1º Capítulo

1.1 O que é Internet?

1.1.1 Uma descrição detalhada da rede:

A Internet pública é uma rede de computadores mundial, isto é, uma rede que interconecta milhões de equipamentos de computação em todo o mundo.O termo rede de computadores está começando a soar um tanto desatualizado, dados os muitos equipamentos não tradicionais que estão sendo ligados à Internet, todos esses equipamentos são denominados **hospedeiros** ou **sistemas finais**.
Sistemas finais são conectados entre si por **enlaces** (links) **de comunicação.** Enlaces diferentes podem transmitir dados em taxas diferentes, sendo a **taxa de transmissão**(banda passante) de um enlace medida em bits/s.
Em geral, sistemas finais não são interligados diretamente por um único enlace de comunicação, em vez disso, são interligados indiretamente por equipamentos intermediários de comutação conhecidos como **comutadores de pacotes** (tipos mais predominantes: roteadores e comutadores de camada de enlace).Eles encaminham a informação que está chegando em um de seus enlaces de entrada para um de seus enlaces de saída**.** Em redes de computadores o bloco de informação é denominado **pacote**.
A seqüência de enlaces de comunicação e comutadores de pacotes que um pacote percorre desde o sistema final remetente até o sistema final receptor é conhecida como **rota** ou **caminho** através da rede. A Internet usa uma técnica conhecida com **comutação de pacote,** que permite que vários sistemas finais comunicantes compartilhem ao mesmo tempo um caminho ou partes dele.
Sistemas finais acessam a Internet por meio de **Provedores de Serviços de Internet (*ISPs*)**. Cada ISP é uma rede de comutadores de pacotes e enlaces de comunicação. ISPs provêem aos sistemas finais uma variedade de tipos de acesso à rede (acesso por modem discado, banda larga) e acesso a provedores de conteúdo, conectando sites Web diretamente à Internet. Para permitir a comunicação entre usuários da Internet e possibilitar a usuários acesso mundial ao conteúdo da Internet, ISPs de nível mais baixo são interconectados por meio de ISPs de nível mais alto(consiste em roteadores de alta velocidade interconectados com enlaces de fibra ótica de alta velocidade).
Os sistemas finais, os comutadores de pacotes e outras peças da Internet executam **protocolos** que controlam o envio e o recebimento de informações dentro da Internet. O **TCP** e o **IP** são dois dos protocolos mais importantes da Internet. O protocolo IP especifica o formato dos pacotes que são enviados e recebidos entre roteadores e sistemas finais. Os principais protocolos da Internet são conhecidos coletivamente como **TCP/IP.**
Os **padrões da Internet** são desenvolvidos pela IETF e os documentos padronizados dela são denominados **RFCs.**
Existem redes privadas cujos hospedeiros não podem trocar mensagens com hospedeiros que estão fora da rede privada, elas são denominadas **intranets**, pois usam o mesmo tipo de hospedeiros,roteadores,enlaces e protocolos da Internet pública.

1.1.2 Uma descrição do serviço:

A Internet permite que **aplicações distribuídas** que executam em seus sistemas finais troquem dados entre si. Ex de aplicações: navegação na Web, mensagem instantânea, áudio e vídeo em tempo real, etc.
A Internet provê dois serviços a suas aplicações distribuídas: um **serviço confiável orientado para conexão** (garante que os dados transmitidos de uma origem a um destino sejam finalmente entregues ao destinatário em ordem e completos) e um **serviço não confiável não orientado para conexão** (não oferece nenhuma garantia quanto à entrega final). A aplicação distribuída usa um ou outro desses serviço, mas não ambos.
Atualmente , a Internet não provê um serviço que ofereça garantias quanto ao tempo que gastará para levar os dados da origem ao destino.

1.1.3 O que é um protocolo?

Um protocolo de rede é semelhante a um protocolo humano, a diferença é que as entidades que trocam mensagens e realizam ações são componentes de hardware ou software de algum equipamento. Todas as atividades na Internet que envolvem duas ou mais entidades remotas comunicantes são governadas por um protocolo.
Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão e/ou no recebimento de uma mensagem ou outro evento.

1.2 A periferia da Internet

1.2.1 Sistemas finais, clientes, servidores:

Os computadores conectados à Internet são usualmente chamados de **sistemas finais(**computadores de mesa,servidores, computadores móveis**)**, por que estão na periferia da internet. Sistemas finais também são denominados hospedeiros(hosts) por que hospedam programas de aplicação. Às vezes, sistemas finais são ainda subdivididos em duas categorias: **clientes** e **servidores.** Normalmente, clientes costumam ser PCs de mesa ou móveis, enquanto servidores tendem a ser máquinas mais poderosas que armazenam e distribuem páginas Web, vídeo em tempo real, etc. Um programa cliente é um programa que funciona em um sistema final, que solicita e recebe um serviço de um programa servidor, que funciona em outro sistema final. Uma vez que um programa cliente normalmente roda em um computador e o programa servidor, em outro, aplicações cliente-servidor de Internet são, por definição, **aplicações destribuídas.**Eles interagem enviando mensagens um para o outro pela Internet. Nem todas as aplicações de hoje consistem em programas puramente clientes ou servidores, as aplicações P2P de compartilhamento de arquivos populares, o sistema final do usuário funciona como um programa cliente(quando requisita um arquivo de outro par) e também como um programa servidor(quando envia um arquivo para outro par).

1.2.2 Serviço não orientado para conexão e serviço orientado para conexão:

Serviço orientado para conexão: Quando uma aplicação usa o serviço orientado para conexão, o programa cliente e o programa servidor(sistemas finais diferentes) enviam pacotes de controle um para o outro antes de remeter pacotes com dados reais que deverão ser transferidos.Este procedimento,apresentação, alerta o cliente e o servidor para se preparem para uma rajada de pacotes.Quando o processo de apresentação for concluído uma conexão foi estabelecida entre os dois sistemas finais. Os serviços orientados para conexão providos pela Internet vêm conjugados com diversos outros serviços, entre eles a **transferência de dados confiável**, que quer dizer que uma aplicação pode confiar que a conexão entregará todos os seus dados sem erro e na ordem certa. A confiabilidade na Internet é conseguida por meio da utilização de confirmações e retransmissões. O **controle de fluxo** garante que nenhum dos lados de uma conexão sobrecarregue o outro enviando demasiados pacotes muito rapidamente. O serviço de **controle de congestionamento** da Internet ajuda a evitar que ela trave, quando os comutadores de pacotes ficam congestionados, seus buffers podem transbordar e pode ocorrer perda de pacotes. O serviço orientado para conexão da Internet tem um nome – **Protocolo de Controle de Transmissão(TCP).** Entre os serviços que o TCP provê a uma aplicação estão transporte confiável, controle de fluxo e controle de congestionamento.

Serviço não orientado para conexão: Não há apresentação mútua no serviço não orientado para conexão da Internet. Quando um lado quer enviar pacotes ao outro lado, ele simplesmente envia. Como não processo de apresentação, os dados podem ser entregues mais rápido, o que torna esse serviço ideal para aplicações simples orientadas para transação. Porém, como não há nenhuma transferência confiável de dados, uma fonte nunca tem certeza de quais pacotes chegaram ao destino. Este serviço também não provê controle de fluxo, nem de congestionamento. O serviço de Internet não orientado para conexão é denominado **Protocolo de Datagrama do Usuário(UDP).**

1.3 O núcleo da rede

1.3.1 Comutação de circuitos e comutação de pacotes:

Há duas abordagens fundamentais para montagem de um núcleo de rede: **comutação de circuitos e comutação de pacotes.** Em redes de comutação de circuitos, os recursos necessários ao longo de um caminho para prover comunicação entre sistemas finais são reservados pelo período da sessão de comunicação. Em redes de comutação de pacotes , esses recursos não são reservados; as mensagens de uma sessão usam os recursos por demanda, e como conseqüência, poderão ter de esperar para conseguir acesso a um enlace de comunicação. As redes de telefonia são exemplos de redes de comutação de circuitos. A Internet é um exemplo de rede de comutação de pacotes.

Comutação de circuitos: Quando dois sistemas finais querem se comunicar, a rede estabelece uma conexão fim-a-fim dedicada entre dois sistemas finais. Para que o sistema final A envie mensagens ao sistema B, a rede deve primeiramente reservar um circuito em cada dois enlaces.

