REDES DE COMPUTADORES: MODELO OSI/X.25

Eleri Cardozo
Maurício F. Magalhães
Departamento de Engenharia de Computação
e Automação Industrial
Faculdade de Engenharia Elétrica
Universidade Estadual de Campinas

1996

Índice

Capítulo 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 e LANs a nível regional (5 - 100 Km), usualmente empregando uma ou mais redes de alto desempenho interconectadas.

4. Redes de Longa Distância (WAN: Wide Area Network): interconectam computadores e LANs 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).

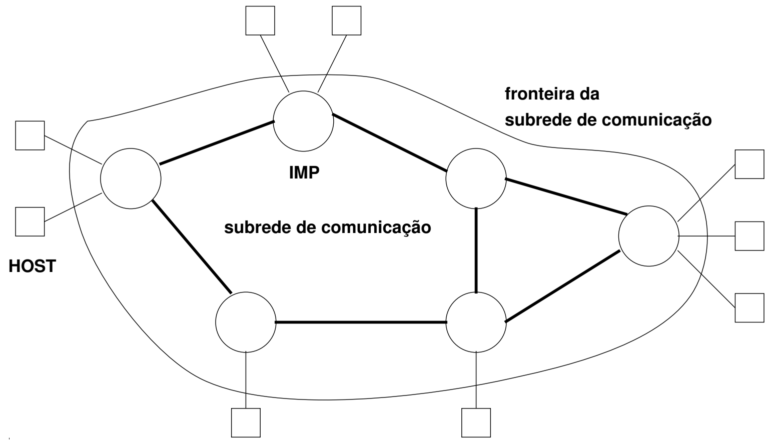

Figura 1.1: Hosts e IMPs numa subrede de comunicacão.

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*.

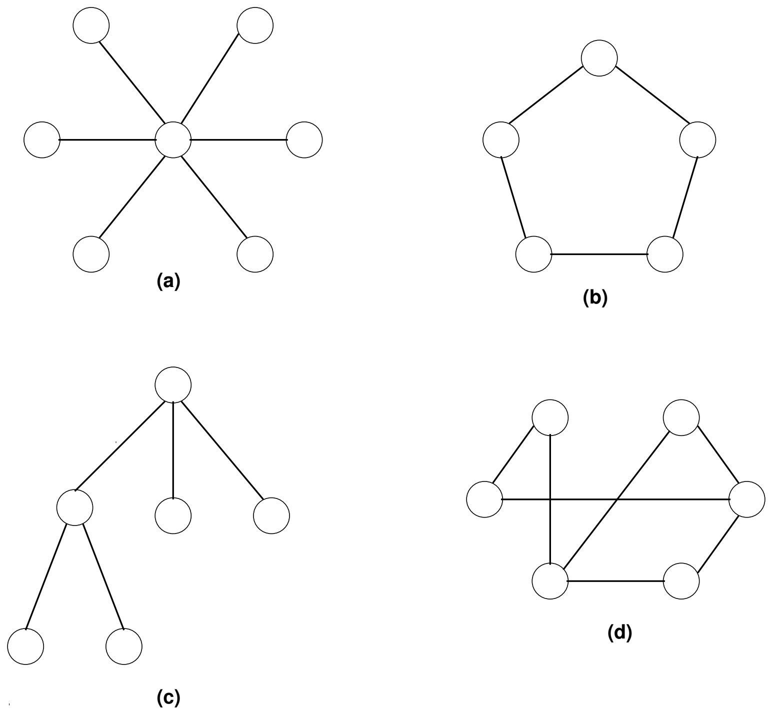

Figura 1.2: Topologias típicas em subredes ponto-a-ponto: (a) Estrela (b) Anel, (c) Árvore, (d) Genérica.

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: ITU-TS (International Telecommunications Union - Telecommunication Standardization), antiga 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.

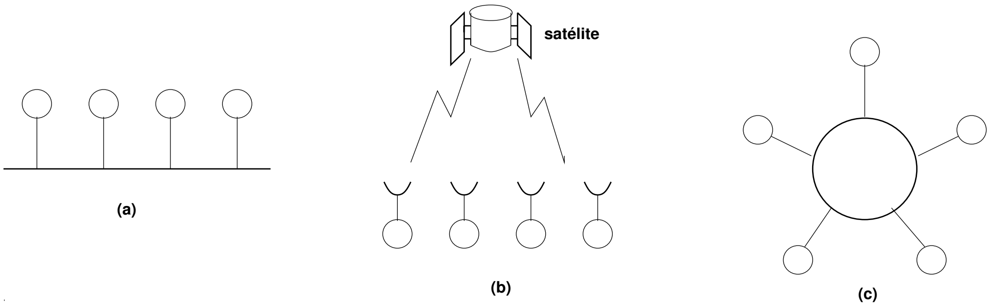

Figura 1.3: Topologias típicas em subredes de difusão: (a) Barramento (b) Radiodifusão via satélite, (c) Anel.

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

Padrões da ITU-TS 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 ITU-TS situam-se mais próximos do hardware que os padrões ISO.

Este texto cobrirá o modelo proposto pela ISO para a *Interconexão de Sistemas Abertos* (OSI[4]) e conhecido como modelo ISO/OSI 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*.

Especialistas estimam que os protocolos Internet (como o TCP/IP) continuarão a existir por um longo tempo, mesmo que oferecidos por redes OSI. Entretanto, novos serviços como aqueles oferecidos por redes digitais de serviços integrados (RDSI) serão totalmente aderentes ao modelo OSI.

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 pela 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 e se a unidade é a última da sequência ou não. 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.

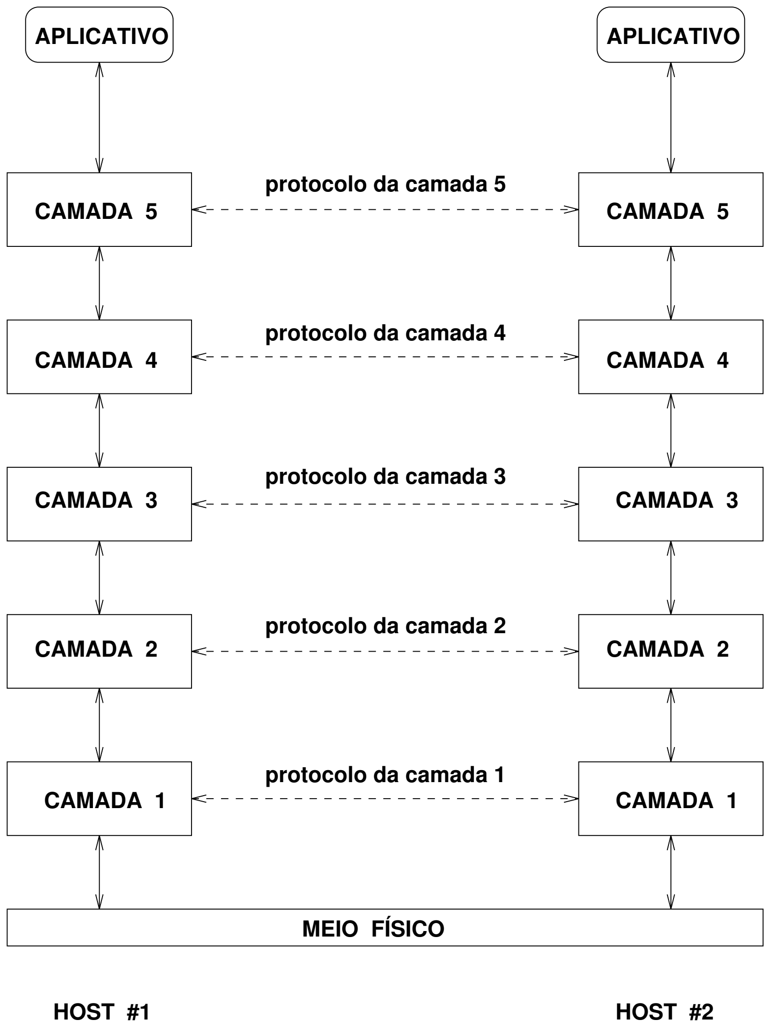

Figura 1.4: Camadas, protocolos e interfaces numa rede de computadores.

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 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.

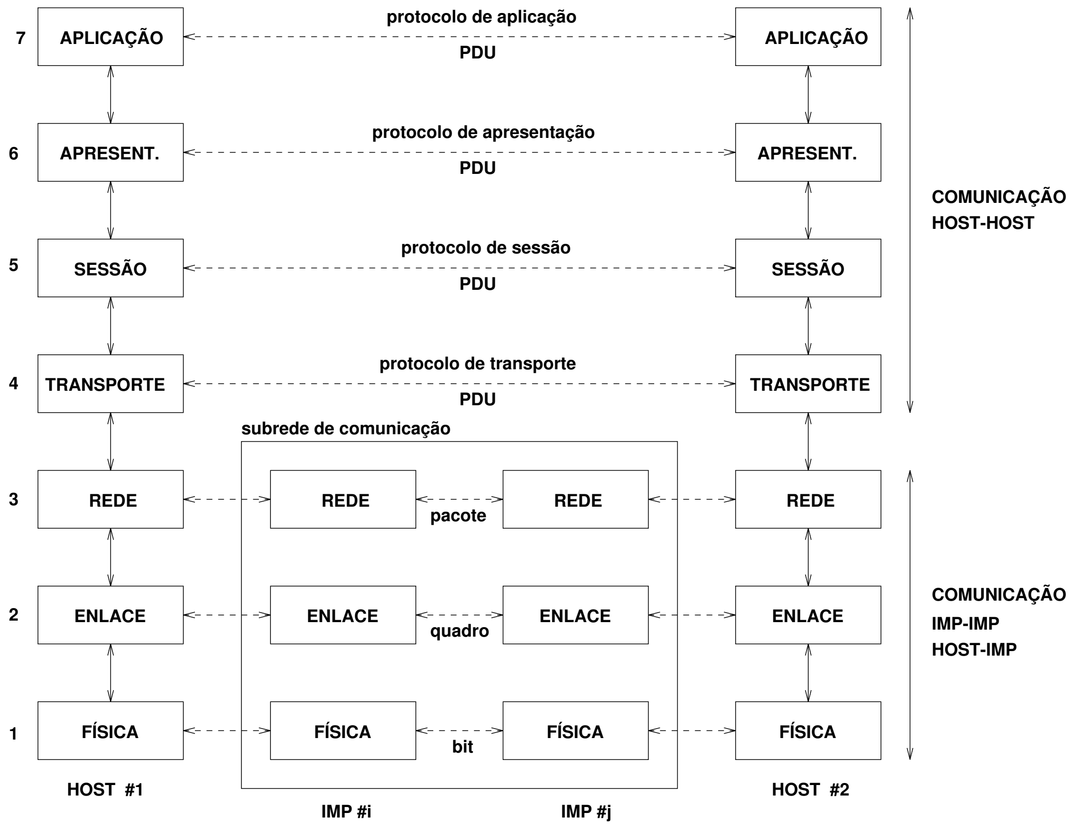

Figura 1.5: O modelo OSI.

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. 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.

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 fragmentação e remontagem de pacotes para atender a limites impostos por determinados segmentos da subrede de comunicação. Em subredes de difusão e redes locais 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: Application Processes) 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.

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

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

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 Serviços

O modelo de referência não especifica por si só o comportamento de sistemas abertos, ou seja, serviços e protocolos. Ele basicamente cria o contexto no qual uma detalhada especificação de serviços e protocolos possa ser suportada pelo sistema aberto. Do ponto de vista da abstração a figura 1.6 ilustra os vários níveis envolvidos relacionando o modelo de referência, serviços, protocolos e implementação de um protocolo.

Da figura anterior podemos observar que o modelo de referência permite a especificação de várias classes de serviço. Cada classe de serviço permite a especificação de várias classes de protocolo onde o nível mais baixo é representado pela implementação do protocolo.

O modelo de referência OSI procura fornecer uma visão global de um sistema organizado através de uma rede. Esta visão, por outro lado, divide a rede em camadas horizontais cuja finalidade consiste em:

- permitir uma discussão da interação entre elementos pares, ou seja, elementos que situam-se em uma mesma camada; esta discussão deve ocorrer sem levar em conta componentes das outras camadas;
- a funcionalidade de cada camada pode ser desenvolvida de forma modular e gradual;
- o sistema aberto pode ser visto como uma sucessão de sub-sistemas facilitando o desenvolvimento de uma implementação modular do mesmo.

Os elementos ativos em cada camada são chamados *entidades*. Uma entidade pode ser um componente de software (um processo, por exemplo) ou de hardware (um controlador

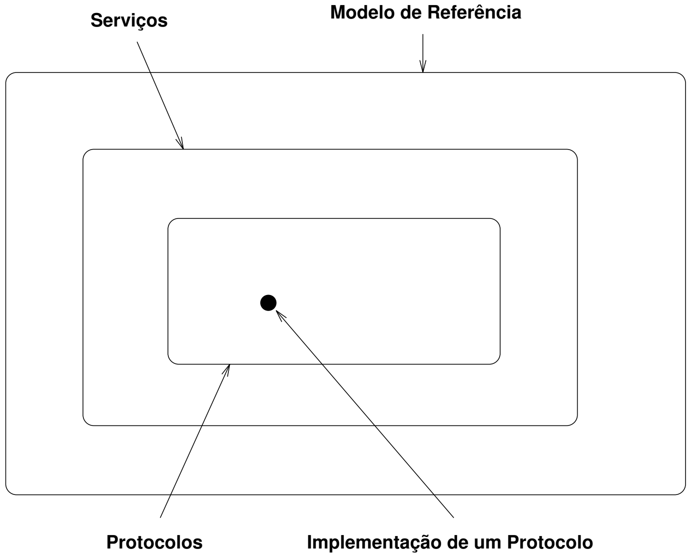

Figura 1.6: Relação entre Modelo de Referência, Especificação de Serviço e Especificação de Protocolo.

de I/O, por exemplo). As entidades na camada N implementam serviços usados na camada $N+1$. Neste contexto, a camada N é denominada *provedora de serviços* enquanto a camada $N+1$ é dita *usuária de serviços*.

Serviços são disponíveis nos SAPs (Service Access Points). A camada $N+1$ acessa os serviços oferecidos pela camada N nos SAPs desta camada. Cada SAP possui um endereço que o identifica única e globalmente.

A figura 1.7 ilustra alguns conceitos associados aos pontos de acesso de serviços. Da figura podemos observar que:

- no máximo uma entidade é responsável pelo suporte ao SAP;
- no máximo uma entidade-$(N+1)$pode acessar um SAP-(N)de cada vez;
- uma entidade-(N)pode suportar qualquer número de SAPs-(N) de cada vez;
- uma entidade-$(N+1)$ pode acessar serviços em mais de um SAP-(N).

A associação entre uma entidade-$(N+1)$ a um SAP-(N) como também a associação de uma entidade-(N) que suporta um SAP-(N) não são associações permanentes podendo ser alteradas dinamicamente.

A natureza dos serviços suportados por uma camada dependem das primitivas que uma entidade-$(N+1)$ e uma entidade-(N) podem evocar relativamente ao SAP-(N).

1.7.1 Serviços Conectados e Sem Conexão

Serviços conectados são análogos a um serviço telefônico. Estabelece-se uma conexão (discagem/atendimento), utiliza-se a conexão para troca de informações (conversação) e termina-se

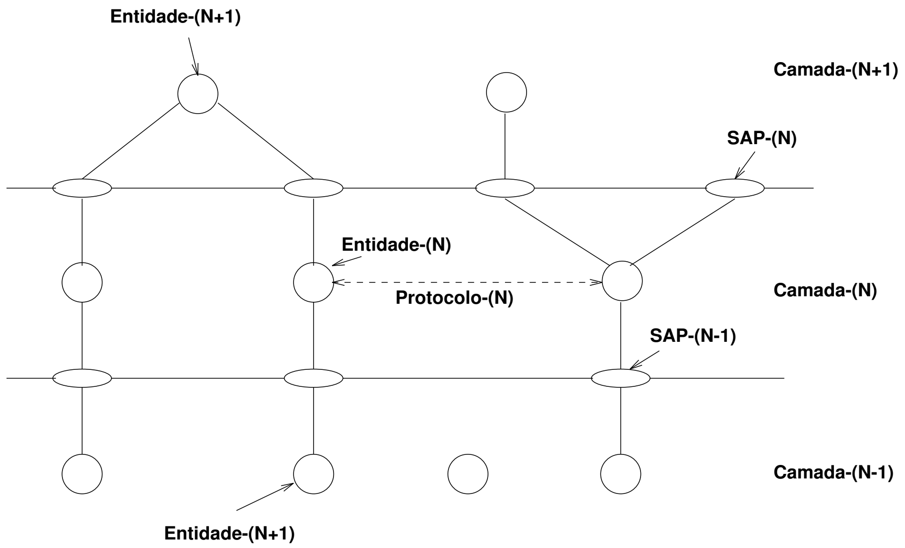

Figura 1.7: Pontos de Acesso de Serviços.

a conexão (volta ao gancho). Serviços conectados estabelecem um canal lógico entre as entidades comunicantes. Neste canal, a ordem temporal no envio da informação é respeitada e duplicações são inexistentes. O canal é dito *lógico* ou *virtual* pelo fato de não dispor de uma conexão física exclusiva, ao contrário, múltiplos canais lógicos compartilham uma mesma conexão física.

Serviços conectados se dividem em dois tipos: mensagens e cadeias de bytes. Mensagens têm suas fronteiras delimitadas, o que não ocorre com as cadeias de bytes. Serviços típicos que demandam conexão são transferência de arquivos e *login* remoto.

Serviços sem conexão são análogos a serviços postais. Envia-se uma mensagem a um destinatário (carta ou telegrama) e faz-se votos que ele a receba. Serviços sem conexão são denominados *serviços de datagrama*. São mais rápidos que os serviços conectados, mas a garantia de entrega, ausência de duplicação e preservação da ordem de envio não são garantidas. Serviços típicos que podem ser baseados em datagrama são os do tipo requisição/resposta (acesso a banco de dados e sincronização de relógios, por exemplo).

Dentro do jargão ISO uma conexão é estabelecida por iniciativa de uma entidade-$(N+1)$ que evoca serviço específico da camada-(N) para estabelecimento de uma conexão lógica com uma entidade-$(N+1)$ remota. A conexão é estabelecida entre SAPs-(N) correspondentes.

Todas as conexões estabelecidas em um mesmo SAP são suportadas pela entidade-(N) correspondente. De modo a permitir que a entidade-$(N+1)$ e a entidade-(N) possam distinguir várias conexões estabelecidas via um mesmo SAP, define-se o conceito de ponto final de conexão (CEP - *connection end point*). Para cada conexão, 2 CEPs são definidos, um para cada extremo da conexão. Desta maneira, um ponto final da conexão-(N) (CEP-(N)) termina uma conexão-(N) em um SAP-(N). Um CEP-(N) relaciona três elementos, ou seja, uma entidade-$(N+1)$, uma entidade-(N) e uma conexão-(N).

No caso do envio sem conexão são utilizados serviços de transferência de dados sem conexão suportados pela camada-(N).

1.7.2 Identificadores

Com o objetivo de permitir que objetos como entidades, SAPs e conexões sejam referenciados no mundo OSI é necessário um esquema de identificação global destes objetos. Os identificadores de entidades são denominados de títulos (*titles*), os de pontos de acesso de serviços (SAPs) são denominados de endereços e as conexões são identificadas pelos identificadores do ponto final de identificação *CEP-identifier*, ou seja:

- uma entidade-(N) possui um título-(N);
- um SAP-(N) possui um endereço-(N);
- um CEP-(N) possui um identificador-CEP-(N).

O sufixo (N) usado acima indica que os identificadores possuem significado no nível da camada-(N).

A associação do título de uma entidade-($N+1$) com o endereço do SAP-(N) ao qual ela está conectada encontra-se em um diretório mantido pela camada-($N+1$), referenciado como diretório-($N+1$). Desta maneira, caso uma entidade-($N+1$) necessite estabelecer uma conexão com uma entidade-($N+1$), ela pode recorrer ao diretório-($N+1$) para determinar o endereço do SAP-(N) ao qual a entidade destino encontra-se conectada.

1.7.3 Unidades de Dados

A troca de dados entre entidades no modelo OSI ocorre de duas formas:

1. entre entidades pares, ou seja, entidades-($N+1$) remotas sendo a troca governada por um protocolo-(N);
2. entre entidades em camadas adjacentes, isto é, entidade-($N+1$) e entidade-(N)através de um mesmo SAP-(N).

As informações trocadas entre as entidades, sejam entidades pares ou adjacentes, podem ser divididas em duas partes: *dados* e *informações de controle*. Desta forma, é possível definir-se quatro tipos de unidades de dado:

1. *informação de controle de protocolo*: informação trocada entre entidades pares com a finalidade de coordenar as suas operações conjuntas;
2. *dado do usuário*: dado transferido entre uma entidade-($N+1$) e uma entidade-($N+1$) par através de serviços suportados por entidades-(N);
3. *informação de controle de interface*: trata-se de informação trocada entre entidade-($N+1$) e entidade-(N) para coordenar as suas interações através do SAP-(N);

4. *dado de interface*: dado transferido da entidade-$(N+1)$ à entidade-(N) para ser enviado à entidade-$(N+1)$ par.

Como, em geral, as informações trocadas entre as entidades pares ou adjacentes incluem informações de dado e controle, determinam-se 2 novas unidades:

1. *unidade de dado do protocolo-(N) (PDU-(N))*: informação trocada entre entidades pares e constituida de controle e dado do usuário;

2. *unidade de dado de interface-$(N+1)$*: informação trocada entre uma entidade-$(N+1)$ e uma entidade-(N) através de um SAP-(N), sendo constituida de controle e dado do usuário.

O conjunto de PDUs trocadas entre entidades-$(N+1)$ pares é especificado pelo protocolo-$(N+1)$ que rege a interação entre estas entidades. Por outro lado, a descrição do serviço não especifica um conjunto de unidades de dados de interface, isto porque a troca deste tipo de unidade de dado se dá entre entidades adjacentes internas a um sistema aberto, ou seja, fora do escopo de padronização OSI. A definição deste tipo de informação é basicamente para permitir a introdução da unidade de dado de serviço-(N).

Uma unidade de dado de serviço (SDU-(N)) é um dado da interface cuja identidade é preservada de um extremo a outro da conexão. Não é relevante como uma SDU-(N) é trocada entre as entidade-$(N+1)$ e a entidade-(N) desde que os limites entre SDUs sejam preservados.

Em determinadas situações uma entidade-$(N+1)$necessita enviar dados urgentes. Neste caso é associada uma unidade de dado de serviço urgente (SDU-urgente-(N)) ao serviço permitindo uma transferência do dado em condições prioritárias.

De forma resumida temos que uma camada fornece um serviço que permite que entidades-$(N+1)$ pares comuniquem-se através da troca de dados, ou seja, PDUs-$(N+1)$. A entidade-$(N+1)$ envia a PDU-$(N+1)$para a entidade-(N) através do SAP-(N)na forma de uma SDU-(N). Esta SDU é entregue remotamente à entidade-$(N+1)$ no SAP correspondente. A entidade-(N) no sistema onde a informação é originada trata a SDU-(N) como dado do usuário e cria uma PDU-(N) anexando ao dado informação de controle do protocolo-(N). Este mapeamento de uma PDU-$(N+1)$ em uma SDU-(N) e da SDU-(N) em uma PDU-(N) é mostrada na figura 1.8

O Modelo de Referência OSI não coloca restrições com relação ao tamanho da unidade de dado. De modo a suportar o comprimento variável das unidades, o Modelo permite que uma unidade de dado seja mapeada em um número de unidades de dado, ou que um número de unidades de dado seja mapeado em uma única unidade de dado. Consequentemente, as formas de mapeamento a seguir são possíveis:

- *segmentação*: é a função realizada por uma entidade-(N)através da qual uma SDU-(N) é mapeada em várias PDUs-(N)(figura 1.9);

- *remontagem*: função inversa da segmentação, ou seja, a entidade-(N) par mapeia múltiplas PDUs-(N) em uma SDU-(N)(figura 1.9);

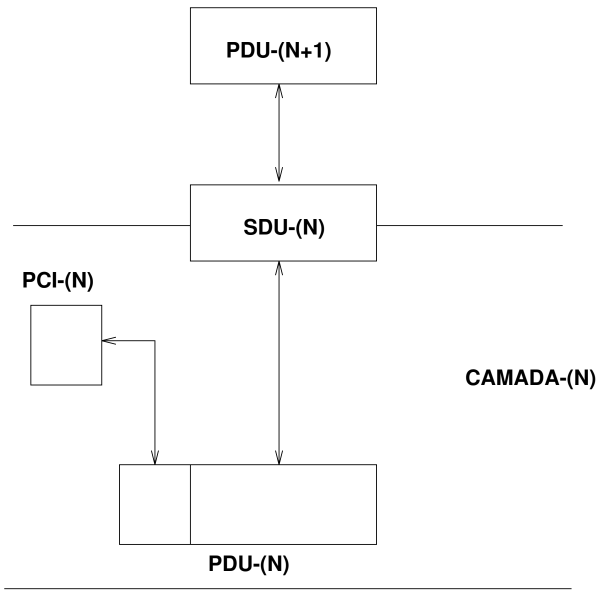

Figura 1.8: Mapeamento de PDUs em SDUs.

- *bloqueio*: função realizada por uma entidade-(N) que mapeia múltiplas SDUs-(N)em uma PDU-(N)(figura 1.10);
- *desbloqueio*: função reversa do bloqueio, ou seja, a entidade-(N) par mapeia uma PDU-(N) em múltiplas SDUs-(N) correspondentes (figura 1.10);
- *concatenação*: é a forma que permite a uma entidade-(N) mapear múltiplas PDUs-(N) em uma SDU-($N-1$) (figura 1.11);
- *separação*: função reversa da concatenação e realizada pela entidade-(N) que mapeia uma SDU-($N-1$) em múltiplas PDUs-(N) correspondentes (figura 1.11).

1.7.4 Primitivas para Requisição de Serviços

A definição dos serviços suportados por uma determinada camada-(N) é caracterizada pelo conjunto de primitivas e os respectivos parâmetros que podem ser evocadas pelo usuário do serviço (entidade-($N+1$)) e pelo provedor do serviço (entidade-(N)) através de um SAP-(N). A interação entre a entidade-($N+1$) e a entidade-(N) através das primitivas de serviço definidas involve a passagem de informações de controle ou de dados, ou ambos.

No modelo OSI, as primitivas para a requisição de serviços são divididas em quatro tipos (figura 1.12).

Vamos tomar o exemplo do estabelecimento de uma conexão entre duas entidades. A entidade que deseja se conectar executa a primitiva *CONNECT.request* (na notação OSI), passando como parâmetro o endereço da segunda entidade. A entidade endereçada recebe um

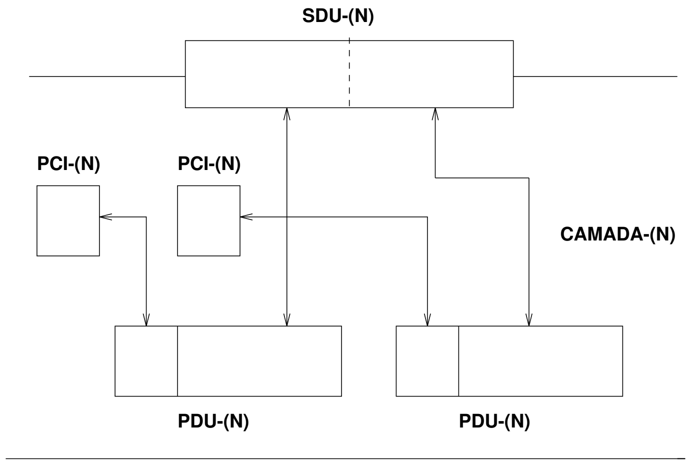

Figura 1.9: Segmentação e Remontagem.

CONNECT.indication avisando que uma conexão está sendo requisitada. Como resposta, executa um *CONNECT.response* indicando se aceita ou rejeita a conexão. A entidade que requisitou está bloqueada num *CONNECT.confirm*, que volta exatamente o resultado do *CONNECT.response* emitido pela outra entidade.

Serviços envolvendo duas partes podem conter *negociação*. Por exemplo, uma conexão pode especificar certa taxa mínima de transferência de dados, máximo tamanho das mensagens, etc. Neste caso, a negociação deve fazer parte do protocolo que regulamenta o serviço.

Serviços podem ser *confirmados* ou *sem confirmação*. Serviços confirmados necessitam das quatro primitivas básicas (requisição, indicação, resposta e confirmação). Serviços sem confirmação necessitam apenas das primitivas do tipo requisição e indicação. Exemplo: serviço de desconexão. A figura 1.13 ilustra estes conceitos.

1.8 Protocolos

Um dos problemas fundamentais abordado na especificação de um sistema de comunicação aberto diz respeito ao protocolo envolvendo entidades pares. O conceito clássico de protocolo define a forma como entidades equivalentes interagem entre si para a realização de um objetivo comum para a prestação de serviços a entidades na camada superior. Desta forma, 2 entidades-$(N+1)$ que desejam se comunicar devem utilizar os serviços de comunicação suportadas pela camada-(N), com exceção da camada física para acesso ao meio físico diretamente.

Para suportarem os serviços de comunicação requisitados no SAP-(N) pelas entidades-$(N+1)$, as entidades-(N) devem se comunicar através de serviços da camada-$(N-1)$. Neste caso, a comunicação entre entidades pares é regida por um conjunto de regras e formatos que as entidades devem seguir para suportar os serviços. Este conjunto de regras e formatos representando por aspectos semânticos, sintáticos e temporais é denominado de protocolo da camada-(N).

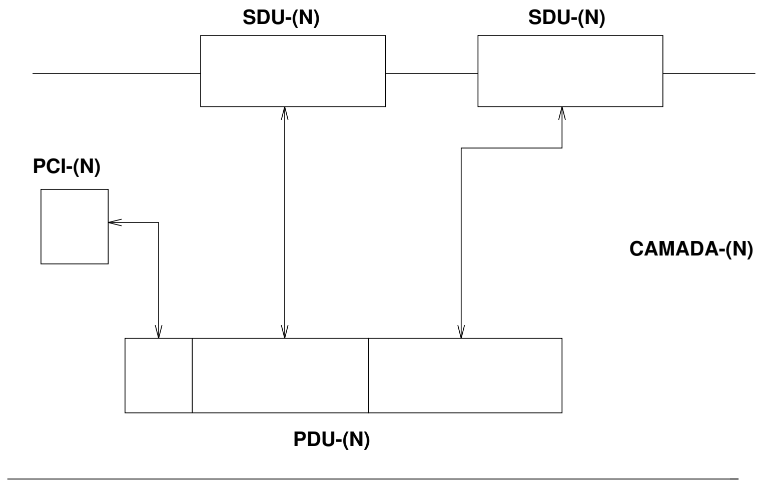

Figura 1.10: Bloqueio e Desbloqueio.

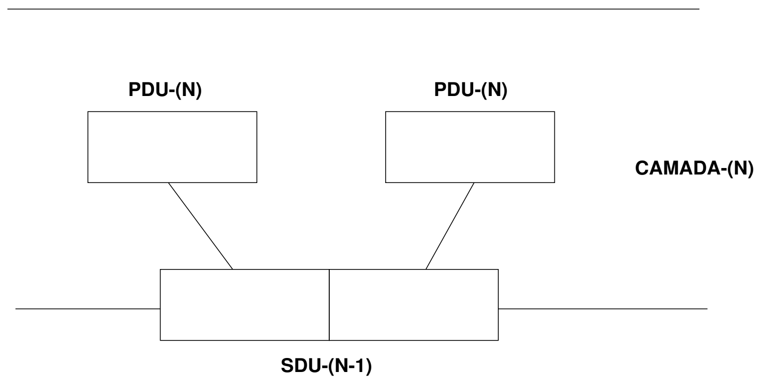

Figura 1.11: Concatenação e Separação.

1.8.1 Especificação de Protocolo

A especificação de um protocolo relativamente a uma camada compreende:

- descrição dos tipos de PDUs;
- descrição dos procedimentos do protocolo e os serviços evocados para transferência de cada tipo de PDU;
- definição formal da estrutura de cada tipo de PDU;
- definição formal da operação da entidade de protocolo.

Primitiva	**Significado**
Requisição	Uma entidade requer a execução de um serviço
Indicação	Uma entidade é informada da ocorrência de um evento
Resposta	Uma entidade deseja responder a um evento
Confirmação	Uma entidade é informada sobre o resultado de sua requisição

Figura 1.12: Tipos de serviços no modelo OSI.

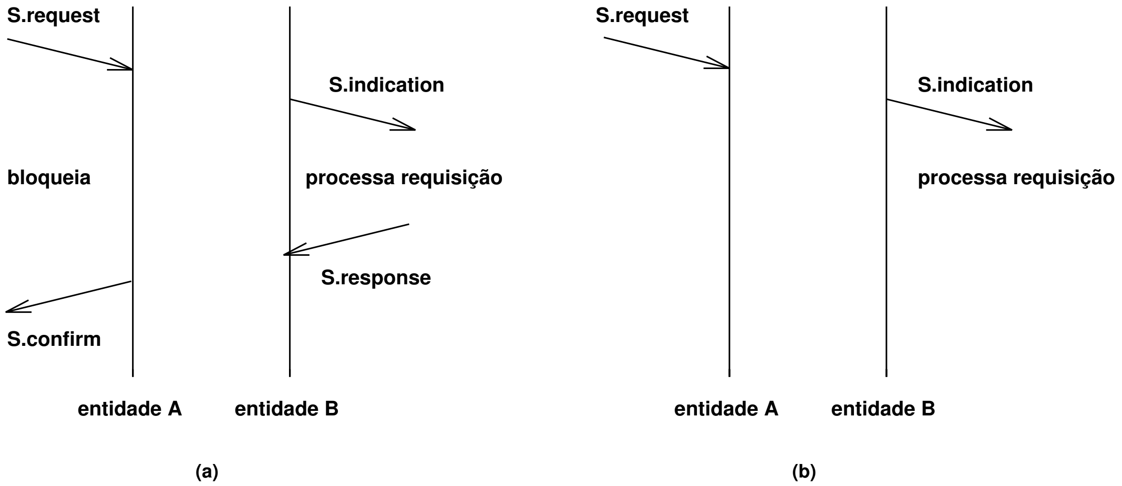

Figura 1.13: Serviços OSI confirmados (a) e sem confirmação (b).

Definição de PDU

Duas entidades pares de um protocolo comunicam-se através da troca de PDUs. Estas são formadas por uma parte correspondente aos dados do usuário e outra parte contendo as informaçẽs de controle relativa ao protocolo (PCI). Como a PDU é uma informação que será trocada entre sistemas diferentes, é necessário que a PDU possua uma estrutura cujo significado seja comum às entidades de ambos os sistemas. Isto é obtido nos documentos de padronização através da utilização de cadeias de bits ou através de uma forma baseada em tipos abstratos de dados (*Abstract Syntax Notation Number One ou ASN.1*) acompanhada de regras de codificação.

A representação baseada em cadeia de bits é mais utilizada na especificação dos protocolos das camadas inferiores. A representação utilizando o ASN.1 é mais comum na especificação de PDUs dos protocolos de nível mais alto, isto é, dos protocolos orientados à aplicação. ASN.1 é baseada na tipificação dos dados como encontrado na maioria das linguagens de programação. Como ASN.1 é uma sintaxe abstrata, isto significa que nada é especificado com relação ao número e ordem dos bits correspondentes a cada tipo de dado encontrado na especificação. Neste caso, utiliza-se um conjunto de regras de codificação que irá gerar PDUs formadas de cadeias de bytes, cadeias estas que serão interpretadas da mesma forma em todos os sistemas.

Operação do Protocolo

Uma entidade de protocolo é modelada na forma de uma máquina de estados finitos (autômato). Isto significa que uma entidade de protocolo deve encontrar-se em um único estado de cada vez dentre um conjunto finito de estados possíveis.

A transição de um estado para outro acontece quando um evento válido ocorre na interface do autômato. Alguns exemplos de eventos são os seguintes:

- recepção de uma primitiva de serviço na interface com a camada superior;
- recepção de uma primitiva na interface com a camada inferior;
- ocorrência de eventos locais.

Em geral, associado à ocorrência de um evento válido, o autômato muda de estado e gera alguma ação interna específica como, por exemplo, o disparo de um relógio ou a evocação de uma primitiva.

O método de especificação de protocolo utiliza uma tabela evento-estado onde cada entrada na tabela especifica o evento saída e o novo estado para o qual o autômato deverá transitar devido a uma combinação *evento-de-entrada—estado-atual* (figura 1.14).

1.9 Problemas

1. Qual o principal benefício proporcionado por uma rede de computadores ?
2. Terminais bancários podem ser considerados hosts de uma rede de computadores ? Justifique.
3. O que são IMPs ? Por que diferenciá-los dos hosts ?
4. Quem (usuário, fabricante, vendedor, etc.) lucra mais com a padronização de redes de computadores ?
5. Quais as vantagens de se estruturar uma rede em camadas ?
6. Qual a diferença entre modelo, protocolo e arquitetura ?
7. Descreva suscintamente as 7 camadas do modelo OSI.
8. Suponha que você decida reimplementar a camada de rede utilizando um algoritmo inteligente de roteamento. Que cuidados você deve tomar para que as camadas de enlace e de transporte permaneçam inalteradas ?
9. Considere o programa *talk* do Unix que abre uma sessão de conversação *on-line* entre dois usuários (o que é teclado num terminal é mostrado no outro e vice-versa). Você o implementaria como um serviço conectado ou sem conexão ? Justifique.

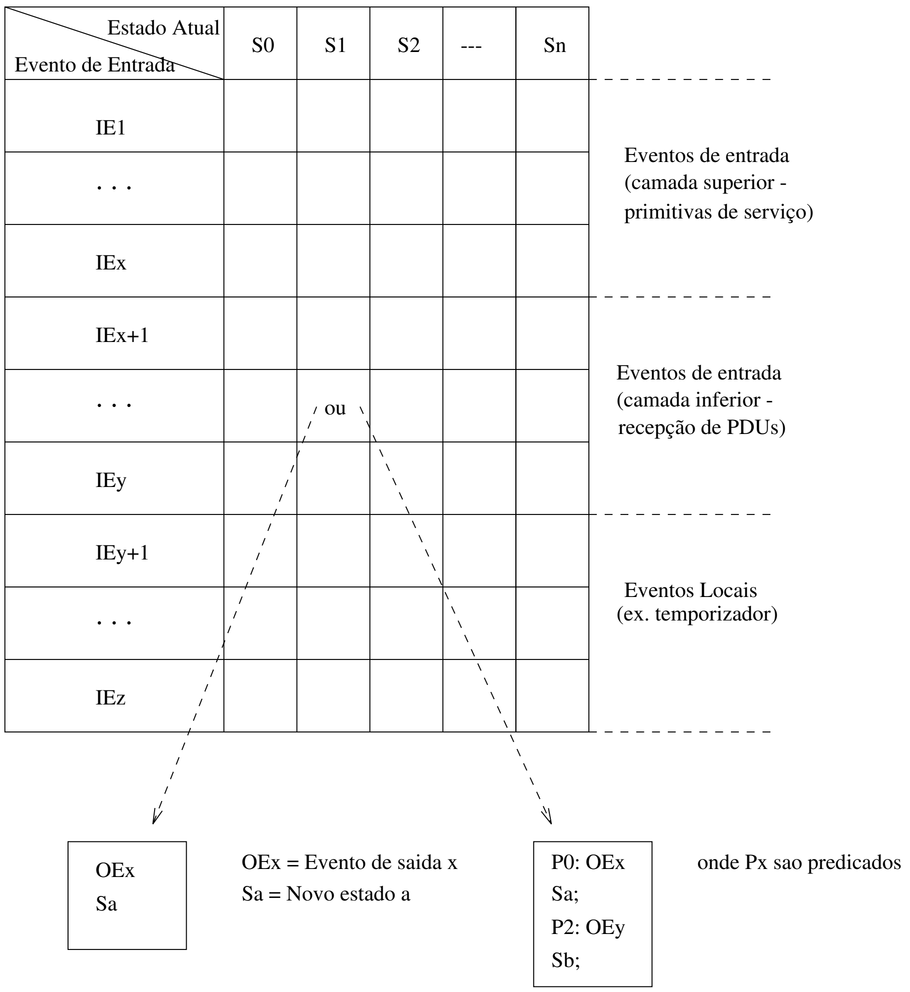

Figura 1.14: Tabela evento-estado utilizada na especificação de protocolos.

10. Considere o programa *mail* do Unix que envia uma mensagem para um usuário em determinado host (correio eletrônico). Você o implementaria como um serviço confirmado ou sem confirmação ? Justifique.

Capítulo 2

A CAMADA FÍSICA

A camada física trata da geração de sinais e sua propagação através do meio físico. A camada deve especificar:

- a natureza do meio físico: sua constituição (cabo coaxial, fibra ótica, etc.), suas dimensões (diâmetro, comprimento, etc.), suas constantes físicas (impedância característica, atenuação, etc.), dentre outros parâmetros;
- a forma como os hosts e IMPs são interconectados no meio físico (conectores, pinagem, etc.);
- a forma como 0s e 1s são codificados em sinais compatíveis com o meio físico;
- os parâmetros e suas respectivas tolerâncias dos sinais: tensões, correntes, freqüências, etc.;
- o procedimento de multiplexação de sinais no meio físico, se houver.

