

## **1.Camada 1 OSI - Física ⇔ Camada de Acesso à rede TCP/IP**

A camada física do modelo OSI fornece os meios para transportar os bits que formam um quadro da camada de enlace de dados no meio físico de rede. Essa camada aceita um quadro completo da camada de enlace de dados e o codifica como uma série de sinais que são transmitidos à mídia local. Os bits codificados que formam um quadro são recebidos por um dispositivo final ou por um dispositivo intermediário.

**Ela define se a transmissão pode ser ou não realizada nos dois sentidos simultaneamente e transporta sinais para todas as camadas superiores.** Esta camada também define especificações elétricas e físicas dos dispositivos, e a relação entre um dispositivo e um meio de transmissão, tal como um cabo de cobre ou um cabo de fibra óptica, isso inclui o layout de pinos, tensões, impedância da linha, especificações do cabo, temporizador, hubs, repetidores, adaptador de rede, adaptadores de barramento de host e entre outros. Os padrões da camada física abordam três áreas funcionais:

### **1.1 Componentes físicos**

Os componentes físicos são os dispositivos de hardware eletrônico, mídia e outros conectores que transmitem os sinais que representam os bits. NICs, interfaces e conectores, materiais de cabo e projetos de cabo são especificados nos padrões associados à camada física.

### **1.2 Codificação**

A codificação de linha é um método para converter um fluxo de bits de dados em um “código” predefinido. Estes códigos são agrupamentos de bits usados para fornecer um padrão previsível, reconhecido tanto pelo transmissor quanto pelo receptor.

### **1.3 Sinalização**

A camada física deve gerar sinais elétricos, ópticos ou sem fio que representem os valores binários no meio físico. A maneira como os bits são representados é chamada de método de sinalização.

## **2.Camada 2 OSI - Enlace de dados ⇔ Camada de acesso à rede TCP/IP**

A camada 2 OSI, mais conhecida como camada de enlace de dados ou link de dados, prepara os dados da rede para a rede física. Ela é responsável pelas comunicações NIC para NIC dentro da mesma rede e ela faz o seguinte.

- Permite que as camadas superiores acessem a mídia
- Aceita dados e os encapsula em quadros da camada 2
- Controla como os dados são colocados e recebidos na mídia

- Troca quadros entre pontos de extremidade através da mídia de rede
- Recebe dados encapsulados e os direciona para o protocolo da camada superior apropriado
- Executa a detecção de erros e rejeita qualquer quadro corrompido

A camada de enlace de dados pode ser projetada de modo a oferecer diversos serviços. Três possibilidades oferecidas com frequência são:

- Serviço sem conexão e sem confirmação
- Serviço sem conexão com confirmação
- Serviço orientado a conexão com confirmação

### **2.1 Serviço sem conexão e confirmação**

O serviço sem conexão e sem confirmação consiste em fazer a máquina de origem enviar quadros independentes à máquina de destino, sem que a máquina de destino confirme o recebimento desses quadros. Nenhuma conexão lógica é estabelecida antes ou liberada depois do processo. Se um quadro for perdido devido a ruídos na linha, não haverá nenhuma tentativa de detectar a perda ou de recuperá-lo na camada de enlace de dados.

### **2.2 Serviços sem conexão com confirmação**

O próximo passo em termos de confiabilidade é o serviço sem conexão com confirmação. Quando esse serviço é oferecido, ainda não há conexões lógicas sendo usadas, mas cada quadro enviado é individualmente confirmado. Dessa forma, o transmissor sabe se um quadro chegou corretamente ou não. Caso não tenha chegado dentro de um intervalo de tempo específico, o quadro poderá ser enviado outra vez. Esse serviço é útil em canais não confiáveis, como os sistemas sem fio.

### **2.3 Serviços orientados a conexão com confirmação**

O serviço mais sofisticado que a camada de enlace de dados é capaz de oferecer à camada de rede é o serviço orientado a conexões. Com ele, as máquinas de origem e destino estabelecem uma conexão antes dos dados serem transferidos. Cada quadro enviado pela conexão é numerado, e a camada de enlace de dados garante que cada quadro será de fato recebido. Além disso, essa camada garante que todos os quadros serão recebidos uma única vez e na ordem correta.

### **2.4 Enquadramento**

Para oferecer serviços à camada de rede, a camada de enlace de dados deve usar o serviço fornecido a ela pela camada física. O que a camada física faz é aceitar um fluxo de bits brutos e tentar entregá-lo ao destino. A estratégia adotada pela camada de enlace de dados é dividir o fluxo de bits em quadros e calcular o total de verificação

(*checksum*) em relação a cada quadro. Quando um quadro chega ao seu destino, o total de verificação é recalculado. Se o total de verificação recém-calculada for diferente do que está contido no quadro, a camada de enlace de dados saberá que houve um erro e tomará providências para lidar com ele.

Uma forma de obter esse enquadramento é inserir intervalos de tempo entre os quadros, de modo muito semelhante aos espaços entre as palavras de um texto comum. No entanto, as redes raramente oferecem qualquer garantia em relação à temporização.

Como é arriscado contar com a temporização para marcar o início e o fim de cada quadro, outros métodos foram criados:

- Contagem de caracteres
- Byte flags
- Flags iniciais
- Flags finais
- Violação de codificação da camada física

## **2.5 Contagem de caracteres**

Este utiliza um campo no cabeçalho para especificar o número de caracteres do quadro. Quando vê a contagem de caracteres, a camada de enlace de dados de destino sabe quantos caracteres devem vir em seguida e, conseqüentemente, onde está o fim do quadro. O problema com esse algoritmo é que a contagem pode ser adulterada por um erro de transmissão.

## **2.6 Byte flags**

Este método de enquadramento contorna o problema de ressincronização após um erro, fazendo cada quadro começar e terminar com bytes especiais. No passado, os bytes iniciais e finais eram diferentes, mas nos últimos anos, a maioria dos protocolos têm utilizado o mesmo byte, chamado de byte de flag, como delimitador de início e de fim.

