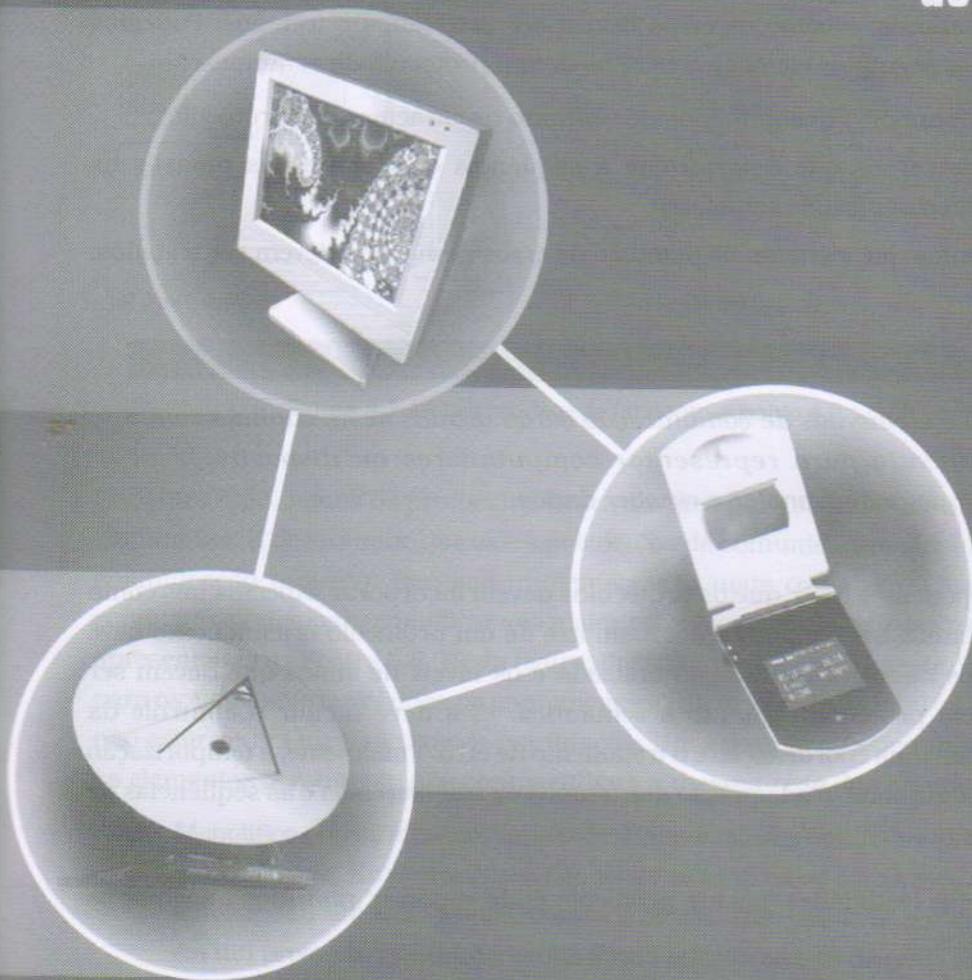


Capítulo

5

Protocolos, Modelos de
Referência e Arquiteturas
de Protocolos



Introdução

Apresentamos até o momento os conceitos básicos de comunicação de dados e os meios físicos de transmissão. Em adição, discutimos como é efetuada a troca de dados entre dois pontos em uma rede, utilizando-se de um protocolo de enlace. Em outras palavras, de maneira intencional fomos acrescentando tópicos importantes em nossas discussões sobre as redes de comunicação e computadores, sem nos preocuparmos com uma formalização das tecnologias das redes.

Desta forma, vamos, neste capítulo, trabalhar os conceitos dos protocolos, dos modelos de referência e das arquiteturas dos protocolos. Esses três elementos representam a espinha dorsal das redes de comunicação e computadores.

Protocolos

Os *protocolos* podem ser entendidos como um conjunto de regras que determinam como deverá ocorrer a comunicação entre duas estações numa rede de comunicação (ou nas redes de computadores) e como os erros devem ser detectados e tratados. Podemos então enumerar os seguintes elementos básicos de um protocolo de comunicação:

- O conjunto de símbolos denominados de conjunto de caracteres do protocolo.
- O conjunto de regras que determinam a seqüência e o tempo das mensagens pertencentes ao conjunto de caracteres.
- Os procedimentos que auxiliam na detecção de erros e em como devem ser tratados.

NOTA

Estações, nós e elementos de comunicação serão usados neste capítulo com o mesmo significado para representar computadores ou dispositivos de comunicação que podem enviar e receber dados.

De outra forma, podemos dizer que os protocolos devem incorporar aspectos tais como a sintaxe, a semântica e a temporização. A sintaxe de um protocolo orienta como deve ser o formato dos dados, a codificação utilizada e os níveis de sinais que devem ser considerados pelo protocolo. Quanto à semântica, esta deve incluir o controle da informação visando uma coordenação e o tratamento de erro. Finalmente, a temporização deve ser interpretada como a adequação dos tempos de transferência e as seqüências de mensagens permitidas.

Dentre as inúmeras facilidades que um protocolo pode suportar, podemos imaginar os seguintes exemplos:

- A comunicação entre um computador e o meio físico de uma rede de computadores (exemplo CSMA/CD).
- O acesso de um computador a uma rede de comunicação (exemplo HDLC).
- O transporte dos dados entre uma aplicação num determinado computador e outra aplicação em um outro computador (exemplos TCP e UDP).

Para uma classificação dos protocolos, é usual a consideração de algumas de suas características. Os aspectos apresentados a seguir são os mais comuns para a caracterização de um protocolo:

1. Direto ou indireto: um protocolo pode ser mais ou menos complexo dependendo do tipo de comunicação que é efetuada entre duas estações numa rede. Na forma direta de comunicação, os protocolos irão prover suporte para configurações de redes ponto-a-ponto e multiponto. Na Figura 5.1, ilustramos um exemplo de uma rede ponto-a-ponto e uma outra multiponto.

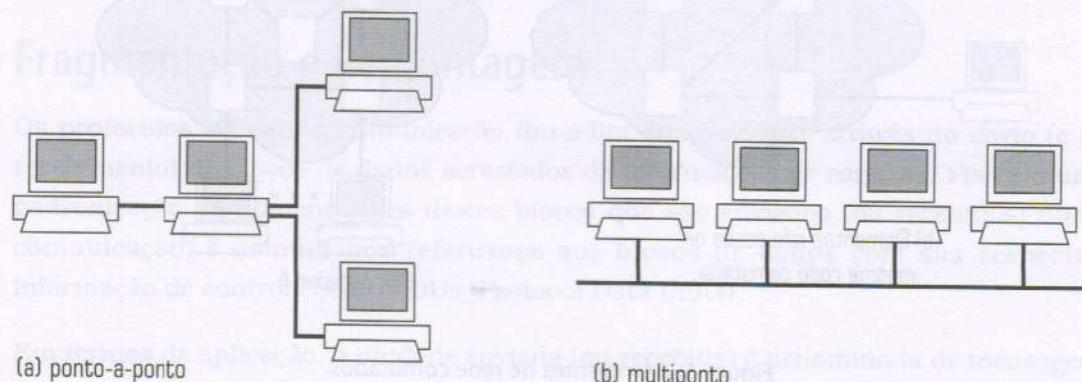


Figura 5.1 Ambientes de rede (a) ponto-a-ponto e (b) multiponto.

Por outro lado, temos os protocolos desenvolvidos para atender aos ambientes de redes comutadas. É de se supor que esses protocolos de comunicação, que irão gerenciar uma comunicação *indireta*, têm sua implementação mais complexa. Quanto ao ambiente comutado, podemos exemplificar sua maior complexidade através das redes de comutação apresentadas na Figura 5.2. Nesta figura, ilustramos: (a), uma rede de comutação comum para os elementos envolvidos na comunicação; (b), uma rede na qual os elementos podem não estar numa mesma rede de comutação. Um exemplo clássico de ambiente no qual os elementos geralmente não compartilham a mesma rede de comutação é a Internet.

2. Monolítico ou estruturado: os protocolos monolíticos têm todas as suas funções num único módulo. Em outras palavras, todas as funções na comunicação entre elementos numa rede ficam sob a responsabilidade de um único pacote de software. Por outro lado, em um protocolo que obedece uma forma estruturada, as diversas funções necessárias na comunicação são estabelecidas e executadas de maneira distribuída em níveis.

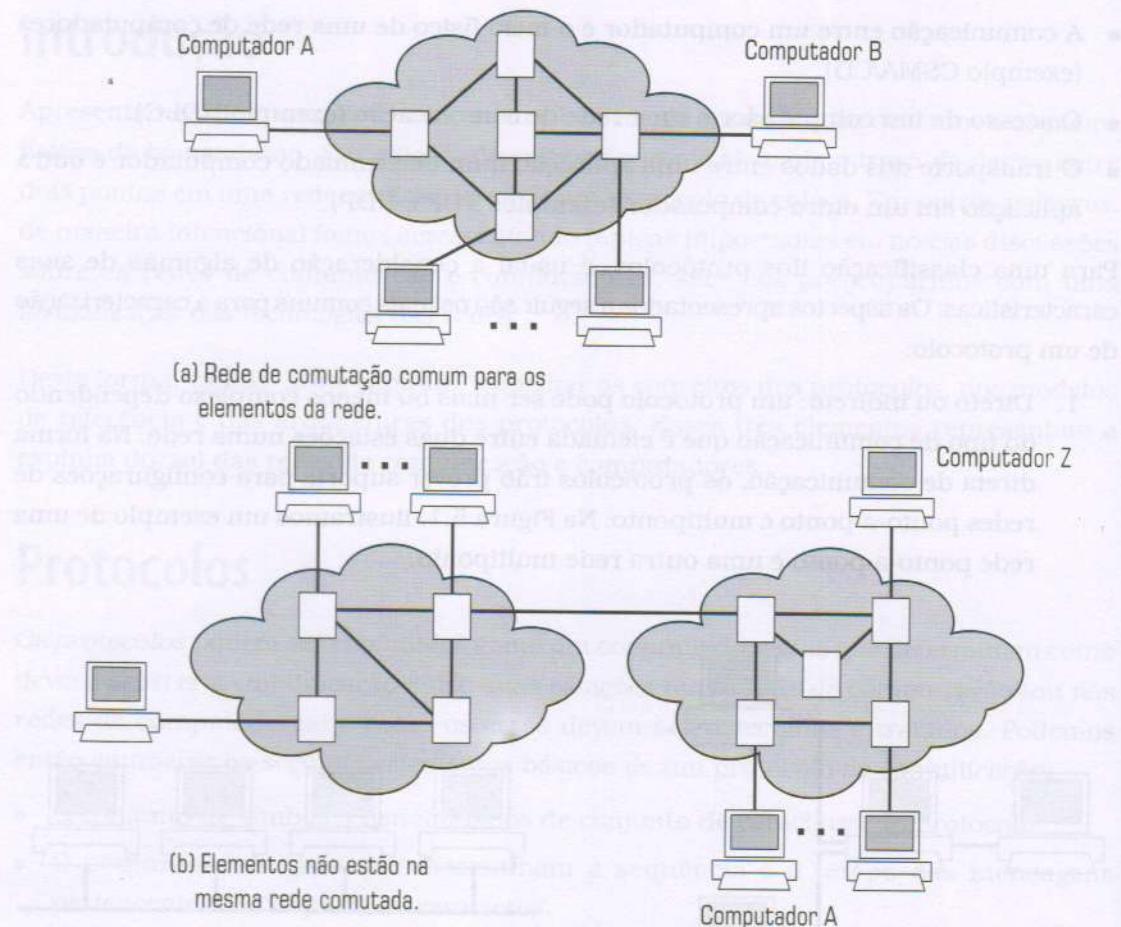


Figura 5.2 Ambientes de rede comutados.

3. Simétrico ou assimétrico: os protocolos ditos simétricos são caracterizados pela comunicação fim-a-fim dos elementos envolvidos na operação. Diferente dos protocolos simétricos, os assimétricos podem ser exemplificados pelo estabelecimento de como deverá ser efetuada uma determinada comunicação por um dos elementos envolvidos na comunicação. Assim, podemos exemplificar um protocolo assimétrico na comunicação entre um cliente e um servidor, ou por exemplo um computador de grande porte que seleciona quais terminais estão autorizados para fazer determinado acesso.
4. Padronizado e não-padronizado: um outro aspecto também muito importante com relação aos protocolos diz respeito a sua padronização (ou não). Os protocolos não-padronizados são, muitas vezes, conhecidos por protocolos fechados ou proprietários. Enquanto os protocolos padronizados são aqueles que seguem uma determinada orientação normativa e são denominados de protocolos abertos.

NOTA

Cabe ressaltar que um protocolo fechado, ou não-padronizado, não é uma abordagem que devemos desconsiderar completamente. É importante lembrar que sempre que possível a utilização do protocolo padronizado é desejável. Todavia, existem situações particulares em que um protocolo de comunicação não-padronizado é a resposta correta para o problema. Em ambientes de controle de dispositivos na indústria, por exemplo, não é incomum encontrarmos protocolos não-padronizados.

Até agora, estudamos uma possível classificação dos protocolos de acordo com algumas de suas características. Um outro aspecto relevante que devemos considerar em relação aos protocolos diz respeito a algumas de suas funções. Funções como a fragmentação e a remontagem, o encapsulamento, o controle de conexão, a entrega ordenada (ou não-ordenada), o controle de fluxo e erros, o endereçamento, a multiplexação e os serviços de transmissão disponíveis. A seguir, detalhamos estas funções.

Fragmentação e Remontagem

Os protocolos efetuam a comunicação fim-a-fim em uma rede através do envio (e do recebimento) de blocos de dados acrescidos de informações de controle. Visando uma padronização de nomenclatura destes blocos que são enviados (ou recebidos) numa comunicação, é comum nos referirmos aos blocos de dados com sua respectiva informação de controle como PDUs (Protocol Data Units).

Em termos de aplicação, a unidade enviada (ou recebida) é denominada de mensagem. O processo de envio de uma mensagem requer que, muitas vezes, nas camadas inferiores, seja efetuada uma quebra da mensagem em PDUs menores. Esta operação de quebra da mensagem é chamada de fragmentação. Quando as PDUs chegam ao destinatário, uma operação inversa é efetuada. Essa operação é chamada de remontagem da mensagem. A Figura 5.3 ilustra as operações (a) de fragmentação e (b) remontagem.