Vamos examinar os principais tópicos relacionados com a camada física, procurando limitar o nível de detalhamento ao escôpo deste texto. Padrões publicados por entidades como IEEE, ANSI e ISO devem ser consultados para um estudo mais aprofundado dos componentes da camada física para uma determinada tecnologia de rede.

2.1 Modos de Transmissão

A camada física é responsável pela transmissão através do meio físico dos quadros oriundos da camada de enlace. De acordo com o hardware empregado, a transmissão pode ocorrer em modo *full duplex* onde dois hosts comunicantes transmitem simultaneamente, ou *half duplex* caso a informação tenha seu fluxo alternado no tempo (apenas um host por vez pode transmitir). Em redes locais de computadores, a comunicação *half duplex* é a mais comum, dado que, na maioria das vezes, o meio físico é composto de duto único.

A transmissão de bits pelo meio físico pode se processar de forma *serial* ou *paralela*. Na transmissão serial um bit é transmitido a cada intervalo de tempo, enquanto na paralela os bits são transmitidos serialmente por N dutos independentes, resultando na transmissão de

N bits por unidade de tempo. Como a transmissão paralela requer um meio físico composto de vários dutos, esta não é empregada em redes locais de computadores.

Uma transmissão de bits pode se dar de forma *síncrona* ou *assíncrona*. Na transmissão síncrona um bit é transmitido a cada intervalo de tempo bem definido, enquando na transmissão assíncrona um bit pode ser transmitido num instante de tempo arbitrário. A transmissão síncrona requer que tanto o host transmissor quanto o host receptor disponham de uma base de tempo comum, o que torna seu emprego muito restrito em redes locais de computadores. A transmissão assíncrona a nível de bits, por outro lado, é um processo ineficiente pois requer a introdução de um separador entre dois bits (caracter especial ou intervalo de tempo sem transmissão).

Na prática, utiliza-se um meio termo entre os modos síncrono e assíncrono, onde a transmissão assíncrona se dá a nível de bloco de bits (transmissão start-stop). Detectado a ocorrência de um bloco, os bits que o compõe são recebidos sincronamente. Cada bloco é iniciado com um delimitador de início (combinação de bits) e terminado com um delimitador de final de bloco. Ao detectar um delimitador de início, o receptor sincroniza seu relógio para receber, sincronamente, os bits que compõem o bloco. Mesmo que o host transmissor e receptor apresentem um certo *drift* em seus relógios, a transmissão start-stop requer sincronismo apenas entre os delimitadores de início e final de bloco.

2.2 Codificação de Sinais Digitais

A representação de bits em circuitos digitais se dá, via de regra, por sinais do tipo ON/OFF onde um valor alto de tensão representa o código 1 enquanto um valor baixo representa o código 0 (ou vice-versa). Esta forma de codificação produz ondas quadradas de formato altamente irregular o que dificulta sua transmissão através do meio físico (justificativa adiante). Outras formas de se codificar um sinal digital ON/OFF para sua posterior transmissão são (figura 2.1):

- Código Polar NRZ (Non Return to Zero): idêntico ao sinal ON/OFF, exceto que duas polaridades são empregadas (positiva e negativa). Raramente empregado em redes locais de computadores;
- Código Unipolar RZ (Return to Zero): o bit 1 coincide com um pulso de relógio, enquanto o bit 0 é representado pela ausência do pulso. Raramente empregado em redes locais de computadores;
- Código Manchester: uma transição sempre ocorre no meio do intervalo de um bit. Se positiva (subida), a transição representa o bit 1, se negativa (descida) o bit 0. Transições são do tipo RZ;
- Código Bifase: idêntico ao Manchester, mas empregando transições NRZ;
- Código Manchester Diferencial: uma transição sempre ocorre no meio do intervalo de um bit. Uma transição adicional no início do intervalo representa o bit 0, enquanto a ausência desta representa o bit 1. Transições são do tipo RZ;

- Código Bifase Diferencial: idêntico ao Manchester Diferencial, mas empregando transições NRZ;

- Código CMI (Code Mark Inversion): o bit 0 é representado por uma transição positiva (subida) no meio do intervalo do bit, e o bit 1 pela ausência de transição positiva. Uma transição no final do intervalo ocorre quando bits 1s se repetem. Transições são do tipo NRZ.

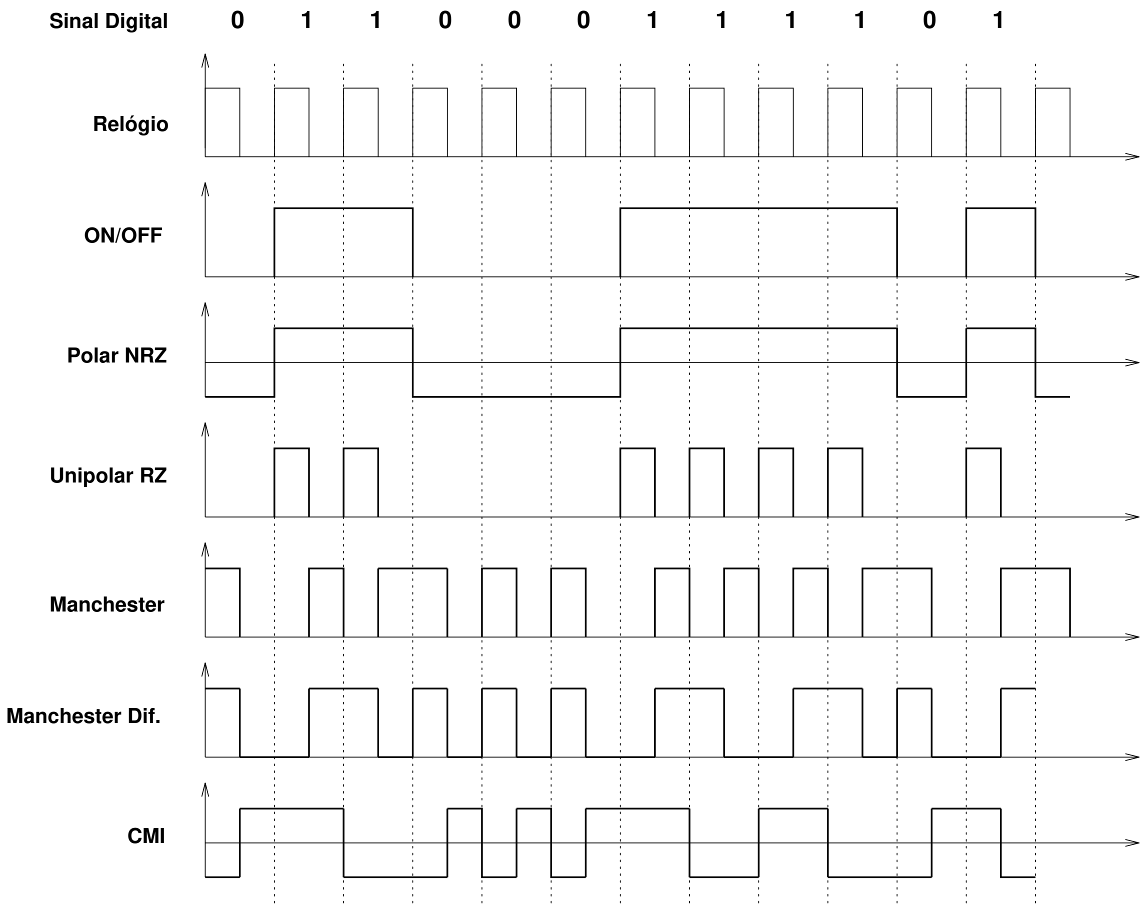

Figura 2.1: Formas de se codificar um sinal digital.

Um sinal codificado pode ser transmitido de forma digital ou analógica. Na forma digital, pulsos elétricos (ou óticos) são gerados no meio físico. O padrão IEEE 802.3 (CSMA/CD) utiliza transmissão digital com codificação Manchester injetando no cabo pulsos de tensão com amplitude variando entre 0 e -2.05 Volts.

Na forma analógica, sinais senoidais são gerados no meio físico. A geração de sinais senoidais a partir de um sinal digital é denominada *modulação*. O padrão IEEE 802.4 (Token Bus) emprega modulação na transmissão de sinais digitais pelo meio físico.

2.3 Modulação de Sinais Digitais

O processo de modulação consiste em variar a amplitude, a freqüência ou a fase de um sinal senoidal para representar determinado nível de um sinal digital.

A *Modulação por Chaveamento de Amplitude* (ASK) emprega uma onda senoidal de freqüência fixa e dois níveis de amplitude que representam os níveis do sinal digital (figura 2.2).

A *Modulação por Chaveamento de Freqüência* (FSK) emprega duas freqüências para codificar os níveis do sinal digital (figura 2.2). A passagem da freqüência alta para a baixa pode se dar de forma contínua ou de forma discreta. Na passagem contínua (modulação FSK-contínua), existe um tempo máximo para ocorrer a transição entre as duas freqüências e uma variação máxima nas amplitudes correspondentes às freqüências alta e baixa. Na passagem discreta (FSK-coerente), normalmente se empregam N ciclos de freqüência baixa para representar determinado nível binário, e $2N$ ciclos de freqüência alta (o dobro da baixa) para o outro nível.

O padrão IEEE 802.4 (Token Bus) prevê ambas as formas de modulação FSK (não coexistentes na mesma rede, obviamente). Na moduação FSK-contínua, as freqüências são 3.75 e 6.25 MHz com uma tolerância de $\pm$ 80 KHz. O período de transição não pode ultrapassar 100 nanosegundos e as amplitudes nas freqüências alta e baixa não devem ter uma variação superior a 10%. Na modulação FSK-coerente, são empregadas freqüências de 5 e 10 (ou 10 e 20) MHz e um ciclo de freqüência baixa para representar o bit 1 ($N = 1$).

Finalmente, a *Modulação por Chaveamento de Fase* (PSK) representa os níveis binários por um avanço ou atraso de fase no sinal senoidal. Por exemplo, uma transição negativa (de 1 para 0) é representada por um avanço de 90^o na fase do sinal senoidal, enquanto uma transição positiva (de 0 para 1) é representada por um atraso equivalente (figura 2.2).

2.4 Multiplexação

Multiplexação é o processo pelo qual o meio físico é compartilhado entre dois ou mais pares transmissor-receptor. A multiplexação pode se dar pela divisão em freqüência (FDM: Frequency Division Multiplex) ou pela divisão no tempo (TDM: Time Division Multiplex).

Na multiplexação FDM, diferentes bandas de freqüências são alocadas a diferentes pares transmissor-receptor. Esta técnica é bastante empregada em troncos telefônicos, sendo rara em redes de computadores. Em redes, a multiplexação TDM é a mais empregada. Esta técnica é análoga ao *time-sharing* onde a CPU de um processador é alocada a cada usuário de forma alternada.

A multiplexação TDM pode ser síncrona ou assíncrona. Na TDM síncrona, o meio físico é alocado alternadamente e por um período fixo a cada host, independente do host querer utilizá-lo ou não. Caso tenha quadros para transmitir, o host o faz no período em que o meio fica a ele alocado. Esta técnica provoca uma sub-utilização do meio físico, mas garante uma taxa mínima de transmissão para cada host. Esta garantia é fundamental em aplicações de tempo real.

Na técnica TDM assíncrona, quando um host necessitar transmitir quadros o mesmo deve executar um procedimento de acesso ao meio físico. Procedimentos convencionais são:

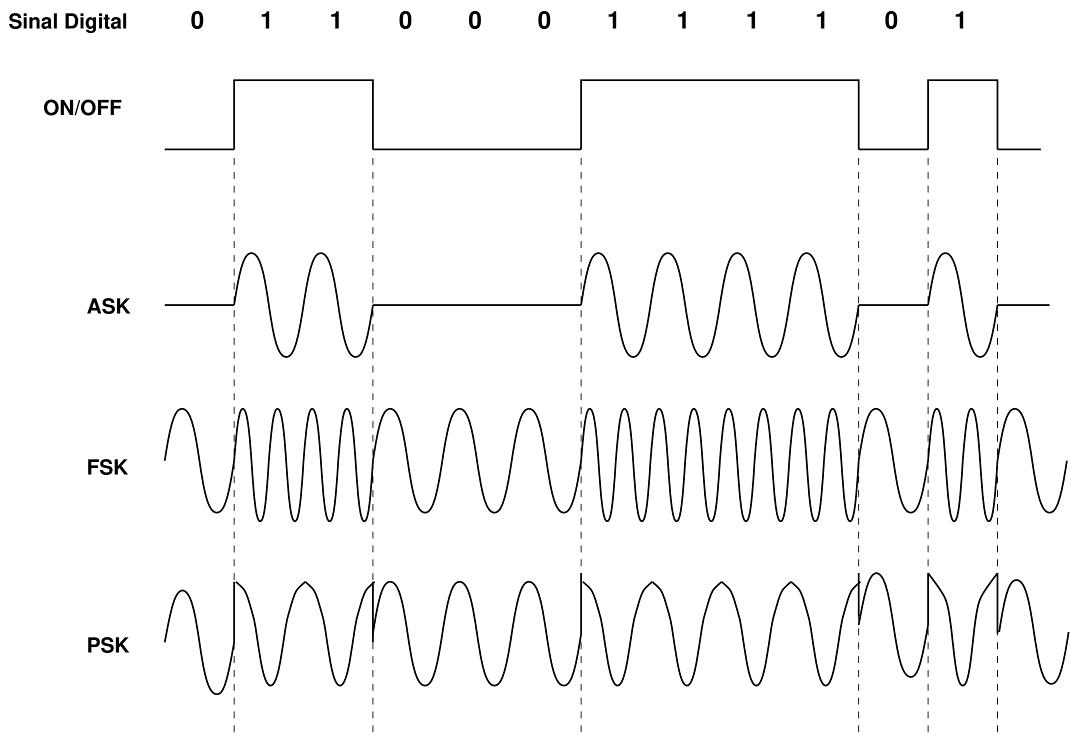

Figura 2.2: Modulação de sinais digitais.

passagem de ficha e sensoriamento do meio (estudados no próximo capítulo). Na técnica TDM assíncrona, o meio físico é utilizado mais eficienetmente, mas, em geral, não existe garantia de taxa mínima de transmissão para os hosts.

2.5 Características dos Meios de Transmissão Elétricos

Via de regra, os meios de transmissão elétricos são do tipo passa-baixas, isto é, permitem a propagação de freqüências baixas até determinado limiar (largura de banda B). Em geral, a largura de banda é definida como a freqüência em que o sinal injetado num extremo tem sua potência diminuida em 50% (-3 dB) na outra extremidade (figura 2.3-a).

Entretanto, se considerarmos os elementos elétricos diretamente conectados ao meio de transmissão, sua característica passa-baixas se altera para uma característica passa-faixa. Considere, por exemplo, que o circuto de transmissão e recepção esteja ligado ao meio através de um transformador de desacoplamento. Como transformadores exercem forte atenuação em baixas freqüências, o conjunto transformador-meio de transmissão passa a operar como um filtro passa-faixa (figura 2.3-b).

A característica passa-faixa de um meio de transmissão é a razão pela qual utiliza-se codificação de sinais diferentes de ON/OFF. Seja um sinal aleatório em codificação ON/OFF com amplitude unitária no caso de 1 e com igual probabilidade de ocorrência de 0s e 1s. Utilizando-se um filtro ideal que permite a passagem apenas se sinais na freqüência f, vamos

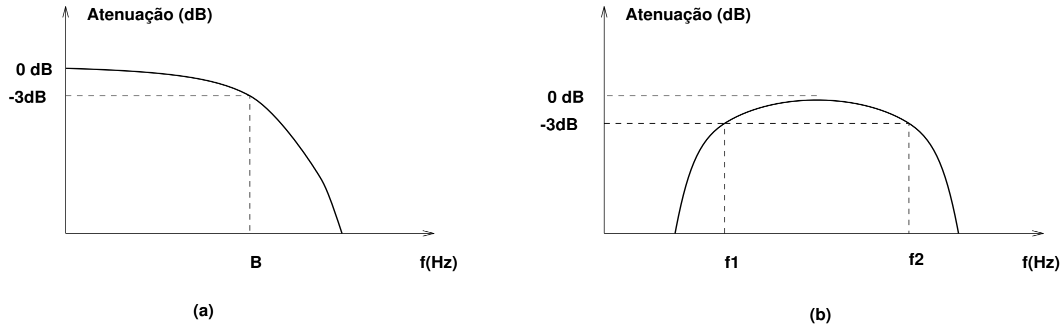

Figura 2.3: Um filtro passa-baixas (a) e passa-faixa (b).

aplicar o sinal ON/OFF ao filtro e computar a potência do sinal filtrado. Variando-se f de zero (nível DC) a infinito, obtem-se o gráfico (espectro de potência) da figura 2.4-a.

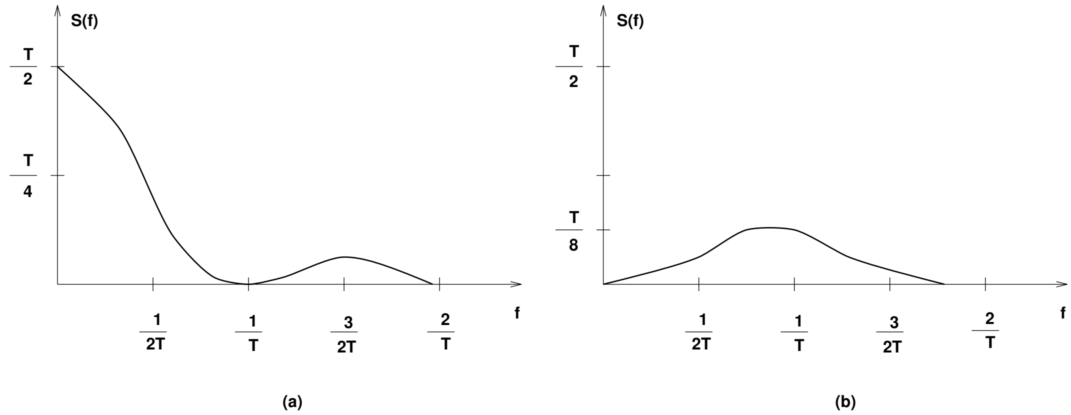

Figura 2.4: Espectro de potência de um sinal de amplitude unitária e período de ocorrência de bits igual a T: (a) codificação ON-OFF; (b) codificação Manchester.

Percebe-se agora que um meio de transmissão com características passa-faixa é inadequado para transmitir um sinal binário em codificação ON/OFF, pois este sinal demanda a propagação de potências consideráveis em baixas freqüências (de 0 até o inverso do período do bit-T). Impedindo-se a propagação de tais freqüências, tem-se um sinal altamente distorcido no receptor.

Passando agora o sinal ON/OFF por um codificador Manchester antes de aplicá-lo ao filtro, obtem-se o espectro de potência da figura 2.4-b. O código Manchester demanda a propagação de potências moderadas tanto de baixas quanto de altas freqüências, causando pouca distorção do sinal quando transmitido através de um meio passa-faixa.

2.6 Fontes de Distorção do Sinal

Característica Passa-faixa do Meio

A figura 2.5 mostra a distorção de um sinal periódico pela ausência (ou atenuação) de suas componentes de alta freqüência. A ausência de componentes de baixa freqüência dificulta ainda mais a identificação de 0s e 1s pelo receptor. Sinais periódicos necessitam apenas da propagação de submultiplos inteiros de sua freqüência (harmônicas da série de Fourier).

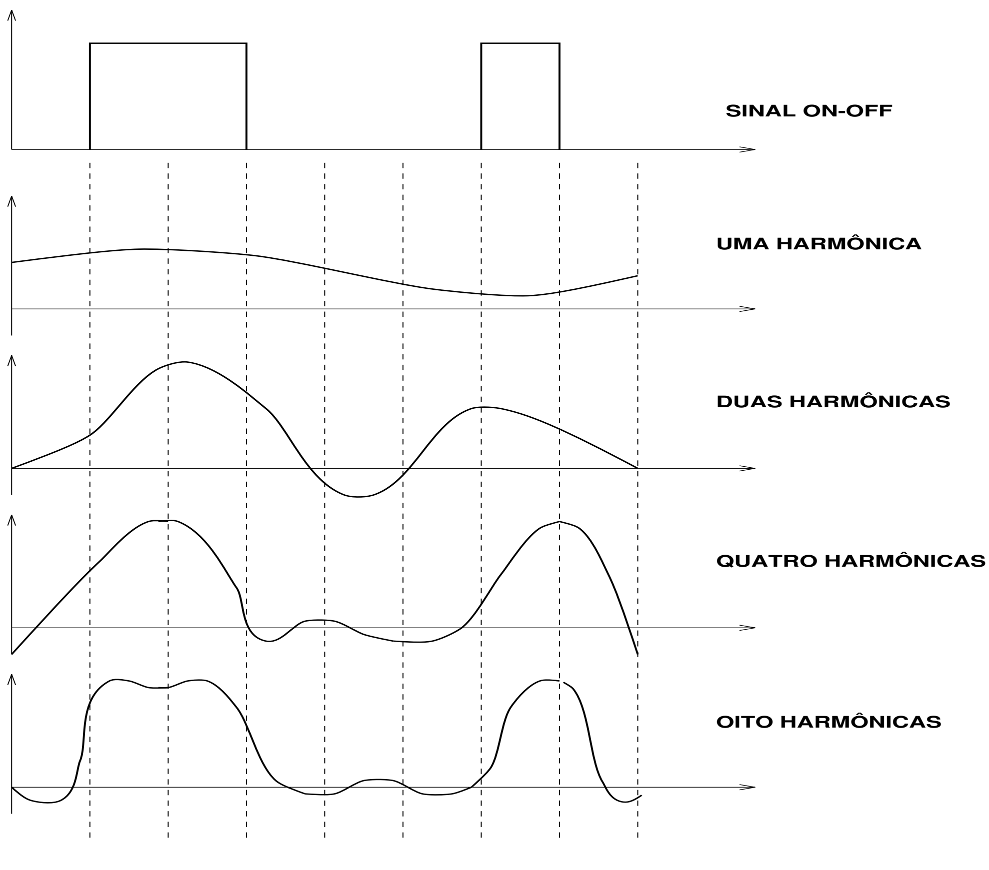

Figura 2.5: Um sinal periódico e suas harmônicas.

Interferência Entre Símbolos

Quando aplicamos uma onda retangular de tensão a um meio de transmissão elétrico, temos no receptor do sinal uma onda não retangular como mostra a figura 2.6. A subida íngreme da tensão é impedida pelas capacitâncias inerentes do meio, que necessitam armazenar energia para ter sua tensão variada. Esta energia é devolvida quando a tensão decai, impondo igualmente um decréscimo suave da tensão no meio.

Suponha a ocorrência do bit 1 seguido do bit 0. O decaimento suave da tensão pode fazer com que o receptor detecte o bit 1 no lugar do bit 0, caso a amostragem se dê mais próxima do início do intervalo do bit. A este fenômeno denomina-se *interferência entre símbolos*.

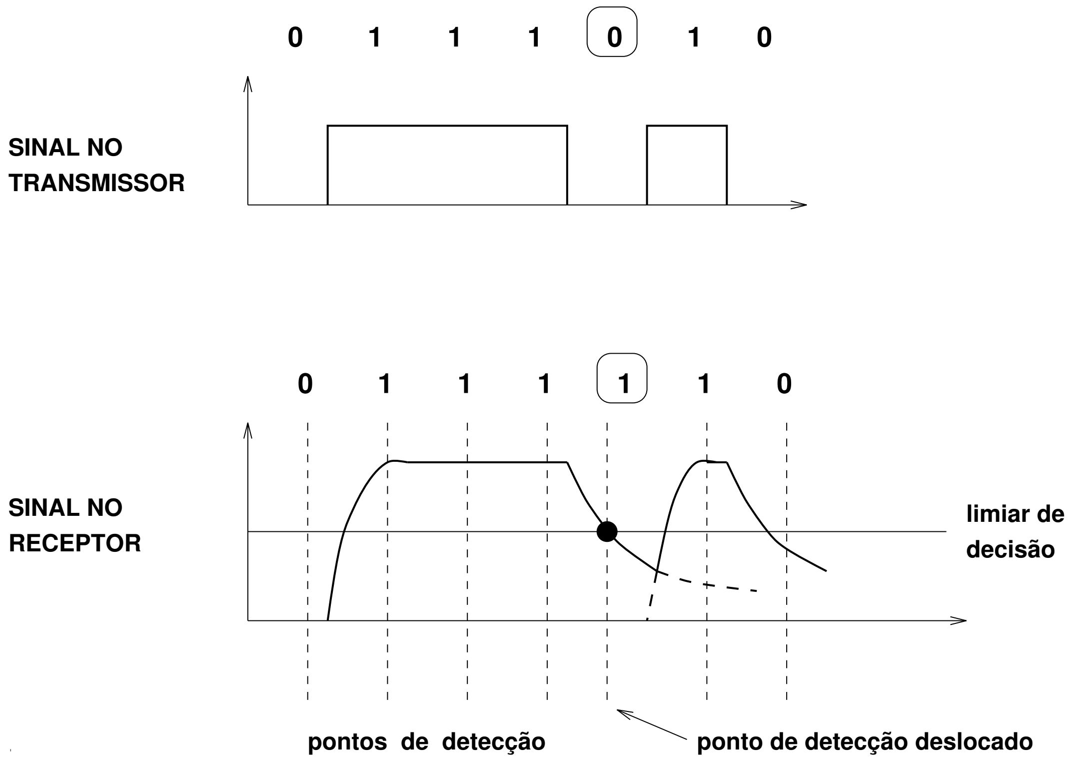

Figura 2.6: Interferência entre símbolos.

Jitter de Fase

Jitter de fase é uma imprecisão na fase do sinal digital causada por alteração do ponto de operação normal dos circuitos eletrônicos devido a variações de temperatura, oscilações da tensão de alimentação, etc. O jitter de fase causa uma mudança no instante de passagem pelo limiar que distingue uma transição (figura 2.7). Como consequência, o período de ocorrência dos bits tem um valor não constante, podendo causar erros na recepção do sinal.

O jitter de fase é medido pela faixa de tempo em que a transição pode variar (em valor absoluto ou relativo ao período do bit).

Ruído Térmico

Ruído térmico é uma perturbação de natureza aleatória que distorce o sinal. Transitórios térmicos e eletromagnéticos são as causas mais comuns de ruído térmico. Transitórios mecânicos é outra fonte de ruído térmico, alterando as características elétricas dos contactos (resistência e capacitância de contacto).

A figura 2.8 ilustra a distorção causada pelo ruído térmico.

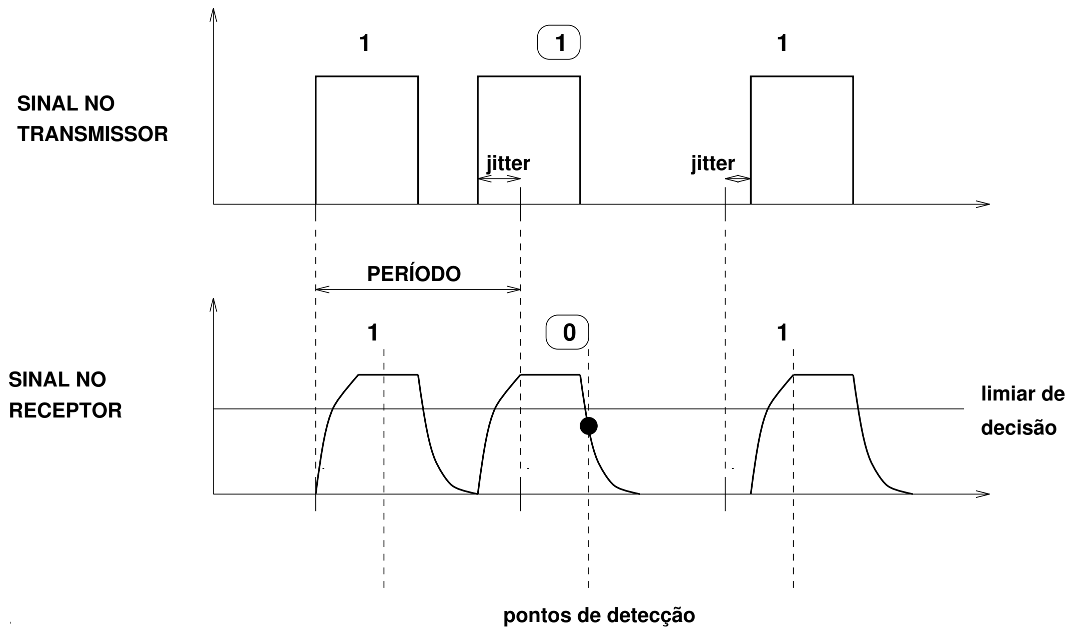

Figura 2.7: Erro de detecção devido ao jitter de fase.

Distorção por Atraso de Propagação

Um sinal pode ser representado por um certo número de componentes harmônicas senoidais. Nos meios físicos usuais a velocidade de propagação de uma onda senoidal depende de sua freqüência. Caso esta dependência seja acentuada, uma harmônica de freqüência alta pode ter sua fase alterada em relação as harmônicas de freqüências baixas. A figura 2.9 mostra este efeito.

2.7 Capacidade de Transmissão de um Canal

A freqüência com que um sinal pode se propagar no meio de transmissão é denominada *baud*. Um canal de 10 Mbaud permite 10^6 variações do sinal por segundo. A relação entre baud e bits por segundo (bits/baud) depende da forma de codificação do sinal. Um canal com capacidade de N bauds pode transportar N bits/s no caso de sinal digital com codificação ON/OFF ou NRZ. No caso de codificação Manchester, teremos N/2 bits/s em média.

Um canal de N bauds pode transportar um número maior que N de bits/s. É o caso de, por exemplo, utilizar-se um sinal digital com codificação ON/OFF e 4 níveis de tensão representando as ocorrências dos bits 00, 01, 10 e 11. Neste caso temos 2N bits/s. A codificação de múltiplos bits numa única variação do sinal é rara em redes locais devido a alta taxa de falhas na decodificação do sinal. Quando utilizada (principalmente em troncos telefônicos), emprega-se transmissão analógica com modulação PSK e ASK combinadas. (figura 2.10).

A capacidade de transmissão de um canal passa-baixas é limitada pelo teorema de Ny-

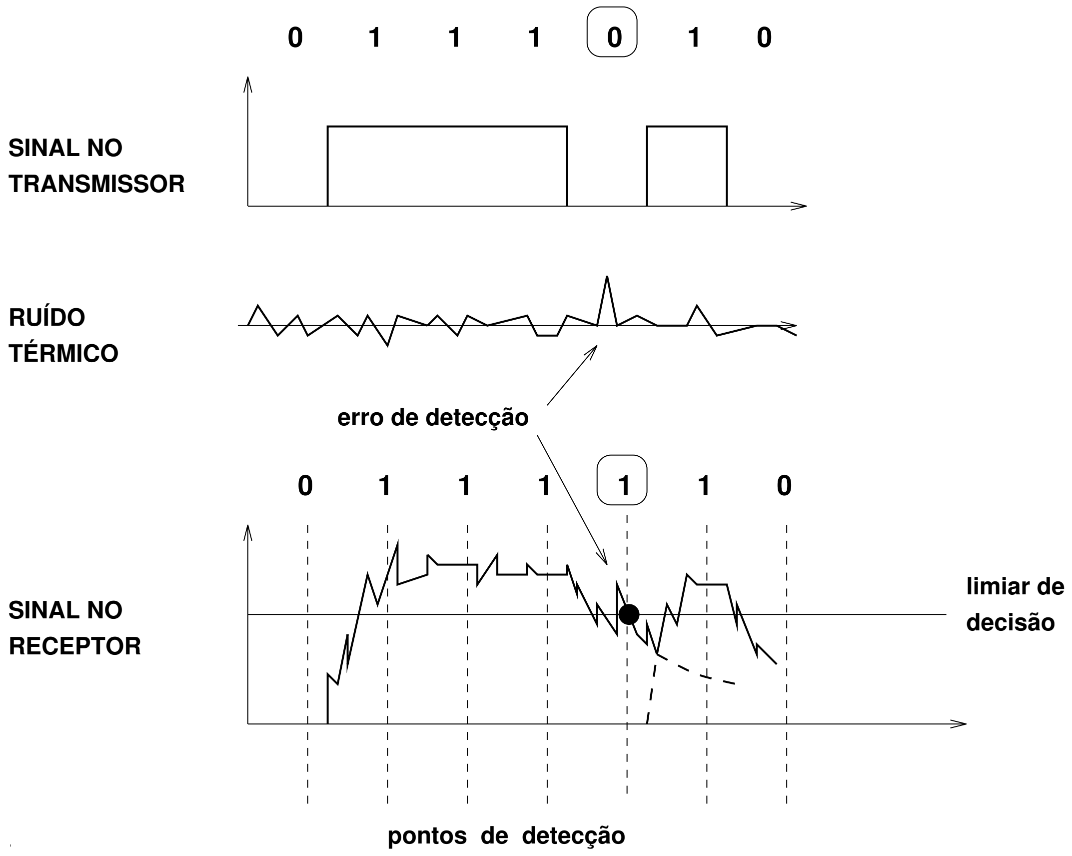

Figura 2.8: Erro de detecção devido ao ruído térmico.

quist: Um canal livre de ruído com largura de banda B transmitindo um sinal com V diferentes amplitudes possui uma taxa de transmissão T (em bits/s) dada por

$$T \leq 2B \log_2 V$$

Por exemplo, um canal com largura de banda 3 KHz apresenta uma capacidade máxima de 6 Kbits/s caso sejam empregadas duas amplitudades (modulação ASK) para codificar os níveis 0 e 1.

Intuitivamente, o limite de Nyquist se deve a filtragem de freqüências altas pelo canal tornando inútil amostragens com taxa maior que 2B.

Devido a presença das fontes de distorção descritas acima, o limite teórico de Nyquist nunca é atingido na prática. A interferência entre símbolos causadas pelas indutâncias e capacitâncias ao longo do meio é um fator preponderante na limitação da taxa de transmissão de um canal elétrico.

Em redes de computadores baseadas em canais elétricos, taxas da ordem de 10 Mbits/s são típicas. Em redes baseadas em fibra ótica, 100 Mbits/s é taxa típica. Redes operando na faixa de Gbits/s como as baseadas em tecnologia ATM (Asynchronous Transfer Mode) já estão disponíveis comercialmente.

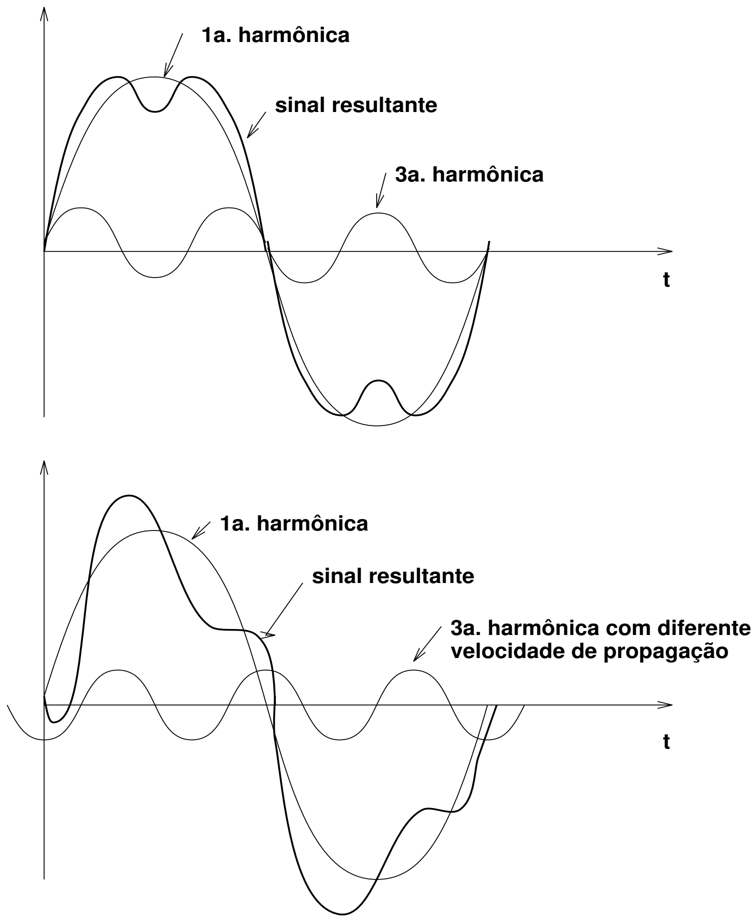

Figura 2.9: Distorção devido a variação da velocidade de propagação das harmônicas com a freqüência.

2.8 Meios de Transmissão

2.8.1 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 exteriores 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 freqüência máxima de transmissão depende do comprimento e da espessura do par de

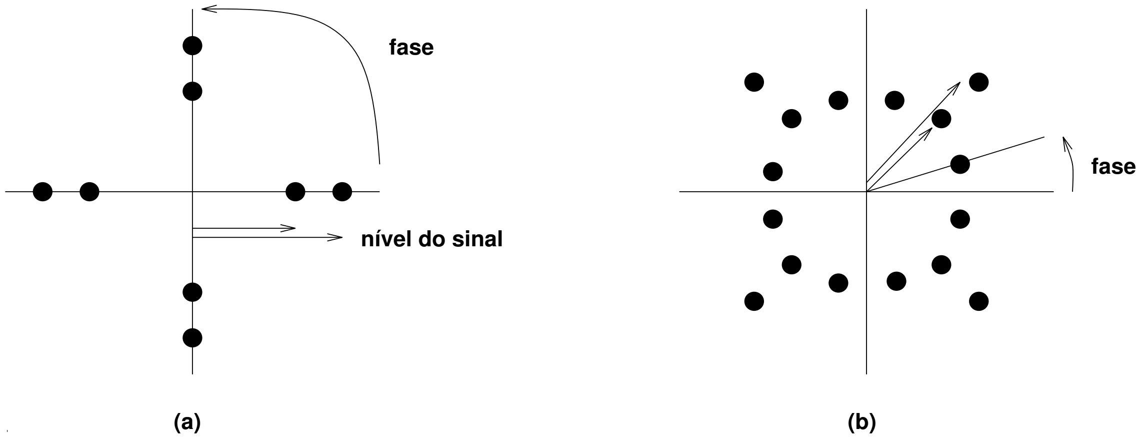

Figura 2.10: Modulações PSK e ASK combinadas para 3 bits/baud (a) e 4 bits/baud (b).

fios, o que em última instância dita a impedância elétrica do par. Para longas distâncias (quilometros) a taxa de transmissão não ultrapassa 20 Kbits/s, podendo atingir 100 Mbits/s em comprimentos de poucas dezenas de metros. É comum redes na faixa de 10 Mbits/s ter como meio de transmissão pares trançados para distâncias inferiores a 1 quilometro.

2.8.2 Cabo Coaxial

É composto de um condutor cilíndrico isolado envolto por uma malha de cobre e uma capa plástica de proteção (figura 2.11). 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 freqüências (efeito pelicular e irradiação). Entretanto, sua estrutura assimétrica contribui para a atenuação imposta à amplitude do sinal. Apesar de apresentar certa complexidade de instalação, o cabo coaxial é o meio de transmissão mais empregado em redes locais de computadores.

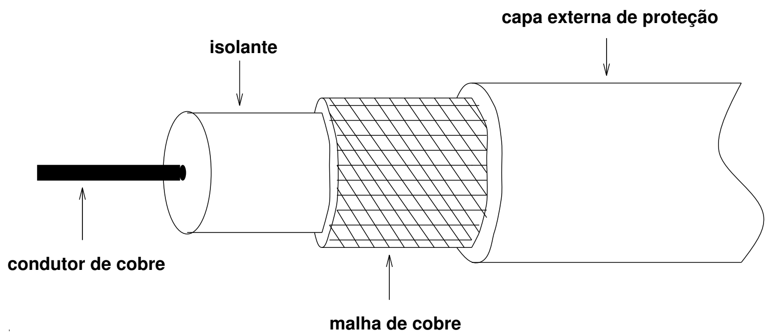

Figura 2.11: Cabo coaxial de cobre.

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 freqüência básica de transmissão (ou duas no caso de modulação FSK). Os cabos de 75Ω são denominados *cabos banda larga* por disporem de largura de faixa estendida permitindo a multiplexação pela divisão em freqüência (FDM) de vários canais.

A distância máxima de um cabo coaxial 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 freqüência do sinal e do número de conectores existentes (onde os hosts se conectam).

O gráfico da figura 2.12 mostra a atenuação da amplitude do sinal por quilometro de cabo em função da freqüência.

