CEFET/SC – UNIDADE DE SÃO JOSÉ
Curso de Telecomunicações

Redes de Computadores e Internet

CURSO DE TELECOMUNICAÇÕES

Redes de Computadores e Internet

Prof. Evandro Cantú
CEFET/SC São José
88.103-310 • Praia Comprida • São José SC
Tel.: (048) 247-3646 • Fax: (048) 247-2542
Email: cantu@sj.cefetsc.edu.br
Primavera 2003

Apresentação

O objetivo desde texto é servir de material de apoio para um curso introdutório de redes de computadores, voltado para o técnico de nível médio. As redes de computadores são estudadas tomando como referência às tecnologias de rede mais difundidas atualmente, como a Internet e as redes locais Ethernet. A partir destas tecnologias, inicialmente dirige-se o foco para as aplicações de rede, tratando sua utilidade e importância na sociedade contemporânea. Depois, procura-se trabalhar os conceitos de base envolvidos nesta temática, explorando o software e o hardware de suporte para as aplicações, onde estão envolvidas questões como: conectividade entre as máquinas, comutação de circuitos e comutação de pacotes, protocolos de comunicação e arquitetura em camadas das redes de computadores. A arquitetura das redes de computadores é estudada em torno dos protocolos TCP/IP, principais protocolos da Internet, no que se refere às redes geograficamente distribuídas, e das tecnologias Ethernet, no que se refere às redes locais de computadores.

A principal referência bibliográfica utilizada na construção deste material foi o livro de KUROSE e ROSS, *Computer Networking: A top-down approach featuring the Internet*, o qual desenvolve uma abordagem, chamada pelos autores de *top-down* (ou "de cima para baixo"), onde o estudo inicia pelas aplicações e depois vai descendo pelas demais camadas que formam a arquitetura das redes de computadores, tomando como foco principal a Internet.

O texto está organizado em quatro partes. A primeira faz uma introdução às redes de computadores e a Internet, dando uma visão ampla das redes e dos principais conceitos envolvidos. A segunda parte abordada as aplicações de rede, apresentando em particular a aplicação WWW, o correio eletrônico e a transferência de arquivos. Na terceira parte discute os protocolos Internet TCP/IP. Finalmente, a quarta parte discute as redes locais e os protocolos de enlace, com destaque para a tecnologia Ethernet. No final do texto foi incluído um glossário de termos técnicos utilizados na área de redes de computadores, elaborado com a colaboração do professor Alexandre Moreira.

Evandro Cantú

São José, setembro de 2003.

Índice analítico

Parte
1

Introdução as Redes de computadores e a Internet

A Internet é hoje a rede de computadores mais utilizada no mundo, estando em franca expansão; em termos de redes de telecomunicações somente perde em abrangência para o sistema telefônico. No que se refere às tecnologias de rede, a Internet é uma entre muitas alternativas, todavia, devido a sua importância na sociedade contemporânea, pode ser tomada como principal veículo para a discussão das redes de computadores.

O que é uma rede de computadores?

Uma rede de computadores é conexão de dois ou mais computadores para permitir o compartilhamento de recursos e a troca de informações entre as máquinas.

Em alguns casos, seria suficiente construir redes de computadores limitadas, que conectam somente algumas máquinas. Por exemplo, num pequeno escritório de advocacia, com alguns computadores e uma impressora, poderia se construir uma pequena rede para permitir o compartilhamento da impressora entre os usuários.

Atualmente, com a importância cada vez maior de se dispor de acesso a informações e facilidades de comunicação, as redes de computadores estão projetadas para crescer indefinidamente, sendo a Internet um bom exemplo. No caso do escritório de advocacia, a pouco citado, além da possibilidade de compartilhamento de recursos, uma conexão com outras redes e à Internet pode oferecer acesso a informações importantes, como códigos de leis e acompanhar o andamento de processos, além de propiciar um meio de comunicação bastante ágil, facilitando o trabalho tanto dos prestadores do serviço de advocacia como dos clientes.

A conectividade dos computadores em rede pode ocorrer em diferentes escalas. A rede mais simples consiste em dois ou mais computadores conectados por um **meio físico**, tal como um par metálico ou um cabo coaxial. O meio físico que conecta dois computadores costuma ser chamado de **enlace de comunicação** e os computadores são chamados de **nós**. Um enlace de comunicação limitado a um par de nós é chamado de **enlace ponto-a-ponto**. Um enlace pode também envolver mais de dois nós, neste caso, podemos chamá-lo de **enlace multiponto** (Figura 1.1). Um enlace multiponto, formando um barramento de múltiplo acesso, é um exemplo de enlace utilizado na tecnologia de **rede local** (**LAN** – *local area network*) do tipo Ethernet.

Se as redes de computadores fossem limitadas a situações onde todos os nós fossem diretamente conectados a um meio físico comum, o número de computadores que poderiam ser interligados seria também muito limitado. Na verdade, numa rede de maior abrangência geográfica, como as **redes metropolitanas** (**MAN** – *metropolitan area network)* ou **redes de alcance global** (**WAN** *wide área network*), nem todos os computadores precisam estar diretamente conectados. Uma conectividade indireta pode ser obtida usando uma **rede comutada**. Nesta rede comutada podemos diferenciar os **nós** da rede que estão na sua **periferia**, como computadores terminais conectados ao núcleo da rede via enlaces ponto-a-ponto ou multiponto, daqueles que estão no **núcleo** da rede, formado por **comutadores** ou **roteadores** (Figura 1.2)

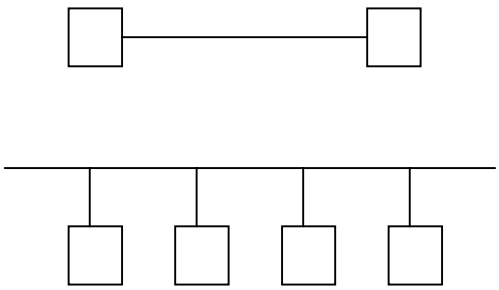


Figura 1.1. Enlace ponto-a-ponto e multiponto

Existem inúmeros tipos de redes comutadas, as quais podemos dividir em redes de **comutação de circuitos** e redes de **comutação de pacotes.** Como exemplo, podemos citar o sistema telefônico e a Internet, respectivamente.

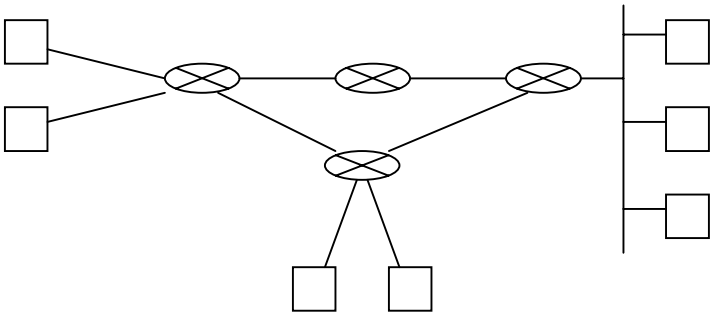


Figura 1.2. Rede comutada interconectando sistemas terminais

O que é a Internet?

A Internet é a **rede mundial de computadores**, que interliga milhões de dispositivos computacionais espalhados ao redor do mundo (Figura 1.3).

A maioria destes dispositivos é formada por **computadores pessoais**, **estações de trabalho**, ou **servidores**, que armazenam e transmitem informações, como por exemplo, páginas *Web*, arquivos de texto ou mensagens eletrônicas. Todos estes dispositivos são chamados **hospedeiros** (*hosts***)** ou **sistemas terminais**.

As **aplicações de rede**, como por exemplo, paginação na *Web*, transferência de arquivos ou correio eletrônico, rodam nos sistemas terminais.

Os sistemas terminais, assim como os principais componentes da Internet, precisam de **protocolos de comunicação,** que servem para controlar o envio e a recepção das informações na Internet. O **TCP** (*Transmission Control Protocol*) e o **IP** (*Internet Protocol*) são os principais protocolos da Internet, daí o fato de a Internet ser também conhecida como rede **TCP/IP**.

Os sistemas terminais são conectados entre si por meio de **enlaces de comunicação**, que por sua vez podem ser de diferentes tipos, como por exemplo, um enlace ponto-a-ponto (tipo o PPP) ou multiponto (como uma rede local Ethernet). Os enlaces de comunicação, por sua vez, são suportados por um **meio físico,** os quais podem ser **cabos coaxiais**, **fios de cobre**, **fibras ópticas** ou o ar a partir do uso do **espectro de freqüência de rádio**.

Na Internet, nem todos os computadores são diretamente conectados, neste caso, utilizam-se dispositivos de chaveamento intermediário, chamados **roteadores** (*routers* ou ainda *gateways*).

Em cada roteador da Internet as mensagens que chegam nos enlaces de entrada são **armazenadas e encaminhadas** (*store-and-forward*) aos enlaces de saída, seguindo de roteador em roteador até seu destino. Neste processo, a técnica de comutação utilizada é conhecida como **comutação de pacotes**, em contraste com a **comutação de circuitos** que é comumente utilizada nos sistemas telefônicos.

Na comutação de pacotes, as **mensagens** que serão transmitidas são fragmentadas em **pacotes** menores, os quais viajaram na Internet de forma independente uns dos outros.

O **protocolo IP** é o responsável por estabelecer a **rota** pela qual seguirá cada pacote na malha de roteadores da Internet. Esta rota é construída tendo como base o endereço de destino de cada pacote, conhecido como **endereço IP**.

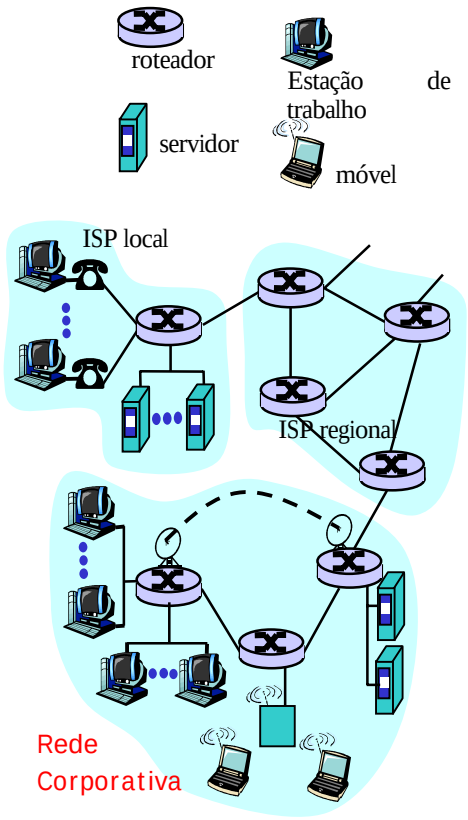


Figura 1.3. Visão dos componentes da Internet

Além de um **endereço IP**, um **nome** também pode ser associado a um sistema terminal a fim de facilitar sua identificação por nós humanos. Por exemplo, 200.135.233.1 é o endereço IP e `www.sj.cefetsc.edu.br` é o nome do servidor do CEFET-SC em São José. A aplicação **DNS** (*domain name system*) associa dinamicamente nomes a endereços IP.

Em outras palavras, pode-se dizer que a Internet é uma **rede de redes**, interconectando redes de computadores públicas e privadas, as quais devem rodar o protocolo IP em conformidade com a **convenção de endereços IP** e **nomes** da Internet.

A topologia da Internet é hierárquica, onde os **sistemas terminais** são conectados a **provedores locais** (ou **ISP** – *Internet Service Provider*), que por sua vez são conectados a **provedores regionais**, e estes últimos a **provedores nacionais** ou **internacionais.** Por exemplo, o provedor local do CEFET-SC em São José está conectado ao provedor regional da RCT-SC (Rede Catarinense de Tecnologia – `www.funcitec.rct-sc.br`), que está conectado ao provedor nacional da RNP (Rede Nacional de Pesquisa – `www.rnp.br`) (veja mapa RNP no endereço `www.rnp.br/backbone`).

A conexão de um computador a um provedor local é feita por meio de uma **rede de acesso**, a qual pode ser um **acesso residencial** (por exemplo, via modem e linha discada) ou **acesso corporativo** via **rede local**.

No nível tecnológico a Internet está construída a partir da criação, teste e implementação de **padrões Internet**. Estes padrões são desenvolvidos e formalizados pelo organismo internacional **IETF** (*Internet Engineering Task Force* – `www.ietf.org`), através de documentos conhecidos como **RFCs** (*Request For Comments* – `www.ietf.org/rfc.html`), que contém a descrição de cada protocolo padrão da Internet.

O que é um protocolo?

No nosso dia-a-dia o relacionamento humano exige alguns **protocolos**, ou *boas maneiras*, como por exemplo, quando os dirigimos a uma pessoa para perguntar as horas. Note que no exemplo de protocolo humano para perguntar as horas, há mensagens específicas que são emitidas e ações específicas que são realizadas em função das respostas recebidas (Figura 1.4).

No caso de um protocolo de rede temos a interação entre componentes de software e hardware dos computadores, ao invés de pessoas. Na Internet todas as atividades de comunicação são governadas por protocolos de comunicação. Por exemplo, protocolos fim-a-fim garantem a integridade dos dados transmitidos através de mecanismos de reconhecimento e retransmissão; protocolos de roteamento determinam o caminho de um pacote de dados da fonte até o destino; protocolos de hardware em um adaptador de rede controlam o fluxo de bits sobre os fios que interligam dois computadores; etc.

Como exemplo de um protocolo de rede, considere o que acontece quando você requisita uma página de um servidor *Web*. O cenário é mostrado na figura 1.4: primeiro seu computador envia uma mensagem requisitando uma conexão com o servidor remoto (*TCP conection request*); o servidor *Web* eventualmente vai receber sua requisição e responder afirmativamente (*TCP conection reply*); sabendo que a conexão esta estabelecida, seu computador requisita então a página procurada (`GET http://www.sj.cefetsc.edu.br/index.htm`) e o servidor remoto envia o arquivo com o código HTML correspondente.

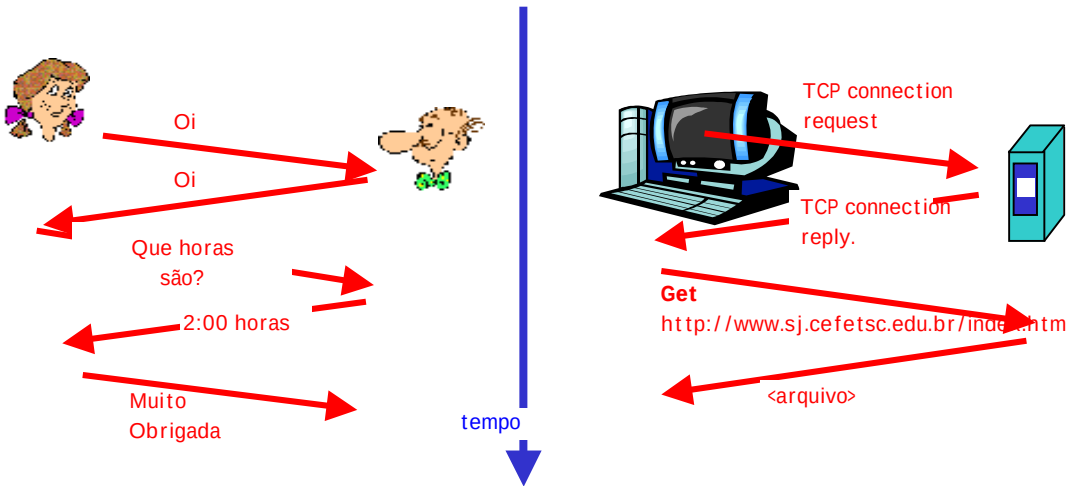


Figura 1.4. Protocolos

> Os protocolos definem o formato e a ordem das mensagens enviadas e recebidas pelas entidades da rede bem como as ações que são tomadas quando da transmissão ou recepção de mensagens.

A periferia da Internet

Olhando a Internet com um pouco mais de detalhe podemos identificar a **periferia da rede**, onde estão os computadores que rodam as aplicações, e o **núcleo da rede**, formado pela malha de roteadores que interligam as redes entre si.

Na periferia da rede estão os **sistemas terminais** ou **hospedeiros** (*hosts*). São referidos como hospedeiros porque hospedam **programas de aplicação.** São programas de aplicação típicos da Internet: o *login* remoto a sistemas (Telnet ou SSH), a transferência de arquivos (FTP), o correio eletrônico (*email*), a paginação na *Web* (WWW), a execução de áudio e vídeo, etc.

Os sistemas terminais são divididos em duas categorias: os **clientes** e os **servidores**. Os clientes são em geral computadores pessoais ou estações de trabalho, e os servidores computadores mais poderosos. Servidores e clientes interagem segundo o **modelo cliente/servidor**, no qual uma aplicação cliente solicita e recebe informações de uma aplicação servidora (Figura 1.5).

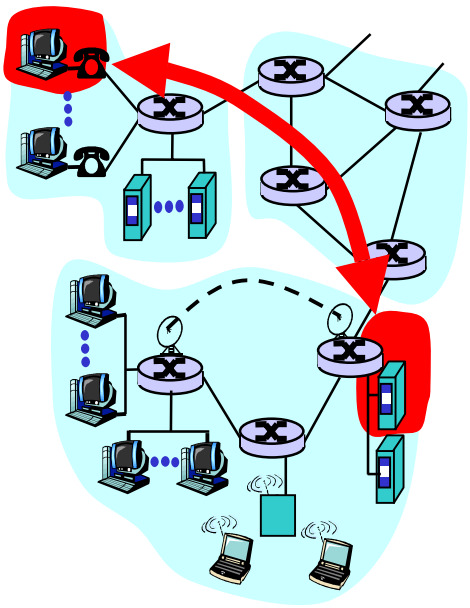


Figura 1.5. Interação cliente/servidor na Internet

Tipicamente a aplicação cliente roda em um computador e a aplicação servidora em outro, sendo por definição as **aplicações cliente/servidor** ditas **aplicações distribuídas**.

Serviços oferecidos pela Internet às aplicações

A Internet, ou mais genericamente as redes TCP/IP, provêem um **canal de comunicação lógico** entre um **processo cliente,** rodando em uma máquina cliente, e um **processo servidor,** rodando em uma máquina servidora, permitindo que as **aplicações distribuídas** troquem informações entre si. Para usar este canal de comunicação, os programas de aplicação têm uma **porta cliente**, através da qual o serviço é solicitado, e uma **porta servidora**, que retorna o serviço requisitado.

Quanto ao tipo de serviço solicitado pelas aplicações à rede podemos ter:

- Serviço tipo **pedido/resposta** (*request/reply*);
- Serviço tipo **fluxo de dados tempo real** (*audio/video streaming*).

A paginação na Web é um exemplo de **serviço tipo pedido/resposta**, onde um processo cliente solicita uma informação e um processo servidor fornece a informação solicitada. Não há restrições de tempo entre o pedido e a resposta, entretanto, é necessário que a informação transmitida seja livre de erros.

Uma conversa telefônica via Internet é um exemplo de **fluxo de dados em tempo real**, neste caso há restrições temporais na transmissão, por outro lado, um pequeno silêncio ocasionado por um erro ou ruído pode não ser um problema grave para o entendimento geral da conversa.

Para estes dois tipos de requisições de serviços, a Internet dispõe de dois tipos de **serviços de transporte**:

- Serviço garantido e orientado a conexão;
- Serviço não garantido e não orientado a conexão.

O **serviço garantido e orientado a conexão** tem o nome de **TCP (***Transmission Control Protocol*). Quando uma aplicação usa o **serviço orientado a conexão** o cliente e o servidor trocam pacotes de controle entre si antes de enviarem os pacotes de dados. Isto é chamado de procedimento de **estabelecimento de conexão** (*handshaking)*, onde se estabelecem os parâmetros para a comunicação. Por exemplo, mensagens TCP são trocadas entre as partes de uma interação WWW para estabelecer a conexão entre o cliente e o servidor. Uma vez concluído o *handshaking* a conexão é dita estabelecida e os dois sistemas terminais podem trocar dados. O serviço de **transferência garantida**, que assegura que os dados trocados são livres de erro, o que é conseguido a partir de mensagens de reconhecimento e retransmissão de pacotes. Por exemplo, quando um sistema terminal B recebe um pacote de A, ele envia um reconhecimento; quando o sistema terminal A recebe o reconhecimento ele sabe que o pacote que ele enviou foi corretamente recebido; caso A não recebe confirmação, ele assume que o pacote não foi recebido por B e retransmite o pacote.

Além das características citadas, o TCP integra ainda um serviço de **controle de fluxo,** que assegura que nenhum dos lados da comunicação envie pacotes rápido demais, pois uma aplicação em um lado pode não conseguir processar a informação na velocidade que está recebendo, e um serviço de **controle de congestão** ajuda a prevenir congestionamentos na rede.

No **serviço não orientado a conexão** não há *handshaking*; quando um lado de uma aplicação quer enviar pacotes ao outro lado ele simplesmente envia os pacotes. Como o serviço é não garantido, também não há reconhecimento, de forma que a fonte nunca tem certeza que o pacote foi recebido pelo destinatário. Também não há nenhum controle de fluxo ou congestão. Como o serviço é mais simples, os dados podem ser enviados mais rapidamente. Na Internet, o **serviço não garantido e não orientado a conexão** tem o nome de **UDP (***User Datagram Protocol***)**.

As aplicações mais familiares da Internet usam o TCP, como por exemplo: Telnet, correio eletrônico, transferência de arquivos e WWW. Todavia existem várias aplicações usam o UDP, incluindo aplicações emergentes como aplicações multimídia, voz sobre Internet, áudio e vídeo conferência.

Núcleo da Internet

O núcleo da rede é formado pela **malha de roteadores**, responsável por interligar as redes entre si, formando as ligações inter-redes, ou **Internet**.

No núcleo da rede as informações trafegam na forma de **pacotes de dados**, chamados de **datagramas**. Em cada roteador os datagramas que chegam nos enlaces de entrada são **armazenados e encaminhados** (*store-and-forward*) aos enlaces de saída, seguindo de roteador em roteador até seu destino.

O **protocolo IP** é o responsável por estabelecer a **rota** pela qual seguirá cada datagrama na malha de roteadores da Internet. Esta rota é construída tendo como base o endereço de destino de cada pacote, conhecido como **endereço IP**.

Como visto anteriormente, os **serviços de transporte** da Internet, através dos **protocolos TCP e UDP**, provêem o serviço de comunicação fim-a-fim entre as **portas dos processos de aplicação**

rodando em dois diferentes sistemas terminais (*hosts*). Para isto, o TCP e UDP usam os serviços do **protocolo IP**, a qual provê um serviço de comunicação para os datagramas entre os dois computadores remotos**,** envolvendo cada roteador da rede no caminho entre o computador origem e o destino da comunicação.

Comutação de pacotes x comutação de circuitos

A Internet usa a comutação de pacotes como tecnologia de comunicação no núcleo da rede, em contraste com as redes telefônicas que usam a comutação de circuitos.

Na **comutação de circuitos**, quando dois sistemas terminais desejam se comunicar a rede estabelece um **circuito dedicado fim-a-fim** entre os dois sistemas. É por exemplo o que acontece numa ligação telefônica; a partir do número discado, a rede estabelece um caminho entre os dois interlocutores e reserva um circuito para possibilitar a conversação; o circuito ficará reservado durante todo o tempo em que durar a comunicação.

Na **comutação de pacotes**, os recursos da rede *não* são reservados; as mensagens usam os recursos a medida da necessidade, podendo como conseqüência, durante uma transmissão de dados ter que esperar (em uma fila) para acessar um enlace, caso o mesmo esteja ocupado.

Como uma analogia simples, considere dois cabeleireiros: um que atende com hora marcada e o outro que não. Para o que atende com hora marcada deve-se antes fazer uma reserva de horário, mas, quando se chega ao cabeleireiro, a princípio, não haverá espera (isto não se aplica às consultas médicas, pois, apesar de hora marcada sempre há espera!). Para o que não atende com hora marcada pode-se chegar a qualquer momento, mas, corre-se o risco de ter que esperar, caso haja outras pessoas sendo atendidas.

A **Internet** é essencialmente uma rede baseada na comutação de pacotes. Considere, por exemplo, o que acontece quando um computador deseja enviar um pacote de dados a outro computador na Internet. Como na comutação de circuitos, o pacote será transmitido sobre uma série de diferentes enlaces de comunicação, todavia, não haverá uma reserva de um circuito fim-a-fim. O pacote será encaminhado de roteador em roteador, e caso o enlace de saída de um roteador de sua rota esteja ocupado, o pacote deverá ser armazenado e aguardar a liberação do enlace em uma fila, sofrendo um atraso.

> Diz-se que a Internet faz o **melhor esforço** (*best effort*) para entregar os dados num tempo apropriado, todavia não dá nenhuma garantia.

Os defensores da comutação de pacotes sempre argumentam que a comutação de circuitos é ineficiente, pois reserva o circuito mesmo durante os **períodos de silêncio** na comunicação. Por exemplo, durante uma conversa telefônica, os silêncios da conversação, ou as esperas para chamar uma outra pessoa, não podem ser utilizados para outras conexões. Em outro exemplo, imagine um médico que usa uma rede de comutação de circuitos para acessar uma série de exames de raios-X de um paciente. O médico estabelece uma conexão, solicita um exame, analisa os resultados e solicita o próximo. No caso, os recursos da rede não são utilizados durante o tempo em que o médico esta analisando os exames. Além disto, os tempos necessários para o estabelecimento de circuitos fim-a-fim são grandes, além de ser uma tarefa complicada e requerer esquemas complexos de sinalização ao longo de todo o caminho da comunicação.

Por outro lado, os opositores da comutação de pacotes argumentam que a mesma não seria apropriada para aplicações tempo real, como por exemplo conversar telefônicas, devido os atrasos

variáveis em filas de espera, difíceis de serem previstos. Todavia, com o avanço tecnológico e o aumento da velocidade dos enlaces, observa-se uma tendência em direção à migração dos serviços telefônicos também para a tecnologia de comutação de pacotes.

Roteamento em redes de comutação de pacotes

Há duas classes de redes de comutação de pacotes, as redes baseadas em **datagramas**, como a Internet, e as redes baseadas em **circuito virtual**. A diferença básica destas duas redes está na forma como os pacotes são roteados em direção ao destino.

Roteamento em redes baseadas em circuito virtual

Nas redes baseadas em **circuito virtual**, a rota para os pacotes é estabelecida a priori, numa fase de estabelecimento do circuito virtual. Uma vez estabelecido o circuito virtual, todos os pacotes seguem pela mesma rota, cada um deles carregando a informação de qual circuito virtual o mesmo deve tomar em cada roteador. Os exemplos de redes que utilizam esta técnica incluem as redes X.25, as redes *frame-relay* e as redes ATM (*asynchronous transfer mode*).

O processo de estabelecimento de um circuito virtual é similar ao estabelecimento de conexão nas redes de comutação de circuitos, entretanto, os enlaces individuais não ficam reservados de forma exclusiva para uma única conexão, podendo, durante uma transmissão, serem compartilhados por outras transmissões.

Fazendo uma analogia, podemos comparar o estabelecimento de um circuito virtual com o planejamento de uma viagem de carro, definindo o trajeto a priori, com a ajuda de um mapa e consulta a Polícia Rodoviária para verificar o estado das rodovias até o destino. Durante a viagem, o motorista segue, com a ajuda do mapa, o trajeto anteriormente estabelecido. Veja também, que as estradas não ficam reservadas para um único veículo; outros carros, que provavelmente seguem a outros destinos, compartilham trechos das rodovias.

