REDES DE COMPUTADORES: ARQUITETURA TCP/IP

Eleri Cardozo
Departamento de Engenharia de Computação
e Automação Industrial
Faculdade de Engenharia Elétrica e de Computação
Universidade Estadual de Campinas

1994

Conteúdo

1 INTRODUÇÃO **4**
- 1.1 Conceitos Básicos 4
- 1.2 Aplicações Típicas 5
- 1.3 Estruturas de Redes 6
- 1.4 Padronização de Redes 6
- 1.5 Estruturação de Redes em Camadas 9
- 1.6 O modelo OSI 9
  - 1.6.1 A Camada Física 10
  - 1.6.2 A Camada de Enlace 10
  - 1.6.3 A Camada de Rede 12
  - 1.6.4 A Camada de Transporte 12
  - 1.6.5 A Camada de Sessão 12
  - 1.6.6 A Camada de Apresentação 13
  - 1.6.7 A Camada de Aplicação 13
- 1.7 A Arquitetura Internet 13
  - 1.7.1 A Camada Interface de Rede 13
  - 1.7.2 A Camada Inter-Redes 14
  - 1.7.3 A Camada de Transporte 14
  - 1.7.4 A Camada de Aplicação 15

2 A CAMADA INTERFACE DE REDE **16**
- 2.1 Meios de Transmissão 16
  - 2.1.1 Par Trançado Metálico 16
  - 2.1.2 Cabo Coaxial 16
  - 2.1.3 Fibra Ótica 17
- 2.2 Padrões Físicos da Camada Interface de Rede 18
  - 2.2.1 Redes Locais 18
  - 2.2.2 Redes Públicas 19
- 2.3 Padrões de Enlace e Acesso ao Meio 19
  - 2.3.1 Métodos de Acesso ao Meio 20
  - 2.3.2 Padrões de Enlace 21
- 2.4 Endereçamento Internet 26
  - 2.4.1 Endereço IP 26
  - 2.4.2 Subredes 28

2.4.3 Endereços Físicos 30
2.4.4 Mapeamento Endereço IP - Endereço Físico: Protocolo ARP 31
2.4.5 Mapeamento Endereço Físico - Endereço IP: Protocolo RARP 32

3 A CAMADA INTER-REDES 33
3.1 Funcionalidades da Camada de Rede (OSI) 33
3.1.1 Entrega de Pacotes 33
3.1.2 Roteamento 34
3.1.3 Controle de Congestionamento 35
3.1.4 Interconexão de Redes 36
3.2 A Camada Inter-redes (TCP/IP) 38
3.2.1 O Papel das Comportas na Interconexão de Redes 39
3.3 Padrões da Camada Inter-Redes 41
3.3.1 Transporte de Datagramas: Protocolo IP 41
3.3.2 Mensagens de Erro e Controle: Protocolo ICMP 43
3.3.3 Roteamento Externo: Protocolo EGP 45
3.3.4 Roteamento Interno: Protocolo RIP 46

4 A CAMADA DE TRANSPORTE 49
4.1 Conceitos Básicos 49
4.1.1 O Serviço de Transporte 49
4.1.2 Classes de Protocolos de Transporte 51
4.1.3 Protocolos de Transporte Classe 4 51
4.2 Padrões da Camada de Transporte: Protocolo TCP/IP 57
4.2.1 O Conceito de Port 57
4.2.2 O Conceito de Dados Urgentes 58
4.2.3 Cabeçalho do Protocolo TCP/IP 59
4.2.4 Temporização e Controle de Congestionamento no Protocolo TCP/IP 61
4.2.5 TCP/IP Sobre Redes de Alto Produto Banda*Atraso 63
4.3 Padrões da Camada de Transporte: Protocolo UDP 64

5 A CAMADA DE APLICAÇÃO 66
5.1 Terminal Remoto 67
5.2 Manipulação de Arquivos 68
5.2.1 Acesso Interativo: Protocolo FTP 68
5.2.2 Acesso On-line: Protocolo NFS 69
5.3 Correio Eletrônico 71
5.3.1 O Protocolo SMTP 71
5.4 Gerenciamento de Redes 73
5.4.1 Protocolo SNMP 75
5.4.2 Protocolo CMOT 76
5.5 Hipertexto/Hipermídia 77
5.5.1 O Protocolo HTTP 77
5.6 Resolução de Nomes 78
5.6.1 O Conceito de Domínio 79

5.6.2 Mapeamento Nome Simbólico - Endereço IP 81
5.6.3 Arquitetura do DNS 83
5.6.4 Mapeamento Inverso 84

Chapter 1

INTRODUÇÃO

1.1 Conceitos Básicos

Definiremos *Rede de Computadores* como um conjunto de computadores *autônomos* e *interconectados*. O termo *autônomo* exclui arranjos de processadores que apresentam relação mestre/escravo ou dispõem de um controle centralizado como os multiprocessadores, as máquinas *data flow* e os *array processors*. Numa rede, nenhum computador obedece a comandos de outro, possuindo inclusive autonomia para se desconectar da rede.

Os meios de interconexão são muitos: cabos de cobre, fibras óticas, rotas de microondas, radiodifusão, etc. Atualmente, os cabos de cobre (coaxiais e pares trançados) são os mais empregados, devendo a fibra ótica assumir este papel num futuro próximo. Os meios de interconexão limitam tanto a taxa de transmissão de informação quanto a extensão geográfica da rede. Quanto a sua extensão geográfica, as redes se classificam em:

1. Redes Locais (LAN: Local Area Network): interconectam computadores localizados numa mesma sala ou edifício (10 m - 1 Km). Tipicamente, um único meio de transmissão é empregado.

2. Redes de Campus (CAN: Campus Area Network): interconectam computadores a nível de campus (fábrica, universidade, etc.) em extensões não superiores a 10 Km. Tipicamente são compostas de várias LANs interligadas por uma rede de alto desempenho *(backbone)*.

3. Redes Metropolitanas (MAN: Metropolitan Area Network): interconectam computadores de uma mesma corporação a nível regional (5 - 100 Km), usualmente empregando linhas telefônicas alugadas de uma mesma operadora.

4. Redes de Longa Distância (WAN: Wide Area Network): interconectam computadores a nível nacional ou continental (100 - 5000 Km). Via de regra são operadas por *holdings* nacionais de telecomunicações.

Uma rede é dita *homogênea* se todos os computadores por ela interconectados são idênticos. Caso contrário, temos uma rede *heterogênea*. Obviamente, redes heterogêneas demandam padronização tanto no nível de hardware (tensões, frequências, etc.) quanto no nível de software (por exemplo, representação de dados e formatação de mensagens).

O objetivo central de uma rede de computadores é o compartilhamento de informação e recursos. Outros benefícios importantes são:

- o crescimento gradual da capacidade de processamento da informação;
- a diversidade de equipamentos e a liberdade de escolha;
- o aumento da confiabilidade (via redundância);
- o processamento da informação *in loco*;
- um meio alternativo de comunicação social.

1.2 Aplicações Típicas

A tecnologia de redes de computadores teve profundo impacto nas atividades relacionadas ao ensino/pesquisa, à produção e serviços, e à administração. No ensino/pesquisa, a interligação de bibliotecas, o correio eletrônico e os serviços de boletim eletrônico aumentam a velocidade de disseminação do conhecimento.

Nas atividades relacionadas à produção, as redes locais suportam a automação da manufatura, os sistemas distribuídos de controle digital (SDCD) e a manufatura integrada. As modernas tecnologias de fabricação demandam uma capacidade de processamento de informação em diversos níveis (desde o controle de sensores e manipuladores até a emissão de faturas) capaz de ser obtida apenas com a interconexão de processadores diversificados (PCs, estações de trabalho, computadores de processo, etc.).

No campo da administração, a automação de escritórios é o melhor exemplo do emprego de redes de computadores. Anterior à popularização do computador pessoal, poucos eram os funcionários administrativos capazes de operar um terminal conectado a *mainframe*. Hoje, nas grandes empresas, com a disseminação dos computadores pessoais (conectados via rede local), documentos são gerados, transmitidos e armazenados sem a necessidade de manipulação de papéis, envelopes, carimbos, etiquetas, etc. A substituição do papel pela mídia eletrônica teve profundo impacto na racionalização dos custos administrativos.

Na década de setenta, as empresas de telecomunicação passaram a oferecer serviços de comunicação de dados, utilizando, muitas vezes, as próprias linhas de voz existentes. Este serviço, no Brasil, é oferecido pela Embratel através da Rede Nacional de Comutação de Pacotes (RENPAC). A comunicação de dados permite a interconexão em longa distância de computadores. Se o canal de comunicação for de banda larga, a informação poderá vir acompanhada de sinais de vídeo e áudio (digitalizados e compactados). Inicia-se assim a era da multimídia, abrindo-se novas fronteiras no emprego do computador como veículo de comunicação e interação humana. Pessoas localizadas em diferentes cidades ou países poderão interagir em sessões de CAD[1] cooperativo, Teleconferência, Telemedicina, etc.

[1] Computer-Aided Design.

1.3 Estruturas de Redes

Um computador conectado à rede é denominado *Host* ou *End System* (ES). Hosts são conectados por uma *subrede de comunicação*. Subredes carregam *mensagens*[2] de um host para outro. Tipicamente, em redes locais, a subrede de comunicação se reduz a um duto elétrico ou ótico. Em redes de longa distância, a subrede de comunicação é composta de *linhas de transmissão* (ou *canais*) e *dispositivos de chaveamento* denominados IMPs (Interface Message Processors) ou ISs (Intermediate Systems). IMPs são computadores especializados que conectam duas ou mais linhas de transmissão e aos quais os hosts se conectam (figura 1.1).

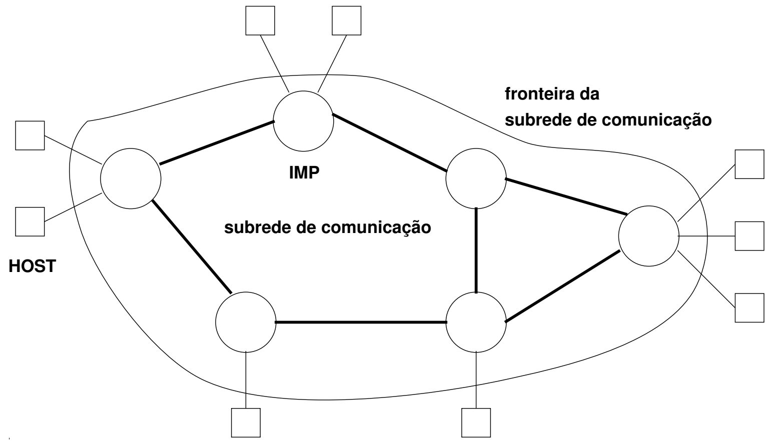

Figura 1.1: Hosts e IMPs numa subrede de comunicacão.

Subredes de comunicação se dividem em dois grupos: ponto-a-ponto[3] e de difusão *(broadcast)*. Em subredes ponto-a-ponto os IMPs são conectados por linhas de transmissão, de sorte que apenas IMPs diretamente conectados se comunicam. Se uma mensagem necessita ser transmitida entre dois IMPs não conectados, a mesma deve ser roteada através de outros IMPs. A figura 1.2 mostra as topologias típicas de subredes ponto-a-ponto.

Em subredes de difusão todos os hosts compartilham uma mesma linha de transmissão. Mensagens enviadas por um host são recebidas por todos os demais. Se o endereço de destino contido na mensagem for diferente do endereço do host que a recebeu, a mensagem é simplesmente descartada. A figura 1.3 mostra as topologias típicas de subredes de difusão.

1.4 Padronização de Redes

Um *padrão* é um conjunto de normas e procedimentos. O cumprimento destas normas e procedimentos pode ser obrigatório (normalmente quando relacionados à segurança do homem) ou recomendável (normalmente quando relacionados à qualidade de produtos e serviços). Padrões visam homogeneizar produtos e serviços num nível aceitável de qualidade

[2]Também denominadas *pacotes*.
[3]Também denominadas *comutação de pacotes*.

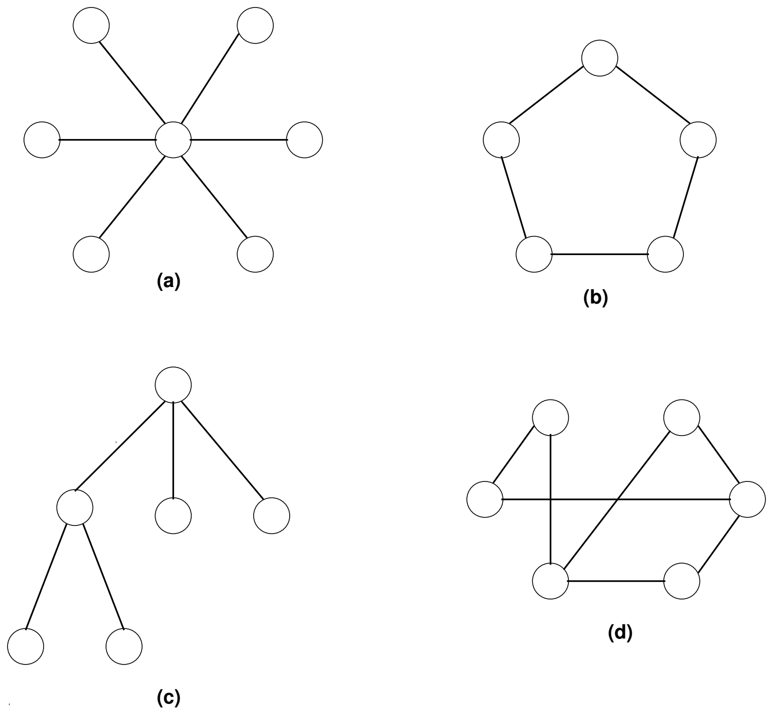

Figura 1.2: Topologias típicas em subredes ponto-a-ponto: (a) Estrela (b) Anel, (c) Árvore, (d) Genérica.

e segurança, minimizar investimentos em estoques, compatibilizar equipamentos de diferentes procedências, etc.

Um padrão é dito *de facto* quando foi adotado sem nenhuma ação de entidade reguladora. Exemplo: IBM-PC. Por outro lado, padrões *de jure* são produzidos por entidades reguladoras, nacionais ou internacionais, governamentais ou não. Exemplo: ISO-9000.

Cada país industrializado ou semi-industrializado possui uma entidade de padronização. As mais conhecidas no ramo da engenharia elétrica são:

- Brasil: Associação Brasileira de Normas Técnicas (ABNT);
- EUA: American National Standard Institute (ANSI) e Institute of Electrical and Electronic Engineers (IEEE);
- Alemanha: Deutsche Industrie-Norm (DIN);
- Inglaterra: Britsh Standard Institution (BSI).

Na área de redes de computadores os padrões *de jure* são estabelecidos por duas entidades: CCITT (Comité Consultatif International de Télégraphique et Téléphonique), que congrega as companhias de telecomunicações nacionais; e a ISO (International Standard Organization), que congrega as entidades de padronização nacionais.

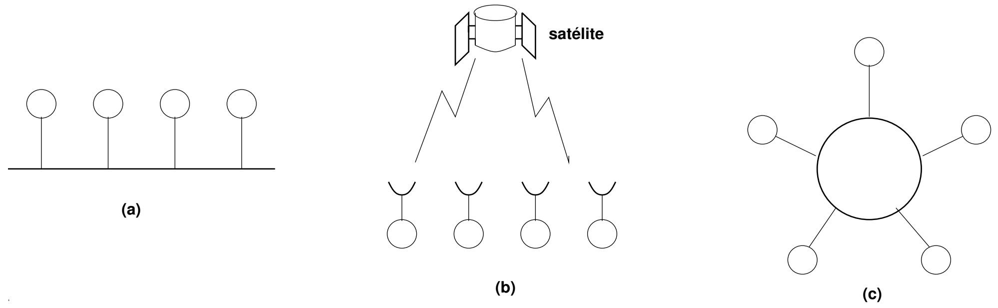

Figura 1.3: Topologias típicas em subredes de difusão: (a) Barramento (b) Radiodifusão via satélite, (c) Anel.

A ISO tem aceito padrões já estabelecidos por outras entidades (principalmente ANSI, IEEE e CCITT) como padrões internacionais, simplesmente redigindo-os e catalogando-os de acordo com os seus critérios. Por exemplo, o padrão IEEE 802 para redes locais (Ethernet, Token Bus e Token Ring) foi adotado integralmente pela ISO (ISO 8802).

Padrões do CCITT normalmente se referem à transmissão de dados a longas distâncias, enquanto padões ISO são mais voltados aos serviços que uma rede geralmente provê e a protocolos de conversação inter-hosts. A grosso modo, pode-se afirmar que padrões CCITT situam-se mais próximos do hardware que os padrões ISO.

A ISO padronizou um modelo de referência para a *Interconexão de Sistemas Abertos* (OSI[4]) e conhecido como modelo OSI/ISO ou simplesmente modelo OSI. Este modelo estipula que uma rede de computadores deve ser estruturada em sete camadas, propondo um ou mais padrões para controlar o funcionamento de cada camada. Os padrões OSI estão ainda em vias de se tornarem padrões *de facto*.

Atualmente, os padrões *de facto* são os chamados padrões Internet[5]. Criados pelo Departamento de Defesa (DoD) dos EUA para a interconexão de seus computadores no final da década de 70, têm sido adotado por todos os fabricantes daquele país para atender às normas de contrato impostas pelo DoD.

O que nos leva a crer que os padrões ISO suplantarão os padrões Internet no futuro? A resposta baseia-se no fato dos padrões OSI cobrirem uma gama maior de serviços, desde aqueles manipulados diretamente pelo usuário (submissão de *jobs* remotos, correio eletrônico, terminal virtual, etc.) até aqueles de interfaceamento com o hardware, passando por compressão de dados e criptografia. Mais resumidamente, a ISO possui padrões cobrindo praticamente todo o espectro da tecnologia de redes, o que não ocorre com os padrões Internet.

Os padrões Internet enfatizam mais o transporte confiável de dados de um host para outro. Inicialmente, apenas três serviços são padronizados no nível de usuário: transferência de arquivos, correio eletrônico e *login* remoto. Outros serviços que utilizam o transporte de dados Internet foram introduzidos pela comunidade de usuários ou por fabricantes. É o caso do Yellow Pages (diretório), RPC (chamada de procedimento remotos) e NFS (sistema

[4]Open Systems Interconnection.

[5]Abreviação de *interconnected networks*.

de compartilhamento de arquivos). Muitos destes serviços foram desenvolvidos pela Sun Microsystems para uso próprio e se tornaram padrões *de facto*.

1.5 Estruturação de Redes em Camadas

Como todo sistema complexo, redes de computadores são normalmente modeladas como blocos funcionais interligados. Estes blocos são referidos na literatura como *camadas*, sendo associado a cada camada um *nível*. A idéia é que cada camada ofereça serviços à camada superior, escondendo os detalhes de como estes serviços são implementados.

Logicamente, a camada N de um host troca informação com a camada N dos outros hosts. As regras envolvidas na conversação formam o *protocolo* da camada N. A especificação de protocolos não trata de detalhes de como são implementados, detalhes estes que dependem da tecnologia empregada. Entre cada par de camadas adjacentes existe uma *interface*, que define claramente as regras de utilização dos serviços oferecidos pela camada inferior. O conjunto de camadas e protocolos definem a *arquitetura* da rede.

Considere uma rede de cinco camadas como mostra a figura 1.4.

Suponha que uma mensagem é gerada por um processo em um host e direcionada para um segundo processo num outro host. Vamos seguir esta mensagem poela rede da figura 1.4. Para processos do usuário, o ponto de entrada na rede é a camada 5. A camada 5 transforma os dados da mensagem numa representação padrão, independente da arquitetura do processador, e submete a mensagem transformada à camada 4. Na camada 4 a mensagem é particionada em unidades menores, sendo a cada unidade adicionado um cabeçalho contendo informações de controle, tais como número de sequência. As unidades geradas na camada 4 são, uma a uma, conduzidas à camada 3.

A camada 3 decide o caminho que as unidades irão percorrer (roteamento). Informações tais como identificação dos hosts emissor e destinatário são contidas num outro cabeçalho adicionado pela camada 3. Na camada 2 é computado um código para fins de detecção de erros *(checksum)*. Esta camada adiciona tanto um novo cabeçalho quanto um rótulo demarcador de final nas unidades oriundas da camada 3. A camada 2 envia suas unidades para a camada 1, armazenando-as caso o protocolo desta camada exija confirmação de recepção. Na camada 1 serão gerados os sinais elétricos (ou óticos, ou eletromagnéticos, dependendo do meio físico de transmissão) que farão as unidades atingirem o host destinatário.

Ao receber (na camada 1) as unidades transmitidas, o host destinatário propaga-as para as camadas superiores, até atingir a camada 5, quando o processo destinatário será notificado. No caminho inverso, as unidades são remontadas de acordo com as informações contidas nos cabeçalhos: a mensagem atinge o processo receptor na forma em que foi enviada pelo processo emissor.

1.6 O modelo OSI

O modelo OSI é composto das 7 camadas apresentadas na figura 1.5. O modelo OSI não é uma arquitetura, posto que não especifica os protocolos empregado pelas camadas. Entretanto, a ISO tem produzido protocolos para as 7 camadas, publicados como padrões

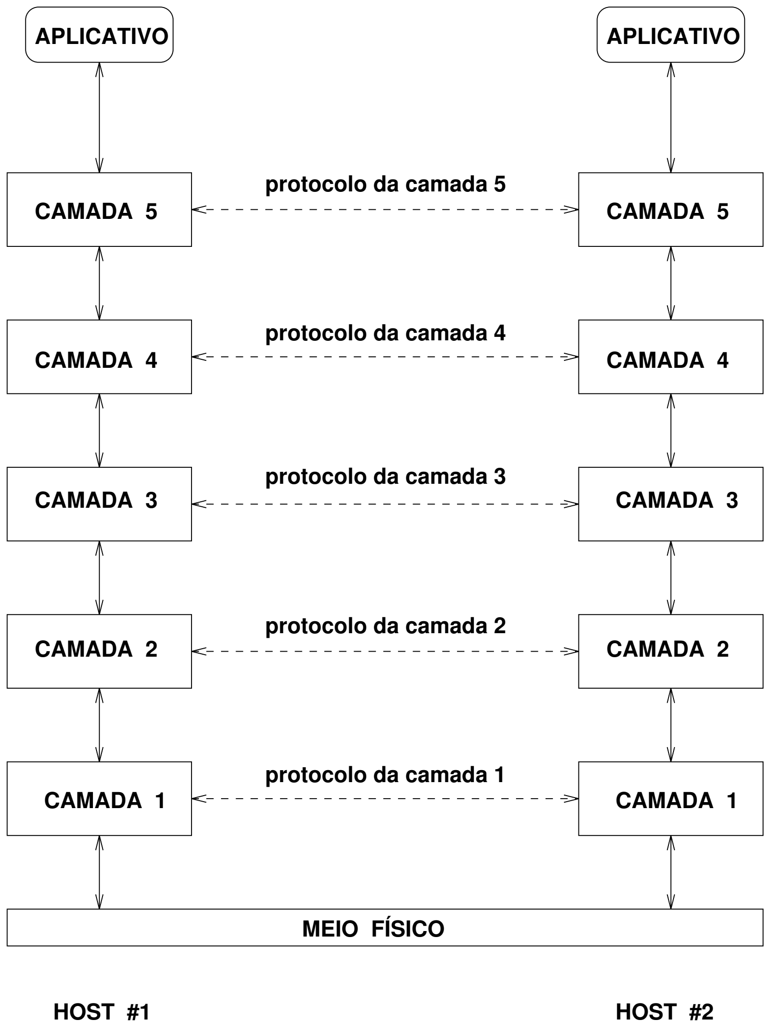

Figura 1.4: Camadas, protocolos e interfaces numa rede de computadores.

internacionais.

1.6.1 A Camada Física

A camada física é a responsável pela geração dos sinais elétricos, óticos ou eletromagnéticos que serão propagados pelo meio físico. Protocolos nesta camada especificam qual a duração e intensidade do sinal, técnica de multiplexação, pinagem, etc. Obviamente esta camada está intimamente relacionada ao meio físico empregado.

1.6.2 A Camada de Enlace

A camada de enlace utiliza a camada física para a transmissão de *quadros de dados (data frames)*. Tipicamente um quadro de dados é composto de algumas centenas de bytes.

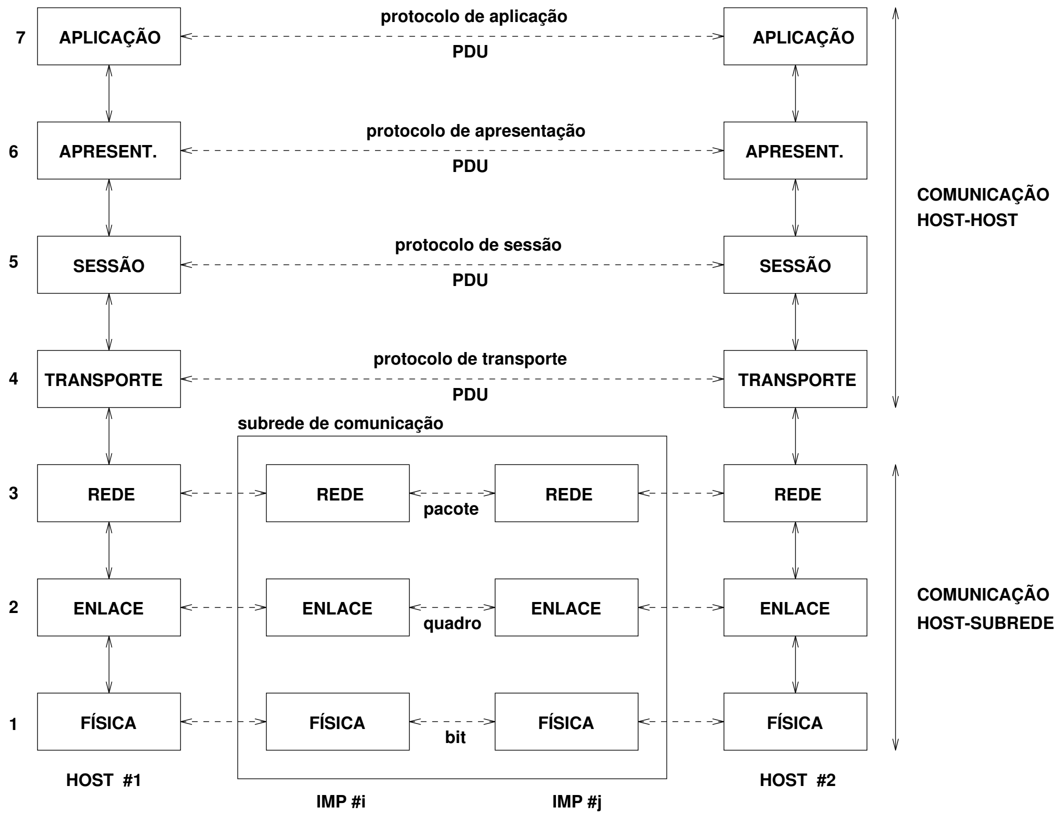

Figura 1.5: O modelo OSI.

Quadros são delimitados por sequências pré-estabelecidas de bits. A camada de enlace transmite (recebe) quadros de dados, aguardando (enviando) o respectivo quadro de reconhecimento de recepção. A transmissão de quadros, mesmo com reconhecimento de recepção, não é confiável. Quadros podem ser duplicados ou chegar fora de ordem. A duplicação ocorre quando o quadro de reconhecimento é deformado, tornando-se ininteligível pelo receptor. O receptor, neste caso, envia novamente o quadro de dados correspondente por falta de reconhecimento, gerando assim a sua duplicação.

Quadros podem ter sua ordem alterada quando são roteados por vários IMPs até atingir o host de destino. Se um quadro for deformado num ponto de seu roteamento (necessitando retransmissão), outros que vinham atrás podem "passar à frente" deste, gerando inversão de ordem no host de destino.

A camada de enlace também controla o fluxo de quadros, evitando que um host envie quadros numa taxa superior a que o receptor é capaz de processar.

1.6.3 A Camada de Rede

A camada de rede controla a operação da subrede. Uma de suas funções é o roteamento de *pacotes*[6] do host de origem ao host de destino. O roteamento pode apresentar características dinâmicas, evitando gargalos em certos IMPs, ou estáticas, empregando-se sempre a mesma rota entre dois hosts.

Outra função desta camada é a conversão de pacotes de uma subrede para outra. Por exemplo, um pacote gerado por um host numa subrede Ethernet pode ser destinado a um host numa subrede Token Ring. Como Ethernet e Token Ring empregam diferentes formatos dos pacotes, a devida conversão se faz necessária.

