ÍNDICE

1. INTRODUÇÃO

Devido ao tremendo impacto dos computadores na sociedade, principalmente na última década, este período da história tem sido denominado “a era da informação”. O lucro e a produtividade das organizações e indivíduos tem sido aumentados de forma significativa pelo uso das redes de computadores como suporte a troca e acesso à informação. Indivíduos utilizam as redes de computadores quase que diariamente de forma a conduzir suas atividades pessoais e empresariais. O que se pode observar é uma aceleração da utilização e das aplicações com base na tecnologia das redes de computadores, à medida que mais pessoas descobrem as potencialidades dos computadores e das redes de comunicação tanto em aplicações domésticas como em empresariais. As transações e atividades diárias em lojas de departamentos, bancos, e outras pessoas e empresas das mais diversas naturezas, estão cada vez mais dependentes das redes de computadores.

Lamentavelmente, a maioria das redes constitui entidades independentes estabelecidas para atender às necessidades de um grupo isolado. Os usuários selecionam uma tecnologia de hardware que seja adequada aos seus problemas de comunicação. É impossível a estruturação de uma rede universal com base em uma única tecnologia de hardware, já que nenhuma rede única atende a todas as aplicações. Alguns usuários precisam de uma rede de alta velocidade para conectar-se a máquinas, mas essas redes não podem ser expandidas para alcançar grandes distâncias. Outros preferem uma rede de velocidade inferior que faça conexão com máquinas a centenas de milhas de distância.

Ao longo dos anos, as agências governamentais norte-americanas perceberam a importância e o potencial de tecnologia de interligação em redes e vêm financiando as pesquisas que possibilitaram a interconexão global de redes. A tecnologia da ARPA (Advanced Research Projects Agency) inclui um conjunto de padrões de rede que especificam os detalhes do sistema pelo qual os computadores se comunicam, bem como um conjunto de convenções para interconexão em redes e para roteamento. Denominado oficialmente Pilha de Protocolos de

interligação em redes TCP/IP, e geralmente citado como TCP/IP, essa pilha pode ser utilizada para comunicação em qualquer conjunto de redes interconectadas. Algumas empresas, por exemplo, utilizam o TCP/IP para interconectar todas as redes de sua organização, ainda que a empresa não se comunique com redes externas. Outros grupos utilizam o TCP/IP para estabelecer comunicações entre sites geograficamente distantes.

Nos Estados Unidos, a National Science Foundation (NSF), o Department of Energy (DOE), o Department of Defense (DOD), a Health and Human Services Agency (HHS) e a National Aearonautics and Space Administration (NASA) participaram do financiamento da Internet e utilizam o TCP/IP para conectar muitas de suas instalações de pesquisa. A interligação em redes resultante permite que os pesquisadores de instituições conectadas compartilhem informações com seus colegas de todo o mundo com a mesma facilidade com que compartilham informações com pesquisadores da sala ao lado. Um sucesso extraordinário, a Internet demonstra a viabilidade da tecnologia TCP/IP e mostra como pode-se lidar com uma diversidade de tecnologias de redes.

1.1. SERVIÇOS EM REDES

Grande parte da abordagem de serviços terá como foco padrões denominados **protocolos**. Protocolos como TCP e IP fornecem as regras para a comunicação. Eles contém os detalhes de formatos de mensagens, descrevem o que um computador faz ao receber uma mensagem e especificam como um computador trata os erros ou outras condições anormais. De certa forma, um protocolo de comunicação permite que alguém especifique ou entenda uma comunicação de dados sem depender de conhecimentos minuciosos do hardware da rede de um fornecedor específico. Todos os serviços de rede são descritos por protocolos.

Os serviços de aplicativos da Internet mais comum e difundidos incluem:

- *Correio Eletrônico*. O correio eletrônico permite que um usuário elabore memorandos e os envie a indivíduos ou grupos. Uma outra parte do aplicativo do correio eletrônico permite que os usuários leiam os memorandos que receberam. O correio eletrônico tem sido tão bem-sucedido que muitos usuários da Internet dependem dele para correspondência comercial normal. Embora existam muitos sistemas de correio eletrônico, a utilização do TCP/IP faz com que a entrega de correio seja mais confiável, já que não depende de computadores para processamentos intermediários na transmissão de mensagens. Um sistema de entrega de correio TCP/IP opera através de contato direto entre a máquina do transmissor e a máquina do receptor. Assim, o transmissor sabe que quando a mensagem deixa a máquina local, ela foi recebida com êxito no destino.

- *Transferência de arquivos*. Embora alguns usuários às vezes transfiram arquivos através do correio eletrônico, ele se destina, sobretudo, a mensagens de pouco texto. Os protocolos TCP/IP incluem um programa aplicativo que permite que os usuários enviem ou recebam arbitrariamente arquivos externos de programas de dados. Ao utilizar, por exemplo, um programa de transferência de arquivos, a pessoa pode copiar de uma máquina para outra uma base de dados extensa contendo imagens de satélite, um programa escrito em Pascal ou C++, ou um dicionário de inglês. O sistema indica uma maneira de checar os usuários autorizados, ou até de evitar acessos. Tal como ocorre com o correio eletrônico, a transferência de arquivos na interligação em redes TCP/IP é confiável porque as duas máquinas envolvidas comunicam-se diretamente, sem depender de máquinas intermediárias que façam cópias do arquivo ao longo do processo.

- *Login remoto*. O login remoto permite que, de seu computador, um usuário entre em conexão com uma máquina remota e estabeleça uma sessão interativa de login. O login remoto faz com que uma janela na tela do usuário pareça conectar-se diretamente com a máquina remota, enviando cada toque no teclado a uma máquina remota e exibindo cada caracter que o computador remoto imprime na janela do usuário. Quando a sessão de login remoto termina, o aplicativo retorna o usuário ao sistema local.

No nível da camada de rede, uma interconexão proporciona extensos tipos de serviços que todos os programas aplicativos utilizam:

- *Serviço de entrega de pacotes sem conexão*. Este serviço, explicado com detalhes ao longo do texto, forma a base para todos os serviços de interligação em redes. A entrega sem conexão constitui uma preocupação do serviço oferecido pela maioria das redes distribuidoras de encomendas. Isso simplesmente significa que a interligação em redes TCP/IP promove o roteamento de pequenas mensagens de uma máquina para outra, com base nas informações do endereço contidas na mensagem. Como o serviço sem conexão promove o roteamento de cada pacote separadamente não há garantia de entrega, e nem de entrega na mesma ordem na qual os pacotes forma transmitidos. Já que quase sempre há um mapeamento direto para o hardware, o serviço sem conexão é extremamente eficiente. O mais importante é que a entrega de pacotes, sem conexão, como base para todos os serviços de interligação em redes, torna os protocolos TCP/IP adaptáveis a uma ampla gama de hardware de redes.

- *Serviço de transporte de streams confiáveis.* A maioria dos aplicativos precisa de muito mais do que uma entrega de pacotes, porque eles exigem que o software de comunicação corrija automaticamente erros de transmissão, pacotes perdidos, ou falhas de comutações ao longo do caminho entre o transmissor e o receptor. O serviço de transporte confiável trata desses problemas. Ele permite que um aplicativo de um computador estabeleça uma "conexão" com um aplicativo de outro computador, e a seguir envie um grande volume de dados através da conexão, como se fosse uma conexão de hardware direta e permanente. Naturalmente, em um nível mais baixo, os protocolos de comunicação dividem a cadeia de dados em mensagens curtas e as envia, uma de cada vez, esperando que o receptor confirme a recepção.

Muitas redes oferecem serviços básicos semelhantes aos mencionados acima, de modo que alguém poderia questionar o que diferencia os serviços TCP/IP de outros. As principais características diferenciadoras são:

- *Independência da tecnologia de redes*. Embora o TCP/IP seja baseado em tecnologia convencional de comutação de pacotes, ele é independente do hardware de qualquer fornecedor específico. A Internet inclui diversas tecnologias de rede, desde as redes projetadas para operar em um prédio até as projetadas para cobrir grandes distâncias. Os protocolos TCP/IP definem a unidade de transmissão de dados denominada **datagrama**, e especifica como transmitir datagramas em uma rede específica.

- *Interconexão universal*. Uma interligação em redes TCP/IP permite a comunicação de que qualquer par de computadores ao qual ela é conectada. A cada computador é atribuído um endereço universalmente reconhecido por toda a interligação em redes. Cada datagramas traz os endereços de sua origem e de seu destino. Os computadores de comutação intermediária utilizam o endereço de destino para tomar decisões sobre roteamento.

- *Confirmações fim-a-fim.* Os protocolos de interligação em redes TCP/IP fornecem uma confirmação entre a origem e o destino final, e não entre máquinas sucessivas ao longo do caminho, mesmo quando as duas máquinas não se conectam a uma mesma rede física.

- *Padrões de protocolo de aplicativos.* Além dos serviços básicos no nível de transporte (como conexões de streams confiáveis), os protocolos TCP/IP incluem padrões para muitos aplicativos comuns, inclusive o correio eletrônico, a transferência de arquivos e o login remoto. Assim, quando estão desenvolvendo programas aplicativos que utilizam TCP/IP, os programadores sempre descobrem que o software existente oferece os serviços de comunicação de que eles precisam.

1.2. ESCOPO DA INTERNET

A arquitetura TCP/IP surgiu com a criação de uma rede patrocinada pelo Departamento de Defesa dos Estados Unidos. Uma das tarefas essenciais dessa rede seria manter comunicados, mesmo que apenas uma parte, órgãos do governo e universidades, numa ocorrência de guerras ou catástrofes que afetassem os meios de comunicação daquele país. Dessa necessidade, surgiu a ARPANET, uma rede que permaneceria intacta caso um dos servidores perdesse a conexão. A ARPANET necessitava então de um modelo de protocolos que assegurasse tal funcionalidade esperada, mostrando-se confiável, flexível e de fácil implementação. É então desenvolvida a arquitetura TCP/IP, que se torna um padrão de fato. A ARPANET cresceu e tornou-se a rede mundial de computadores – Internet. A utilização (e facilidades) do padrão TCP/IP utilizado pelos fabricantes de outras redes, com a finalidade da conectividade com a Internet. A normalização do TCP/IP chegou após a sua utilização em massa.

A Internet cresceu, abrangendo centenas de redes individuais localizadas nos Estados Unidos e na Europa. Conectou aproximadamente 20.000 computadores de universidades, órgãos públicos e laboratórios de pesquisa organizacional. O tamanho e a utilização da Internet continuou em ascensão muito mais acelerada do que o previsto. No final de 1987, estimou-se que o crescimento alcançara 15% ao mês. Em torno de 1994, a Internet alcançava mais de 3 milhões de computadores em 61 países.

A utilização de protocolos TCP/IP e o crescimento da Internet não se limitaram a projetos financiados pelo governo. Grandes companhias voltadas para o setor de computadores conectaram-se à Internet, bem como muitas outras organizações de grande porte como companhias de petróleo, indústria automobilística, empresas de eletrônica, companhias farmacêuticas e portadoras de telecomunicações. As empresas de pequeno e médio porte começaram a conectar-se na década de 1990. Além disso, muitas outras utilizavam os protocolos TCP/IP em suas interligações em redes corporativas, mesmo tendo optado por não participar da Internet.

Uma expansão acelerada trouxe problemas de escala não previstos no projeto original e motivou os pesquisadores a encontrar técnicas para gerenciar numerosos recursos distribuídos. No projeto original, por exemplo, os nomes e endereços de todos os computadores conectados à Internet eram mantidos em um único arquivo que era editado manualmente e, a seguir, distribuído a todos os sites da Internet. Em meados da década de 1980, tornou-se óbvio que um banco de dados de origem não seria suficiente. Primeiro, os pedidos para atualização de arquivos rapidamente provocaria excesso do pessoal disponível para processá-los. Segundo, ainda que existisse um arquivo-fonte correto, a capacidade da rede seria insuficiente para permitir a distribuição freqüente para todos os sites ou o acesso on-line a cada site.

Novos protocolos foram desenvolvidos e um sistema de atribuição de nome foi colocado em vigor através da Internet para permitir que qualquer usuário automaticamente determinasse o nome de uma máquina remota. Conhecido como Domain Name System (DNS), o mecanismo conta com máquinas denominadas servidoras de nome para responder a consultas sobre nomes. Nenhuma máquina contém todo o banco de dados de nomes de domínios. Em vez de uma máquina, os dados são distribuídos por um conjunto de máquinas que utilizam protocolos TCP/IP para se comunicarem entre si quando estiverem respondendo a uma consulta. Tão logo a Internet tornou-se popular e os usuários passaram a buscar informações através de serviços como Gopher e a World Wide Web, novamente o tráfego aumentou.

A Internet não é controlada por nenhum órgão governamental ou comercial, mas sim por organizações voluntárias que controlam os usuários e os artigos publicados na Internet. Eis algumas organizações:

IAB A IAB (Internet Advisory Board) é constituída de várias organizações e seu objetivo principal é coordenar a organização geral da Internet.

InterNIC A InterNIC (Internet Network Information Center) foi criado pela NSF para distribuir endereços IP.

IRTF O IRTF (Internet Research Task Force) é um dos comitês que constituem a IAB. Ele é responsável por várias atividades a nível de pesquisa, como o desenvolvimento de protocolos.

RFC RFC (Request for Comments) são documentos técnicos relacionados aos protocolos da Internet. Alguns deles contém padrões para os protocolos, outros são recentemente desenvolvidos, podendo obter sucesso e se tornarem padrões. Esses documentos formam a documentação da Internet.

FNC FNC (Federal Networking Council) é um comitê que exerce a parte informativa da Internet. A FNC realiza o intermédio entre a IAB e as instituições governamentais, além de prestar suporte a agências no uso da Internet.

IETF IETF (Internet Engeneering Task Force) é um subcomitê da IAB que realiza a manutenção de problemas construtivos e também a implementação de novas tecnologias.

A Internet é considerada por muitos como um dos mais importantes e revolucionários desenvolvimentos da história da humanidade. Pela primeira vez no mundo um cidadão comum ou uma pequena empresa pode (facilmente e a um custo muito baixo) não só ter acesso a informações localizadas nos mais distantes pontos do globo como também – e é isso que torna a coisa revolucionária – criar, gerenciar e distribuir informações em larga escala, no âmbito mundial, algo que somente uma grande organização poderia fazer usando os meios de comunicação convencionais. Isso com certeza afetará substancialmente toda a estrutura de disseminação de informações existente no mundo, a qual é controlada primariamente por grandes empresas. Com a Internet um pessoa qualquer (um jornalista, por exemplo) pode, de sua própria casa, oferecer um serviço de informação baseado na Internet, a partir de um microcomputador, sem precisar da estrutura que no passado só uma empresa de grande porte poderia manter. Essa perspectiva abre um enorme mercado para profissionais e empresas interessadas em oferecer serviços de informação específicos.

2. BÁSICO DE UMA REDE

Independente do tipo de conexão que façam, seja entre computadores ou entre terminais e computadores, as redes de comunicação dividem-se em dois tipos básicos: de comutação de circuitos (também conhecidas como redes baseadas em conexões) e de comutação de pacotes (conhecidas, ainda, como redes sem conexão).

- A comutação por circuitos opera formando uma conexão dedicada entre duas pontas. O sistema telefônico dos Estados Unidos utiliza uma tecnologia de comutação de circuitos – uma chamada telefônica estabelece um circuito da linha de quem telefona, através de uma central de comutação local, passando por linhas do tronco, até uma central de comutação remota e, finalmente, ao destinatário da chamada. Enquanto um circuito estiver aberto, o equipamento telefônico testa o microfone várias vezes, converte os sinais para o formato digital e os transmite através do circuito para o receptor. O transmissor tem a garantia de que os sinais serão distribuídos e reproduzidos, pois o circuito oferece um percurso de dados seguro, de 64 kpbs (mil bits por segundo), o mínimo necessário para o envio de voz digitalizada. A vantagem da comutação de circuitos reside na sua capacidade segura: uma vez que um circuito é estabelecido, nenhuma outra atividade de rede poderá reduzir a capacidade do circuito. A desvantagem da comutação de circuitos é o alto custo: o preço é fixo, independente do tráfego. Por exemplo, o preço de uma ligação telefônica é o mesmo, ainda quando as duas pontas não se comunicam.

- Nas redes de comutação de pacotes, as mensagens a serem transmitidas através das estações da rede são divididas em pequenas unidades chamadas **pacotes** que são multiplexados por meio de conexões entre máquinas de alta capacidade. Um pacote que geralmente contém apenas pequenas unidades de informações transporta uma identificação que capacita o hardware da rede a enviar as informações a determinado destino. Por exemplo, a transmissão de um arquivo extenso entre dois equipamentos deve ser feita a partir da divisão do arquivo em vários pacotes antes de encaminhá-los à rede. O

hardware da rede envia os pacotes aos seus respectivos destinos onde o software os reúne novamente em um único arquivo. A grande vantagem da comutação de pacotes é a possibilidade de realizar simultaneamente várias comunicações entre computadores, com conexões entre equipamentos compartilhados por todos os pares de equipamentos que estão se comunicando. A desvantagem é que à medida que a atividade se intensifica, um determinado par de computadores conectados entre si recebe uma capacidade menor da rede. Ou seja, toda vez que uma rede de comutação de pacotes estiver sobrecarregada, os computadores conectados a ela terão que esperar até poderem enviar pacotes adicionais.

2.1. WANs e LANs

As tecnologias de comutação de pacotes são freqüentemente divididas em duas categorias, de acordo com a extensão: redes de longas distâncias (WANs) e redes locais (LANs). As duas categorias não possuem definições formais. Ao contrário, os fornecedores utilizam os termos de forma coloquial para que os consumidores saibam diferenciar as duas tecnologias.

As tecnologias de rede remota ou de redes de longas distâncias possibilitam a comunicação entre grandes distâncias. A maioria das tecnologias de rede de longas distâncias não impõe um limite na extensão da distância; permite que os dois extremos se comuniquem a uma distância arbitrária. Por exemplo, uma rede de longa distância pode operar em um continente ou conectar computadores de continentes diferentes. Geralmente, as redes de longa distâncias operam em velocidades mais lentas do que as redes locais, e necessitam de um retardo de transmissão bem maior entre as conexões. A velocidade de uma rede de longas distâncias varia de 56 Kbps a 155 Mbps (um milhão de bits por segundo). O retardo de transmissão pode variar desde alguns milissegundos até vários décimos de segundo.

As tecnologias de rede local possuem uma velocidade de conexão entre comutadores bem mais rápida, mas deixam a desejar na capacidade de operar em longas distâncias. Por exemplo, uma rede local típica abrange uma área pequena, como um único edifício ou um

campus, e funciona entre 10 Mbps e 2 Gbps (bilhões de bits por segundo). Já que essas tecnologias operam em pequenas áreas, o retardo de transmissão é bem menor do que o das tecnologias de rede de longas distâncias, o qual pode durar desde alguns décimos de um milissegundo, até no máximo dez milissegundos.

2.2. A TECNOLOGIA ETHERNET

Ethernet é o nome dado a uma tecnologia de rede local popular, de comutação de pacotes; a maioria das empresas de médio e grande porte a utiliza. Cada cabo da Ethernet possui aproximadamente 0,5 polegadas de diâmetro e até 500 metros de comprimento. Para oferecer o máximo de proteção contra interferência elétrica de dispositivos como motores elétricos, o cabo possui uma forte proteção que o torna difícil de ser dobrado. O esquema de fiação funciona perfeitamente quando vários computadores ocupam um mesmo compartimento. O cabo percorre o trajeto diretamente de um computador a outro. Para incluir um novo computador, basta conectá-lo à cadeia.

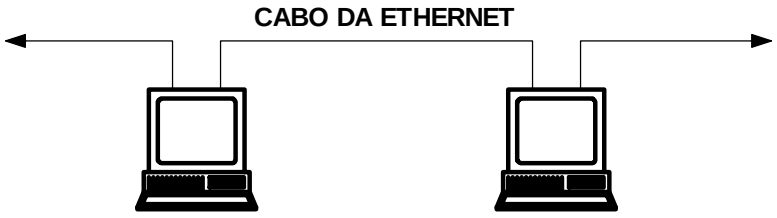

Figura 2.1. Conexão física entre dois computadores.