É bem possível que o padrão de bits do byte de flag ocorra nos dados. Em geral, essa situação irá interferir no enquadramento. Uma forma de solucionar esse problema é fazer com que a camada de enlace de dados do transmissor inclua um caractere de escape especial (ESC) imediatamente antes de cada byte de flag. A camada de enlace de dados da extremidade receptora remove o byte de escape antes de entregar os dados à camada de rede. Essa técnica é chamada de inserção de bytes ou inserção de caracteres. Desse modo, é possível distinguir um byte de flag de enquadramento de um byte nos dados pela ausência ou presença de um byte de escape antes dele.

Uma das principais desvantagens da utilização desse método de enquadramento é que ele depende da utilização de caracteres de 8 bits. Nem todos os códigos de caracteres utilizam caracteres de 8 bits.

## **2.7 Flags Finais e Flags iniciais**

Este permite que os quadros de dados conttenham um número arbitrário de bits, e possibilita a utilização de códigos de caracteres com um número arbitrário de bits por caractere. De acordo com essa técnica, cada quadro começa e termina com um padrão de bits. Sempre que encontrar cinco valores 1 consecutivos nos dados, a camada de enlace de dados do transmissor insere um bit 0 no fluxo de bits que está sendo enviado. Essa inserção de bits é semelhante à inserção de bytes, na qual um byte de escape é inserido no fluxo de caracteres enviados antes de ocorrer um byte de flag nos dados.

Ao ver cinco bits 1 consecutivos sendo recebidos, seguidos por um bit 0, o receptor remove automaticamente o bit 0. A inserção de bits, assim como a inserção de bytes, é completamente transparente para a camada de rede de ambos os computadores. Se os dados contiverem o padrão de flag 01111110, esse flag será transmitido 01111010, mas será armazenado na memória do receptor como 01111110.

## **2.8 Violação de codificação da Camada Física**

Este só se aplica a redes nas quais a decodificação no meio físico contém algum tipo de redundância. Algumas LANs codificam 1 bit de dados utilizando 2 bits físicos. Normalmente, 1 bit é um par alto-baixo, e um bit 0 é um par baixo-alto. O esquema significa que todo bit de dados tem uma transição intermediária, facilitando a localização dos limites de bits pelo receptor. As combinações alto-alto e baixo-baixo não são usadas para dados, mas são empregadas na delimitação de quadros em alguns protocolos.

Para proporcionar uma segurança ainda maior, muitos protocolos de enlace de dados utilizam uma combinação de contagem de caracteres com um dos outros métodos. Quando um quadro é recebido, o campo de contagem é utilizado para localizar o fim do quadro. O quadro só é aceito como válido se o delimitador correto estiver presente na posição adequada e se o total de verificação estiver correto.

## **2.9 As Subcamadas**

Os padrões IEEE 802 LAN/MAN são específicos para LANs Ethernet, WLANs, WPANs e outros tipos de redes locais. A camada de enlace de dados consiste nas seguintes duas subcamadas:

### **2.9.1 Local Link Control (LLC)**

Esta subcamada IEEE 802.2 comunica entre o software de rede nas camadas superiores e o hardware do dispositivo nas camadas inferiores. Ela coloca a

informação no quadro que identifica qual protocolo de camada de rede está sendo usado para o quadro. Essas informações permitem que vários protocolos usem a mesma interface de rede e mídia. Esta subcamada pega os dados do protocolo de rede, que geralmente é um pacote IPv4 ou IPv6, e adiciona informações de controle da camada 2 para ajudar a entregar o pacote ao nó de destino.

### 2.9.2 Controle de Acesso à Mídia (MAC)

Implementada no hardware, é responsável pelo encapsulamento de dados e controle de acesso à mídia e fornece endereçamento de camada de enlace de dados e é integrado com várias tecnologias da camada 1. Ela controla a NIC e outro hardware responsável pelo envio e recebimento de dados no meio LAN/MAN com ou sem fio.

A subcamada MAC é responsável e fornece encapsulamento de dados da seguinte maneira (IEEE 802.3).

- **Delimitação de quadros Ethernet:** O processo de enquadramento fornece delimitadores importantes para identificar campos dentro de um quadro. Esses bits de delimitação promovem a sincronização entre os nós de transmissão e de recepção.
- **Endereçamento Ethernet:** O quadro Ethernet incluiu um endereço MAC de origem e de destino para fornecer o quadro Ethernet da NIC Ethernet para a NIC Ethernet na mesma LAN.
- **Deteção de erros Ethernet:** Inclui um *trailer* usado para detectar erros de transmissão (FCS).

Cada ambiente de rede que os pacotes encontram à medida que eles viajam de um host local a um host remoto pode ter diferentes características. Em qualquer troca de pacotes de camada de rede, pode haver várias transições de camada de enlace de dados e de meios físicos. Em cada salto ao longo do caminho, um roteador executa as seguintes funções de camada 2: Aceita um quadro, desencapsula o quadro, encapsula novamente o pacote em um novo quadro, encaminha o novo quadro apropriado para o meio desse segmento da rede física. Cada tipo de quadro tem três partes básicas: Cabeçalho, dados e trailer.

A camada de enlace de dados acrescenta informações na forma de um trailer no final do quadro. Todos os protocolos da camada de dados encapsulam os dados dentro do campo de dados do quadro, porém, a estrutura do quadro e os campos contidos no cabeçalho e trailer variam de acordo com o protocolo. Em um ambiente frágil são necessários mais controles para garantir a entrega. Os campos de cabeçalho e de trailer aumentam à medida que mais informações de controle são necessárias.

### 2.10 Enquadramento

O enquadramento quebra o fluxo em agrupamentos decifráveis, com a informação de controle inserida no cabeçalho e trailer, com valores em diferentes campos. Nem todos os protocolos incluem esses campos. Os padrões para um protocolo de enlace de dados específico definem o formato real do quadro.

Os campos de quadro incluem o seguinte.

