela de conversa aberta.

Uma única instância do BitTorrent, por exemplo, pode chegar a abrir mais de 20 conexões, já que baixa e serve o arquivo para vários hosts simultaneamente. Preste atenção nas conexões em que o destino é seu próprio IP, pois elas indicam gente se conectando a servidores ativos na sua máquina.

▼ Conexões ativas				
Origem	Destino	Porta	Serviço	Programa
192.168.1.100	207.46.107.124	1863	Desconhecido	gaim
192.168.1.100	192.168.1.2	22	SSH	ssh
192.168.1.2	192.168.1.100	22	SSH	
192.168.1.100	64.12.24.152	5190	AOL IM	gaim

Caso o compartilhamento de conexão esteja ativo, a lista mostra todas as conexões, de todos os micros da rede local (ou seja, uma lista possivelmente bem grande). Isso pode ser usado para detectar micros que estão rodando programas que consomem muita banda, como programas P2P em geral, e tomar as providências necessárias, advertindo o usuário ou bloqueando as portas ou IPs das estações.

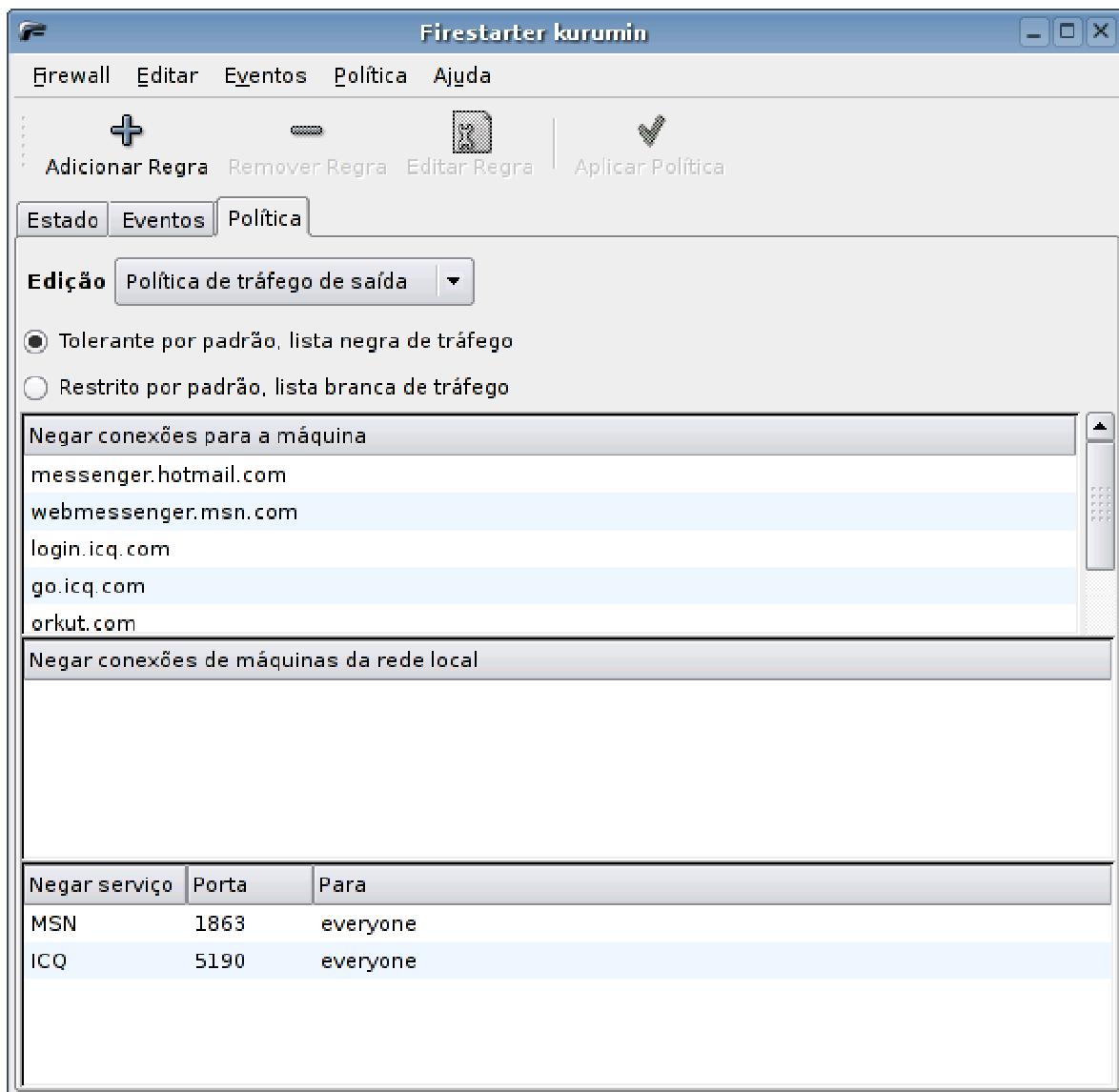
Para isso, acesse a aba "Política". Mude a opção no botão "Edição" para "Política de tráfego de saída". As opções agora permitem bloquear tráfego de dentro para fora, impedindo que determinados programas-clientes funcionem, ou que certos servidores ou sites sejam acessados.

