Unidade II

5 PADRÕES E PROTOCOLOS DE CAMADA DE REDE – PARTE 1

O usuário que acessa uma aplicação conectada à internet não tem e não deve ter a preocupação sobre como os seus dados chegarão até o destino. O caminho percorrido pelas mensagens não é um problema do usuário, que está na camada de aplicação, e muito menos da camada de transporte.

Determinar o trajeto da mensagem que sai de um host e chega até outro é uma responsabilidade da camada de rede que, utilizando diversos processos, encontra os melhores caminhos a serem trilhados pelos pacotes. Tudo isto ocorre considerando um esquema de endereçamento lógico, que na maioria os casos é determinado pelo tão conhecido IP, além dos protocolos de roteamento executados nesta camada.

Como há uma grande complexidade associada as funções e protocolos da camada de rede, teremos este tópico e o próximo dedicado a ela. Neste tópico vamos abordar as funcionalidades da camada de rede e os serviços por ela oferecidos, dando um destaque especial para o roteamento e os seus algoritmos.

5.1 A camada de rede

5.1.1 Processos da camada de rede

Para favorecer a entrega de pacotes host-a-host, a camada de rede executa os seguintes processos: encapsulamento; desencapsulamento; roteamento; encaminhamento; controle de erros (FOROUZAN; MOSHARRAF, 2013).

O primeiro processo é o encapsulamento, também chamado de empacotamento, que consiste em receber o segmento da camada de transporte e encapsulá-lo em um pacote, em que teremos um cabeçalho contendo informações importantes (por exemplo, os endereços lógicos de origem e destino) para encaminhamento e roteamento adequado.

Lembrete

A camada de rede tem como PDU (Unidade de Dados de Protocolo) o pacote.

O desencapsulamento é o segundo processo executado pela camada de rede e ocorre no destino, quando o host recebe o pacote e retira dele a carga útil (segmento), entregando-o para a camada de transporte. É o processo inverso ao do encapsulamento.

O roteamento é o terceiro processo prestado pela camada de rede e consiste na determinação do melhor caminho para um pacote. O objetivo principal deste processo é encontrar a melhor rota (relacionadas em uma tabela), que pode ser definida a partir de um algoritmo de roteamento ou estabelecida de forma estática por um administrador da rede (FOROUZAN; MOSHARRAF, 2013).

Lembrete

O roteador é um dispositivo intermediário de camada de rede e atua no processo de roteamento de pacotes.

O encaminhamento é o quarto processo, executado por um dispositivo de camada 3 (que pode ser um roteador) e determina por qual interface o pacote será encaminhado. Contudo, é importante frisar que o encaminhamento só ocorre quando há uma tabela de roteamento criada dinâmica ou estaticamente. Portanto, o encaminhamento e roteamento estão intimamente ligados. A figura a seguir ilustra esses dois processos.

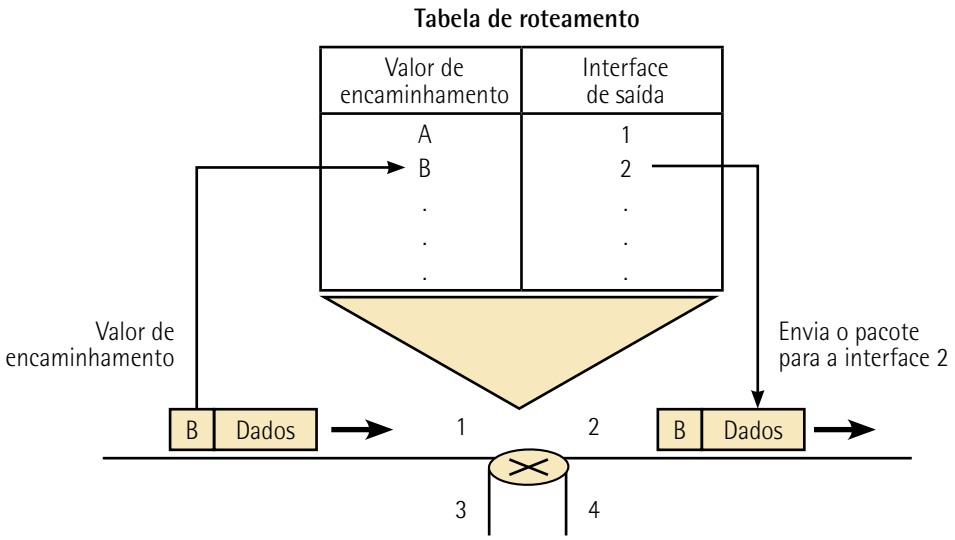


Figura 72 – Encaminhamento e roteamento

Adaptado de: Forouzan e Mosharraf (2013, p. 242).

É importante distinguir roteamento de encaminhamento. Tanenbaum, Feamster e Wetherall (2021, p. 461) mencionam que:

> Algumas vezes, é útil fazer distinção entre o roteamento, que é a tomada de decisão sobre quais rotas utilizar, e o encaminhamento, que acontece quando um pacote chega. Podemos imaginar que um roteador tem dois processos internamente. Um deles trata cada pacote que chega, procurando a linha de saída que será usada em sua tabela de roteamento. Esse processo é o encaminhamento. O outro processo é responsável pelo preenchimento e pela atualização das tabelas de roteamento. É nele que o algoritmo entra em cena.

O próximo processo é o controle de erros. Embora pareça uma tarefa executada apenas pela camada de enlace (conforme já visto), este processo é executado com primazia tanto na camada de transporte como na camada de rede, em um contexto diferente. O primeiro detalhe importante é que a maior parte dos protocolos de camada 3 (o IP é um deles) trabalha com controle de erros no cabeçalho do pacote, e não no campo de dados (em que está situado o segmento). O segundo detalhe é que há um protocolo na camada de rede chamado ICMP, que é responsável pelas mensagens de erro na internet e grande auxiliar do IP (FOROUZAN; MOSHARRAF, 2013).

5.1.2 Serviços oferecidos à camada de transporte

A lógica da arquitetura de redes em camadas estabelece que uma camada inferior sempre presta serviços para a camada superior. Assim, a camada de rede oferece serviços à camada de transporte, e são de dois tipos diferentes: orientado à conexão e não orientado à conexão (TANENBAUM; FEAMSTER; WETHERALL, 2021).

Se considerarmos a internet, o serviço oferecido à camada de transporte sempre é não orientado à conexão. O motivo é que a internet foi criada a partir da utilização do IP, considerado protocolo do melhor esforço e sem conexão, fazendo com que a camada de rede se preocupe apenas com o envio e recepção de pacotes – deixando todo controle de fluxo, controle de congestionamento e controle de erros para as camadas superiores (MAIA, 2013).

No entanto, é bom acrescentar que, em um passado não muito distante (décadas de 1970 e 1980), existia uma série de protocolos de camada de rede que operavam orientados à conexão. Um desses protocolos era o X25 (TANENBAUM; FEAMSTER; WETHERALL, 2021).

5.1.3 Abordagem de datagramas e abordagem de circuitos virtuais

Quando o serviço oferecido à camada de transporte é não orientado à conexão, temos uma abordagem de datagramas. A figura a seguir apresenta uma topologia de rede operando esse tipo de abordagem.

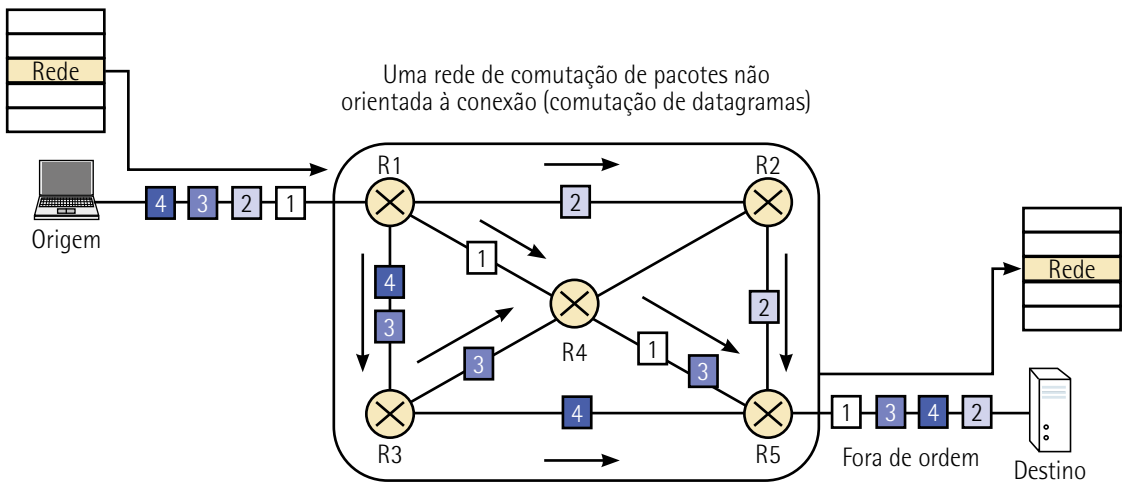


Figura 73 – Abordagem de datagrama

Adaptado de: Forouzan e Mosharraf (2013, p. 244).

Na abordagem de datagrama não temos um caminho prefixado por onde os pacotes (também chamados de datagrama) trafegam. Eles podem tomar os mais diversos trajetos, e todo o roteamento/encaminhamento é feito a cada salto dado em um roteador (MAIA, 2013).

A figura a seguir apresenta a ideia desse encaminhamento. Nela temos um roteador com quatro interfaces: ele recebe um pacote pela interface 1; verificado o endereço lógico de destino, consulta a tabela de roteamento e depois procede com o encaminhamento do pacote pela interface 2.

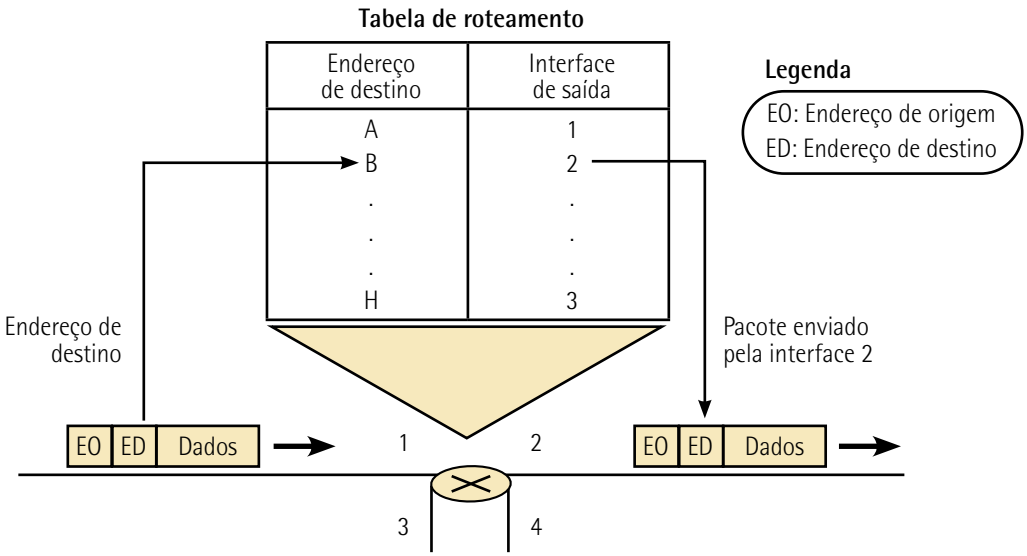


Figura 74 – Roteamento e encaminhamento na abordagem de datagrama

Adaptado de: Forouzan e Mosharraf (2013, p. 245).

Observação

A abordagem de datagrama é a mais comum na internet.

Quando o serviço oferecido à camada de transporte é orientado à conexão, temos uma abordagem de circuitos virtuais. A principal característica dessa abordagem reside no estabelecimento de uma conexão virtual para a transmissão de pacotes. Desta forma, os pacotes (todos eles gerados por um fluxo de dados) seguem o mesmo caminho (MAIA, 2013).

A figura a seguir apresenta uma topologia funcionando em abordagem de circuitos virtuais.

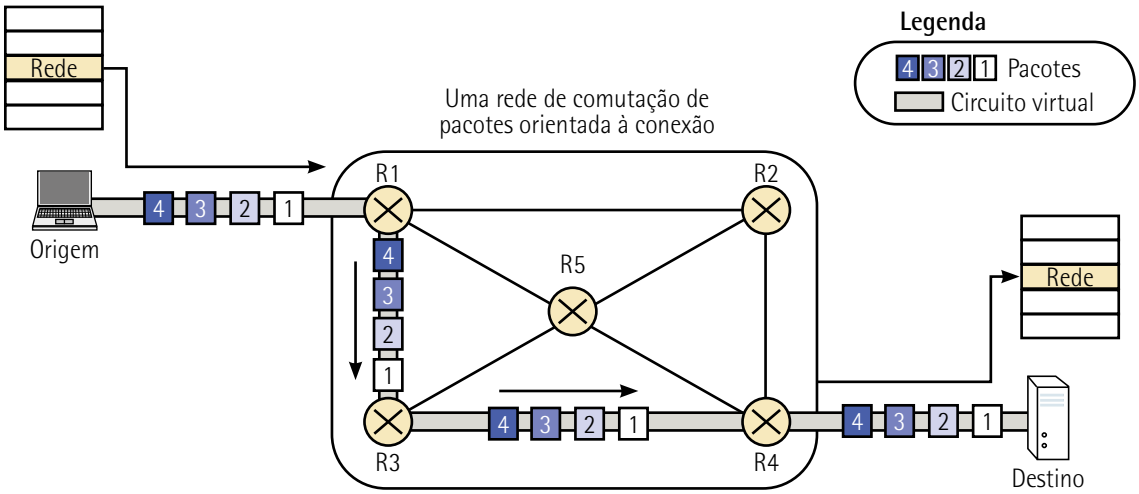


Figura 75 – Abordagem de circuitos virtuais

Adaptado de: Forouzan e Mosharraf (2013, p. 245).

Na abordagem de circuitos virtuais, os pacotes recebem, em seus cabeçalhos, identificadores do circuito virtual (também chamados de rótulos), que servem de base para o processo de roteamento. Compondo esta forma de trabalho da camada de rede, temos três fases: configuração do circuito virtual (fase 1); transferência de dados (fase 2); e finalização do circuito virtual (fase 3) (FOROUZAN; MOSHARRAF, 2013).

A figura a seguir apresenta esse processo de roteamento na abordagem de circuitos virtuais.

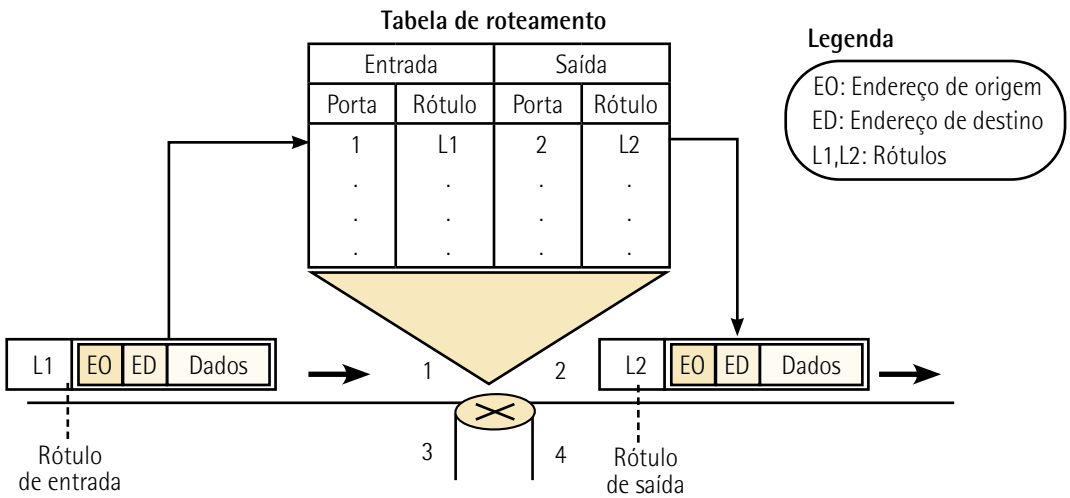


Figura 76 – Roteamento e encaminhamento na abordagem de circuitos virtuais

Adaptado de: Forouzan e Mosharraf (2013, p. 246).

O quadro a seguir apresenta uma comparação entre as abordagens de datagrama e de circuito virtual.

Quadro 10 – Comparação entre as abordagens de datagrama e de circuitos virtuais

Questão	Rede de datagramas	Rede de circuitos virtuais
Configuração de circuitos	Desnecessária	Obrigatória
Endereçamento	Cada pacote contém os endereços completos de origem e de destino	Cada pacote contém um pequeno número do circuito virtual
Informações sobre o estado	Os roteadores não armazenam informações sobre o estado das conexões	Cada circuito virtual requer espaço em tabelas de roteadores por conexão
Roteamento	Cada pacote é roteado independentemente	A rota é escolhida quando o circuito virtual é estabelecido e todos os pacotes a seguem
Efeito de falhas no roteador	Nenhum, com exceção dos pacotes perdidos durante a falha	Todos os circuitos virtuais que tiverem passado pelo roteador que apresentou o defeito serão encerrados
Qualidade de serviço	Difícil	Fácil, se for possível alocar recursos suficientes com antecedência para cada circuito virtual
Controle de congestionamento	Difícil	Fácil, se for possível alocar recursos suficientes com antecedência para cada circuito virtual

Adaptado de: Tanenbaum, Feamster e Wetherall (2021, p. 261).

Saiba mais

Para conhecer um pouco mais sobre a implementação da abordagem de datagramas e da abordagem de circuitos virtuais, leia o capítulo 5, "A camada de rede", do livro a seguir:

TANENBAUM, A; FEAMSTER, N; WETHERALL, D. *Redes de computadores*. 5. ed. Rio de Janeiro: Pearson Prentice Hall, 2021. p. 231-320.

5.1.4 Qualidade de serviço, desempenho e controle de congestionamento na camada de rede

O conceito de QoS (*Quality of Service* – Qualidade de Serviço) foi se estabelecendo na camada de rede à medida que os usuários necessitavam de uma grande atenção para o trafego de voz e imagem, principalmente em tempo real.

Na abordagem de circuitos virtuais, a QoS é um pouco mais garantida, quando comparada com a abordagem de datagramas. Ao utilizar protocolos de melhor esforço na camada de rede, como o IP, o trabalho com QoS vai se configurando como um desafio – mas não impossível – quando trabalhamos os aspectos que norteiam o desempenho da camada 3, principalmente quando consideramos a aplicação

a ser utilizada. Com a evolução do IP da versão 4 para a versão 6, percebemos um melhor atendimento das questões voltadas para QoS.

O quadro a seguir apresenta uma relação entre o desempenho da camada de rede em algumas aplicações. Os parâmetros considerados são: largura de banda, atraso, flutuação (variação do atraso) e perda.

