Unidade III

Esta unidade tem como principal objetivo apresentar as camadas, sem dúvida as mais importantes, que devem ser largamente entendidas pelo profissional da área de tecnologia da informação. Isso porque essa unidade apresenta as camadas que são responsáveis pela comunicação fim a fim nas redes de computadores (Camada de Transporte) e detalha a maneira com que as mensagens são encaminhadas através das redes, em todo seu percurso, até chegar ao destino final (Camada de Rede).

5 CAMADA DE TRANSPORTE

A **Camada de Transporte**, camada central da pilha de protocolos (figura 46), desempenha o papel fundamental de fornecer serviços de comunicação diretamente aos processos de aplicação, que rodam em hospedeiros diferentes.

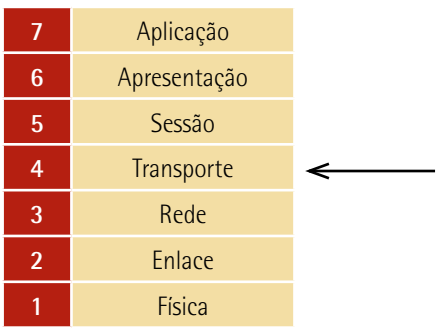

Figura 46 - Camada de Transporte do modelo OSI (KOVACH, 2009)

5.1 Serviços de transporte

Agora vamos entender o funcionamento da Camada de Transporte e sua relação com a Camada de Rede que está logo abaixo dela. Você deve se lembrar daquele exemplo que usamos comparando as redes aos serviços de correio. Vamos agora aprofundar um pouco mais.

É muito comum a confusão entre as funções pertinentes à Camada de Transporte e as pertinentes à Camada de Rede. Como falamos, a Camada de Transporte estabelece comunicação lógica entre processos em hospedeiros diferentes. Já a Camada de Rede fornece comunicação lógica entre os hospedeiros. Para entender melhor, vamos ilustrar com uma história de amor entre João e Maria.

João era um jovem loucamente apaixonado por Maria. Ele morava em Florianópolis e ela, em São Paulo. Eles se conheceram quando ela foi passar férias na casa de uma de suas tias. Durante as férias, João nunca teve coragem de convidar Maria para um cinema e ela acabou indo embora sem que ele tivesse se declarado a ela.

Depois disso, arrependido, João passou a enviar uma carta para sua amada todas as semanas (pobre João, não tinha MSN nessa nossa história). Sempre com o mesmo pedido, que Maria aceitasse se encontrar com ele. Ela nunca respondia a seus lamentos.

No edifício onde João morava, o correio passava toda sexta-feira para recolher as correspondências dos moradores e deixar outras. José, o porteiro no edifício, era o responsável por recolher as cartas dos moradores, entregá-las ao correio e por receber as cartas do carteiro e distribuí-las em cada um dos apartamentos. José era um porteiro que fazia o seu trabalho, que era distribuir as cartas aos apartamentos e entregar ao correio as cartas dos moradores.

João passou meses nesse sofrimento, mandando cartas para seu amor, e nunca recebeu qualquer resposta. Estava quase desistindo.

Foi quando surgiu Carlos na história. Carlos se chamava na verdade Teófilo Carlos Prates, mas gostava de ser chamado de Carlos mesmo. Isso aconteceu em uma das férias do José. Carlos passou a ser o responsável pela entrega e recepção das cartas no prédio de João. Mas ele era um porteiro e tanto e tomava todo o cuidado para garantir que as cartas haviam sido entregues aos seus destinos. Perguntava ao carteiro todas as vezes se tudo tinha corrido bem.

Logo ficou sabendo que as cartas de Carlos não estavam sendo entregues à sua amada. O carteiro contou que o endereço tinha um erro e que os envelopes estavam sendo descartados.

Carlos correu para avisar João, que não sabia se ficava triste ou feliz com a novidade. Enquanto estava na dúvida, correu na casa da tia da Maria e confirmou o endereço. Ele tinha anotado errado!

Imediatamente escreveu uma nova carta para Maria e recomendou que Carlos fosse o responsável por encaminhá-la ao correio.

Meu amado João,

Há meses venho esperando essa sua carta, sonhando com o dia em que poderia lhe dizer, meu amor, que essa nossa história, na verdade, não passa de uma metáfora para explicar o funcionamento dos protocolos de transporte UDP (José) e TCP (Carlos).

Maria

Agora vamos ilustrar nossa história falando como acontece no ambiente das redes de computadores.

O processo se inicia com o preparo de sua mensagem na Camada de Aplicação, que em nossa história foi representada pelas cartas de João, para ser enviada à rede através da API, que cumpre as regras de envio. Esse pacote, ainda da Camada de Aplicação, é enviado para a Camada de Transporte, representada pelos porteiros do prédio de João, José e Carlos, como mostra a figura 47:

Figura 47 - Mensagem sendo enviada à Camada de Transporte (elaborada pela autora)

A Camada de Transporte é responsável por adicionar o seu cabeçalho. Dependendo do serviço solicitado, adiciona o cabeçalho UDP ou TCP. Na nossa história, José pode ser representado pelo protocolo UDP, que não garantia a entrega das cartas, enquanto Carlos, pelo TCP, que tomava todo cuidado para garantir as entregas. Em seguida, é direcionado à Camada de Rede, representada pelo correio em nossa história e ilustrada na figura 48:

Figura 48 - O protocolo de transporte encaminhando a mensagem à Camada de Rede (elaborada pela autora)

A Camada de Rede é responsável por encaminhar os pacotes ao seu destino e lá entregá-los à Camada de Transporte. Em nossa história podemos representá-la pelo correio entregando a mensagem ao porteiro do prédio de Maria, quando o endereço não tinha mais erros. Como vimos, a Camada de Rede (ou correio) não se preocupa com as mensagens endereçadas erradas. Se ela não encontra o destino, simplesmente descarta sem nem mesmo enviar qualquer tipo de aviso ao remetente. A figura 49 mostra a mensagem chegando na Camada de Transporte do destinatário:

Figura 49 - A Camada de Rede entregando a mensagem à camada de transporte, no destino (elaborada pela autora)

No destino, a Camada de Transporte executa suas funções e remove os cabeçalhos, entregando a mensagem à Camada de Aplicação. Em nossa história é representada finalmente pelo recebimento da carta de João à Maria. A figura 50 ilustra essa entrega da Camada de Transporte no número de porta correspondente à aplicação de origem, na Camada de Aplicação de destino:

Figura 50 - A Camada de Transporte entregando a mensagem à Camada de Aplicação, no destino (elaborada pela autora)

E assim a aplicação distribuída na rede é executada no sistema final do destino.

Detalhando um pouco mais como todo o processo da Camada de Transporte se dá, na origem, ela converte as mensagens que recebe de um processo de aplicação remetente em pacotes de Camada de Transporte (que, como vimos, chamamos neste nível de segmentos). Isso é possível, pois as mensagens que vêm da Camada de Aplicação podem ser fragmentadas em pedaços menores e, adicionado a cada pedaço, um cabeçalho da Camada de Transporte para criar o segmento dessa camada.

A Camada de Transporte é também responsável pela qualidade na entrega e recebimento dos dados. No destino, depois de os dados já estarem endereçados na Camada de Rede, é hora de começar o transporte dos mesmos. A Camada de Transporte é que gerencia esse processo. Nesse

momento, vale destacar que, para assegurar, de maneira confiável, o sucesso no transporte dos dados, é considerado um serviço que atua de forma interativa chamado QoS – Qualidade de Serviço (Quality of Service).

A figura 51 traz uma visão dos sistemas finais nas duas pontas. A aplicação origem entrega a mensagem para o protocolo da Camada de Transporte, e o protocolo da Camada de Transporte entrega a mensagem para o protocolo de rede, que no destino entrega-a ao protocolo de transporte e, consequentemente, à aplicação de destino.

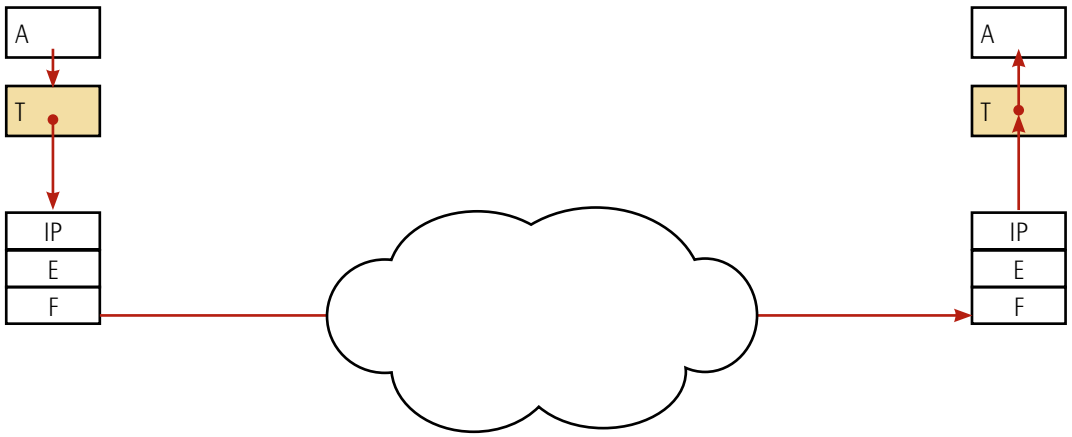

Figura 51 - As operações da Camada de Transporte (KOVACH, 2009)

Se considerarmos a rede vista pela aplicação, é como se a comunicação entre as camadas de transporte na origem e no destino fosse direta, sem passar pelas demais camadas abaixo, como está representado na figura 52:

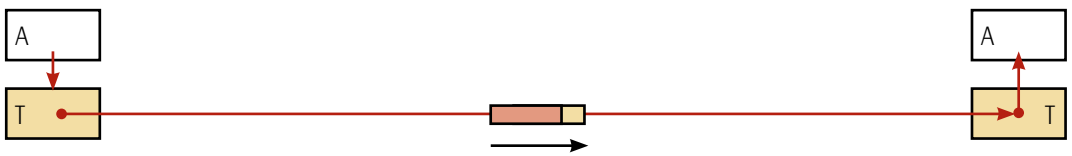

Figura 52 - A rede vista pela aplicação (KOVACH, 2009)

Na Camada de Transporte, os protocolos proveem comunicação lógica, e não física, entre processos de aplicação, executando em hospedeiros diferentes. Pode haver mais de um protocolo de Camada de Transporte disponível para aplicações de rede, mas os protocolos de transporte executam em sistemas finais, e tudo se passa como se os hospedeiros estivessem conectados diretamente, como mostra a figura 53:

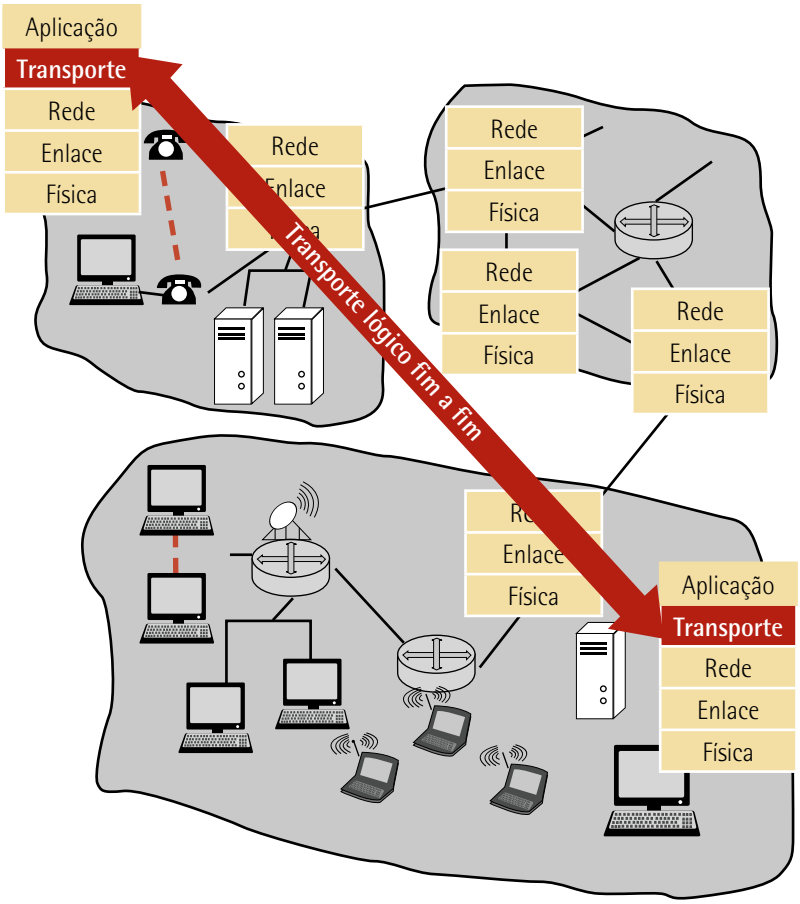

Figura 53 - Elementos que compõem os correios eletrônicos (KUROSE; ROSS, 2010)

5.1.1 A origem e o destino das mensagens

A identificação das aplicações de destino, que devem receber as mensagens na Camada de Transporte, é feita por meio de processos de aplicação, ou seja, a comunicação lógica é feita entre processos.

Como vimos anteriormente, processo é um programa que está rodando em uma determinada máquina. Os processos são identificados por meio de portas (números de 16 *bits*). Esses números são conhecidos como **portas de protocolo**. Antes de fazer uma comunicação, as aplicações devem ser identificadas por um número, isto é, devem se associar a um número de porta. Na origem, somente após se associar a um número as aplicações passam a ter condições de solicitar serviços da Camada de Transporte. O sistema operacional local fornece uma interface por meio da qual os processos podem se associar a uma porta.