Roteamento em redes baseadas em datagrama

Nas redes baseadas em **datagramas**, não há estabelecimento de conexão ou circuito virtual. Os pacotes são encaminhados em função do endereço do destino. No caso da Internet, é o **endereço IP** que vai ser utilizado para a definir a rota que o pacote vai seguir.

Voltando a analogia da viagem de carro, no caso de uma rede de datagramas, podemos comparar com a realização da viagem pedindo informações em cada entroncamento, onde o motorista não conhece os caminhos e nem possui mapas. Por exemplo, suponha que você vai realizar uma viagem de Florianópolis para a cidade de Araraquara no interior de São Paulo usando este processo. Você chega ao primeiro posto na saída de Florianópolis e pergunta como chegar a Araraquara. Visto que o estado é São Paulo, o informante lhe diz para pegar a BR-101 no sentido norte e quando chegar a Curitiba perguntar novamente. Chegando em Curitiba, você faz novamente a pergunta a um policial rodoviário e ele lhe diz que a BR-116, rumo a São Paulo, está bem congestionada e lhe recomenda a saída para o estado de São Paulo via o norte do Paraná, orientando para que pergunte novamente quando chegar na divisa dos estados, na cidade de Ourinhos. Em Ourinhos, lhe indicam a estrada rumo a cidade de Bauru, onde você deverá fazer nova pergunta. Finalmente, em Bauru, lhe indicam a auto-estrada que vai diretamente a Araraquara. Neste exemplo, veja que as decisões em cada entroncamento são tomadas tendo como base o endereço final.

Tomando um exemplo diferente, em muitos aspectos as redes baseadas em datagramas são análogas aos serviços postais. Quando alguém vai enviar uma carta a um destinatário, o mesmo coloca a carta em um envelope e escreve o endereço do destino sobre o envelope. O endereço tem uma estrutura hierárquica, incluindo, no caso do Brasil, o país, o estado, a cidade, a rua e o número da casa. Por exemplo, se alguém enviar uma carta da França para nossa escola, o correio da França primeiro vai

direcionar a carta para o centro postal do Brasil (por exemplo, situado em São Paulo). O centro postal do Brasil vai então direcionar a carta para Santa Catarina, estado destino da carta (na agência central de Florianópolis, por exemplo). A agência de Florianópolis vai então direcioná-la a agência de São José, que por sua vez vai repassar ao carteiro para entregar a carta aqui na escola.

Na rede baseada em datagrama, cada pacote atravessa a rede contendo no cabeçalho o endereço do nó destino, que como o endereço postal, tem uma estrutura hierárquica. Quando o pacote chega a um roteador, o mesmo examina uma parte do endereço e o encaminha ao roteador adjacente.

A figura 1.6 mostra uma taxonomia das redes de telecomunicações.

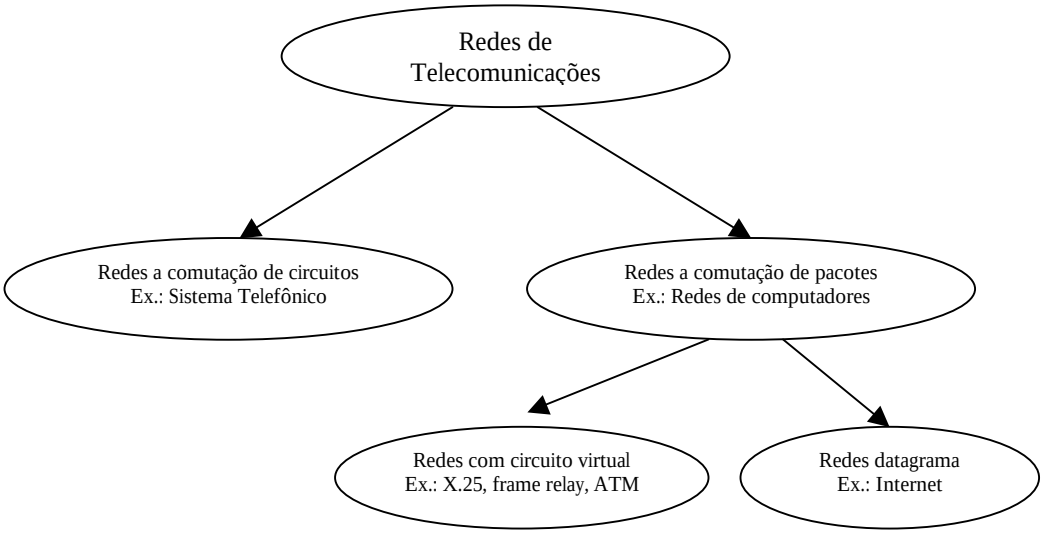


Figura 1.6. Taxonomia das redes de telecomunicações

Redes de acesso a Internet e meios físicos

Como vimos, na periferia da Internet estão os sistemas terminais que rodam as aplicações e no núcleo da rede estão os roteadores, responsáveis pela interconexão das redes. Neste item vamos abordar as **redes de acesso** a Internet, ou seja, quais as diferentes maneiras de conectar um computador a Internet.

Grosso modo podemos dividir as **redes de acesso** em duas categorias:

- Redes de **acesso residencial**;
- Redes de **acesso corporativo**.

Uma rede de **acesso residencial** conecta tipicamente um computador pessoal, instalado na casa de um usuário, a um roteador de borda, provavelmente de um provedor de acesso doméstico. A forma mais comum de acesso residencial é o **acesso via modem e linha discada**. O modem residencial converte o sinal digital do computador num formato analógico para ser transmitido sobre a linha telefônica analógica. No lado do provedor, outro modem vai converter o sinal analógico de volta a forma digital. Neste caso, a rede de acesso é um simples enlace ponto-a-ponto sobre o par trançado da linha telefônica. Em termos de velocidade de transmissão, os modems analógicos transmitem em taxas que vão até 56 Kbps, todavia, devido à má qualidade das linhas, dificilmente este valor é atendido de forma plena. (Figura 1.7)

Outra forma de acesso residencial, a qual não necessita a conversão analógica/digital, é a utilização da tecnologia **RDSI** (Rede Digital de Serviços Integrada), disponível em algumas centrais telefônicas das concessionárias de telecomunicações.

Novas tecnologias, como o **ADSL** (*asymmetric digital subscriber line*) e o **HFC** (*hibric fiber coaxial cable*) também tem sido empregada para acesso residencial.

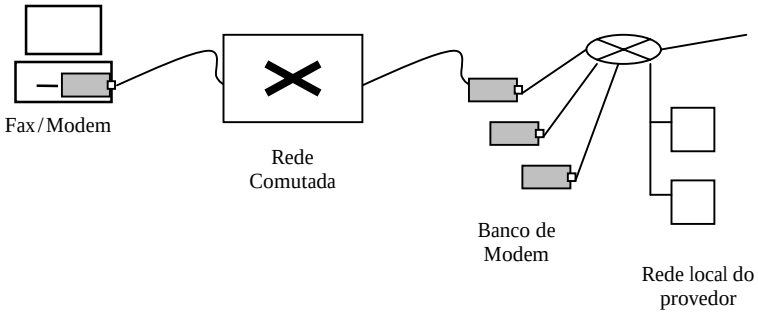


Figura 1.7. Acesso residencial via modem e linha discada

O **ADSL** usa **multiplexação por divisão da freqüência** para dividir o enlace de comunicação entre a casa do usuário e o provedor em três faixas de freqüência:

- Um canal de alta velocidade (*downstream*) de até 8 Mbps, na faixa de 50 kHz a 1 MHz;
- Um canal de média velocidade (*upstream*) de até 1 Mbps, na faixa de 4 kHz a 50 kHz;
- Um canal de baixa velocidade para o sinal telefônico de voz, na faixa de 0 a 4 kHz.

O ADSL permite velocidades de até 8 Mbps do provedor a residência (*downstream*) e no sentido reverso (*upstream*) até 1 Mbps. Esta assimetria é uma das características do ADSL e reflete as características de uso do usuário residencial, muito mais um consumidor do que um fornecedor de informações da Internet.

A tecnolocia **HFC,** também conhecida como *cable modem*, requer modems especiais para permitir um acesso doméstico a partir dos sistemas de distribuição de **TV a cabo**. O ***cable modem*** é um dispositivo externo, conectado ao computador pessoal a partir de uma porta Ethernet (Ethernet é uma tecnologia de rede local). Como no caso do ADSL, o *cable modem* divide o canal de acesso em duas bandas, um canal do provedor a residência de até 10 Mbps e 768 Kbps no sentido reverso. No HFC (e não no ADSL) estas velocidades de acesso são compartilhadas entre os usuários, pois a distribuição da TV a cabo usa um meio compartilhado entre vários usuários (*broadcast*).

Uma **rede de acesso corporativo** é tipicamente uma **rede local** de computadores conectada a um roteador de borda. Existem várias tecnologias de rede local, todavia, a tecnologia **Ethernet** é hoje uma das mais disseminadas. A Ethernet opera em velocidades de 10 Mbps a 100 Mbps (existe ainda a Ethernet a 1 Gbps). Ela usa par trançado de cobre ou cabo coaxial para conexão entre as máquinas, que compartilham um barramento comum, sendo portando a velocidade de acesso também compartilhada entre os usuários (Figura 1.8).

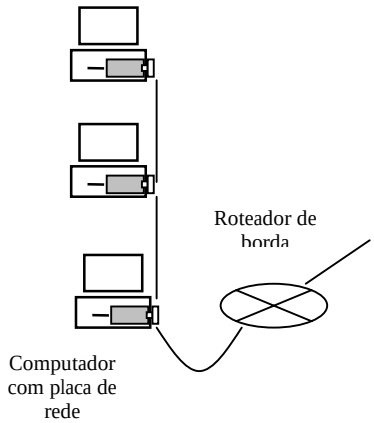


Figura 1.8. Acesso corporativo via rede local

Meios físicos

Como meio físico podemos ter, por exemplo, **par trançado**, **cabo coaxial**, **fibra óptica** ou a utilização do ar e do **espectro de freqüência de rádio**. A conexão ao meio físico pode se dar de diversas maneiras, onde cada uma delas utiliza protocolos específicos, necessitando de **dispositivos**

adaptadores, por exemplo, placas fax/modem e placas de rede. O tipo de acesso e o meio físico utilizado determinará uma **taxa de transmissão** de dados para o enlace de comunicação.

O que são camadas de protocolos?

Uma rede de computadores é um sistema bastante complexo. Como vimos, a interação entre os computadores e as diversas aplicações pode se dar de diferentes maneiras e a partir da utilização de um número variado de protocolos. Para lidar com esta complexidade, a **arquitetura das redes de computadores** procurou estabelecer uma série de **camadas de protocolos** cada uma delas tratando de uma funcionalidade específica da comunicação.

Analogia com sistema postal (Correios)

Para entender o papel das **camadas de protocolo** utilizadas nas redes de computadores, vamos fazer uma analogia com um sistema postal hipotético.

Por exemplo, para enviar uma carta neste **sistema postal** o **usuário** deverá primeiramente acondicioná-las em um **envelope** padronizado. Em seguida, ele deve escrever, também segundo algumas regras, o **endereço** do destinatário. Note que o endereço é hierarquizado, onde consta o nome do usuário final, o nome da rua, a cidade, o estado e o país. Feito isto o usuário deve selar a carta e depositá-la em uma **caixa coletora** do serviço postal.

Os **carteiros** do sistema postal são responsáveis por diariamente coletar as correspondências nas caixas coletoras e levá-las até a **agência de triagem local** dos correios.

A **agência de triagem local** realiza um primeiro **serviço de triagem** das correspondências, a partir do endereço dos destinatários, **e** define o **encaminhamento** seguinte das mesmas. Para alguns destinos pode haver um encaminhamento direto a partir da agência local (por exemplo, uma localidade vizinha). Para outros destinos (por exemplo, uma cidade de outro estado) o encaminhamento pode se dar via outra **agência de triagem intermediária**. Para encaminhar as correspondências ao próximo destino, todas as cartas cujas **rotas** devem seguir por esta destinação são acondicionadas em um **malote**, e seguirão por um **serviço de malote**.

O **serviço de malote** carrega os malotes entre as "**agências vizinhas"** (isto é, as quais possuem serviço de malote direto). Dependendo das agências em questão, o transporte dos malotes pode ser realizado de diferentes maneiras. Por exemplo, via linha aérea comercial, via linha de transporte rodoviário, com transporte rodoviário próprio, etc.

Uma vez na próxima **agência de triagem** o malote é aberto e nova triagem é realizada. Este processo de **roteamento** das correspondências entre as agências de triagem prossegue até que a correspondência chegue a **agência destino**, responsável pela jurisdição onde habita o destinatário final.

Uma vez na **agência destino** as cartas são separadas e repassadas aos **carteiros** para fazerem a entrega a domicílio das cartas aos **destinatários finais**. (veja diagrama mostrado na Figura 1.9)

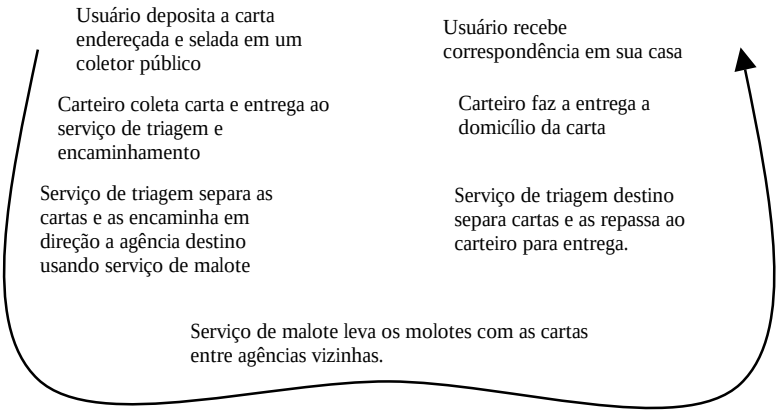


Figura 1.9. Ações para encaminhar uma carta no serviço postal

Estruturação do sistema em camadas

Todo este processo tem analogia com as redes de computadores. Por exemplo, uma mensagem entre um computador conectado a uma rede e outro de uma rede remota deve ser encaminhada desde a rede do remetente, seguindo uma determinada rota, até atingir o computador destino. Todavia, a analogia que estamos buscando está na **estrutura** mostrada na figura 1.9.

Como podemos observar, cada funcionalidade no processo de envio de uma correspondência tem uma etapa correspondente no lado do destinatário. Poderíamos então organizar estas funcionalidades organizando-as em **camadas horizontais** (Figura 1.10).

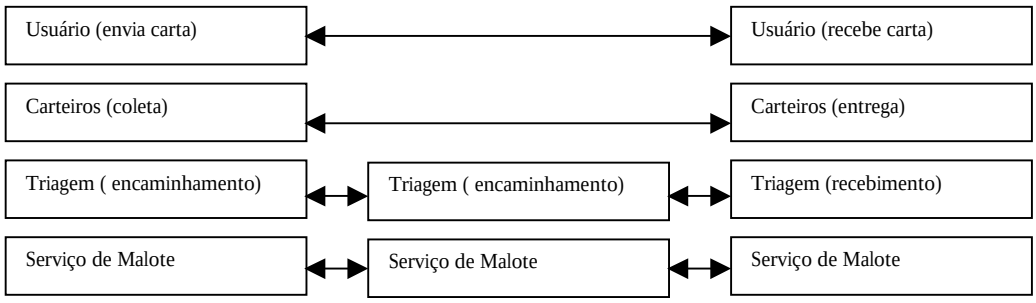


Figura 1.10. Estrutura em camadas do serviço de correios

Estas camadas horizontais permitem que cada funcionalidade seja descrita de forma separada, onde cada camada guarda uma certa independência das demais. Por exemplo, para o **usuário**, uma vez que ele depositou uma carta no coletor, não lhe interessa como a mesma vai ser entregue ao destinatário. Ele simplesmente conta com o sistema postal para isto.

Por sua vez, os **carteiros** não se preocupam com o conteúdo das correspondências e nem em como serão enviadas ao destinatário. Seu serviço é coletar as correspondências e levá-las o setor de triagem. A forma como os carteiros realizam seu trabalho pode ser alterada sem afetar as demais camadas. Por exemplo, utilizar uma bicicleta ao invés de ir a pé para coletar as cartas.

Quanto às **agências de triagem e encaminhamento**, da mesma forma, sua organização interna pode ser alterada sem prejudicar o restante do processo. Por exemplo, uma agência de triagem mais importante pode contar com serviços automatizados para separação de cartas, outras menores, podem realizar a separação manualmente.

O mesmo tipo de comentários poderia ser feito com respeito ao **serviço de malote** das correspondências. Por exemplo, entre duas agencias de triagem que possuem um grande fluxo de correspondências, como entre duas capitais, poderia haver um serviço de malote dedicado via aérea.

Note que nas **agências terminais**, **todas** as **camadas** do sistema postal precisam ser implementadas, incluindo caixas coletoras e os serviços de carteiros para coleta/entrega de cartas. Por outro lado, podemos ter algumas **agências intermediárias** dedicadas somente à **triagem e encaminhamento**, localizadas, por exemplo, em nós importantes do sistema. Neste caso, as camadas superiores não precisam ser implementadas.

Cada camada oferece um **serviço** à camada superior:

- **A partir da realização de algumas ações,** por exemplo, os carteiros coletam as cartas e as repassam ao serviço de triagem e encaminhamento**;**
- **Utilizando os serviços da camada inferior,** por exemplo, os carteiros contam com o serviço de triagem e encaminhamento para que continue o processo de entrega até o destinatário**.**

Camadas de protocolos nas redes de computadores

Durante os primeiros tempos das redes de computadores os diversos fabricantes trabalharam de forma separada no desenvolvimento de suas tecnologias, muitas delas incompatíveis entre si. Com o intuito de estabelecer alguma padronização e permitir uma integração entre as diversas tecnologias, a **ISO** (*International Standard Organization* – `www.iso.org`), juntamente com o **ITU** (*International Telecommunication Union* – `www.itu.org`), organismos responsáveis pelo estabelecimento de normas e padrões em telecomunicações no mundo, definiram um **modelo de referência** com **sete camadas** de protocolos. Este modelo ficou conhecido como **modelo OSI** (*open system interconnection*).

As **camadas de protocolos** facilitam o projeto e a implementação das redes de computadores, e no nosso caso, também o ensino e a aprendizagem das redes. Através das camadas de protocolos, o problema de construir uma rede fica decomposto em diversos módulos, onde cada camada pode ser implementada separadamente, sem afetar as demais.

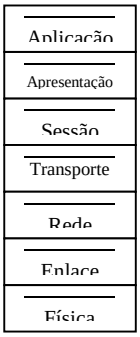


Figura 1.15. Modelo de 7 camadas ISO

As sete camadas do modelo OSI, nomeadas como **aplicação, apresentação, sessão, transporte, rede, enlace** e **física** (Figura 1.15), tiveram muito sucesso na literatura de redes de computadores, todavia, não tiveram o mesmo sucesso comercial. Hoje, não há nenhum produto que siga a risca as recomendações do modelo OSI. Dentre os modelos comerciais, certamente a arquitetura Internet é a que tem hoje maior sucesso. Grosso modo, pode-se dizer que o **modelo Internet** é uma simplificação do **modelo OSI**, onde algumas camadas agrupam funcionalidades de mais de uma camada do modelo OSI.

Modelo em camadas da Internet

O modelo em camadas da Internet agrupa as funcionalidades das redes de computadores em quatro camadas. A camada superior, que define regras para a troca de mensagens entre os **processos de aplicação**. A segunda camada que oferece um **canal de comunicação lógico fim-a-fim entre os processos de aplicação,** oferecendo um serviço apropriado para que os processos de aplicação troquem mensagens. A terceira camada, que trata os problemas relativos ao roteamento de pacotes entre dois computadores remotos, permitindo a **conectividade fim-a-fim entre dois computadores.** Por fim, a camada inferior que trata os problemas relacionados aos enlaces de comunicação entre nós vizinhos e os problemas relacionados à transmissão física de bits sobre os enlaces (Figura 1.11).

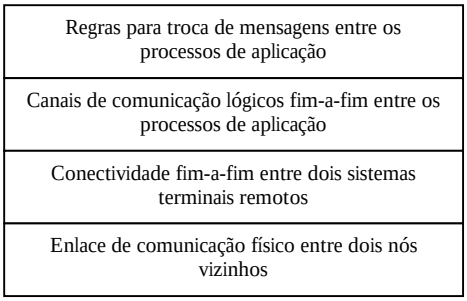


Figura 1.11. Diferentes camadas para as redes de computadores

No caso dos **canais de comunicação lógico fim-a-fim entre os processos de aplicação**, para atender aos dois tipos de aplicações descritos anteriormente (aplicações tipo **pedido/resposta** e aplicações tipo **fluxo de dados tempo real**), poder-se-ia ter dois canais distintos, conforme mostra a figura 1.12.

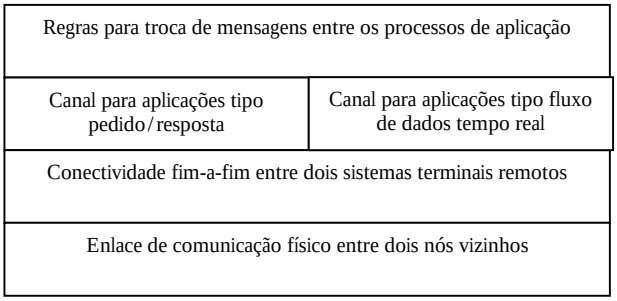


Figura 1.12. Diferentes canais para diferentes tipos de aplicação

Dentro do padrão Internet, esta **pilha de protocolos** tem as camadas assim denominadas: **camada aplicação, camada transporte, camada rede** e **camada enlace/física**. (Figura 1.13)

Figura 1.13. Pilha de protocolos da Internet

Camada Aplicação

Os protocolos da **camada de aplicação** definem as regras e o formato das mensagens que são trocadas entre as aplicações de rede, por exemplo, a aplicação WWW (*world wide web*) é governada pelas regras do protocolo de aplicação **HTTP** (*hiper text transfer protocol*); o correio eletrônico envia as mensagens usando o protocolo de aplicação **SMTP** (*simple mail transfer protocol*); a transferência de arquivos usa o protocolo de aplicação **FTP** (*file transfer protocol*). As **mensagens** trocadas entre as entidades da camada aplicação utilizam os canais disponibilizados pelos protocolos da camada inferior.

A camada de aplicação estabelece as regras para a troca de mensagens entre as aplicações.

Voltando ao exemplo do sistema postal, a camada de aplicação corresponderia às regras que os usuários devem obedecer para utilizar os serviços postais, como utilizar envelope apropriado, escrever o endereço e o CEP corretamente, selar a carta e depositar na caixa de coleta.

Camada Transporte

Os protocolos da **camada de transporte** garantem um **canal de comunicação lógico fim-a-fim entre os processos** rodando no lado do cliente e no lado do servidor, para que as aplicações possam trocar mensagens entre si. Para atender aos dois tipos de aplicações comentados anteriormente, **aplicações tipo pedido/resposta** e **aplicações tipo fluxo de dados em tempo real**, a Internet implementa dois protocolos de transporte, o **TCP** e o **UDP**. O TCP fornece um serviço confiável e orientado a conexão. O UDP fornece um serviço sem conexão (*connectionless*) e não confiável.

Como em cada computador da rede podemos ter diferentes processos de aplicação rodando, por exemplo, várias seções de navegadores *Web*, um dos serviços oferecidos pela camada de transporte é a **multiplexação/demultiplexação de aplicações**, entregando as mensagens na **porta** apropriada de cada processo.

A camada de transporte estabelece um canal de comunicação lógico para a transferência de mensagens porta-a-porta entre os processos de aplicação rodando em dois computadores remotos.

Camada Rede

Dentro da Internet, as mensagens são fragmentadas em **pacotes**, chamados **datagramas**, e atravessam a rede de roteador em roteador desde o computador origem até o computador destino usando a técnica de **comutação de pacotes**. Nesta viagem, uma das tarefas dos protocolos da **camada de rede** é definir a **rota** que seguirão os datagramas. A camada rede da Internet tem dois componentes principais, o **protocolo IP**, que define o formato do datagrama e a forma de endereçamento, e os **algoritmos de roteamento**.

A camada de rede realiza a transferência de pacotes, ou datagramas, entre dois computadores remotos.

A camada rede envolve cada computador e roteador do caminho entre o computador origem e o destino, diferentemente das camadas de aplicação e transporte que somente precisam implementadas nas duas pontas da comunicação.

Retomando nosso exemplo do sistema postal, o serviço executado pela camada rede é análogo ao serviço executado pelas agências postais. As agências recebem as correspondências coletadas pelos carteiros, realizam os **serviços de triagem e encaminhamento** de correspondências entre agências e por fim repassam novamente a um carteiro da agencia remota para entregar na casa do destinatário.

Relacionamento entre as camadas de transporte e de rede

A camada de transporte se situa logo acima da camada de rede na pilha de protocolos. Enquanto os protocolos de transporte oferecem comunicação lógica entre processos rodando em diferentes computadores, a camada de rede oferece comunicação lógica entre os computadores. A diferença é sutil, mas importante. Vamos analisá-la fazendo uma analogia com residências atendidas pelo nosso sistema postal hipotético.

Neste sistema, a agência da jurisdição do destinatário entrega as cartas no endereço da residência do usuário com a ajuda dos carteiros. Todavia, um mesmo endereço pode pertencer a mais de uma pessoa. Quando chega uma correspondência a uma pessoa da residência, alguém deve se encarregar de recebê-la e entregá-la ao usuário final. Neste exemplo, a pessoa que recebeu a correspondência do carteiro faz um papel análogo ao serviço de multiplexaxão de aplicações realizado pelos protocolos da camada transporte.

Camada Enlace

Para mover um pacote de um **nó** até o **nó adjacente**, dentro de uma determinada rota, a camada rede necessita dos serviços dos protocolos da **camada de enlace**. Por exemplo, para transferir dados entre dois computadores conectados em uma rede local, o protocolo de enlace de múltiplo acesso **Ethernet** pode ser utilizado. Já no caso de dois computadores conectados via linha discada, o protocolo de enlace ponto-a-ponto **PPP** poderia ser utilizado.

> **A camada de enlace realiza a transferência de dados entre nós vizinhos da rede.**

Comparando com o sistema postal, a camada enlace é análoga a camada que realiza os serviços de transporte das cartas entre agências vizinhas e entre agências e usuários. Isto engloba tanto o serviço de malote entre agências, quanto o trabalho realizado pelos carteiros levando as cartas entre as agências de correio e as residências dos usuários.

Camada Física

Vinculado à camada enlace está a **camada física,** que é responsável por mover os **bits** que compõe os dados entre um nó e outro utilizando um meio físico específico. Os **meios físicos** podem ser cabos coaxiais, fios de cobre, fibras ópticas ou o ar a partir do uso do espectro de freqüência de rádio.

> **A camada física realiza o transporte de bits sobre o meio físico de um enlace de comunicação.**

No caso do sistema postal, a camada física corresponderia ao meio de transporte utilizado pelos carteiros ou pelo serviço de malote para transportar as cartas, como por exemplo, bicicleta, carro, ônibus, etc.