Com o avanço da tecnologia foi possível construir redes Ethernets que não necessitam da proteção elétrica de um cabo coaxial. Chamada de **Ethernet de pares trançados**, essa tecnologia permite que um computador acesse uma rede Ethernet utilizando um par de fios de cobre normais sem proteção, semelhantes aos utilizados para fazer conexões entre equipamentos telefônicos. A vantagem desse tipo de tecnologia é que, além de reduzir os custos, oferece proteção a outros computadores da rede no caso de um usuário desconectar um único computador. Em alguns casos, uma tecnologia de pares trançados possibilita que

uma instituição utilize a Ethernet com a fiação telefônica já existente, sem a adição de novos cabos. Conhecido tecnicamente como 10Base-T, o esquema de fiação de pares trançados conecta cada computador a um HUB da Ethernet, como ilustra a Figura 2.2.

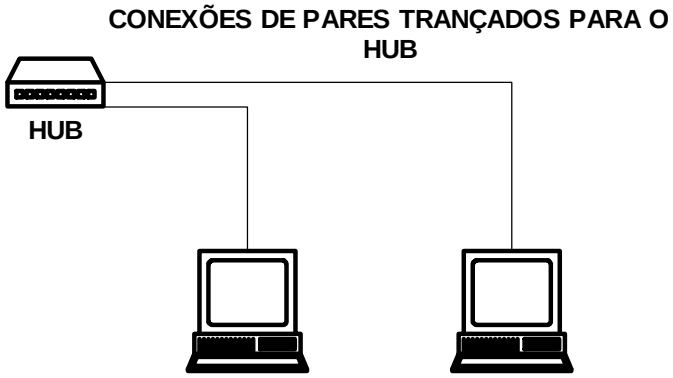

Figura 2.2. Cada computador conecta-se a um HUB por cima de um par de fios.

O HUB é um dispositivo eletrônico que estimula os sinais num cabo Ethernet. Fisicamente, o HUB é formado por uma pequena caixa que geralmente é alojada em um gabinete de fiação; uma conexão entre um HUB e um computador deve ter menos de cem metros de extensão. Um HUB necessita de energia elétrica e, talvez, de pessoal qualificado para fazer o monitoramento e o controle de sua operação na rede. Para a interface com um computador, a conexão a um HUB parece funcionar do mesmo modo que a conexão direta com coaxial.

A Ethernet é uma tecnologia de barramento de difusão de 10 Mbps com método de entrega sem garantia e controle de acesso distribuído. É um **barramento** porque todas as estações compartilham um único canal de comunicação; é de difusão (broadcast) porque todos os transceptores recebem cada uma das transmissões. O controle de acesso é distribuído porque, ao contrário de algumas tecnologias de rede, a Ethernet não possui nenhuma autoridade central para permitir o acesso, vários equipamentos podem acessar a Ethernet simultaneamente e cada um deles estabelece se o meio está ou não livre, detectando a presença ou não de sinal. Quando uma interface do host tem um pacote para transmitir, ela verifica o meio para saber se há alguma mensagem sendo transmitida. Se nenhuma transmissão for detectada, a interface do host inicializa a transmissão. Cada uma das transmissões possui um limite de duração (porque há um tamanho máximo de pacote). Além do

mais, o hardware deve observar um intervalo mínimo de tempo entre as transmissões, o que significa que nenhum par de equipamentos comunicantes pode utilizar a rede sem oferecer aos demais equipamentos uma oportunidade de acesso.

Apesar de um cabo da Ethernet possuir um comprimento máximo, a rede pode ser aumentada de duas maneiras: com o auxílio de repetidores e de pontes.

- O **repetidor** pode ser utilizado para transmitir sinais elétricos de um cabo a outro. Entretanto, no máximo dois repetidores podem ser colocados entre duas máquinas, de modo que o comprimento total de uma única Ethernet continua muito pequeno (três segmentos de 500 metros cada).

- As **pontes** são melhores do que os repetidores porque não repercutem os ruídos, as falhas ou os quadros[ψ] com má formação; um quadro inteiramente válido deve ser recebido antes que a ponte o aceite e o transmita para outro segmento.

De acordo com a visão do TCP/IP, as Ethernets ligadas por pontes são simplesmente uma outra forma de conexão física de rede. O importante é que: *Em virtude de a conexão entre cabos físicos, fornecida pelas pontes e pelos repetidores, ser transparente para os equipamentos conectados à Ethernet, um sistema único de rede física.*

2.3. FDDI (Fiber Distributed Data Interconnect)

A FDDI é uma conhecida tecnologia de rede que opera em pequenas áreas geográficas e oferece uma largura de banda maior do que a Ethernet. Ao contrário da Ethernet e de outras tecnologias de redes locais que utilizam cabos para transportar os sinais elétricos, a FDDI utiliza fibras de vidro e transmite as informações, convertendo-as em feixes de luz.

ψ O termo quadro deriva-se da comunicação por linhas seriais, nas quais o transmissor "configura" a informação, acrescentando sinais especiais antes e após a transmissão das informações.

A fibra óptica possui duas vantagens a mais do que o fio de cobre:

- são imunes à interferência eletromagnética, podendo ficar próximas a dispositivos elétricos potentes;

- como utilizam luz, a quantidade de informação transportadas por um único canal de fibra óptica é significativamente maior do que a dos cabos que transportam sinais elétricos.

A FDDI é uma rede token ring de 100 Mbps dotada de um recurso de auto-reparo É uma rede **em anel** porque forma um circuito fechado, iniciando em um computador, passando por todos os outros, e novamente retornando ao computador de origem.
Trata-se de uma rede de tecnologia token ring porque utiliza um token como forma de controlar a transmissão. Quando a rede está inativa, um quadro especial denominado token passa de estação a estação. Sempre que uma estação tiver que enviar um pacote, ela terá que esperar a chegada de um token, enviar o pacote e depois passar o token à estação seguinte. O token que está sendo utilizado garante o equilíbrio: concede a todas as estações a oportunidade de enviar um pacote antes que outra estação envie um segundo pacote.

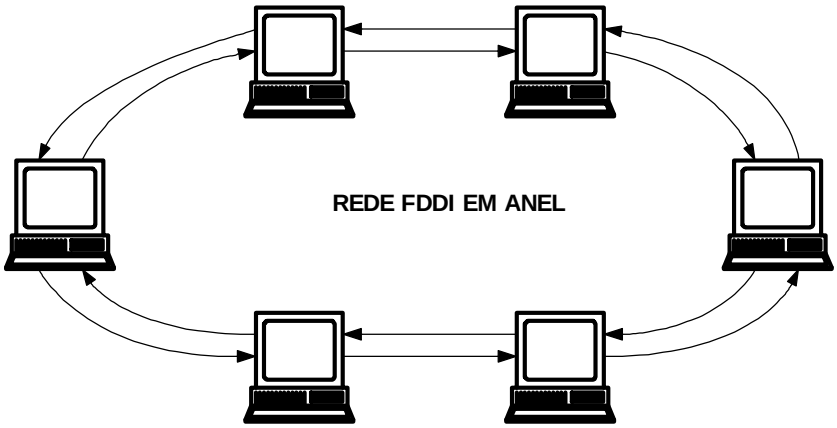

Figura 2.3. Uma rede FDDI com fibras ópticas fazendo a conexão de seis computadores. As setas indicam a direção do tráfego nas fibras e nos computadores conectados entre si.

Para oferecer um recurso de recuperação automática de falhas, o hardware da FDDI utiliza dois anéis independentes que se conectam a cada computador. Quando a interface percebe que não pode comunicar-se com o computador ao lado, o anel de garantia é utilizado para contornar a falha.

A falha talvez seja decorrente da desconexão da fibra (p. ex., um corte acidental). Se as fibras de ambos os anéis seguirem o mesmo percurso físico, é muito provável que a Segunda fibra também seja desconectada. O hardware da FDDI automaticamente utiliza o anel de rotação oposta para formar o circulo fechado na direção em que ele ainda está operando. Com isso, os outros computadores podem continuar se comunicando, mesmo com a ocorrência da falha.

2.4. ATM (Asynchronous Transfer Mode)

O ATM é o nome dado a uma tecnologia de rede de alta velocidade, baseada em conexão, que vem sendo usada tanto nas redes que operam em pequenas como em grandes áreas geográficas. Pelos padrões correntes, as redes de alta velocidade são aquelas que operam a uma velocidade de, no mínimo, 100 Mbps; o ATM pode intercambiar informações com velocidades de gigabit/segundo. É claro que para se obterem velocidades tão altas é necessário um equipamento complexo de última geração. Consequentemente, as redes ATM possuem um custo mais alto do que as demais tecnologias.

Uma rede ATM utiliza técnicas de hardware e software especiais:

- Uma rede ATM é formada por um ou mais comutadores de alta velocidade que são conectados aos computadores host e a outros comutadores ATM.

- O ATM utiliza fibras ópticas para fazer conexões, inclusive conexões entre um computador host e um comutador ATM. As fibras ópticas possuem uma velocidade de transferência

maior do que a dos fios de cobre; normalmente, a conexão entre um host e um comutador ATM opera a uma velocidade de 100 ou 155 Mbps.

- As camadas mais baixas de uma rede ATM utilizam quadros de tamanhos fixos chamados células. Como as células possuem exatamente o mesmo tamanho, o hardware do comutador ATM pode processá-la rapidamente.

A rede ATM difere das redes de comutação de pacotes porque oferece um serviço **baseado em conexão**. Antes de um computador host conectado a um ATM enviar células, ele deve primeiramente interagir com o comutador para especificar o endereço do destinatário. A interação é análoga a uma ligação telefônica[Ψ]. O host especifica o endereço do computador remoto e espera o comutador ATM entrar em contato com o sistema remoto e estabelecer um caminho (rota fixa). Se o computador remoto não aceitar o pedido, não responder ou se o comutador não puder alcançar o computador remoto, o pedido para estabelecer a comunicação falha.

Quando uma conexão é feita, o comutador ATM local escolhe um identificador da conexão e passa-o para o host, juntamente com um mensagem informando o sucesso da conexão. O host utiliza o identificador da conexão ao enviar ou receber células.

Ao terminar a conexão, o host comunica-se novamente com o comutador para que a conexão seja desfeita. O comutador desconecta os dois computadores. A desconexão equivale a tirar um telefone do gancho no final de uma ligação telefônica; após a desconexão, o comutador pode novamente utilizar o identificador da conexão.

[Ψ] Em virtude de a rede ATM ter sido idealizada para transportar voz, assim como dados, há uma forte relação entre o ATM e a comutação telefônica.

2.5. INTERLIGAÇÃO EM REDES

Fisicamente, duas redes só podem ser conectadas por um computador que esteja ligado às duas. No entanto, uma ligação física não fornece a interconexão que imaginamos, porque tal ligação não garante que o computador vai cooperar com outras máquinas que desejam se comunicar. Para se ter uma interconexão de rede viável, necessitamos de computadores que queiram repassar pacotes de uma rede para outra. Os computadores que conectam entre si duas redes e repassam pacotes de um para o outro são chamados de gateways de interligação em redes ou roteadores de interligação em redes. Considere um exemplo que consista de duas redes físicas mostradas na Figura 2.4, o roteador R está conectado às redes 1 e 2. Para R atuar como um roteador, precisa capturar pacotes na rede 1 que são destinados às máquinas na rede 2 e transferi-los e vice-versa.

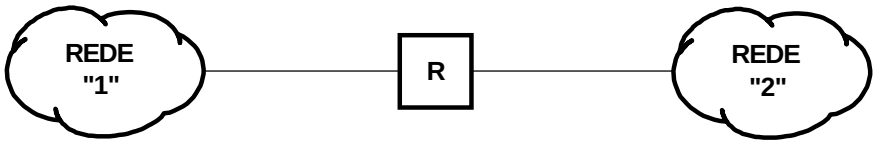

Figura 2.4. Duas redes físicas conectadas entre si por um roteador.

Na figura acima, são usadas nuvens para demonstrar redes físicas, porque o hardware específico não tem importância. Cada rede pode ser uma rede local ou uma rede de longa distância, e cada uma pode ter muitos ou poucos hosts acoplados.

Quando conexões de interligação em redes se tornam mais complexas, os roteadores necessitam saber sobre a topologia da interligação em redes, além das redes às quais estão conectados.

Figura 2.5. Três redes interconectadas por dois roteadores.

Neste exemplo, o roteador R1 deve transferir da rede 1 para a rede 2 todos os pacotes destinados a máquinas em qualquer das redes 2 e 3. Para uma grande interligação em redes composta de muitas redes, a tarefa do roteador de tomar decisões sobre por onde enviar pacotes se torna mais complexa.

Além de roteadores que conectam entre si redes físicas, o software é necessário em cada host para permitir que programas aplicativos usem a interligação em redes como se fosse uma única rede realmente física. Do ponto de vista da interligação em redes, qualquer sistema de comunicação capaz de transferir pacotes é considerado como uma única rede, independente do retardo, da vazão, do tamanho máximo do pacote ou da escala geográfica.

O protocolo da interligação em redes do TCP/IP trata todas as redes do mesmo modo. Uma rede local, como uma Ethernet, uma rede de área maior, como o backbone de ANSNET, ou uma ligação de ponto a ponto entre duas máquinas, cada uma conta como uma rede.

3. ENDEREÇAMENTO

Pensando em uma interligação em redes como uma grande rede igual a qualquer outra rede física. A diferença é que a interligação em redes é uma estrutura virtual, idealizada por seus projetistas e totalmente implantada em software. Assim, os projetistas estão livres para escolher formatos e tamanhos de pacotes, endereços, técnicas de entrega e assim por diante; nada é orientado pelo hardware. Para endereços, os projetistas de TCP/IP optaram por um esquema análogo ao endereçamento de rede física, no qual a cada host da interligação é atribuído um endereço com número inteiro de 32 bits, denominado seu endereço IP. A parte interessante do endereçamento da interligação é que os números inteiros são escolhidos cuidadosamente para tornar o roteamento eficiente. Especificamente, um endereço IP codifica a identificação da rede à qual um host se acopla, assim como a identificação de um único host nessa rede. Resumindo:

"A cada host de uma interligação em redes TCP/IP é atribuído um endereço de interligação em redes único de 32 bits que é usado em todas as comunicações com aquele host."

Os bits dos endereços IP para todos os hosts de uma rede dada compartilham um mesmo prefixo. Conceitualmente, cada endereço é um par (*netid, hostid*) em que *netid* identifica uma rede e *hostid* identifica um host naquela rede. Na prática, cada endereço IP deve Ter uma das três primeiras formas mostradas na Figura 3.1. Dado um endereço IP, seu tipo pode ser determinado a partir de três bits de alta ordem, sendo dois bits suficientes para distinguir entre as três classes principais:

- **Endereços do tipo A**, são usados pelas numerosas redes que não possuem mais de 2^{16} (ou seja, 65.536) hosts, dedicam sete bits para netid e 24 bits para hostid.

- **Endereços do tipo B**, que são usados para redes de tamanho médio que possuem entre 2^8 (ou seja, 256) e 2^{16} hosts, alocam 14 bits para o netid e 16 bits para o hostid.

- **Endereços do tipo C**, que possuem menos de 2^8 hosts, alocam 21 bits para o netid e somente 8 bits para hostid.

O endereço IP foi definido de tal modo que é possível extrair as partes do netid ou do hostid rapidamente. Os roteadores, que usam a parte netid de um endereço ao decidir qual o destino de um pacote, dependem de uma extração eficiente para alcançar velocidade alta.

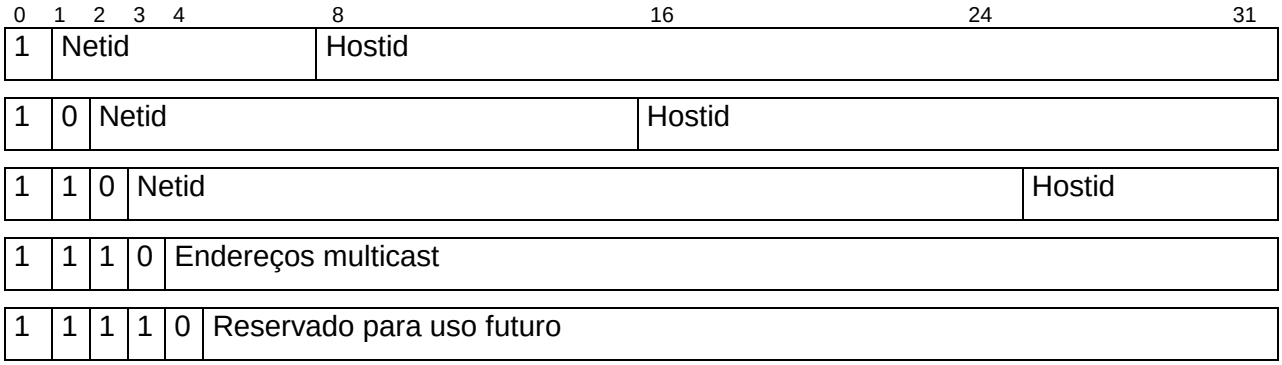

Figura 3.1. As cinco formas de endereços da Internet (IP). As três primeiras formas, classes A, B e C podem ser diferenciadas pelos três primeiros bits.

Quando os computadores convencionais possuem duas ou mais conexões físicas são denominados hosts *multi-homed*. Esses hosts e os roteadores necessitam de endereço IP múltiplos. Cada endereço corresponde a uma das conexões de rede da máquina. Portanto, como os endereços IP codificam não apenas uma rede, como também um host daquela rede, os endereços IP não especificam um computador individual, e sim uma conexão à rede. Assim, um roteador conectando *n* redes tem *n* endereços diferentes de IP, um para cada conexão de rede.

Os endereços IP são escritos como quatro números inteiros decimais separados por pontos decimais, no qual cada número inteiro fornece o valor de um octeto de endereço IP. Assim, o endereço de 32 bits:

10000000 00001010 00000010 00011110

é representado por:

128.10.2.30

Na realidade, a maioria dos softwares TCP/IP que apresenta ou requer uma pessoa para manipular um endereço IP usa a notação decimal com ponto. Assim, compreender a relação entre tipos de endereços IP e números decimais pode ajudar. A tabela da Figura 3.2 resume a escala de valores para cada tipo.

Classe	Endereço mais baixo	Endereço mais alto
A	0.1.0.0	126.0.0.0
B	128.0.0.0	191.255.0.0
C	192.0.1.0	223.255.255.0
D	224.0.0.0	239.255.255.255
E	240.0.0.0	247.255.255.255

Figura 3.2. A escala de valores decimais que correspondem a cada tipo de endereço IP.

3.1. ENDEREÇOS DE REDES E DE BROADCAST

A maior vantagem de codificar informações de rede em endereços de interligação em redes é o roteamento eficiente. Outra vantagem é que os endereços de interligação em redes podem se referir a redes, assim como a hosts. Por convenção, hostid zero nunca é atribuído a um host individual. Em vez disso, um endereço IP com hostid zero é usado para se referir a essa própria rede. O esquema de endereçamento da interligação em redes inclui um endereço de difusão que se refere a todos os hosts da rede. De acordo com o padrão, qualquer hostid que consista de todos os 1s é reservado para difusão. Em muitas tecnologias de rede, difusão pode ser tão eficiente quanto as transmissões normais; em outras, difusão é apoiada pelo software da rede, mas requer uma demora muito mais substancial do que uma transmissão única. Algumas redes não aceitam qualquer difusão. Assim, ter um endereço de difusão de IP não assegura a disponibilidade ou a eficiência da entrega de difusão.

Um endereço de difusão direcionado pode ser interpretado sem ambivalência em qualquer etapa de uma interligação em redes porque identifica, de modo único, a rede de destino, além de especificar difusão naquela rede. Os endereços de difusão direcionados fornecem um poderoso mecanismo que permite a um sistema remoto enviar um pacote único que será

transmitido por difusão na rede especificada. Do ponto de vista de endereçamento, a principal desvantagem da difusão direcionada é que ela requer conhecimento do endereço da rede. Outra forma de endereço de difusão, denominada endereço de difusão limitado ou de rede local, fornece um endereço de difusão para a rede local, independente do endereço atribuído de IP. Consiste em trinta e dois 1s. Um host pode usar os endereços de difusão limitados coo parte de um procedimento padrão antes que ele aprenda seu endereço de IP ou o endereço para a rede local. Entretanto, uma vez que o host aprenda o endereço correto de IP para a rede local, ele deve usar difusão direcionada.