A figura 54 representa as portas numeradas associadas a cada uma das aplicações correspondentes entre os níveis de aplicação e transporte, já que a aplicação entrega ao protocolo de transporte, dependendo do número de porta. Repare que a porta serve para identificar a aplicação em execução:

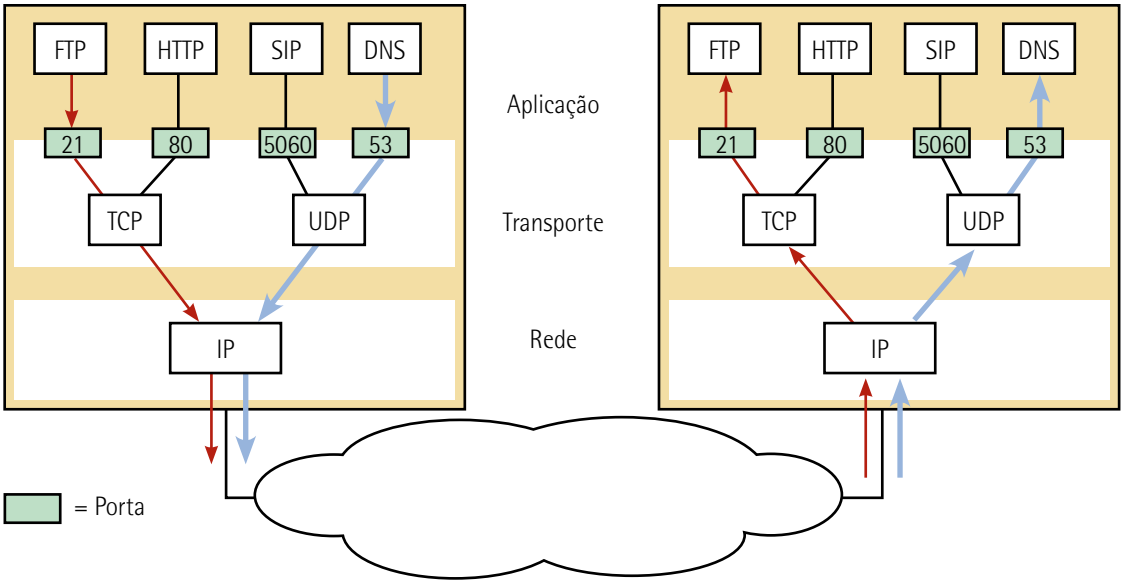

Figura 54 - Elementos que compõem os correios eletrônicos (KOVACH, 2009)

Lembrete

Lembre-se do processo de encapsulamento e desencapsulamento, que permite a identificação das aplicações no destino por meio do cabeçalho correspondente à camada.

Existem dois tipos de portas: as **estáticas** e as **dinâmicas**. Portas estáticas ou conhecidas são portas associadas a processos que fornecem serviços (programas servidores) e que, normalmente, não mudam com o tempo. Por exemplo, o servidor SMTP está sempre associado à porta 25. Portas dinâmicas são portas associadas a processos que solicitam serviços a servidores (programas clientes) e são normalmente assinaladas dinamicamente pelo sistema operacional, ou seja, mudam a cada execução do programa.

Portas com faixas de 0 a 1023 são chamadas de portas *conhecidas*, ou seja, estão associadas a uma aplicação comum, conhecida. Acima de 1023 são chamadas portas altas e é possível associá-las a uma aplicação desconhecida. Alguns exemplos de portas baixas conhecidas estão na Tabela 1:

Tabela 4 - Exemplos de portas baixas conhecidas

Protocolo	Número da porta
FTP	21/TCP
Telnet	23/TCP
SMTP	25/TCP
BOOTP	67/UDP
HFTP	69/UDP
HTTP	80/TCP
HOSTNAME	101/TCP

POP3	110/TCP
NTP	123/UDP
SNMP	161/UDP
BGP	179/TCP
IRC	194/TCP
IMAP	220/TCP, UDP

Na entrega da mensagem no destino, a Camada de Transporte é responsável por examinar os campos que contêm a identificação da porta que deverá ser a receptora da mensagem. A tarefa de entregar os dados contidos em um segmento da Camada de Transporte à porta correta é denominada **demultiplexação**. Para que a demultiplexação aconteça dessa forma, é certo que na origem as informações provenientes das diversas portas foram reunidas em um único segmento para passarem à Camada de Rede e serem transmitidas ao destino. A esse trabalho de reunir diversas informações em um único segmento denominamos **multiplexação**.

5.2 Protocolos de transporte

Cada aplicação da internet usa pelo menos um protocolo da Camada de Transporte para enviar e receber dados. São dois os principais protocolos de Camada de Transporte: TCP (Transmission Control Program) e UDP (User Datagram Protocol). Veremos que, embora tenham o mesmo objetivo, tais protocolos possuem características muito diferentes entre si, e a escolha de associação com as aplicações depende das características que se esperam da aplicação.

Tabela 5 – Aplicações populares da internet e seus protocolos de transporte

Aplicação	Protocolo de Camada de Aplicação	Protocolo de Camada de Transporte
Correio eletrônico	SMTP	TCP
Acesso a terminal remoto	Telnet	TCP
Web	HTTP	TCP
Transferência de arquivo	FTP	TCP
Recepção de multimídia	Tipicamente proprietária	UDP ou TCP
Telefonia por internet	Tipicamente proprietária	UDP ou TCP
Gerenciamento de rede	SNMP	Tipicamente UDP
Protocolo de roteamento	RIP	Tipicamente UDP
Tradução de nome	DNS	Tipicamente UDP

5.2.1 O protocolo UDP

O UDP *(*User Datagram Protocol*)* é um protocolo da Camada de Transporte que fornece um serviço de entrega rápida aos protocolos de aplicação e é considerado um protocolo de transporte da internet mínimo, "sem frescura". Ele executa o serviço de "melhor esforço", ou seja,

segmentos UDP podem ser perdidos ou entregues à aplicação fora de ordem. Não existe nenhum tipo de configuração inicial entre remetente e receptor, e os segmentos são tratados de forma independente (sem conexão).

Destaca-se por ser um protocolo **não orientado à conexão**, ou seja, eliminando o estabelecimento de conexão, torna-se rápido, mas possui transferência não confiável de dados. Além disso, não possui controle de fluxo e de congestionamento, ou seja, pode transmitir o mais rápido possível. O UDP é considerado simples, pois não se mantém o "estado" da conexão no remetente/receptor e possui cabeçalho de segmento bastante pequeno e simples, se comparado ao TCP.

O UDP é muito utilizado para aplicações de meios contínuos (voz, vídeo), que são tolerantes a perdas e sensíveis à taxa de transmissão, assim como todas as aplicações isócronas (aplicações quem precisam reproduzir-se na mesma taxa com que foram geradas). Também é utilizado nas aplicações de DNS e SNMP (protocolo de gerenciamento de rede). Nas aplicações que utilizam UDP, é comum a necessidade de transferência confiável mínima. Nesses casos, é necessário incluir a confiabilidade na Camada de Aplicação, e a recuperação de erro também fica específica à aplicação.

Segmento UDP

O cabeçalho do segmento UDP é simples, se comparado ao TCP, que será apresentado na próxima seção. Ele é composto por 4 campos essenciais, como mostra a figura 55:

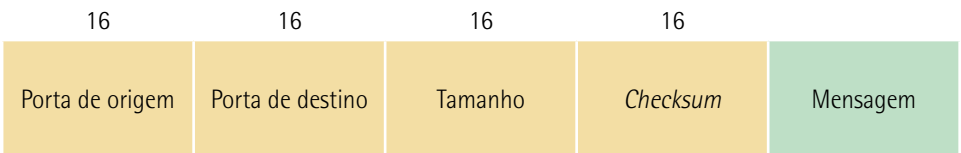

Figura 55 - Cabeçalho UDP (KOVACH, 2009)

- **Porta de origem**: identifica o número de porta relacionado com a aplicação de origem. Este campo representa a direção de resposta do destinatário. Entretanto, é um campo não obrigatório, ou seja, seu preenchimento pela aplicação de origem é opcional e, neste caso, será preenchido com zero (utilizado para mensagens unidirecionais).
- **Porta de destino**: identifica o número de porta relacionado com a aplicação de destino.
- **Tamanho da mensagem**: identifica o tamanho total do segmento UDP, incluindo-se o cabeçalho.
- ***Checksum***: campo reservado para verificação de integridade do segmento no destino.

Repare que do lado do remetente, na Camada de Transporte, o cabeçalho UDP é agrupado à mensagem, formando o segmento UDP, que é passado integralmente à Camada de Rede, que adicionará seu cabeçalho IP, como mostra a figura 56:

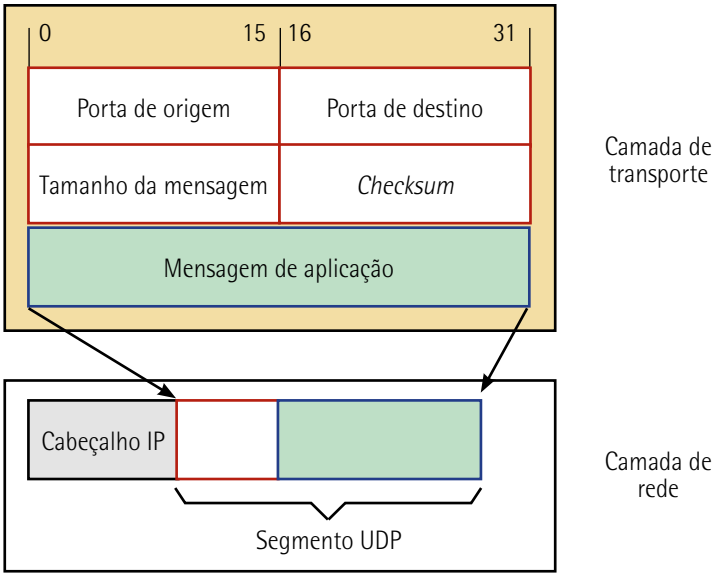

Figura 56 - Segmento UDP (KOVACH, 2009)

O segmento TCP recebe a mensagem da Camada de Aplicação, que adiciona o número da porta de destino no campo correspondente. O endereço IP de destino, apontado pela aplicação, é enviado diretamente para a Camada de Rede, pois ela cuidará de entregar ao IP de destino correto. Lembre-se de que o segmento UDP fica encapsulado no nível de rede, como mostra a figura 57:

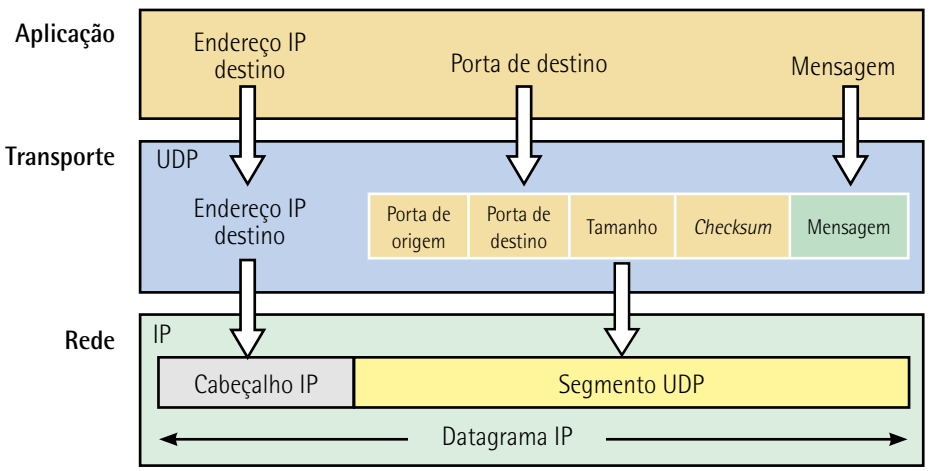

Figura 57 - Encapsulamento UDP (KOVACH, 2009)

Recordando como funciona o processo de encapsulamento nos sistemas finais, ao utilizar-se o UDP como protocolo de transporte a uma aplicação, é o cabeçalho do UDP que vai sendo passado à camada do encapsulamento, como vimos anteriormente. A figura 58 ilustra essa ação:

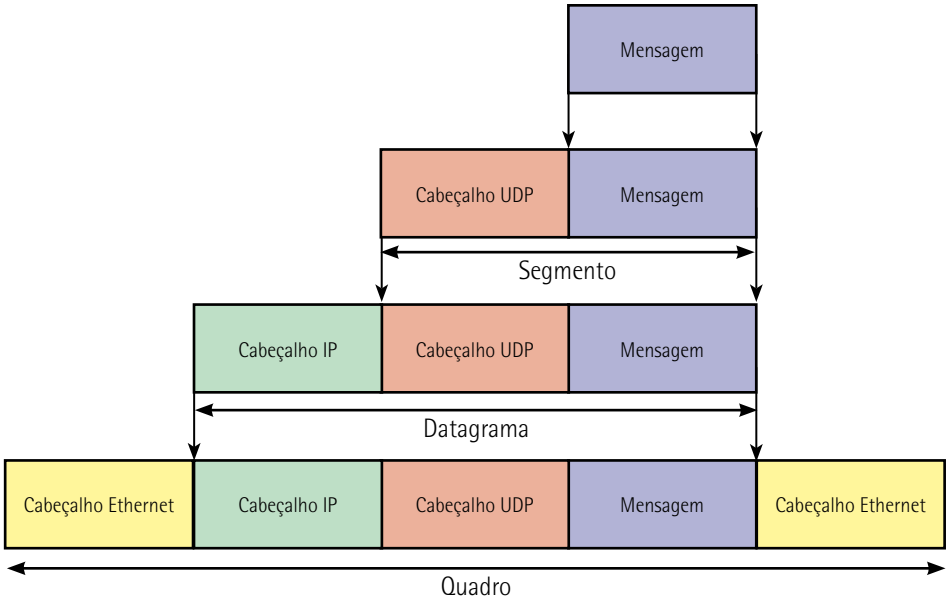

Figura 58 - Encapsulamento de transporte com o protocolo UDP (KOVACH, 2009)

Observação

Checksum é o nome dado ao processo de verificação da integridade da mensagem transmitida. De forma ampla, pode-se dizer que na origem o remetente é responsável por fazer uma "conta mágica" com os dados (*bits*) da mensagem. O resultado dessa conta é armazenado no cabeçalho na origem. Quando chega ao destino, o receptor refaz a "conta mágica" com os dados da mensagem recebidos. O resultado ele compara com o valor armazenado no cabeçalho de origem. Se for igual, é porque a mensagem chegou íntegra, ou seja, não houve perda ou alteração da mensagem original.