Em subredes de difusão esta camada é extremamente simples, dado que sua principal atribuição (roteamento) é inexistente nestas subredes.

1.6.4 A Camada de Transporte

A função principal da camada de transporte é receber dados da camada de sessão, particionar estes dados em unidades menores e, em certos casos, garantir que estas unidades cheguem a seu destino sem duplicação e na ordem correta.

Esta camada possui tipicamente dois tipos de serviços: um serviço rápido onde mensagens são limitadas em tamanho e não existe garantia de entrega, ordem ou ausência de duplicação; e um serviço mais lento, porém altamente confiável e sem limites de tamanho nas mensagens. Um terceiro serviço, difusão de mensagens para todos os hosts da subrede, pode estar disponível nesta camada.

No caso de serviço com entrega confiável, a camada de transporte é responsável pela remontagem dos quadros oriundos da camada de rede, respeitando a ordem em que foram enviados e descartando duplicações.

É função também desta camada o controle do fluxo de dados entre dois processos comunicantes (a camada de rede controla o fluxo apenas entre IMPs).

As camadas anteriores (física, de enlace e de rede) são empregadas na comunicação IMP-IMP. A camada de transporte é a primeira a promover comunicação host-host (ver figura 1.5).

1.6.5 A Camada de Sessão

Esta camada permite dois processos de aplicação (APs) estabelecerem sessões entre si a fim de organizar e sincronizar a troca de informação. Para tal, uma *conexão de sessão* é estabelecida, definindo-se as regras de diálogo entre os dois APs. Existem três variantes de diálogo quanto ao sentido do fluxo de dados: TWS (Two Way Simultaneous): bidirecional simultaneamente, TWA (Two Way Alternate): bidirecional alternadamente (um por vez), e OW (One Way): unidirecional.

[6] A taxa de utilização de uma subrede é medida pelo fluxo de pacotes por unidade de tempo.

1.6.6 A Camada de Apresentação

Esta camada fornece serviços de representação canônica de dados, compressão de dados e criptografia. Uma representação canônica dos dados se faz necessária quando hosts de arquiteturas diferentes devem se comunicar. Por exemplo, a representação de números em ponto flutuante varia de arquitetura para arquitetura. Quando um *float* é transmitido, o mesmo é convertido para uma representação padronizada, enviado via rede, e reconvertido na representação adotada pelo host de destino. A camada de apresentação dispõe de um protocolo para tal (EDR: External Data Representation).

Compressão de dados é útil para o envio de grandes massas de dados como imagens e textos. Criptografia é utilizada quando os dados a serem transmitidos são confidenciais e visa evitar sua interceptação em trânsito por pessoas não autorizadas. Protocolos para compressão e criptografia de dados também são definidos nesta camada.

1.6.7 A Camada de Aplicação

Esta camada dispõe de serviços comumente utilizados por usuários de redes. Correio eletrônico, transferência de arquivos, *login* remoto, serviços de diretório e submissão de *jobs* remotos são exemplos destes serviços.

Esta camada também se constitui no ponto de acesso à rede por processos de aplicação (APs). Estão em vias de padronização as chamadas APIs (Application Program Interfaces), que são bibliotecas de funções para envio/recepção de mensagens, estabelecimento de conexões, etc.

1.7 A Arquitetura Internet

A arquitetura Internet (ou arquitetura TCP/IP) é composta de apenas quatro camadas: interface de rede, inter-redes, transporte e aplicação. A figura 1.6 relaciona as camadas da arquitetura Internet com as correspondentes do modelo OSI. Internet é uma arquitetura porque especifica protocolos para cada uma de suas camadas (o que não ocorre com o modelo OSI).

1.7.1 A Camada Interface de Rede

Esta camada corresponde às camadas Física e de Enlace do modelo OSI. A interface de rede pode operar sobre uma rede local ou uma rede de longa distância. No primeiro caso, a interface de rede é uma placa que implementa um protocolo de enlace e de acesso ao meio, por exemplo, uma placa Ethernet operando segundo o padrão IEEE 802.3. No segundo caso, a interface de rede é um subsistema mais complexo que implementa um protocolo de conexão física host-IMP (por exemplo o X.21) e um protocolo de enlace IMP-IMP (por exemplo o protocolo HDLC: High-level Data Link Control).

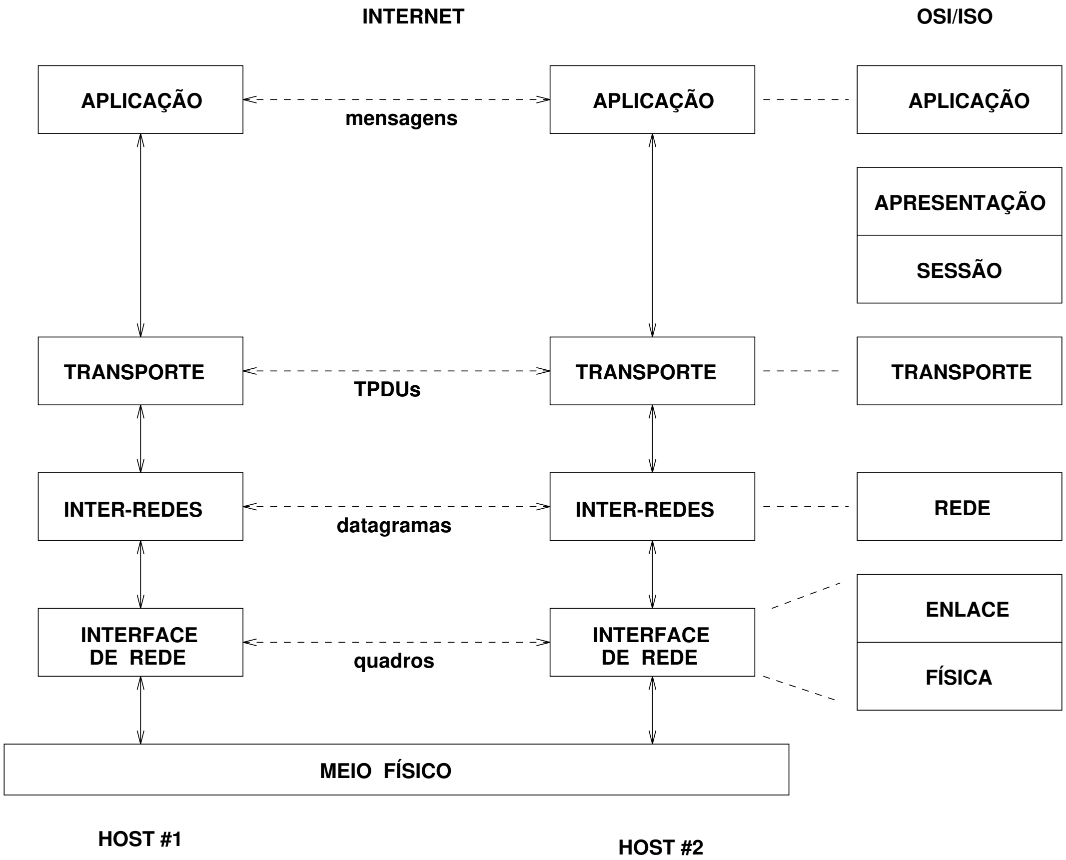

Figura 1.6: A arquitetura Internet e sua comparação com o modelo OSI/ISO.

1.7.2 A Camada Inter-Redes

É equivalente à camada de rede do modelo OSI. Esta camada define protocolos para:

1. Transporte não confiável de mensagens: o protocolo IP (Internet Protocol).
2. Controle da comunicação e informe de erros: o protocolo ICMP (Internet Control Message Protocol).
3. Roteamento de mensagens: protocolos GGP (Gateway-to-Gateway Protocol), RIP (Routing Information Protocol), etc.

1.7.3 A Camada de Transporte

É equivalente à camada 4 do modelo OSI. Esta camada define dois protocolos: TCP/IP (Transfer Control Protocol/Internet Protocol) que provê um transporte confiável de dados, e UDP (User Datagram Protocol) que deixa a confiabilidade do transporte a cargo das camadas inferiores. O protocolo TCP/IP garante um transporte confiável mesmo operando em subredes de baixa confiabilidade onde as mensagens estão sujeitas a perda, duplicação e corrupção.

1.7.4 A Camada de Aplicação

Especifica protocolos comumente implementados em programas interativos. Exemplo:

- TELNET, rlogin: terminal remoto;
- FTP (File Transfer Protocol): transferência de arquivos;
- SMTP (Simple Mail Transfer Program): correio eletrônico;
- SNMP (Simple Network Management Protocol): gerenciamento de rede;
- NNTP (Network News Transfer Protocol): boletim eletrônico;
- etc.

Chapter 2

A CAMADA INTERFACE DE REDE

2.1 Meios de Transmissão

2.1.1 Par Trançado Metálico

Um par trançado constitui-se de dois fios enrolados de forma helicoidal. Esta forma evita que os fios assumam características de uma antena, o que os tornaria susceptíveis a interferências eletromagnéticas, bem como minimizam a componente indutiva da impedância. A componente resistiva da impedância sofre o chamado *efeito pelicular*, segundo o qual a corrente elétrica tende a se concentrar nas bordas do condutor, aumentando sua resistência efetiva.

Pares trançados são utilizados tanto para transmissão de sinais analógicos (em redes telefônicas), quanto digitais (em redes de computadores). Este meio de transmissão tem como atrativo o baixo custo e a facilidade de instalação, aproveitando-se, em muitos casos, a própria fiação telefônica existente.

A frequência máxima de transmissão depende do comprimento e da espessura do par de fios, o que em última instância dita a impedância elétrica do par. Para longas distâncias (alguns quilometros) a taxa de transmissão não ultrapassa 20 Kbits/s, podendo atingir 100 megabits/s em comprimentos de poucos metros. É comum rede padrão de 10 megabits/s (como o padrão Ethernet) ter como meio de transmissão pares trançados para distâncias inferiores a 1 quilometro. Atualmente, padrões na faixa de 100 megabits/s como o FDDI (Fiber Distributed Data Interface) e o *Fast Ethernet* operam com pares trançados para distâncias inferiores a 100 metros.

2.1.2 Cabo Coaxial

É composto de um condutor cilíndrico isolado envolto por uma malha de cobre e uma capa plástica de proteção. A blindagem forma uma proteção eletrostática ao condutor central tornando o cabo mais imune a interferências eletromagnéticas. Sua forma de construção minimiza as perdas em altas frequências (efeito pelicular e irradiação). Entretanto, sua estrutura assimétrica contribui para a atenuação imposta à amplitude do sinal. Apesar de apresentar certa complexidade de instalação, cabos coaxiais são atualmente o meio de transmissão mais empregado em redes locais de computadores.

Tipicamente cabos com impedâncias características de 75Ω e 50Ω são empregados. Os cabos de 50Ω são denominados *cabos banda básica* pois são adequados ao suporte de uma frequência básica de transmissão (ou duas no caso de modulação por chaveamento de frequência). Os cabos de 75Ω são denominados *cabos banda larga* por disporem de largura de faixa estendida permitindo a multiplexação pela divisão em frequência de vários canais.

A distância máxima em que um cabo coaxial pode ser empregado depende da atenuação imposta ao sinal. Um limite máximo de 30 dB é comumente imposto. A atenuação depende do comprimento do cabo, de suas características elétricas, da frequência do sinal e do número de conectores existentes (onde os hosts se conectam).

2.1.3 Fibra Ótica

A fibra ótica é composta de um núcleo de sílica envolto por uma casca também de sílica, tudo protegido por uma capa plástica. O núcleo e a casca apresentam índices de refração distintos, apesar de construídos de materiais similares. A tecnologia mais comum é a chamadas *fibra multimodo* onde a luz é mantida no núcleo por reflexão na casca (figura 2.1). Fibras multimodo possuem diâmetros entre 50 e 200 μm. Atenuações de 1 a 5 dB por quilometro na potência do sinal ótico são típicas.

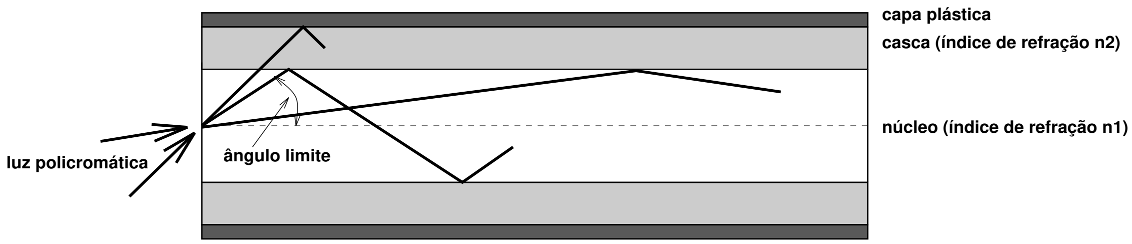

Figura 2.1: Fibra ótica multimodo.

O sinal ótico consite de luz policromática de comprimento de onda centrado em $0.8\mu m$. O sinal é produzido por diodos LED e captado por fotodetectores, sendo totalmente imune a interferências eletromagnéticas.

A fibras óticas são de difícil instalação, e utilizadas em redes com topologia em anel onde o tráfego da informação se dá num sentido único. A conexão de um host numa rede de fibra ótica é um processo complicado. O sinal ótico é convertido num sinal elétrico correspondente, passado ao computador que, quando for o caso, o retransmite empregando o processo inverso. Essa regeneração do sinal coopera para o aumento da distância de cobertura da rede.

Redes baseadas em fibra ótica (FDDI e DQDB, por exemplo) operam a taxas de 100-500 megabits/s. Taxas de gigabits/s com percursos de longas distâncias necessitam fibras monomodo (diâmetros de 5 a 10 μm) e luz monocromática produzida por diodos laser.

2.2 Padrões Físicos da Camada Interface de Rede

2.2.1 Redes Locais

Para redes locais, os padrões IEEE 802 são os mais empregados. A família de padrões IEEE 802 dividem a camada física em cinco componentes:

1. PLS (Physical Layer Signaling): é o ponto de saída do host em direção meio físico. É parte da placa de rede inserida num *slot* do barramento de dados do computador.

2. PMA (Physical Medium Attachment): é o dispositivo responsável pela transmissão e recepção de sinais no meio físico. É comandada pela PLS. No jargão Ethernet é o *transceiver* (transmitter-receiver).

3. MDI (Medium Dependent Unit): conecta a PMA ao meio físico. Em cabos coaxiais pode ser um conector tipo T (que requer o seccionamento do cabo) ou um conector *vampiro*, que "morde" o cabo sem secioná-lo. A PMA mais a MDI formam a MAU (Medium Attachment Unit).

4. AUI (Attachment Unit Interface): conecta a PLS à PMA. Via de regra consiste de cabo de pares trançado com blindagem externa e comprimento máximo de 50 metros.

5. Meio de Transmissão.

Atualmente, a vasta maioria das redes locais são baseadas nos padrões IEEE 802: 802.3 (CSMA/CD), 802.4 (Token Bus) e 802.5 (Token Ring). Tais padrões foram integralmente aceitos pela ISO (padrões 8802).

Por exemplo, o documento ANSI/IEEE Std 802.3-1984 denominado *IEEE Standards for Local Area Networks: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) - Access Methods and Physical Layer Specifications*, de 143 páginas, descreve os padrões elétricos e mecânicos dos cinco componentes da camada física listados acima para redes CSMA/CD (Ethernet).

Tal documento é imprescindível para o fabricante de placas e componentes de redes locais, sendo de pouca valia para o projetista, instalador e usuário de redes. Em geral, dados como comprimento máximo do cabo, taxa de transmissão, etc, são suficientes para o projeto, instalação e operação de redes locais de computadores no que tange a camada física.

O 802.3 (padrão Ethernet) admite 4 meios de transmissão: cabo coaxial em banda básica, cabo coaxial em banda larga, par trançado e fibra ótica em banda básica. As taxas de transmissão podem ser de 1, 5, 10 (a mais usual) e 20 megabits/s. Para esta faixa de frequência, o par trançado está em vias de reinar absoluto, apesar de atualmente o cabo coaxial de banda básica ainda ser muito empregado. O cabo coaxial de banda larga está em franco desuso e a fibra ótica, nesta faixa de frequência, somente se justifica para casos especiais (exemplo: ambientes industriais com elevado ruído eletromagnético).

2.2.2 Redes Públicas

Em redes públicas de longa distância, o único padrão existente na camada física é a interconexão do host ou terminal de dados (DTE: Data Terminal Equipment) ao equipamento de comunicação de dados (DCE: Data Communication Equipment). O padrão mais comum é a interface X.21 da CCITT, muito parecida com a interface serial RS-232C, presente em praticamente todos os computadores. A interface X.21 prevê oito conexões para comunicação full-duplex DCE-DTE (figura 2.2).

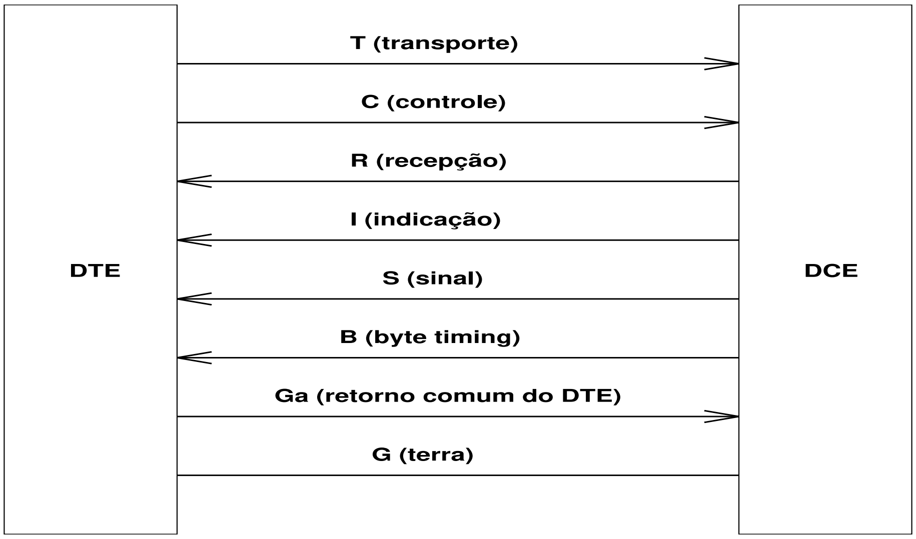

Figura 2.2: Linhas da interface X.21.

As linhas T e C são empregadas para dados e controle no sentido do DTE para o DCE; as linhas R e I para dados e controle no sentido inverso. A linha S provê ao DTE um sinal de relógio. A linha B (opcional) provê uma referência que delimita um byte. As linhas Ga e G são empregadas para retorno de corrente e terra.

Como na interface RS-232C, emprega-se codificação sem retorno a zero (NRZ) onde o bit 0 é representado por uma tensão positiva maior que 4 Volts, e o bit 1 por uma tensão negativa de módulo superior a 3 Volts. Um cabo composto de núcleos de cobre envoltos por blindagem conecta o DTE ao DCE a uma distância máxima de 15 m.

2.3 Padrões de Enlace e Acesso ao Meio

O *enlace* é responsável pela detecção e recuperação de erros ocorridos na camada física, oferecendo às camadas superiores um transporte de dados mais confiável. O enlace é uma conexão virtual por onde fluem os quadros. Esta conexão implementa protocolos simples de detecção e recuperação de erros, por exemplo detecção através de *checksum* e recuperação por retransmissão.

Em redes de longa distância e metropolitanas o enlace se dá entre IMPs. A rota física por onde os quadros fluem é exclusiva dos IMPs por ela conectados (o meio físico é decomposto em segmentos que conectam dois, e somente dois, IMPs). Ao decidir transmitir um quadro, um IMP simplesmente seleciona o segmento conectando ao destinatário e invoca os serviços da camada física referente ao segmento.

Em redes locais, o meio físico é compartilhado por todos os hosts. A transmissão de um quadro em redes locais requer antes um procedimento de acesso ao meio. Este procedimento é denominado MAC (Medium Access Control) e varia em complexidade em função da topologia e demais características da rede.

Dada a importância do controle de acesso ao meio em redes locais, é comum reservar uma subcamada no modelo de rede exclusiva para tal. No modelo OSI, a *subcamada de acesso ao meio* é parte da camada de enlace e situa-se na interface desta com a camada física. O restante da camada de enlace denomina-se Controle de Enlace Lógico (LLC: Logical Link Control) e podemos considerá-la como uma segunda subcamada, acima da subcamada de acesso ao meio. Na arquitetura Internet a subcamada de acesso ao meio, o controle de enlace lógico e a interface física de conexão ao meio estão agrupados na camada de interface de rede. A figura 2.3 ilustra as três subcamadas que compõem a interface de rede na arquitetura TCP/IP.

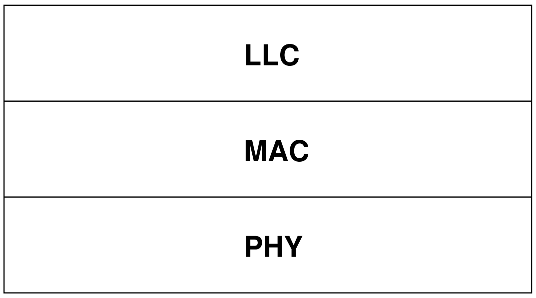

Figura 2.3: Componentes da interface de rede: LLC (enlace), MAC (acesso ao meio) e PHY (conexão ao meio físico).

2.3.1 Métodos de Acesso ao Meio

A subcamada de acesso ao meio implementa uma disciplina (seguida à risca por todos os hosts) de acesso ao meio físico. As mais difundidas técnicas de controle de acesso ao meio são as baseadas em *acesso aleatório* (ou de contenção) e *passagem de permissão*. O método de acesso aleatório mais empregado é o CSMA/CD (Carrier Sense Multiple Access-Collision Detection) empregados em redes Ethernet.

No método CSMA/CD, os hosts acessam o meio caso este esteja sem atividade. Obviamente, dois hosts podem detectar o meio inativo e acessá-lo simultaneamente gerando uma colisão. Detectada uma colisão, o host interrompe imediatamente a transmissão, entrando em seguida num processo de retransmissão. O processo de detecção de colisão é simples: durante uma transmissão o host escuta o meio, comparando o sinal no meio com aquele sendo transmitido. Ocorrida uma diferença, o host conclui que uma segunda transmissão está se sobrepondo à sua.

Detectada uma colisão, o host reforça a colisão com a injeção de sinais espúrios no meio *(jamming)* a fim de que os demais hosts transmitindo detectem imediatamente a colisão e suspendam igualmente a transmissão. O algoritmo é composto dos seguintes passos:

1. Escute o meio até ser detectada a condição de inatividade.

2. Inicie a transmissão do quadro, escutando o meio para se certificar que apenas esta transmissão está em curso. Encerrada a transmissão do quadro sem colisão, fim.

3. Reforce a colisão (jamming).

4. Caso o número de colisões (c) na transmissão deste quadro exceder um limite, sinalize um erro à camada superior e termine.

5. Gere um número aleatório (r) entre 0 e R(c).

6. Vá para 1 após r unidades de tempo.

Como o método CSMA/CD detecta colisões independente do reconhecimento por parte do receptor, esta técnica pode suportar serviços de datagrama sem confirmação.

Os métodos baseados em passagem de permissão foram desenvolvidas para redes com topologia em anel. A idéia básica é ter-se uma ficha *(token)* circulando pelo anel, de host para host. O host que detiver o *token* está autorizado a transmitir. Transmitido um quadro, este circula pelo anel até atingir o host destino. Recebido sem erros no destino, o host ativa no próprio quadro um bit de reconhecimento e transmite ao seu sucessor até atingir o host que o emitiu (note que um quadro sempre dá uma volta completa pelo anel). O emissor pode então se certificar que o quadro foi corretamente recebido ou ignorado (devido a erros ocorridos na camada física ou inexistência do destinatário), drenando-o do anel.

Nenhum host pode manter a posse do *token* por um intervalo de tempo superior a um limite pré-estabelecido. Efetuadas as transmissões ou expirado o tempo máximo de possessão do *token*, o host passa o *token* para o seu sucessor. Redes com topologia em anel que empregam passagem de permissão como método de acesso ao meio são denominadas redes *token ring*.

Métodos baseados em passagem de permissão apresentam duas características básicas: inexistência de colisões e tempo máximo de espera para acessar o meio (este tempo, em teoria, é infinito para os métodos de acesso aleatório).

2.3.2 Padrões de Enlace

Os padrões de enlace estabelecem um formato para os quadros que serão transmitidos pelo meio físico e protocolos de interação entre as subcamadas de enlace lógico comunicantes.

O enlace (conexão virtual) pode se dar entre dois hosts ou emanando de um host para vários outros. No primeiro caso o enlace é dito *ponto-a-ponto*, enquanto no segundo tem-se um enlace *multiponto*. Um enlace ponto-a-ponto ocorre, por exemplo, quando dois hosts se conectam para efetuar a transferência de um arquivo. Exemplo de enlace multiponto é um computador central controlando vários terminais geograficamente dispersos.

Podemos identificar três tipos de estações (hosts) quanto suas responsabilidades pelo enlace:

1. estações primárias: controlam totalmente o enlace;
2. estações secundárias: recebem comandos da primária, podendo transmitir pelo enlace somente quando autorizadas por esta;
3. estações combinadas: atuam de forma dual, ora como primária ora como secundária, dependendo do contexto.

Um protocolo de enlace possui tipicamente quatro fases:

1. Estabelecimento do enlace, onde um host toma a iniciativa do estabelecimento de uma conexão virtual com um ou mais hosts.
2. Transferência de dados, onde os hosts que compõem a conexão trocam quadros de informação através desta.
3. Encerramento do enlace, onde um dos hosts toma a iniciativa de propor o encerramento da conexão.
4. Reinicialização do enlace, onde um host toma a iniciativa de reinicializar o protocolo de transferência de quadros pelo enlace pela ocorrência de um erro irrecuperável.

A título de exemplo, a figura 2.4 estabelece o formato de quadros definido pelo padrão IEEE 802.3.

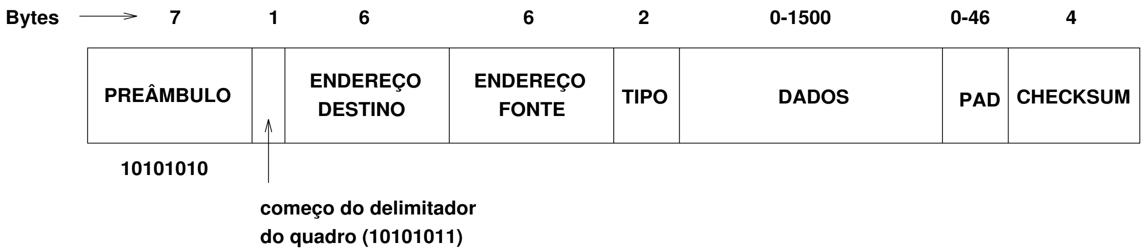

Figura 2.4: Formato dos quadros no IEEE 802.3.

O quadro inicia com um preâmbulo de 7 bytes composto dos bits 10101010. A seguir vem um byte de início de quadro composto dos bits 10101011. Duas sequências de 6 bytes estabelecem os endereços do destinatário e do emissor, respectivamente. Seguem 2 bytes contendo o tipo da informação contida no quadro e os bytes correspondentes (1500 máximo). Caso o número de bytes da informação contida no quadro sejam insuficientes para atingir o tamanho mínimo de quadro (64 bytes a partir do byte de início), um *pad* de 0 a 46 bytes completa as informações do quadro. Finalmente, 4 bytes são reservados para *checksum*.

A imposição de um tamanho mínimo de quadro se dá por duas razões:

1. quadros muitos curtos emitidos nos extremos do cabo entrar em colisão sem que os respectivos emissores a detectem (isto eé, quanto um quadro atinge o extremo oposto a tansmissão do quadro neste extremo já foi concluída);

2. reforçar o *checksum*, diminuindo a probabilidade de diferentes arranjos de bits gerarem o mesmo *checksum*.

O Protocolo ISO HDLC para Redes Públicas

O protocolo HDLC (High-level Data Link Control) é padronizado pela ISO, servindo de base para o X.25/camada 2 que emprega um subconjunto denominado LAP (Link Access Procedure). Vamos apresentar apenas este subconjunto.

No HDLC, os quadros têm o formato apresentado na figura 2.5. O quadro possui um delimitador de início e final composto dos bits 01111110. Um procedimento de codificação de bits é empregado para evitar a ocorrência desta sequência no quadro. O campo de endereço é utilizado em conexões multiponto para identificar as estações secundárias da conexão. Em conexões ponto-a-ponto este campo é utilizado para distinguir quadros de comandos dos de resposta.