Questões

1. A **conectividade** entre computadores pode se dar em diferentes escalas. Comente sobre as formas de se conectar computadores, citando exemplos de redes existentes na prática.

2. Quais tipos de dispositivos podem ser conectados a Internet além de computadores pessoais. Cite exemplos e pesquise endereços URL que apresentem algum dispositivo deste tipo.
3. O que é um **sistema terminal** ou **hospedeiro** (*host*)? Explique o porquê deste nome.
4. O que é um **roteador**? Quais são suas funções nas redes de computadores?
5. Explique a expressão ***store-and-forward***, relativa ao funcionamento de um roteador.
6. Quais as vantagens e desvantagens da **comutação de circuitos** em relação com a **comutação de pacotes**?
7. Pesquise sobre a **comutação de mensagens** e diferencie esta técnica da **comutação de pacotes.**
8. O que é uma **aplicação de rede**? Cite exemplos e mostre a utilidade de cada aplicação citada.
9. O que é um **protocolo**? Cite um exemplo de um protocolo humano que você usa no seu dia-a-dia.
10. Quais os principais **protocolos da Internet**?
11. Qual a origem no nome Internet?
12. O que é um **endereço IP**?
13. O que significa ter os computadores conectados em **rede local**? Como uma rede local pode ser conectada a Internet?
14. Explique o que é o **modelo cliente/servidor**, obedecido pela maioria das aplicações Internet.
15. As aplicações Internet requisitam serviços da rede subjacente. Diferencie os **serviços do tipo pedido/resposta** dos **serviços tipo fluxo de dados tempo real**. Cite exemplos.
16. Porque se diz que a Internet é dita uma rede ***best-effort***? Explique.
17. Pesquise sobre a forma de **acesso doméstico** a Internet utilizando **RDSI**, disponível na região da Grande Florianópolis. Explicitar tanto os aspectos tecnológicos quanto os comerciais, descrevendo também a tecnologia utilizada para a transmissão dos dados e os equipamentos necessários.
18. Idem para a tecnologia **ADSL.**
19. Idem para a tecnologia ***cable modem***.
20. Na tecnologia ***cable modem*** o canal é dedicado ou compartilhado entre os usuários? Explique.
21. Comente sobre pelo menos três vantagens de se dividir a arquitetura das redes de computadores em **camadas**.
22. Quais as principais funções de cada uma das camadas da **arquitetura Internet**?
23. Qual camada da Internet faz o processo de **roteamento**?

24. Explique porque os protocolos da **camada rede** (como o IP) devem ser implementados em todos os nós da rede (como sistemas terminais e roteadores) e os protocolos da **camada transporte** (como o TCP) somente precisam ser implementados nos sistemas terminais.

25. Faça um levantamento da **topologia da Internet no Brasil** (rede acadêmica e privada), mostrando os *backbones* e provedores nacionais, regionais e locais.

26. O que é **telefonia na Internet**? Ache algum URL sobre telefonia na Internet que descreva alguns produtos existentes.

27. Pesquise sobre **distribuição de áudio** na Internet. Ache algum URL que ofereça este serviço.

28. O que é **vídeo conferência** na Internet? Explique como funciona.

Parte 2

Aplicações de Rede

As aplicações de rede são a "razão de ser" da Internet, permitindo que os usuários possam fazer coisas úteis e interessantes na rede. Sem as aplicações, a Internet não teria sentido.

O que é uma aplicação de rede?

As **aplicações de rede** são programas que rodam nos **sistemas terminais** ou **hospedeiros** (*hosts*) e se comunicam entre si através da rede. São programas de aplicação típicos da Internet: o ***login* remoto a sistemas** (Telnet ou SSH), a **transferência de arquivos** (FTP), o **correio eletrônico** (*e-mail*), a **paginação na *Web*** ou **WWW** (*world wide web*), o **bate-papo** em rede (*chat*), **telefonia** na Internet (*VoIP*), a **vídeo conferência**, a **execução de áudio e vídeo**, etc.

As aplicações de rede são programas ou, como dizem no jargão dos sistemas operacionais, processos que se comunicam entre si pela da troca de **mensagens** através da rede.

A Internet oferece o suporte para a troca de mensagens entre as aplicações através de **canais de comunicação lógicos**, que são oferecidos pelos **protocolos TCP/IP**.

Protocolos de aplicação

Além do TCP/IP, cada aplicação utiliza protocolos específicos, chamados **protocolos de aplicação**, que definem como os processos de aplicação, rodando em diferentes computadores, trocam mensagens entre si. Em particular, os protocolos de aplicação definem:

- Os tipos de mensagens trocadas, por exemplo, uma mensagem de solicitação ou resposta;
- A sintaxe e a semântica das mensagens, definindo os campos de cada mensagem e seu significado;
- As regras definindo quando e como um processo envia ou responde uma mensagem.

Os **protocolos de aplicação**, apesar de importantes, são apenas uma pequena parte de uma aplicação de rede. Por exemplo, a aplicação **WWW** é uma aplicação de rede que permite aos usuários obterem "documentos" da *Web* sob demanda. Os componentes da aplicação WWW incluem documentos em formato HTML (*hypertext markup language*), navegadores *Web* (como o Netscape ou o Internet Explorer), servidores de páginas *Web* (como o Apache do Linux, o

Microsoft Internet Information Server ou Netscape Server) e o **protocolo de aplicação HTTP** (*hiper text transfer protocol*).

Da mesma forma, o **correio eletrônico** (*e-mail*) tem vários componentes, incluindo os servidores que hospedam as caixas postais dos usuários, os leitores de correio eletrônico que permitem ler e criar mensagens, um padrão que define a estrutura das mensagens eletrônicas, os **protocolos de aplicação**, cujo principal é o **SMTP** (*simple mail transfer protocol*), que definem como as mensagens são trocadas entre os servidores e entre os servidores e os leitores de correio eletrônico.

Clientes e servidores

Uma aplicação de rede tem tipicamente duas partes, um lado **cliente** e um lado **servidor** que se comunicam entre si. Por exemplo, um navegador *Web* implementa o lado cliente do HTTP, e um servidor *Web* implementa o lado servidor http (Figura 2.1).

Para algumas aplicações, um computador pode implementar ora o lado cliente ora o lado servidor. Por exemplo, considere um de acesso remoto via Telnet entre um computador A e um computador B. Se o computador A inicia a seção Telnet, então A é o cliente e B é o servidor. Por outro lado, se o computador B inicia a seção, ele é que será o cliente e A o servidor. Da mesma forma, na aplicação de correio eletrônico, o servidor que envia uma mensagem, implementa o lado cliente do SMTP e o servidor que recebe a mensagem implementa o lado servidor.

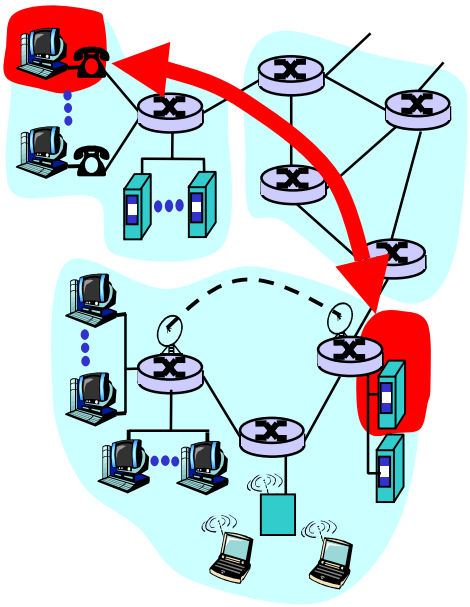


Figura 2.1. Interação cliente/servidor na Internet

Comunicação através da rede

Como visto, uma aplicação envolve a comunicação de dois processos através da rede. Os dois processos se comunicam através do envio e recebimento de mensagens através mecanismos chamados **portas** (*sockets*). Os processos assumem que há uma infraestrutura de transporte no outro lado da porta do processo emissor que transportará as mensagens até a porta do processo destino.

O conceito de **portas** faz parte da implementação dos protocolos de transporte da Internet **TCP** e **UDP**. Em resumo, pode-se dizer que os protocolos de transporte estabelecem um **canal de comunicação lógico** para a transferência de mensagens **porta-a-porta** entre os processos de aplicação rodando em dois computadores remotos.

Na Internet cada uma das aplicações mais conhecidas, utilizam portas padronizadas, por exemplo, um servidor *Web* é identificado pela porta 80. Um servidor de correio eletrônico pela porta 25. Um servidor Telnet pela porta 23.

Endereçamento

Para que um processo em um computador possa enviar uma mensagem a um computador remoto, ele deve endereçar quem vai receber a mensagem. O **endereço** envolve duas peças de informação: (1) o **nome** ou o **endereço IP** da máquina destino; (2) o **número da porta** do processo do lado do receptor.

Por exemplo, para **endereçar** o servidor *Web* do CEFET em São José devemos fornecer o endereço IP ou o nome de domínio da máquina responsável por servir as páginas Internet, ou seja, IP: 200.135.233.1 ou www.sj.cefetsc.edu.br, respectivamente. Quanto ao **número da porta**, como algumas aplicações tem o número padronizado, em geral, o agente usuário escolhe o número de porta automaticamente em função da aplicação em uso.

Agente usuário

O **agente usuário** é a interface entre o usuário e a aplicação de rede. Mais especificamente, o agente usuário é um programa de computador, comercial ou de domínio público, que implementa a interface do lado cliente de uma aplicação de rede.

Por exemplo, o agente usuário para um cliente WWW pode ser, por exemplo, o Internet Explorer da Microsoft, o Netscape Navigator ou o Mozilla. O agente usuário para um leitor de correio eletrônico pode ser, por exemplo, o Outlook da Microsoft, o Eudora ou o Netscape Messager.

Qual serviço de transporte uma aplicação precisa?

A Internet dispõe de dois serviços de transporte para os protocolos de aplicação, o **UDP** (*user datagram protocol*) e o **TCP** (t*ransmission control protocol*). Quando uma aplicação é projetada para a Internet, a primeira decisão do projetista deve ser definir qual protocolo de transporte será utilizado.

A escolha dependerá do tipo **serviço** que a aplicação vai necessitar. Quanto aos tipos de serviços requisitados pelas aplicações, podemos classificá-los em três dimensões:

- Quanto à **perda de dados**

  Algumas aplicações, como por exemplo, transmissão de áudio, podem tolerar algumas perdas; outras aplicações, como por exemplo, uma transferência de arquivos ou um Telnet, requerem transferência 100% confiável.

- Quanto aos **requisitos temporais**

  Algumas aplicações, como por exemplo, telefonia na Internet e jogos interativos, requerem baixo retardo para serem "viáveis", outras, como uma mensagem de correio eletrônico, não tem restrições temporais.

- Quanto à **largura de banda**

  Algumas aplicações, como por exemplo, multimídia, requerem quantia mínima de banda para serem "viáveis"; outras aplicações são mais "elásticas" e conseguem usar qualquer quantia de banda disponível, como por exemplo, a paginação na *Web*.

A tabela abaixo apresenta algumas aplicações típicas da Internet e os requisitos em termos de transporte.

Aplicação	Perdas	Banda	Requisitos temporais
transferência de arquivos	sem perdas	elástica	não
Correio eletrônico	sem perdas	elástica	não
WWW	sem perdas	elástica	não
áudio/vídeo de tempo real	tolerante	áudio: 5Kb-1Mb vídeo:10Kb-5Mb	sim, 100's mseg
áudio/vídeo gravado	tolerante	como anterior	sim, alguns segs
jogos interativos	tolerante	> alguns Kbps	sim, 100's mseg
aplicações financeiras	sem perdas	elástica	sim e não

Para atender a estes requisitos, os dois protocolos de transporte da Internet oferecem as seguintes facilidades:

Serviço TCP:

- **Serviço orientado a conexão:** uma abertura de conexão é requerida entre cliente e servidor;
- **Transporte confiável:** garante comunicação livre de erros entre o processo emissor e receptor;
- **Controle de fluxo**: evita que o emissor possa "afogar" com dados um receptor mais lento;
- **Controle de congestionamento**: permite "estrangular" o emissor quando a rede está sobrecarregada.
- **Não provê:** garantias temporais ou de banda mínima.

Serviço UDP:

- **Transferência de dados não confiável:** não há garantia de entrega de dados livre de erros entre o processo emissor e receptor**;**
- **Não Provê:** abertura de conexão, confiabilidade, controle de fluxo, controle de congestionamento, garantias temporais ou de banda mínima.

A tabela a seguir mostra algumas aplicações típicas e os respectivos protocolos de transporte utilizados.

APLICAÇÃO	Protocolo de aplicação	Protocolo de transporte
Correio eletrônico	SMTP	TCP
Login remoto	Telnet	TCP
WWW	HTTP	TCP
Transferência de arquivos	FTP	TCP
Servidor de arquivos remoto	NFS	tipicamente UDP
Gerenciamento de rede	SNMP	tipicamente UDP
Protocolo de roteamento	RIP	tipicamente UDP
Tradução de nomes	DNS	tipicamente UDP
Multimídia	proprietário	TCP ou UDP
Telefonia na Internet	proprietário	tipicamente UDP

A aplicação WWW

A aplicação **WWW** é uma aplicação de rede que permite aos usuários obterem "documentos", ou páginas *Web*, sob demanda. Uma **página *Web*** consiste de **objetos,** os quais podem ser arquivos HTML (*hypertext markup language*), imagens JPEG, imagens GIF, *applets* Java, clip de áudio e vídeo, etc, endereçados por um **endereço URL** (*universal resource locator*). A maioria das páginas *Web* consiste de uma **página base HTML** e várias referências, conhecidas como *hiperlinks*, para outros objetos. Páginas pessoais dos usuários são conhecidas como *home pages*.

Um **endereço URL** tem duas componentes: (1) o **nome do computador** que hospeda as páginas Web e (2) o **nome do objeto e o caminho** onde o mesmo esta localizado. Por exemplo, para acessar a página do Curso de Telecomunicações do CEFET, o endereço URL é

`www.sj.cefet.edu.br/principal/areas/tele/inicio.htm`

onde `www.sj.cefet.edu.br` é o endereço do servidor e `/principal/areas/tele/inicio.htm` especifica o caminho e o arquivo com o objeto solicitado.

Um **navegador Web**, como o Internet Explorer ou o Netscape Navigator, é o **agente usuário** para a aplicação WWW e implementa o lado **cliente** do protocolo HTTP. Um **servidor Web** hospeda as páginas Web, as quais são acessadas por seu endereço URL. Um servidor Web implementa o lado **servidor** do protocolo HTTP, sendo que, entre os servidores mais populares temos o Apache do Linux, o IIS (*Internet Information Server*) da Microsoft e o Netscape Server.

O protocolo HTTP

O protocolo HTTP define como os navegadores Web (clientes) requisitam páginas de servidores Web. Quando um usuário requisita um objeto, por exemplo clicando em uma referência de uma página Web, o navegador envia **mensagens de requisição HTTP** para o servidor Web. O servidor recebe a requisição e responde com uma **mensagem de resposta HTTP** que contém os objetos solicitados (Figura 2.2).

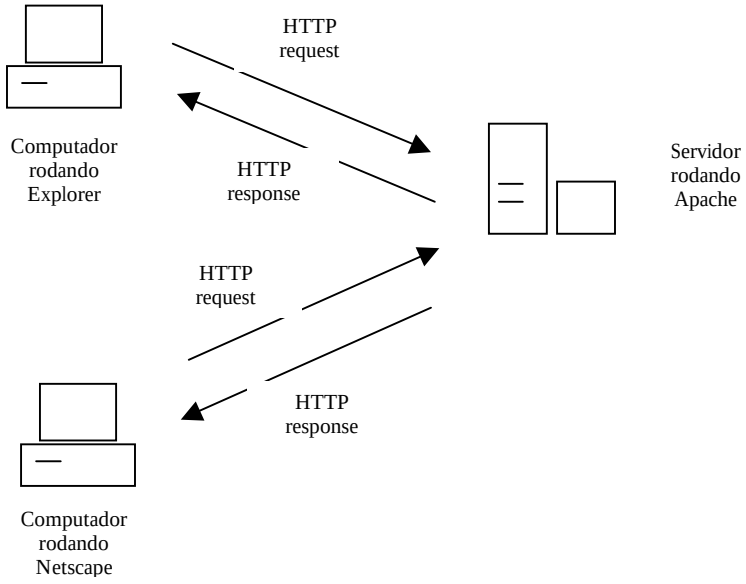


Figura 2.2. Protocolo HTTP

Há duas versões do protocolo HTTP implementadas pelos navegadores, o **HTTP/1.0** e o **HTTP/1.1** e ambas as versões usam como protocolo de transporte o **TCP**. Para requisitar uma página Web, o cliente HTTP primeiramente abre uma conexão TCP com o servidor. Uma vez aberta a conexão TCP o cliente e o servidor podem trocar mensagens através das suas **portas** de interface. A **porta 80** é o padrão para a aplicação WWW.

O **HTTP/1.0** usa o que se chama **conexões não persistentes**, onde após a requisição de cada objeto, o servidor responde e encerra a conexão TCP. Por exemplo, para uma página Web composta de um arquivo base HTML e mais 5 imagens JPEG, após a recepção de cada arquivo, a conexão TCP é encerrada e deverá ser reaberta para cada novo objeto requisitado (isto é feito automaticamente pelo agente usuário).

O **HTTP/1.1** permitiu melhorar o desempenho dos navegadores Web através do uso de **conexões persistentes**, onde o servidor mantém a conexão TCP aberta após o envio da resposta. Desta forma, as requisições e as respostas subsequentes entre o mesmo par cliente/servidor podem utilizar a mesma conexão já aberta, eliminando o tempo de abertura de conexão. Caso a conexão deixe de ser utilizada por um certo tempo o servidor se encarrega de liberar a conexão.

Formato das mensagens HTTP

O protocolo HTTP é baseado no **paradigma pedido/resposta**, havendo dois tipos de mensagens: **mensagens de requisição** e **mensagens de resposta**.

A **mensagens de requisição** (*request*) tem a seguinte estrutura:

```
GET /diretorio/pagina.html
Host: www.escolatecnica.edu.br
Connection: close
User-agent: Mozilla/4.0
Accept-language:pt
(extra carriage return, line feed)
```

A primeira linha apresenta o comando básico para **requisição** de uma página Web, seguido pela parte do URL que indica o caminho e o nome do objeto que se deseja (`GET /diretorio/pagina.html`). As linhas seguintes, chamadas de cabeçalho, são opcionais. A segunda linha (`Host: www.escolatecnica.edu.br`) indica o nome computador onde reside o objeto; a terceira linha (`Connection: close`) informa para fechar a conexão após envio da resposta; a quarta linha (`User-agent: Mozilla/4.0`) indica o tipo do agente usuário utilizado e a linha (`Accept-language:pt`) indica que o português é a língua preferencial.

Do ponto de vista do usuário o mesmo só "enxerga" o endereço URL que digitou e o navegador monta e envia as mensagens HTTP de forma transparente.

A **mensagens de resposta** (*response*) tem a seguinte estrutura:

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html

(data data data data data . . .)
```

A **resposta** tem três partes, uma primeira linha informando o estado (*status*) da solicitação, seis linhas de cabeçalho e os dados que compõe objeto solicitado. A primeira linha indica a versão do protocolo, o código e estado da mensagem (`HTTP/1.1 200 OK`). A segunda linha (`Connection: close`) indica que a conexão será encerrada; a terceira linha (`Date: Thu, 06 Aug 1998 12:00:15 GMT`) informa a data da última modificação no objeto solicitado, utilizada por servidores *proxy*; a quarta linha (`Server: Apache/1.3.0 (Unix)`) indica o tipo do servidor, a quinta linha (`Content-Length: 6821`) indica o tamanho do objeto em bytes e a última linha (`Content-Type: text/html`) informa o conteúdo da mensagem. Os dados vem em seguida.

Os códigos de estado (*status*) mais comuns são:

`200 OK`: Requisição OK e o objeto solicitado vai em anexo.
`301 Moved Permanently`: O objeto solicitado foi movido para outra URL.
`400 Bad Request`: Requisição não entendida pelo servidor.
`404 Not Found`: O objeto requisitado não existe no servidor.
`505 HTTP Version Not Supported`: Esta versão do protocolo HTTP não é suportada pelo servidor.

Exercício

É possível "ver" as mensagens trocadas pelo protocolo HTTP, executando manualmente os comandos em uma conexão TCP, na porta 80, com um servidor Web. Para tal, pode-se fazer primeiramente um Telnet (acesso remoto), na porta 80 de um servidor Web. Em seguida, uma vez estabelecida a conexão, pode-se trocar comandos HTTP manualmente.

Por exemplo:

> telnet www.sj.cefetsc.edu.br 80

GET /~cantu/index.html

permite estabelecer um canal TCP na porta 80 com servidor do CEFET em São José e acessar a página do professor Cantú.

Os navegadores Web

Os **navegadores Web**, como o Internet Explorer da Microsoft, o Netscape Navigator ou o Mozilla, implementam de forma transparente ao usuário o conjunto de comandos do HTTP, incluindo facilidades que permitem aos usuários ter um acesso às páginas Web de modo bem mais amigável.

Aplicação **de transferência de arquivos**

A **aplicação de transferência de arquivos** é suportada pelo protocolo de aplicação **FTP** (*file transfer protocol*) que é um protocolo para transferir arquivos de um computador para outro.

Numa típica **sessão FTP** um usuário pode transferir arquivos de um computador remoto para um computador local e vice-versa (*download* e *upload*, respectivamente). O usuário interage com o FTP através de um **agente usuário.** Primeiro fornece o **nome** (ou o endereço IP) do computador remoto, estabelecendo com isto uma **conexão TCP** entre o processo **FTP cliente** e **servidor**. Depois o usuário deve fornecer sua **identificação** e sua **senha**, para então poder executar **comandos FTP** para transferir arquivos (Figura 2.3).

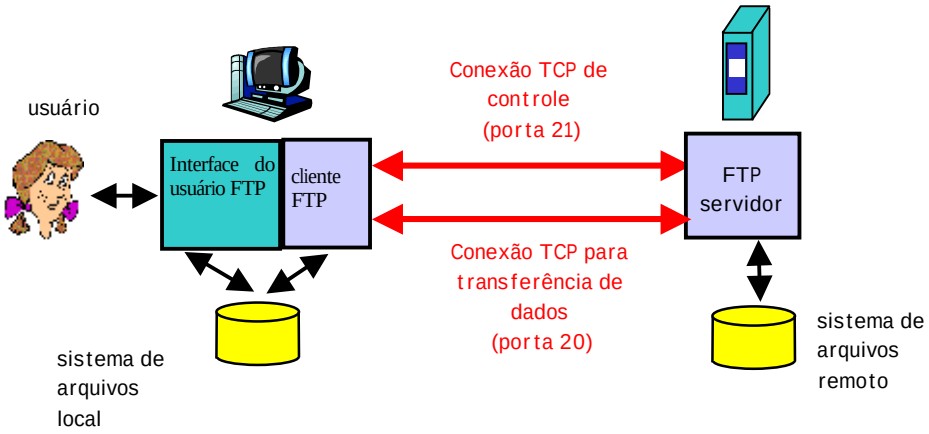


Figura 2.3. Aplicação FTP

Agentes usuário FTP

Como **agente usuário**, o usuário pode executar comandos **ftp** modo texto diretamente em uma janela Unix ou no Prompt de Comandos do Microsoft MS-DOS. Também há disponível no mercado aplicativos especializados, como por exemplo, o WS-FTP (gratuito para uso acadêmico e disponível para *download* no endereço `ftp.ipswitch.com`).

Protocolo FTP

O **protocolo FTP**, assim como o HTTP, roda sobre o **TCP**. A diferença é que o FTP usa duas conexões paralelas TCP para transferir arquivos: uma para **controle da conexão** e outra para a **transferência de dados**. O controle de conexão é usado para trocar informações como a identificação do usuário e senha, e para transferir os comandos FTP para mudar de diretório (`cd`), solicitar arquivos (`get`) ou enviar arquivos (`put`). A conexão de dados é usada para transferir os arquivos. Cada uma destas duas conexões TCP usa uma porta específica: a conexão de controle de conexão usa a porta 21 e a conexão de dados usa a porta 20 (Figura 2.3).

Comandos do usuário e mensagens do protocolo FTP

A cada comando do usuário, o protocolo FTP envia mensagens do cliente ao servidor e vice-versa. As mensagens são enviadas através da conexão TCP de controle, em formato ASCII, com quatro caracteres maiúsculos. Veja alguns exemplos de comandos digitados pelo usuário e as mensagens do protocolo FTP enviadas do servidor ao cliente:

Ação executada pelo usuário	Comando FTP digitado pelo usuário	Mensagem enviada pelo protocolo cliente FTP ao servidor
Autenticação	O usuário digita seu nome e senha	`USER nome_do_usuário` `PASS senha`
O usuário lista o conteúdo de um diretório	`dir`	`LIST`
O usuário solicita um arquivo	`get nome_arquivo`	`RETR nome_arquivo`
O usuário envia um arquivo ao servidor	`put nome_arquivo`	`STOR nome_arquivo`

Cada comando do cliente é seguido por uma resposta do servidor. As respostas são sempre de três dígitos, com uma mensagem opcional seguindo o número. Veja algumas respostas típicas:

```
331 User name OK, password required
125 Data conection already open; transfer starting
425 Can´t open data conection
452 Error writing file.
```

Correio eletrônico

O **correio eletrônico**, ou ***e-mail*** (*electronic mail*), é uma das aplicações mais populares da Internet. É uma aplicação assíncrona, onde os usuários enviam e lêem suas mensagens quando acharem conveniente. As mensagens modernas incluem *hyperlinks* HTML, texto formatado, imagens, sons e até vídeo.

Numa visão geral, o correio eletrônico possui três grandes componentes: os **agentes usuários**, os **servidores de *e-mail*** e o **protocolo SMTP** (*simple mail transfer protocol*) (Figura 2.4).

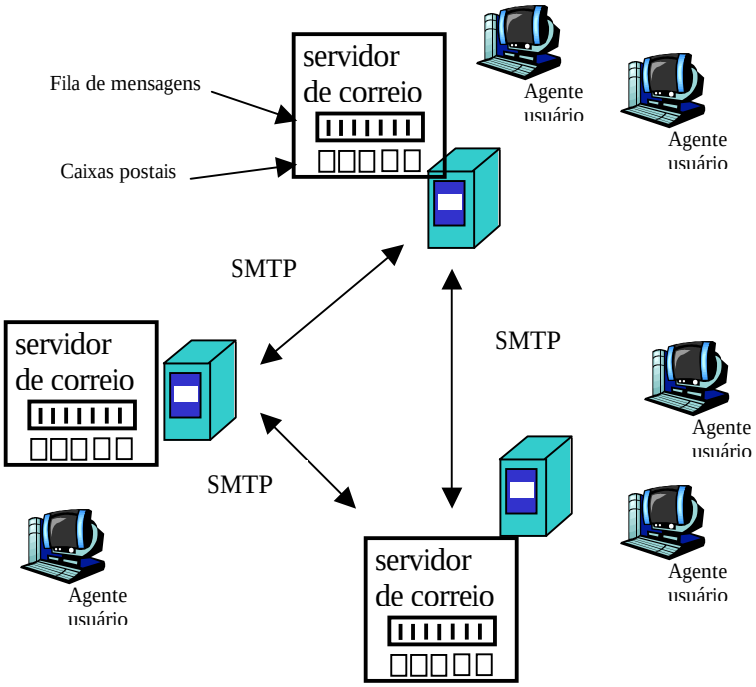


Figura 2.4. Componentes do Correio

Leitores de *e-mail*

Os **agentes usuário**, muitas vezes chamados de **leitores de *e-mail***, permitem aos usuários lerem (*read*), responderem (*reply*) ou encaminharem (*forward*) a outra pessoa uma mensagem recebida, bem como comporem (*compose*) e enviar (*send*) uma nova mensagem. Os modernos leitores de *e-mail* apresentam **interface gráfica**, como por exemplo, o Eudora, Microsoft Outlook ou Netscape Messenger; entretanto, muitos ainda utilizam leitores de *e-mail* em **modo texto**, como o mail, pine e elm. Mais recentemente, também tem sido bastante difundida a leitura de *e-mail* diretamente com os navegadores *Web*, os quais acessam servidores conhecidos como de ***web-mail***.

Servidores de *e-mail*

Os **servidores de *e-mail*** são os componentes centrais da infraestrutura do correio eletrônico. Para enviar uma mensagem à caixa postal de uma pessoa, uma vez que o remetente digitou mensagem, seu agente usuário a envia ao seu servidor de *e-mail*, que coloca a mensagem em uma **fila de saída**.

Através do **protocolo SMTP**, o servidor de *e-mail* envia as mensagens que estão na sua fila de saída em direção ao servidor destino. Caso o servidor destino não esteja acessível, o servidor de *e-mail* tentará enviá-la novamente a cada 30 minutos, persistindo nestas tentativas por alguns dias, quando então remove a mensagem e notifica quem a tinha enviado.

Cada servidor de *e-mail* tem também um conjunto de **caixas postais** (*mailbox*) para cada um de seus usuários cadastrados. Uma vez que uma mensagem chegou ao servidor de *e-mail* destino, a mesma é armazenada na caixa postal do respectivo usuário.

Para ler uma mensagem em sua caixa postal o destinatário da mensagem deve, a partir de seu **leitor de *e-mail,*** requisitá-la de seu servidor. O servidor de *e-mail* então requisita uma autenticação do usuário, através de uma **identificação** e uma **senha**, para depois repassar as mensagens que porventura chegaram a esta pessoa.

Protocolo SMTP

O **protocolo SMTP** é o protocolo de aplicação mais importante para o funcionamento do correio eletrônico. Ele usa o serviço de transferência de dados confiável do **TCP** para transferir uma mensagem desde o remetente até a caixa postal do destinatário. O SMTP, como outros protocolos de aplicação, tem dois lados, o lado cliente e o lado servidor. Quem envia a mensagem faz o papel do cliente e quem recebe de servidor, todavia, ambos os lados do SMTP devem ser implementados em cada servidor de *e-mail.*

As mensagens trocadas pelo protocolo SMTP são mensagens em caracteres ASCII. Para enviar uma mensagem, o **cliente SMTP** estabelece uma **conexão TCP**, na **porta 25**, com o **servidor SMTP**. Uma vez estabelecida à conexão TCP, cliente e servidores de *e-mail* entram em uma fase de apresentação mútua (*handshaking*), trocando algumas informações (como, o cliente indica o endereço de *e-mail* do emissor e do destinatário), antes de enviarem a mensagem eletrônica em si.

Veja um exemplo de uma seqüência de mensagens SMTP trocadas entre um cliente (C) e um servidor (S). O nome do cliente é `smtp.das.ufsc.br` e o nome do servidor é `mail.sj.cefetsc.edu.br`. Cada linha do exemplo corresponde exatamente aos textos ASCII trocados depois de aberto o canal TCP. A abertura do canal TCP pode ser feita a partir de um Telnet no servidor de *email* (porta 25), por exemplo:

> telnet `mail.sj.cefetsc.edu.br` 25

```
S:    220 hendrix.sj.cefetsc.edu.br ESMTP Postfix
C:    HELO smtp.das.ufsc.br
S:    250 hendrix.sj.cefetsc.edu.br
C:    MAIL FROM: cantu@das.ufsc.br
S:    250 OK
C:    RCPT TO: cantu@sj.cefetsc.edu.br
S:    250 OK
C:    DATA
S:    354 End data with <CR> <LF> . <CR> <LF>
C:    Ola Evandro,
C:    Este eh um teste de troca de mensagens SMTP de modo manual.
C:    Ele serve para enriquecer nossas aulas de laboratorio.
C:    .
S:    OK: queued as ...
C:    QUIT
S:    221 bye
```