5.2.2 O protocolo TCP

TCP (Transmission Control Protocol) é um protocolo da Camada de Transporte que fornece um serviço de entrega confiável aos programas de aplicação. Diferente do UDP, o TCP é um protocolo mais lento, por possuir alguns tipos de validação que tornam os segmentos confiáveis e garantem a entrega das mensagens enviadas pela origem ao destino.

Considerando suas principais características, o protocolo TCP destaca-se por ser orientado à conexão, em que, antes de enviar os dados, o aplicativo deve solicitar o estabelecimento de uma conexão com o outro aplicativo, isto é, deve fornecer o endereço antes de passar os dados, como mostra a figura 59:

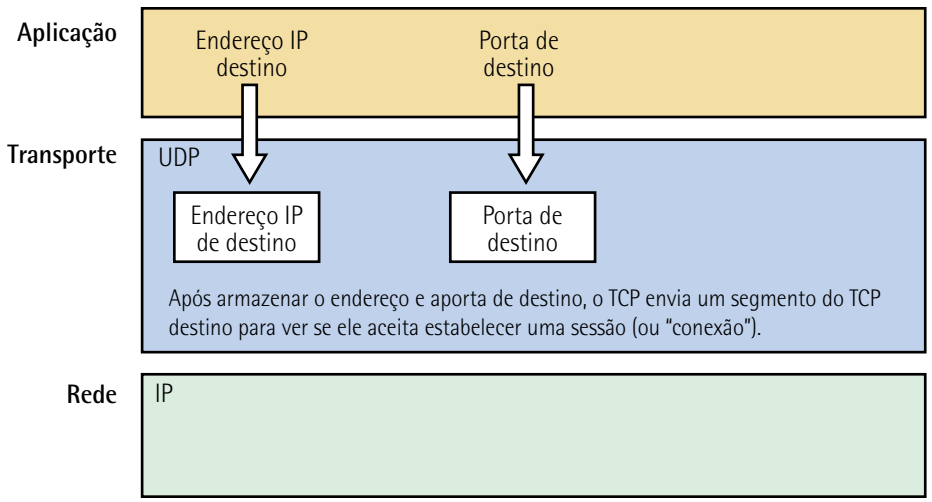

Figura 59 - Protocolo TCP, fornecendo apenas o endereço e porta de destino no estabelecimento da conexão (KOVACH, 2009)

Depois, a aplicação passa apenas a mensagem até o término da conexão, como está representado na figura 60:

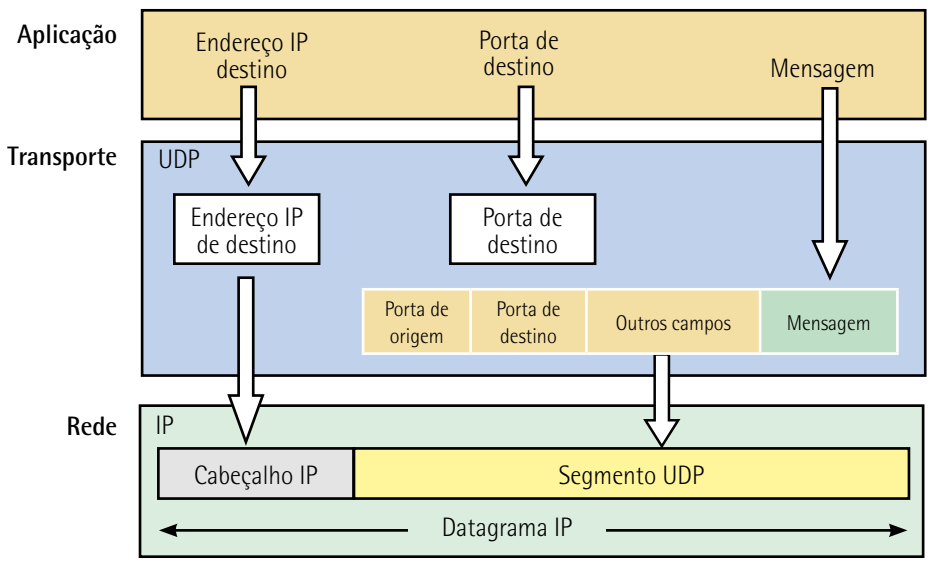

Figura 60 - A aplicação passando a mensagem depois (KOVACH, 2009)

Dizemos que o TCP possui transferência confiável de dados, pois ele usa números de sequência e reconhecimento positivo com retransmissão para entrega confiável dos dados. Assim, o TCP é um protocolo utilizado por diversas aplicações que não aceitam perdas de informações e devem garantir a entrega e a integridade das mesmas.

Os números de sequência são usados para determinar a ordem dos dados que chegam e para detectar pacotes que estão faltando. O reconhecimento positivo com retransmissão exige que o receptor envie um pacote de reconhecimento (Ack) ao remetente sempre que recebe um dado.

A figura 61 mostra o remetente enviando um dado que não chega ao destino porque sofreu algum problema no meio do caminho. O remetente, depois de enviado o dado, fica aguardando por um tempo programado uma resposta de reconhecimento do dado enviado. Se não recebê-la durante o tempo programado, ele reenvia o dado.

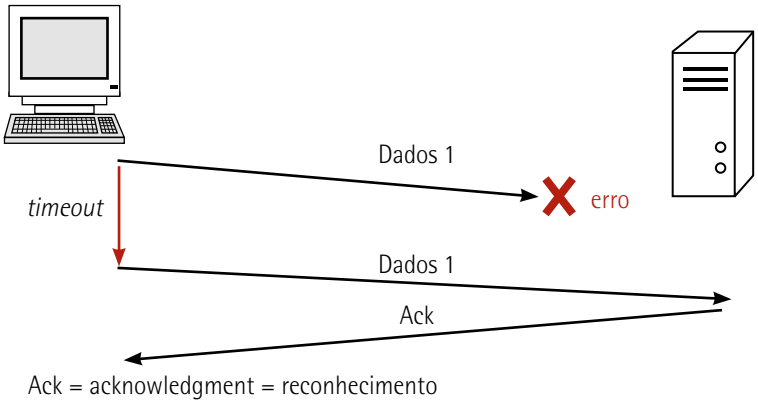

Figura 61 - Transferência confiável de dados do TCP (elaborada pela autora)

Segmento TCP

Da mesma forma que o UDP, segmento é a unidade de transferência de dados trocada entre as estações que usam o protocolo TCP, pois ambos são de camada de transporte. O segmento TCP é composto pela mensagem que veio da Camada de Aplicação mais os campos do cabeçalho TCP, como mostra a figura 62:

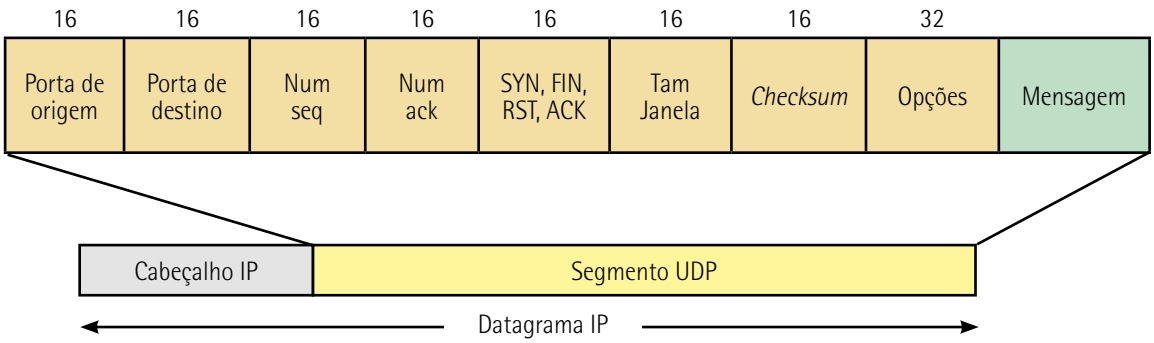

Figura 62 - O segmento TCP (KOVACH, 2009)

Vale lembrar aqui que os segmentos são usados para:

- estabelecer conexões;
- terminar conexões;
- transferir dados;
- enviar reconhecimentos;
- fazer controle de fluxo.

O formato do segmento TCP, diferente do UDP, é mais complexo e possui uma variedade de campos e controles, justamente por ter que se preocupar com os detalhes da entrega confiável dos dados.

Assim como no UDP, o segmento TCP, mostrado na figura 63, possui os dois primeiros campos referentes aos números de porta (porta origem e destino). Mostraremos a seguir o significado de cada um dos campos do cabeçalho TCP.

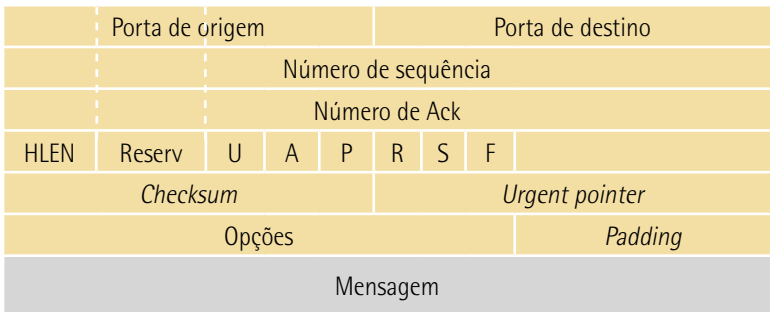

Figura 63 - O formato do cabeçalho TCP (KOVACH, 2009)

- **Portas de origem e destino**: identificam as aplicações na origem.
- **Número de sequência**: indica o número de sequência do primeiro *byte* deste segmento, ou seja, a posição relativa do primeiro *byte* que está sendo carregado. Este número não é absoluto e sim relativo, devido a questões de segurança.
- **Número de ACK**: é o reconhecimento de que ele está recebendo e indica o número do próximo *byte* que o destino espera receber. Este campo fica válido apenas quando o *bit* ACK estiver ativado (campo "A").
- **HLEN**: é o comprimento do cabeçalho, representado por um número inteiro, que especifica o tamanho do cabeçalho em blocos de 32 *bits*. Geralmente possui 20 *bytes* (5 blocos de 32 *bits*), ou seja, o tamanho normal deste segmento, sem o campo opções.
- **U (URG)**: indica que o campo *Urgent Pointer* é válido e deve ser interpretado. Assim, o módulo TCP deve processar o dado urgente antes de processar qualquer outro dado.
- **A (ACK)**: *bit* usado para indicar que o segmento contém um reconhecimento. Quando A=1, significa que tem Ack. Se A=0, esse campo torna-se insignificante e não é tratado.
- **P (PSH)**: solicita ao módulo TCP receptor para enviar os dados imediatamente para a aplicação, ou seja, força o envio imediato de dados (sem esperar dados adicionais). Normalmente, TCP "bufferiza" os dados que chegam até atingir certo valor antes de enviar para a aplicação. Por exemplo, Telnet usa esse *bit* para forçar a entrada do caractere digitado no servidor, diminuindo o atraso na geração dos ecos.
- **R (RST)**: é utilizado para rejeitar um estabelecimento de conexão. Normalmente, o TCP envia um segmento com este *bit* setado (=1) quando detecta um problema com a conexão.
- **S (SYN)**: *bit* usado para indicar fase de estabelecimento de conexão. Ele informa também o número inicial de sequência.

- **F (FIN)**: *bit* usado para indicar fase de término de conexão. Ele fecha apenas o fluxo de dados no sentido da sua transmissão. O módulo receptor deve também enviar uma mensagem com FIN ativado para fechar completamente a conexão.
- **Tamanho da janela:** indica quantos *bytes* o receptor está disposto a aceitar o envio sem confirmação (ack). É usado pelo lado do receptor para informar o tamanho máximo do seu buffer - controle de fluxo.
- ***Checksum***: abrange o cabeçalho, os dados e o pseudocabeçalho. É baseado na soma em complemento de um.
- **Opções**: esse campo é usado pelo módulo TCP para uma das pontas da conexão informar à outra ponta o tamanho máximo de segmento (MSS - *Maximum Segment Size*) que ele está disposto a receber. A opção MSS só é válida num segmento com o campo SYN ativado. Se o MSS não for transmitido, o TCP assume um MSS *default* que é de 536 *bytes*.
- ***Urgent Pointer***: é usado para identificar um bloco de dados urgentes dentro do campo de dados.
- ***Padding***: campo de preenchimento usado para garantir que o tamanho do cabeçalho seja múltiplo de 32 *bits*.

Veja na figura 64 como funciona o estabelecimento de uma conexão TCP, que utiliza os campos SYN, Número de sequência, Ack e Número de Ack, conhecido como *tree way handshake*.