Bits →	8	8	8	>= 0	16	8
	01111110	ENDEREÇO	CONTROLE	DADOS	CHECKSUM	01111110

Figura 2.5: Formato do quadro no protocolo HDLC.

O campo de dados possui tamanho arbitrário, mas normalmente limitado por restrições impostas pela camada física.

O campo de *checksum* é computado pelo polinômio gerador $x^{16} + x^{12} + x^5 + 1$.

O campo de controle define três tipos de quadros: de informação (I), de supervisão (S) e não numerados (N), sendo os dois últimos quadros de controle (figura 2.6).

Os quadros de informação carregam os contadores N(S) e N(R) para reconhecimento contínuo do fluxo de quadros.

Os quadros de supervisão são empregados no controle de fluxo e como reconhecimento da recepção de quadros. Três tipos instruções podem estar contidas no campo SS:

1. RR (Receiver Ready): informa que o receptor está pronto para continuar a recepção de quadros a partir de N(R), inclusive.

2. RNR (Receiver Not Ready): informa que o receptor reconhece o recebimento de todos os quadros até N(R)-1, e solicita a suspensão temporária do envio de novos quadros.

3. REJ (Reject): informa que o receptor detectou um erro na recepção do quadro N(R) e solicita o envio a partir deste.

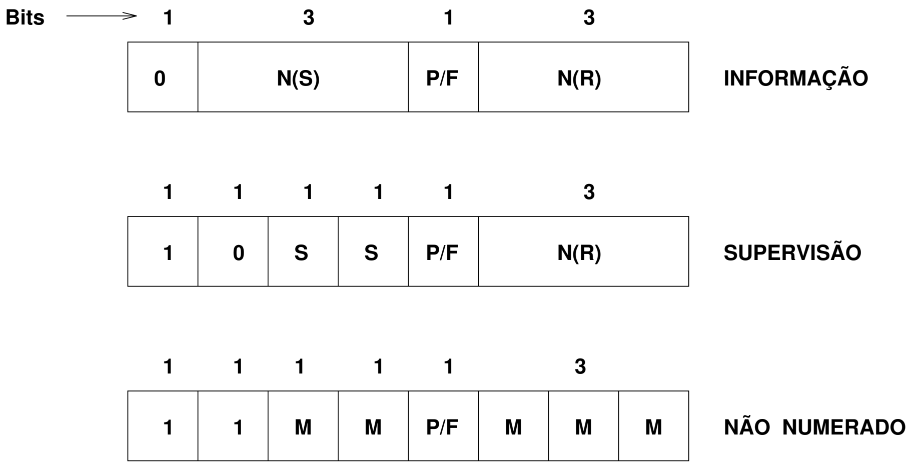

Figura 2.6: Formatos do campo de controle no protocolo HDLC.

Os quadros não numerados são utilizados para estabelecimento e término de conexões. Seis tipos de intruções são definidas:

1. SNRM (Set Normal Response Mode): estabelece uma conexão entre estação primária e secundária.

2. SABM (Set Asynchronous Balanced Mode): estabelece uma conexão entre estações combinadas.

3. DISC (Disconnect): informa o outro extremo da conexão que este host deseja terminar o enlace.

4. DM (Disconnect Mode): Informa que uma conexão solicitada não pode ser estabelecida (por exemplo, por falta de espaço para armazenar os parâmetros de controle da conexão).

5. UA (Unnumbered Acknowledgement): reconhece positivamente comandos de estabelecimento, término e reinicialização de conexão.

6. FRMR (Frame Reject): um quadro foi recebido com erro através da conexão e seu conteúdo não pôde ser identificado.

O bit P/F (Pool/Final) é ativado por uma estação primária quando em consulta a uma secundária. Caso tenha quadros para transmitir, a estação secundária o faz com o bit P/F ativado, até o último quadro onde o bit P/F é desativado, indicando o final da transmissão.

O Protocolo IEEE 802.2 para Redes Locais

O protocolo de enlace definido pelo IEEE no escopo do padrão 802 é denominado LLC (Logical Link Control) e foi inspirado no HDLC descrito acima. No padrão 802, os quadros têm o formato como o da figura 2.4. O LLC especifica o formato do PDU que é parte dos dados do quadro. Este PDU, denominado LPDU, tem o formato dado pela figura 2.7.

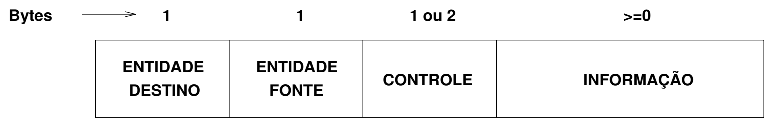

Figura 2.7: Formatos de um LPDU.

O campo ENTIDADE FONTE especifica qual entidade gerou o PDU, enquanto o campo ENTIDADE DESTINO estipula a entidade para a qual o PDU se destina. O protocolo prevê serviços sem conexão (com e sem reconhecimento) e os serviços orientados a conexão oferecidos pelo protocolo LLC. A camada de rede invoca estes serviços através de primitivas definidas pela subcamada de enlace lógico.

As famílias de primitivas são:

- L-DATA: envio de quadros de dados sem conexão;
- L-CONNECT: estabelecimento de conexão;
- L-DISCONNECT: término de conexão;
- L-DATA_CONNECT: envio de quadros de dados através de uma conexão;
- L-RESET: reinicialização de conexão;
- L-CONNECTION_FLOWCONTROL: regula o fluxo de informação entre a camada de rede e a camada de enlace locais;

A interface LLC/MAC é do tipo sem conexão, já que a comunicação entre ambas é local (interior à camada de enlace ou interface de rede). Uma única primitiva, MA-DATA, é empregada para transferir informação entre as subcamadas LLC e MAC.

O campo de controle do LPDU é inspirado no HDLC. Os três tipos de quadros do HDLC (informação, supervisão e não numerados) são definidos no protocolo LLC.

Quadros de informação são subdivididos em dois grupos: tipo I (Information) que transportam dados através de conexões, e tipo UI (Unnumbered Information) que transportam datagramas.

Quadros de supervisão são do tipo RR, RNR, REJ (idênticos ao X.25).

Quadros não numerados são do tipo SABM[1], DISC, DM, UA e FRMR (também idênticos ao X.25). Dois tipos de quadros não numerados, ausentes no X.25, são listados a seguir:

1. TEST: utilizado para testar uma conexão, fazendo com que o outro lado envie quadro similar.
2. XID (Exchange Information): utilizado para um host comunicar os tipos de serviços providos e tamanhos de janelas.

[1] O LLC suporta apenas o modo de resposta assíncrono balanceado, isto é, todas as estações são combinadas.

2.4 Endereçamento Internet

Toda comunicação via rede supõe a existência de um emissor e um destinatário identificados por *endereços*. Diferentes camadas do modelo de redes tratam endereços de forma distinta. Para a camada de aplicação, endereços devem assumir uma forma próxima a comunicação humana: hosts, usuários, domínios e serviços são identificados por nomes simbólicos. Endereços neste nível identificam processos de aplicação em comunicação.

Para a camada inter-redes, o endereço deve identificar um host e a subrede na qual o host está conectado. Neste nível o endereço é composto por números que identificam univocamente o par (subrede, host). Endereços neste nível identificam hosts comunicantes.

Para a camada interface de rede, o endereço deve identificar um dispositivo físico ligado ao meio. Neste nível, o endereço é composto por uma cadeia de bits (atribuido pelo fabricante do dispositivo).

2.4.1 Endereço IP

Endereços IP são utilizados pelo protocolo IP (Internet Protocol) para o transporte de datagrama entre dois hosts (se referem portanto a camada inter-redes). Endereços IP ocupam 32 bits e são divididos em 5 classes conforme mostrado na figura 2.8.

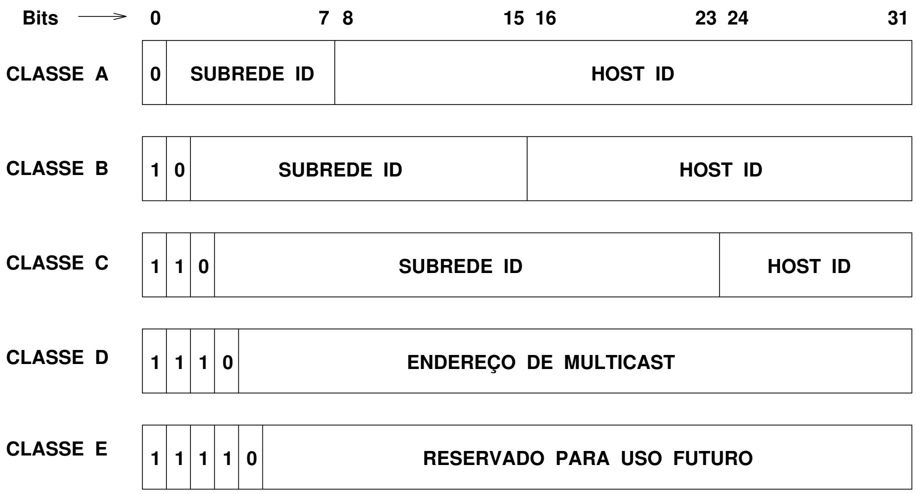

Figura 2.8: Classes de endereços IP.

A classe A é utilizada para subredes que comportam muitos hosts. Como apenas 7 bits são utilizados para identificar a subrede, podem existir apenas $2^7 = 128$ subredes da classe A (cada uma com no máximo $2^{24} = 16M$ hosts). A classe C é o oposto: embora o número de hosts na subrede é limitado (256), pode-se dispor de $2^{21} = 2M$ subredes desta classe.

Endereços da classe D são utilizados para *multicast* (comunicação envolvendo múltiplos destinatários). Nesta classe de endereços não há separação entre hosts e subrede.

Finalmente, a classe E é reservada para uso futuro, por exemplo para redes especiais.

Notação Decimal

É comum representar o endereço IP numa notação decimal. Para tal, divide-se o endereço em 4 grupos de 8 bits, converte-se os 8 bits para notação decimal, separando-os um ponto. Exemplo:

```
 10000000000010100000001000011110
10000000 00001010 00000010 00011110
            128.10.2.30
```

A notação decimal é utilizada por administradores de sistema para, por exemplo, atribuir endereço IP a um novo host, configurar *gateways*, identificar servidores, etc.

Relativamente ao valor do primeiro byte (x) do endereço IP podemos observar que:

- $x < 128$: endereçamento classe A;
- $128 \leq x < 192$: endereço classe B;
- $192 \leq x < 224$: endereço classe C;
- $x \geq 224$: reservado.

Endereços Especiais

A figura 2.9 mostra endereços especiais e os correspondentes significados. Trata-se apenas de convenção seguida pelos protocolos da camada inter-redes.

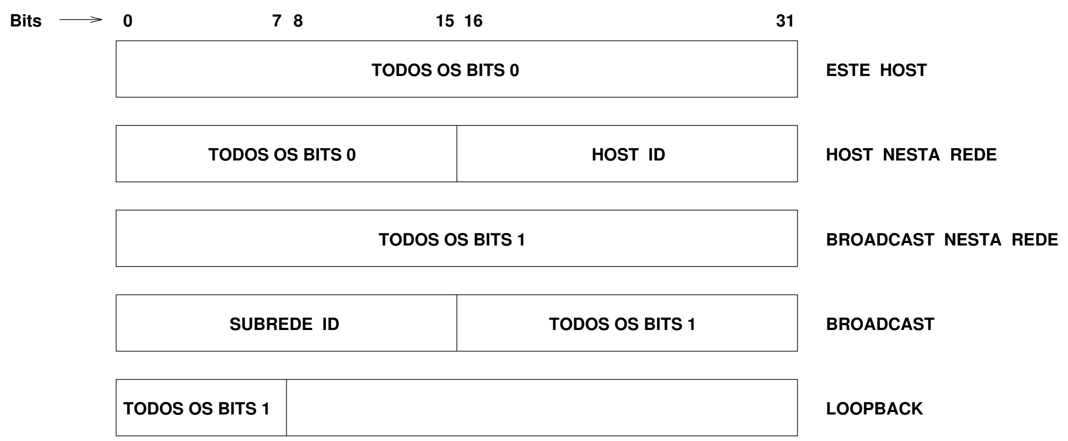

Figura 2.9: Formatos especiais de endereços IP.

Duas observações importantes. A primeira refere-se a *broadcast*, onde um datagrama é endereçado a todos os hosts de uma subrede. Via de regra, *broadcast* tende a deteriorar o desempenho da rede, pois interrompe todos os hosts daquela subrede. Por esta razão, requisições de *broadcast* são comumente limitadas à subrede do host que o emitiu.

A segunda observação refere-se a endereços *loopback* da forma 127.0.0.0. Tais endereços são utilizados para "curto-circuitar" a rede em situações onde a comunicação deve ser circunscrita ao host. Ao reconhecer um endereço *loopback* a camada inter-redes não aciona a camada física para propagar a mensagem pois trata-se de uma comunicação local.

2.4.2 Subredes

Para ilustrar o conceito de subredes, considere a figura 2.10. A figura apresenta uma empresa fictícia que utiliza 4 subredes. Considere ainda que a empresa detém um endereço classe C (215.194.97). Isto significa que a empresa em tese pode alocar até 256 (ie, 2^8) hosts.

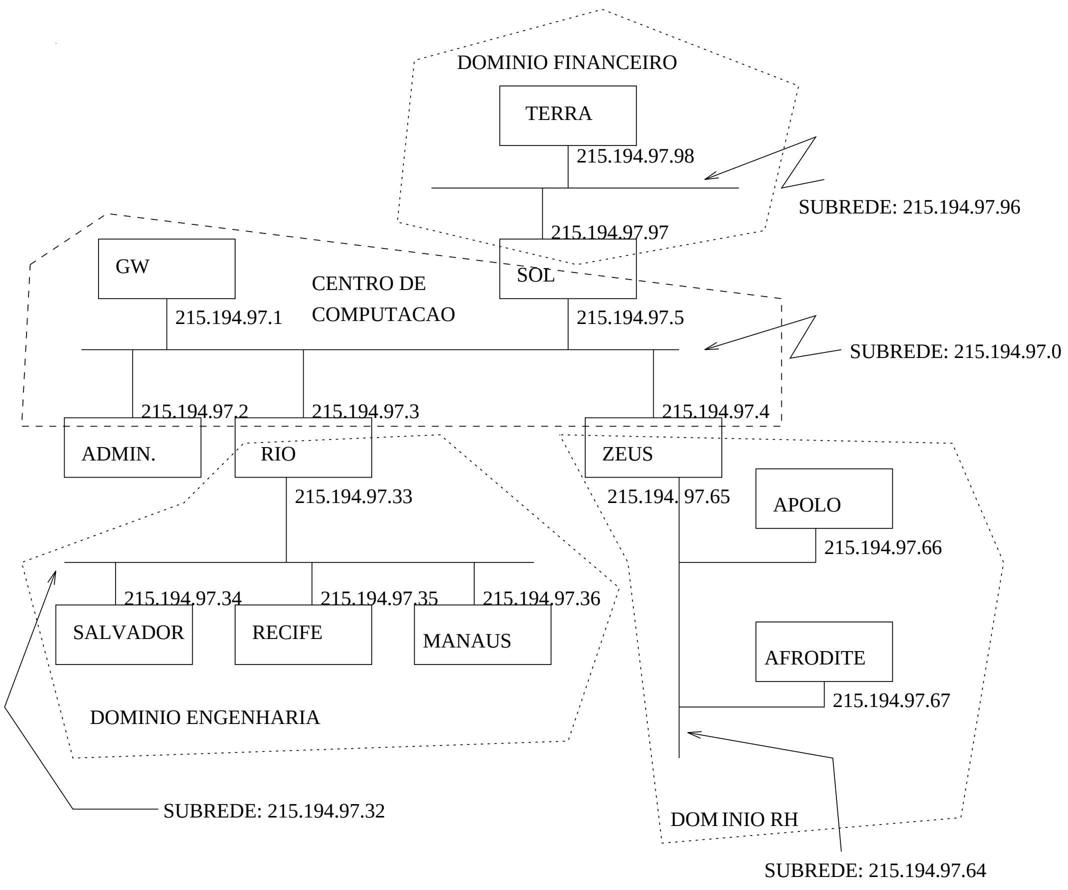

Figura 2.10: Uma empresa hipotética dividida em 4 subredes.

Entretanto, para qualquer classe de endereço IP os endereços de estação 0 e 255 são reservados. Um endereço IP com todos os bits do endereço de estação iguais a 0 identifica a própria subrede. No caso, 215.194.97.0 (endereço classe C) refere-se à subrede 215.196.97. Este tipo de endereço é utilizado nas tabelas de roteamento para referenciar uma subrede na Internet. Um endereço IP com todos os bits do endereço da estação iguais a 1 representa o endereço de *broadcast*, permitindo endereçar simultaneamente todas as estações da subrede.

Por exemplo, um datagrama com endereço 215.194.97.255 é recebido por todas as estações presentes na subrede 215.194.97.

Deve ser destacado que o endereço IP refere-se a uma interface de rede e não ao host propriamente dito. No caso da figura 2.10, os hosts GW, Rio, Zeus e Sol possuem 2 endereços IP, um para cada rede na qual o host encontra-se conectadao. O protocolo IP utiliza o endereço de subrede do endereço IP para roteamento do datagrama entre subredes. O endereço completo, incluindo o endereço do host é utilizado para entregar o datagrama quando o mesmo alcança a subrede de destino.

Estabelecimento de Subredes

A estrutura padrão do endereço IP pode ser modificado internamente ao domínio através da utlização de alguns bits do endereço do host como bits adicionais do endereço de subrede. Desta forma, a linha divisória entre os endereços do host e subrede (ver figura 2.8) é deslocada criando subredes adicionais e reduzindo o número máximo de hosts que podem participar de uma subrede. Os novos bits definem uma nova subrede dentro de uma subrede maior. A decisão de se criar subredes é normalmente associada a decisões topológicas ou administrativas. A criação de subredes permite uma descentralização no gerenciamento de endereços das estações. No caso do esquema de endereçamento tradicional um único administrador é responsável pelo gerenciamento dos endereços de todas os hosts conectados à subrede. Com a criação de subredes esta tarefa é descentralizada. No caso da figura 2.10, caso o administrador não tenha interesse em gerenciar as informações do Departamento de Engenharia da empresa, uma subrede pode ser designada para o Departamento e o seu gerenciamento ser realizado internamente ao Departamento.

Uma subrede é definida através da aplicação de uma máscara ao endereço IP. Os bits 1 da máscara definem que os bits equivalentes no endereço IP devem ser interpretados como bits do endereço de rede. Consequentemente, os bits 0 da máscara definem a parte do endereço IP que deve ser interpretada como endereço de host. A subrede é conhecida somente do ponto de vista local. Do ponto de vista externo, o endereço é interpretado como um endereço IP padrão. No caso, por exemplo, de um endereço classe B padrão, a máscara associada é 255.255.0.0. Uma possibilidade frequentemente utilizada estende em um byte a parte do endereçamento de rede da classe B. Neste caso a máscara de subrede é 255.255.255.0. Os dois primeiros bytes definem um endereço de rede classe B; o terceiro byte define o endereço de subrede e o quarto byte define o host naquela subrede.

Muitos administradores de rede preferem utilizar uma máscara orientada a byte porque é mais fácil de ser lida e compreendida. Entretanto, a máscara pode ser orientada ao bit e utilizada em qualquer classe de endereço. No caso do exemplo da figura 2.10 o endereço classe C foi subdivido em 8 subredes, sendo que na figura aparecem somente 4 subredes ficando os outros 4 endereços reservados para subredes futuras. Ao endereço IP 215.194.97.0 foi associada a máscara 255.255.255.224[2]. A aplicação desta máscara a um endereço classe C define os três bits de ordem mais alta no quarto byte como a parte que especifica a subrede. Os itens a seguir ilustram o efeito provocado pela adoção da máscara 255.255.255.224:

[2] 224 equivale à máscara binária 11100000.

Assim sendo, temos para as 8 subredes os endereços de subrede e *broadcast* dados pela tabela 2.1.

subrede	broadcast
214.194.97.0	214.194.97.31
214.194.97.32	214.194.97.63
214.194.97.64	214.194.97.95
214.194.97.128	214.194.97.159
214.194.97.96	214.194.97.127
214.194.97.160	214.194.97.191
214.194.97.192	214.194.97.223
214.194.97.224	214.194.97.251

Tabela 2.1: Endereços de subrede e broadcast para a rede 214.194.97.0 e máscara 255.255.255.224.

Exemplos:

- endereço IP: 215.194.97.1 interpretação: máquina 1 na subrede 215.194.97.0;
- endereço IP: 215.194.97.35 interpretação: máquina 3 na subrede 215.194.97.32;
- endereço IP: 215.194.97.67 interpretação: máquina 3 na subrede 215.194.97.64;
- endereço IP: 215.194.97.97 interpretação: máquina 1 na subrede 215.194.97.96;

2.4.3 Endereços Físicos

A camada interface de rede utiliza endereços físicos para localizar a interface que gerou o quadro e a interface destinatária. Na figura 2.4 os campos ENDEREÇO FONTE e ENDEREÇO DESTINO são endereços físicos, não endereços IP. No caso do padrão IEEE 802.3 (Ethernet) o endereço físico é composto de 6 bytes (48 bits) atribuídos de forma única pelo fabricante da interface[3]. Este endereço é comumente fornecido como 6 números hexadecimais separados por ponto, sendo que cada número corresponde a um byte do endereço. Por exemplo, o endereço 67.F3.AF.3E.12.FF identifica a interface

```
011001110111100110101111001111100001001011111111
```

Ao receber um quadro, a interface compara o campo ENDEREÇO DESTINO com o seu endereço. Se coincidir, o quadro é processado pela interface de rede, caso contrário é descartado.

Por convenção, um endereço Ethernet do tipo FF.FF.FF.FF.FF.FF (todos os bits 1) significa um quadro de *broadcast* e, apesar de diferir do endereço local da interface, o quadro é processado pela interface.

[3]No caso de endereços Ethernet, o IEEE é a entidade que supre os fabricantes com endereços evitando assim a possibilidade de existirem duas interfaces com o mesmo endereço.

2.4.4 Mapeamento Endereço IP - Endereço Físico: Protocolo ARP

Para enviar um quadro para um host destino, a camada interface de rede necessita do endereço físico do host. Este endereço deve ficar circunscrito à camada interface de rede, permitindo que as camadas superiores utilizem formas de endereçamento mais abstratas. Surge então o problema: dado um endereço IP de determinado host, como descobrir seu endereço físico?

A solução ideal seria as interfaces de rede armazenarem endereços IP ao invés de endereços físicos. Como tal não ocorre com a maioria das tecnologias de rede (inclusive Ethernet, a tecnologia mais utilizada) uma forma de mapeamento entre endereços IP e físico é imprescindível.

Uma solução simples é manter uma tabela relacionando endereço IP com o correspondente endereço físico. Tal solução é impraticável dadas as dimensões das redes atuais. Uma outra alternativa é difundir um quadro em *broadcast* com a seguinte requisição: quem possuir tal endereço IP, mande o seu correspondente endereço físico. O protocolo ARP (Address Resolution Protocol) adota exatamente esta idéia.

O protocolo ARP possui um PDU dado pela figura 2.11. O PDU é transportado como dados de um quadro. O quadro tem como destino um endereço de broadcast (FF.FF.FF.-FF.FF.FF no caso de redes Ethernet). Quadros contendo PDUs do protocolo ARP são identificados por um tipo armazenado no cabeçalho do quadro. No caso de um quadro Ethernet, um valor de 2054 no campo TIPO indica um quadro carregando um PDU do protocolo ARP.

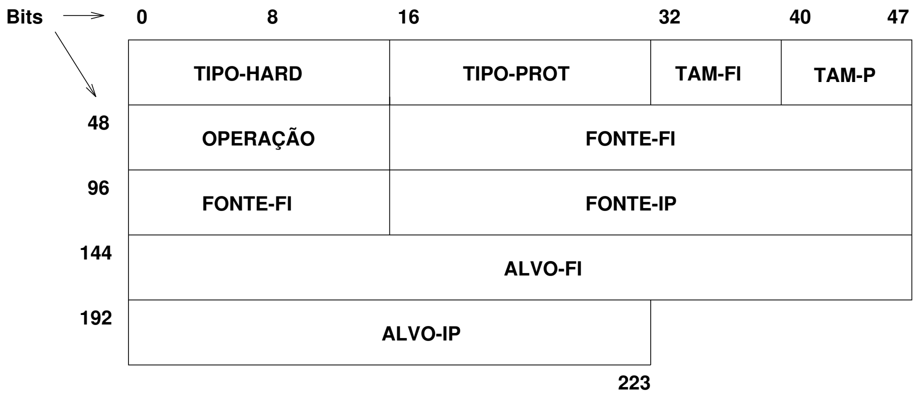

Figura 2.11: Formato do PDU para o protocolo ARP.

O campo TIPO-HARD especifica a tecnologia da subrede (1 para redes Ethernet). TIPO-PROT estipula o tipo do endereço utilizado pelo protocolo da camada inter-redes (2048 para endereços IP). TAM-FI estabelece o tamanho do endereço físico (48 bits para redes Ethernet). TAM-P estabelece o tamanho do endereço de alto nível (32 bits para endereços IP). OPERAÇÃO diferencia uma requisição (valor 1) de uma resposta (valor 2). Os enderecos físico e IP do emissor estão presentes, respectivamente, nos campos FONTE-FI e FONTE-IP. Os enderecos físico e IP do host alvo estão presentes, respectivamente, nos campos ALVO-FI e ALVO-IP. Obviamente, em se tratando de uma requisição o campo ALVO-FI não possui significado.

O protocolo mantém uma memória cache que armazena os últimos mapeamentos obtidos. Isto evita que antes de cada transmissão de quadro um *broadcast* para resolução de endereços seja efetuado. Duas outras possíveis melhoras no protocolo podem ser obtidas:

1. Como uma requisição é difundida por toda a subrede e contém os endereços IP e físico do requisitante (campos FONTE-FI e fonte-IP), todos os hosts da subrede irão receber este quadro e podem armazenar esta informação em sua memória cache para uso futuro.

2. Quando um novo host ingressa na rede, o mesmo pode voluntariamente anunciar seus endereços físico e IP num quadro ARP. Isto evita que todos os demais hosts executem o protocolo ARP quando necessitarem se comunicar com o host ingressante.

2.4.5 Mapeamento Endereço Físico - Endereço IP: Protocolo RARP

O endereço IP de um host é mantido em disco e acessado pelo sistema operacional durante o processo de *boot*. Esta informação é fundamental para a instalação dos processos que compõem o software de rede. O que ocorre no caso de um host *diskless* (sem disco)? Estações *diskless* acessam uma imagem do sistema operacional de uma estação servidora. Esta imagem não possui nenhuma referência a endereços IP pois é comumente utilizada por várias estações *diskless*. Neste caso, a estação dispõe de seu endereço físico e necessita descobrir seu próprio endereço IP. Este processo é exatamente o inverso do que estabelece o protocolo ARP.

O mapeamento endereço físico - endereço IP é estabelecido pelo protocolo RARP (Reverse ARP). O protocolo RARP utiliza o mesmo PDU do protocolo ARP (figura 2.11). Quadros referentes ao protocolo RARP possuem TIPO igual a 32821. O campo OPERAÇÃO carrega o valor 3 para requisições e 4 para respostas.

No caso do protocolo ARP, um único host responde ao *broadcast*: aquele que possui o endereço físico procurado. Entretanto, para o protocolo RARP deve haver um host especial na rede que conheça os endereços físicos dos demais. Este host especial é denominado *servidor de RARP* e possui uma tabela em disco contendo o mapeamento entre endereços físico e IP de um subconjunto de máquinas da rede (tipicamente aquelas *diskless*).

É importante notar que um host utiliza o protocolo RARP uma única vez durante o *boot*: após obtido o seu endereço IP, este dado é armazenado permanentemente em memória.

Chapter 3

A CAMADA INTER-REDES

3.1 Funcionalidades da Camada de Rede (OSI)

A camada de rede no modelo OSI (ou inter-redes na arquitetura TCP/IP) é a última camada onde o fluxo de informação leva em conta as peculiariedades da subrede de comunicação tais como sua topologia e capacidade de suas linhas físicas. A partir desta, as camadas acima empregam mecanismos de comunicação fim a fim (host a host), abstraindo a existência da subrede de comunicação. Para que a camada de transporte possa abstrair os detalhes da subrede de comunicação, a camada de rede deve prover os seguintes serviços:

- entrega de pacotes: recolher um pacote do host emissor e encaminhá-lo ao host receptor;
- roteamento: escolha da rota de comunicação por onde os pacotes oriundos da camada de transporte irão trafegar;
- controle de congestionamento: evitar que IMPs fiquem congestionados de pacotes devido a surtos de tráfego ou roteamento mal conduzido;
- interconexão de redes *(internetworking)*: possibilitar que hosts em subredes heterogêneas possam se comunicar (por exemplo, um host numa subrede Ethernet e outro numa subrede Token Ring).