Quadro 11 – Requisitos de desempenho na camada de rede

Aplicação	Largura de banda	Atraso	Flutuação	Perda
E-mail	Baixa	Baixo	Baixa	Média
Compartilhamento de arquivos	Alta	Baixo	Baixa	Média
Acesso à web	Média	Médio	Baixa	Média
Login remoto	Baixa	Médio	Média	Média
Áudio por demanda	Baixa	Baixo	Alta	Baixa
Vídeo por demanda	Alta	Baixo	Alta	Baixa
Telefonia	Baixa	Alto	Alta	Baixa
Videoconferência	Alta	Alto	Alta	Baixa

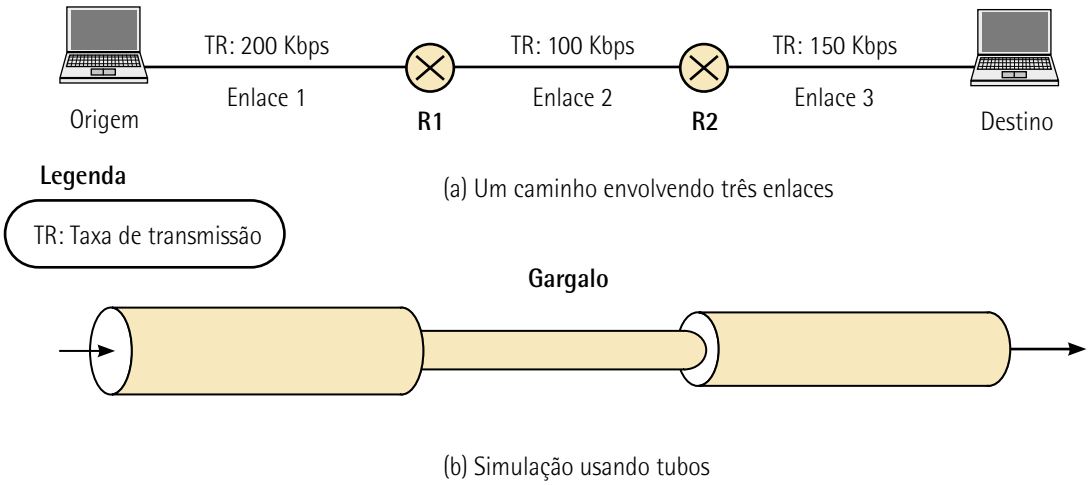
Adaptado de: Tanenbaum, Feamster e Wetherall (2021, p. 275).

Forouzan e Mosharraf (2013) mencionam apenas três parâmetros que influenciam no desempenho da camada de rede – e, consequentemente, na QoS. São eles: atraso, vazão e perda de pacotes.

Os atrasos expressam a ausência de respostas instantâneas no processo de comunicação e podem ser divididos de quatro formas: atraso de transmissão devido a inserção de bits um a um na linha; atraso de propagação oriundo do tempo que o bit leva para cruzar todo o meio de transmissão; atraso de processamento do roteador, considerando o tempo que ele leva para executar o controle de erro e envio do pacote pela próxima interface; e atraso na fila de entrada da interface do roteador (FOROUZAN; MOSHARRAF, 2013).

A vazão é o segundo parâmetro que influencia o desempenho da camada de rede e expressa a quantidade de bits que podem ser transmitidos em um segundo. Alguns autores chamam a vazão de velocidade, de largura de banda ou taxa de transmissão. A vazão pode sofrer variações ao longo de todo o percurso do bit (FOROUZAN; MOSHARRAF, 2013; TANENBAUM; FEAMSTER; WETHERALL, 2021).

A figura a seguir apresenta essa ideia de vazão envolvendo variações ao longo do canal de comunicação:



(a) Um caminho envolvendo três enlaces

(b) Simulação usando tubos

Figura 77 – Vazão

Adaptado de: Forouzan e Mosharraf (2013, p. 249).

Como último parâmetro, há a perda de pacotes, oriundos dos problemas de fila no *buffer* de entrada dos roteadores. Sobre a perda de pacotes, Forouzan e Mosharraf (2013, p. 252) mencionam que:

> Quando um roteador recebe um pacote enquanto ele processa outro, o pacote recebido deve ser armazenado em um buffer de entrada e esperar pela sua vez. Um roteador, contudo, tem um buffer de entrada de tamanho limitado. Pode chegar um momento em que o buffer esteja cheio, de modo que o próximo pacote a chegar deve ser descartado. A consequência da perda de pacotes na camada de rede da internet é que o pacote precisa ser reenviado, o que, por sua vez, pode criar um tráfego excessivo e causar a perda de mais pacotes. Existem diversos estudos teóricos na área da teoria de filas que buscam evitar a sobrecarga das filas e a perda de pacotes.

Somado a todas essas questões de desempenho, temos outro problema enfrentado na camada de rede: o controle de congestionamento. Um detalhe importante é que o controle de congestionamento não é tratado de modo explícito na internet, porque o IP é um protocolo do melhor esforço, tornando-se uma responsabilidade da camada de transporte (quando orientada à conexão). Contudo, quando está em uso a abordagem de circuitos virtuais na camada de rede, o controle de congestionamento é uma necessidade.

Os parâmetros de desempenho, vazão e atraso estão intimamente relacionados ao controle de congestionamento, conforme pode ser visto na figura a seguir.

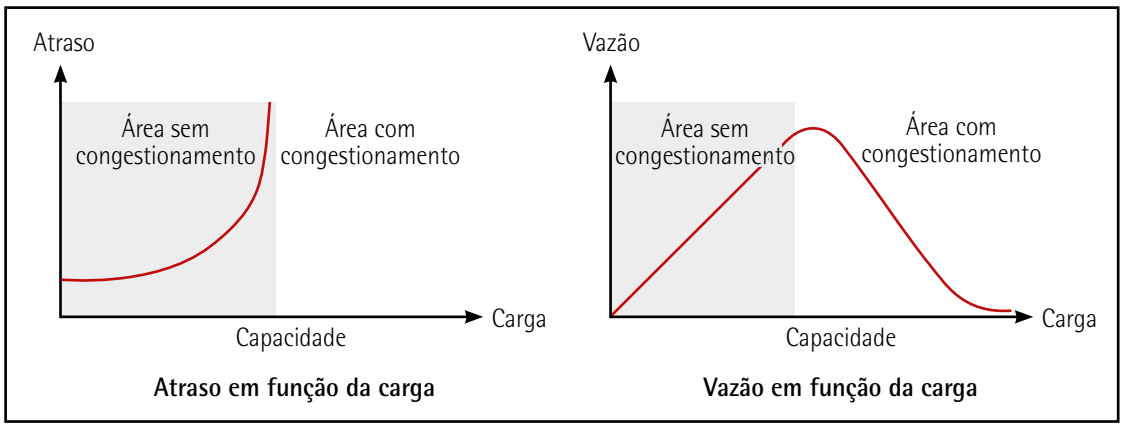


Figura 78 – Vazão e atraso relacionados ao controle de congestionamento

Adaptado de: Forouzan e Mosharraf (2013, p. 252).

A leitura adequada do primeiro gráfico (capacidade × atraso) revela que, quando há uma carga inferior à capacidade, o atraso é ínfimo e não temos congestionamento; quando a carga está acima da capacidade, porém, o atraso sobe exponencialmente e o congestionamento é extremamente perceptível (FOROUZAN; MOSHARRAF, 2013).

Fazendo a leitura do segundo gráfico (capacidade × vazão), percebemos que, quando a carga é inferior à capacidade, a vazão sobe a níveis considerados altos e sem congestionamento. À medida que a carga se torna maior que a capacidade, a vazão reduz vertiginosamente e o congestionamento é inevitável (FOROUZAN; MOSHARRAF, 2013).

Forouzan e Mosharraf (2013) mencionam que, para tratá-lo bem, é possível controlar o congestionamento por meio de ferramentas agrupadas em duas categorias: controle em malha aberta e controle em malha fechada.

O controle em malha aberta prevê uma série de ações que evitam o congestionamento antes que ele aconteça. As ações que compõem esse controle são: retransmissão de pacotes perdidos ou corrompidos; janelamento que limita o número de pacotes enviados em um determinado período; confirmações de pacotes recebidos; e políticas adequadas de descarte a fim de evitar o congestionamento (FOROUZAN; MOSHARRAF, 2013).

O controle em malha fechada é considerado reativo porque atua depois de o congestionamento ter acontecido. Dentre as ações de controle desta categoria, merecem destaque a contrapressão (suspensão da recepção de dados em um nó congestionado) e o pacote de bloqueio (envio de pacote para o host que gerou o fluxo de dados com o intuito de informar a existência de congestionamento). A figura a seguir apresenta uma ilustração sobre a contrapressão e o pacote bloqueio (FOROUZAN; MOSHARRAF, 2013).

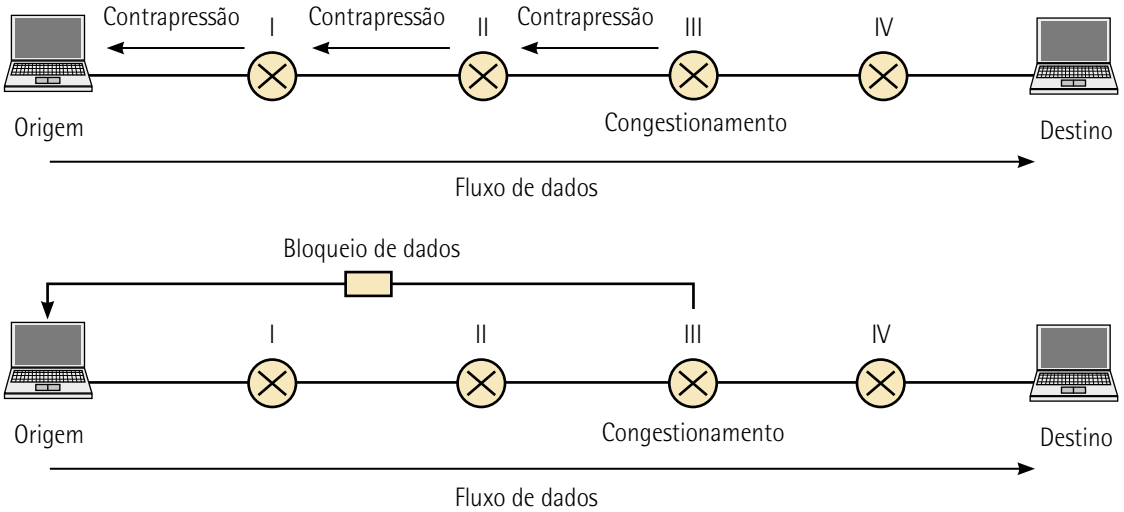


Figura 79 – Contrapressão e pacote de bloqueio

Adaptado de: Forouzan e Mosharraf (2013, p. 255).

Saiba mais

Para saber um pouco mais sobre o congestionamento e as técnicas utilizadas para o controle de tráfego, leia a seção 5.3, "Controle de tráfego na camada de rede", do livro a seguir:

TANENBAUM, A; FEAMSTER, N; WETHERALL, D. *Redes de computadores*. Tradução: Daniel Vieira. 5. ed. Rio de Janeiro: Pearson Prentice Hall, 2021.

Complementando todas essas necessidades de controlar melhor o congestionamento, o desempenho e o tráfego nas redes de computadores, Maia (2013) menciona alguns mecanismos para implementação do QoS na camada de rede. São eles: controle de admissão, reserva de recursos, políticas de escalonamento e modelagem do tráfego.

Especificamente na internet há um problema de garantia do QoS, e muitos esforços têm sido empenhados para garantir a entrega mais adequada de pacotes, dos quais mais se destacam os serviços integrados e os serviços diferenciados.

Saiba mais

Para conhecer um pouco mais sobre mecanismos de implementação de QoS, leia a seção 6.9, "Qualidade de serviço", do livro a seguir:

MAIA, L. P. *Arquitetura de redes de computadores*. 2. ed. Rio de Janeiro: LTC, 2013.

5.2 Algoritmos de roteamento e roteadores

5.2.1 Roteamento

O roteamento é um dos processos executados na camada de rede por um dispositivo específico chamado roteador, que determina o melhor caminho para um pacote. Tudo ocorre a partir da execução do algoritmo de roteamento para criar a tabela de roteamento (MAIA, 2013; TANENBAUM; FEAMSTER; WETHERALL, 2021).

A finalidade última do roteamento é fazer com que os pacotes cheguem ao seu destino. As redes de computadores se comportam como um conjunto de estradas, em que diversos caminhos permitem que os carros trafeguem e cheguem ao seu destino. A figura a seguir apresenta esta analogia entre o roteamento e as alternativas de estradas para os carros percorrerem e chegarem aos seus destinos.

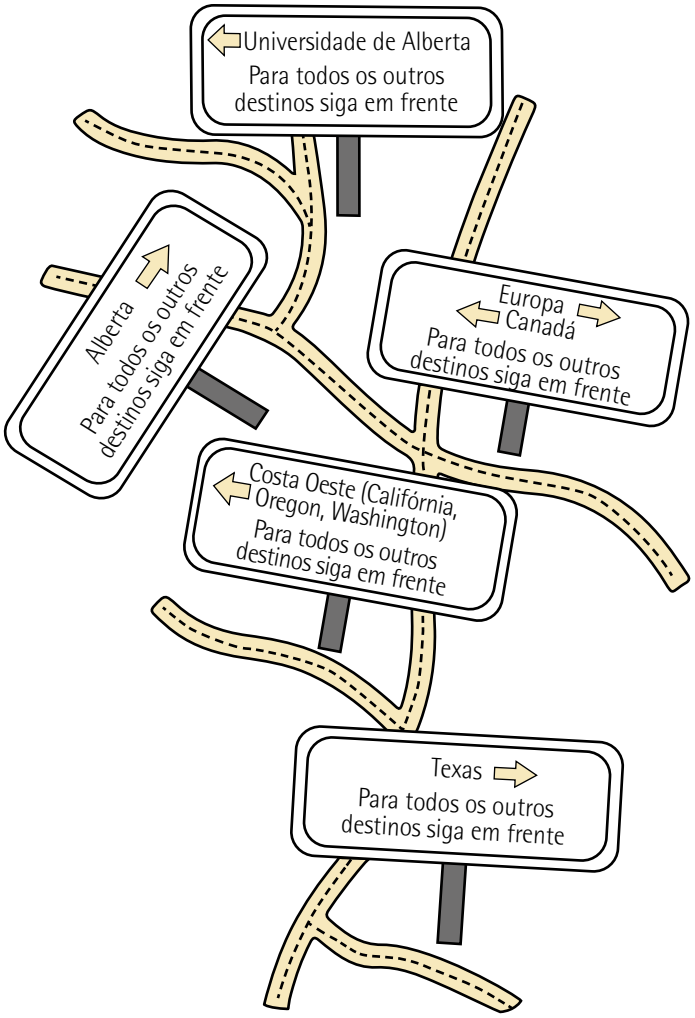


Figura 80 – Comparação entre o roteamento e as estradas

Adaptado de: Fitzgerald e Dennis (2010).

Partindo para um entendimento mais prático do roteamento nas redes de computadores, a figura a seguir apresenta uma topologia contendo três roteadores (R1, R2 e R3) e quatro redes específicas (20.0.0.0; 30.0.0.0; 40.0.0.0; 50.0.0.0). Para cada um dos roteadores da topologia temos uma tabela de roteamento contendo a relação entre as interfaces e os endereços lógicos de camada 3.

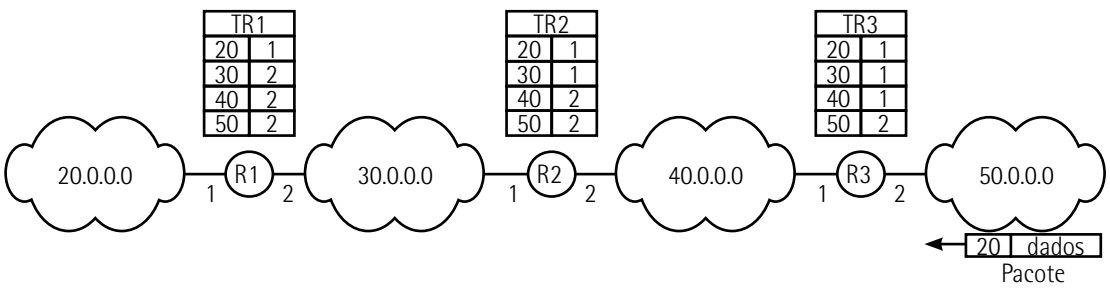


Figura 81 – Roteamento

Adaptado de: Maia (2013, p. 170).

O roteamento pode ser unicast (o pacote é roteado de um host de origem para apenas um host de destino), multicast (o pacote é roteado de um host de origem para um grupo de hosts de destino) e broadcast (o pacote é roteado de um host de origem para todos os hosts de uma rede) (MAIA, 2013).

O roteador opera com a tabela de roteamento, que fornece as informações completas sobre como encaminhar adequadamente os pacotes. A construção da tabela pode se dar de forma estática, em que as rotas são criadas manualmente pelo administrador da rede. Também é possível, e mais comum, que as tabelas de roteamento sejam criadas pelos protocolos de roteamento, que trabalham baseados em um algoritmo.

5.2.2 Roteador

O roteador é, com certeza, o dispositivo de camada 3 mais importante. Ele funciona como um dispositivo computacional e é composto de quatro componentes: portas de entrada, portas de saída, processador de roteamento e malha de comutação. A figura a seguir ilustra um roteador com os seus componentes.

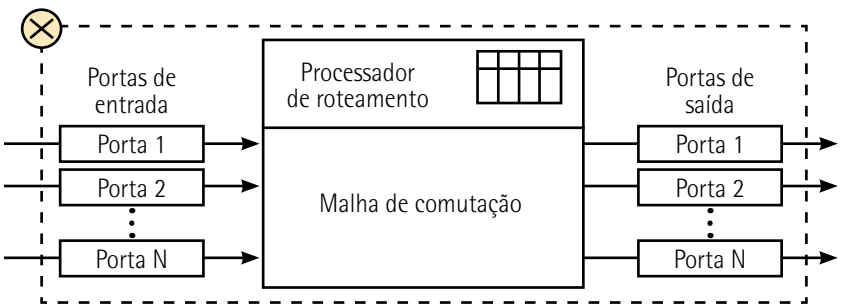


Figura 82 – Roteador e seus componentes

Adaptado de: Forouzan e Mosharraf (2013, p. 256).

O processador de roteamento opera apenas na camada de rede e, baseado no esquema de endereçamento lógico, efetua o encaminhamento de pacotes. Todo o processamento da camada física (nível de bits) e da camada de enlace (nível de quadros) é executado nas portas de entrada e de saída, conforme pode ser visto na figura a seguir.

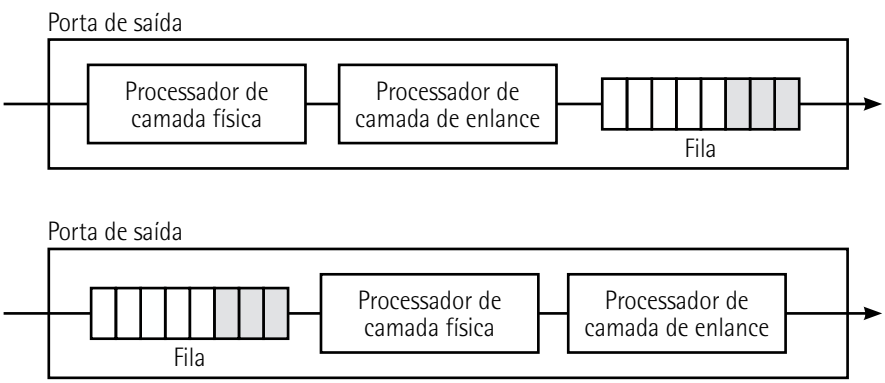


Figura 83 – Porta de entrada e porta de saída do roteador

Adaptado de: Forouzan e Mosharraf (2013, p. 257).

O processador de roteamento executa o algoritmo de roteamento e indica o endereço do próximo salto que o pacote precisa dar, além da porta de saída. As malhas de comutação, por sua vez, são utilizadas para mover o pacote de uma porta de entrada para uma porta de saída. A figura a seguir apresenta a ideia de uma malha de comutação.