Em geral, o software da interligação em redes interpreta os campos que consistem em 0s (zeros) para significar “este”. Assim, um endereço de IP com hostid igual a zero refere-se a “este” host e outro de netid igual a zero refere-se a “esta” rede. Usar netid zero é especialmente importante nas circunstâncias em que um host deseja comunicar-se em uma rede, mais ainda não sabe o endereço IP dela. O host usa a netid zero temporariamente, e outros hosts da rede interpretam o endereço como “esta” rede.

3.2. ENDEREÇO DE LOOPBACK

A tabela da Figura 3.2 mostra que nem todos os endereços possíveis foram atribuídos a classes. Por exemplo, o endereço 127.0.0.0, num valor da escala da classe A, é reservado para *loopback*, e é utilizado no teste TCP/IP e para a comunicação na máquina local. Quando algum programa usa o endereço de loopback como destino, o software de protocolo retorna os dados sem enviar o tráfego através de qualquer rede. Um pacote enviado a um endereço 127 da rede não deve aparecer em nenhuma rede. Além disso, um host ou um roteador nunca deve difundir informações sobre roteamento ou alcance para o número de rede 127; este não é um endereço de rede.

3.3. PONTOS FRACOS NO ENDEREÇAMENTO

Codificar informações de rede em um endereço pode ter suas desvantagens. A mais óbvia delas é que os endereços referem-se às conexões de redes, não ao host: *se um host se move de uma rede para outra, seu endereço IP deve mudar.*

Outra falha do esquema de endereçamento é que, quando qualquer rede tipo C cresce além de 255 hosts, deve ter seu endereço mudado para um endereço tipo B. Apesar de isto parecer um problema menor, mudar endereços de redes pode tomar muito tempo e ser difícil para depurar. Coo muitos softwares não são projetados para trabalhar com endereços múltiplos para a mesma rede física, os administradores não podem planejar uma transição tranqüila, na qual introduzam novos endereços lentamente. Ao contrário, devem interromper o uso de um endereço de rede, mudar os endereços de todas as máquinas e, então, recuperar a comunicação usando o novo endereço de rede.

Saber apenas um endereço de IP, para determinado destino, talvez não seja suficiente; pode ser impossível alcançar o destino usando tal endereço. Considere o exemplo da interligação em redes mostrado na Figura 3.3. Nesta figura, dois hosts, A e B, conectam-se à rede 1 quase sempre se comunicam diretamente usando aquela rede. Assim, os usuários do host A normalmente devem referir-se ao host B usando endereço IP I_3. Existe outro caminho de A até B, através do roteador R, e é usado sempre que A envia um pacote com endereço IP I_5 (endereço de B na rede 2). Agora, suponha que a conexão de B para a rede 1 falhe, mas a máquina em si continue trabalhando. Os usuários em A, que especificam endereços IP I_3, não alcançam B apesar de os usuários que especificam endereços I_5 poderem alcançar.

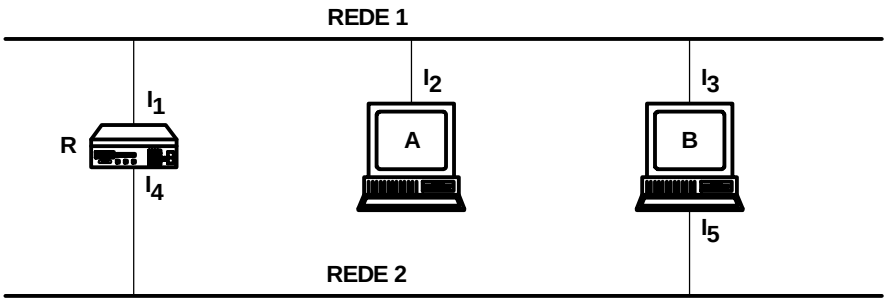

Figura 3.3. Um exemplo de interligação em redes com um host de multi-homed, B, que indica um problema com o esquema de endereçamento IP. Se a interface I_3 é desconectada, A deve usar o endereço I_5 para alcançar B, enviando pacotes através do roteador R.

3.4. UM EXEMPLO

Para esclarecer o esquema de endereçamento IP, considere um exemplo de duas redes de Universidades conectadas à IBPI-NET. A Figura 3.4 mostra os endereços de redes e ilustra como os roteadores conectam redes entre si.

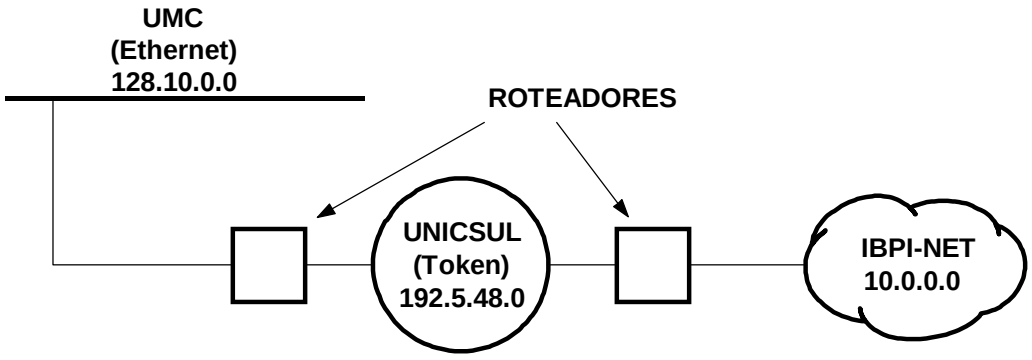

Figura 3.4. A conexão lógica de duas redes ao provedor IBPI-NET.

O exemplo mostra três redes e os números de rede que lhes foram designados: a IBPI-NET (10.0.0.0), a UMC (128.10.0.0) e a UNICSUL (192.5.48.0). De acordo com a tabela da Figura 3.2, os endereços têm classes A, B e C, respectivamente.

A Figura 3.5 mostra as mesmas redes com host, e endereços Internet designados para cada conexão de rede. Quatro hosts cognominados HOST1, HOST2, HOST3 e HOST4 acoplam-se às redes; ROUTER1 é um roteador que conecta a IBPI-NET e a rede token ring (UNICSUL), e ROUTER2 é um roteador que conecta a rede UNICSUL à Ethernet (UMC). O HOST2 tem conexões tanto na Ethernet quanto na rede token ring, podendo, assim, alcançar destinos diretamente em qualquer rede. Apesar de um host de multi-homed, como HOST2, poder ser configurado para rotear pacotes entre as duas redes, a maioria dos sites usa computadores dedicados como roteadores, a fim de evitar sobrecarregar sistemas de computação convencionais com o processamento requerido pelo roteamento. Na figura, um roteador dedicado, ROUTER2 , executa a tarefa de rotear tráfego entre a Ethernet e as redes token ring.

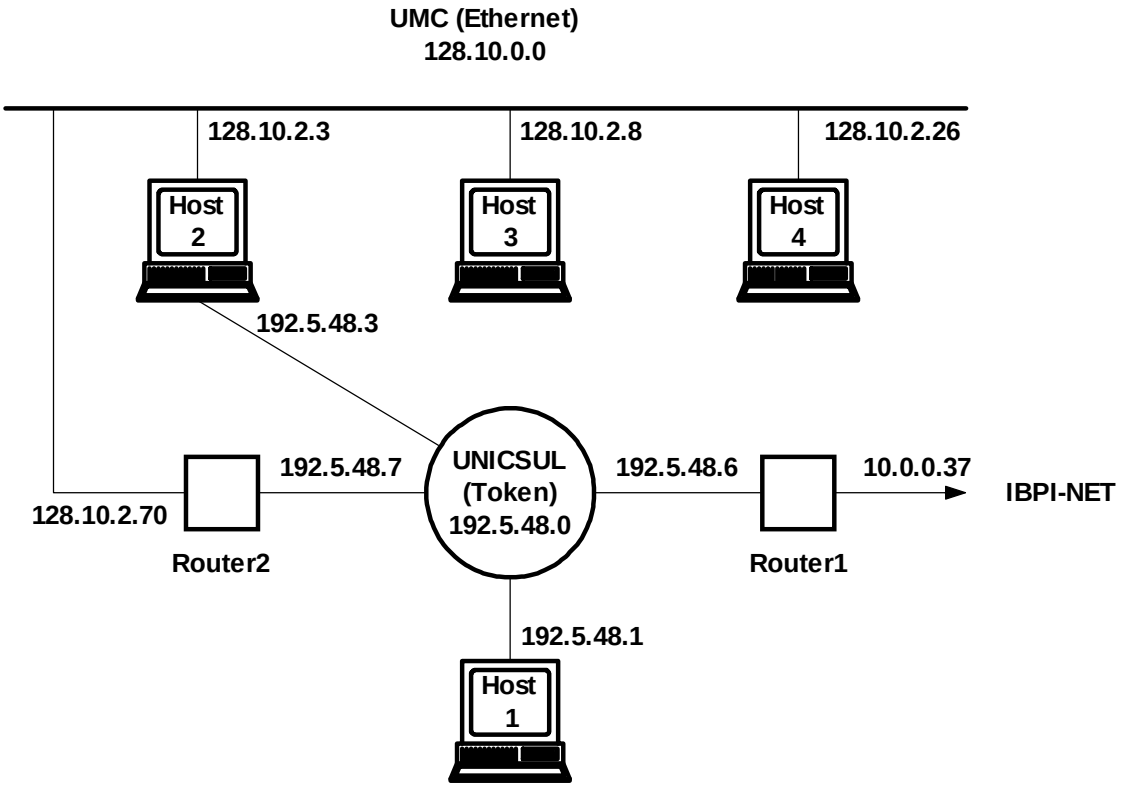

Figura 3.5. Exemplo de designação de endereços IP para roteadores e hosts acoplados a três redes da figura anterior.

4. PROTOCOLOS

Os protocolos são os padrões que especificam como os dados são apresentados ao serem transmitidos de uma máquina para outra. Os protocolos representam para a comunicação computadorizada o que a linguagem de programação é para a computação.

Os complexos sistemas de comunicação de dados não usam um único protocolo para tratar com todas as tarefas de transmissão. Ao contrário, requerem uma pilha de protocolos cooperativos, algumas vezes chamados *família de protocolo* ou *pilha de protocolo*. Para compreender a razão, vejamos os problemas que surgem quando as máquinas se comunicam através de uma rede de dados:

- *Falha de hardware*. Um host ou roteador pode falhar tanto porque o hardware falha como porque o sistema operacional entra em colapso. Um enlace de transmissão de rede pode falhar ou ser desconectado acidentalmente. O software de protocolo necessita detectar tais falhas e recuperar-se delas, se possível.

- *Congestionamento de redes*. Mesmo quando todo o hardware e software operam corretamente, as redes têm capacidade finita que pode ser ultrapassada. Os protocolos precisam encontrar formas para que uma máquina em congestionamento possa suprimir o excesso de tráfego.

- *Demora ou perda de pacotes*. Algumas vezes, os pacotes demoram muito ou são perdidos. Os protocolos precisam aprender sobre as falhas ou adaptar-se a longas demoras.

- *Danificação de dados*. Interferência elétrica ou magnética ou falhas de hardware podem causar a transmissão de erros que danificam os conteúdos dos dados transmitidos. Os protocolos necessitam detectar e recuperar tais erros.

- *Duplicação de dados ou erros seqüenciais*. Redes que oferecem rotas múltiplas podem transmitir dados fora de seqüência ou podem transmitir duplicatas de pacotes. Os protocolos necessitam reorganizar os pacotes e remover algumas duplicatas.

4.1. MODELO DA DIVISÃO EM CAMADAS OSI

Existe um modelo desenvolvido pela ISO (International Standards Organization) usado para descrever a estrutura e funcionamento dos protocolos de comunicação de dados é denominado Modelo de Referência OSI (Open Systems Interconnect). A base deste modelo é a divisão da complexidade do projeto organizando a rede em camadas, com níveis de abstração diferentes definindo uma pilha de protocolos. Ele contém sete camadas, sendo cada uma, responsável por oferecer serviços às camadas superiores de uma forma transparente, ou seja, as demais camadas não precisam saber de detalhes da implementação do serviço implementado nesta camada.

Camada	Funcionalidade
7	Aplicativo
6	Apresentação
5	Sessão
4	Transporte
3	Rede
2	Enlaces de Dados
1	Conexão Física

Figura 4.1. Modelo OSI de referência das sete camadas para o protocolo.

Descrição básica das camadas do modelo OSI:

- Camada física. Especifica um padrão para a interconexão física entre hosts e comutadores de pacote de rede, e também os procedimentos usados para transferir pacotes de uma máquina para outra.

- Camada de enlaces de dados. Especifica como os dados transitam entre um comutador de pacote e um host ao qual está conectado. Define o formato dos quadros e especifica como as duas máquinas reconhecem os limites do quadro. Já que a transmissão de erros pode destruir os dados, o protocolo de nível inclui a detecção de erro, permitindo que as duas saibam quando a transferência de um quadro foi bem-sucedida.

- Camada de rede. Contém a funcionalidade que completa a definição da interação entre o host e a rede. Denominado camada de rede ou sub-rede de comunicação, esse nível define a unidade básica de transferência na rede e inclui os conceitos de endereçamento e roteamento de destino.

- Camada de transporte. Garante que o destino recebe os dados extamente da forma que eles tenham sido mandados. Para isso ela também oferece serviços com conexão e sem conexão, como a camada de redes, mas estes são melhorados. A camada de rede faz parte da sub-rede de comunicação, pertence à concessionária e pode variar de uma rede para outra. Com a camada de transporte é possível rodar vários programas de aplicação sobre redes diferentes, uma vez que se utiliza primitivos padrões da camada de transporte.

- Camada sessão. Permite que os usuários estabeleçam uma sessão, ou seja, um ambiente iniciado a partir de uma conexão e que permite a transferência organizada de dados. Além disso, gerencia os diálogos, ou seja, o controle de quem deve ser a vez de conversar. Existem conexões full-duplex, onde os dados se movem nos dois sentidos simultaneamente, e half-duplex, onde somente um lado “fala” a cada vez.

- Camada de apresentação. Cuida dos problemas relativos a representação dos dados transmitidos, como conversão, criptografia e compressão.

- Camada de aplicativo. Contém os programas com os quais o usuário interfaceia mais diretamente. Alguns desses programas tornaram-se extremamente úteis e acabaram por definir uma série de padronizações. Entre os exemplos possíveis estão: correio eletrônico, transferência de arquivos, acessos a arquivos remotos, entre outros.

4.2. MODELO DA DIVISÃO EM CAMADAS TCP/IP

O software TCP/IP é organizado em quatro camadas conceituais construídas em uma quinta camada de hardware. A Figura 4.2 mostra as camadas conceituais, assim como a forma dos dados à medida que passa entre elas.

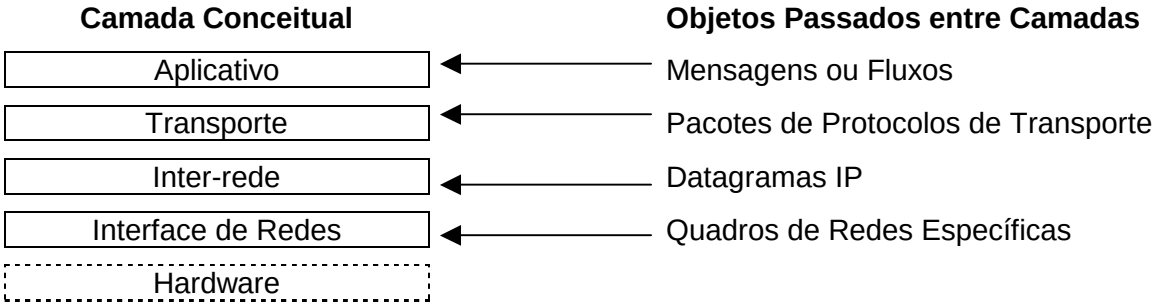

Figura 4.2. As quatro camadas conceituais do software TCP/IP e a forma dos objetos que passam entre elas.

- *Camada de aplicativos*. No nível mais alto, os usuários rodam programas aplicativos que acessam serviços disponíveis através de uma interligação em redes TCP/IP. Um aplicativo interage com um dos protocolos do nível de transporte para enviar ou receber dados. Cada programa aplicativo escolhe o estilo de transporte necessário, que tanto pode ser uma seqüência de mensagens individuais ou um stream contínuo de bytes. O programa aplicativo passa, para o nível de transporte, os dados na forma adequada, para que possam, então, ser transmitidos.

- *Camada de Transporte*. A primeira função da camada de transporte é prover a comunicação de um programa aplicativo para outro. Tal comunicação é sempre chamada fim-a-fim. A camada de transporte pode regular o fluxo de informações. Ela pode fornecer transporte confiável, assegurando que os dados cheguem sem erros e em seqüência. Para isso, o protocolo de transporte faz com que o lado receptor envie confirmações e o lado transmissor retransmita pacotes perdidos. O software da camada de transporte divide o fluxo de dados transmitidos em pequenas partes (algumas vezes chamadas **pacotes**) e

passe cada pacote, juntamente com o endereço de destino, à camada seguinte para ser transmitido.

- *Camada Internet*. Como vimos, a camada da Internet trata das informações de uma máquina para outra. Aceita um pedido para enviar um pacote originário da camada de transporte juntamente com um identificação da máquina para a qual o pacote deve ser enviado. Encapsula o pacote em um datagrama IP, preenche o cabeçalho do datagrama, usa o algoritmo de roteamento para decidir se entrega o datagrama diretamente ou o envia para um roteador e passa o datagrama para a interface de rede apropriada para transmissão. A camada Internet também lida com datagramas de entrada, verificando sua validade, e usa o algoritmo de roteamento para decidir se o datagrama deve ser processado no local ou se deve ser enviado. Para os datagramas endereçados à máquina local, o software da camada de interligação em redes apaga o cabeçalho do datagrama e, entre vários protocolos de transporte, escolhe aquele que vai cuidar do pacote.

- *Camada de interface de rede*. O nível mais baixo do software TCP/IP compreende uma camada da interface de rede responsável pela aceitação de datagramas IP e por sua transmissão através de uma rede específica. Uma interface de rede pode consistir em um driver de dispositivo ou em um subsistema complexo que usa seu próprio protocolo de enlace de dados.

4.3. PROTOCOLOS DE TRANSPORTE

4.3.1. UDP (User Datagram Protocol)

Na pilha de protocolos TCP/IP, o UDP, fornece o mecanismo principal utilizado pelos programas aplicativos para enviar datagramas a outros programas iguais. O UDP fornece um serviço de transmissão sem conexão, não-confiável, usando o IP para transportar mensagens entre

máquinas. Usa o IP para transportar mensagens, porém acrescenta a habilidade de distinguir entre múltiplos destinos em um certo host.

Cada mensagem UDP é conhecida como um datagrama de usuário, que consiste em duas partes: um cabeçalho UDP e uma área de dados UDP. Como a Figura 4.3 mostra, o cabeçalho está dividido em quatro campos de 16 bits que especificam a porta da qual a mensagem foi enviada, a porta à qual a mensagem é destinada, o comprimento da mensagem e a soma de verificação UDP.

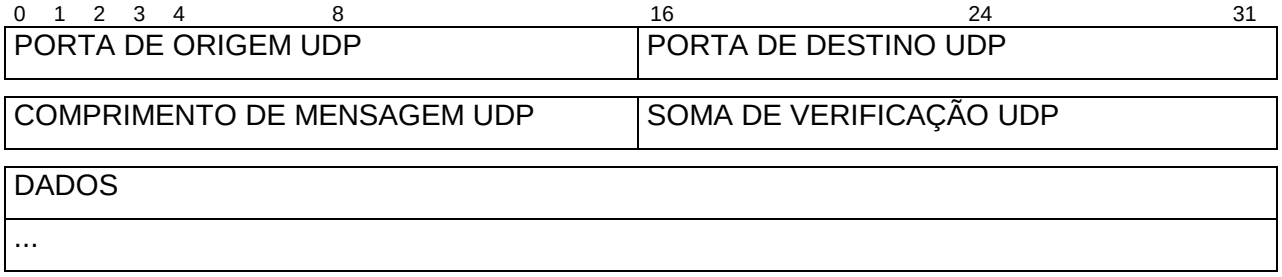

Figura 4.3. O formato dos campos em um datagrama UDP.

Os campos PORTA DE ORIGEM e PORTA DE DESTINO contêm os números de porta do protocolo UDP de 16 bits usados para demultiplexar os datagramas entre os processos que esperam para recebê-los. A PORTA DE ORIGEM é opcional. Quando usada, especifica a porta para a qual devem ser enviadas as respostas; se não usada, deverá ser zero.