- **Sinalizadores de início e fim de quadro:** Usado para identificar os limites de início e fim do quadro.
- **Endereçamento:** Indica os nós de origem e destino na mídia
- **Tipo:** Identifica o protocolo da camada 3 no campo de dados
- **Controle:** Identifica serviços especiais de controle de fluxo, como QoS.
- **Dados:** Contém a carga útil do quadro (cabeçalho do pacote, cabeçalho do segmento e os dados)
- **Deteção de erro:** Incluído após os dados para formar o trailer

A forma mais comum de garantir uma entrega confiável é dar ao transmissor algum tipo de feedback sobre o que está acontecendo no outro extremo da linha. Normalmente, o protocolo solicita que o receptor retorne quadros de controle especiais com confirmações positivas ou negativas sobre os quadros recebidos. Se receber uma confirmação positiva sobre um quadro, o transmissor saberá que o quadro chegou em segurança ao destino. Por outro lado, uma confirmação negativa significa que algo saiu errado e que o quadro deve ser retransmitido.

Em um processo de detecção de erros, o *trailer* determina se o quadro chegou sem erros. Ele coloca um resumo lógico ou matemático dos bits que compõem o quadro no trailer. A camada de enlace de dados adiciona detecção de erros porque os sinais na mídia podem estar sujeitos a interferência, distorções ou perdas que alternaria substancialmente os valores de bits que esses sinais representam.

Um nó de transmissão cria um resumo lógico dos conteúdos do quadro, também conhecido como valor de verificação de redundância cíclica. Este valor é colocado no campo de sequência de verificação de quadro (**FCS**) para exibição ou conteúdo do quadro. No trailer Ethernet, o FCS fornece um método para o nó de recebimento determinar se o quadro apresentou erros de transmissão.

A camada de enlace provê o endereçamento usado no transporte de quadro através de uma mídia local compartilhada. Os endereços de dispositivos nesta camada são chamados de endereços físicos. O endereçamento da camada de enlace de dados está contido no cabeçalho do quadro e especifica o nó destino no quadro local. O cabeçalho do quadro também pode conter o endereço de origem do quadro.

O endereço físico não indica em qual rede o dispositivo está localizado, em vez disso, o endereço físico é um endereço exclusivo do dispositivo específico. Um dispositivo ainda funcionará com o mesmo endereço físico da camada 2, mesmo que o dispositivo se mova para outra rede ou sub-rede.

Conforme o pacote IP viaja do host para o roteador, de roteador para roteador e de roteador para host, em cada ponto ao longo do caminho, o pacote IP é encapsulado em um novo quadro de enlace de dados. Cada quadro de enlace contém o endereço de enlace de dados da NIC de origem e destino.

Se a confirmação ou o quadro se perder, o timer será desativado, alternando o transmissor para um problema potencial. A solução é simplesmente transmitir o quadro outra vez. Entretanto, quando os quadros são transmitidos várias vezes, existe o perigo do receptor aceitar o mesmo quadro duas ou mais vezes e de repassá-lo à camada de rede mais de uma vez. Para impedir que isso aconteça, geralmente é necessário atribuir números de sequência aos quadros enviados, para que o receptor possa distinguir as retransmissões dos quadros originais.

### **2.11 Protocolo HDLC (High Level Data Link Control) da Camada de Enlace**

Os protocolos se baseiam nos mesmos princípios, todos são orientados a bits, e todos utilizam a técnica de inserção de bits para transferência de dados. Todos os protocolos orientados a bits utilizam a estrutura de quadro.

**O campo Endereço** é importante principalmente nas linhas com vários terminais, onde ele é utilizado para identificar um dos terminais. No caso de linhas ponto-a-ponto, às vezes esse campo é utilizado para fazer distinção entre comandos e respostas.

**O campo Controle** é usado para números de sequência, confirmações e outras finalidades.

**O campo Dados** pode conter qualquer informação. Ele pode ser arbitrariamente longo, embora a eficiência do total de verificação diminua com o aumento do comprimento do quadro, devido à maior probabilidade de ocorrerem vários erros em rajada.

**O campo Total** de verificação é uma variação do código de redundância cíclica.

O quadro é delimitado por outra sequência de flag (01111110). Nas linhas ponto-a-ponto ociosas, as sequências de flags são transmitidas de forma contínua. O quadro mínimo contém três campos e totaliza 32 bits, excluindo os flags de cada extremidade. Existem 3 tipos de quadros: Quadro de informação, Quadro supervisor e Quadro não numerado.

### **2.12 A Camada de Enlace de dados na Internet**

A Internet consiste em máquinas individuais (hosts e roteadores) e na infraestrutura de comunicação que as conecta. Dentro de um único prédio, as LANs são bastante utilizadas para interconexões, mas grande parte da infraestrutura geograficamente distribuída é construída a partir de linhas privadas ponto-a-ponto.

Na prática, a comunicação ponto-a-ponto é utilizada principalmente em duas situações. Na primeira delas, milhares de organizações têm uma LAN ou mais, cada uma com um determinado número de hosts e um roteador. Com frequência, os roteadores são interconectados por uma LAN de backbone.

A segunda situação em que as linhas ponto-a-ponto executam uma função importante na internet diz respeito aos milhões de indivíduos que estabelecem conexões domésticas com a Internet utilizando modems e linhas telefônicas com acesso por discagem. O PC doméstico do usuário estabelece uma conexão com o roteador de um provedor de serviços da Internet, e depois atua como um host da Internet completo.