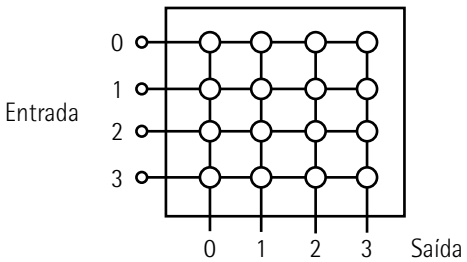


Figura 84 – Malha de comutação

Adaptado de: Forouzan e Mosharraf (2013, p. 257).

5.2.3 Algoritmos de roteamento

No cerne do funcionamento da camada de rede, encontramos o algoritmo de roteamento. Segundo Tanenbaum, Feamster e Wetherall (2021, p. 420):

> 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 rede internamente utilizar datagramas, essa decisão deverá ser tomada mais uma vez para cada pacote de dados recebido, pois a melhor rota pode ter sido alterada desde a última vez. Se a rede internamente utilizar circuitos virtuais, 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.

Os algoritmos de roteamento podem ser agrupados em adaptativos (adaptam as decisões às condições e contexto de tráfego e conectividade) e não adaptativos (não se adaptam aos indicadores ou questões relacionados ao tráfego e conectividade). Os algoritmos não adaptativos produzem o roteamento estático, que normalmente é configurado/estabelecido manualmente por um administrador de redes por meio de rotas fixas, sem que haja quaisquer alterações dinâmicas. Já os algoritmos adaptativos produzem o roteamento dinâmico, que é sempre provido por um protocolo de roteamento (TANENBAUM; FEAMSTER; WETHERALL, 2021).

Segundo Maia (2013), os algoritmos de roteamento dispõem das seguintes características:

- **Seleção do melhor caminho**: os algoritmos de roteamento sempre procuram o melhor caminho para o pacote trilhar conforme regras e métricas estabelecidas.

- **Rápida convergência**: os algoritmos de roteamento, principalmente os adaptativos, procuram estabelecer rapidamente as tabelas de roteamento para os roteadores, favorecendo rapidamente a conectividade das redes.

- **Oferecimento de robustez**: os algoritmos de roteamento funcionam diante de adversidades, como problemas de congestionamento, falhas de hardware, dentre outros.

- **Oferecimento de escalabilidade**: os algoritmos de roteamento, principalmente os adaptativos, assimilam os aumentos de tamanho das redes (principalmente a inserção de novos roteadores).

- **Eficiência no consumo de recursos**: os algoritmos de roteamento cumprem seus objetivos, consumindo apenas a medida eficiente de recursos de hardware.

Tanenbaum, Feamster e Wetherall (2021) mencionam ainda a existência de propriedades desejáveis em um algoritmo de roteamento. São elas: exatidão nas informações sobre rotas; simplicidade nos processos para evitar sobrecarga de hardware; máxima robustez para funcionar diante de instabilidades; estabilidade na convergência para um conjunto de rotas; equidade na escolha das rotas; e eficiência no uso de recursos.

Observação

Perceba que as propriedades desejáveis a um algoritmo de roteamento estão alinhadas com as suas características.

Existem diversos algoritmos de roteamento que consideram os mais variados parâmetros possíveis na construção das rotas. Alguns algoritmos consideram, por exemplo, a quantidade de saltos entre a origem e o destino para determinar a melhor rota para o pacote. Outros algoritmos consideram parâmetros como largura de banda, atraso e outras métricas para determinar as suas rotas.

Saiba mais

Para conhecer um pouco mais sobre algoritmos de roteamento, leia a seção 5.2, "Algoritmos de roteamento em uma única rede", do livro a seguir:

TANENBAUM, A; FEAMSTER, N; WETHERALL, D. *Redes de computadores*. Tradução: Daniel Vieira. 5. ed. Rio de Janeiro: Pearson Prentice Hall, 2021.

5.2.4 Protocolos de roteamento

Os protocolos de roteamento apresentam-se como um conjunto de regras que regem o processo de roteamento. Todo o trabalho se baseia na execução de algoritmos de roteamento adaptativos que propiciam o roteamento dinâmico de pacotes em uma rede (TANENBAUM; FEAMSTER; WETHERALL, 2021).

Observação

Protocolos de roteamento não são a mesma coisa que protocolos roteáveis. Os protocolos roteáveis são aqueles que provêm encapsulamento, endereçamento, fragmentação e demais atividades da camada de rede, enquanto os protocolos de roteamento são responsáveis unicamente pelo processo de roteamento.

Podemos classificar os protocolos de roteamento de algumas formas. Uma delas considera a abrangência do AS (*Autonomous System* – Sistema Autônomo), definido por Oliveira e Melo (2021) como um conjunto de redes e roteadores comandado por um domínio administrativo. Assim, temos, quanto ao AS:

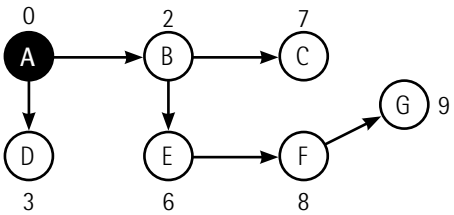
- **Protocolo de gateway interno, protocolo de roteamento local ou *Interior Gateway Protocol*** (IGP): caracterizado pela atuação do algoritmo de roteamento dentro de um único AS.

- **Protocolo de gateway externo, protocolo de roteamento global ou *Exterior Gateway Protocol*** (EGP): caracterizado pela atuação do algoritmo de roteamento dentro de múltiplos ASs.

Outra forma de classificar os protocolos de roteamento é de acordo com a métrica utilizada pelo algoritmo de roteamento. Assim, eles podem ser: protocolos de roteamento por vetor de distância e protocolos de roteamento por estado de enlace (FOROUZAN; MOSHARRAF, 2013).

Os protocolos de roteamento por vetor de distância (*distance vector*) fundamentam-se no algoritmo Belmann-Ford, que trabalha com a criação de uma "árvore de menor custo" para a determinação do melhor caminho. Assim, cada roteador monta, por meio do algoritmo, a sua tabela de roteamento que apresenta o custo para chegar às redes disponíveis.

Um detalhe importante dos protocolos de roteamento por vetor de distância é que as informações da tabela de roteamento são periodicamente trocadas entre os vizinhos de forma que as "árvores de menor custo" tenham dados atualizados sobre o roteamento. A figura a seguir apresenta um exemplo simplificado de uma árvore para um determinado nó A (que pode ser um roteador) e o seu respectivo vetor de distância.



Árvore para o nó A

	A
A	0
B	2
C	7
D	3
E	6
F	8
G	9

Vetor de distâncias para o nó A

Figura 85 – Árvore para o nó A e o seu vetor de distância

Adaptado de: Forouzan e Mosharraf (2013, p. 299).

Dentre os principais protocolos de roteamento por vetor de distância, encontramos o RIP (*Routing Information Protocol* – Protocolo de Informações de Roteamento). O RIP é um dos protocolos de roteamento mais conhecidos no mundo das redes de computadores. Ele utiliza como métrica de menor custo a contagem de saltos, que representa a quantidade de redes pelas quais um pacote transitará até chegar ao seu destino. A figura a seguir apresenta um exemplo de contagem de saltos do RIP (FILIPPETTI, 2017).

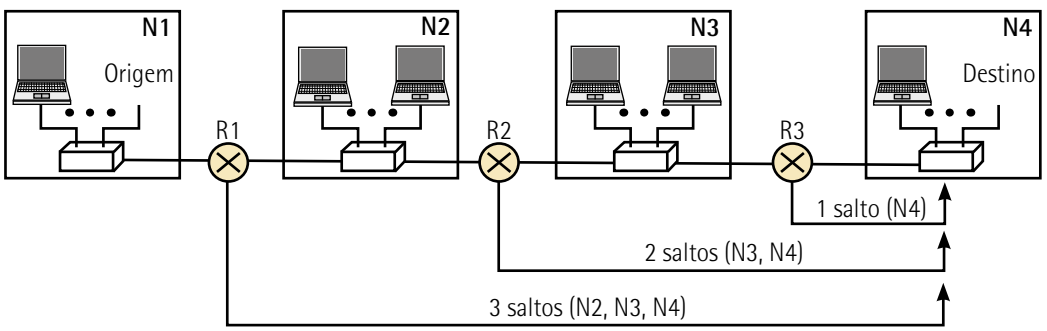


Figura 86 – Contagem de saltos no RIP

Adaptado de: Forouzan e Mosharraf (2013, p. 312).

As tabelas de roteamento em roteadores que operam com o RIP apresentam o número de saltos a serem dados até que se chegue na rede de destino. Considerando o mesmo exemplo descrito na topologia da figura 86, vemos na tabela a seguir as tabelas de roteamento dos roteadores R1, R2 e R3.

Tabela 1 – Tabelas de roteamento dos roteadores R1, R2 e R3 utilizando o RIP

Roteamento de R1			Roteamento de R2			Roteamento de R3		
Rede de destino	Próximo roteador	Custo em saltos	Rede de destino	Próximo roteador	Custo em saltos	Rede de destino	Próximo roteador	Custo em saltos
N1	–	1	N1	R1	2	N1	R2	3
N2	–	1	N2	–	1	N2	R2	2
N3	R2	2	N3	–	1	N3	–	1
N4	R2	3	N4	R3	2	N4	–	1

Adaptado de: Forouzan e Mosharraf (2013, p. 313).

Para favorecer a construção das tabelas de roteamento, o RIP trabalha com a troca integral, no intervalo de 25 a 35 segundos, da tabela de roteamento entre vizinhos de forma que todos conheçam bem a situação da sua vizinhança em matéria de conectividade. O RIP também especifica temporizadores de validade de rota e retirada da mesma da tabela de roteamento (nos casos da rota ser inválida).

Observação

Os temporizadores do RIP podem ser configurados da maneira mais adequada possível pelo administrador de redes em cada um dos roteadores.

Outro protocolo de roteamento por vetor de distância é o BGP (*Border Gateway Protocol* – Protocolo de Gateway de Borda). O BGP é um EGP utilizado na internet para trocar informações de roteamento entre sistemas autônomos. Sua configuração é bem mais complexa do que os IGP e raramente é utilizado em empresas com poucos sistemas autônomos (OLIVEIRA; MELO, 2021).

A figura a seguir apresenta uma topologia utilizando o BGP.

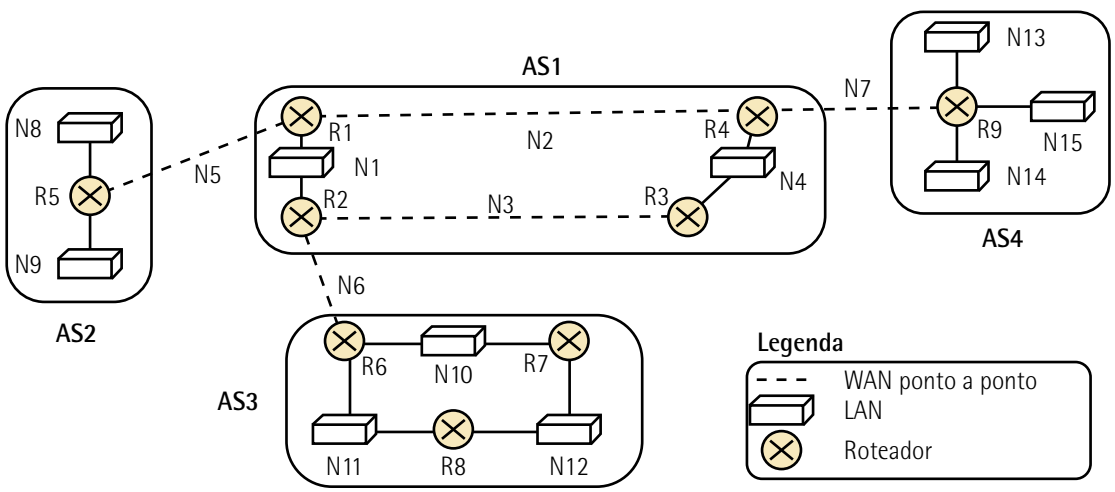


Figura 87 – Protocolo BGP

Adaptado de: Forouzan e Mosharraf (2013, p. 321).

Continuando com a classificação dos protocolos de roteamento de acordo com a métrica utilizada pelo algoritmo de roteamento, encontramos aqueles chamados de estado de enlace (link state). Os protocolos de roteamento por estado de enlace vêm suprir uma série de deficiências facilmente constatadas pelos protocolos de roteamento por vetor de distância.

O algoritmo fundamentado no estado de enlace trabalha com uma visão global do roteamento (contrastando com o vetor de distância que trabalha apenas com a visão de vizinhos). Outro ponto interessante a destacar é que a métrica estabelecida pelo estado de enlace não é simplesmente uma contagem de saltos, mas um conjunto de indicadores que demonstram o estado atual do enlace, como a largura de banda e o atraso, por exemplo. Completando as vantagens do estado de enlace, quando comparado com o vetor de distância, nele não há o envio periódico de toda a tabela de roteamento, mas apenas das atualizações (TANENBAUM; FEAMSTER; WETHERALL, 2021).

Um dos algoritmos mais conhecidos em protocolos de roteamento por estado de enlace é o Dijkstra, que é utilizado pelo OSPF (*Open Shortest Path First* – Protocolo Aberto de Menor Rota Primeiro). O OSPF é um IGP muito mais eficiente que o RIP, porque trabalha com um custo que combina banda passante, confiabilidade e outras métricas. A figura a seguir apresenta uma topologia que utiliza o OSPF como protocolo de roteamento, e a tabela subsequente, os respectivos roteamentos criados (TANENBAUM; FEAMSTER; WETHERALL, 2021).

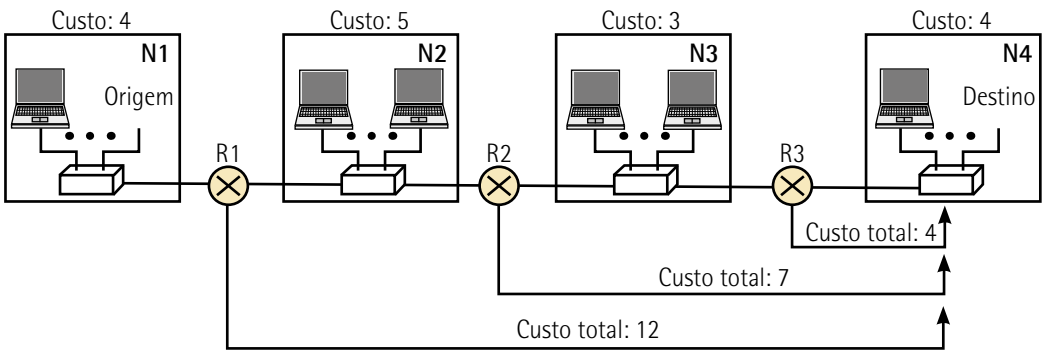


Figura 88 – Contagem de saltos no OSPF

Tabela 2 – Tabelas de roteamento dos roteadores R1, R2 e R3 utilizando o OSPF

Roteamento de R1			Roteamento de R2			Roteamento de R3		
Rede de destino	Próximo roteador	Custo em saltos	Rede de destino	Próximo roteador	Custo em saltos	Rede de destino	Próximo roteador	Custo em saltos
N1	–	4	**N1**	R1	9	**N1**	R2	12
N2	–	5	**N2**	–	5	**N2**	R2	8
N3	R2	8	**N3**	–	3	**N3**	–	3
N4	R2	12	**N4**	R3	7	**N4**	–	4

Adaptado de: Forouzan e Mosharraf (2013, p. 318).

O OSPF foi projetado para operar em um grande AS que pode ser dividido em áreas. A área 0 (zero) do OSPF é conhecida como backbone e está interligada às outras áreas por meio dos roteadores de borda de área. Na área 0 é possível encontrar outros dois tipos de roteadores: roteador de borda do AS e os roteadores de backbone. A figura a seguir apresenta a utilização do OSPF com múltiplas áreas dentro de um AS (FILIPPETTI, 2017).

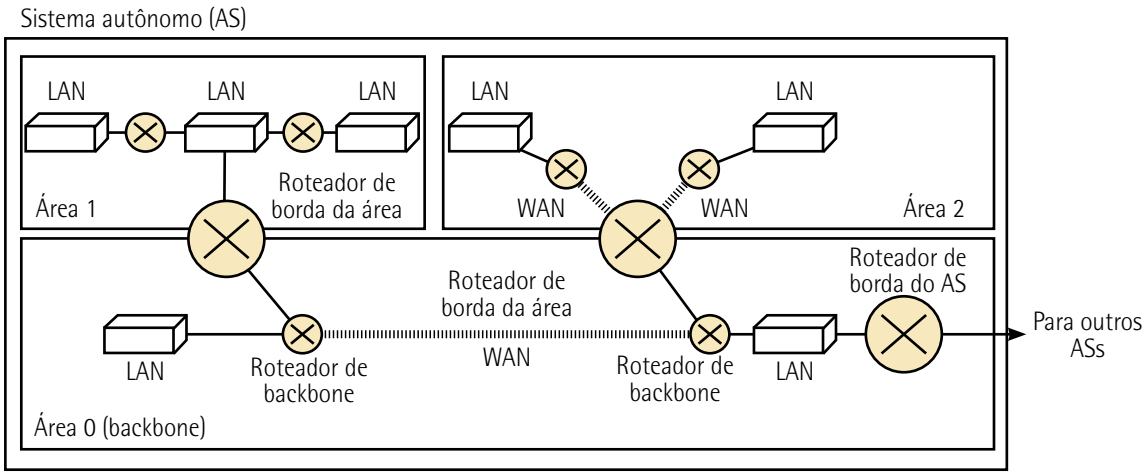


Figura 89 – OSPF e as múltiplas áreas

Adaptado de: Forouzan e Mosharraf (2013, p. 318).

Outro protocolo de roteamento por estado de enlace é conhecido como IS-IS (*Intermediate System-Intermediate System* – Protocolo de Estado de Enlace Intersistemas). A vantagem do IS-IS se comparado ao OSPF é que ele pode transportar diversos protocolos da camada de rede (IP, IPX, AppleTalk, dentre outros).

Saiba mais

Para conhecer mais sobre o OSPF, leia o capítulo 7, "Endereçamento IPv4", do livro a seguir:

OLIVEIRA, A. V.; MELO, J. L. *Certificação CCNA*: guia preparatório para o exame 200-301. SF Editorial: Rio de Janeiro, 2021.

6 PADRÕES E PROTOCOLOS DE CAMADA DE REDE – PARTE 2

Tendo conhecido os fundamentos que circundam a camada de rede, incluindo os protocolos de roteamento, agora trataremos sobre o protocolo roteado IP nas suas versões 4 e 6. Adentraremos um pouco os esquemas de endereçamento lógico e seu uso na internet, concluindo os estudos sobre as camadas que tratam da conectividade ao compreender o funcionamento do protocolo de internet.

6.1 Protocolo de internet

6.1.1 IPv4

Além dos protocolos de roteamento (que têm grande importância na disponibilização dos melhores caminhos para os pacotes), encontramos na camada de rede outros protocolos que atendem às suas funcionalidades básicas. Estes são os protocolos roteáveis: IPv4, IPv6, IPX, AppleTalk, dentre outros.

Embora já convivamos com a versão 6 do IP, constatamos que o IPv4 ainda é um protocolo de camada de rede muito difundido e utilizado pelas redes de computadores. Um bom exemplo de sua aplicação é a rede de comunicação de dados provida pela internet, que possibilita todas as facilidades de roteamento e endereçamento lógico necessários.

O IPv4 foi especificado e alterado nas RFCs 791, 950, 919, 922, 1349 e 2474. Seu grande mérito é a utilização permitida em qualquer tipo de rede física com interoperabilidade, propiciando a comunicação entre as diversas tecnologias de rede existentes tanto em nível físico quanto de transporte (TANENBAUM; FEAMSTER; WETHERALL, 2021).