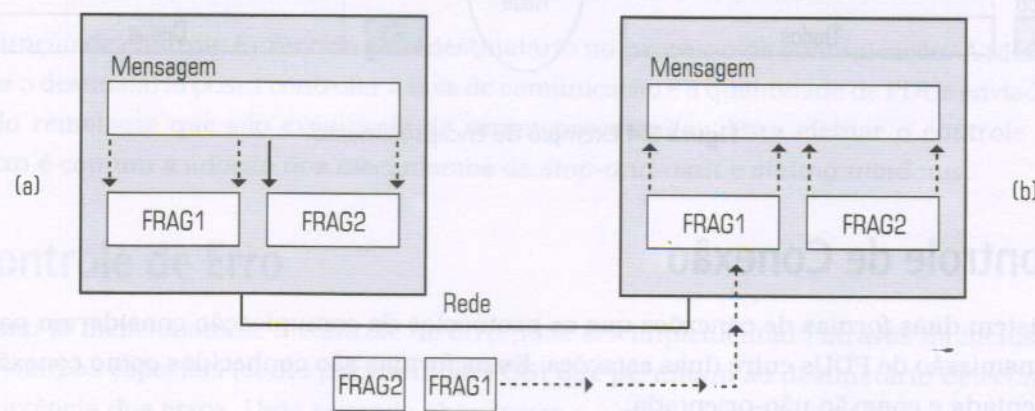


Figura 5.3 Fragmentação e remontagem.

Dentre as vantagens da fragmentação, podemos imaginar, a título de exemplo, o caso da adequação do tamanho da mensagem para um determinado tamanho máximo permitido por um meio físico de transmissão. Por outro lado, como desvantagem desta técnica, podemos dizer que um maior número de PDUs nos leva a despender mais tempo para o processamento de cada bloco de dados.

NOTA

Este parâmetro é denominado de MTU (Maximum Transfer Unit) e representa o tamanho máximo de dados permitidos para transferência em uma determinada rede.

Encapsulamento

A operação de adição de informações de controle aos dados que devem ser transmitidos é conhecida como encapsulamento. A Figura 5.4 mostra a idéia do processo de encapsulamento. Uma PDU é caracterizada por conter, na maioria da vezes, dados e informações de controle (existem casos de PDUs que contêm apenas informações de controle). As informações de controle podem ser agrupadas em três categorias:

- Endereço: indica o endereço do destinatário e (ou) remetente.
- Detecção de erro: algum mecanismo é acrescido para auxiliar na detecção de erros.
- Controle do protocolo: alguma função particular do protocolo pode ser efetuada pela inserção de informações adicionais.

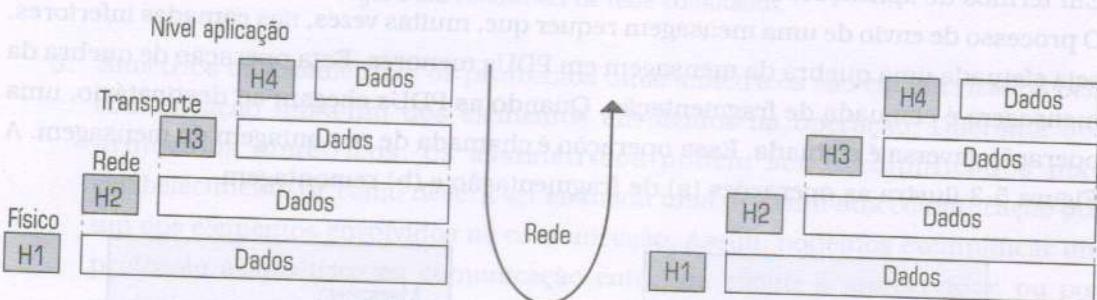


Figura 5.4 Exemplo de encapsulamento.

Controle de Conexão

Existem duas formas de conexões que os protocolos de comunicação consideram para transmissão de PDUs entre duas estações. Estas formas são conhecidas como conexão-orientada e conexão não-orientada.

Em um ambiente de conexão-orientada, primeiro é estabelecida uma ligação dedicada entre as estações de origem e destino. Em seguida, é efetuada a transmissão das PDUs. A etapa final é o fechamento da conexão.

Em uma conexão não-orientada, a transmissão é caracterizada pelo uso das ligações existentes da rede de comunicação para o envio das PDUs. Em outras palavras, as PDUs são enviadas da origem para o seu destino utilizando-se de ligações existentes e não dedicadas.

NOTA

Algumas vezes na literatura de redes, em português, encontramos referências às conexões não-orientadas como sendo sem conexão. É importante notar que uma conexão-orientada pode ser entendida como uma ligação dedicada entre dois elementos de comunicação durante o tempo de transmissão de PDUs. Por outro lado, a conexão não-orientada significa que não será utilizado nenhum canal dedicado para a transmissão das PDUs. Em outras palavras, as PDUs vão utilizar as ligações já existentes da rede de comunicação para chegar ao destinatário. Desta forma, a conexão existe, mas não é dedicada.

Entrega Ordenada

Quando as PDUs estiverem sendo enviadas/recebidas por duas estações interligadas por uma rede comutada, é possível que cheguem fora de ordem. A razão para a chegada fora de ordem é explicada pela existência de diversos caminhos diferentes entre dois pontos numa rede comutada. Quando o protocolo emprega uma conexão-orientada, é, geralmente, garantida a ordenação das PDUs. Por outro lado, se o serviço oferecido pelo protocolo não é orientado, muito provavelmente ocorrerá a chegada fora de ordem das PDUs. Por este motivo, mecanismos de ordenação devem ser previstos para que a mesma seja efetuada.

Controle de Fluxo

A função de controle é exercida pelo destinatário no processo de comunicação. A idéia é que o destinatário possa controlar a taxa de comunicação e a quantidade de PDUs enviadas pelo remetente que são exequíveis de serem processadas. Para efetuar o controle de fluxo é comum a adoção dos mecanismos de *stop-and-wait* e *sliding windows*.

Controle de Erro

Como já mencionamos, o controle de erro pode ser implementado através da inclusão de campos especiais (como por exemplo FCS) que permitam ao destinatário detectar a ocorrência dos erros. Uma segunda abordagem é a retransmissão das PDUs.

Endereçamento

O endereçamento num protocolo de comunicação é um dos pontos mais complexos e que requer uma atenção especial. Os endereços são, usualmente, classificados como endereços de nível, escopo e modo.

Um endereço de nível é aquele que identifica um elemento computacional (por exemplo: são computadores e roteadores) de uma forma única numa determinada arquitetura de protocolo. Em outras palavras, podemos dizer que o endereço de nível se refere ao endereço da camada de rede. Na arquitetura TCP/IP (como estudaremos na seção de TCP/IP), o endereço de nível significa o endereço IP.

O segundo tipo de endereço, o endereçamento de escopo, pode ser entendido como aquele endereço que irá servir para que computadores interligados nas redes de comunicação se comuniquem e não sejam confundidos com outros computadores. Um clássico exemplo é o endereço de acesso ao meio (MAC – *Medium Access Control – Address*). Cada computador ligado numa rede local tem uma placa de rede com um único endereço de rede.

NOTA

As placas de rede são usualmente chamadas de NIC – Network Interface Cards.

NOTA

O endereço de escopo pode ser comparado com um número de identificação tipo CPF, ou número de identidade mundial. Em outras palavras, não existe duas ocorrências desse número.

O modo de endereçamento se refere à abrangência de endereço para qual(ais) elemento(s) queremos comunicar. Em outras palavras, uma estação pode enviar PDUs para um único destinatário, para um grupo de destinatários ou para todas estações de uma rede. O envio de PDUs para um único destinatário é conhecido como endereço *unicast*. Quando endereçamos PDUs para um grupo de estações, estamos utilizando um endereço de *multicast*. Por fim, o endereço de *broadcast* é aquele pelo qual todas as estações de uma determinada rede receberão as PDUs enviadas.

Multiplexação

A multiplexação é um conceito com estreita relação com o endereçamento. Para ilustrar, podemos imaginar um conjunto de aplicações solicitando serviços orientados à conexão

para um determinado endereço. É esperado que, por exemplo, um protocolo de transporte receba todas as solicitações dos protocolos de aplicação e efetue uma multiplexação para implementar uma comunicação eficiente entre estações.

Serviços Diferenciados de Transmissão

Finalizando, temos ainda os serviços diferenciados de transmissão como uma função desejável de um protocolo. Este deve prover facilidades tais como prioridade, classes de serviços e segurança. Em outras palavras, devemos imaginar que uma diferenciação de PDUs é necessária sob determinados aspectos numa transmissão. Assim, os serviços diferenciados de transmissão, por exemplo, auxiliam na adequação de prioridades e alocação de largura de banda.

Modelos

Um fato, verificado ao longo de vários anos na área de redes de computadores, foi a falta de modelos de referência padronizados sobre as especificações detalhadas e claras das funções dos protocolos e seu inter-relacionamento.

NOTA

É importante ressaltar que esta preocupação é antiga e praticada na área de redes de comunicação. Conforme estudamos no capítulo de conceitos básicos, a padronização é um fator presente e de sucesso na área das redes de comunicação. Um exemplo que confirma nossa afirmação é a despreocupação de um usuário do sistema telefônico quando este deseja fazer uma ligação nacional/internacional. Em outras palavras, não existe uma preocupação quanto ao tipo de aparelho ou sistema telefônico da pessoa com quem se deseja falar.

Dentre muitas vantagens da abordagem de modelos de referência, podemos citar uma modularização de funções e a interoperabilidade entre protocolos de diferentes vendedores. Por esta razão, a metodologia atualmente mais aceita é a especificação dos protocolos e sua estruturação em níveis que são chamadas de camadas.

Entendemos os *modelos de referência dos protocolos* como uma estrutura onde existe um detalhamento da função de cada nível, das relações entre as interfaces das camadas e dos protocolos. Em outras palavras, o modelo de referência representa uma abstração na qual existem as especificações de como o ambiente deve funcionar. Todavia, não existe a menção de uma implementação de um protocolo específico.

NOTA

Modelos de referência dos protocolos pode, também, ser entendido como modelos de protocolos. Desta forma, usamos os dois termos ao longo deste capítulo e do livro.

A abordagem funcional das entidades e da estrutura em níveis permite que os modelos de protocolos sejam propostos, e que estas especificações possam ser abertas. De outra forma, podemos dizer que os modelos de referência permitem que um fabricante implemente de sua maneira um determinado conjunto de protocolos e, assim, poderemos ter a interoperabilidade deste pacote de software com outro pacote padronizado desenvolvido por outro fabricante.

NOTA

O conceito de modelo referência de protocolo, na literatura de rede, algumas vezes é também apresentado como arquitetura de redes. Entendemos que, nas arquiteturas de protocolos, temos os protocolos já implementados e distribuídos nos níveis da arquitetura. Em nossa visão, o modelo de protocolo é somente a definição estrutural e funcional do ambiente, ou seja, não existem protocolos implementados. Um fato que confirma nossa abordagem é a denominação dos modelos como referência. Em outras palavras, é comum a denominação do modelo de referência OSI ou do modelo de referência TCP/IP. Em adição, é interessante mencionar que os modelos de protocolos são usualmente dispostos numa forma de pilha.

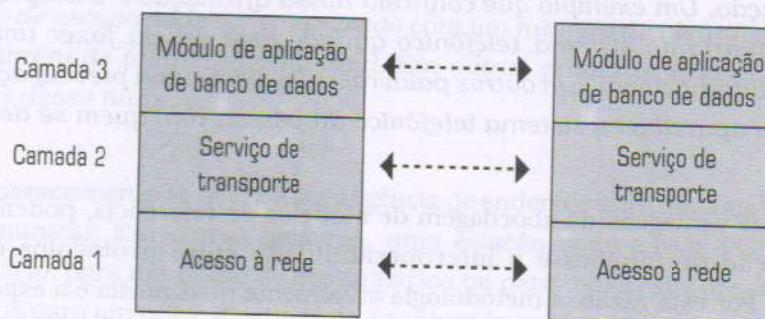


Figura 5.5 Modelo genérico para aplicações de banco de dados.

A Figura 5.5 ilustra um exemplo de modelo genérico simplificado de protocolo em camadas para suporte de aplicações de banco de dados. Neste exemplo, estabelecemos que o modelo de protocolo tem suas funções distribuídas em três camadas. Na primeira camada, determinamos que deverá existir um módulo de acesso à rede, incluindo-se os serviços

físicos e de enlace. O próximo nível, denominado de serviço de transporte e rede, é responsável por um serviço confiável orientado à conexão e serviço de roteamento para a Internet. No terceiro nível, o módulo de aplicação de banco de dados é proposto para atender as solicitações distribuídas, inclusive entre ambientes de fabricantes distintos de software. Em adição, o modelo de referência do exemplo propõe uma interação transparente nas interfaces dos níveis e que os protocolos obedeçam a uma abordagem estruturada.

Observe que, no modelo genérico simplificado da Figura 5.5, não existe a menção a um determinado protocolo já existente. Também é possível que, em um determinado nível, mais de um protocolo possa existir para abranger todas as funções. Por outro lado, novos protocolos futuramente podem ser agregados a um determinado nível para efetuar uma função específica.

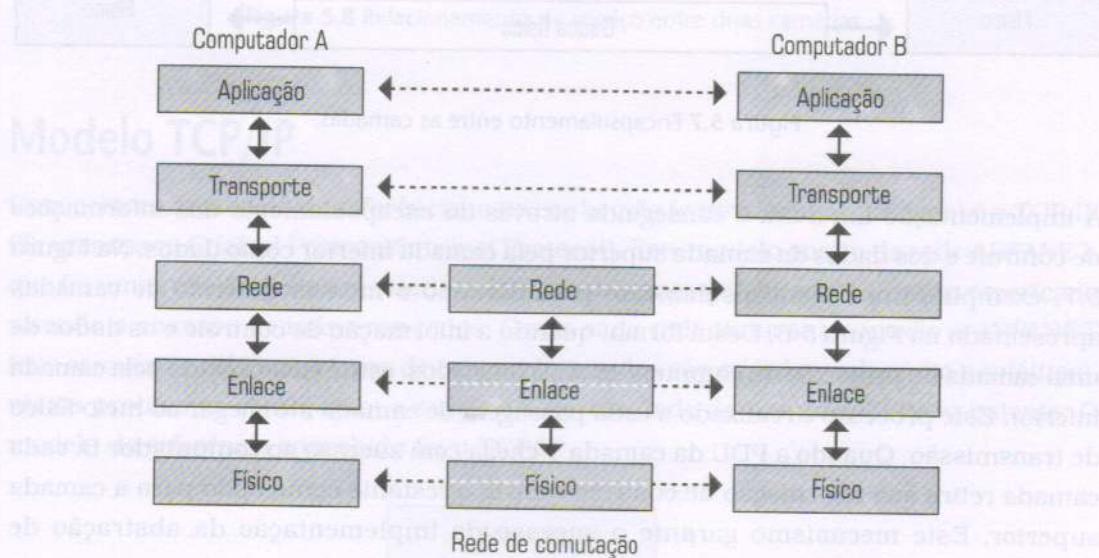


Figura 5.6 Relacionamento das camadas, protocolos e interfaces.