### **2.13 Point-to-Point Protocol (PPP)**

O PPP trata da detecção de erros, aceita vários protocolos, permite que endereços IP sejam negociados em tempo de conexão, permite a autenticação e inclui outras características. O PPP dispõe de 3 recursos:

- Um método de enquadramento que delinea de forma não ambígua o fim de um quadro e o início do quadro seguinte. O formato do quadro também lida com a detecção de erros.
- Um protocolo de controle de enlace usado para ativar linhas, testá-las, negociar opções e desativá-las novamente quando não forem mais necessárias. Esse protocolo é chamado LCP (Link Control Protocol)
- Uma maneira de negociar as opções da camada de rede de modo independente do protocolo da camada de rede a ser utilizado. O método escolhido deve ter um NCP (Network Control Protocol)

Primeiro, o PC chama o roteador do provedor por meio de um modem. Depois que o modem do roteador atende ao telefone e estabelece uma conexão física, o PC envia ao roteador uma série de pacotes LCP no campo de carga útil de um ou mais quadros PPP. Esses pacotes e suas respostas selecionam os parâmetros PPP a serem utilizados. Quando todos esses parâmetros estão corretamente definidos de comum acordo, uma série de pacotes NCP é enviada para configurar a camada de rede. O PC quer executar uma pilha de protocolos TCP/IP, e assim necessita de um endereço IP. Como não há endereços IP suficientes, normalmente cada provedor da Internet obtém um bloco de endereços e, em seguida, atribui dinamicamente um endereço a cada PC recém-conectado durante sua sessão de login. O NCP para o IP atribui os endereços IP.



**A principal diferença entre PPP e HDLC é que o primeiro é orientado a caracteres, e não bits.** Especificamente, o PPP utiliza a técnica de inserção de bytes em linhas de discagem por modem, logo, todos os quadros representam um número inteiro de bytes. Não é possível enviar um quadro formado por 30,25 bytes, como ocorre com HDLC. Os quadros PPP não só podem ser enviados por linhas telefônicas de acesso por discagem, mas também podem ser enviados por linhas SONET ou por verdadeiras linhas HDLC orientadas a bits.

Todos os quadros PPP começam pelo byte de flag padrão do HDLC (01111110), que é complementado por inserção de bytes se ocorrer dentro do campo de carga útil (*payload*). Em seguida, temos o campo **Endereço**, que sempre é definido como o valor binário 11111111, indicando que todas as estações devem aceitar o quadro. A utilização desse valor evita o problema da necessidade de atribuição de endereços de enlace de dados.

O campo **Controle** é exibido após o campo **Endereço** e seu valor padrão é 00000011. Esse valor indica um quadro não numerado. Em outras palavras, o PPP não oferece uma transmissão confiável com o uso de números de sequência e confirmações como o padrão.

O quarto campo do quadro PPP é o campo *Protocolo*. Sua tarefa é informar o tipo de pacote que se encontra no campo Carga Útil (*payload*). Os códigos são definidos para representar os protocolos LCP, NCP, IP, IPX, AppleTalk e outros.

O campo Carga Útil (*payload*) tem comprimento variável, podendo se estender até o tamanho máximo negociado. Se o comprimento não for negociado com o uso do LCP durante a configuração da linha, será empregado um comprimento padrão de 1.500 bytes.

Depois do campo Carga Útil (*payload*), temos o campo **Total de verificação**, que normalmente tem 2 bytes, embora seja possível negociar um total de verificação de 4 bytes.

## **2.13 Encapsulamento**

### **2.13.1 Host para roteador**

O host de origem encapsula o pacote IP da camada 3 em um quadro da camada 2. No cabeçalho do quadro, o host adiciona seu endereço da camada 2 como origem e o endereço da camada 2 do roteador como destino.

### **2.13.2 Roteador para roteador**

O pacote IP da camada 3 é encapsulado em um novo quadro da camada 2. No cabeçalho do quadro, o endereço de roteador de camada 2 é adicionado como origem e o endereço de camada 2 para o seguinte roteador como destino.

### **2.13.3 Roteador para host**

O roteador encapsula o pacote IP da camada 3 em um novo quadro da camada 2. No cabeçalho do quadro, o roteador adiciona seu endereço de camada 2 como origem e o endereço de camada 2 para o servidor de destino. O endereço da camada de enlace de dados é usado apenas para entrega local e não tem significado além da rede local.

### 3. Camada 3 OSI - Rede ⇔ Camada Internet TCP/IP

Esta camada fornece os meios funcionais e de procedimento de transferência de comprimento variável de dados de sequências de uma fonte de acolhimento de uma rede para um host de destino numa rede diferente. Transforma os segmentos da camada de transporte em pacotes e também pode realizar a fragmentação e remontagem e os erros de entrega de relatório. Os roteadores operam nesta camada tornando a Internet possível. Os principais protocolos de comunicação da camada de rede são IPv4 e IPv6. Outros protocolos incluem protocolos de roteamento como OSPF (*Open Shortest Path First*) e protocolos de mensagem como ICMP (*Internet Control Message Protocol*).

A camada de rede oferece serviços à camada de transporte na interface entre a camadas e rede e a camada de transporte. Uma questão importante é identificar os tipos de serviços que a camada de rede oferece à camada de transporte. Os serviços da camada de rede foram projetados tendo em vista os seguintes objetivos:

- Os serviços devem ser independentes da tecnologia de roteadores
- A camada de transporte deve ser isolada do número, do tipo e da tecnologia dos roteadores presentes

#### 3.1 Sub-redes de Circuitos Virtuais e Datagramas

São possíveis duas organizações, dependendo do tipo de serviço oferecido. Se for o serviço sem conexões, os pacotes serão injetados individualmente na sub-rede e roteadores de modo independente uns dos outros. Não será necessária nenhuma configuração antecipada. Nesse contexto, os pacotes são chamados de *datagramas*.

Se for usado o serviço orientado a conexões, antes de ser possível enviar quaisquer pacotes de dados. Essa conexão é chamada de *circuito virtual* em analogia com os circuitos físicos estabelecidos pelo sistema telefônico, e a sub-rede é denominada sub-rede de circuitos virtuais. O algoritmo que gerencia os *datagramas* e toma as decisões de roteamento é chamado de *algoritmo de roteamento*.

Os circuitos virtuais permitem que os pacotes contêm números de circuitos em vez de endereços de destino completos. Se os pacotes tenderem a ser muito pequenos, um endereço de destino completo em cada pacote poderá representar um volume significativo de *overhead* e, portanto, haverá desperdício de largura de banda.