Multiplexação: Um circuito é implementado em um enlace por **multiplexação por divisão de freqüência(FDM)** ou por **multiplexação por divisão de tempo(TDM)**. Com FDM, o espectro de freqüência de um enlace é compartilhado entre as conexões

estabelecidas através do enlace que reserva uma banda de freqüência para cada conexão durante o período da ligação.Em um enlace TDM, o tempo é dividido em quadros de duração fixa, e cada quadro é dividido em um numero fixo de compartimentos(slots) Quando estabelece uma conexão por meio de um enlace, a rede dedica à conexão um compartimento de tempo em cada quadro.Esses compartimentos são reservados para o uso exclusivo dessa conexão, e um dos compartimentos de tempo (em cada quadro) fica disponível para transmitir os dados dela.

Comutação de pacotes: Em redes de computadores modernas, o originador fragmenta mensagens longas em porções de dados menores, os pacotes. Entre origem e destino, cada um desses pacotes percorre enlaces de comunicação e comutadores de pacotes. Pacotes são transmitidos por casa enlace de comunicação a uma taxa igual à de transmissão total do enlace. A maioria dos comutadores de pacotes armazena e reenvia os pacotes nas entradas dos enlaces, onde ele deve receber o pacote inteiro antes de poder começar a transmitir o primeiro bit do pacote para o enlace de saída.Assim, eles introduzem um atraso de armazenamento e reenvio na entrada de cada enlace(se um pacote consiste em L bits e deve ser reenviado por um enlace de saída de R bps, então atraso = L/R).

Comutação de pacotes *versus* comutação de circuitos: multiplexação estatística: Comutação de circuitos aloca previamente a utilização do enlace de transmissão independentemente de demanda, com desperdício de tempo de enlace desnecessário alocado e não utilizado. Comutação de pacotes , por outro lado, aloca utilização de enlace por demanda. A capacidade de transmissão do enlace será compartilhada pacote por pacote somente entre os usuários que tenham pacotes que precisem ser transmitidos pelo enlace. Tal comportamento de recursos por demanda (e não por alocação prévia) às vezes é denominado **multiplexação estatística** de recursos.

1.3.2 Redes de comutação de pacotes: redes datagramas e redes de circuitos digitais

Há duas grandes classes de redes de comutação de pacotes: redes de datagramas(qualquer rede que transmite pacotes segundo endereços de sistemas finais de destino, ex: roteadores) e redes de circuitos virtuais(qualquer rede que transmita pacotes segundo números de circuitos virtuais).

Redes de circuitos virtuais: Um circuito virtual pode ser imaginado como uma conexão virtual entre um sistema final de origem e um sistema final de destino. Um identificador de circuito virtual será atribuído a CV quando ele for estabelecido pela primeira vez entre a fonte e o destino. Qualquer pacote que faça parte do CV terá esse identificador em seu cabeçalho. Quando um pacote chega a um comutador de pacotes, este examina o ID CV, indexa a sua tabela e transmite o pacote ao enlace de saída. Um comutador em uma rede de CVs mantém informação de estado para suas conexões em curso, cada vez que uma nova conexão é ativada por um comutador, um novo registro de conexão deve ser adicionado à tabela de tradução dele, e cada vez que uma conexão é desativada, um registro deve ser removido da tabela.

Redes de datagramas: As redes de datagramas são análogas, em muitos aspectos, ao serviço postal. Quando um remetente envia uma carta a um destino, ele a coloca em um envelope e nele escreve o endereço do destinatário. Em uma rede de datagramas, cada

pacote que transita por ela contém em seu cabeçalho o endereço de destino. Quando um pacote chega a um comutador de pacotes da rede, ele examina uma parte do endereço de destino do pacote e o remete a um comutador adjacente. Redes de datagramas não mantém informação de estado de conexão em seus comutadores.

1.4 Redes de acesso e meios físicos

1.4.1 Redes de acesso

Acesso Residencial: Refere-se à conexão de um sistema final residencial a um roteador de borda(primeiro roteador de um caminho entre um sistema final e qualquer outro sistema final remoto). Uma forma comum de acesso residencial é o **modem discado** ligado por uma linha telefônica a um ISP residencial. O modem converte o sinal digital de saída do PC em formato analógico para transmissão pela linha telefônica(é a mesma usada para fazer chamadas telefônicas normais). Na outra extremidade da linha telefônica analógica , um modem no ISP converte o sinal analógico novamente para sinal digital para entrar no roteador ISP. A rede de acesso é simplesmente um par de modens juntamente com uma linha telefônica ponto a ponto discada. Como o acesso discado é lento e impede a utilização normal da linha telefônica, novas tecnologias surgiram, como a banda larga que oferece taxas mais altas de bits a usuários residenciais, além de fornecer meios para que os usuários acesse a Internet e falem ao telefone ao mesmo tempo. Há dois tipos comuns de acesso banda larga, o DLS e o HFC. O acesso DSL normalmente é fornecido por uma companhia telefônica, às vezes em parceria com um ISP independente. A DSL usa multiplexação por divisão de freqüência. Diferentemente de modens discados, as DSLs foram explicitamente projetadas para distancias curtas entre modens residenciais e modens de ISP, o que permite taxas de transmissão substancialmente mais altas do que as de acesso discado. Enquanto DSL e modens discados usam linhas telefônicas comuns, redes de acesso HFC são extensões das redes de cabos existentes usadas para transmissão de TV a cabo. Como acontece com a DSL, o HFC requer modens especiais, denominados modens a cabo, eles dividem a rede HFC em dois canais, um canal na direção do usuário(descida) e um na direção do provedor(subida). Como na DSL, taxa de transmissão de descida normalmente é maior do que a de subida. A rede HFC é um meio de transmissão compartilhado. DSL e HFC possuem serviços sempre disponíveis.

Acesso corporativo: Refere-se à conexão de sistemas finais de uma empresa ou instituição educacional à rede(roteador de borda). Nesses locais, normalmente é usada uma rede local(LAN). O roteador de borda é responsável pelo roteamento de pacotes cujo destino é externo à LAN. A tecnologia Ethernet é, hoje, a que predomina em redes corporativas, como o HFC, ela usa um meio compartilhado, de modo que usuários finais compartilham a velocidade de transmissão da LAN.

Acesso sem fio: Há duas categorias amplas de acesso sem fio à Internet Nas LANS sem fio, os usuários sem fio transmitem/recebem pacotes de/para uma estação-base dentro de um raio de algumas dezenas de metros. No acesso sem fio usa-se um espectro de rádio para conectar um sistema final portátil a uma estação-base, que estará conectada um roteador de borda.

1.4.2 Meios físicos:

Os meios físicos se enquadram em duas categorias, **meios guiados e meios não guiados**. Nos meios guiados, as ondas são dirigidas ao longo de um meio sólido, tal como um cabo de fibra ótica, um par de fios de cobre trançado ou um cabo coaxial. Nos meios não guiados, as ondas se propagam na atmosfera e no espaço, como é o caso de uma LAN sem fio ou de um canal digital de satélite. O custo de um enlace físico é em geral relativamente insignificante em comparação a outros custos da rede.

Par de fios de cobre trançado: Meio de transmissão guiado mais barato e mais comumente usado. Esse par é constituído de dois fios de cobre isolados, cada um com aproximadamente um milímetro de espessura, enrolados em espiral. Os fios são trançados para reduzir a interferência elétrica de pares semelhantes que estejam próximos. Tecnologia de modem discado e DSL usam pares trançados.

Cabo coaxial: Como o par trançado, o cabo coaxial é constituído de dois condutores de cobre, porém concêntricos e não paralelos, é bidirecional. Cabos coaxiais são muito comuns em sistemas de televisão a cabo.

Fibras óticas: A fibra ótica é um meio delgado e flexível que conduz pulsos de luz, sendo que cada um desses pulsos representa um bit. Suporta taxas de transmissão elevadíssimas. Fibras óticas são imunes à interferência eletromagnética, têm baixíssima atenuação de sinal de até cem km e são muito difíceis de derivar.Essas características fizeram da fibra ótica o meio preferido para a transmissão guiada de grande alcance, em particular para cabos submarinos.

Canais de rádio terrestres: Canais de rádio carregam sinais dentro do espectro eletromagnético. Sua instalação não requer cabos físicos, podem atravessar paredes, dão conectividade ao usuário móvel e, podem transmitir um sinal a longas distancias. Suas características dependem do ambiente de propagação e da distancia pela qual o sinal deve ser transmitido. Condições ambientais determinam perda de sinal no caminho e atenuação por efeito de sombra e interferência. Podem ser classificados em dois grupos, os de pequeno alcance, e os de longo alcance.