Na Figura 5.6, exemplificamos o relacionamento das camadas, seus protocolos e interfaces entre dois computadores num ambiente de rede genérico. Algumas observações sobre a Figura 5.6 são:

- Os protocolos de cada camada, de cada computador, se comunicam com seu respectivo par no outro computador. Este tipo de comunicação é chamado de fim-a-fim. Assim, um protocolo da camada n no computador A troca informação com o protocolo do computador B da camada n .

NOTA

Uma dúvida muito comum que surge neste ponto é o perfeito entendimento de como é implementada a comunicação fim-a-fim entre as camadas de diferentes

computadores. Esta dúvida é bastante compreensível, de vez que, para que a comunicação transparente entre camadas iguais, entre diferentes computadores, seja efetuada, temos que passar pelas camadas inferiores.

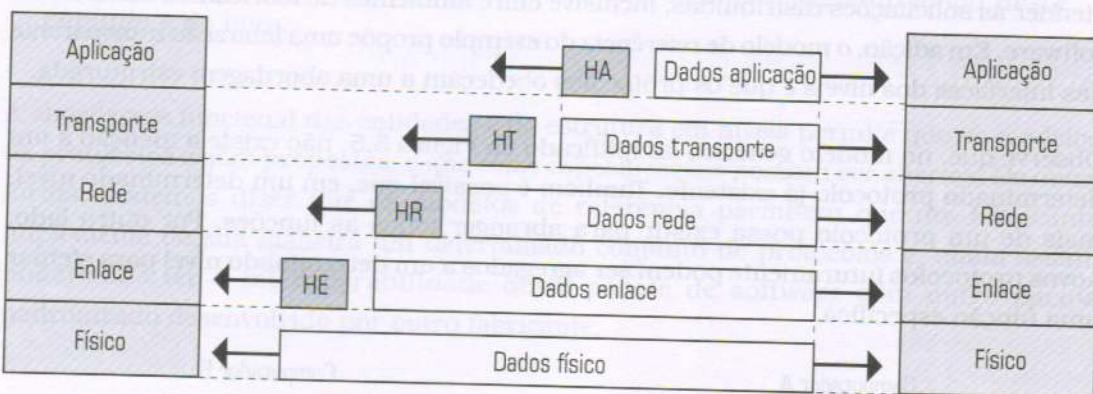
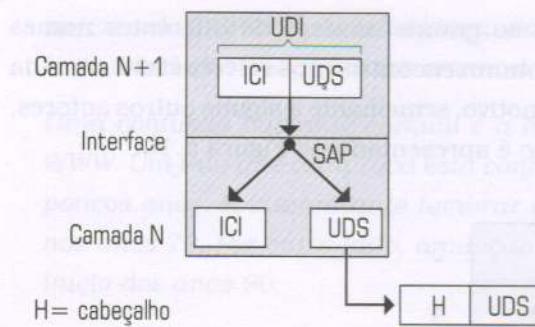


Figura 5.7 Encapsulamento entre as camadas.

A implementação fim-a-fim é conseguida através do encapsulamento das informações de controle e dos dados da camada superior pela camada inferior como dados. Na Figura 5.7, exemplificamos o encapsulamento considerando o modelo genérico de camadas apresentado na Figura 5.6. Desta forma, quando a informação de controle e os dados de uma camada do protocolo do computador A são enviados, este é encapsulado pela camada inferior. Este processo é realizado a cada passagem de camada até chegar ao meio físico de transmissão. Quando a PDU da camada 1 chega com sucesso ao computador B, cada camada retira sua informação de controle e envia o restante como dado para a camada superior. Este mecanismo garante o sucesso da implementação da abstração de comunicação transparente entre camadas.

- As interfaces entre as camadas definem as operações e serviços que são oferecidos pela camada inferior para a camada superior. A forma de acesso aos serviços é possível por intermédio dos pontos denominados de SAPs (Service Access Points). Cada SAP é caracterizado por um endereço exclusivo através do qual este serviço é identificado de maneira unívoca. Para um melhor entendimento da abstração dos SAPs, é clássica a comparação dos SAPs com as tomadas e números dos sistemas telefônicos e os endereços nos sistemas de correios, respectivamente. De uma outra forma, é importante que você saiba o número telefônico de uma determinada pessoa para que possa se comunicar com ela. Por outro lado, é necessário o endereço do destinatário para que você envie uma correspondência. Na Figura 5.8, ilustramos a relação entre duas camadas.



- SAP = Service Access Point (ou ponto de acesso de serviço entre camadas).
- IDU = Interface Data Unit (ou UDI = Unidade de Dado da Interface).
- SDU = Service Data Unit (ou UDS = Unidade de Dado do Serviço).
- ICI = Control Information Interface (ou ICI = Informação de Controle da Interface).
- PDU = Protocol Data Unit (ou UDP = Unidade de Dado do Protocolo).

Figura 5.8 Relacionamento de serviço entre duas camadas.

Modelo TCP/IP

Com certeza, o modelo de referência mais conhecido (e um dos mais antigos) é o TCP/IP (Transmission Control Protocol/Internet Protocol). Este modelo surgiu da rede ARPANET, que foi uma rede de pesquisa criada pelo Departamento de Defesa do governo americano visando a conexão de inúmeras redes. Como cada rede tinha sua conexão à ARPANET feita através de diferentes tipos de enlaces (exemplos são os enlaces de rádio e satélites), vários problemas começaram a surgir e a necessidade de um modelo ficou patente. O modelo de referência concebido foi o TCP/IP.

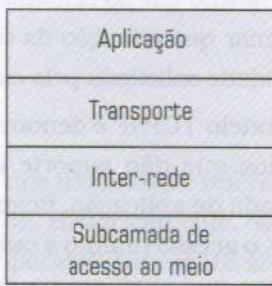


Figura 5.9 O modelo de referência TCP/IP.

O modelo TCP/IP, originalmente, foi projetado em quatro camadas como exemplificado na Figura 5.9. As quatro camadas do modelo imaginadas foram:

- Interface de rede: esta é a primeira camada do modelo TCP/IP. Descrita de maneira superficial em vários documentos, atribui a função de suporte à camada de rede, que é a camada imediatamente superior à camada 1. Como deve ser efetuado, o suporte à camada superior não é bem definido. Os serviços que deveriam ser definidos compreendem as funções de acesso físico e lógico ao meio físico. Uma outra observação

quanto a esta primeira camada se refere ao grande número de diferentes nomes atribuídos e encontrados na literatura. É comum encontrarmos referências à camada host/rede, rede, hardware e física. Por este motivo, semelhante a alguns outros autores, o modelo TCP/IP que vou adotar neste livro é apresentado na Figura 5.10.

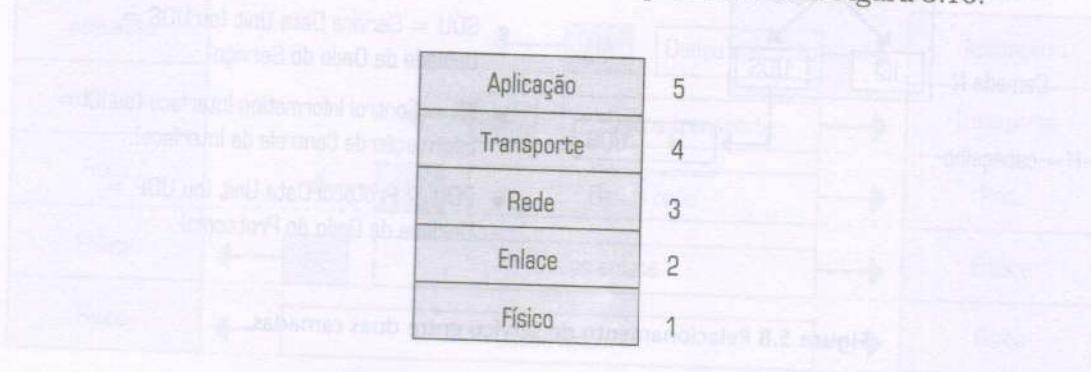


Figura 5.10 Modelo de referência TCP/IP modificado.

- Inter-rede: no modelo TCP/IP, o nível de inter-rede é o responsável pelo envio dos datagramas de um computador qualquer para outro computador, independente de suas localizações na rede. Os datagramas são a denominação da PDU neste nível do modelo. O protocolo IP (Internet Protocol), da arquitetura TCP/IP que vamos estudar na próxima seção, é um exemplo de protocolo da camada de inter-rede.
- Transporte: a camada de transporte é responsável por prover suporte à camada de aplicação de maneira confiável (ou não), independente dos serviços oferecidos pelas camadas de interface de rede e inter-rede. Então, podemos afirmar que a camada de transporte deve oferecer um serviço de qualidade, mesmo que as camadas 1 e 2 não ofereçam tal serviço. Em outras palavras, podemos afirmar que a função da camada de transporte é garantir uma conexão fim-a-fim com a qualidade solicitada pela camada de aplicação.
- Aplicação: a quarta camada do modelo TCP/IP é denominada de camada de aplicação. Nesta camada, estão os protocolos que dão suporte às aplicações dos usuários. É importante observar que, na camada de aplicação, ficam protocolos que auxiliam, por exemplo, a transferência de dados, o acesso remoto a outros computadores, o protocolo de correio eletrônico, os protocolos que auxiliam na gerência das redes, o protocolo que faz o mapeamento dos nomes dos computadores para seus endereços de rede, e ainda o protocolo que implementa a busca de páginas na WWW (World Wide Web).

NOTA

Alguns comentários são interessantes quanto ao modelo de referência TCP/IP. A primeira observação fica por conta da fraca definição original da camada 1. Todavia, é razoável imaginar que, sendo um dos primeiros modelos padronizados de rede, não existia experiência suficiente para definição da interoperabilidade requerida. O modelo foi concebido em quatro níveis. Entretanto, com retorno do

sucesso do TCP/IP após o largo uso do ambiente WWW (World Wide Web), ficou impossível não imaginarmos um modelo sem distintas camadas física e lógica.

Uma confusão bastante comum é a associação de igualdade entre Internet e WWW. Um fato que comprova esta confusão é a afirmação de que a Internet tem poucos anos. É interessante lembrar que a Internet foi um modelo imaginado nos anos 70. Por outro lado, a adoção da idéia de uso do ambiente WWW é do início dos anos 90.

Ainda com relação ao modelo TCP/IP, a Figura 5.11 ilustra seus cinco níveis e a nomenclatura que vamos adotar quanto às unidades de protocolo. Em outras palavras, a figura indica que, em termos de aplicação, vamos nos referir às mensagens; que os pacotes são as unidades manipuladas na camada de transporte; que os datagramas são tratados no âmbito de rede; que os quadros são as unidades do nível 1.

5	Aplicação	}	Mensagem
4	Transporte		Pacote
3	Rede		Datagrama
2	Enlace		
1	Físico		Quadro

Figura 5.11 Níveis e unidades de protocolos.

Modelo ISO/OSI

Um fato interessante a respeito dos modelos de referência é que muitos imaginam que sua adoção é um fato simples e natural. Historicamente, os grandes fabricantes de computadores não tinham interesse em interoperabilidade entre sistemas diferentes. A razão para este fato é óbvia: cada fornecedor desejava garantir um mercado cativo de usuários. Nenhum grande fabricante tinha interesse em que seus usuários pudessem ter interoperabilidade com outros ambientes computacionais, em termos de hardware e software.

Por outro lado, os usuários e as organizações tinham a cada dia seus sistemas mais heterogêneos em termos de hardware e software, sem a possibilidade de compartilhamento efetivo dos seus recursos. Este fato levou a ISO à proposição de um modelo de referência conhecido por RM-OSI (Reference Model – Open Systems Interconnection).

NOTA

O leitor neste ponto deve estar se perguntando por que os usuários na época não optavam pelo uso do TCP/IP. Esta é uma boa dúvida, pois o modelo, que hoje é indiscutivelmente adotado no mercado, em todo o mundo, não se apresentava como uma solução. Podemos afirmar que, pelo contrário, muitas vezes o modelo TCP/IP era desprezado como solução. Era rotulado como um modelo eficiente para universidades e centros de pesquisa. O ressurgimento do TCP/IP se deu após a adoção no início dos anos 90 do paradigma de acesso fácil à rede através do ambiente WWW. Neste ambiente, temos protocolos com interfaces gráficas mais amigáveis para, por exemplo, a transferência de dados, como é o exemplo do HTTP (HyperText Transfer Protocol).

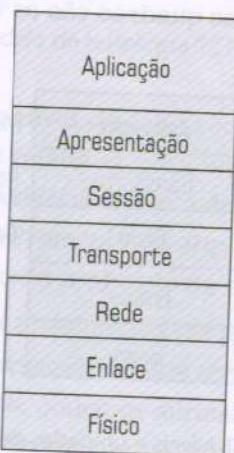


Figura 5.12 O modelo de referência RM-OSI.

Nossas observações sobre o TCP/IP no parágrafo anterior são importantes para que o leitor entenda a relevância do modelo de referência RM-OSI. Em adição, o modelo foi proposto devido ao sucesso na padronização da arquitetura de protocolos TCP/IP. A idéia foi melhorar o modelo de referência TCP/IP, inclusive considerando alguns aspectos sugeridos pelos fabricantes. Como resultado dos fatos apresentados, em 1983 a ISO propôs seu modelo de referência, constituído de sete camadas, cada qual com suas funções específicas. Apresentamos, a seguir, uma descrição das camadas do modelo de referência RM-OSI que é ilustrado na Figura 5.12.

- **Física:** esta é a primeira camada do modelo; seu objetivo é a definição elétrica e mecânica da interface de rede. Por esta razão, na camada física, são estabelecidos a forma de representação em volts dos 0s e 1s, o tempo de duração dos bits, as direções possíveis de transmissão, a quantidade de pinos da placa de rede e outros aspectos elétricos e mecânicos relativos à interface de rede.