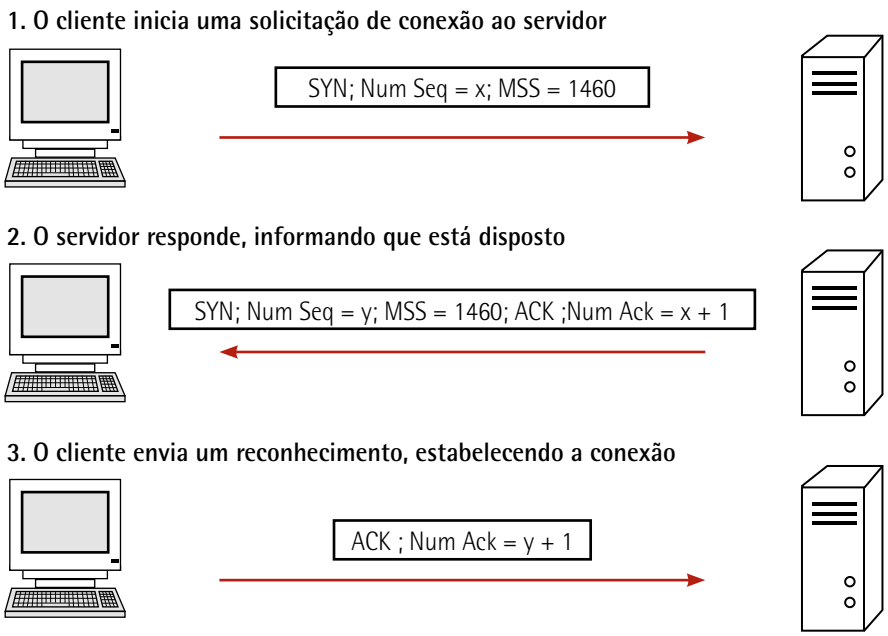

Figura 64 - Estabelecimento de uma conexão TCP (KOVACH, 2009)

- No exemplo da figura, é como se o cliente iniciasse a solicitação dizendo para o servidor receptor não enviar mais do que 1460 de tamanho máximo do segmento.
- O servidor analisa o pedido e, se ele aceitar as condições da conexão (SYN=1), responde informando que está disposto.

- O cliente então envia um reconhecimento de que recebeu a aceitação, estabelecendo a conexão.

A figura 65 mostra agora como funciona o término de uma conexão TCP:

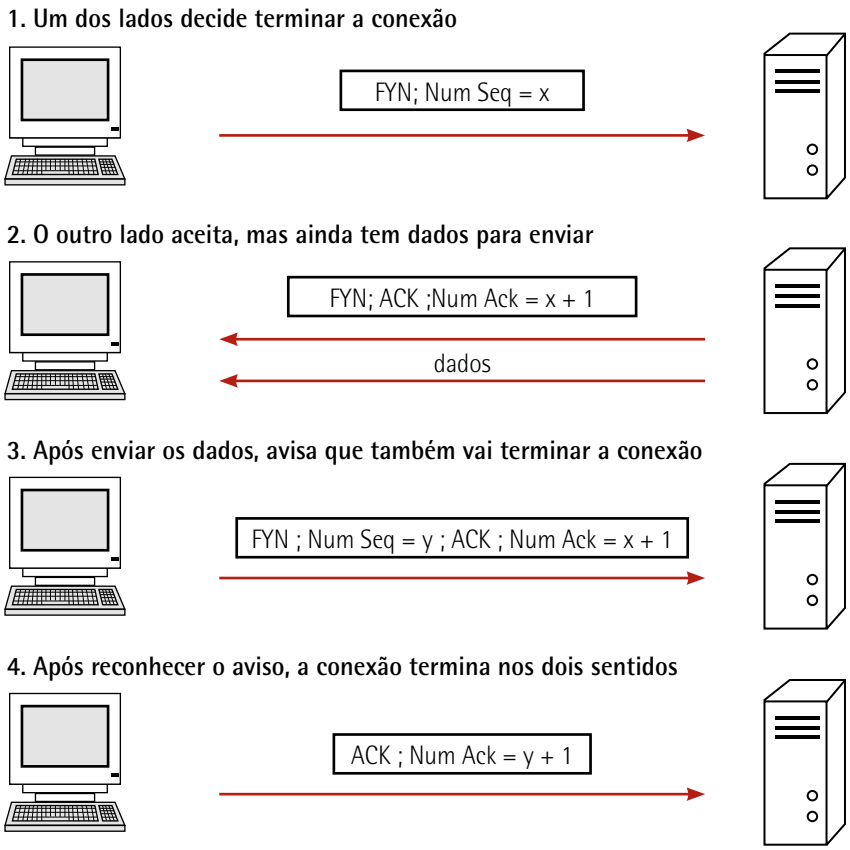

Figura 65 - Encerramento de uma conexão TCP (KOVACH, 2009)

- O pedido de encerramento da conexão é feito por um dos lados. O campo FIN é ativado e, junto a esta mensagem, é apresentado o último número de sequência conhecido.
- O outro lado pode aceitar, mesmo se ainda tiver dados para enviar. Assim, ele habilita também o campo FIN, mostrando que concorda com o encerramento da conexão, mas mostra que seu número de Ack é maior que o número de sequência enviado na solicitação de encerramento da conexão. E, em seguida, envia os dados.
- Após enviar, avisa que também vai encerrar a conexão, agora com o novo número de sequência.
- O lado que solicitou o término da conexão, após reconhecer o aviso do outro lado, envia nova mensagem, encerrando a conexão nos dois sentidos.

Como vimos, o cabeçalho TCP contém um campo que indica o número sequencial do primeiro *byte* contido no campo de dados (número de sequência) e, assim, coloca na ordem correta os segmentos que chegam fora de ordem e descarta os duplicados. O TCP reconhece apenas o maior número dos segmentos recebidos sem erro. A figura 66 mostra que o número de sequência no cabeçalho pula de acordo com a quantidade de *bytes* existentes dentro dos segmentos.

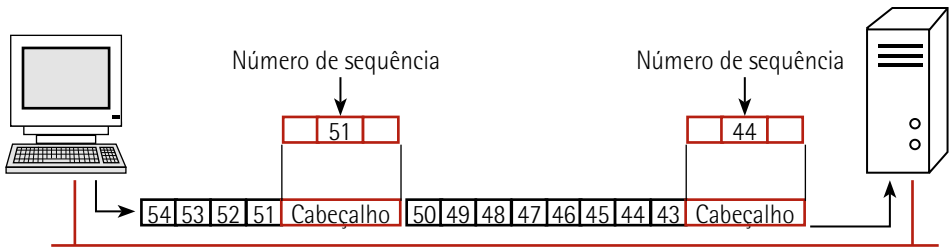

Figura 66 - No TCP, cada byte possui um número sequencial (KOVACH, 2009)

Existem diversas implementações do *software* TCP para trabalhar com as janelas deslizantes. Em geral servem para medir as condições da rede e identificar, previamente, se há condições de transmitir. Vejamos algumas das implementações TCP importantes a seguir.

No comportamento que chamamos de pare-espere, o transmissor tem que esperar pelo Ack do pacote anterior para transmitir um novo pacote. Assim, se a distância for grande, o tempo de espera também pode ser grande. Veja a figura 67:

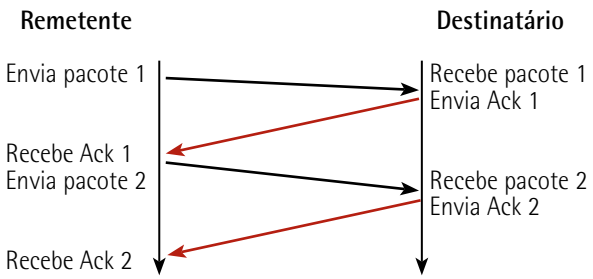

Figura 67 - Comportamento pare-espere, do TCP (KOVACH, 2009)

O TCP pode utilizar um esquema de reconhecimento acumulativo em que o receptor vai acumulando sequencialmente os *bytes* recebidos desde o estabelecimento da conexão. Assim, cada segmento de reconhecimento (ACK) especifica no campo Número de Ack o número do próximo *byte* que o receptor espera receber da fonte.

Vale lembrar que nesta implementação o receptor não gera o Ack de um segmento se o segmento anterior não foi recebido. Em vez disso, o receptor retransmite o último Ack enviado. Isso significa que, se um ou mais Acks forem perdidos, mas um Ack do segmento que foi enviado posteriormente for recebido pelo transmissor, ele pode considerar que todos os segmentos anteriores foram recebidos pelo receptor.

Dizemos também que o protocolo TCP faz controle de fluxo, ou seja, o TCP receptor envia um valor (tamanho da janela) ao transmissor nos pacotes de reconhecimento, que especifica o número de *bytes* que o transmissor pode transmitir sem esperar pelo reconhecimento dos mesmos. A janela desliza à medida que chegam os reconhecimentos e, quando esse valor for igual a zero, o transmissor para de enviar os dados. A este processo chamamos de **janela deslizante** e é utilizado para aumentar a taxa de transmissão dos pacotes.

Na figura 66 está representado como a janela desliza, à medida que chegam os reconhecimentos de transmissão. A janela inicial é igual a 8, ou seja, permite enviar até 8 números de sequência (no exemplo, de 3 a 10).

À medida que recebe os Acks do destinatário, a janela desliza, então neste exemplo vemos que, após remetente receber o Ack 3 (reconhecimento do pacote 3 enviado pelo destinatário), a janela passa a permitir o envio dos números de sequência de 4 a 11, e assim por diante.

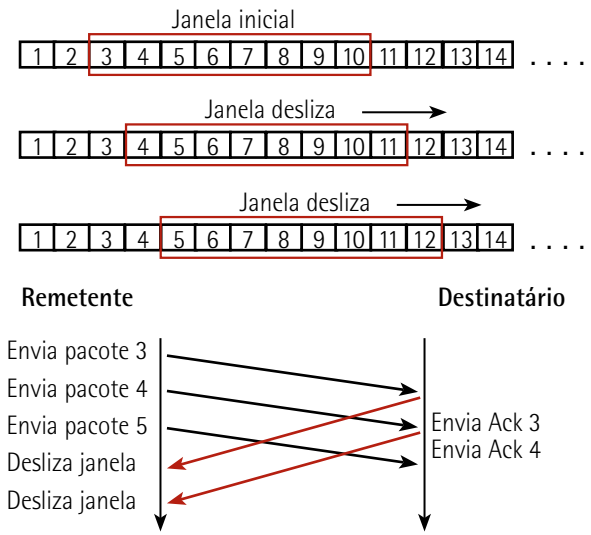

Figura 68 - Exemplo de TCP com janela deslizante (KOVACH, 2009)

Se ainda está difícil de entender como funciona a janela deslizante, veja esse outro exemplo, representado pela figura 69:

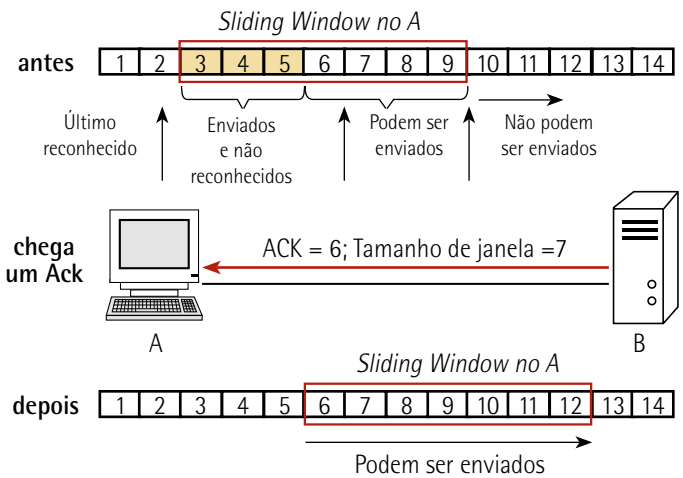

Figura 69 - Outro exemplo de TCP com janela deslizante (KOVACH, 2009)

Antes da chegada do Ack=6, o último número de sequência reconhecido (que o remetente recebeu o Ack) era o 2; os de número 3, 4 e 5 foram enviados pelo remetente, mas ainda não foram reconhecidos. Como o tamanho da janela é 7, os números de sequência 6, 7, 8 e 9 podem ser enviados neste intervalo; os demais, ainda não.

Com o reconhecimento do número de sequência 6, a janela desliza, permitindo enviar os próximos 7 números de sequência, a partir do 6, ou seja, pode enviar o 6, 7, 8, 9, 10, 11 e 12.

Observação

Note que a mensagem de Ack contém o número do próximo, que o destinatário está esperando receber, ou seja, se a mensagem é enviada com Ack = 6, significa que o destinatário recebeu até o número de sequência 5 e está esperando receber o de número 6.

Além disso, o TCP é um protocolo capaz de fazer multiplexação de várias conexões em uma mesma porta, através do endereço IP e número de porta. Como uma conexão TCP é formada por quatro variáveis (endereço IP de origem, porta de origem, endereço IP de destino, porta de destino), variando apenas um dos parâmetros, já se tem nova conexão, podendo ser gerada simultaneamente a outras.

A figura 70 mostra que o sistema final A, cujo endereço IP é o 1.1.1.1, tem duas aplicações sendo executadas, uma na porta 200 e outra na porta 100, e ambas se comunicando com a aplicação que roda na porta 25 do host C, de IP 3.3.3.3. Assim, embora o endereço IP de origem, endereço IP de destino e porta de destino sejam os mesmos, a porta de origem varia para cada uma das aplicações de origem, então o TCP estabelece uma conexão diferente para cada uma delas, pois pelo menos 1 dos 4 pontos do par ordenado (IP origem, porta de origem; IP destino, porta de destino) é diferente.

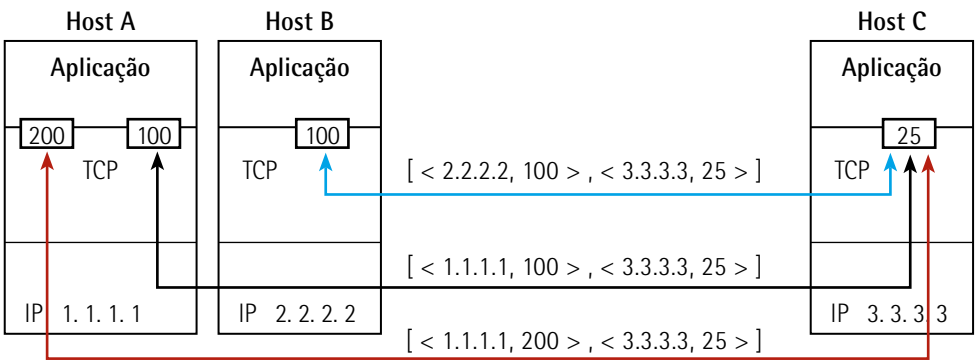

Figura 70 - Multiplexação de conexões simultâneas (KOVACH, 2009)

Saiba mais

Abaixo há duas dicas de leitura sobre a Arquitetura TCP/IP.