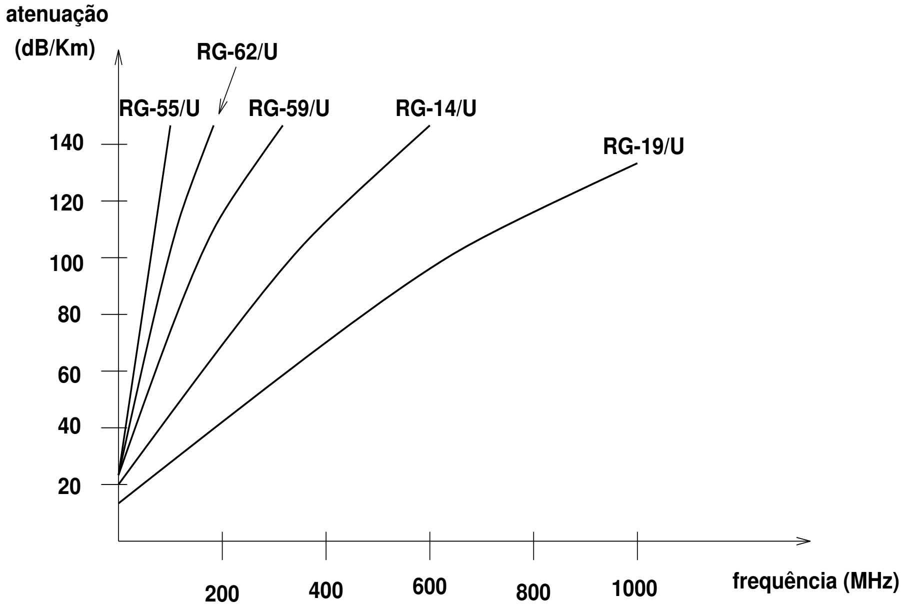

Figura 2.12: Atenuação da amplitude por quilometro em função da freqüência.

2.8.3 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 chamada fibra *multimodo* onde a luz é mantida no núcleo por reflexão na casca (figura 2.13). Fibras multimodo possuem diâmetros entre 50 e 200 μm. Atenuações de 1 a 5 dB por quiilometro na potência do sinal ótico são típicas.

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.

As 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

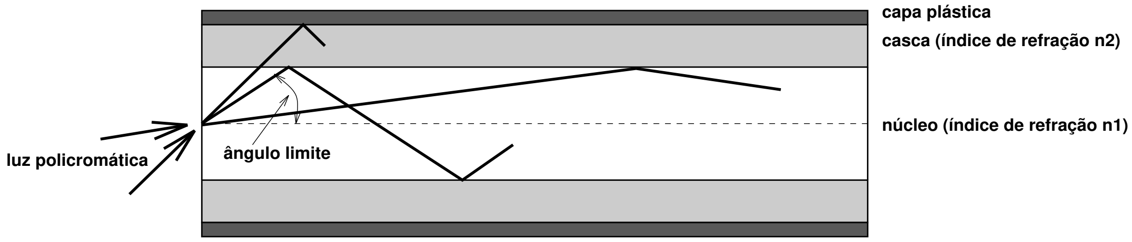

Figura 2.13: Fibra ótica multimodo.

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 da rede (figura 2.14).

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

2.9 Componentes da Camada Física

Os 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 seccioná-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.

Em redes de longa distância, define-se apenas dois componentes para a camada física:

1. DTE (Data Termininal Equipment): o equipamento do usuário conectado a rede (computador, videotexto, etc.);

2. DCE (Data Circuit-terminating Equipment) O *front-end* da rede colocado pela operadora junto ao DTE. Via de regra, é um modulador/demodulador (modem).

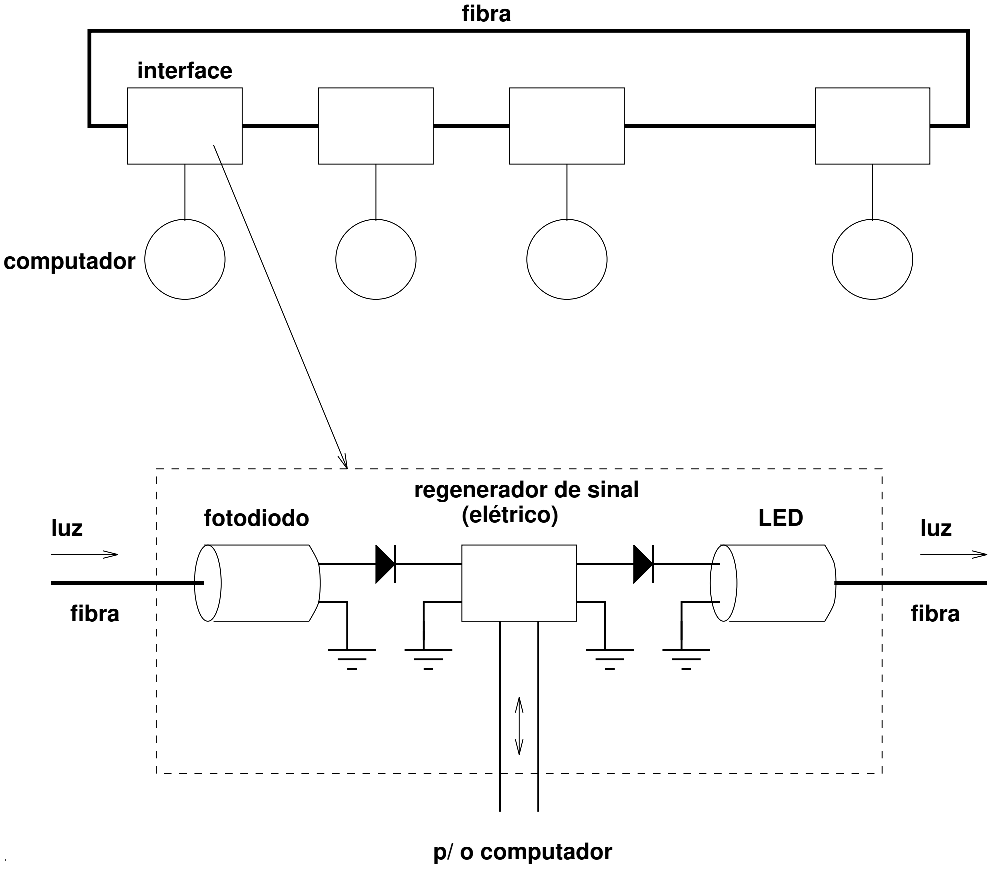

Figura 2.14: Rede de fibra ótica em anel com repetidores ativos.

2.10 Padrões da Camada Física

2.10.1 Redes locais

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 tipo do cabo, comprimento máximo, distância mínima entre MDIs, etc., são suficientes para o projeto, instalação e operação de redes locais de computadores no que tange à camada física.

Apenas a título de exemplo, vamos examinar algumas especificações do padrão IEEE 802.3. O 802.3 admite 3 meios de transmissão: cabo coaxial em banda básica (o mais utilizado), cabo coaxial em banda larga e fibra ótica em banda básica. As taxas de transmissão podem ser de 1, 5, 10 (a mais usual), e 20 Mbits/s.

A MAU emprega codificação Manchester com tensões variando de zero a -2.05 Volts. Os tempos de subida e descida do sinal devem estar compreendidos entre 25 $\pm$ 5 nanosegundos. A faixa limiar de tensão que distingue uma transição deve situar-se entre -1.492 e -1.629 Volts.

A MDI deve impor uma resistência de contacto não superior a 50 nanoohms (tanto para o condutor central quanto para a blindagem) e apresentar capacitância inferior a 2 pF a 10 MHz. O jitter máximo introduzido pela MAU não deve ser superior a 1.7 nanosegundos, e a atraso de propagação na MAU deve ser inferior a 257 nanosegundos.

Para cabos de 50Ω (banda básica), a atenuação máxima é de 17 dB/Km a velocidade de propagação do sinal elétrico deve ser superior a 77% da velocidade da luz no vácuo, e o jitter máximo inferior a 7 nanosegundos, tudo para freqüência de 10 MHz e 500 metros de cabo.

Dezenas de outros parâmetros elétricos e mecânicos fazem parte do documento IEEE 802.3.

2.10.2 Redes Públicas

Em redes públicas de longa distância, o único padrão existente na camada física é a interconexão do DTE ao DCE. 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 2.15).

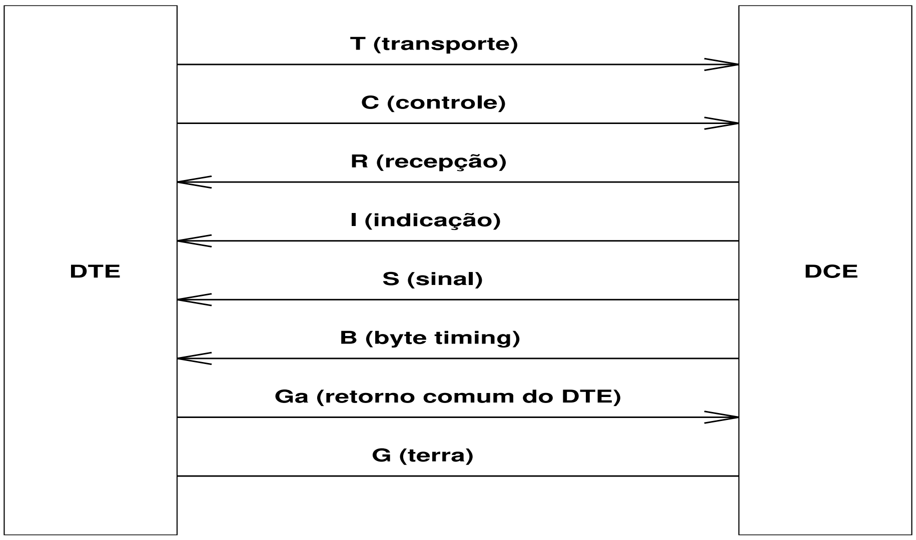

Figura 2.15: Linhas da interface X.21.

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 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.

2.11 Problemas

1. Quais os motivos de se empregar esquemas de codificação diferente do ON/OFF ?

2. Desenhe a forma de onda nos códigos Manchester e Manchester Diferencial do sinal composto dos bits 0001110101.

3. Diz-se que o código Manchester colabora para a manutenção do sincronismo entre o emissor e o receptor de um sinal. Comente esta afirmação.

4. Quais as formas de se modular um sinal digital ?

5. Qual a taxa de transmissão de um canal de 6 MHz utilizando-se modulação PSK com 4 avanços de fases distintos ?

6. O que é *Interferência Entre Símbolos* e *Jitter de Fase* e quais os seus efeitos na transmissão de sinais digitais ?

7. Quando se utiliza cabos coaxiais banda básica e banda larga ?

8. Por que a grande maioria das redes baseadas em fibra ótica apresentam topologias em anel ?

9. Descreva os componentes da camada física do padrão IEEE 802.

10. Redes públicas apresentam a camada física mais simples ou mais complexa que as redes locais ? Justifique.

Capítulo 3

A CAMADA DE ENLACE

A camada de 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. 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.

3.1 A Subcamada de Acesso ao Meio (MAC)

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*.

3.1.1 Técnicas de Acesso Aleatório

As técnicas de acesso aleatório são utilizadas em redes com topologia de barramento. Dois métodos de acesso aleatório serão descritos: métodos ALOHA (pioneiros) e sua evolução para os métodos de acesso múltiplo com detecção de portadora (CSMA: Carrier Sense Multiple Access).

Método ALOHA Puro

Este médodo foi concebido na Universidade do Hawaii para uma rede conectando unidades em quatro ilhas via rádio. A técnica necessita de reconhecimento por parte do receptor e segue o algoritmo abaixo:

1. transmita o quadro;
2. aguarde o reconhecimento da recepção por T unidades de tempo; se recebido, fim;
3. gere um número aleatório (r) entre 0 e R;
4. vá para 1 após r unidades de tempo.

A técnica ALOHA pura é basteante simples. Sempre que necessitar transmitir um quadro, o host simplesmente o faz. Caso ocorra colisão (interferência entre duas transmissões), o quadro será propagado com erro, causando o seu descarte pelo destinatário. O emissor detecta colisão pelo não recebimento do reconhecimento. Neste caso, a próxima retransmissão se dará após um intervalo de tempo aleatório. É importante tentar nova transmissão após um intervalo de tempo aleatório, pois, caso contrário, uma nova colisão certamente ocorrerá se ambos os hosts colidentes tentarem a retransmissão ao mesmo tempo.

A técnica ALOHA pura apresenta baixa eficiência na utilização do canal pois uma transmissão em curso está sempre sujeita a interferência de outra que se inicia. A variante a seguir impede interferências numa transmissão em curso.

Método ALOHA Particionado

Esta variante do ALOHA puro permite que transmissões se iniciem em intervalos de tempo bem definidos (partições). Se o período das partições for superior ao tempo de transmissão de um quadro, uma transmissão que se iniciou sem colisão será concluída sem colisão. Como desvantagem, o host deve esperar o início do próxima partição para transmitir, mesmo que o meio esteja livre. O algoritmo é dado abaixo:

1. aguarde o *beep* de início de partição (fornecido por uma estação mestre);
2. transmita o quadro;
3. aguarde o reconhecimento da recepção por T unidades de tempo; se recebido, fim.
4. gere um número aleatório (r) entre 0 e R;
5. vá para 1 após r unidades de tempo.

Método CSMA Não Persistente

Os métodos da família CSMA têm em comum a capacidade de *escutar* o meio físico para a detecção de uma transmissão em curso. Um host somente inicia a transmissão se detectar o meio em repouso (sem transições, no caso de transmissão digital). Colisões ainda podem ocorrer, se dois hosts detectarem o meio em repouso e iniciarem a transmissão ao mesmo tempo. Neste caso, a confirmação por parte do receptor, como nos métodos ALOHA, também é imprescindível.

O método CSMA Não Persistente opera segundo o algoritmo:

1. escute o meio;
2. se o meio estiver em repouso:
   (a) transmita o quadro;
   (b) aguarde o reconhecimento da recepção por T unidades de tempo; se recebido, fim;
   (c) vá para 1.
3. caso contrário (transmissão em curso):
   (a) gere um número aleatório (r) entre 0 e R;
   (b) vá para 1 após r unidades de tempo.

Quando detectada uma transmissão em curso, o método aguarda um intervalo aleatório antes de reiniciar a escuta do meio a fim de aguardar a sua liberação. Se a transmissão terminar logo após o início do intervalo aleatório, uma sub-utilização do meio é acarretada.

Método CSMA 1-Persistente

Este método é idêntico ao anterior, apenas fazendo o intervalo aleatório igual zero (escuta permanente do meio até cessar a transmissão em curso). Este método evita as esperas com o meio em repouso do anterior (aumentando portanto a utilização do canal) sob pena de um aumento da possibilidade de colisões quando dois hosts estão sensoriando o meio ocupado por um terceiro.

Método CSMA p-Persistente

É um meio termo entre o método Não Persistente e o 1-Persistente. O método detecta o meio permanentemente até a transmissão em curso se encerrar. Neste ponto, o método pode transmitir ou suspender a transmissão por um intervalo de tempo aleatório. Os passos do algoritmo CSMA p-Persistente são os seguintes:

1. escute o meio até ser detectada a condição de repouso;
2. gere um número aleatório (s) entre 0 e 1;
3. Se s $\geq$ p:

   (a) transmita o quadro;
   (b) aguarde o reconhecimento da recepção por T unidades de tempo; se recebido, fim;
   (c) vá para 1.

4. Caso contrário (s < p):
   (a) gere um número aleatório (r) entre 0 e R;
   (b) aguarde r unidades de tempo;
   (c) escute o meio; se em repouso vá para 2;
   (d) caso contrário (transmissão em curso):
      i. gere um número aleatório (u) entre 0 e U;
      ii. vá para 1 após u unidades de tempo.

O método CSMA p-Persistente apresenta uma boa taxa de utilização do meio com baixa probabilidade da de ocorrência de colisões caso p seja ajustado às características do tráfego.

Método CSMA-CD

O método CSMA-CD (Collision Detection) adiciona aos métodos CSMA a detecção de colisões *sem a necessidade de aguardar reconhecimento por parte do receptor.* 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 repouso;
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.

A rede Ethernet foi pioneira na introdução do método CSMA-CD, sendo comum empregar-se o termo *Ethernet* para este método de acesso ao meio.

CSMA-CD no Padrão IEEE 802.3

O padrão IEEE 802.3 estabelece o formato de quadros apresentado na figura 3.1.

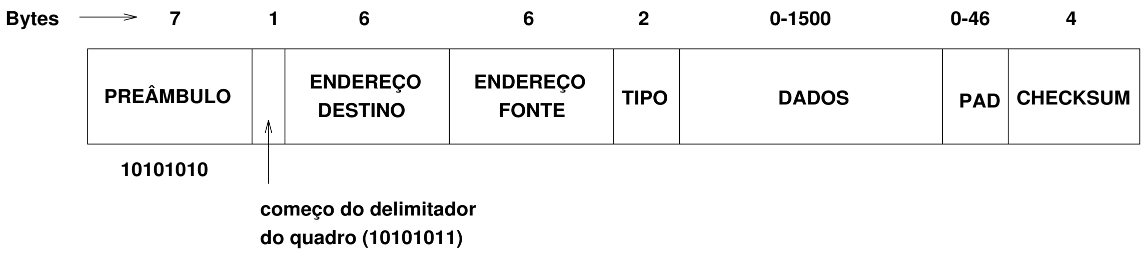

Figura 3.1: Formato dos quadros no IEEE 802.3

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 seqüências 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 seja insuficiente 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 podem entrar em colisão sem que os respectivos emissores a detectem (isto é, quando 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*.

3.1.2 Métodos Baseados em Passagem de Permissão

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 posse do *token*, o host a passa para 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).

O anel pode conter uma *estação mestre* que tem como função verificar se o *token* não se perdeu, reiniciar o anel em caso de falhas, remover quadros corrompidos, etc. Em caso de falha desta estação, outra é eleita como mestre (automaticamente ou com a intervenção do operador).

Apesar da simplicidade do conceito, métodos de acesso ao meio baseados na passagem de permissão são bem mais complexos que os métodos de acesso aleatório. A seguintes situações devem ser tratadas:

- iniciação do anel quando o primeiro host é ligado;
- inserção de novos hosts no anel;
- reconfiguração do anel ante a falha ou o desligamento programado de hosts.

Apesar de associados à topologia em anel, métodos baseados em passagem de permissão também se aplicam a topologia de barramento. Neste caso, forma-se um anel lógico ordenando os hosts de acordo com algum critério, por exemplo, seus endereços. O *token* circula neste anel lógico, dando permissão ao host que o detém de difundir quadros no meio físico. Redes com topologia de barramento que empregam passagem de permissão como método de acesso ao meio são denominadas redes *token bus* (figura 3.2).

Nestas redes o hosts emissor não dispõe de confirmação por parte do receptor como nas redes *token ring* visto que o quadro é simplesmente difundido no barramento.

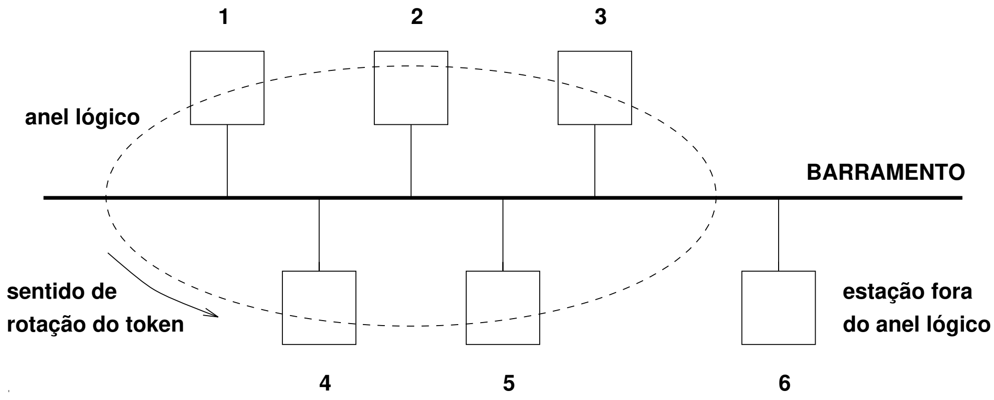

Figura 3.2: Passagem de permissão em redes com topologia de barramento.

Redes *token bus* apresentam um atrativo adicional em relação a redes *token ring*: um host pode receber mensagens sem estar participando do anel lógico (host 6 na figura 3.2), posto que todas as mensagem são transmitidas em difusão pelo meio físico. Tais hosts são incapazes de transmitir, pois jamais estarão de posse do *token* (no máximo podem responder a uma mensagem a eles direcionada). Esta característica das redes *token bus* viabiliza a inclusão de processadores extremamente simples na rede (microcontroladores, por exemplo), sem dotá-los da capacidade plena de acesso ao meio (e portanto sem empregar todas as funcionalidades das sete camadas do modelo OSI).

3.1.3 Passagem de Permissão no Padrão IEEE 802.5

O padrão IEEE 802.5 *(token ring)* é bem mais complexo que o CSMA-CD do 802.3. Duas características não presentes no 802.3 são definidas no 802.5: prioridade de acesso ao meio e reserva do meio.

O *token* é composto de 3 bytes: DI (delimitador de início) CA (controle de acesso) e TIPO. O primeito byte, DI, identifica o início do *token* e é formado por transições inválidas do código Manchester Diferencial (transições que jamais ocorrem em cadeias de 0s e 1s tais como duas transições positivas ou negativas seguidas). O segundo byte, CA, é utilizado para controle de acesso ao meio, sendo composto de agrupamentos de bits em 4 categorias:

1. status (1 bit): se o *token* está livre ou não;

2. monitor (1 bit): se o *token* passou pela estação mestre ou não. A estação mestre ativa este campo sempre que um *token* passar por ela;

3. prioridade (3 bits): estipula a prioridade mínima dos quadros que podem ser transmitidos com a captura do *token*. Se o valor deste campo for N, um host detentor do *token* pode transmitir quadros de prioridade maior ou igual a N;

4. reserva (3 bits): determina a prioridade do próximo *token* livre. Ao gerar um novo *token*, caso o valor da reserva seja maior que o valor da prioridade do token capturado, o host atribui o valor da reserva como prioridade do *token*. Se um host deseja reservar o *token*, este atribui ao campo de reserva a prioridade dos quadros que tem para transmitir, caso esta prioridade seja maior que a prioridade de reserva corrente. Para evitar que a prioridade do *token* cresça indefinidamente, todo o host que aumentar a prioridade da reserva e capturar o *token* se compromete a liberá-lo com prioridade menor.

O byte de TIPO estipula o tipo da informação que o quadro carrega: dados oriundos das camadas superiores ou controle (este último utilizado para a manutenção do anel).

É freqüente na literatura a afirmação que redes *token ring* são determinísticas, isto é, apresentam um tempo de acesso ao meio limitado (aproximadamente o tempo máximo de retenção do *token* multiplicado pelo número de hosts no anel). Tal propriedade ocorre somente se o esquema de prioridade e reserva não seja utilizado.

A figura 3.3 mostra o formato de um quadro no padrão IEEE 802.5.

Capturado o *token*, o quadro é injetado no anel logo a seguir. Os campos de endereço e *checksum* são idênticos ao IEEE 802.3. A quantidade de dados é ilimitada (a rigor, limitada pelo tempo máximo que um host pode reter o *token*). O campo DF (delimitador de final) também é composto por transições Manchester Diferencial inválidas. Um campo após o DF, ST (status) contém dois bits: A e C. O bit A é ativado pelo host destino e informa o host fonte que o destinatário tomou conhecimento do quadro a ele endereçado. O bit C

Bytes →	1	1	1	6	6	sem limites	4	1	1
	DI	CA	TIPO	ENDEREÇO DESTINO	ENDEREÇO FONTE	DADOS	CHECKSUM	DF	ST

Figura 3.3: Formato dos quadros no IEEE 802.5

é ativado se o destinatário aceitou o quadro (pode tê-lo rejeitado por falta de área para armazenamento, por exemplo).

Duas observações importantes:

1. um quadro no IEEE 802.5 não define campo de tamanho do quadro; o campo de dados começa 14 bytes após o campo DI do *token* e termina 4 bytes antes do campo DF do quadro;

2. o campo DF deve obrigatoriamente preceder o campo ST; o destinatário está em condições de aceitar um quadro (bit C do campo ST) somente após computar o *checksum* e, como não existe quadro de tamanho de dados, o campo de *checksum* só é definido após o recebimento do campo DF.

Manutenção do Anel

Uma das estações do anel é rotulada como estação mestre (EM). Via de regra, é a primeira estação a completar o procedimento de *boot*. Caso esta estação falhe, uma nova EM é eleita. Periodicamente, a EM circula um *token* com o campo TIPO contendo a informação ACTIVE_MONITOR_PRESENT. Se este *token* ficar sem circular por determinado período, inicia-se um procedimento de escolha de uma nova EM.

Assim que uma estação termina o procedimento de *boot*, ela aguarda a passagem do *token* ou um quadro de ACTIVE_MONITOR_PRESENT. Expirado este tempo de espera, a estação gera um quadro de controle com a informação CLAIM_TOKEN. Se este quadro circular sem alteração, a estação que o emitiu se torna a estação mestre. Se já existir uma EM, a mesma altera o quadro de CLAIM_TOKEN, informando ao emissor do quadro a sua existência.

São atribuições da estação mestre:

- drenar quadros corrompidos do anel;

- drenar quadros órfãos do anel (quadros não drenados pelo emissor por falhas de hardware ou software);

- verificar se o *token* não se perdeu (o host detentor do *token* falha em injetar um novo *token* no anel). Neste caso, a estação mestre gera um novo *token* no anel.

O dreno de quadros pela estação mestre é feito caso o bit de monitor do quadro CA estiver ativado quando o quadro passar pela EM (indicando que se trata de uma segunda passagem).

Finalmente, quando um host suspeitar da ruptura do anel (tempo longo sem a passagem de *tokens*), este injeta um *token* de controle com a informação BEACON no campo TIPO. Se o quadro voltar ao emissor, este supõe que o problema foi sanado. Caso contrário, a estação entra num estado de *standby* aguardando o reestabelecimento do anel.

3.1.4 Passagem de Permissão no Padrão IEEE 802.4

O padrão IEEE 802.4 *(token bus)* emprega topologia de barramento, sendo o meio controlado por passagem de permissão. O anel lógico (figura 3.4) é estabelecido ordenando-se os hosts de acordo com os respectivos endereços (número formado pelos 48 bits que compõem o endereço). O *token* circula no sentido do endereço mais alto para o mais baixo.

O padrão define quatro prioridades para os quadros: 0, 2, 4 e 6 (a mais alta). Logicamente, é como se cada host tivesse quatro filas de quadros para serem transmitidos. De posse do *token*, o host transmite os quadros de prioridade 6. Esgotados estes, os de prioridade 4 são transmitidos e assim por diante até que todos os quadros pendentes se esgotem ou o tempo máximo de posse do *token* for atingido.

Os quadros do padrão IEEE 802.4 têm o formato dado pela figura 3.4.

Bytes →	>= 1	1	1	6	6	0-8174	4	1
	PA	DI	TIPO	ENDEREÇO DESTINO	ENDEREÇO FONTE	DADOS	CHECKSUM	DF

Figura 3.4: Formato dos quadros no IEEE 802.4

O preâmbulo (PA) tem duracão superior ou igual a um período de um byte. Sua função básica é permitir que os hosts se preparem para receber o quadro, sincronizando seus relógios com o do host emissor. A seguir, o campo DI (delimitador de início) contém codificações inválidas como no IEEE 802.5. O campo TIPO contém os seguintes dados:

1. tipo do quadro (2 bits): se quadro de controle, de dados ou de gerenciamento do anel.

2. prioridade (3 bits): prioridade mínima dos quadros que podem ser transmitidos com a captura do token.

O padrão IEEE 802.4 não provê mecanismo de reserva como no IEEE 802.5. O campo de dados pode conter no máximo 8174 bytes. O campo de *checksum* e o delimitador de final (DF) são similares ao 802.5.

Manutenção do Anel Lógico

Cada estação mantém o endereço de sua antecessora e de sua sucessora no anel. Não existe estação mestre. O anel é iniciado quando a primeira estação termina o procedimento de *boot*. A estação monitora o meio por determinado período de tempo. Caso nenhuma atividade seja detectada, a estação emite um quadro de controle to tipo CLAIM_TOKEN. Se não houver contestação, a estação se torna detentora do *token* (e neste caso, sozinha no anel).

De posse do *token* e terminadas as transmissões, caso o período de posse do *token* não tenha se expirado, a estação cria uma *janela de inclusão*, dando chance a outras estações de ingressarem no anel lógico. A estação detentora do *token* propaga um quadro do tipo SOLICIT_SUCESSOR com o seu endereço no campo de endereço fonte e o de sua atual sucessora no campo de endereço destino do quadro. Caso alguma estação esteja esperando para ingressar no anel (isto é, sua tentativa de iniciar o anel falhou) e seu endereço esteja situado entre estas estações, a mesma responde informando seu endereço. A estação emissora armazena o novo endereço de sua sucessora, e informa a sua antiga sucessora que sua antecessora mudou. A estação ingressante tem os endereços de sua antecessora e sucessora no quadro que iniciou o procedimento. Caso mais de uma estação responda à proposta de inclusão, um procedimento de contenção seleciona apenas uma (ficando as demais aguardando as próximas janelas de inclusão).

A saída de uma estação no anel é um procedimento mais simples que o ingresso. De posse do *token*, a estação propaga um *token* do tipo SET_SUCESSOR dirigida a sua antecessora com o endereço de sua sucessora no quadro. A estação antecessora armazena sua nova sucessora, eliminando assim a estação que iniciou o processo do anel lógico.

A passagem do *token* é um processo delicado. A estação cria um quadro de controle do tipo TOKEN com o campo de endereço destino igual ao de sua sucessora. Recebido pela sucessora, esta se considera de posse do *token*. O host que passou o *token* certifica-se de seu recebimento através da escuta do meio. A estação que recebeu o *token* acessa o meio logo a seguir seja para transmitir seus quadros, seja para passar o *token* adiante. Caso o meio fique inativo, a estação que passou o *token* propaga um quadro do tipo WHO_FOLLOWS fornecendo o endereço de sua sucessora no campo de endereço destino do quadro. A estação que tiver como antecessora este endereço responde, tornando-se a nova sucessora a receber o *token* (a estação que falhou em receber o *token* é eliminada do anel lógico, podendo ingressar mais tarde via janela de inclusão). Caso o procedimento de WHO_FOLLOWS falhe, um quadro do tipo SOLICIT_SUCESSOR2 é propagado solicitando que todas as estações que seguem a emissora no anel lógico respondam. Neste procedimento, mais de uma estação é eliminada do anel lógico.

Duas situações devem ainda ser gerenciadas:

- *tokens* duplicados: se uma estação detentora do *token* "ouvir" uma transmissão, esta o descarta (eventualmente todos os *tokens* duplicados são descartados, vide abaixo);

- perda do *token*: se uma estação não receber o *token* por um período longo, esta inicia um procedimento de CLAIM_TOKEN. Caso mais de uma estação o faça, um algoritmo de contenção indica a estação vencedora (que terá a posse do *token*).

3.2 A Subcamada de Enlace Lógico (LLC)

A subcamada LLC provê serviços à camada de rede através de SAPs (ver Cap. 1) denominadas LSAPs. Tipicamente, 3 serviços podem estar disponíveis nas LSAPs (detalhados a seguir):

1. serviço sem conexão e sem reconhecimento;
2. serviço sem conexão com reconhecimento;
3. serviço orientado a conexão.

Para prover tais serviços, a subcamada LLC deve dispor das seguintes funcionalidades:

- partição dos pacotes oriundos da camada de rede em quadros compatíveis com a subcamada de acesso ao meio;
- detecção de quadros corrompidos por erros de transmissão;
- recuperação de quadros corrompidos;
- estabelecimento e gerenciamento de conexões;
- controle do fluxo de quadros.

3.2.1 Montagem de Quadros

Os quadros são definidos de forma diferente para redes locais ou públicas. Em redes locais, os quadros têm o formato imposto pela subcamada de acesso ao meio e os dados referentes ao protocolo de enlace são incorporados num PDU (ver Cap. 1) que trafega no campo de dados do quadro.

Em redes públicas, não existe subcamadada de acesso ao meio, e os quadros são definidos justamente para implementar os protocolos de enlace. Neste caso, os dados do protocolo possuem campos exclusivos no quadro.

Na montagem de quadros, a delimitação dos mesmos merece alguns comentários. Caso os quadros sejam compostos exclusivamente de caracteres (o que é raro na atualidade) pode-se definir caracteres especiais para a delimitação de quadros. Por exemplo, o código ASCII reserva alguns códigos especiais para delimitação de blocos de dados como o *Data Link Escape* (DLE), o *Start of Text* (STX) e o *End of Text* (ETX). Assim, um quadro pode ser delimitado por:

```
DLE STX ... <texto em ASCII> ... DLE ETX
```

Os caracteres DLE, STX e ETX não ocorrem no interior do texto, pois são caracteres de controle.

Atualmente, os dados que compõem o quadro são seqüências arbitrárias de bits (números, segmentos de memória, etc) que não podem ser representados como seqüência de caracteres (texto). Nestes casos, duas técnicas são bastante empregadas:

Enchimento de Bits ***(bit stuffing)***

Escolhe-se um delimitador de ínicio de quadro (por exemplo 01111110) e previne-se de sua ocorrência nos dados com uma regra simples como: *A cada ocorrência de cinco 1s adiciona-se um 0.* O receptor do quadro executa procedimento inverso na recepção. A figura 3.5 ilustra este procedimento.

(a) **0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0**

(b) **0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0**

bits adicionados

(c) **0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0**

Figura 3.5: Enchimento de bits: (a) dados originais, (b) dados no quadro, (c) dados decodificados pelo receptor.

Violação de Códigos da Camada Física

Delimita-se o quadro com códigos inválidos empregados pela camada física [1]. Dado que tais códigos jamais ocorrem na codificação de 0s e 1s, sua detecção indica o início ou o final de quadros. No código Manchester Diferencial (empregado pelo 802.5) utiliza-se transições positivas (H) seguidas de negativas (L): HHLLHHLL. Na modulação FSK-coerente (empregada no 802.4), pode-se misturar freqüências altas e baixas num mesmo período do bit, gerando assim um código inválido (figura 3.6).

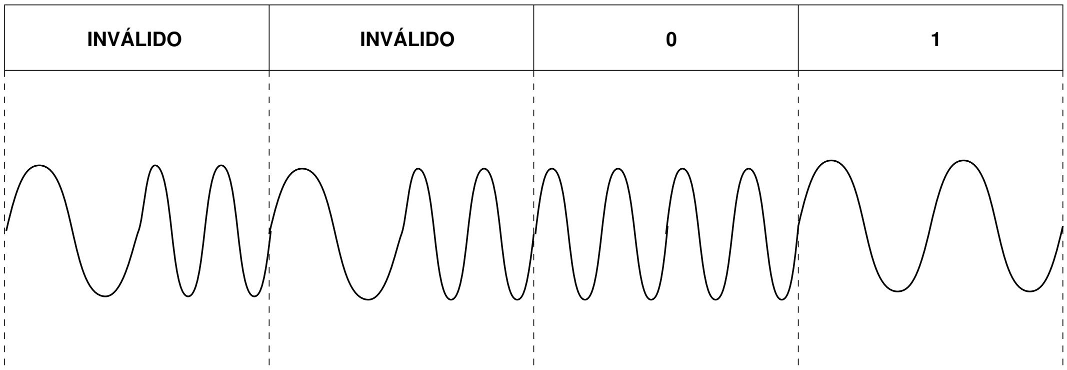

Figura 3.6: Exemplo de modulação FSK-coerente com códigos inválidos.

[1]A rigor, quem gera os quadros inválidos é a camada física a partir de indicador suprido pela LLC.

3.2.2 Detecção de Erros

A técnica mais usual para detecção de erros[2] durante a transmissão de quadros é a *Redundância Cíclica* (CRC). Esta técnica não utiliza bits de redundância o suficiente para correção do erro. Via de regra, detectado que um quadro chegou com erro, o receptor solicita ao emissor sua retransmissão. Em redes de computadores, a retransmissão é a técnica mais usual de correção (recuperação) de erros.

A técnica CRC é similar aos *dígitos de controle* presentes no CPF, contas bancárias, etc. O dígito de controle é computado por uma seqüência de operações bem definida na origem e transmitido ao destino que o recomputa. Em havendo diferença entre os valores transmitido e computado, conclui-se sobre a invalidade do número.

Na técnica CRC pode-se imaginar que os bits do quadro formam um (gigantesco !) número inteiro. Se o quadro é composto por k bits (numerados de 0 a k-1), este número é dado pelo polinômio:

$$P(x) = b(k-1).x^{k-1} + b(k-2).x^{k-2} + ...b(1).x + b(0)$$

onde b(j) é o valor do bit na posição j (0 ou 1) e x é a base da representação (2). Por exemplo, a seqüência de 10 bits 1101011011 é representada pelo polinômio $x^9 + x^8 + x^6 + x^4 + x^3 + x + 1$.

A técnica CRC reserva um número arbitrário de bits (n) para detecção de erros *(checksum)*. Assim, são transmitidos n+k bits, sendo k de informação seguidos de n bits de *checksum*. Os n bits de *checksum* são computados de forma que os n+k bits do quadro sejam representados por um polinômio $F(x).P(x)$, sendo $F(x)$ de ordem n.

Seja um polinômio de referência, $G(x)$ de ordem n (denominado *polinômio gerador*). Podemos escrever a seguinte igualdade:

$$F(x).P(x) = Q(x).G(x) + R(x)$$

onde Q(x) é de ordem k-1 e R(x) de ordem n-1.

Na técnica CRC escolhe-se $F(x) = x^n$ e computa-se $R(x)$ dividindo-se os inteiros (não os polinômios !) $F(x).P(x)$ por $Q(x)$. Neste caso, $R(x)$ formará os bits de *checksum*.

As operações são feitas em aritmética módulo 2 (sem o "vai 1"):

Adição	Subtração	Multiplic.	Divisão
0 + 0 = 0	0 - 0 = 0	0 x 0 = 0	0 ÷ 1 = 0
0 + 1 = 1	0 - 1 = 1	0 x 1 = 0	1 ÷ 1 = 1
1 + 0 = 1	1 - 0 = 1	1 x 0 = 0	
1 + 1 = 0	1 - 1 = 0	1 x 1 = 1	

Em aritmética módulo 2, pode-se escrever

$$F(x).P(x) = Q(x).G(x) + R(x)$$

[2] Erro neste contexto é a inversão do valor de um ou mais bits. Inversões de N bits em seqüência é denominada *erro de rajada de comprimento N*.

como

$$F(x).P(x) + R(x) = Q(x).G(x)$$

Como $F(x) = x^n$ o lado esquerdo da igualdade acima representa os bits originais acrescidos dos bits referentes a R(x) a direita.

No exemplo da seqüência 1101011011, considerando-se $n = 4$ (4 bits de *checksum*) e tomando-se o polinômio gerador: $G(x) = x^4 + x + 1$ temos

$$Q(x) = x^9 + x^8 + x^3 + x$$

$$R(x) = x^3 + x^2 + x$$

A figura 3.7 ilustra as operações, que nas implementações práticas são efetuadas por hardware.

Asssim os bits 1101011011 são transmitidos com quatro bits adicionais: 1101011011-1110

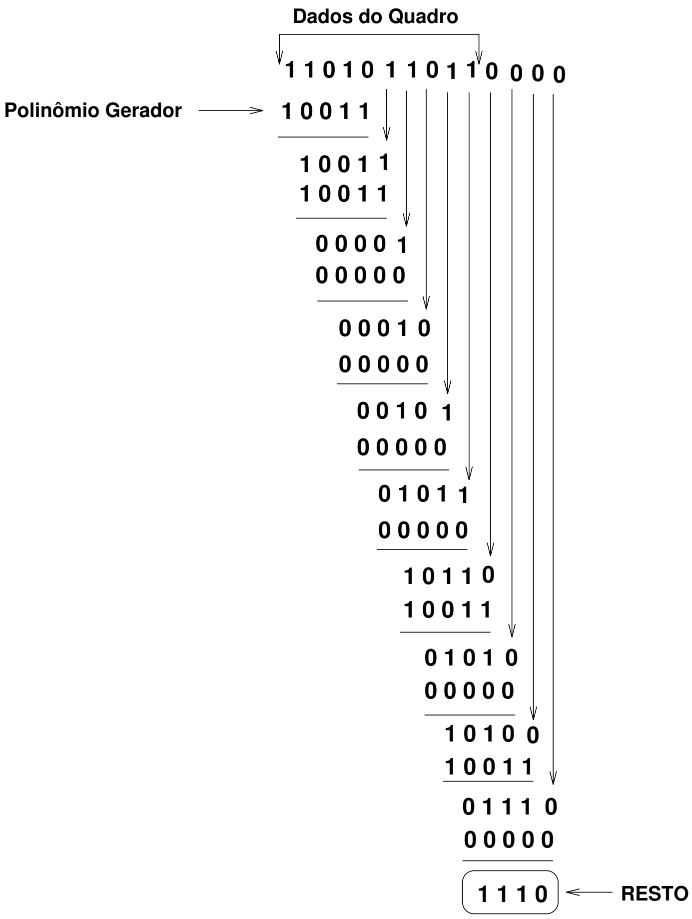

Figura 3.7: Exemplo do cômputo do *checksum*.

Tanto a ITU-T como o IEEE padronizaram polinômios geradores para o cômputo de *checksum*.

$CCR - 16 : x^{16} + x^{15} + x^2 + 1$
$CCR - CCITT : x^{16} + x^{12} + x^5 + 1$
$IEEE - 802.2 : x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$
Os polinomios CCR-16 e CCR-CCITT, para 16 bits de *checksum* são capazes de detectar:

- inversões de 1 ou 2 bits;
- inversões de um número ímpar de bits;
- rajadas de comprimento menor ou igual a 16;
- 99,997% das rajadas de comprimento 17;
- 99,998% das rajadas de comprimento 18.

3.2.3 Técnicas de Recuperação de Erros por Retransmissão

As técnicas mais usuais de recuperação de erros por retransmissão são baseadas em reconhecimento.