- Enlace: esta segunda camada proposta no modelo RM-OSI, já verificando a falta de uma definição clara apresentada no modelo TCP/IP, é responsável pela fragmentação dos dados recebidos pela camada superior em quadros e por seu envio. Quanto ao mecanismo de envio dos dados, é relevante lembrar que este inclui, também, o processamento dos quadros de controle enviados pelo receptor. Como a função da camada 1 é apenas física (elétrica e mecânica), a camada de enlace é responsável pelo reconhecimento do início e final dos quadros, pelo controle de fluxo entre remetente e destinatário e ainda pela forma de acesso ao meio.
- Rede: a terceira camada do modelo é chamada de camada rede. Alguns dos serviços oferecidos nesta camada são a conexão com outros sistemas computacionais, roteamento dos datagramas entre uma determinada origem e seu destino, o estabelecimento/manutenção e fechamento das conexões e ainda o controle do congestionamento da rede.
- Transporte: a camada de transporte (ou camada 4) provê serviços orientados e não-orientados. O serviço orientado à conexão assegura uma transferência confiável fim-a-fim entre dois pontos na rede. De maneira geral, exemplos de outras funções encontradas na camada de transporte são a detecção e correção de erros, o controle de fluxo do transporte, a fragmentação e remontagem, o tratamento da seqüência dos PDUs de transporte, a multiplexação e os serviços diferenciados de transmissão (exemplo: prioridade de determinados PDUs em relação aos demais).
- Sessão: a camada 5 foi projetada para permitir a comunicação com sucesso entre aplicações. Então, funções clássicas deste nível são o estabelecimento, o gerenciamento e o término de sessões. Na interoperabilidade entre sessões de diferentes ambientes computacionais, devemos cuidar dos aspectos de comunicação do tráfego (simplex, half-duplex ou full-duplex), da sincronização das partes e do gerenciamento de permissões entre as sessões.
- Apresentação: a camada de apresentação fornece um serviço para as aplicações de independência da representação de seus dados. Podemos imaginar aplicações que utilizam a codificação EBCDIC solicitando uma interação com outra que utiliza a codificação ASCII. Desta forma, a camada 6 gerencia estas diferenças, fazendo com que a representação seja apropriada em cada computador.

NOTA

Um exemplo que auxilia na compreensão das funções das camadas de sessão e apresentação é um acesso do seu computador pessoal a um computador IBM de grande porte. Seu PC não tem um esquema de acesso (sessão) às aplicações e recursos compatível com o IBM de grande porte. Por outro lado, a codificação utilizada no IBM é EBCDIC, que não é a mesma do ASCII adotada no seu computador pessoal.

- Aplicação: na camada de aplicação encontramos os protocolos que auxiliam os processos dos usuários. Exemplos são a conexão de terminais entre diferentes ambientes computacionais, transferência de arquivos e gerência de nomes e endereços na rede.

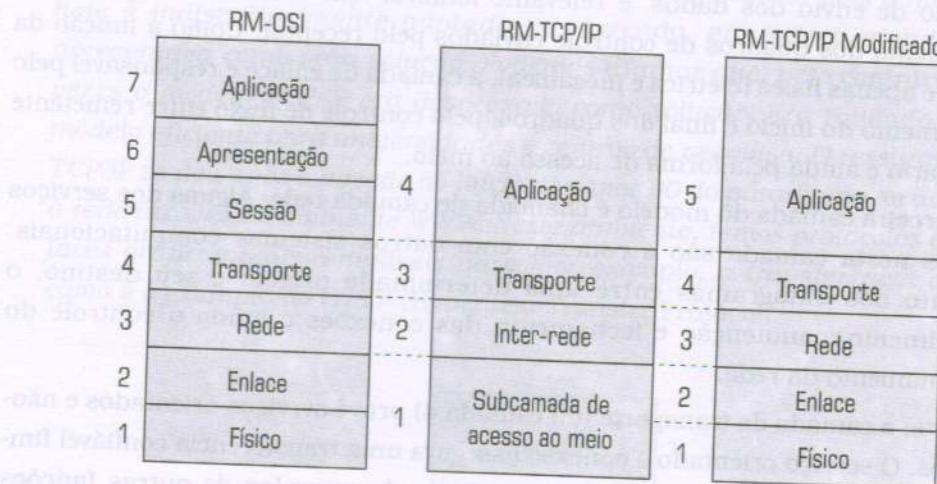


Figura 5.13 Comparação dos modelos de referência RM-OSI e TCP/IP.

Na Figura 5.13, fazemos uma comparação entre os modelos TCP/IP e RM-OSI. Verifique que, no caso do protocolo TCP/IP, estamos apresentando o modelo convencional e o modificado, que adotamos neste livro.

Modelo IEEE 802

Com a falta de definição observada na camada 1 no modelo TCP/IP, inúmeras soluções proprietárias surgiram no mercado. Com o objetivo de padronizar esta grande quantidade de soluções que vinham surgindo para as redes locais (LANs), a Sociedade de Computação do Instituto de Engenheiros Elétricos e Eletrônicos (Computer Society do IEEE), nos Estados Unidos, criou um comitê de padronização em 1980.

O IEEE publicou um conjunto de padrões que, primeiro, foi adotado pelo Instituto Nacional Americano de Padronização (ANSI) e, após uma determinada revisão, foi aceito pela ISO. Na ISO, o modelo ficou conhecido como ISO 8802.

NOTA

O leitor deve observar, quando estiver participando em um projeto de rede, que a especificação IEEE 802 e a ISO 8802 são semelhantes. A diferença é que o escopo de abrangência do IEEE 802 é americano, enquanto o ISO 8802 é internacional.

Na Figura 5.14, fazemos uma apresentação do modelo de referência IEEE 802 comparando-o com o modelo RM-OSI. O modelo IEEE 802 é composto por três camadas, as quais cobrem as funções essenciais de uma rede local. Numa rede local, a maior preocupação de qualquer tecnologia é voltada para implementações eficientes das funções físicas e de enlace. Exemplos de funções importantes no âmbito de enlace numa rede local são os serviços de acesso concorrentes dos usuários ao meio físico, a fragmentação/remontagem de quadros com seus respectivos campos de controle, e gerência de comunicação em nível de enlace. Por outro lado, o número de pinos e suas funções são aspectos que devem obedecer a certos padrões.

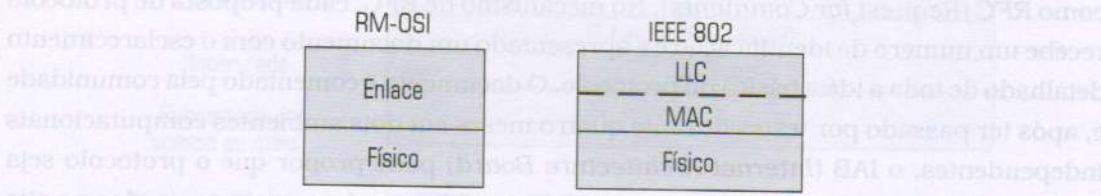


Figura 5.14 Comparação IEEE 802 e RM-OSI.

Arquiteturas de Protocolos

Uma *arquitetura de protocolos* pode ser interpretada como a representação de protocolos já implementados e dispostos de forma estruturada. Podemos dizer, de outra forma, que a arquitetura de protocolos é baseada em um determinado modelo de referência. A estrutura mais comum encontrada nos modelos de referência é a forma de pilha; por conseguinte, nas arquiteturas, temos os protocolos também organizados em pilhas funcionais. Podemos dizer que cada nível da pilha tem um certo número de protocolos que executam serviços para os protocolos de uma camada superior. Um determinado nível de protocolo se comunica com outro nível, superior ou inferior, através de entidades denominadas de interfaces. Todavia, na comunicação fim-a-fim, cada nível só percebe seu nível correspondente na outra ponta.

As arquiteturas de protocolos proprietários são aqueles ambientes implementados por fabricantes na comunicação de suas redes fechadas. Exemplos de arquiteturas de protocolos proprietários são o SNA (System Network Architecture) da IBM, o DNA (Digital Network Architecture) da Digital, o Netware da Novell, e o Apple Talk da Apple Computer.

Por outro lado, as arquiteturas de protocolos abertos são caracterizadas por serem projetos independentes de fabricantes e por suas definições serem estabelecidas em órgãos normativos, tais como o IEEE, o ITU-T e a ISO. Exemplos clássicos são as arquiteturas de protocolos TCP/IP, IEEE 802.x e ISO/OSI 8802.x. Vamos, a seguir, conhecer mais sobre as arquiteturas TCP/IP e IEEE 802.

Arquitetura TCP/IP

A importância e o potencial da tecnologia de *internetworking* (inter-rede) foram visualizados pelas agências governamentais americanas de pesquisa no final dos anos 60. Por esta razão, foi inicializado o desenvolvimento de uma arquitetura de protocolos que pudesse interoperar diversos computadores com diferentes ambientes de software e hardware. Entre os anos de 1977 e 1979, esta arquitetura de protocolos foi concluída e denominada de TCP/IP Internet Protocol Suite.

Antes de começarmos a estudar os protocolos da arquitetura TCP/IP, é importante saber que um protocolo se torna padrão da arquitetura através de um mecanismo conhecido como *RFC* (*Request for Comments*). No mecanismo de RFC, cada proposta de protocolo recebe um número de identificação e é apresentado um documento com o esclarecimento detalhado de toda a idéia básica do protocolo. O documento é comentado pela comunidade e, após ter passado por testes durante quatro meses em dois ambientes computacionais independentes, o IAB (*Internet Architecture Board*) pode propor que o protocolo seja aceito como um padrão da arquitetura TCP/IP (as RFCs podem ser encontradas no site do IETF apresentado na seção de referências deste capítulo).

NOTA

Um outro ponto que o leitor deve estar atento é o uso indevido da palavra Internet. Algumas vezes a palavra é utilizada com o significado de WWW (World Wide Web). Outras vezes, é empregada como referência à arquitetura de ligações dos enlaces dos computadores na rede. Por último, verificamos o uso da palavra como o conjunto de protocolos TCP/IP. Por causa dos motivos expostos, o termo Internet foi regulamentado em Outubro de 1995, nos Estados Unidos, pelo Federal Networking Council (FNC). A definição foi formulada de acordo com consultas realizadas com as principais entidades envolvidas no desenvolvimento da tecnologia Internet. Em adição, também participaram as comunidades de direitos de propriedade intelectual (Intellectual Property Rights – IPR). Apresentamos a seguir o resultado final da proposta (veja referência FNC (1995)).

A Internet refere-se a um sistema global de informação que:

- É logicamente ligado por um único conjunto de endereços globais baseados no protocolo IP (Internet Protocol) ou nas suas subsequentes extensões/implementações futuras.
- Está apto para o suporte de comunicação usando o conjunto de protocolos da arquitetura *Transmission Control Protocol/Internet Protocol* (TCP/IP) e suas subsequentes extensões/implementações futuras, e/ou outros protocolos compatíveis com o IP.
- Fornece, utiliza ou faz acessível, tanto de forma privada quanto pública, os serviços de alto nível baseados na comunicação e infraestrutura relativa descrita neste documento.

A arquitetura *Transmission Control Protocol/Internet Protocol* (TCP/IP) é ilustrada na Figura 5.15, apresentando seus inúmeros protocolos distribuídos nas quatro camadas segundo o modelo de referência TCP/IP.

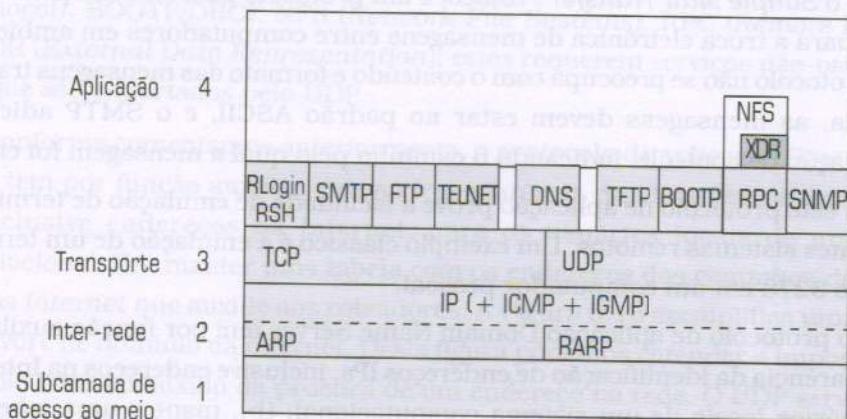


Figura 5.15 Arquitetura *Transmission Control Protocol/Internet Protocol* (TCP/IP).

Camada de Aplicação

A *camada de aplicação*, que corresponde ao nível 4 da arquitetura TCP/IP, é caracterizada por protocolos que solicitam à camada de transporte os serviços orientados e não-orientados à conexão. No primeiro caso, o TCP prove o suporte às aplicações de uma maneira confiável. Por outro lado, os protocolos de aplicação que solicitam serviços não-orientados são atendidos pelo UDP.

Os protocolos BGP (*Border Gateway Protocol*), FTP (*File Transfer Protocol*), HTTP (*Hypertext Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), Telnet, DNS (*Domain Name Server*) e RPC (*Remote Procedure Call*) são exemplos de protocolos que fazem solicitação de serviços orientados à conexão ao TCP. A seguir, apresentamos um sumário com a função desses protocolos que solicitam serviços ao TCP:

- BGP: o protocolo *Border Gateway Protocol* (BGP) tem por função o roteamento de datagramas entre distintos sistemas computacionais de rede em ambientes TCP/IP. Entendemos como distintos sistemas computacionais de rede as redes de computadores de distintas corporações. Estes ambientes são denominados de *sistemas autônomos* (SA). Geralmente, dentro de um sistema autônomo utiliza-se um protocolo de *roteamento interno* do tipo OSPF (*Open Shortest Path First*), e, entre os SAs, um protocolo de *roteamento externo* como o BGP.
- FTP: utilizando-se do protocolo TCP, o *File Transfer Protocol* efetua a transferência de arquivos binários e textos de uma maneira confiável entre sistemas computacionais distintos com arquitetura TCP/IP.