Outro compromisso é o que se dá entre o tempo de configuração e o tempo de análise de endereço. O uso de circuitos virtuais requer uma fase de configuração, o que leva tempo e consome recursos. Porém, é fácil descobrir o que fazer com um pacote de dados que em uma sub-rede de circuitos virtuais: o roteador só utiliza o número do circuito para criar um índice em uma tabela e descobrir para onde vai o pacote. Em uma sub-rede de datagramas, é necessário um procedimento de pesquisa mais complicado para localizar a entrada correspondente ao destino.

Questão	Sub-rede de datagramas	Sub-rede de circuitos virtuais
<b>Configuração de circuitos</b>	Desnecessária	Obrigatória
<b>Endereçamento</b>	Cada pacote contém os endereços de origem e de destino completos	Cada pacote contém um número de circuito virtual curto
<b>Informações sobre o estado</b>	Os roteadores não armazenam informações sobre o estado das conexões	Cada circuito virtual requer espaço em tabelas de roteadores por conexão
<b>Roteamento</b>	Cada pacote é roteado independentemente	A rota é escolhida quando o circuito virtual é estabelecido
<b>Efeito de falhas no roteador</b>	nenhum, com exceção dos pacotes perdidos durante a falha	Todos os circuitos virtuais que tiverem passado pelo roteador que apresentou o defeito serão encerrados
<b>QoS</b>	Difícil	Fácil, se possível alocar recursos o suficiente
<b>Controle de congestionamento</b>	Difícil	Fácil, se possível alocar recursos o suficiente

**Os circuitos virtuais têm algumas vantagens na garantia de qualidade de serviço e ao evitarem o congestionamento dentro da sub-rede**, pois os recursos podem ser reservados antecipadamente, quando a conexão é estabelecida. Quando os pacotes começam a chegar, a largura de banda e a capacidade do roteador necessárias já estarão instaladas. No caso de sistemas de processamento de transações, o *overhead* necessário para configurar e limpar um circuito virtual pode reduzir facilmente o uso do circuito. Caso se espere que a maior parte do tráfego seja desse tipo, o uso de circuitos virtuais comutados dentro da sub-rede fará pouco sentido.

**Os circuitos virtuais também têm um problema de vulnerabilidade**. Se um roteador apresentar uma falha e perder sua memória, mesmo que volte um segundo depois, todos os circuitos virtuais que estiverem passando por ele terão de ser

interrompidos. Por outro lado, se um roteador de datagramas ficar fora do ar, somente os usuários cujos pacotes estiverem enfileirados no roteador naquele momento serão afetados, e talvez nem todos eles, dependendo do fato de já terem sido confirmados.

Para realizar comunicações de ponta a ponta através dos limites da rede, os protocolos de camada de rede executam quatro operações básicas:

### **3.1.1 Endereçamento de dispositivos finais**

Os dispositivos finais devem ser configurados com um endereço IP exclusivo para identificação na rede.

### **3.1.2 Encapsulamento**

A camada de rede encapsula a unidade de dados de protocolo (PDU) da camada de transporte em um pacote. O processo de encapsulamento adiciona informações de cabeçalho IP, como os endereços IP dos hosts origem e destino. É executado pela origem do pacote IP.

### **3.1.3 Roteamento**

A camada de rede fornece serviços para direcionar os pacotes para um host de destino em outra rede. Para trafegar para outras redes, o pacote deve ser processados por um roteador. A função do roteador é escolher o melhor caminho e direcionar os pacotes para o host de destino em um processo conhecido como roteamento. Um pacote pode atravessar muitos roteadores antes de chegar ao seu host de destino. Cada roteador que um pacote atravessa é chamado de salto.

### **3.1.4 Algoritmos de roteamento**

O algoritmo de roteamento é a parte do software da camada de rede responsável pela decisão sobre a linha de saída a ser usada na transmissão do pacote de entrada. Se a sub-rede utilizar circuitos virtuais internamente, as decisões de roteamento serão tomadas somente quando um novo circuito virtual estiver sendo estabelecido. Daí em diante, os pacotes de dados seguirão a rota previamente estabelecida.

Mesmo que as rotas sejam escolhidas independentemente para cada pacote ou apenas quando novas conexões são estabelecidas, certas propriedades são desejáveis em um algoritmo de roteamento: correção, simplicidade, robustez, estabilidade, equidade e otimização.

Os algoritmos de roteamento podem ser agrupados em duas classes principais:

- **Adaptativas:** Os algoritmos adaptativos mudam suas decisões de roteamento para refletir mudanças na topologia e, normalmente, também no tráfego. Os algoritmos adaptativos diferem em termos do lugar em que obtêm suas informações, do momento em que alteram as rotas e da unidade métrica, utilizada para a otimização.

- **Não adaptativas:** Os algoritmos adaptativos não baseiam suas decisões de roteamento em medidas ou estimativas de tráfego e da topologia atual. Em vez disso, a escolha da rota a ser utilizada para ir de um lugar até o outro é previamente calculada off-line, sendo transferida para os roteadores quando a rede é inicializada.

### 3.2 O princípio da otimização e algoritmos de roteamento pelo caminho mais curto

É possível criar uma descrição geral das rotas ótimas sem levar em conta a topologia ou o tráfego de rede. Essa descrição é conhecida como ***princípio de otimização***. Esse princípio estabelece que, se o roteador A estiver no caminho ótimo entre o roteador B e o roteador C, o caminho ótimo entre A e C também estará na mesma rota. Como consequência direta desse princípio, é possível observar que o conjunto de rotas ótimas de todas as origens para um determinado destino forma uma árvore com raiz no destino. Uma árvore dessas é conhecida como ***árvore de escoamento***.