Reconhecimento Positivo

Neste método, ao transmitir um quadro, o emissor aguarda outro de reconhecimento por parte do receptor, caso o primeiro tenha sido recebido livre de erros. Recebido um quadro de reconhecimento, o emissor envia o próximo quadro e assim sucessivamente. Caso o receptor tenha detectado um erro (por exemplo, diferença entre o *checksum* computado e enviado), este simplesmente suprime o envio do reconhecimento. Expirado o tempo de espera pelo reconhecimento, o emissor retransmite o quadro.

Reconhecimento Negativo

Neste método o receptor sempre transmite um quadro de reconhecimento imediatamente após a recepção de um quadro: reconhecimento positivo, caso nenhum erro tenha sido detectado, ou negativo, caso contrário. A recepção de um reconhecimento negativo faz o emissor retransmitir prontamente o quadro, sem a necessidade de espera como no método anterior.

Reconhecimento Contínuo

Os métodos de reconhecimento positivo e negativo apresentam dois inconvenientes:

1. duplicação de quadros: 0 receptor recebe quadros duplicados em duas situações: quando o reconhecimento chegar após ter-se expirado seu tempo de espera; ou quando o reconhecimento é descartado na sua recepção devido a erros;
2. baixa eficiência: para cada quadro transmitido, circula outro de controle (reconhecimento) no sentido inverso.

O método de reconhecimento contínuo permite o emissor transmitir até N (largura da janela) quadros sem a necessidade de espera por reconhecimento. Cada quadro carrega dois contadores, N(S) e N(R), cujos valores dependem do tipo do quadro (de informação ou reconhecimento).

A figura 3.8 compara o reconhecimento contínuo com o reconhecimento negativo.

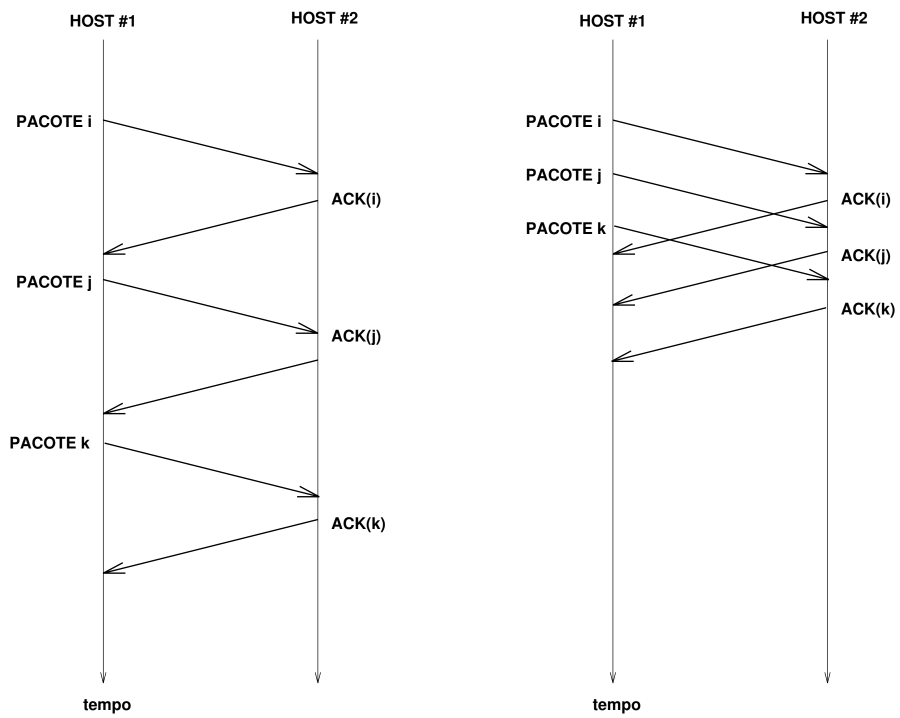

Figura 3.8: Reconhecimento negativo (esquerda) comparado com reconhecimento contínuo com janela de tamanho 3.

Nos quadros de informação (que fluem no sentido emissor-receptor), N(S) indica o número do quadro sendo transmitido e N(R) o número do próximo quadro esperado pelo emissor.

Nos quadros de reconhecimento, N(R) indica o próximo quadro sendo esperado (implicitamente os quadros numerados até N(R)-1 foram recebidos livre de erros). N(S) e N(R) ocupam poucos bits, tipicamente 3, sendo neste caso um contador em módulo 8. Isto justifica a introdução da janela em curso, permitindo maior segurança na identificação dos quadros.

O reconhecimento contínuo permite duas formas de operação quanto ao reconhecimento:

1. um quadro de reconhecimento para toda a janela;

2. quadros individuais de reconhecimeto.

No primeiro caso o emissor verifica se N(R) indica que a janela em curso se completou com sucesso. Neste caso, uma nova janela é iniciada com a transmissão dos próximos N quadros. Caso contrário, o emissor retransmite os quadros a partir de N(R) até o último quadro da janela, aguardando novo reconhecimento. Esta técnica reduz considravelmente os quadros de reconhecimento, aumentando a eficiência do enlace.

No segundo caso (figura 3.8) o transmissor *desliza* (avança) a janela a cada quadro de reconhecimento recebido. Como para cada quadro de dado existe um de reconhecimento, a janela desliza continuamente, o que não ocorre no primeito caso. Esta técnica minimiza as retransmissões, sendo atrativa para enlaces sujeitos a altas taxas de erro.

Quadros duplicados podem ainda ocorrer (por exemplo, quando um quadro de reconhecimento é corrompido), mas sua detecção é simples, posto que os quadros são identificados um a um.

3.2.4 Formas de Estabelecimento do Enlace

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 dispersos geograficamente.

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.

Enlaces ponto-a-ponto são constituídos tipicamente por duas estações combinadas, enquanto enlaces multiponto são formados por uma primária e várias secundárias.

Um enlace pode ser estabelecido para operar em um dos três modos abaixo:

1. modo de resposta normal: a estação primária envia um quadro de consulta para as suas secundárias inquirindo sobre a existência de quadros para transmitir; em caso positivo, a secundária transmite imediatamente após ter recebido o quadro de consulta;
2. modo de resposta assíncrono: as estações secundárias transmitem independentemente da consulta por parte da primária; este modo é geralmente empregado quando as estações secundárias se conectam à primária por canais exclusivos (por exemplo, linhas seriais); este modo não é empregado em redes de computadores;
3. modo de resposta assíncrono balanceado: utilizado em enlaces ponto-a-ponto envolvendo apenas estações combinadas; as estações gerenciam o enlace de acordo com um protocolo definido pelas camadas de enlace dos hosts comunicantes.

Finalmente, um enlace é governado por um protocolo composto de 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. reiniciação do enlace, onde um host toma a iniciativa de reinicializar o protocolo de transferência de quadros pelo enlace pela ocorrência de um erro irrecuperável.

3.2.5 Controle do Fluxo de Quadros

O controle de fluxo é necessário quando um receptor de quadros vê-se na impossibilidade momentânea de continuar a recepção. Várias são as razões para isso ocorrer: exaustão de buffers, ocorrência de erros internos de hardware ou software, atendimento de atividades de comunicação mais prioritárias, etc.

Em protocolos baseados no reconhecimento positivo ou negativo, o controle do fluxo é desnecessário, pois o emissor só envia o próximo quadro após o recebimento do reconhecimento do receptor. Caso o receptor deseje a suspensão temporária do envio de quadros, este simplesmente deixa de enviar os quadros de reconhecimento.

Em protocolos que empregam o reconhecimento contínuo, dois quadros de controle são empregados para o controle de fluxo:

- quadros RR (Receiver Ready): informa o emissor que o receptor está pronto para iniciar ou continuar a recepção de quadros;

- quadros RNR (Receiver Not Ready): informa o emissor que o receptor está impossibilitado temporariamente de receber quadros.

Os quadros RR e RNR visam aumentar a eficiência do canal compartilhado evitando a circulação de quadros de dados que com certeza serão descartados pelo receptor.

3.3 Protocolos de Enlace

3.3.1 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 LAPB (Link Access Procedure, Balanced)[3].

No HDLC, os quadros têm o formato apresentado na figura 3.9. 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

[3] O LAPB restringe o HDLC por permitir apenas enlace no modo de resposta assíncrono balanceado - ver subseção 3.2.4.

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 3.9: 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 3.10).

Bits →	1	3	1	3	
	0	N(S)	P/F	N(R)	INFORMAÇÃO

1	1	1	1	1	3	
1	0	S	S	P/F	N(R)	SUPERVISÃO

1	1	1	1	1		3		
1	1	M	M	P/F	M	M	M	NÃO NUMERADO

Figura 3.10: Formatos do campo de controle no protocolo HDLC.

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.

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.

3.3.2 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 3.1. O LLC especifica o formato do PDU que é parte dos dados do quadro. Este PDU, denominado LPDU, tem o formato dado pela figura 3.11.

Bytes ⟶	1	1	1 ou 2	>=0
	ENTIDADE DESTINO	ENTIDADE FONTE	CONTROLE	INFORMAÇÃO

Figura 3.11: 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

L-CONNECT.request(end_local, end_remoto, classe_de_serviço) L-CONNECT.indication(end_local, end_remoto, classe_de_serviço) L-CONNECT.response(end_local, end_remoto, classe_de_serviço) L-CONNECT.confirm(end_local, end_remoto, classe_de_serviço)
L-DISCONNECT.request(end_local, end_remoto) L-DISCONNECT.indication(end_local, end_remoto)
L-DATA.request(end_local, end_remoto, l_sdu) L-DATA.indication(end_local, end_remoto, l_sdu) L-EXPEDITED-DATA.request(end_local, end_remoto, l_sdu) L-EXPEDITED-DATA.indication(end_local, end_remoto, l_sdu)
L-RESET.request(end_local, end_remoto) L-RESET.indication(end_local, end_remoto) L-RESET.response(end_local, end_remoto) L-RESET.confirm(end_local, end_remoto)
L-ERROR-REPORT.indication(razão)
L-UNIT-DATA.request(end_local, end_remoto, l_sdu, classe_de_serviço) L-UNIT-DATA.indication(end_local, end_remoto, l_sdu, classe_de_serviço)

Tabela 3.1: Primitivas OSI de enlace. L-UNIT-DATA é utilizada para os serviços sem conexão; as demais para serviços com 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-CONNECT: estabelecimento de conexão (com confirmação);
- L-DISCONNECT: término de conexão (sem confirmação);
- L-DATA: envio de quadros de dados através da conexão (sem confirmação);
- L-RESET: reinicialização de conexão (com confirmação);
- L-ERROR-REPORT: informe de erros pelo provedor (indicação apenas);
- L-EXPEDITED-DATA: envio de dados urgentes através da conexão (sem confirmação);
- L-UNIT-DATA: envio de dados sem conexão (sem corfirmação).

A tabela 3.1 ilustra o formato destas primitivas.

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[4], 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.

3.4 Problemas

1. Descreva o funcionamento dos principais métodos de acesso ao meio para redes locais.

2. Descreva os campos que compõem o quadro dos padrões IEEE 802.3, 802.4 e 802.5.

3. Descreva os campos que compõem o *token* dos padrões IEEE 802.4 e 802.5.

4. Descreva a manutenção do anel no padrão IEEE 802.5.

5. Descreva as técnicas mais usuais para delimitação de quadros e em que situações são empregadas.

6. Por que protocolos de enlace baseados em caracter estão superados ?

7. Compute o *checksum* utilizando a técnica CRC para a seqüência de 12 bits 111001100010 e polinômio gerador $x^4 + x^2 + x + 1$.

8. Faça um diagrama temporal mostrando a transmissão de quadros com reconhecimento contínuo e largura de janela 3, utilizando quadros de controle tipo RR e RNR.

9. Descreva os quadros não numerados empregados pelo padrão X.25/camada 2.

10. Faça um diagrama temporal mostrando o estabelecimento de enlace, envio de quadros e término do enlace utilizando os quadros do protocolo HDLC.

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

Capítulo 4

A CAMADA DE REDE

A camada de rede é a terceira e última camada onde o fluxo de informação leva em conta as peculiariedades da subrede de comunicação, tais como a existência de IMPs, como os IMPs se interconectam, etc. A partir desta, as camadas acima empregam mecanismos de comunicação host a host, abstraindo a existência da subrede de comunicação. Para que a camada de transporte venha independer da topologia e tecnologia da subrede de comunicação, a camada de rede deve prover:

- entrega de pacotes: recolher um pacote do host emissor e entregá-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).

4.1 Entrega de Pacotes

Pacote é a unidade de informação que trafega entre os IMPs de uma rede de longa distância. Quando um host (DTE) necessitar transferir um pacote, este o entrega ao IMP ao qual ele se conecta fisicamente (via DCE). O IMP escolhe uma rota para o pacote tendo como destinatário o IMP ao qual o host de destino está conectado, para, a partir deste, ser entregue ao destinatário final (novamente, via DCE).

O tráfego entre os IMPs pode se dar através de uma conexão pré-estabelecida ou sem o estabelecimento prévio de conexão. No primeiro caso a camada de rede é dita *orientada a conexão*. Se a camada de rede não suporta circuitos virtuais (conexões), a mesma é dita *sem conexão* ou *orientada a datagrama*.

A camada de transporte acessa o serviço de entrega de pacotes através de NSAPs (Network Service Acess Points). NSAPs são endereços que identificam no nível de rede as entidades provedoras do serviço. O modelo OSI especifica um NSAP (figura 4.1) contendo valores decimais, não ocupando mais que 40 dígitos (20 bytes). O NSAP é dividido em três campos:

1. AFI (Authority and Format Indicator): indica o tipo de numeração presente no terceiro campo (número de telefone, endereço Internet, etc). Ocupa 1 byte codificando dois dígitos decimais com valores entre 10 e 99;

2. IDI (Initial Domain Identifier): especifica o domínio ao qual o terceiro campo pertence. Por exemplo, se o terceiro campo armazena o número de um telefone, o IDI pode conter o código do país. Este campo armazena um número variável de bytes, função do campo AFI;

3. DSP (Domain Specific Part): armazena o endereço do serviço propriamente dito. Contém um número variável de bytes, função dos campos AFI e IDI.

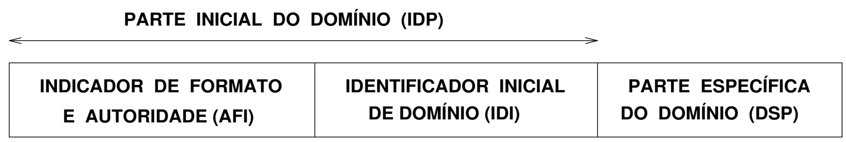

Figura 4.1: Formato de NSAP para redes OSI.

4.1.1 Camadas de Rede Orientadas a Conexão

No caso de redes locais, conexões no nível de camada de rede são similares às conexões no nível de camada de enlace: apenas dois hosts são envolvidos e apenas estes dela tomam parte. 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 (via DCE). O objetivo desta conexão é a entrega confiável de quadros mesmo que o DCE esteja fisicamente conectado ao IMP através de em canal sujeito a ruído.

Uma conexão no nível de camada de rede, por outro lado, requer a intermediação de múltiplos IMPs, pois a conexão se estende de host a host, via IMPs. No estabelecimento de uma conexão deste tipo, escolhe-se uma rota (vide próxima seção) e o tráfego de pacotes se dará 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.

4.1.2 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, estes estão sujeitos ao descarte quando um IMP não for capaz processá-los.

Cada pacote é roteado individualmente, independente dos demais. Neste caso, a garantia de entrega de pacotes na ordem em que foram emitidos não é verificada. Pacotes podem ser duplicados caso pacotes de reconhecimento (RR) da camada superior não atinjam o destino.

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 outas palavras, fica a cargo dos hosts (isto é, dos usuário 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 IMP (isto é, sobre a própria subrede de comunicação).

A tabela 4.1 ilustra as primitivas OSI ofereciadas pela camada de rede à camada de transporte para serviços sem e com conexão.

4.2 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.

N-UNITDATA.request(end_local, end_remoto, QoS, dados) N-UNITDATA.request(end_local, end_remoto, QoS, dados)
N-FACILITY.request(QoS) N-FACILITY.indication(end_remoto, QoS, motivo) N-REPORT.indication(end_remoto, QoS, motivo)
N-CONNECT.request(NSAP_local, NSAP_remoto, ACK, EXP, QoS, dados) N-CONNECT.indication(NSAP_local, NSAP_remoto, ACK, EXP, QoS, dados) N-CONNECT.response(NSAP_remoto, ACK, EXP, QoS, dados) N-CONNECT.confirm(NSAP_remoto, ACK, EXP, QoS, dados)
N-DISCONNECT.request(requerente, motivo, dados, end_resposta) N-DISCONNECT.indication(ent_solicitante, motivo, dados, end_resposta)
N-DATA.request(dados) N-DATA.indication(dados) N-DATA-ACKNOLEDGE.request() N-DATA-ACKNOLEDGE.indication() N-EXPEDITED-DATA.request(dados) N-EXPEDITED-DATA.indication(dados)
N-RESET.request(ent_solicitante, motivo) N-RESET.indication(ent_solicitante, motivo) N-RESET.response() N-RESET.confirm()

Tabela 4.1: Serviços ofereciados pela camada de rede: as primitivas N-UNITDATA, N-FACILITY e N-REPORT são empregadas em serviços sem conexão; as demais em serviços com conexão.

A última opção é análoga a redes de computadores. Os IMPs são depósitos de pacotes e somente o IMP diretamente conectado à subrede do host destinatário é que 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 obtenção 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 característica dinâmica.

A forma mais simples de rotear pacotes numa subrede é dotar cada IMP de uma *tabela de roteamento* contendo o melhor circuito ligando este IMP aos demais. Via de regra, a tabela armazena mais de um circuito entre dois IMPs, na eventualidade de uma rota de comunicação ou um IMP do circuito sair de serviço. Num roteamento dinâmico, a tabela de roteamento varia de acordo com as condições de tráfego da subrede.

Em redes de computadores, o roteamento dinâmico pode ser centralizado ou distribuído.

4.2.1 Roteamento Centralizado

No roteamento centralizado, os IMPs enviam periodicamente informações de tráfego a uma central de roteamento (NCC: Network Control Center). A central dispõe de informações sobre:

- os IMPs que estão ativos e os que estão inativos (por falha, manutenção, etc);
- as rotas de comunicação que estão ativas e as que estão inativas;
- o fluxo médio de pacotes nas rotas ativas;
- o tempo médio que um pacote fica armazenado num IMP;
- etc.

A partir destas informações o NCC, computa qual seria o roteamento ótimo ou sub-ótimo de pacotes para uma dada situação de tráfego. Computado o roteamento, o NCC distribui novas tabelas de roteamento aos IMPs.

O roteamento centralizado é viável apenas nos domínios da operadora da subrede de comunicação.

---

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

4.2.2 Roteamento Distribuído

O roteamento distribuído não utiliza um elemento centralizador como o NCC. A primeira conseqüência da distribuição é a indisponibilidade de uma visão global do tráfego na subrede. No roteamento distribuído, cada IMP é responsável pela atualização de sua tabela de roteamento. Em geral, esta atualização é fruto da troca de informações de tráfego entre IMPs diretamente conectados.

A tabela de roteamento dispõe de um índice para cada IMP da subrede e armazena informações de roteamento para os circuitos iniciando no IMP detentor da tabela. Cada posição na tabela armazena duas informações: o próximo host do circuito (diretamente conectado ao detentor da tabela), e uma estimativa do tempo que o pacote levará para atingir o destinatário final.

Estimativas de tempo são obtidas circulando periodicamente *pacotes de eco* entre IMPs diretamente conectados. O emissor de um pacote de eco marca no pacote o instante de tempo em que o pacote foi gerado e o transmite ao destinatário. O destinatário, ao recebe-lo, simplesmente o envia de volta ao emissor. O tempo de circulação de um pacote de eco é uma boa métrica das condições de tráfego entre dois IMPs (o tempo médio de tráfego no trecho é igual ao tempo de circulação do pacote dividido por dois).

Considere a subrede da figura 4.2. A subrede é representada por um grafo onde os nós representam os IMPs e os arcos rotas de comunicação entre IMPs. Suponha que ao transmitir um pacote de eco, o IMP envia também as medidas de tempo médio de tráfego para todos os demais IMPs a partir deste. Com isto, um IMP J dispõe do tempo médio de tráfego:

- entre J e I_j (todos os IMPs diretamente conectados ao IMP J);
- entre I_j e os demais IMPs.

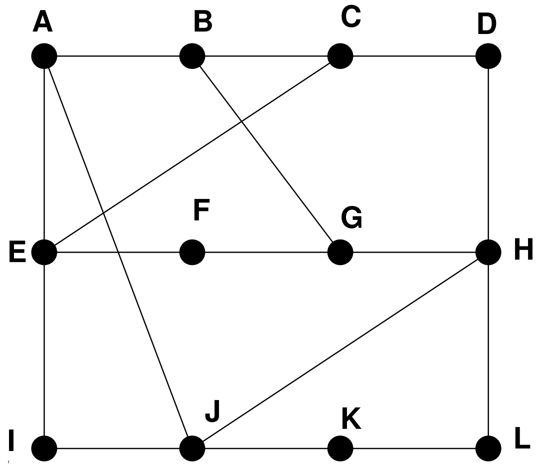

Figura 4.2: Uma subrede de comunicação.

As quatro primeiras colunas da tabela 4.2 consistem de vetores recebidos por J de seus vizinhos A, I, H, K. Na primeira coluna A informa J que o tempo médio para atingir B é 12

A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	-
K	24	22	22	0	6	K
L	29	33	9	9	15	K

Tabela 4.2: Quadros de eco recebidos pelo IMP J dos IMPs A, I, H e K (colunas 1-4) e sua tabela de roteamento (última coluna).

mseg, 25 mseg para C, e assim por diante. A última coluna da tabela contém a estimativa levantada pelo próprio IMP J.

De posse dos quatro vetores, J pode computar o tempo de tráfego mínimo entre ele e qualquer outro IMP (quinta coluna da tabela 4.2). A segunda coluna da tabela informa o IMP que deve ser enviado o pacote caso o destinatário final seja o IMP dado pela linha correspondente. Vajamos como esta tabela de roteamento é gerada. Os IMPs medem periodicamente o tempo de comunicação para com os vizinhos. Sempre que um pacote de eco é enviado, o receptor adiciona a ele as estimativas de tráfego de que dispõe (obtidas por ele ou enviadas em pacotes de eco pelos vizinhos). Como a subrede de comunicação forma um grafo conectado, pouco a pouco as tabelas vão se completando, até que todos os IMPs disponham de estimativas de tráfego a partir dos IMPs vizinhos para qualquer outro IMP da rede.

Por exemplo, qual o melhor roteamento de J para F ? Certamente este roteamento deve passar por A, I, K ou H. Passando por A, J dispõe do custo J-A (8 mseg) por medida própria e 23 mseg de A para F, informado por A no vetor recebido por J (coluna 1). Um tempo de 31 mseg é incorrido se o roteamento de J para F passar por A. Note que J desconhece como A irá rotear o pacote para F, sabendo apenas que o menor tempo para tal é 23 mseg. Analogamente, tempos de 30, 31 e 46 mseg são incorridos caso o roteamento se dê via I, H e K, respectivamente. Neste caso, sempre que necessitar rotear um pacote por F, J o faz via I com um custo de 30 mseg.

A tabela de roteamento de J é recomputada todas as vezes que J receber um novo vetor de seus vizinhos ou obter uma medida por iniciativa própria.

4.3 Controle de Congestionamento

Vimos no capítulo anterior que a camada de enlace é capaz de controlar o fluxo de quadros entre dois hosts comunicantes. Cabe a 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 (como é feito no IEEE 802.2 - ver Cap. 3).

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.

4.3.1 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 quanto for o tamanho da janela, 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, outro circuito pode ser escolhido ou a conexão é recusada.

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), reiniciada ou um sinal de *ocupado* é enviado ao IMP que emitiu o pacote, solicitando sua retransmissão mais tarde.

4.3.2 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 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 reiniciação de conexão causam o descarte de pacotes ainda fluindo pela conexão.

Seja um IMP com k buffers disponíveis para armazenamento de pacotes (buffers alocados à recepção não contam) e s linhas de saída. Como cada pacote recebido é roteado para uma dada linha de saída, pode-se dizer que os buffers armazenando pacotes são alocados às linhas de saída. Obviamente, uma única linha não pode monopolizar os k buffers do *pool*. Então, qual o número máximo de buffers, m, que devemos alocar por linha ? Estudos empíricos mostram um número adequado de buffers é dado pela relação $m = k/\sqrt{s}$.

A rede ARPANET utilizava esta estratégia de controle de congestionamento, mas alocando um número mínimo de buffers por linha de saída. Isto evita que linhas de alto tráfego mantenham permanentemente os buffers a ela alocados causando o descarte permanente de pacotes que fluem por linhas de pouco tráfego.

4.4 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 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 pode ocorrer nível de:

- 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.

A figura 4.3 ilustra estas possibilidades.

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

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.

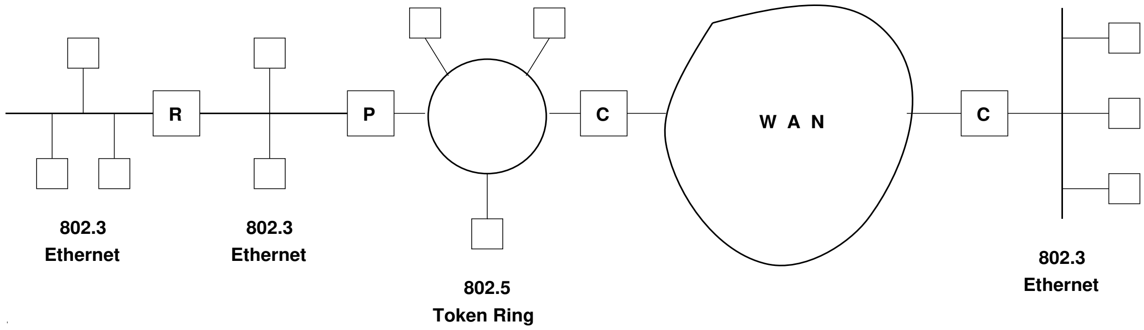

Figura 4.3: Exemplo de interconexão de redes. Retângulos com a letra R são repetidores; com a letra P pontes; e com a letra C comportas. Demais retângulos são hosts.

Pontes (Bridges)

Pontes 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).

Comportas (Gateways)

Comportas 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 uma comporta (formato dos pacotes e quadros, protocolos de enlace e acesso ao meio, e padrões da camada física requerem conversão). Comportas atuam nas três primeiras camadas do modelo OSI (ou nas duas primeiras da arquitetura TCP/IP).

Conversores de Protocolos

Conversores de protocolos 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 um conversor de protocolo.

4.4.1 Interconexão de Redes e o Modelo OSI

Interconexão de redes não teve a devida atenção por parte da ISO na elaboração do modelo OSI. No modelo OSI, interligação de redes é propiciada pela divisão da camada de rede em três subcamadas:

1. subcamada de acesso à subrede: trata do processamento dos protocolos das camadas de rede referentes às subredes sendo interconectadas;

2. subcamada de aprimoramento: compatibiliza os diferentes serviços oferecidos pelas subredes interconectadas;

3. subcamada inter-redes: processa os pacotes como se as redes interconectadas fossem homogêneas.

A figura 4.4 mostra a interconexão entre duas subredes utilizando a proposta do modelo OSI (subdivisão da camada de rede). Esta proposta nada mais é que estabelecer uma camada de rede padrão na subcamada inter-redes (OSI, obviamente), cabendo a subcamada de aprimoramento processar as devidas conversões de e para esta camada padrão. Como veremos, conversões causam invariavelmente degradação do desempenho e perda de funcionalidades.

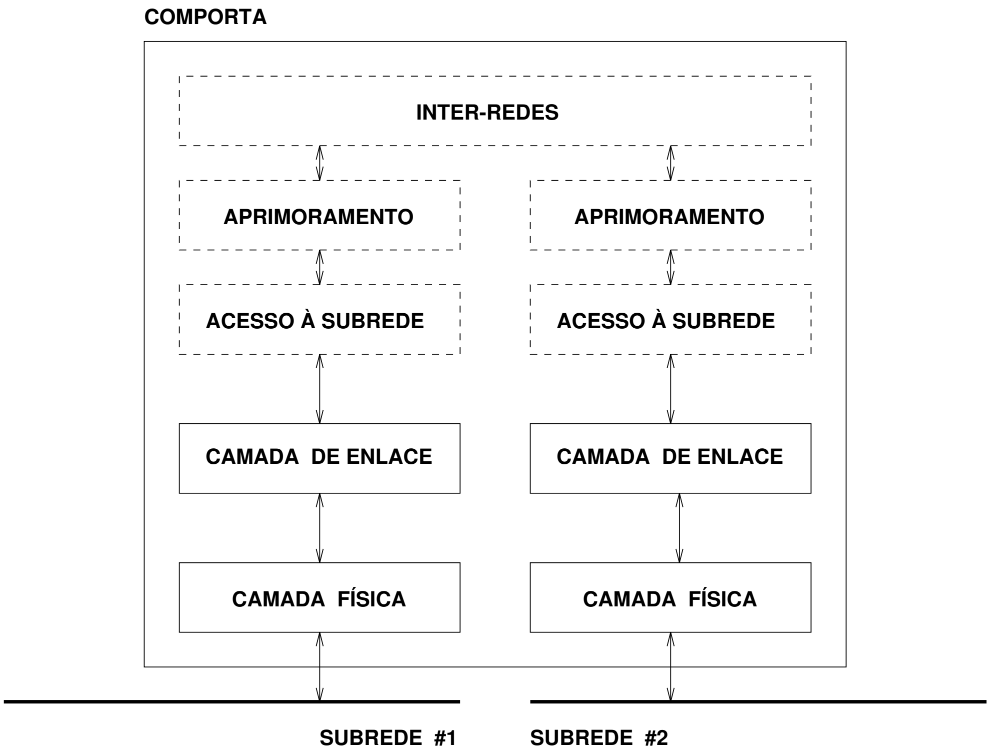

Figura 4.4: Proposta do modelo OSI para interconexão de redes.

Por exemplo, seja as subredes da figura 4.4 redes pública distintas. Quando um pacote enviado da subrede 1 se destinar a subrede 2, o roteamento obrigatoriamente passará pela comporta. A subcamada de acesso a subrede processará o pacote, fornecendo eventualmente um reconhecimento ao emissor (no protocolo da subrede 1). O pacote é então passado à subcamada de aprimoramento, que o converte num formato dado pelo padrão da subcamada inter-redes. Na subcamada inter-redes, o pacote é processado como se o mesmo circulasse por uma subrede homogênea. Feito este processamento, o pacote é passado para a subcamada de aprimoramento da subrede 2, onde será convertido para o formato requerido por esta subrede e encaminhado a subcamada de acesso a subrede. Esta, completa a transmissão do pacote, recebendo eventualmente o reconhecimento do destinatário (no protocolo da subrede 2).

4.4.2 Pontes

Dado que a camada de rede praticamente inexiste em redes locais, pontes são comumente empregadas para interconectá-las. Neste caso é imprescindível que as subredes interconectadas operem com o mesmo protocolo nas camadas de 4, 5, 6 e 7 (caso contrário, conversores de protocolos devem ser empregados).

A figura 4.5 ilustra uma ponte interconectando duas subredes da família IEEE 802 (802.3 e 802.5). Estas subredes utilizam o mesmo protocolo de enlace, mas formatos de quadros, disciplina de acesso ao meio e camada física distintos. O mesmo conceito de interconexão de redes proposto pelo modelo OSI é empregado, exceto que, neste caso, a conversão se dá no nível quadros (não de pacotes).

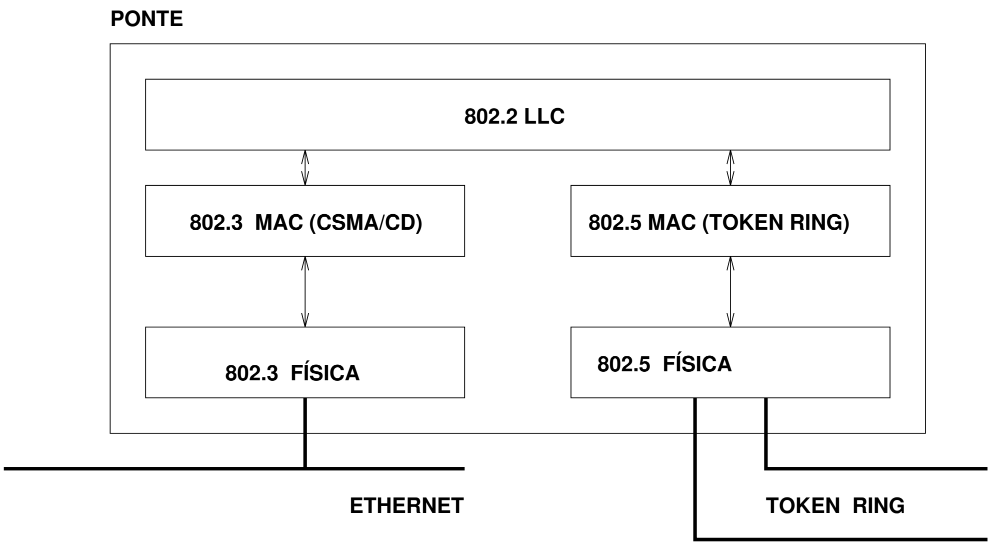

Figura 4.5: Ponte interconectando subredes IEEE 802.3 e 802.5.

A ponte converte o formato de pacote e o protocolo de enlace (quando for o caso) de uma rede para outra. A ponte dispõe de tantas camadas físicas e subcamadas de acesso ao meio quantas forem as subredes por ela conectadas. A subcamada de enlace lógico é quem perfaz o elo de ligação entre as subredes. Via de regra, pontes são hosts não exclusivamente dedicados à esta tarefa.

Infelizmente, muitos problemas decorrem da interligação de subredes. Vamos exemplificar alguns para o caso da interconexão de subredes IEEE 802.3 (CSMA-CD) e 802.5 (Token Ring).

Tamanho dos quadros

Quadros no padrão 802.3 têm um comprimento máximo da ordem de 1500 bytes enquando no padrão 802.5 é limitado pela taxa de transmissão e pelo tempo máximo de posse do *token*, tipicamente 5000 bytes. O que ocorre quando uma subrede 802.5 envia um quadro de, por exemplo, 3000 bytes para a subrede 802.3? Como o protocolo LLC impõe a indivisibilidade do quadro, tal quadro simplesmente é descartado pela ponte.

Taxa de Transmissão

Suponha a subrede 802.3 operando a 10 Mbits/seg e a subrede 802.5 operando a 4 Mbits/seg. Se um host na 802.3 enviar uma rajada de quadros para outro na subrede 802.5, a ponte irá receber quadros numa taxa mais alta que é capaz de escoar. Como consequência, pode ocorrer uma exaustão de buffers na ponte acarretando a perda de quadros.

Perda de Funcionalidades

O padrão 802.5 suporta priorização no envio de quadros e reserva do meio. A transmissão de um quadro de altíssima prioridade de uma rede 802.5 para outra 802.3 não garante sua pronta recepção. Ocorre que a ponte, ao receber e converter o quadro necessita executar o procedimento CSMA-CD para acessar o meio no lado 802.3. Este procedimento não suporta prioridade, sendo o tempo de acesso função do nível de tráfego na subrede.

Atraso na Comunicação

Pontes invariavelmente impõem um atraso na comunicação entre dois hosts. A ponte adiciona ao tempo de comunicação normal dois procedimentos de acesso ao meio, um cômputo do *checksum* e a conversão do pacote de um formato para outro (desprezando-se o tempo extra na propagação do sinal). Este atraso na comunicação pode causar a ocorrência de *timeouts* no lado emissor gerando retransmissões desnecessárias (que contribuem para mais atrasos !).

Perda de Confiabilidade

Ao receber um pacote da subrede 802.5, a ponte computa o *checksum*, e, se livre de erros, o converte e retransmite para a subrede 802.3, enviando imediatamente um reconhecimento positivo ao emissor no lado 802.5. Este reconhecimento é "mentiroso" pois nada garante que o receptor na subrede 802.3 recebeu o quadro livre de erros. Isto causa perda de confiabilidade nos protocolos de transporte onde a garantia de entrega de pacotes é amarrada à garantia de entrega de quadros na camada de enlace (como o protocolo UDP, por exemplo).

4.4.3 Comportas

Comportas conectam subredes no nível da camada 3 sendo mais empregadas na interconexão de redes públicas. O emprego de uma única comporta na interconexão de duas redes públicas gera conflitos entre as operadoras, pois aquela que operar a comporta tem influência na operação da outra rede. A solução é se empregar duas comportas, uma de cada lado, e interconectá-las por uma via de comunicação.

Comportas podem ser orientadas a conexão ou orientadas a datagrama, dependendo da natureza da camada 3 da rede.

Comportas Orientada a Conexão

Se ambas as subredes interconectadas são baseadas em conexão, as comportas tomam parte do circuito virtual (conexão), como mostrado na figura 4.6. Neste caso, diz-se que as comportas são orientadas a conexão. Este tipo de comporta aloca os recursos requeridos pela conexão, antes de estabelece-la, garantindo uma entrega confiável dos pacotes que fluem de uma subrede para outra. Como desvantagem temos:

- se uma comporta falhar, a conexão e os pacotes em trânsito estarão comprometidos;
- inflexibilidade no roteamento, pois uma vez estabelecida uma conexão, o fluxo de pacotes associados a esta conexão permanece fixo.

Comportas orientadas a conexão se comunicam através do protocolo CCITT X.75 (quase idêntico ao X.25).

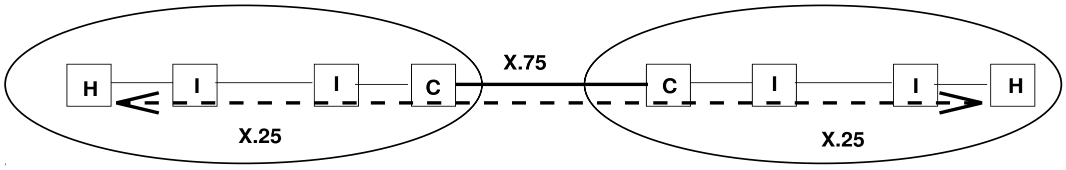

Figura 4.6: Interligação de duas subredes via comportas orientadas a conexão (C: comporta; H: host; I: IMP). A linha pontilhada indica uma conexão.

Comportas Orientadas a Datagrama

Comportas orientadas a datagrama são empregadas quando uma ou ambas as subredes interconectadas não suportam circuitos virtuais no nível da camada de rede. Neste caso, os pacotes que compõem uma dada mensagem podem fluir por rotas diferentes (figura 4.7). Esta propriedade facilita o roteamento, adaptando-o prontamente às condições de tráfego na interligação.

Quando comportas orientadas a datagrama são empregadas, pacotes podem se perder (por falta de buffer numa comporta, por exemplo), ou serem recebidos fora de ordem (caso fluam por diferentes comportas, por exemplo). Cabe a camada de transporte nos hosts comunicantes tratar estas situações.

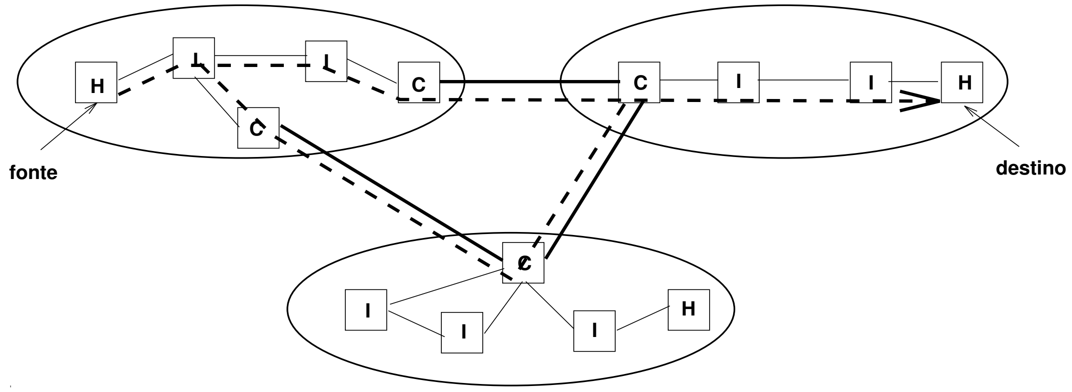

Figura 4.7: Interligação de duas subredes via comportas orientadas a datagrama. As linhas tracejadas indicam rotas de datagramas.

4.5 O Protocolo X.25/Camada 3

O protocolo X.25 para a camada de rede difere dos correspondentes X.25 para as camadas 1 e 2 no que tange ao estabelecimento das conexões. No X.25/camada 1 a conexão DTE-DCE é física (elétrica) e estabelecida permanentemente. No X.25/camada 2, uma conexão involve apenas dois computadores: o host (DTE) e o IMP a ele conectado (via DCE). Aqui, o DTE deve iniciar a conexão com o IMP utilizando uma via física permanente ou temporária (linha discada, por exemplo). Para o X.25/camada 3, a conexão une dois DTEs e se estende por pelo menos 1 (tipicamente vários) IMPs. Diferente dos anteriores, o estabelecimento de uma conexão irá envolver múltiplas entidades (DTEs e IMPs).