3.1.1 Entrega de Pacotes

Pacote é a unidade de informação que trafega entre os IMPs de uma rede de longa distância. Quando um host necessitar transferir um pacote, este o entrega a um IMP em sua subrede. O IMP escolhe uma rota para o pacote tendo como destino a subrede ao qual o host de destino está conectado.

O tráfego entre os IMPs pode se dar através de uma circuito virtual (conexão) pré-estabelecido. Neste caso a camada de rede é dita *orientada a conexão*. Se a camada de rede não suporta circuitos virtuais a mesma é dita *sem conexão* ou *orientada a datagrama*.

Camadas de Rede Orientadas a Conexão

No caso de redes locais, conexões no nível de camada de rede são similares a conexões no nível de camada de enlace (modelo OSI): apenas dois hosts são envolvidos e apenas estes tomam parte da conexão. Em redes de longa distância, uma conexão no nível da camada de enlace envolve também dois computadores: o host e o IMP ao qual se conecta.

Uma conexão no nível de camada de rede, por outro lado, requer a intermediação de múltiplos IMPs, pois os hosts comunicantes não necessariamente se encontram na mesma subrede. No estabelecimento de uma conexão deste tipo, escolhe-se uma rota (como descrito na próxima seção) e o tráfego de pacotes dar-se-á sempre por esta rota. Um IMP que toma parte de uma conexão, recebe um pacote do IMP antecessor (ou host emissor), armazena-o temporariamente até que pacotes recebidos previamente possam fluir, e o entrega ao IMP sucessor (ou host de destino).

As vantagens de se estabelecer conexão é a alocação de recursos a priori para a transmissão de pacotes, garantindo que pacotes não serão descartados por falta de recursos para seu armazenamento quando em trânsito. Como desvantagem do emprego de conexões podemos citar:

- se um IMP que toma parte da conexão falhar, a mesma estará comprometida;
- a rede se torna mais susceptível a congestionamentos, dada a impossibilidade de se reorientar conexões já estabelecidas;
- os recursos alocados a uma conexão se tornam sub-utilizados se a taxa de transmissão de pacotes pela conexão for baixa.

Camadas de Rede Orientadas a Datagrama

Camadas de rede orientadas a datagrama não estabelecem conexão para o envio de pacotes. O serviço de entrega é pouco confiável, cabendo à camada de transporte gerenciar pacotes perdidos, duplicados ou que chegam fora de ordem. Como recursos não são previamente alocados para o tráfego de pacotes, datagramas estão sujeitos ao descarte quando um IMP não dispor de recursos para processá-lo.

Cada pacote é roteado individualmente, independente dos demais. Neste caso, a garantia de entrega de pacotes na ordem em que foram emitidos não se verifica. Pacotes podem ser duplicados caso pacotes de reconhecimento não atinjam o emissor.

A vantagem das camadas de rede orientadas a datagrama está na sua simplicidade e capacidade de se adaptar prontamente às condições de tráfego da rede e a falhas de IMPs. Como desvantagens, uma sofisticada camada de transporte é requerida. Em outras palavras, fica a cargo dos hosts (isto é, dos usuários da subrede de comunicação) gerenciar a confiabilidade do serviço de entrega de pacotes. Nas camadas de rede orientadas a conexão, esta confiabilidade recai sobre os IMPs (isto é, sobre a própria subrede de comunicação).

3.1.2 Roteamento

O problema de rotear pacotes inexiste quando a comunicação se dá numa rede local: o pacote atinge o host destinatário sem a intermediação de IMPs. Em redes MAN e WAN, a existência de IMPs obriga, na transmissão de um pacote, a se proceder a escolha de um circuito entre o emissor e o destinatário do pacote. No circuito, um ou mais IMPs irão receber o pacote, armazená-lo temporariamente e retransmití-lo ao próximo IMP do circuito.

Roteamento é um problema presente em muitas atividades. Imagine um serviço de entrega rápida com vários depósitos e uma frota de veículos. Quando um veículo apanha uma encomenda de um cliente, vários procedimentos podem ser adotados:

1. entregar a encomenda imediatamente ao destinatário com o mesmo veículo;

2. armazenar a encomenda num depósito para ser entregue com outras encomendas destinadas à mesma região;

3. mover a encomenda de depósito em depósito até atingir o mais próximo do cliente, de onde será entregue.

A última opção é análoga a roteamento em redes de computadores. Os IMPs são depósitos de pacotes e somente o IMP diretamente conectado à subrede do host destinatário está habilitado a fazer a entrega.

O problema de roteamento é de natureza combinatória, sendo portanto computacionalmente complexo caso desejamos obter a solução ótima[1]. A área de Pesquisa Operacional tem produzido vários algoritmos para este problema, bem como heurísticas para a procura de soluções sub-ótimas. Infelizmente, tais algoritmos e heurísticas requerem o conhecimento prévio do fluxo dos objetos roteados, ou uma boa estatística do mesmo.

Em redes de computadores não dispomos da informação antecipada quanto ao fluxo de pacotes. Isto demanda o cômputo do fluxo *on-line*, periodicamente, ou frente a uma condição de congestionamento na rede. Em outra palavras, em redes de computadores o roteamento deve ter certa característica dinâmica.

A forma mais simples de rotear pacotes numa subrede é dotar cada IMP de uma *tabela de roteamento* contendo qual o próximo IMP dada uma subrede de destino. Via de regra, a tabela armazena mais de um circuito na eventualidade de uma via de comunicação ou um IMP do circuito sair de serviço.

Num roteamento dinâmico, a tabela de roteamento é atualizada periodicamente em função das condições de tráfego e mudança de topologia da subrede de comunicação.

3.1.3 Controle de Congestionamento

Vimos no capítulo anterior que a camada de enlace (modelo OSI) é capaz de controlar o fluxo de quadros entre dois hosts comunicantes através dos quadros de controle RR e RNR. Cabe à camada de rede controlar o fluxo de pacotes entre IMPs evitando que um IMP receba mais pacotes do que é capaz de processar. Tal como roteamento, esta atividade é inexistente em redes locais, bastando neste caso um controle entre as camadas de rede e de enlace no mesmo host.

[1] O critério de otimalidade pode ser: mínimo tempo de trânsito, mínima distância percorrida, máxima confiabilidade, etc

Vamos examinar duas estratégias de controle de congestionamento: pré-alocação de buffers e descarte de pacotes. A primeira é mais empregada para comunicação conectada e a segunda para comunicação sem conexão.

Pré-alocação de Buffers

Um recurso importante que um IMP dispõe para o processamento de pacotes são buffers de memória. Em caso do estabelecimento de uma conexão (no nível de camada de rede), todos os IMPs envolvidos alocam buffers capazes de abrigar tantos pacotes quantos forem enviados, garantindo que pacotes que trafegam nos dois sentidos terão espaço de armazenamento alocado a priori. Caso este espaço não possa ser alocado no momento do estabelecimento da conexão por um IMP, a camada de rede escolhe outro circuito ou recusa o pedido de conexão.

Esta estratégia, entretanto, pode representar uma sub-utilização dos recursos, no caso de uma conexão ser estabelecida e poucos pacotes circularem por ela. Pior ainda, novas conexões podem ser recusadas em favor de "conexões ociosas" já estabelecidas. Este problema pode ser resolvido criando-se um *pool* de buffers e ir alocando e desalocando à medida da necessidade. Utilizando-se um *pool* de buffers corre-se o risco de um pacote chegar num IMP e não existir buffer disponível para armazená-lo. Neste caso, a conexão é rearranjada (tirando este IMP do circuito) ou um sinal de *ocupado* é enviado ao IMP que emitiu o pacote, solicitando sua retransmissão a posteriori.

Descarte de Pacotes

Esta estratégia de controle de congestionamento é oposta à apresentada acima. Cada IMP possui um conjunto de buffers para a recepção (um por linha física chegando no IMP) e para o armazenamento local de pacotes antes de sua retransmissão. Caso um pacote seja recebido e não exista espaço disponível para armazená-lo, o mesmo é simplesmente descartado.

Em geral, pacotes de controle não são descartados, sendo processados no próprio buffer de recepção. A razão para tal é que pacotes de controle geralmente causam a liberação de pacotes de dados armazenados. Por exemplo:

- pacotes de reconhecimento liberam pacotes de dados já transmitidos mas ainda armazenados caso sua retransmissão se faça necessária;
- pacotes de encerramento ou reinicialização de conexão causam o descarte de pacotes ainda fluindo pela conexão.

3.1.4 Interconexão de Redes

Dois fatos: redes são heterogêneas e redes necessitam ser interligadas. Redes são heterogêneas porque cada fornecedor desenvolveu e comercializa sua própria arquitetura de rede. É o caso da arquitetura SNA da IBM, DECNET da Digital, NetWare da Novell, e assim por diante. Se todas as redes empregassem o modelo OSI/ISO ou a arquitetura TCP/IP com idêntica pilha de protocolos, a interconexão de redes seria uma tarefa trivial.

Redes necessitam ser interconectadas para possibilitar o compartilhamento de recursos e informação. A interconexão de redes pode envolver:

- LAN-LAN: conexão entre duas LANs numa mesma organização;
- LAN-WAN: conexão entre uma LAN e uma rede pública ou corporativa;
- WAN-WAN: conexão entre duas redes públicas ou corporativas operadas por diferentes entidades;
- LAN-WAN-LAN: conexão entre duas LANs de uma mesma organização por intermédio de uma rede pública ou corporativa.

Dispositivos para Interconexão de Redes

De acordo com a heterogeneidade das redes sendo interligadas, os dispositivos de interconexão variam em complexidade. Tais dispositivos se classificam em:

- Repetidores: regeneram o sinal entre segmentos de uma mesma rede. Por exemplo, uma rede Ethernet possui um limitante de 500m para o comprimento do cabo em banda larga. Entretanto, pode-se estender uma Ethernet até 2500m utilizando-se 4 repetidores. Repetidores estão relacionados unicamente com a camada física do modelo OSI.
- Pontes (Bridges): interconectam redes cujas diferenças não passam da camada de enlace do modelo OSI. É o caso, por exemplo, das redes padrão IEEE 802.3, 802.4 e 802.5, que possuem o mesmo protocolo de enlace (802.2-LLC) mas formato de quadros, protocolo de acesso ao meio e camada física diferenciados. Pontes convertem o enlace de dados, o protocolo de acesso ao meio e os padrões da camada física de uma rede para outra, atuando nas duas primeiras camadas do modelo OSI (ou unicamente na camada interface de rede da arquitetura TCP/IP).
- Roteadores (Routers): interconectam redes cujas diferenças chegam até a camada de rede (protocolo de enlace, formato dos pacotes, etc). Por exemplo, a conexão de uma rede local a uma rede pública se dá através de um roteador (formato dos pacotes e quadros, protocolos de enlace e acesso ao meio, e padrões da camada física requerem conversão). Roteadores atuam nas três primeiras camadas do modelo OSI (ou nas duas primeiras da arquitetura TCP/IP).
- Comportas (Gateways): interconectam redes cujas diferenças ultrapassam a camada de rede. Por exemplo, uma rede local Ethernet com protocolo de transporte TCP/IP se conecta a uma rede pública com protocolo de transporte OSI (ISO 8073) através de uma comporta. Comportas também são denominadas "conversores de protocolos".

A figura 3.1 ilustra estas possibilidades.

Na arquitetura TCP/IP o elemento chave na interconexão de redes são os roteadores e as comportas (denominados genericamente de comportas no jargão TCP/IP). Como veremos adiante, comportas são responsáveis pelo roteamento de pacotes entre subredes por ela conectadas.

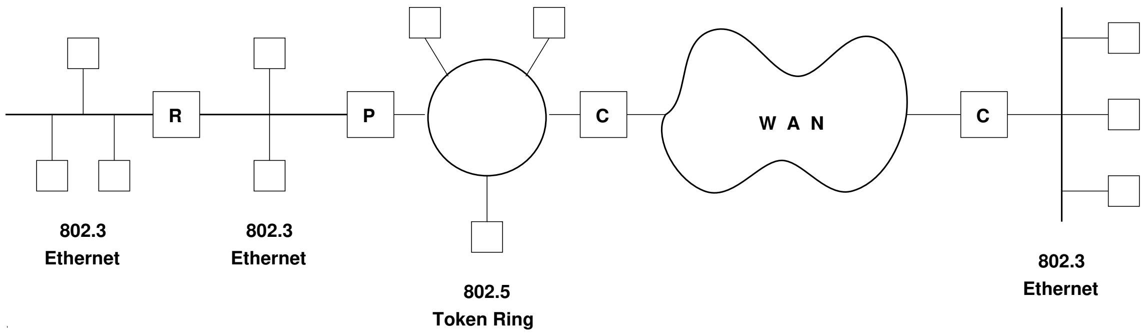

Figura 3.1: Exemplo de interconexão de redes. Retângulos com a letra R são repetidores; com a letra P pontes; e com a letra C comportas. Demais retângulos são hosts.

3.2 A Camada Inter-redes (TCP/IP)

A camada de inter-redes na arquitetura TCP/IP é equivalente a uma camada de rede do modelo OSI orientada a datagrama (sem conexão). Algumas particularidades importantes desta camada na arquitetura TCP/IP:

1. não é garantido que pacotes (datagramas na terminologia TCP/IP) serão corretamente entregues ao endereço destino;

2. nenhuma forma de reconhecimento informando o emissor sobre a recepção correta de datagramas é provida;

3. o controle de congestionamento se dá pelo descarte de pacotes;

4. procedimentos de reconhecimento e retransmissão que garantem confiabilidade na comunicação são deixados a cargo da camada de transporte.

A camada inter-redes na arquitetura TCP/IP enfoca mais o aspecto de interconexão de subredes que a confiabilidade na transmissão de datagramas. De fato, a ausência do estabelecimento prévio de conexão entre os hosts comunicantes pode acarretar:

- perda de datagrama por corrupção do quadro que o encapsula, falha de IMPs no trajeto, etc;

- duplicação de datagramas, devido a perda de quadros de reconhecimento na nivel de enlace;

- recepção de datagramas fora da ordem em que foram transmitidos, devido a mudança da rota durante a transmissão.

A camada inter-redes se utiliza da sub-camada de enlace (LLC) para transmitir datagramas. Um datagrama trafega como dados num quadro e seu conteúdo é completamente

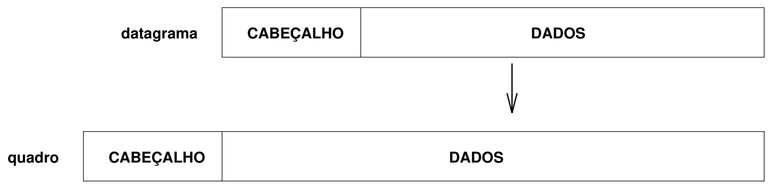

Figura 3.2: Quadros transportam datagramas no campo de dados.

ignorado pela sub-camada de enlace. A figura 3.2 estabelece a relação entre quadros e datagramas.

Via de regra as redes limitam o campo de dados dos quadros que por ela trafegam (1500 bytes no caso de redes Ethernet). Isto implica que datagramas superiores ao tamanho máximo do quadro necessitam ser *fragmentados* para a transmissão e *remontados* no destino.

Finalmente, deve ser imposto um tempo de vida máximo para o datagrama atingir seu destino. Quando um IMP detecta o expiramento do tempo de vida de um datagrama ele simplesmente o descarta. A razão desta imposição é evitar que datagramas fiquem "circulando" indefinidamente pela subrede de comunicação devido, por exemplo, a falhas no roteamento.

3.2.1 O Papel das Comportas na Interconexão de Redes

Na arquitetura TCP/IP as comportas *(gateways)* desempenham um papel importante: são o elo de conexão entre duas subredes. Uma *internet* é um conjunto de subredes interligadas por comportas. A maioria destas subredes são redes locais ou de campus interligadas a uma rede *backbone*. Backbones são redes de longa distância que servem de elo principal de interconexão. Comportas diretamente conectadas ao backbone são denominadas comportas núcleo *(core gateways)*.

Um conjunto de subredes interligadas e administradas por uma mesma entidade é denominado *sistema autônomo*. Num sistema autônomo, as comportas que o interligam ao meio exterior são denominadas *comportas exteriores*. As demais são ditas *comportas interiores*.

A figura 3.3 ilustra estes conceitos.

Em se tratando de roteamento, a importância dos diversos componentes envolvidos seguem esta ordem: comportas núcleo, comportas exteriores, comportas interiores e hosts. Hosts possuem um papel limitado no roteamento de datagramas: deve decidir apenas a partir de qual comporta o datagrama deve iniciar sua rota. Como é muito usual existir apenas uma comporta na subrede, a participação dos hosts no roteamento é mínima.

Ao receber um datagrama a comporta verifica se o mesmo se destina a uma das subredes às quais a comporta se conecta. Se for o caso o datagrama é entregue pela comporta diretamente ao host destinatário. Caso contrário, a comporta deve decidir para que outra comporta o datagrama será enviado. Na arquitetura TCP/IP esta decisão quanto ao roteamento de datagramas apresenta algumas propriedades importantes:

- a comporta é autônoma na decisão sobre roteamento de datagramas;

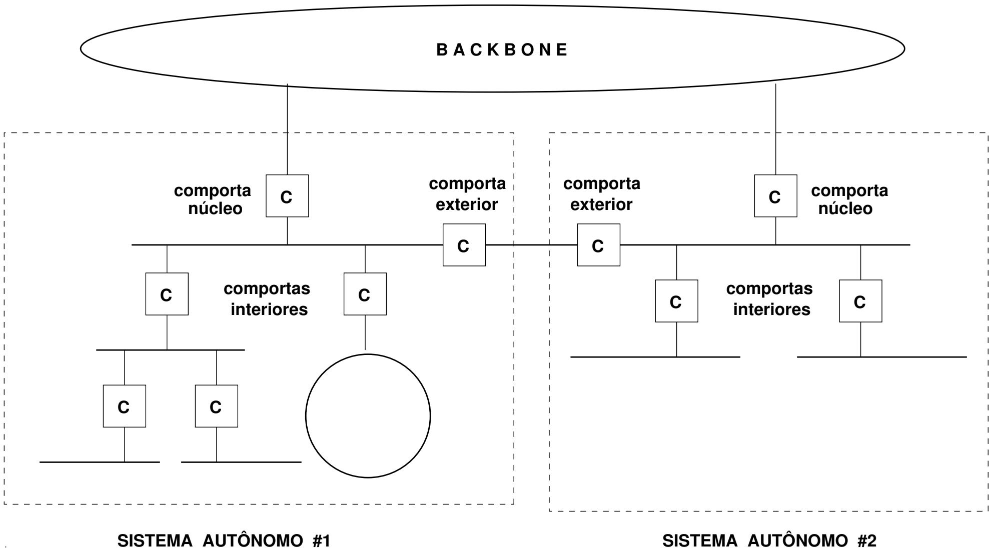

Figura 3.3: Comportas núcleo, exteriores e interiores.

- a decisão é tomada com informação parcial, notadamente a topologia da subrede de comunicação e destino final do datagrama;
- comportas comunicam entre si para atualizar informações relevantes ao roteamento: topologia da subrede de comunicação, atraso de propagação nas conexões, etc[2].

A idéia básica do procedimento de roteamento numa *internet* segue uma estruturação hierárquica. Cada comporta mantém uma tabela com três colunas: subrede de destino, próxima comporta, e custo estimado da rota. Via de regra, o custo é dado pelo número de comportas que o datagrama deve passar até atingir a subrede destino. O número de entradas nesta tabela é função da importância da comporta:

1. Comportas núcleo: mantém tabelas contendo todas as subredes da *internet*. Estas comportas mantém suas tabelas de roteamento atualizadas através do protocolo GGP (Gateaway to Gateway Protocol).
2. Comportas exteriores: mantém tabelas de umas poucas subredes "à frente". Se a subrede de destino não se encontra na tabela, uma subrede *default* mais próxima do backbone é selecionada. Estas comportas mantém suas tabelas de roteamento atualizadas através do protocolo EGP (Exterior Gateway Protocol) ou BGP (Border Gateway Protocol).
3. Comportas interiores: mantém tabelas envolvendo somente as subredes pertencentes ao sistema autônomo. Caso o datagrama se destine a uma subrede exterior ao domínio, o datagrama é passado a uma comporta exterior *default*. Estas comportas mantém suas

[2]Esta troca de informação torna o roteamento dinâmico.

tabelas de roteamento atualizadas através do protocolo RIP (Routing Information Protocol) ou HELLO.

4. Hosts: mantém tabelas estáticas indicando apenas as subredes que possam ser atingidas diretamente a partir da subrede à qual se conecta. Não se utilizam de protocolos de roteamento.

Na maioria dos sistemas UNIX, o processo *gated* é responsável pelo roteamento e tem incorporado os prorocolos GGP, EGP, RIP e HELLO. A seleção do protocolo de roteamento depende de onde *gated* executa.

3.3 Padrões da Camada Inter-Redes

3.3.1 Transporte de Datagramas: Protocolo IP

O protocolo IP (Internet Protocol) é a base da arquitetura TCP/IP. A interconexão de redes na arquitetura TCP/IP supõe que todas as subredes são capazes de manipular datagramas (pacotes) padronizados. O protocolo IP fornece exatamente um padrão para a construção e manipulação de datagramas que irão circular pelas subredes de comunicação.

Um datagrama possui um cabeçalho e um campo de dados contendo a informação que o datagrama transporta. Exceto para datagramas especiais, o conteúdo semântico dos dados é completamente ignorado pelo protocolo IP, sendo de interesse apenas das camadas superiores.

Um datagrama IP tem seu tamanho limitado em 65.535 bytes, incluindo o cabeçalho. Os campos e seus respectivos tamanhos que compõem o cabeçalho de um datagrama é apresentado na figura 3.4.

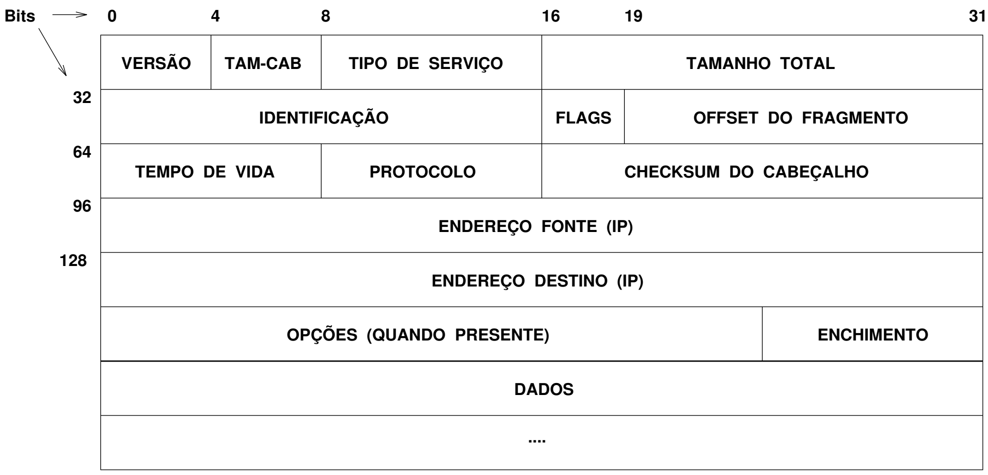

Figura 3.4: Cabeçalho do protocolo IP.

O campo VERSÃO contém a versão do protocolo, sendo utilizado como garantia que os hosts comunicantes dispõem da mesma versão do protocolo IP.

O campo TAM-CAB supre o tamanho do cabeçalho, medido em múltiplos de 32 bits.

O campo TIPO DE SERVIÇO (comumente ignorado) possui 4 sub-campos:

1. Os três primeiros bits do campo identificam o tipo de datagrama, e pode servir de indicativo de sua prioridade. O valor 0 identifica um datagrama de dados. O valor 7 identifica um datagrama de controle (merecendo portanto alta prioridade no tratamento).

2. O quarto bit (bit D: *delay*) se ativo (1) solicita um roteamento com baixo atraso.

3. O quinto bit (bit T: *throughput*) se ativo solicita um roteamento por vias de alta vazão.

4. o sexto bit (bit R: *reliability*) se ativo solicita um roteamento por vias de alta confiabilidade.

Os bits D, T e R quando utilizados fornecem informações adicionais para o roteamento. Comumente são ignorados.

O campo TAMANHO TOTAL indica o tamanho total do datagrama medido em bytes (8 bits). Por dispor de apenas 16 bits, o tamanho do datagrama fica limitado a $2^{16} = 65.535$ bytes.

Os campos IDENTIFICAÇÃO, FLAGS e OFFSET DO FRAGMENTO dizem respeito a fragmentação e remontagem de datagramas. O campo IDENTIFICAÇÃO carrega um número gerado pelo emissor e comum a todos o fragmentos. O primeiro bit do campo FLAGS (bit DF: *disable fragmentation*) quando ativado informa que o datagrama não deve ser fragmentado. O segundo bit indica se o datagrama carrega o último fragmento do datagrama original, ou um fragmento que não o último. O campo OFFSET DO FRAGMENTO especifica em múltiplos de 8 bytes (64 bits) a posição que o fragmento ocupa no datagrama original. Com estes campos é possível remontar-se um datagrama fragmentado, mesmo que seus fragmentos chegem duplicados ou fora de ordem. A perda de um fragmento impossibilita a remontagem, causando o descarte de todos os demais fragmentos (e portanto do datagrama original).

O campo TEMPO-DE-VIDA (TTL: *time-to-live*) especifica o tempo máximo que o datagrama pode permanecer na *internet* (segundos). Por cada comporta que o datagrama passa, seu tempo de vida é decrementado. A comporta pode computar o decremento de uma forma simplificada (subtraindo uma unidade do campo), ou mais elaborada (subtraindo o tempo de permanência do datagrama na comporta). Em ambos os casos o tempo de trânsito entre comportas não é levado em conta. Quando o tempo de vida de um datagrama atinge o valor zero, o mesmo é descartado.

O campo PROTOCOLO indica o tipo de dados que o datagrama carrega. Usualmente especifica um protocolo de transporte como o TCP/IP ou o UDP.

O campo CHECKSUM DO CABEÇALHO contém o valor do checksum computado apenas para o cabeçalho. O receptor do datagrama computa novamente o checksum do cabeçalho, comparando-o com o valor armazenado no campo. Em havendo diferença o datagrama é descartado. O protocolo IP não computa checksum para os dados transportados em datagramas, deixando esta tarefa para a camada de transporte.

Os campos ENDEREÇO FONTE e ENDEREÇO DESTINO contém, respectivamente, os endereços IP do host emissor e do host destinatário. Note que nenhuma informação de rota está presente no cabeçalho.

O campo OPÇÕES não está presente em datagramas comuns que carregam dados oriundos da camada de transporte. Quando presente[3], o campo é composto de 1 byte dividido em 3 sub-campos:

1. O primeiro bit, quando ativo (1) informa que o campo deve ser copiado em todos os fragmentos, caso o datagrama necessite ser fragmentado em seu trajeto. Quando inativo (0), o campo é copiado apenas no primeiro fragmento.
2. Os dois bits seguintes estabelecem a classe da opção, podendo assumir 4 valores:
   - valor 0: controle da subrede;
   - valor 1: reservado para uso futuro;
   - valor 2: depuração e medição;
   - valor 3: reservado para uso futuro.
3. Os 5 bits seguintes especificam a opção propriamente dita. Exemplo de opções:
   - registro de rota, para teste do roteamento;
   - rota específica, para enviar um datagrama por uma rota estipulada;
   - manipulação restrita do datagrama, para datagramas carregando dados confidenciais;
   - registro de tempo, para medida do tempo gasto na rota.

Os dados referentes à opção seguem o cabeçalho. Por exemplo, na opção de registro de rota, cada comporta adiciona ao campo de dados do datagrama um novo segmento de rota.

Finalmente, ENCHIMENTO não é um campo mas uma simples adição de bits (sem nenhum significado) para que o tamanho do cabeçalho do datagrama seja múltiplo de 32 bits.

3.3.2 Mensagens de Erro e Controle: Protocolo ICMP

O protocolo ICMP (Internet Control Message Protocol) é utilizado quando hosts e comportas necessitam reportar erros, testar se uma via de comunicação está operacional, medir seu atraso, etc. O protocolo utiliza o campo de dados de datagramas para o transporte de suas mensagens e possui um cabeçalho extremamente compacto (figura 3.5).