Na operação do IPv4, cada pacote recebe tratamento isolado durante todo o seu percurso na rede, podendo trilhar caminhos diferentes uns dos outros – o que caracteriza uma abordagem em datagramas, contraposta às abordagens mais antigas de circuito virtuais. Assim, o IPv4 é um protocolo não orientado à conexão, e seus pacotes são tratados e avaliados a cada nó – ou seja, a cada equipamento pelo qual trafegam (OLIVEIRA; MELO, 2021).

Uma das características do tratamento desses pacotes em uma rede de comunicação de dados é que eles podem ser entregues a seu destino sem obedecer à ordem de saída. Desta forma, a tarefa de reagrupamento de dados na ordem correta fica sob responsabilidade das camadas superiores – como a de transporte, por exemplo.

O pacote IPv4 é muito simples de ser compreendido. Basicamente, temos um campo de cabeçalho e um campo de dados. O campo cabeçalho do IPv4 é composto por diversos campos que são utilizados para permitir o endereçamento e o roteamento correto dos pacotes pela rede. A figura a seguir apresenta os campos contidos no pacote IPv4 (MAIA, 2013).

<table>
<tr><th colspan="2">← 1 byte →</th><th>← 1 byte →</th><th colspan="2">← 1 byte →</th><th>← 1 byte →</th></tr>
<tr><td>Versão</td><td>Tamanho header</td><td>Tipo serviço</td><td colspan="3">Tamanho do pacote</td></tr>
<tr><td colspan="3">Identificação</td><td>Flags</td><td colspan="2">Deslocamento</td></tr>
<tr><td colspan="2">TTL</td><td>Protocolo</td><td colspan="3">Checksum do cabeçalho</td></tr>
<tr><td colspan="6">Endereço de origem</td></tr>
<tr><td colspan="6">Endereço de destino</td></tr>
<tr><td colspan="5">Opções do pacote IP</td><td>Preenchimento</td></tr>
</table>

Figura 90 – Pacote IPv4

Os campos do cabeçalho IP são os seguintes:

- **Versão**: apresenta a versão do protocolo, que no caso do IPv4 é o número 4.
- **Tamheader**: apresenta o tamanho do cabeçalho contado em números de palavras de 32 bits (4 bytes).
- **Tipo serviço**: apresenta a indicação de qualidade do serviço desejado para o encaminhamento do pacote por meio dos seus 8 bits.
- **Tampacote**: apresenta o tamanho do pacote em quantidade de octetos (bytes), com o valor máximo de 65.535 bits.
- **Identificação**: é o campo preenchido pela origem do pacote que o identifica. É usado na montagem da sequência dos pacotes no destino. Um pacote que precisa ser fragmentado por outro equipamento no caminho até o seu destino utiliza, neste campo, o mesmo valor para todos os fragmentos resultantes.
- **Flags**: campo de 3 bits que identifica se o pacote pode ser fragmentado no caminho até o destino e se já ocorreu fragmentação. O primeiro bit é sempre 0, o segundo bit indica se pode ou não fragmentar (0 = pode fragmentar, 1 = não pode fragmentar), e o terceiro bit indica se este pacote é (1) ou não é (0) o último fragmento.
- **Deslocamento**: caso tenha ocorrido fragmentação, este campo indica o deslocamento dos dados do pacote em relação ao campo de dados do pacote original (antes da fragmentação). Este campo é primordial para a remontagem do pacote e considera como unidade um octeto (1 byte).
- **TTL** (Tempo de vida): representa a quantidade de saltos por onde um pacote pode trafegar. Cada ativo de rede que roteia este pacote diminui o TTL de 1, sendo descartado quando o valor chega a zero.
- **Protocolo**: campo preenchido com um valor numérico que identifica para qual protocolo da camada superior a camada de rede deve entregar o conteúdo deste pacote no momento em que ele chegar ao destino. Exemplo: 6 – TCP, 17 – UDP, 1 – ICMP, 89 – OSPF etc.
- **Checksum do cabeçalho**: é o campo calculado e checado para cada salto que o pacote passa na rede, a fim de verificar a integridade do cabeçalho.
- **Endereço de origem**: é o endereço de origem do pacote, composto por 32 bits.
- **Endereço de destino**: é o endereço de destino do pacote, composto por 32 bits.

- **Opções do pacote IP**: este campo é opcional, mas requerido para algumas implementações. A origem do pacote colocará nesse campo as opções selecionadas. Esse campo é variável em seu tamanho e vai depender das opções definidas pela origem.

- **Preenchimento**: é o campo para preencher o cabeçalho, mantendo sempre o alinhamento em 32 bits.

6.1.2 Endereçamento IPv4

As redes da atualidade encontram-se quase todas interligadas e são compostas de uma quantidade considerável de equipamentos e hosts integrados. O melhor exemplo dessa integração é a existência da internet, em que calculam-se milhões de hosts interligados por meio de uma malha complexa de conexões de dados, trocando informações e pacotes.

A estrutura do endereçamento IPv4 possibilita essa integração e identificação. Ela foi idealizada e implementada com alguns requisitos importantes, que são:

- Cada host é único em relação a seu endereço lógico na rede e não podem existir dois endereços lógicos iguais no mesmo segmento.

- As redes podem ser divididas em sub-redes para garantir um gerenciamento eficiente de sua interligação com redes diferentes.

- A possibilidade de envio de informações para diversos hosts a partir de um único pacote com transmissão em broadcast.

Outra característica importante do esquema de endereçamento IPv4 reside na sua estrutura hierárquica. Em uma rede é possível identificar cada host de uma maneira única, e, com isso, ao juntá-las, elas conseguem se identificar em parte ou em sua totalidade a cada equipamento conectado, a partir dos gateways e roteadores, e ainda entregar os pacotes a seus destinos corretamente.

O endereço IPv4 é representado por um conjunto de 32 bits divididos em quatro octetos (bytes). Um exemplo de endereço IP pode ser visto na figura a seguir.

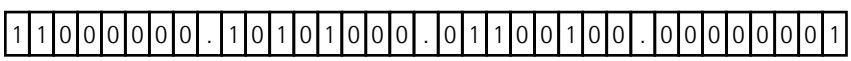


Figura 91 – Endereço IPv4

Esses bits podem ser representados por seu formato binário (notação binária pontuada) ou em formato decimal (notação decimal pontuada), separados por pontos. Sabendo que cada octeto possui 1 byte de tamanho e que 1 byte possui 8 bits, a conversão decimal binária em nosso exemplo da figura anterior pode ser calculada pelo valor/referência para cada octeto:

- Primeiro bit da esquerda para direita tem o valor decimal = 128.
- Segundo bit da esquerda para direita tem o valor decimal = 64.
- Terceiro bit da esquerda para direita tem o valor decimal = 32.
- Quarto bit da esquerda para direita tem o valor decimal = 16.
- Quinto bit da esquerda para direita tem o valor decimal = 8.
- Sexto bit da esquerda para direita tem o valor decimal = 4.
- Sétimo bit da esquerda para direita tem o valor decimal = 2.
- Oitavo bit da esquerda para direita tem o valor decimal = 1.

O primeiro octeto do exemplo tem o valor binário 11000000 e, somando os valores decimais dos bits com sinalização igual a 1 e desprezando os bits com sinalização igual a 0, temos o resultado dos cálculos: 128 + 64 = 192, correspondente ao valor decimal desse octeto.

O segundo octeto tem o valor binário 10101000 e, somando os valores decimais dos bits com sinalização igual a 1 e desprezando os bits com sinalização igual a 0, temos o resultado do cálculo: 128 + 32 + 8 = 168, correspondente ao valor decimal desse octeto.

O terceiro octeto tem o valor binário 01100100 e, somando os valores decimais dos bits com sinalização igual a 1 e desprezando os bits com sinalização igual a 0, temos o resultado do cálculo: 64 + 32 + 4 = 100, correspondente ao valor decimal desse octeto.

O quarto octeto do exemplo tem o valor binário 00000001 e, somando os valores decimais dos bits com sinalização igual a 1 e desprezando os bits com sinalização igual a 0, temos o resultado do cálculo: 1, correspondente ao valor decimal desse octeto.

Assim, a representação decimal do exemplo mencionado na figura anterior resulta no endereço com notação decimal pontuada (ou pontilhada): 192.168.100.1. O quadro a seguir apresenta outros exemplos de endereços IPv4, trazidos do formato binário para o formato decimal.

Quadro 12 – Exemplos de endereços IPv4

Número binário de 32 bits	Notação decimal pontilhada equivalente
10000001 00110100 00000110 00000000	129.52.6.0
11000000 00000101 00110000 00000011	192.5.48.3
00001010 00000010 00000000 00100101	10.2.0.37
10000000 00001010 00000010 00000011	128.10.2.3
10000000 10000000 11111111 00000000	128.128.255.0

Adaptado de: Comer (2016, p. 305).

Assim, em uma forma hierárquica, os endereços IPv4 podem ser divididos em cinco classes diferentes: A, B, C, D e E. As classes A, B e C são utilizadas para endereçamento de hosts. A classe D está destinada às comunicações em multicast e a classe E está reservada para uso futuro e para atividades experimentais. A identificação dos endereços dessas classes está nos valores dos primeiros bits do primeiro octeto. Na classe A, o primeiro bit do primeiro octeto é fixo e sempre igual a 1. Já na classe B, o primeiro e o segundo bits do primeiro octeto são fixos e respectivamente iguais a 1 e 0. Na classe C, o primeiro, o segundo e o terceiro bits são fixos e respectivamente iguais a 1, 1 e 0. Para a classe D, o primeiro, o segundo, o terceiro e o quarto bits são fixos e respectivamente iguais a 1, 1, 1 e 0. Na classe E, o primeiro, o segundo, o terceiro e o quarto bits são fixos e respectivamente iguais a 1, 1, 1 e 1.

A figura a seguir apresenta uma ilustração de cada uma dessas classes. Perceba que nas classes A, B e C temos o prefixo de rede (composto pelos bits fixos do primeiro octeto acompanhados dos bits que formam a parte denominada de "network") e o sufixo (que é a porção de host). Assim, temos na classe A apenas 1 octeto para a prefixo de rede e 3 octetos para a porção de host. Na classe B temos 2 octetos para o prefixo de rede e 2 octetos para a porção de host. Na classe C temos 3 octetos para o prefixo de rede e apenas 1 octeto para host. As classes D e E não são utilizadas para endereçar hosts.

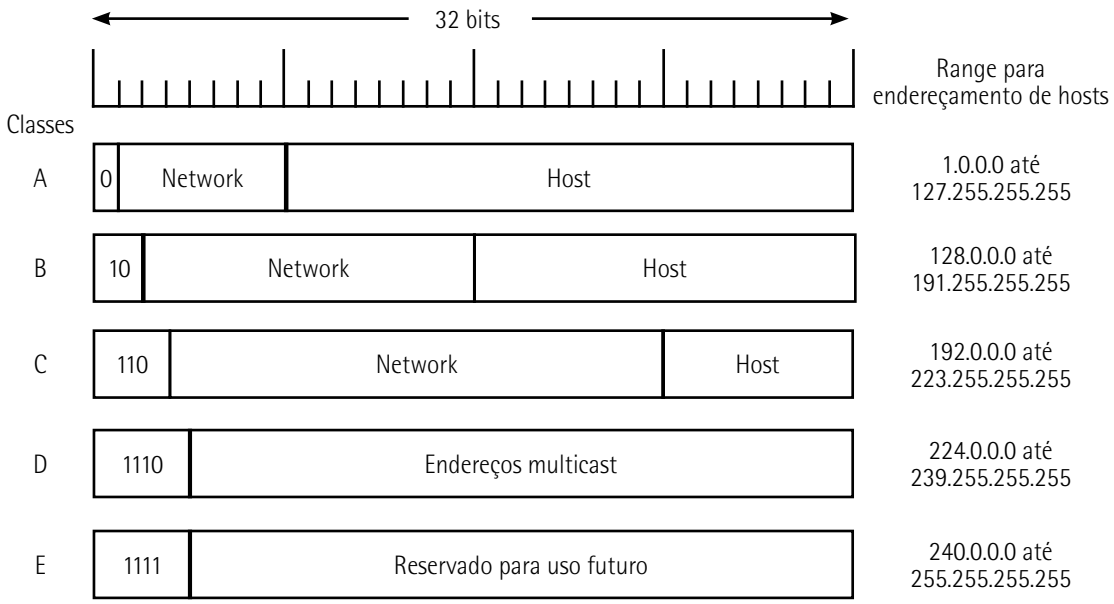


Figura 92 – Classes de endereços IP

Voltando para o exemplo encontrado na figura 91, temos o endereço IPv4 192.168.100.1 pertencente a classe C. No entanto, temos ainda mais um parâmetro para identificar adequadamente o endereço IPv4. Trata-se da máscara de rede, que é um conjunto de 32 bits de correspondência binária/decimal para a determinação do prefixo de rede (porção de rede do endereço IPv4), da quantidade de hosts possíveis no segmento e da identificação do endereço de broadcast.

Dessa forma, teremos para o endereço IPv4 192.168.100.1 a máscara classe C associada, que é /24. A representação do número 24 após uma barra indica a quantidade de bits que formam a porção de rede

(prefixo), que em nosso caso são 24 bits (três primeiros octetos). A figura a seguir apresenta a máscara em outras duas notações (binário pontilhada e decimal pontilhada).

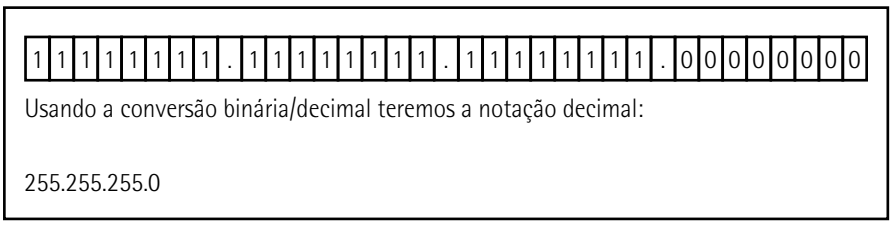


Figura 93 – Máscara de rede

Da mesma forma, para as classes A e B temos as máscaras descritas a seguir nas três notações (binário pontilhada, decimal pontilhada e barrada).

- **Classe A**: 11111111.00000000.00000000.00000000 – 255.0.0.0 – /8
- **Classe B**: 11111111.11111111.00000000.00000000 – 255.255.0.0 – /16

Estabelecidas as notações binárias/decimais do endereço e as suas máscaras, podemos executar o cálculo para descobrir o endereço de rede (o primeiro endereço IPv4 de uma rede que não pode ser usado para hosts e identifica diretamente a qual rede pertence o host) e o endereço de broadcast (que é o último endereço IPv4 de uma rede, utilizado para transmissão em broadcast desta rede e não pode ser usado para identificar hosts). A figura a seguir apresenta a busca pelo endereço de rede, utilizando a máscara e o endereço IPv4 em uma operação AND booleano ao longo de todos os bits.

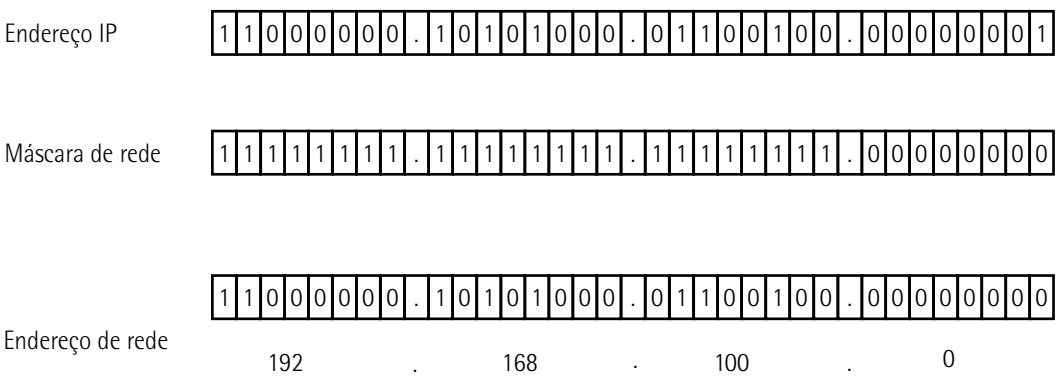


Figura 94 – Endereço de rede obtido por meio da máscara de rede

Para encontrarmos o endereço de rede, utilizamos o mesmo processo de AND booleano entre endereço IPv4 e máscara de rede no prefixo de rede. Para a porção de host levamos todos os bits para o valor igual a 1. A figura a seguir apresenta este processo.

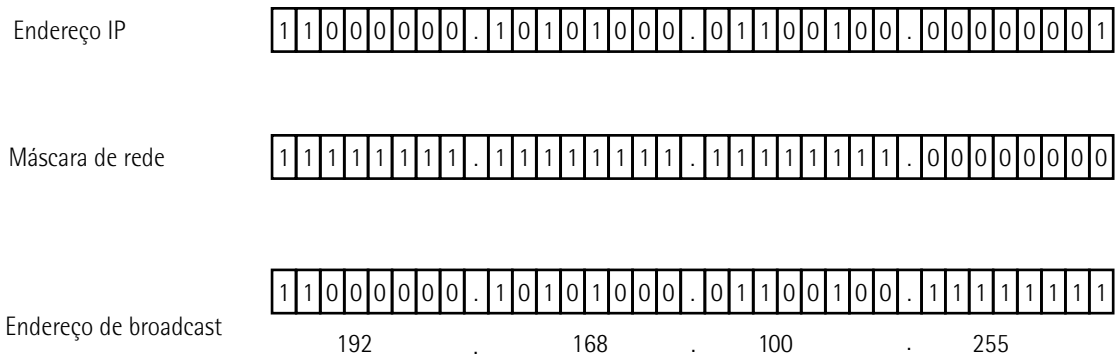


Figura 95 – Endereço de broadcast obtido por meio da máscara de rede

Os endereços IPv4 são classificados em privados e públicos. Os endereços IPv4 públicos são roteáveis em toda a internet. Já os endereços IPv4 privados são utilizados internamente em organizações, e pacotes que carregam estes endereços não são roteáveis na internet. O quadro a seguir apresenta a lista de endereços IPv4 privados.

Quadro 13 – Endereços IPv4 privados

Classe	Endereço inicial	Endereço final
A	10.0.0.0/8	10.255.255.255/8
B	172.16.0.0/16	172.16.255.255/16
C	192.168.0.0/24	192.168.255.255/24

Adaptado de: Maia (2013, p. 168).

Estes endereços IPv4 privados são utilizados na conectividade dentro de organização. Em todos os momentos em que há a necessidade do trânsito de um pacote da rede interna para a rede externa, o roteador (que é o gateway da rede) executa um processo de NAT (*Network Address Translation* – Tradução de Endereço de Rede). Maia (2013, p. 168) afirma que o NAT:

> permite que uma instituição opere com apenas um endereço na internet, mesmo possuindo inúmeros hosts na rede interna. Para isso, existe um dispositivo que implementa o NAT, geralmente um roteador, fazendo a ligação entre a rede interna e a internet. O NAT permite que uma instituição opere com apenas um endereço classe C, reduzindo assim a necessidade dos poucos endereços classe B ainda disponíveis. O NAT utiliza o conceito de endereços privados, que são endereços que não podem ser utilizados na internet, mas apenas dentro da rede interna.

Observação

Por meio do NAT e da utilização de endereços privados, foi possível obter uma grande economia na utilização de endereços IPv4.

