SIM Gateway: roteador para redes sem fios baseado em FreeBSD

Carlos Oberdan Rolim[1] **, Carlos A. M. dos Santos**[2]

[1] SIM Telecom – Rua Imigrantes, 500 — 98800-000 Santo Ângelo, RS

[2]Universidade Regional Integrada do Alto Uruguai e das Missões — URI
Av. Universidade das Missões, 464 — 98802-470 Santo Ângelo, RS

ober@sol.psi.br, casantos@urisan.tche.br

Resumo. *Este artigo descreve a implementação de um roteador de baixo custo destinado a prover serviços de interconexão entre diversas empresas através de uma rede sem fios. O sistema é baseado em uma versão reduzida do sistema operacional FreeBSD, carregada a partir de um* flash disk*, e oferece uma interface para fácil configuração via WWW.*

1. Introdução

Redes baseadas em cabeamento têm se mostrado inadequadas para atender locais de difícil acesso, devido aos altos custos de instalação e manutenção. Em função disto, há uma demanda crescente por conexões sem fio, que independem de cabos ou da infra-estrutura provida pelas operadoras de telefonia. Para atender a essa demanda os provedores de acesso necessitam de equipamentos adequados. Este artigo descreve um roteador para redes sem fio que foi desenvolvido para ser utilizado comercialmente por uma empresa de telecomunicações.

O texto a seguir está organizado da seguinte forma: a seção 2 discute alguns conceitos relevantes de redes wireless; a seção 3 descreve a proposta inicial do projeto, os serviços fornecidos pelo roteador, os principais problemas encontrados e as soluções adotadas, e a administração do sistema; a seção 4 descreve as modificações e melhorias feitas a partir da versão inicial; na seção 5, por fim, são apresentadas as conclusões e as perspectivas para trabalhos futuros.

2. Conceitos básicos

As redes sem fio mais populares atualmente seguem o padrão IEEE 802.11b e usam portadora com frequência de 2,4GHz. Devido à saturação da faixa espectral, algumas redes estão migrando para a faixa dos 5,8GHz, mas o custo dos equipamentos que operam nessa faixa ainda é muito elevado (tipicamente o dobro do preço).

Há muitos aspectos comuns enre as redes 802.11b e 802.3; a diferença fundamental é o modo como se controla a comunicação entre as estações. O modo mais simples é o Independent Basic Service Set (IBSS), também chamado de *Ad-Hoc*, no qual cada máquina de um grupo pode se comunicar com outra máquina do mesmo grupo de forma independente. IBSS é mais utilizado em conexões ponto-a-ponto ou em ambientes livres de obstáculos que obstruam a comunicação.

Já no modo Basic Service Set (BSS) um equipamento chamado Access Point (AP) atua como intermediário, fazendo o papel de ponte (*bridge*) entre as máquinas conectadas

Tabela 1: Sensibilidade dos cartões

	Sensibilidade	
Velocidade	PRISM II	Lucent
1MBit/s	-91dBm	-94dBm
2MBit/s	-88dBm	-91dBm
5.5MBit/s	-87dBm	-87dBm
11MBit/s	-82dBm	-84dBm

Fonte: [Intersil, 2003, ORiNOCO, 2003]

via rádio e as conectadas via cabo [Flickenger, 2002]. Quando uma máquina precisa se comunicar com outra ela manda os pacotes para o Access Point e este os encaminha para a estação destino apropriada. Para usar os serviços de uma rede 802.11b uma estação deve fazer a *associação* a um Access Point.

3. Solução desenvolvida

3.1. Proposta inicial

As interfaces de rádio 802.11b em geral vêm embutidas em cartões PCMCIA. Todos eles podem ser configurados para operar tando em modo IBSS quanto BSS, mas a maioria dos "drivers" não permite configurá-los para trabalhar como AP. Com o driver *wi* do sistema operacional FreeBSD, entretanto, cartões com *chipset* PRISM II (Intersil) podem ser usados com esta finalidade.

A idéia inicial do trabalho foi, utilizando-se um PC e um cartão de rádio, construir um AP com as mesmas funcionalidades de um equipamento comercial, mas a um custo inferior. Ao comparar cartões com chipsets Hermes (Lucent) e PRISM II, entretanto, descobriu-se que este possui características que o tornam inadequado para operar como AP em ambientes abertos. Equipamentos com esse tipo de cartão possuem uma potência exagerada, que polui com ruído desnecessário a faixa espectral à sua volta, e baixa sensibilidade, fator extremamente importante para que um AP possa "ouvir " os clientes que desejam associarem-se à rede.