O campo TIPO especifica o tipo da mensagem. Tipos usuais são:

- valor 0: resposta a uma requisição de eco;
- valor 3: destino inatingível;

[3]A presença é indicada quando o tamanho do cabeçalho excede a 5 unidades de 32 bits.

Figura 3.5: Cabeçalho do protocolo ICMP.

- valor 8: requisição de eco;
- valor 11: tempo de vida do datagrama excedeu;
- valor 13: requisição de marca de tempo;
- valor 14: resposta a uma requisição de marca de tempo.

O campo CÓDIGO fornece informações complementares ao tipo. Por exemplo, o tipo 3 (destino inatingível) poderá vir acompanhado de um código que fornece informação mais explícita tal como:

- valor 0: subrede inatingível;
- valor 1: host inatingível;
- valor 2: protocolo não disponível;
- valor 3: port não disponível;
- valor 4: fragmentação necessária mas bit DF ativado;
- valor 5: falha de roteamento;
- valor 6: subrede desconhecida;
- valor 7: host desconhecido.

O campo CHECKSUM visa garantir a integridade da mensagem (cabeçalho mais dados).

Além dos tipos e códigos exemplificados acima, mensagens do protocolo ICMP lidam com:

1. Controle de fluxo: toda a vez que uma comporta descarta um datagrama devido a congestionamento, uma mensagem ICMP é enviada ao host emissor informando o datagrama descartado e o motivo. O emissor deve então diminuir o fluxo de datagramas.
2. Redireção de rota: quando uma comporta recebe um datagrama do host emissor e descobre que outra comporta possui melhor rota, a mesma envia uma mensagem ICMP ao host instruindo-o a alterar sua tabela de roteamento.

3. Detecção de rotas longas ou circulares: quando o tempo de vida de um datagrama expira sem ter ocorrido congestionamento, o host emissor é notificado através de uma mensagem ICMP. Esta mensagem pode indicar falha no roteamento (laços) ou seleção de rotas muito longas.

4. Sincronização de relógios: um host pode ter seu relógio sincronizado com uma comporta ou com outro host através da requisição de uma marca de tempo via mensagem ICMP.

3.3.3 Roteamento Externo: Protocolo EGP

O protocolo de roteamento EGP (Exterior Gateway Protocol) é empregado para que comportas externas cooperem nas atividades de roteamento. As mensagens possuem um cabeçalho dado pela figura 3.6.

Figura 3.6: Cabeçalho do protocolo EGP.

O campo VERSÃO especifica a versão corrente do protocolo. Os campos TIPO e CÓDIGO têm a mesma função destes campos no protocolo ICMP. O campo STATUS carrega informação de status dependente do tipo de mensagem. O campo CHECKSUM garante a detecção de mensagens corrompidas. O campo SISTEMA AUTÔNOMO identifica o sistema que gerou a mensagem. O campo NÚMERO DE SEQUÊNCIA associa mensagens (por exemplo, a que requisição uma resposta se refere).

O protocolo é baseado no conceito de *comportas vizinhas*. Duas comportas são vizinhas se ambas concordam em trocar informação de roteamento. Informações de roteamento causam a atualização das tabelas de roteamento das comportas exteriores e comportas núcleo. Consiste basicamente na informação de que existe um caminho para determinada subrede a partir de uma dada comporta.

O protocolo permite que uma comporta externa ou núcleo requisite a outra o estabelecimento de uma relação de vizinhança; testar se uma comporta vizinha está ativa; cessar uma relação de vizinhança; e trocar informação de roteamento. Estas operações são identificadas pelo tipo da mensagem. O campo de código estipula se a mensagem é uma requisição, uma confirmação ou uma recusa.

As informações de roteamento se constituem de blocos como ilustrado na figura 3.7. Cada comporta informa apenas as distâncias das subredes que fazem parte de seu sistema autônomo.

Cada bloco contém uma comporta (especificada pelo seu endereço IP). Para esta comporta segue o número de distâncias listadas a partir desta comporta. No protocolo EGP

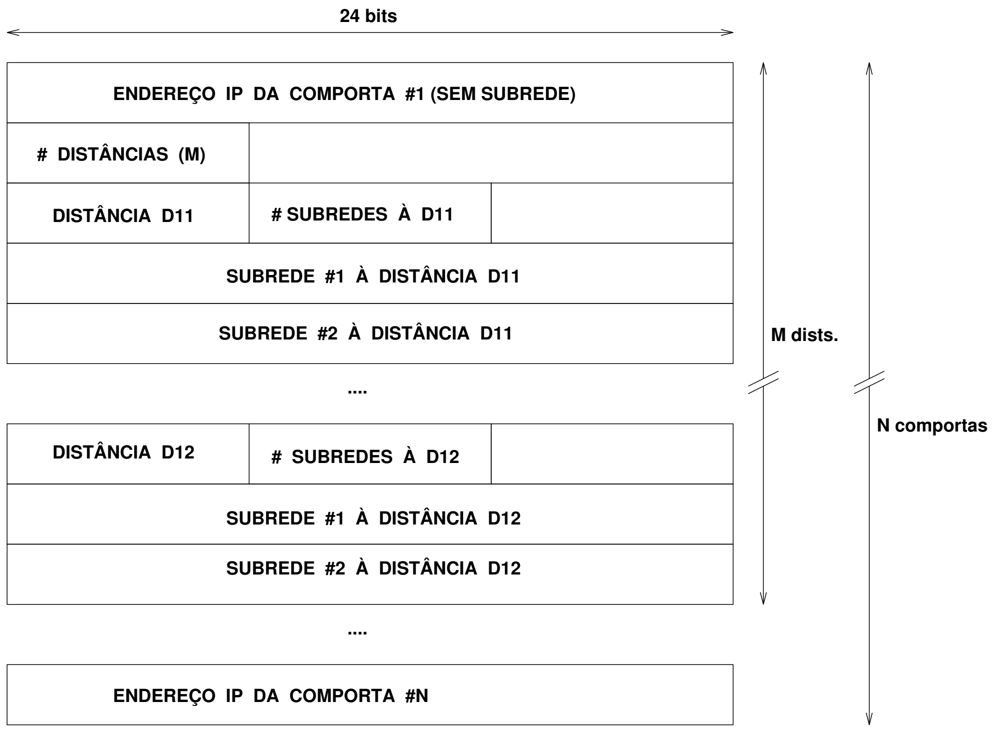

Figura 3.7: Como informações de roteamento são trocadas no protocolo EGP.

distâncias possuem apenas significado relativo a um sistema autônomo e não são utilizadas como métricas de custo. Em outras palavras, informam apenas que um caminho existe para determinada subrede. Segue o número de subredes que possuem tal distância da comporta em questão e os endereços IPs destas subredes.

Com estas informações, comportas mantém suas tabelas de roteamento atualizadas. Infelizmente dado que as distâncias não possuem significado de custo, comportas que adquirem caminhos via protocolo EGP não se utilizam de algoritmos que computam o roteamento através da minimização da distância a ser percorrida pelo datagrama. Neste caso, o roteamento é feito com base apenas na existência de rotas (abstraindo o custo das mesmas).

3.3.4 Roteamento Interno: Protocolo RIP

O protocolo de roteamento RIP (Routing Information Protocol) é empregado para que comportas internas cooperem nas atividades de roteamento. É o protocolo de roteamento interno introduzido nos sistemas UNIX BSD.

Ao contrário do protocolo EGP, distâncias no protocolo RIP possuem significado de custo: medem o número de comportas entre duas subredes. Este número varia de 1 a 15, sendo 16 utilizado para distância infinita (inexistência de caminho).

O protocolo se RIP utiliza um protocolo de transporte (UDP) para a condução de suas mensagens. No protocolo RIP informações de roteamento são obtidas pela ação de um *daemon* (processo *routed* no UNIX) que executa nas comportas internas e nos hosts. Daemons executando em comportas difundem a cada 30 segundos informações de roteamento. Daemons executando em hosts apenas se utilizam destas informações.

As mensagens possuem um formato pela figura 3.8.

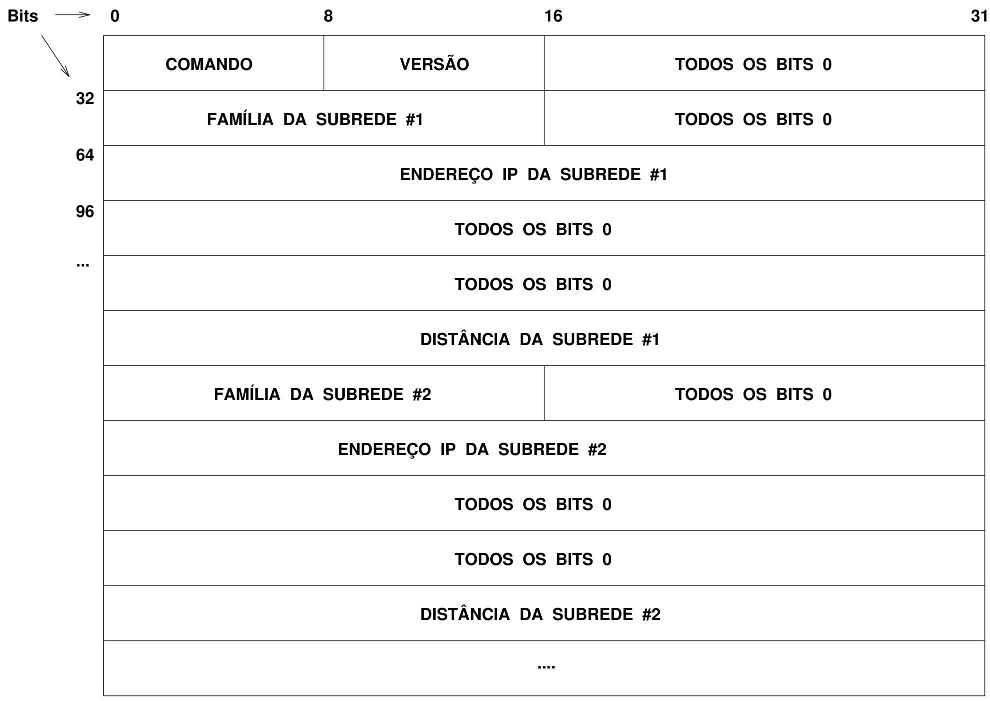

Figura 3.8: Formato de mensagens do protocolo RIP.

O campo COMANDO pode assumir o valor 1 para requisição ou 2 para resposta. Informações de roteamento se constituem de pares (subrede, distância). A comporta pode enviar toda a sua tabela de roteamento nestes pares. O campo FAMÍLIA DA SUBREDE usualmente assume valor 2 para subredes operando com o protocolo IP.

Daemons "aprendem" novas rotas recebendo mensagens RIP ou solicitando que comportas enviem sua tabela de roteamento. Por exemplo, seja o sistema autônomo da figura 3.9.

A comporta C_1 propaga na subrede 2 uma mensagem (1, 1) informando que pode atingir a subrede 1 a um custo 1. As comportas C_2 e C_3 recebem esta mensagem e atualizam em suas tabelas a informação que a subrede 1 pode ser atingida a partir de C_1 a um custo 2. Mais tarde, comportas C_2 e C_3 propagam mensagens para as subredes abaixo informando que podem atingir a subrede 1 a partir de um custo 3 (1, 3) e a subrede 2 a partir de um custo 2 (2, 2). Sucessivamente, todos os hosts e comportas terão a partir de que comporta cada subrede do sistema autônomo pode ser atingida e a que custo.

Sempre que um host ou comporta é informado de uma rota já existente em sua tabela de roteamento, o mesmo a substitui caso o custo da nova rota seja inferior.

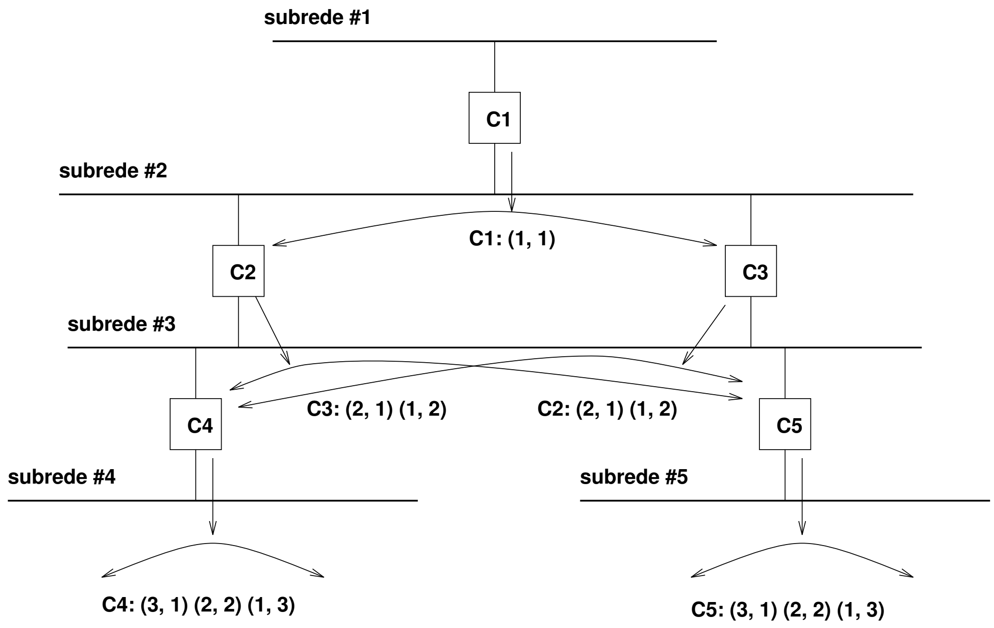

Figura 3.9: Sistema autônomo empregando o protocolo de roteamento interno RIP.

Chapter 4

A CAMADA DE TRANSPORTE

A camada de transporte é a primeira camada a abstrair a topologia e a tecnologia da subrede de comunicação. Esta camada utiliza o serviço de entrega de pacotes da camada de rede para prover comunicação confiável host a host. Se a camada de rede for baseada em datagrama, todo o trabalho referente a uma comunicação confiável fica a cargo da camada de transporte. Neste caso, a camada de transporte deve gerenciar a perda, duplicação e inversão de ordem de pacotes ocorridas na subrede de comunicação. No caso da camada de rede prover entrega confiável de pacotes, a camada de transporte tem uma estrutura mais simplificada, gerenciando principalmente o fluxo de mensagens entre os hosts comunicantes e as quebras de conexões estabelecidas no nível de camada de rede.

4.1 Conceitos Básicos

4.1.1 O Serviço de Transporte

O serviço de transporte pode ser orientado a conexão ou a datagrama (sem conexão). Em redes locais, dada a alta confiabilidade da camada de rede, o serviço sem conexão é atrativo para muitas aplicações devido ao seu baixo *overhead*. Em redes públicas, mesmo com camada de rede orientada a conexão (como no caso do X.25/camada 3), o serviço de transporte orientado a conexão é o mais seguro para a maioria das aplicações. A arquitetura TCP/IP oferece estes dois serviços de transporte.

As conexões de transporte geralmente não delimitam as mensagens, estabelecendo uma cadeia contínua de bytes. Por exemplo, o emissor poderá enviar dois segmentos de 1 Kbyte e o receptor recolher os 2 Kbytes sem nenhuma indicação de quantos segmentos foram enviados. Este modo de operação deixa a cargo da aplicação a segmentação das mensagens.

Conexões de transporte classificam-se em *full-duplex* ou *half duplex*. Conexões full-duplex suportam comunicação bidirecional; enquanto conexões half-duplex suportam apenas fluxo unidirecional. Conexões full-duplex são mais convenientes dado que as aplicações que utilizam a conexão podem enviar e receber dados pela mesma.

A transferência de dados por uma conexão se dá via de regra de forma “bufferizada”, isto é, a camada de transporte aguarda um determinado número de bytes para enviá-los num único pacote. A utilização de buffers aumenta e eficiência da comunicação pois evita o tráfego de pacotes carregando uns poucos bytes.

O protocolo TCP/IP emprega conexões full-duplex com transferência "bufferizada".

A camada de transporte pode definir alguns parâmetros associados à qualidade do transporte de dados. Estes parâmetros são negociados no estabelecimento de conexões de transporte entre os hosts e a subrede de comunicação. Qualidade de serviço depende da subrede de comunicação e não é levada em conta na maioria das redes atuais. Os principais parâmetros que definem a qualidade de serviço são:

- tempo de estabelecimento da conexão (milisegundos): máximo intervalo de tempo entre a entidade usuária da camada de transporte solicitar uma conexão e o seu estabelecimento;
- vazão (bytes/s): taxa assegurada de transmissão de mensagens pela conexão de transporte;
- atraso de propagação (milisegundos): tempo máximo de trânsito de um TPDU entre as camadas de transporte dos hosts comunicantes;
- taxa de erro residual: taxa de mensagens perdidas ou corrompidas na conexão de transporte (teoricamente zero, mas assumindo um valor muito pequeno na prática);
- tempo de encerramento de conexão (milisegundos): tempo máximo para o término de uma conexão;
- proteção: estipula a privacidade dos dados fluindo pela conexão de transporte quanto à sua interceptação por terceiros;
- confiabilidade: mede a probabilidade da camada de transporte falhar espontaneamente (devido a "bugs", situações não previstas, etc).

Um subconjunto destes parâmetros é passado à camada de transporte no estabelecimento de uma conexão. A conexão não é estabelecida se a qualidade do serviço sendo requisitada não puder ser honrada:

1. pela camada de transporte do host local;
2. pela camada de transporte do host remoto (sendo conectado);
3. pela subrede de comunicação.

A arquitetura TCP/IP não prevê qualidade de serviço no estabelecimento de conexões. Esta lacuna deve desaparecer com a introdução de serviços multimídia.

4.1.2 Classes de Protocolos de Transporte

Os protocolos de transporte são classificados pela ISO em cinco categorias em função da camada de rede da qual se utilizam e de certas funcionalidades que provêem.

As subredes de comunicação se classificam em três tipos:

- Tipo A (livres de erros): apenas as LANs se aproximam desta classe;
- Tipo B (transporte confiável de pacotes, mas com a ocorrência de quebras de conexões): redes públicas X.25 pertencem a esta classe;
- Tipo C (transporte de pacotes não confiável): redes públicas com protocolo IP como a ARPANET pertencem a esta classe.

As cinco categorias de protocolos de transporte são descritas a seguir.

- Classe 0: Operam em subredes tipo A. São os protocolos mas simples pois toda a confiabilidade do transporte de dados recai sobre a subrede de comunicação. Na arquitetura TCP/IP o protocolo UDP pertence a esta classe.
- Classe 1: Para subredes tipo B, provêem recuperação básica de erros. No mais, deixam a cargo da subrede de comunicação a confiabilidade do transporte de dados.
- Classe 2: Para redes tipo A, são idênticos aos de Classe 0, permitindo apenas a multiplexação de várias conexões de transporte numa única conexão de rede.
- Classe 3: Para redes tipo B, provêem as funcionalidas das classes 1 e 2.
- Classe 4: Os únicos a operarem em subredes tipo C, provêem detecção e recuperação de erros ocorridos na subrede de comunicação. São os protocolos de transporte mais complexos como é o caso do TCP/IP.

4.1.3 Protocolos de Transporte Classe 4

Protocolos classe 4 operam em subredes tipo C que oferecem serviços de datagramas sem garantia de entrega, preservação da ordem ou ausência de duplicação. Portanto, cabe ao serviço de transporte adicionar a confiabilidade não oferecida pela camada de rede. Protocolos classe 4 são empregados também em subredes tipo B, onde a camada de transporte pode se valer de serviços orientados à conexão oferecidos pela camada de rede. Neste caso, as diferenças entre um protocolo classe 3 e 4 são desprezíveis em termos de desempenho, já que ambos terão que gerenciar apenas as quebras de conexão no nível da camada de rede.

Um protocolo de transporte classe 4 deve gerenciar a perda, duplicação e inversão de ordem dos pacotes em três situações:

1. durante o estabelecimento de conexões de transporte;
2. durante a transferência de TPDUs de dados;
3. durante o encerramento da conexão.

Vamos examinar as soluções comumente empregadas nos protocolos classe 4 (principalmente o TCP/IP) para estas três situações.

Estabelecimento de Conexões

Dois cenários devem ser gerenciados por um protocolo de transporte classe 4: duplicação de um TPDU que carrega uma requisição de estabelecimento de conexão; e perda de um TPDU que carrega a confirmação do estabelecimento da conexão.

No primeiro caso, ao receber um TPDU duplicado, o destinatário irá supor que uma segunda conexão foi solicitada e alocará recursos para a mesma. No segundo cenário, o host que solicitou a conexão supõe que sua requisição se perdeu e a envia novamente. Uma segunda conexão será estabelecida pelo receptor, como no caso anterior.

Os dois cenários descritos acima apresentam uma característica em comum: uma conexão estabelecida apenas num host, e que portanto jamais será utilizada. Os recursos alocados a tais conexões são totalmente inúteis e permanecem alocados até esta situação ser detectada e corrigida.

Uma solução comumente empregada é o estabelecimento de conexões em três fases. Esta técnica soluciona os problemas discutidos acima. Ao enviar um pedido de conexão (SYN), o host solicitante informa um número inicial de sequência na numeração dos TPDUs de dados que ele enviará pela conexão: X (primeira fase). Ao receber a solicitação, o destinatário reconhece X e informa que seu número inicial de sequência será Y (segunda fase). Ao receber a confirmação da conexão, o host que a iniciou reconhece Y num TPDU de reconhecimento (ACK) ou no primeiro TPDU de dados que enviar (terceira fase). Este método de estabelecimento de conexões (figura 4.1) é imune a duplicação e perda de TPDUs empregados no estabelecimento de conexões.

No primeiro cenário, o host que recebeu a segunda confirmação de conexão responderá com um REJECT na terceira fase, causando a desativação imediata da conexão gerada pelo TPDU duplicado. No segundo cenário, a primeira conexão será desativada por falta do recebimento de confirmação na terceira fase (após um *timeout* na espera da confirmação).

O protocolo TCP/IP emprega o estabelecimento de conexões em três fases.

Transferência de TPDUs de Dados

Protocolos de transporte classe 4 como o TCP/IP empregam como mecanismo de confiabilidade o *reconhecimento* e a *retransmissão*. O reconhecimento usualmente é positivo, isto é, uma confirmação de recepção é emitida no sentido inverso ao fluxo de TPDUs de dados. O protocolo pode esperar para enviar o próximo TPDU apenas quando receber o reconhecimento do anterior; ou definir uma *janela* de N TPDUs que podem ser enviados sem a necessidade de reconhecimento.

No caso de se empregar janelas, vai-se *deslizando* a janela à medida que TPDUs de reconhecimento vão sendo recebidos. Este método é denominado *janela deslizante* sendo ilustrado na figura 4.2.

Uma janela começando em k e terminando em $k+N$ divide os TPDUs em três categorias:

1. TPDUs de numeração anterior a k podem ser descartados pois foram enviados e tiveram seu recebimento confirmados;

2. um TPDUs de numeração entre k e $k+N$ pode:

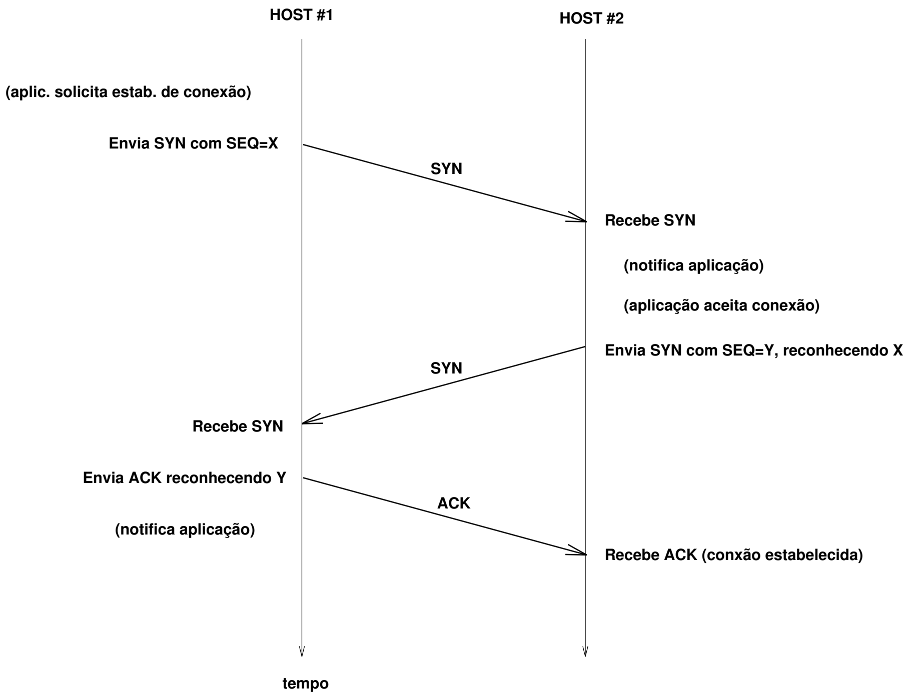

Figura 4.1: Diagrama de eventos para o estabelecimento de conexão em 3 fases.

- ter sido enviado e seu recebimento confirmado;
- ter sido enviado e seu recebimento ainda não confirmado;
- ainda não ter sido enviado.

3. TPDUs de numeração posterior a $k + N$ ainda não foram enviados (e somente o serão quando a janela se mover).

A figura 4.2 mostra que o emprego de janela deslizante aumenta a taxa de transmissão de TPDUs. Quando a janela para de se mover devido à ausência de reconhecimento de TPDUs localizados no início da mesma, estes TPDUs são retransmitidos ante a possibilidade de não terem atingido o receptor. Caso seja o TPDU de reconhecimento que se perdeu, o correspondente TPDU de dados chegará duplicado no seu destino, sendo simplesmente descartado pelo receptor.

O tamanho da janela (N) deve ser selecionado com cautela. Uma janela de grandes dimensões aumenta a taxa de transmissão de TPDUs, mas requer uma capacidade maior de armazenamento temporário tanto por parte do emissor quanto do receptor.

Se a conexão for full-duplex (como as conexões TCP/IP), pode-se otimizar o processo de reconhecimento através de uma técnica denominada *piggybacking*. Seja uma conexão de transporte ligando as aplicações A e B. Sempre que um TPDU de dados for enviado por B, este pode conter em seu cabeçalho o número do último TPDU enviado por A e que B recebeu em sequência. Suponha que A enviou o TPDU número 106 correspondente a janela 100-109.

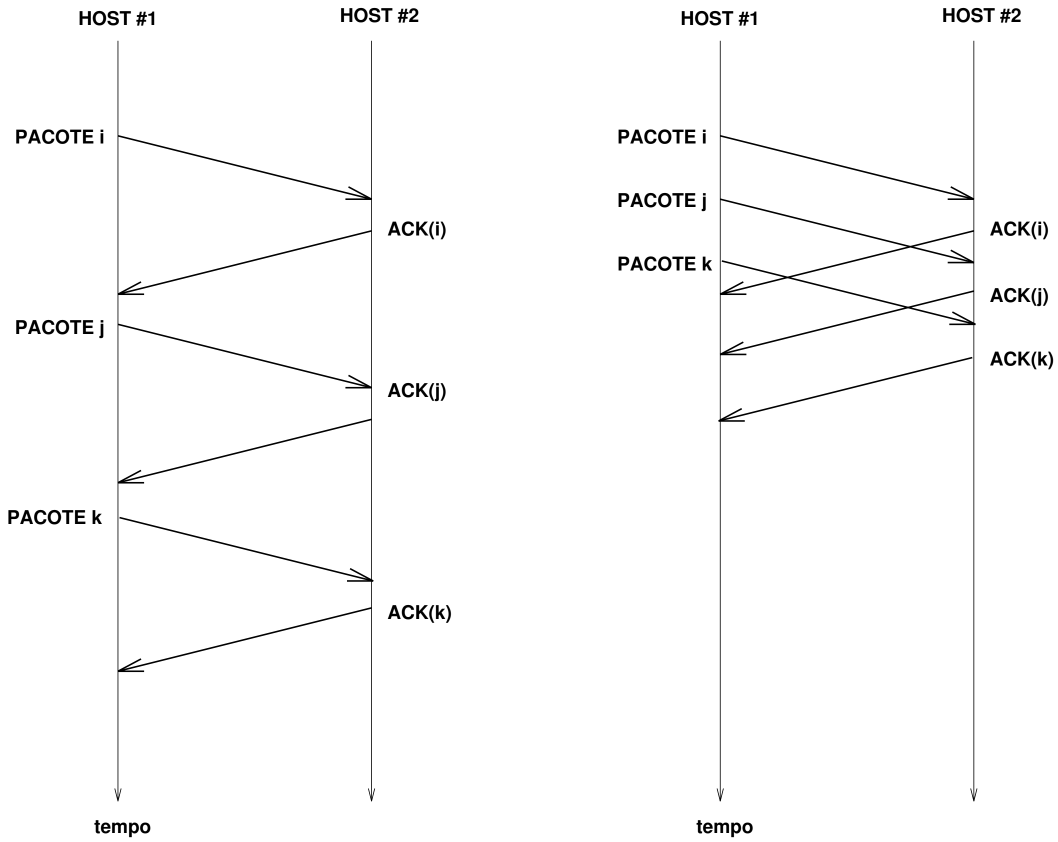

Figura 4.2: Reconhecimento individual (esquerda) comparado com janela deslizante de tamanho 3.