Canais de rádio por satélite: Existem dois tipos de satélites usados para comunicação, satélites geoestacionários e satélites de baixa altitude. Os geoestacionarios ficam permanentemente sobre o mesmo lugar na Terra, estão a grandes distancias da Terra e causam atraso de propagação. Os de baixa altitude são posicionados muito mais próximos da Terra e não ficam permanentemente sobre um único lugar, giram ao redor da Terra.

1.5 ISPs e backbones da Internet

Na Internet pública, redes de acessos situadas na borda da Internet são conectadas ao restante segundo uma hierarquia de níveis de ISPs. Os ISPs de acesso estão no nível mais baixo dessa hierarquia. No topo dela está um numero relativamente pequeno de ISPs denominados ISPs de nível 1, eles apresentam as seguintes características : conectam-se diretamente a cada um dos outros ISPs de nível 1, conectam-se a um grande numero de ISPs de nível 2 e a outras redes clientes, têm cobertura internacional. Esses ISPs também são conhecidos como redes de **backbone da Internet**.

Um ISP de nível 2 normalmente tem alcance regional ou nacional e conecta-se apenas a uns poucos ISPs de nível 1. Para alcançar uma grande parcela da Internet global, um ISP de nível 2 tem de direcionar o tráfego por um dos ISPs de nível 1 com o qual está conectado o qual é o **provedor** do **cliente** (ISP de nível 2). Uma rede de nível 2 também pode preferir conectar-se diretamente a outras redes de mesmo nível. Alguns provedores de nível 1 também são provedores de nível 2(integrados verticalmente). Quando dois ISPs estão ligados diretamente um ao outro são denominados **peers** um do outro.
Dentro de uma rede de um ISP, os pontos em que ele se conecta a outros ISPs são conhecidos como **pontos de presença**(POP). Um POP é simplesmente um grupo de um ou mais roteadores na rede do ISP com os quais roteadores em outros ISPs, ou em redes pertencentes a clientes do ISP, podem se conectar. Além de se conectarem entre si em pontos privados de formação de par, ISPs muitas vezes se interconectam em **pontos de acesso de rede** (NAPs), podendo cada um deles ser controlado e operado por alguma empresa privada de telecomunicações ou por um provedor de backbone de Internet, eles trocam enormes volumes de trafego entre muitos ISPs.
A topologia da Internet é complexa, consistindo em dezenas de ISPs de níveis 1 e 2 e milhares de ISPs de níveis mais baixos. A cobertura dos ISPs é bastante diversificada. Os ISPs de níveis mais baixos conectam-se a ISPs de níveis mais altos e estes se interconectam em pontos privadosde emparelhamento(onde ISPs de nível 1 fazem interconexão entre si) e NAPs(onde ISPs de nível 2 fazem interconexão entre eles mesmos e de nível 1).

1.6 Atraso e perda em redes de comutação de pacotes

Quando um pacote viaja da sua origem para seu destino, ele sofre, ao longo do caminho, diversos tipos de atraso em cada nó(sistema final ou roteador) existente no caminho.

1.6.1 Tipos de atraso

Um pacote é enviado do nó anterior por meio do roteador A até o roteador B, o roteador A possui um enlace de saída que o leva ao B, este enlace e precedido de uma fila(buffer). Quando o pacote chega em A, o roteador examina o cabeçalho do pacote para determinar o enlace de saída apropriado e então o direciona ao mesmo.

Atraso de processamento: O tempo requerido para examinar o cabeçalho do pacote e determinar para onde direcioná-lo é parte do atraso de processamento, que pode incluir outros fatores, como o tempo necessário para verificar erros em bits existentes. Geralmente esses atrasos são da ordem de microssegundos.

Atraso de fila: O pacote sofre um atraso de fila enquanto espera para ser transmitido no enlace. O tamanho desse atraso dependerá da quantidade de outros pacotes que chegaram antes e que já estiverem na fila. Se a fila estiver vazia, então o tempo de atraso de fila do pacote será zero, porém se o trafego estiver pesado e houver muitos pacotes também esperando, o atraso será longo.Esses atrasos são da ordem de micro a milissegundos.

Atraso de transmissão: Um pacote é transmitido depois que todos os pacotes que chegaram antes dele tenham sido enviados. O atraso de transmissão é **L/R** (onde L é o tamanho do pacote e R é a velocidade de transmissão do enlace do roteador A ao

roteador B). Esta é a quantidade de tempo requerida para empurrar(transmitir) todos os bits do pacote para o enlace. Da ordem de micro a milissegundos.

Atraso de propagação: Assim que lançado no enlace, um bit precisa se propagar até o roteador B. O tempo necessário para propagar o bit desde o inicio do enlace até o roteador B é o atraso de propagação. O atraso de propagação é **D/S** (onde D é a distancia entre os dois roteadores e S é a velocidade de propagação do enlace).

Comparação entre atrasos de transmissão e de propagação: O atraso de transmissão é a quantidade de tempo requerida para o roteador empurrar o pacote para fora(não tem nada a ver com a distancia entre os dois roteadores) e atraso de propagação é o tempo que leva para um bit se propagar de um roteador ao outro(não tem nada a ver com o tamanho do pacote).

O atraso nodal(em um único roteador) é a soma de todos os atrasos.

1.6.2 Atraso de fila e perda de pacote:

Quando o atraso de fila é grande ou insignificante?
Depende da **intensidade de tráfego** a qual é dada por **L.A/R** (onde L é o tamanho de todos os pacotes, A é a taxa média com que os pacotes chegam à fila e R é a taxa com que os bits são retirados da fila). Se a intensidade for maior que 1, então a velocidade média com que os bits chegam à fila excederá a velocidade com que eles podem ser transmitidos para fora, então a fila irá aumentar sem limite e o atraso tenderá ao infinito. Porém se a intensidade for próxima a zero, então as chegadas de pacotes serão poucas e bem espaçadas e é improvável que um pacote que esteja chegando encontre outro na fila.

Perda de pacote: Quando um pacote chega e encontra uma fila cheia, sem espaço disponível para armazena-lo, o roteador descartará esse pacote, isto é, ele será perdido.

1.6.3 Atraso e rotas na Internet

Utilizando o Traceroute, o caminho da origem ao destino será mostrado, onde os pacotes passam por uma série de roteadores. O resultado é formado por seis colunas, a primeira indica o numero do roteador ao longo da rota, a segunda o nome delem a terceira o endereço do roteador, as três ultimas indicam os atrasos de ida e volta para as três tentativas

1.7 Camadas de protocolo e seus modelos de serviço

1.7.1 Arquitetura de camadas

Camada de aplicação: É onde residem aplicações de rede e seu protocolos. Inclui muitos protocolos como o HTTP, SMTP e o FTP.

Camada de transporte: Transporta mensagens da camada de aplicação entre os lados do cliente e servidor. Há dois protocolos de transporte na Internet, o TCP e o UDP, e qualquer um deles pode levar mensagens de camada de aplicação.

Camada de rede: É responsável pela movimentação, de uma máquina para outra, de pacotes de camada de rede conhecidos como datagramas. Tem dois componentes principais. Um deles é o protocolo que define os campos no datagrama, bem como o modo como os sistemas finais e os roteadores agem nesses campos(protocolo IP). O outro componente é o protocolo de roteamento que determina as rotas que os datagramas seguem entre origens e destinos.

Camada de enlace: Roteia um datagrama por meio de uma série de comutadores de pacotes entre a origem e o destino. Pacotes de camada de enlace serão denominados quadros.

Camada física: A tarefa da camada física é de movimentar os bits individuais que estão dentro do quadro de um nó para o seguinte.

1.7.2 Camadas, mensagens, segmentos, datagramas e quadros.

Uma mensagem de camada de aplicação na máquina emissora é passada para a camada de transporte, esta pega a mensagem e anexa informações adicionais que serão usadas pela camada de transporte do receptor. A mensagem de camada de aplicação e as informações de cabeçalho da camada de transporte, justas, constituem o segmento de camada de transporte, que encapsula a mensagem de camada de aplicação.A camada de transporte então passa o segmente à camada de rede que adiciona informações de cabeçalho de camada de rede, como endereços de sistemas finais de origem e de destino, criando um datagrama de camada de rede, este então é passado para a camada de enlace que adiciona suas informações e cria um quadro de camada de enlace. O nome disso é **encapsulamento**.

1.8 História das redes de computadores e da Internet

.....