São conhecidos diversos algoritmos para se calcular o caminho mais curto entre dois nós de um grafo. O algoritmo de Dijkstra é um exemplo. Cada nó é identificado por sua distância a partir do nó de origem ao longo do melhor caminho conhecido. Inicialmente, nenhum caminho é conhecido, portanto, todos os nós são rotulados com infinito. À medida que o algoritmo prossegue e os caminhos são encontrados, os rótulos podem mudar, refletindo melhores caminhos. Um rótulo pode ser provisório ou permanente. Quando se descobre que um rótulo representa o caminho mais curto possível até a origem desse nó, ele se torna permanente e nunca mais é alterado daí em diante.

#### 3.2.1 Roteamento por *flooding*

Outro algoritmo estático é o algoritmo de inundação (***flooding***), no qual cada pacote de entrada é enviado para toda a linha de saída, exceto para aquele em que chegou. Este algoritmo gera uma vasta quantidade de pacotes duplicados. É preciso ter um controlador de ***hops*** contido no topo do cabeçalho de cada pacote.

O algoritmo de inundação não é prático na maioria das aplicações, mas tem utilidade como em aplicações militares, em que muitos roteadores podem ser destruídos a qualquer momento, a grande robustez do algoritmo de inundação é altamente desejável. Em aplicações de bancos de dados distribuídos também. O algoritmo de inundação sempre escolhe o caminho mais curto, pois todos os caminhos possíveis são selecionados em paralelo.

#### 3.2.2 Roteamento com vetor de distância

Os algoritmos de roteamento com vetor de distância operam fazendo cada roteador manter uma tabela que fornece a melhor distância conhecida até cada destino

e determina qual linha deve ser utilizada para chegar lá. Essas tabelas são atualizadas através da troca de informações com os vizinhos. Também pode receber outro nome, sendo o mais comum o algoritmo de roteamento distribuído de Bellman-Ford.

O roteamento com vetor de distância funciona na teoria, mas tem um sério inconveniente na prática: apesar de convergir para a resposta correta, ele pode fazê-lo muito lentamente. Em particular, ele reage com rapidez a boas notícias, mas reage devagar a más notícias.

### **3.2.3 Roteamento hierárquico**

Em um determinado momento, uma rede pode crescer até o ponto em que deixará de ser viável cada roteador ter uma entrada correspondente a cada outro roteador, então o roteamento terá de ser feito de forma hierárquica, como na rede telefônica. Quando o roteamento hierárquico for utilizado, os roteadores serão divididos naquilo que se denomina regiões, com cada roteador conhecendo todos os detalhes sobre como rotear pacotes para destinos dentro de sua própria região, mas sem conhecer nada sobre a estrutura interna de outras regiões.

### **3.2.4 Roteamento por difusão**

Em algumas aplicações, os hosts precisam enviar mensagens a muitos outros hosts. Um método de difusão que não exige recursos especiais da sub-rede permite à origem simplesmente enviar um pacote específico a cada destino. O método não só desperdiça largura de banda, como também exige que a origem tenha uma lista completa de todos os destinos. O algoritmo de inundação é um candidato. Outro algoritmo é o roteamento para vários destinos, onde se este método for utilizado, cada pacote conterá uma lista de destinos ou um mapa de bits indicando os destinos desejados. Uma árvore de amplitude é um subconjunto da sub-rede que inclui todos os roteadores, mas não contém nenhum loop. Este seria também outro algoritmo a ser utilizado.

A ideia chamada encaminhamento pelo caminho inverso é simples, depois que um pacote de difusão chega a um roteador, o roteador verifica se o pacote chegou pela linha que normalmente é utilizada para o envio de pacotes à origem da difusão. Em caso afirmativo, há uma excelente possibilidade de que o pacote de difusão tenha seguido a melhor rota a partir do roteador e seja, portanto, a primeira cópia a chegar ao roteador. A principal vantagem do encaminhamento pelo caminho inverso é que ele é ao mesmo tempo razoavelmente eficiente e é fácil de implementar, além de não exigir que os roteadores saibam nada sobre árvores de amplitude, nem têm o overhead de uma lista de destino ou um mapa de bits em cada pacote de difusão.

### **3.2.5 Roteamento por multidifusão**

A multidifusão exige o gerenciamento de grupos. É preciso usar algum método para criar e destruir grupos, e para permitir que os processos entrem e saiam de grupos.

O modo como essas tarefas serão realizadas não interessa ao algoritmo de roteamento. O que interessa é que, quando um processo se associar a um grupo, ele informará seu host desse fato. É importante que os roteadores saibam quais de seus hosts pertencem a cada um dos grupos. Os hosts devem informar seus roteadores sobre alterações na associação a grupos, ou então os roteadores terão de consultar seus hosts periodicamente.

Existem vários métodos que podem ser usados para podar a árvore de amplitude. O mais simples pode ser usado se o roteamento por estado de enlace for empregado e cada roteador estiver ciente da topologia completa, inclusive de quais hosts pertencem a cada um dos grupos. Uma desvantagem potencial desse algoritmo é que ele é mal dimensionado para redes grandes. Um projeto alternativo utiliza árvores baseadas no núcleo.

### **3.2.6 Roteamento por Inundação**

O contador é decrementado em cada *hop*, com o pacote sendo descartado quando o contador atingir zero. O ideal é que o contador de *hops* seja inicializado com o comprimento do caminho desde a origem até o destino. Se não souber o tamanho do caminho, o transmissor poderá inicializar o contador com o valor referente ao pior caso, ou seja, o diâmetro total da sub-rede.

Uma técnica alternativa para conter o processo de inundação é controlar quais pacotes foram transmitidos por inundação, a fim de evitar transmiti-los uma segunda vez. Uma forma de conseguir isso é fazer o roteador de origem inserir um número de sequência em cada pacote recebido de seus hosts, isto é, uma lista por roteador de origem. Para evitar que as listas cresçam indefinidamente, cada lista deve ser incrementada de acordo com um contador  $k$ , o que significa que todos os números de sequência até  $k$  foram vistos.