Neste momento A recebe um TPDU de dados de B no qual B reconhece o recebimento da sequência até o número 103 (inclusive). Neste momento:

- A pode descartar os TPDU 100, 101, 102, 103;
- A pode avançar a janela mais 4 unidades, podendo transmitir até o TPDU 113 sem necessitar aguardar qualquer reconhecimento por parte de B.

Piggybacking reduz o número de reconhecimentos caso o tráfego de TPDUs de dados pela conexão seja intenso nos dois sentidos. Esta redução se dá pelo envio de reconhecimentos em TPDUs de dados (economizando o envio de TDPUs de reconhecimento).

O protocolo TCP/IP emprega um mecanismo de janela deslizante na transmissão de TPDUs. Entretanto, ao invés de reconhecer individualmente os TPDUs enviados, o receptor reconhece a posição do último byte recebido na sequência de bytes transmitida deste o início da conexão[1]. Neste caso, o tamanho da janela se refere a uma quantidade de bytes, não a um número de TPDUs.

[1] A rigor, reconhecimento no TCP/IP estipula o próximo byte que o receptor espera receber.

Esta forma reconhecimento não requer o envio de um TPDU de reconhecimento para cada TPDU recebido corretamente pelo emissor. De fato, se os TPDUs estão atingindo o receptor corretamente, um único TPDU de reconhecimento poderá ser enviado confirmando de uma única vez todos os bytes transmitidos na janela.

Entretanto, se um TPDU de dados não atingir o receptor, esta forma de reconhecimento é ineficiente pois nada se pode concluir sobre os TPDUs da janela que se seguiram a este. Considere uma janela de 8000 bytes, sendo que cada TPDU carrega 1000 bytes. Seja a situação em que o emissor enviou os 8 primeiros TPDUs e aguarda reconhecimento. Expirado o período de tempo que o emissor aguarda o reconhecimento, suponha que o receptor confirmou apenas o byte 3000. Neste momento, o emissor pode afirmar que:

- os 3 primeiros TPDUs chegaram corretamente e portanto podem ser descartados;
- o quarto TPDU não atingiu o receptor (ou sua confirmação se perdeu);
- nada se pode concluir sobre os demais TPDUs da janela.

No exemplo acima, as seguintes ações podem ser tomadas pelo emissor:

1. Enviar o quarto TPDU e aguardar reconhecimento na esperança que apenas este tenha se perdido. Neste caso o recebimento deste TPDU causaria a confirmação de toda a janela (byte 8000).
2. Enviar os 5 últimos TPDUs (isto é, todo o restante da janela).

Usualmente a segunda opção é escolhida.

Encerramento de Conexões

O encerramento de conexões em protocolos de transporte classe 4 é também um processo que requer cuidados. Dois cenários devem ser considerados:

1. o TPDU solicitando o encerramento da conexão se perdeu, gerando o fechamento da conexão em apenas um lado;
2. um TPDU de dados *atrasado* chega mas a conexão pela qual fluia já foi encerrada.

Analogamente ao estabelecimento de conexões, um esquema de três fases pode ser empregado para o encerramento de conexões. Ao enviar uma requisição de encerramento de conexão, o emissor dispara um contador de tempo. Chegada a confirmação do encerramento, a "confirmação da confirmação" é enviada e o host encerra a conexão. Expirado o tempo do contador sem a chegada de confirmação, o pedido de encerramento de conexão é retransmitido. O processo se repete por determinado número de vezes quando a conexão é encerrada unilateralmente pelo lado do emissor.

Ao receber um TPDU solicitando o encerramento de conexão, o receptor também dispara um contador de tempo (com intervalo bem superior ao do emissor), enviando prontamente a confirmação ao solicitante. Se este tempo se expirar sem o recebimento da "confirmação da confirmação", o host encerra unilateralmente a conexão.

O protocolo TCP/IP utiliza um mecanismo de encerramento em três fases ligeiramente diferente do descrito acima. O lado que toma a iniciativa do encerramento envia um TPDU solicitando o fechamento da conexão (FIN). O lado oposto recebe a requisição e confirma o encerramento (ACK). A conexão permanece ativa no outro lado até que a aplicação seja notificada do encerramento e proceda o fechamento no seu respectivo lado, enviando um FIN e aguardando um ACK. Este mecanismo requer que o protocolo mantenha um histórico de conexões estabelecidas. Mecanismos de retransmissão são empregados caso TPDUs carregando FIN ou ACK não atinjam o destino.

A figura 4.3 ilustra este procedimento.

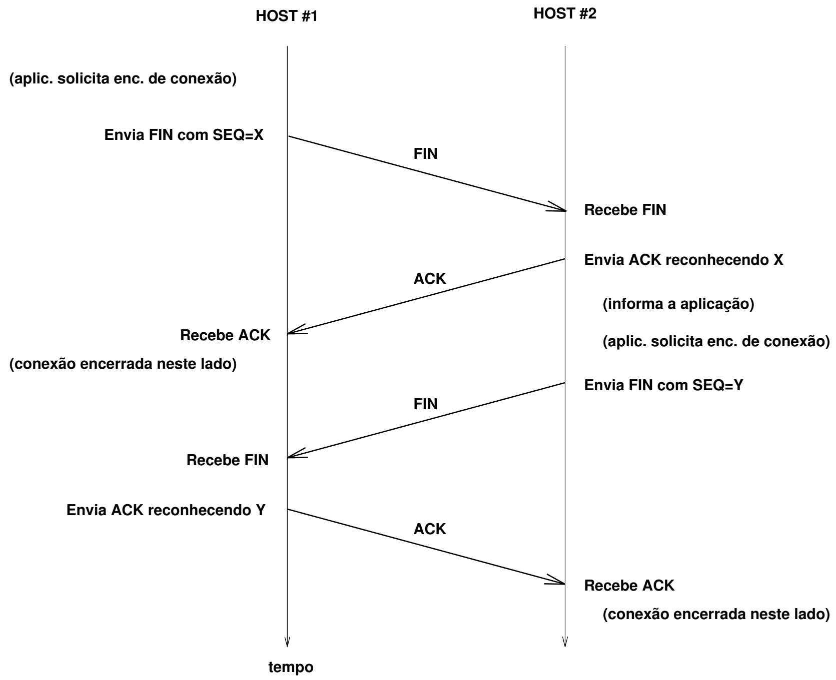

Figura 4.3: Encerramento de conexão no protocolo TCP/IP.

Este procedimento garante também que TPDUs "atrasados" não serão descartados. Simplesmente, um host não aceita o encerramento da conexão se um ou mais TPDUs de dados ainda não tiveram seu recebimento confirmado.

Os mecanismos de encerramento de conexões apresentados não garantem que uma conexão será encerrada em cem por cento dos casos (por exemplo, quando nenhum dos TPDUs solicitando o encerramento da conexão atingem o destino). Uma segurança adicional é encerrar uma conexão caso permaneça inativa por um determinado período de tempo. TPDUs de conteúdo nulo podem ser empregados para manter a conexão viva, fazendo com que sua recepção zere o contador de tempo que monitora e inatividade da conexão. O protocolo TCP/IP encerra conexões inativas por um longo período de tempo.

4.2 Padrões da Camada de Transporte: Protocolo TCP/IP

A camada de transporte da arquitetura TCP/IP provê dois serviços: um serviço orientado a conexão e um seviço orientado a datagrama. O primeiro é suprido pelo protocolo TCP/IP (Transfer Control Protocol / Internet Protocol) e o segundo pelo protocolo UDP (User Datagram Protocol).

O protocolo TCP/IP pertence à classe 4 e opera sob uma camada de rede orientada a datagrama. As principais características do protocolo foram introduzidas nas seções acima:

- utilização de conexões full-duplex;
- estabelecimento e fechamento de conexões em três fases;
- transferência "bufferizada" de dados;
- mensagens sem delimitação de fronteiras;
- controle de fluxo por janela deslizante.

Estas características são comuns à maioria dos protocolos classe 4. Nesta seção apresentaremos algumas peculiariedades do protocolo TCP/IP.

4.2.1 O Conceito de Port

Para as camadas interface de rede e inter-redes o destinatário é identificado univocamente por um endereço de host apenas. A camada de transporte atende a múltiplas aplicações e portanto deve ser capaz de diferenciar os quadros endereçados a estas aplicações. Obviamente, um componente adicional ao host deve estar presente no endereçamento empregado pela camada de transporte. Este componente é denominado *port* e se presta a identicar uma aplicação usuária da camada de transporte num dado host. O port é único num host, sendo o par (host, port) suficiente para endereçar uma aplicação.

Ports identificam também recursos utilizados para o envio e recepção de mensagens. Estes recursos constituem-se de buffers onde mensagens em trânsito são armazenadas temporariamente. Em sendo recurso de máquina, ports são alocados pelo sistema operacional via chamada de sistema. No Protocolo TCP/IP ports são identificados por inteiros sem sinal ocupando 16 bits.

A figura 4.4 ilustra múltiplas aplicações compartilhando um única instância de protocolo de transporte.

O protocolo TCP/IP identifica duas aplicações comunicantes atráves da conexão estabelecida entre elas. Uma conexão é referenciada por duas duplas (host, port). Por exemplo o par

```
(143.106.11.188, 1234)  (143.106.1.10, 21)
```

indica que a aplicação que possui o port 1234 no host 143.106.11.188 está conectada à que possui o port 21 no host 143.106.1.10.

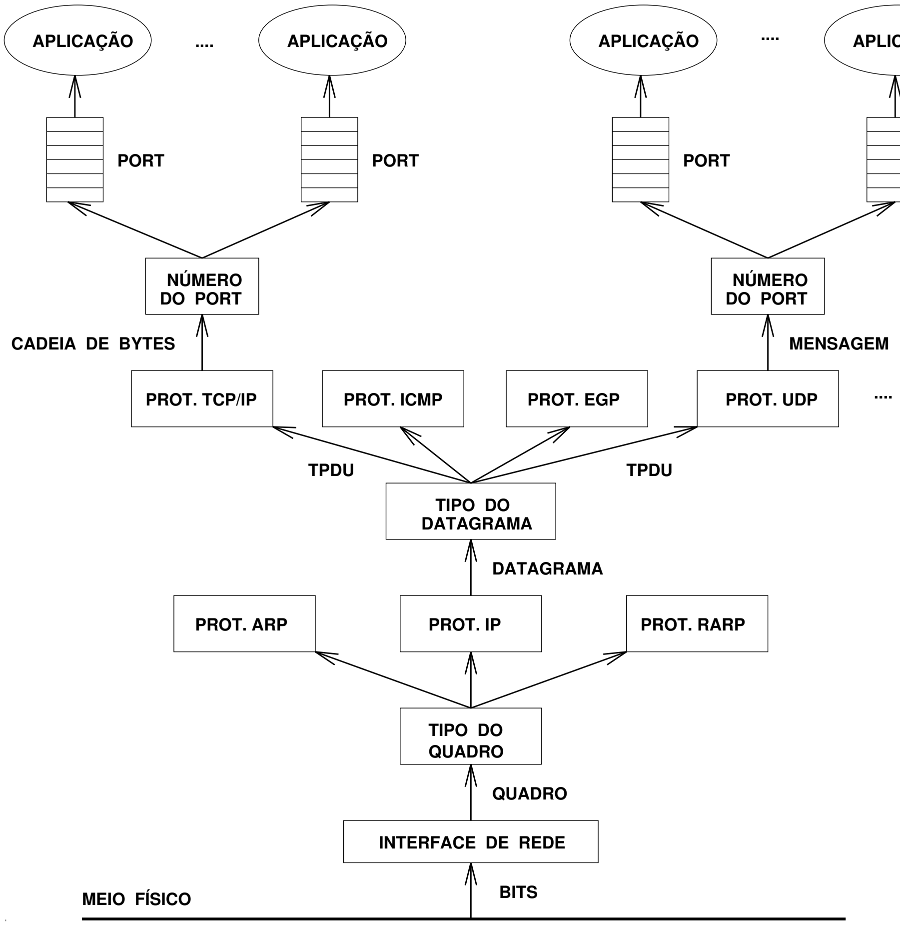

Figura 4.4: Ports permitem uma única instância de protocolo de transporte atender múltiplas aplicações.

Um port é dito *ativo* se através deste uma aplicação toma a iniciativa, via chamada de sistema, para o estabelecimento de uma conexão. Um dos parâmetros da chamada é o par (host, port) que constituirá o outro extremo da conexão. Este segundo port é dito *passivo*, sendo que a aplicação que o detém executa uma chamada de sistema “aceitando” o estabelecimento da conexão.

4.2.2 O Conceito de Dados Urgentes

Dados urgentes (Out-of-band data) são empregados quando uma mensagem deve ser tratada com maior prioridade. Seja o exemplo de um sistema de automação onde uma aplicação coleta dados de vários sensores, processa-os e os envia através de uma conexão de transporte a uma segunda aplicação para apresentação ao operador. Suponha que a aplicacão que coleta dados percebe um dado fora do limite normal (por exemplo, uma temperatura muito elevada). Esta informação deve ser tratada prioritariamente às demais.

Dado que a conexão de transporte honra a sequência em que os dados foram submetidos para transmissão, o dado urgente só será tratado quando todos os dados que o precedem tiverem sido consumidos. Para esta situação, o protocolo TCP/IP permite que dados enviados por uma conexão de transporte sejam taxados como "urgentes". Tais dados permanecem em sua posição relativa aos demais. Entretanto, a aplicação no outro extremo será notificada da existência de dados urgentes na conexão, bem como sua posição (a partir dos dados ainda não consumidos). Usualmente, a aplicação consome os dados[2] até que os dados urgentes atinjam o início da cadeia. A figura 4.5 ilustra este procedimento.

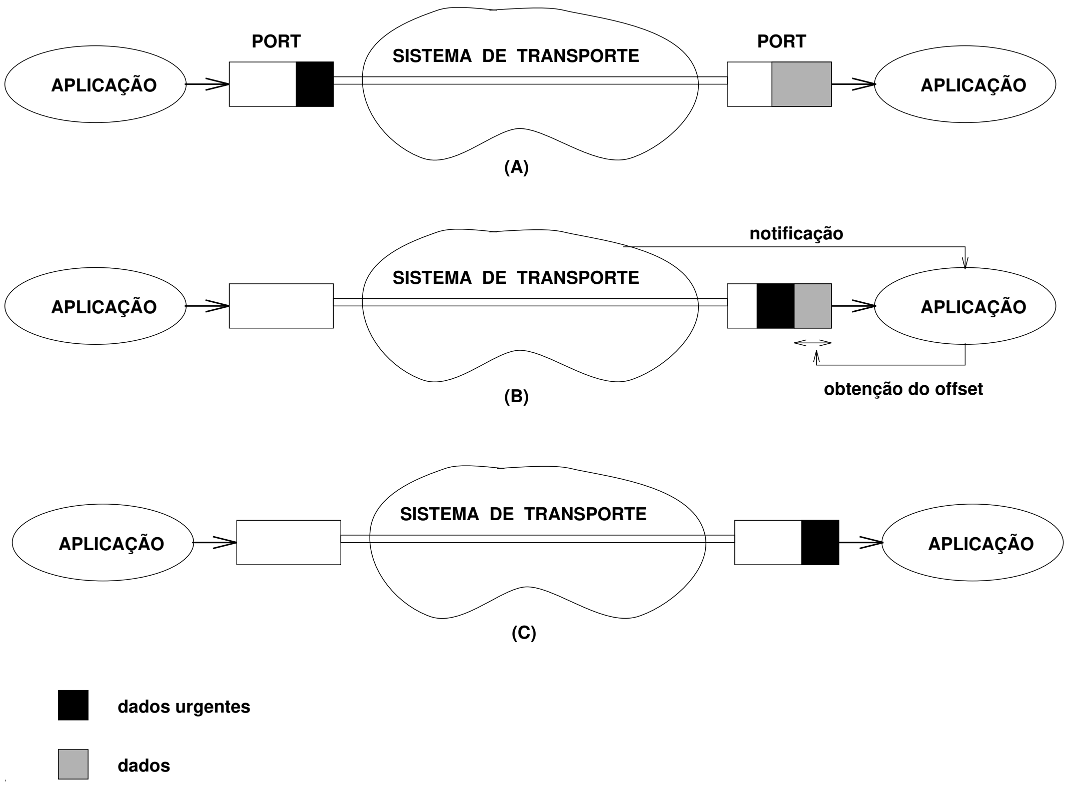

Figura 4.5: A sequência do processamento de dados urgentes. (A) - Aplicação submete dados urgentes. (B) - Dados urgentes chegam ao destino; aplicação é notificada; aplicação obtem a posição dos dados urgentes. (C) - Aplicação consome dados regulares até atingir os dados urgentes.

4.2.3 Cabeçalho do Protocolo TCP/IP

Mensagens trocadas via protocolo TCP/IP são transportadas no campo de dados de datagramas e possuem o cabeçalho dado pela figura 4.6.

Os campos PORT DE ORIGEM e PORT DE DESTINO identificam os ports da conexão de transporte. Note que o hosts não são estipulados no cabeçalho pois os mesmos estão presentes no cabeçalho do protocolo IP (veja figura 3.4).

[2]Amazenando-os para processamento futuro ou simplesmente descartando-os.

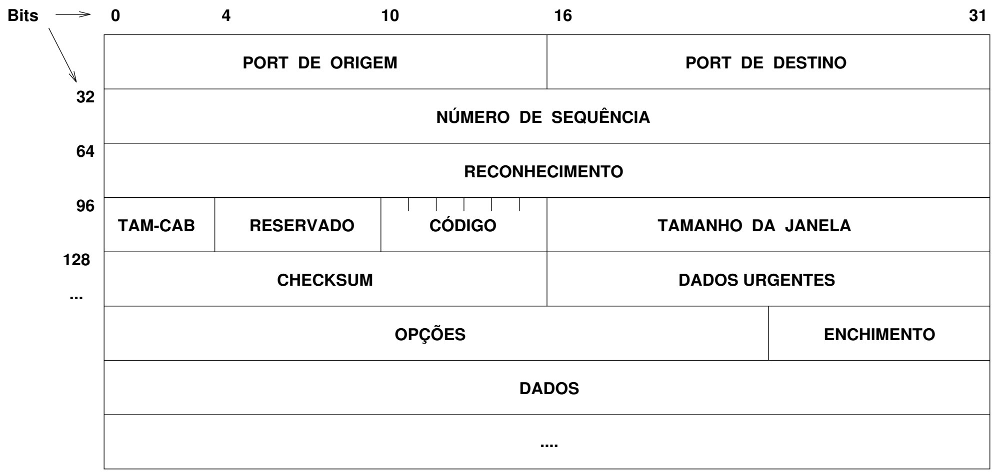

Figura 4.6: Cabeçalho do protocolo TCP/IP

O campo NÚMERO DE SEQUÊNCIA estabelece a posição dos dados que o pacote carrega com relação à cadeia de bytes transferida desde o estabelecimento da conexão.

O campo RECONHECIMENTO identifica a posição na cadeia de bytes que o emissor espera receber no próximo pacote a ele endereçado. Este campo provê reconhecimento expontâneo *(piggybacking)*.

O campo TAM-CAB contém o tamanho do cabeçalho em múltiplos de 32 bits. Este campo se faz necessário dado que o campo OPÇÕES possui tamanho variável.

O campo CÓDIGO possui seis bits assim distribuídos:

1. bit URG: o pacote carrega dados urgentes;
2. bit ACK: o pacote carrega um TPDU de reconhecimento;
3. bit PSH: solicita à camada de transporte um *push*, isto é, um envio sem "bufferização";
4. bit RST: solicita ao destinatário um *reset* da conexão;
5. bit SYN: solicita ao destinatário o estabelecimento de conexão;
6. bit FIN: solicita ao destinatário o encerramento da conexão.

O campo TAMANHO DA JANELA informa a quantidade de bytes o emissor deseja receber sem a necessidade de reconhecimento. Note que o tamanho da janela é dinâmico e não necessariamente idêntico nos dois sentidos da conexão.

O campo CHECKSUM contêm o cômputo do checksum para o cabeçalho mais dados.

O campo DADOS URGENTES especifica a posição na cadeia em que os dados urgentes terminam[3]. Este campo é válido apenas se o bit URG esteja ativo.

[3] Dados urgentes iniciam sempre no primeiro byte do campo de dados.

Finalmente, o campo OPÇÕES é utilizado para troca de informação entre as camadas de transporte que mantém a conexão. Uma informação importante é o tamanho máximo do segmento (MSS: Maximum Segment Size), que dita o tamanho máximo dos dados num pacote que o emissor está apto a processar. Usualmente, MSS possui um valor que é função do tipo de rede à qual o host se conecta. Cada rede possui um MTU (Maximum Tranfer Unit) que representa o número ótimo de bytes que se acomoda num quadro de enlace. Para redes Ethernet (IEEE 802.3), MTU tem valor de 1500 bytes; para redes FDDI (Fiber Distributed Data Interface) MTU tem valor de 4352 bytes; e assim por diante. Numa rede local, MSS é um múltiplo inteiro de MTU, função da quantidade de buffers o host dispõe. Para internets, MSS têm valor *default* de 536 bytes[4].

de 4288 bytes[5], mas pequenos microcomputadores ou determinadas subredes podem impor um tamanho menor para os pacotes.

4.2.4 Temporização e Controle de Congestionamento no Protocolo TCP/IP

O protocolo TCP/IP possui um mecanismo interno de temporização utilizado para fins de retransmissão de janelas não confirmadas. O desempenho do protocolo TCP/IP está intimamente relacionado com este mecanismo de temporização. Vimos anteriormente que a confiabilidade do protocolo TCP/IP está baseada no binômio reconhecimento-retransmissão. A cada segmento de janela enviado, o protocolo aguarda sua confirmação por um determinado intervalo de tempo. A medida que segmentos vão sendo reconhecidos a janela se move permitindo a transmissão de novos segmentos. Sempre que a falta de reconhecimento impedir o movimento da janela, um esquema de retransmissão é executado. Isto ocorre quando expirar o tempo de espera do reconhecimento para o primeiro segmento da janela, e, neste caso, toda a janela é retransmitida. A questão é: qual o valor do intervalo que o emissor deve aguardar a confirmação de um segmento ? Obviamente, este valor é função do tempo que um segmento leva até atingir o receptor mais o tempo que o reconhecimento leva para atingir o emissor[6]. Estes tempos são, em geral, equivalentes e a soma de ambos é denominado RTT (Round Trip Time: tempo de ida e volta).

O protocolo TCP/IP computa RTT para cada conexão aberta. Este cômputo é efetuado associando-se o valor do relógio interno a cada segmento transmitido. Recebido um reconhecimento, computa-se o quanto o relógio interno progrediu desde a transmissão do segmento. Em geral, a cada novo reconhecimento o valor de RTT é alterado através de um mecanismo de média móvel:

$$RTT = \alpha * RTT_{atual} + (1 - \alpha) * RTT_{medido}$$

Nas implementações primitivas, o *timeout* para fins de retransmissão era computado como:

[4]Comportas, por norma, devem manipular datagramas de pelo menos 576 bytes. O valor de 536 é este valor mínimo menos o tamanho dos cabeçalhos dos protocolos IP e TCP.

[5]Por razões de ordem prática, nunca o limite de 64 Kbytes que um datagrama pode transportar é empregado.

[6]Despreza-se aqui o tempo de processamento nos lados emissor e receptor.

$$timeout = \beta * RTT$$

Este mecanismo de temporização é deficiente por várias razões:

- não contempla altas variações de RTT. Esta situação ocorre quando o caminho percorrido pelos datagramas passa por múltiplas comportas;
- em caso de retransmissão o cômputo de RTT é dúbio: o reconhecimento refere-se ao primeiro segmento transmitido ou à uma subsequente retranmissão ? (ambiguidade do reconhecimento);
- não colabora para aliviar congestionamentos na rede. *Timeouts* podem indicar congestionamento e a pronta retransmissão da janela irá agravar ainda mais a situação de congestionamento.

A seguir examinaremos como as implementações mais modernas do protocolo TCP/IP contornam as deficiências acima.

Cômputo de RTT

Atualmente as implementações TCP/IP computam RTT levando em conta tanto a média das estimativas quanto sua variânça. A técnica sugerida para o cômputo de RTT é dada a seguir:

$$\widehat{RTT} = RTT_{medido} - RTT_{atual}$$

$$RTT = RTT_{atual} + \delta * \widehat{RTT}$$

$$DEV = DEV_{atual} + \rho * (|\widehat{RTT}| - DEV_{atual})$$

$$Timeout = RTT + \eta * DEV$$

As expressões acima são computadas em aritmética inteira por razões de desempenho. Os valores de δ, ρ e η são escolhidos a fim de facilitar as operações via operadores de deslocamento. Estudos mostram que $\delta = \frac{1}{2^3}$; $\rho = \frac{1}{2^2}$; e $\eta = 4$ permitem uma boa adaptação de RTT às variações do atraso de transmissão.

RTT Face à Retransmissões

A ambiguidade do reconhecimento é contornada com uma regra simples: não se computa RTT para segmentos retransmitidos. Entretanto, a ocorrência de uma retransmissão em geral indica uma situação de congestionamento. Neste caso, mantendo-se inalterado o valor de RTT pode causar retransmissão de todos os segmentos até que a condição de congestionamento seja superada. Para esta situação, utiliza-se uma técnica de retração (backoff). Numa situação de congestionamento, opera-se com um *timeout* dilatado:

$$Timeout = \gamma * Timeout_{atual}$$

onde γ usualmente possui valor 2. Este valor de *timeout* é mantido até que um novo valor de RTT seja computado.

A manutenção do valor de RTT somada a estratégia de retração é conhecida na literatura como Algoritmo de Karn.

Controle de Congestionamento

O protocolo TCP/IP emprega duas políticas de controle de congestionamento:

1. Decréscimo Multiplicativo;

2. Partida Lenta (Slow Start).

Para implementar estas políticas, o protocolo TCP/IP emprega duas janelas: a janela "negociada" (JN) estabelecida no campo TAMANHO DA JANELA do cabeçalho do protocolo; e uma janela de "congestionamento" (JC) computada dinamicamente pelo transmissor. O transmissor usa como janela (J):

$$J = min(JN, JC)$$

A técnica do Decréscimo Multiplicativo estipula que, ocorrido um *timeout*, o tamanho da janela de congestionamento é dividido por 2. Para os segmentos na janela ainda por transmitir, é utilizada retração no *timeout*. Esta técnica colabora para aliviar o congestionamento, pois diminuindo a janela diminui-se também o tráfego médio pela conexão.

A técnica da Partida Lenta evita que novas conexões causem congestionamento na rede. Assim que a conexão é estabelecida, a janela de congestionamento é feita igual a um tamanho máximo de segmento (MSS): $JC = MSS$. A cada reconhecimento recebido (sem necessidade de retransmissão) JC é aumentada de um MSS até atingir a metade da janela negociada (JN). Neste ponto, o protocolo entra numa fase de prevenção contra congestionamento. Nesta fase o incremento de um MSS só se dá caso todos os segmentos da janela sejam reconhecidos (istó é, caso a janela esteja avançando a uma taxa constante).

4.2.5 TCP/IP Sobre Redes de Alto Produto Banda*Atraso

O produto banda*atraso (BD: bandwidth*delay) de um segmento de rede define uma quantidade de bits em trânsito pela rede, sendo considerado alto quando supera a casa dos 10^6. Exemplo de redes com alto produto BD são as que utilizam *links* de satélite (devido ao alto atraso) e de fibra ótica (devido a alta banda). Protocolos de transporte em tais redes operando com janelas de tamanho padrão não conseguem manter a rede permanentemente ocupada (isto é, com bytes permanentemente em trânsito). Esta "lacuna de bytes" compromete a banda que uma conexão TCP/IP pode atingir em redes com alto produto BD.