2º Capítulo

2.1 Princípios de aplicações de rede

Exemplos de aplicações de rede: correio eletrônico, a Web, mensagem instantânea, login em computador remoto como Telnet e SSH, compartilhamento de arquivos P2P, transferência de arquivos, etc.
O cerne do desenvolvimento de aplicação de rede é escrever programas que rodem em sistemas finais diferentes e se comuniquem entre si pela rede. Por exemplo, a na Web há dois programas distintos que se comunicam, o programa do browser (roda na máquina do ususário) e o programa servidor Web (roda na máquina do servidor Web).

2.1.1 Arquiteturas de aplicação de rede:

A arquitetura da aplicação determina como a aplicação é organizada nos vários sistemas finais.
Em uma arquitetura cliente-servidor há um hospedeiro sempre em funcionamento, denominado servidor, que atende a requisições de muitos outros hospedeiros, denominados clientes,estes podem estar em funcionamento às vezes ou sempre. Os clientes não se comunicam diretamente uns com os outros. O servidor tem um endereço fixo, denominado endereço de IP, o cliente sempre pode contatá-lo, enviando um pacote ao endereço do servidor. Em aplicações cliente-servidor, muitas vezes aocntece de um único hospedeiro servidor ser incapaz de atender a todas as requisições de seus clientes, por essa razão, muitas vezes são utilizados conjuntos de hospedeiros (**server farm)** para criar servidor virtual poderoso em arquiteturas cliente-servidor.
Em uma arquitetura P2P pura, não há um servidor sempre funcionando no centro da aplicação, em vez disso pares arbitrários de hospedeiros comunicam-se diretamente entre si. Como os pares se comunicam sem passar por nenhum servidor especial, a arquitetura é denominada peer-to-peer, onde nela nenhuma das máquinas participantes precisa estar sempre em funcionamento. Um de suas características mais fortes é a escalabilidade, onde cada par adicional não apenas aumenta a demanda, mas também a capacidade de serviço. Por outro lado, devido à sua natureza altamente distribuída e descentralizada, pode ser difícil de gerenciar aplicações P2P.
Muitas aplicações são organizadas segundo arquiteturas hibridas cliente/servidor/P2P, a Napster era um exemplo disso, no sentido de que era P2P porque arquivos MP3 eram trocados diretamente entre pares, sem passar por servidores dedicados, sempre em funcionamento, mas também era cliente-servidor, já que um par consultava um servidor central para determinar quais pares que estavam em funcionamento tinham um arquivo MP3 desejado.

2.1.2 Comunicação entre processos:

No jargão de sistemas operacionais, na verdade não são programas que se comunicam, mas processos, que podem ser imaginados com programas que estão sendo rodados dentro de um sistema final. Processos que rodam em sistemas finais diferentes se comunicam pela troca de mensagens por meio da rede de computadores.

Processos clientes e processos servidores: Para cada par de processos comunicantes normalmente rotulamos um dos dois processos de cliente(que inicia a comunicação) e o outro de servidor(processo que é contatado para iniciar a sessão).
Na Web, um processo browser inicia o contato com um processo do servidor Web(cliente) e o processo do servidor Web é o servidor. No compartilhamento de arquivos P2P, quando o ParA solicita ao ParB o envio de um arquivo especifico, o ParA é o cliente enquanto o ParB é o servidor.

Sockets: Um processo envia mensagens para a rede e recebe mensagens dela através de seu **socket**. Um analogia para se entender: Um processo é análogo a uma casa e seu socket à porta da casa, quando um processo quer enviar uma mensagem a um outro processo em outro hospedeiro, ele a empurra porta(socket) afora para dentro da rede, ao chegar ao hospedeiro destinatário, a mensagem passa através da porta(socket) do processo receptor. Um socket é a interface entre a camada de aplicação e a de transporte dentro de uma máquina.

Endereçamento de processos: Para que um processo em um hospedeiro envie uma mensagem a um processo em outro, o processo de origem tem de identificar o processo destinatário. Para isso ele precisa de duas informações, o nome ou o endereço da máquina hospedeira e um identificador que especifique o processo destinatário.
O processo destinatário é identificado por seu **endereço de IP** que é uma quantidade de 32 bits que identifica exclusivamente o sistema final. Além de saber o endereço, o processo de origem tem de identificar o processo que está rodando no outro hospedeiro, um **número de porta** de destino atende a essa finalidade.

2.1.3 Protocolos de camada de aplicação:

Um protocolo de camada de aplicação define como processos de uma aplicação, que funcionam em sistemas finais diferentes, passam mensagens entre si, em particular ele define os tipos de mensagens trocadas, a sintaxe dos vários tipos de mensagem, a semântica dos campos, regras para determinar quando e como um processo envia e responde mensagens. Muitos protocolos de camada de aplicação são proprietários e não estão disponíveis ao publico. É importante distinguir aplicações de rede de protocolos de camada de aplicação, um protocolo de camada de aplicação é apenas um pedaço de aplicação de rede.

2.1.4 De que serviços uma aplicação necessita?

Transferência confiável de dados: Algumas aplicações exigem transferência de dados totalmente confiável, isto é, não pode haver perda de dados(que pode ter conseqüências devastadoras). Outras aplicações podem tolerar uma certa perda de dados, mais notavelmente aplicações de multimídia.

Largura de banda: Algumas aplicações têm de transmitir dados a uma certa velocidade para serem efetivas. Se essa largura de banda não estiver disponível, a aplicação precisará codificar a uma taxa diferente ou então desistir, já que receber metade da largura de banda que precisa de nada adianta para tal aplicação sensível à largura de banda. Embora aplicações sensíveis à largura de banda exijam uma dada quantidade de largura de banda, aplicações elásticas(correio eletrônico, transferência de arquivos,etc)

podem fazer uso de qualquer quantidade mínima ou máxima que por acaso esteja disponível.

Temporização: O requisito final de serviço é a temporização. Aplicações interativas em tempo real, exigem limitações estritas de temporização na entrega de dados para serem efetivas.

2.1.5 Serviços providos pelos Protocolos de Transporte da Internet:

Serviços do TCP: O modelo do serviço TCP inclui um serviço orientado para conexão e um serviço confiável de transferência de dados. Faz com que o cliente e o servidor troquem informações de controle de camada de transporte antes que as mensagens de camada de aplicação comecem a fluir. Depois da apresentação diz-se que existe uma conexão TCP. A conexão é full-duplex(simultânea), visto que os dois processos podem enviar mensagens um ao outro pela conexão ao mesmo tempo. Os processos podem confiar no TCP para a entrega de todos os dados enviados sem erro e na ordem correta. O TCP também inclui um controle de congestionamento(voltado ao bem estar geral da Internet e não ao beneficio direto dos processos comunicantes) que limita a capacidade de transmissão de um processo quando a **rede** está congestionada. O TCP não garante uma taxa de transmissão mínima, o processo de origem não pode transmitir com a taxa que quiser, pois a taxa é regulada pelo controle de congestionamento. Ele também não garante absolutamente nenhum limite de tempo para que os dados cheguem ao receptor, ele garante a entrega de todos os dados, mas não dá nenhuma garantia quanto à velocidade de entrega ou aos atrasos.

Serviços do UDP: Protocolo de transporte simplificado, não orientado para conexão, portanto não há apresentação antes que os dois processos comecem a se comunicar. Provê um serviço não confiável, ele não oferece nenhuma garantia de que a mensagem chegará ao processo receptor e as realmente chegam podem ser fora de ordem. Não inclui um mecanismo de controlo de congestionamento, portanto um processo de origem pode mandar os mandos à taxa que quiser. O UDP também não oferece nenhuma garantia quanto a atrasos.

2.2 A Web e o HTTP

A Web funciona por demanda.

2.2.1 Descrição geral do HTTP

O HTTP (Protocolo de Transferência de Hipertexto) é um protocolo da camada de aplicação, ele é implementado em dois programas, um cliente e outro servidor. Os dois programas, executados em sistemas finais diferentes, conversam um com o outro por meio da troca de mensagens HTTP. O protocolo define a estrutura dessas mensagens e o modo como o cliente e o servidor as trocam.
Uma **página Web** é constituída de objetos que são simplesmente arquivos que se podem acessar com um único URL. A maioria das páginas Web é constituída de um arquivo-base HTML e diversos objetos referenciados. Cada URL tem dois componentes, o nome do hospedeiro do servidor que abriga o objeto e o nome do caminho do objeto.