O protocolo X.25 para a camada de rede define um endereçamento para os hosts segundo a recomendação X.121 da CCITT. Este endereçamento é diferente dos NSAPs padrão OSI, e composto de 14 dígitos decimais. Os primeiros 3 dígitos identificam o país (exemplo: 302 a 307 para o Canadá), o quarto dígito a rede pública naquele país (totalizando $6 \times 10 = 60$ para o Canadá), e os últimos 10 dígitos o IMP da rede.

O X.25/camada 3 utiliza um protocolo denominado PLP (Packet Layer Protocol). Este protocolo define 3 tipos de pacotes: um tipo para a requisição de conexões, um tipo para controle, e um para dados. A figura 4.8 ilustra o formato de um pacote utilizado no estabelecimento de conexões.

Neste pacote, os campos GRUPO e CANAL ocupam 12 bytes e armazenam o identificador do circuto virtual sendo estabelecido. A seguir vem os endereços (formato X.121) dos IMPs de origem e destino. O campo RECURSOS é utilizado para requerer alguns serviços providos pela rede, por exemplo, tipo de circuito (full-duplex, simplex, etc), tamanho da janela (controle de fluxo), velocidade de comunicação (1200 bps, 2400 bps, etc), taxação no IMP destino (chamada a cobrar), etc. Finalmente, o DTE que iniciou a conexão pode dispor de, no máximo, 16 bytes no campo DADOS-DO-USUÁRIO. Este campo é interpretado pelo DTE sendo conectado e ignorado pela rede.

Um pacote de controle (figura 4.9) possui campos para identificar a conexão GRUPO-

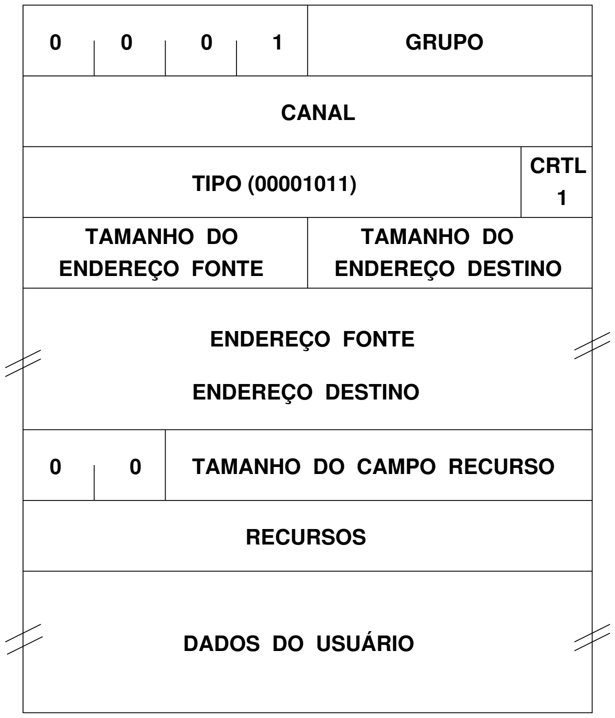

Figura 4.8: Pacote X.25 utilizado no estabelecimento de conexões.

CANAL e ação de controle sendo tomada. Um ou dois bytes de informação adicional acompanha o cabeçalho e indica tipicamente o porque da ação de controle (por exemplo, a recusa do estabelecimento de uma conexão).

As ações de controle possíveis são:

- CALL_ACCEPT: indica o aceite de pedido de conexão;
- CLEAR_REQUEST: indica o término de conexão (ou o não aceite);
- CLEAR_CONFIRMATION: confirma o término de conexão;
- RECEIVER_READY, RECEIVER_NOT_READY, REJECT: utilizados no controle do fluxo do pacotes de dados;
- INTERRUPT: envio de, no máximo, 32 bytes fora de seqüência (com alta prioridade);
- INTERRUPT_CONFIRMATION: reconhece um INTERRUPT;
- RESET_REQUEST: reinicia uma conexão (zerando o número de sequência dos pacotes de dados);
- RESET_CONFIRMATION: reconhece um RESET_REQUEST;

Figura 4.9: Pacote X.25 de controle.

- RESTART_REQUEST: reinicia todas as conexões que este DTE participa (emitido após a volta de um *crash*);
- RESTART_CONFIRMATION: reconhece um RESTART_REQUEST;
- DIAGNOSTIC: informa o DTE de problemas ocorridos na subrede de comunicação (por exemplo, campo TIPO inválido).

Pacotes de dados apresentam o formato da figura 4.10.

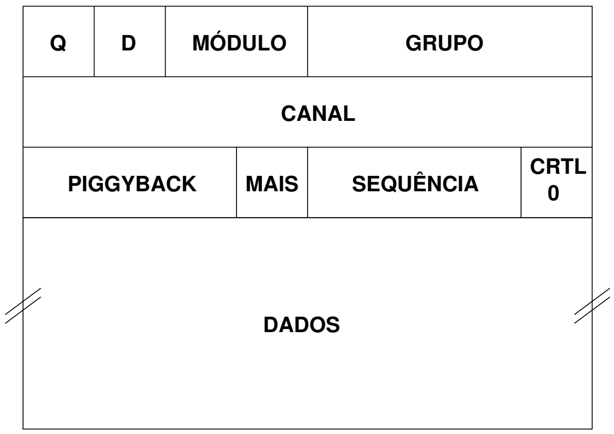

Figura 4.10: Pacote X.25 de dados.

O bit Q indica *dados qualificados* e é utilizado pela camada de transporte para distinguir seus PDUs de dados dos de controle. O bit D estipula o significado do campo *piggyback*. Se D for igual a 1, *piggyback* indica um reconhecimento por parte do host (DTE) receptor. Caso contrário (Q igual a zero), o reconhecimento da entrega é oriundo do DCE local. O campo módulo pode assumir o valor 01 para numeração em módulo 8, ou 10 para numeração em módulo 128.

Piggyback é um reconhecimento sem a utilização de pacotes de controle. Se o host receptor necessitar enviar um pacote da dados durante a recepção de uma seqüência de pacotes, este, voluntariamente, informa o último pacote da sequência recebido sem erros (permitindo o outro lado avançar a janela). Isto minimiza a necessidade de pacotes de reconhecimento quando a comunicação é intensa nos dois sentidos.

O campo MAIS indica que mais pacotes pertencentes a este grupo estão por vir. Somente quando MAIS for zero, a mensagem é entregue ao receptor. O campo SEQÜÊNCIA numera os pacotes (em módulo 8 ou 128 conforme o campo Q).

4.6 Problemas

1. Quais os serviços providos pela camada de rede ?

2. Quais as vantagens e desvantagens das camadas de rede orientadas a conexão e sem conexão ?

3. Descreva o algoritmo do caminho mais curto para roteamento de pacotes em redes de computadores.

4. Quais as técnicas mais empregadas para controle de congestinamento em subredes de comunicação ?

5. Quais os dispositivos existentes para a interconexão de redes e em que situações são empregados ?

6. Cite algumas perdas de funcionalidades causadas pela interconexão de redes 802.3 (CSMA-CD) e 802.4 (TOKEN BUS).

7. Como o modelo OSI trata a interconexão de redes ?

8. Como é estabelecida uma conexão no nível da camada de rede no protocolo X.25 ?

9. Quais os pacotes de controle empregados no X.25/Camada 3 ? Faça um diagrama temporal mostrando os pacotes emitidos pelas duas entidades.

10. O que é *piggybacking* ?

Capítulo 5

A CAMADA DE TRANSPORTE

A camada de transporte é a primeira camada do modelo OSI 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 para 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 (TPDUs) entre os hosts comunicantes e as quebras de conexões de rede (N-RESETs).

5.1 Qualidade do Serviço

A camada de transporte define 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. Os principais parâmetros são:

- vazão (bytes/s): taxa mínima 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;
- *jitter* (milisegundos): variação máxima do atraso de propagação;
- tempo de estabelecimento da conexão (milisegundos): máximo intervalo de tempo entre a entidade usuária da camada de transporte solicitar uma conexão e a sua confirmação;
- probabilidade de falha no estabelecimento da conexão: probabilidade da conexão não ser estabelecida no tempo estipulado (devido a falta de buffers nos hosts, congestionamento na subrede de comunicação, etc);
- 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;
- probabilidade de falha no término de conexão: probabilidade do encerramento de uma conexão não se completar no tempo de encerramento estipulado;
- proteção: estipula a privacidade dos dados fluindo pela conexão de transporte quanto a sua interceptação por terceiros;
- prioridade: estipula a prioridade desta conexão em relação às demais (a prioridade da transmissão de TPDUs é função da prioridade das conexões por onde os TPDUs fluem);
- 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 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.

5.2 O Serviço de Transporte

O serviço de transporte pode ser orientado a conexão ou a datagrama (sem conexão). Em redes locais, dada a alta confiabiliade da camada de rede, o serviço sem conexão é atrativo para muitas aplicações, dado 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.

O modelo OSI especifica apenas quatro primitivas para transporte orientado a conexão e uma para transporte orientado a datagrama (tabela 5.1). Note a ausência de primitiva do tipo RESET: uma conexão de transporte é muito mais confiável que uma conexão de rede (camada 3).

As primitivas T-CONNECT, T-DISCONNECT e T-DATA estabelecem, terminam e transmitem TPDUs por uma conexão. São equivalentes àquelas da camada de rede.

A primitiva T-EXPEDITED-DATA transmite *dados expressos*: TPDUs com alta prioridade que são entregues à aplicação fora de seqüência. São empregados para transmitir informações urgentes como um caracter de controle (BREAK, DELETE, etc) teclado pelo usuário numa sessão de *login* remoto [1].

[1] Imagine um programa em loop que supostamente deveria ler caracteres já digitados pelo usuário. O que aconteceria se o caracter BREAK, teclado pelo usuário para abortar o programa, for entregue em seqüência?

T-CONNECT.request(TSAP_origem, TSAP_destino, exp_req, QoS, dados)
T-CONNECT.indication(TSAP_origem, TSAP_destino, exp_req, QoS, dados)
T-CONNECT.response(QoS, TSAP_destino, exp_req, dados)
T-CONNECT.confirm(QoS, TSAP_destino, exp_req, dados)
T-DISCONNECT.request(dados)
T-DISCONNECT.indication(justificativa, dados)
T-DATA.request(dados)
T-DATA.indication(dados)
T-EXPEDITED-DATA.request(dados)
T-EXPEDITED-DATA.indication(dados)
T-UNITDATA.request(TSAP_origem, TSAP_destino, QoS, dados)
T-UNITDATA.indication(TSAP_origem, TSAP_destino, QoS, dados)

Tabela 5.1: Serviços ofereciados pela camada de transporte: as primitivas T-CONNECT, T-DISCONNECT e T-DATA são empregadas em serviços com conexão; T_UNITDATA em serviços sem conexão. *exp_req* é um flag que indica se dados expressos serão enviados pela conexão ou não. QoS estabelece a qualidade de serviço proposta ou aceita.

A primitiva T-UNIDATA transmite um TPDU sem o estabelecimento de conexão e sem garantia de entrega, seqüência ou ausência de duplicação.

A figura 5.1 mostra seqüências típicas de ocorrência destas primitivas.

A camada de transporte provê serviços nos TSAPs (Transport Service Access Points). Na camada de rede um NSAP identificava um host. Na camada de transporte, um TSAP identifica um *port* de comunicação mantido por um processo (em host conhecido). A entidade usuária da camada de transporte normalmente indica o host e o port aos quais a mensagem se destina. A camada de transporte manipula o port, passando o host (no formato adequado de NSAPs) à camada de rede.

Via de regra, um TSAP é uma estrutura muito mais simples que um NSAP: tipicamente um inteiro ocupando 2 bytes que identifica o port. Um TSAP pode ser entendido como um identificador de uma fila para onde mensagens são enviadas (ou de onde mensagens são recebidas). A forma como um processo solicita este recurso depende da interface de acesso aos serviços de transporte provida pelo sistema operacional.

O estado em que um TSAP pode se encontrar (e o que causa as transições de estado) é apresentado no diagrama da figura 5.2.

5.3 Classes de Protocolos de Transporte

Os protocolos de transporte são classificados 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 e de N-RESETs. Apenas as LANs se aproximam desta classe.

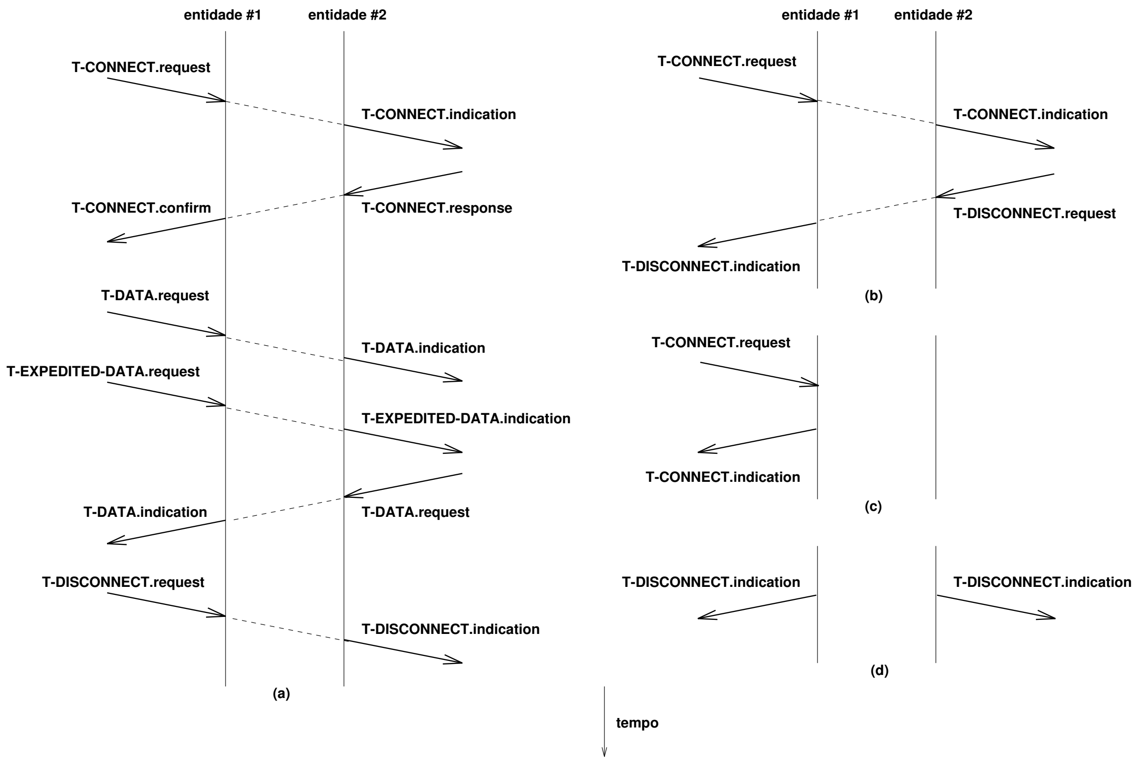

Figura 5.1: Algumas seqüências de ocorrência das primitivas OSI: (a) seqüência normal de transferência de dados; (b) conexão recusada pelo host remoto; (c) conexão recusada pela camada de transporte; (d) término de conexão requisitado pela camada de transporte.

Tipo B : Transporte confiável de pacotes, mas com a ocorrência de N-RESETs (quebras de conexões). Redes públicas X.25 pertencem a esta classe.

Tipo C : Transporte de pacotes não confiável. Redes com protocolo IP (Internet Protocol) como a Internet, por exemplo, 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 transprte de dados recai sobre a subrede de comunicação.

Classe 1 : Para subredes tipo B, provêem recuperação básica de erros (principalmente N-RESETs). 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.

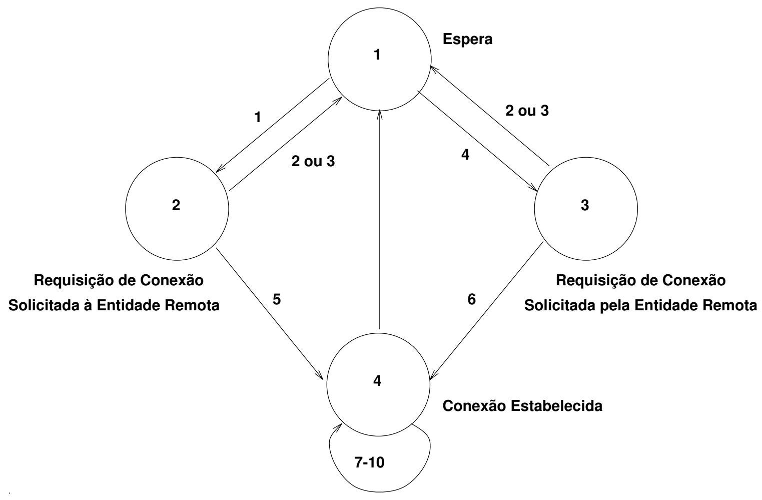

Figura 5.2: Estados de um TSAP. Os eventos que causam transição de estado são: (1) T-CONNECT.request gerado pela camada usuária; (2) T-DISCONNECT.indication recebido da camada de transporte; (3) T-DISCONNECT.request gerado pela camada usuária; (4) T-CONNECT.indication recebido da camada de transporte; (5) T-CONNECT.confirm recebido da camada de transporte; (6) T-CONNECT.response gerado pela camada usuária; (7) T-DATA.request gerado pela camada usuária; (8) T-DATA.indication recebido da camada de transporte; (9) T-EXPEDITED-DATA.request gerado pela camada usuária; (10) T-EXPEDITED-DATA.indication recebida da camada de transporte.

Classe 4 : Os únicos a operarem em subredes tipo C, provêem detecção e recuperação de erros não tratados pela subrede de comunicação (usualmente baseadas em datagrama). São os protocolos de transporte mais complexos.

5.4 Protocolos de Transporte Classe 4

Protocolos classe 4 operam em subredes tipo C que oferem serviços de datagramas sem garantia de entrega, 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 classe 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 fechamento de conexões.

Vamos examinar as soluções comumente empregadas nos protocolos classe 4 para estas três situações.

5.4.1 Estabelecimento de Conexões

Dois cenários devem ser gerenciados por um protocolo de transporte classe 4: duplicação de um TPDU que carrega uma requisição de conexão (gerado pela chamada *T-CONNECT.request*); e perda de um TPDU que carrega a confirmação do estabelecimento da conexão (que iria gerar um *T-CONNECT.confirm*).

No primeiro caso, ao receber um TPDU duplicado, o destinatário irá supor que uma segunda conexão foi solicitada. 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 evita conexões inúteis geradas por duplicação ou perda de TPDUs.

Ao enviar um pedido de conexão, o host solicitante informa um número inicial de seqüência na numeração dos TPDUs de dados que ele enviar pela conexão: x (primeira fase). Ao receber a solicitação, o destinatário reconhece x e informa que seu número inicial de seqüência será y (segunda fase). Ao receber a confirmação da conexão, o host que a iniciou reconhece y num TPDU de reconhecimento ou no primeiro TPDU de dados que enviar (terceira fase). Este método de estabelecimento de conexões (figura 5.3) é imune a duplicação ou 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 *timeout*).

5.4.2 Transferência de TPDUs de Dados

Perdas, duplicações e inversões de ordem são tratadas de forma adequada por protocolos de controle de fluxo com reconhecimento e *piggybacking*. Um problema adicional nos protocolos de transporte classe 4 reside na alocação de buffers. TPDUs transmitidos devem permanecer armazenados no emissor até a chegada de reconhecimento por parte do receptor. O receptor necessita também armazenar TPDUs até que se complete uma janela (no caso de reconhecimento contínuo) ou uma mensagem (composta de vários TPDUs).

Via de regra, os esquemas de *bufferização* empregam um *pool* de buffers compartilhados por todas as conexões ou um buffer de grandes dimensões para cada conexão. Neste ultimo caso, o buffer único abriga vários TPDUs e sua manutenção lembra gerenciamento de memória em sistemas operacionais.

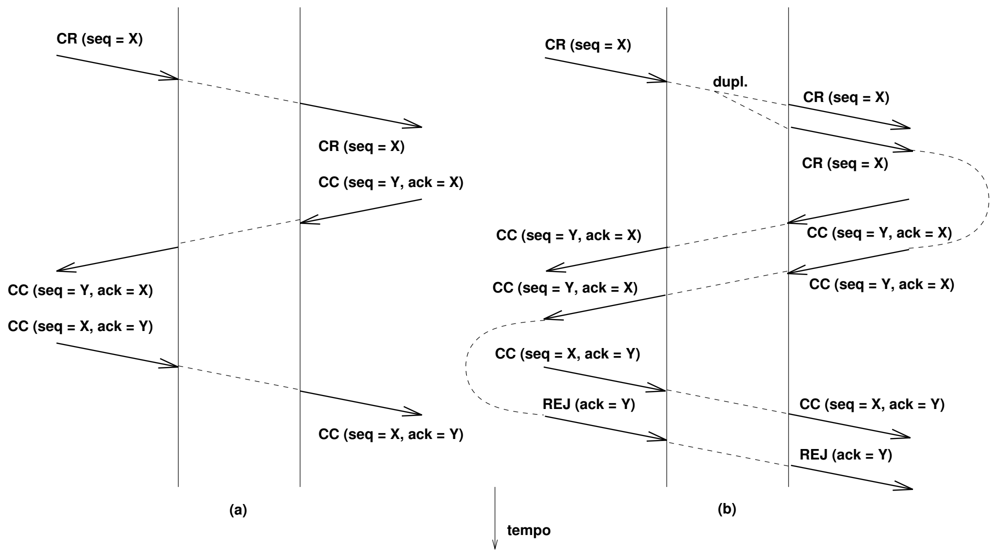

Figura 5.3: Estabelecimento de conexão em 3 fases: (a) situação normal; (b) duplicação de TPDU de requisição.

Tanto no emprego de um *pool* de buffers quanto no emprego de um buffer único fica a questão: qual o tamanho dos buffers? Esta questão é importante pois o tamanho dos TPDUs varia de poucos bytes como numa sessão de *login* remoto, à milhares de bytes como em transferência de arquivos.

No caso de *pool* de buffers, pode-se empregar buffers de tamanho variado. Isto causa um *overhead* adicional na procura de um buffer no *pool* (de tamanho adequado ao TPDU que o buffer irá abrigar).

No caso de se empregar buffer único, pode-se negociar um tamanho inicial de buffer (ao invés do tamanho da janela) no estabelecimento da conexão. O host que está recebendo TPDUs de dados informa nos TPDUs de reconhecimento quantos bytes existem disponíveis no buffer. O host emitindo os TPDUs ajusta sua taxa de transmissão em função dos TPDUs que tem para transmitir e da quantidade de espaço que o receptor tem para armazená-los. Note que o tamanho do buffer único no receptor (e no transmissor, caso esteja usando o mesmo esquema de *bufferização*) pode permanecer constante, aumentar ou diminuir de tamanho, em função da memória disponível à camada de transporte como um todo.

5.4.3 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 chega após a conexão pela qual fluia ter sido 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.

O 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. A figura 5.4 ilustra este procedimento.

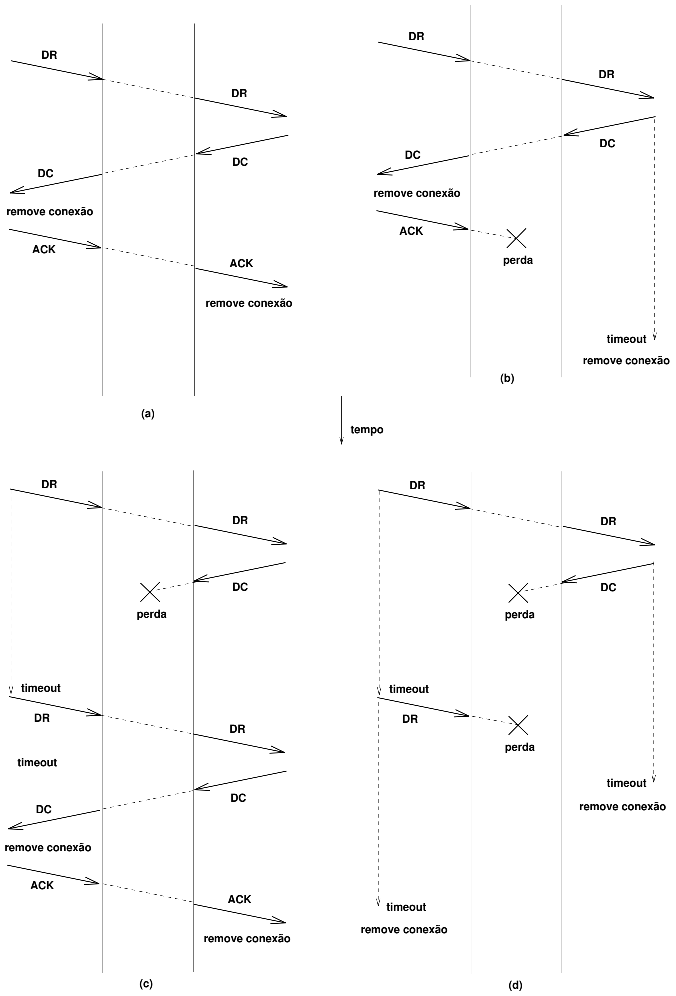

Figura 5.4: Encerramento de conexão em 3 fases: (a) situação Normal; (b) falha na terceira fase; (c) falha na segunda fase; (d) falhas na segunda e terceira fase.

Este procedimento garante também que TPDUs "atrasados" não serão descartados. Simplesmente, um host não aceita o encerramento da conexão (emitindo um REJECT na terceira fase) se um ou mais TPDU de dados ainda não teve seu recebimento confirmado.

Encerramento de conexão em três fases não garante 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.

5.5 Multiplexação

Multiplexação a nível de camada de transporte é a capacidade de:

1. operar múltiplas conexões de transporte através de uma única conexão de rede (multiplexação *upward*); ou

2. uma única conexão de transporte utilizar múltiplas conexões de rede (multiplexação *downward*).

O primeiro caso é comum em redes públicas onde a taxação é função do número de conexões requisitadas à subrede de comunicação. Obviamente a qualidade do serviço é pobre em conexões de transporte multiplexadas numa mesma conexão de rede. Protocolos de transporte classe 2 em diante são dotados de capacidade de multiplexação *upward*.

O segundo caso (multiplexação *downward*) ocorre quando uma conexão de transporte necessita determinada vazão não assegurada pela conexão de rede. Neste caso, distribui-se os TPDUs por várias conexões de rede, aumentando a vazão da conexão de transporte.

5.6 Protocolos de Transporte OSI Orientados a Conexão

Os protocolos de transporte OSI é um padrão ISO (8073) tipicamente em conjunção com o X.25. Este protocolo define 10 classes de TPDUs. Os formatos são os mesmos para as 5 classes de protocolos, exceto que certas classes não necessitam de todos os TPDUs.

Os três primeiros campos dos TPDUs são comuns. O primeiro, LI (Length Indicator), ocupa 1 byte e contém o tamanho do cabeçalho (partes fixa e variável). Segue o segundo campo, também de 1 byte, contendo nos primeiros 4 bits o tipo de TPDU e nos 4 bits restantes o tamanho da janela (Cdt: crédito), quando for o caso. O terceiro campo ocupa 2 bytes e contém um identificador da conexão. Uma conexão é identificada por dois inteiros denominados referência destino (estabelecido por quem aceita a conexão) e referência fonte (estabelecido por quem inicia a conexão).

Os demais campos dependem do tipo de TPDU. Os 10 tipos são descritos a seguir e apresentados na figura 5.5.

CR (Connection Request): inicia uma conexão de transporte. O campo CLASSE propõe uma classe de protocolo para a conversação (geralmente a classe 4 é empregada). A parte variável consiste de:

1. os TSAPs de origem e destino;
2. o tamanho máximo dos TPDUs (de 128 a 8192, em potências de 2);
3. a versão do protocolo;
4. uma chave de proteção (para fins de criptografia, direito de acesso ao serviço, etc);
5. um indicador de uso de *checksum*;
6. parâmetros de qualidade do serviço.

Este TPDU pode carregar até 32 bytes de dados do usuário (entregue à entidade que mantém o TSAP destino).

CC (Connection Confirm): Provê confirmação positiva ou negativa de uma requisição de conexão. Possui formato idêntico ao CC, carregando na parte variável a contraproposta referente aos parâmetros de qualidade de serviço contidos na requisição.

DR (Disconnect Request): solicita o encerramento da conexão. O campo MOTIVO provê um indicativo do motivo de tal requisição. A parte variável, quando for o caso, carrega informações complementares ao campo MOTIVO.

DC (Disconnect Confirm): confirma o término da conexão. A parte variável carrega apenas o *checksum*, se sua utilização foi decidida no estabelecimento da conexão.

DT (Data): transporta dados. O campo EOT (End of Text) ocupa 1 bit e indica se este TPDU é o último de uma mensagem ou não (em sendo o último, a mensagem pode ser entregue à entidade receptora). Completando o byte, o campo TPDU-N, de 7 bits, provê o número de seqüência do TPDU em módulo 7.

ED (Expedited Data): carrega dados expressos. A parte variável contém o *checksum*, quando for o caso.

AK (Acknowledgement): reconhece TPDUs de dados (DT ou ED). O campo Cdt pode alterar a janela previamente em uso para fins de controle de fluxo. A parte variável contém o *checksum*, quando for o caso.

RJ (Reject): utilizado nos protocolos de classe 1 e 3, solicita a transmissão de todos os TPDUs a partir e inclusive do número de seqüência constante no campo TPDU-ESPERADO.

ER (Error Report): informa a ocorrência de erros tais como parâmetros desconhecidos, tipo de TPDU inválido, etc. O campo MOTIVO informa o erro detectado. Caso necessário, maiores informações são supridas na parte variável.

5.7 Protocolos de Transporte OSI sem Conexão

O protocolo de transporte OSI sem conexão define um único tipo de TPDU utilizado no transporte de dados (figura 5.5). A parte variável do cabeçalho contém os TSAPs de origem e destino, e, opcionalmente, o *checksum*.

Cada TPDU é transmitido num pacote separado, sem confirmação de recebimento. Não há garantia de entrega, ordem, ou ausência de duplicação. Protocolos de transporte sem conexão é utilizado principalmente em redes classes A (locais).

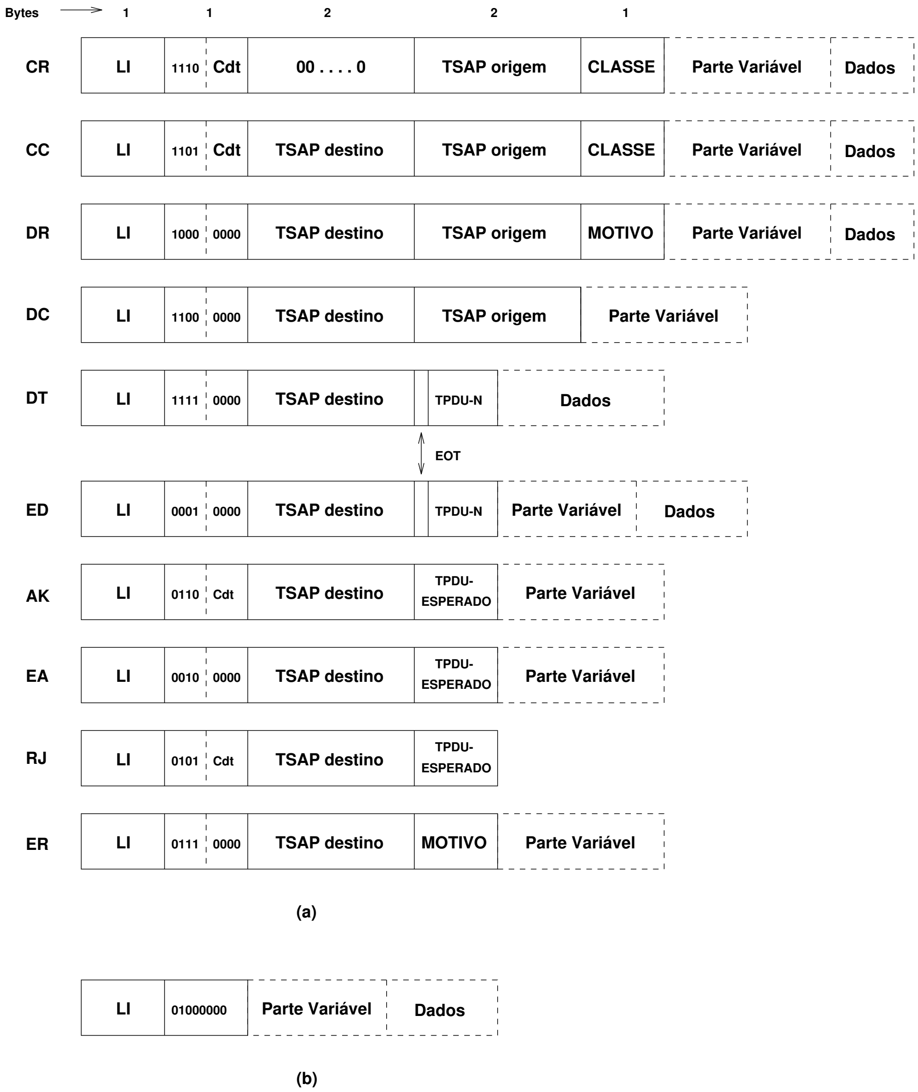

Figura 5.5: PDUs de transporte definidos pela ISO: (a) transporte orientado a conexão; (b) transporte orientado a datagrama.

5.8 Problemas

1. Quais as funções precípuas da camada de transporte ?
2. O que é "qualidade de serviço" e sob que parâmetros é determinada ?
3. Qual a finalidade de se definir classes de protocolos de transporte ?
4. Descreva como conexões são estabelecidas em protocolos de transporte classe 4.
5. Como se dá o controle de fluxo em protocolos de transporte classe 4 ?
6. Quais as diferenças no procedimento de três fases para se abrir e fechar conexões ?
7. Defina multiplexação *upward* e *downward*.
8. Mapeie as primitivas OSI nos PDUs de transporte definidos pela ISO.
9. Defina um algoritmo a ser utilizado num protocolo de transporte para tratar os N-RESETs produzidos pela camada de rede.
10. É razoável se empregar um protocolo de transporte sem conexão em subredes de comunicação orientadas à conexão (X.25, por exemplo) ? Justifique.

Capítulo 6

A CAMADA DE SESSÃO

A camada de sessão tem por finalidade organizar a troca de informação (diálogo) entre dois hosts comunicantes. Esta camada provê mecanismos para:

- estabelecer e terminar sessões de diálogo;
- trocar informação numa sessão de diálogo;
- gerenciar o diálogo;
- sincronizar o diálogo;
- definir e gerenciar unidades de diálogo (atividades);
- reportar erros detectados no decorrer do diálogo.

Vamos examinar cada uma das funcionalidades acima.

6.1 Estabelecimento e Término do Diálogo

Dependendo do protocolo de transporte, uma conexão de transporte pode se encerrar adruptamente por uma das partes, causando, eventualmente, perda de informação. A figura 6.1(a) ilustra esta situação onde uma conexão foi terminada antes da outra parte encerrar o envio de dados. Isto de deve ao fato das primitivas OSI de transporte prever o encerramento da conexão como um serviço sem confirmação (primitivas do tipo *request* e *indication* apenas, ver figura 5.1).

A camada de sessão provê encerramento de uma conexão de sessão como um serviço confirmado (primitivas *request, indication, response* e *confirm*). Neste caso, uma aplicação envolvida numa sessão de diálogo, ao receber uma indicação de desconexão, emite a resposta afirmativa apenas caso não tenha mais informação para transmitir. Isto garante que a conexão se encerra sem a perda de informação, mesmo que a conexão de transporte da qual a conexão de sessão se utilizava se encerre adruptamente. A figura 6.1(b) ilustra o encerramento ordenado de uma sessão de diálogo

Uma conexão de sessão pode se encerrar adruptamente caso a própria camada de sessão tome a iniciativa do encerramento. Um exemplo desta situação é a quebra da conexão de transporte da qual a conexão de sessão fazia uso.

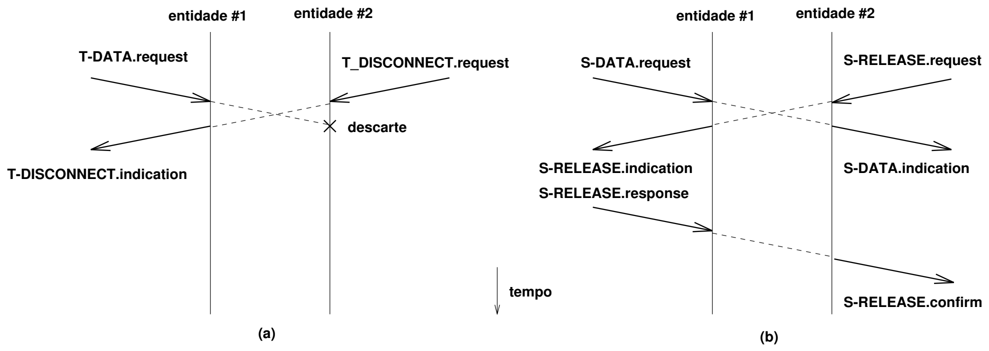

Figura 6.1: Encerramento de conexões: (a) de transporte; (b) de sessão (RELEASE tem o significado de DISCONNECT).

6.2 Troca de Informação

Sessões de diálogo podem conduzir quatro tipos de dados:

1. dados regulares: informação que diz respeito apenas às entidades comunicantes da aplicação;

2. dados expressos: informação urgente (gerada pela aplicação) entregue fora de sequência;

3. dados tipados: informação enviada sem que o transmissor detenha o *token* de envio (veja próxima sessão);

4. dados de capacidade: empregados exclusivamente para controle interno da camada de sessão, permitem alterar parâmetros do diálogo.

As primitivas para envio dos três primeiros tipos de dados de sessão são sem confirmação. O envio de dados de capacidade é um serviço confirmado, pois a alteração de parâmetros do diálogo dependem do aceite da outra parte envolvida no diálogo.

6.3 Gerenciamento do Diálogo

Conexões de transporte usualmente são do tipo full-duplex, onde a informação flui nos dois sentidos simultaneamente. Entretanto, muitas (senão a maioria) aplicações se comunicam em modo half-duplex, onde a informação flui num sentido de cada vez. Comunicação half-duplex pode ser implementada em conexão full-duplex pela introdução de um *token*: apenas quem detiver o *token* está habilitado a enviar informação pela conexão.

No estabelecimento de uma conexão de sessão é especificado o tipo da conexão half- ou full-duplex. No caso de ser escolhida conexão half-duplex, deve-se especificar também a qual das partes será atribuído o *token* primeiro.

Terminado o envio de informação, o detentor do *token* passa-o à outra parte. O *token* pode ser requisitado a qualquer momento pela entidade que não o detém. Recebida a requisição, a entidade detentora do *token* pode, a seu critério, passá-lo a quem o solicitou.

A figura 6.2 ilustra as o emprego do *token* e dados tipados em comunicação half-duplex.

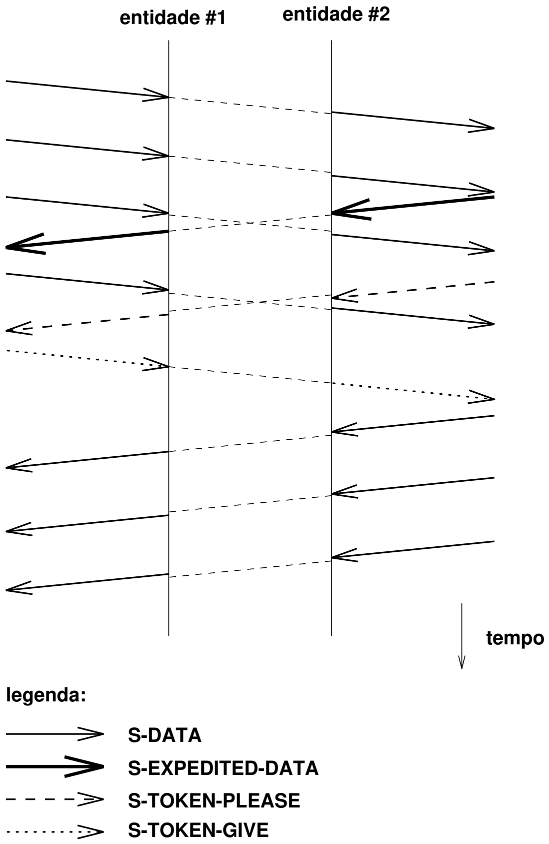

Figura 6.2: Comunicação típica no nível de sessão.

6.4 Sincronização do Diálogo

Numa sessão de diálogo pode-se definir *pontos de sincronismo*. Pontos de sincronismo estabelecem marcas no diálogo para fins de retomada da comunicação a partir destas marcas. Pontos de sincronismo são definidos pelas entidades comunicantes, não pela camada de sessão (esta os adiciona unicamente sob requisição da aplicação). A introdução de pontos de sincronismo necessita o reconhecimento da outra entidade comunicante, sendo portanto um serviço cofirmado.

Seja o exemplo de um a rede se serviços integrados que oferece transmissão por fac-simile (fax). Cada página do texto é transmitida como uma série de mensagens de dados. Antes do envio de cada página, o transmissor adiciona um ponto de sincronismo. Suponha que o receptor roteia os dados dos SPDUs diretamente para uma impressora, sem armazená-los localmente. Se ocorrer uma falha na impressão (por exemplo, travamento do papel), o receptor solicita ao transmissor que volte ao ponto de sincronismo referente à página afetada pelo problema, passando a receber todas as mensagens a partir daí. O que ocorreu neste caso foi uma ressincronização do diálogo a partir de um ponto do passado.