6.1.3 Sub-redes IP

O esquema de endereçamento IPv4 apresenta um endereço com 32 bits divididos em duas porções distintas, chamadas de prefixo (ou porção de rede) e sufixo (ou porção de host). Por meio da utilização das máscaras de rede é possível encontrar os endereços de rede, os endereços de broadcast e a faixa válida de endereços a ser utilizada em uma rede.

Por maior ou menor que seja a porção de host em uma classe endereçável qualquer (A, B ou C), sempre haverá um desperdício de endereços IPv4, ensejando numa falta de eficiência na administração desses endereços. Apenas para termos uma ideia, o quadro a seguir apresenta a quantidade de hosts por rede disponíveis em cada uma das classes endereçáveis do IPv4.

Quadro 14 – Quantidade de hosts por rede em cada classe de endereços IPv4

Classe	Número de redes	Números de hosts
A	128	16.777.214
B	16.384	65.534
C	2.097.152	254

Ao atribuir um endereço de classe A para uma empresa, ela receberá uma rede com 16.777.214 hosts. Nem mesmo grandes empresas dispõem de hosts suficientes para ocupar todo o espaço de endereçamento de uma rede classe A. Já no caso de uma rede classe B, são 65.534 hosts e, embora seja um número bem menor, ainda é bastante grande, porque alocar uma classe B para uma rede de 500 hosts deixaria 65.034 endereços sem uso. Já uma classe C ofereceria somente 254 hosts, valor muito baixo para a grande maioria de empresas do mercado. Inevitavelmente, as empresas acabam aumentando de tamanho e precisando de mais endereços de rede de classe C. Vemos que muitas empresas utilizam endereços classe A, como Apple, Xerox, HP e IBM.

Considerando essa questão e o iminente esgotamento da faixa de endereços IPv4, criou-se o mecanismo de divisão em sub-redes, que consiste em um cálculo, utilizando bits da porção de host para a divisão em sub-redes – aumentando, assim, a capacidade de criação de redes com faixas menores de endereços IPv4, e contribuindo para maior eficiência no uso do esquema de endereçamento.

Lembrete

Quem define a porção de rede e a porção de hosts é a máscara de rede. É ela que permite identificar quantos bits há em cada porção.

A figura a seguir apresenta a ideia da divisão em sub-redes.

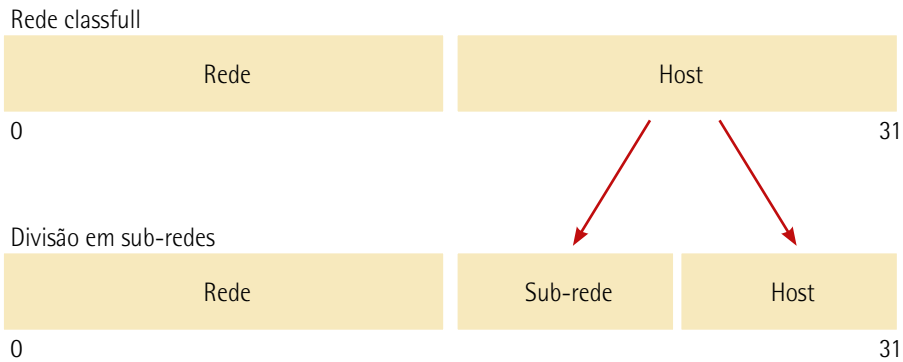


Figura 96 – Porções do endereço de rede

Para melhor compreendermos, usemos um exemplo utilizando o endereço 172.16.0.0/16 para uma divisão em quatro sub-redes. Vemos que este endereço é de classe B, então temos que tomar bits emprestados da porção de host. No endereço de classe B, a porção de rede corresponde aos primeiros 16 bits, e a de host, aos 16 bits seguintes. Peguemos emprestados os bits mais significativos ou, ainda, o mais à esquerda da porção de host para atribuir quatro sub-redes. Depois, peguemos bits suficientes para endereçá-las.

Para chegar ao número quatro usando a regra de 2^b, em que b é o número de bits que pegamos emprestados, precisaremos de 2 bits. Esses bits que foram retirados da porção de host vão fazer parte da porção de sub-rede e também serão contabilizados pela máscara de sub-rede. A máscara de sub-rede nos indica o que é porção de rede e o que é porção de host. A seguir, vamos ver o processo finalizado para atribuir os quatro novos endereços IP.

Precisaremos de dois bits da porção de host, pois $2^2 = 4$. A máscara nos mostra que a porção de rede é composta dos dois primeiros octetos e a porção de host, dos dois octetos restantes. A figura a seguir apresenta a porção de rede e host em relação à máscara padrão e a escolha dos dois bits mais significativos que serão tomados para criar as sub-redes.

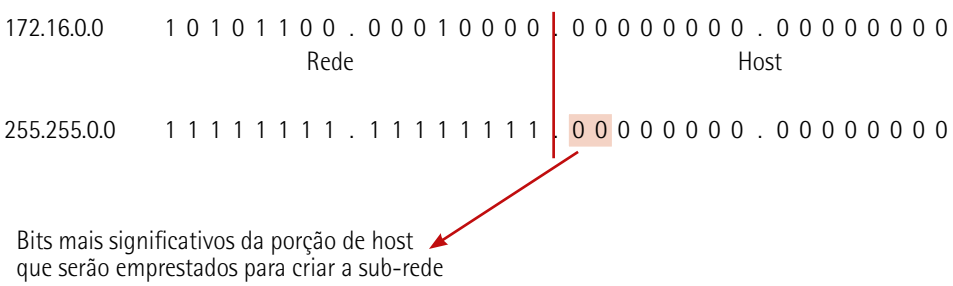


Figura 97 – Identificação da porção de rede

No momento em que os bits necessários para obter as quatro sub-redes são selecionados, eles passam a compor a proporção de sub-rede e ainda ocorre uma alteração significativa da máscara, que passa a ter uma nova denominação: máscara de sub-rede. Essa nova máscara indicará uma porção de rede estendida, pois complementa os bits que foram emprestados para criar as sub-redes.

A figura a seguir nos mostra a nova máscara, os bits emprestados para a porção de sub-rede e as três porções que compõem o endereçamento IP.

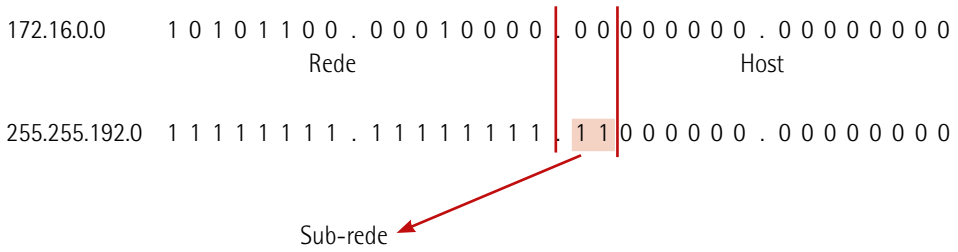


Figura 98 – Bits emprestados para a sub-rede

Calculada a nova máscara, ela passa a criar as sub-redes. Já sabemos que o endereço de rede possui todos os bits da porção de host que foram definidos como o número 0 e que o endereço de broadcast possui todos os bits da porção de hosts que foram definidos como número 1. Já é possível identificar o primeiro endereço de sub-rede, que será o próprio endereço da rede usado para realizar essa divisão, mas com uma nova máscara. Para identificar o endereço de broadcast, colocamos todos os bits da porção de hosts com os números 1 em binário. A figura a seguir apresenta a primeira sub-rede.

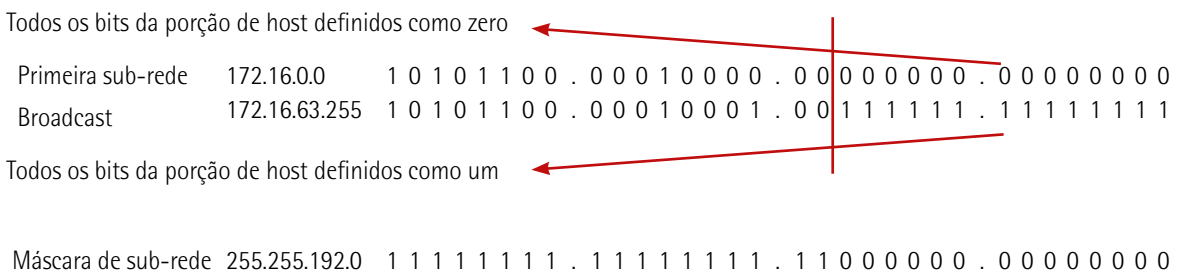


Figura 99 – Demonstração da primeira sub-rede

Os outros três endereços remanescentes de sub-rede serão obtidos por meio da manipulação dos dois bits da porção de sub-rede. Então, realizando todas as combinações possíveis para obter os quatro endereços de sub-rede, as próximas sub-redes serão demonstradas na figura a seguir.

Figura 100 – Combinando os bits das sub-redes

Conforme visto na sequência de divisão em sub-redes, foi necessário manipular apenas os 2 bits para obter todas as combinações possíveis; entretanto, quando se manipula mais bits, essa tarefa é bem difícil e complexa.

Por exemplo: se desejarmos 28 sub-redes, precisamos tomar emprestado 5 bits. Apenas 5 bits nos possibilitam obter 32 novas sub-redes – ou seja, 4 sub-redes a mais do que precisamos. Porém, se usássemos apenas 4 bits, teríamos somente 16 sub-redes, um número bem menor do desejado.

Devemos então considerar que, ao usar 5 bits, teremos 4 redes disponíveis para usar futuramente. Essa é uma boa técnica para prever futuros crescimentos na infraestrutura. Ao optar por redes com o número mínimo de hosts, é preciso observar atentamente quantos bits serão necessários para ter uma porção de hosts. Com o número de hosts a serem escolhidos, os bits que sobraram da porção de hosts serão os que emprestaremos para criar essas sub-redes.

A figura a seguir apresenta esse processo considerando um endereço IPv4 privado 172.16.0.0/16.

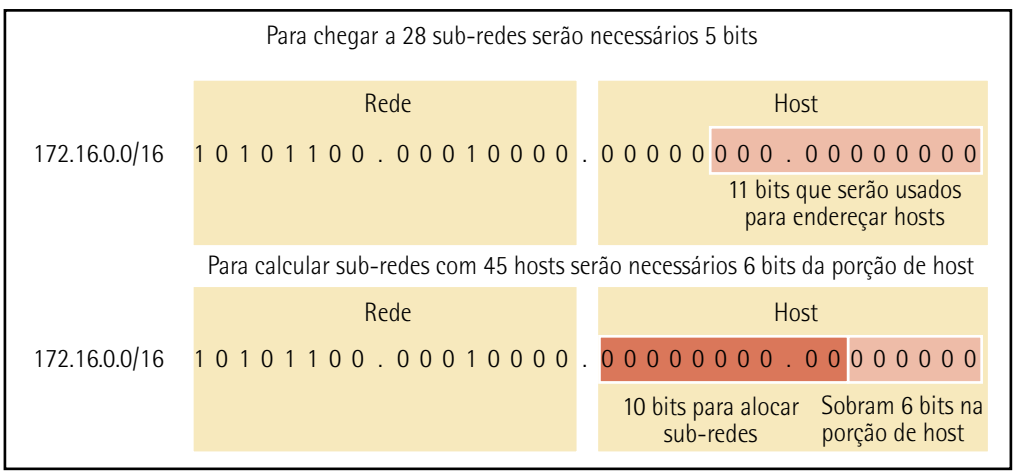


Figura 101 – Exemplo de divisão em sub-redes

Esta figura mostra que, quando feita a opção por 28 sub-redes, tem-se na verdade 32 sub-redes, embora cada uma delas tenha 2.046 hosts. Quando feita a opção por 45 hosts, teremos exatamente 62 hosts no lugar dos 10 bits restantes, que serão reservados para a sub-rede, dando um total de 1.024 sub-redes, cada uma delas com 62 hosts.

Escolhido o número de bits que vão ser emprestados da porção de hosts, os incluiremos na máscara com a função de determinar as sub-redes e os endereços de hosts. Finalmente, pode-se atribuir os endereços aos hosts de rede. No exemplo usado, as máscaras seriam as seguintes:

- Para a opção pelo número de redes: 255.255.248.0
- Para a opção pelo número de hosts: 255.255.255.192

Sempre que manipulamos os bits de um endereço para criar as sub-redes, devemos atentar à classe a que pertence aquele endereço. Se for um endereço de classe A, a porção de redes possui 8 bits, e a porção de hosts, logicamente, 24 bits. Podemos ainda pegar os bits emprestados da porção de hosts a partir do nono bit do endereço. No caso de endereço de classe B, a porção de redes possui apenas 16 bits, assim como a de hosts.

O empréstimo de bits de host para sub-rede deve ser feito do mais significativo para o menos significativo – ou seja, a leitura deve ser feita da esquerda para a direita, em sequência, sem faltar nenhum bit. Os bits disponíveis que podem ser emprestados em cada classe são mostrados na figura a seguir.

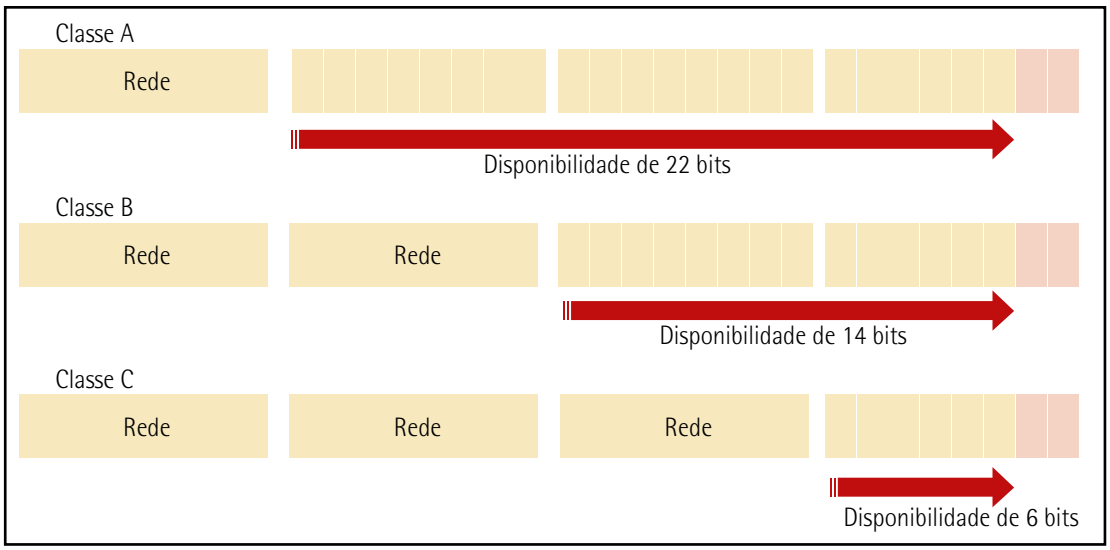


Figura 102 – Disponibilidade de bits para sub-redes em cada uma das classes

Depois de tomarmos os bits emprestados, é preciso deixar pelo menos 2 bits para porção de host. Isso é necessário para ter sempre 2 hosts válidos em cada rede. Dois bits nos permitem obter 2 hosts, pois $2^2 - 2 = 2$, o que permite, nessa condição, números de host válidos para um endereço de rede que possui 2 bits disponíveis na sua porção de host, que é o caso da máscara /30.

Partindo para outro exemplo, consideremos um endereço IPv4 192.168.1.0/24. Precisamos dividir o endereço em três sub-redes. Usaremos o endereço de classe C 192.168.1.0, que tem como máscara padrão 255.255.255.0. Sabendo o número de sub-redes, precisaremos saber quantos bits são necessários para chegarmos ao número três ou maior utilizando a regra de 2^b, em que b é o número de bits necessários para o cálculo.

Para chegar a três sub-redes, precisamos apenas de 2 bits, pois $2^2 = 4$. Ao fazer este cálculo de 2^b, saberemos que é preciso pegar 2 bits emprestados da porção de host para serem aplicados na porção de sub-rede. Vamos emprestar os 2 bits mais significativos da porção de host, conforme pode ser visto na figura a seguir. É importante observar que a porção de host ficou com 6 bits. Esses bits serão usados para endereçar os hosts, chegando ao total de 62 hosts por sub-rede.

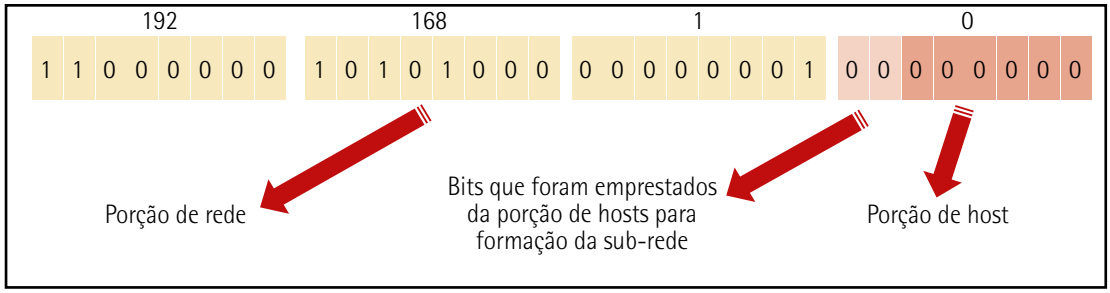


Figura 103 – Empréstimo de bits de host para sub-rede

Para chegar ao primeiro endereço de rede e seu endereço de broadcast, precisamos saber o valor de todos os bits da porção de rede com 0 e 1, respectivamente. A figura anterior mostra a porção de host com seus bits em zero, e a figura a seguir, com todos os bits de host definidos em 1, gerando o endereço de broadcast.

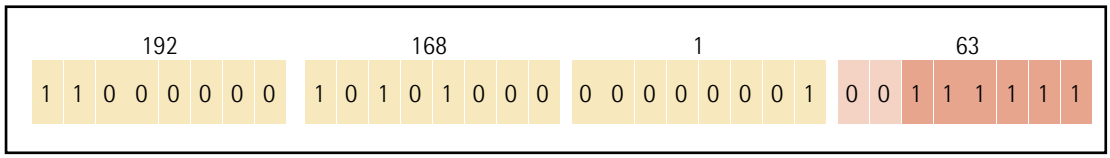


Figura 104 – Endereço de broadcast

Para chegar ao primeiro endereço de host válido para a rede 192.168.1.0/26, basta calcular todos os bits de host como zero, exceto o último, que é o menos significativo.

Dessa forma, chegaremos ao endereço 192.168.1.1 como o primeiro endereço válido para essa rede. Para saber o último endereço, basta diminuir em uma unidade o valor do último octeto do endereço de broadcast. Chegamos ao endereço 192.168.1.62 como o último endereço válido.

Os cálculos dos endereços restantes serão alcançados pela manipulação dos 2 bits emprestados para a porção de sub-rede. Precisamos fazer todas as combinações possíveis de 0 e 1 para chegar ao valor das sub-redes. Para obter o próximo endereço de rede, precisamos apenas somar uma unidade ao último octeto do endereço de broadcast. Fazendo essa somatória, chegaremos ao número 64. O valor obtido depois da adição será 192.168.1.64, o segundo endereço de rede depois da divisão. O primeiro endereço válido da segunda rede será obtido da mesma forma como feito na primeira rede, atribuindo o bit menos significativo da porção de host da máscara como um. O resultado é o endereço 192.168.1.65 como primeiro endereço válido para a segunda rede. Em relação ao último endereço válido, subtraímos uma unidade do último octeto do endereço de broadcast e assim teremos o endereço 192.168.1.126. A figura a seguir nos mostra o endereço de rede e de broadcast em binários e mostra os endereços de rede e broadcast para a segunda rede.