Um **browser** é um agente de usuário para a Web, apresenta a página requisitada ao usuário e fornece numerosas características de navegação e de configuração. Um **servidor Web** abriga objetos Web, cada um endereçado por um URL.
Quando um usuário requisita uma pagina Web, o browser envia ao servidor mensagens de requisição HTTP para os objetos da página, o servidor recebe as requisições e responde com mensagens de resposta HTTP que contêm os objetos.
Até 1997, essencialmente todos os browser e servidores Web implementavam a versão HTTP/1.0, a partir de 1998 eles começaram a implementar a versão HTTP/1.1. O HTTP /1.1 é compatível com o HTTP /1.0, um servidor Web que execuuta a versão 1.1 pode se comunicar com um servidor que executa a 1.0.
O HTTP usa o TCP como seu protocolo de transporte subjacente.
O servidor HTTP não mantém nenhuma informação sobre clientes, por isso é denominado protocolo sem estado.

2.2.2 Conexões persistentes e não persistentes:

Em seu modo default, o HTTP usa conexões persistentes, mas pode ser configurado para utiliza as não persistentes.

Conexões não persistentes:
Possui cinco etapas de transferência:

1. O processo cliente HTTP inicia uma conexão TCP com o servidor na porta de número 80.
2. O cliente envia uma mensagem(inclui o caminho) de requisição ao servidor através de seu socket associado com a conexão TCP.
3. O processo servidor recebe a mensagem de requisição através de seu socket associado à conexão, extrai o objeto de seu armazenamento, encapsula o objeto em uma mensagem de resposta e a envia ao cliente.
4. O processo servidor ordena ao TCP que encerre a conexão TCP, que só encerrará quando tiver certeza de que o cliente recebeu a mensagem de resposta.
5. O cliente recebe a mensagem de resposta e a conexão TCP é encerrada.

Cada conexão TCP é encerrada após o servidor enviar o objeto, ela não persiste para os outros objetos. Usuários podem configurar browsers modernos para controlar o grau de paralelismo(conexões paralelas podem ser abertas).
O tempo de requisição que transcorre entre a requisição e o recebimento de um arquivo-base HTTP por um cliente é denominado tempo de viagem de ida e volta (RTT), já inclui atrasos.
Possui algumas desvantagens: uma nova conexão deve ser estabelecida e mantida para cada objeto solicitado. Para cada uma delas, devem ser alocados buffers TCP e conservadas variáveis TCP tanto no cliente quanto no servidor.

Conexões persistentes:
Em conexões persistentes, o servidor deixa a conexão TCP aberta após enviar a resposta, requisições e repostas subseqüentes entre os mesmos cliente e servidor podem ser enviadas por meio da mesma conexão. Normalmente, o servidor HTTP fecha uma conexão quando ela não usada durante um certo tempo.

Há duas versões de conexões persistentes:

Sem paralelismo: O cliente emite uma nova requisição somente quando a resposta anterior for recebida. Sofre um RTT para requisitar e receber cada um dos objetos.Após o servidor enviar um objeto, a conexão fica ociosa enquanto espera a chegada de outra requisição.

Com paralelismo: O cliente emite uma requisição logo que encontra uma referência, assim pode fazer requisições seqüenciais para os objetos relacionados, isto é, pode fazer uma nova requisição antes de receber uma resposta a uma requisição anterior. É possível gastar somente um RTT para todos os objetos. A conexão TCP com paralelismo fica ociosa durante uma fração menor de tempo.

2.2.3 Formato da mensagem HTTP

Há dois tipos de mensagens HTTP: de requisição e de resposta.

Mensagem de requisição HTTP

1. GET /somedir/page.html HTTP/1.1
2. Host: www.someschool.edu
3. User-agent: Mozilla/4.0
4. Connection: close
5. Accept-language:fr

Escrita em ASCII, é constituída de cinco linhas, cada uma seguida de um ‘carriage return’ e ‘line feed’. Embora esta mensagem tenha 5 linhas, uma mensagem pode ter muito mais ou menos que isso. A primeira linha de uma mensagem é denominada linha de requisição, as subseqüentes são denominadas linhas de cabeçalho.

1. O método GET é usado quando o browser requisita um objeto e este é identificado no campo do URL.
2. Especifica o hospedeiro no qual o objeto se encontra.
3. O browser está dizendo que não quer usar conexões persistentes.
4. Especifica o agente de usuário.
5. Mostra que o usuário prefere receber uma versão ,na língua especificada, do objeto se esse existir no servidor.

Método POST: Utilizado quando o usuário preenche um formulário.
Método HEAD: É semelhante ao GET, quando um servidor recebe uma requisição com ele, responde com uma mensagem HTTP, mas deixa de fora o objeto requisitado.

HTTP/1.0 permite somente três tipos de métodos GET, POST e HEAD. Além desses três métodos, a especificação HTTP/1.1 permite vários métodos adicionais, entre eles PUT(permite que um usuário carregue um objeto para um caminho especifico) e DELETE(permite que o usuário elimine um objeto).

Mensagem de resposta HTTP

1. HTTP/1.1 200 OK
2. Connection close
3. Date: Thu, 06 Aug 1998 12:00:15 GMT
4. Server: Apache/1.3.0 (Unix)
5. Last-Modified: Mon, 22 Jun 1998
6. Content-Length: 6821
7. Content-Type: text/html

1. Linha de estado.
2. Conexão não persistente
3. Indica a hora e a data em que a resposta foi criada e enviada pelo servidor.
4. Mostra que a mensagem foi gerada por um servidor Web Apache.
5. Indica a hora e data em que o objeto foi criado ou sofre a ultima modificação.
6. Indica o número de bytes do objeto que está sendo enviado.
7. Tipo do objeto.

Códigos de estado:
200 OK : requisição bem-sucedida e informação é entregue com a resposta.
301 Moved Permanently : Objeto requisitado foi removido permanentemente, novo URL.
400 Bad Request: Código genérico de erro que indica que a requisição não pôde ser entendida pelo servidor.
400 Not Found: o documento requisitado não existe no servidor.
505 HTTP Version Not Supported: a versão do protocolo requisitada não é suportada pelo servidor.

2.2.4 Interação usuário-servidor: cookies

Cookies, permitem que sites monitorem seus usuários. A tecnologia dos cookies tem quatro componentes, uma linha de cabeçalho de cookie na mensagem de resposta HTTP ; uma linha de cabeçalho de cookie na mensagem de requisição HTTP; um arquivo de cookie mantido no sistema final do usuário e gerenciado pelo browser do usuário; um banco de dados de apoio no site Web.

2.2.5 Conteúdo HTTP

Além da Web, é usado para aplicações de comércio eletrônico para transferência de arquivos XML de uma máquina a outra; para transferir VoiceXML, WML e frequentemente é usado como o protocolo de transferência de arquivos no compartilhamento de arquivos P2P.

2.2.6 Caches Web

Um cachê Web(servidor proxy) é uma entidade da rede que atende requisições HTTP em nome de um servidor Web de origem. Tem seu próprio disco de armazenamento, e mantém ,dentro dele, cópias de objetos recentemente requisitados.
O browser de um usuário pode ser configurado de modo que todas as suas requisições HTTP sejam dirigidas primeiramente ao cache Web. Quando um browser requisita um objeto, ele estabelece uma conexão TCP com o cachê Web e envia a ele uma requisição HTTP para um objeto, o cachê verifica se tem uma cópia do objeto armazenada, se tiver, a envia para o browser do cliente, se não tiver, o cache abre uma conexão TCP com o servidor de origem, envia uma requisição do objeto para a conexão TCP, após receber essa requisição, o servidor de origem envia o objeto ao cache, quando este recebe o objeto, ele guarda uma cópia em seu armazenamento local e envia outra, ao browser do cliente.
Um cache é, ao mesmo tempo, um servidor(quando recebe requisições de um browser) e um cliente(quando envia requisições para um servidor de origem).
.O cache na Web tem sido utilizado amplamente na Internet por duas razões: um cache Web pode reduzir substancialmente o tempo de resposta para a requisição de um cliente; chaces Web podem reduzir substancialmente o tráfego no enlace de acesso de uma instituição qualquer à Internet e também o da Internet como um todo, melhorando o desempenho para todas as aplicações.

2.2.7 GET condicional

O HTTP tem um mecanismo que permite que um cache verifique se seus objetos estão atualizados, o GET condicional.