Caso pontos de sincronismo sejam empregados, o transmissor deve armazenar as mensagens caso o receptor ressincronize o diálogo. Obviamente, a quantidade de mensagens armazenadas deve ser limitada. Para tal, define-se dois tipos de pontos de sincronismo: menor (como os do exemplo acima) e maior.

Pontos de sincronismo maior (PSM) definem barreiras cujo retrocesso não pode ser ultrapassado, limitando, portanto, a quantidade de mensagens armazenadas para fins de ressincronização. Retomando o exemplo anterior, suponha que o transmissor adicione um PSM a cada cinco pontos de sincronismo menor (que delimitam as páginas). Agora, ao reconhecer um PSM, o receptor deve ter certeza que todas as páginas transmitidas até então foram corretamente impressas, pois as mensagens transmitidas antes de um PSM são descartadas pelo transmissor.

A quantidade de informação transmitida entre dois PSMs é denominada *unidade de diálogo* ou *atividade* (figura 6.3).

6.5 Gerenciamento de Atividades

Atividades são unidades delimitadas de uma sessão de diálogo. Geralmente a delimitação das atividades está relacionada com fronteiras naturais reconhecidas pelas entidades comunicantes. Por exemplo, numa sessão de transferência de arquivos, a transmissão de cada arquivo estaria associada uma atividade (figura 6.3).

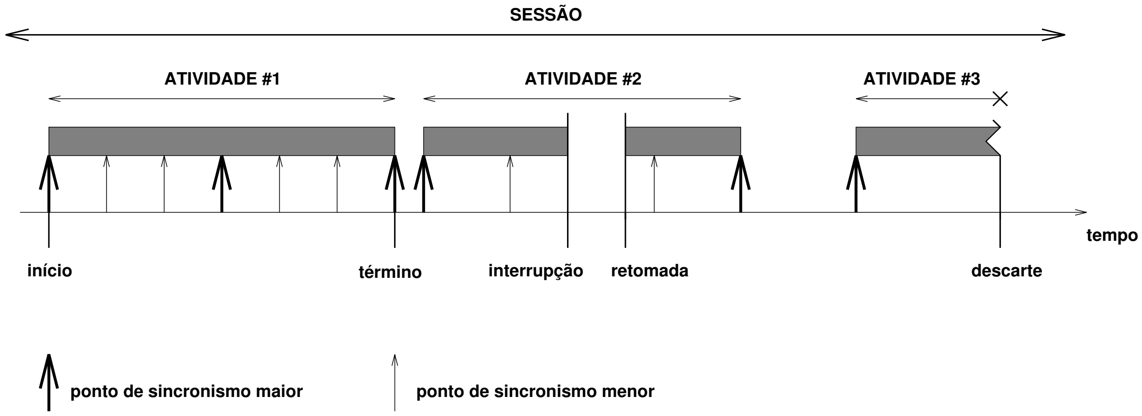

Figura 6.3: Partição de uma sessão de diálogo em atividades.

Atividades numa mesma sessão de diálogo estão delimitadas por pontos de sincronismo maior. Como consequência, durante o processamento de uma atividade não podemos ressincronizar o diálogo a um ponto pertencente à atividade anterior.

As operações relacionadas com atividades são:

- início: marca o início de nova atividade inserindo um ponto de sincronismo maior (exemplo: inicio da transmissão de arquivo);

- término: informa que a atividade corrente terminou de forma normal (exemplo: fim da transmissão de arquivo);
- descarte (término anormal): abandona a atividade corrente (exemplo: descarte da parte já transmitida do arquivo);
- interrupção: suspende temporariamente uma atividade (exemplo: suspensão de transmissão de arquivo);
- retomada: retoma uma atividade a partir do ponto em que foi interrompida (exemplo: retoma a transmissão de arquivo).

As consequências de se iniciar, terminar, descartar, suspender e retomar atividades dizem respeito apenas às aplicações usuárias da camada de sessão. A camada de sessão apenas provê as primitivas para tal, gerando os respectivos SPDUs de controle que causam as respectivas indicações na camada de sessão oposta. As primitivas para iniciar e retomar atividade interrompida são serviços sem confirmação; as demais são serviços confirmados.

Finalmente, nada impede ao usuário inserir pontos de sincronismo maior menor no interior de uma atividade.

6.6 Detecção de Erros

A camada de sessão provê apenas duas primitivas para o informe de situações anormais. A primeira é utilizada pela aplicação. Por exemplo, uma aplicação remota solicita à local a transferência de um arquivo inexistente. Esta primeira primitiva define um serviço sem confirmação.

A segunda primitiva é utilizada pela própria camada de sessão quando esta detecta alguma situação anormal. As aplicações conectadas numa sessão de diálogo recebem apenas uma indicação do erro, podendo proceder como desejarem. Exemplo de erro neste nível é a perda do sequenciamento na transmissão dos SPDUs que compõem uma mensagem.

6.7 Primitivas OSI de Sessão

A tabela 6.1 apresenta as primitivas OSI para a camada de sessão. As primitivas fornecem os serviços (orientados à conexão) descritos nas seções precedentes. Primitivas iniciando com S-P possuem o modo *indication* apenas e se prestam para o informe de eventos detectados pela entidade provedora do serviço (neste caso, os protocolos da camada de sessão). Uma única primitiva, S-UNITDATA, provê transmissão de informação sem o estabelecimento de conexão.

6.8 Protocolo OSI de Sessão

O protocolo OSI de sessão (ISO 8327) é inspirado num protocolo antigo da CCITT para teletexto. A forma geral dos SPDUs é dada na figura 6.4(a). A campo SI (Session Identifier)

primitiva	**finalidade**
S-CONNECT	Estabelecimento de uma conexão de sessão
S-RELEASE	Término de uma conexão de sessão
S-U-ABORT	Termina adruptamente uma conexão
S-P-ABORT	Informa o término adrupto de uma conexão
S-DATA	Transfere de dados
S-EXPEDITED-DATA	Transfere de dados expressos
S-TYPED-DATA	Transfere de dados fora de faixa (out-of-band)
S-CAPABILITY-DATA	Transfere de informação de controle
S-TOKEN-GIVE	Passa o *token* à outra entidade
S-TOKEN-PLEASE	Solicita o *token* de outra entidade
S-CONTROL-GIVE	Passa todos os *tokens* à outra entidade
S-SYNC-MAJOR	Insere um ponto de sincronização maior
S-SYNC-MINOR	Insere um ponto de sincronização menor
S-RESYNCHRONIZE	Retorna a um ponto de sincronização
S-ATIVITY-START	Inicia uma atividade
S-ACTIVITY-END	Termina uma atividade
S-ACTIVITY-DISCARD	Descarta uma atividade
S-ACTIVITY-INTERRUPT	Interrompe uma atividade em curso
S-ATIVITY-RESUME	Retoma uma atividade interrompida
S-U-EXCEPTION-REPORT	O usuário informa sobre uma exceção
S-P-EXCEPTION-REPORT	A entidade provedora informa sobre uma exceção
S-UNITDATA	Transfere dados sem o estabelecimento de conexão

Tabela 6.1: Primitivas OSI para a camada de sessão. As primitivas em negrito são serviços confirmados.

de um byte contém o tipo do SPDU. O campo LI (Length Identifier), também de um byte, indica o tamanho (em bytes) do campo de parâmetros. LI varia entre 0 e 254: um valor de 255 indica a existência de 2 bytes adicionais para este campo.

Após os campos SI e LI, vêm os parametros do SPDU. Parâmetros têm o formato da figura 6.4(b). PI (Parameter Identifier) informa o tipo do parâmetro; LI seu tamanho; PV (Parameter Value) seu valor.

Certos SPDUs provêem campo de dados do usuário para que os usuários da camada de sessão enviem informação adicional àquelas passadas nas primitivas (a camada de sessão não interpreta estes dados).

Grosso modo, a chamada de cada primitiva de sessão gera um SPDU do tipo da primitiva. Primitivas que implementam serviços confirmados definem também seus correspondentes SPDUs de reconhecimento.

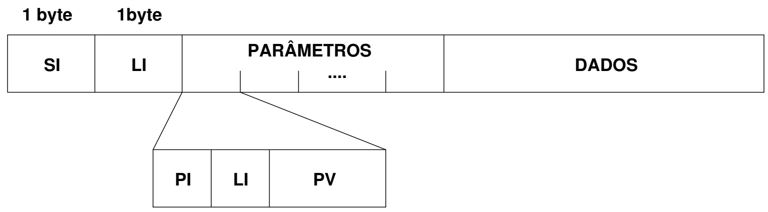

Figura 6.4: Formato geral de um SPDU (para compatibilidade com o protocolo CCITT para teletexto).

6.9 Problemas

1. Quais as funcionalidades que a camada de sessão provê ?
2. Quais as diferenças entre uma conexão de transporte e uma conexão de sessão ?
3. Como a camada de sessão implementa comunicação half-duplex ?
4. Como a camada de sessão classifica os dados por ela transmitidos ?
5. Como funciona o mecanismo de sincronização da camada de sessão ?
6. Quais as vantangens de se organizar o diálogo em atividades ?
7. Quais as operações de gerenciamento de atividades ?
8. Faça um diagrama temporal mostrando uma ressincronização do diálogo. Idem para o início, suspensão, retomada e término de uma atividade.
9. Cite as primitivas OSI de sessão.
10. Descreva os campos de um SPDU do protocolo OSI de sessão.

Capítulo 7

A CAMADA DE APRESENTAÇÃO

A camada de apresentação é empregada para alterar a representação de dados transmitidos via rede para fins de:

1. compatibilizar a comunicação;
2. segurança e privacidade;
3. compactação de dados.

O primeiro item diz respeito a representação canônica de dados: uma representação inteligível por todos os hosts comunicantes independentemente de suas arquiteturas de máquina. Os dois últimos itens se relacionam com criptografia e compressão de dados. A camada de apresentação é o local para se implementar tais funções. Criptografia e compressão de dados são tópicos extensos e dependentes do contexto no qual a aplicação está inserida. Estes tópicos não serão cobertos neste texto.

7.1 Representação Canônica de Dados

Até então nos referimos a *dados* sem nos preocupar com sua representação. Infelizmente, representar dados não é tarefa trivial, posto que diferentes computadores adotam representações diferentes para o mesmo dado. Por exemplo, a IBM de longa data emprega o código EBCDIC em seus *mainframes* para representar texto. Os demais fabricantes empregam o códico ASCII. Outro problema é a sequência de bits numa palavra: alguns computadores empregam o bit mais significativo à esquerda, enquanto outros à direita. Por exemplo, o número 11 é representado como 0...01011 numa SUN e 11010...0 num VAX. Como regra geral, a forma de representação interna de dados adotada por computadores é fortemente influenciada por suas arquiteturas de hardware.

Com o advento das redes heterogêneas (interconectando hardware de diferentes arquiteturas), tornou-se necessário uma representação canônica de dados (ou sintaxe de tranferência). A idéia é simples. Antes de transmitir um dado, o host converte de sua representação interna para a representação canônica. Ao recebê-lo, o host receptor converte da representação canônica para sua representação interna. Em outras palavras, os dados que fluem pela rede têm uma representação padronizada. Isto evita que hosts comunicantes necessitem conhecer mutuamente a forma de representação de dados por eles empregadas [1].

No modelo OSI, a representação canônica de dados baseia-se numa sintaxe denominada ASN.1 (Abstract Syntax Notation, version 1).

7.2 A Sintaxe ASN.1

ASN.1 é uma linguagem formal de especificação de tipos de dados padronizada pela ISO (documento 8824). Uma segunda padronização, BER (Basic Encoding Rules), especifica como dados na sintaxe ASN.1 são formatados numa mensagem (por exemplo, um TPDU). BER é também um padrão ISO (documento 8825).

Aplicações definem todas as estruturas de mensagens válidas (tipos de PDUs) em ASN.1, agrupando estas definições num único módulo: o *dicionário de dados*. Ao transmitir um PDU, a aplicação passa à camada de apresentação o tipo de PDU sendo transmitido e o conteúdo do seus campos. A camada de apresentação consulta o dicionário de dados em busca da definição dos campos do PDU e, utilizando a norma BER, gera a mensagem a ser transmitida. A figura 7.1 ilustra este processo.

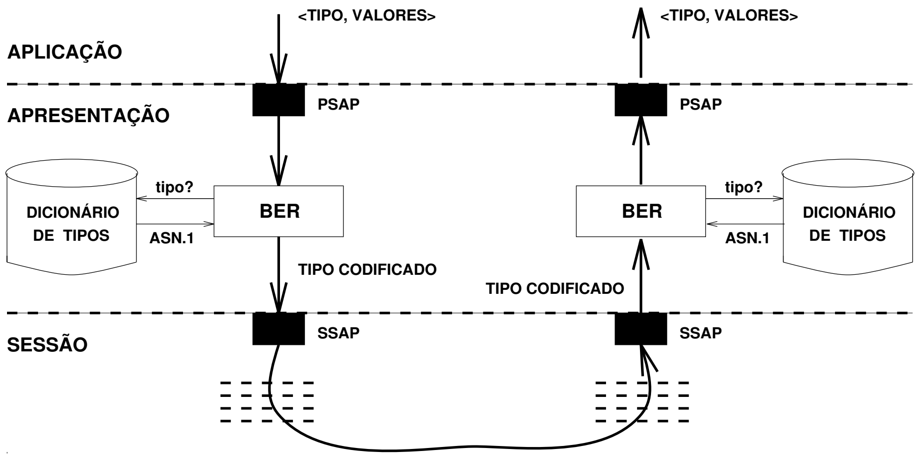

Figura 7.1: O serviço OSI de apresentação.

Quando a mensagem atingir a camada de apresentação do receptor, é verificado o tipo de PDU. O dicionário de dados é então consultado para se obter a estrutura do PDU recebido. Com base sua na estrutura e, sabendo-se que a codificação segue ASN.1/BER, o receptor pode remontar o PDU segundo sua representação própria de dados.

Como toda a linguagem formal, ASN.1 consiste num conjunto de elementos básicos *(tokens)* e regras de combinação destes elementos. ASN.1 define quatro tipos de *tokens*:

[1] Esta solução é inviável por três razões: a quantidade de arquiteturas (e portanto representações) distintas; o aparecimento contínuo de novas arquiteturas; e a impossibilidade de difusão de mensagens num formato inteligível por todos os hosts.

palavras: sequências de letras maiúsculas ou minúsculas, dígitos e hífem. O primeiro caractere deve ser uma letra. Palavras definem tipos de dados, identificadores e palavras-chave.

números: formações compostas apenas de dígitos.

strings: cadeia de caracteres to tipo

- alfanumérico: qualquer conteúdo delimitado por " "
- hexadecimal: conteúdo entre 0-9 e A-F delimitado por ' 'H
- binário: 0 ou 1 delimitados por ' 'B

pontuação: caracteres do tipo

```
. : = , { } < > - ' " |
```

7.2.1 Tipos Primitivos de Dados

ASN.1 define um conjunto de tipos primitivos de dados. Tipos mais complexos são obtidos através da agregação dos tipos primitivos. Os tipos primitivos são:

boolean: valores lógicos (TRUE-FALSE);

integer: números inteiros;

real: números reais (mantissa, base, expoente);

bit String: sequência ordenada de 0s e 1s;

octet string: sequência ordenada de bytes (octetos);

any: tipo a ser especificado;

null: tipo sem valor;

object descriptor: referencia algum objeto sem sintaxe formal (usado mais a título de comentário);

object identifier: define univocamente um objeto, por exemplo, um protocolo de transferência de arquivos;

encripted: tipo encriptado.

Exemplos:

```
ChecksumInUse ::= BOOLEAN
ChecksumInUse ::= FALSE

TimeToLive ::= INTEGER
TimeToLive ::= 60

Timeout ::= REAL
Timeout ::= {250, 10, -3}

Preamble ::= BITSTRING
Preamble ::= '01010101'B

ForFutureUse ::= ANY

NoLongerInUse ::= NULL

Ftam ::= OBJECT DESCRIPTOR
Ftam ::= "File Transfer, Access and Management"

Ftam ::= OBJECT IDENTFIER
Ftam ::= { iso standard 8571 }
```

7.2.2 Tipos Complexos (Construtores)

Tipos constutores são formados a partir dos tipos primitivos descritos na seção anterior. São eles:

sequence: lista ordenada de tipos primitivos arbitrários;

sequence of: lista ordenada de tipos primitivos homogêneos;

set: lista não ordenada de tipos primitivos arbitrários;

set of: lista não ordenada de tipos primitivos homogêneos;

choice: um conjunto de tipos dos quais apenas um deve ser considerado.

Exemplos:

```
PDUHeader ::= SEQUENCE { Length INTEGER,
                         Flags BITSTRING,
                         DestinationTSAP INTEGER,
                         SourceTSAP INTEGER }
```

```
ProtocolClassOption ::= CHOICE { class-0 OCTETSTRING,
                                 class-1 OCTETSTRING,
                                 class-2 OCTETSTRING,
                                 class-3 OCTETSTRING,
                                 class-4 OCTETSTRING }
```

7.2.3 Tipos Marcados *(Tagged)*

Tipos marcados carregam uma informação adicional para fins de eliminação de ambiguidades. Vimos pelos exemplos acima que tipos construtores são estruturas compostas de múltiplos campos. Durante a transmissão de um tipo construtor, o tipo tamanho e valor de cada um de seus campos é codificado segundo a norma BER. Um problema decorre da existência de campos opcionais. A palavra-chave OPTIONAL pode ser usada para identificar um campo como opcional. Outra palavra-chave, DEFAULT, estabelece o valor que deve ser tomado caso o campo (opcional) seja omitido. Por exemplo:

```
PDUHeader ::= SEQUENCE {
  Length INTEGER,
  Flags BITSTRING OPTIONAL DEFAULT '00000000'B,
  DestinationTSAP INTEGER,
  SourceTSAP INTEGER
  }
```

Ao receber um PDU com o cabeçalho dado pela sequência acima, como o receptor reconhece se o campo opcional foi transmitido ? A solução é marcar explicitamente os componentes da sequência e transmitir a marca junto com o dado. Exemplo:

```
PDUHeader ::= SEQUENCE {
  Length [0] INTEGER,
  Flags [1] BITSTRING OPTIONAL DEFAULT '00000000'B,
  DestinationTSAP [2] INTEGER,
  SourceTSAP [3] INTEGER
  }
```

Assim sendo, se o campo *Flags* for omitido, o receptor toma conhecimento pela ausência da marca [1]. Com a transmissão da marca, o tipo fica redundante pois pode ser acessado pelo receptor na definição do tipo construtor. Esta redundância pode ser mantida como uma verificação adicional. Para eliminá-la deve-se utilizar a palavra-chave IMPLICIT:

```
PDUHeader ::= SEQUENCE {
  Length [0] IMPLICIT INTEGER,
  Flags [1] IMPLICIT BITSTRING OPTIONAL DEFAULT '00000000'B,
  DestinationTSAP [2] IMPLICIT INTEGER,
  SourceTSAP [3] IMPLICIT INTEGER
  }
```

Suponha agora que o protocolo possui vários tipos de PDUs (por exemplo, 9 para o TP-4). Ao receber um PDU, como o receptor descobre de que tipo se trata ? Se o cabeçalho do exemplo anterior for, digamos, do terceiro tipo, podemos marcar não apenas seus campos, mas também sua estrutura como um todo:

```
PDUHeader ::= [APPLICATION 3] SEQUENCE {
  Length [0] IMPLICIT INTEGER,
  Flags [1] IMPLICIT BITSTRING OPTIONAL DEFAULT '00000000'B,
  DestinationTSAP [2] IMPLICIT INTEGER,
  SourceTSAP [3] IMPLICIT INTEGER
  }
```

A palavra-chave APPLICATION denota um tipo construtor no dicionário de dados da aplicação OSI. A marca, 3, indica se tratar da terceira definição presente no dicionário. APPLICATION pode ser substituída por PRIVATE (aplicação privada), CONTEXT-SPECIFIC (interpretação dependente do conexto) e UNIVERSAL (tipos primitivos do ASN.1).

7.3 Sintaxe de Transferência

Sintaxe de transferência especifica como um tipo ASN.1 é formatado numa mensagem. A norma BER fornece regras para esta formatação. Dada a sua extensão, veremos apenas os aspectos fundamentais do BER.

BER adota inteiramente o código ASCII para caracteres. As representações adotam o bit mais significativo a esquerda. Números são representados pelos seus complementos de dois.

Tipos primitivos ou não são formatados em três campos: a identificação do tipo, o tamanho do dado, e o valor propriamente dito. Caso o tipo não seja primitivo, o valor irá conter também outros três campos (tipo, tamanho, valor), e assim sucessivamente até chegarmos unicamente a tipos primitivos.

7.3.1 Identificação do Tipo, Tamanho e Valor

O tipo é identificado por um byte. Os primeiros dois bits informam a classe do tipo (UNIVERSAL, APPLICATION, CONTEXT-SPECIFIC e PRIVATE). O próximo bit denota se o tipo é primitivo ou não. Os cinco bits restantes identificam o tipo na classe. Para a classe UNIVERSAL, 1 denota BOOLEAN, 2 denota INTEGER, 3 denota BITSTRING, 4 denota OCTETSTRING, e assim por diante.

Caso uma classe tenha mais que 30 tipos, ativa-se todos os 5 bits do tipo seguindo-se tantos bytes quantos forem necessários para abrigar o índice máximo, todos começando com 1 no primeiro bit (restando sete para o tipo), exceto o último.

O tamanho do dado, se menor que 128 ocupa um único byte, tendo o primeiro bit com valor 0. Se maior que 128, o primeiro bit é feito 1 e os sete restantes estipulam quantos bytes a seguir armazenam o valor. Um tamanho de conteúdo nulo (todos os bits 0) indica que o valor será delimitado por dois bytes de conteúdo nulo em cada extremo.

A codificação de valores varia com o tipo. Por exemplo, um OCTETSTRING de N bytes é codificado empregando-se um byte por caractere, segundo o código ASCII. Exemplos:

```
numero 100:     00000010 00000001 01100100

numero 32768:   00000010 00000010 01000000 00000000

string "A":     00000100 00000001 01100001
```

Como exemplo de tipos construtores, tomemos o caso de uma SEQUENCE. Neste caso, o byte que descreve o tipo indica se tratar de uma sequência, o byte tamanho indica o número de elementos da sequência (N) e o byte valor dá lugar a N triplas (tipo, tamanho, valor). Por exemplo, uma sequência composta do número 100 e do string "A" é codificada como:

```
Exemplo ::= [UNIVERSAL 7] SEQUENCE {
                elem-1 [1] IMPLICIT INTEGER,
                elem-2 [2] IMPLICIT OCTETSTRING }
```

Para valores <100, "A">, a sequência acima é codificada como:

```
00001000 00000010 00000010 00000001 01100100 00000100 00000001 01100001
```

7.4 Primitivas OSI de Apresentação

As primitivas OSI para a camada de apresentação são as mesmas, exceto uma, da camada de sessão [2] (ver figura 6.1). Simplesmente tais primitivas "curto-circuitam" a camada de

[2] Começando com P- ao invés de S-

apresentação, permitindo à camada de aplicação o acesso às funções da camada de sessão. Tais primitivas existem para manter o conceito de camadas.

Uma nova primitiva S-ALTER-CONTEXT provê um mecanismo de mudança de contexto, isto é, passar de uma atividade de sessão para outra sem suuspender a primeira. Esta primitiva é um serviço com confirmação.

7.5 Problemas

1. Quais as funções da camada de apresentação ?
2. Especifique os NPDUs do X.25/camada 3 em ASN.1.

Capítulo 8

A CAMADA DE APLICAÇÃO

8.1 Funcionalidades da Camada de Aplicação

A camada de aplicação do modelo OSI define protocolos (e entidades que os implementam) utilizados por aplicativos denominados APs (Application Processes ou Processos de Aplicação). Por exemplo, um AP que implementa um serviço de mensagens (via caixa postal, por exemplo) quando operando num ambiente OSI utilizaria o protocolo MHS (Message Handling System) definido na camada de aplicação.

A camada de aplicação tem por finalidade prover os seguintes serviços aos APs que dela se utilizam:

- identificar elementos remotos de aplicação, verificando sua disponibilidades (servidores, por exemplo);
- negociar com estes elementos uma qualidade de serviço apropriada;
- negociar contextos de apresentação para troca de informação;
- negociar serviços de sessão;
- transportar dados entre aplicativos;
- autenticar as entidades comunicantes.

Alguns exemplos de serviços providos pela camada de aplicação são listados abaixo:

- serviço de acesso e transferência de arquivos;
- serviço de troca de mensagens;
- serviço de diretório;
- serviço de manipulação de tarefas (jobs) remotas;
- serviço de controle de concorrência e recuperação;

- serviço de terminal remoto;
- etc.

Para o provimento destes serviços a camada de aplicação possui uma estruturação relativamente complexa quando comparada às demais camadas do modelo OSI. Esta estruturação é sintetizada na sequência.

8.2 Estruturação da Camada de Aplicação

A camada de aplicação agrega *entidades de aplicação* responsáveis pelo provimento de serviços aos APs. Estes serviços podem ser de propósito geral ou específico. Os serviços de propósito geral são providos por entidades denominadas CASE (Common Application Service Elements), enquanto os de propósito específicos são providos pelas entidades SASE (Specific Application Service Elements). Em linhas gerais, APs usam serviços oferecidos por SASEs que por sua vez usam serviços oferecidos pelos CASEs conforme ilustrado na figura 8.1.

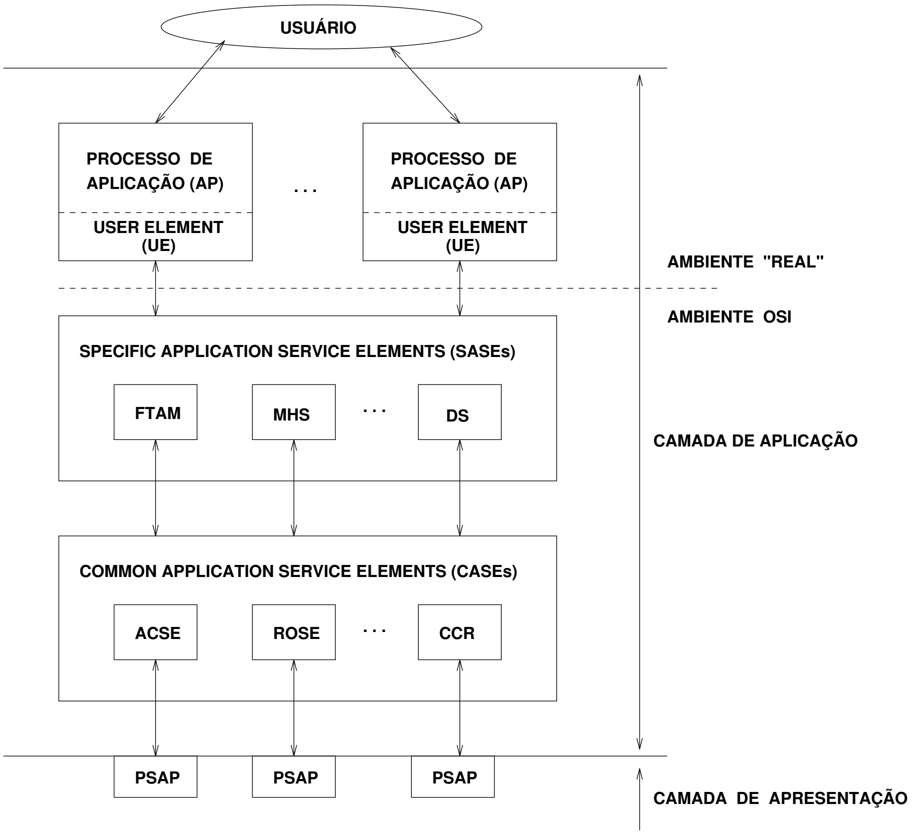

Figura 8.1: Os componentes da camada de aplicação do modelo OSI.

Um conceito importante na camada de aplicação é o conceito de *associação*. As demais camadas do modelo OSI oferecem a abstração de conexão (primitivas tipo CONNECT) como o meio interação entre duas entidades da camada N+1 via serviços oferecidos pela camada N. O que ocorre se $N = 7$, ou, em outas palavras, *quem* utilizaria as conexões da camada de aplicação ?

Este problema foi resolvido através das regras abaixo:

1. a camada de aplicação define associação (não conexão) como forma de provimento de serviços aos APs;

2. cada associação de aplicação utiliza uma única conexão de apresentação (isto é, uma associação pode ser vista como uma extensão de uma conexão de apresentação).

Um CASE denominado ACSE (Association Control Service Element) oferece o serviço de estabelecimento de associação para as demais entidades da camada de aplicação.

Atualmente, a tendência é evitar a distinção entre CASE e SASE e denominar estas entidades simplesmente de ASE (Application Service Element). Uma quentão crucial é: como combinar ASEs para implementar um serviço complexo ? Esta questão está vinculada com o conceito de associação. Intuitivamente, um ASE proveria uma associação através de uma conexão de apresentação agregando valor à conexão oferecida pela camada inferior[1]. Este modelo apresenta alguns problemas quando ASEs são combinados. O principal problema refere-se à complexidade da coordenação de múltiplas associações (cada qual utilizando uma única conexão de apresentação). O que ocorre, por exemplo, se uma das associações é encerrada pelo provedor ?; ou, como gerenciar o diálogo a nível de aplicação se diferentes associações utilizam diferentes serviços de sessão[2] ?

A proposta da ISO para esta questão é a definição de um componente agregador denominado ASO (Application Service Object). Um ASO é um objeto provedor de serviço que agrega:

- Elementos de Serviço de Aplicação (ASE);

- Outros ASOs;

- Uma única Função de Controle (CF: Control Function) que especifica como os componentes de um ASO são agregados.

Um ASO estabelece uma única associação eliminando, portanto, o problema de gerenciamento de múltiplas associações (uma por ASE). Obviamente, caso um ASO agregue outros ASOs múltimas associações emanarão deste, mas as demais associações são encapsuladas pelos ASOs agregados.

Entidades de aplicação agora são definidas em termos de ASOs, não mais em termos de ASEs. Em linhas gerais, neste modelo os ASEs definem mensagens (APDUs), enquanto os ASOs (via CF) determinam a dinâmica da interação através destas mensagens.

[1] As demais camadas operam segundo este princípio.

[2] Por exemplo, uma utiliza *tokens* para gerenciar o diálogo e a outra não.

A figura 8.2 ilustra esta nova concepção da camada de aplicação. Infelizmente, esta concepção foi elaborada após muitos ASEs já terem sido padronizados. Esta padronização não levou em conta a possibilidade destes ASEs serem agregados em torno de um ASO. A reformulação destes padrões para adequarem os ASEs à nova estruturação da camada de aplicação é um processo incerto.

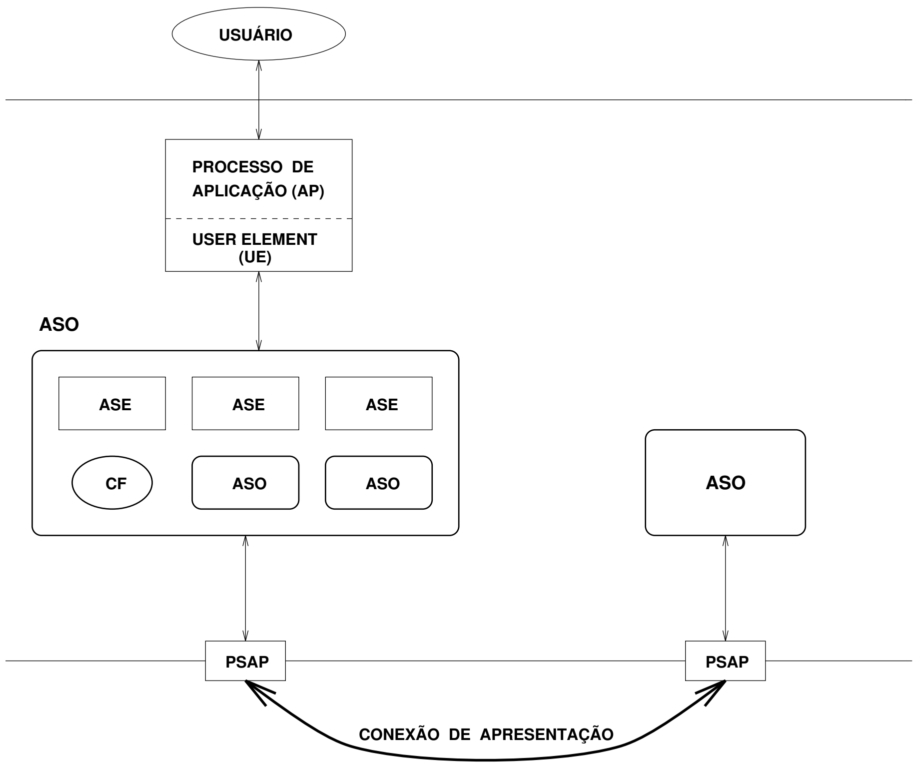

Figura 8.2: Camada de aplicação baseada em ASOs.

A seguir apresentaremos alguns elementos de serviço da camada de aplicação.

8.3 Serviço de Associação (ACSE)

Conforme mensionado anteriormente, o elemento de aplicação responsável pelo estabelecimento de associação é o ACSE. O ACSE utiliza uma conexão de apresentação para cada associação que estabelece. Um dado interessante é que após estabelecida a associação, a entidade que a requisitou passa a utilizar diretamente a conexão de apresentação aberta para esta associação para a troca de dados. Entretanto, o encerramento da associação se dá através do ACSE.

Além dos parâmetros requeridos pela primitiva P-CONNECT, o estabelecimento de uma associação necessita:

- identificadores das entidades de aplicação envolvidas na associação (iniciadora e respondedora);

- um contexto de aplicação: um identificador de objeto em ASN.1 utilizado para identificar as regras de intercâmbio de informação através da associação (por exemplo, protocolo de aplicação);

- dados supridos pela entidade usuária (não interpretados pelo ACSE).

- um contexto de autenticação: um identificador de objeto em ASN.1 utilizado para identificar as regras de autenticação (por exemplo, chaves criptográficas).

Os serviços providos pelo ACSE são acessados através das primitivas abaixo:

1. A-ASSOCIATE: estabelece uma associação utilizando a primitiva P-CONNECT da camada de apresentação. Este serviço é confirmado.

2. A-RELEASE: termina uma associação utilizando a primitiva P-RELEASE de apresentação. Este serviço é confirmado.

3. A-ABORT: termina adruptamente (aborta) uma associação por solicitação da entidade usuária utilizando a primitiva P-U-ABORT de apresentação. Este serviço é sem confirmação.

4. A-P-ABORT: informa o término adrupto de uma associação pelo provedor do serviço. Esta primitiva é ativada quando da ocorrência de um P-P-ABORT na camada de apresentação e possui o modo indicação apenas.

5. A-UNITDATA: transfere uma unidade de dados de aplicação sem o estabelecimento de associação. Este serviço é sem confirmação e utiliza a primitiva P-UNITDATA de apresentação.

Note a inexistência da primitiva A-DATA para envio de dados através de uma associação. A primitiva P-DATA é empregada para esta finalidade.

Um ponto importante do ACSE é como entidades de aplicação são identificadas. A identificação AEs se dá através de nomes que o ACSE mapeia em endereços de apresentação (PSAPs) utilizando o próprio serviço de diretório da camada de aplicação.

8.4 Serviço de Operações Remotas (ROSE)

ROSE (Remote Operations Service Element) é um ASE que intermedia a submissão remota de operações. Exemplo de tais operações são: armazenamento de mensagens e submissão de tarefas (jobs) remotas.

O ROSE fornece um serviço similar à uma chamada de procedimento remoto (RPC: Remote Procedure Call), sendo que este serviço pode ter natureza síncrona ou assíncrona. No modo síncrono a entidade solicitante permanece bloqueada até a entidade executora retornar o resultado da operação. No modo assíncrono, a entidade solicitante pode evocar múltiplas operações e só então coletar os respectivos resultados. Sob esta ótica, as operações remotas são agrupadas em cinco classes:

- classe 1: operação síncrona que retorna indicativo de sucesso ou falha;
- classe 2: operação assíncrona que retorna indicativo de sucesso ou falha;
- classe 3: operação assíncrona que retorna somente indicativo de falha;
- classe 4: operação assíncrona que retorna somente indicativo de sucesso;
- classe 5: operação assíncrona que não retorna qualquer indicativo.

O ROSE utiliza associações para a comunicação entre entidades remotas[3]. Uma associação aberta para suporte à operaçoes remotas define através de sua classe qual entidade está autorizada a evocar procedimentos remotos:

- classe 1: permite apenas a entidade que solicitou a associação evocar operações remotas;
- classe 2: permite apenas a entidade que aceitou a associação evocar operações remotas;
- classe 3: permite ambas as entidades evocarem operações remotas.

O ROSE provê seus serviços através de cinco primitivas:

1. RO-INVOKE: solicita a execução de uma operação remota;
2. RO-RESULT: retorna o resultado da operação remota;
3. RO-ERROR: reporta um erro na execução de uma operação remota;
4. RO-REJECT-U: rejeita uma solicitação ou retorno se a entidade usuária detectar um erro (por exemplo, operação inexistente ou falha de autenticação);
5. RO-REJECT-P: rejeita uma solicitação ou retorno se o provedor de serviço detectar um erro.

Todos os serviços são sem confirmação (RO-REJECT-P possui o modo indicação apenas).

O ROSE define quatro classes de de PDUs, um para cada tipo de primitiva. Argumentos, resultados e erros das operações são especificados em ASN.1.

O ROSE provê ainda o conceito de operações ligadas (linked). Neste mecanismo um argumento de uma operação identifica outra operação dirigida para o própria entidade solicitante. Um dos objetivos é evitar a passagem de argumentos volumosos (matrizes, por exemplo) através da execução de operações no local onde os argumentos estão armazenados (emulando a passagem de argumentos por referência). A figura 8.3 ilustra este exemplo.

[3]Propostas recentes permitem operações remotas sem o estabelecimento de associação através da primitiva A-UNITDATA.

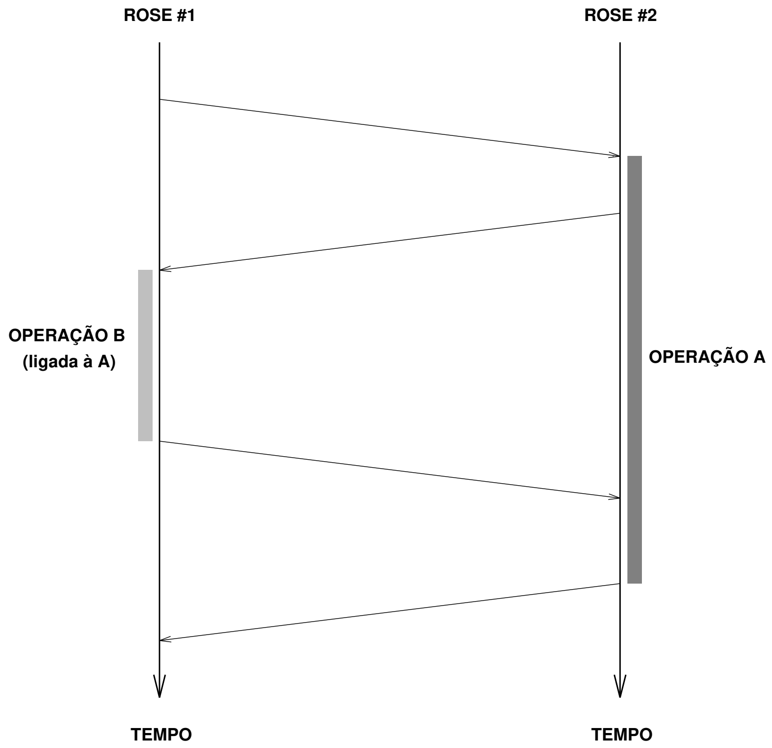

Figura 8.3: Operações ligadas no ROSE.

8.5 Serviço de Transferência Confiável (RTSE)

RTSE (Reliable Transfer Service Element) é um ASE que provê a transferência confiável de dados entre duas entidades de aplicação. Por confiável subentende-se a capacidade de recuperação de erros (quebras de conexões, panes de hardware, etc.).

O RTSE fornece uma interface mais simples à camada de sessão, além de "esconder" a existência do ACSE. Por exemplo, o RTSE insere automaticamente pontos de sincronismo no fluxo de dados sendo transferido, além de gerenciar a atividade de sessão aberta para a transferência. O RTSE é utilizado, por exemplo, nas implementações MHS (Message Handling System) para transferência confiável de mensagens entre comutadores (relays).

O RTSE emprega transferência alternada nos dois sentidos (TWA: Two Way Alternate), com mecanismo de *token* para gerenciamento do sentido da comunicação.

Para cada PDU submetido para transferência, o RTSE abre uma nova atividade a nível de sessão. O PDU é então segmentado em blocos, sendo a transferência de cada bloco precedida por um ponto de sincronismo menor. Quando todos os blocos tiverem sua recepção confirmada o RTSE confirma à entidade usuária a entrega do PDU e encerra a atividade. Caso um bloco não tenha sido confirmado, o RTSE solicita ao outro extremo uma retroação ao ponto de sincronismo inserido anteriormente à este bloco.