```
> Connection closed by foreign host.
```

No exemplo acima o cliente envia uma mensagem do servidor `smtp.das.ufsc.br` ao servidor `sj.cefetsc.edu.br`. Os comandos usados pelos cliente foram: HELO (olá) , MAIL FROM (de), RCPT TO (para), DATA (dados) e QUIT (fim). O servidor responde cada comando usando um código (a mensagem explanatória em inglês é opcional).

Exercício

Envie você também um *email* manualmente a um colega trocando mensagens SMTP diretamente com um servidor. Para tal, voce deverá fazer primeiramente um Telnet (acesso remoto), na porta 25

(porta do SMTP), de um servidor de *e-mail*. Uma vez estabelecida à conexão TCP, você poderá trocar comandos SMTP.

Protocolo para leitura de *e-mail* POP3

Uma vez enviada uma mensagem eletrônica, ela é colocada na caixa postal do destinatário. Uma maneira natural para o destinatário de ler as mensagens de sua caixa postal, seria acessar diretamente o seu servidor de *e-mail*. Isto é na verdade o que se faz quando se acessa remotamente o servidor (por exemplo, através de um Telnet) e utiliza um agente usuário como o pine. Os novos *web-mail* também fazem isto, acessando as caixas postais diretamente no servidor de *e-mail*.

No caso do usuário destino utilizar um **leitor de *e-mail*** diretamente em seu computador pessoal (como o Eurora ou Outlook), vai haver a necessidade de transferir as mensagens do usuário do seu servidor de *e-mail* para seu computador. Para realizar esta tarefa, normalmente utiliza-se um protocolo de acesso para *e-mail* extremamente simples, o **protocolo POP3**. O POP3 inicia quando o agente usuário (cliente) abre uma conexão **TCP** com o servidor de *e-mail*, na **porta 110**. Com a conexão TCP estabelecida, o POP3 processa três fases: autorização (quando o usuário envia seu nome e senha e recebe suas mensagens), transação (quando o usuário requisita ações sobre as mensagens, como por exemplo marcando algumas para serem apagadas) e atualização (quando o usuário encerra a sessão e o servidor apaga as mensagens marcadas para serem removidas).

Outro protocolo com a mesma função do POP3 é o protocolo **IMAP** (*Interactive Mail Acess Protocol*).

Exercício

Se você usa seu computador pessoal para *e-mail*, verifique a configuração do seu aplicativo, anotando o endereço do servidor para que você possa enviar mensagens, chamado de **endereço SMTP**, e o endereço do servidor onde você vai verificar suas mensagens ainda não lidas, chamado de **endereço POP3**.

Questões

1. O que é um **protocolo de aplicação** e qual sua relação com as aplicações?
2. Mostre para pelo menos três aplicações, quem faz o papel do **cliente** e quem faz o papel de **servidor**.
3. Como as aplicações se comunicam através da rede? Que tipo de mecanismo é utilizado nesta comunicação?
4. Explique o mecanismo de **endereçamento** utilizado pelas aplicações Internet.
5. O que é um **agente usuário**? Cite exemplo de agentes usuário para pelo menos cinco aplicações.
6. Que tipo de **serviços** as **aplicações** requerem dos protocolos da camada transporte da Internet?
29. Que tipo de **serviços** os protocolos da **camada transporte** da Internet, TCP e UDP, oferecem às aplicações? Cite algumas características de cada um dos serviços.
7. Porque o HTTP, o FTP e o SMTP rodam sobre o TCP e não sobre o UDP?
8. Quais os componentes de um **endereço URL**, utilizado na aplicação WWW?

9. Explique qual o papel e como funciona o protocolo **HTTP**.
10. Pesquise na Internet sobre a história da aplicação WWW, levantando as características desta aplicação que a tornaram uma das mais populares da Internet.
11. Pesquise como o protocolo HTTP pode prover mecanismos de autenticação de usuários para acesso às informações na Web. Ache uma URL que discuta esta questão.
12. Pesquise e descreva como funcionam os servidores de ***cache* Web** (também conhecidos como **servidores *proxy***). Ache uma URL que discuta esta questão.
13. Faça uma descrição das características de dois **navegador Web** (Internet Explorer e Netscape, por exemplo), mostrando as diferenças entre ambos.
14. Faça um Telnet (ou SSH) em um servidor Unix e utilize o navegador em modo texto **lynx**. Descreva suas características e utilidade.
15. Qual a utilidade da aplicação de **transferência de arquivos** (FTP). Descreva os procedimentos para transferir um arquivo de um servidor até uma estação cliente, utilizando como **agente usuário** comandos **ftp** modo texto.
16. Qual as principais diferenças entre o protocolo de aplicação HTTP e o protocolo FTP?
17. Explique como funciona o **correio eletrônico**, descrevendo a função dos principais componentes desta aplicação.
18. O que é um **leitor de *e-mail***? Cite exemplo de produtos comerciais.
19. Qual a diferença entre o protocolo **SMTP** e o protocolo **POP3**? Onde cada um é utilizado?
20. Descreva as diversas partes que compõe uma **mensagem eletrônica**. Mostre através de um exemplo.
21. Faça um Telnet (ou SSH) em um servidor Unix e utilize o **agente usuário** de correio eletrônico em modo texto **mail**. Descreva suas características e utilidade.
22. O que é **ICQ**? Pesquise sobre que protocolos esta aplicação utiliza.

Parte 3

Protocolos Internet TCP/IP

Os protocolos da Internet TCP/IP foram primeiramente apresentados a mais de 15 anos, muito tempo considerando a era da informação; todavia, muitos de seus princípios fundamentais continuam atuais, e mais, com a grande difusão da Internet, estes protocolos formam hoje a tecnologia hegemônica das redes de computadores.

Arquitetura da Internet TCP/IP

O conjunto de **protocolos TCP/IP** (*Transmission Control Protocol/Internet Protocol*) é um padrão industrial de protocolos destinados a redes geograficamente distribuídas, ou WANs (*wide area networks*), sendo as principais peças da **arquitetura Internet**.

A **arquitetura Internet** objetiva a interligação de **computadores**, não importando em qual tipo de rede os mesmos estejam conectados, a qualquer outro computador da rede mundial de computadores. Para interligar redes distintas a arquitetura Internet usa uma máquina como ponto de ligação entre as redes, sendo está máquina conhecida como **roteador** (ou *gateway)*. Os roteadores são os responsáveis pelo roteamento das mensagens na malha que forma a Internet. (Figura 3.1)

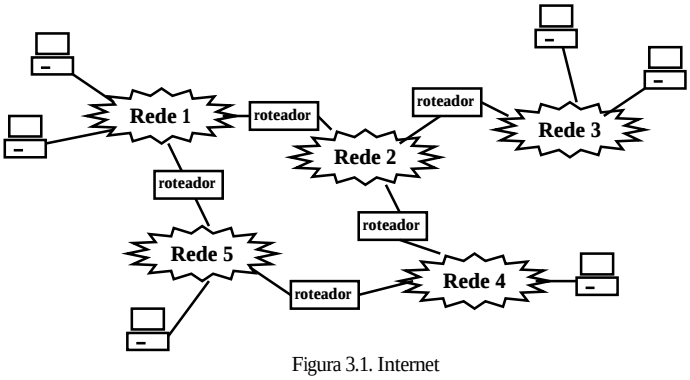


Figura 3.1. Internet

Os protocolos da arquitetura Internet TCP/IP estão organizados em quatro camadas: a **camada de aplicação**, a **camada de transporte**, a **camada de rede,** interligando as inter-redes, e a **camada enlace/física**, inferior, representando os protocolos de enlace e a rede física (Figura 3.2).

Na **camada de transporte** a arquitetura baseia-se principalmente em um serviço de transporte orientado a conexão, fornecido pelo protocolo **TCP** (*Transmission Control Protocol*). Todavia, um serviço de datagrama, não orientado a conexão, também é disponível com o protocolo **UDP** (*User Datagram Protocol*). Na **camada de rede**, temos um serviço não-orientado a conexão, fornecido pelo protocolo **IP** (*Internet Protocol*).

Figura 3.2. Pilha de protocolos da Internet

Camada de Transporte

Situada entre a camada de aplicação e a camada de rede, a **camada de transporte** tem a função de prover um **canal de comunicação lógico fim-a-fim** entre os processos de aplicação rodando em diferentes computadores, sem se preocupar com os detalhes da infra-estrutura física usada para carregar as mensagens entre eles.

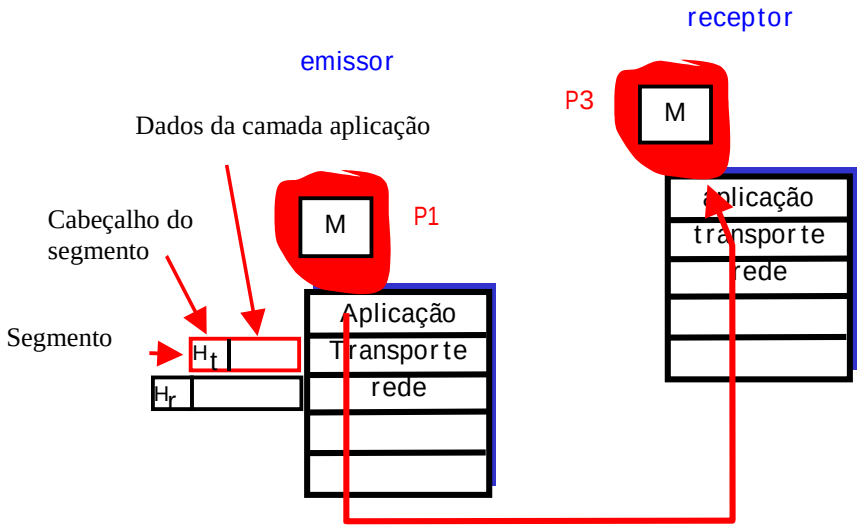
Os protocolos de transporte são implementados nos sistemas terminais, já que oferecem um canal lógico fim-a-fim às aplicações, não necessitando, no entando, serem implementados nos roteadores da rede, os quais atuam somente até a camada rede.

No lado do emissor, as **mensagens** redebidas das aplicações são fragmentadas e encapsuladas em **unidades de dados de protocolos,** ou **PDUs** (*protocol data unit*), chamadas **segmentos,** aos quais adiciona-se um cabeçalho (Figura 3.3). Cada segmento é então repassado a camada rede que por sua vez encapsula em **unidades de dados de protocolos** da camada de rede, ou **datagramas**.

Relação entre a camada de transporte e a camada de rede

Na Internet o protocolo da **camada rede** é chamado **IP** e fornece um serviço de comunicação de **computador-a-computador** na inter-rede. O modelo de serviço do **protocolo IP** é do tipo **"melhor esforço"** (*best effort*), isto é, ele faz o melhor esforço para o envio de um datagrama entre computadores, mas não dá nenhuma garantia. Em particular, não garante a entrega do datagrama, não garante que sejam entregues em ordem e nem garante a integridade dos dados. É por isso chamado de **serviço não garantido**.

Os **protocolos de transporte TCP e UDP** estendem a entrega computador-a-computador, fornecida pelo IP, pela entrega **processo-a-processo**, que é chamada de **multiplexação/demultiplexação de aplicações**. O TCP e o UDP oferecem também checagem da integridade dos dados, incluindo campos para **detecção de erros** no seu cabeçalho. No caso do **UDP**, a **multiplexação/demultiplexação de aplicações** e a **checagem de erros** nos dados, são os dois únicos serviços oferecidos, sendo portanto um **serviço não garantido.** O **TCP**, além destes dois, oferece ainda a **transferência garantida**, usando controle de fluxo, números de sequência, reconhecimentos e temporizadores.



O serviço de multiplexação e demultiplexação de aplicações

O protocolo IP entrega dados entre dois sistemas terminais (*hosts*), cada qual identificado por seu **endereço IP**. A responsabilidade dos protocolos de transporte é entregar estes dados (segmentos) a aplicação apropriada rodando em cada *host*.

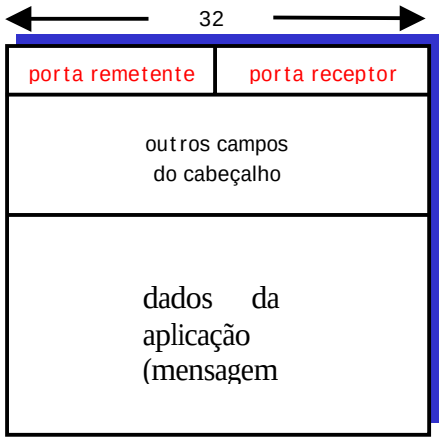


Figura 3.4. Segmento da camada transporte

Cada um dos **segmentos** da camada transporte tem em seu cabeçalho um campo que indica a qual processo o mesmo deve ser entregue. Estes campos são conhecidos como **números de porta**. O cabeçalho inclui um campo com o número de **porta do emissor** e o número de **porta do receptor**. (Figura 3.4)

Os números de porta variam de 0 a 65535, sendo que até a porta 1023 são números reservados para aplicações específicas. Por exemplo:

HTTP – porta 80

SMNP – porta 25

TELNET – porta 23

SSH – porta 22

FTP – porta 21.

Considere como exemplo a aplicação **Telnet**, que utiliza a **porta 23**. Quando um **cliente Telnet** inicia uma seção, ele envia ao **servidor Telnet** um segmento TCP destinado à porta 23 (porta reservada para a aplicação Telnet) e coloca como número de porta da fonte uma porta que não esteja sendo utilizada por nenhum processo no *host* cliente, por exemplo, porta **X**. Quando o servidor recebe o segmento, ele verifica que o mesmo é endereçado a porta 23 e então sabe que se

trata da aplicação Telnet e a porta da fonte **X** vai identificar um processo Telnet específico (já que

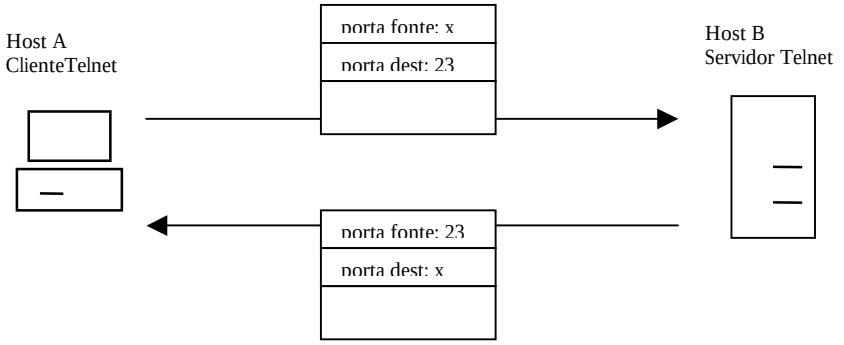


Figura 3.5. Uso de portas para acessar servidor de aplicação Telnet

pode haver outras solicitações). (Figura 3.5).

UDP (*User Datagram Protocol*)

O **protocolo UDP** adiciona ao IP a **multiplexação e demultiplexação** de aplicações e o mecanismo de **detecção de erros**.

No UDP não há processo de abertura de conexão para o envio de dados, por isto é chamado de **protocolo sem conexão** (*connectionless*).

Características:

- Sem estabelecimento de conexão, não introduzindo, portanto, atrasos para esta tarefa.
- Não mantém estado da conexão, que implicaria em *buffers* (memórias) de envio e recepção, números de seqüência e reconhecimento.
- Tem pequeno *overhead* (informações de controle) no cabeçalho.
- Taxa de envio sem regulação ou controle de fluxo.

Por estas características é apropriado para aplicações tempo real, como telefonia e transferência de áudio e vídeo sobre a Internet.

O formato do "**segmento" UDP** (alguns autores chamam de *datagrama UDP*, pois pouco acrescenta ao *datagrama IP*) é bastante simples (Figura 3.6), além dos campos reservados para as **portas de origem e destino**, há um campo que indica o **comprimento do segmento** e o ***checksum***, utilizado para o reconhecimento de erros no segmento. O campo de **dados da aplicação** é preenchido com os dados da aplicação, por exemplo, para a aplicação DNS os dados podem ser mensagens de consulta e resposta, para aplicações de aúdio tempo real, o campo é preenchido com amostras de aúdio.

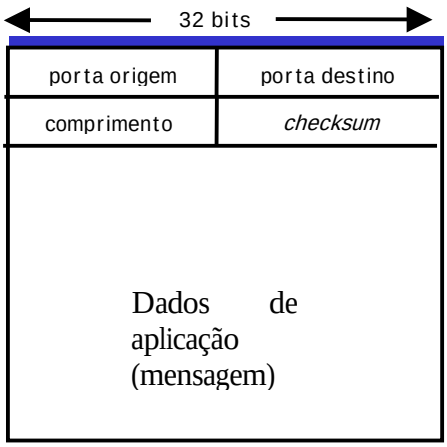


Figura 3.6. Formato do segmento UDP

Checksum

O ***checksum*** do UDP permite a **detecção de erros** nos dados transmitidos. Para isto, o **emissor UDP** faz o **complemento 1** da **soma** de todos as palavras de 16 bits do segmento e coloca o resultado no campo ***cheksum***. Por exemplo, suponha que temos três palavras de 16 bits:

0110011001100110
0101010101010101
0000111100001111

A soma será

0110011001100110
0101010101010101
\+ _____
1011101110111011

Adicionando a terceira palavra a esta soma

1011101110111011
0000111100001111
\+ _____
1100101011001010

O **complemento 1** é obtido invertendo cada bit 1 por 0 e vice-versa. Desta forma o complemento da soma será 0011010100110101, o qual será o *cheksum*. No lado do **receptor UDP**, todas as palavras de 16 bits recebidas são adicionadas, incluindo o *cheksum*.Se não houve erros na transmissão, a soma será 1111111111111111. Se um dos bits for 0, então é sabido que houve erros.

TCP (*Transmission Control Protocol*)

O **protocolo TCP,** como o UDP, também oferece a **multiplexação/demultiplexação** de aplicações e o mecanismo de **detecção de erros**. A grande diferença é que o TCP é um **protocolo orientado a conexão** e com **transferência garantida,** onde os dois processos devem acordar entre eles uma abertura de conexão para que os dados possam ser transferidos. Além destas características, o TCP integra ainda um serviço de **controle de fluxo,** que assegura que nenhum dos lados da comunicação envie pacotes rápido demais, pois uma aplicação em um lado pode não conseguir processar a informação na velocidade que está recebendo, e um serviço de **controle de congestão** ajuda a prevenir congestionamentos na rede.

Uma conexão TCP é uma conexão *full-duplex* (isto é, em ambos os sentidos e simultânea) e é sempre **fim-a-fim**, entre o *host* emissor e o *host* receptor. Uma vez estabelecida à conexão os dois processos podem trocar informações. O processo cliente, no lado **emissor,** passa o bloco de dados através da **porta** apropriada. O TCP então manipula estes dados, dirigindo para o ***buffer* de envio**. Os dados são então fragmentados e encapsulados na forma de **segmentos**. Os segmentos, por sua vez, são passados a camada rede onde eles são separadamente encapsulados em **datagramas IP**, que são enviados através da rede. Quando o TCP do **receptor** recebe os dados, os mesmos são recebidos no ***buffer* de recepção**. A aplicação no lado do receptor então lê os dados a partir deste *buffer*.

Transferência garantida: analogia com a compra de uma enciclopédia

Voltando a analogia com o sistema postal, pode-se dizer que o serviço de entrega de correspondências entre usuários é um serviço tipo **melhor esforço** (*best effort*), isto é, ele faz o melhor esforço para o envio de uma carta entre usuários, mas não dá nenhuma garantia. Em particular, não garante a entrega da carta, pois a mesma pode se perder e não há formas de avisar o emissor sobre o ocorrido. Da mesma forma, não há um serviço de confirmação de recebimento do receptor ao emissor. É por isso que pode ser chamado de **serviço não garantido**.

Este serviço é análogo ao serviço oferecido pelo protocolo da camada rede da Internet, o IP. Na Internet o **serviço garantido** é implementado pelo TCP, e roda sobre o serviço não garantido fornecido pelo IP, utilizando números de sequência, reconhecimentos e temporizadores.

Vamos comparar o serviço garantido fornecido pelo TCP utilizando uma analogia com o que acontece na compra de uma enciclopédia em fascículos.

Suponha que você resolva adquirir uma enciclopédia, cujos volumes são vendidos em fascículos que são entregues pelo correio. Imagine que a coleção completa tenha 100 fascículos sendo eles enviados um a cada semana.

Quando você resolve fazer a compra, você envia uma carta a editora responsável pela venda da enciclopédia com seu pedido. A editora então faz a **abertura de um cadastro** de cliente para você e na semana seguinte lhe envia a confirmação do seu cadastro, juntamente com o primeiro fascículo e os procedimentos para confirmação de recebimento e pagamento.

Suponha que a cada cinco fascículos recebidos, você deve enviar uma correspondência de **confirmação de recebimento**, juntamente com a parcela de pagamento correspondente.

Como a entrega dos fascículos usa o **serviço não garantido** dos correios, os mesmos podem ser perdidos ou mesmo danificados no transporte. Caso isto ocorra, você deverá enviar a editora uma carta de aviso informando o número fascículo não chegou ou que chegou danificado. A editora então fará o **reenvio** do fascículo com problemas.

As trocas de mensagens entre o comprador e a editora continuam até que o total de 100 fascículos sejam entregues e a última confirmação e o respectivo pagamento seja efetuado. Neste momento, a editora **encerrará o cadastro** do cliente e você poderá usufruir da enciclopédia completa.

Voltando aos protocolos da Internet, para implementar o serviço garantido no TCP, ocorrem procedimentos similares aos efetuados entre o cliente e o vendedor da enciclopédia. No caso do TCP, primeiro há uma fase chamada de **abertura de conexão**, onde se estabelece os parâmetros para a comunicação, como inicalização de variáveis e *buffers*. Em seguida, inicia-se a **troca de dados**, onde cada pacote de informação trocado entre o emissor e o receptor tem um **número de sequência**, o qual vai ser tomado como base para o receptor **reconhecer** o recebimento. Caso o reconhecimento não seja confirmado dentro de um tempo limite, o emissor **retransmite** o pacote.

Protocolo com Transmissão Garantida

Para garantir uma entrega de dados livre de erros, os protocolos com transmissão garantida, como o TCP, utilizam uma técnica conhecida como **confirmação positiva com retransmissão**. A técnica exige que um receptor comunique-se com a origem, retornando uma mensagem de **reconhecimento** (*acknoledge*), a medida que recebe os dados. O transmissor, por sua vez, inicia um temporizador para cada pacote que envia, e **retransmite** o pacote se este temporizador se complete antes que chegue uma confirmação de recebimento.

A figura 3.7 mostra um exemplo de confirmação positiva.

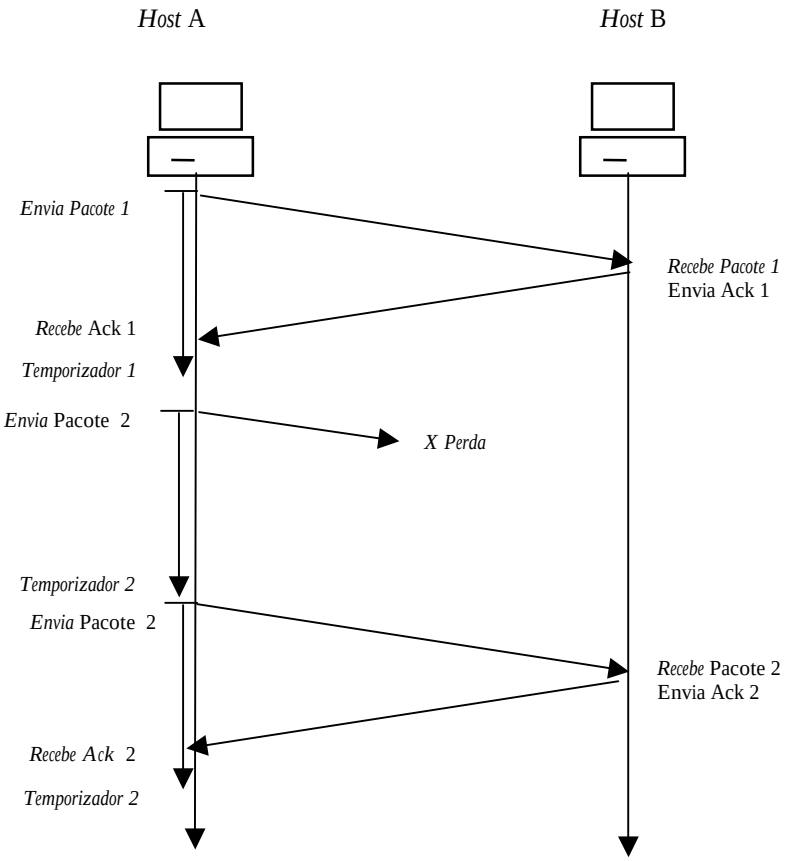


Figura 3.7. Protocolo de confirmação positiva.

O problema de um protocolo como o da figura 3.7 é que o emissor deve esperar o reconhecimento de cada pacote antes que um novo pacote possa ser enviado, o que torna a transmissão bastante ineficiente. Protocolos mais elaborados, como o TCP, permitem que o emissor transmita múltiplos pacotes antes de esperar uma confirmação. No TCP isto é implementado através de um mecanismo conhecido como **janelas deslizantes**.

No mecanismo de janelas deslizantes, mostrado na figura 3.8, o emissor pode enviar uma sequência de pacotes, contidos dentro de uma "janela" de tamanho fixo, antes de esperar uma confirmação.

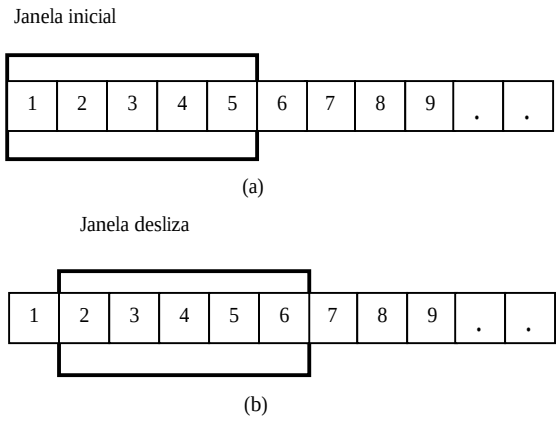


Figura 3.8 Mecanismo de janelas deslizantes

Na figura 3.8 (a) , os pacotes contidos dentro da janela (numerados de 1 a 5) podem ser enviados em sequência. Quando o transmissor redece a confirmação do primeiro pacote da janela, a janela "desliza", figura 3.8 (b), permitindo que um novo pacote seja enviado.

A figura 3.9 mostra uma sequência de três pacores sendo transmitida com o mecanismo de janela deslizante.

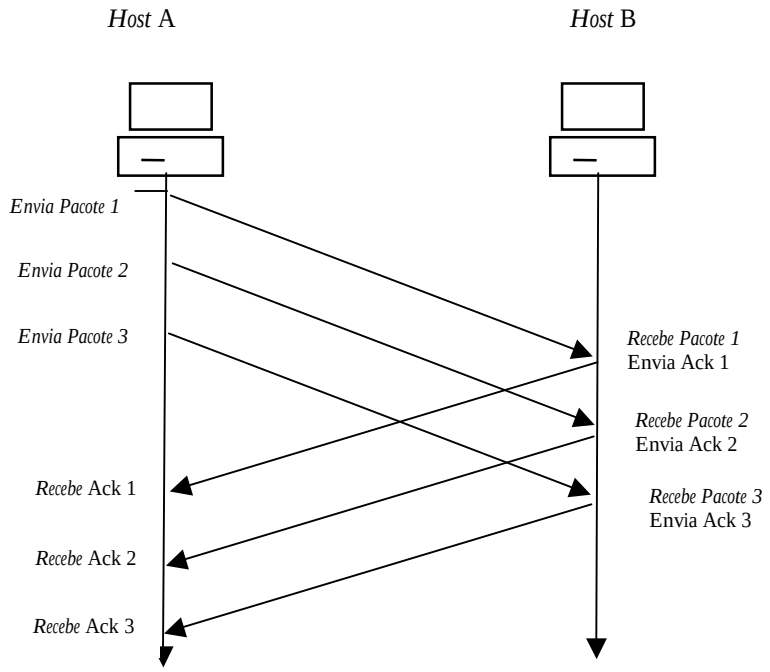


Figura 3.9. Sequência de pacotes transmitidos com janelas deslizantes.

Segmento TCP

A figura 3.7 mostra a estrutura do **segmento TCP**. No cabeçalho, além dos **números de porta** e ***checksum*** que também existem no UDP, há outros campos com informações necessárias a implementação do serviço de transferência garantida, controle de fluxo e controle de congestionamento.

O campo de **dados da aplicação** do segmento TCP (Figura 3.10), contém um fragmento ou pedaço dos dados da aplicação, cujo tamanho máximo, chamado de **MSS** (*maximum segment size*), depende da implementação do TCP. Os valores típicos são 1.500 bytes, 536 bytes e 512 bytes, não incluindo o cabeçalho. (Em geral o valor de MSS é escolhido para evitar a fragmentação do datagrama IP na camada inferior, conforme veremos a frente. Este valor em algumas implementações pode ser configurado manualmente ou estabelecido automaticamente pelo protocolo).

Outros campos fundamentais do segmento TCP são os seguintes:

- **Número de seqüência** e **reconhecimento**, utilizado para o emissor e receptor implementarem o serviço de transferência garantida.
- Tamanho da **janela do receptor**, usado para o controle de fluxo, e indica o número de bytes que o receptor é capaz de receber.
- **Tamanho do cabeçalho**, especifica o tamanho da cabeçalho, que pode variar em funções do campo de opções, todavia, tipicamente, o tamanho do cabeçalho é de 20 bytes.
- O campo de **opções** é usado quando o emissor e receptor precisam negociar o tamanho máximo de segmento (MSS).
- Os ***flags*** (bandeiras) contém 6 bits. O **Ack** é usado para indicar que o campo de reconhecimento é válido, O **Rst**, **Syn** e **Fin** são usados para abertura e encerramento de conexão, o **Psh** indica que o receptor deve passar imeidatamente o dado a camada superior e o **Urg** indica um dado urgente.

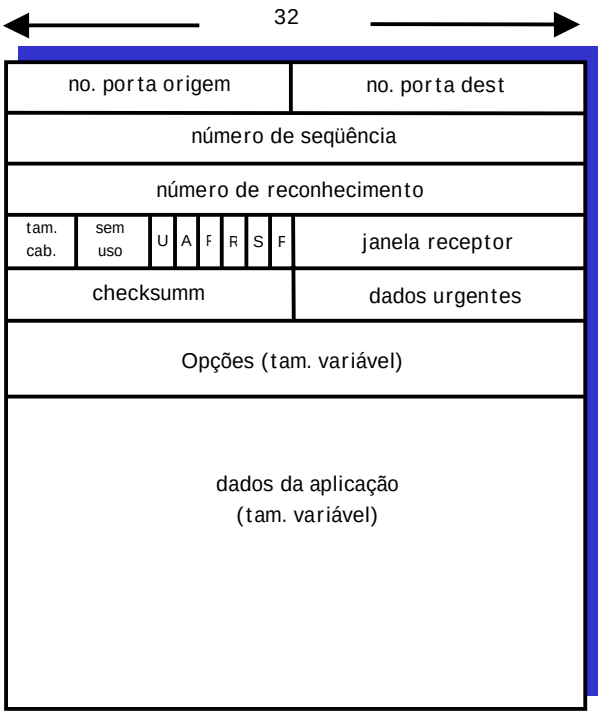


Figura 3.10. Formato do segmento TCP

Números de seqüência e reconhecimento no TCP

Dois campos importantes do segmento TCP são os **números de seqüência** e **reconhecimento**, os quais fazem a parte crítica do trabalho de **transferência de dados confiável**.

Como vimos, os **dados das aplicações** são transportados pelos segmentos TCP. Caso as mensagens forem maior que o valor de **MSS, tamanho máximo do segmento**, as mesmas são fragmentadas para poderem ser acomodadas na parte de dados do segmento. Por exemplo, um arquivo GIF de 500K bytes trocado pelo HTTP será fragmentado em vários pedaços para ser transmitido pelo TCP. Os **números de seqüência** servem, portanto, para que o lado receptor TCP possa reordenar corretamente os dados recebidos.