O campo COMPRIMENTO contém uma contagem de octetos no datagrama UDP, incluindo o cabeçalho UDP e os dados de usuário. Assim, o valor mínimo para COMPRIMENTO é oito, que é o comprimento apenas do cabeçalho.

A soma de verificação UDP é opcional e não precisa ser usada; um valor de zero no campo SOMA DE VERIFICAÇÃO significa que a soma de verificação não foi calculada. Os projetistas resolveram tornar opcional a soma de verificação a fim de permitir que as implementações operassem com pouco overhead ao usar UDP através de uma rede local altamente confiável. O IP não calcula uma soma de verificação na parte dos dados de um datagrama IP, portanto, a

soma de verificação oferece o único modo de assegurar que os dados tenham chegado intactos e possam ser usados.

A camada IP é responsável apenas pela transferência de dados entre um par de hosts em uma interligação em redes, enquanto a camada UDP é responsável apenas pela diferenciação entre múltiplas origens ou destinos em um host. Desse modo, apenas o cabeçalho IP identifica os hosts de origem e destino; apenas a camada UDP identifica as portas de origem e destino em um host.

O software colocado em cada camada de uma hierarquia de protocolo deve ser capaz de multiplexar ou demultiplexar múltiplos objetos da camada seguinte. O software UDP apresenta outro exemplo de multiplexação e demultiplexação. Aceita datagramas UDP de muitos programas aplicativos e os passa ao IP, para transmissão, e aceita datagramas UDP recebidos de IP e os passa ao programa aplicativo apropriado. Conceitualmente, toda a multiplexação e demultiplexação entre o software UDP e os programas aplicativos ocorre através do mecanismo da porta. Na prática, cada programa aplicativo deve negociar com o sistema operacional a fim de obter uma porta de protocolo e um número de porta correspondente, antes que ele possa enviar um datagrama UDP.

O modo mais simples de conceber uma porta UDP é uma fila. Na maioria das implementações, quando um programa aplicativo negocia com um sistema operacional para usar determinada porta, o sistema operacional cria uma fila interna que pode reter as mensagens que estão chegando. Freqüentemente, o aplicativo pode especificar ou mudar o tamanho da fila. Quando o UDP recebe um datagrama, verifica se o número de porta de destino confere com uma das portas atualmente em uso. Se não conferir, envia uma mensagem de erro de porta não-atingida e descarta o datagrama. Se for encontrada uma correspondência, o UDP enfileira o novo datagrama na porta onde o programa aplicativo pode acessá-lo. Naturalmente, ocorrerá um erro se a porta estiver cheia e o UDP descartar o datagrama recebido.

O UDP é um protocolo mais rápido do que o TCP, pelo fato de não verificar o reconhecimento das mensagens enviadas. Por este mesmo motivo, não é confiável como o TCP.

4.3.2. TCP (Transmission Control Protocol)

Em um nível mais baixo, as redes de comunicação fornecem uma entrega de pacotes não-confiável, os pacotes podem ser perdidos ou danificados quando erros de transmissão interferem nos dados. As redes que roteiam pacotes dinamicamente podem entregá-los fora de ordem, entregá-los após um intervalo substancial, ou entregar reproduções.

Em um nível mais alto, os programas aplicativos freqüentemente precisam enviar grandes volumes de dados de um computador a outro. A utilização de um sistema de transmissão sem conexão e não-confiável torna-se tediosa e irritante, e requer que os programadores criem detecção e recuperação de erros em cada programa aplicativo. Um dos objetivos da pesquisa de protocolos de rede foi encontrar soluções de fins gerais para problemas de transmissão confiável de streams de pacotes, possibilitando aos especialistas criar um único protocolo para transmissão de stream de dados que todos os programas aplicativos pudessem utilizar.

O serviço é definido pelo TCP, o serviço de stream confiável é tão importante que toda a pilha de protocolos é citada como TCP/IP. O TCP é um protocolo de comunicação, e não um software, o protocolo especifica o formato dos dados e das confirmações que os dois computadores trocam para oferecer uma transferência confiável e, também, os procedimentos de que se valem os computadores para assegurar que os dados cheguem corretamente.

O TCP pode ser utilizado com uma variedade de sistema de transmissão de pacotes, inclusive o serviço de transmissão de datagramas IP. O TCP pode, por exemplo, ser implementado para utilizar linhas telefônicas por discagem, rede local, rede de fibra óptica de alta velocidade, ou uma rede de longas distâncias de velocidade mínima. Na realidade, um dos pontos fortes do TCP é a grande variedade de sistemas de transmissão que ele pode usar.

O TCP posiciona-se acima do IP no esquema de divisão em camadas do protocolo, permite que programas aplicativos múltiplos, de determinada máquina, comuniquem-se simultaneamente, e

ele demultiplexa o tráfego TCP de entrada entre os programas aplicativos. Como o UDP, o TCP utiliza números de porta de protocolo para identificar o destino final em uma máquina. Porém, as portas TCP são muito mais complexas, porque determinado número delas não corresponde a um objeto único. Ao contrário, o TCP foi estruturado na abstração de conexão, em que os objetos a serem identificados são conexões de circuito virtual, e não portas isoladas.

A unidade de transferência entre o software TCP de duas máquinas é denominada segmento. Os segmentos são trocados para estabelecer conexões, transferir dados, enviar confirmações, informar tamanhos de janelas e encerrar conexões. A Figura 4.4 mostra o formato do segmento TCP.

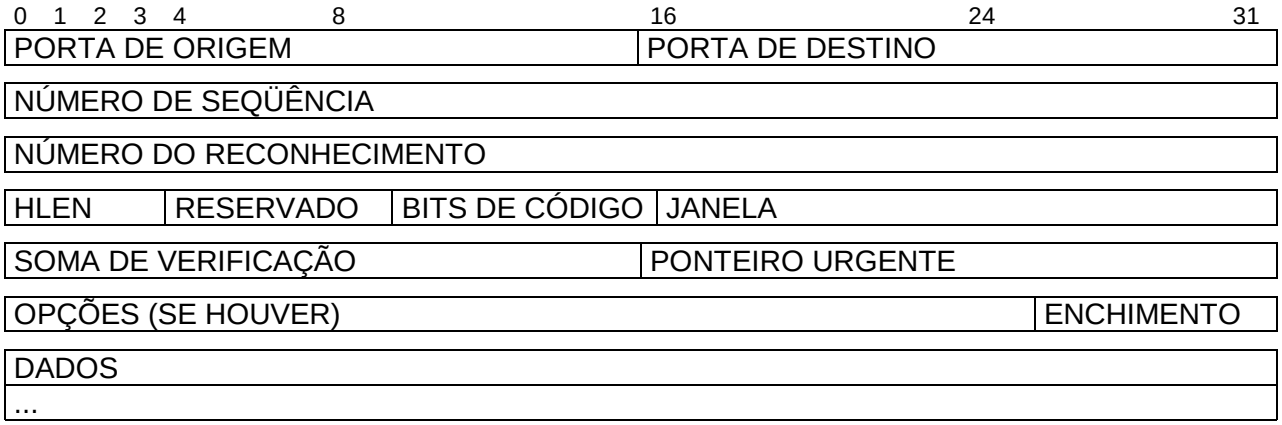

Figura 4.4. O formato de um segmento TCP com o cabeçalho TCP seguido de dados.

Cada segmento é dividido em duas partes: um cabeçalho seguido de dados. O cabeçalho, conhecido como cabeçalho TCP, transporta a identificação esperada e as informações de controle. Os campos PORTA DE ORIGEM e PORTA DE DESTINO contêm os números de portas TCP que identificam os programas aplicativos nas extremidades da conexão. O campo NÚMERO DE SEQÜÊNCIA identifica a posição no stream de bytes do transmissor dos dados no segmento. O campo NÚMERO DO RECONHECIMENTO identifica o número do octeto que a origem espera receber depois. Observe que o número seqüencial refere-se ao stream que segue na mesma direção que o segmento, enquanto o número do reconhecimento refere-se ao stream que segue em direção oposta ao segmento.

O campo HLEN[Ψ] contém um número inteiro que especifica o comprimento do cabeçalho do segmento, medido em múltiplos de 32 bits. Isso é necessário porque o campo OPÇÕES varia em comprimento, dependendo de quais opções foram incluídas. Assim, o tamanho do cabeçalho TCP varia de acordo com as opções selecionadas.

Uma das idéias mais importantes e complexas em TCP concentra-se na maneira como ele trata o timeout e a retransmissão. A exemplo de outro protocolos confiáveis, o TCP espera que o destino envie confirmação toda vez que recebe novos octetos, com êxito, do stream de dados. Sempre que envia um segmento, o TCP inicia um temporizador e espera uma confirmação. Se o temporizador terminar antes que os dados do segmento tenham sido confirmados, o TCP aceita que esse segmento foi perdido ou destruído e o retransmite. O software TCP precisa aceitar as duas grandes diferenças entre o tempo necessário para atingir vários destinos e as alterações necessárias no tempo para atingir um determinado destino, de acordo com a variação de carga do tráfego. O TCP monitora o desempenho de cada conexão e deduz valores razoáveis para timeout. À medida que o desempenho de uma conexão muda, o TCP revisa seu valor de timeout (ou seja, adapta-se à mudança).

O congestionamento é uma condição de retardo longo causado por uma sobrecarga de datagramas em um ou mais pontos de comutação (p. ex., em roteadores). Quando ocorre um congestionamento, os intervalos aumentam e o roteador começa a enfileirar datagramas até que possa distribuí-los. Precisamos lembrar que cada roteador possui uma capacidade limitada de armazenamento e que os datagramas concorrem para esse armazenamento. Na pior hipótese, o número total de datagramas que chega ao roteador congestionado cresce até que o roteador alcance a capacidade e comece a eliminar datagramas. Lamentavelmente, a maioria dos protocolos de transporte utiliza timeout e retransmissão, de modo que reajam ao aumento do retardo com a retransmissão de datagramas. As retransmissões agravam o congestionamento em vez de o amenizarem. Se não for verificado, o aumento do tráfego produzirá um aumento do retardo, provocando uma elevação do tráfego, e assim por diante, até que a rede torne-se inútil; mas os protocolos de transporte, como o TCP, podem evitar o congestionamento reduzindo automaticamente as taxas de transmissão sempre que ocorrerem

[Ψ] De acordo com a especificação, o campo HLEN é o deslocamento dos dados da área situada dentro do segmento.

retardo. Logicamente, os algoritmos para evitar congestionamentos precisam ser criados cuidadosamente porque, mesmo em condições operacionais normais, uma interligação em redes exibirá grande variação de retardo de ida e volta.

4.4. PROTOCOLOS DE REDE

4.4.1. IP (Internet Protocol)

O serviço mais importante de uma rede consiste em um sistema de entrega de pacotes. Tecnicamente, o serviço é definido como um sistema de transmissão sem conexão, e não-confiável; é análogo ao serviço oferecido por hardwares de redes. O serviço é conhecido como não-confiável porque a entrega não é garantida. O pacote pode ser perdido, reproduzido, atrasar-se ou ser entregue com problemas, mas o serviço não detectará tais condições, nem informará isso ao transmissor nem ao receptor. Ele é denominado sem conexão porque cada pacote é independente dos outros, uma seqüência de pacotes enviados de um computador a outro pode trafegar por caminhos diferentes, ou alguns podem ser perdidos enquanto outros são entregues.

O protocolo que define o mecanismo de transmissão sem conexão e não-confiável é conhecido como Internet Protocol. O IP oferece três definições importantes:

- O protocolo IP define a unidade básica de transferência de dados utilizada através de uma interligação em redes TCP/IP, assim, ela especifica o formato exato de todos os dados à medida que ela passa pela interligação em redes TCP/IP.

- O software IP desempenha a função de roteamento, escolhendo um caminho por onde os dados serão enviados.

- O IP inclui um conjunto de regras que concentram a idéia da entrega não-confiável de pacotes, que definem como os hosts e os roteadores devem processar os pacotes, como e quando as mensagens de erro devem ser geradas e as condições segundo as quais os pacotes podem ser descarregados.

Numa rede física, a unidade de transferência é um quadro que contém um cabeçalho e dados, onde o cabeçalho fornece informações como endereço de origem e de destino (físicos). A interligação em redes denomina sua unidade básica de transferência de um datagrama IP, que é dividido em cabeçalho e áreas de dados. A diferença é que o cabeçalho do datagrama contém endereços IP, enquanto o quadro contém os endereços físicos.

A Figura 4.5 mostra a organização dos campos em um datagrama:

Figura 4.5. Formato de um datagrama da Internet, a unidade básica de transferência em um interligação em redes TCP/IP.

Já que o processamento de datagramas se dá em softwares, o conteúdo e o formato não são retringidos por quaisquer hardwares. O primeiro campo de quatro bits de um datagrama (VERS), por exemplo, contém a versão do protocolo IP utilizada para criar o datagrama. Ele é utilizado para verificar se o transmissor, o receptor e quaisquer roteadores existentes entre eles concordam quanto ao formato do datagrama. Todo software IP precisa verificar o campo de versão antes de processar um datagrama, para assegurar-se de que ele se adapta ao formato

que o software espera. Se os padrões mudarem, as máquinas rejeitarão datagramas com versões de protocolo diferentes dos seus, impedindo que eles deturpem o conteúdo do datagrama com um formato desatualizado. A versão atual do protocolo IP é a quatro.

O campo de comprimento do cabeçalho (HLEN), também de quatro bits, fornece o comprimento do cabeçalho do datagrama medido em palavras de 32 bits. Todos os campos do cabeçalho contêm um comprimento fixo, exceto para OPÇÕES IP e os campos correspondentes PADDING. O cabeçalho mais comum, que não contém qualquer opção e nenhum preenchimento, mede 20 octetos e o campo de comprimento de cabeçalho é igual a cinco.

O campo COMPRIMENTO TOTAL, fornece o comprimento do datagrama IP medido em octetos, incluindo octetos no cabeçalho e nos dados. O tamanho da área de dados pode ser calculado subtraindo-se de COMPRIMENTO TOTAL o comprimento do cabeçalho (HLEN). Já que o campo COMPRIMENTO TOTAL, possui 16 bits de coprimento, o maior tamanho possível para um datagrama IP é 2^{16} ou 65.535 octetos. Na maioria do aplicativos, essa não é uma limitação rígida. No futuro pode tornar-se mais importante, se as redes de velocidade mais alta puderem transportar pacotes de dados maiores que 65.535 octetos.

Três campos no cabeçalho do datagrama, IDENTIFICAÇÃO, FLAGS e DESLOCAMENTO DO FRAGMENTO, controlam a fragmentação e a remontagem de datagramas. O campo IDENTIFICAÇÃO contém um número inteiro único que identifica o datagrama, sua finalidade principal é permitir que o destino saiba quais datagramas estão chegando e a que datagramas pertencem. O campo DESLOCAMENTO DO FRAGMENTO é responsável pela remontagem do datagrama, o destino precisa obter todos os fragmentos que iniciam com o fragmento que possui deslocamento zero até o fragmento de maior deslocamento. O campo FLAGS controlam a fragmentação, define se o datagrama pode ou não ser fragmentado e se o destino reuniu todos os fragmentos.

O campo TEMPO DE VIDA especifica quanto tempo, em segundos, o datagrama pode permanecer no sistema de interligação em redes, os roteadores e os hosts que processam datagramas precisam decrementar o campo TEMPO DE VIDA à medida que o tempo passa e remover o datagrama da interligação quando seu tempo expira. PROTOCOLO especifica qual

protocolo de alto nível foi utilizado para criar a mensagem que está sendo transportada na área DADOS do datagrama.

O campo VERIFICAÇÃO DA SOMA DO CABEÇALHO assegura a integridade dos valores de cabeçalho. A verificação IP é formada com o tratamento do cabeçalho como uma seqüência de números inteiros de 16 bits (na ordem de bytes da rede), reunindo-os com uma aritmética complemento de um, e a seguir considerando o complemento de um como o resultado. Para a finalidade de calcular a soma de verificação, considera-se que o campo VERIFICAÇÃO DA SOMA DO CABEÇALHO contenha zero.

Os campos ENDEREÇO IP DE ORIGEM e ENDEREÇO IP DE DESTINO contêm endereços IP de 32 bits do transmissor do datagrama e do receptor desejado. Embora o datagrama possa ser roteado através de muiots roteadores intermediários, os campos da origem e destino nunca mudam; eles especificam os endereços IP da origem e do último destino.

O campo denominado DADOS mostra o início da área de dados do datagrama, seu comprimento depende, logicamente, do que está sendo enviado no datagrama. O campo OPÇÕES IP que se segue ao endereço de destino não é necessário em todo datagrama, e as opções são incluídas principalmente para testes ou depuração da rede. Contudo, o processamento de opções é parte integrante do protocolo IP; assim, todas as implementações de padrões precisam incluí-lo. O campo PADDING depende das opções selecionadas. Ele representa bits contendo zero e que podem ser necessários para garantir que o cabeçalho do datagrama se estenda até o múltiplo exato de 32 bits.

O objetivo do IP é fornecer uma rede virtual que abranja várias redes físicas e ofereça um serviço de entrega de datagrama sem conexão. O algoritmo de roteamento IP deve escolher a forma pela qual enviará um datagrama através de várias redes físicas. O roteamento se classifica em dois tipos: **encaminhamento direto** e **encaminhamento indireto**. O encaminhamento direto é a transmissão de um datagrama, através de uma única rede física para outra máquina. Duas máquinas só podem executar o encaminhamento direta se ambas se conectarem diretamente a uma mesma rede física (p. ex., uma única Ethernet). O

encaminhamento indireto ocorre quando o destino não se encontra na mesma rede física, forçando o transmissor a passar o datagrama para um roteador executar a entrega.

O algoritmo de roteamento IP pode ser descrito da seguinte forma:

Extrai o endereço IP de destino, D, do datagrama e calcule o prefixo da rede, N;
Se N corresponder com qualquer endereço de rede conectado
diretamente entregue datagrama ao destino D através desta rede
(Isto envolve converter D para um endereço físico, encapsular o datagrama e enviar o quadro.)
Se a tabela contiver uma rota específica do host para D
envie o datagrama para o próximo passo da rota especificada na tabela
Se a tabela contiver uma rota para a rede N
envie o datagrama para o próximo passo da rota especificada na tabela
Se a tabela contiver uma rota padrão
envie o datagrama para o roteador padrão especificado na tabela
Caso contrário declare um erro de roteamento;

O endereço IP selecionado pelo algoritmo de roteamento IP é conhecido como endereço do próximo passo da rota, porque indica para onde o datagrama deve ser enviado (mesmo que não seja o último destino). O software da interface de rede vincula o endereço do próximo passo da rota a um endereço físico, forma um quadro usando aquele endereço físico, coloca o datagrama na parte de dados do quadro e envia o resultado. Após usar o endereço do próximo passo da rota para encontrar um endereço físico, o software de interface da rede descarta o endereço do próximo passo da rota.

Quando um datagrama IP chega a um host, o software de interface da rede entrega-o ao software IP para processamento. Se o endereço de destino do datagrama corresponder ao endereço de IP do host, o software IP do host aceita o datagrama e passa-o ao software apropriado, do protocolo de nível mais alto, para posterior processamento. Se não houver correspondência com o IP de destino, o host deve descartar o datagrama (ou seja, os hosts ficam proibidos de tentar encaminhar datagramas acidentalmente roteados para a máquina errada).

O algoritmo de roteamento da interligação em redes é orientado por tabela e usa apenas endereços IP. Embora seja possível para uma tabela de roteamento conter um endereço de

destino específico ao host, a maior parte dessas tabelas contém apenas endereços de rede, mantendo pequenas as tabelas de roteamento. O uso de uma rota padrão também pode manter pequena a tabela de roteamento, principalmente para hosts que só conseguem acessar um roteador.

4.4.2. ICMP (Internet Control Message Protocol)

Para permitir que os roteadores de uma interligação em redes informem os erros ou forneçam informações sobre ocorrências inesperadas, os projetistas acrescentaram aos protocolos TCP/IP um mecanismo de mensagem para fins específicos, conhecido como ICMP – Internet Control Message Protocol (Protocolo Internet de Mensagem de Controle), é considerado uma parte necessária do IP e deve ser incluído em cada implementação de IP.