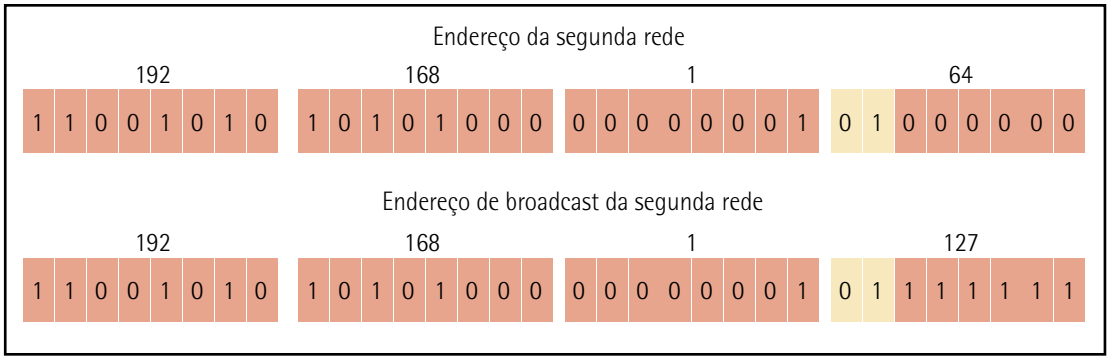


Figura 105 – Endereço da segunda sub-rede e de broadcast

O quadro a seguir apresenta um resumo dos endereços das sub-redes acompanhados dos endereços de broadcast.

Quadro 15 – Endereços de sub-redes e de broadcast

	Endereço de rede	Endereço de broadcast
1º endereço	192.168.1.0	192.168.1.63
2º endereço	192.168.1.64	192.168.1.127
3º endereço	192.168.1.128	192.168.1.191
4º endereço	192.168.1.192	192.168.1.255

Observação

Quando trabalhamos sem sub-redes, costumamos chamar as máscaras e os endereços de classful. Quando trabalhamos com sub-redes, costumamos chamar os endereços e as máscaras de classless.

Partindo agora para um outro exemplo, vamos utilizar um endereço IP classe A 10.0.0.0/8. Precisamos dividir o endereço em 400 sub-redes. Usaremos o endereço de classe A 10.0.0.0, que tem como máscara padrão 255.0.0.0. Sabendo o número de sub-redes, temos que verificar quantos bits são necessários para termos o número 400, ou maior, utilizando a regra de 2^b em que b é o número de bits necessários.

No caso de 400 sub-redes, precisaremos de 9 bits, pois $2^9 = 512$. Caso usássemos 8 bits, teríamos somente 256 sub-redes, número insuficiente para a nossa necessidade. Fazendo o cálculo de 2^b, descobrimos que devemos pegar 9 bits emprestados da porção de host para que sejam utilizados na porção de sub-rede.

Pegamos emprestados os 9 bits mais significativos da porção de host e identificamos que a porção de host ficou com 15 bits. Esses bits serão utilizados para endereçar os hosts, totalizando 32.766 hosts por sub-rede, número atingido no cálculo.

Para saber o primeiro endereço de rede e seu endereço de broadcast, é preciso definir todos os bits da porção de rede com 0 e 1, na ordem. Assim temos a porção de host com todos os bits em zero e a figura seguinte, com todos os bits de host definidos em um (broadcast).

Nesse exemplo, identificamos que o primeiro endereço de host válido para a rede 10.0.0.0/17 é obtido marcando todos os bits de host como zero, menos o último, ou seja, o menos significativo.

Desse jeito teremos o endereço 10.0.0.1 como primeiro endereço válido. Para obter o último endereço, basta diminuir em uma unidade o valor do último octeto do endereço de broadcast e teremos o endereço 10.0.127.254 como o último endereço válido, ou ainda usar a fórmula conhecida $2^n - 2 =$ hosts, onde n é a quantidade de zeros mais à direita na máscara resultante.

Ainda, temos que alterar a máscara de 255.0.0.0 para a máscara de sub-rede. Precisamos disso para definir como binários o número de bits referentes à porção de sub-rede, neste caso 9 bits.

O resto dos endereços serão resultantes do cálculo, por meio da manipulação dos 9 bits emprestados para a porção de sub-rede. Precisamos fazer todas as combinações possíveis de 0 e 1 para chegar a todas as sub-redes, no entanto, realizar essa operação para muitos bits é cansativo.

Para obter o próximo endereço de rede, basta adicionar uma unidade ao último octeto do endereço de broadcast. Entretanto, ao fazer essa soma, chegaremos ao número 256. Como o valor de cada octeto deve estar entre 0 e 255, em vez de colocar 256, colocamos zero e adicionamos uma unidade saltando ao terceiro octeto. Teremos então o número 128 no terceiro octeto. O endereço calculado depois das adições será 10.0.128.0, o segundo endereço de rede da divisão.

O primeiro endereço válido da segunda rede será obtido da mesma forma que na primeira rede, definindo o bit menos significativo da porção de host como um. Teremos o endereço 10.0.128.1 como primeiro endereço válido para a segunda rede. No caso do endereço do último endereço válido, diminuímos uma unidade do último octeto do endereço de broadcast, ou seja, teremos o endereço 10.0.255.254, e assim sucessivamente.

Para chegar ao terceiro endereço de rede, o procedimento é o mesmo realizado para obter a segunda rede. Ao somar uma unidade ao quarto octeto, teremos o valor 256, ou seja, mudamos o octeto para zero, saltando para o próximo octeto, e adicionamos um ao terceiro octeto. Entretanto, o terceiro octeto também nos dará o valor 256 ao ser adicionado em um. Devemos alterar o terceiro octeto para zero e adicionar uma unidade ao segundo octeto. Teremos o valor um no segundo octeto e obteremos o terceiro endereço de rede, 10.1.0.0.

O quadro a seguir apresenta os primeiros e últimos endereços de rede e seus respectivos endereços de broadcast para a divisão em sub-redes do endereço aplicado a este primeiro exemplo.

Quadro 16 – Endereços de sub-rede e broadcast

	Endereço de rede	Endereço de broadcast
1º endereço	10.0.0.0	10.0.127.255
2º endereço	10.0.128.0	10.0.255.255
3º endereço	10.1.0.0	10.1.127.255
510º endereço	10.254.128.0	10.254.255.255
511º endereço	10.255.0.0	10.255.127.255
512º endereço	10.255.128.0	10.255.255.255

Exemplo de aplicação

Com o intuito de treinar um pouco o cálculo de sub-redes, utilize um endereço IP de qualquer classe, sendo público e faça um processo de divisão em sub-redes deixando apenas 5 bits de host.

6.1.4 IPv4 × IPv6

Devido ao ritmo acelerado de evolução das redes de computadores, ao ingresso de novos dispositivos móveis e ao crescimento da população com acesso à internet em todas as localidades do planeta, surgiu a necessidade de mais endereços no padrão IP e, com o fim prematuro do protocolo IPv4, tornou-se necessária a evolução desse protocolo.

O endereçamento IPv4, ainda em uso atualmente, não suportou esse crescimento de dispositivos e a demanda de acesso à internet, extinguindo rapidamente os seus recursos de endereçamento. Certamente esse é o principal motivo para a idealização de um novo protocolo de endereçamento que fosse suportado pelos próximos anos, o que levou à criação do protocolo IPv6.

O IPv6 foi projetado para ser o sucessor do IPv4. Ele tem maior espaço de endereços, que desta vez possuem 128 bits, fornecendo 340 undecilhões de endereços. Esse valor é escrito com o número 340 seguido de 36 zeros. Entretanto o IPv6 é muito mais do que números em quantidades maiores. Quando o comitê IETF (*Internet Engineering Task Force*) iniciou seu desenvolvimento, aproveitou para corrigir muitas das limitações do IPv4 e ainda incluir novos aprimoramentos (OLIVEIRA; MELO, 2021).

Um bom exemplo é o ICMP versão 6, que inclui a resolução de endereço com uma configuração automática, que não é encontrada nos ICMP da versão 4.

Algumas características desse novo protocolo são:

- maior espaço de endereçamento;
- mobilidade;

- segurança;
- autoconfiguração;
- compatibilidade com o IPv4.

A redução das reservas de endereços IPv4 certamente foi o principal fator para a criação e migração de um novo protocolo. Conforme continentes como África, Ásia e algumas outras partes do mundo forem se conectando à internet, não haverá endereços IPv4 suficientes para absorver todo esse crescimento.

O IPv4, como sabemos, tem um máximo teórico de 4,3 bilhões de endereços possíveis combinados ao NAT (*Network Address Translation* – Tradução de Endereços de Rede). Os endereços privados foram imprescindíveis para retardar e conter a redução do espaço dos endereçamentos IPv4; o NAT, porém, danifica o funcionamento de muitos aplicativos e tem determinadas limitações que impedem, principalmente, comunicações ponto a ponto (FILIPPETTI, 2017).

6.2 IPv6 e outros protocolos da camada de rede

6.2.1 Pacote IPv6

O pacote IPv6 guarda algumas semelhanças com a versão 4 e é composto de duas partes: cabeçalho e dados. A grande diferença reside no tamanho do cabeçalho, que é bem mais simples e otimizado, a fim de agilizar o encaminhamento das informações através das redes de computadores. A figura a seguir apresenta o pacote IPv6.

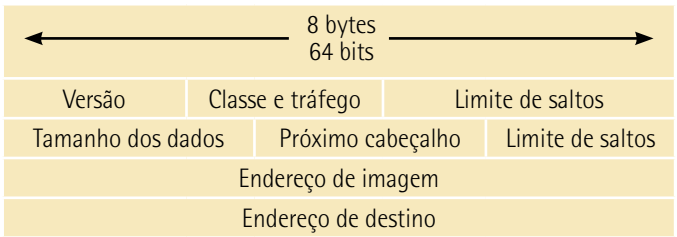


Figura 106 – Pacote IPv6

A definição dos campos do cabeçalho é a seguinte:

- **Versão**: é a versão do protocolo, no caso, 6.
- **Classe e tráfego**: indica a prioridade do pacote.
- **Identificador de fluxo**: QoS (Qualidade do Serviço).
- **Tamanho dos dados**: informa o tamanho da parte de dados do pacote IPv6.

- **Próximo header** (cabeçalho): é o campo que aponta para o próximo header do IPv6. Essa característica de possuir mais de um header foi criada para simplificar o cabeçalho padrão, e, caso sejam necessárias funções especiais, cabeçalhos extras são alocados e inseridos na parte de dados do pacote IP.

- **Limite de saltos**: oficializando o que já acontecia com o campo TTL (Tempo de Vida) do IPv4, este campo limita a quantidade de dispositivos que roteiam os pacotes por onde este pacote pode passar. Caso esse número chegue a zero, o pacote é descartado.

- **Endereço de origem**: é o endereço do dispositivo de origem representado por um campo de 128 bits.

- **Endereço de destino**: é o endereço do dispositivo de destino representado por um campo de 128 bits.

6.2.2 Endereçamento IPv6

O protocolo IPv6 usa como endereçamento uma palavra com 128 bits, capaz de gerar um total de 3.4×10^{38} de endereços possíveis, garantindo uma longevidade considerável.

Da mesma maneira que no protocolo IPv4, a forma de representação do endereçamento do IPv6 não é realizada no formato binário, pois, pelo tamanho, seria muito difícil a sua representação. Então, no IPv6, a representação do endereço é feita pelo agrupamento de 16 em 16 bits separados pelo sinal de dois-pontos (:).

Como demostrado a seguir, o formato preferencial para se escrever um endereço do padrão IPv6 é X:X:X:X:X:X:X:X, em que cada X consiste em quatro valores hexadecimais. Ao falarmos de endereçamento IPv4, nos referenciamos a 8 bits com o termo octeto. Entretanto, o termo usado no IPv6 é o hexteto, um termo ainda informal e que é empregado basicamente para fazer referência a um segmento de 16 bits, ou 4 valores hexadecimais, sabendo que cada X equivale a um único hexteto, ou 16 bits, ou ainda a 4 dígitos hexadecimais.

O formato preferencial significa essencialmente que os endereçamentos IPv6 são gravados usando todos os 32 dígitos hexadecimais; isso, entretanto, não significa que esse seja o método ideal para representar os endereçamentos em IPv6. A seguir, veremos as regras que nos ajudam a reproduzir os números e os dígitos que são imprescindíveis para a representação de endereço IP versão 6.

Esses grupos de 16 bits são representados usando uma notação hexadecimal, em que cada dígito hexadecimal representa 4 bits separados. Dessa forma, teremos:

XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX

Em que X é um dígito hexadecimal representado pelos valores (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

Exemplos:

FADA:FADA:0000:FFFF:FFFF:4AFD:5EAA1:0000

Ou:

FFFF:0000:0000:0000:0000:0000:0000:0001

Caso existam valores 0 à esquerda do número nos grupos de 16 bits, no momento da representação esses zeros podem ser suprimidos, por exemplo: 001A pode ser representado apenas por 1A.

2017:0000:1F3A:0000:0000:1A:2345:5678

Se existirem agrupamentos de 4 dígitos zero (0000), eles podem ser suprimidos e representados desta forma:

2017:0000:1F3A:::FF1A:2345:5678

Ou:

2017::1F3A:0000:0000:FF1A:2345:5678

O endereçamento IPv6 também especifica três tipos diferentes de endereçamento: o unicast, anycast e o multicast.

O unicast endereça apenas uma interface, ou seja, não há mais de uma interface respondendo ao mesmo endereço. O endereço IPv6 unicast identifica exclusivamente uma interface de um dispositivo que esteja habilitado para IPv6. Observe na figura a seguir um mecanismo de endereço IP versão 6 origem, que deve ser um endereço unicast.

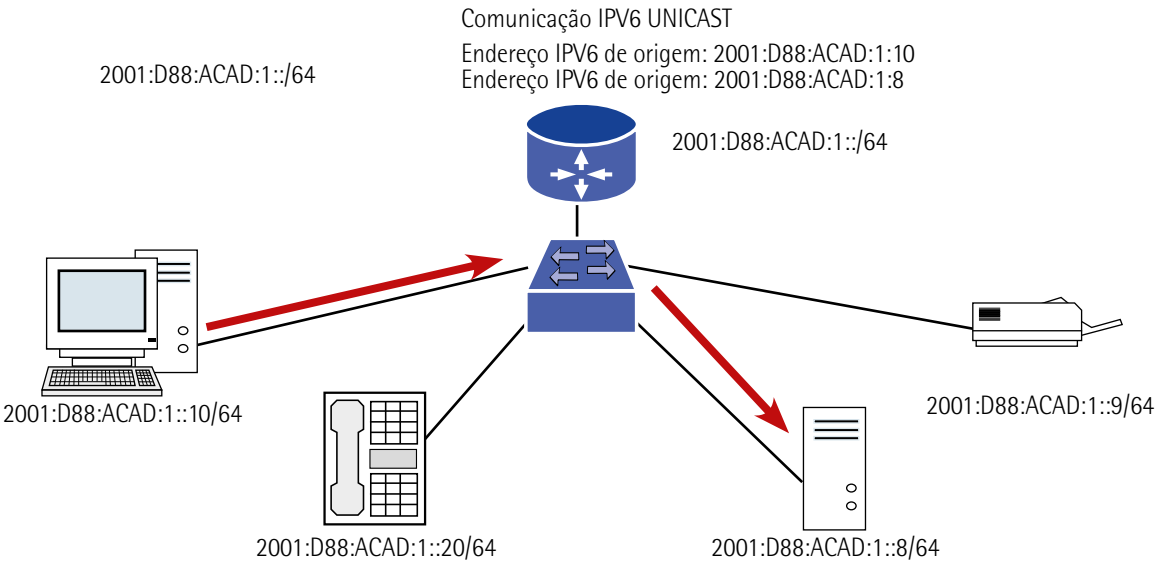


Figura 107 – IPv6 unicast

Devemos lembrar que o prefixo, que é a parte da rede de endereço padrão IPv4, deve ser identificado pelo comprimento, pela notação em sua barra ou por uma máscara de sub-rede em formato decimal com pontos. Um exemplo é o endereço IPv4 192.168.1.10 com a máscara de sub-rede em formato decimal com pontos 255.255.255.0, que é equivalente à notação decimal 192.168.1.10/24.

O endereçamento IPv6 usa um comprimento de prefixo a fim de representar a parte de prefixo do endereço. O IPv6 não utiliza uma notação de máscara de sub-rede decimal com pontos, como acontece no IPv4. O comprimento desse prefixo indica a parte de rede de um endereço IPv6 no formato do endereço IPv6/comprimento do prefixo.

O comprimento do prefixo pode variar de 0 a 128. Um comprimento do prefixo IPv6 padrão para LANs e para a maioria dos outros tipos de redes é /64. Isso significa que o prefixo ou a parte de rede do endereço é de 64 bits, restando outros 64 bits para a ID da interface (parte de host) do endereço. A figura a seguir apresenta o prefixo do endereço IPv6.

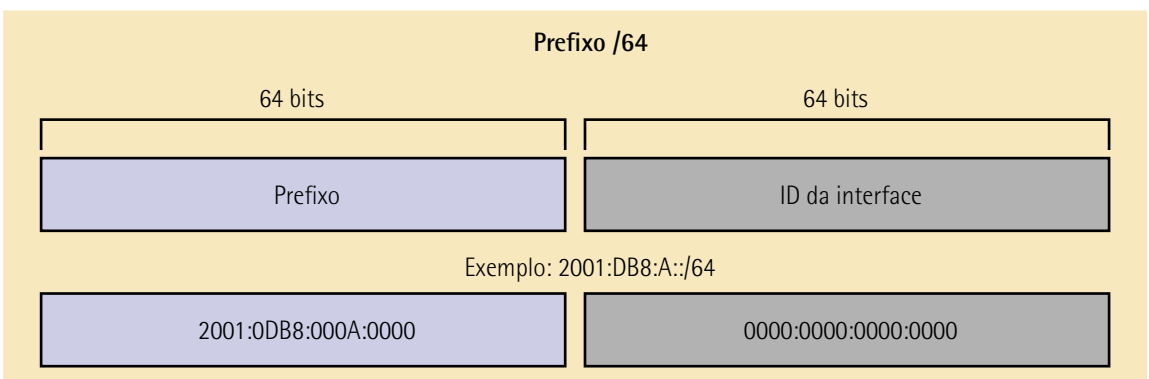


Figura 108 – Prefixo do IPv6

No momento de representar o endereço IPv6 unicast, ele identifica, exclusivamente, uma interface em um tipo de dispositivo que esteja habilitado para IPv6. Um pacote que seja enviado a um endereço unicast é recebido por uma interface atribuída diretamente a esse endereço. Muito semelhante ao IPv4, os endereços IPv6 de origem devem ser um endereço unicast, mas o endereço IPv6 de destino ainda pode ser um endereço unicast ou multicast.

Os tipos mais comuns de endereços IP versão 6 unicast são endereços unicast globais, ou GUA (*Global Unicast Addresses*), e os endereços unicast de link local. A figura a seguir apresenta os endereços IPv6 unicast.