Um ponto importante do RTSE é que este assume que os dados sendo transferido são constituídos de *string* de bytes, isto é, o tipo ASN.1 OCTET STRING. A razão desta imposição é facilitar a inserção automática de pontos de sincronismo menor para fins de retroação ante a ocorrência de falhas. Isto sem duvida consiste numa limitação severa posto que as entidades usuárias do RTSE certamente desejariam transferir dados bem mais estruturados. Para contornar esta limitação, a especificação RTSE define um *serviço de compatibilização sintática*[4]. Este serviço permite que as entidades usuárias definam implementações locais que transformam os tipos de dados peculiares da entidade em *string* de octetos. Regras para esta transformação está fora da especificação do RTSE.

O RTSE dispõe de sete primitivas:

- RT-OPEN: solicita o estabelecimento de uma associação. É um serviço confirmado.
- RT-CLOSE: solicita o encerramento de uma associação. É um serviço confirmado.
- RT-TRANSFER: transfere dados. É um serviço confirmado.
- RT-TURN-PLEASE: solicita à outra entidade habilitação para transferir dados. É um serviço não confirmado.
- RT-TURN-GIVE: passa o *token* à outra entidade, habilitando-a a transferir dados. É um serviço não confirmado.
- RT-U-ABORT: termina adruptamente uma associação. É um serviço não confirmado.
- RT-P-ABORT: o provedor sinaliza a impossibilidade de manter a associação. Possui o modo indicação apenas.

8.6 Serviço de Controle de Concorrência e Recuperação (CCR)

CCR (Commitment, Concurrency and Recovery) é um serviço de suporte a transações atômicas. Uma transação atômica é um conjunto de operações (tipicamente sobre uma massa de dados) cuja execução deve satisfazer a três propriedades:

1. atomicidade: ou todas as operações se completam com sucesso, ou tudo que a transação produziu é desfeito;
2. serialização: as operações pertencentes a transações atômicas distintas não se entrelaçam;
3. persistência: quando uma transação atômica termina com sucesso, o resultado de suas operações não se perde ante a falhas não catastróficas tais como queda de hosts e rede.

Transações atômicas têm por finalidade manter consistente o estado de determinados objetos (massas de dados, programas, etc) quando manipulados por operações sujeitas a falhas.

[4]Syntax Matching Service.

Uma transação atômica opera sobre objetos localizados em um ou mais hosts. No último caso, um host é eleito como *coordenador* da transação (geralmente o host da aplicação que solicitou a transação) sendo os demais *participantes*. Tanto o coordenador quanto os participantes são responsáveis pela garantia das três propriedades acima. Um instante crítico para esta garantia é o encerramento da transação. Um protocolo de consenso se faz necessário e o mais utilizado é o protocolo de *compromissamento em duas fases*[5].

A implementação de transações atômicas não é tarefa trivial. Mecanismos de recuperação *(rollback)* são necessários para a garantia da atomicidade; de bloqueio *(locking)* para garantir a serialização; e de *logging* para a garantia da persistência.

O CCR não oferece tais mecanismos, posto que são fortemente dependentes da aplicação. Em resumo, o CCR provê primitivas para iniciar, abortar e terminar uma transação segundo o protocolo de compromissamento em duas fases.

As primitivas CCR são:

- C-BEGIN: o coordenador informa aos participantes o início de uma transação atômica (sem confirmação);
- C-PREPARE: o coordenador solicita aos participantes o encerramento da transação (sem confirmação);
- C-READY: o participante responde o coordenador que está pronto para encerrar a transação (sem confirmação). Neste momento, todas as operações efetuadas pela transação são feitas permanentes (gravadas em disco).
- C-REFUSE: o participante responde o coordenador que está impossibilitado de encerrar a transação com sucesso [6] (sem confirmação);
- C-COMMIT: o coordenador informa aos participantes que a transação pode se encerrar com sucesso (caso todos os participantes tenham respondido C-READY). Este serviço é confirmado;
- C-ROLLBACK: o coordenador informa aos participantes que a transação deve ser abortada (caso pelo menos um participante respondeu C-REFUSE). Coordenador e participantes desfazem todas as operações efetuadas pela transação. Este serviço é confirmado;
- C-RESTART: reinicia, se possível, uma transação a partir de um ponto especificado. Este serviço é confirmado.

Note a inexistência de uma primitiva tipo C-DATA. Tal qual o RTSE, o CCR utiliza os serviços da camada de apresentação para transferência de dados. O CCR fornece apenas os mecanismos de controle para o processamento de transações. As operações realizadas no âmbito de uma transação estão fora da especificação CCR. Estas podem se basear no ROSE ou simplesmente utilizar a primitiva P-DATA de apresentação para o envio de informação.

[5] *Two-fase commit.*

[6] Provavelmente porque alguma de suas operações falhou, ou está impossibilitado de tornar as operações permanentes.

Do ponto de vista da implementação, uma transação é delimitada por pontos de sincronismo maior. As primitivas C-REFUSE, C-ROLLBACK e C-RESTART emitem um P-RESYNCHRONIZE a fim de retroagir (via facilidades de *logging*) ao ponto de sincronismo inserido no início da transação.

Uma especificação correlata denominada TP (Transaction Processing) incorpora todas as primitivas do CCR, definindo inclusive uma primitiva TP-DATA.

8.7 Serviço de Troca de Mensagens (MHS)

MHS (Message Handling System) é um serviço de manipulação[7] de mensagens. Este serviço é conhecido como X.400 por ter se originado na CCITT e publicado com esta identificação.

MHS se destina tanto ao suporte de sistemas de mensagens inter-pessoal (correio eletrônico) quanto documentos comerciais (EDI: Electronic Data Exchange). O MHS não se restringe a caracteres ASCII, como o SMTP[8], seu equivalente da arquitetura TCP/IP. Mensagens compostas de textos, imagens, gráficos, etc., podem ser manipuladas pelo MHS.

O modelo MHS consiste de vários elementos funcionais, descritos a seguir. Um Agente do Usuário (UA: User Agent) pode ser visto como um aplicativo que interage com o usuário. Por usuário subentende-se uma pessoa ou um programa que faz uso do MHS. O UA é encarregado de preparar, enviar e receber mensagens.

O envio de mensagens se dá através de Agentes de Transferência de Mensagens (MTA: Message Transfer Agent). MTAs são comutadores de mensagens, isto é, armazenam temporariamente e enviam a outro MTA (política "Armazena e Envia") até ser entregue ao UA servindo o destinatário. Note que não é aberta conexão direta entre o originador e o destinatário da mensagem; ao contrário, a mensagem é "chaveada" entre MTAs analogamente à comutação de pacotes. MTAs cooperam segundo um protocolo denominado P1 e utilizam o RTSE para a transferência confiável da mensagem entre MTAs. Acesso aos MTAs se dá através de um protocolo denominado P3.

MS (Message Store) é um elemento funcional que atua entre o UA e o MTS. Sua função é armazenar temporariamente a mensagem até ser entregue ao MTS (no caso de envio) ou ao UA (no caso de recepção). Operações de armazenamento e recuperação nos MSs são executadas segundo um protocolo denominado P7. Tais operações são executadas remotamente nos MSs através dos serviços providos pelo ROSE.

MTS (Message Transfer System) é uma coleção de MTAs que atuam cooperativamente no roteamento de mensagens. É análogo a uma rede de comutação de pacotes (onde os MTAs seriam os IMPs).

A figura 8.4 ilustra a interligação dos elementos funcionais descritos acima.

Uma mensagem X.400 é composta de um *envelope* e um *conteúdo* conforme ilustrado na figura 8.5.

O endereçamento X.400 é utilizado tanto para identificar originador e destinatário(s) quanto para rotear as mensagens. Um endereço X.400 é composto dos seguintes campos:

[7]Endereçamento, armazenamento, envio, notificação, etc.

[8]Simple Mail Transfer Protocol.

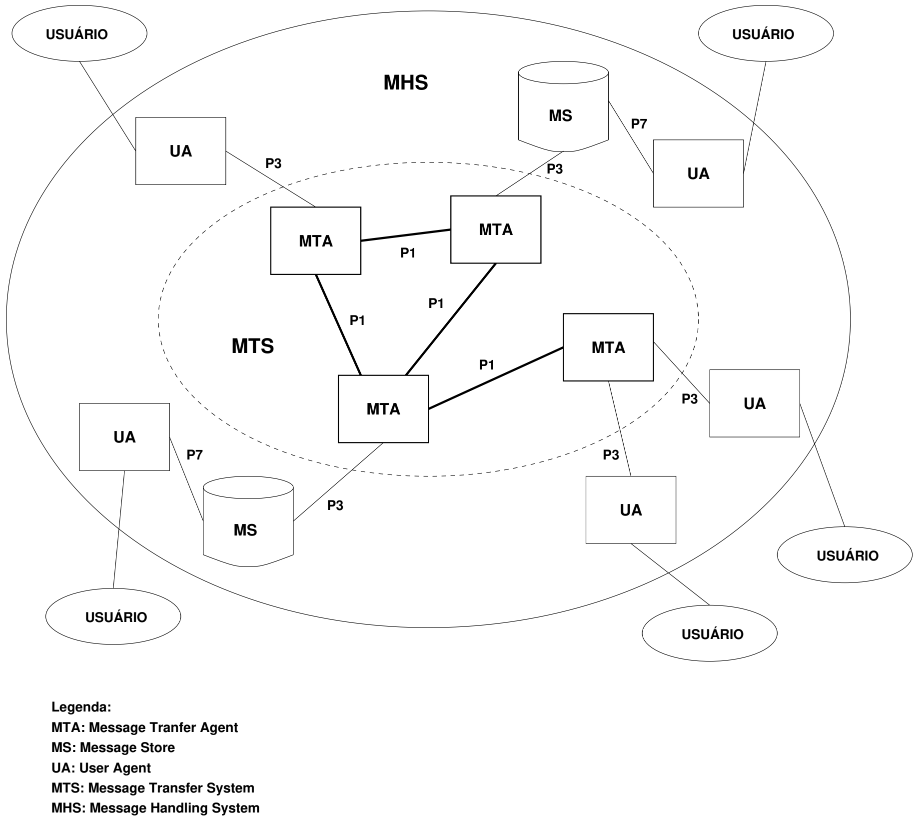

Figura 8.4: Elementos funcionais do MHS.

- Nome de País: código de 2 (ISO 3166) ou 3 (CCITT X.121) caracteres indicando o país do usuário (Ex: BR);
- Nome ADMD (Administrative Management Domain): especifica a organização provedora do serviço ao qual o usuário está conectado (Ex: Embratel);
- Nome PDMD (Private Management Domain): eventual corporação que intermedia o oferecimento do serviço (ex: uma BBS);
- Nome da Organização à qual o usuário pertence;
- Nome da Unidade Organizacional à qual o usuário pertence;
- Nome Pessoal do usuário;
- Atributos próprios do domínio;
- Endereço X.121 do terminal do usuário.

Nem todos os atributos acima podem estar presentes no envelope de uma mensagem. Por exemplo, o endereço do terminal pode suprimir o nome da organização e usuário.

Quanto a segurança e privacidade, o protocolo X.400 cobre três áreas:

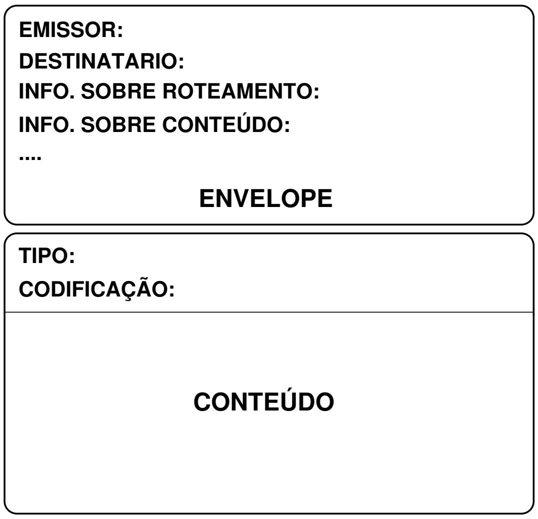

Figura 8.5: Estrutura de mensagens no MHS.

1. confidencialidade: garante que apenas o destinatário terá acesso ao conteúdo da mensagem;

2. autenticação: garante a identidade do originador[9];

3. provas de submissão e entrega: notificações que comprovam o envio/recepção.

8.8 Serviço de Diretório (DS)

DS (Directory Service) é um serviço de base de dados global (DIB: Directory Information Base). Esta base de dados forma um "diretório eletrônico" que armazena objetos tais como pessoas, endereços, senhas, organizações, etc. Este serviço é parte das recomendações da série X.500 da ex-CCITT.

Cada objeto do diretório possui um ou mais *nomes distintos* que servem como chave de acesso para este objeto. Nomes distintos são coleções ordenadas de atributos tais como:

- cn: common name (nome usual);
- ou: organizational unit (unidade organizacional);
- o: organização;
- st: state (estado);
- c: country (país).

[9]Isto é, impede que uma mensagem seja enviada em nome de outrém.

Um objeto é uma sequência de atributos especificado em ASN.1 onde cada atributo possui um identificador de objeto que descreve o tipo de atributo e o(s) correspondente(s) valor(es). Tipos usualmente são *PrintableString* em ASN.1 o que possibilita uma interpretação livre de arquitetura de máquina. A figura 8.6 ilustra um objeto genérico.

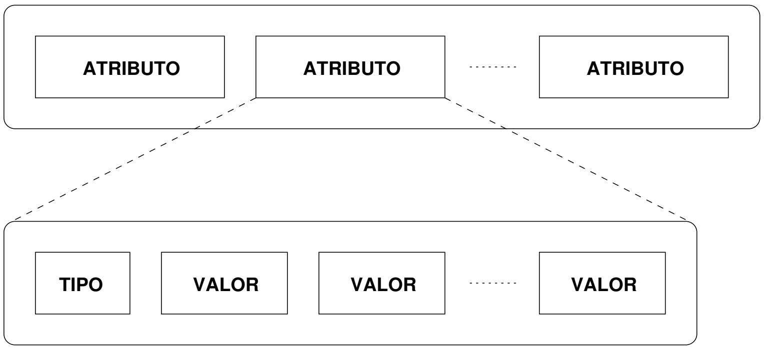

Figura 8.6: Modelo de objeto para o Serviço de Diretório.

Objetos são agrupados em classes, sendo algumas classes padronizadas pela recomendação X.520. O diretório em si é estruturado na DIB na forma de árvore (denominada DIT: Directory Information Tree) conforme ilustrado na figura 8.7.

Do ponto de vista funcional, o DS é composto que quatro componentes. O primeiro, DSA (Directory Service Agent) armazena uma parte do diretório distribuído. O segundo componente, DUA (Directory User Agent) intermedia o acesso ao diretório por parte de um usuário. O terceiro componente, DAP (Directory Access Protocol) viabiliza a interação DUA-DSA; sendo o quarto componente, DSP (Directory Service Protocol) um protocolo de interação DSA-DSA. DAP e DSP são definidos como operações ROSE. Este modelo funcional é ilustrado na figura 8.8.

O DAP é um protocolo tipo requisição-resposta. A requisição carrega tipicamente um nome distinto de um objeto e a resposta o conteúdo deste objeto. Caso o objeto não esteja armazenado no DSA, este pode:

1. retornar o endereço de apresentação de outro DSA "mais próximo" da informação requisitada;

2. encadear a solicitação para outro DSA;

3. difundir a solicitação para um conjunto de DSAs, repassando uma eventual resposta ao DUA que solicitou a informação.

8.9 Serviço de Transferência de Arquivos (FTAM)

FTAM (File Transfer, Access and Management) é um ASE que provê funcionalidades para acesso, transferência e gerência de arquivos remotos, diretórios e *links* simbólicos[10]. O FTAM

[10] Diretórios e links simbólicos foram incorporados como adendos à especificação em 1990.

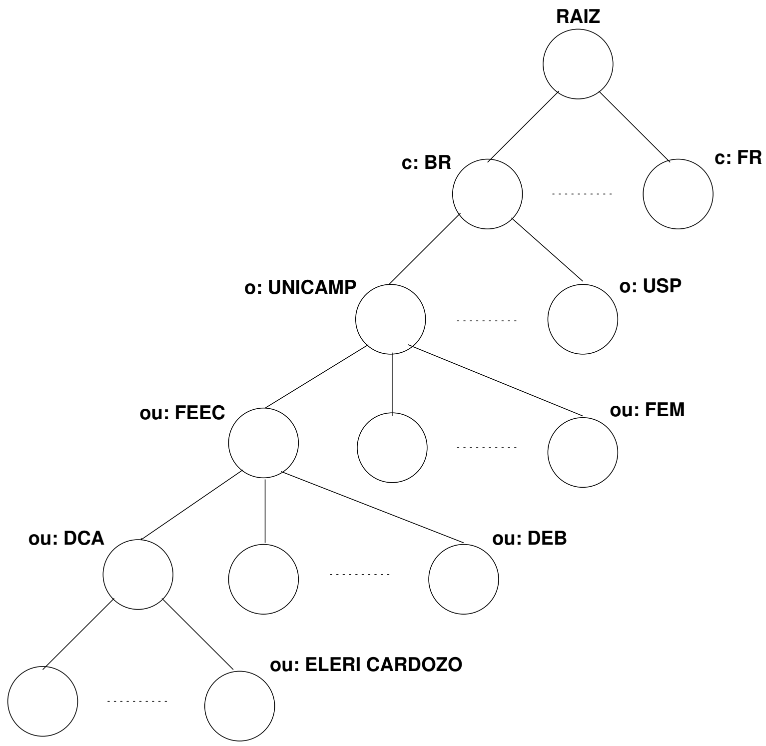

Figura 8.7: DIT (Directory Information Tree).

define um sistema de arquivos "virtual", independente do sistema de arquivos "real" da máquina. As operações FTAM são definidas sobre o primeiro e mapeadas no segundo de acordo com a implementação (figura 8.9).

O FTAM utiliza o ACSE para o estabelecimento de uma sessão de operação sobre arquivos remotos e o CCR para agrupar numa mesma sessão uma sequência de operações.

Um arquivo é composto de zero ou mais unidades de dados (DU: Data Units). As DUs são organizadas em uma estrutura hierárquica (árvore) que determina a inter-relação entre as DUs. Esta estrutura é denominada FADU (File Access Data Unit). Caso o arquivo possua apenas uma FADU, o mesmo é denominado *não-estruturado*; se a hierarquia de FADUs possuir apenas 2 níves, o arquivo é dito *sequencial*; com mais de 2 níveis, tem-se um arquivo *estruturado em blocos*. A figura 8.10 ilustra estas três possibilidades.

O FTAM define nove tipos de arquivos, desde arquivos de texto não estruturados até arquivos especiais para aplicações gráficas.

O FTAM possui 25 primitivas. Seis primitivas são definidas para abertura/fechamento de sessão:

1. F-INITIALIZE: inicia uma sessão FTAM (serviço confirmado);

2. F-TERMINATE: encerra uma sessão FTAM (serviço confirmado);

3. F-U-ABORT: interrompe (aborta) uma sessão FTAM (serviço não confirmado);

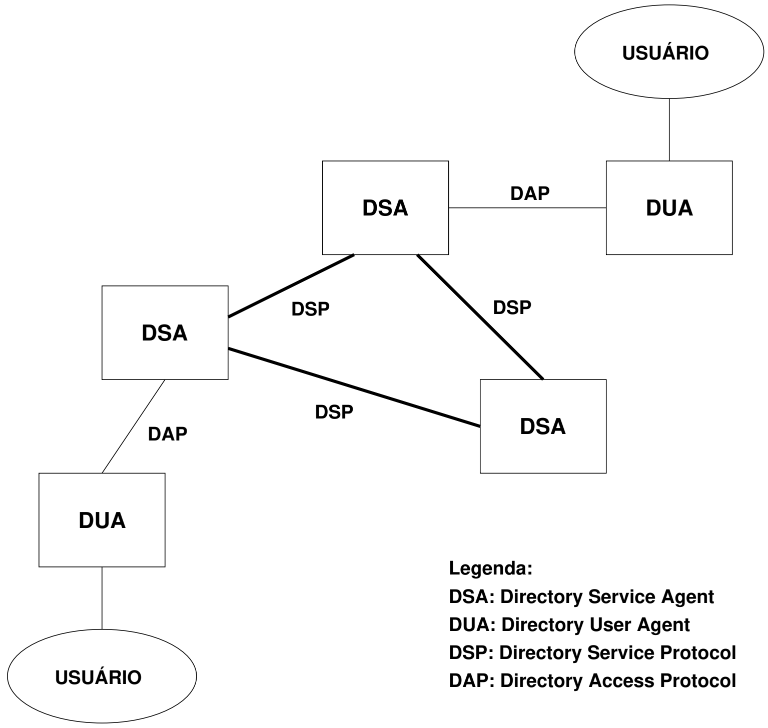

Figura 8.8: Modelo funcional do Serviço de Diretório (DS)

4. F-P-ABORT: o provedor de serviço interrompe (aborta) uma sessão FTAM (primitiva tipo indicação apenas);
5. F-SELECT: seleciona um arquivo para operação (serviço confirmado);
6. F-DESELECT: desfaz uma seleção (serviço confirmado).

Quatro primitivas são definidas para gerenciamento de arquivos (todas as primitivas são serviços confirmados):

1. F-CREATE: cria um novo arquivo;
2. F-DELETE: remove um arquivo.
3. F-READ-ATTRIB: lê os atributos do arquivo (permissões, tipo, etc.);
4. F-CHANGE-ATTRIB: altera os atributos do arquivo.

Seis primitivas são utilizadas para gerenciamento de falhas e atomicidade (todas as primitivas são serviços confirmados):

1. F-BEGIN-GROUP: marca o início de uma operação atômica;
2. F-END-GROUP: sinaliza o término de uma operação atômica;

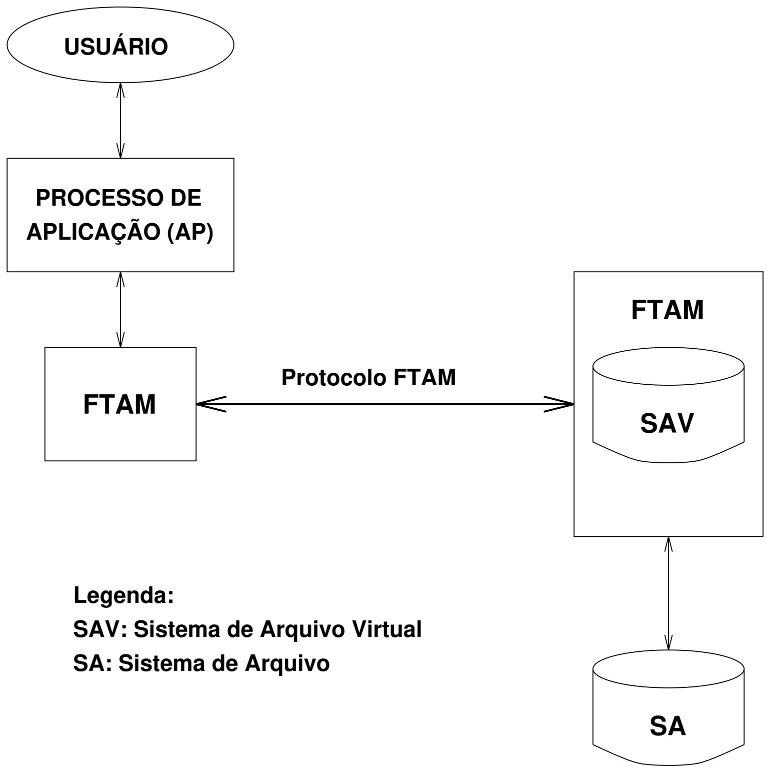

Figura 8.9: FTAM: Sistema de Arquivos Virtual.

3. F-RECOVER: restabelece um regime de transferência após a ocorrência de falha;
4. F-CANCEL: termina adruptamente uma transferência de arquivo;
5. F-CHECK: estabelece um *checkpoint* no regime de transferência[11];
6. F-RESTART: retorna a um *checkpoint* anterior.

Finalmente, nove primitivas oferecem o serviço de transferência propriamente dito:

1. F-OPEN: abre um arquivo para leitura/escrita (serviço confirmado);
2. F-CLOSE: fecha um arquivo aberto (serviço confirmado);
3. F-LOCATE: posiciona o ponteiro de transferência numa FADU específica (serviço confirmado);
4. F-ERASE: destrói uma FADU (serviço confirmado);
5. F-READ: lê uma ou mais FADUs (serviço sem confirmação);

[11] O FTAM não utiliza pontos de sincronização para este serviço.

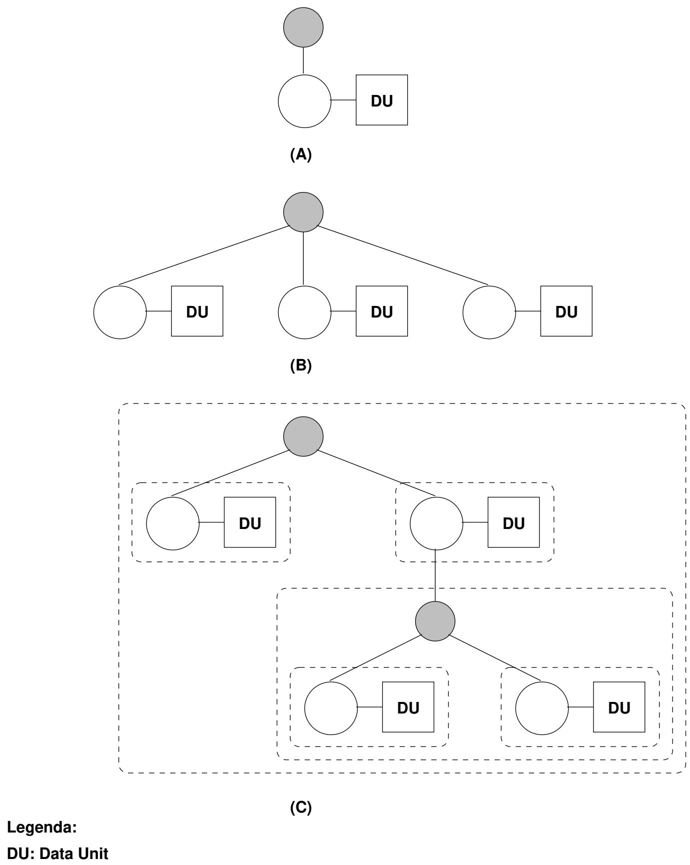

Figura 8.10: FTAM: Arquivos não estruturado (A); sequencial (B); estruturado em blocos (C). As linhas pontilhadas do item (C) definem FADUs.

6. F-WRITE: escreve numa ou mais FADUs (serviço sem confirmação);
7. F-DATA: sinaliza à entidade respondedora a presença de dados (primitiva tipo indicação apenas);
8. F-DATA-END: sinaliza à entidade respondedora o término de trasferência de uma FADU (primitiva tipo indicação apenas);
9. F-TRANFER-END: encerra o regime de transferência (serviço confirmado).

A figura 8.11 ilustra um diagrama de estados típico de operação do FTAM.

8.10 Terminal Virtual (VT)

VT (Virtual Terminal) é um ASE que define terminais virtuais utilizados para interação via rede com aplicações remotas. A necessidade de tal serviço está relacionada à vasta gama de terminais existentes no mercado. Ao invés de incorporar as particularidades de cada

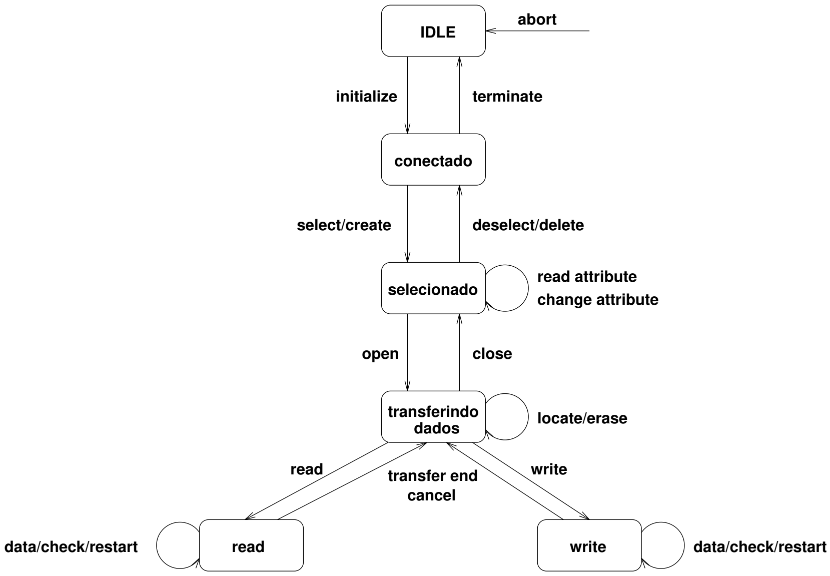

Figura 8.11: FTAM: diagrama de estados típico.

terminal, uma aplicação opera sobre este terminal virtual e uma outra entidade mapeia este dispositivo virtual num dispositivo real. A idéia é similar ao conceito de sistema de arquivos virtual (figura 8.9) do FTAM.

A ISO define várias classes de terminais de acordo com a complexidade dos objetos que o terminal é capaz de manipular. Atualmente, apenas a classe referente aos terminais baseados em caractere está padronizada no âmbito do VT.

O VT é um ASE similar aos descritos anteriormente: utiliza o ACSE e uma conexão de apresentação para o estabelecimento de associação e transporte de PDUs, respectivamente. O VT utiliza também os serviços de sessão para organização do diálogo (via token) caso as entidades estabeleçam comunicação alternada nos dois sentidos (TWA).

O protocolo VT prevê:

1. estabelecimento de associação entre as entidades VT[12];
2. término normal ou adrúpto da associação;
3. transferência de dados pela associação;
4. negociação e renegociação de um perfil de operação para o terminal virtual;
5. gerenciamento do diálogo[13].

As primitivas para abertura e encerramento de associação são idênticas às do ACSE com o prefixo VT: VT-ASSOCIATE, VT-RELEASE, VT-U-ABORT, VT-P-ABORT.

[12]Tipicamente uma vinculada ao terminal do usuário e outra vinculada à aplicação remota.

[13]O protocolo VT define dois modos de operação: no modo A inexiste qualquer forma de gerenciamento, enquanto no modo B o gerenciamento é baseado em *tokens*.

A primitiva principal para transferência de dados, VT-DATA, define um serviço sem confirmação. Um dos parâmetros desta primitiva é a prioridade dos dados: normal, alta ou urgente. Um par de primitivas também sem confirmação, VT-DELIVER e VT-ACK-RECEIPT são utilizadas, respectivamente, para transferência de dados e reconhecimento da recepção.

A negociação do perfil de operação se dá através das seguintes primitivas:

- VT-START-NEG: solicita à outra entidade o início de uma sessão de negociação (serviço confirmado);
- VT-OFFER-NEG: consulta à outra entidade sobre a conveniência de se estabelecer uma sessão de negociação (serviço sem confirmação);
- VT-ACCEPT-NEG/VT-REJECT-NEG: responde afirmativa ou negativamente à consulta acima (serviço sem confirmação);
- VT-END-NEG: termina uma sessão de negociação (serviço confirmado);
- VT-SWITCH-PROFILE: altera o perfil de operação de um terminal virtual (serviço confirmado). Um ponto de sincronismo maior é inserido caso a associação seja reiniciada via VT-BREAK.

O gerenciamento do diálogo (modo B) se dá através das primitivas VT-REQUEST-TOKENS e VT-GIVE-TOKENS, ambas sem confirmação.

Finalmente, o protocolo VT oferece uma primitiva para reiniciar a conexão (via P-RESYNCHRONIZE): VT-BREAK. Este serviço é confirmado.

8.11 Problemas

1. Cite a filosofia de integração entre os ambientes local e OSI.
2. Como a camada de aplicação é estruturada ?
3. Qual a diferença entre *conexão* e *associação* ?
4. Qual a vantagem de se definir objetos de serviço (ASOs) ?
5. Descreva a utilidade do ACSE, ROSE e RTSE.
6. Cite dois empregos típicos de transações atômicas.
7. Desenhe um diagrama temporal para uma transação atômica envolvendo coordenador e 3 participantes. Considere os casos de término normal e anormal *(rollback)*.
8. Descreva funcionalmente o MHS.
9. Descreva funcionalmente o DS.
10. Descreva funcionalmente o FTAM.

Capítulo 9

GERENCIAMENTO DE REDES

Devido à expansão das redes, dois aspectos são fundamentais:

- a rede e os recursos associados têm se tornado indispensáveis às organizações;
- aumento da probabilidade de comportamento errado devido a falha em parte da rede ou degradação do desempenho a níveis inaceitáveis.

Uma rede de maior complexidade não pode ser gerenciada somente com o esforço humano, necessitando que ferramentas automáticas de gerenciamento sejam utilizadas. A dificuldade de fornecimento destas ferramentas aumenta se a rede inclui equipamentos de vários fabricantes. Estas dificuldades estão presentes nos níveis das redes locais (LAN - Local Area Network), redes metropolitanas (MANs - Metropolitan Area Networks) e redes de longa distância (WAN - Wide Area Networks). Entretanto, as redes locais e, em boa parte, as redes metropolitanas, encontram-se no centro de qualquer estratégia de rede de uma empresa e devem ser uma das preocupações principais de todo programa de gerenciamento de rede. Para gerenciar uma rede é fundamental o conhecimento sobre o status e comportamento da rede. Neste sentido, é necessário um sistema de gerenciamento de rede que inclua um conjunto compreensível de ferramentas de captura de dados e controle integrado ao software e hardware da rede.

9.1 Requisitos do Gerenciamento de Rede

A ISO (International Standard Organization) lista as seguintes áreas como chaves no gerenciamento de redes:

- gerenciamento de falhas;
- gerenciamento de contabilidade;
- gerenciamento de configuração e nome;
- gerenciamento de desempenho;
- gerenciamento de segurança.

Cada uma destas gerências encontra-se discutida na sequência.

9.1.1 Gerenciamento de Falha

De modo a manter uma operação adequada de uma rede complexa, cuidados devem ser tomados de modo a que o sistema como um todo, e cada componente essencial individualmente, encontrem-se operando em ordem. Quando da ocorrência de uma falha é importante que as seguintes ações sejam tomadas o mais rapidamente possivel:

- determinar exatamente o local da falha; isolar o resto da rede da falha de modo que continue a funcionar sem interferência;
- reconfigurar ou modificar a rede de modo a minimizar o impacto na operação sem a presença do(s) componente(s) em falha;
- reparar ou substituir o(s) componente(s) com problema permitindo à rede o retorno ao estado inicial.

Importante na questão da definição do gerenciamento de falha é o próprio conceito de falha. Esta deve ser distinguida do erro. Uma falha é uma condição anormal que requer atenção (ou ação) do gerenciamento para reparação. Uma falha é normalmente indicada por uma operação incorreta ou por um número excessivo de erros. Por exemplo, no caso de uma linha de comunicação interrompida, nenhum sinal flui através da linha, ou ainda, uma emenda em um cabo pode causar fortes distorções de modo a provocar uma taxa elevada e persistente de erro no nível de bit. Certos erros, por outro lado, podem ocorrer ocasionalmente e não são normalmente considerados como falha, por exemplo, o erro em um único bit em uma linha de comunicação. Em geral, é possível compensar a ocorrência de erros através de mecanismos de controle de erros presentes nos protocolos de comunicação.

Requisitos do Usuário

Os usuários esperam uma solução rápida e confiável do problema. Muitos usuários finais poderão, eventualmente, tolerar perdas ocasionais. Entretanto, quando da ocorrência não frequente destas perdas, o usuário espera receber uma notificação e a solução quase que imediata do problema Para fornecer este nível de resolução de falhas, é necessária a disponibilidade de funções de gerenciamento rápidas e confiáveis voltadas para deteção de falhas e os respectivos diagnósticos. O impacto e a duração das falhas também podem ser minimizados pelo uso de componentes redundantes e rotas de comunicação alternativas com o objetivo de fornecer à rede um alto grau de tolerância a falhas. A própria capacidade de gerenciamento de falhas deve ser redundante a fim de aumentar a confiabilidade da rede. Os usuários esperam ser informados do status da rede, incluindo-se a programação da manutenção da rede, como também , a manutenção não programada e feita em função da ocorrência de problemas na rede. Após correção de uma falha e restauração do sistema à sua condição plena de operação, o serviço de gerenciamento de falhas deve garantir que o problema foi resolvido e que nenhum outro problema foi introduzido. Assim como em outras áreas do gerenciamento de redes, o gerenciamento de falhas deve ter um efeito mínimo no desempenho da rede.

9.1.2 Gerenciamento de Contabilidade

Em muitas redes corporativas, divisões ou projetos são cobrados pelo uso dos serviços de rede. Estes procedimentos internos de contabilidade não envolvem, em geral, transferência real de fundos, mas são importantes para a empresa. Mesmo que este tipo de contabilidade interna não seja empregada, o gerente da rede deve ser capaz de acompanhar o uso dos recursos da rede por parte de um usuário ou de grupos de usuários com as seguintes finalidades:

- um usuário, ou grupo de usuários, pode estar abusando graças a um acesso privilegiado e, desta maneira, sobrecarregando a rede em prejuízo de outros usuários;
- usuários podem estar fazendo um uso ineficiente da rede; o gerente da rede poderá assisti-los na mudança de procedimentos visando uma melhoria no desempenho.

O gerente da rede estará em uma melhor posição para planejar uma eventual expansão da rede se as atividades dos usuários forem conhecidas em detalhe suficiente.

Requisitos do Usuário

O gerente da rede deve ser capaz de especificar os tipos de informação de contabilidade que deverão ser armazenados nos vários nós, o intervalo desejado de envio das informações armazenadas aos níveis superiores de gerenciamento e os algoritmos a serem utilizados nos cálculos de carga do uso da rede pelos usuários. Relatórios de contabilidade deverão ser gerados sob controle do gerente de rede. Com o objetivo de limitar o acesso às informações de contabilidade, o sistema deve ser capaz de verificar a autorização de acesso e manipulação da informação por parte dos usuários.

9.1.3 Gerenciamento de Nome e Configuração

As redes modernas de comunicação são formadas de componentes individuais e subsistemas lógicos que podem ser configurados para suportar muitas aplicações diferentes. O mesmo dispositivo pode ser configurado para agir como um roteador, como um nó do sistema ou ambos. Uma vez decidido como o dispositivo será usado, o gerente de configuração pode escolher o software apropriado e o conjunto de atividades e valores (ex: tempo de retransmissão na camada de transporte) para aquele dispositivo. O gerente de configuração é responsável pela iniciação da rede e pela parada controlada da rede ou parte da rede. Ainda sob a sua responsabilidade encontram-se a manutenção, adição e atualização das relações entre os componentes e o estado destes componentes durante a operação da rede.

Requisitos do Usuário

As operações de partida e parada em uma rede são responsabilidades específicas do gerente de configuração. O gerente deve ser capaz de identificar os componentes da rede e definir a conectividade desejada destes componentes. Quando a rede é configurada sempre com o mesmo conjunto de atributos é necessário poder definir e modificar os atributos default, como também, carregar estes conjuntos pré-definidos de atributos em componentes específicos da rede. O gerente deve ser capaz de alterar a conectividade dos componentes da rede quando houver alteração nas necessidades do usuário. Reconfiguração de uma rede é geralmente desejável em resposta à avaliação de desempenho ou para suportar expansão da rede, recuperação de falha e teste de segurança. Usuários geralmente necessitam, ou desejam ser informados do estado dos recursos e componentes da rede. Portanto, na ocorrência de mudanças na configuração, os usuários deverão ser notificados desta alteração. Relatórios de configuração podem ser gerados de forma periódica ou em resposta a uma requisição. Gerentes de rede aceitam somente que usuários autorizados (operadores) gerenciem e controlem a operação de rede (ex: distribuição e atualização de software).

9.1.4 Gerenciamento de Desempenho

As redes de computadores modernas são compostas de muitos componentes variados que necessitam se intercomunicar e compartilhar dados e recursos. Em muitos casos a efetividade da aplicação requer que a comunicação via rede esteja dentro de certos limites de desempenho. O gerenciamento de desempenho de uma rede de computadores compreende duas amplas categorias: monitoramento e controle. Monitoramento é a função que "persegue" as atividades na rede. A função de controle permite o gerenciamento de desempenho da rede. Algumas das questões de desempenho relacionadas ao gerenciamento da rede são:

- qual o nível da capacidade de utilização?
- existe tráfego excessivo?
- o throughput tem sido reduzido a níveis aceitáveis?
- o tempo de resposta está aumentando?

Para responder a estas questões, o gerente de rede deve focalizar um conjunto inicial de recursos para ser monitorado de modo a estimar os níveis de desempenho. Isto inclui a associação de métricas e valores adequados a recursos importantes da rede como indicadores dos diferentes níveis de desempenho. Por exemplo, qual o número de retransmissões em uma conexão de transporte que deve indicar problemas de desempenho necessitando de atenção? O gerenciamento de desempenho deve, portanto, monitorar muitos recursos para fornecer informações a respeito de um determinado nível de operação da rede. Através da coleção e análise destas informações e do uso do resultado desta análise como realimentação ao conjunto de valores recomendados, o gerente da rede pode adquirir cada vez mais a perícia para reconhecer situações indicativas de degradação do desempenho.