Uma referência muito conceituada: *TCP/IP: a bíblia*, de Meeta Gupta, Mridula Parihar, Paul Lasalle e Rob Scrimger, Editora Campus/Elsevier, 2002.

Guia Ilustrado do TCP/IP, de Matthew Naugle, Editora Berkeley Brasil, 2001.

6 CAMADA DE REDE

A camada 3 está logo abaixo da Camada de Transporte na pilha de protocolos do modelo OSI, como mostra a figura 71, e é responsável pelo processo de interconexão de redes. As redes são interligadas por dispositivos chamados roteadores.

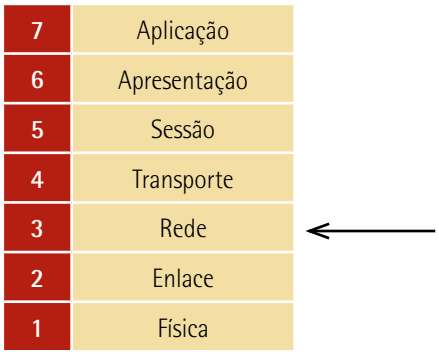

Figura 71 - Camada de Rede do Modelo OSI (KOVACH, 2009)

Como mostra a figura 72, quando um computador da rede 1 quer enviar um dado para um computador da rede 2, ele envia o pacote de dados ao roteador 1, que fica responsável por encaminhar esse pacote ao computador de destino. No caso de um computador da rede 1 querer enviar um pacote de dados para um computador na rede 3, ele envia o pacote ao roteador 1, que então passará esse pacote diretamente ao roteador 2 e que então se encarregará de entregar esse pacote ao computador de destino na rede 3.

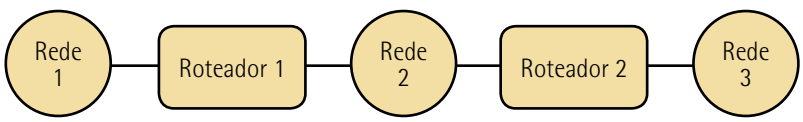

Figura 72 - Exemplo de interconexão de redes através de roteadores (KOVACH, 2009)

O roteador é, sem dúvida, o principal agente no processo de interconexão das redes, pois determina as rotas baseado nos seus critérios, endereçando os dados pelas redes e gerenciando suas tabelas de roteamento. A entrega de pacotes é feita facilmente pelo roteador porque os pacotes de dados possuem o endereço IP do computador de destino. No endereço IP há a informação de que a rede o pacote deve ser entregue. Além disso, os roteadores possuem internamente interfaces de saída, para onde os enlaces de entrada repassam os pacotes no interior dos roteadores. As interfaces de saída servem para que o roteador encaminhe os pacotes ao roteador vizinho da rota selecionada por ele como a melhor para transmitir ao destino.

É assim que as redes baseadas no protocolo TCP/IP funcionam. Elas têm um ponto de saída da rede, chamado de *gateway*, que é para onde vão todos os pacotes de dados recebidos e que não são para aquela rede. As redes subsequentes vão, por sua vez, enviando o pacote aos seus *gateways* até que o pacote atinja a rede de destino.

A partir de dispositivos, como roteadores e protocolos de roteamento, é a Camada de Rede que decide qual o melhor caminho para se chegar ao destino, bem como estabelece as rotas. Algumas arquiteturas de rede requerem determinar o caminho antes de enviar os dados. Essa camada também já reconhece o endereço físico, que é convertido para endereço lógico (o endereço IP) por meio de um protocolo específico chamado ARP, responsável por tal tradução nesse nível.

A figura 73 mostra uma rede simples, com dois sistemas finais (S1 e S2) e diversos roteadores no caminho. Para a comunicação entre os dois sistemas finais, a Camada de Rede do sistema final de origem (S1) encapsula as informações em um pacote (datagrama) e encaminha ao roteador vizinho, que encaminhará ao seu próximo vizinho, e assim por diante, até chegar ao sistema final S2, onde a Camada de Rede extrairá os segmentos de Camada de Transporte e os entregará a esta camada para enviar à Camada de Aplicação de destino. Note que, nos sistemas finais, a pilha de protocolos é completa, enquanto nos roteadores só vai até a Camada de Rede, pois esses equipamentos não rodam protocolos das outras camadas superiores.

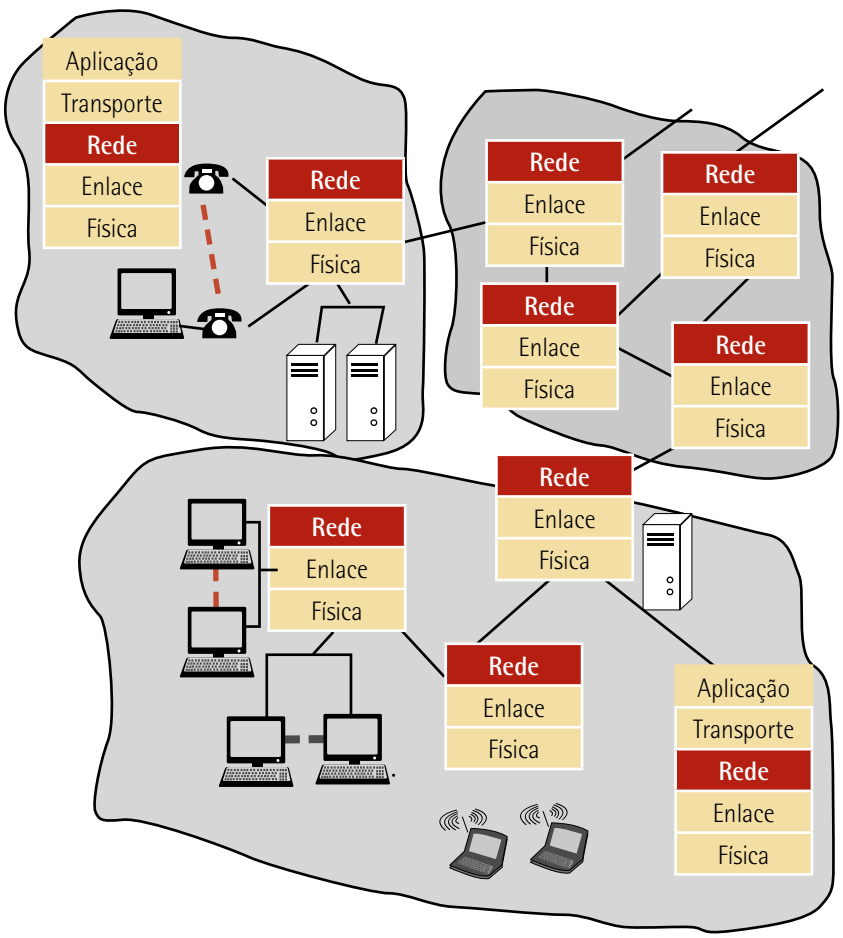

Figura 73 - Exemplo de comunicação entre sistemas finais através da Camada de Rede (KUROSE; ROSS, 2010)

Saiba mais

Agora que você está familiarizado com as camadas de Aplicação e Transporte e está iniciando seu estudo da Camada de Rede, assista ao vídeo *Guerreiros da internet*. É muito interessante e mostra como a internet funciona, ilustrando de forma bem clara o trajeto dos pacotes de dados, desde a hora que você faz uma consulta em seu navegador até a hora que a página começa a aparecer na tela. Acesse <www.warriorsofthe.net>. O vídeo está disponível para baixar em português também.

6.1 O protocolo IP

O IP (Internet Protocol) é o protocolo responsável pelo encaminhamento dos datagramas desde a origem até o destino através da internet, como pode ser visto na figura 74. Assim como alguns protocolos de Camada de Transporte, como o UDP, o IP é um protocolo não orientado à conexão.

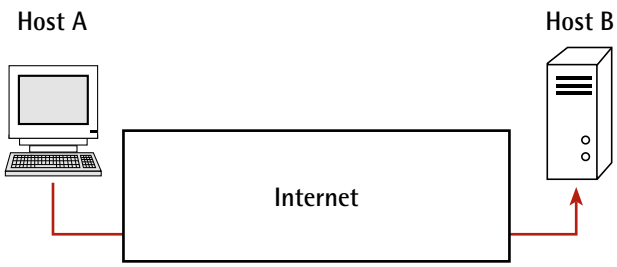

Figura 74 - Protocolo IP responsável pelo encaminhamento dos datagramas (elaborada pela autora)

O serviço oferecido pelo protocolo IP fornece um modelo conhecido como "serviço de melhor esforço", pois ele utilizará a maior banda possível disponível na rede para encaminhar seus pacotes, tentando fazer isso sem atraso e de modo que seus datagramas cheguem de forma ordenada, ou seja, na ordem com que foram enviados pela origem. Embora ele se esforce ao máximo, não existem garantias de que a transmissão será livre de erros, que os pacotes serão entregues e que não haverá perda de pacotes ou atrasos.

Dizemos, assim, que o IP é um protocolo não confiável, por não implementar mensagens de confirmação (como faz o TCP, na camada 4) de que os datagramas foram entregues ao destino. Se houver qualquer tipo de perda, elas serão corrigidas apenas pela camada 4, com ajuda do protocolo TCP de transporte. Esta é uma grande vantagem do IP: simplicidade. O IP tenta ser o mais rápido que pode, é capaz de detectar erros, mas se apoia nas correções desses erros por meio de protocolos de transporte.

O Datagrama IP

Assim como os segmentos na Camada de Transporte, o datagrama está dividido em duas partes. Entretanto, aqui as partes são o cabeçalho IP e o segmento TCP, como mostrado na figura 75:

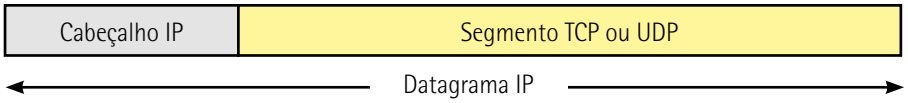

Figura 75 - Protocolo IP responsável pelo encaminhamento dos datagramas (elaborada pela autora)

O cabeçalho contém toda a informação necessária que identifica o conteúdo do datagrama. Na área de dados está encapsulado o pacote do nível superior, ou seja, um pacote TCP ou UDP.

O formato do datagrama IP tem 32 *bits* e está mostrado na figura 76:

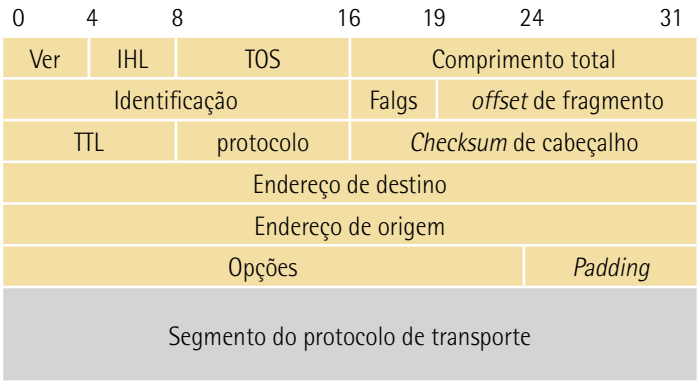

Figura 76 - O datagrama IP (elaborada pela autora)

Vamos conhecer o que são cada um desses campos do IP:

- **Versão**: indica a versão do protocolo IP sendo usada (IPv4 ou IPv6).
- **IHL (Internet Header Length)**: indica o comprimento do cabeçalho em múltiplos de 32 *bits* (no caso de IPv4).
- **TOS (Type of service - Tipo de Serviço)**: fornece uma indicação dos parâmetros da qualidade desejada, para um nó especificar uma preferência de como os datagramas poderiam ser manuseados.

Os três *bits* de precedência especificam a prioridade dos pacotes:

– 0 para pacotes normais e 7 para pacotes mais prioritários.

– É o campo mais importante para distinguir pacotes de voz dos pacotes de dados.

Os *bits* D, T e R indicam o tipo de transporte desejado pelo pacote.

– O bit D solicita minimizar atraso.

– O bit T solicita maximizar o *throughput*.

– O bit R solicita maximizar confiabilidade.

- **Comprimento total**: fornece o comprimento total do datagrama IP, incluindo cabeçalho e dados, medido em *bytes* de oito *bits*.
- **Tempo de vida (*Time-to-Live* - *TTL*)**: indica o tempo máximo que um datagrama pode trafegar em uma rede internet. Tornou-se um campo de contagem de nós caminhados. Assim, cada roteador decrementa este campo de um. Se o valor deste campo chegar a zero antes de atingir o destino, o datagrama é descartado.
- **Protocolo**: indica qual protocolo seguinte será usado.

Se for ICMP, o campo é preenchido com 1, se for TCP, 6 e se for UDP, 17.

- ***Checksum* do cabeçalho**: campo de verificação para o cabeçalho IP. Se um erro for detectado na recepção, o datagrama é descartado.
- **Endereço de origem**: endereço IP de origem.
- **Endereço de destino**: endereço IP de destino.
- ***Padding*** (variável): serve para garantir que o comprimento do cabeçalho seja sempre múltiplo de 32 *bits*.
- **Opções** (variável): utilizado para teste e depuração de aplicações de *softwares* de rede.

Os três campos abaixo estão relacionados com a fragmentação:

- **Identificação**: é usado para identificar um datagrama. Todos os fragmentos de um datagrama possuem a mesma identificação.
- **Flag**: identifica o controle de fragmentação:
  - O *bit* 0 é reservado;
  - Se o *bit* 1 for 0, permite fragmentação; se for 1, não permite fragmentação.
  - Se o *bit* 2 for 0, significa que é o último fragmento; se for 1, significa que ainda tem mais fragmentos.
- ***Offset* do fragmento**: indica a posição do fragmento dentro do datagrama original. É medido em unidades de 8 *bytes*.