### **3.2.7 Roteamento em redes Ad Hoc**

Um caso ainda mais extremo é aquele em que os próprios roteadores são móveis. Entre as possibilidades estão:

- Veículos militares em um campo de batalha sem qualquer infraestrutura
- Uma frota de navios no mar
- Trabalhos de emergência em calamidades de destruição de infraestrutura
- Um grupo de pessoas com notebooks em uma área que não tem instalações

802.11

Em todos esses casos e em outros, cada nó consiste em um roteador e um host, em geral, no mesmo computador. O que torna as redes ad hoc diferentes das redes fisicamente conectadas é que todas as regras habituais a respeito de topologias fixas,

vizinhos fixos e conhecidos, relacionamento fixo entre endereço IP e localização e outras são repentinamente abandonadas.

Os roteadores podem ir e vir, ou aparecer em novos lugares de um momento para outro. Com uma rede fisicamente conectada, se um roteador tiver um caminho válido para algum destino, esse caminho continuará a ser válido indefinidamente. Em qualquer instante, uma rede ad hoc pode ser descrita por um grafo dos nós roteadores + hosts.

### **3.3 Roteamento por dispositivos móveis**

Os hosts que nunca se movem são chamados de estacionários. Eles estão conectados à rede por fios de cobre ou fibra óptica. Por outro lado, é possível distinguir dois outros tipos de hosts. Os hosts migrantes são basicamente hosts estacionários que se deslocam de um local fixo para outro de tempos em tempos, mas que utiliza a rede apenas quando estão fisicamente conectados a ela. Os hosts visitantes realmente utilizam seus computadores em trânsito e querem manter suas conexões à medida que se deslocam.

Quando um novo host entra em uma área, seja conectando-se a ela ou simplesmente percorrendo a célula, seu computador deve se registrar com o agente externo dessa área. O procedimento de registro normalmente funciona da seguinte forma:

- Periodicamente, cada agente externo transmite um pacote anunciando sua existência e seu endereço. Um host móvel recém-chegado pode aguardar uma dessas mensagens, no entanto, se nenhuma mensagem chegar rápido o suficiente, o host móvel poderá transmitir por difusão um pacote com a mensagem “há algum agente externo por aí”
- O host móvel se registra com o agente externo, fornecendo seu endereço local, o endereço atual da camada de enlace de dados e algumas informações de segurança
- O agente externo entra em contato com o agente local do host móvel e diz: “um de seus hosts está por aqui. A mensagem do agente externo para o agente local contém o endereço de rede do agente externo. A mensagem contém ainda as informações de segurança, a fim de convencer o agente local de que o host móvel realmente está lá
- O agente local examina as informações de segurança, que contém um timbre de hora, para provar que foi gerado há alguns segundos. Se tudo estiver correto, o agente local diz ao agente externo para prosseguir
- Quando o agente externo obtém a confirmação do agente local, ele cria uma entrada em suas tabelas e informa ao host móvel que agora ele está registrado



Quando um pacote é enviado a um host móvel, ele é roteado até a LAN local do host, pois é isso que o endereço informa que deve ser feito. Pacotes enviados ao host móvel em sua LAN local são interceptados pelo agente local daquela parte. Em seguida, o agente local consulta a nova localização do host móvel e encontra o endereço do agente externo que está tratando do host móvel. Depois, o agente executa duas ações. Primeiro, ele encapsula o pacote no campo de *payload* de um pacote externo e envia este último ao agente externo. Esse mecanismo é chamado de *tunneling*. Em segundo, o agente local diz ao transmissor que daí em diante ele deverá enviar pacotes ao host móvel encapsulando-os no campo de *payload* de pacotes explicitamente endereçados ao agente externo, ao invés de enviar ao host móvel.

Os esquemas diferem no modo de como conseguem lidar com pacotes endereçados a um destino e que devem ser entregues a um destino diferente. Uma opção é alterar o endereço de destino e simplesmente retransmitir o pacote modificado.

### **3.4 Desencapsulamento**

Quando o pacote chega na camada e na rede do host de destino, o host verifica o cabeçalho IP do pacote. Se o endereço IP de destino no cabeçalho corresponder ao seu próprio endereço IP, o cabeçalho será removido do pacote. Depois que o pacote é desencadeado pela camada de rede, a PDU resultante da camada 4 é transferida para o serviço apropriado na camada de transporte. O processo de desencapsulamento é executado pelo host de destino do pacote IP.

Os protocolos de comunicação (IPv4 e IPv6) especificam a estrutura de pacotes e o processamento usado para transportar os dados de um host para outro hospedeiro. A operação sem levar em consideração os dados contidos em cada pacote, permite que a camada de rede transporte pacotes para diversos tipos de comunicações entre vários hosts.

Os pacotes são sempre criados no host de origem, que deve ser capaz de direcionar os pacotes para o host de destino. Para fazer isso, os dispositivos finais do host criam sua própria tabela de roteamento. Outra função da camada de rede é direcionar pacotes entre hosts. Um host pode enviar um pacote para:

#### **3.4.1 Itself**

Um host pode executar um ping em si mesmo enviando um pacote para um endereço IPv4 especial (127.0.0.1) ou um endereço IPv4 (::1), que é comumente referido como a interface de loopback. O ping na interface de loopback testa a pilha de protocolos TCP/IP no host.

#### **3.4.2 Local host**

Este é um host de destino que está na mesma rede local que o host de envio. Os hosts de origem e destino compartilham o mesmo endereço de rede.

### **3.4.3 Remote host**

Este é um host de destino de uma rede remota. Os hosts de origem e destino não compartilham o mesmo endereço de rede.

## **4. Camada 4 OSI - Transporte ⇔ Camada de Transporte TCP/IP**

A camada de transporte é responsável pela comunicação lógica entre aplicativos executados em hosts diferentes. A principal função desta camada é receber os dados pela camada de sessão e segmentá-los para que sejam enviadas à camada de rede proporcionando um serviço eficiente, confiável e de baixo custo. Isso pode incluir serviços como o estabelecimento de uma sessão temporária entre dois hosts e a transmissão confiável de informações para um aplicativo. As operações são feitas com controle de fluxo, ordenação dos pacotes e a correção de erros, tipicamente enviando para o transmissor uma informação de recebimento. Outra função da camada de transporte é interligar as camadas 5-7 com 1-3.