Neste caso, existem duas abordagens. Você pode bloquear a porta usada pelo cliente, ou pode bloquear o acesso ao servidor a que ele se conecta. Por exemplo, o MSN envia mensagens através da porta 1863 e se conecta ao servidor messenger.hotmail.com. Bloqueando qualquer um dos dois, o cliente já deixa de funcionar. Mas, para garantir, você bloqueia ambos. Existe ainda um cliente disponível via navegador, através da página <http://webmessenger.msn.com>, que você pode bloquear também.

O ICQ se conecta ao servidor login.icq.com, através da porta 5190. Assim como no caso do MSN, existe uma versão via navegador, disponível no site <http://go.icq.com>. Você pode bloquear as três coisas.

Na mesma tela é possível bloquear também sites específicos, incluindo domínio por domínio. A idéia aqui é bloquear páginas específicas nas quais os usuários estejam gastando muito tempo, ou páginas de conteúdo impróprio. Lembre-se de que:

- 1- Ao bloquear um domínio, os usuários ainda conseguirão acessar a página diretamente pelo IP; é necessário bloquear as duas coisas.
- 2- No caso de páginas de conteúdo impróprio, é mais prático usar um servidor Squid com o DansGuardian do que ficar tentando bloquear endereço por endereço no firewall.
- 3- Bloquear um domínio ou IP aqui vai bloquear o acesso de todos os protocolos (POP3, SMTP, SSH, FTP, etc.), não apenas http, ou seja, significa realmente cortar relações. Caso você bloqueie o acesso ao IP de um servidor que hospeda vários sites, vai bloquear o acesso a todos eles.



Veja que aqui estou usando a opção "Tolerante por padrão", na qual o firewall por padrão permite todo o tráfego de saída e você especifica manualmente o que bloquear. Você pode utilizar também o modo "Restrito por padrão", onde o firewall bloqueia tudo e você precisa ir abrindo uma a uma as portas que serão utilizadas. O mínimo, neste caso, é abrir as portas 53 (DNS), 80 (http) e 443 (https) para permitir o acesso à web básico e, a partir daí, ir abrindo um a um os demais protocolos necessários.

Uma vez ativado o firewall, as regras ficam ativas, mesmo que você feche a interface principal, mas você perde a possibilidade de monitorar as tentativas de acesso e aceitar conexões. O Firestarter fica residente na forma do serviço de sistema "firestarter". Você pode usar o comando "Iptables -L", que lista as regras de firewall ativas para comprovar isso.

Para realmente parar o firewall, você precisa reabrir a interface e clicar no "Parar firewall" ou usar o comando "/etc/init.d/firestarter stop". Imagine que, ao contrário dos firewalls para Windows, o firewall em si é independente da interface.

Para que o firewall seja inicializado automaticamente durante o boot, é importante que o sistema esteja configurado para inicializar o serviço "firestarter" durante o boot. No Mandriva, você pode habilitá-lo no menu de serviços, dentro do Painel de Controle. No Fedora, e em outras distribuições derivadas do Red Hat, use o comando "**chkconfig firestarter on**" e, nas distribuições derivadas do Debian, use o comando "**update-rc.d -f firestarter defaults**".

Uma ressalva é que, ao instalar a partir do código fonte, é preciso copiar manualmente o script "/etc/init.d/firestarter" para que ele trabalhe como um serviço de sistema. Dentro da árvore com o código-fonte, você encontra scripts para várias distribuições.

Você pode também configurar a interface do Firestarter para ficar residente, como um ícone do lado do relógio ao ser fechado no "Editar > Preferências > Interface > Minimizar para a bandeja ao fechar a janela".

Mais um problema comum é a necessidade de fornecer a senha de root cada vez que a interface do Firestarter é aberta, um detalhe bastante chato. Você pode eliminar essa necessidade utilizando o sudo para permitir que o seu usuário possa abrir o Firestarter como root, sem precisar fornecer a senha. Para isso, instale o pacote "**sudo**" (usando o gerenciador de pacotes da distribuição que estiver utilizando) e adicione as seguintes linhas no final do arquivo "**/etc/sudoers**":

```
usuário      ALL= NOPASSWD: /usr/bin/firestarter
usuário ALL= NOPASSWD: /usr/sbin/firestarter
```

Substitua o "usuário" pelo login desejado. Note que coloquei duas linhas, pois em algumas distribuições o binário do Firestarter é instalado dentro da pasta "/usr/bin" e, em outras (Debian, por exemplo), na pasta "/usr/sbin". Na verdade, você vai precisar de apenas uma delas.

Feito isso, você pode passar a inicializar o Firestarter usando o comando "sudo firestarter" ou "sudo firestarter --start-hidden", se preferir que ele já inicie minimizado ao lado do relógio.

Para que ele seja aberto automaticamente junto com o KDE, crie um arquivo de texto chamado "firestarter.desktop", na pasta ".kde/Autostart/" dentro da sua pasta home, contendo o seguinte:

```
[Desktop]                                         Entry]
Exec=sudo                                         firestarter
Name=Firestarter                                     --start-hidden
Type=Application
```

Todos os ícones de aplicativos colocados dentro desta pasta são executados durante a abertura do KDE. Você pode arrastar um ícone do menu para ela, usando o Konqueror ou criando um arquivo de texto manualmente. Note que os ícones colocados dentro desta pasta contém uma sintaxe especial e terminam com a extensão ".desktop".

»2^a EDIÇÃO 2006