Em particular, no protocolo TCP/IP as técnicas de temporização e controle de congestionamento apresentadas a seguir são ineficiantes para redes com alto produto BD. A literatura apresenta algumas possíveis extensões do protocolo, geralmente adicionando-se novas opções no cabeçalho sem a necessidade de alterar os campos padrão. Isto permite que implementações cujas extensões ainda não foram implementadas simplesmente ignorem estas novas opções e funcionem normalmente sem levá-las em conta. Duas estratégias compensatória para redes com alto produto BD são:

1. janelas de alta capacidade;
2. reconhecimentos seletivos.

Janelas de Alta Capacidade

O cabeçalho do protocolo TCP/IP reserva 16 bits para a especificação do tamanho da janela, estando portanto limitado em 64 Kbytes. Redes com alto produto BD podem abrigar várias janelas de 64 Kbytes em trânsito. Propõe-se aumentar este limite para além dos 64 KBytes para manter a rede permanentemente ocupada. Entretanto como o TCP/IP retransmite toda a janela, a perda de um pacote irá forçar uma retransmissão de uma vasta quantidade de bytes, comprometendo o desempenho da conexão. A proposta seguinte visa exatamente evitar a retransmissão de toda a janela.

Reconhecimentos Seletivos (NAKs)

A informação contida num reconhecimento enviado do receptor ao transmissor explicita única e exclusivamente o último byte recebido em sequência: um reconhecimento não informa sobre outros segmentos recebidos. Por exemplo suponha uma janela de 10 MSSs em que o primeiro e o sexto segmentos não atingiram o receptor. O receptor continuará reconhecendo o último byte da janela anterior, até a ocorrência de *timeout* seguida da retransmissão de janela segundo uma política de controle de congestionamento.

Reconhecimentos seletivos podem ser utilizados para confirmar negativamente (NAK: not acknowledged) o recebimento de segmentos. No exemplo anterior o receptor enviaria NAKs para os primeiro e sexto segmentos da janela, fazendo com que o transmissor retransmita apenas estes segmentos. NAKs permitem o envio seletivos de segmentos que não atingiram o receptor evitando tanto a retransmissão de toda a janela quanto a execução desnecessária de políticas de controle de congestionamento.

4.3 Padrões da Camada de Transporte: Protocolo UDP

O protocolo UDP (User Datagram Protocol) é orientado a datagrama, não suportando portanto conexões de transporte. O protocolo UDP é da classe 0, isto é, deixa a cargo da camada de rede toda a confiabilidade do transporte. Como o protocolo IP não provê nenhum mecanismo de confiabilidade, o protocolo UDP é mais empregado quando as aplicações que dele fazem uso executam em hosts de uma mesma rede local.

Como no TCP/IP, o protocolo UDP também se utiliza do serviço de entrega de datagramas da camada IP. Dado que o protocolo UDP não emprega nenhum mecanismo de reconhecimento, retransmissão e controle de fluxo, o mesmo possui um cabeçalho muito simples (figura 4.7).

Os campos PORT DE ORIGEM e PORT DE DESTINO especificam o port emissor e o port destinatário. No protocolo TCP/IP estes campos são empregados para identificar a

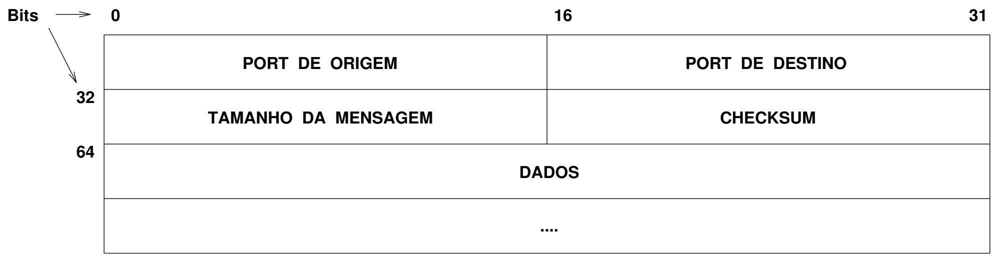

Figura 4.7: Cabeçalho do protocolo UDP

conexão de transporte (juntamente com os respectivos hosts) e garantir portanto a existência do receptor. Tal não ocorre no protocolo UDP: o destinatário é identificado apenas pelo seu port (e respectivo host). Caso este port inexista no host destino, o host simplesmente descarta a mensagem sem qualquer notificação ao emissor.

O campo TAMANHO DA MENSAGEM determina o tamanho em bytes da mensagem (cabeçalho mais dados). O campo CHECKSUM é utilizado para garantir a integridade do cabeçalho mais dados. Quando este campo for zero significa que o checksum não deve ser levado em conta. Não existe campo de opções no protocolo UDP.

Chapter 5

A CAMADA DE APLICAÇÃO

A camada de aplicação define um conjunto de serviços manipulados diretamente pelo usuário ou pelo administrador de sistema. Tais serviços são acessados através de protocolos e disponíveis em aplicativos diversos. Muitos destes aplicativos fazem parte do sistema, enquanto outros mais sofisticados são produtos comercializados por terceiros.

Na arquitetura TCP/IP os serviços da camada de aplicação utilizam a filosofia cliente-servidor. Os aplicativos são clientes de serviços definidos pelas implementações TCP/IP. Clientes e servidores se comunicam através de protocolos de aplicação que regem a interação cliente-servidor. Protocolos de aplicação se utilizam dos protocolos TCP/IP e UDP para o envio/recepção de suas mensagens.

Servidores provêem ports de comunicação para fins de acesso aos serviços. Tais ports são reservados e possuem número fixo independente da implementação (os chamados *ports notáveis*). A maioria das implementações TCP/IP definem um arquivo no diretório */etc* denominado *services*. Algumas linhas deste arquivo são apresentadas abaixo:

```
# @(#)services 1.16 90/01/03 SMI
#
# Network services, Internet style
#
tcpmux 1/tcp # rfc-1078
systat 11/tcp users
netstat 15/tcp
ftp-data 20/tcp
ftp 21/tcp
telnet 23/tcp
smtp 25/tcp mail
time 37/tcp timserver
time 37/udp timserver
name 42/udp nameserver
whois 43/tcp nicname # usually to sri-nic
sunrpc 111/udp
sunrpc 111/tcp
```

Por exemplo, o serviço de terminal virtual, denominado TELNET é provido no port 23 através do protocolo TCP/IP. Note que um mesmo serviço pode estar disponível através de diferentes protocolos de transporte com o mesmo número de port.

Serviços interativos de longa duração como *login* remoto e transferência de arquivos devem possuir um modo de operação de forma a evitar a indisponibilidade do serviço quando um cliente dele faz uso. Neste caso, o servidor quando recebe uma conexão de um cliente cria uma instância de si próprio[1] para interagir com o cliente, retomando seu estado de espera de conexão. Com isto, múltiplos clientes podem fazer uso do mesmo serviço simultaneamente.

A seguir apresentaremos os serviços mais comuns oferecidos pela arquitetura TCP/IP. Daremos ênfase em como o serviço é implementado, omitindo os detalhes do protocolo que governa o acesso ao serviço.

5.1 Terminal Remoto

Terminal remoto é um serviço através do qual um terminal conectado a um computador se apresenta ao usuário como se estivesse conectado a outro computador. Em outras palavras, a subrede de comunicação faz o papel de uma conexão física terminal-computador (como uma linha serial, por exemplo).

A implementação do serviço de terminal remoto é ilustrada na figura 5.1. O processo cliente coleta dados do terminal do usuário e os transforma numa representação segundo determinado padrão, enviando-os a um servidor remoto. O servidor converte da representação padronizada para sua representação interna; submete estes dados a um *pseudo terminal*[2]; e coleta sua resposta. Tal resposta (eco, *scroll*, etc) faz o caminho inverso dos dados para ser apresentada ao terminal do cliente.

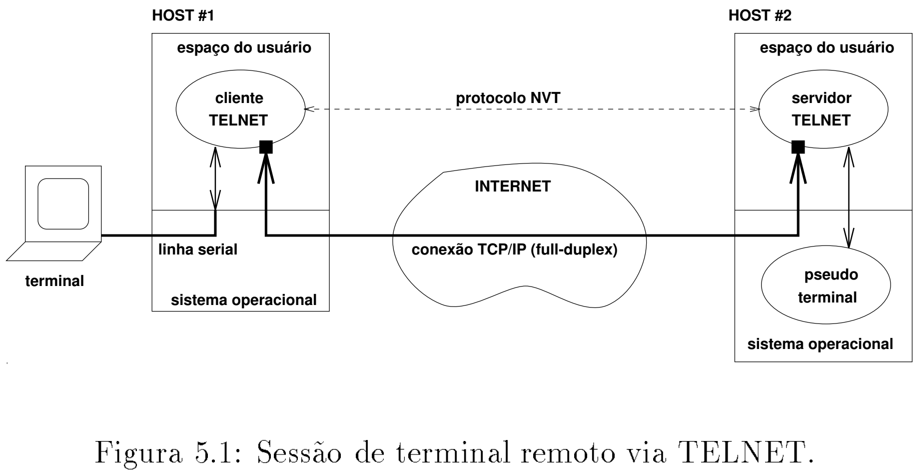

Figura 5.1: Sessão de terminal remoto via TELNET.

O protocolo TELNET é um padrão de comunicação entre cliente e servidor de terminal remoto. O protocolo define uma interface entre cliente e servidor denominada *Network*

[1] No UNIX, tal se dá pela chamada de sistema *fork*.

[2] Um processo que implementa um terminal lógico.

Virtual Terminal (NVT). NVT disciplina uma troca de dados orientada a byte. Os caracteres definidos pelo protocolo para troca de dados são baseados no padrão ASCII norte-americano (USASCII). O padrão USASCII define representação em bytes para caracteres de texto e de controle (como aqueles associados ao movimento do cursor, por exemplo).

Um problema presente na implementação de terminais remotos é o tratamento de caracteres especiais, como o CRTL-C. Via de regra os sistemas operacionais reservam determinadas sequências de caracteres para controle de execução. No UNIX os caracteres CRTL-C e CRTL-Z não são interpretados como entrada de dados, mas sim como, respectivamente, interrupção e passagem para *background* do programa em execução. A existência destes caracteres demanda um tratamento especial numa sessão de terminal remoto. Por exemplo, um CRTL-C deve interromper o programa executando no terminal remoto não no terminal local (o cliente TELNET !).

O protocolo TELNET redefine no lado do cliente os caracteres de interrupção como caracteres normais de dados. Em outras palavras, o sistema operacional local vai ignorá-los. Tais caracteres são codificados numa sequência não ASCII padronizada pelo NVT e enviados ao servidor remoto. O servidor os reconverte para a representação local e os submete ao pseudo terminal. Como o servidor não redefiniu estes caracteres como de dados, os mesmos irão ser tratados como tal pelo sistema operacional remoto (por exemplo, terminando o processo em execução).

O protocolo TELNET permite ainda a definição de parâmetros tal como o tipo de terminal utilizado pelo cliente. O protocolo TELNET é disponível na camada de aplicação através de programa interativo de mesmo nome.

5.2 Manipulação de Arquivos

Trataremos aqui de dois protocolos para manipulação de arquivos: FTP (File Transfer Protocol) e NFS (Network File System).

5.2.1 Acesso Interativo: Protocolo FTP

FTP é um protocolo para transferência de arquivos via Internet. O modo que o protocolo opera é ilustrado na figura 5.2. Duas conexões de transporte são estabelecidas: uma conexão utilizada para interação cliente-servidor (denominada conexão de controle) e outra exclusiva para transferência de arquivos (denominada conexão de dados). A conexão de dados permanece ativa durante toda a interação cliente-servidor.

O protocolo FTP utiliza as definições NVT para a interação cliente-servidor (via conexão de controle). O modo de transferência de arquivos pode ser: ASCII, binário ou EBCDIC. O modo de transferência é estabelecido pelo cliente.

Arquivos podem ser transferidos no sentido do host do cliente para o host do servidor ou vice-versa. O arquivo no host destino pode receber o mesmo nome do host de origem ou um outro nome escolhido pelo cliente. O protocolo permite ainda que o cliente opere diretórios tanto no lado do cliente quanto no lado do servidor (listando conteúdos, criando novos diretórios, etc).

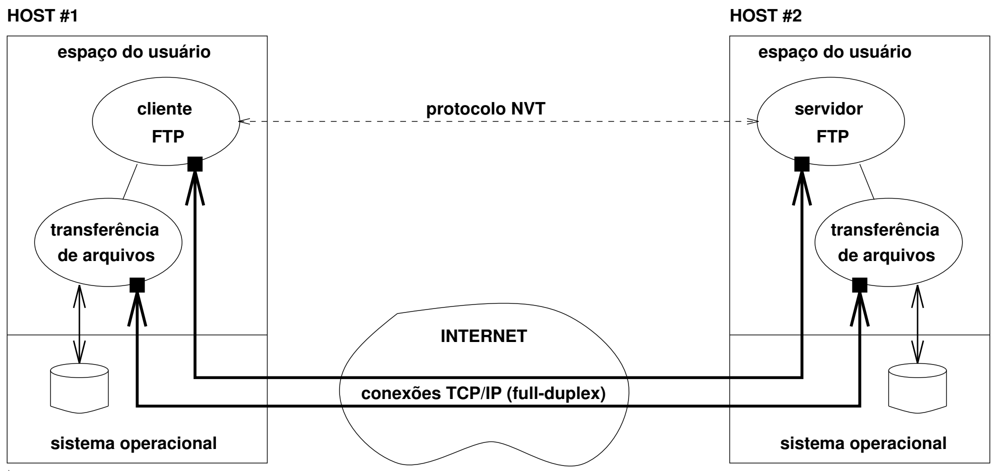

Figura 5.2: Esquema de transferência de arquivos via protocolo FTP.

Como o protocolo TELNET, o protocolo FTP é disponível na camada de aplicação através de programa interativo de mesmo nome.

5.2.2 Acesso On-line: Protocolo NFS

O protocolo NFS foi desenvolvido pela SUN Microsystems para compartilhamento *on-line* de arquivos. Diferente do acesso interativo provido pelo FTP, o acesso on-line é muito mais conveniente: permite a manipulação de arquivos remotos da mesma forma que arquivos locais. Como o protocolo não é proprietário da SUN, praticamente todos os fabricantes de computadores o adotaram para fins de interoperabilidade em termos de compartilhamento de arquivos.

O protocolo pode utilizar tanto o transporte sem conexão (UDP) como o transporte com conexão (TCP/IP). O transporte sem conexão é mais recomendado para redes locais ou próximas. Implementações do protocolo NFS sobre o UDP empregam seu próprio esquema de reconhecimento e retransmissão, tornando-o imune a eventuais falhas de comunicação.

O protocolo permite a servidores *exportar* sistemas de arquivos para um conjunto de clientes (ou para toda a Internet). Clientes podem *montar* estes sistemas de arquivos criando referências locais para os mesmos (sem copiá-los). A montagem de um sistema de arquivos remoto exige uma interação cliente-servidor para fins de autenticação.

A implementação do protocolo NFS exige alteração do sistema operacional tanto no lado cliente quanto no lado servidor. O protocolo supõe a existência de um *sistema de arquivos virtual* (ver figura 5.3). Quando um arquivo é referenciado o sistema de arquivos virtual examina se o mesmo faz parte do sistema de arquivos local ou pertence a um sistema de arquivos remoto (montado localmente). Se o arquivo for local o sistema de arquivos virtual deixa a cargo do sistema de arquivos local a sua manipulação. Caso contrário, se o arquivo pertencer a um sistema de arquivos remoto, o sistema de arquivos virtual interage com o servidor NFS para a sua operação.

A interação cliente-servidor NFS permite que todas as operações sobre arquivos locais sejam conduzidas sobre arquivos remotos de forma totalmente transparente. Por exemplo, um cliente NFS pode:

- abrir um arquivo remoto;
- ler dados de um arquivo remoto;
- gravar dados num arquivo remoto;
- remover um arquivo remoto.

A figura 5.3 ilustra a arquitetura necessária à implementação do protocolo NFS.

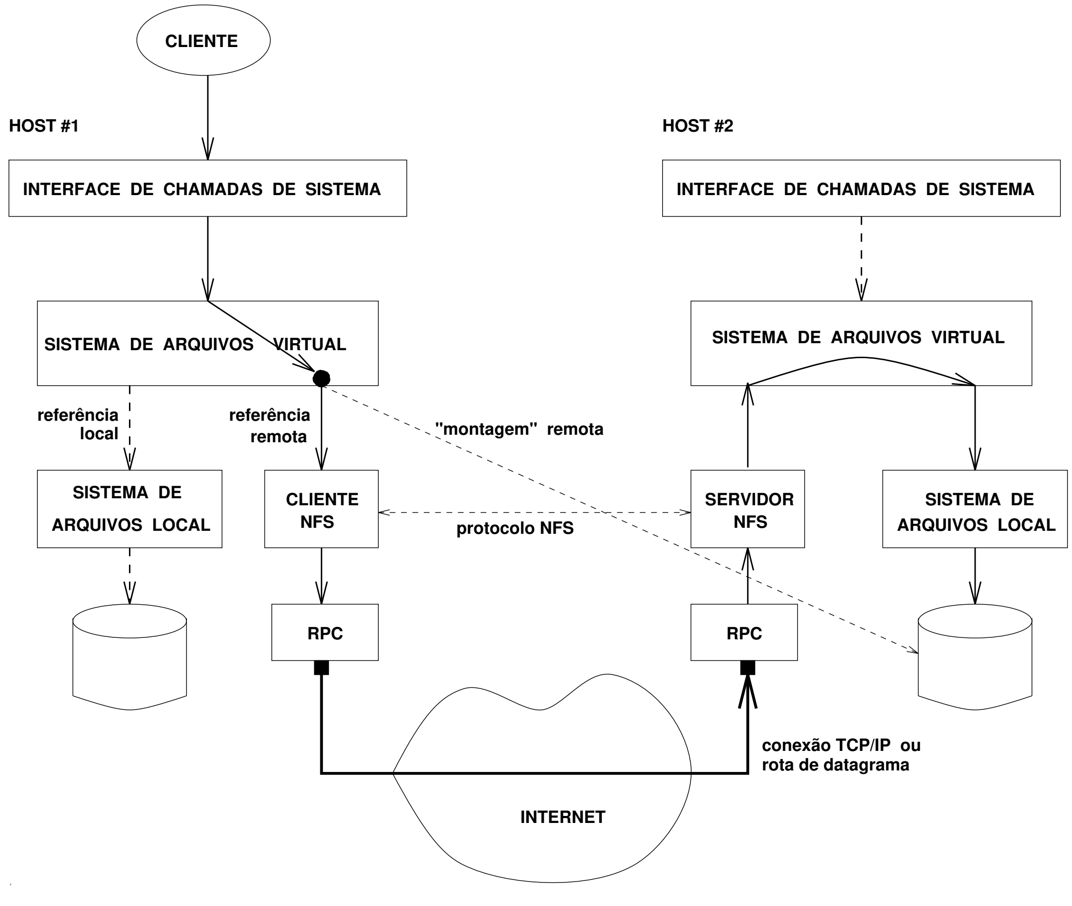

Figura 5.3: Arquitetura do NFS (Network File System).

A rigor, o protocolo NFS não é implementado diretamente sobre os protocolos de transporte. O NFS utiliza um protocolo de interação cliente-servidor denominado RPC (Remote Procedure Call), que por sua vez utiliza um protocolo de representação de dados denominado XDR (eXternal Data Representation). Tanto o RPC quanto o XDR foram introduzidos pela

SUN com o propósito de servir de base para o NFS. Muitos autores consideram o RPC como um protocolo de sessão (camada 5 do modelo OSI/ISO) e o XDR um protocolo de apresentação (camada 6 do modelo OSI/ISO).

5.3 Correio Eletrônico

Correio eletrônico é talvez a aplicação mais utilizada numa internet. O correio eletrônico permite um usuário num determinado host enviar mensagens (texto) a outro usuário em qualquer host da Internet. A arquitetura TCP/IP emprega um protocolo muito simples para transferência de mensagens, o SMTP (Simple Mail Transfer Protocol).

Cada usuário dispõe de uma área para armazenamento temporário de mensagens. Esta área é denominada *spool*. Quando o usuário envia uma mensagem através de um aplicativo, a mensagem é depositada no *spool* e permanece aguardando envio. Mensagens não são enviadas assim que são submetidas. Analogamente, quando o mesmo aplicativo é empregado para a leitura de mensagens, o *spool* do usuário é examinado para verificar a presença de novas mensagens.

Via de regra um mesmo programa atua como cliente e servidor de mensagens (no sistema UNIX trata-se do programa *sendmail*). O servidor de correio eletrônico é ativado em duas situações:

1. periodicamente pelo sistema operacional[3];
2. quando uma conexão por parte de um cliente remoto é requisitada para fins de transferência de mensagens.

No primeiro caso, o servidor examina as áreas de *spool* dos usuários e cada mensagem pendente é enviada (vide abaixo).

No segundo caso, o servidor aceita a conexão, recebe o texto da mensagem e armazena no *spool* do destinatário.

5.3.1 O Protocolo SMTP

Os aplicativos adicionam às mensagens de correio eletrônico um cabeçalho exemplificado abaixo:

```
From Jaime.Sichman@imag.fr Tue Nov  8 16:10:57 1994
Return-Path: <Jaime.Sichman@imag.fr>
Received: from imag.fr by dca.fee.unicamp.br (4.1/SMI-4.1)
id AA13594; Tue, 8 Nov 94 16:09:45 EDT
Date: Tue, 8 Nov 94 19:06:25 +0100
From: Jaime.Sichman@imag.fr (Jaime SICHMAN)
Message-Id: <9411081806.AA22714@cosmos.imag.fr>
```

[3] Usualmente a cada 60 segundos.

```
To: eleri@dca.fee.unicamp.br
Subject: Visita do prof. Demazeau ao Brasil
Cc: Yves.Demazeau@imag.fr
Status: RO
```

Os campos possuem o seguinte significado:

- From: o usuário que gerou a mensagem;
- Return-Path: o endereço Internet para réplica da mensagem;
- Received: hosts que armazenaram temporariamente a mensagem;
- Date: dia e hora em que a mensagem foi gerada;
- Message-Id: identificador da mensagem atribuído pelo host emissor;
- To: destinatário;
- Subject: assunto tratado no texto da mensagem;
- Cc: usuário(s) que recebeu cópia da mensagem;
- Status: status corrente.

O protocolo SMTP emprega mensagens ASCII contendo códigos e/ou palavras-chave. Mensagens são textos em ASCII e nenhum caractere especial deve fazer parte da mensagem. O protocolo requer que a mensagem termine pela sequência

```
<CR><LF>.<CR><LF>
```

Na transmissão da mensagem acima, o host emissor (imag.fr) conecta-se ao port 25 (servidor SMTP) do host receptor (dca.fee.unicamp.br). Estabelecida a conexão, cliente e servidor interagem via SMTP. Caso a conexão não possa ser estabelecida, o cliente mantém a mensagem no *spool* para envio futuro.

O exemplo abaixo ilustra a transferência de uma mensagem de correio eletrônico através do protocolo SMTP. A interação começa com um *handshake*.

```
dca:  220 dca.fee.unicamp.br Simple Mail Transfer Service Ready
imag: HELLO imag.fr
dca:  250 dca.fee.unicamp.br
```

A seguir o cliente identifica o emissor e o destinatário da mensagem:

```
imag: MAIL FROM:<Jaime.Sichman@imag.fr>
dca:   250 OK
imag: RCPT TO:<eleri@dca.fee.unicamp.br>
dca:  250 OK
```

Caso o destinatário não seja reconhecido pelo servidor, a sessão se encerra com a mensagem:

```
dca:  550 No such user here
```

Reconhecido o destinatário, o cliente solicita que servidor receba a mensagem:

```
imag: DATA
dca:  354 Start mail input; end with <CR><LF>.<CR><LF>
imag: From Jaime.Sichman@imag.fr Tue Nov  8 16:10:57 1994
      Return-Path: <Jaime.Sichman@imag.fr>
      ...
      Alo Eleri <CR><LF> Estou lhe enviando esta mensagem para ...
      ...
      Abracos <CR><LF> Jaime <CR><LF>.<CR><LF>
dca:  250 OK
```

Finalmente, a sessão se encerra com o fechamento da conexão TCP/IP de ambos os lados.

```
imag: QUIT
dca: 221 dca.fee.unicamp.br Service closing transmission channel
```

5.4 Gerenciamento de Redes

Gerenciamento de redes é uma atividade ligada à administração de sistemas, não ao usuário final. A atividade de gerenciamento inclui:

- estatísticas de uso da rede;
- localização e correção de problemas;
- atualizações e ajustes da estrutura da rede.

Como exemplo destas atividades temos: determinar o número médio de pacotes que circula pela rede; determinar se um host está ativo; introduzir novas tabelas de roteamento. A arquitetura TCP/IP tem nas comportas seu ponto crítico e, justamente por esta razão, a atividade de gerenciamento numa Internet recai mais sobre as comportas que sobre os hosts.

Em redes homogêneas o gerenciamento se dá em níveis próximos à camada física, permitindo um total controle dos hosts e comportas mesmo que as camadas acima falhem. Em redes heterogênes, a diversidade de equipamentos e formas de interconexão obriga a atividade de gerenciamento se processar no nível da camada de aplicação. Esta estratégia permite uma cobertura maior das atividades de gerenciamento, mas apresenta a desvantagem de depender do funcionamento de todas as camadas abaixo da camada de aplicação.

A arquitetura para gerenciamento de redes TCP/IP é apresentada na figura 5.4 e também segue o modelo cliente-servidor. Em cada comporta e host existe um servidor de gerenciamento (SG) que provê informações coletadas por um cliente. O cliente é parte de um aplicativo de gerenciamento que interage com os servidores através de um protocolo específico.

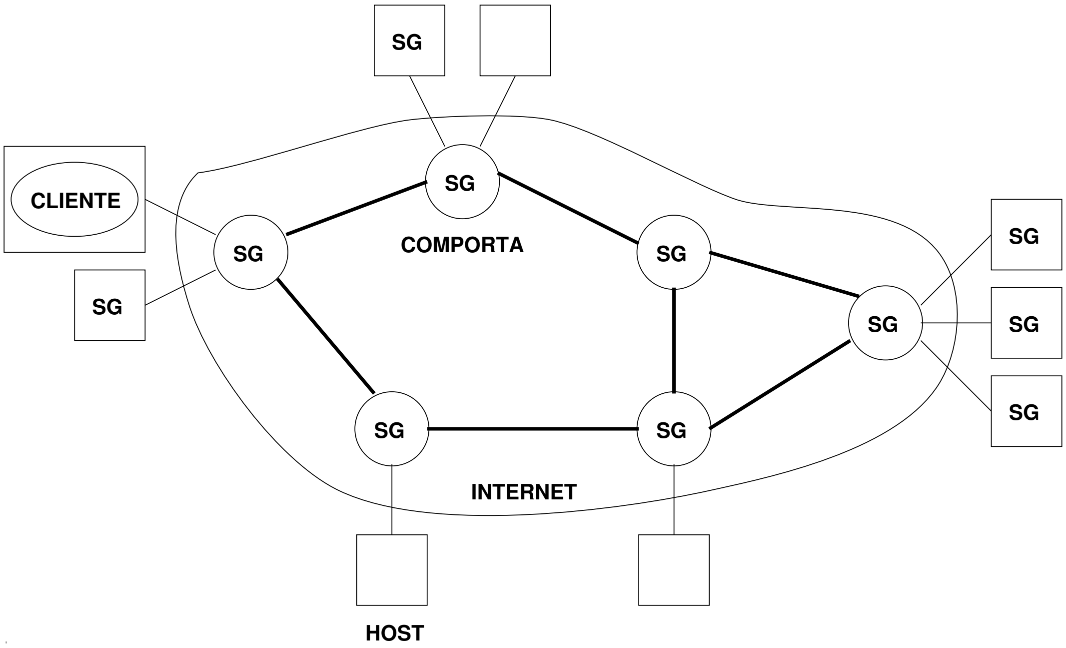

Figura 5.4: Arquitetura para gerenciamento de redes TCP/IP. SG significa a presença de um servidor de gerenciamento no host ou comporta.

Um processo de autenticação existe de forma a restringir as atividades de gerenciamento ao domínio do qual o administrador faz parte.

Examinaremos dois protocolos de gerenciamento: SNMP (Simple Network Management Protocol) e CMOT (Common Management Information Over TCP/IP). Ambos os protocolos determinam que comportas e hosts dotados de capacidade de gerenciamento mantenham informações de estado que clientes possam vir a acessar.