2.3 Transferência de arquivo: FTP

O usuário quer transferir arquivos de ou para um hospedeiro remoto, para acessar a conta remota, ele deve fornecer uma identificação e uma senha, após fazer isso, ele pode transferir arquivos do sistema local de arquivos para o sistema remoto e vice-versa. O usuário interage com o FTP por meio de um agente de usuário FTP. Primeiro, ele fornece o nome do hospedeiro remoto(estabelece conexão TCP com o processo servidor), depois ele fornece sua identificação e senha sendo enviadas pela conexão TCP.
O HTTP e o FTP são protocolos de transferência de arquivos e têm muitas características em comum: por exemplo, ambos utilizam o TCP.Mas também possuem diferenças importantes, o FTP usa duas conexões TCP paralelas para transferir um arquivo, uma de controle e uma de dados. A primeira é usada para enviar informações de controle entre os dois hospedeiros, a segunda é usada para efetivamente enviar um arquivo. Como o FTP usa uma conexão de controle separada, dizemos que ele envia suas informações de controle fora da banda. O HTTP envia linhas de cabeçalho de requisição e de resposta pela mesma conexão TCP que carrega o próprio arquivo transferido, por essa razão diz-se que ele envia suas informações de controle na banda.
O FTP envia exatamente um arquivo pela conexão de dados e em seguida a fecha, se o usuário quiser enviar um outro arquivo, o FTP abrirá outra conexão de dados. Durante

uma sessão, o servidor FTP deve manter informações de estado sobre o usuário, ele deve associar a conexão de controle com uma conta de usuário específica e também deve monitorar o diretório corrente do usuário enquanto este passeia pela arvore do diretório remoto.

2.3.1 Comandos e respostas FTP:

Alguns comandos:

USER username: enviar identificação do usuário ao servidor.
PASS password: enviar a senha do usuário ao servidor
LIST: pedir ao servidor que envie uma lista com todos os arquivos existentes no atual diretório remoto.
RETR filename: obter um arquivo do diretório atual do hospedeiro remoto.
STOR filename: armazenar um arquivo no diretório atual.

Respostas típicas:

331 Nome de usuário OK, senha requisitada
125 Conexão de dados já aberta; iniciando transferência
425 Não é possível abrir a conexão de dados
452 Erro ao escrever o arquivo

2.4 Correio eletrônico na Internet

Tal como o correio normal, o e-mail é um meio de comunicação assíncrono, rápido, fácil de distribuir e barato.No sistema de correio da Internet há três componentes, os agentes de usuários, servidores de correio e o SMTP. Servidores de correio formam o núcleo da infra-estrutura do e-mail, cada destino.Cada destinatário tem uma caixa postal localizada em um dos servidores de correio. Uma mensagem típica inicia sua jornada no agente de usuário do remetente, vai até o servidor de correio dele e viaja até o servidor de correio do destinatário, onde é depositada na caixa postal. Quando o destinatário que acessar as mensagens de sua caixa postal, o servidor de correio que contem sua caixa postal o autentica. Se o servidor de correio do remetente não puder entregar a correspondência ao servidor dele, manterá a mensagem em uma fila de mensagens e tentará transferi-la depois.
O SMTP é o principal protocolo de camada de aplicação do correio eletrônico da Internet.Usa o serviço confiável de transferência de dados do TCP.

2.4.1 SMTP

O SMTP transfere mensagens de servidores de correio remetentes para servidores de correio destinatários. O SMTP é uma tecnologia antiga que possui certas características arcaicas, por exemplo, restringe o corpo de todas as mensagens de correio ao simples formato ASCII de 7 bits.

Enviando uma mensagem:

1. O remetente chama seu agente de usuário para email, fornece o endereço do destinatário, compõe uma mensagem e instrui o agente a enviar a mensagem.
2. O agente de usuário do remetente envia a mensagem para o seu servidor de correio, onde ela é colocada em uma fila de mensagens.
3. O lado cliente do SMTP, que funciona no servidor de correio do remetente, vê a mensagem na fila e abre uma conexão TCP para um servidor SMTP, que funciona no servidor de correio do destinatário.
4. Após alguns procedimentos iniciais de apresentação, o cliente SMTP envia a mensagem do remetente para dentro da conexão TCP.
5. No servidor de correio do destinatário, o lado servidor SMTP recebe a mensagem e a coloca na caixa postal dele.
6. O destinatário chama seu agente de usuário para ler a mensagem quando for mais conveniente para ele.

O SMTP não usa servidores de correio intermediários para enviar correspondência, mesmo quando os dois servidores estão localizados em lados opostos do mundo.

Como o SMTP transfere uma mensagem de um servidor de correio remetente para um servidor de correio destinatário:
O Cliente SMTP faz com que o TCP estabeleça uma conexão na porta 25 com o servidor SMTP, se o servidor não estiver em funcionamento, o cliente tenta novamente depois, senão, o servidor e o cliente trocam alguns procedimentos de apresentação, assim que acabam de se apresentar, o cliente envia a mensagem.

2.4.2 Comparação com o HTTP:

Ambos os protocolos são usados para transferir arquivos de um hospedeiro para outro. O HTTP transfere arquivos de um servidor Web para um cliente Web. O SMTP transfere arquivos de um servidor de correio para outro. Possuem características em comum (conexões persistentes) mas diferenças importantes também. O HTTP é um protocolo de recuperação de informações e o SMTP é um protocolo de envio de informações. Outra diferença é que o SMTP exige que cada mensagem, inclusive o corpo, esteja no formato ASCII de 7 bits e o HTTP não impõe essa restrição. O HTTP encapsula cada objeto em sua própria mensagem HTTP, o correio pela Internet coloca todos os objetos de mensagem em um única mensagem.

2.4.3 Formatos de mensagem de correio e MIME:

Formato: Possui um cabeçalho contendo informações(From, To, Subject) e o corpo da mensagem.

A extensão MIME para dados que não seguem o padrão ASCII: Para enviar conteúdo que não seja texto ASCII, o agente de usuário remetente deve incluir cabeçalhos adicionais na mensagem, esses cabeçalhos são Content-Type(permite que o agente de usuário destinatário realiza uma ação adequada sobre a mensagem) e Content-Transfer-Enconding(converter o corpo da mensagem à sua forma original).
O cabeçalho Received: Especifica o nome do servidor SMTP que enviou a mensagem, o do que recebeu e o horário em que o servidor destinatário recebeu a mensagem.

2.4.4 Protocolos de acesso ao correio:

Como o agente de usuário do destinatário não pode usar SMTP para obter as mensagens por que essa é uma operação de recuperação e o SMTP é um protocolo de envio, existem protocolos especiais que acessam o correio, entre eles o POP3, IMAP e HTTTP.

POP3: Protocolo de acesso de correio extremamente simples, possui funcionalidade limitada. O POP3 começa quando o agente de usuário(o cliente) abre uma conexão TCP com o servidor de correio(o servidor) na porta 110. Com a conexão ativada, o protocolo passa por três fases: autorização(o agente de usuário envia um nome de usuário e uma senha para autenticar o usuário), transação(recupera mensagens, o agente de usuário pode marcar mensagens que devem ser apagadas,), atualização(apaga as mensagens que foram marcadas). Em uma transação POP3, o agente de usuário emite comandos e o servidor, uma resposta para cada um deles, há duas respostas possíveis +OK (quando ocorreu tudo bem) e –ERR:(informa que houve algo de errado). A fase de autorização tem dois comandos principais user e pass. O servidor POP3 mantém como informação de estado apenas quais mensagens devem ser apagadas.O modo ler-e-apagar não permite que o usuário acesse uma mensagem já lida através de um outro computador. O modo ler-e-guardar permite que isso aconteça.

IMAP: Protocolo com mais recursos e mais complexo que o POP3. Um servidor IMAP associa cada mensagem a uma pasta. Este protocolo provê comandos que permitem que os usuários criem pastas e transfiram mensagens de uma para outra e comandos que os usuários podem usar para pesquisar pastas remotas em busca de mensagens que obedeçam a critérios específicos.Mantém informação de estado de usuário. Tem comandos que permitem que um agente de usuário obtenha componentes de mensagens.

E-mail pela Web: O agente de usuário é um browser Web comum e o usuário se comunica com sua caixa postal via HTTP. E quando se quer enviar uma mensagem de e-mail, esta é enviada do browser do remetente para seu servidor de correio e não por SMTP.

2.5 DNS: o serviço de diretório da Internet

Hospedeiros da Internet podem ser identificados de muitas maneiras. Um identificador é seu nome de hospedeiro(hostname), fáceis de lembrar, porém eles fornecem pouca, se é que alguma, informação sobre a localização de um hospedeiro na Internet.Além disso, como nomes de hospedeiros podem consistir em caracteres allfanuméricos de comprimento variável, seriam difíceis de serem processados por roteadores. Hospedeiros também são identificados pelos endereços de IP(constituído de 4 bytes), que possui uma estrutura hierárquica(ao examiná-lo da esquerda para a direita, obtem-se informações específicas sobre onde o hospedeiro se encontra).