O ICMP permite que os roteadores enviem mensagens de erro ou de controle aos outros roteadores ou aos hosts; possibilita a comunicação entre o software do IP em uma máquina, e o software de IP em outra. Um host pode usar o ICMP para se corresponder com um roteador ou com outro host. A principal vantagem de permitir que os hosts usem ICMP é que isto fornece um mecanismo único usado para todas as mensagens de controle de informação. Do ponto de vista técnico, o ICMP é um mecanismo para relatar erros. Fornece um meio pelo qual os roteadores que encontram erros possam levá-los ao conhecimento do transmissor.

As mensagens ICMP são geradas por gateways na rota de transporte de um datagrama ou pela estação de destino. Quando ocorre algum problema previsto pelo ICMP. A mensagem ICMP descrevendo a situação é preparada e entregue à camada IP (Figura 4.6), que adiciona à mensagem ICMP o cabeçalho IP e envia ao emissor do datagrama com o qual ocorreu o problema.

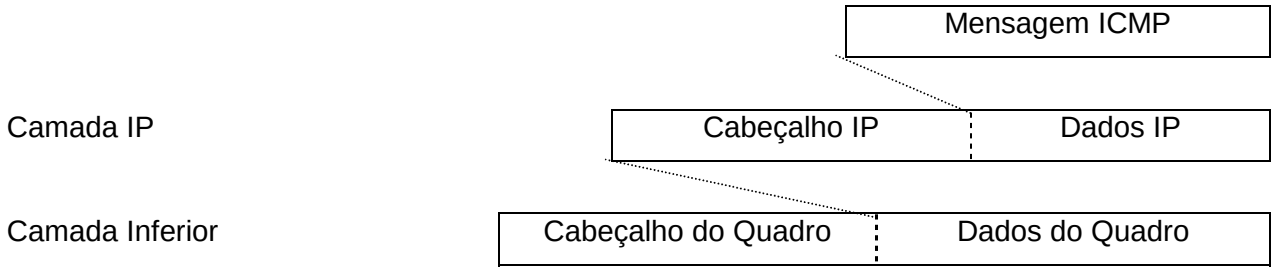

Figura 4.6. Encapsulamento da mensagem ICMP em um Datagrama IP.

4.4.3. ARP (Address Resolution Protocol)

Os endereços IP são atribuídos independente de um endereço de hardware físico de uma máquina. Para enviar um pacote de interligação em redes através de uma rede física de um computador para outro, o software de rede deve mapear o endereço IP em um endereço físico de hardware e usar o endereço de hardware para transmitir o quadro. Se os endereços de hardware forem menores do que os endereços IP, um mapeamento direto pode ser estabelecido tendo o endereço físico da máquina codificado em seu endereço IP. De outra maneira, os mapeamentos devem ser executados dinamicamente. O ARP executa a conversão de endereço dinâmica, usando somente o sistema de comunicação de rede de baixo nível. ARP permite que as máquinas convertam endereços sem manter um registro permanente das vinculações.

Uma máquina usa ARP para descobrir o endereço de hardware de outra máquina, difundindo uma solicitação ARP. A solicitação contém o endereço IP da máquina para o qual um endereço de hardware é requisitado. Todas as máquinas de uma rede recebem uma solicitação ARP. Se a solicitação combina com um endereço IP de máquina, a máquina responde enviando uma resposta que contém o endereço de hardware necessitado. As respostas são dirigidas para uma máquina; não são transmitidas por difusão.

Para tornar ARP eficiente, cada máquina trata as vinculações entre endereços físicos IP. Como o tráfego de interligação em redes geralmente consiste em uma seqüência de interações entre pares de máquinas, o cache elimina muitas solicitações de difusão ARP.

4.4.4. RARP (Reverse Address Resolution Protocol)

Este protocolo destina-se à solução do problema inverso ao resolvido pelo ARP. O problema RARP é um host que não conhece o seu próprio endereço IP ou o de um outro host, mas possui o endereço físico correspondente. O protocolo RARP permite que a partir do endereço físico, seja obtido o endereço IP correspondente. Para que o RARP funcione, é necessário ao menos um servidor RARP, que possui informações de mapeamento de todos os hosts da rede. Pode haver um ou vários servidores RARP na mesma rede.

A principal vantagem de possuir diversas máquinas funcionando como servidores RARP é tornar o sistema mais confiável. Se um servidor estiver desativado, ou excessivamente sobrecarregado para responder, um outro responderá à solicitação. Assim, é muito mais provável que o serviço esteja disponível. A principal desvantagem de se utilizarem muitos servidores é que quando uma máquina difunde uma solicitação RARP, a rede torna-se sobrecarregada se todos os servidores tentam responder. Em uma Ethernet, por exemplo, utilizar servidores RARP múltiplos torna mais provável uma colisão.

Para evitar respostas múltiplas e simultâneas, há pelos menos duas possibilidades, e ambas envolvem o aumento do intervalo entre as respostas:

- A cada máquina que faz solicitações RARP é atribuído um servidor principal. Em circunstâncias normais, somente o servidor principal da máquina responde à sua solicitação RARP. Todos os servidores que não sejam principais recebem a solicitação, mas apenas registram seu tempo de chegada. Se o servidor principal não estiver disponível, a máquina original fará um intervalo para aguardar uma resposta e, a seguir, difundir novamente a

solicitação. Sempre que um servidor, que não seja o principal, receber uma segunda cópia de uma solicitação RARP em um curto espaço após a primeira, ele responde.

- A segunda solução utiliza um esquema semelhante, mas tenta impedir que todos os servidores não-principais transmitam respostas simultaneamente. Cada máquina não-principal que receber uma solicitação calcula um intervalo aleatório e, a seguir, envia uma resposta. Em circunstâncias normais, o servidor principal responde imediatamente e com intervalos entre as respostas sucessivas, de modo que há pouca probabilidade de que cheguem ao mesmo tempo. Quando o servidor principal não está disponível, a máquina solicitante sofre um pequeno retardo antes que uma resposta seja recebida. Escolhendo os intervalos com cuidado, o projetista assegura que as máquinas solicitantes não difundiram antes de receber uma resposta.

5. ROTEAMENTO

5.1. ROTEAMENTO BASEADO EM TABELAS

Este algoritmo do protocolo IP utiliza uma tabela de roteamento que armazena informações sobre como atingir cada sub-rede de rede *internet*. Sempre que a camada IP, em uma estação ou em um *gateway,* precisa transmitir um datagrama para uma estação que não está diretamente conectada à mesma sub-rede, ela consulta a tabela de roteamento a fim de determinar o *gateway* para o qual esse datagrama deve ser enviado.

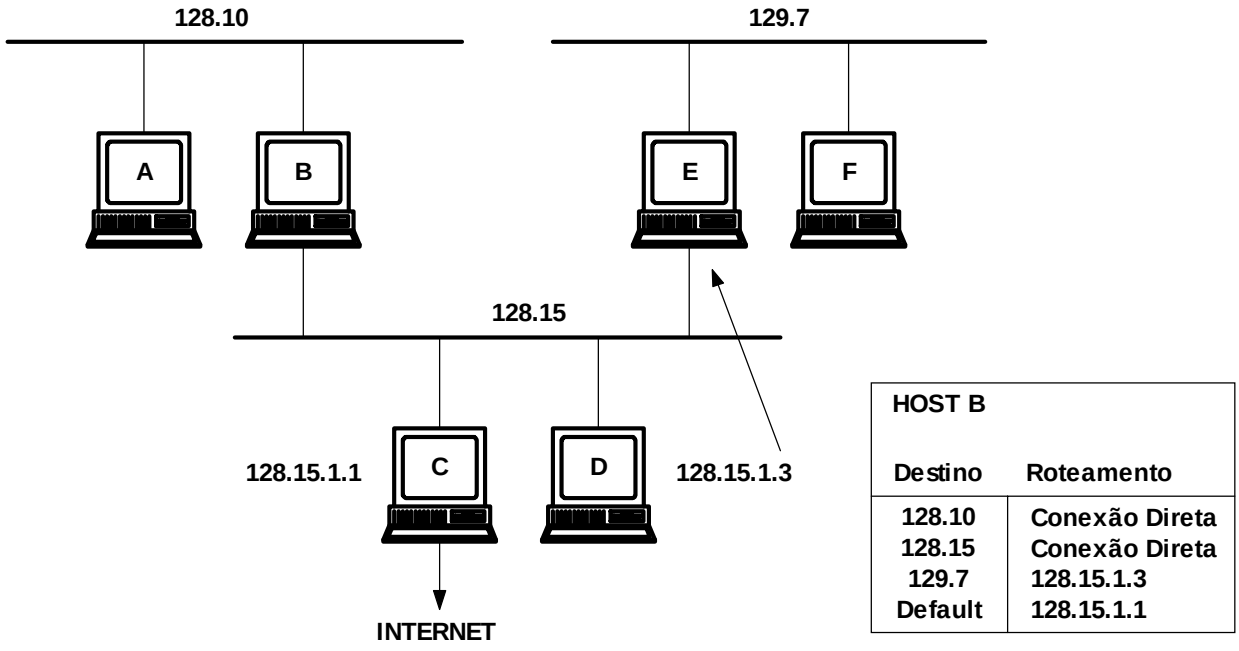

Figura 5.1. Tabela de Roteamento do Gateway B.

Tipicamente, a tabela de roteamento do IP contém entradas do tipo (N,G), onde N é um endereço IP (endereço de destino) e G é endereço IP do próximo *gateway* para atingir N. Essa tabela, portanto, só determina o próximo passo no caminho para um destinatário. Nem a estação emissora nem o gateway conhecem a rota completa até a estação a estação destinatária. Vale ressaltar que as entradas dessa tabela só referenciam *gateways* que podem ser atingidos diretamente, isto é, todos os gateways listados na tabela de roteamento de uma máquina M estão conectados às sub-redes físicas nas quais a máquina M está conectada. Um exemplo de tabela de roteamento é dado na Figura 5.1.

A tabela de roteamento do IP pode conter informações sobre todos os destinatários de uma rede *internet,* já que a maioria das máquinas não teria espaço em memória suficiente para isso Por esse motivo, são armazenados os endereços das sub-redes. Outra técnica utilizada para minimizar o tamanho das tabelas é a utilização de rotas predefinidas *(default*) para o qual um datagrama deve ser enviado sempre que não for encontrada na tabela uma entrada específica para o endereço IP destino. A utilização de rotas predefinidas é particularmente útil em redes com um único gateway, já que todas as demais sub-redes da rede *Internet* devem ser atingidas mediante esse *gateway.*

5.2. ALGORITMOS DE ROTEAMENTO

O algoritmo de roteamento é a técnica utilizada pelos *gateways* para se localizarem mutuamente e para conseguirem comunicação com as diversas redes de uma rede *Internet*. A tabela de roteamento de um *gateway* é atualizada a partir de informações obtidas na execução do algoritmo de roteamento utilizados na arquitetura TCP/IP: Vetor-Distância *(Vector-Distance)* e Estado-do-Enlace (*Link-State*), que são discutidos a seguir.

5.2.1. ROTEAMENTO VECTOR-DISTANCE

Inicialmente, cada *gateway* possui uma tabela contendo uma entrada para cada sub-rede à qual está conectado. A cada sub-rede especificada na tabela está associada a distância entre a mesma e o *gateway* que mantém a tabela. Esta distância pode ser medida em *hops* (número de *gateways* a atravessar para atingir uma sub-rede) ou em retardo (tempo necessário para a sub-rede).Inicialmente, os campos de distância devem valer zero, pois somente as sub-redes às quais o *gateway* está diretamente conectado são especificadas na tabela. Periodicamente, cada *gateway* envia uma cópia de sua tabela para todo o *gateway* que possa atingir diretamente. O *gateway* que recebe a tabela, a compara com a sua própria e modifica esta última nos seguintes casos:

- se o *gateway* emissor conhecer um caminho maios curto para determinada sub-rede, ou seja se a distância apresentada na tabela do emissor for menor do que a da tabela do receptor;

- se o *gateway* emissor apresentar uma sub-rede que o receptor não conhece, ou seja, se na tabela do emissor existir uma entrada que não está presente na tabela do receptor; esta entrada é inserida na tabela do receptor;

- se uma rota que passa pelo emissor tiver sido modificada, ou seja, se a distância associada a uma sub-rede que passa pelo emissor tiver mudado.

Na atualização dos campos de distância da tabela do receptor, deve-se considerar a distância entre os *gateways* emissor e receptor, (por exemplo, é necessário somar 1 no caso da métrica baseada em *hops*). Vale lembrar que, para cada sub-rede especificada na tabela, existe associada um campo que indica o próximo *gateway* na rota para essa sub-rede. A Figura 5.2 ilustra este tipo de roteamento.

O algoritmo é simples e de fácil implementação; porém, em ambientes dinâmicos, onde novas conexões surgem e outras são desativadas com freqüência, a informação de atualização propaga-se muito lentamente e, durante esse período de propagação, alguns *gateways* possuem informações de roteamento inconsistentes. Além disso, as mensagens de atualização tornam-se enormes, pois são diretamente proporcionais ao número total de redes e *gateways* presentes na rede *Internet* (todos os gateways devem participar, senão o algoritmo não converge).

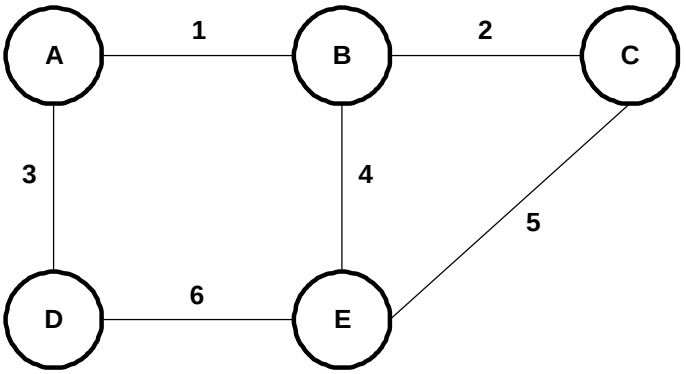

Tabela do Nó A

Destino	Ligação	Distância
A	local	0
B	1	1
C	1	2
D	3	1
E	1	2

Figura 5.2. Roteamento Vector-Distância.

5.2.2. ROTEAMENTO LINK-STATE (Shortest Path First)

Neste algoritmo, cada *gateways* deve conhecer a topologia completa da rede *Internet*. Isto é feito descrevendo-se os *gateways* interconectados entre si por enlaces *(links).* Existe um enlace entre dois *gateways* se ambos puderem comunicar-se diretamente, ou seja se estiverem à mesma rede física.

Cada *gateway* exerce duas funções principais. A primeira é testar continuamente o estado dos enlaces com os *gateways* vizinhos. A segunda é enviar periodicamente os dados de estado de seus enlaces a todos os outros *gateways* da rede *Internet.* O teste de estado é realizado através do envio de mensagens curtas que exigem resposta. Se acontecer uma resposta, sob

condições que variam segundo a implementação do protocolo, o enlace está ativo, senão está inativo. Os dados de estado indicam, simplesmente, se há possibilidade de comunicação entre dois *gateways.* .Estes dados são em geral enviados em modo difusão *(broadcast)* individualmente.

Ao receber uma informação de estado, um *gateway* atualiza seu mapa da rede *Internet* ativado ou desativado os enlaces em questão e recalcula as rotas para todos os destinos possíveis através do algoritmo *Shortest-Path-First* (SPF), de Dijskstra, aplicado à topologia da rede *Internet*. A figura 5.3 ilustra um exemplo deste tipo de roteamento.

Em relação ao algoritmo *Vector-Distance, o* SPF possui diversas vantagens. O cálculo das rotas é realizado localmente, não dependendo de máquinas intermediárias. O tamanho das mensagens não depende do número de *gateways* diretamente conectados ao *gateway* emissor. Como as mensagens trafegam inalteradas a detecção de problemas torna-se mais fácil.

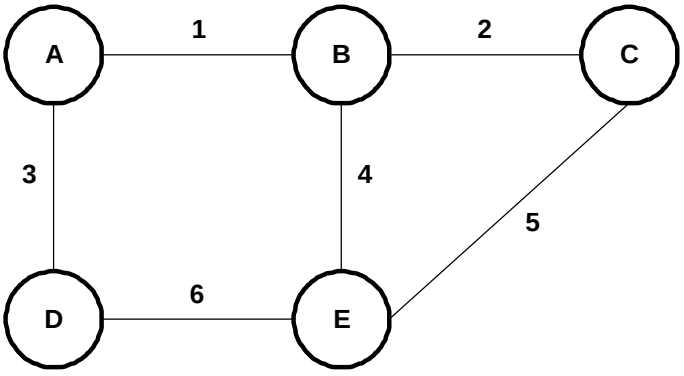

Tabela do Nó A

Fonte	Ligação	Distância
A	B	1
B	C	2
A	D	3
C	E	5
E	D	6
B	E	4

Figura 5.3. Roteamento Link-State.

Os algoritmos descritos anteriormente assumem, a priori, a existência de uma tabela de roteamento devidamente iniciada. Esta iniciação depende do próprio sistema computacional no qual se situa a camada IP. Várias soluções manuais existem como, por exemplo, a carga de uma tabela pré-configurada com dados limitados ou a carga de uma tabela vazia que são preenchidas através de comandos. Entretanto, o problema principal é a manutenção dessas tabelas devido à dinâmica das redes. Para resolver esse problema torna-se imprescindível o

uso de mecanismos automáticos, previstos nos protocolos de roteamento descritos nos itens seguintes.

5.3. PROTOCOLOS DE ROTEAMENTO

O protocolo de roteamento determina a forma pela qual os *gateways* devem trocar informações necessárias à execução do algoritmo de roteamento. Por exemplo, se o algoritmo de roteamento implementado for do tipo Vector-Distance, o protocolo de roteamento deve definir como cada *gateway* envia aos demais a sua distância em relação a cada sub-rede *Internet.*

5.3.1. IGP – Interior Gateway Protocol

O termo IGP é utilizado para designar o protocolo usado na troca de informações de roteamento entre *Interior Gateways* (IG).

Em sistemas autônomos (SA) pequenos, as tabelas de roteamento podem ser determinadas, manualmente, pelo administrador do sistema. O administrador mantém uma tabela de redes do SA que é atualizada sempre que uma rede é removida ou inserida no SA . Além de não ser confiável, esse método torna-se inviável com o crescimento do SA .

Não existe um protocolo padrão entre os *gateways* de um mesmo SA . Por esse motivo, o termo IGP é usado para referenciar qualquer protocolo de roteamento entre Interior Gateways (IG). Os protocolos IGP mais conhecidos são: RIP (*Routing Information Protocol*), *Hello Protocol* e OSPF (*Open Shortest-Path-Frist Protocol).*

RIP - *Routing Information Protocol*

O RIP foi originalmente desenvolvido pela Universidade da Califórnia em Berkeley. Este protocolo permite a troca de informações utilizadas pelo algoritmo de roteamento *Vector-Distance* em uma sub-rede dotada do serviço de difusão de mensagens.

O RIP divide as máquinas da sub-rede em ativas e passivas. As máquinas ativas divulgam informações de roteamento para as outras, enquanto as máquinas passivas recebem as informações e atualizam suas rotas, sem divulga-las. Tipicamente, os *gateways* executam o RIP no modo ativo, enquanto as estações o executam no modo passivo.

Um *gateway* executando o RIP no modo ativo difunde as mensagens a cada 30 segundos e também quando recebe uma solicitação de informação de outro *gateway.* A mensagem difundida normalmente contém informações sobre todas as sub-redes do SA, extraídas da tabela de roteamento do *gateway*. Cada mensagem enviada por um *gateway* G consiste em pares de informações. Cada par é composto de um endereço de sub-rede IP e da distância do *gateway* G à sub-rede . A métrica utilizada para o cálculo de distância é baseada no número de *hops* (número de *gateways)* na melhor rota entre o gateway G e a rede. Curiosamente o RIP assume o valor "1" para a distância de um *gateway* a uma sub-rede à qual ele está diretamente conectado. Para compensar diferenças de tecnologia de redes, algumas implementações do RIP informam uma distância maior quando a rota atravessa uma rede lenta. Participantes ativos e passivos do RIP, quando recebem uma mensagem, atualizam suas rotas de acordo com o algoritmo de roteamento *Vector-Distance,* descrito no item 5.2.1 Para evitar que uma rota oscile entre dois ou mais caminhos com a mesma métrica, o RIP especifica que uma rota deve ser atualizada somente quando a nova rota possuir distância menor que a atual.