6.1.1 Fragmentação

O protocolo IP utiliza a técnica de fragmentação quando um datagrama atravessa uma rede com MTU (Maximum Transfer Unit – Unidade Máxima de Transferência) menor do que o número de *bytes* contidos nele. MTU é o tamanho máximo de *bytes* que podem ser transferidos dentro de uma rede física. Por exemplo, a MTU de uma rede Ethernet é 1.500 *bytes* e a de uma rede Token Ring é 4.464 *bytes*, o que significa que, se pacotes maiores que tais valores tiverem que atravessar essas redes, provavelmente serão fragmentados.

É muito comum implementações usarem datagramas de 576 *bytes* sempre que eles não podem verificar se o caminho inteiro é capaz de manipular grandes pacotes. Esse é o tamanho "seguro" máximo do datagrama que um nó requer, o que significa que, se o tamanho dos datagramas for menor que 576 *bytes*, é bem provável que não serão fragmentados.

Fragmentação consiste em dividir um datagrama em pedaços menores denominados fragmentos. Os fragmentos sempre serão transportados como datagramas independentes. Ao receber o primeiro fragmento, a estação inicia uma contagem de tempo para aguardar o conjunto completo de fragmentos. Se faltar algum, o datagrama é descartado.

É importante ressaltar que os datagramas não são remontados, ou seja, uma vez fragmentados, continuam fragmentados mesmo que depois passem a encontrar redes físicas com MTU com grande capacidade, como mostra a figura 77:

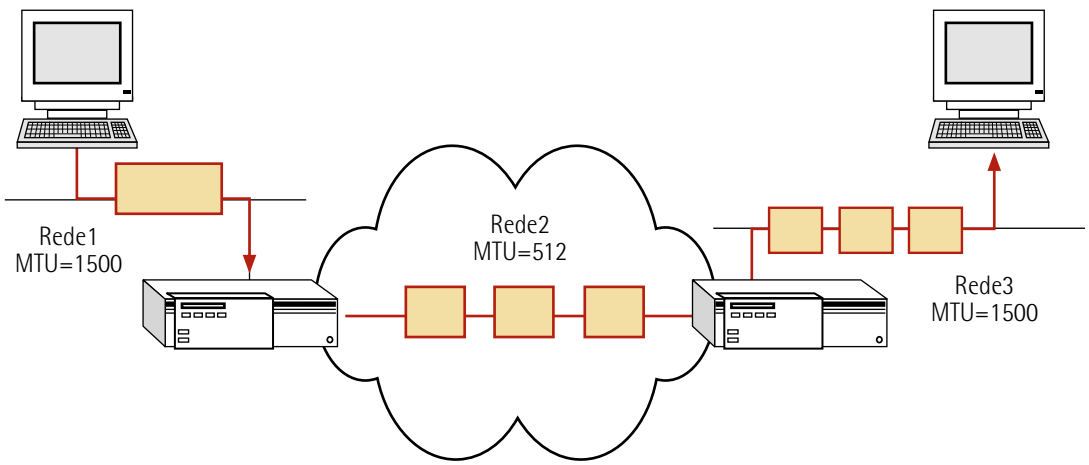

Figura 77 - Fragmentação IP (elaborada pela autora)

Se qualquer fragmento for perdido no caminho, o datagrama não poderá ser remontado.

6.2 Endereçamento IPv4

Em uma rede IP, cada ponto de interconexão de um dispositivo é identificado por um número de 32 *bits*, equivalente a 4 *bytes*, denominado endereço IP. Considerando os 32 *bits*, há um total de 2^{32} endereços IP possíveis, o equivalente a cerca de 4 bilhões de endereços IP possíveis. Tais endereços são escritos em quatro conjuntos de *bytes*, representados por números decimais, separados por pontos.

Tais endereços são representados com quatro algarismos decimais separados por ponto decimal, por exemplo, 128.10.2.30, em que 128 é o número decimal referente aos primeiros 8 *bits* do endereço; o 10, o número decimal referente ao segundo conjunto de 8 *bits* do endereço e assim por diante. Assim, este endereço transformado para *bits*, é:

10000000 00001010 00000010 00011110

Dizemos que o endereço IP é o endereço lógico de rede e cada endereço IP está associado com uma interface física de rede (por exemplo, uma placa de rede), e não com o computador. Cada endereço IP é globalmente exclusivo, mas não pode ser escolhido de qualquer maneira. Uma parte desse endereço será determinada pela sub-rede à qual ela está conectada. Chamamos de sub-rede a divisão de uma rede em redes menores, cujo tráfego fica reduzido, facilitando sua administração e melhorando o desempenho da rede. Veja uma representação na figura 78:

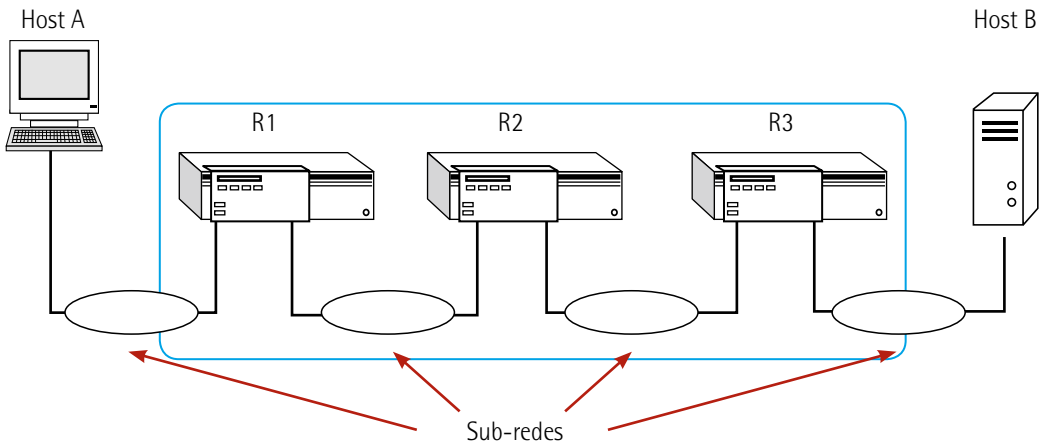

Figura 78 - Exemplo de sub-redes (elaborada pela autora)

6.2.1 Classes e formatos de endereço IP

A estratégia de atribuição de endereços da internet é conhecida como roteamento interdomínio sem classes (CIDR – Classless Interdomain Routing), que generaliza a noção de endereçamento de sub-rede. Antes da adoção do CIDR, os tamanhos das parcelas de um endereço IP estavam limitados a 8, 16 ou 24 *bits*, um esquema de endereçamento definido por classes de endereços conhecidas como classes A, B e C, respectivamente, como mostra a figura 79:

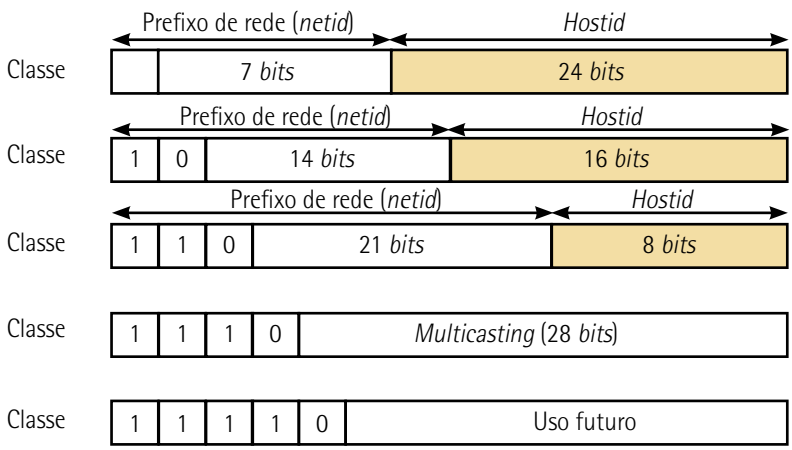

Figura 79 - Classes e formatos de endereços IP (KOVACH, 2009)

Assim, o *range* de endereçamento em cada classe está demonstrado na tabela 3:

Tabela 6 - *Range* de endereços IP das classes

Classe	Endereço mais baixo	Endereço mais alto
A	1.0.0.0	126.0.0.0
B	128.1.0.0	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	255.255.255.254

Nesse modelo, cada endereço IP é constituído por duas partes: uma se refere à rede e a outra, ao dispositivo nessa rede. Essa identificação é feita pelo par (*netid*, *hostid*), em que *netid* identifica o prefixo da rede, pelo qual o dispositivo está conectado, e *hostid* identifica o dispositivo nessa rede, como mostra a figura 80. No caso de termos um roteador conectando n redes distintas, teremos n endereços IP distintos também.

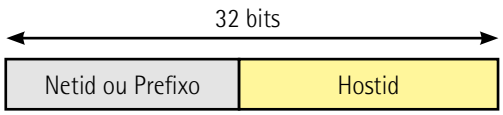

Figura 80 - Endereço IP: *netid* e *hostid* (elaborada pela autora)

Na classe A, os 8 primeiros *bits* definem o prefixo de rede, e os restantes 24 *bits*, o dispositivo na rede; na classe B, têm-se 16 *bits* para identificar o prefixo de rede e outros 16 *bits* para a identificação do dispositivo na rede; já a classe C é o inverso da classe A, em que os primeiros 24 *bits* definem prefixos de rede, e os 8 seguintes, dispositivos na rede. A classe D é utilizada para aplicações de *multicasting*, muito útil, por exemplo, para transmissão de vídeo, em que os receptores que quiserem receber esse pacote de *multicasting* podem sintonizar suas interfaces nesses endereços, solicitando recebimento. A classe E, por muito tempo reservada para uso futuro, tem sido utilizada para testes de otimização do protocolo IP pelo IETF (Internet Engineering Task Force).

Nessa notação de endereçamento IP, temos que os endereços reservados para a classe A são normalmente aplicados em empresas muito grandes, nacionais ou internacionais, ou universidades muito grandes, como a Universidade de São Paulo (USP). A tabela 3 mostra que, nessa classe, os endereços variam seus prefixos de 1 a 126 no primeiro octeto, já que os outros três octetos definem o *host*.

Os endereços de classe B são utilizados para redes de empresas ou universidades de tamanho médio e contemplam os endereços de prefixos de inicial de 128 a 191, que representam seus dois octetos que identificam os prefixos de rede, já que os outros dois octetos identificam os *hosts* na rede.

Os endereços de classe C, geralmente usados para empresas de pequeno e médio portes, têm seu *range* de prefixos de rede de três octetos variando de 192 a 223, com seu último octeto utilizado para identificação dos *hosts* na rede.

Classe A (**a**.b.c.d)	**a** identifica a rede b.c.d identificam o *host*	Exemplo: 10.10.5.1
Classe B (**a.b**.c.d) a = 128 – 191	**a.b** identificam a rede c.d identificam o *host*	Exemplo: 129.10.5.1
Classe C (**a.b.c**.d) a = 192 – 223	**a.b.c** identificam a rede d identifica o *host*	Exemplo: 194.10.5.1

Figura 81 - Classes e *ranges* de endereçamento IP (KOVACH, 2009)

Na internet, cada sub-rede possui um prefixo de rede e é para onde os roteadores encaminham os pacotes. Cada roteador armazena as informações dos prefixos de rede em uma tabela de roteamento, que contém apenas prefixos de rede e não endereços completos de dispositivos na rede. Nos próximos itens daremos mais detalhes sobre roteamento.

6.2.2 Máscaras de sub-rede

São baseadas no prefixo de rede, em que os roteadores vão escolhendo seus caminhos (o roteamento é feito) até chegar ao último roteador antes da rede de destino. Quando chegar à sub-rede de destino, o endereço referente ao *hostid* será olhado para buscar dentro da sub-rede o dispositivo final a que se destina a mensagem.

Chamamos de máscaras de sub-rede os *bits* que determinam o prefixo de rede. Assim como o endereço IP, são valores de 32 *bits* que permitem ao receptor de pacotes IP identificar quais são os *bits* do endereço que fazem parte do *netid* e quais fazem parte do *hostid*. Os *bits* em 1 da máscara identificam os *bits* do endereço IP que são usados como prefixo de rede; os *bits* em 0 da máscara identificam os *bits* do endereço IP que são usados para identificar o dispositivo na rede (*hostid*).

Para o roteamento, utilizam-se máscaras-padrão (*default*) de cada classe até chegar à rede de destino, como mostram as figuras 82 e 83:

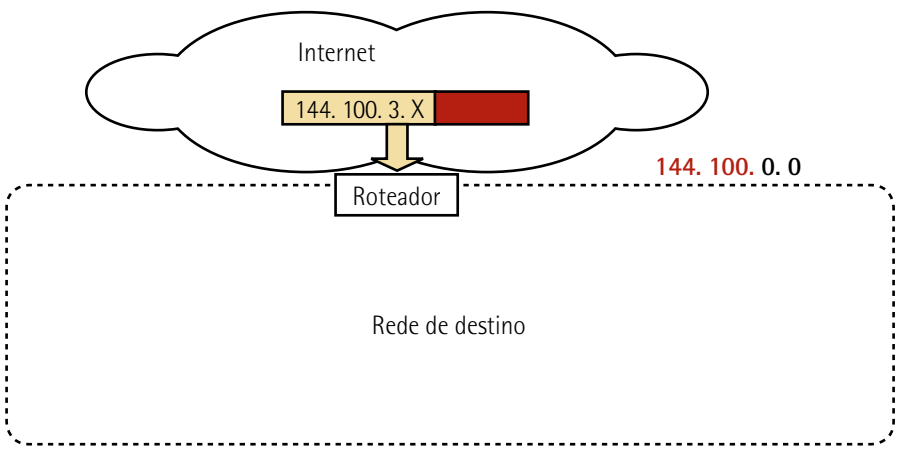

Figura 82 - Exemplo de utilização de máscara (elaborada pela autora)

Ao atingir a rede de destino, o roteador aplica a máscara no endereço e faz o roteamento utilizando o prefixo resultante.