A tabela 1 demonstra a sensibilidade dos diferentes cartões. Um AP com um cartão PRISM II se comportaria como uma pessoa que gritasse o tempo todo e fosse surda ao mesmo tempo. Decidiu-se então direcionar o projeto para a construção de um roteador, o que atenderia a uma outra necessidade do mercado, já que os clientes BSS, em sua maioria, possuem capacidade muito limitada para operar como roteadores.

3.2. Requisitos e serviços básicos a serem prestados

DHCP (Dynamic Host Configuration Protocol) Em redes sem fio o DHCP é extremamente útil devido à facilidade com que o usuário passa a fazer parte da rede. O roteador é capaz de trabalhar tanto como servidor DHCP quanto como cliente.

DNS (Domain Name Service) Inicialmente considerou-se desnecessário prover um servidor de nomes. O sistema somente seria cliente DNS dos servidores já existentes na rede.

NAT (Network Address Translation) O sistema possui suporte a NAT 1:N e NAT 1:1.

Roteamento O ideal seria prover roteamento dinâmico, mas o *daemon* de roteamento do FreeBSD (routed) suporta apenas o protocolo RIP, cujo uso é impraticável em redes sem fio, devido à quantidade de mensagens de atualização de rotas enviadas aos pontos da rede. No momento está sendo usado apenas roteamento estático.

Filtro de Pacotes É um mecanismo essencial para a segurança da rede. O sistema provê filtragem como base em tipo de protocolo, tipo de pacote, endereço de origem e endereço de destino.

Modelagem de Tráfego (Traffic Shaping) Esse serviço é cada vez mais necessário nas redes atuais, pois possibilita limitar a parcela da banda consumida por cada um dos pontos de presença da rede.

SNMP Um agente SNMP permite coletar dados estatísticos da rede. Os administradores de rede necessitam desses dados, em especial o tráfego de pacotes, para otimizarem o funcionamento da rede. Foi instalada uma versão reduzida do net-snmp[1] somente para consulta de informações.

VPN (Virtual Private Network) permite trafegar informações de forma segura passando por redes não confiáveis. Esse serviço era necessário para que alguns usuários conectassem suas redes locais a redes remotas para compartilhamento de recursos.

3.3. Sistema operacional

Desde o inicio foi procurado um sistema operacional que permitisse o gerenciamento e monitoramento remoto e que, acima de tudo, suportasse os principais cartões wireless do mercado. Deveria também ser robusto ao ponto de não corromper o seu sistema de arquivos em caso de queda de energia e poder ser recuperado rapidamente em caso de um desastre maior.

Optou-se pelo FreeBSD [2], por sua flexibilidade, estabilidade e segurança. FreeBSD é derivado do BSD4.4 (da universidade da Califórnia, em Berkley) e é desenvolvido desde 1993 em um trabalho coletivo feito através da Internet. Por ser um software de código aberto, e de grande qualidade, revelou-se vantajoso em relação aos sistemas operacionais comerciais. Ao contrário do Linux, outro software muito popular, o FreeBSD é um *sistema operacional* e não um kernel com outras coisas por cima que o fazem, então, uma "distribuição".

3.4. Servidor sem disco rígido, monitor ou teclado

Em caso de uma queda de energia o sistema operacional testa o sistema de arquivos no próximo *boot*, identificando e corrigindo os erros. Em casos críticos, porém, é necessária a execução manual de comandos para o reparo. Isto não é desejável em um produto sem teclado nem monitor, e que muitas vezes ficará em locais de difícil acesso. A solução foi montar o sistema de arquivos em modo *read-only*. Dessa forma ele permaneceria intacto mesmo ocorrendo uma queda de energia ou outro tipo de problema no sistema.

O ideal seria não depender de meios mecânicos para armazenamento como disquete, disco rígido ou CD-ROM. Adotou-se então um *flash disk*, reconhecido pelo BIOS da máquina como um disco padrão IDE. Há flash disks de 16MB, 32MB, 64MB e maiores, com preços igualmente crescentes. Uma unidade de 32MB ficou num patamar aceitável de preço e tamanho para o projeto, mas não comportava sequer a instalação mínima do FreeBSD.

A solução foi criar um FreeBSD personalizado, que ocupa apenas 12MB. Para isto foi gerardo um *kernel* enxuto, que ocupa apenas 1,1MB. Os dispositivos do diretório /dev foram compactados também. No momento do boot do sistema o kernel e o diretório /dev são descompactados e colocados em um *ram disk*. O diretório /var também é montado em memória (os arquivos de registro do sistema são perdidos na hora de reinicialização). Obteve-se assim um sistema enxuto, que utiliza um meio de armazenamento quase imune a falhas a um custo baixo se comparado aos similares comerciais.