OSPF - *Open Shortest-Path-First Protocol*

O protocolo OSPF foi elaborado por um grupo de trabalho da *Internet Engineering Task Force* com o propósito de atender às exigências de roteamento de grandes redes, ou seja, um IGP para sistemas autônomos de porte. É um protocolo que usa o algoritmo SPS e compreende

uma série de facilidades adicionais listados a seguir, as quais permitem diminuir a sobrecarga necessária para a manutenção da topologia atualizada de uma rede *Internet*:

- roteamento levando em consideração o tipo de serviço;
- balanceamento de carga entre rotas de mesmo tamanho;
- participação dos *gateways* e redes em subgrupos denominados áreas, sendo a topologia de uma área conhecida apenas dentro da mesma, facilitando o crescimento modular do SA ;
- definição da topologia de rede virtual que abstraia detalhes de rede real;
- divulgação e informações recebidas de *exterior gateways*. O formato da mensagem permite distinguir informações recebidas de fontes externas daquelas recebidas dentro do SA ;

O protocolo OSPF é baseado nas mensagens: *Hello, Database Description, Link Status Request e Link Status Uptade*. Quando um *gateway* OSPF é inicializado, sua primeira ação é contatar os *gateways* vizinhos, através de mensagens *Hello*. Os gateways trocam mensagens entre si para eleger o ***gateway mestre*** (DR -Designated Router). Este *gateway* torna-se responsável pela notificação de informações de roteamento a todos os *gateways* presentes na rede (*gateways* secundários). Nos protocolos de roteamento discutidos anteriormente todos os *gateways* enviavam e recebiam informações de roteamento, gerando tráfego excessivo. A figura de um *gateway* mestre, com o função de gerador/distribuidor de informações, reduz significativamente o tráfego relativo às mensagens de roteamento, que são trocadas somente entre o *gateway* mestre e os demais gateways secundários.

O OSPF usa o roteamento *link state.* As informações de roteamento trocados entre *gateways,* através da mensagem *Database Description*, indicam o estado e o custo associado às interfaces e aos *gateways* vizinhos. Estas mensagens são confirmadas pelos gateways que a recebem. Como as bases podem ser grandes, uma base de topologia pode gerar várias mensagens. A mensagem *Link Status Request* é usada por um gateway na requisição de dados atualizados a outro *gateway.* Na mensagem *Link Status Updade* é usada por um *gateway* no envio de informações sobre o estado de seus enlaces.

Uma vez estabelecido o *gateway* mestre da cada sub-rede *Internet* ,realizada a troca de informações de roteamento entre o *gateways* mestres das várias sub-redes em que esteja conectado, o gateway monta a sua base de dados de roteamento. O algoritmo SPF é, então

executado a partir dessa base e, como resultado, é obtida uma árvore de roteamento com o *gateway* na raiz, indicando a conectividade com outras redes. A partir dos dados de custo, são calculados os custos totais das rotas até cada sub-rede da *Internet.*

5.3.2. EGP - Exterior Gateway Protocol

O protocolo EGP não está vinculado a nenhum algoritmo de roteamento. Isto é, para que dois *gateways* se comuniquem através do EGP não é necessário que eles executem um mesmo algoritmo de roteamento. O EGP define as informações a serem trocadas entre EG (basicamente as tabelas de roteamento) e os elementos de protocolo necessários à troca dessas informações.

Tais informações permitem que um ou mais sistemas autônomos sejam utilizados como intermediários do tráfego originado em algum sistema autônomo e destinado a outro, sem que o usuário da rede perceba que a rede é composta por mais de um sistema autônomo.

O EGP é um protocolo de roteamento elaborado para uma rede de sistemas autônomos organizados em uma estrutura tipo árvore, ou seja, uma rede sem *loops* (ciclos) na sua topologia. As informações trocadas neste protocolo não impedem que ocorram *loops* no roteamento. Uma situação de *loop* pode ocorrer, por exemplo, quando uma tabela de roteamento de um *gateway* G indica o *gateway* G' como a melhor saída para uma rede N, e a tabela de roteamento de G' indica G como a melhor saída para a rede N.

As mensagens do EGP são associadas a cada Sistema Autônomo através de uma identificação de 16 *bits* que é colocada no cabeçalho da cada mensagem do EGP. Essas mensagens trafegam somente entre *gateways* vizinhos. Dois *gateways* podem tornar-se vizinhos quando:

- estão diretamente conectadas por um, cabo, por exemplo; ou

- estão conectados por uma rede transparente para eles, isto é, uma rede cuja estrutura interna eles não conhecem.

Não faz parte do protocolo EGP determinar quando dois gateways devem tornar-se vizinhos. Esta é uma tarefa do administrador do SA . Dois *gateways* tornam-se vizinhos através da troca de mensagens de **Aquisição de Vizinho**.

Após se tornarem vizinhos, dois gateways passam a trocar mensagens de **Disponibilidade**, para conhecer o estado do vizinho (conectado/desconectado), e mensagens de **Alcance**, para identificar quais redes podem ser acessadas através do vizinho.

O EGP é um protocolo do tipo solicitação *(polling).* As mensagens são trocadas somente quando ocorre uma solicitação de um dos vizinhos. Por isso, o EGP permite que cada *gateway* controle a sua taxa de envio e recebimento de informação de roteamento. Além disso, esse protocolo permite que um S A tenha um mecanismo de roteamento interno que não é afetado por falhas em outros sistemas.

5.3.3. BGP – Border Gateway Protocol

Com o crescimento da Internet, o uso do EGP tornou-se limitado. Existia a necessidade de acrescentar funções de policiamento no roteamento e o protocolo devia suportar topologias complexas. Consequentemente surgiu o BGP, para suprir as deficiências do EGP no roteamento entre sistemas autônomos.

Roteadores com BGP se preocupam com critérios políticos de roteamento. Um sistema autônomo (SA) deve querer habilidade de enviar pacotes para algum site e receber pacotes de outro site de seu interesse. Entretanto, ele não deve gostar de conduzir pacotes entre sistemas autônomos (SA's) que não seja de seu interesse. Por exemplo, companhias telefônicas devem atuar como portadora de seus clientes, mas não dos outros.

O protocolo BGP foi projetado para permitir muitos critérios de roteamento a serem aplicadas no tráfego entre SA's. Critérios típicos envolvem considerações de ordem política, de segurança, ou econômicas. Alguns exemplos de limites de roteamento são: nunca coloque o Iraque na rota para o Pentágono; tráfego iniciando ou terminando na IBM, não trafega para Microsoft. Os critérios são configurados manualmente em cada roteador BGP.

Do ponto de vista do roteador BGP, o mundo consiste de outros roteadores BGP interconectados. Dois roteadores BGP são considerados conectados se eles compartilham uma rede comum. Dado o interesse de um BGP especial no tráfego, as redes são grupadas em três categorias. A primeira categoria é ***stubs networks***, na qual somente tem conexão para um roteador BGP. Estas não podem ser usadas para trânsito na rede, porque só tem uma ligação.

A segunda é as redes ***multiconnected networks***. Podem ser usadas para tráfego em trânsito, exceto se recusarem. Finalmente, existem as redes ***transit networks***, como um backbone, que estão dispostas a manipular pacotes de outros, possivelmente com algumas restrições.

Pares de roteadores BGP se comunicam através de conexões TCP. Operando deste modo eles fornecem uma comunicação confiável e escondem os detalhes da rede que os pacotes estão passando.

BGP é um protocolo que usa o algoritmo *vector distance*, mas com uma pequena diferença. Ao invés de manter a distância de cada destino, cada roteador BGP mantém o caminho usado. Similarmente, ao invés de periodicamente dar a cada vizinho a distância estimada para cada possível destino, cada roteador diz a seus vizinhos o caminho exato que está usando.

Como um exemplo, considerar os roteadores conforme a Figura 5.3. Em particular, considerar a tabela de roteamento de F. Supor que ele usa o caminho FGCD para alcançar D. Quando os vizinhos sua informação de roteamento, eles fornecem seus caminhos completos, como mostra a Figura 5.3 (por simplicidade, somente o destino D é ilustrado).

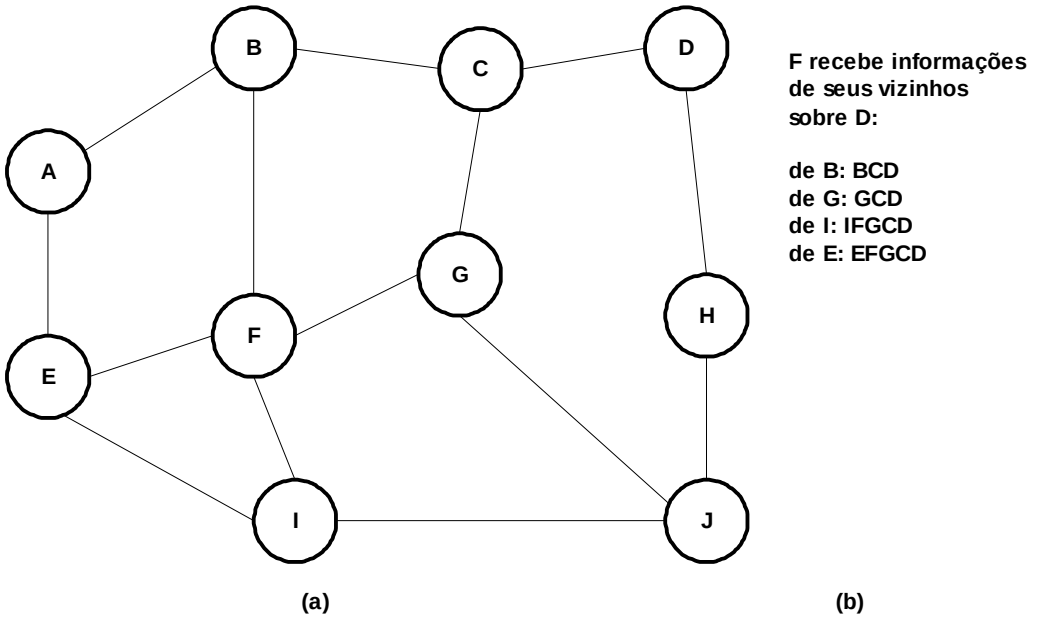

Figura 5.3. Roteamento com o protocolo BGP.

Após todos os caminhos chegarem dos vizinhos. F examina-os para ver qual é o melhor. Rapidamente descarta os caminhos de I e E, porque estes caminhos passam por F. A escolha está entre B e G. Cada roteador BGP contém um módulo que examina rotas para um dado destino e dá um valor a eles, retornando um número da distância para o destino de cada rota. Alguma rota viola um critério de roteamento e seu valor é infinito. O roteador então adota a rota de menor distância.

BGP facilmente resolve o problema de contagem infinita que causa problema a outros algoritmos de roteamento. Por exemplo, supor que G quebre ou a linha FG esteja desativada. F então recebe rotas dos outros três vizinhos. Estas rotas são BCD, IFGCD e EFGCD. Verifica-se que duas rotas estão sem sentido, porque passam por F, então é escolhido FBCD como a nova rota.

Vector-distance, uma diferente estratégia de podar a árvore deve ser seguida. O algoritmo básico é o *reverse path forwarding*. Entretanto, quando um roteador que não faz parte do grupo,

recebe uma mensagem multicast, ele responde para que o emissor não envie mensagens para ele.

5.4. ROTEAMENTO MULTICAST

A comunicação IP normal é ponto-a-ponto. Entretanto, para algumas aplicações, a comunicação multiponto é útil para o processo de enviar mensagens para um grande número de receptores simultaneamente. Exemplos de aplicações multiponto são replicação de dados, banco de dados distribuídos e multiconferência.

O IP suporta *multicast*, usando a classe de endereços D. Cada endereço da classe D identifica um grupo de estações. No endereço IP, 28 bits estão disponíveis para identificar grupos. Quando um processo envia um pacote para endereço de classe D, o pacote é liberado para todos os membros do grupo, mas não garante que todos receberão o pacote.

Existe dois tipos de grupos de endereços: permanente e temporário. Um grupo permanente sempre existirá e não precisa ser configurado. Alguns exemplos de endereços de grupo permanente são:

- 224.0.0.1 - todos os sistemas numa rede local;
- 224.0.0.2 - todos os roteadores numa rede local;
- 224.0.0.5 - todos os roteadores OSPF numa rede local.

Multicasting é implementado por roteadores *multicast* especiais. Cerca de uma vez a cada minuto, cada roteador *multicast* envia um pacote para as estações de sua rede local (endereço 224.0.0.1) perguntando para eles responderem de volta, quais os grupos que seus processos pertencem. Estes pacotes de consultas e respostas usam um protocolo chamado IGMP (*Internet Group Management Protocol)* , que é similar ao ICMP. Ele tem dois tipos de pacote: consulta e resposta, cada um com formato fixo contendo alguma informação de controle na primeira palavra do campo *payload* e um endereço classe D na segunda palavra.

Para algumas aplicações, os processos estão separados em vários locais, mas trabalham juntos em grupo. Por exemplo, um grupo de processos implementando um sistema de banco de dados distribuído. Nele é freqüente um processo enviar uma mensagem para todos os outros membros do grupo. Então necessita-se de um modo de enviar mensagens para grupos bem definidos que são numericamente grandes, mas pequenos comparados ao tamanho da rede. Para realizar esta tarefa é necessário utilizar uma técnica de roteamento multiponto.

Para fazer ***multicasting*****,** cada roteador constrói sua ***spanning tree*** selecionando enlaces na rede formando uma árvore, de forma a cobrir todos os outros roteadores na sub-rede. Por exemplo, na Figura 5.4(a) temos uma sub-rede com dois grupos, 1 e 2. Alguns roteadores são ligados nos grupos, como indicado na Figura 5.4(a). A ***spanning tree*** para o roteador mais a esquerda é mostrado na figura 5.4(b).

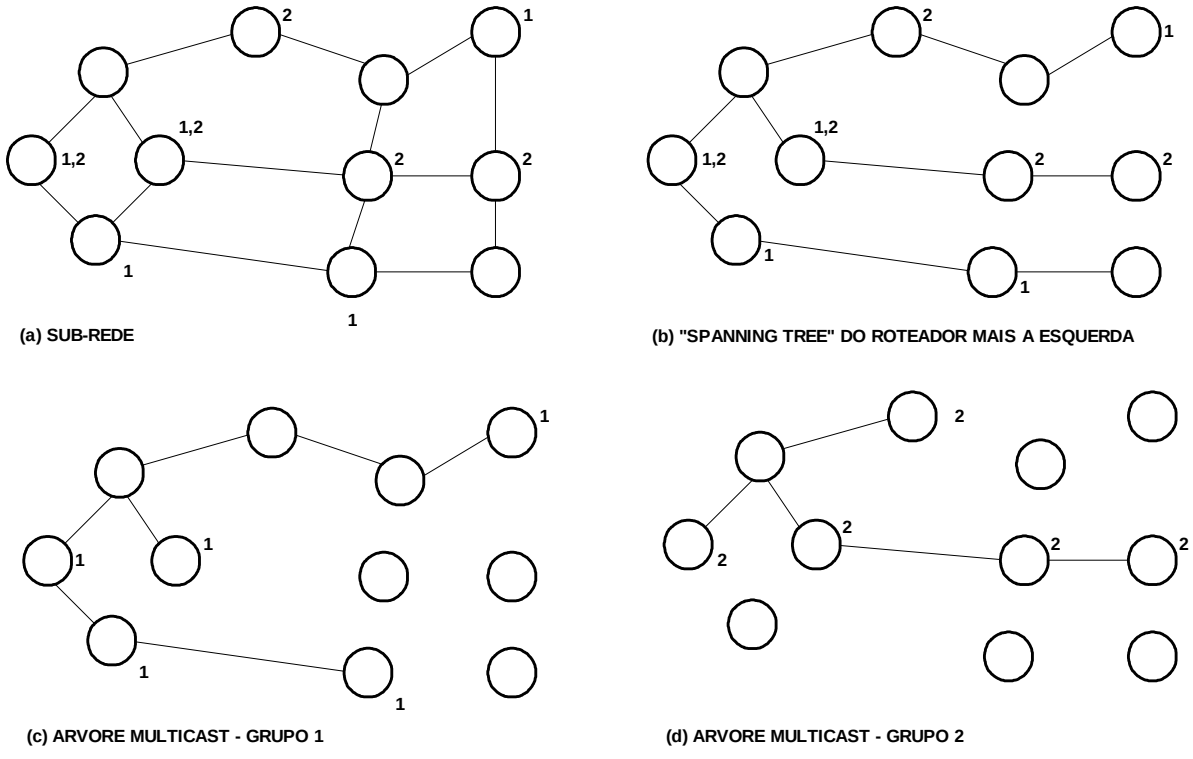

Figura 5.4. Roteamento Multicast.

Quando um processo envia um pacote multicast para o grupo, o primeiro roteador examina sua *spanning tree* e poda a árvore, removendo toda as linhas que não pertencem ao seu grupo. A Figura 5.4(c) ilustra a árvore do grupo 1 e a Figura 5.4(d) ilustra a árvore do grupo 2. Pacotes *multicast* são enviados somente na *spanning tree* apropriada.

Vários modos de podar a árvore são possíveis. O modo mais simples é usar o roteamento *link-state*, sendo que cada roteador deve conhecer a topologia completa da sub-rede. Então a *spanning tree* pode ser construída iniciando no final de cada caminho até a raiz, eliminando os roteadores que não pertencem ao grupo.

Com o roteamento *vector-distance*, uma diferente estratégia de podar a árvore deve ser seguida. O algoritmo básico é o *reverse path forwarding*. Entretanto, quando um roteador que não faz parte do grupo, recebe uma mensagem multicast, ele responde para que o emissor não envie mensagens para ele.

Uma desvantagem deste algoritmo é que para grandes redes muita memória é necessária. Supor que uma rede tem ***n*** grupos, cada um com a média de ***m*** membros. Para cada grupo, ***m*** *spanning trees* podadas são armazenadas, para um total de ***m.n*** árvores. Quando muitos grupos grandes existem, é gasto muito memória para armazenar as árvores.

Uma alternativa é usar árvores chamadas *core-base tree*. Aqui uma única árvore *spanning tree* por grupo é computada, com a raiz ("the core") perto do meio do grupo. Para enviar uma mensagem *multicast*, uma estação envia-a para a raiz, que então envia para os nós do grupo. Embora esta técnica não seja ótima, ela reduz os custos de armazenagem de ***m*** árvores para uma árvore por grupo.

6. TCP/IP EM REDES ATM

O ATM é uma tecnologia de rede de alta velocidade na qual a rede possui um ou mais comutadores conectados entre si para formar uma estrutura de comutação. Na lógica, uma estrutura de comutação funciona como uma única grande rede que permite a comunicação entre quaisquer hosts.

Já que o ATM é uma tecnologia baseada em conexão, dois computadores devem estabelecer um circuito virtual através da rede antes de transmitir dados. Um host pode optar por um circuito virtual comutado ou permanente. Os circuitos comutados são criados por demanda. Os permanentes requerem uma configuração manual. Em ambos os casos, o ATM atribui a cada circuito aberto um identificador de número inteiro. Cada quadro enviado pelo host ou pela rede possui um identificador de circuito. Um quadro não possui um endereço de origem e de destino.

Apesar dos níveis mais baixos do ATM utilizarem células de 53 octetos para a transmissão de dados, o ATM possui mecanismos adicionais em sua camada de adaptação que são utilizados pelos aplicativos. A AAL5 (Adaptation Layer 5) do ATM, em particular, é utilizada para o envio de dados através de uma rede ATM. A AAL5 oferece uma interface que recebe e envia blocos de dados de tamanhos variados, os quais podem ter octetos de até 64kb.

Para enviar um datagrama IP através de uma rede ATM, o transmissor precisa formar uma conexão de circuito virtual para o destino que utiliza a AAL5 e enviar o datagrama à AAL5 coo um único bloco de dados. A AAL5 acrescenta um trailer, divide o datagrama e o trailer em células menores para a transmissão através da rede e, depois, reagrupa o datagrama antes de encaminhá-lo ao sistema operacional do computador de destino. Assim, ao enviar datagramas através da rede ATM, o IP não fragmenta no tamanho da célula ATM. Ao contrário, o IP utiliza uma MTU[Ψ] de 9.180 octetos e permite que a AAL5 divida o datagrama em células.

[Ψ] MTU (Maximum Transfer Unit) significa o maior volume de dados que pode ser transferido em determinada rede física, a MTU é determinada pelo hardware da rede.