Requisitos do Usuário

Antes de usar uma rede para uma aplicação particular, um usuário pode querer conhecer informações como os tempos médios e de pior caso e a confiabilidade dos serviços de rede. O desempenho deve ser conhecido em detalhe suficiente para poder responder questões específicas dos usuários. Os usuários finais esperam que os serviços de rede sejam gerenciados de forma que possam proporcionar, consistentemente, bons tempos de resposta às suas aplicações. Os gerentes de rede necessitam de estatísticas de desempenho que os auxiliem a planejar, gerenciar e manter redes complexas. Estatísticas de desempenho podem ser usadas para reconhecer potenciais gargalos antes que estes causem problemas aos usuários finais. Ações corretivas podem então ser tomadas. Estas ações podem traduzir-se na mudança de tabelas de roteamento para balancear ou redistribuir a carga do tráfego durante períodos de pico ou quando um gargalo é identificado pelo crescimento muito rápido da carga em uma determinada área. A longo prazo, a capacidade de planejamento baseada em tais informações de desempenho pode indicar a decisão, por exemplo, de expansão das linhas naquela área.

9.1.5 Gerenciamento de Segurança

O gerenciamento de segurança está associado com a geração, distribuição e armazenamento de chaves de criptografia. Password e outras informações de controle de acesso devem ser mantidas e distribuídas. Gerenciamento de segurança também encontra-se associado com o monitoramento e controle de acesso a redes de computadores e acesso a toda, ou parte, da informação de gerenciamento obtida de nós da rede. Arquivos de registro são uma ferramenta de segurança importante o que faz com que o gerenciamento de segurança envolva a coleção, armazenamento e exame de arquivos de segurança, assim como, a habilitação e desabilitação destas facilidades de registros (logs).

Requisitos do Usuário

O gerenciamento de segurança fornece facilidades para proteção de recursos da rede e da informação do usuário. Facilidades de segurança de rede devem ser disponíveis somente para usuários autorizados. Os usuários desejam saber que políticas de segurança estão implantadas na rede e que o gerenciamento das facilidades de segurança também é seguro.

9.2 Sistemas de Gerenciamento de Rede

Um sistema de gerenciamento de rede é uma coleção de ferramentas para monitoramento e controle da rede com as seguintes características:

- uma única interface de operador com um poderoso e amigável conjunto de comandos para realização da maioria das tarefas de gerenciamento;
- uma quantidade mínima de equipamentos separados, ou seja, a maioria do hardware e do software requerido para o gerenciamento da rede deve ser incorporado no equipamento existente do usuário.

Um sistema de gerenciamento de redes consiste de adições incrementais de hardware e software implementados entre os componentes existentes na redes. O software utilizado nas tarefas de gerenciamento reside nos computadores nós da rede e nos processadores de comunicação (ex: processadores front-end, controladores de terminal, pontes, roteadores, etc.). Um sistema de gerenciamento de rede deve ser projetado para enxergar a rede como uma arquitetura unificada, com endereços e nomes atribuídos a cada ponto e conhecer os atributos e ligações de cada elemento no sistema. Os elementos ativos da rede fornecem regularmente uma realimentação da informação do estado para o centro de controle da rede. A figura a seguir sugere a arquitetura de um sistema de gerenciamento de rede. Cada nó da rede contém uma coleção de software voltado para a tarefa de gerenciamento de rede, indicada na figura como entidade de gerenciamento de rede (Network Management Entity - NME). Cada NME realiza as seguintes tarefas:

- coleciona estatísticas de comunicação e atividades relacionadas à rede;
- armazena estatística local;
- responde aos comandos do centro de controle da rede, incluindo:
  - comandos para transmissão ao NCC de estatísticas coletadas;
  - alteração de parâmetros (ex: temporizador utilizado no protocolo de transporte);
  - fornecimento de informações sobre estado de componentes (ex: valores de parâmetros, ligações ativas) e geração de tráfego artificial para realização de testes.

Pelo menos um hospedeiro no sistema é designado como centro de controle da rede. Além do software NME, o hospedeiro que atua como controle da rede inclui uma coleção de softwares denominada Network Control Center (NCC). O NCC inclui uma interface de operador para permitir a um usuário autorizado o gerenciamento da rede. O NCC responde a comandos do usuário através de informações enviadas para o vídeo, ou então, comandos são dirigidos às NMEs através da rede. Esta comunicação é realizada via protocolo de gerenciamento de rede no nível da camada de aplicação, empregando uma arquitetura de comunicação da mesma forma que qualquer outra aplicação distribuída.

Algumas observações são importantes:

1. como o software de gerenciamento de rede baseia-se no sistema operacional do hospedeiro e na arquitetura de comunicação, muitos sistemas disponíveis até recentemente eram projetados para serem empregados em equipamentos de um único fabricante. No caso de computadores pessoais, existe um grande número de pacotes de software de gerenciamento que permitem a integração de computadores pessoais de vários fabricantes. Padrões nesta área ainda estão imaturos, entretanto, sistemas de gerenciamento de redes baseados em padrões de modo a permitir o gerenciamento de equipamentos de vários fabricantes estão tornando-se mais comuns;

2. como mostrado na figura anterior, o centro de controle de rede comunica-se e controla, inicialmente, monitores de software em outros sistemas. A arquitetura pode ser estendida para incluir elementos de hardware especializados para função de monitoramento e controle;

3. para permitir uma alta disponibilidade da função de gerenciamento da rede, dois ou mais centros de controle são utilizados. Em operação normal, um dos centros encontra-se inativo, ou simplesmente coletando estatísticas, enquanto o outro é utilizado para controle. Caso o centro de controle de rede primário falhe, o sistema de back-up poderá ser usado.

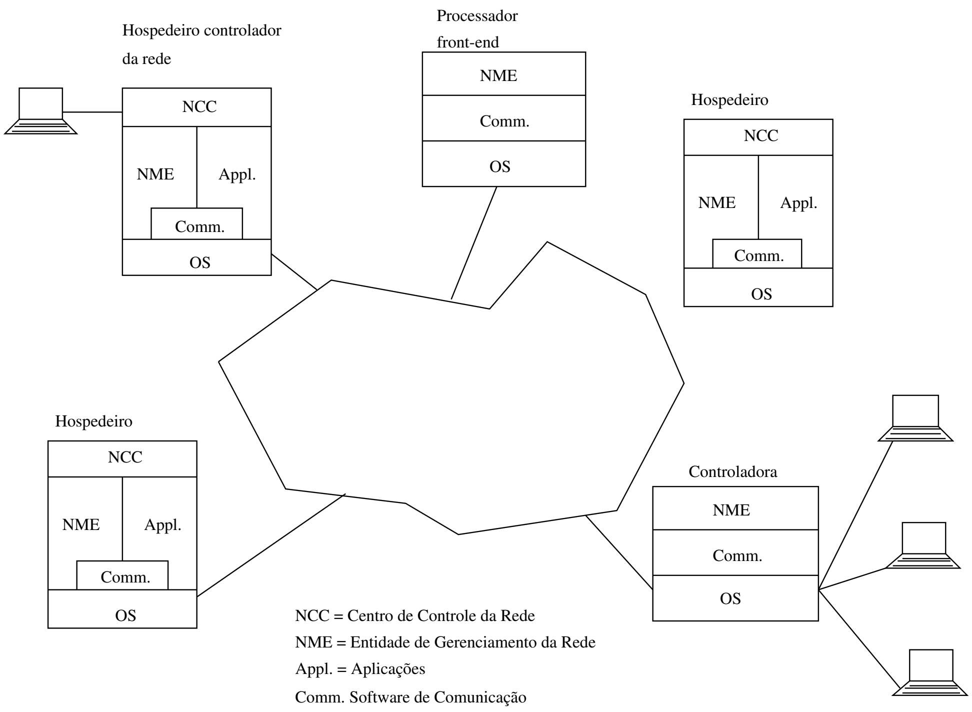

Figura 9.1: Elementos de um Sistema de Gerenciamento de Rede

9.3 Gerenciamento de Rede OSI

A ISO (International Organization for Standardization) é responsável por um conjunto de padrões de gerenciamento de rede. O primeiro padrão relacionado ao gerenciamento de rede foi denominado ISO 7498-4 e especifica o contexto de gerenciamento para o modelo OSI. Este documento indica que o gerenciamento OSI suporta os seguintes requisitos dos usuários:

- atividades que permitem aos gerentes planejar, organizar, supervisionar, controlar e contabilizar o uso de serviços de interconexão;
- habilidade de responder aos requisitos de mudança;
- facilidades que garantam comportamento previsível na comunicação;
- facilidade para proteção da informação e autenticação da origem e do destino dos dados transmitidos.

A partir deste documento, a ISO tem publicado um conjunto volumoso de padrões voltados ao gerenciamento de rede. O antigo CCITT (Consultative Comittee on Interna-tional Telegraphy and Telephony), atualmente ITU-TS (Telecommunications Strandardization Sector), e a ISO atuam conjuntamente nestes esforços de padronização sendo o primeiro responsável pela série de recomendações X.700. Os padrões situam-se em cinco categorias:

1. visão geral e contexto do gerenciamento OSI: inclui o documento OSI 7498-4 que fornece uma introdução geral aos conceitos de gerenciamento e o ISO 10040, que fornece uma visão geral dos documentos restantes;

2. CMIS/CMIP: define os serviços de informação de gerenciamento comuns (Common Management Information Service- CMIS), necessários às aplicações de gerenciamento OSI, e o protocolo comum de gerenciamento da informação (Common Management Information Protocol - CMIP), protocolo este que fornece a capacidade de troca de informações necessárias ao suporte do CMIS;

3. funções do sistema de gerenciamento: define as funções específicas que são realizadas pelo sistema de gerenciamento OSI;

4. estrutura da informação gerenciada: define a base de informações de gerenciamento (Management Information Base - MIB), a qual contém uma representação de todos os objetos do ambiente OSI sujeitos ao gerenciamento;

5. gerenciamento de camada: define as informações, serviços e funções do gerenciamento relacionado às várias camadas do modelo OSI.

9.3.1 Contexto do Gerenciamento OSI

A figura na sequência mostra a arquitetura OSI para o gerenciamento de rede.

Os elementos chaves desta arquitetura incluem:

1. processo de aplicação (System Management Aplication Process - SMAP): é o software local responsável pela execução das funções de gerenciamento de rede em um único sistema (hospedeiro, processador front-end, roteador, etc.). Possui acesso aos parâmetros do sistema e coordena ações com os outros SMAPs do sistema;

2. entidade de aplicação (System Management Application Entity - SMAE): é responsável pela comunicação com outros nós, especialmente com o centro de controle da rede (NCC). Um protocolo padrão, CMIP (Common Management Information Protocol) é utilizado para esta facilidade. A entidade de aplicação (SMAE) fornece um serviço comum de informações de gerenciamento (Common Management Information Service - CMIS);

3. entidade de gerenciamento de camada (Layer Management Entity - LME): lógica é acrescida a cada camada da arquitetura OSI para fornecer funções de gerenciamento específicas de cada camada;

4. base de informações de gerenciamento (Management Information Base - MIB): coleção de informações, situada em cada nó, pertencentes ao gerenciamento da rede.

Os elementos ilustrados na figura anterior devem ser implementados de forma distribuída em todos os sistemas que estão sujeitos ao gerenciamento de rede. As interações que ocorrem entre os sistemas estão ilustradas, de forma abstrata, na figura a seguir.

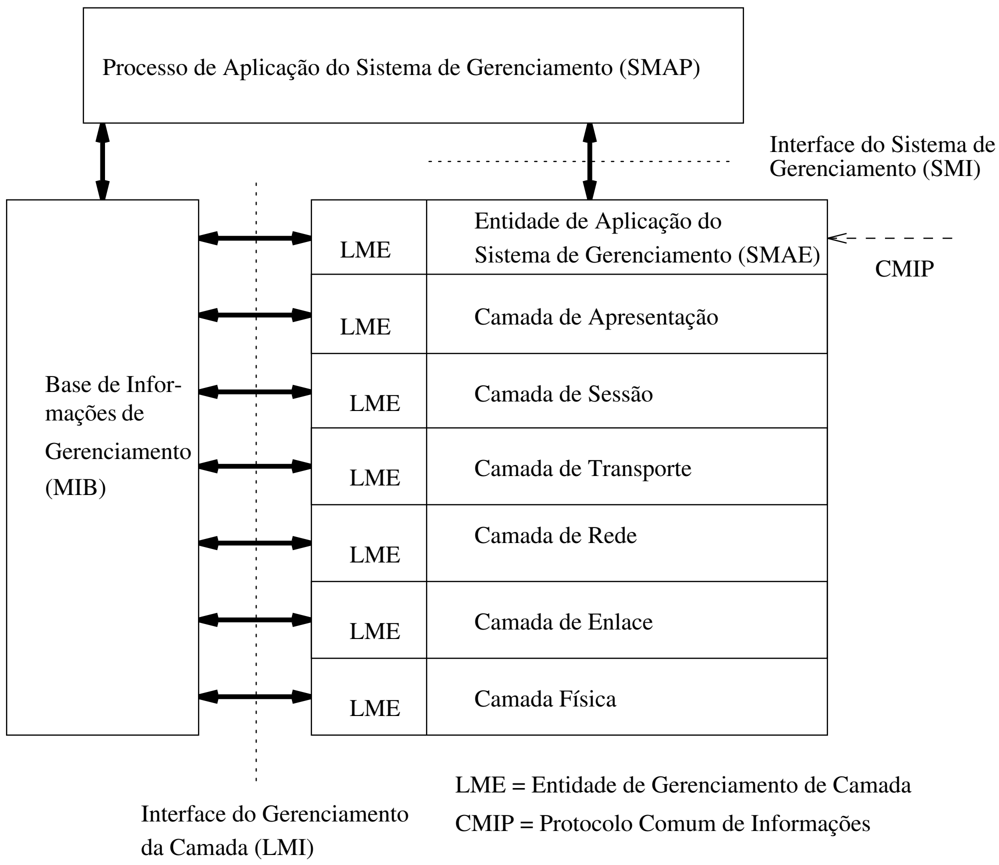

Figura 9.2: Modelo da Arquitetura do Gerenciamento OSI

As atividades de gerenciamento são realizadas através da manipulação de objetos gerenciados. Cada sistema contém um certo número destes objetos. Cada objeto é uma estrutura de dados que corresponde a uma entidade a ser gerenciada. O SMAP presente em um sistema pode possuir um dentre dois possíveis papéis: papel de agente ou papel de gerente. O SMAP possui o papel de gerente em um sistema que atua como gerente de rede ou centro de controle da rede. O gerente envia requisições de informação e comandos de operação para execução nos sistemas gerenciados. Em cada sistema gerenciado o processo agente, responsável pela gerência dos objetos locais, interage com o gerente.

9.3.2 Funções de Gerenciamento

Um conjunto de padrões tem sido publicado na categoria denominada funções de gerenciamento (Systems Management Functions - SMF). Cada padrão SMF define a funcionalidade necessária para suportar os requisitos da área funcional do gerenciamento do sistema (System Management Function Area - SMFA). As áreas funcionais, descritas anteriormente, são:gerenciamento de falha,gerenciamento de contabilidade, gerenciamento de configuração, gerenciamento de desempenho e gerenciamento de segurança. Uma determinada SMF pode

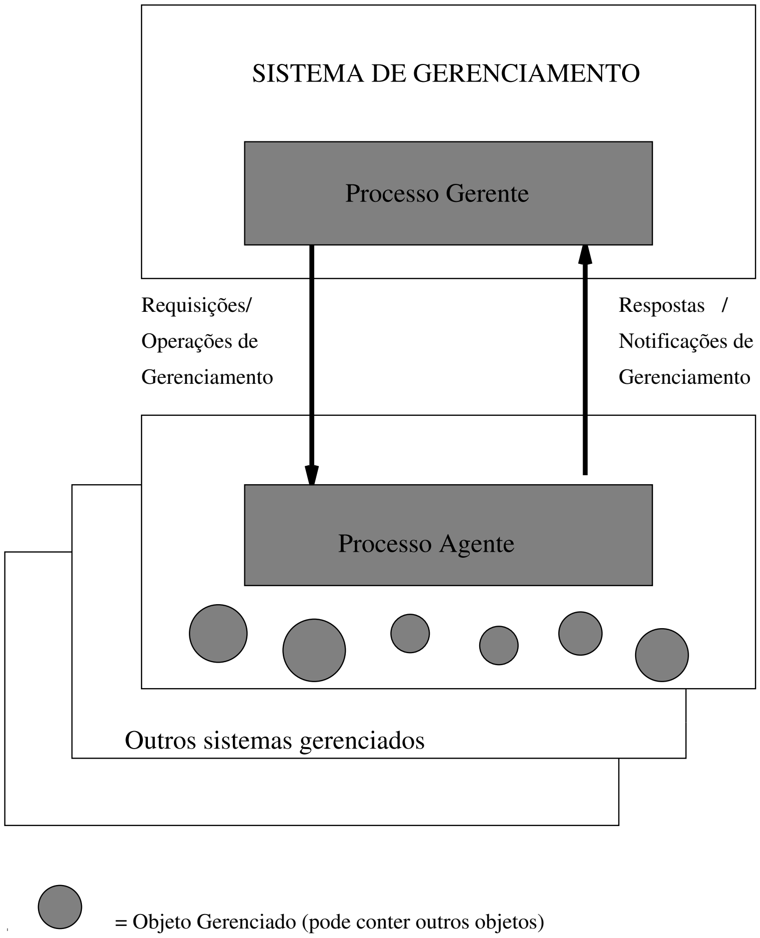

Figura 9.3: Interações no Sistema de Gerenciamento

suportar requisitos em uma ou mais das cinco SMFAs, por exemplo: a função de gerenciamento voltada à elaboração de eventos pode ser aplicável a todas as SMFAs . Cada uma das SMFAs requer várias SMFs. Cada padrão SMF define a funcionalidade da SMF e fornece um mapeamento entre os serviços fornecidos pela SMF e os serviços de informação de gerenciamento comum (CMIS). A figura a seguir ilustra estas relações.

9.3.3 Estrutura da Informação de Gerenciamento

A base da atividade de gerenciamento dos sistemas é a base de informações de gerenciamento (MIB) que possui a representação de todos os recursos do sistema de gerenciamento. A estrutura da informação de gerenciamento (Structure of Management Information - SMI) define o contexto geral na qual a MIB pode ser definida e construida. O SMI identifica os tipos de dados que podem ser usados na MIB e como os recursos dentro da MIB são representados e nomeados. O gerenciamento de sistemas OSI baseia-se fortemente nos conceitos de projeto orientado a objetos. Cada recurso monitorado e controlado pelo gerenciamento OSI é

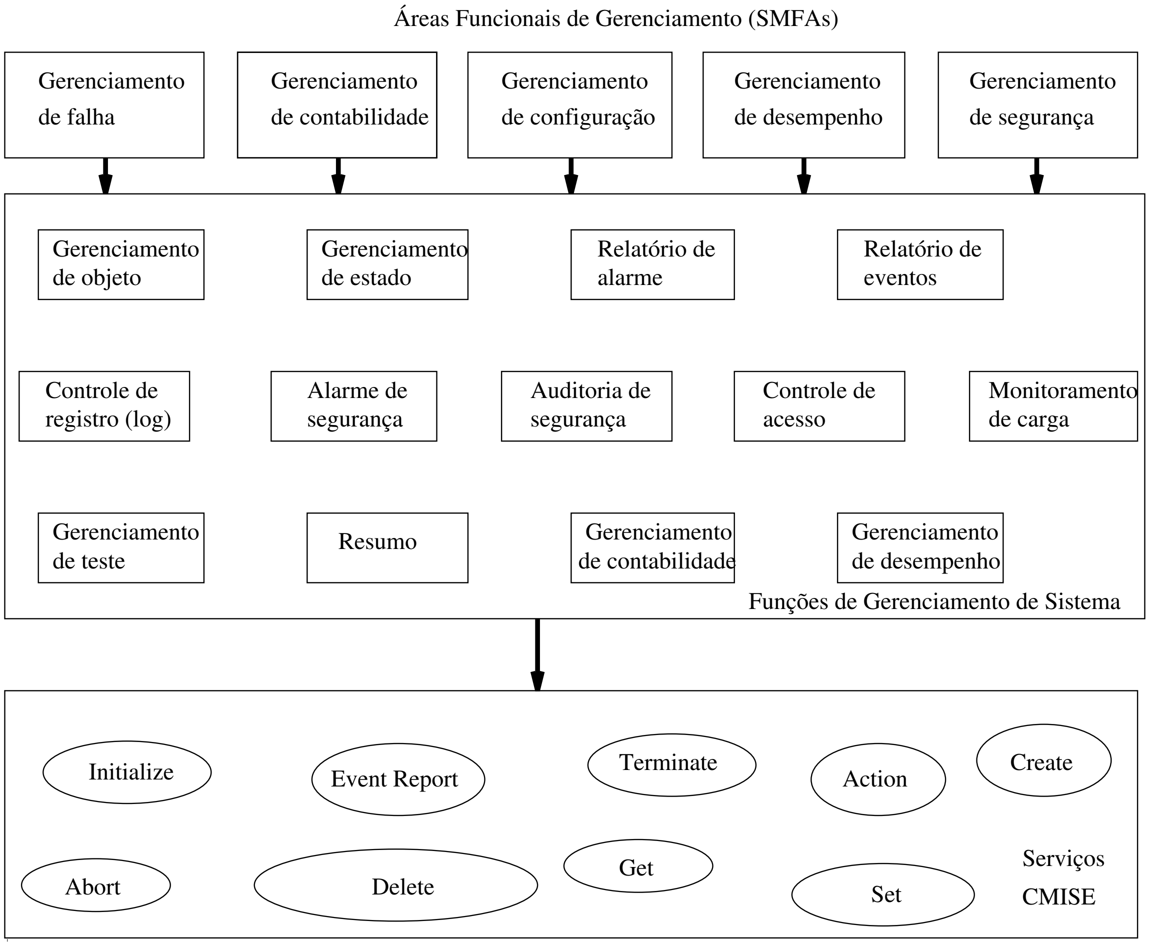

Figura 9.4: Visão Geral dos Sistemas de Gerenciamento

representado por um objeto gerenciado. Este objeto pode ser definido para qualquer recurso que uma organização deseje monitorar e/ou controlar. Exemplos de recursos de hardware são: chaves (switches), estações de trabalho, PBXs, porta de placa de rede local, multiplexadores, etc. Exemplos de recursos de software são: algoritmos de roteamento, rotinas de gerenciamento de armazenamento, etc. Objetos gerenciados relativos a recursos específicos de uma camada são denominados objetos gerenciados da camada-N. Objetos gerenciados relativos a recursos que englobam mais de uma camada denominam-se objetos gerenciados de sistema. Um objeto gerenciado é definido em termos dos atributos que ele possui, das operações que podem ser realizadas sobre ele, das notificações que ele pode enviar e das relações com outros objetos gerenciados. De modo a estruturar a definição de uma MIB, cada objeto gerenciado é uma instância de uma classe de objeto gerenciado. Esta classe é um modelo para as instâncias do objeto gerenciado que compartilham os mesmos atributos, notificação e operações de gerenciamento. A definição de uma classe de objeto gerenciado consiste em:

- atributos visíveis na interface do objeto;
- operações de gerenciamento que podem ser aplicadas ao objeto;
- comportamento exibido pelo objeto gerenciado em resposta às operações de gerenciamento;
- notificações que podem ser emitidas pelo objeto.

Atributos: os dados reais contidos em um objeto gerenciado são denominados atributos. Cada atributo traduz uma propriedade de recurso que o objeto representa como, por exemplo, características operacionais, estado atual ou condições de operação. O tipo de dado de um atributo pode ser inteiro, real, booleano, cadeia de caracteres ou algum tipo composto construído a partir de tipos básicos. Um atributo pode ter um único valor ou um conjunto de valores. Em adição ao tipo de dado, cada atributo possui regras de acesso (leitura, escrita, leitura/escrita) e regras pelas quais ele pode ser localizado como resultado de uma busca.

Operações: as operações de gerenciamento aplicam-se aos atributos de um objeto ou ao objeto gerenciado como um todo. Uma operação realizada em um objeto gerenciado pode ter sucesso somente se o sistema de gerenciamento que está evocando a operação possui autorização necessária para realizar a operação e as restrições de consistência não são violadas.

Comportamento: um objeto gerenciado exige certas características de comportamento, incluindo a forma como o objeto reage a operações realizadas sobre ele e as restrições associadas ao seu comportamento. O comportamento do objeto ocorre em resposta a um estímulo tanto externo como interno. Estímulos externos traduzem-se em operações de gerenciamento liberadas na forma de mensagens CMIP. Estímulos internos são eventos internos ao objeto gerenciado e aos seus recurso tal como, por exemplo, o temporizador associado ao objeto.

Notificações: objetos gerenciados emitem notificações quando alguma ocorrência interna ou externa que afete o sistema é detetada. Notificações podem ser transmitidas externamente através de um protocolo ou então ser arquivadas (logged). Sistemas de gerenciamento podem requisitar que algumas, ou todas, as notificações emitidas por um objeto gerenciado lhes sejam enviadas. Notificações que são enviadas a um gerente estão contidas em um relatório de evento (event report).

9.3.4 Serviço Comum de Informação de Gerenciamento (CMIS)

O CMIS define os serviços suportados pelo gerenciamento de sistemas OSI. Estes serviços são evocados pelos processos de gerenciamento para comunicar remotamente. Estes serviços são de dois tipos: serviços confirmados e serviços não-confirmados. No primeiro caso, é necessário que o processo de gerenciamento remoto envie uma resposta para indicar a recepção e o sucesso/falha da operação requisitada; no segundo caso não há o envio de respostas. Três categorias de serviços são relevantes para o CMIS:

1. serviços de associação: os usuários do CMIS necessitam estabelecer uma associação de aplicação para comunicação. O usuário CMIS baseia-se, neste caso, no elemento de serviço de controle de associação (Association Control Service Element- ACSE), que é uma entidade de aplicação OSI separada, para controle das associações de aplicação;

2. serviço de notificação de gerenciamento: este serviço é usado para transportar informação de notificação. A definição de uma notificação e o consequente comportamento das entidades de comunicação são dependentes da especificação do objeto gerenciado que gerou a notificação, especificação esta que se encontra fora do escopo do CMIS. Primitiva do serviço de notificação:

   Serviço: M-EVENT-REPORT; Tipo: confirmado/não confirmado; Definição: relata um evento sobre um objeto gerenciado a um usuário par do serviço CMIS.

3. serviço de operação de gerenciamento: existem seis serviços que são utilizados para transportar informações de gerenciamento associadas às operações de gerenciamento. A definição da operação e o consequente comportamento das entidades de comunicação são dependentes da especificação do objeto gerenciado para o qual a operação é dirigida estando também fora do escopo do CMIS. Primitivas de serviços de operação:

   Serviço: M-GET; Tipo: confirmado, Definição: requisita informação de gerenciamento de um usuário par do serviço CMIS;

   Serviço: M-SET; Tipo: confirmado/não confirmado; Definição : requisita a modificação da informação de gerenciamento por parte de um usuário par de um serviço CMIS;

   Serviço: M-ACTION; Tipo: confirmado/não confirmado, Definição: requisita a um usuário par do serviço CMIS a realização de uma ação;

   Serviço: M-CREATE; Tipo: confirmado, Definição: requisita a um usário par do serviço CMIS para criar uma instância de um objeto gerenciado;

   Serviço: M-DELETE; Tipo: confirmado, Definição: requisita a um usuário par do serviço CMIS para deletar uma instância de um objeto gerenciado;

   Serviço: M-CANCEL-GET; Tipo: confirmado, Definição: requisita a um usuário par do serviço CMIS para cancelar uma requisição previamente solicitada e aguarda a evocação de um serviço M-GET;

O CMIS fornece 2 facilidades de estruturação:

1. múltiplas respostas a uma operação confirmada podem ser associadas a uma operação através do uso de um parâmetro de identificação de ligação;

2. operações podem ser realizadas em múltiplos objetos gerenciados, selecionados segundo algum critério e seguindo a condição de sincronização.

As primitivas de serviço M-GET, M-SET, M-ACTION e M-DELETE incluem um parâmetro para selecionar o(s) objeto(s) a ser(em) usado(s) na operação. A seleção do(s) objeto(s) envolve três conceitos: escopo, filtragem e sincronização. Estas três facilidades são fornecidas como parâmetros. **Escopo:** refere-se à identificação de um objeto (ou objetos) para o qual um filtro é aplicado. O escopo refere-se a uma base de objetos gerenciados. Esta base é o ponto de partida para a seleção de um ou mais objetos a partir da aplicação do filtro. Usando a base de objeto como raiz, uma sub-árvore é obtida a partir da árvore global de objetos;

Filtragem: trata-se de uma expressão booleana, contendo uma ou mais asserções sobre a presença ou valores de atributos de um objeto gerenciado. Cada asserção pode ser um teste de igualdade, ordem, presença ou comparação de um conjunto. A filtragem é aplicada em todos os objetos gerenciados selecionados pelo parâmetro do escopo e somente os objetos que atendem ao filtro são selecionados para a realização da operação **Sincronização:** o parâmetro de escopo pode resultar na seleção de mais de um objeto gerenciado a ser submetido à filtragem. A questão que se coloca diz respeito ao fato de que a ordem na qual estes objetos serão selecionados pelo filtro não é especificada (deixada como questão de implementação local). O usuário do serviço CMISE pode requisitar dois tipos de sincronização:

1. atômica: todos os objetos gerenciados selecionados para a operação são tratados para detetar se eles estão aptos à realização com sucesso da operação. Caso um ou mais objetos não estejam aptos, nenhuma operação é realizada;

2. melhor esforço: todos os objetos selecionados são solicitados a realizarem a operação.

9.3.5 Protocolo de Gerenciamento Comum de Informação (Common Management Information Protocol - CMIP)

O CMIP suporta os serviços fornecidos pelo gerenciamento de sistemas OSI através de um conjunto de entidades de dado de protocolo que implementam o CMIS. Estas PDUs são transmitidas em resposta às primitivas de serviço do CMIS enviada pelos usuários do serviço CMIS.

9.4 Problemas

1. Quais os requisitos que a atividade de gerenciamento de rede deve atender ? Descreva brevemente cada um deles.

2. Quais os elementos que compõem um sistema de gerenciamento de redes ?

3. Qual a arquitetura necessária para o gerenciamento OSI ?

4. Quais os elementos funcionais que compõem a arquitetura de gerenciamento OSI ?

5. Quais as funcionalidades do CMIS ?

Apêndice A

Perfil OSI Governamental (GOSIP)

Os maiores compradores de computadores e redes de comunicação são os órgãos governamentais, principalmente os do governo federal norte-americano. É natural que cada órgão dê preferência a uma arquitetura de rede que melhor atenda às suas necessidades. Surge então um problema: como interligar os diversos órgãos governamentais para fins de troca de informações técnicas e administrativas ?

O governo federal norte-americano publicou uma recomendação em 1988 no sentido de padronizar uma arquitetura mínima de interconexão de computadores. O título da recomentação, *Goverment Open Systems Interconection Profile*, gerou um termo comumente associado à publicações semelhantes: GOSIP.

A figura A.1 ilustra os componentes da especificação GOSIP norte-americana. Existem três versões desta especificação: versão 1 (estável); versão 2 (proposta) e versão 3 (futura). Uma observação importante é a agregação das camadas de sessão e apresentação à camada de aplicação[1] (isto é, os serviços de sessão e apresentação estão embutidos na aplicação que por sua vez acessa diretamente o serviço de transporte).

No Brasil, o Decreto 518 de 8 de maio de 1992 estabelece o POSIG (Perfil OSI do Governo Brasileiro). A figura A.2 ilustra os componentes do POSIG.

[1]Esta estruturação reduz as sete camadas do modelo OSI às quatro camadas da arquitetura TCP/IP.

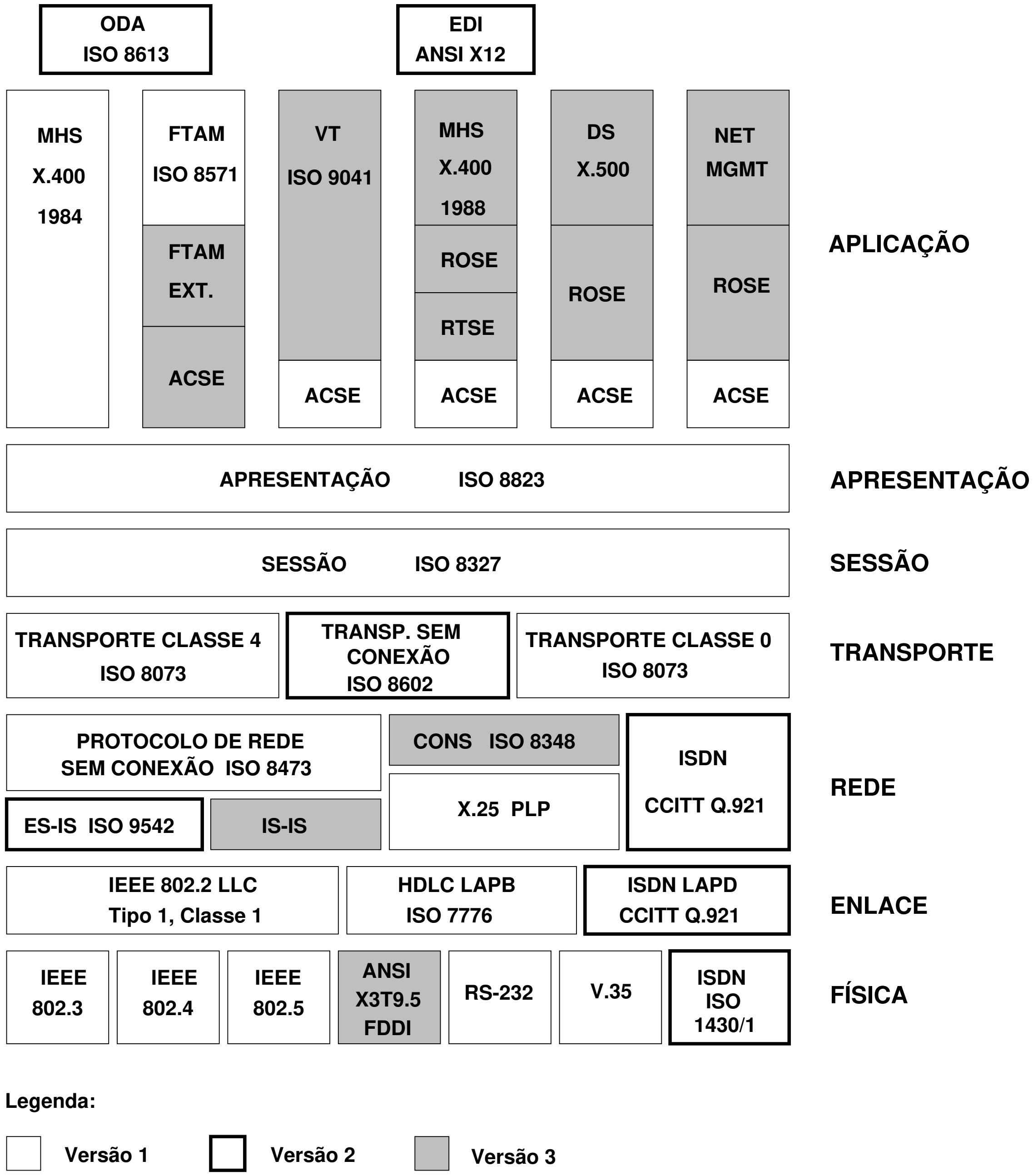

Figura A.1: GOSIP (Government Open Systems Interconection Profile) norte-americano.

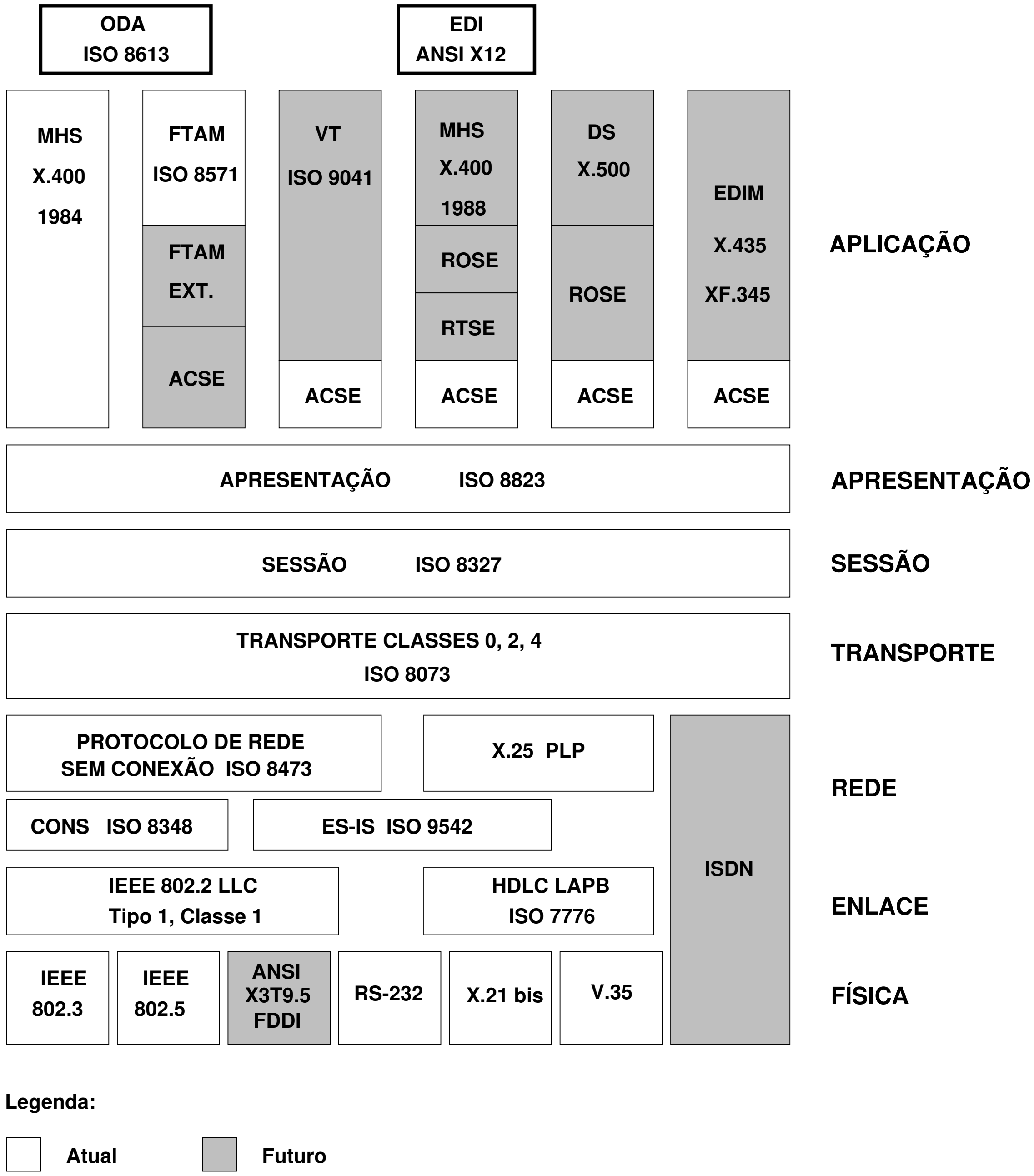

Figura A.2: POSIG (Perfil OSI do Governo Brasileiro).

Apêndice B

DECnet/OSI

DECnet Fase V ou DECnet/OSI é uma implementação da pilha OSI/ISO comercializada pela Digital Equipment Corp. A figura B.1 ilustra os protocos OSI implementados, bem como os protocolos da versão anterior (fase IV) mantidos para fins de interoperabilidade.

	DECnet/OSI	Protocolos FASE IV
APLICAÇÃO	ACSE ROSE FTAM VT	DAP
APRESENTAÇÃO	PROTOCOLO OSI DE APRESENTAÇÃO	
SESSÃO	PROTOCOLO OSI DE SESSÃO	CONTROLE DE SESSÃO
TRANSPORTE	TP0 TP2 TP4	NSP
REDE	X.25 CONS CLNS ES-IS	
ENLACE	HDLC IEEE 802.2 (LLC)	DDCMP
FÍSICA	V.24 V.25 V.35 RS-232C RS-242 RS-243 Ethernet Token Ring FDDI	

Figura B.1: Arquitetura DECnet/OSI.

O nível físico implementa as interfaces RS-232-C, RS-422, RS-423, V.35 e redes Ethernet, token Ring e FDDI.

O nível de enlace implementa os protocolos HDLC/LAPB utilizados no X.25 e 802.2 (LLC) utilizados nas LANs Ethernet, Token Ring e FDDI.

O nível de rede implementa os protocolos de rede X.25 PLP, CONS (Connection Oriented Network Service)[1] e CLNS (Connectionless Mode Network Service); além do protocolo ES-IS de roteamento.

O nível de transporte implementa os protocolos OSI orientados à conexão de classe 0, 2 e 4; além do protocolo NSP presente na DECnet Fase IV.

[1] Implementado sobre o X.25.

O nível de sessão implementa o protocolo ISO de sessão orientado à conexão; além do protocolo DNA da Fase IV.

O nível de apresentação implementa o protocolo de OSI apresentação (orientado a conexão) que oferece basicamente os mesmos serviços de sessão.

Finalmente o nível de aplicação implementa os elementos ACSE (Association Control Service Element) e ROSE (Remote Operations Service Element), bem como os serviços FTAM (File Transfer Access and Management) e VT (Virtual Terminal).