2.5.1 Serviços fornecidos pelo DNS

As pessoas preferem o identificador nome de hospedeiro e os roteadores os endereços de IP, para conciliar essa preferências, é necessário um serviço de diretório que traduza nomes de hospedeiro para endereços de IP. Esta é a tarefa principal do DNS(sistema de nomes de domínio). O DNS é um banco de dados distribuído implementado em uma

hierarquia de servidores nome(servidor DNS) e um protocolo de camada de aplicação que permite que hospedeiros consultem o banco de dados distribuídos. É comumente empregado por outras entidades da camada de aplicação(como HTTP, SMTP, FTP).
Exemplo: Um browser(cliente HTTP) quer que roda na máquina de um usuário requisita um URL. Para que a máquina do usuário possa enviar uma mensagem de requisição HTTP ao servidor WEB, ela precisa primeiramente obter o endereço de IP e isso é feito da seguinte maneira:

1. A própria máquina do usuário executa o lado cliente da aplicação DNS
2. O browser extra o nome de hospedeiro do URL e passa o nome para o lado cliente da aplicação DNS.
3. O cliente DNS envia uma consulta contendo o nome do hospedeiro para um servidor DNS.
4. O cliente DNS finalmente recebe uma resposta, que inclui o endereço IP para o nome do hospedeiro.
5. Tão logo o browser receba o endereço do DNS, pode abrir uma conexão TCP com o processo servidor http localizado naquele endereço IP.

Outros serviços importantes do DNS:

Apelidos de hospedeiro: Um hospedeiro com nome complicado pode ter um ou mais apelidos. Um nome como ralayl.west-coast.enterprise.com (noime canônico) pode ter, por exemplo, enterprise.com como apelido. O DNS pode ser chamado por uma aplicação para obter o nome canônico correspondente a um apelido fornecido.

Apelidos de servidor de correio: Endereços de e-mail são fáceis de lembrar, por exemplo bob@hotmail.com, porém o nome de hospedeiro do hospedeiro do servidor do Hotmail é mais complicado do que hotmail.com, então o DNS pode ser chamado por uma aplicação de correio para obter o nome canônico a partir do apelido fornecido.

Distribuição de carga: O DNS também é usado para realizar distribuição de carga entre servidores replicados, tais como os servidores Web replicados. Sites movimentados são replicados em vários servidores, sendo que cada servidor roda em um sistema final diferente e tem um endereço IP diferente.Assim, um conjunto de endereços IP fica associado a um único nome canônico e contido no banco de dados do DNS.

2.5.2 Visão geral do modo de funcionamento do DNS:

Suponha que uma certa aplicação que executa na máquina de um usuário, precise traduzir um nome de hospedeiro para um endereço de IP. A aplicação chamará o lado cliente do DNS, especificando o nome de hospedeiro que precise ser traduzido. A partir daí, o DNS do hospedeiro do usuário assume o controle, enviando uma mensagem(enviada dentro de datagramas UDP à porta 53) de consulta para dentr da rede. O DNS no hospedeiro do usuário recebe uma mensagem de resposta DNS fornecendo o mapeamento desejado. O DNS é uma caixa-preta que provê um serviço de tradução simples e direto. Um arranjo simples para DNS seria ter um servidor de nomes contendo todos os mapeamentos, porém existem problemas em fazer isso.

Um único ponto de falha: Se o servidor de nomes quebrar, a Interne inteira quebrará.
Volume de tráfego: Um único servidor teria de manipular todas as consultas DNS.
Banco de dados centralizado distante: Um único servidor de nomes nunca poderia estar 'próximo' de todos os clientes que fazem consultas.
Manutenção: O único servidor de nomes teria de manter registros de todos os hospedeiros da Internet.

Um banco de dados distribuído, hierárquico: O DNS usa um grande numero de servidores, organizados de maneira hierárquica e distribuídos por todo mundo. Há três classes de servidores de nomes: de nomes raiz, de domínio de alto nível(TLD) e servidores DNS com autoridade.

Servidores de nomes raiz: Existem 13 servidores de nomes raiz.

Servidores de nomes de domínio de alto nível: São responsáveis por domínios de alto nível como .com, .org, .net, .edu e também por todos os domínios de alto nível de paises como .fr, .uk, .br.

Servidores de nomes com autoridade: Toda organização que tiver hospedeiros que possam ser acessados publicamente na Internet deve fornecer registros DNS também acessíveis publicamente que mapeiam os nomes desses hospedeiros para endereços de IP.

Exemplo: O hospedeiro cis.poly.edu deseja o endereço IP de gaia.cs.umass.edu

Consulta Iterativa: O hospedeiro envia uma mensagem de consulta DNS a seu servidor de nomes local(dns.poly.edu). Essa mensagem contém o nome de hospedeiro a ser traduzido. O servidor de nomes local transmite a mensagem de consulta a um servidor de nomes raiz, que percebe o sufixo edu e retorna ao servidor de nomes local uma lista de endereços IP contendo servidores TLD responsáveis por edu. Então, o servidor de nomes local retransmite a mensagem de consulta a um desses servidores TLD, que recebe o sufixo umass.edu e responde com o endereço IP do servidor de nomes com autoridade para a University of Massachusetts (dns.umass.edu). Finalmente, o servidor de nomes local reenvia a mensagem de consulta diretamente a dns.umass.edu que responde com o endereço IP de gaia.cs.umass.edu

Consulta Recursiva: O hospedeiro requisita o endereço IP, este envia uma mensagem de consulta para o servidor de nomes loca, que envia ao servidor de nomes raiz, que envia ao de nomes TLD, que envia ao de nomes com autoridade, que retorna o endereço IP para o TLD, que retorna ao de nomes raiz, que retorna ao de nomes local.

Cache DNS: O DNS explora extensivamente o cache para melhorar o desempenho quanto ao atraso e reduzir o numero de mensagem DNS que ricocheteia pela Internet. A idéia por trás do cache é muito simples. Em uma cadeia de consultas, quando um servidor de nomes recebe uma resposta DNS, ele pode fazer cache das informações da resposta em sua memória local.

2.5.3 Registros e mensagens DNS:

Os servidores de nomes que juntos implementam o banco de dados distribuído do DNS armazenam registros de recursos, que fornecem mapeamentos de nomes de

hospedeiros para endereços IP. Cada mensagem DNS carrega um ou mais registros de recursos.
Um registro de recurso é uma tupla de quatro elementos:
(Name, Value, Type, TLL)
TLL é o tempo de vida útil do registro. Os significados de Name e Value dependem de Type.

Type = A -> Name é o noem de hospedeiro e Value é o endereço IP para o nome de hospedeiro.
Type = NS -> Name é um domínio e Value é o nome de um servidor de nomes com autoridade que sabe como obter endereços IP para hospedeiros do domínio.
Type = CNAME -> Value é um nome canônico de hospedeiro para o apelido de hospedeiro contido em Name.
Type = MX -> Value é o nome canônico de um servidor de correio cujo apelido de hospedeiro está contido em Name.

Mensagens DNS: As duas únicas espécies de mensagens DNS são mensagens de consulta e de resposta DNS.
A semântica de vários campos de uma mensagem é a seguinte:
1. Os primeiros 12 bytes formam a seção de cabeçalho que contém vários campos: O primeiro campo é o identificador da consulta. Campos de flag (flag para dizer se a mensagem é de consulta ou resposta, *flag de autoridade* que é marcado em uma mensagem de resposta quando o servidor de nomes é um servidor com autoridade para um nome consultado, *flag de recursão*, quando um cliente quer que um servidor de nomes proceda recursivamente sempre que não tem registro). Há também quatro campos de 'numero de'.
2. A seção pergunta contém informações sobre a consulta que está sendo feita.Inclui um campo de nome que contém o nome que está sendo consultado e um campo de tipo que indica o tipo de pergunta que está sendo feito sobre o nome.
3. Em uma resposta de um servidor de nomes, a seção resposta contém os registros de recursos par ao nome que foi consultado originalmente.
4. A seção de autoridade contém registros de outros servidores com autoridade.
5. A seção adicional contém outros registros úteis.