A camada de transporte não tem conhecimento do tipo de host de destino, o tipo de mídia pela qual os dados devem percorrer, o caminho percorrido pelos dados, o congestionamento em um link ou o tamanho da rede. A camada de transporte inclui dois protocolos: TCP e UDP

### **4.1 Rastreamento de conversações individuais**

Na camada de transporte, cada conjunto de dados que flui entre um aplicativo de origem e um aplicativo de destino é conhecido como conversa e é rastreado separadamente. É responsabilidade da camada de transporte manter e monitorar essas várias conversações.

A maioria das redes tem uma limitação da quantidade de dados que pode ser incluída em um único pacote. Portanto, os dados devem ser divididos em partes gerenciáveis.

### **4.2 Segmentação de dados e remontagem de segmentos**

É responsabilidade da camada de transporte dividir os dados do aplicativo em blocos de tamanho adequado. Dependendo do protocolo de camada de transporte usado, os blocos de camada de transporte são chamados de segmentos ou datagramas. A camada de transporte divide os dados em blocos menores.

### **4.3 Adicionar informações de cabeçalho**

O protocolo da camada de transporte também adiciona informações de cabeçalho contendo dados binários organizados em vários campos a cada bloco de dados. São os valores nesses campos que permitem que os vários protocolos da camada de transporte realizem diferentes funções no gerenciamento da comunicação de dados.

#### **4.4 Identificação das aplicações**

A camada de transporte deve separar e gerenciar várias comunicações com as diferentes necessidades de requisitos de transporte. Para passar fluxos de dados para os aplicativos adequados, a camada de transporte identifica o aplicativo de destino usando um identificador chamado número da porta.

#### **4.5 Multiplexação das conversas**

O envio de alguns tipos de dados através de uma rede, como um fluxo de comunicação completo, pode consumir toda a largura de banda disponível. Isso impediria que outras conversas de comunicação ocorressem ao mesmo tempo. Isso também dificultaria a recuperação de erro e retransmissão dos dados danificados.

A verificação de erros pode ser realizada nos dados do segmento, para determinar se o segmento foi alterado durante a transmissão.

Os protocolos de camada de transporte especificam como transferir mensagens entre hosts e são responsáveis pelo gerenciamento dos requisitos de confiabilidade de uma conversa. A camada de transporte inclui os protocolos TCP e UDP. Diferentes aplicações têm diferentes necessidades de confiabilidade de transporte.

### **5. Camada 5, 6 e 7 OSI - Sessão, Apresentação e Aplicação | Camada de Aplicação TCP/IP**

#### **5.1 Camada de Sessão**

As funções na camada de sessão criam e mantêm diálogos entre as aplicações origem e destino. A camada de sessão processa a troca de informações para iniciar diálogos, mantê-los ativos e reiniciar sessões interrompidas ou ociosas por um longo período. Se a rede falhar, os computadores reiniciam a transmissão dos dados a partir da última marcação recebida pelo computador receptor.

#### **5.2 Camada de Apresentação**

A camada de apresentação tem três funções principais:

- Formatar ou apresentar dados no dispositivo de origem em um formato compatível para recebimento pelo dispositivo de destino
- Comprimir dados de uma maneira que possa ser descompactada pelo dispositivo de destino
- Criptografar dados para transmissão e descriptografar dados após o recebimento.

A camada de apresentação formata dados para a camada de aplicação e define padrões para formatos de arquivo. Alguns padrões bem conhecidos para vídeo incluem Matroska Video (MKV), Motion Picture Experts Group (MPG) e QuickTime Video

(MOV). Alguns formatos conhecidos de imagem gráfica são o formato Graphics Interchange Format (GIF), o Joint Photographic Experts Group (JPG) e o formato Portable Network Graphics (PNG).

Os dados recebidos da camada 7 estão descomprimidos, e a camada 6 do dispositivo transmissor fica responsável por comprimir esses dados tornando a transmissão mais rápida, já que haverá menos dados a serem transmitidos. Para aumentar a segurança, pode-se usar algum esquema de criptografia neste nível.

### 5.3 Camada de Aplicação

Os protocolos de aplicativos TCP/IP especificam o formato e as informações de controle necessárias para muitas funções comuns de comunicação da Internet. Os protocolos da camada de aplicação são utilizados pelos dispositivos de origem e destino durante uma sessão de comunicação. Para que as comunicações sejam bem-sucedidas, os protocolos da camada de aplicativo implementados no host de origem e destino devem ser compatíveis.

- **Sistemas de nomes:** DNS
- **Configurações de host:** BOOTP e DHCP
- **E-mail:** SMTP, POP3, IMAP
- **Transferência de arquivos:** FTP, TFTP
- **Web:** HTTP e HTTPS

Um modelo em camadas para descrever operações e protocolos de rede auxilia no projeto de protocolos, fomenta a concorrência, impede alterações de tecnologia ou capacidade e fornece uma linguagem comum para descrever funções e a capacidade de rede. Existem dois modelos em camadas que são usados para descrever operações em rede: TCP/IP e OSI.

Uma comunicação poderia ser enviada através de uma rede de uma origem até destino como um fluxo maciço e ininterrupto de bits, mas isso iria criar problemas para outros dispositivos que precisassem usar os mesmos canais de comunicação. Além disso, se algum canal na infraestrutura de rede interconectada falhasse durante a transmissão, a mensagem completa seria perdida e teria que ser retransmitida. Uma melhor abordagem é dividir os dados em pedaços menores e mais gerenciáveis para o envio pela rede. A segmentação é o processo de dividir um fluxo de dados em unidades menores para transmissões através da rede.

Quando um endereço da Web ou URL é digitado em um navegador da Web, ele estabelece uma conexão com o serviço da Web. O serviço Web está em execução no servidor que está a utilizar o protocolo HTTP. URLs e URIs (*Uniform Resource Identifiers*) são os nomes que a maioria das pessoas associa aos endereços da Web.