- HTTP: o protocolo *HyperText Transfer Protocol* é considerado um dos motivos do sucesso da Web. Este protocolo de comunicação permite a transferência de arquivos de servidores numa rede TCP/IP de maneira amigável através do uso de *hyperlinks*.
- SMTP: o *Simple Mail Transfer Protocol* é um protocolo padronizado pelas RFCs 821 e 822 para a troca eletrônica de mensagens entre computadores em ambientes TCP/IP. O protocolo não se preocupa com o conteúdo e formato das mensagens transferidas. Todavia, as mensagens devem estar no padrão ASCII, e o SMTP adiciona uma informação de controle, indicando o caminho pelo qual a mensagem foi enviada.
- Telnet: este protocolo de aplicação prove a facilidade de emulação de terminais entre diferentes sistemas remotos. Um exemplo clássico é a emulação de um terminal IBM modelo 3278 em um computador pessoal.
- DNS: o protocolo de aplicação Domain Name Server tem por função auxiliar: (a), na transparência da identificação de endereços IPs, inclusive endereços na Internet, para os usuários locais de um sistema computacional; (b), manter uma tabela com os endereços dos caminhos de algumas redes na Internet que auxilie aos roteadores. A Figura 5.16 exemplifica uma parte de uma árvore de domínio da Internet. Nesta figura, podemos entender a importância do protocolo DNS no auxílio da procura de um endereço na rede. O DNS utiliza o protocolo TCP para requerer uma transferência confiável de uma grande quantidade de informações para sua tabela.

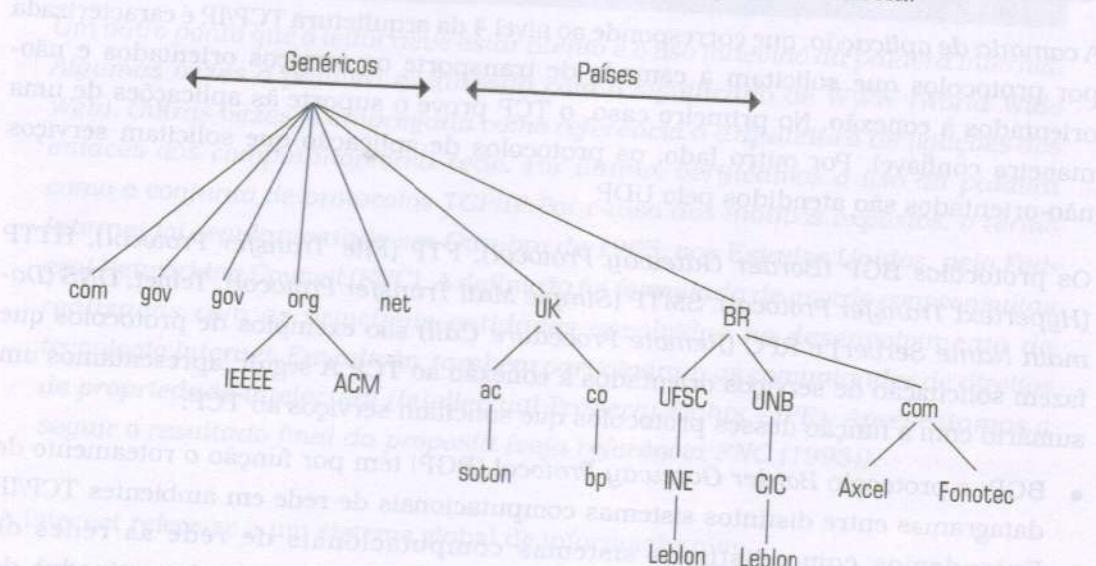


Figura 5.16 Exemplo de parte de um domínio da Internet.

- RPC: uma *Remote Procedure Call* é um mecanismo muito popular em ambientes distribuídos onde são efetuadas as chamadas remotas de procedimentos com a função de encapsular a comunicação entre duas aplicações numa rede. A comunicação entre a aplicação cliente e servidora pode ser efetuada de maneira orientada à conexão ou

- RPC: uma *Remote Procedure Call* é um mecanismo muito popular em ambientes distribuídos onde são efetuadas as chamadas remotas de procedimentos com a função de encapsular a comunicação entre duas aplicações numa rede. A comunicação entre a aplicação cliente e servidora pode ser efetuada de uma maneira orientada à conexão ou não-orientada à conexão. Por esta razão, este protocolo de aplicação pode solicitar serviços ao TCP (conexão orientada) ou UDP (conexão não-orientada). A Figura 5.17 exemplifica uma comunicação via RPC.

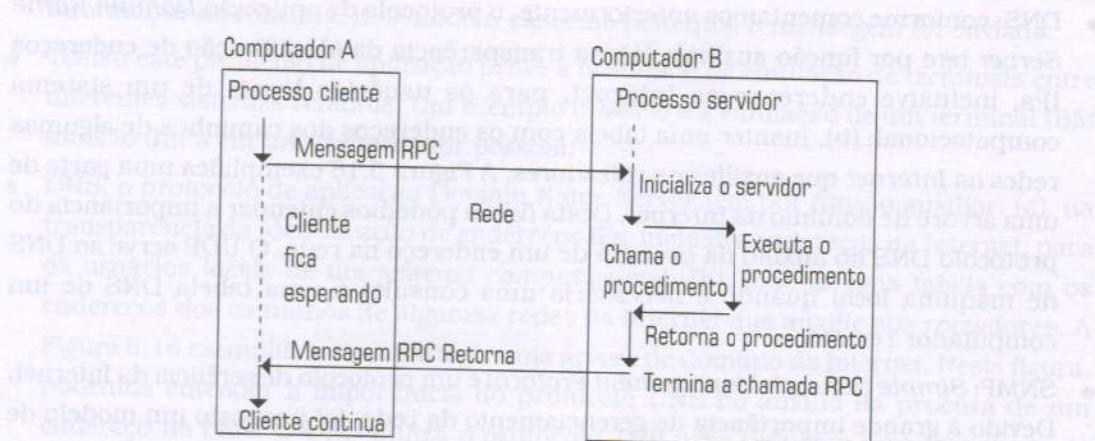


Figura 5.17 Comunicação via RPC.

- XDR: o protocolo *External Data Representation* provê uma transparência para os programadores de aplicações distribuídas no tocante ao formato dos dados que devem ser trocados entre computadores com diferentes formatos de dados. Para atingir esta transparência, o protocolo faz uma codificação própria para representação independente de um tipo específico de computador. Quando uma troca é efetuada entre dois processos em diferentes computadores, com distintas formas de representação, o primeiro processo faz uma chamada ao XDR, que converte sua representação para forma especificada do XDR. No destino, ocorre o processo inverso, no qual o protocolo XDR remoto é chamado para fazer a conversão do seu formato para a representação local.

NOTA

É interessante o leitor notar que, quando um protocolo de aplicação estiver fazendo uso do protocolo UDP, a pilha de protocolos utilizada é UDP/IP e não TCP/IP. Esta observação ilustra nossa preocupação com algum formalismo quanto a definições, como é o caso da Internet. Devido ao uso extensivo do protocolo TCP, é muito comum o usuário se referir à pilha TCP/IP. Todavia, existem casos em que o TCP não está sendo utilizado.

Camada de Transporte

A camada 3 na arquitetura TCP/IP (não confundir com a camada 3 do modelo RM-OSI, que representa o nível de rede) é o nível dos protocolos de transporte. Os dois protocolos que atualmente fazem parte da arquitetura são o TCP e o UDP (lembre-se da definição da Internet, na qual explicitamente é dito que novos protocolos podem ser propostos no futuro para complementar as funções dos protocolos existentes).

Transmission Control Protocol (TCP)

O TCP é um protocolo caracterizado por oferecer um serviço confiável entre aplicações. Com o objetivo de efetuar suas tarefas com sucesso, o protocolo identifica os pacotes recebidos fazendo uma correlação de cada pacote com suas respectivas conexões. Exemplos de serviços providos pelo TCP são a identificação dos pacotes, a correção numa eventual perda de pacotes e a garantia da seqüência de entrega dos pacotes.

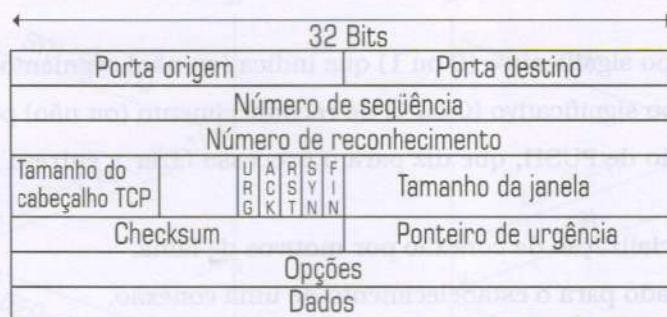


Figura 5.18 Formato de um pacote TCP.

A PDU do TCP é conhecida por *segmento*. Vamos, também, utilizar a palavras *pacote* para representar a PDU do TCP. O formato do pacote TCP é ilustrado, na Figura 5.18, em que temos os seguintes campos:

- Portas de origem e destino (16 bits cada campo): são as identificações dos processos de origem e de destino envolvidos numa conexão TCP. Os números das portas obedecem a uma padronização no sentido de que algumas portas são reservadas.

A reserva no número das portas significa que estas podem ser usadas em qualquer implementação com o mesmo significado. Exemplo de portas padrões TCP são 5 (*Remote Job Entry* – aplicativo de submissão remota em computadores de grande porte), 21 (FTP), 23 (Telnet), 25 (SMTP), 53 (*Domain Name Server* – DNS), 80 (World Wide Web Http), 88 (Kerberos), 108 (SNA Gateway Server) e 179 (BGP). A Figura 5.19 ilustra um exemplo de multiplexação que o TCP realiza para a entrega de pacotes para as suas respectivas portas.

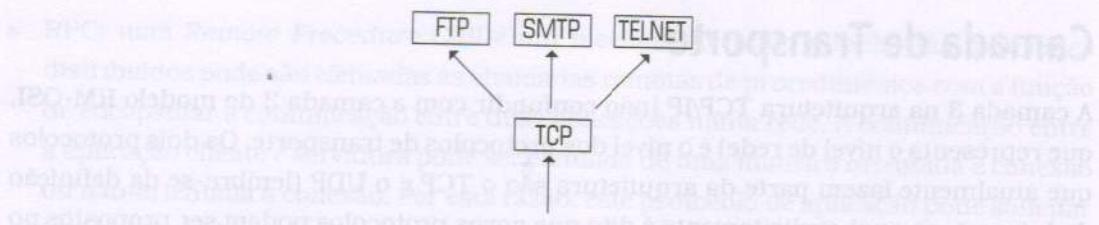


Figura 5.19 Multiplexação a nível do TCP.

- Número de seqüência (32 bits): é o número de um segmento numa conexão TCP, com exceção quando o pacote é SYN. Quando um segmento começa com SYN, esta é a primeira representação da seqüência e o primeiro octeto de dados vem a seguir.
- Número de reconhecimento positivo de pacotes (32 bits): é o número do reconhecimento dos segmentos ACKs.
- Tamanho do cabeçalho (4 bits): informa o número de palavras de 32 bits que existem no segmento TCP.
- Flags (6 bits):
 - URG – campo significativo (0 ou 1) que indica (ou não) segmento urgente.
 - ACK – campo significativo (0 ou 1) de reconhecimento (ou não) positivo.
 - PSH – função de PUSH, que diz para o processo fazer a entrega imediata para a aplicação.
 - RST – reinicialização de conexão por motivos de falha.
 - SYN – utilizado para o estabelecimento de uma conexão.
 - FIN – utilizado para finalizar uma conexão, nenhum dado será mais enviado pelo remetente.
- Window Size (16 bits): é o tamanho da janela para o controle de fluxo.
- Checksum (16 bits): é o campo de verificação do cabeçalho através do cálculo baseado em todos os campos do segmento.
- Urgent point (16 bits): este campo permite que o destinatário saiba quantos dados urgentes serão enviados.
- Options: projetado para prover serviços extras, como por exemplo o tamanho máximo de um segmento que deverá ser aceito.

Um vez que já conhecemos o formato e as funções de um pacote TCP, é interessante saber como é efetuado o mecanismo de comunicação entre duas entidades TCP. A comunicação TCP é caracterizada pelas seguintes etapas:

- Abertura de uma conexão.
- Envio e recebimento de dados.
- Obtenção de informações sobre a conexão.
- Fechamento da conexão.

A abertura de uma conexão TCP é efetuada empregando-se uma técnica chamada de *three-way handshake*. Nesta técnica, apresentada na Figura 5.20 (a), um processo no computador A envia um segmento de abertura de conexão (SYN) a um outro segmento X. No computador B, um segundo processo recebe os pacotes do computador A e responde com os segmentos de reconhecimento positivo (ACK $X+1$) e um outro segmento de sincronização de abertura de conexão Y. Finalmente, o computador A retorna para o computador B um pacote de reconhecimento positivo (ACK $Y+1$).

Por outro lado, o fechamento da conexão TCP emprega quatro segmentos, como ilustrado na Figura 5.20 (b). O computador A envia um pacote de fim de conexão (FIN). O computador B retorna um reconhecimento positivo do primeiro segmento (ACK $X+1$) e logo após solicita o fechamento de sua conexão com A (FIN). O computador A retorna ao B seu reconhecimento positivo de fechamento de conexão.

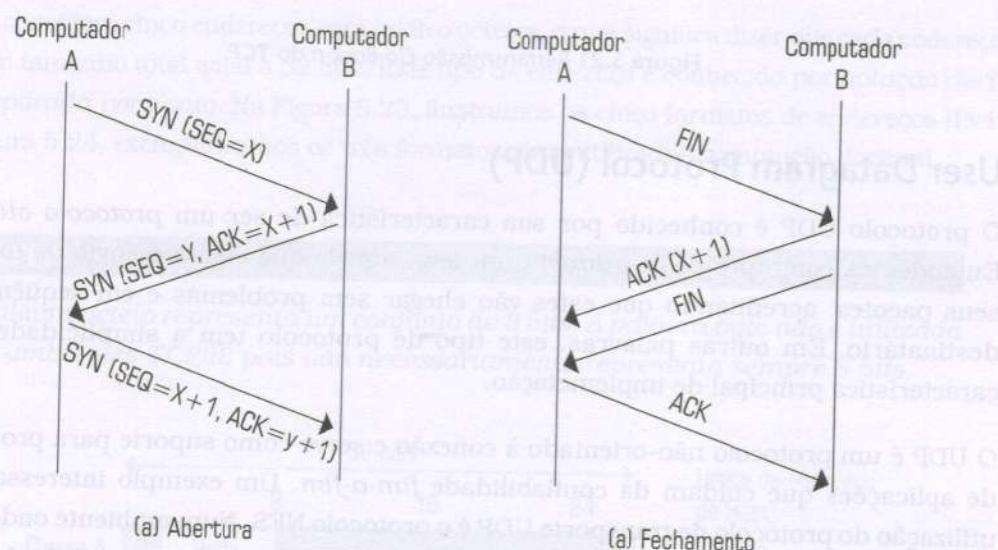


Figura 5.20 Exemplo de (a) abertura e (b) fechamento TCP.