Para inserir registros no banco de dados do DNS: Quando deseja-se registrar o nome de domínio de uma empresa, isto deve ser feito em uma entidade registradora, que é uma entidade comercial que verifica se o nome de domínio é exclusivo, registra-o no banco de dados DNS e cobra uma pequena taxa por seus serviços. Ao registrar o nome de usuário, também deve-se informar os nomes e endereços IP dos seus servidores DNS com autoridade, primários e secundários. Também deve-se inserir em seus servidores de nomes com autoridade do registro de recurso Type A e o registro de recurso Type MX para ser servidor de correio.

2.6 Compartilhamento de arquivos P2P

Exclusivamente em termos de tráfego, o compartilhamento de arquivos P2P pode ser considerado a aplicação mais importante da Internet. Sistemas modernos de compartilhamento de arquivos P2P não somente compartilham MP3, mas também vídeos,software,documentos e imagens.

Um usuário está ligado à Internet e lança sua aplicação de compartilhamento de arquivos P2P. Ele consulta um MP3 que quer, logo após dar o comando de busca, a aplicação exibirá uma lista de pares que têm uma cópia da canção para compartilhar e que estão conectados na Internet no momento. Uma conexão direta é estabelecida entre os dois computadores e o arquivo MP3 é enviado do par que o tem para o par do usuário que o pediu. Se o par que o tem inadvertidamente desconectar seu PC da Internet durante a transferência, então o software de compartilhamento de arquivos P2P do usuário pode tentar obter o restante do arquivo de um outro par que o tenha.

O compartilhamento de arquivos P2P é um paradigma de distribuição atraente porque todo o conteúdo é transferido diretamente entre pares comuns, sem passar por servidores de terceiros. Ele é altamente escalável. Embora não exista envolvimento de nenhum servidor centralizado, o compartilhamento de arquivos P2P ainda se baseia no paradigma cliente-servidor. O par requisitante é o cliente e o par escolhido é o servidor. O arquivo é enviado do par servidor ao par cliente com um protocolo de transferência de arquivos.Todos os pares podem executar tanto o lado cliente quanto o servidor.
Suponha que o protocolo de transferência de arquivos seja HTTP . Quando o ‘requisitante’ seleciona o ‘escolhido’ para baixar a canção, seu computador envia a ele uma requisição HTTP para essa canção e este envia uma resposta HTTP contendo a canção. Enquanto o ‘requisitante’ estiver executando a aplicação de compartilhamento de arquivos P2P, seu computador é um cliente Web e também um servidor Web transitório. Seu par é um servidor Web porque está servindo conteúdo dentro de respostas HTTP; é transitório porque está conectado apenas intermitentemente com a Internet e pode obter um novo endereço IP toda vez que se conectar novamente com a rede.

Diretório centralizado: Uma das abordagens mais diretas da localização de conteúdo é prover um diretório centralizado (como fazia o Napster). O serviço de compartilhamento de arquivos P2P usa um servidor de grande porte para prover o serviço de diretório.Quando um usuário lança a aplicação de compartilhamento de arquivos P2P, ela contata o servidor de diretório. Ela informa ao servidor de diretório seu endereço IP e os nomes dos objetos que estão disponíveis para compartilhamento em seu disco local. O servidor de diretório coleta essa informação de cada par que fica ativo, criando, um banco de dados centralizado. Para manter seu banco de dados atualizado, o servidor de diretório deve poder determinar quando um par se desconecta. Um modo de monitorar quais pares continuam conectados é enviar mensagens periodicamente para verificar se eles respondem. De o servidor de diretório determinar que um par não está mais conectado, ele remove do banco de dados o endereço de IP dele. Utilizar um diretório centralizado tem várias desvantagens:

1. Um único ponto de falha: Se o servidor de diretório cair, toda a aplicação P2P cairá.

2. Gargalo de desempenho: Um servidor centralizado tem de manter um banco de dados imenso e deve responder a milhares de consultas por segundo.
3. Violação de direitos autorais: A industria fonográfica está preocupada que sistemas de compartilhamento de arquivos P2P permitam que usuários obtenham facilmente acesso gratuito a conteúdo protegido por direitos autorais.

Inundação de consultas: Gnutella, uma aplicação de compartilhamento de arquivos de domínio público. Diferentemente do Npster, a Gnutella não utiliza um servidor entralizado para monitorar conteúdo nos pares. O cliente Gnutella implementa o protocolo Gnutella e executa em um par comum. Em Gnutella, os pares formam uma rede abstrata, lógica, denominada rede de sobreposição. Se o par X mantiver uma conexão TCP com um outro par Y, então dizemos que há uma aresta(abstrata) entre X e Y. O grafo que contém todos os pares ativos e arestas de conexão define a rede de sobreposição.
Em Gnutella, pares enviam mensagens a pares próximos na rede de sobreposição montada sobre conexões TCP existentes. Quando o 'requisitante' quer localizar uma canção, seu cliente Gnutella envia a todos os seus vizinhos uma mensagem Gnutella Query que inclui as palavras chave, esses vizinhos por sua vez, retransmitem a mensagem a todos os seus vizinhos e assim por diante. Esse processo é denominado inundação de consultas.
Embora o projeto descentralizado da Gnutella seja simples e bem estruturado, muitas vezes é criticado por não ser escalável.Em particular, com inundação de mensagens,sempre que um par inicia uma consulta, ela se propaga para todos os outros pares presentes na rede de sobreposição. Os projetistas da Gnutella enfrentaram esse problema utilizando inundação de consultas de escopo limitado.
Descrevendo o que acontece quando um par X quer juntar-se à rede Gnutella.

1. Em primeiro lugar, o par X deve achar algum outro par já que esteja na rede de sobreposição. Uma abordagem para resolver esse autocarregamento é o cliente Gnutella de X manter uma lista de pares que estão frequentemente ativos na rede Gnutella.

2. Tão logo obtenha acesso a essa lista, X faz tentativas seqüenciais para estabelecer uma conexão com pares presentes na lista até ser criada uma conexão com algum par Y.

3. Após estabelecida a conexão TCP entre X e Y, o par X envia a Y uma mensagem Gnutell Ping. Ao receber a mensagem Ping, Y transmite a todos os seus vizinhos na rede de sobreposição.

4. Sempre que um par Z recebe uma mensagem Ping, responde retornando uma mensagem Gnutella Pong para X através da rede de sobreposição.

5. Quando recebe as mensagens Pong, X sabe os endereços IP de muitos pares presentes na rede Gnutella,além de Y. Então pode estabelecer conexões TCP com alguns desses outros pares.

Explorando a heterogeneidade: O Napster utiliza um servidor de diretório centralizado e sempre localiza conteúdo quando este estiver presente em algum par participante.Gnutella utiliza uma arquitetura totalemnte distribuída, mas localiza conteúdo somente em pares próximos na rede de sobreposição.O Kazaa toma emprestadas idéias do Napster e Gnutella, resultando em um poderoso sistema de compartilhamento de arquivos P2P. A tecnologia Kazaa é proprietária e ,além disso, criptografa todo o controle de trafego.O Kazaa explora a heterogeneidade de uma maneira intuitiva e natural. Ele se parece com o Gnutella, no sentido de que não usa um servidor dedicado para monitorar e localizar conteúdo.Entretanto, diferentemente da Gnutella, nem todos os pares são iguais no Kazaa. Os pares mais poderosos (grande largura de banda e alta conectividade com a Internet) são lideres de grupo e têm maiores responsabilidades.

Quando um par lança a aplicação Kazaa, estabelece uma conexão TCP com um dos lideres de grupo. Então, o par informa a seu líder de grupo todos os arquivos que está disponibilizando para compartilhamento.

No Kazaa cada arquivo é identificado por um hash do arquivo. Cada objeto tem um descritor, que inclui o nome do arquivo e um texto descritivo não estruturado do objeto.

A arquitetura do Kazaa explora heterogeneidade dos pares, designando como lideres de grupo uma pequena fração dos pares mais poderosos que formam a camada superior de uma rede de sobreposição plana e inundação de escopo limitado, o projeto hierárquico permite a verificação de compatibilidade em um numero significativamente maior de pares sem criar um trafego de consultas excessivo.

Também emprega varias técnicas que melhoram seu desempenho: enfileiramento de requisições, o usuário pode configurar seu par de modo a limitar o numero d transferências simultâneas em qualquer valor. Prioridades de incentivo o ‘escolhido’ dará prioridade de enfileiramento a usuários que, no passado, tenham carregado mais arquivos do que baixado. Transferência paralela , um ‘requisitante’ pode baixar o arquivo paralelamente, uma metade de usuário e outra de outro.