Os **números de seqüência** não correspondem a uma série de segmentos transmitidos, mas refletem a quantidade de bytes que o TCP está transmitindo. Por exemplo, suponha que o bloco total de dados que será transmitido tenha 500.000 bytes, que o valor de MSS é de 1.000 bytes, e que o primeiro byte dos dados é numerado como zero. Para transmitir esta quantidade de bytes o TCP formará 500 segmentos. Ao primeiro segmento atribui-se o número de seqüência zero, ao segundo 1000, ao terceiro 2000 e assim por diante. (Figura 3.11).

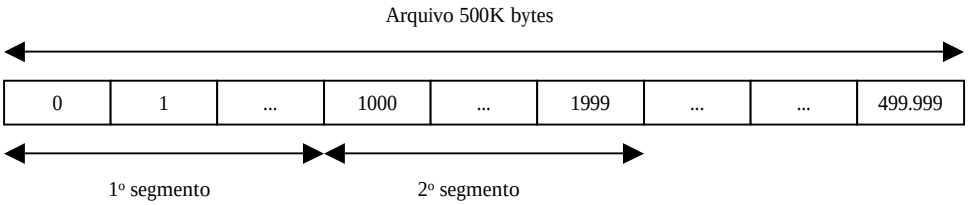


Figura 3.11. Divisão de um arquivo de dados em

Os **reconhecimentos** servem para o receptor informar o emissor quais blocos que foram recebidos corretamente. Todavia, lembre-se que uma comunicação TCP é sempre *full-duplex*, o que significa que o *host* A pode estar recebendo dados do *host* B ao mesmo tempo em que está enviando dados ao *host* B (como parte da mesma conexão TCP). Desta forma, haverá números de reconhecimentos para dados seguindo de A para B e outros para dados seguindo de B para A.

> O **número de reconhecimento** que o *host* A coloca no seu segmento é o **número de seqüência** do próximo byte que o *host* A espera receber do *host* B.

Por exemplo, suponha que o *host* A recebeu todos os bytes numerados de 0 a 535 de B e que está prestes a enviar um segmento a B. Neste caso, o *host* A coloca como número de reconhecimento 536, o que vai indicar a B que o mesmo recebeu todos os bytes até este número.

Em outro exemplo, suponha que o *host* A recebeu todos os bytes numerados de 0 a 535 de B e em seguida recebeu de B um segmento contendo bytes de 900 a 1000. Note que A não recebeu os bytes que vão de 536 a 899. Como A ainda está esperando bytes a partir de 536, ele reenvia a B um segmento com número de reconhecimento 536. Continuando este exemplo, suponha agora que A receba o segmento que faltava, com os bytes que vão de 536 a 899. Neste caso, como ele já recebeu inclusive os dados contendo os bytes de 900 a 1000, ele envia um reconhecimento com número 1001. Isto é chamado de **reconhecimento cumulativo**, que indica que recebeu todos os bytes até este número

Telnet: Caso de estudo para números de seqüência e reconhecimento

O **Telnet** é uma aplicação interativa usada para acesso remoto a sistemas e roda sobre o protocolo de transporte **TCP**.

O Telnet permite que um usuário utilize uma máquina A e estabeleça uma seção interativa em uma máquina B, como se estivesse utilizando um terminal. Quem solicita o Telnet assume o papel de cliente. Cada caractere digitado pelo usuário cliente será enviado ao computador remoto; o computador remoto então enviará uma cópia de cada caractere para ser mostrado na tela do cliente. Desta forma, cada caractere atravessa a rede duas vezes entre o tempo em que o usuário digita uma tecla e a visualização da mesma na tela.

Vamos examinar os segmentos TCP trocados durante uma seção Telnet (Figura 3.12). Suponha que o usuário tecla a letra "C". Suponha ainda que os **números de seqüência** iniciais usados pelo cliente e pelo servidor sejam 42 e 79, respectivamente. Isto indica que o primeiro byte a ser enviado pelo cliente ao servidor terá o número de seqüência 42 e o primeiro byte a ser enviado pelo servidor ao cliente terá o número de seqüência 79. Lembre também que o **número de reconhecimento** indica o número de seqüência do próximo byte esperado. Desta forma, depois de estabelecia a

conexão TCP, e antes do envio de quaisquer dados, o cliente esta esperando pelo byte 79 e o servidor está esperando pelo byte 42.

A figura 3.12 mostra três segmentos trocados entre o cliente e o servidor. O primeiro segmento é enviado pelo cliente, contendo um caractere ASCII com a letra "C" (número de seqüência 42). O segundo segmento é enviado pelo servidor ao cliente e serve para dois propósitos: provê um reconhecimento do caractere recebido (número de reconhecimento 43) e envia o caractere "C" de volta para ser apresentado na tela do cliente (número de seqüência 79). No terceiro segmento trocado, o cliente reconhece o caractere recebido (número de reconhecimento 80).

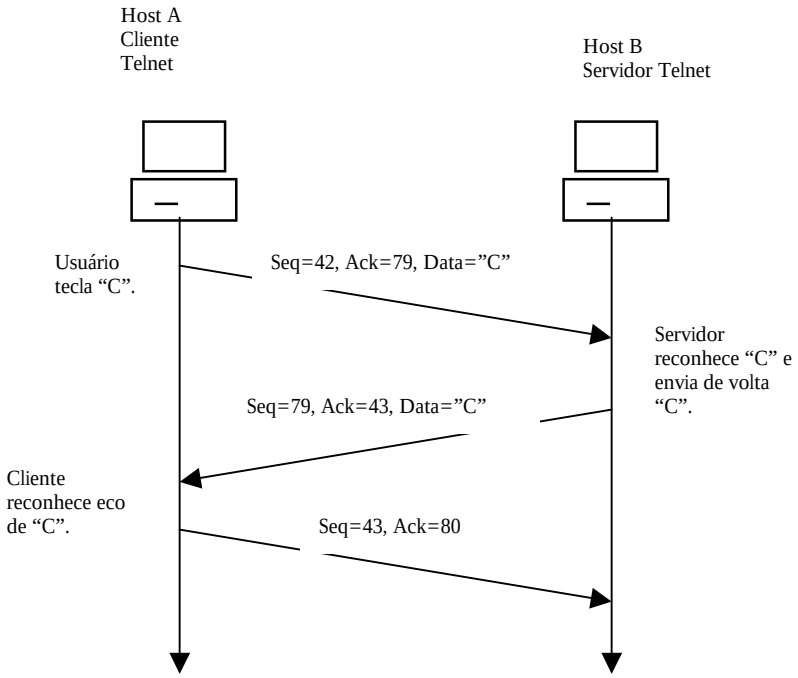


Figura 3.12. Números de seqüência e reconhecimento para a aplicação Telnet

O serviço transferência de dados garantida no TCP

Para criar o **serviço de transferência de dados garantida** o TCP manipula três grandes eventos relacionados à transmissão/retransmissão de dados.

1. Quando recebe dados da camada aplicação o TCP cria segmentos com **números de seqüência**, correspondentes aos próximos número de seqüência a serem transmitidos, e inicia um **temporizador** para cada segmento criado.
2. Caso o temporizador de um segmento enviado **estoure o tempo** (*time-out*), o TCP retransmite este segmento.
3. Caso o TCP receba um **reconhecimento** um segmento enviado (ou de um conjunto de segmentos), ele cancela os temporizadores remanescentes a estes segmentos; ou ainda, caso receba reconhecimentos de segmentos que já haviam sido reconhecidos (reconhecimentos cumulativos), ele retransmite os segmentos cujos números de seqüência são superiores ao reconhecimento cumulativo.

Vamos explicar como estes eventos que são tratados pelo TCP analisando alguns **cenários**. No primeiro cenário (Figura 3.13), o *host* A envia 8 bytes de dados ao *host* B (com número de seqüência

92). O *host* B envia reconhecimento dos 8 bytes recebidos (reconhecimento 100), o qual é perdido. Depois do estouro do temporizador do segmento 92, o mesmo é reenviado pelo *host* A. Quando o *host* B recebe o segmento duplicado, ele reenvia o reconhecimento.

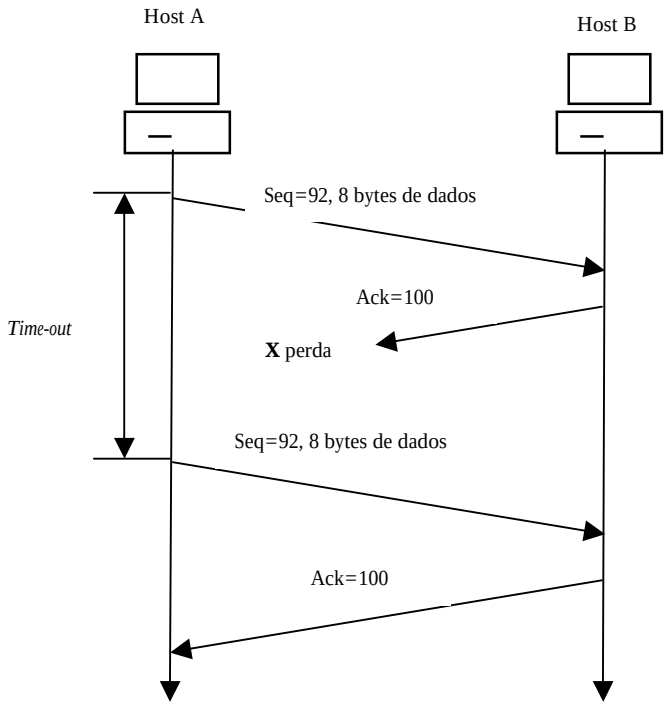


Figura 3.13. Retransmissão devido a reconhecimentos perdidos

No segundo cenário (Figura 3.14), o *host* A transmitiu ao *host* B um segmento com 8 bytes (número de seqüência 92) e em seguida mais um segmento com 20 bytes (número de seqüência 100). O *host* B recebeu estes segmentos e enviou números de reconhecimento (100 e 120 respectivamente). Todavia, o reconhecimento do segmento 92 (reconhecimento 100) chegou depois do *time-out*. Logo o *host* A retransmitiu o segmento com número de seqüência 92. Como o *host* B já havia recebido este segmento e também o seguinte (com número de seqüência 100), ele reenviou o reconhecimento cumulativo deste último segmento (reconhecimento 120).

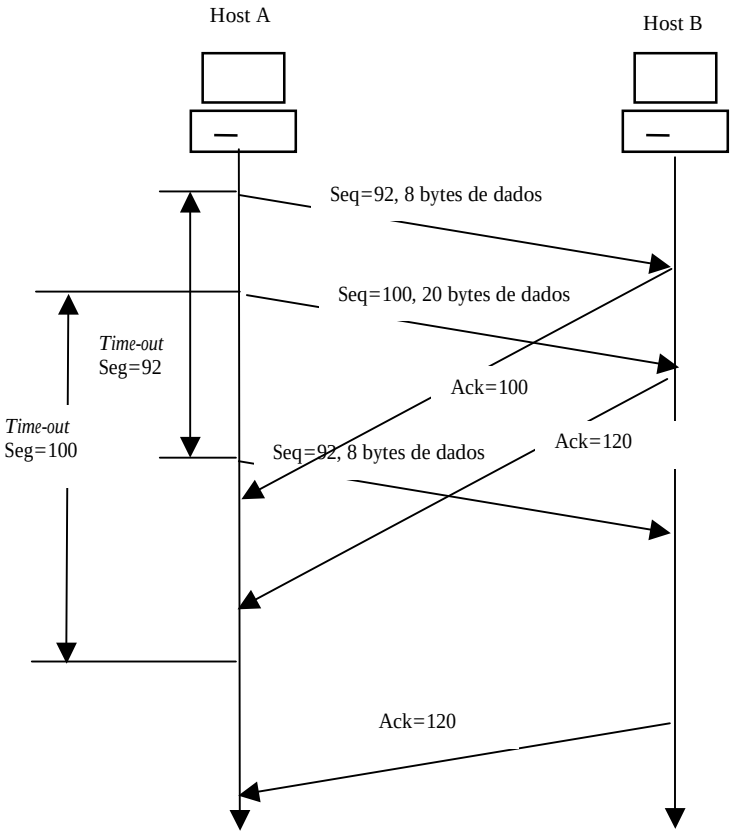


Figura 3.14. Segmento retransmitido porque seu reconhecimento chegou depois do time-out

No próximo cenário (Figura 3.15), como no caso anterior, o *host* A transmitiu ao *host* B um segmento com 8 bytes (número de seqüência 92) e em seguida mais um segmento com 20 bytes (número de seqüência 100). O *host* B recebeu estes segmentos e enviou números de reconhecimento (100 e 120 respectivamente). Todavia, o reconhecimento do segmento 92 (número de reconhecimento 100) se perdeu, o que não aconteceu com o segmento 100 (número de reconhecimento 120). Como o *host* A recebeu este último reconhecimento, ele sabe que o *host* B recebeu todos os segmentos por ele enviados, e está esperando agora segmentos com número de seqüência 120.

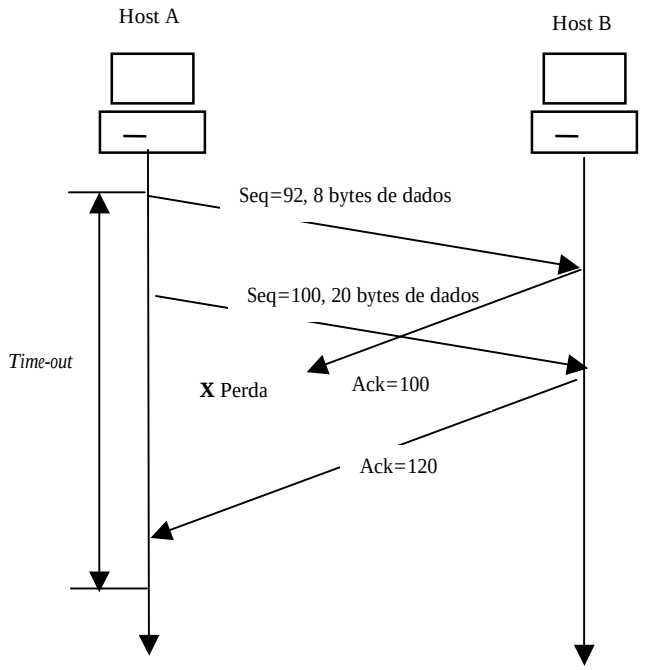


Figura 3.15. Reconhecimento cumulativo evita retransmissão do primeiro segmento

Gerenciamento de conexões no TCP

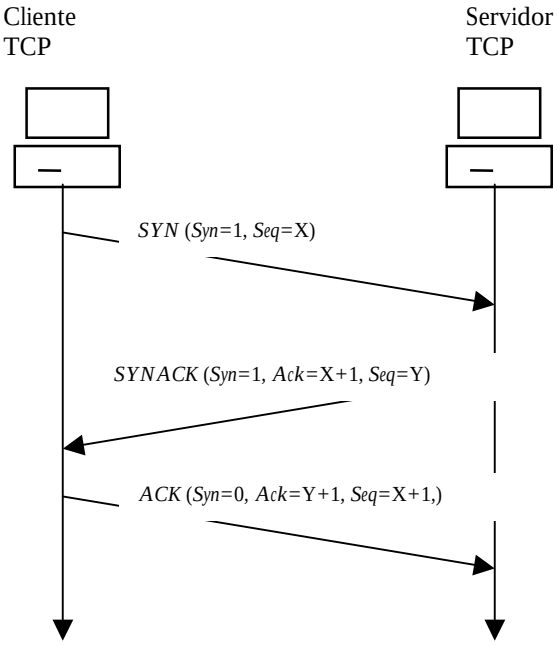
Para trocarem segmentos de dados utilizando o TCP o emissor e o receptor estabelecer uma "conexão" TCP através da troca de pacotes de controle entre si. Isto é chamado de **procedimento de estabelecimento de conexão** (*handshaking*), onde se estabelecem os parâmetros para a comunicação. Uma vez concluído o *handshaking* a conexão é dita estabelecida e os dois sistemas terminais podem trocar dados.

ABERTURA DE CONEXÃO

Na fase de estabelecimento de conexão, são inicializadas as variáveis do protocolo TCP, como os números de seqüência e o tamanho de *buffers*. O processo cliente é o que inicia o estabelecimento da conexão sendo o servidor contatado pelo cliente.

O estabelecimento da conexão se dá em três passos (Figura 3.16):

1. O lado cliente do TCP envia um segmento de sincronização, chamado SYN (com o *flag* Syn setado em 1), ao lado servidor do TCP, especificando um número inicial de seqüência.
2. O servidor recebe o SYN, aloca *buffers* e inicializa variáveis, e envia

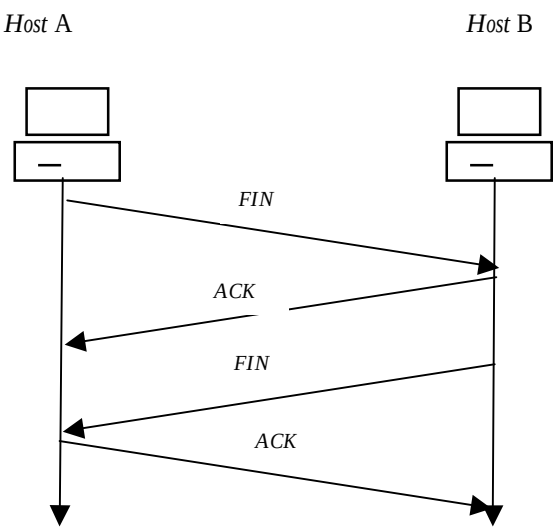


3.16. Três passos da abertura de conexão TCP

uma mensagem de aceite da conexão, chamada SYNACK (com o *flag* Syn e *flag* Ack setados em 1), onde reconhece o pedido de conexão e especifica seu número inicial de seqüência.

3. Uma vez recebido o aceite da conexão pelo servidor, o cliente confirma o recebimento com um segmento chamado ACK (*flag* Syn agora em 0 e *flag* Ack setado em 1 indicando um reconhecimento válido) e também aloca *buffers* e inicializa variáveis da conexão.

Uma vez que os três passos do estabelecimento da conexão forem completados, os *hosts* cliente e servidor podem trocar segmentos contendo dados entre eles.



3.17. Encerramento de conexão TCP

ENCERRAMENTO DE CONEXÃO

Para o enceramento da conexão quatro segmentos são trocados (Figura 3.17). Quem inicia a desconexão envia de um segmento especial, chamado FIN (com *flag* Fin setado em 1). Quem recebe o segmento solicitando o fim da conexão, primeiro reconhece o segmento recebido e depois envia ele também um segmento FIN. O encerramento definitivo da conexão se dá quando o que iniciou a desconexão recebe e reconhece o segundo segmento FIN.

Camada Rede

A **camada de transporte** provê um canal lógico **processo-a-processo** para as aplicações rodando em diferentes *hosts*. Para prover este serviço, a camada de transporte usa a **camada rede**, a qual provê um serviço de comunicação de **computador-a-computador** na inter-rede.

Papéis da camada rede:

- **Determinação da rota** que tomarão os datagramas desde o computador origem até o destino, a partir de **algorítmos de roteamento**.
- **Chaveamento de datagramas** chegando nos enlaces de entrada de cada roteador para a saída apropriada.

Protocolo IP (*Internet protocol*)

Na Internet a **camada rede** é implementada pelo **protocolo IP**, o qual oferece um **serviço de datagramas,** onde cada datagrama é tratado como uma unidade independente e não recebe nenhum tratamento de erros ou reconhecimento fim a fim. O datagrama permanece inalterado enquanto passa da origem ao destino.

Quando a camada de rede do lado de um emissor recebe um **segmento** da camada de transporte ela o encapsula em um **datagrama IP,** escreve o **endereço** do destino e outros campos do cabeçalho e envia ao primeiro roteador em direção ao *host* destino. Para que o datagrama atinga o destino, a camada rede envolve cada ***host*** e cada **roteador** no caminho entre a origem e o destino dos segmentos.

Os três principais componentes da camada rede da Internet são:

- **Protocolo IP**, que provê uma forma de **endereçamento**, **formato do datagrama** e convenções de empacotamento.
- **Protocolos de roteamento,** que permitem a determinação de rotas e elaboração de **tabelas de roteamento**. Os protocolos de roteamento mais conhecidos são o **RIP, OSPF** e **BGP**.
- **Protocolo ICMP,** utilizado para reportagem de erros e sinalização entre os roteadores.

Datagrama IP

Um **datagrama IP** é a unidade básica de transferência na Internet. O formato do datagrama apresenta um cabeçalho, que contém os **endereços IP da fonte** e do **destino**, além de outros campos, e uma área de **dados**. (Figura 3.18)

- O campo **versão** indica a versão do protocolo.
- O **comprimento do cabeçalho**, indica o comprimento do cabeçalho, em função dos

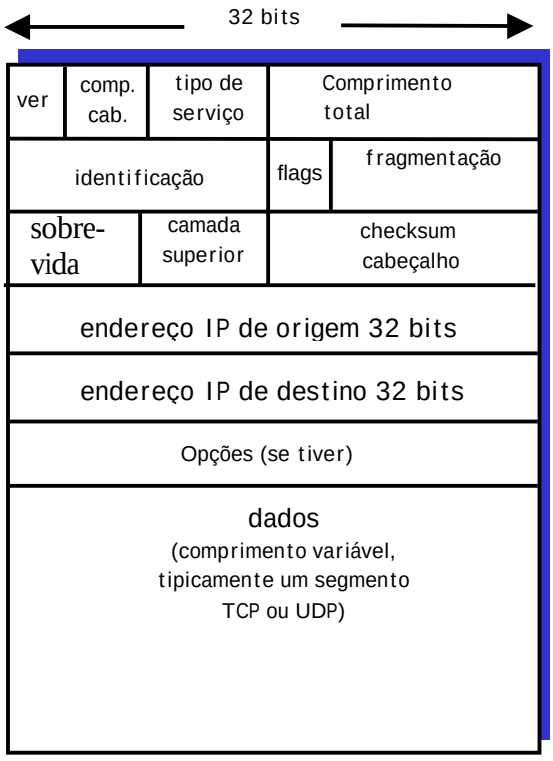


Figura 3.18. Formato do datagrama IP

campos opcionais, tipicamente o datagrama tem 20 bytes.

- O **tipo de serviço** permite diferenciar diferentes datagramas, como mensagens de controle (como ICMP) e dados normais (como mensagens HTTP), datagramas tempo-real (como aplicações de telefonia), etc.
- Os ***flags*** e **fragmentação** são usados em caso de fragmentação do datagrama IP.
- O **tempo de sobrevida**, **TTL** (*time-to-live*), indica o tempo de vida do datagrama, após o qual o mesmo é descartado.
- O **protocolo da camada superior** utilizado, como por exemplo TCP ou UDP.
- O ***checksum***, utilizado para detecção de erros no cabeçalho.
- O campo de **opções** é raramente usado.
- O campo de **dados**, que é a razão de ser do datagrama, e tipicamente carrega segmentos TCP ou UDP.

O **comprimento total** do datagrama, teoricamente poderia ser de 64K bytes (em função dos 16 bits do campo), todavia, na prática, nunca é maior que 1.500 bytes e freqüentemente é limitado em 576 bytes. Isto é feito para evitar a fragmentação do datagrama na rede física, já que o mesmo é encapsulado em um quadro da camada enlace e nem todas tem quadros de mesmo tamanho. No caso das redes locais Ethernet o tamanho do quadro é de 1.500 bytes e em outros enlaces é de 576 bytes. O tamanho máximo dos pacotes que podem ser transportados pela camada enlace é chamado de **MTU** (*maximum transfer unit*) (Figura 3.19).

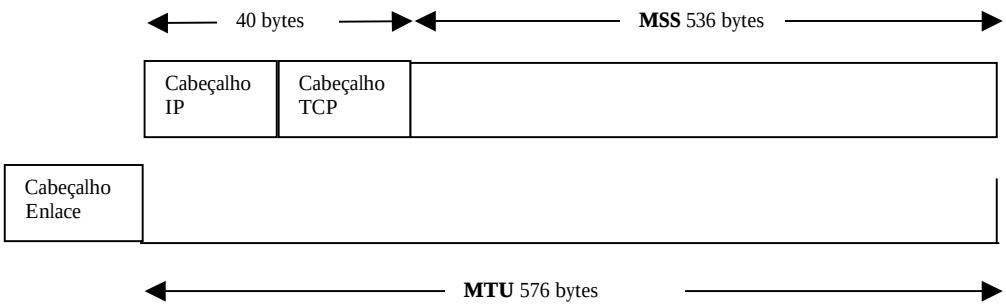


Figura 3.19. Valores práticos de MSS e MTU

Endereçamento IP

Endereço IP é um endereço lógico de **32 bits**, escrito em quatro octetos representados em decimal, cada um variando de O a 255. Os números são separados por pontos. Por exemplo, 193.32.216.9 seria um endereço válido, e sua notação em binário seria:

11000001 00100000 11011000 00001001.

Cada computador que esteja rodando o TCP/IP exige um endereço IP exclusivo. A exclusividade de endereço deve ser sempre mantida, mesmo ao se conectar a Internet.

Cada endereço IP engloba duas partes: o **identificador da rede** e o **identificador do *host***. O identificador da rede identifica a rede onde se encontram todos os *hosts* da mesma rede local. O identificador do *host* identifica um dispositivo em uma rede local, como um computador ou roteador. Por exemplo, a figura 3.20 ilustra três redes locais interconectadas por um roteador com

três interfaces. Olhando para os endereços IP atribuídos a cada computador e a cada interface do roteador, podemos notar, por exemplo, que os dispositivos conectados a rede local da esquerda e acima tem os endereços IP da forma 200.1.2.X. Isto é, compartilham os 24 bits mais à esquerda do endereço IP. No jargão IP, esta parte do endereço forma o identificador da rede. Os 8 bits restantes permitem identificar cada *host* da rede local. O endereço da rede local seria 200.1.2.0/24, onde a notação "/24" é também conhecida como **máscara de rede**, e indica que os 24 bits mais à esquerda dos 32 bits do IP identificam a rede.

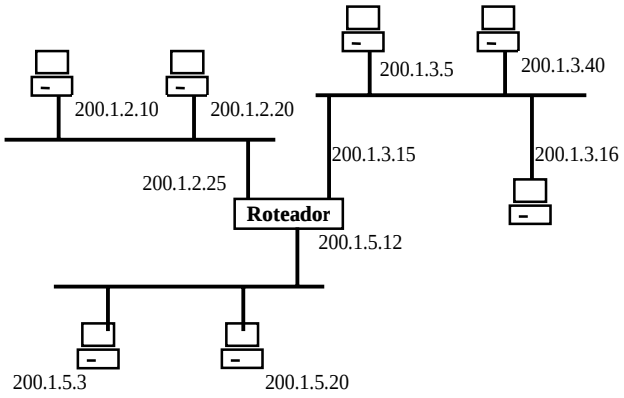


Figura 3.20. Endereçamento IP

Classes de endereçamento de IP

Para garantir endereços exclusivos em âmbito mundial, os endereços IP são licenciados a partir de uma localização central[1]. Quando foi criado, havia quatro classes básicas de licenças para endereços IP, cada uma especificando uma gama de endereços que podem ser atribuídos à licença (Figura 3.21). Na **classe A** os primeiros 8 bits identificavam a rede e os últimos 24 bits poderiam ser atribuídos aos *hosts* nesta rede, o que permitiria 2^{24} endereços. Na **classe B** o espaço de endereçamento para *hosts* seria de 2^{16} endereços. Já na **classe C,** a menor delas, deixaria 8 bits para serem atribuídos a *hosts*, ou 2^{8} endereços. A **classe D** é reservada para endereços de *multicast*.

Estas classes de endereçamento não são mais utilizadas como parte formal dos da arquitetura de endereçamento IP, pois, com o crescimento do número de organizações de pequeno e médio porte o espaço de endereçamento ficou limitado. Por exemplo, uma rede classe C (/24) pode acomodar

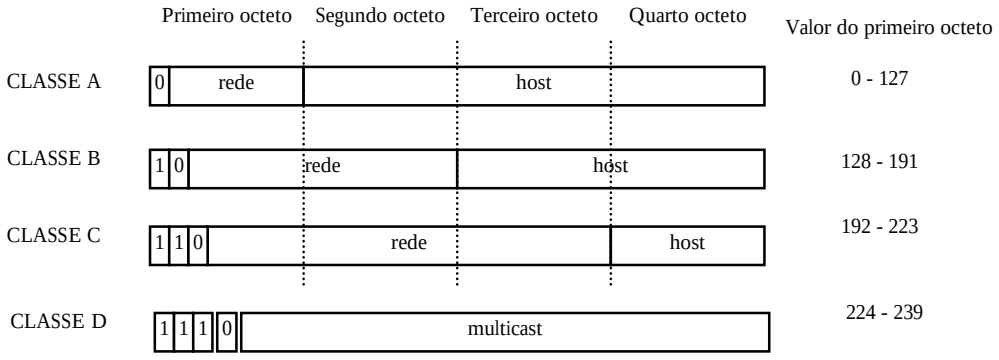


Figura 3.21. Classes de endereços IP

até 2^8 endereços, ou seja 256 *hosts*, o que pode ser muito pouco para muitas organizações. Já uma classe B (/16), poderia acomodar 2^{16} endereços, ou 64.634 endereços, o que seria demais para uma organização com, por exemplo, 2000 computadores.

Isto foi resolvido pelo IETF com a definição do padrão chamado **CIDR** (*classes interdomain routing*), que permite as organizações obterem um identificador de rede com qualquer tamanho. A notação utilizada pelo CIDR é **a.b.c.d/x**, onde o **x** é a **máscara de rede** que indica o número de bits reservados para a identificação da rede. Por exemplo, uma organização com 2000 computadores poderia solicitar um bloco de 2048 endereços, cuja notação seria a.b.c.d/21, e indica que os primeiros 21 bits identificam a rede e os 11 bits restantes (2^{11} =2048) caracterizam o espaço de endereçamento. No caso da nossa rede no CEFET em São José, licença é 200.135.233.0/24, a qual nos permite atribuir internamente até 256 endereços.

Alguns endereços IP têm utilização especial. Por convenção, um **endereço de rede** tem o campo identificador de *host* com todos os bits iguais a **0**. Podemos também nos referir a todos os *hosts* de uma rede através de um **endereço de difusão**, onde todos os bits são iguais a **1**. Um endereço com todos os 32 bits iguais a 1 é considerado um endereço de difusão para a rede do *host* origem do datagrama. O endereço 127.0.0.0 é reservado para teste (*loopback*) e comunicação entre processos da mesma máquina. Os endereços com o primeiro octeto entre 240 e 255 são reservados para uso futuro.

Roteamento

O **roteamento** inter-redes é a principal função do protocolo IP. O protocolo assume que um *host* é capaz de enviar datagramas a qualquer outro *host* conectado á mesma rede local. Caso o destinatário não esteja na mesma rede, parte da função de roteamento é transferida para os **roteadores** *(gateways*).

Os roteadores podem ser equipamentos específicos ou computadores normais que possuem mais de uma interface de rede. O roteamento no IP baseia-se exclusivamente no **identificador de rede** do endereço destino. Cada roteador possui uma tabela, chamada **tabela de roteamento**, cujas entradas são pares: **endereço de rede / endereço de roteador**. Por exemplo, quando um *host* deseja enviar um datagrama, inicialmente ele verifica se o destinatário está conectado a rede local. Se for o caso, ele entrega o datagrama a interface de rede que se encarrega de mapear o IP no endereço físico do *host* destino, encapsular o datagrama IP em um quadro da rede e transmiti-lo. Caso o *host* destino não se encontre na rede local, ele envia o datagrama ao **roteador padrão** (*gateway default*) da rede local. O roteador procura na sua **tabela de roteamento** o endereço do roteador que deve ser usado para alcançar a rede onde está conectado o destinatário do datagrama. O roteador encontrado pode não fazer parte da rede destino, mas, deve fazer parte do caminho a ser percorrido para alcançá-la.

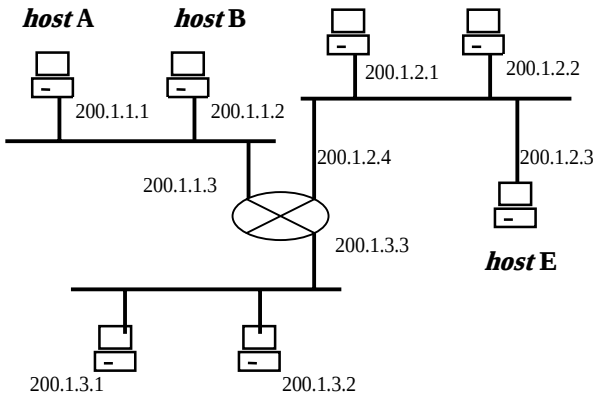


Figura 3.22. Redes e roteamento

Veja um exemplo de como funcionam as

[1] No Brasil o fornecimento de endereços IP é realizado pela FAPESP em São Paulo (www.fapesp.br).

tabelas de roteamento, considerando o contexto da rede apresentada na figura 3.22.

Suponha que o *host* A tenha a tabela de roteamento dada na tabela a seguir e deseja enviar um datagrama IP ao *host* B. Neste caso, o *host* A consulta sua tabela de roteamento e descobre que a rede 200.1.1.0/24 casa com o identificador da rede do *host* B. A tabela indica que o número de *hops* (número de enlaces a percorrer) é 1, o que quer dizer que está na mesma rede local. Então o *host* A passa o datagrama diretamente a camada enlace para proceder à entrega ao *host* B.

Tabela de roteamento do *host* A		
Rede destino	Próximo roteador	N. hops
200.1.1.0/24	-	1
200.1.2.0/24	200.1.1.3	2
200.1.3.0/24	200.1.1.3	2

Suponha agora o caso em que o *host* A queira enviar um datagrama ao *host* E, situado em outra rede, no caso a rede 200.1.2.0/24. Consultando sua tabela de roteamento ele verifica que o número de *hops* é 2, logo não está na mesma rede local, e que o acesso ao *host* E deve se dar através do roteador 200.1.1.3. Então ele passa o datagrama ao roteador para dar prosseguimento a entrega.

O roteador então consulta sua tabela de roteamento (veja tabela abaixo) e verifica que a rede 200.1.2.0/24 é acessível diretamente através da sua interface endereçada por 200.1.2.4. Sendo assim, ele entrega o datagrama a camada de enlace da rede 200.1.2.0/24 para fazer a entrega ao *host* E.

Tabela de roteamento do roteador			
Rede destino	Próximo roteador	N. hops	Interface
200.1.1.0/24	-	1	200.1.1.3
200.1.2.0/24	-	1	200.1.2.4
200.1.3.0/24	-	1	200.1.3.3

Protocolo de roteamento RIP

Na Internet, um **algoritmo de roteamento** ainda bastante utilizado é o **RIP** (*routing information protocol*) e apresenta **tabelas de roteamento** bastante parecidas com as do exemplo anterior.

As tabelas de roteamento RIP são construídas dinamicamente, baseadas em um algoritmo de roteamento que calcula as rotas tendo como base o número de enlaces a percorrer, escolhendo a rota que percorre o menor número de enlaces.

A partir do comando Unix `netstat -rn` pode-se visualizar as tabelas de roteamento RIP de um roteador Unix.

Parâmetros básicos para configuração do TCP/IP

Qualquer computador utilizando o TCP/IP possui três parâmetros básicos de configuração: **endereço IP**, **máscara de rede** e **roteador padrão**.

Endereço de IP

Endereço lógico exclusivo de 32 bits, escrito em quatro octetos representados em decimal.

Máscara de Rede

A máscara de rede é utilizada para "mascarar" uma parte do endereço IP para que se possa distinguir o identificador da rede do identificador do *host*. Quando dois *hosts* desejam se comunicar, a máscara da rede é utilizada para determinar se um *host* está localizado na rede local ou em uma rede remota.

Exemplos de máscara de rede:

Classe	N. de hosts	Bits usados para a máscara	Notação em decimal
/20	2^{12} = 4096	11111111 11111111 11110000 00000000	255.255.240.0
/21	2^{11} = 2048	11111111 11111111 11111000 00000000	255.255.248.0
/24	2^{8} = 256	11111111 11111111 11111111 00000000	255.255.255.0

Para se extrair o **identificador da rede** a partir do endereço IP completo, uma operação lógica **AND** é realizada com a máscara de rede.

Por exemplo, para descobrir o identificador de rede do *host* Joplin cujo endereço IP é 200.135.233.4 e cuja máscara de rede é 255.255.255.0, devemos fazer uma operação AND desdes dois valores:

```
        11001000 10000111 11101001 00000100
        11111111 11111111 11111111 00000000
AND     ___________________________________
        11001000 10000111 1110100100000000
```

o qual será igual a 200.135.233.0 (rede do CEFET em São José).

Roteador Padrão

Para comunicação com um *host* de uma outra rede, deve-se configurar um endereço IP para o **roteador padrão** (*defaut gateway*). O roteador padrão é o local para onde o TCP/IP envia pacotes destinados a redes remotas. Se um roteador padrão não for especificado, as comunicações se limitarão à rede local.

Exercício

A partir do Painel de Controle do Microsoft Windows, verifique a configuração do TCP/IP de seu computador.

Mapeamento do IP em um endereço físico da rede local

Protocolo ARP

Quando um *host* deseja enviar um datagrama a um destinatário conectado à sua rede local, ele entrega o datagrama a interface de rede para ela mapear o **endereço IP** no **endereço físico** (endereço de placa[2]) do *host* destino.

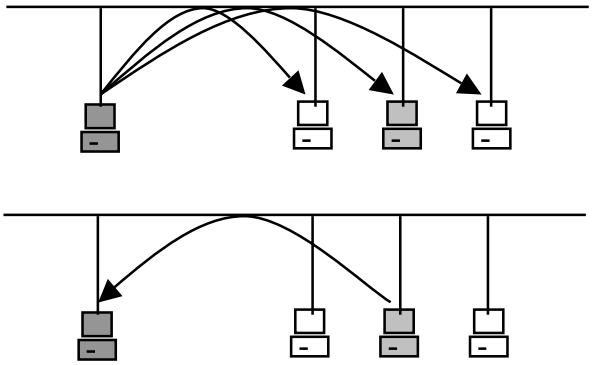


Figura 3.23. Protocolo ARP

O **protocolo ARP** permite encontrar o endereço físico a partir do endereço IP da máquina alvo. Para tal, o protocolo usa um mecanismo de difusão (*broadcast)*, enviando uma solicitação a todas as máquinas da rede, sendo que a máquina alvo responde indicando o par **endereço IP / endereço físico** (Figura 3.23).

Para melhorar a performance do protocolo, cada máquina possui uma memória (*cache)* com as últimas consultas realizadas, evitando múltiplos *broadcasts*. Ainda como refinamento, junto com o *broadcast*, a estação solicitante envia seu par endereço IP / endereço físico, permitindo que todas as máquinas da rede incluam este par em suas *caches* locais.

Quando um hardware é trocado, a máquina que sofreu a mudança se anuncia na rede com o novo par endereço IP / endereço físico, logo após sua entrada em operação.

Protocolo RARP

O **protocolo RARP** realiza a operação inversa do ARP, isto é, a partir de um **endereço físico** permite encontrar o **endereço IP** da máquina.

É geralmente utilizado por máquinas sem disco rígido (*disk-less)* para obter um endereço IP de um servidor. Para tal, um *host* RARP envia um *broadcast* com o seu endereço físico solicitando um endereço IP. A máquina autorizada a responder o pedido RARP envia a resposta.

Alternativas mais modernas ao protocolo RARP são o **BOOTP** e o **DHCP**, ambos construídos sobre protocolos de mais alto nível, como o IP e o UDP.

Protocolo BOOTP

O RARP é um protocolo de baixo nível, que exige um acesso direto ao *hardware* de rede para obter um IP.

Pelo fato de usar o UDP e o IP, o **BOOTP** pode ser implementado como um programa de aplicação. Além disto, é mais eficiente que o RARP, especificando vários itens necessários para a inicialização além do endereço de IP, como o endereço de um roteador ou de um servidor.

O BOOTP usa o UDP para carregar uma mensagem que é encapsulada em um datagrama IP. Para realizar o broadcast deste datagrama com a solicitação de um endereço IP, é utilizado o *broadcast* limitado na rede local (endereço IP: 255.255.255.255), mesmo antes de se saber qual o endereço IP da rede local ou do *host*.

---

[2] Por exemplo, as redes Ethernet possuem um endereço físico de 48 bits, gravados em memória Eprom pelo fabricante da placa.

Alocação dinâmica de IP

O **protocolo DHCP** (*dynamic host configuration protocol*) é uma extensão do protocolo BOOTP e permite a alocação dinâmica de endereços IP (o BOOTP é baseado em tabelas estáticas). Com o DHCP, um **servidor DHCP** recebe uma solicitação de um **cliente** e aloca dinamicamente um **endereço IP** em resposta ao pedido do cliente. Com o DHCP um computador cliente pode adquirir toda a configuração necessária em uma única mensagem (por exemplo, o endereço IP, máscara de rede, roteador padrão, servidor DNS, etc).

O **servidor DHCP** deve ser configurado com a **faixa de endereços IP** disponíveis para oferecer. Quando um computador se conecta na rede, ele solicita um endereço IP se apresentando com seu endereço físico. O servidor então escolhe um endereço IP dentro da faixa disponível e aloca ao solicitante.

Protocolo ICMP

Conforme já mencionado, o **protocolo IP** fornece um serviço de datagramas não confiável e não orientado a conexão, onde um datagrama segue de roteador em roteador até alcançar seu destino final. Se um roteador não consegue encontrar uma rota ou entregar um datagrama, ou se uma condição anormal é detectada, o roteador precisa informar a fonte original dos dados para que esta tome alguma ação ou corrija o problema. O **protocolo ICMP** (*Internet Control amd Message Protocol*) permite que os roteadores enviem **mensagens de erro e controle** a outros roteadores ou *hosts*; oferecendo uma comunicação entre a camada rede de uma máquina e a camada rede de outra máquina.

Tecnicamente o ICMP é um mecanismo de reportagem de erros. Ou seja, quando um datagrama causa um erro, o ICMP pode reportar a condição de erro de volta a fonte original do datagrama; a fonte então relata o erro para a aplicação ou realiza uma ação com vistas a corrigir o erro. Por exemplo, quando rodando uma aplicação Telnet ou HTTP, podemos encontrar mensagens como "rede destino não encontrada" (*destination network unreachable*), que tem origem no protocolo ICMP.

O ICMP é normalmente considerado como parte do IP, todavia está situado logo acima. As mensagens ICMP são carregadas na porção de dados de um datagrama IP, que as identifica como tipo ICMP. Os datagramas contendo as mensagens ICMP seguem de volta, seguindo exatamente o caminho que tomaram os dados do usuário, podendo elas também serem perdidas ou corrompidas.

Formato das Mensagens ICMP

Cada mensagem ICMP tem um campo de **tipo** e um campo de **código**, e também contém os primeiros 8 bytes do datagrama que causou o erro (com isto o emissor pode determinar o pacote que causou o erro).

Algumas mensagens ICMP:

ICMP Tipo	Código	Descrição
0	0	*echo reply (*para o Ping*)*
3	0	*destination network unreachchable*
3	1	*destination host unreachchable*
3	6	*destination network unknow*
3	7	*destination host unknow*
8	0	*echo request*
11	0	*TTL* (*time to live*) *expire*

Nem todas as mensagens ICMP são de reportagem de erros. A aplicação Ping, por exemplo, utiliza mensagens ICMP *Echo Request* e *Echo Reply* para verificar se um host está disponível e sua respectiva resposta.

O Traceroute, que é capaz de traçar a rota que liga um *host* a outro *host*, também usa mensagens ICMP. Para determinar o nome e o endereço dos roteadores entre a fonte e o destino, o Traceroute na fonte envia uma série de datagrama IP ordinários ao destino. O primeiro datagrama tem o TTL igual a 1, o segundo 2, o terceiro 3, e assim por diante, e inicia temporizadores para cada datagrama. Quando o enésimo datagrama chega ao enésimo roteador, este verifica que o tempo de sobrevida do datagrama acaba de terminar. Pelas regras do IP, o datagrama é então descartado e uma mensagem ICMP de advertência é enviada a fonte (tipo 11 código 0), com o nome do roteador e seu endereço IP. Quando a resposta chega de volta a fonte, a mesma calcula o tempo de viagem em função dos temporizadores.

Sistema de Nomes de Domínio

Um nome de domínio é um nome hierárquico implementado com a utilização de um **Sistema de Nomes de Domínio** (**DNS** *domain name system*).

O **DNS** proporciona um banco de dados *on-line* e distribuído para resolver nomes de domínios a seus endereços IP correspondentes. Isto facilita na medida em que não precisamos mais memorizar endereços IP, mas sim **nomes de domínio**, muito mais fáceis de serem lembrados e ao mesmo tempo identificados com o proprietário do domínio.

Se uma organização deseja participar da Internet, deve registrar o seu nome de domínio no Centro de Informações de Rede.

Principais nomes de domínio Internet

Nome de Domínio	Significado
edu	Instituição educacional
com	Organização comercial
gov	Instituição governamental
org	Organização não governamental
<código de país>	Cada país (esquema geográfico)

Exemplos:

ufsc.br cefetesc.edu.br mec.gov.br matrix.com.br
mit.edu national.com (nos USA não há sigla de país)

Além da sintaxe para os nomes, o esquema DNS inclui um sistema distribuído eficiente, seguro e de propósito geral para se mapear nomes em endereços.

O DNS consiste da união de sistemas cooperativos independentes, chamados **servidores de nomes**, que fazem a translação do **nome de domínio** em **endereço IP**. O software cliente, chamado **resolvedor de nomes**, usa um ou mais servidores de nomes para traduzir um nome.

Resolução de Nomes

A resolução de nomes esta baseada em uma árvore hierárquica de nomes. Conceitualmente a resolução inicia de cima para baixo (*top-down)*, começando no servidor raiz e seguindo para os servidores localizados nos ramos da árvore.

Há dois modos possíveis para um servidor resolver um nome: **resolução interativa** (passo-a-passo) ou **resolução recursiva**.

Em ambos os casos, o servidor consultado verifica se o nome solicitado pertence a um sub-domínio seu. Se for o caso, traduz o nome ao endereço de acordo com sua base de dados. Se não puder resolver o nome completamente, verifica o tipo de solicitação feita pelo cliente. Se o cliente solicitou busca recursiva o servidor contata um DNS que possa resolver o nome e devolve a resposta ao cliente. Caso a solicitação foi do tipo interativa, ele fornece o nome de um DNS ao cliente e não a resposta da resolução completa do nome.

Para iniciar a busca, um cliente precisa saber como conectar pelo menos um servidor de nomes raiz. Em adição, um servidor de nomes deve saber o endereço de um servidor de nomes de domínio imediatamente superior (servidor pai).

Como a maioria das consultas é de âmbito local, a eficiência do sistema é aumentada iniciando-se a busca em um servidor de nomes local. Além disto, os servidores de nomes da Internet usam memória *cache* para otimizar os custos da busca de nomes não locais. Todos os nomes recentemente usados são armazenados na sua memória *cache*, bem como a informação de como foram obtidos. Como a informação em memória pode estar desatualizada, o servidor de nomes marca como não autoritativa (*non authoritative)*, podendo o cliente contatar a autoridade para ver se o nome ainda é válido.

Questões

1. Qual o papel dos protocolos da **camada de transporte** da Internet?
2. Explique a relação entre os protocolos da **camada transporte** e da **camada rede** da Internet.
3. Em que consiste o serviço de **multiplexação** de aplicações oferecido pelos protocolos de transporte TCP e UDP.
4. Qual informação é utilizada por um processo que está executando em um computador, para identificar um processo que está executando em outro computador remoto.
5. Pesquise na Internet a lista completa das **portas** TCP e UDP reservadas para aplicações específicas. Ache um endereço URL com esta informação.
6. Suponha que você está desenvolvendo uma aplicação para a Internet. Que tipo de protocolo de transporte você utilizaria, **TCP** ou **UDP**? Explique, tendo como base à aplicação que será desenvolvida
7. Quais são os princípios utilizados pelos **protocolos de transporte confiável**, como o TCP, para garantir que os dados transmitidos são livres de erros?
8. Diferencie os objetivos dos serviços de **controle de fluxo** e de **controle de congestionamento**, presentes no protocolo de transporte TCP.
9. Para que serve e como funciona o mecanismo de ***chesksum*** utilizado pelo TCP, UDP e IP? Cite um exemplo prático.
10. Para que servem os **números de seqüência** e **reconhecimento** presentes no cabeçalho do segmento TCP? Explique o processo utilizado para numerar os segmentos.

11. Em que consiste o ***handshaking*** do TCP? Explique as informações que são trocadas neste processo.
12. Quais os papéis da **camada rede** da Internet?
13. O que é **MSS**? Explique.
14. O que é **MTU**? Explique.
15. Explique o formato do **endereço IP**, em termos de número de bits e sua representação em decimal. Qual é o número binário equivalente aos endereços IP 200.135.233.1 (www.sj.cefetsc.edu.br) e 150.161.1.150 (www.ufsc.br)?
16. Explique as diferentes **classes de endereços IP** existentes, comentando também para que serve o padrão **CIDR**.
17. Explique o processo de **roteamento estático de datagramas**, realizado com a ajuda de tabelas de roteamento.
18. Cite algumas tecnologias de rede que usam roteamento tipo **circuito virtual**. Ache alguns endereços URL que expliquem estas tecnologias.
19. Quais os **parâmetros básicos de configuração do TCP/IP** em um computador conectado a Internet. Explique o papel de cada parâmetro. Mostre os passos para configurar o TCP/IP em um computador com o sistema operacional Windows e com o Linux.
20. Para que serve o protocolo **ARP**? Explique.
21. Explique para que serve a aplicação **DNS**.
22. Explique para que serve o protocolo **DHCP**.

Parte 4

Protocolos de Enlace e Redes Locais

As redes locais são redes de computadores concentradas em uma área geográfica restrita, por exemplo no âmbito de uma escola ou universidade, e permitem aos computadores e usuários da rede compartilharem recursos. Com a possibilidade de as redes locais serem conectadas entre si, formando a Internet, cresceu de forma extraordinária as possibilidades de acesso a recursos e serviços, de forma que hoje praticamente não se pensa mais uma rede local isolada.

O que é um protocolo de enlace?

A camada de rede da Internet oferece um serviço de comunicação de datagramas entre dois sistemas terminais. Esta comunicação passa por caminhos que iniciam no *host* de origem, passando por uma série de roteadores e termina no *host* destino. Cada equipamento, como *hosts* e roteadores, é chamado de nó e o **canal de comunicação entre dois nós adjacentes ao longo de uma rota** é chamado de **enlace**. Desta forma, para mover um datagrama desde sua origem até seu destino, ele precisa percorrer cada um dos enlaces individuais entre os diversos nós.

Os enlaces entre nós vizinhos podem ser suportados por diferentes tecnologias, utilizando protocolos específicos, os quais são chamados de **protocolos de enlace**. As **unidades de dados de protocolos** trocadas pelos protocolos de enlace são chamadas **quadros** (*frames*) e tipicamente encapsulam um datagrama da camada rede.

> Assim como os protocolos de rede são protocolos fim-a-fim que movem datagramas de um *host* a outro, os **protocolos de enlace** são **protocolos nó-a-nó**, movendo **quadros** sobre um simples enlace.

Serviços oferecidos pelos protocolos de enlace

Dentre os possíveis serviços oferecidos pelos protocolos de enlace está o **acesso ao meio físico** e o **encapsulamento** (*framing*). No caso, os datagramas da camada rede são encapsulados em quadros e o acesso ao meio vai depender do tipo de protocolo utilizado. Grosso modo podemos dividir os protocolos de enlace em dois grandes grupos: os **protocolos de enlace ponto-a-ponto** e os

protocolos de enlace multiponto, que caracterizam as **redes locais**. No caso de um protocolo ponto-a-ponto o acesso ao meio é bastante simples, aceitando o envio de um quadro caso o meio estiver livre. Já no caso de protocolos multiponto, como o protocolo de rede local Ethernet, há necessidades de mecanismos especiais para acesso ao meio.

Além destes serviços básicos, dependendo do protocolo, outras ações podem ser executadas sobre os quadros, como: comunicação *full-duplex* ou *half-duplex*, detecção e correção de erros, entrega de dados garantida e controle de fluxo.

Placas adaptadoras

Os **protocolos de enlace** são em geral implementados sobre **placas adaptadoras** que fazem a interface entre o *host*, seja ele um computador terminal ou um roteador, e o meio físico. Desta forma, os principais componentes de um **adaptador de rede** são sua interface com o barramento do *host* e sua interface com o enlace físico (Figura 4.1). Por exemplo, uma **placa de rede Ethernet** de 10 Mbps possui uma interface para conexão da mesma diretamente no barramento do computador, e uma interface de rede, que pode ser tipo RJ45 (conexão com par trançado) ou BNC (conexão com cabo coaxial).

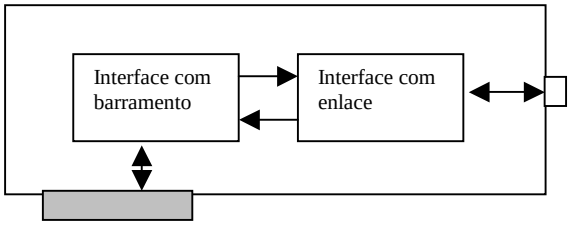


Figura 4.1. Adaptador de rede

Técnicas de detecção e correção de erros

Em qualquer transmissão de informação existe o risco do erro sob o efeito de perturbações aleatórias ou de ruídos (*noise*). De um modo geral, os **erros** nos dados transmitidos através da rede podem ser:

- Erros de bit introduzidos nos dados;
- Perda de pacotes;
- Falha nos enlaces de comunicação.

Os **erros de bits** são bastante raros, havendo técnicas para detectá-los e mesmo corrigi-los. Se o erro for muito grave, o pacote pode ser descartado e terá que ser retransmitido. No caso da **perda de pacotes**, a retransmissão é a solução. Já no caso de **falha de um enlace**, algumas vezes é possível utilizar uma rota alternativa, evitando a ligação com defeito.

A **detecção e correção de erros no nível de bits** dos quadros enviados de um nó a outro nó fisicamente conectado são geralmente serviços oferecidos pelos protocolos da camada de enlace.

Três técnicas simples de detecção e correção de erros no nível de bits são a **checagem de paridade**, os **métodos de *checksum*** e os **métodos de redundância cíclica**.

Checagem de paridade

Talvez a forma mais simples de detecção de erros de bits seja utilizar um simples **bit de paridade**. Por exemplo, suponha que um dado D a ser transmitida tenha *d* bits. Usando um esquema de paridade, o emissor acrescenta ao dado um bit adicional e escolhe seu valor como o total de bits em 1 de *d* + 1 bits (o total de bits em D mais o bit de paridade), de forma que seja par (Figura 4.2).

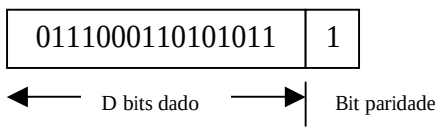


Figura 4.2. Bit de paridade

Quando o receptor recebe o dado, ele computa os bits em 1, incluindo o bit de paridade, e verifica se o resultado é par. Caso não seja, o receptor sabe que algum bit teve seu valor alterado.

Algumas técnicas permitem, além de detectar erros em bits, de corrigi-los. Estas técnicas são conhecidas como **FEC** (*forward error correction*). Elas são úteis, pois permitem diminuir a necessidade de retransmissões pelo emissor.

Checksum

Na técnica de ***checksum*** o dado D é tratado como uma seqüência de palavras binárias. O método consiste em somar a seqüência de palavras e usar a soma para detectar erros nos bits. Este é método utilizado pelos protocolos Internet (veja exemplo anterior sobre o *checksum* do UDP).

Checagem de redundância cíclica

Os **códigos de redundância cíclica**, ou códigos **CRC** (*cyclic redundancy check*), estão entre os métodos mais utilizados nas redes de computadores para detecção de erros, pois podem descobrir mais erros que um *checksum.*

Os códigos CRC são também conhecidos como **códigos polinomiais**, já que podem ser vistos como um polinômio onde os coeficientes são 0 e 1. Por exemplo, o número binário de 4 bits, 1011, corresponde ao polinômio

$$M(x) = 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0 = x^3 + x^1 + 1, \text{ cujo grau é } 3.$$

Os **códigos CRC** operam como segue. Para uma peça de dados D a ser transmitida, o emissor e o receptor devem acordar primeiramente sobre um **polinômio gerador**, G, de grau *r*. Assim, o emissor adiciona ao dado D mais *r* bits, de forma que o resultado da soma *d + r* seja exatamente divisível por G usando aritmética **módulo 2** (Figura 4.3).

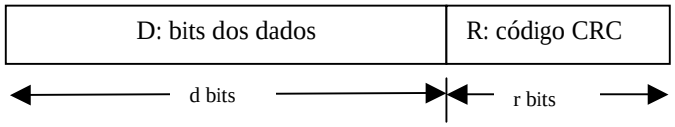


Figura 4.2. Código CRC

Quando o receptor recebe os dados, ele divide *d + r* por G; caso a divisão não seja exata, ele sabe que há erros nos dados; caso contrário, o dado é considerado correto.

Protocolos de enlace ponto-a-ponto

Um **protocolo de enlace ponto-a-ponto** consiste de um simples emissor em uma extremidade de um enlace e um simples receptor na outra ponta. Muitos protocolos tem sido desenvolvido para este tipo de comunicação, como por exemplo, o protocolo **PPP** (*point-to-point protocol*) e o **HDLC** (*high-level data link control*).

O PPP é tipicamente o protocolo escolhido para conectar um computador pessoal residencial a um provedor de acesso a Internet, usando uma linha telefônica, sendo sem dúvida um dos protocolos ponto-a-ponto mais utilizados atualmente.

Protocolo PPP

O **protocolo PPP** pode operar sobre uma linha telefônica (usando por exemplo uma conexão via modem de 54K bps), sobre um enlace SONET/SDH (*syncrhonous optical network/syncrhonous digital hierarchy*), sobre uma conexão X.25 ou sobre um circuito digital RDSI (rede digital de serviços integrados).

O protocolo PPP recebe um pacote da camada rede (por exemplo, um datagrama IP) e o encapsula em um **quadro da camada enlace PPP**, de forma que o receptor será capaz de identificar o início e o fim do quadro, bem como o pacote da camada rede que ele contém.

O formato do **quadro PPP** (Figura 4.4) sempre inicia e termina com 01111110 (chamado de *flag*), o segundo byte é sempre 11111111 (chamado de endereço) e o terceiro byte é sempre 00000011 (chamado de controle). Os demais campos são os seguintes:

- *Protocol* (1 ou 2 bytes), indica ao receptor qual o protocolo da camada de rede que está sendo encapsulado no quadro. No caso de um datagrama IP, este campa tem o valor hexadecimal 21.
- *Information* (tamanho variável, podendo ter no máximo 1500 bytes), contém o pacote encapsulado (dado), por exemplo um datagrama IP.
- *Checksum* (2 a 4 bytes), usado para detectar erros nos bits do quadro transmitido.

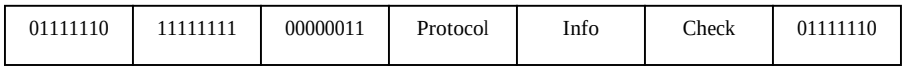


Figura 4.4. Quadro PPP

O protocolo **SLIP** (*Serial Line Internet Protocol*) é outro protocolo similar ao protocolo PPP.

Protocolos de enlace de múltiplo acesso

A **Ethernet** talvez seja a tecnologia mais utilizada em **enlaces de múltiplo acesso** (*broadcast*), freqüentemente utilizada em **redes locais** (**LAN** – *local area networks*).

O problema central nos enlaces de múltiplo acesso é determinar quem deve transmitir e quando. Com vários podem transmitir quadros ao mesmo tempo, estes poderão **colidir** e serão perdidos. Os **protocolos de acesso múltiplo ao meio** permitem coordenar as colisões.

A distribuição aberta de TV é um exemplo clássico de sistema tipo *broadcast*, todavia, este sistema opera apenas em um sentido, difundindo a informação. Por outro lado, os computadores conectados a um enlace múltiplo acesso devem poder receber e transmitir informações. São exemplos deste último tipo de sistema as redes locais que compartilham um barramento (por exemplo, a Ethernet), as redes locais sem fio (redes *wireless*), sistemas de comunicação via satélite, etc (Figura 4.5). Um exemplo humano de um sistema tipo *broadcast* é uma assembléia de trabalhadores, onde vários podem falar ao mesmo tempo, sendo o problema controlar quem deve falar, quando falar e por quanto tempo.

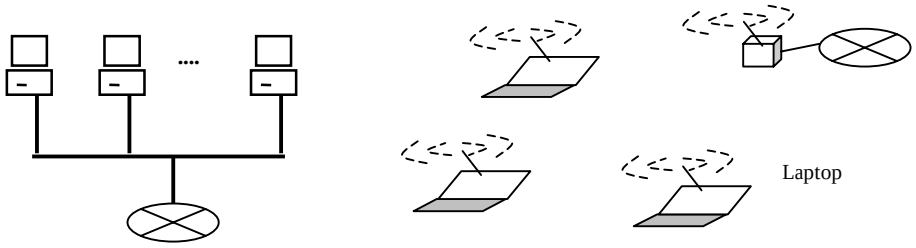


Figura 4.5 Barramento Ethernet e rede *wireless*

Protocolos para particionar um canal comum

Sistemas, como a distribuição de rádio e TV, usam a técnica de **multiplexação por divisão da freqüência** (**FDM** – *frequency division multiplexing*) para separar cada canal a ser transmitido no meio comum, no caso o meio físico é o ar a partir do uso do espectro de freqüência de rádio. Por exemplo, na grande Florianópolis, a rádio FM Itapema transmite na freqüência de 93,7 MHz, a rádio Atlântica em 100,9 MHz, a Antena 1 em 92,1 MHz, etc. Por outro lado, alguns enlaces de fibra óptica, por exemplo, usam a técnica de **multiplexação por divisão do tempo** (**TDM** – *time division multiplexing*), para separar vários canais a serem transmitidos sobre uma única fibra. Cada canal utiliza um intervalo de tempo específico para transmissão (*time slot)*, enquanto os demais aguardam sua vez para transmitirem. No caso da assembléia de trabalhadores comentada a pouco, se fosse utilizada a técnica TDM, poderia ser estabelecida uma ordem e em tempo especifico para cada um falar, de forma que não houvesse dois falando ao mesmo tempo. Estas duas técnicas, **FDM** e **TDM**, separam cada canal de transmissão de forma que não há colisões entre os dados de cada canal.

Outra técnica para particionar um canal comum é, por exemplo, a técnica de **acesso múltiplo por divisão de código** (**CDMA** – *code division multiple access*), usada em alguns sistemas de telefonia móvel, onde é atribuído um código diferente para cada nó.

Nas redes de computadores as técnicas mais utilizadas são conhecidas como **protocolos de acesso randômico**, ou ainda, **multiplexação estatística**. Vários protocolos deste tipo foram desenvolvidos, onde os mais conhecidos derivam do **protocolo ALOHA**, desenvolvido no final dos anos sessenta para permitir interligar via rádio os computadores espalhados pelo campus da

universidade do Hawaii (USA), situados em diferentes ilhas do Pacífico. Entre estes estão os **protocolos de múltiplo acesso baseados em escuta da portadora** (**CSMA** – *carrier sense multiple access*), que faz parte da definição do protocolo **Ethernet.**

Protocolo ALOHA

Na primeira versão do **protocolo ALOHA**, quando um nó tinha um quadro a ser transmitido, ele o transmitia imediatamente. Se após um tempo de atraso o emissor ouvisse sua transmissão (reflexão do sinal de rádio transmitido), ele assumia que não havia ocorrido conflito. Caso contrário, assumia que havia ocorrido que uma colisão e retransmitia o quadro com uma probabilidade *p*, senão esperava um tempo correspondente ao tempo de transmissão e tentava enviar novamente com probabilidade *p*.

Protocolo CSMA

O protocolo CSMA foi projetado para funcionar com computadores conectados em barramento. Foi inspirado no protocolo ALHOA e introduziu dois novos princípios:

- Escutar a portadora antes de enviar um quadro (*carrier sense*) (o que não era possível no ALHOA devido ao tempo de propagação do sinal de rádio). Neste processo, o nó escuta o canal: caso o canal estiver livre transmite o quadro imediatamente; caso o canal estiver ocupado, volta a escuta-lo depois de decorrido um tempo randômico para tentar nova transmissão.
- Se alguém começar a transmitir no mesmo tempo, pára a transmissão. Este procedimento é chamado de **detecção de colisões** (*colision detection*), onde os nós continuam ouvindo o canal enquanto transmitem: caso detectem uma sobreposição de transmissões (colisões), param imediatamente a transmissão.

Estas duas regras são as características principais do protocolo (**CSMA/CD** – *carrier sense multiple access/colision detection*), utilizado nas **redes locais** baseadas no protocolo **Ethernet**.

Redes Locais

Os protocolos de múltiplo acesso são largamente utilizados nas **redes locais** de computadores, ou **LAN**s (*local area networks*), que são redes de computadores concentradas em uma área geográfica relativamente pequena, como um edifício, uma escola ou uma universidade.

Numa rede local, todos os computadores e demais dispositivos de rede são diretamente conectados. Desta forma, usam o mesmo tipo de protocolo de enlace, em geral. Um roteador conectando a rede local a Internet provê uma forma de acesso a Internet a todos os equipamentos da rede local (Figura 4.6).

Nos anos 1980 até o início dos anos 1990 duas classes **tecnologias de redes locais** eram bastante populares: a tecnologia **Ethernet** (padronizada como **IEEE 802.3**), baseada em um protocolo de acesso randômico; e as tecnologias ***token-ring*** (padronizada como **IEEE 802.5**) e **FDDI** (*fiber digital distributed interface*), onde os *host* são conectados

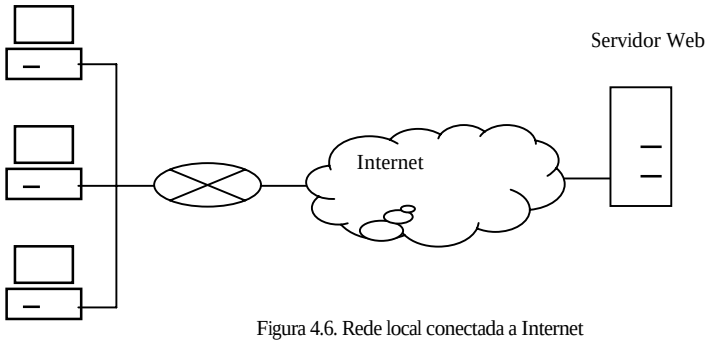


Figura 4.6. Rede local conectada a Internet

em um anel e o protocolo de acesso é baseado em passagem de ficha (*token*) entre as estações.

Nos dias atuais, há um incontestável domínio da tecnologia Ethernet frente às demais tecnologias, sendo este o modelo de redes locais que estudaremos. Além disto, cabe ressaltar que, com a popularização dos computadores portáteis (*laptops*) em ambientes empresariais, as **redes locais sem fio** (*wireless*) (padronizadas como **IEEE 802.11**) também tem tido espaço no mercado.

Endereços físicos

Como visto, os nós das redes locais trocam **quadros** (*frames*) entre si através de um **canal comum** (*broadcast*). Isto significa que, quando um nó transmite um quadro, todos os demais nós vão receber este quadro. Todavia, em geral, um nó não quer enviar quadros a todos nós, mas sim a um nó particular. Para prover esta funcionalidade, os nós de uma rede local devem ser capaz de endereçar os demais nós quando enviam um quadro. Desta maneira, quando um nó recebe um quadro, ele pode determinar se o quadro está endereçado a ele ou a outro nó da rede:

- Se o endereço do quadro recebido casa com o **endereço físico** do nó que o recebeu, então o nó extrai o datagrama (da camada de rede) do quadro recebido (camada de enlace) e repassa para cima na sua pilha de protocolos.
- Se o endereço do quadro recebido **não** casa com o endereço físico do nó o recebeu, o nó simplesmente descarta o quadro.