O TCP é conhecido por ser um protocolo *pessimista*, uma vez que o mesmo acredita que no envio dos segmentos sempre irão ocorrer perdas e que os pacotes vão chegar fora de ordem. Por causa desta abordagem, o protocolo tem uma implementação complexa. Quando ocorre uma perda de segmentos, o TCP adota uma política *Go-Back-n*. Nesta abordagem, exemplificada na Figura 5.21, quando um pacote n de uma janela de transmissão é perdido, todos os pacotes de n até o final da janela deverão ser retransmitidos pelo remetente. Em outras palavras, considerando-se uma transmissão de uma janela de tamanho igual a 500 pacotes, em uma eventual perda do segmento 10, todos os pacotes de 10 até 500 serão novamente transmitidos.

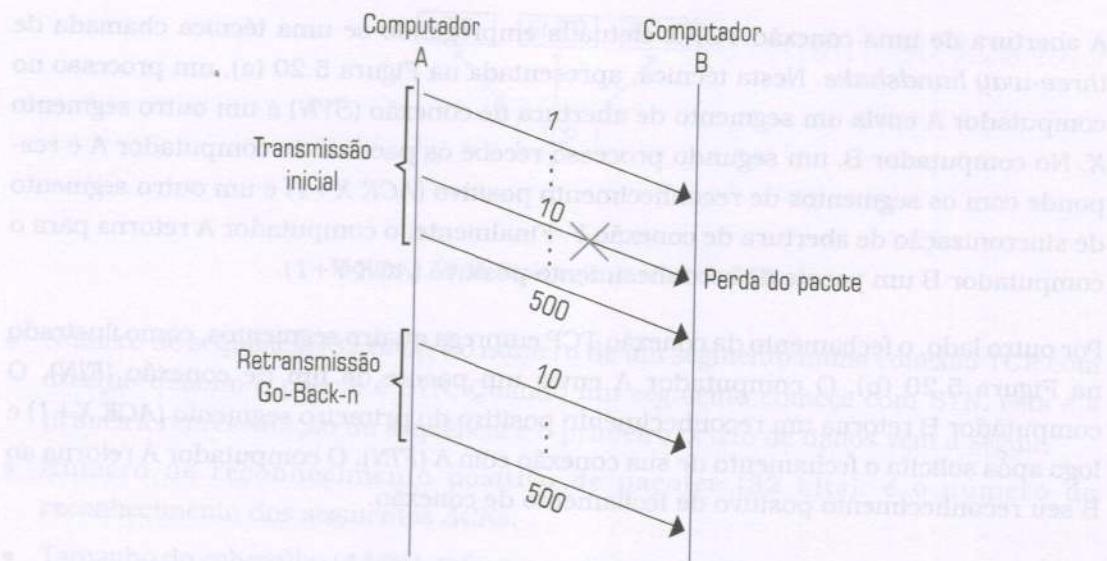


Figura 5.21 Retransmissão Go-Back-n do TCP.

User Datagram Protocol (UDP)

O protocolo UDP é conhecido por sua característica de ser um *protocolo otimista*. Entendemos como protocolo *otimista* (ou leve) aquele que efetua o envio de todos os seus pacotes, acreditando que estes vão chegar sem problemas e em sequência no destinatário. Em outras palavras, este tipo de protocolo tem a simplicidade como característica principal de implementação.

O UDP é um protocolo não-orientado à conexão e serve como suporte para protocolos de aplicações que cuidam da confiabilidade *fim-a-fim*. Um exemplo interessante de utilização do protocolo de transporte UDP é o protocolo NFS. Num ambiente onde existe o protocolo NFS, a garantia de consistência das informações compartilhadas é de responsabilidade do NFS. Por outro lado, a responsabilidade do UDP é fazer uma comunicação rápida entre os computadores envolvidos na transmissão.

A Figura 5.22 apresenta os campos de origem, destinatário, tamanho e campo de controle de um pacote UDP.

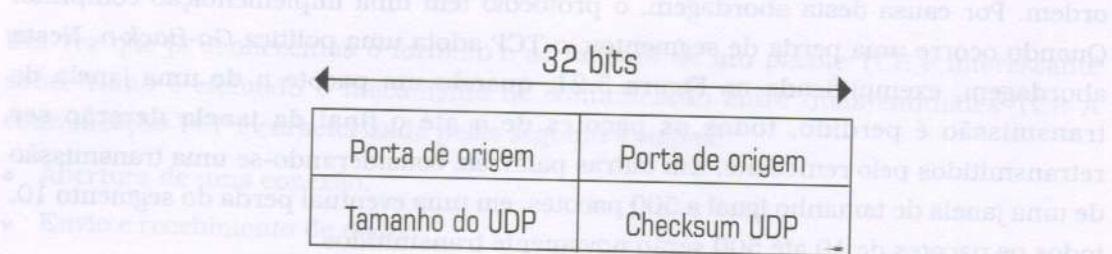


Figura 5.22 Formato de um pacote UDP.

Camada de Inter-rede

A camada de *inter-rede* é o segundo nível da arquitetura TCP/IP. Os principais protocolos alocados nesta camada são o IP (*Internet Protocol*), o ICMP (*Internet Control Message Protocol*), IGMP (*Internet Group Management Protocol*), o ARP (*Address Resolution Protocol*) e o RARP (*Reverse Address Resolution Protocol*).

Internet Protocol (IP)

O IP é o principal protocolo do nível de *inter-rede* na arquitetura TCP/IP. O endereçamento IP é o responsável pelo roteamento em ambientes de redes TCP/IP. A versão do protocolo IP, utilizada atualmente na Internet, é a versão IPv4. Todavia, já existe uma nova proposta do protocolo que visa atacar os problemas encontrados na versão atual. Esta nova implementação é conhecida como IPv6, e vamos abordá-la ao final desta seção.

O IPv4 considera cinco endereços com quatro octetos, o que significa dizer que cada endereço tem um tamanho total igual a 32 bits. Este tipo de endereço é conhecido por *notação decimal separada por ponto*. Na Figura 5.23, ilustramos os cinco formatos de endereços IPv4. Na Figura 5.24, exemplificamos os três formatos mais utilizados na notação decimal.

NOTA

A palavra octeto representa um conjunto de 8 bits. A palavra byte não é utilizada nos ambientes TCP/IP, pois não necessariamente representa sempre 8 bits.

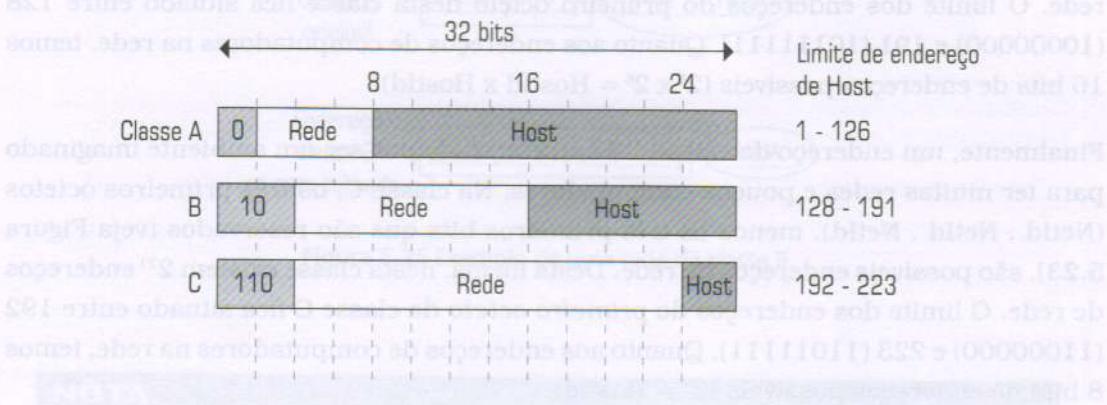


Figura 5.23 Formatos dos endereços IPv4.

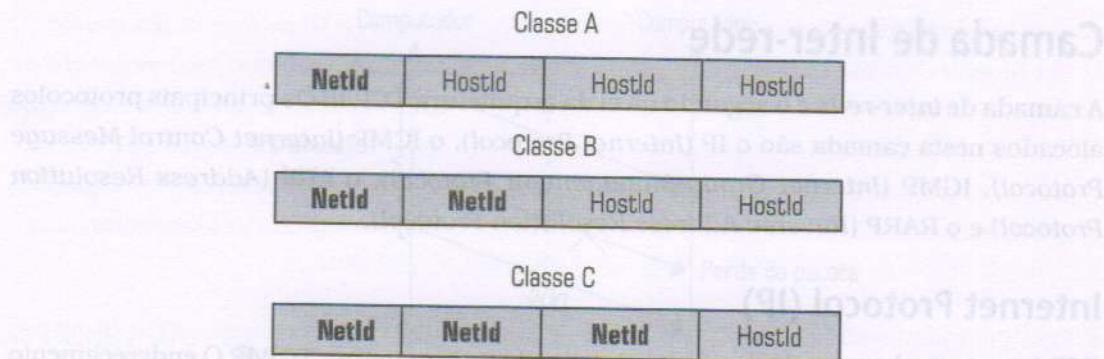


Figura 5.24 Endereços classes A, B e C.

O endereço da classe A foi imaginado para um ambiente no qual teríamos poucas redes e uma grande quantidade de computadores. Nesta classe de endereço, dispomos só o primeiro octeto (NetId), menos o primeiro dígito que é reservado (veja Figura 5.23), de possíveis endereços de rede. Então, dispomos de 2^7 endereços de rede. Os endereços de rede 0 (00000000) e 127 (01111111) são reservados, o que resulta no limite de 1 a 126 como endereços válidos. Quanto aos endereços de computadores na rede, temos 24 bits de endereços possíveis ($2^8 \times 2^8 \times 2^8 = \text{HostId} \times \text{HostId} \times \text{HostId}$).

O endereço da classe B foi projetado para um ambiente no qual teríamos uma quantidade equivalente no número de redes e de computadores. A classe B dispõe os dois primeiros octetos (NetId . NetId), menos os dois primeiros bits que são reservados (veja Figura 5.23), de possíveis endereços de rede. Assim, nesta classe existem 2^{14} endereços de rede. O limite dos endereços do primeiro octeto desta classe fica situado entre 128 (10000000) e 191 (10111111). Quanto aos endereços de computadores na rede, temos 16 bits de endereços possíveis ($2^8 \times 2^8 = \text{HostId} \times \text{HostId}$).

Finalmente, um endereço da classe C é caracterizado por ser um ambiente imaginado para ter muitas redes e poucos computadores. Na classe C, os três primeiros octetos (NetId . NetId . NetId), menos os três primeiros bits que são reservados (veja Figura 5.23), são possíveis endereços de rede. Desta forma, nesta classe existem 2^{21} endereços de rede. O limite dos endereços do *primeiro octeto* da classe C fica situado entre 192 (11000000) e 223 (11011111). Quanto aos endereços de computadores na rede, temos 8 bits de endereços possíveis ($2^8 = \text{HostId}$).

NOTA

O leitor deve compreender que a definição das classes A, B e C foi proposta num projeto pioneiro de arquitetura de rede, o TCP/IP. A realidade mostrou alguns outros caminhos na utilização dos endereços, no tocante à relação-de quantidade de redes e computadores. Um outro aspecto a ser observado é que o endereçamento IPv4 estabelece dois níveis de endereços. No primeiro nível, existe

a atribuição formal da IANA (Internet Assigned Number Authority), órgão gestor da Internet para atribuição de números na Internet. No caso dos endereços de rede, a IANA atribui um único NetId na Internet para a corporação que solicitou sua entrada na rede. No segundo nível, cada instituição é responsável pela administração local dos seus HostIds. Vamos estudar mais adiante a versão IPv6, na qual o estabelecimento de níveis é um problema a ser solucionado.

O endereçamento IPv4 é composto por dois níveis macros, *NetId* e *HostId*. O primeiro nível é composto pelo endereço de rede (*NetId*), que é fornecido pela IANA (Internet Assigned Number Authority), órgão gestor da Internet para atribuição de endereços. Neste nível, não existe flexibilidade quanto à manipulação de endereços. Em outras palavras, um único endereço no âmbito de toda a Internet é fornecido à organização e este não pode ser trocado. Por outro lado, o segundo nível (*HostId*) é de responsabilidade da organização. Desta forma, a atribuição dos endereços fica por conta da autoridade local da corporação. Um exemplo bastante clássico é o uso do endereçamento da classe B. O terceiro octeto, que corresponde ao primeiro *HostId*, pode ser usado como endereço de *subrede*. Em outras palavras, o primeiro endereço de *HostId* é utilizado para a criação de *subredes* na empresa. O roteamento dos datagramas é responsabilidade da pessoa (ou grupo) responsável pela rede da empresa. Para fora da rede da empresa, na classe B, aquele endereço significa *HostId*. Para a empresa, na verdade aquele endereço é de *subrede*. Na Figura 5.25, apresentamos um caso de uso de uma rede da classe B e o uso do terceiro octeto como endereço de *subrede*.

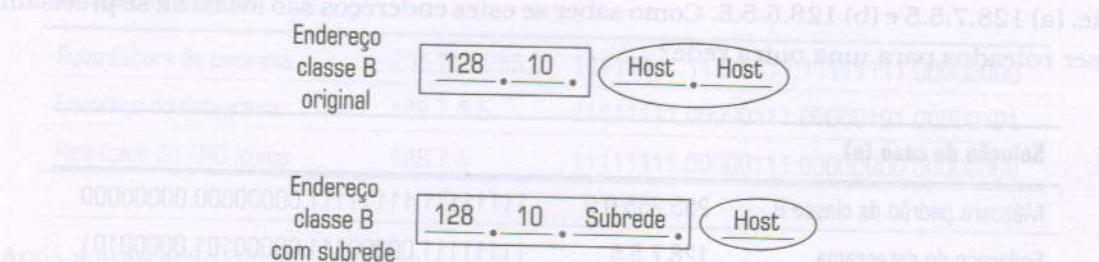


Figura 5.25 Exemplo de uma rede da classe B.

NOTA

Nosso exemplo da rede classe B de uso de subredes pode ser também empregado nas classes A e C. No caso da classe A, é possível o uso a partir do segundo octeto, uma vez que somente o primeiro é NetId. Por outro lado, na classe C, somente parte do último octeto pode ser usado para criar subrede. Por outro lado, como temos na classe C somente os últimos 8 bits (HostId) para serem utilizados para subrede, podemos usar uma quantidade de bits para indicar uma subrede e o restante para indicar computadores na rede.