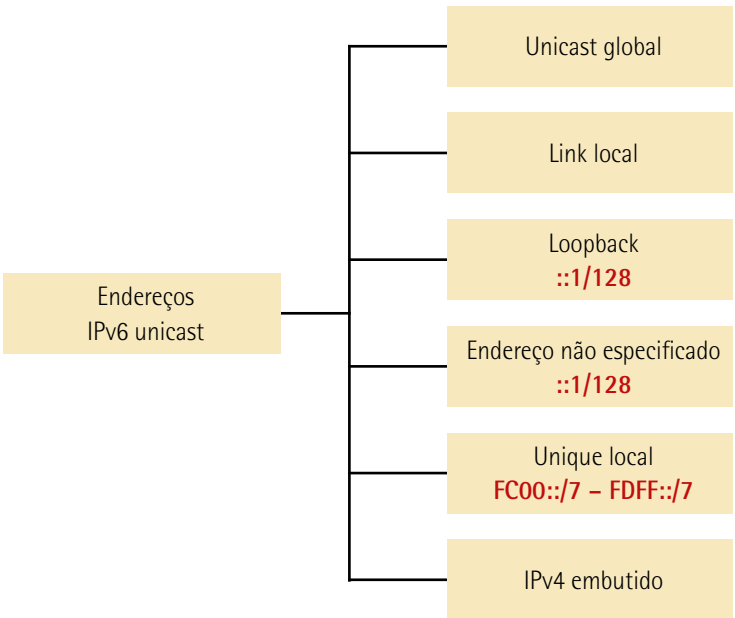


Figura 109 – Endereço IPv6 unicast

O endereço unicast global é bem parecido com o endereço IPv4 público. São endereços de internet basicamente roteáveis e globalmente exclusivos. Os endereços unicast globais podem ser configurados estaticamente ou serem atribuídos em formato dinâmico. A figura a seguir apresenta endereços unicast globais.

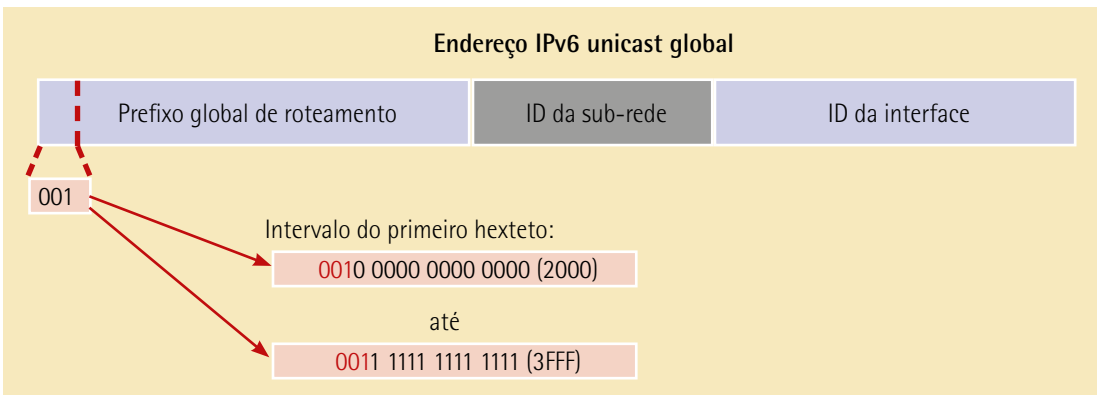


Figura 110 – Endereço IPv6 unicast globais

Os endereços de link local são utilizados para estabelecer a comunicação com outros dispositivos que estejam presentes no mesmo segmento do link local. No caso do IPv6, o termo link refere-se a uma sub-rede e os endereços de link local são limitados a um único link. Essa exclusividade só deve ser afirmada nesse link porque eles não são roteáveis para além do link – ou seja, os roteadores não encaminham pacotes com endereços de link com local de origem ou de destino. A figura a seguir apresenta o estabelecimento de links locais em redes operando com IPv6.

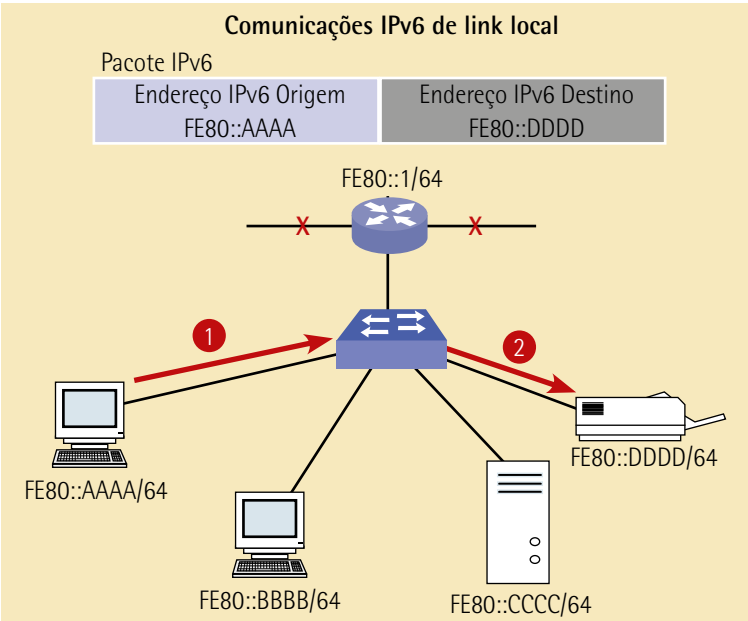


Figura 111 – O estabelecimento do link local em redes operando IPv6

Outra classe de endereçamento IPv6 unicast é conhecida como unique local. Os endereços IPv6 unique local possuem certas semelhanças com endereços privados da RFC 1918 para IPv4 (exemplo 127.0.0.1), mas essas semelhanças param por aí. Os endereços unique local são usados para o endereçamento local dentro de um site ou dentro de um número limitado de sites. Esses endereços não devem, em hipótese alguma, ser roteados pelo IPv6 global e nem passar por tradução de endereços (NAT) IPv6 global. Os endereços unique local estão no intervalo FC00::/7 a FDFF::/7.

No endereçamento IPv4, os endereços privados são combinados com mecanismos de tradução de rede ou tradução de porta. Endereços de vários para um, privados para públicos, por exemplo. Isso acontece em função da limitante disponibilidade do espaçamento de endereços IPv4. Muitos sites utilizam mecanismos de natureza privada para endereços RFC 1918 com a intenção clara de proteger a própria rede contra potenciais vulnerabilidades da segurança ou até mesmo ocultá-la; essa técnica, entretanto, nunca foi definida para estas tecnologias. A IETF recomenda que sites tomem suas devidas precauções de segurança em seu roteador de borda da internet. Os endereços unique local podem ser aplicados para dispositivos que nunca precisaram, precisarão ou terão acesso por qualquer outra rede.

O endereçamento IPv6 unicast global, como o nome já diz, é exclusivo globalmente. Roteável na internet IPv6, esses endereços equivalem aos endereços públicos no IPv4. O ICANN (*Internet Commitee for Assigned Names and Numbers*), operador do IANA (*Internet Assigned Numbers Authoriry*) para a

versão IPv6, designa e aloca blocos de endereço IPv6 para cinco RIRs (Registro Regional de Internet, entidade que reúne as cinco organizações que regulamentam o uso dos endereços IP pelo mundo, formado por LACNIC, ARIN, APNIC, RIPE NCC e AFRINIC).

No entanto, atualmente são distribuídos apenas endereços unicast globais com os primeiros 3 bits iguais a 001 ou 2000::/ 3. Observe que isso representa apenas 1/8 do espaço total de endereços IPv6 disponíveis, excetuando uma parte muito pequena de outros tipos de endereços unicast e multicast.

A figura a seguir mostra a estrutura e a faixa dos endereços unicast globais.

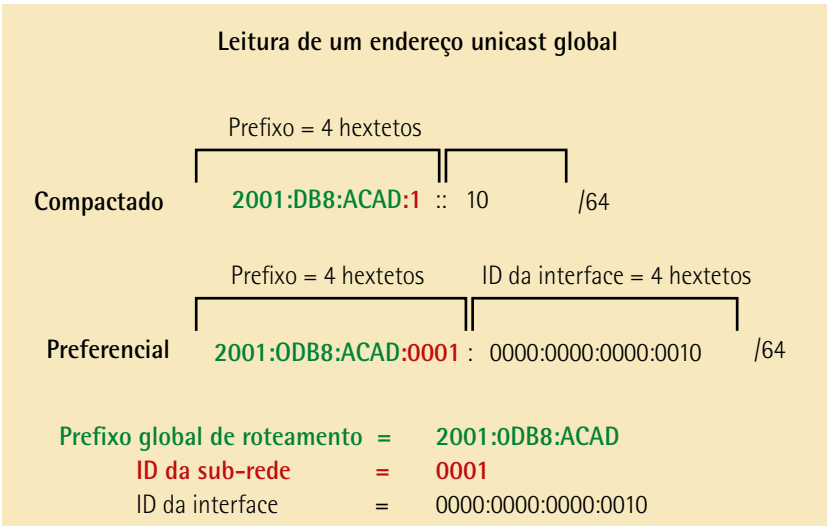


Figura 112 – Exemplo de denominação unicast global

Um endereço unicast global é formado por três partes: prefixo global de roteamento, ID da sub-rede e ID da interface.

Prefixo global de roteamento é o prefixo parte de rede do endereço IPv6 que é atribuído pelos provedores como uma ISP (*Internet Solution Provider*) diretamente a um cliente ou a um site. No momento, os RIRs atribuem o prefixo global de roteamento /48 a seus clientes, partindo de residencias até redes corporativas.

A figura a seguir mostra a estrutura de um endereço unicast global usando um prefixo global de roteamento /48. Os prefixos /48 são os prefixos de roteamento global mais comumente atribuídos.

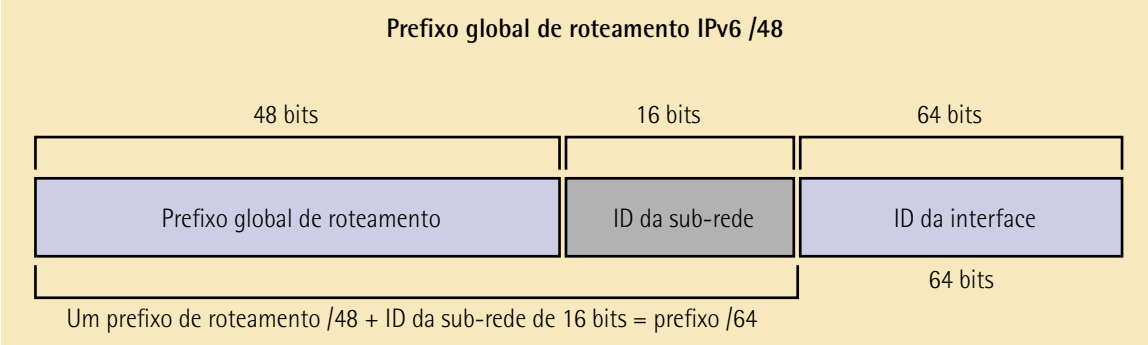


Figura 113 – Exemplo de endereço unicast global /48

Por exemplo, o endereço IPv6 2001:0DB8:ACAD::/48 tem um prefixo que indica que os primeiros 48 bits (3 hextetos: 2001:0DB8:ACAD) são o prefixo ou a parte de rede do endereço. Os dois-pontos duplos (::) antes do comprimento de prefixo /48 significam que o restante do endereço contém apenas zeros.

O tamanho do prefixo global de roteamento determina o tamanho da ID da sub-rede. A ID da sub-rede é empregada por uma empresa para identificar sub-redes locais. Quanto maior a ID da sub-rede, mais sub-redes disponíveis ela terá.

A ID da interface IPv6 é equivalente à parte de host do endereço IPv4. O termo ID de interface é usado para um único host que pode ter diversas interfaces, cada uma com um ou mais endereços IPv6. É bem provável e também recomendável que as sub-redes /64 sejam as usadas na maioria dos casos.

6.2.3 ICMP

O ICMP (*Internet Control Message Protocol* – Protocolo de Controle de Mensagens da Internet) é um protocolo de camada de rede que trabalha conjuntamente com o IP. O ICMP, entretanto, não é usado especificamente para transmissão dos dados, mas sim como protocolo de controle e de mensagens de erro (MAIA, 2013).

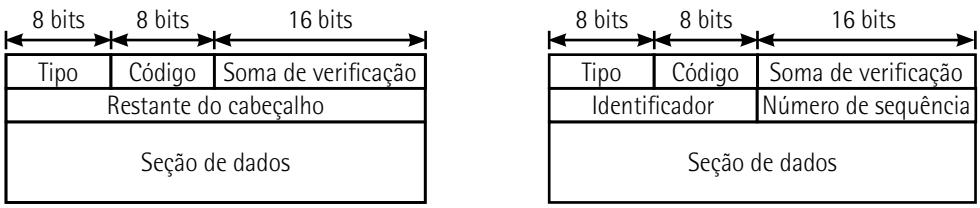
O IP, por ser um protocolo de melhor esforço, não se preocupa com mensagens de erros, qualidade, mensagens de controle e outras questões relacionadas a protocolos orientados à conexão. É justamente daí que nascem as motivações de uso do ICMP, que trabalha com dois tipos de mensagens: relatório de erro e consulta.

A funcionalidade efetiva do ICMP permite que equipamentos roteadores e ativos de rede interligados possam informar erros ou quaisquer problemas inesperados ocorridos durante uma transmissão de dados.

O ICMP é um mecanismo que informa os erros e possibilita que roteadores possam avisar às entidades transmissoras as causas de um erro. Ele, entretanto, não especifica totalmente a ação que precisa ser realizada para a correção de um erro.

Imaginemos que, durante uma transmissão, um pacote passa por vários roteadores até o seu destino. Caso o destinatário receba informações erradas sobre o roteamento, esse pacote será encaminhado para um roteador errado. Logo, o que recebeu os dados não tem condições de enviar informações de erro ao destinatário original, mas consegue avisar ao transmissor original do pacote a anomalia ocorrida. Dessa maneira, concluímos que o transmissor não tem qualquer influência sobre os problemas de roteamento que podem acontecer durante o trajeto do pacote, e também não tem condições de identificar em qual roteador aconteceu o problema.

As mensagens ICMP são encapsuladas em um pacote IP. Dois exemplos disso podem ser vistos na figura a seguir.



Valor de tipo e de código

Mensagens de relatório de erro	Mensagens de consulta
03: Destino inalcançável (códigos 0 a 15)	**08 e 00:** Pedido e resposta de eco (apenas código 0)
04: Origem extinta (apenas código 0)	**13 e 14:** Pedido e resposta de carimbo de tempo (*timestamp*)(apenas código 0)
05: Redirecionamento (códigos 0 a 3)	
11: Tempo excedido (códigos 0 e 1)	
12: Problema de parâmetro (códigos 0 e 1)	

Figura 114 – Mensagens ICMP

Adaptado de: Forouzan e Mosharraf (2013, p. 291).

Por meio do ICMP podemos utilizar duas ferramentas importantes para verificação de conectividade: o ping e o traceroute.

O ping é o utilitário de teste que usa o protocolo ICMP, além de suas mensagens de solicitação de eco e de uma resposta de eco, para aferir a conectividade entre dois hosts. O ping tem funcionalidade garantida com hosts IPv4 e hosts IPv6.

Para aferir a conectividade com outro host em uma rede, uma solicitação de eco é enviada ao host usando um comando ping. Se for o endereço específico a receber tal requisição de eco, ele enviará uma resposta de eco equivalente. Assim que a resposta de eco é recebida, o ping fornece uma resposta sobre o tempo de envio da requisição e o recebimento da resposta, que pode ser uma medida de desempenho da rede referenciada em milissegundos.

O ping normalmente tem um valor de tempo limite para sua resposta. Se a resposta não é recebida dentro do tempo que se espera, o ping informa que não houve resposta, o que significa a existência

de problemas – mas também pode indicar que recursos de segurança que são capazes de bloquear mensagens estão ativados na rede, como o bloqueio por um firewall, por exemplo.

Depois que todas as requisições estiverem encaminhadas, o ping exibirá um resumo que inclui a taxa de sucesso ou insucesso, além do tempo médio de ida e volta do pacote até o seu destino.

Existem casos especiais de teste de verificação de conectividade em que podemos usar o ping. Um deles é a aferição de configuração interna de IPv4 ou de IPv6 diretamente no host local. Para realizar o teste, faz-se um ping no endereço loopback local (127.0.0.1 para IPv4, ::1 para IPv6).

Podemos usar o ping também para testar a capacidade do host de se comunicar com a rede e com outros hosts. Basta executar o ping para o endereço IP do gateway do host. O ping no gateway indica que o host e a interface do roteador que serve basicamente como gateway estão operacionais e ativados na rede local.

Para tal teste costuma-se usar o endereço do gateway, dado que o roteador no momento está operacional. Se o endereço do cliente não responder, poderá ser enviado um ping para o endereço IP de outro host da rede local que saiba que o roteador está operacional.

Se o gateway ou algum outro host efetuar a resposta, o host local conseguirá se comunicar pela rede local. Se não houver resposta, mas outro host responder, isso pode indicar um problema com a interface do roteador que serve como gateway no momento.

Outra possibilidade é que o endereço do gateway tenha sido configurado incorretamente na configuração interna do host, ou ainda que a interface do roteador esteja plenamente operacional, mas tenha algum nível de segurança aplicado a ela que a impeça de processar ou responder solicitações ICMP como o ping.

O ping também deve ser usado para testar a capacidade de um host local de se comunicar com uma rede interconectada. Esses hosts podem fazer uso do ping a um host IPv4 operacional em uma rede remota. Se correr tudo bem, uma operação de grande parte da rede interconectada poderá ser verificada de basicamente todo o segmento interno até as bordas externas. Um ping bem-sucedido pela rede interconectada confirma também a comunicação pela rede local, o funcionamento do roteador que serve como gateway e o funcionamento de todos os outros dispositivos a ela conectados, como outros roteadores que podem estar no caminho entre a rede local e o host remoto.

O traceroute, por sua vez, é um utilitário que gera uma lista de saltos que foram sendo atingidos ao longo de um caminho. Esse relatório pode fornecer informações importantes sobre verificação e solução de eventuais erros. Caso os dados atinjam seu destino, o rastreamento lista a interface de cada roteador no caminho entre esses dois hosts. Caso ainda ocorram falhas dos dados em alguns saltos ao longo do caminho, o endereço do último roteador que responder ao rastreamento fornecerá uma indicação de onde está o problema ou das restrições de segurança que foram encontrados ao longo do percurso.

O protocolo ICMP está disponível tanto para a versão IPv4 como para a IPv6. O ICMPv4 é um protocolo de mensagens específicas para o IPv4, já o ICMPv6 oferece os mesmos serviços, porém para o protocolo IPv6, e com funcionalidades adicionais importantes na análise de tráfego.

Algumas das mensagens ICMP mais comuns, tanto para ICMPv4 e ICMPv6, são:

- **Confirmação de host**: uma mensagem proveniente do eco ICMP pode ser usada para determinar se o host está ou não operacional. O host local envia uma solicitação de eco no padrão ICMP (echo request) para um host. Se o host estiver ativo e disponível, o host de destino enviará uma resposta de eco (echo reply).

- **Destino ou serviço inalcançável**: no momento que o host ou gateway recebe um pacote que não pode ser entregue, ele pode fazer uso de uma mensagem ICMP de destino inalcançável para notificar à origem do datagrama que o destino ou serviço está inalcançável. Essa mensagem conterá um código que indica o motivo pelo qual não foi possível entregar o pacote.

- **Tempo excedido**: uma mensagem ICMPv4 de tempo excedido é usada por um roteador para indicar que um determinado pacote não pode ser encaminhado porque seu tempo de vida útil TTL (*Time To Live*) foi reduzido a zero. Caso o roteador receba um novo pacote, o campo TTL do pacote IPv4 diminui para zero, e ele então descartará o pacote e enviará uma mensagem de tempo excedido para o host da origem.