[1] `http://net-snmp.sourceforge.net/`
[2] `http://www.FreeBSD.org`

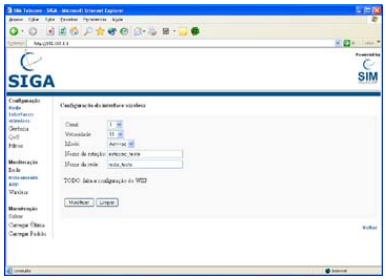

Figura 1: Interface de configuração via WWW

O problema do acesso ao sistema foi resolvido usando a porta serial como console. Quando necessário, basta conectar um terminal (ou um PC rodando um emulador de terminal).

3.5. Gerenciamento

A configuração do roteador pode ser feita de forma manual. Basta usar um terminal, ou iniciar uma sessão remota via SSH, e então editar os arquivos necessários. Para um usuário experiente do FreeBSD essa tarefa é trivial, mas para os novatos é um tanto complexa. Como um das metas do projeto é desenvolver um produto de fácil configuração e gerenciamento, a alternativa foi desenvolver uma interface de gerenciamento amigável, para que mesmo pessoas não especializadas pudessem administrá-lo.

A primeira opção considerada foi desenvolver um agente SNMP, que ficaria no roteador, e uma ferramenta de administração, que ficaria da estação usada para gerenciar o sistema. Esta solução se revelou pouco flexível, pois exigiria que o programa de administração fosse instalado em cada estação. Observou-se que seria muito mais simples desenvolver uma aplicação que rodasse no roteador, desempenhando as tarefas de gerenciamento, e fosse acessível via WWW. O agente SNMP continou a ser usado, mas apenas para coletar estatísticas de operação do roteador.

Usa-se um servidor HTTP capaz de autenticar usuários por sessão. Quando um usuário quer acessar a interface de administração, ele precisa fornecer um nome e uma senha válidos. A figura 1 mostra a tela de configuração de uma interface wireless. As configurações e informações são manipuladas através de scripts CGI escritos na linguagem de programação Perl (uma versão estática do interpretador que fornece todas as funções básicas em apenas 1.3MB). Apesar de a linguagem ser interpretada, não se observou qualquer problema de desempenho.

4. Evolução do sistema

O projeto foi colocado em operação em 25 provedores de acesso no Rio Grande do Sul em meados de 2003. Durante o período de aproximadamente um ano diversas correções de erros foram feitas e novas necessidades observadas. Melhorias acabaram sendo desenvolvidas para suprir essas necessidades. A seguir são descritas algumas dessas melhorias no sistema:

Suporte a PPPoE (PPP over Ethernet) Com a inclusão de suporte a PPPoE no kernel o sistema passou a funcionar como servidor PPPoE para clientes da LAN local, ao mesmo tempo em que pode trabalhar como cliente PPPoE autenticando-se em um servidor remoto. Os dados que trafegam na rede sao transportandos usando MPPE (Microsoft Point-To-Point Encryption Protocol) [Pall and Zorn, 2001] usando criptografia de 40 a 128 bits.

Suporte a PPTP O suporte a VPN inicialmente foi feito usando o aplicativo vtun [3], que tem como vantagens a alta taxa de compactação obtida, o que resulta em grande desempenho no tráfego dos dados, e a facilidade de configuração. Entretanto o vtun usa seu próprio protocolo de comunicação sobre TCP e UDP e não suporta PPTP, L2TP ou IPsec. É necessário que tanto o lado cliente quanto o servidor rodem em plataforma Unix. Como era necessário fazer o sistema autenticar em servidores Windows, que suportam apenas PPTP e L2TP, substituiu-se o vtun pelo PPTP Client.

Suporte a IPSec no kernel para quando houver a necessidade de conexões com criptografia forte de 128 bits. A troca de chaves dinâmicas foi possibilitada devido ao uso do racoon[4] para gerenciamento das mesmas.

Troca de pacotes de firewall e nat Os aplicativos IPFW e NATD existem desde a versão 2.0 do FreeBSD. Eles são eficientes, mas não oferecem alguns recursos necessários ao sistema, como por exemplo NAT para mais de um endereço IP e construção de regras de firewall mais flexiveis, robustas e enxutas. Dessa forma passou-se a usar o IPFilter em conjunto com o IPNat, disponíveis no FreeBSD versão 5, para fazer firewall e NAT. Isto tornou necessário atualizar a versão do sistema operacional utilizada.