Em verdade, não é o nó da rede que tem um **endereço físico**, mas sim, cada **adaptador de rede** da rede local. Nas redes locais Ethernet, o endereço físico é também chamado de **endereço Ethernet** ou ainda **endereço MAC** (*media access control*). Um endereço Ethernet é um número expresso na notação hexadecimal, de seis bytes, dando 2^{48} possíveis endereços. Este endereço é permanente, sendo gravado pelo fabricante do adaptador de rede em uma memória ROM (*read only memory*) (Figura 4.7).

Resolução de endereço físico

Quando um datagrama da camada rede (por exemplo, um datagrama IP), endereçado a um computador de uma rede local chega ao roteador de borda, a partir da Internet, o roteador deverá encapsular este datagrama em um quadro da camada enlace para poder entregá-lo ao computador destino. Para que isto seja feito, o roteador deverá mapear o **endereço IP** no **endereço físico** do computador destino. Como vimos anteriormente, esta tarefa é realizada pelo protocolo **ARP** (*address resolution protocol*).

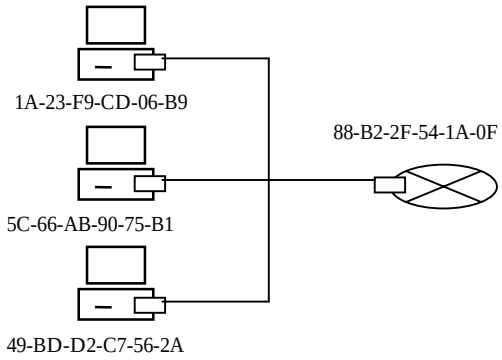


Figura 4.7. Cada adaptador de rede tem um endereço físico

Ethernet

Ethernet é a tecnologia de redes locais mais difundida atualmente. Pode-se dizer que a Ethernet está para as redes locais, assim como a Internet está para as redes geograficamente distribuídas de alcance global.

A Ethernet usa o protocolo de acesso randômico **CSMA-CD**, que é completamente descentralizado, o que facilita o projeto, e o *hardware* (em particular a **placa de rede Ethernet**) tem um custo bastante atrativo.

Existem várias tecnologias de rede local Ethernet, que operam em velocidades de 10 Mbps, 100 Mbps e 1 Gbps. Podem rodar sobre cabo coaxial, par trançado de cobre ou ainda fibra óptica,

sendo que, ao nível lógico, todas as máquinas compartilham um barramento comum, sendo portando a velocidade de acesso também compartilhada entre as estações.

Quadro Ethernet

O **quadro** (*frame*) Ethernet tem as seguintes características (Figura 4.8):

Preâmbulo	End. Dest.	End. Origem	Tipo	Dados	CRC

Figura 4.8. Quadro Ethernet

- Preâmbulo (8 bytes), cada um dos primeiros sete bytes do preâmbulo tem o valor 10101010 e o oitavo byte tem o valor 10101011.
- Endereço Destino e Origem (6 + 6 bytes), contém o endereço físico da origem e destino do quadro, nomeados AA-AA-AA-AA-AA-AA e BB-BB-BB-BB-BB-BB, respectivamente.
- Tipo (2 bytes), permite identificar o tipo do protocolo da camada superior, por exemplo, o protocolo IP (ou outro como Novell IPX).
- Dados (46 a 1500 bytes), carrega o datagrama IP, sendo o MTU (*maximum tranfer unit*) o quadro Ethernet 1.500 bytes.
- CRC (*cyclic redundancy check*) (4 bytes), permite ao receptor detectar quaisquer erros introduzidos nos bits do quadro recebido.

Tecnologias Ethernet

As tecnologias Ethernet estão padronizadas na norma IEEE 802.3 podendo ser implementadas de diversas formas:

- A tecnologia **Ethernet 10Base2** (praticamente em desuso) (Figura 4.9), usa cabos coaxiais em uma topologia em barramento e tem velocidade de transmissão de 10 Mbps (o 10 de 10Base2 indica a velocidade de 10 Mbps e o 2 denota 200 metros como a distância máxima entre dois nós). A conexão das estações, através de cabos coaxiais e placas de rede, é feita por meio de conectores BNC e terminadores nas duas extremidades.

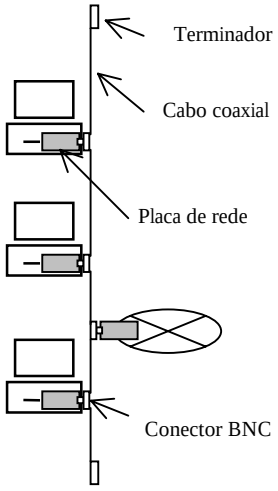


Figura 4.9. Rede local Ethernet 10Base2

- A tecnologia **Ethernet 10BaseT** e **100BaseT** (esta última também conhecida como ***fast* Ethernet**), usam par trançado de cobre em uma topologia em estrela usando um concentrador (ou *hub*) (a nível lógico a conexão também é do tipo barramento). As velocidades vão de 10 Mbps a 100 Mbps, dependendo da placa de rede e da categoria do cabeamento, sendo que muitos adaptadores de rede são 10/100Mbps. Para as conexões com cabos categoria 5, utiliza-se conectores RJ-45 e *hub* (Figura 4.10). Ambas as tecnologias podem também usar enlaces de fibra óptica, geralmente utilizados para conectar dois *hubs* localizados em diferentes edifícios de um campus, por exemplo.

- **Gigabit Ethernet**, tecnologia emergente, permite velocidades de até 1 Gbps e também usa topologia em estrela com *hub* ou *switch* no ponto central. Opera sobre fibra óptica e também sobre par trançado categoria 5e ou 6, podendo ser empregada em *backbones*.

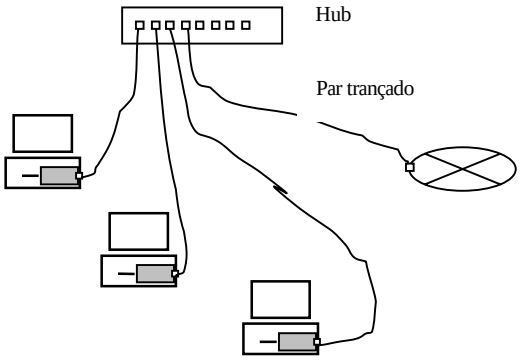


Figura 4.10. Rede local Ethernet 10/100BaseT

Hubs, pontes e *switches*

Hubs

O modo mais simples para interconectar computadores numa rede local é através de um *hub*. Um ***hub***, ou **concentrador**, é um dispositivo que simplesmente pega os bits dos quadros de uma porta de entrada e retransmite às portas de saída.

> ***Hubs*** são essencialmente repetidores e operam sobre os bits, atuando portanto ao nível da **camada física**.

Pontes

Uma **ponte** (*bridge*) é um dispositivo eletrônico que permite que várias redes locais sejam concatenadas. Cada ponte conecta dois segmentos de rede e faz com que uma cópia de cada quadro que chega a um segmento seja transmitida ao outro segmento. Deste modo, os dois segmentos da rede local operam como se fosse uma rede única.

> Diferentemente do *hub*, uma **ponte** manipula quadros completos, atuando portanto ao nível da **camada enlace**.

Switches

Um ***switch*** (ou comutador) é um dispositivo eletrônico capaz de comutar o tráfego de uma LAN, diminuindo o espaço de conflitos no acesso ao meio comum. Fisicamente um *switch* assemelha-se a um *hub*; a diferença surge do modo como os dois dispositivos operam: o *hub* simula um meio único compartilhado por todos os computadores, enquanto o *switch* segmenta a rede local, onde cada computador tem um segmento para si próprio.

Questões

1. O que é um **protocolo de enlace** e quais suas principais funções dentro da arquitetura Internet?
2. Quais os possíveis **serviços** que a **camada enlace** pode oferecer a camada rede? Estes serviços tem correspondência com serviços oferecidos pelo IP ou mesmo TCP?
3. Se todos os enlaces da Internet oferecessem um serviço de entrega confiável, o serviço de entrega confiável do TCP seria redundante? Explique.
4. Qual a função das **placas adaptadoras** de rede?

5. Quais são, em geral, as causas de **erros** nos dados transmitidos através da rede?
6. Explique o que é **checagem de paridade**. Mostre um exemplo prático.
7. Suponha que uma informação contida em um pacote tem o padrão de bits 1010101010101011 e que um esquema de paridade é utilizado. Qual será o valor do campo de paridade?
8. Pesquise sobre a **aritmética módulo 2** utilizada nos códigos CRC. Mostre um exemplo prático de como se processam as quatro operações (soma, subtração, multiplicação e divisão) neste sistema.
9. Diferencie um **enlace ponto-a-ponto** de um **enlace multiponto**, citando exemplo de tecnologias existentes.
10. Qual a principal situação onde é utilizado o protocolo **PPP**? Exemplifique citando equipamentos necessários e velocidade de transmissão.
30. Pesquise e descreva as **técnicas de multiplexação** utilizadas para partilhar um meio físico de um enlace em múltiplos canais, como o **FMD** e **TDM**. Cite exemplos.
11. Pesquise a história e o funcionamento do protocolo **ALHOA**, descrevendo-o em detalhes. Ache alguma URL sobre o assunto.
12. O que o protocolo **CSMA-CD** tem de parecido e de diferente com o protocolo **ALOHA**? Porque no ALOHA não é possível implementar a detecção de colisões que existe no CSMA-CD?
13. Pesquise os diferentes **padrões IEEE 802** para redes locais. Cite exemplos de tecnologias de uso corrente que usam estes padrões.
14. Explique o papel dos **endereços físicos** nas redes locais. Como estes endereços são configurados nos adaptadores de rede.
15. Caracterize as **redes locais Ethernet**, precisando os protocolos utilizados, forma de endereçamento físico e tecnologias existentes.
16. Na tecnologia **Ethernet 10BaseT** o canal é compartilhado entre os usuários. Como fica a velocidade de acesso para cada usuário?
17. Cite alguns **meios físicos** que podem ser utilizados pelas tecnologias **Ethernet**, relacionando cada tecnologia com o respectivo meio.
18. Descreva em detalhes a tecnologia **Ethernet 10/100BaseT**, descrevendo suas características e equipamentos necessários para instalação.
19. O que é um ***hub***?
20. O que é uma **ponte**?
21. O que é um ***switch***? Qual a diferença em relação a um *hub*?
22. Pesquise na Internet endereços URL de fabricantes que ofereçam produtos como ***hubs,*** **pontes,** ***switches*** e **roteadores**, fazendo uma lista de produtos de diferentes fabricantes com suas características.
23. Um roteador pode ser um dispositivo especializado, fabricado com esta função, ou implementado em um computador. Pesquise como posso implementar um **roteador** em um computador, citando as necessidades em termos de hardware e software.

Glossário

Termo	Definição	Referências no texto
ADSL	(*Asymmetric Digital Subscriber Line*) Sistema que possibilita transmissão de sinais em banda larga nos cabos telefônicos metálicos. O ADSL usa multiplexação por divisão de freqüência (FDM) para dividir o enlace de comunicação entre o usuário e o provedor em três faixas de freqüências: uma para envio de dados, uma para receber dados e outra para voz. 12	12
Apache	Aplicativo que implementa o lado servidor da aplicação WWW em máquinas servidoras com sistema operacional Linux..	25
ARP	(*Address Resolution Protocol*) Protocolo que permite de forma dinâmica realizar o mapeamento do endereço de físico a partir do endereço IP (ver pág. 49)	51
ATM	(*Asynchronous Transfer Mode*) Técnica de transferência de dados baseada em células fixas de 53 bytes que permite a comunicação de dados digitais em alta velocidade e grandes volumes. A técnica ATM se aplica ao transporte, a multiplexação e a comutação de informações que chegam na forma de pacotes denominados células ATM. O princípio fundamental dessa técnica consiste na segmentação do fluxo de informações de diversos tipos (contínuo ou descontínuo) em uma seqüência de células elementares para serem transmitidas e comutadas.	10
Backbone	(Espinha dorsal) Normalmente utilizado para se referir ao enlace principal ou de alta densidade de tráfego em uma rede de telecomunicações, geralmente transporta um grande volume de tráfego.	65
Best effort	(melhor esforço) Utilizado para caracterizar redes de datagramas, baseadas em serviços sem conexão e não garantido, como a camada rede da Internet.	9, 34
BGP	(*Border Gateway Protocol*) Protocolo de Roteamento de Gateway Externo, bastante utilizado na Internet.	45
B-ISDN	(Broadband Integrated Services Digital Network).	*RDSI*-FL
BOOTP	(*Bootstrap Protocol*). Protocolo que pode ser implementado como um programa de aplicação para encontrar endereços físicos a partir de endereços IP.	51
Broadcast	(difusão) Modo de transmissão de um sinal sobre um determinado meio a ser recebido por dois ou mais elementos de recepção. Normalmente utilizado para caracterizar sistemas de radiodifusão. Na área de redes também é utilizado para caracterizar os protocolos de acesso ao meio em redes locais do tipo barramento.	51, 61
Buffer	Memória temporária, normalmente utilizada para armazenar dados que estão em espera pela liberação de um recurso, como por exemplo, serem transmitidas em um enlace que está ocupado.	36, 37, 43
Cable Moden	Modem que utiliza a rede cabos coaxiais para transferir informações em alta velocidade.	12
Cache	Memória temporária, normalmente utilizada para armazenar informações de uso recursivo, evitando nova consulta à fonte original da informação.	51, 54
Checksum	Soma de verificação utilizada para detecção de erros em transmissão de dados.	37
CIDR	(*Classes Interdomain Routing*) Padrão que permite as organizações obterem um identificador de rede com qualquer tamanho.	48
Circuito Virtual	Técnica de roteamento em redes de comutação de pacotes, onde a rota dos pacotes é estabelecida a priori, numa fase chamada de estabelecimento de circuito virtual.	10

Conection reply	Resposta de um pedido de conexão.	6
Conection request	Requisição ou pedido de conexão.	6
Conectionless	(Sem conexão) Normalmente se refere a protocolos onde não há procedimento de estabelecimento de conexão antes de iniciar uma transmissão de dados.	17, 36
CRC	(*Cyclic Redundancy Check*) Método de detecção de erros utilizado em vários protocolos de comunicação de dados. Emprega um algoritmo matemático onde são adicionados bits de redundância no mesmo pacote. O receptor usa o mesmo algoritmo para recalcular os bits de redundância e compara este resultado com o valor recebido. Se as duas seqüências forem iguais o pacote é considerado livre de erro.	59
Datagrama	Pacotes de dados, normalmente utilizado para se referir ao pacote de dados transportado pelo protocolo IP.	17, 45
DHCP	(*Dynamic Host Configuration Protocol*) Protocolo que permite a alocação dinâmica de endereços IP.	52
DNS	(*Domain Name System*) Banco de dados da internet usado para converter os nomes dos domínios em endereços IP.	53
Download	Transferência de arquivo de um computador remoto para um computador local via rede.	27
Downstream	Fluxo de dados sendo transferido em um enlace de um computador remoto a um computador local.	12
DSL	(*Digital Subscriber Line*) Tecnologia digital de transmissão de informações por meio de fios de cobre. As taxas de transferência dependem da tecnologia que se usa (por exemplo: ADSL, HDSL, SDSL). Concebida, em princípio, para aplicações em redes telefônicas.	*ADSL*
e-mail	Correio eletrônico. Sistema pelo qual um usuário de computador pode trocar mensagens com outros usuários (ou grupos de usuários) via uma rede de comunicações. O correio eletrônico é uma das aplicações mais populares da Internet.	21, 22, 29
Ethernet	Padrão para redes locais a 10 Mbps e 100 Mbps. Todos os hosts são conectados em um enlace tipo barramento e utiliza o protocolo de acesso múltiplo CSMA/CD.	58, 61, 62
Fast Ethernet	Padrão de Ethernet que opera a 100 Mbps.	64
FDM	(*frequency division multiplexing*) multiplexação por divisão da freqüência. Técnica de para separar cada canal a ser transmitido em um meio comum a partir do uso do espectro de freqüência de rádio, no caso o meio físico é o ar. Sistemas como a distribuição de rádio e TV usam esta técnica.	61
Flag	(bandeira) Campos de utilizados normalmente no início e no fim de um quadro da camada enlace para delimitá-lo. Contém o valor de 01111110 em protocolos como o PPP e o HDLC.	38, 60
Frame	(quadro).	*Quadro*
Frame-Relay	Protocolo de acesso do nível da camada enlace, que usa circuitos virtuais para transportar dados.	10
FTP	(*File Transfer Protocol*) Protocolo de transferência de arquivos, utilizado para a transferência de arquivos de um computador para outro na Internet.	16, 21, 27
Full-duplex	São conexões que permitem a transferência de dados em ambas às direções e simultaneamente.	37, 39
Gateway	Nó da rede equipado para atuar como interface com outras redes que usam protocolos diferentes. Também utilizado como referência a um roteador.	*Roteador*
Gateway default	(roteador padrão).	*Roteador padrão*

Gigabit Ethernet	Tecnologia Ethernet emergente que aumenta a velocidade de transmissão para 1 Gbps podendo ser empregada em *backbones.*	65
Half-duplex	São conexões que permitem a transferência de dados em ambas às direções, mas não simultaneamente.	58
Handshaking	Procedimento de estabelecimento de conexão.	8, 43
HDLC	(*High Level Data Link Control*) Protocolo do nível da camada enlace, utilizado em transmissão de dados orientados a bit.	60
HFC	(*Hibric Fiber Coaxial Cable*) Rede híbrida fibra ótica / cabo coaxial. Normalmente utilizada para acesso doméstico a Internet utilizando o sistema de distribuição de TV a cabo.	12
Hiperlink	Ponto de chamada em hipertextos, a partir dos quais pode-se acessar outras informações relacionadas a esta chamada.	24
Home-page	Página pessoal. Documento hipertexto, em linguagem HTML, para disponibilizar informações na Internet.	24
Host	(hospedeiro) Computador ou sistema terminal que permite que usuários se comuniquem com outros computadores em uma rede usando programas de aplicação.	4, 7
HTML	(*Hipertext Markup Language*) Linguagem de programação constituída de diretivas em código ASCII e utilizada na elaboração de documentos hipertexto e páginas pessoais da Web (*home-pages*). Para a visualização de documentos HTML usa-se um navegador Internet.	21, 24
HTTP	(*Hipertext Transfer Protocol*) Protocolo para mover arquivos hipertexto através da Internet. é o protocolo mais importante usado na aplicação WWW.	16, 25
Hub	Dispositivo que permite conectar diversos computadores (ou outros dispositivos), usualmente em uma topologia em estrela, simulando um barramento.	65
IAB	(*Internet Architecture Board*) Corpo técnico que supervisiona o desenvolvimento dos protocolos Internet. Possui duas forças-tarefa: o IETF e o IRTF.	*IETF*
ICMP	(*Internet Control and Message Protocol*) Protocolo para reportagem de erros no roteamento de datagramas IP.	52
IETF	(*Internet Engineering Task Force*) Organização aberta, composta por projetistas de rede, operadores, vendedores e pesquisadores cujo propósito é coordenar a operação, a gerência e a evolução da Internet e resolver questões de curto e médio prazos concernentes a protocolo e arquitetura. É uma fonte importante de propostas para padrões de protocolo, os quais são submetidos ao IAB para a aprovação final.	5
IIS	(*Internet Information Server*) Aplicativo que implementa o lado servidor da aplicação WWW nos sistemas Windows da Microsoft.	25
IMAP	(*Interactive Mail Acess Protocol*) Protocolo utilizado para acesso a servidores de correio eletrônico. Outro protocolo com função similar é o POP3.	31
Internet	(Inter-rede) Rede mundial de computadores surgida nos anos 60 e popularizada a partir dos anos 90. Permite que usuários de vários tipos de computadores e redes no mundo inteiro se comuniquem por meio de protocolos comuns.	4, 7, 8
Intranet	Rede interna de uma empresa, interligada segundo os protocolos da Internet. Enquanto a Internet é uma rede aberta, uma Intranet existe apenas dentro de uma organização, estando protegida do mundo exterior por *firewalls*, o que permitem que os empregados tenham acesso ao mundo externo mas, evita que outros tenham acesso a ela. Uma Intranet serve para distribuir notícias, responder perguntas dos empregados, atualizar registros funcionais, conectar funcionários em áreas distantes, etc.	*Internet*
IP	(*internet protocol*) É um dos principais protocolos da Internet, compondo a camada rede, loco abaixo da camada transporte. É um protocolo de comutação de pacotes não-orientado a conexão. O protocolo IP é responsável por estabelecer a rota pela qual	8, 17, 45

	seguirá cada pacote na malha de roteadores da Internet.	
IRTF	(*Internet Research Task Force*) É uma força tarefa do IAB para considerar questões Internet de longo prazo, do ponto de vista teórico.	*IETF*
ISDN	(*Integrated Services Digital Network*).	*RDSI*
ISO	(*International Organization for Standardization*) É uma organização internacional formada por órgãos de diversos países, tais como o ANSI (americano), o BSI (inglês), o AFNOR (francês) e a ABNT (brasileira), que estabelece padrões industriais de aceitação mundial.	15
ISP	(*Internet Service Provider*) Provedor de serviços Internet. Uma organização oferecendo e provendo serviços e conexão à Internet ao público e possuindo seus próprios servidores para prover os serviços oferecidos.	5
ITU	(*International Telecommunication Union*) Órgão da ONU responsável pelo estabelecimento de normas e padrões em telecomunicações e radiodifusão no mundo. O ITU-T é o setor da ITU responsável pela padronização em telecomunicações (antigo CCITT).	15
LAN	(*Local Area Network*) Rede Local. Ambiente de comunicação local que utiliza enlaces de múltiplos acesso em um meio compartilhado. Tipicamente construída para operar em ambiente privado.	3, 61, 62
Linux	Sistema operacional de código aberto, baseado no sistema Unix.	25
MAN	(*Metropolitan Area Network*) Redes Metropolitanas. Uma rede de dados servindo uma área mais ou menos do tamanho de uma cidade.	4
Modelo OSI	Modelo conceitual de protocolo com sete camadas, desenvolvido em conjunto pela ISSO e ITU, visando prover um conjunto de padrões para interconexão de sistemas abertos de tratamento da informação.	15
Modem	Contração de "modulador demodulador", utilizada para designar o equipamento resultante da associação de um modulador e de um demodulador. Este equipamento serve para transmitir sinais digitais através dos meios de comunicação, que são naturalmente analógicos.	11
MS-DOS	(*Microsoft Disk Operation Sistem*) Antigo sistema operacional em modo texto da Microsoft.	28
MSS	(*Maximum Segment Size*) Tamanho máximo do segmento da camada transporte.	38
MTU	(*Maximum Transfer Unit*) Tamanho máximo dos pacotes que podem ser transportados pela camada enlace.	46
Multicast	Pacote endereçado a um grupo de nós da rede.	47
OSI	(*Open System Interconnection*).	modelo OSI. *modelo OSI*
OSPF	(*Open Shortest Path First*) Protocolo de roteamento utilizado na Internet.	45
Overhead	Informações de controle que é agregada ao dado que será transmitido na forma de um cabeçalho.	36
Pacote	Seqüência de bits formada por dados do usuário precedidos por um cabeçalho de controle que permite que o pacote seja encaminhado, através da rede, para seu destino.	5, 9
PDU	(*Protocol Data Unit*) Unidade de dados de protocolo. Consiste no pacote que será transportado por uma determinada camada de protocolo, composto por dados da camada superior precedidos por um cabeçalho de controle.	34
Ping	Aplicativo que permite verificar se um *host* está disponível na rede.	53
POP3	(*Post Office Protocol 3*) Protocolo utilizado para acesso a servidores de correio eletrônico. Outro protocolo com função similar é o IMAP.	31
Porta	(*socket*) Denominação utilizada nos protocolos de transporte da Internet, no processo de multiplexação de aplicações, para diferenciar os canais utilizados por cada processo de	17, 35

	aplicação.	
PPP	(*Point to Point Protocol*) Protocolo de enlace que provê um método para transportar quadros sobre enlaces ponto-a-ponto. É bastante usado na comunicação entre dois computadores via modem e linha telefônica. Outro protocolo com função similar é o protocolo SLIP.	60
Quadro	(*frame*) Unidades de dados de protocolos da camada de enlace.	64
RARP	(*Reverse Address Resolution Protocol*) O protocolo RARP é uma adaptação do ARP e permite a uma estação descobrir o seu endereço IP a partir de um endereço físico.	51
RDSI	Rede digital de serviços integrados, ou ISDN (*Integrated Services Digital Network*). Rede de telecomunicações digital capaz de transportar indistintamente sinais integrantes de diversos serviços. Presta serviços de até 2 Mbps em que é utilizada a rede de pares metálicos até o assinante.	12, 60
RDSI-FL	Rede Digital de Serviços Integrados de Faixa Larga, ou B-ISDN (*Broadband Integrated Services Digital Network*). Rede de alta velocidade que suporta o tráfego de todo tipo de serviço (voz, dados e vídeo) a taxas acima de 2 Mbps. É uma evolução da RDSI (ISDN). O ITU-T escolheu o ATM como transporte para essa rede.	*RDSI*
RFC	(*Request for Comments*) Série de documentos que descreve a suíte de protocolos Internet. Nem todas as RFCs descrevem padrões Internet, mas todos os padrões Internet são escritos como RFCs.	5
RIP	(*Routing Information Protocol*) Algoritmo para roteamento utilizado na Internet. As tabelas do roteamento RIP são construídas dinamicamente.	45
Roteador	(*router* ou *gateway*) Dispositivo de conexão e de chaveamento entre redes. A decisão de chaveamento é baseada em informação de camada de rede e tabelas de roteamento, geralmente construídas por protocolos específicos.	5, 33, 48
Roteador Padrão	(*gateway default*) Em uma rede local, o roteador padrão é o dispositivo para onde o TCP/IP envia pacotes destinados a redes remotas.	48, 50
Router	(roteador).	*Roteador*
SDH	(*Synchronous Digital Hierarchy*) Hierarquia Digital Síncrona. Abreviatura mantida na linguagem técnica para se referir a sistemas da hierarquia digital síncrona. É um padrão de transporte de informações em redes digitais.	*SONET/SDH*
SLIP	(*Serial Line Internet Protocol*) Protocolo usado para transportar pacotes sobre linhas seriais, por exemplo, na conexão entre dois computadores via modem e linha telefônica, ou via cabo RS-232. Outro protocolo utilizado para fins similares é o PPP.	60
SMTP	(*Simple Mail Transfer Protocol*) Principal protocolo da aplicação de correio eletrônico da Internet. Define como as mensagens são trocadas entre os servidores e entre os servidores e os leitores de correio eletrônico.	16, 30
Socket	(porta).	*Porta*
SONET/ SDH	(*Syncrhonous Optical Network / Syncrhonous Digital Hierarchy*) Hierarquia Digital Síncrona sobre enlaces de fibra óptica.	60
SSH	(*Secure Shell*) Protocolo utilizado para acesso remoto a sistemas, similar ao Telnet, mas considerado mais seguro por usa criptografia na transferência das informações de autenticação.	21
Store-and-forward	Armazena e encaminha. Comumente refere-se aos roteadores da rede, onde um datagrama que chega em um enlace de entrada e armazenado e depois encaminhado a um enlace de saída.	5, 8
Switch	Dispositivo capaz de comutar o tráfego em uma rede local. Fisicamente é similar a um hub, mas na prática, reduz o domínio de colisões na rede local, aumentando sua eficiência.	65

TCP	(*Transmission Control Protocol*) É um protocolo de transporte padronizado para a interligação de redes baseadas em IP. Operando no topo do IP, é um protocolo orientado à conexão, responsável pela multiplexação de aplicações, garantindo a confiabilidade da ligação extremo a extremo, possuindo ainda controle do fluxo e congestionamento.	8, 17, 24, 37
TCP / IP	(*Transmission Control Protocol / Internet Protocol*) Conhecido como o conjunto de protocolos da Internet, que combina o TCP e o IP.	4, 21, 33
TDM	(*Time Division Multiplex*) Multiplexação por divisão em tempo. Sistema de multiplexação no qual um canal é constituído, intermitentemente, a intervalos de tempo regulares através de uma distribuição automática, a um canal comum.	61
Telnet	(emulação de terminal via rede) Protocolo padrão da Internet para o serviço de acesso remoto.	21, 22, 35, 39
Time-out	Evento que ocorre quando um dispositivo de rede espera uma resposta de outro dispositivo de rede, mas não a obtém dentro de um intervalo especificado de tempo.	40, 41
Time-to-live	(TTL) Tempo de sobrevida, indica o tempo de vida do datagrama, após o qual o mesmo é descartado.	46
Top-down	("de cima para baixo") Que inicia a partir do topo e depois vai descendo pelos níveis inferiores.	i, 53
Traceroute	Aplicativo capaz de traçar a rota que liga um *host* a outro *host* na Internet.	53
UDP	(*User Datagram Protocol*) Protocolo padrão da Internet de camada de transporte. É um protocolo não orientado à conexão e sem transferência de dados garantida.	8, 17, 24, 36
UNIX	Sistema operacional multiusuário desenvolvido na década de 1960 e ainda hoje bastante utilizado, principalmente em provedores de serviços para a Internet.	26, 28
Upload	Transferência de arquivo de um computador para um computador remoto via rede.	27
Upstream	Fluxo de dados sendo transferido em um enlace de um computador a um computador remoto.	12
URL	(*Uniform Resource Locator*) Endereço para localização e identificação de informações na *Web*. Contém informação do nome de domínio do servidor e caminho usado para especificar a localização de um documento.	24
VoIP	Voz sobre IP. Técnica de transmissão do sinal de voz do sistema telefônico sobre a Internet.	21
WAN	(*Wide Area Network*) Rede geograficamente distribuída. Rede que cobre uma grande área geográfica, podendo constituir-se de várias redes locais interligadas. A Internet é uma WAN.	4
Web	Literalmente significa teia, em geral refere-se a conhecida aplicação Internet para acessar hipertextos em formato HTML onde os usuários podem criar, editar ou ler, através de um navegador de documentos.	4, 21
WWW	(*World Wide Web*) O mesmo que *Web*. .	21, 24
X-25	Tipo de técnica orientada a conexão para comunicação de dados e pacotes.	10, 60

Referências Bibliográficas

KUROSE, J. F. e ROSS, K. W. **Computer Networking: A top down approach featuring the Internet**, Addison Wesley, 2001. (Principal referência, disponível com Prof. Cantú)

KUROSE, J. F. e ROSS, K. W. **Redes de Computadores e a Internet: Uma nova abordagem**, Addison Wesley, São Paulo, 2003. (Biblioteca CEFET-SJ - Reserva)

TANENBAUM, A. **Redes de Computadores**, 3º Edição, Campus, 1996. (Biblioteca CEFET-SJ)

SOARES, Luiz Fernando Gomes; LEMOS, Guido; COLCHER, Ségio. **Redes de Computadores – Das LANs, MANs e WANs às Redes ATM**, Editora Campos, Rio de Janeiro, 1995. (Biblioteca CEFET-SJ)

COMER, D. E. **Interligação em rede com TCP/IP, Vol 1 – Princípios, protocolos and arquitetura**, trad. da 3º Edição, Campus, 1998. (Biblioteca CEFET-SJ)

COMER, D. E. **Redes de Computadores e Internet**, 2º Edição, Bookman, 2001. (Biblioteca CEFET-SJ - Reserva)

PETERSON, L. L. e DAVIE, B. S. **Computer Networks: A systems approach**, Morgan Kaufmann, 2nd Edition, 2000. (Disponível com Prof. Cantú)