Uma LIS (Logical IP Subnet) é formada por um conjunto de computadores que utiliza a rede ATM em vez de uma rede local. Os computadores formam circuitos virtuais entre si por meio dos quais alteram os datagramas. A presença tanto dos circuitos virtuais como dos circuitos permanentes em uma LIS torna ainda mais complicada a questão da vinculação de endereços. Um protocolo ARP modificado, conhecido como ATMARP, faz a vinculação de endereços para os computadores em uma LIS conectada por um circuito virtual comutado. Os computadores de uma LIS contam com um servidor ATMARP para vincular o endereço IP de outro computador da LIS a um endereço ATM equivalente. Cada computador de uma LIS deve fazer o registro com o servidor, fornecendo seus endereços IP e ATM ao servidor. Desse modo, outros computadores podem entrar em contato com o servidor para obter uma vinculação, conforme necessário. Como no caso do ARP convencional, uma vinculação deve ser revalidada ou descartada. Um protocolo relativo ao ATMARP inverso é utilizado para descoberta dos endereços ATM e IP de um computador remoto conectado por um circuito virtual permanente.

7. DNS (DOMAIN NAME SYSTEM)

O DNS (Domain Name System) é um esquema de gerenciamento de nomes, hierarquico e distribuído. O DNS define a sintaxe dos nomes usados na Internet, regras para delegação de autoridade na definição de nomes, um banco de dados distribuído que associa nomes a atributos (entre eles o endereço IP) e um algoritmo distribuído para mapear nomes em endereços. O DNS é especificado nas RFCs[Ψ] 882, 883 e 973.

As aplicações normalmente utilizam um endereço IP de 32 bits no sentido de abrir uma conexão ou enviar um datagrama IP. Entretanto, os usuários preferem identificar as máquinas através de nomes ao invés de números. Assim e necessário um banco de dados que permita a uma aplicação encontrar um endereço, dado que ela conhece o nome da máquina com a qual se deseja comunicar.

Um conjunto de servidores de nomes mantém o banco de dados com os nomes e endereços das máquinas conectadas a Internet. Na realidade este é apenas um tipo de informação armazenada no domain system (sistema de domínios). Note que é usado um conjunto de servidores interconectados, ao invés de um único servidor centralizado. Existem atualmente tantas instituições conectadas a Internet que seria impraticável exigir que elas notificassem uma autoridade central toda vez que uma máquina fosse instalada ou trocasse de lugar. Assim, a autoridade para atribuição de nomes e delegada a instituições individuais. Os servidores de nome formam uma árvore, correspondendo a estrutura institucional. Os nomes também adotam uma estrutura similar.

Um exemplo típico é o nome **chupeta.jxh.xyz.br**. Para encontrar seu endereço Internet, pode ser necessário o acesso a até quatro servidores de nomes. Inicialmente deve ser consultado um servidor central, denominado servidor raiz, para descobrir onde está o servidor br. O servidor br

é o responsável pela gerencia dos nomes das instituições/empresas brasileiras ligadas a Internet. O servidor raiz informa como resultado da consulta o endereço IP de vários servidores de nome para o nível br (pode existir mais de um servidor de nomes em cada nível, para garantir a continuidade da operação quando um deles para de funcionar). Um servidor do nível br pode então ser consultado, devolvendo o endereço IP do servidor xyz.

De posse do endereço de um servidor xyz é possível solicitar que ele informe o endereço de um servidor jxh, quando, finalmente, pode-se consultar o servidor jxh sobre o endereço da máquina chupeta. O resultado final da busca é o endereço Internet correspondente ao nome chupeta.jxh.xyz.br.

Cada um dos níveis percorridos e referenciado como sendo um domínio. O nome completo chupeta.jxh.xyz.br é um nome de domínio.

Na maioria dos casos, não é necessário ter acesso a todos os domínios de um nome para encontrar o endereço correspondente, pois os servidores de nome muitas vezes possuem informações sobre mais de um nível de domínio o que elimina uma ou mais consultas. Além disso, as aplicações normalmente tem acesso ao DNS através de um processo local (servidor para as aplicações e um cliente DNS), que pode ser implementado de modo a guardar os últimos acessos feitos, e assim resolver a consulta em nível local. Essa abordagem de acesso através de um processo local, simplifica e otimiza a tarefa das aplicações no que tange ao mapeamento de nomes em endereços, uma vez que elimina a necessidade de implementar, em todas as aplicações que fazem uso do DNS, o algoritmo de caminhamento na árvore de domínios descrito anteriormente.

O DNS não se limita a manter e gerenciar endereços Internet. Cada nome de domínio e um nó em um banco de dados, que pode conter registros definindo varias propriedades. Por exemplo, o tipo da máquina e a lista de serviços fornecidos por ela. O DNS permite que seja definido um

[Ψ] RFC (Request For Comments), nome de um conjunto de notas que contêm levantamentos, avaliações, idéias, técnicas e comentários, bem como padrões de protocolos TCP/IP sugeridos e aceitos. As RFCs estão disponíveis on-line.

alias (nome alternativo) para o nó. Também é possível utilizar o DNS para armazenar informações sobre usuários, listas de distribuição ou outros objetos.

O DNS é particularmente importante para o sistema de correio eletrônico. No DNS são definidos registros que identificam a máquina que manipula as correspondências relativas a um dado nome, identificado assim onde um determinado usuário recebe suas correspondências. O DNS pode ser usado também para definição de listas para distribuição de correspondências.

8. APLICAÇÕES

As aplicações, no modelo TCP/IP, não possuem uma padronização comum. Cada uma possui um RFC próprio. O endereçamento das aplicações é feito através de portas (chamadas padronizadas a serviços dos protocolos TCP e UDP), por onde são passados as mensagens. É na camada de Aplicação que se trata a compatibilidade entre os diversos formatos representados pelos variados tipos de estações da rede.

8.1. TELNET

O protocolo TCP/IP inclui um protocolo simples de terminal remoto denominado TELNET. O TELNET permite que um usuário em determinado site estabeleça uma conexão TCP com um servidor login situado em outro site. O TELNET transmite, então os toques no teclado do usuário diretamente ao computador remoto, como se estivessem sendo digitados no teclado conectado à máquina remota. Esse terminal também retorna a saída da máquina remota até a tela do usuário. O servidor recebe o nome de *transparente*, porque faz com que o teclado e o monitor do usuário pareçam estar conectados diretamente à máquina remota.

Embora o TELNET não seja sofisticado se comparado a alguns protocolos de terminal remoto, ele é amplamente aceito. Geralmente o software do cliente do TELNET permite que o usuário especifique a máquina remota fornecendo seu nome de domínio ou seu endereço IP. Por aceitar endereços IP, o TELNET pode ser usado com hosts mesmo que o vínculo nome/endereço não possa ser estabelecido (p. ex., quando o software de atribuição de nomes de domínio estiver sendo depurado).

O TELNET oferece três serviços básicos:

- Define um terminal virtual da rede fornecedora de uma interface padrão para sistemas remotos. Os programas clientes não precisam entender os detalhes de todos os sistemas remotos possíveis; eles são projetados para usar a interface padrão.

- Inclui um mecanismo que permite ao cliente e ao servidor negociar opções e fornece um conjunto de opções padronizadas (p. ex., uma das opções verifica se os dados passados pela conexão usam o caracter padrão ASCII de sete bits ou o conjunto de oito bits).

- Trata ambas as pontas da conexão de forma simétrica. Em particular, não obriga a entrada do cliente via teclado, nem obriga o cliente a ter a saída indicada na tela. Dessa forma, o TELNET permite que o programa arbitrário torne-se um cliente. Além do mais, qualquer ponta pode negociar opções.

A Figura 8.1 ilustra como os programas aplicativos implementam um cliente e servidor TELNET, quando um usuário chama o TELNET, um programa aplicativo existente na máquina do usuário torna-se o cliente. O cliente estabelece uma conexão TCP com o servidor por intermédio da qual irão se comunicar. Uma vez estabelecida a conexão, o cliente aceita toques de teclado do usuário e os envia ao servidor enquanto, simultaneamente, aceita caracteres que o servidor envia de volta e apresenta-os na tela do usuário. O servidor deve aceitar uma conexão TCP de um cliente e, a seguir, retransmitir dados entre a conexão TCP e o sistema operacional local.

Na prática, o servidor é mais complexo do que a figura representa porque precisa conduzir várias conexões simultâneas. Em geral, um processo de servidor-mestre aguarda novas conexões e cria um novo escravo para cuidar de uma conexão em particular. Desse modo, “o servidor TELNET” mostrado na Figura 8.1 representa o escravo que trata de uma conexão em particular. A figura não mostra o servidor-mestre que espera novas solicitações, nem mostra os escravos cuidando das outras conexões.

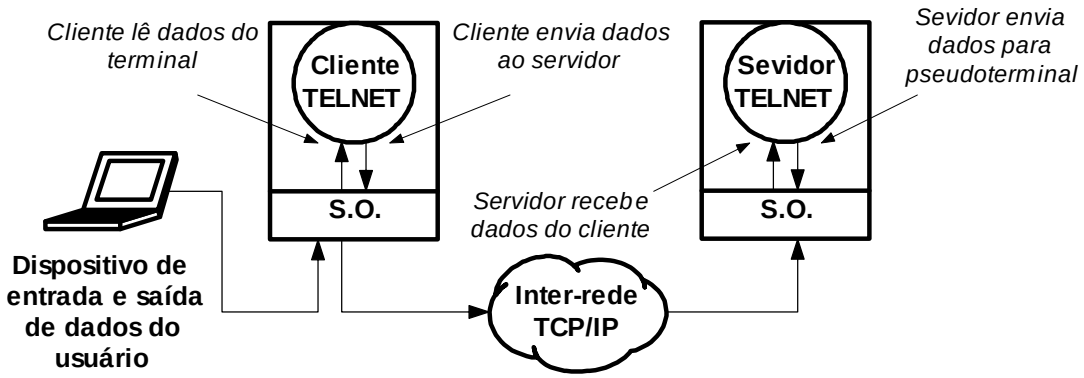

Figura 8.1. Trajeto de dados em uma sessão de terminal remoto TELNET, enquanto trafega do teclado do usuário até um sistema de operação remoto. Acrescenta um servidor TELNET a um sistema timesharing geralmente requer modificações no sistema operacional.

O termo pseudoterminal descreve o ponto de entrada do sistema operacional que permite que o servidor de um programa em funcionamento, como o TELNET, transfira caracteres ao sistema operacional como se estivessem vindo de um teclado. É impossível construir um servidor TELNET, a não ser que o sistema operacional forneça tal recurso. Se o sistema suporta tal abstração de pseudoterminal, o servidor TELNET pode ser implementado com programas aplicativos. Cada servidor-escravo conecta um canal TCP de um cliente a um pseudoterminal específico.

8.2. FTP (File Transfer Protocol)

A transferência de arquivos é um dos aplicativos TCP/IP usados com mais freqüência e responde por grande parte do tráfego de rede. Os protocolos de transferência de arquivos padrão existiram para ARPANET antes que o TCP/IP se tornasse operacional. Essas versões anteriores do software de transferência de arquivos evoluíram para um padrão atual conhecido como FTP (File Transfer Protocol).

O FTP oferece muitas vantagens além da função de transferência propriamente dita:

- *Acesso interativo*. Apesar do TCP ser projetado para uso por programas, muitas implementações fornecem uma interface interativa que permite que as pessoas interajam facilmente com servidores remotos. Por exemplo, um usuário pode pedir uma listagem de todos os arquivos de um diretório em uma máquina remota. O cliente também responde normalmente à entrada "help", mostrando a informação do usuário sobre possíveis comandos que podem ser chamados.

- *Especificação de formato (representação)*. O FTP permite que o cliente especifique o tipo e o formato dos dados armazenados. Por exemplo, o usuário pode determinar se um arquivo contém texto ou números inteiros binários e se os arquivos textos usam os conjuntos de caracteres ASCII ou EBCDIC.

- *Controle de autenticação*. O FTP requer que os clientes autorizem a si próprios enviando um nome de login e a senha ao servidor antes de requisitar a transferência e arquivos. O servidor recusa acesso aos clientes que não podem fornecer um login válido e uma senha.

A exemplo de outros servidores, a maioria das implementações de servidor FTP permite o acesso simultâneo de vários clientes. Os clientes usam o TCP para se conectar a um servidor. Um processo de servidor principal único guarda concessões e cria um processo escravo para tratar cada conexão. De maneira diferente de outros servidores, no entanto, o processo escravo não realiza toda a computação necessária. Ao contrário, o escravo aceita e trata a conexão de controle do cliente, mas usa um processo ou processos adicionais para tratar uma conexão de transferência de dados à parte. A conexão de controle transporta comandos que informam ao servidor qual arquivo transferir. A conexão de transferência de dados, que também usa o TCP como protocolo de transferência, transporta todas as transferências de dados.

Geralmente tanto o cliente como o servidor criam um processo separado para tratar da transferência de dados. Considerando que os detalhes exatos da arquitetura do processo dependam do sistema operacional usado, a Figura 8.2 ilustra o conceito:

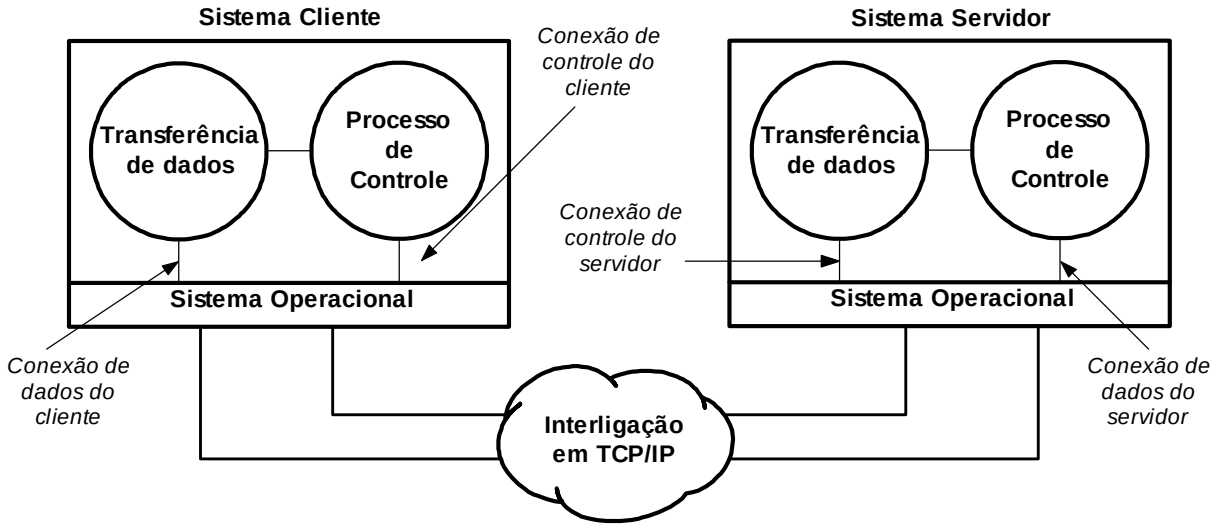

Figura 8.2. Um cliente e um servidor FTP com uma conexão de controle TCP entre eles e uma conexão TCP à parte entre seus processos e transferência de dados relacionados.

A conexões de transferência de dados e os processos de transferência de dados que as utilizam podem ser criados dinamicamente, quando necessário, mas a conexão de controle persiste através de uma sessão. Uma vez que a conexão de controle desaparece, a sessão é finalizada e o software de ambas as extremidades encerra todos os processos de transferência de dados.

8.3. NFS (Network File System)

Desenvolvido inicialmente pela Sun Microsystems, o NFS (Network File System) fornece acesso de arquivo online compartilhado que é transparente e integrado. Muitos sites TCP/IP usam NFS para interconectar seus sistemas de arquivo de computadores. Da perspectiva do usuário, o NFS é praticamente invisível. Um usuário pode executar um programa aplicativo arbitrário e usar arquivos arbitrários para entrada ou saída. O próprio nome dos arquivos não indica se eles são locais ou remotos.

A Figura 8.3 ilustra como o NFS está inserido no sistema operacional. Quando um programa aplicativo é executado, este chama o sistema operacional para abrir um arquivo ou para armazenar e recuperar dados de um arquivo. O mecanismo de acesso a arquivos aceita o pedido e automaticamente passa-o ou para o software de sistema de arquivo local ou para o cliente NFS, dependendo de o arquivo estar no disco local ou em uma máquina remota. Quando ele recebe um pedido, o software do cliente usa o protocolo NFS para contatar o servidor apropriado em uma máquina remota e executar a operação requisitada. Quando o servidor remoto responde, o software do cliente devolve os resultados ao programa aplicativo.

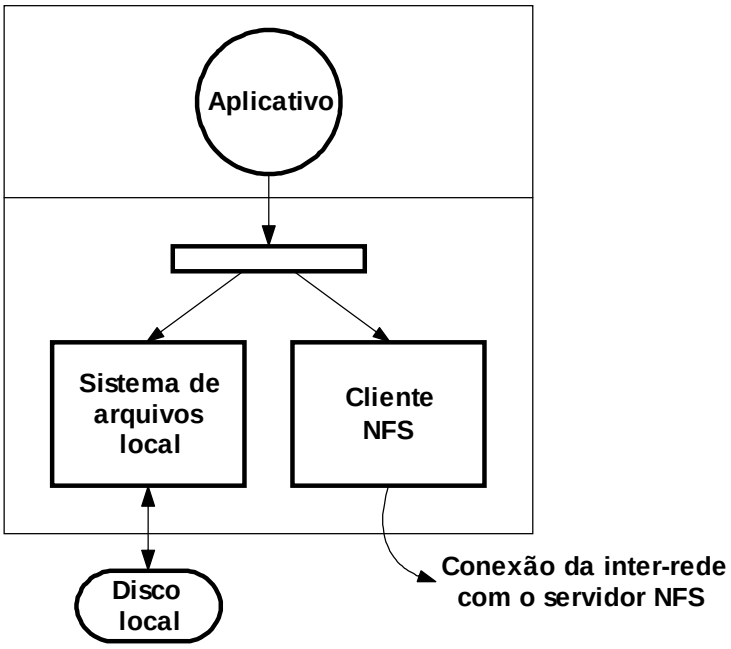

Figura 8.3. Código NFS em um sistema operacional. Quando um programa aplicativo requisita uma operação de arquivo, o sistema operacional deve passar o pedido para o sistema de arquivo local ou para o software do cliente NFS.

8.4. RPC (Remote Procedure Call)

Em vez de definir o protocolo NFS a partir de riscos, os projetistas deviam preferir montar três peças independentes: o próprio protocolo NFS, um mecanismo de RPC (Remote Procedure Call) para fins gerais e um XDR (eXternal Data Representation), também para fins gerais. O objetivo seria separar os três para possibilitar o uso de RPC e XDR em outro software, inclusive em programas aplicativos e também em outros produtos.

Do ponto de vista do programador, o próprio NFS não fornece novos procedimentos que possam ser chamados por um programa. Ao contrário, uma vez que um administrador tenha configurado o NFS, ele programa arquivos remotos de acesso usando exatamente as mesmas operações utilizadas para arquivos locais. No entanto, o RPC e o XDR fornecem macanismos que os programadores podem usar para construir programas distribuídos. Por exemplo, um programador pode dividir um programa entre o lado do cliente e o lado do servidor que usam RPC como o principal mecanismo de comunicação. No lado do cliente, o programador atribui alguns procedimentos como remotos, obrigando o compilador a incorporar a esses procedimentos os códigos de RPC. No lado do servidor, o programador implementa os procedimentos desejados e usa outros recursos RPC para declará-los parte do servidor. Quando o programa de execução do cliente chama um dos programas remotos, o RPC automaticamente coleta valores para argumentos, monta uma mensagem, envia a mensagem ao servidor remoto, espera uma resposta e armazena os valores devolvidos nos argumentos atribuídos. Basicamente, a comunicação com o servidor remoto ocorre automaticamente como efeito parcial de uma chamada de procedimento remoto. O mecanismo RPC concentra todos os detalhes do protocolo, possibilitando aos programadores que têm pouco conhecimento de protocolos de comunicação básicos desenvolver programas distribuídos.