Troca de *traffic shaper* Durante a transmissão de dados os pacotes que chegam ao sistema são colocados em uma fila de espera até serem tratados pelo sistema e então despachados pela interface adequada segundo critérios de roteamento. O algoritmo normalmente usado para despachar os pacotes nessa baseia-se e um FIFO (Firt In First Out). Como o sistema passou a ser utilizado para roteamento de VoIP (Voice over IP) surgiram os seguintes problemas:

latência O atraso geralmente é ocasionado pela concorrência entre os vários serviços pela da banda de rede existente.

jitter É o resultado de pacotes que chegam ao destino em intervalos irregulares. Este problema afeta drasticamente a qualidade da transamissão de voz.

perda de pacotes Devido aos pacotes de voz serem transmitidos usando UDP, não há garantia de entrega. Pacotes que forem perdidos acabam tornando uma conversa ininteligível.

Esses problemas eram causados pelo dummynet, utilizado como *traffic shaper*. Ele utiliza o algoritmo WFQ para gerenciar a fila de pacotes e por isto não consegue prover uma real qualidade do serviço (QoS).

O pacote ALTQ (Alternate Queueing) [Cho, 2004], portado do OpenBSD para o FreeBSD, provê um conjunto de políticas de gerenciamento de filas e componentes para QoS. Para habilitar suporte ao mesmo foi necessário aplicar um *patch* no kernel e então recompilar alguns módulos e binários.

Com a inclusão do ALTQ o sistema passou a ter suporte múltiplas políticas de gerenciamento de banda. Atualmente estão sendo usadas somente CBQ (Class-Based Queueing) [Floys and Jacobson, 1995] e PRIQ (Priority Queue), os quais garantem além de gerenciamento de quantidade de banda disponível ao cliente a possibilidade de dar maior prioridade aos pacotes de voz, fazendo com que sejam despachados antes que os demais.

5. Conclusões e planos futuros

O trabalho apresentado neste artigo foi desenvolvido inicialmente para solucionar alguns problemas encontrados no dia-a-dia em redes sem fios. Como resultado disto, foi criado

[3] `http://vtun.sourceforge.net/`
[4] `http://www.kame.net/racoon/`

um produto que integra diversos conceitos e tecnologias, é versátil, seguro e de baixo custo e que apresenta várias vantagens em relação a um AP comercial.

Como exemplo, podemos citar o equipamento AP2000, da Proxim. Seu custo é de aproximadamente R$ 2300,00 e ele não é capaz de prestar alguns dos serviços mais simples, como roteamento, QoS, NAT, autenticação PPTP, tunelamento. Além disso seu suporte a *firewall* se resume a um filtro de pacotes baseado em identificação de protocolo e número de porta.

O SIM Gateway passou rapidamente do meio acadêmico para o uso comercial em empresas provedoras de serviços de conexão sem fios via rádio no Estado do RS.

O uso de software de código aberto foi um fator decisivo para o sucesso do empreendimento. Apesar do caráter comercial do produto, não foram feitas alterções proprietárias no software utilizado, mantendo a total compatibilidade com o software como ele é distribuído. Isto garante a possibilidade de usar as novas versões que venham a ser desenvolvidas no futuro.

Mesmo sendo o produto considerado atualmente completo, em ambiente de produção, identificamos três necessidades que devem ser atendidas no futuro. A primeira delas é uma atualização da interface de configuração via WWW, para adaptá-la aos novos serviços que foram incluídos depois de estar pronta a versão inicial.

A segunda necessidade é a inclusão de um servidor de nomes, pois atualmente o sistema atua apenas como cliente. O principal empecilho para isto é a limitação de espaço disponível no *flash disk*.

A maior dificuldade ainda está em suportar roteamento dinâmico. Usar OSPF seria a melhor alternativa, mas até o momento não foi encontrado um servidor OSPF com um tamanho aceitável para ser armazenado no *flash disk*.

Referências

Cho, K. (2004). Altq: Alternate queueing for BSD UNIX. `http://www.csl.sony.co.jp/person/kjc/kjc/software.html#ALTQ`.

Flickenger, R. (2002). *Building Wireless Community Networks*. O'Reilly & Associates, Sebastopol.

Floys, S. and Jacobson, V. (1995). Link-sharing and resource management models for packet networks. *IEEE/ACM Transactions on Networking*, 3(4):365–386.

Intersil (2003). PRISM II: the complete "antenna-to-computer" solution. `http://www.intersil.com/data/fn/fn4904.pdf`.

ORiNOCO (2003). ORiNOCO world pc card. `http://www.orinocowireless.com/products/all/orinoco/docs/ds/PC-card.pdf%`.

Pall, G. and Zorn, G. (2001). Microsoft Point-To-Point Encryption (MPPE) protocol. `http://www.ietf.org/rfc/rfc3078.txt`.