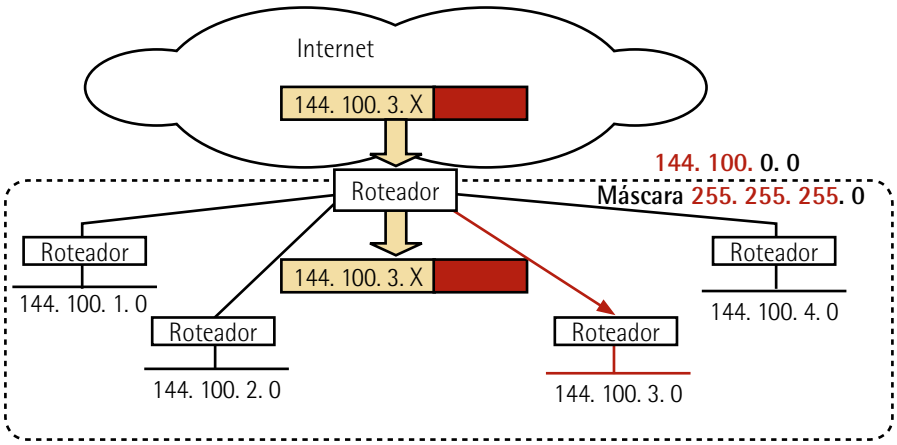

Figura 83 - Exemplo de utilização de máscara - continuação (elaborada pela autora)

As máscaras-padrão utilizadas pelo roteador para cada classe de endereçamento IP são:

- Classe A: máscara de sub-rede 255.0.0.0
- Classe B: máscara de sub-rede 255.255.0.0
- Classe C: máscara de sub-rede 255.255.255.0

Sabendo que os endereços lógicos IP na rede são únicos e que sua utilização aumenta muito a cada ano, rapidamente percebeu-se que os números de endereçamento IPv4 exclusivos rapidamente acabariam. Para contornar esse problema surgiu a ideia de se utilizarem máscaras de sub-rede com valores diferentes das máscaras-padrão, fazendo, assim, ganhar novos endereços IP na divisão em sub-redes.

Somente então, ao atingir a rede de destino, é que o roteador aplica a máscara no endereço IP de destino, passando a rotear baseando-se no prefixo de sub-rede. Em uma rede física, todos os computadores devem ter o mesmo prefixo e usar a mesma máscara. Aplicando-se máscaras de sub-rede com valores diferentes dos valores das máscaras padrão, é possível ganhar ou economizar sub-redes e, portanto, endereços IP. A figura 84 mostra como a aplicação de máscaras de sub-rede diferentes das de *default*, representada pelos *bits* em 1, faz com que *bits* inicialmente reservados ao *hostid* passam a ser *bits* de endereço de sub-redes. Os *bits* em 0 mostrarão quantos *bits* por sub-rede serão utilizados para representar os dispositivos na sub-rede.

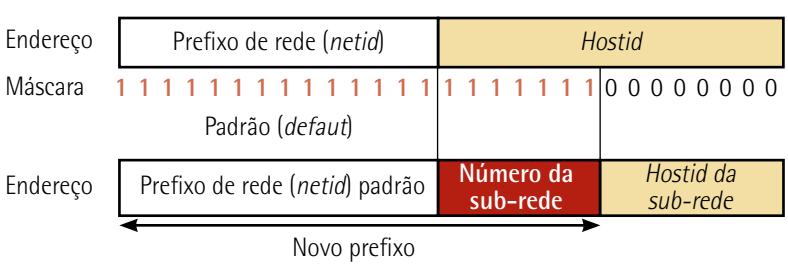

Figura 84 - Máscara de sub-rede (KOVACH, 2009)

Veja um exemplo real de aplicação de máscara de sub-rede. Considerando o endereço 129.10.0.0, de classe B, sua máscara padrão é a 255.255.0.0, que provê um único segmento de rede com 65.536 *hosts* ou 64 K *hosts*. Se considerarmos o terceiro conjunto de *bits* da máscara diferente de zero, perceberemos que é possível aumentar o número de segmentos de rede (sub-redes) e variar o número de *hosts* por sub-rede, como mostra a tabela 5, a seguir.

Lembrete

Lembre-se: é a máscara de rede a responsável por mapear as sub-redes. Repare sempre na quantidade de *bits* 1 que existe no endereço. São eles que indicam o prefixo de rede e, portanto, a quantidade de sub-redes possível no endereço.

Aplicando, por exemplo, uma máscara de sub-rede de 255.255.240.0, passando o 240 para binário, vê-se que agora a máscara de sub-rede tem quatro *bits* em 1 e quatro *bits* em 0. Os *bits* em 1 representam 16 sub-redes, pois $2^4 = 16$. Os quatro *bits* em 0 somados aos outros oito *bits* 0 da máscara representam a quantidade de $2^4 + 2^8 = 2^{12} = 4.096$ ou 4 K *hosts* para cada uma das 16 sub-redes possíveis.

Vale lembrar aqui que, na prática, a primeira e a última sub-redes são descartadas, pois o primeiro IP da primeira sub-rede representa o endereço de rede e o último IP da última sub-rede representa o endereço de *broadcast*, que não é considerado pelos roteadores como endereço válido na rede. Esse endereço é utilizado quando se deseja transmitir uma mensagem a todos os receptores da rede ao mesmo tempo. Dessa forma, na prática, a quantidade real de sub-redes é a apresentada na tabela 4, menos duas.

Tabela 7 - Classes e ranges de endereçamento IP

Prefixo de rede classe B: 129.10. 0. 0 Máscara de rede padrão: 255.255.0.0		1 segmento de rede 64 K *hosts*
Máscara de sub-rede	255. 255.**128**. 0 **1**0000000	2 sub-redes 32 K *hosts* por sub-rede
	255. 255.**192**.0 **11**000000	4 sub-redes 16 K *hosts* por sub-rede
	255. 255. **224**.0 **111**00000	8 sub-redes 8 K *hosts* por sub-rede
	255. 255. **240**.0 **1111**0000	16 sub-redes 4 K *hosts* por sub-rede
	255. 255. **248**. 0 **11111**000	32 sub-redes 2 K *hosts* por sub-rede
	255. 255.**252**.0 **111111**00	64 sub-redes 1 K *hosts* por sub-rede
	255.255. **254**. 0 **1111111**0	128 sub-redes 512 *hosts* por sub-rede
	255. 255.**255**. 0 **11111111**	256 sub-redes 256 *hosts* por sub-rede

Fonte: Kovach, 2009.

Saiba mais

Conheça este simulador de sub-redes *online*: <www.subnet-calculator.com>. Ele pode ajudar a exercitar os cálculos de sub-rede IP, informando também quantos *hosts* válidos podemos ter em cada sub-rede e os respectivos endereços de *broadcast*. Também informa se determinado endereço IP pertence a uma classe A, B, C ou D.

6.2.3 Endereços IP reservados

Ao definir o *range* de endereços IP utilizado por cada classe de endereçamento IP, você talvez tenha se perguntado por que o *range* de endereços que começam com 127 não foi especificado.

Existem alguns endereços IP que são especiais, reservados para funções específicas e que não podem ser utilizados como endereços de uma máquina da rede ou não são levados em conta pelos roteadores. A seguir, são apresentados estes endereços:

- 0. 0. 0. 0: especifica um endereço desconhecido e é usado por uma máquina da rede quando ela não sabe o seu endereço. Muitas vezes esse endereço é utilizado pelas máquinas na rede como endereço de origem nos protocolos de configuração dinâmica, como o DHCP.
- 255. 255. 255. 255 ou *hostid* com todos os *bits* em 1: é usado como endereço de *broadcast* local (dento de uma sub-rede). Como vimos anteriormente, um pacote de *broadcast* é destinado a todos os dispositivos conectados à rede. Aqui, vale lembrar que os pacotes de *broadcast* são bloqueados pelos roteadores.
- 127.0. 0. 0 - 127. 255. 255. 255: são utilizados como endereço de *loopback* local, reservado para testes de *debugging* da configuração de rede da máquina ou para a comunicação entre processos na mesma máquina local. Ao utilizar o endereço de *loopback* para enviar os dados, o *software* IP não deixa a mensagem prosseguir através da rede, devolvendo-a à Camada de Transporte.

Outros endereços especiais, chamados endereços privados, são reservados para redes privadas e servem para montar uma rede TCP/IP sem gerar conflitos com os endereços IP da internet. Os roteadores reconhecem esses endereços como independentes, de uma rede particular, e não repassam os pedidos de datagramas que façam referência a esses endereços para o resto da internet. Esses endereços nunca serão considerados pelos roteadores no sistema de roteamento global da internet. Assim, tais endereços podem ser utilizados simultaneamente por várias organizações.

São estes os endereços privados considerados pela IANA (Internet Assigned Numbers Authority):

- 10. 0. 0. 0 - 10. 255. 255. 255
- 172. 16. 0. 0 -172. 31. 255. 255
- 192. 168. 0. 0 -192. 168. 255. 255

A interconexão com a internet de redes que utilizam endereços privativos é feita normalmente por meio de dispositivos conhecidos como NAT (Network Address Translator).

6.2.4 NAT e DHCP

Para acessar a rede internet, cada computador deve ter o protocolo TCP/IP configurado corretamente. O NAT surgiu como uma alternativa real para o problema de falta de endereços IPv4. Com as redes privadas, tornou-se necessária uma solução para que essas redes recebessem respostas de seus pedidos feitos para fora da rede internet. NAT é um serviço que permite que essa tradução seja feita por meio de um mapeamento de endereços IP públicos e privados.

No NAT, as traduções mais comuns de endereço privado para endereço público e vice-versa são: tradução estática e tradução dinâmica.

- Tradução estática: um endereço privado é sempre convertido em um mesmo endereço público. Essa solução é normalmente utilizada em servidores que necessitam ter sempre o mesmo endereço IP para responder a seus clientes.

- Tradução dinâmica: o endereço privado poderá não utilizar sempre o mesmo endereço público. Dessa forma, é possível que mais de um endereço privado, usado na rede local, acesse a internet usando um mesmo endereço IP público. Essa tradução é usada por clientes, isto é, computadores que não prestam serviços à rede (KOVACH, 2009).

Na tradução dinâmica, é muito comum o uso do protocolo DHCP (Dynamic Host Configuration Protocol). Um servidor DHCP distribui aos computadores clientes um IP válido na internet sempre que um cliente solicita. Se um dispositivo da rede interna solicitar uma página *web* da internet, o servidor DHCP fornecerá um endereço IP público válido para ele poder se conectar à internet. Ao final do carregamento da página solicitada, o servidor DHCP recuperará esse endereço IP de volta. No caso de o usuário desse cliente continuar navegando, o servidor poderá atribuir o mesmo endereço IP ou atribuir outro endereço IP público disponível em sua tabela de traduções.

6.3 Roteamento

Como vimos anteriormente, o prefixo de rede determina se um endereço IP está localizado na mesma rede local ou em uma rede remota. A determinação do prefixo de rede é feita executando uma operação booliana AND do endereço IP de destino com a máscara de sub-rede. Roteamento é o processo de escolher um caminho para enviar os datagramas. Veremos mais adiante como é feita essa escolha de caminho.

Existem duas formas de se fazer roteamento: direta ou indiretamente.

- Roteamento direto: ocorre se ambas as máquinas estiverem conectadas na mesma rede física, isto é, se tiverem os mesmos prefixos de sub-rede.

- Roteamento indireto: ocorre quando o destino não está conectado na mesma rede física, forçando o remetente a passar o datagrama a um roteador conectado na mesma rede física.

O roteamento de datagramas IP é feito por meio de uma tabela denominada tabela de roteamento, existente em cada máquina. A tabela de roteamento contém os prefixos de rede e o endereço IP do próximo roteador no caminho (vizinho), além de outras informações. Na tabela, existe uma linha para cada prefixo de endereço que o roteador conhece. As linhas nas tabelas de roteamento são conhecidas como **rotas**.

O roteador usa o prefixo de rede calculado para consultar a tabela de roteamento. A consulta retorna a linha da tabela que mais combina com o endereço de destino, ou seja, o endereço IP do próximo roteador e a interface de saída na qual este roteador está conectado.

A tabela de roteamento sempre aponta para roteadores que possam ser alcançados diretamente, isto é, que estejam conectados na mesma sub-rede. A tabela de roteamento mantém apenas os prefixos na tabela de roteamento e não os endereços de máquinas individuais na rede. Durante o processo de roteamento, é buscado o prefixo da rede de destino, primeiro na tabela. Se não encontrar, o datagrama é enviado a um roteador padrão (*default*). O endereço IP do roteador padrão é normalmente configurado roteador. Se nenhum roteador padrão estiver configurado, o datagrama não será repassado e, portanto, será descartado.

6.3.1 Algoritmos e protocolos de roteamento

A tabela de roteamento é criada e atualizada de tempos em tempos com os prefixos de rede que o roteador conhece e endereça. Antigamente, as tabelas de roteamento eram criadas e mantidas manualmente pelo administrador da rede, não havendo troca de informações entre os roteadores. É possível imaginar que erros de configuração de rotas eram difíceis de ser detectados, e sempre que um endereço era alterado na rede, era necessário fazer alterações na tabela de roteamento. Esse tipo de roteamento era chamado de **roteamento estático**.

A fim de garantir que não houvesse erros na criação e manutenção das tabelas, surgiu o que chamamos de **roteamento dinâmico**, quando os roteadores passaram a construir suas tabelas automaticamente, de forma dinâmica, trocando informações entre si por meio de **protocolos de roteamento**.