O XDR é uma ferramenta relacionada que fornece meios para que os programadores passem dados entre máquinas heterogêneas sem codificar procedimentos que convertam as representações de dados de hardware. Por exemplo, nem todos os computadores armazenam números inteiros binários de 32 bits no mesmo formato. Alguns armazenam o byte mais

significativo no endereço de memória mais alta, enquanto outros armazenam o byte menos significativo no endereço mais alto. Assim, se os programadores usam uma rede simplesmente para mover bytes de número inteiro, de uma máquina para outra, sem redistribuí-los, o valor do número inteiro pode mudar. O XDR resolve o problema definindo uma representação de máquina independente. Em uma extremidade de um canal de comunicação, um programa chama procedimentos XDR para fazer a conversão da representação do hardware local para a representação independente da máquina. Uma vez que os dados tenham sido transferidos para outra máquina, o programa receptor solicita rotinas XDR para proceder à conversão da representação independente da máquina para a representação local da máquina.

A principal vantagem do XDR é automatizar grande parte da tarefa de conversão de dados. Os programadores não precisam digitar chamadas de procedimentos XDR normalmente. Ao contrário, eles fornecem o compilador XDR com os estatutos de declaração do programa para o qual os dados devem ser transformados e o compilador automaticamente gera um programa com as chamadas de biblioteca XDR necessárias.

8.5. SMTP (Simple Mail Transfer Protocol)

O SMTP (Simple Mail Transfer Protocol) e o protocolo usado no sistema de correio eletrônico na arquitetura Internet TCP/IP. Um usuário, ao desejar enviar uma mensagem, utiliza o módulo interface com o usuário para compor a mensagem e solicita ao sistema de correio eletrônico que a entregue ao destinatário. Quando recebe a mensagem do usuário, o sistema de correio eletrônico armazena uma cópia da mensagem em seu spool (área do dispositivo de armazenamento), junto com o horário do armazenamento e a identificação do remetente e do destinatário. A transferência da mensagem é executada por um processo em background, permitindo que o usuário remetente, após entregar a mensagem ao sistema de correio eletrônico, possa executar outras aplicações.

O processo de transferência de mensagens, executando em background, mapeia o nome da máquina de destino em seu endereço IP, e tenta estabelecer uma conexão TCP com o servidor

de correio eletrônico da máquina de destino. Note que o processo de transferência atua como cliente do servidor do correio eletrônico. Se a conexão for estabelecida, o cliente envia uma cópia da mensagem para o servidor, que a armazena em seu spool. Caso a mensagem seja transferida com sucesso, o servidor avisa ao cliente que recebeu e armazenou uma cópia da mensagem. Quando recebe a confirmação do recebimento e armazenamento, o cliente retira a cópia da mensagem que mantinha em seu spool local. Se a mensagem, por algum motivo, não for transmitida com sucesso, o cliente anota o horário da tentativa e suspende sua execução. Periodicamente o cliente acorda e verifica se existem mensagens a serem enviadas na área de spool e tenta transmiti-las. Se uma mensagem não for enviada por um período, por exemplo de dois dias, o serviço de correio eletrônico devolve a mensagem ao remetente, informando que não conseguiu transmiti-la.

Em geral, quando um usuário se conecta ao sistema, o sistema de correio eletrônico e ativado para verificar se existem mensagens na caixa postal do usuário. Se existirem, o sistema de correio eletrônico emite um aviso para o usuário que, quando achar conveniente, ativa o módulo de interface com o usuário para receber as correspondências.

Um mensagem SMTP divide-se em duas partes: cabeçalho e corpo, separados por uma linha em branco. No cabeçalho são especificadas as informações necessárias para a transferência da mensagem. O cabeçalho e composto por linhas, que contém uma palavra-chave seguida de um valor. Por exemplo, identificação do remetente (palavra-chave “to:” seguida do seu endereço), identificação do destinatário, assunto da mensagem, etc... No corpo são transportadas as informações da mensagem propriamente dita. O formato do texto é livre e as mensagens são transferidas no formato texto.

Os usuários do sistema de correio eletrônico são localizados através de um par de identificadores. Um deles especifica o nome da máquina de destino e o outro identifica a caixa postal do usuário. Um remetente pode enviar simultaneamente varias cópias de uma mensagem, para diferentes destinatários utilizando o conceito de lista de distribuição (um nome que identifica um grupo de usuários). O formato dos endereços SMTP é o seguinte:

Nome_local@Nome do domínio

Onde o *nome_do_domínio* identifica o domínio ao qual a máquina de destino pertence (esse endereço deve identificar um grupo de máquinas gerenciado por um servidor de correio eletrônico). O *nome_local* identifica a caixa postal do destinatário.

O SMTP especifica como o sistema de correio eletrônico transfere mensagens de uma máquina para outra. O módulo interface com usuário e a forma como as mensagens são armazenadas não são definidos pelo SMTP. O sistema de correio eletrônico pode também ser utilizado por processos de aplicação para transmitir mensagens contendo textos.

9. FUTURO DO TCP/IP (IPv6)

Nem a Internet nem os protocolos TCP/IP são estáticos. Através de sua Força Tarefa de Engenharia da Internet, a Diretoria de Arquitetura da Internet (IAB, acrônimo de Internet Architecture Board) promove esforços efetivos e constantes que mantêm a tecnologia elástica e em evolução. O estímulo para a mudança ocorre à medida que aumentos em volume e em tamanho forçam as melhorias necessárias para manter o serviço, uma vez que novos aplicativos exigem mais da tecnologia denominada e já que novas tecnologias tornam possível fornecer novos serviços.

A versão 4 do Internet Protocol (IPv4, versão atual) fornece o mecanismo básico de comunicação da pilha TCP/IP e da Internet. Essa versão permaneceu quase inalterada desde seu início, no final da década de 1970. A longevidade da versão 4 mostra que o projeto é flexível e poderoso. Desde quando o IPv4 foi projetado, o desempenho do processador aumentou mais de duas ordens de magnitude, os tamanhos típicos de memória aumentaram 32 vezes, a largura de banda de rede cresceu 800 vezes, tecnologias de rede local afloraram e o número de hosts na Internet cresceu para 4 milhões. Além disso, as mudanças não ocorreram simultaneamente – o IP conciliou mudanças em uma tecnologia, diante das mudanças em outras.

O protocolo IPv6 proposto mantém muitas das características que contribuíram para o sucesso do IPv4. Na verdade, os projetistas dotaram o IPv6 basicamente com as mesmas características do IPv4, com algumas modificações. Por exemplo, o IPv6 ainda aceita entrega sem conexão (permite que cada datagrama seja roteado independentemente), permite que o transmissor escolha o tamanho de um datagrama e requer que o transmissor especifique o número máximo de passos da rota que um datagrama pode fazer antes de ser concluído. O IPv6 também retém a maioria dos conceitos fornecidos pelas opções do IPv4, inclusive os recursos para fragmentação e roteamento de origem.

A despeito de muitas semelhanças conceituais, o IPv6 muda a maioria dos detalhes do protocolo. Por exemplo, o IPv6 usa endereços maiores e acrescenta algumas características novas. Mais importante, revisa completamente o formato de datagrama, substituindo o campo de opções de comprimento variável do IPv4 por uma série de cabeçalhos de formato fixo. As mudanças introduzidas pelo IPv6 podem ser agrupadas em cinco categorias:

- *Endereços Maiores*. O novo tamanho de endereço é a mudança mais visível. O IPv6 quadruplica o tamanho de um endereço de IPv4, de 32 para 128 bits. O espaço de endereço de IPv6 é tão grande que não pode ser consumido em futuro previsível.

- *Formato Flexível de Cabeçalho*. O IPv6 usa um formato de datagrama inteiramente novo e incompatível. A contrário do IPv4, que usa um cabeçalho de datagrama de formato fixo onde todos os campos, exceto o de opções, ocupam um número fixo de octetos com um deslocamento fixo, o IPv6 usa um conjunto de cabeçalhos opcionais.

- *Opções Aprimoradas*. Como o IPv4, o IPv6 permite que um datagrama inclua informações de controle opcionais. O IPv6 inclui novas opções que oferecem recursos adicionais não disponíveis no IPv4.

- *Suporte para Alocações de Recursos*. O IPv6 substitui a especificação de tipo de serviço do IPv4 por um mecanismo que permite pré-alocação de recursos de rede. Particularmente, o novo mecanismo aceita aplicativos tais como vídeo em tempo real, que requer garantias de largura de banda e retardo de transmissão.

- *Provisão para Extensão de Protocolo*. Talvez a mudança mais significativa no IPv6 seja uma transição de um protocolo que especifica inteiramente todos os detalhes, para um protocolo que pode permitir recursos adicionais.

9.1. FORMATO DO DATAGRAMA

O IPv6 muda completamente o formato de datagrama. Como mostra a Figura 9.1, um datagrama IPv6 tem um cabeçalho básico de tamanho fixo seguido de zero, ou mais cabeçalhos de extensão seguidos de dados.

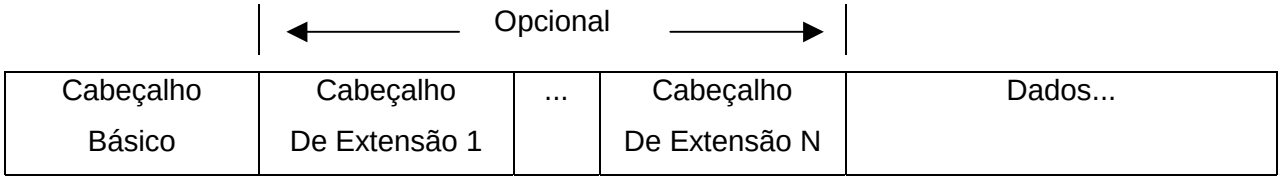

Figura 9.1. Forma geral de um datagrama IPv6 com vários cabeçalhos. Apenas o cabeçalho básico é exigido; os de extensão são opcionais.

Curiosamente, embora deva acomodar endereços maiores, um cabeçalho básico IPv6 contém menos informações do que um cabeçalho de datagrama IPv4. As opções e alguns dos campos fixos que aparecem em um cabeçalho de datagrama IPv4 foram removidos para cabeçalhos de extensão no IPv6. Em geral, as mudanças no cabeçalho de datagrama refletem mudanças no protocolo:

- O alinhamento foi mudado de múltiplos de 32 bits para múltiplos de 64 bits
- O campo de comprimento de cabeçalho foi eliminado e o campo de comprimento de datagrama foi substituído por um campo COMPRIMENTO DO PAYLOAD.
- O tamanho dos campos de endereço de origem e de destino foi aumentado para 16 octetos cada.
- As informações de fragmentação foram retiradas de campos fixos do cabeçalho básico, para um cabeçalho de extensão.

- O campo TEMPO DE VIDA foi substituído por um campo LIMITE DE PASSOS DE ROTA.

- O campo TIPO DE SERVIÇO foi substituído por um campo RÓTULO DE FLUXOS.

- O campo PROTOCOLO foi substituído por um campo que especifica o tipo do próximo cabeçalho.

A Figura 9.2 mostra o conteúdo e o formato do cabeçalho básico do IPv6.

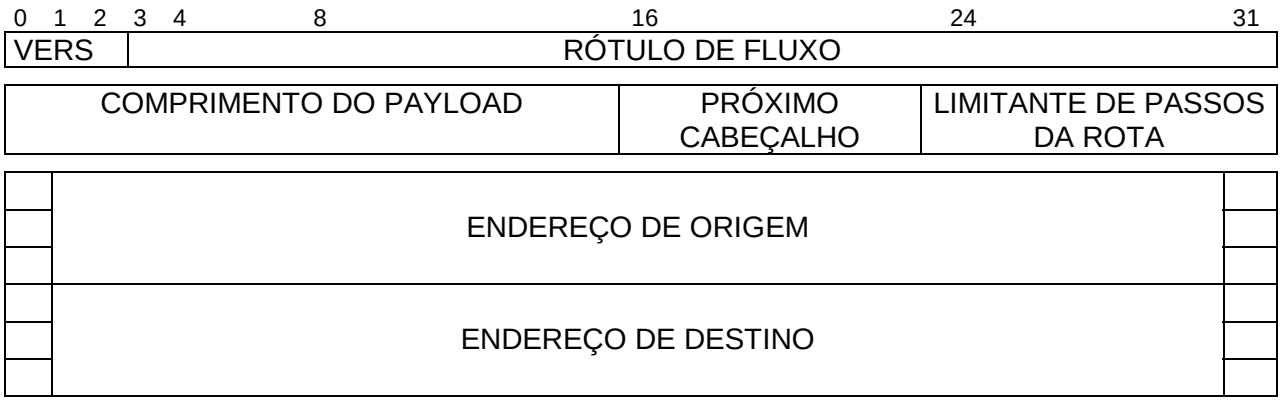

Figura 9.2. Formato de cabeçalho básico de 40 octetos do IPv6. Cada datagrama do IPv6 começa com um cabeçalho básico.

Vários campos de um cabeçalho básico do IPv6 correspondem diretamente aos campos de um cabeçalho do IPv4. Como no IPv4, o campo inicial VERS de 4 bits especifica a versão do protocolo; VERS sempre contém 6 em um datagrama IPv6. Como no IPv4, os campos ENDEREÇO DE ORIGEM e ENDEREÇO DE DESTINO especificam os endereços do transmissor e do destinatário pretendido. No IPv6, entretanto, cada endereço requer 16 octetos. O campo LIMITE DE PASSOS DA ROTA corresponde ao campo TEMPO DE VIDA (TIME TO LIVE) do IPv4. Ao contrário do IPv4, que interpreta um tempo de vida como uma combinação de contagem de passos da rota e do tempo máximo, o IPv6 interpreta o valor atribuindo limite estrito ao número máximo de passos da rota que um datagrama pode fazer antes de ser descartado.

9.2. TAMANHO DO ESPAÇO DE ENDEREÇO

Em IPv6, cada endereço ocupa 16 octetos, quatro vezes o tamanho de um endereço de IPv4. O espaço grande de endereço garante que o IPv6 pode tolerar qualquer esquema razoável de atribuição de endereço. De fato, se posteriormente os projetistas decidirem mudar o esquema de endereçamento, o espaço de endereço será suficientemente grande para acomodar uma nova atribuição.

É difícil compreender o tamanho do espaço de endereço de IPv6. Um modo de examiná-lo consiste em relacionar a magnitude ao tamanho da população: o espaço de endereço é tão grande que cada pessoa do planeta pode ter endereços suficientes para ter sua própria interligação em redes tão grande quanto a Internet atual. Um outro modo de compreender o tamanho é o relacionar ao esgotamento do endereço. Por exemplo, considere quanto tempo você levaria para atribuir todos os endereços possíveis. Um número inteiro de 16 octetos pode conter 2^{128} valores. Assim, o espaço de endereço é maior do que $3{,}4 \times 10^{38}$. Se os endereços forem atribuídos à razão de um milhão de endereços a cada microssegundo, serão necessários mais de vinte anos para atribuir todos os endereços possíveis.

Embora solucione os problemas de capacidade insuficiente, o tamanho grande do endereço cria um problema novo e interessante: as pessoas que mantêm interligações em rede precisam ler, dar entrada e manipular tais endereços. Obviamente, a notação binária é indefensável. Contudo, a notação decimal pontuada, usada para IPv4, também não torna tais endereços suficientemente compactos. Para compreender por que, considere um número de 128 bits, como um exemplo, expresso na notação decimal pontuada:

104.230.140.100.255.255.255.255.0.0.17.128.150.10.255.255

Para ajudar o endereço a tornar-se ligeiramente mais compacto e mais fácil de dar entrada, os projetistas do IPv6 propõem o uso de notação hexadecimal de dois pontos, na qual o valor de cada conjunto de 16 bits é representado em hexadecimal separado por dois pontos. Por

exemplo, quando o valor mostrado acima em notação decimal pontuada tiver sido convertido em notação hexadecimal de dois pontos e impresso usando o mesmo espaçamento, ele se tornará:

68E6:8C64:FFFF:FFFF:0:1180:96A:FFF

A notação hexadecimal de dois pontos tem a vantagem óbvia de requerer menos dígitos e menos caracteres separadores do que o decimal pontuada. Além disso, a notação hexadecimal de dois pontos inclui duas técnicas que a tornam extremamente útil. Primeiro, a notação hexadecimal de dois pontos permite a compressão de zero, em que um string de zeros repetidos é substituído por um par de dois pontos. Para assegurar que a compressão de zero produz uma interpretação não-ambígua, a proposta determina que ela pode ser aplicada apenas uma vez em qualquer endereço. A compressão de zero é especialmente útil quando usada com o esquema de atribuição de endereço proposto, pois muitos endereços vão conter strings de zero contíguos. Segundo, a notação hexadecimal de dois pontos incorpora sufixos de notação hexadecimal pontuada. Veremos que tais combinações destinam-se ao uso durante a transição do IPv4 para IPv6. Por exemplo, o string a seguir é uma notação hexadecimal válida, de dois pontos:

0:0:0:0:0:0:128.10.2.1

Observe que, embora os números separados cada um por dois pontos especifiquem o valor de uma quantidade de 16 bits, cada número da parte de notação hexadecimal pontuada especifica o valor de um octeto. Naturalmente, a compressão de zero pode ser usada com o número acima a fim de produzir um string equivalente de notação hexadecimal de dois pontos que parece ser totalmente semelhante a um endereço IPv4:

::128.10.2.1

9.3. TRÊS TIPOS BÁSICOS DE ENDEREÇO DO IPv6

Como o IPv4, o IPv6 associa um endereço a uma conexão de rede específica, não a um computador específico. Assim, atribuições de endereço são semelhantes a IPv4: um roteador IPv6 tem dois ou mais endereços, e um host IPv6 com uma conexão de rede precisa de apenas um endereço. O IPv6 também retém (e estende) a hierarquia de endereço de IPv4 em que um prefixo é atribuído a uma rede física. Entretanto, para facilitar a atribuição e a modificação de endereço, o IPv6 permite que vários prefixos sejam atribuídos a determinada rede e permite que um computador tenha vários endereços simultâneos atribuídos a determinada interface.

Além de permitir vários endereços simultâneos por conexão de rede, o IPv6 expande e, em alguns casos, unifica endereços especiais do IPv4. Geralmente, um endereço de destino de um datagrama situa-se em uma das três categorias:

Unicast O endereço de destino especifica um único computador (host ou roteador); o datagrama deverá ser roteado para o destino ao longo do caminho mais curto possível.

Cluster O destino é um conjunto de computadores que juntos dividem um único prefixo de endereço (ex.: vinculam-se à mesma rede física). O datagrama deverá ser roteado para o grupo ao longo de um caminho o mais curto possível e, então, entregue a exatamente um membro do grupo (ex.: o membro mais próximo).

Multicast O destino é um conjunto de computadores, possivelmente em diversos locais. Uma cópia do datagrama será entregue a cada membro do grupo usando hardware multicast ou broadcast, conforme o caso.

Um endereço IPv6 tem um comprimento de 128 bits, tornando o espaço de endereço tão longo que cada pessoa do planeta poderia ter uma interligação em redes tão grande quanto a atual Internet. O IPv6 divide os endereços em tipos, do mesmo modo que o IPv4 os divide em

classes. Um prefixo de endereço determina o local e a interpretação dos campos restantes do endereço. Muitos endereços de IPv6 serão atribuídos por provedores de serviços de rede autorizados. Esses endereços têm campos eu contêm uma ID de provedor, uma ID de assinante, uma ID de sub-rede e uma ID de nó.

10. BIBLIOGRAFIA

- Sociedade Brasileira para Interconexão de Sistemas Abertos (BRISA), ARQUITETURA DE REDES DE COMPUTADORES OSI e TCP/IP, Makron Books, 1994.

- Alves, Luiz, COMUNICAÇÃO DE DADOS, Makron Books, 1994.

- Comer, Douglas E., INTERLIGAÇÃO EM REDES COM TCP/IP, Editora Campus, 1998.

- Tanenbaum, Andrew S., COMPUTERS NETWORKS, Third Edition, Prentice Hall PTR, 1996.

- Huitema, Christian, ROUTING IN THE INTERNET, Prentice Hall PTR, 1995.

- O'Sullivan, Bryan, THE INTERNET MULTICAST BACKBONE, February 1996. http://www.serpentine.com/~bos/tech/mbone

- Kumar, V., MBone: INTERACTIVE MULTIMEDIA ON THE INTERNET, Indianapolis, IN: New Riders, 1996.

- Deering, S., HOST EXTENSIONS FOR IP MULTICASTING, August 1989. http://ds.internic.net/rfc/rfc1112.txt

- Schulzinne, H., Casner, S., Frederick, R., Jacobson, V.,RTP: A TRANSPORT PROTOCOL FOR REAL TIME APPLICATIONS, January 1996. http://ds.internic.net/rfc/rfc1889.txt