O conceito de máscara de endereçamento é uma abordagem existente no protocolo IP com o objetivo de melhoria de desempenho no roteamento dos datagramas. Uma máscara é uma técnica que ajuda a determinar se o endereço de um datagrama é local ou se precisa de um roteamento para uma outra rede. Aplicando um AND lógico com os endereços da máscara e do datagrama, fazemos uma eliminação do endereço de HostId. Em outras palavras, resta apenas o endereço de rede. De posse deste resultado, fica fácil saber se é necessário (ou não) efetuarmos um roteamento do datagrama. A Tabela 5.1 ilustra as classes A, B e C de endereços IPv4 e suas máscaras padrões.

Tabela 5.1 Máscaras padrões das classes A, B e C.

Classe de Endereço	Máscara Padrão (binária)	Máscara Padrão (decimal)
A	11111111.00000000.00000000.00000000	255.0.0.0
B	11111111.11111111.00000000.00000000	255.255.0.0
C	11111111.11111111.11111111.00000000	255.255.255.0

Exemplo A

Imagine que uma empresa chamada Grid Computing tenha um endereço de rede da classe B igual a 128.7. Um determinado computador da rede da empresa recebe a solicitação de envio de dois datagramas. Os endereços dos datagramas são, respectivamente, (a) 128.7.5.5 e (b) 128.6.5.5. Como saber se estes endereços são locais ou se precisam ser roteados para uma outra rede?

Solução do caso (a)

Máscara padrão da classe B	255.255.0.0	11111111.11111111.00000000.00000000
Endereço do datagrama	128.7.5.5	11111111.00000111.00000101.00000101
Resultado do AND lógico	128.7	11111111.00000111.00000000.00000000

Após a aplicação da máscara padrão, o protocolo IP do computador entende que o datagrama é para ser entregue para um nó local. Em outras palavras, o endereço está na rede local e basta um broadcast para que o datagrama alcance seu destino.

Solução do caso (b)

Máscara padrão da classe B	255.255.0.0	11111111.11111111.00000000.00000000
Endereço do datagrama	128.6.5.5	11111111.00000100.00000101.00000101
Resultado do AND lógico	128.6	11111111.00000100.00000000.00000000

Após a aplicação da máscara padrão, o protocolo IP do computador entende que o datagrama não deverá ser entregue para um computador local. Em outras palavras, o endereço está em outra rede, e precisa de um nó que faça o roteamento do datagrama para que este alcance seu destino.

O conceito de máscaras também é empregado para as subredes, o que nos leva às submáscaras. Na utilização das submáscaras, o administrador local (ou o grupo de administração) da empresa fica responsável por designar determinados nós na configuração da rede para que sirvam como elementos de roteamento. No próximo exemplo, apresentamos uma situação que deverá auxiliar a compreensão no uso de submáscaras.

Exemplo B

Imagine que uma empresa chamada Grid Computing tenha um endereço de rede da classe B igual a 128.7. O administrador local estabeleceu o terceiro octeto com endereço de subrede e ainda que a submáscara é igual a 255.255.255. Ainda com relação à rede da empresa, existem as subredes 128.7.5 e 128.7.6.

Um determinado computador da subrede 128.7.5 da empresa recebe a solicitação de envio de dois datagramas. Os endereços dos datagramas são, respectivamente, (a) 128.7.5.5 e (b) 128.7.6.5. Como saber se estes endereços são locais à subrede 128.7.5 ou se precisam ser roteados para a subrede 128.7.6?

Solução do caso (a)

Submáscara da empresa	255.255.255.0	11111111.11111111.11111111.00000000
Endereço do datagrama	128.7.5.5	11111111.00000111.00000101.00000101
Resultado do AND lógico	128.7.5	11111111.00000111.00000000.00000000

Após a aplicação da submáscara padrão, o protocolo IP do computador entende que o datagrama é para ser entregue para um nó local da subrede 128.7.5. Em outras palavras, o endereço está na subrede local e basta um broadcast para que o datagrama alcance seu destino.

Solução do caso (b)

Submáscara da empresa	255.255.255.0	11111111.11111111.11111111.00000000
Endereço do datagrama	128.7.6.5	11111111.00000100.00000101.00000101
Resultado do AND lógico	128.7.6	11111111.00000100.00000000.00000000

Após a aplicação da submáscara padrão, o protocolo IP do computador entende que o datagrama não deverá ser entregue para um computador na subrede local 128.7.5. Em outras palavras, o endereço está em outra subrede (128.7.6), e precisa de um nó que faça o roteamento do datagrama para que este alcance seu destino.

Um outro aspecto interessante sobre o IPv4 é a forma que o protocolo implementa a multiplexação dos datagramas recebidos. Na Figura 5.26, é demonstrado como o protocolo IP sabe que um determinado datagrama deve ser encaminhado para o protocolo ICMP (1), IGMP (2), TCP (6), UDP (17), RSVP (46) e OSPF (89). Em outras palavras, reconhecendo o número que vem dentro do campo protocolo do datagrama, o IP está apto para redirecioná-lo para o protocolo correto.

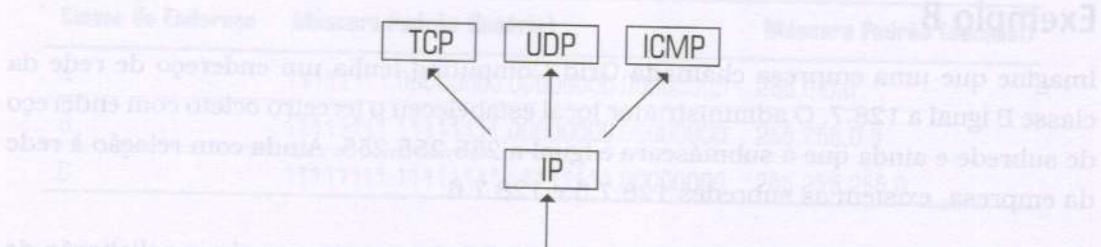


Figura 5.26 Multiplexação do protocolo IP.

Internet Control Message Protocol (ICMP)

O protocolo ICMP tem por objetivo prover mensagens de controle na comunicação entre nós num ambiente de rede TCP/IP. Então, o ICMP permite que roteadores enviem mensagens de erros (ou controle) para outros roteadores (ou nós). Exemplos de mensagens são: a máquina está ativa, destinatário não atingível, tempo excedido pelo datagrama e problema de parâmetro no datagrama. Na Figura 5.27, apresentamos o encapsulamento do ICMP, indicando algumas de suas mensagens.

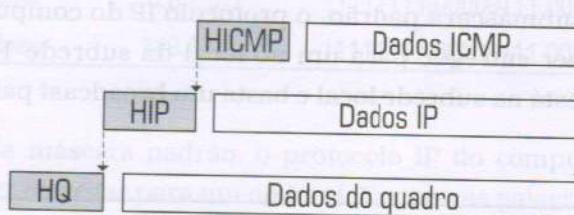


Figura 5.27 Encapsulamento do ICMP.

Address Resolution Protocol (ARP)

Na arquitetura TCP/IP, a comunicação entre processos requer o conhecimento dos endereços IP e de porta. Todavia, o leitor deve saber que em qualquer arquitetura de protocolos o

endereço físico (ou endereço MAC – Medium Access Control) é aquele que efetivamente promove a comunicação através dos meios físicos. Assim, para uma comunicação entre processos, é necessário, também, que os endereços físicos sejam conhecidos.

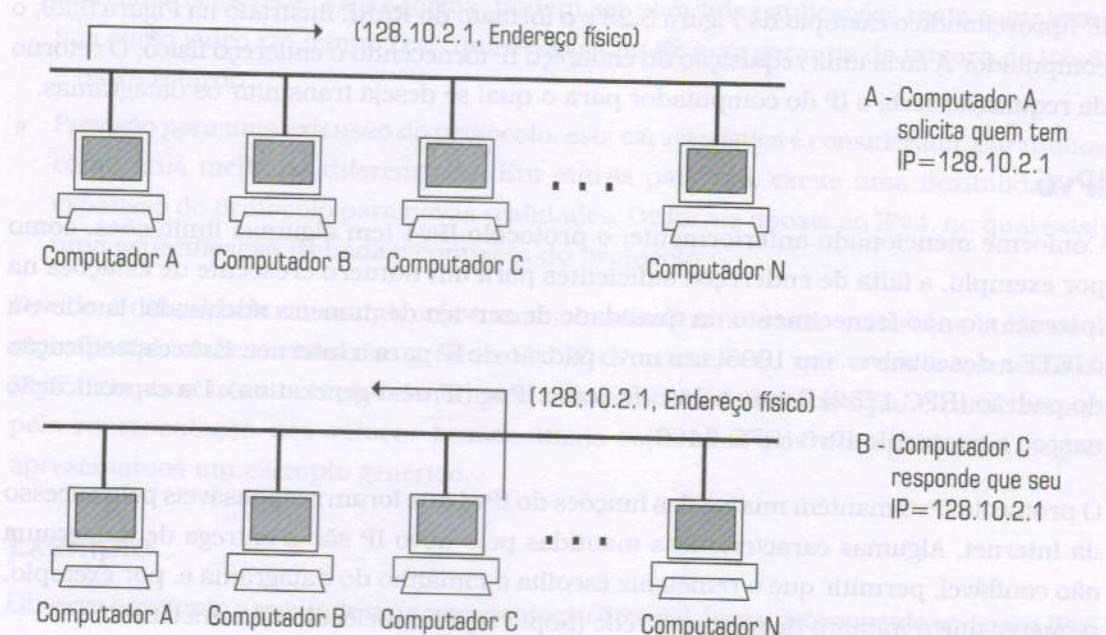


Figura 5.28 Funcionamento do protocolo ARP.

O protocolo ARP tem por função o mapeamento de endereços IP para endereços físicos (endereços de MAC) de rede. De outra forma, podemos dizer que o ARP tem por função a resolução de endereços físicos, uma vez fornecidos endereços IPs. A forma de resolução adotada pelo ARP está representada na Figura 5.28. Neste exemplo, um computador A envia uma mensagem no formato de endereçamento broadcast perguntando qual computador da rede tem um determinado IP. O computador que tem o IP solicitado responde seu endereço físico. A partir do recebimento do endereço físico, o computador A pode atender à solicitação de comunicação do processo. A Figura 5.29 demonstra o formato do ARP e RARP.

32 Bits		
End. Hardware	Tamanho End. Hardware	Protocolo
Endereço de hardware origem	Endereço IP origem	Operação
Endereço de hardware origem	Endereço IP origem	Endereço Hardware Destino
Endereço IP origem	Endereço Hardware Destino	Endereço IP Destino
Endereço Hardware Destino		
Endereço IP Destino		

Figura 5.29 Formato do ARP/RARP.

Reverse Address Resolution Protocol (RARP)

O paradigma de resolução do protocolo RARP é o inverso do ARP. Para a comunicação entre os processos existe a informação do endereço físico; todavia, não está disponível o endereço IP. Aproveitando o exemplo da Figura 5.28 e o formato do RARP, ilustrado na Figura 5.29, o computador A faria uma requisição do endereço IP fornecendo o endereço físico. O retorno da requisição seria o IP do computador para o qual se deseja transmitir os datagramas.

IPv6

Conforme mencionado anteriormente, o protocolo IPv4 tem algumas limitações, como por exemplo, a falta de endereços suficientes para um número crescente de estações na Internet e o não fornecimento da qualidade de serviço de maneira adequada. Isto levou o IETF a desenvolver, em 1995, um novo padrão de IP para a Internet. Esta especificação do padrão (RFC 1752) ficou conhecida como IPng (IP next-generation). Da especificação nasceu o protocolo IPv6 (RFC 2460).

O protocolo IPv6 mantém muitas das funções do IPv4 que foram responsáveis pelo sucesso da Internet. Algumas características mantidas pelo novo IP são a entrega de datagrama não confiável, permitir que o remetente escolha o tamanho do datagrama e, por exemplo, requerer que o número de saltos na rede (hops) seja estabelecido no remetente.

Apesar de alguma similaridade entre os protocolos IPv4 e IPv6, em vários outros pontos o novo protocolo adota mudanças significativas. Exemplo de modificações são o tamanho de endereçamento e algumas facilidades adicionais. Um bom exemplo é a maior flexibilidade para o uso da facilidade de QoS (Qualidade de Serviço).

Podemos agrupar as modificações implementadas no IPv6 em cinco grandes grupos:

- **Maior endereço:** é uma das características mais marcantes da nova versão do IP. O IPv6 quadruplicou o número de octetos do IPv4 para o IPv6. Desta forma, o tamanho de endereço passou de 32 bits para 128 bits (esta expansão foi projetada para atender um futuro ainda não imaginado). Em outras palavras, no IPv6 temos 16 octetos. Um endereçamento de 16 octetos representa 2^{128} valores válidos de endereços. Em outras palavras, o tamanho de endereços é da ordem de 3.4×10^{38} . Se os endereços fossem alocados à taxa de um milhão de endereços a cada microsegundo, teríamos que ter 20 anos para que todos os endereços fossem distribuídos. Da idéia abaixo, pode-se imaginar a quantidade de endereços que o novo protocolo pretender cobrir: "cada pessoa no globo poderá ter endereços suficientes para ter a sua própria Internet tão grande quanto a Internet atual".
- **Formato de cabeçalho flexível:** o formato do datagrama IP não é compatível com o antigo datagrama IPv4. A abordagem foi implementar um formato com uma série de cabeçalhos adicionais, diferente do IPv4, no qual é usado um tamanho fixo para o formato onde os campos têm tamanhos fixos.

- Opções melhoradas: as opções disponíveis no IPv6 são mais poderosas quando comparadas com o IPv4.
- Suporte para reserva de recursos: existe um mecanismo que permite a alocação prévia de recursos da rede. Desta forma, podem ser atendidas aplicações mais complexas, tais como vídeo em tempo real, que necessitam de uma garantia de largura de banda e baixo retardo.
- Previsão para uma extensão do protocolo: esta característica é considerada, por muitos, como uma melhoria diferenciada. Em outras palavras, existe uma flexibilidade de expansão do protocolo para novas realidades. De forma oposta ao IPv4, no qual existe uma especificação fechada e completa do protocolo.