Para encontrar o melhor caminho para o destino de um datagrama através da rede, os roteadores utilizam **algoritmos de roteamento**. É importante notar que o melhor caminho não é sempre considerado o caminho mais curto ou o mais rápido. O melhor caminho é definido pelo algoritmo de roteamento, que o calcula baseado em diversos parâmetros, como velocidade de transmissão e tempo de atraso, entre outros, que formam uma métrica particular de cada algoritmo de roteamento.

Os algoritmos de roteamento procuram manter a tabela sempre com o melhor caminho, baseando-se nas informações recebidas pelos protocolos de roteamento. Periodicamente, os roteadores trocam entre si informações a respeito de suas rotas. Sempre que detectar alguma

alteração na rede, como a existência de novo caminho ou a remoção de um caminho, a nova informação é divulgada aos demais, que atualizam suas tabelas de roteamento.

Para cada prefixo da tabela de roteamento, é o endereço IP do próximo roteador (*next hop*) que deve ser usado para atingir este prefixo. À medida que ocorrem mudanças na rede, os protocolos de roteamento reavaliam os prefixos que podem ser alcançados e os *next hops* a serem usados para cada prefixo. O processo de encontrar o próximo passo após uma mudança na rede é denominado **convergência**.

Existem dois tipos básicos de algoritmo de roteamento que devem ser considerados:

- *Distance vector*: é baseado no número de saltos na rede (*hops*), como pode ser visto na figura 85. Esse algoritmo tem como princípio que o melhor caminho (métrica) para se chegar ao destino é o das rotas mais curtas, independentemente se a rota mais curta for a mais congestionada. Um exemplo de protocolo de roteamento que utiliza o *distance vector* como algoritmo é o RIP (Routing Information Protocol). Embora ele não leve em conta outros parâmetros sobre as condições da rede e tenha uma convergência lenta, o RIP é ainda muito utilizado como protocolo-padrão de roteamento interno de muitas organizações pela sua simplicidade, afinal, não se tem normalmente muitos roteadores na topologia de rede de uma empresa de médio porte, por exemplo.

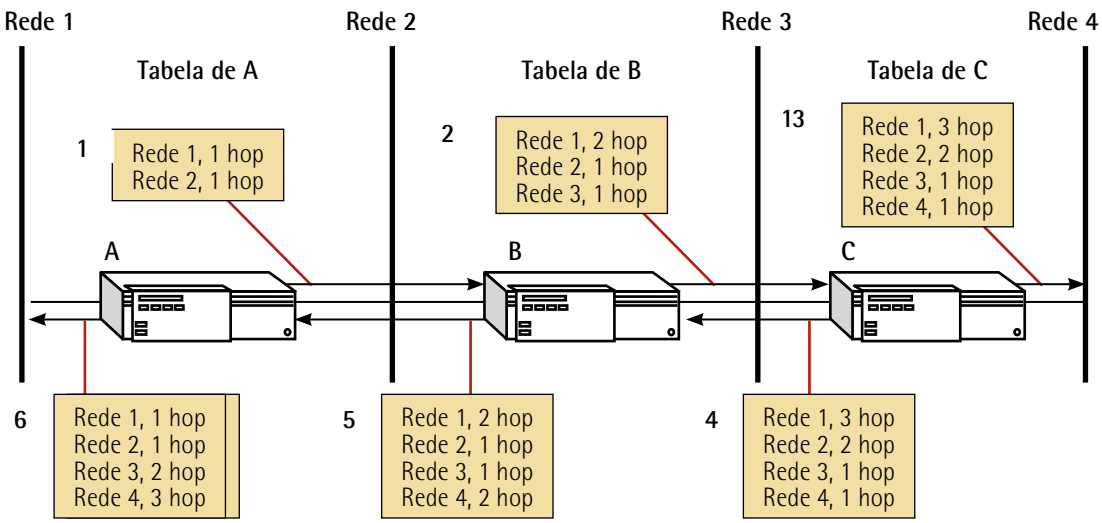

Figura 85 - O protocolo RIP, mostrando o número de saltos (KOVACH, 2009)

- *Link state*: é baseado no estado dos enlaces. Esse algoritmo considera diversos parâmetros na rede para calcular a métrica e a melhor rota para se chegar ao destino. Um dos parâmetros que ele considera é a largura de banda, que determina a velocidade de transmissão de um pacote. Ele é capaz de identificar, por exemplo, se um pacote demora mais para chegar ao destino por um caminho mais curto ou um caminho mais longo, fazendo a escolha sempre da menor métrica calculada para se chegar ao destino. OPSF (Open Shortest Path First) é um exemplo de protocolo de roteamento comum utilizado em redes de grande porte.

Tanto o RIP como o OSPF são exemplos de protocolos de roteamento utilizados apenas para interconectar roteadores internamente em uma rede, chamados de IGP (Interior Gateway Protocols). Para interconectar redes independentes, utilizamos protocolos chamados EGP (Exterior Gateway Protocols). Um exemplo de protocolo de roteamento EGP é o BGP (Border Gateway Protocol), baseado no algoritmo de *distance vector*.

Saiba mais

O Protocolo OSPF é dos mais utilizados em grandes redes coorporativas. Leia o artigo "O Protocolo OSPF", de Jailton Santos das Neves e Waldeck Ribeiro Torres. Este trabalho fornece, de forma bastante didática, uma descrição simples de como o protocolo de roteamento OSPF trabalha, suas funcionalidades e estrutura, usos e limitações, além de mostrar de que forma ele trabalha com p IPv6. Acesse:

<http://www.midiacom.uff.br/~debora/redes1/pdf/trab042/OSPF.pdf>.

Cada roteador possui uma tabela de roteamento diferente, refletindo sua posição única dentro da internet. À medida que se aproximam do núcleo da rede (*core*), os endereços vão sendo agregados, formando prefixos menores. Os roteadores que estão nas pontas da internet usam muitas rotas *default* e têm suas tabelas de roteamento com poucas rotas específicas. Já os roteadores do núcleo da internet possuem cerca de milhares de linhas de entrada em suas tabelas, sem nenhuma rota *default*.

Resumo

Esta unidade focou as camadas intermediárias e fundamentais para o bom funcionamento da arquitetura de rede: a Camada de Transporte e a Camada de Rede.

A **Camada de Transporte** é responsável por fornecer serviços diretamente aos **processos** da aplicação. Na origem, a mensagem pode ser dividida em **segmentos**, e a Camada de Transporte adiciona ao seu cabeçalho informações importantes para serem tratadas no destino, e a envia à Camada de Rede, que vai direcionar a mensagem até o destino. Lá, no nível de transporte, é responsável por executar suas funções antes de descartar seu cabeçalho e entregar à Camada de Aplicação acima. Dentre as principais funções da Camada de Transporte, destacam-se:

- controle de fluxo fim a fim;
- detecção de erro;
- correção de erro.

Os protocolos principais que atuam nesta camada são **UDP** e **TCP**. Vimos que enquanto o UDP trabalha de forma rápida, e por isso não garante a entrega das mensagens, o TCP trabalha de forma mais lenta, porque trabalha detalhadamente para garantir que todos os segmentos de dados sejam entregues ao destino.

Vamos lembrar algumas características importantes do UDP:

- Oferece serviço não orientado à conexão;
- Formato do segmento simples, permitindo rapidez de processamento;
- Não garante a entrega e, portanto, a transferência de dados não é confiável;
- Não possui controle de fluxo;
- Utilizado em aplicações de voz e vídeo, tolerantes a perdas, aplicações de DNS, entre outros.

Vale recordar também as características do TCP:

- Oferece serviço orientado à conexão;
- Faz a entrega confiável dos segmentos através de sequencialização de dados;
- Tem formato do cabeçalho mais complexo que o UDP, que o torna mais lento;
- Trabalha com janela deslizante e faz controle de fluxo;
- Estabelece conexão antes de enviar os dados e encerra após o envio;
- É utilizado para as aplicações que exigem integridade da mensagem original, sem perdas;
- Permite multiplexação de várias conexões simultaneamente.

A **Camada de Rede** é a responsável por interligar as redes na internet através dos **roteadores**. Vimos o conceito de ***gateway***, que é o ponto de acesso a outras redes.

A Camada de Rede, por meio dos protocolos específicos e dos roteadores, estabelece qual o melhor caminho a ser seguido para se chegar ao destino.

O protocolo principal que atua nesse nível é o **IP**, que encaminha os datagramas da origem ao destino. O **roteamento** também é realizado por meio do endereço IP de cada máquina.

O IP é um protocolo **não confiável**, por não implementar mensagens de confirmação e trabalhar executando o serviço de "melhor esforço", ou seja, faz de tudo para entregar. Entretanto, se é perdido no meio do caminho, não recupera e nem pede retransmissão: o pacote é simplesmente descartado. Nessa ocasião, a Camada de Transporte, no destino, é que deverá solicitar a retransmissão à origem.

O **Datagrama IP** possui 32 *bits* e conhecemos cada um dos seus campos. Ele utiliza a técnica de **fragmentação** quando precisa atravessar redes com capacidades de transferência menor que seu tamanho. Nesse caso, o datagrama é dividido em **fragmentos**. Ao receber o primeiro fragmento, a estação inicia uma contagem de tempo para aguardar o conjunto completo de fragmentos. Se faltar algum, o datagrama é descartado e não pode ser remontado. Os fragmentos nunca são remontados.

Vimos o endereço IPv4, conhecendo as **classes-padrão de endereçamento** e seus *ranges* de endereços IP, e exemplificamos como é possível construir **sub-redes**. O segredo aqui são as **máscaras de sub-rede** que, quando manipuladas, permitem ganhar ou economizar sub-redes.

Os endereços IP são **reservados** para funções específicas, como é o caso dos endereços de *broadcast*, por exemplo. Outros endereços especiais são chamados de **endereços privados** e servem para montar uma rede TCP/IP sem gerar conflitos.

NAT e **DHCP** são duas alternativas para o problema da falta de endereços IPv4. Ambas trabalham na tradução de endereço privado para endereço público.

O **roteamento** é, sem dúvida, a tarefa mais importante da Camada de Rede. Ele funciona por meio de uma tabela de roteamento que ele contém. Basicamente, a informação das **rotas**, ou seja, dos prefixos de rede, e o endereço IP do próximo roteador no caminho. A **tabela de roteamento** sempre aponta para roteadores que podem ser alcançados diretamente, isto é, que estão conectados na mesma sub-rede.

Os roteadores constroem suas tabelas de roteamento automaticamente, de forma dinâmica, trocando informações entre si por meio de **protocolos de roteamento.** Para encontrar o melhor caminho para o destino de um datagrama através da rede, os roteadores utilizam **algoritmos de roteamento.**

O processo de encontrar o próximo passo após uma mudança na rede é denominado **convergência.**

Existem dois tipos básicos de algoritmo de roteamento que devem ser considerados:

- *Distance vector*: baseado no número de saltos na rede (*hops*), como o RIP. Sua métrica é simples, é rápido de convergir, mas possui limitações.

- *Link state*: é baseado no estado do enlace e sua métrica considera diversos parâmetros da rede. Um exemplo é o OSPF, bastante difundido em redes de grande porte.

BGP é um exemplo de protocolo de roteamento externo, do tipo EGP, e serve para interconectar redes independentes.

Exercícios

Questão 1 (adaptada de: Forouzan, Behrouz A. *Comunicação de dados e redes de computadores*). Qual o papel da Camada de Transporte?

A) Encaminhar o pacote.
B) Empacotamento.
C) Controle de erros.
D) Controle de acesso.
E) Fornecer serviços.

Resposta correta: alternativa E.

Análise das alternativas:

A) Alternativa incorreta.

Justificativa:

A Camada de Rede é responsável por encaminhar os pacotes ao seu destino e lá entregá-los à Camada de Transporte.

B) Alternativa incorreta.

Justificativa:

O empacotamento é uma responsabilidade da Camada de Enlace. A Camada de Enlace de dados divide o fluxo de *bits* recebidos da Camada de Rede em unidades de dados gerenciáveis denominados *frames*.

C) Alternativa incorreta.

Justificativa:

O controle de erros é uma responsabilidade da Camada de Enlace. A Camada de Enlace de dados acrescenta confiabilidade à Camada Física, adicionando mecanismos para detectar e retransmitir *frames* danificados ou perdidos. Ela também usa mecanismos para reconhecer *frames* duplicados. Normalmente, o controle de erros é obtido por meio de um trailer acrescentado ao final do quadro.

D) Alternativa incorreta.

Justificativa:

O controle de acesso é uma responsabilidade da Camada de Enlace. Quando dois ou mais dispositivos estiverem conectados ao mesmo *link*, serão necessários protocolos da Camada de Enlace de dados para determinar qual dispositivo assumirá o controle do *link* em dado instante.

E) Alternativa correta.

Justificativa:

A Camada de Transporte, camada central da pilha de protocolos, desempenha o papel fundamental de fornecer serviços de comunicação diretamente aos processos de aplicação, que rodam em hospedeiros diferentes.

Questão 2 (Forouzan, Behrouz A. *Comunicação de dados e redes de computadores*). Considere as afirmativas abaixo:

I) Um endereço IPv4 é um endereço de 32 *bits* que define de forma única e universal a conexão de um dispositivo (por exemplo, um computador ou um roteador) à Internet.

II) Os endereços IPv4 são exclusivos no sentido de que cada endereço define uma, e somente uma, conexão com a internet. Dois dispositivos na internet jamais podem ter o mesmo endereço ao mesmo tempo. Pelo uso de algumas estratégias, um endereço pode ser designado a um dispositivo por determinado período e, em seguida, retirado e atribuído a um outro dispositivo.

III) O espaço de endereços do IPv4 é igual a 2^{32}, ou seja, 4.294.967.296.

Está(ão) correta(s) a(s) afirmativa(s):

A) Somente a I.
B) I e II.
C) II e III.
D) I e III.
E) I, II e III.

Resolução desta questão na Plataforma.