Tanto o SMNP quanto o CMOT se baseiam num padrão denoninado MIB (Management Information Base). MIB divide o gerenciamento nas categorias listadas abaixo, sendo que cada categoria suporta determinado número de objetos "gerenciáveis":

sistema : o sistema operacional do host ou comporta (3 objetos).

interfaces : conexão a rede(s) (22 objetos);

translação de endereços : mapeamentos de endereços (3 objetos);

ip : software que implementa o protocolo IP (33 objetos);

icmp : software que implementa o protocolo ICMP (26 objetos);

tcp : software que implementa o protocolo TCP/IP (17 objetos);

udp : software que implementa o protocolo UDP (4 objetos);

egp : software que implementa o protocolo EGP (6 objetos).

O propósito do padrão MIB é balisar que tipo de informação os protocolos de gerenciamento comumente necessitam. A tabela 5.1 ilustra o tipo de informação presente em cada categoria.

Variável MIB	Categoria	Significado
sysUpTime	sistema	tempo decorrido do último reboot
ifNumber	interfaces	Número de interfaces de rede conectadas
ifMtu	interfaces	MTU para uma particular interface
ipDefaultTTL	ip	Valor do campo TEMPO DE VIDA no datagrama
iplnReceives	ip	Número de datagramas IP recebidos
ipForwDatagrams	ip	Número de datagramas IP roteados
ipOutNoRoutes	ip	Número de falhas de roteamento
ipReasmOKs	ip	Número de datagramas IP remontados
ipFragOKs	ip	Número de datagramas IP fragmentados
ipRoutingTable	ip	Tabela de roteamento em uso
icmplnEchos	icmp	Número de requisições ICMP de eco recebidas
tcpRtoMin	tcp	Máximo tempo para retransmissões no TCP/IP
tcpMaxConn	tcp	Máximo número de conexões TCP/IP permitidas
tcpInSegs	tcp	Número de segmentos TCP/IP recebidos
udpInDatagrams	udp	Número de datagramas UDP recebidos
egpInMsgs	egp	Número de mensagens EGP recebidas

Tabela 5.1: Algumas variáveis MIB disponíveis aos protocolos de gerenciamento.

5.4.1 Protocolo SNMP

O protocolo SNMP permite um cliente inspecionar uma variável MIB ou alterar seu conteúdo num determinado host ou comporta. A determinação de quais variáves MIB devem ser lidas ou alteradas é tarefa dos aplicativos de gerenciamento de rede, não do administrador de sistema. O aplicativo interage com o administrador num nível muito mais alto que o SNMP.

O protocolo SNMP estabelece que as mensagens trocadas entre clientes e servidores devem ser codificadas na notação ASN.1 (Abstract Syntax Notation One) padronizada pela ISO. ASN.1 representa um conjunto de atributos numa notação similar a estruturas em C ou registros em Pascal.

O protocolo SNMP define apenas cinco operações:

get-request : acessa um valor de determinada variável MIB;

get-next-request : acessa a próxima variável MIB a partir da variável especificada na mensagem;

get-response : responde a uma requisição de acesso;

set-request : armazena um valor numa variável MIB;

trap : envia resposta ante a ocorrência de um evento (exemplo: reportar uma falha numa linha de comunicação).

A figura 5.5 ilustra a interação cliente-servidor no protocolo SNMP.

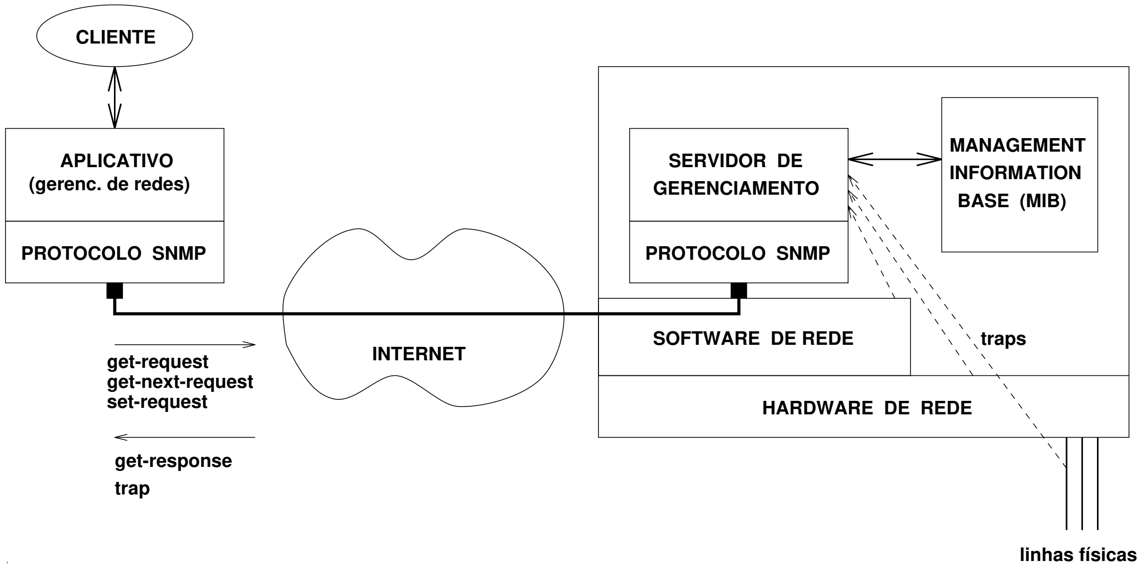

Figura 5.5: Interação cliente-servidor no protocolo SNMP.

5.4.2 Protocolo CMOT

CMOT é uma implementação do protocolo CMIP (Common Management Information Protocol) para redes TCP/IP. CMIP é um protocolo ISO e portanto aderente ao modelo de referência OSI. Do ponto de vista estrutural, o protocolo CMOT é similar ao SNMP: utiliza bases de informação de gerenciamento (MIBs) e codifica as mensagens na notação ASN.1.

CMOT divide os componentes de gerenciamento em *gerentes* e *agentes*. Os agentes coletam informação de gerenciamento, executam comandos e efetuam testes. Os gerentes recebem informação dos agentes, processam esta informação e geram comandos para os agentes.

Cada camada do modelo OSI define certas entidades de gerenciamento, as LME (Layer Management Entities). As LMEs são mantidas por processos de aplicação denominados SMAP (System Management Application Process) que se comunicam via protocolo CMIP. Os SMAP podem ter o papel de gerente ou agente dependendo de quem está interagindo diretamente com a aplicação de gerenciamento de redes. A figura 5.6 ilustra a arquitetura CMOT.

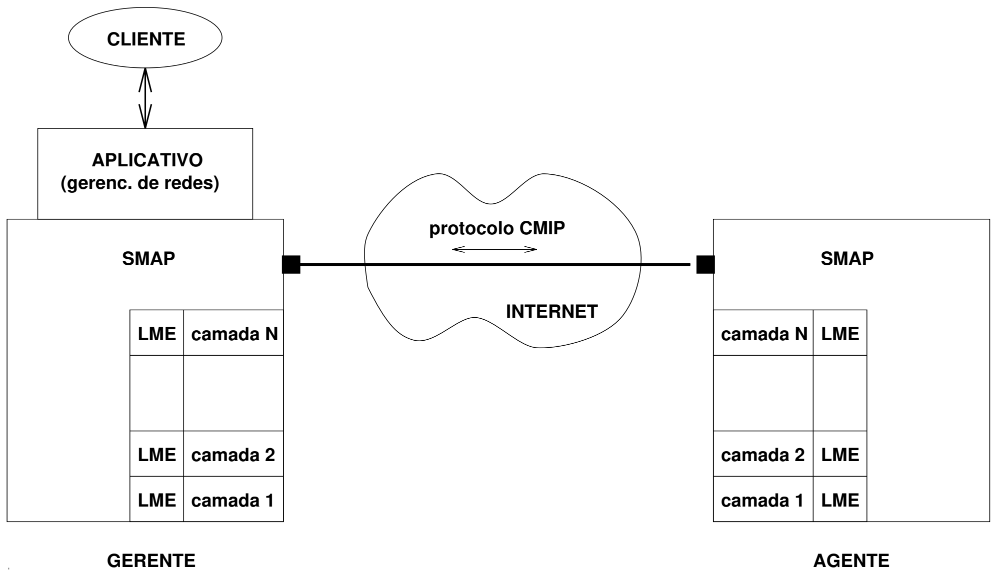

Figura 5.6: Arquitetura CMOT.

5.5 Hipertexto/Hipermídia

Encontra-se em vias de padronização protocolos para permitir o acesso a documentos hipermídia numa internet. Tais documentos combinam texto, imagens, áudio e vídeo e vem substituindo comunicação pela Internet via texto. Protocolos hipermídia suportam, por exemplo, as seguintes aplicações:

- mensagens de correio eletrônico onde o texto cede lugar ao áudio e vídeo;
- acesso *on-line* a publicações técnicas e artísticas;
- acesso *on-line* a eventos (simpósios, mostras, etc.);
- educação e treinamento (cursos *on-line*);
- oferecimento de produtos e serviços através de catálogos *on-line*;
- acesso a obras raras (pinturas, textos, etc.) digitalizadas a partir do original.

5.5.1 O Protocolo HTTP

HTTP (Hypertext Transfer Protocol) é um protocolo de representação de documentos hipertexto/hipermídia. Tais documentos são armazenados em servidores denominados WWW (World Wide Web) e transferidos via Internet para apresentação no terminal do cliente. Toda a composição do documento é realizada pelo cliente, cabendo ao servidor apenas seu armazenamento e localização.

Os documentos transferidos pelo protocolo HTTP são identificados por um localizador representado no padrão URL (Uniform Request Locator). Este padrão localiza o documento através do host e do diretório no qual está armazenado. Por exemplo, o localizador

```
http://dca.fee.unicamp.br/artigos/indice.html
```

aponta para o documento *indice.html* localizado no host *dca.fee.unicamp.br* sob o diretório *artigos*.

HTTP define mensagens de requisição e resposta com um cabeçalho baseado em texto (como no SMTP).

Documentos hipertexto/hipermídia são compostos numa linguagem denominada HTML (Hypertext Markup Language). HTML integra:

1. texto e hipertexto;
2. imagens estáticas nos mais variados formatos (JPEG, TIFF, etc.);
3. áudio e vídeo nos mais variados formatos (MPEG, AVS, etc.);

HTML define um conjunto de comandos para compor um documento. Os comandos têm funcionalidades similares aos comandos básicos do LaTeX mais uma sintaxe que permite ligar uma palavra-chave do texto a um outro objeto (texto, imagem, documento, etc).

A figura 5.7 ilustra um trecho de um documento no formato HTML e a figura 5.8 exibe este documento através do software *mosaic* (que implementa o lado cliente do protocolo HTTP e um interpretador HTML).

5.6 Resolução de Nomes

Vimos na seção 2.4 que diferentes camadas utilizam diferentes formas de endereçamento. A camada física utiliza endereços de hardware codificados numa sequência de bits (48 bits para redes Ethernet). A camada inter-redes utiliza endereços numéricos de 32 bits na forma (classe, subrede, host) comumente representados em notação decimal do tipo 143.106.11.129. Para processos de aplicação que interagem com o ser humano, uma forma de endereçamento mais abstrata (simbólica) se faz necessária. Por exemplo, o aplicativo *telnet* requer o endereço do host onde a sessão de terminal remoto se desenvolverá. Assim,

```
telnet 143.106.11.129
```

solicita a abertura de sessão de terminal remoto no host cujo endereço IP é 143.106.11.129.

Hosts conectados numa internet possuem nomes simbólicos do tipo *leblon, maxwell, brahma,* etc. Via de regra, uma organização atribui nomes de hosts segundo um tema. Por exemplo, o Departamento de Engenharia de Computação e Automação Industrial (DCA) da Faculdade de Engenharia Elétrica (FEE), UNICAMP, atribui nomes aos hosts segundo o tema *praias* onde os hosts são denominados *leblon, ubatuba, buzios,* etc. Com isto, um usuário pode executar o aplicativo *telnet* passando o endereço simbólico ao invés do endereço IP:

```
<HEAD><TITLE>DCA - FEE - Unicamp Home Page</TITLE></HEAD>
<BODY>

<h3>Department of Computer Engineering and Industrial Automation</h3>
<he>School of Electrical Engineering - State University of Campinas</h3>
<IMG ALIGN=MIDDLE src="/images/dca.gif">
<IMAG ALIGN=MIDDLE src="/images/logo.gif>
<h1> Welcome to D C A</h1>
<hr>

<h4> Pressione <a href="Benvindo.html>aqui</a> para obter vers&atilde;o em
Portugu&ecirc;s desta p&aacute;gina.</h4>

<HR><P>

This is the <a ref=http://info.cern.ch/hypertext/WWW/LineMode/Defaults/default.html>
World Wide Web</A> home page for the
Department of Computer Engineering and Industrial Automation (DCA) of the
School of Electrical Engineering(<A href="http://www.fee.unicamp.br">FEE</A>)
at the University of Campinas
<A href="http://www.unicamp.br/">(UNICAMP)</A>.
We are located in
<A href="http://www.iqm.unicamp.br/about-campinas.html">Campinas</A>,
S&atilde;o Paulo, <A href=http://www.iqm.unicamp.br/brasil.html>Brazil</A>. <P>
```

Figura 5.7: Exemplo de documento HTML.

```
telnet leblon
```

Obviamente em algum momento o aplicativo necessitará do endereço IP para proceder a conexão com o servidor TELNET.

Endereçamento simbólico impõe dois problemas:

1. como atribuir nomes ?

2. como resolver nomes (descobrir o correspondente endereço IP) ?

5.6.1 O Conceito de Domínio

Numa vasta internet é impossivel atribuir nomes simbólicos com a garantia de unicidade ou resolver nomes de forma eficaz. A solução encontrada foi descentralizar tanto a atribuição quanto a resolução de nomes. Nomes são compostos segundo uma hierarquia de *domínios*. Domínios se dividem em geográficos e oficiais. Domínios geográficos possuem o código do país no nível mais alto (br para o Brasil).

Domínios oficiais possuem os seguintes códigos no nível mais alto:

com : organizações comerciais;

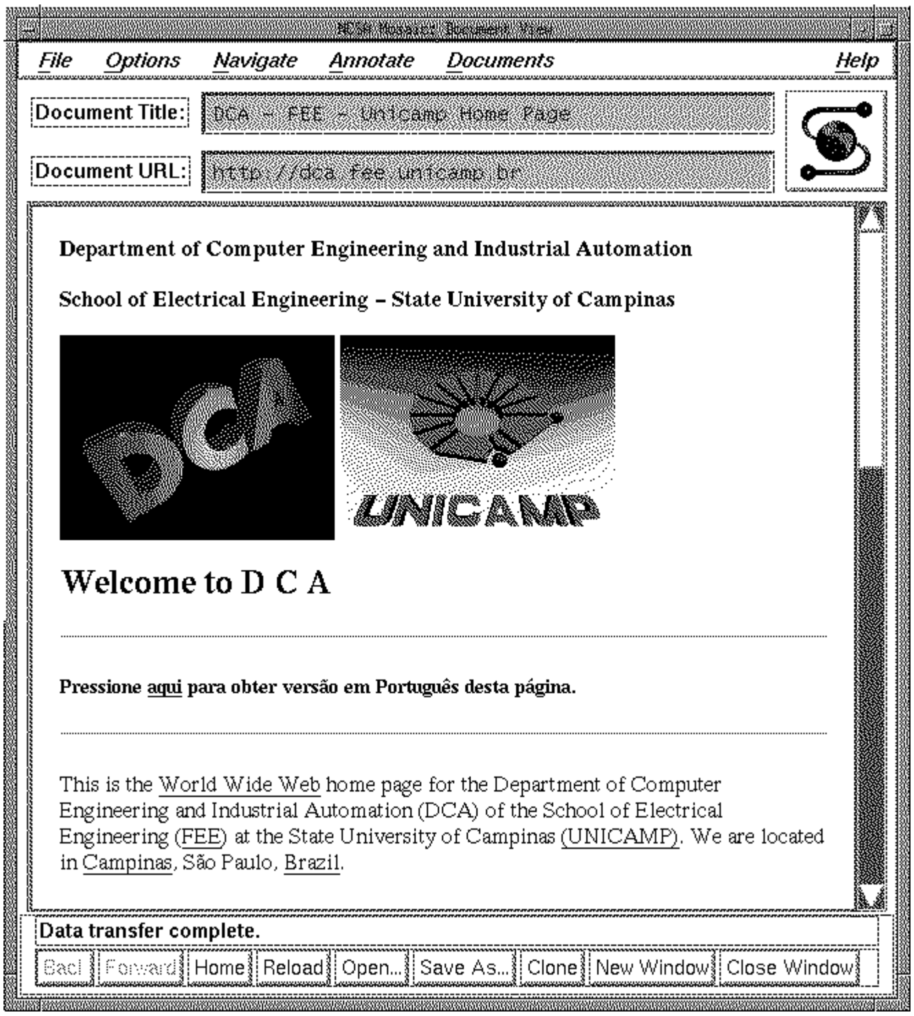

Figura 5.8: Documento HTML apresentado pelo *mosaic*.

edu : instituições de ensino;

gov : instituições governamentais civis;

mil : instituições militares;

net : centros de suporte e controle da Internet;

org : organizações que não se enquadram nas categorias acima;

int : organizações internacionais.

Domínios oficiais surgiram com a rede ARPANET nos Estados Unidos e são mais utilizados naquele país. Nos demais países, adota-se preferencialmente a representação geográfica.

A partir do nível mais alto, os domínios se dividem em *sub-domínios* como ilustra a figura 5.9. Por exemplo, UNICAMP é um sub-domínio do domínio BR; FEE é um sub-domínio de UNICAMP e DCA é um sub-domínio de FEE. Assim, para se evitar conflitos na atribuição de nomes simbólicos, adiciona-se a hierarquia de domínios ao nome. Por exemplo,

```
leblon.dca.fee.unicamp.br
```

se refere ao host de nome simbólico *leblon* pertencente ao domínio DCA contido no domínio FEE contido no domínio UNICAMP contido no domínio BR.

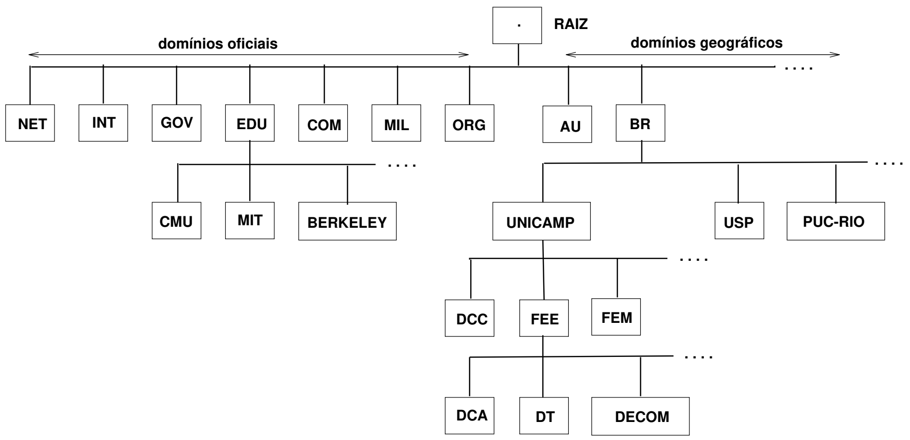

Figura 5.9: Hierarquia de domínios na Internet.

Domínios são atribuídos de forma independente da topologia da rede. Por exemplo, um domínio pode operar várias subredes sendo o inverso também permitido. A entidade responsável pela atribuição de domínios no nível mais alto é o InterNIC (Internet Network Information Center), entidade financiada pelo governo dos Estados Unidos. No Brasil, a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) é a entidade responsável pela atribuição de domínios no segundo nível e faixa de números IP para os hosts do domínio. A partir dai, as entidades estabelecem seus sub-domínios e os endereços IP dentro da faixa estipulada para o domínio.

5.6.2 Mapeamento Nome Simbólico - Endereço IP

O mapeamento nome-endereço IP é feito de forma descentralizada por um sistema denominado Sistema de Nomes por Domínio (DNS: Domain Name System). Neste sistema, um conjunto de *servidores de nomes* cooperam na resolução de nomes. Cada domínio possui um ou mais servidor de nomes. Este servidor é capaz de resolver os nomes pertencentes ao domínio ou encontrar um servidor mais apropriado para a resolução.

Ao receber uma requisição para resolução de nome, o servidor inicia a resolução do domínio mais específico para o mais geral. Por exemplo, seja o comando

```
telnet brahma
```

a partir do host jureia.dca.fee.unicamp.br. O host *jureia* solicita ao servidor de nomes do domínio DCA que resolva o nome *brahma*. Não sendo um nome local, o servidor sugere que a requisição seja direcionada para o servidor do domínio FEE[4]. Este servidor é capaz de resolver todos os nomes do domínio e descobre que o host *brahma* pertence ao domínio DENSIS (Departamento de Engenharia de Sistemas), possui nome completo brahma.densis.fee.unicamp.br e endereço IP 143.106.12.132.

Obviamente, nem todos os servidores possuem capacidade de resolução completa de nomes. Por exemplo, o servidor de nomes do domínio BR teria dificuldades de manter de forma atualizada todos os nomes e endereços IP dos hosts localizados no Brasil. Neste caso, o servidor inicia um processo de resolução reversa. Considere o comando

```
telnet caxambu.dcc.ufmg.br
```

a partir do host jureia.dca.fee.unicamp.br. Neste exemplo, os servidores dos domínios DCA, FEE e UNICAMP são incapazes de resolver o nome. O servidor do domínio UNICAMP sugere o servidor do domínio BR localizado na FAPESP que irá resolver no sentido inverso. Este servidor identifica um servidor para o domínio UFMG (Universidade Federal de Minas Gerais) e o repassa para o solicitante. Contactado este servidor, o mesmo sugere um servidor abaixo capaz de resolver nomes do domínio DCC (Departamento de Ciência da Computação). Finalmente, após contactado este servidor supre o endereq IP do host *caxambu* (150.164.10.134). A figura 5.10 ilustra o processo.

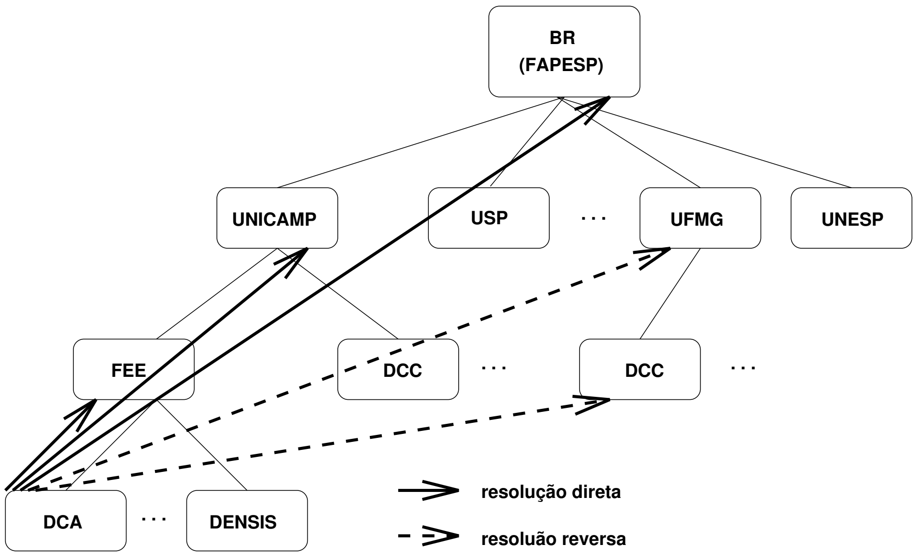

Figura 5.10: Exemplo de resolucão do endereço caxambu.dcc.ufmg.br a partir do domínio dca.fee.unicamp.br

[4]Neste caso temos a chamada resolução *não recursiva*. Na resolução recursiva o próprio servidor de nomes se encarrega de redirecionar a requisição a outro servidor.

Usualmente os servidores de nomes, além de resolver os nomes de seu domínio são capazes de resolver alguns nomes pertencentes a domínios adjacentes. Estes nomes usualmente pertencem a máquinas "importantes" tais como comportas e servidoras.

Servidores de nomes mantem cache em disco sobre os últimos nomes que foram resolvidos externamente. A existência de cache otimiza o processo de resolução de nomes como um todo, minimizando o redirecionamento de requisições.

5.6.3 Arquitetura do DNS

O Domain Name System (DNS) é estruturado segundo dois componentes: um servidor de nomes e um "resolvedor" de nomes. O servidor de nomes é implementado como um processo[5] que executa nas máquinas dotadas de capacidade de resolução de nomes (tipicamente comportas). O port notável 53 é reservado aos servidores DNS. O resolvedor de nomes é uma biblioteca de funções ligada a um programa de aplicação[6]. Estas funções basicamente executam *queries* nos servidores. Resolvedores e servidores se comunicam via protocolos UDP ou TCP/IP.

Um *query* a um servidor DNS pode obter como resposta o endereço IP procurado ou o endereço IP de um servidor "mais próximo". Vê-se assim que a resolução de nomes é um processo recorrente que termina com o endereço IP procurado ou com uma indicação de erro (nome inexistente).

Servidores DNS se classificam como:

1. primários: mantém tabelas atualizadas nome-endereço IP sobre um domínio. Existe um único servidor primário por domínio (usualmente executando na comporta de entrada do domínio);

2. secundários: mantém cópias das tabelas de servidores primários localizados em domínios superiores (usualmente um ou dois níveis acima). Cópias de tabelas são processadas periodicamente;

3. *cache-only*: não mantém tabelas de mapeamento, armazenando em cache as ultimas resoluções obtidas de servidores primários ou secundários. São capazes de resolver apenas os nomes armazenados em sua memória cache.

Administradores de sistema mantém tabelas de mapeamento atualizadas apenas nos servidores primários. Estas tabelas são disseminadas automaticamente para outros servidores via cópia ou armazenamento em cache. Uma tabela de nomes é armazenada num arquivo de extensão *zone* e nome coincidente com o nome do domínio (exemplo: fee.zone). Um host é identificado neste arquivo conforme ilustrado abaixo:

```
; hostname    domain   type    resource record data
;
s302           IN      A       143.106.11.188
               IN      HINFO   "PC-486" "OS/2 V2.11"
s302           IN      CNAME   s302.dca.fee.unicamp.br.
```

[5] *named* no UNIX.

[6] Por exemplo, o *nslookup* no UNIX.

Neste exemplo, o host de nome s302 pertence ao domínio IN (Internet); possui como endereço IP 143.106.11.188 (A: address); é um hardware PC-486 rodando OS/2 V2.11 (HINFO: host info); e tem como nome canônico (CNAME) s302.dca.fee.unicamp.br.

5.6.4 Mapeamento Inverso

Pode ocorrer que uma aplicação deseje conhecer o nome simbólico de um host a partir de seu endereço IP. Por exemplo, a chamada *gethostbyaddr* no UNIX retorna informações sobre um host (inclusive seu nome simbólico) a partir de seu endereço IP. Temos agora um mapeamento inverso que é processado também pelo DNS. Nese caso o DNS utiliza um arquivo de mapeamento reverso (exemplo: fee.rev). Neste arquivo, um host com o endereço IP 143.106.11.188 possui como entrada:

```
; hostname                      domain   type    resource record data
;
188.11.106.143.in-addr.arpa     IN        PTR    s302.dca.fee.unicamp.br
```

Em caso de mapeamento inverso, o servidor DNS resolve como numa resolução direta procurando pelo host formado pela conjunção de seu endereço IP em notação decimal reversa e do domínio fictício in-addr.arpa.

BIBLIOGRAFIA

1. Comer, D.E., *Internetworking with TCP/IP - Volume I: Principles, Protocols and Architecture*, Prentice-Hall International Editions, 1991.

2. Comer, D.E., Stevens, D.L., *Internetworking with TCP/IP - Volume II: Design, Implementation and Internals*, Prentice-Hall International Editions, 1991.

3. Comer, D.E., Stevens, D.L., *Internetworking with TCP/IP - Volume III: Client-Server Programming and Applications*, Prentice-Hall International Editions, 1993.

4. Feit, S., *TCP/IP - Architecture, Protocols and Implementation*, McGraw-Hill, 1993.

5. Hunt, C., *TCP/IP Network Administration*, O'Reilly & Associates, Inc., 1992.

6. Kochan, S.G., Wood, P.H., *Unix Networking*, Hyden Books, 1989.

7. Rose, M.T., *The Simple Book - An Introduction to Management of TCP/IP-based Internets*, Prentice-Hall, 1991.

8. Sun Microsystems Inc., *Networking Programming Guide*, 1990.