Devido à dificuldade natural de nós humanos trabalharmos com endereços binários (e em especial números binários grandes), o IETF estabeleceu que o endereçamento do IPv6 teria uma nova notação, denominada de *colon hex*. Esta notação é caracterizada pela representação dos valores hexadecimais separados por dois pontos. A seguir, apresentamos um exemplo genérico.

Exemplo

Observe o exemplo a seguir de uma representação decimal de um determinado endereço IPv6.

Representação decimal:

255.254.10.150.128.17.0.0.254.255.254.255.100.140.230.104

A representação no formato colon hex do endereço IPv6 seria:

FFFE:A96:8011:0:FEFF:648C:E668

Ainda com relação ao endereço IPv6, este dispõe de três categorias de endereços básicos:

- Unicast: o endereço especifica a um host simples (computador ou roteador) que o datagrama deverá ser enviado pelo menor caminho.
- Cluster: o destino é um conjunto de computadores que compartilham um único prefixo de endereço (todos conectados a uma mesma rede física). O datagrama deverá ser encaminhado para o grupo e entregue a um membro do grupo (o mais perto possível).
- Multicast: o destino é um conjunto de computadores, possivelmente em locais diferentes. Assim, o datagrama deverá ser entregue para cada membro do grupo multicast usando facilidade de multicast de hardware ou broadcast, se possível.

O formato geral do IPv6 é apresentado na Figura 5.30. Somente o campo do cabeçalho básico é necessário, os demais são opcionais. É interessante notar que o IPv6 *base header* tem menos informação do que um cabeçalho (*header*) do datagrama IPv4. O motivo da redução do cabeçalho foi a remoção dos campos de opções e alguns campos fixos para o *extension headers*. Assim valem os comentários:

- TTL foi substituído pelo HOP LIMIT.
- Service Type foi trocado pelo FLOW LABEL
- Campo do protocolo foi trocado pelo campo NEXT HEADER. Um next header permite que dados com diferentes características tenham diferentes cabeçalhos. Ao invés de adotar um cabeçalho fixo, como é a abordagem do IPv4, o IPv6 disponibiliza a facilidade de diferentes cabeçalhos.
- Campo HEADER LENGTH foi substituído pelo campo PAYLOAD LENGTH.

Version	Header Length	Type of service	Total Length
Identification		D M F F	Fragment Offset
Time to live	Protocol	Header Checksum	
Endereço origem			
Endereço destino			
Opções			

(a) Formato do IPv4

Versão	Priority	Flow Label
Tamanho dos dados	Next Header	Hop Limit
Endereço origem		
Endereço destino		

(b) Formato do IPv6

Figura 5.30 Formato dos datagramas (a) IPv4 e (b) IPv6.

O novo campo *Flow Label* permite que os pacotes que tenham que ter um tratamento diferenciado sejam assim tratados. O campo tem tamanho de 20 bits, composto pelo endereço de origem e IP destino, permitindo que os roteadores mantenham o estado durante o fluxo, ao invés de estimar a cada novo pacote. As aplicações são obrigadas a gerar um *flow label* a cada nova requisição. A reutilização do *flow label* é permitida quando um fluxo já está terminado ou quando o mesmo foi fechado. A utilização do campo *flow label* provê aos roteadores uma maneira fácil de manter as conexões e de manter o fluxo de tráfego numa mesma taxa.

A utilização do campo *priority* fornece aos processos a facilidade de identificar a necessidade de tráfego que estes precisam. O uso efetivo, ou normalização, de como este campo, junto com o *flow label*, deve plenamente operar, ainda está em discussão. O campo de 8 bits destinado à classe está, no momento, em nível de desenvolvimento. Todavia, os 4 bits de prioridade podem nos ajudar a entender o que poderemos ter pela frente.

A forma de distribuição dos endereços tem gerado muitas discussões, que ficam concentradas em dois pontos principais:

1) Como fazer a gerência de distribuição dos endereços?

Esta discussão é baseada em qual autoridade deve ser criada para gerenciar a distribuição de endereços. Na Internet atual, temos dois níveis de hierarquia. Em outras palavras, temos um primeiro nível que é responsabilidade da autoridade da Internet (IANA). No segundo nível, temos a responsabilidade de cada empresa. O IPv6 permite múltiplos níveis. Existe uma proposta em tipos de níveis do IPv6 semelhante a do IPv4. Uma proposta é apresentada no exemplo da Figura 5.31.

010	Provider ID	Subscriber ID	Subnet ID	Node ID
-----	-------------	---------------	-----------	---------

Figura 5.31 Proposta de níveis para o IPv6.

Legenda:

- 010 – tipo de endereço; no caso 010 é um endereço que diz o tipo de provedor auferido.
- Provider ID – identificação do provedor.
- Subscriber ID – identificador do assinante.
- Subnet ID – informação da rede do assinante.
- Node ID – informação sobre um nó do assinante.

2) Como mapear um endereço para um destino?

Esta pergunta deve ser respondida com o desempenho com meta. De outra forma, a eficiência computacional deverá ser levada em conta. Independente de autoridades na rede, um datagrama deverá ser analisado e os melhores caminhos deverão ser escolhidos.

NOTA

Podemos inferir, após nossa introdução ao protocolo IPv6, que possivelmente a migração de toda a Internet para esta nova proposta de endereçamento ainda deverá durar muito anos.

Encapsulamento

Para finalizar esta seção, exemplificamos na Figura 5.32 como é efetuado o encapsulamento das unidades de protocolos (PDUs) na arquitetura TCP/IP. Neste exemplo, partimos de uma mensagem enviada por protocolo de aplicação até um quadro apropriado para ser enviado numa rede local.

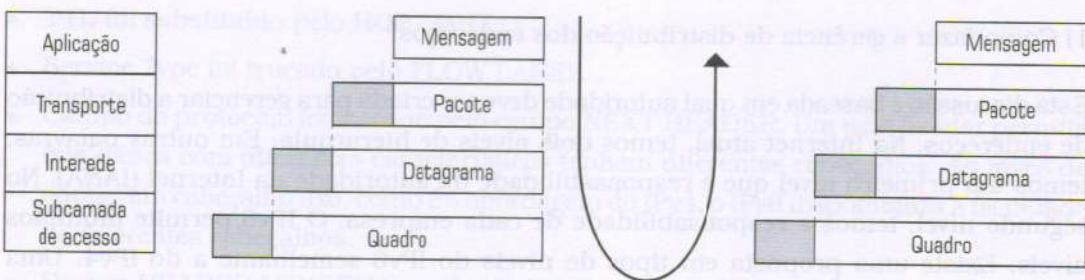


Figura 5.32 Encapsulamento na arquitetura TCP/IP.

A Arquitetura IEEE

Como já mencionamos, a falta de padrões no âmbito das camadas físicas e de enlace, levou a Sociedade de Computação do IEEE a propor uma série de especificações. Na Figura 5.33, podemos observar que a camada física apresenta os padrões 802.3 (Ethernet), 802.4 (Token-Bus), 802.5 (Token-Ring) e 802.6 (DQDB). Cada um destes padrões (que serão discutidos no capítulo de redes locais) cobre as funções físicas estabelecidas no modelo de referência RM-OSI. Em outras palavras, os meio de transmissão (exemplo são os cabos coaxiais, pares trançados e fibra óptica) e as especificações das placas de rede são as funções englobadas neste nível.

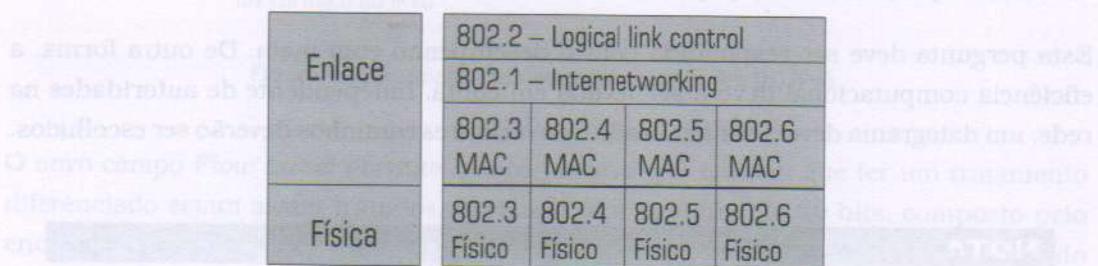


Figura 5.33 Modelo IEEE e a comparação com o RM-OSI.

Em termos de enlace do padrão IEEE 802, como ilustrado na mesma figura, compreendem as funções de LLC (*Logical Link Control*) e MAC (*Medium Access Control*). O Controle de Enlace Lógico (ou LLC) tem por função prover tantos pontos de acesso de serviços (SAPs – *Service Access Points*) quantos forem solicitados pelos usuários de rede. Quanto ao MAC, podemos ter diferentes implementações de tecnologias de acesso ao meio. Exemplos são CSMA/CD (802.3), Token-Bus (802.4), Token-Ring (802.5) e DQDB (802.6).

NOTA

É interesse observar que alguns estudantes e profissionais fazem confusão entre as funções respectivas dos meios físicos e dos protocolos de enlace. Um exemplo clássico é a afirmação de que um cabo Ethernet só admite o protocolo CSMA/CD. Um cabo Ethernet pode ser utilizado por outro protocolo de acesso ao meio que não seja o CSMA/CD. Outra confusão é aquela que diz que o Ethernet é o padrão IEEE 802.3. O padrão Ethernet foi proposto pela Digital, Intel e Xerox (DIX) e tem diferenças em relação ao padrão IEEE 802.3.

Exercícios

- 1) Defina protocolo, modelo e arquitetura de protocolos.
- 2) Considerando uma possível classificação dos protocolos, diga quais características você julga importante serem levadas em conta?
- 3) Explique o funcionamento dos mecanismos de fragmentação e remontagem.
- 4) Quanto ao controle de conexão, qual a diferença fundamental entre uma conexão orientada e outra não-orientada?
- 5) O que significa endereço de nível e escopo?
- 6) Quais os modos de endereçamento que os protocolos geralmente utilizam?
- 7) Qual a função da facilidade de serviços diferenciados?
- 8) Apresente e critique o modelo de referência TCP/IP.
- 9) Apresente e critique o modelo de referência RM-OSI.
- 10) Apresente o modelo IEEE detalhando sua função e importância.
- 11) Quanto à arquitetura TCP/IP, apresente os principais protocolos em suas respectivas camadas.
- 12) Qual a função da camada de aplicação? Apresente dois exemplos de protocolos.
- 13) Comente as principais diferenças dos dois protocolos padrões da camada de transporte da arquitetura TCP/IP.
- 14) Descreva a função da camada de rede em uma arquitetura de protocolos.
- 15) Apresente a camada de rede da arquitetura TCP/IP e descreva a função de cada protocolo.
- 16) Explique o endereçamento do IPv4 quanto ao seu formato e níveis.
- 17) Qual a função de uma máscara para o protocolo IP?
- 18) O que significa uma subrede e como esta é implementada na arquitetura TCP/IP?
- 19) Explique o funcionamento dos protocolos:
 - ARP
 - RARP

- ICMP
- NFS
- BOOTP
- DHCP
- XDR
- RPC
- BGP
- OSPF
- SNMP
- DNS

- 20) Qual a responsabilidade do órgão IANA para a Internet?
 21) Qual a função e importância de uma RFC na arquitetura TCP/IP?

Referências

A formalização e o conhecimento mais detalhado dos protocolos, modelos de protocolos e arquitetura de protocolos são de vital importância nas tecnologias de redes de comunicação e computadores. Estes três elementos compõem a espinha dorsal das redes de comunicação e computadores. A preocupação com este tópico é evidenciada na literatura acadêmica [Comer (2001), Comer (2000), IEEE (1986), Soares (1995), Stallings (1997), Stallings (1999), Stevens (1994, 1995, 1996), Tanenbaum (1996)] e profissional [Albuquerque (2001), Stallings (2000), Torres (2001)].

Uma outra referência interessante é relativa à programação na arquitetura TCP/IP. A documentação sobre este assunto pode encontrada em Comer (1996), Comer (1994) e Stevens (1999, 1998).

Finalmente, quanto às RFCs, que são documentos importantes na arquitetura TCP/IP, estas podem ser encontradas na referência IETF (2001).

Bibliografia

- ALBUQUERQUE, F. *TCP/IP Internet – Protocols & Tecnologias 3.ed.* Axcel Books, 2001.
- COMER, D.E. *Internetworking with TCP/IP, Volume I – Principles, Protocols and Architecture.* 3rd ed. Prentice Hall, 2000.
- COMER, D.E. *Internetworking with TCP/IP, Volume II – Design Implementation and Internal.* 2nd ed. Prentice Hall, 1994.

- COMER, D.E. *Internetworking with TCP/IP, Volume III – Client-Server Programming And Applications*. 2nd ed. Prentice-Hall, 1996.
- COMER, D.E. *Computer Networks and Internet*. 3rd ed. Prentice Hall, 2001.
- FNC. Definition of Internet. 1995. <http://www.fnc.gov>.
- HELD, G. *Understanding Data Communications*. New Riders Publishing, 1999.
- IEEE Computer Society. *IEEE Standard 802.1: Overview, Internetworking, and Systems Management*. ANSI/IEEE Standard 802.4, 1986.
- IETF. Internet Engineering Task Force. 2001. <http://www.ietf.org>.
- SOARES, L.F. *Redes de Computadores – Das LANs, MANs e WANs às Redes ATM*, 2. ed. Editora Campus, 1995.
- STALLING, W. *Local and Metropolitan Area Networks*. 6th ed. Prentice Hall, 1999.
- STALLING, W. *Business Data Communications*. 4th ed. Prentice Hall, 2000.
- Stalling, W. *Data and Computer Communications*. 5th ed. Prentice Hall, 1997
- STEVENS, W.R. *TCP/IP Illustrated V.1 – The Protocols*. 1st ed. Addison Wesley, 1994.
- STEVENS, W.R. *TCP/IP Illustrated V.2 – The Implementation*. 1st ed. Addison Wesley, 1995.
- STEVENS, W.R. *TCP/IP Illustrated V.3 – TCP for Transactions, HTTP, NNTP and Unix*. 1st ed. Addison Wesley, 1996.
- STEVENS, W.R. *Unix Networking Programming V.1 – Networking, APIs – Sockets and XTI*. 2nd ed. Addison Wesley, 1998.
- STEVENS, W.R. *Unix Networking Programming V.2 – Interprocess Communication*. 2nd ed. Addison Wesley, 1998.
- TANENBAUM, A.S. *Computer Networks*. 3rd ed. Prentice Hall, 1996.
- TORRES, G. *Redes de Computadores – Curso Completo*. Axcel Books, 2001.