6.2.4 ARP

O ARP (*Address Resolution Protocol* – Protocolo de Resolução de Endereços) foi criado pela RFC 826 para adicionar uma funcionalidade que dá permissão aos equipamentos de rede para executar um mapeamento entre os endereços físicos e lógicos em seu segmento.

A atribuição de endereço físico é responsabilidade da camada enlace, mas, em uma comunicação enviada por uma rede, para além do endereço lógico (que é o endereço atribuído na camada de rede – por exemplo, IPv4 ou IPv6), ainda precisamos saber qual é o endereço físico correspondente para que os dados possam ser enviados corretamente, permitindo a entrega dessas informações a seu destinatário.

O ARP, na verdade, auxilia o protocolo da camada de rede, mas é implementado na camada enlace.

No momento que um dispositivo precisa conhecer o endereço físico de outro dispositivo, é construída uma mensagem do tipo broadcast internamente. A mensagem, que contém o endereço da camada de rede, é enviada pela rede para a descoberta do endereço físico do correspondente.

Essa descoberta acontece no momento em que há o retorno de uma mensagem através da rede indicando o endereço físico para onde devem ser direcionados os pacotes.

A fim de mitigar o tráfego de broadcast dentro da rede, os equipamentos constroem uma tabela ARP que armazena temporariamente essa associação de endereço físico e lógico dos dispositivos conhecidos

dentro da rede. Então, em vez de constantemente enviar uma solicitação de ARP pela rede, o dispositivo antes verifica a própria tabela ARP.

A figura a seguir apresenta o fluxograma que demonstra o funcionamento do protocolo ARP.

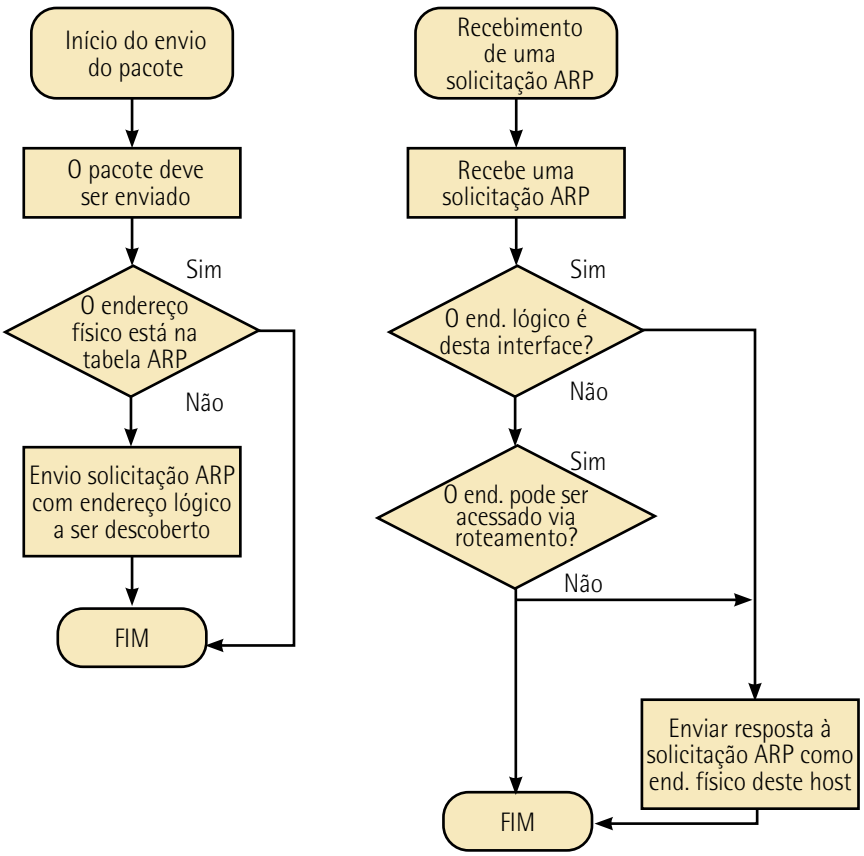


Figura 115 – Fluxograma de funcionamento do ARP

7 PADRÕES E PROTOCOLOS DE CAMADA DE TRANSPORTE

Após conhecermos as camadas mais inferiores (física, de enlace de dados e de redes), passemos agora para a camada de transporte, chamada de camada 4 do modelo híbrido aqui abordado.

O objetivo deste tópico é abordar as funcionalidades de transporte nas redes de computadores e os seus dois protocolos mais conhecidos: TCP e UDP.

7.1 Serviços da camada de transporte

7.1.1 A camada de transporte

Para conhecer os detalhes da camada de transporte, é preciso recordar-se um pouco sobre a camada de rede, vizinha no limite inferior da arquitetura de redes. Na camada de rede, que normalmente utiliza

o IP como protocolo roteável, não há garantia de que os dados e pacotes cheguem ao seu destino, porque a abordagem de datagrama e a falta de orientação a conexão domina a "cena" da camada 3.

Também observamos que na camada de rede há uma preocupação com as rotas que serão trilhadas e o encaminhamento de pacotes que precisa ser feito. Tudo isso é provido por algoritmos e protocolos de roteamento que favorecem a conectividade entre hosts.

No entanto, não há uma preocupação com os mais variados dados gerados pelo host a respeito da utilização de aplicações, e também não há uma estratégia de comunicação fim a fim. São justamente essas e outras funcionalidades providas pela camada de transporte, que isola os níveis superiores da transmissão da rede.

A entidade do nível de transporte da máquina que origina a comunicação se comunica com a entidade do nível de transporte da máquina a que se destina a comunicação. Isso não necessariamente acontece em níveis físicos do enlace ou da rede onde a comunicação se dá entre máquinas adjacentes ou máquinas vizinhas na sua rede. Essa funcionalidade da camada de transporte é conhecida como comunicação processo-a-processo e está descrita na figura a seguir.

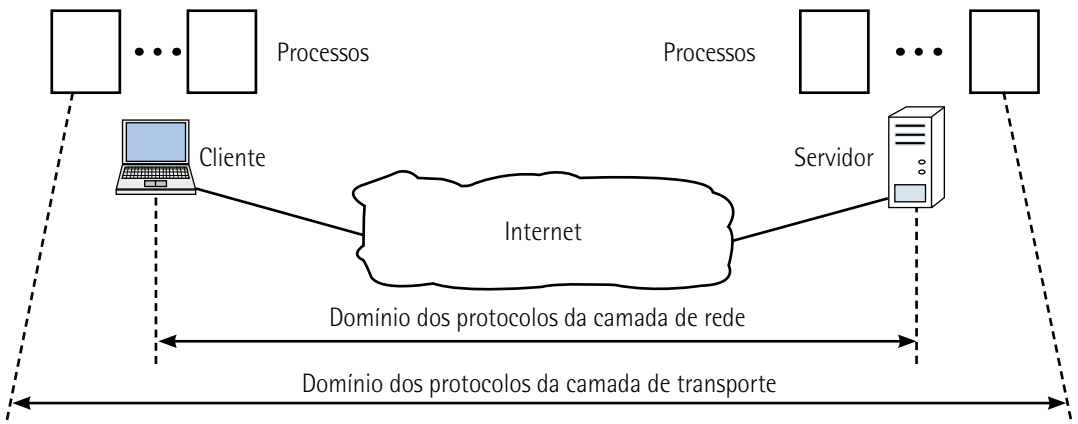


Figura 116 – Comunicação processo-a-processo na camada de transporte

Adaptado de: Forouzan e Mosharraf (2013, p. 141).

7.1.2 Números de porta

A camada de transporte recebe toda a massa de dados oriunda da camada de aplicação e as encapsula em segmentos. O usuário que gera esses dados normalmente trabalha ao mesmo tempo com diversos aplicativos, e cada um deles gera fluxos de dados diferentes que não podem se misturar nem ser confundidos, sob pena de grandes prejuízos no processo de comunicação.

Lembrete

A PDU da camada de transporte é o segmento.

Dessa forma, a camada de transporte vai identificando o fluxo de dados oriundo de cada uma das aplicações com um identificador chamado de número de porta, habilitando comunicações simultâneas. Números de portas são utilizados para identificação dessas comunicações pelas diversas aplicações do usuário.

Assim, quando o dispositivo inicia uma comunicação, ele atribui um número de porta de origem e outro número de porta para o destino. Essa porta de origem identifica a comunicação na sua origem enquanto a porta do destino vai identificar a aplicação, que vai receber a informação ao seu destino. No retorno da comunicação, esses números são trocados sistematicamente.

As portas da camada de transporte são classificadas em: conhecidas, registradas e privadas. As portas conhecidas (0 a 1023) estão entre aquelas garantidas pela IANA (*Internet Assigned Numbers Authority*) para os principais e primeiros serviços e aplicações de rede. As portas registradas (1024 a 49151) não estão sob gestão e controle da IANA, e são atribuídas as aplicações não controladas. As portas privadas, também chamadas de dinâmicas (49152 a 65535), são disponibilizadas de forma temporária pelo sistema operacional todas as vezes em que há a necessidade de uma identificação de porta.

A figura a seguir apresenta o exemplo de uma comunicação entre hosts de origem e de destino, e a atribuição de número de portas.

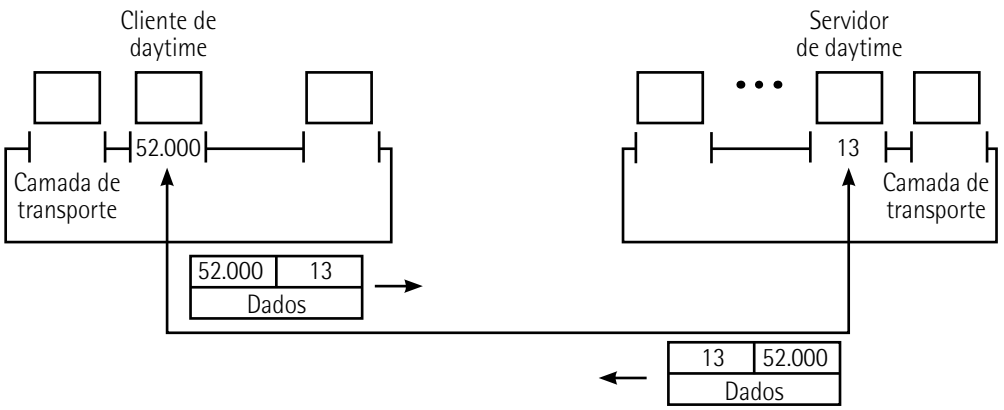


Figura 117 – Atribuição de número de portas

Adaptado de: Forouzan e Mosharraf (2013, p. 142).

Observe que no exemplo desta figura temos uma aplicação em um host designado "cliente de daytime" que está se comunicando com uma aplicação de destino situada no "servidor de daytime". O fluxo de dados da aplicação de origem é encapsulado em um segmento e recebe um número de porta de origem igual a 52.000 (perceba que é uma porta dinâmica/privada) e número de porta de destino igual a 13 (perceba que é uma porta conhecida).

Observação

Quando unimos o número de porta constante no cabeçalho do segmento com o endereço IP situado no cabeçalho do pacote, temos um soquete.

7.1.3 Funcionalidade de multiplexação e de demultiplexação

Outra funcionalidade da camada de transporte é multiplexação (no host de origem) e a demultiplexação (no host de destino). Sobre estas tarefas, Forouzan e Mosharraf (2013, p. 144) mencionam que:

> Sempre que uma entidade aceita itens de mais de uma origem, temos algo denominado multiplexação (muitos para um); sempre que uma entidade encaminha itens a mais de uma origem, temos algo denominado demultiplexação (um para muitos).

Para exemplificar essa tarefa de transporte, a figura a seguir apresenta três processos: P1, P2 e P3. Os processos P1 e P3 sinalizam uma comunicação fim-a-fim diferente do processo P2. Para que haja a transmissão "simultânea" de dados oriundos destes três processos, a camada de transporte opera com a multiplexação.

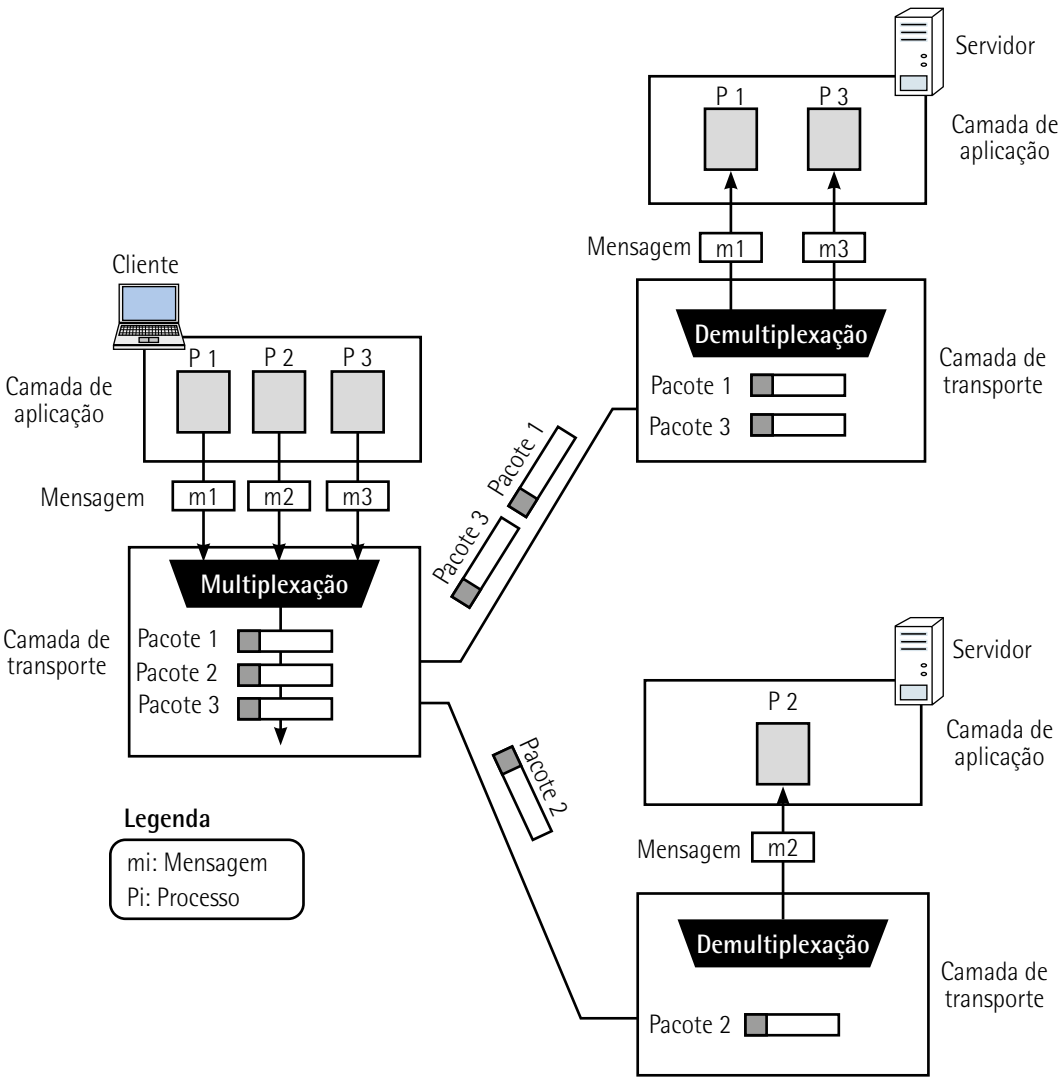


Figura 118 – Multiplexação e demultiplexação na camada de transporte

Adaptado de: Forouzan e Mosharraf (2013, p. 143).

7.1.4 Serviços prestados pela camada de transporte

Na camada de transporte, a depender do protocolo utilizado, trabalhamos com o serviço orientado à conexão e o serviço não orientado à conexão. Quando trabalhamos com o protocolo TCP, há uma orientação à conexão. Quando temos o protocolo UDP, a camada de transporte opera com o serviço não orientado à conexão.

O serviço orientado à conexão tem grande preocupação com a qualidade da transmissão dos segmentos, atuando com processos que serão detalhados mais adiante, tais como: janelamento; controle de erros; correção de erros; controle de fluxo; estabelecimento de sessão; encerramento de sessão; entre outros. Já no serviço orientado à conexão, a grande preocupação está voltada para a velocidade, prescindindo da qualidade.

7.2 TCP e UDP

7.2.1 TCP

O TCP (*Transport Control Protocol* – Protocolo de Controle de Transmissão) é o protocolo orientado à conexão da camada de transporte. Ele é requisitado por aplicações que necessitam de confiabilidade no transporte de segmentos.

Como primeira característica do TCP temos o estabelecimento de sessão entre destino e origem antes de transmitir dados. Após essa sessão ser estabelecida, os dados poderão ser transmitidos, e após o término da transmissão dos dados a sessão será encerrada na camada de transporte.

O estabelecimento da sessão se dá por meio do *handshake* triplo, que consiste na sincronização iniciada pelo cliente ao servidor em três fases:

- **Fase 1**: a entidade que está iniciando a comunicação que transmite o segmento contendo o número de sequência para inicialização, indicando o início da comunicação com um ==> SYN inicial.

- **Fase 2**: a entidade receptora responde com um ==> SYN/ACK, confirmando o estabelecimento da comunicação.

- **Fase 3**: a entidade que iniciou a comunicação responde a confirmação completando a fase de estabelecimento e a sincronização da comunicação.

A figura a seguir esboça o estabelecimento de sessão na camada de transporte por meio do *handshake* triplo.

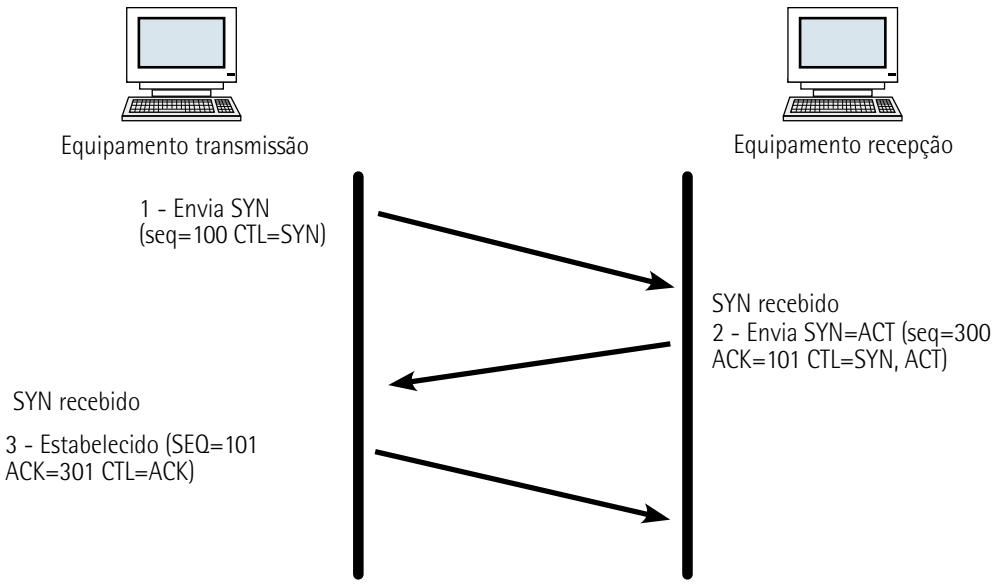


Figura 119 – Estabelecimento de sessão

No momento em que a comunicação é estabelecida, essa fase já se encontra concluída e os dados podem ser transmitidos. Somente após o *handshake* triplo os dados são enviados pela entidade de origem e, depois de transmitidos, a sessão precisa ser encerrada. A figura a seguir apresenta o encerramento de sessão.

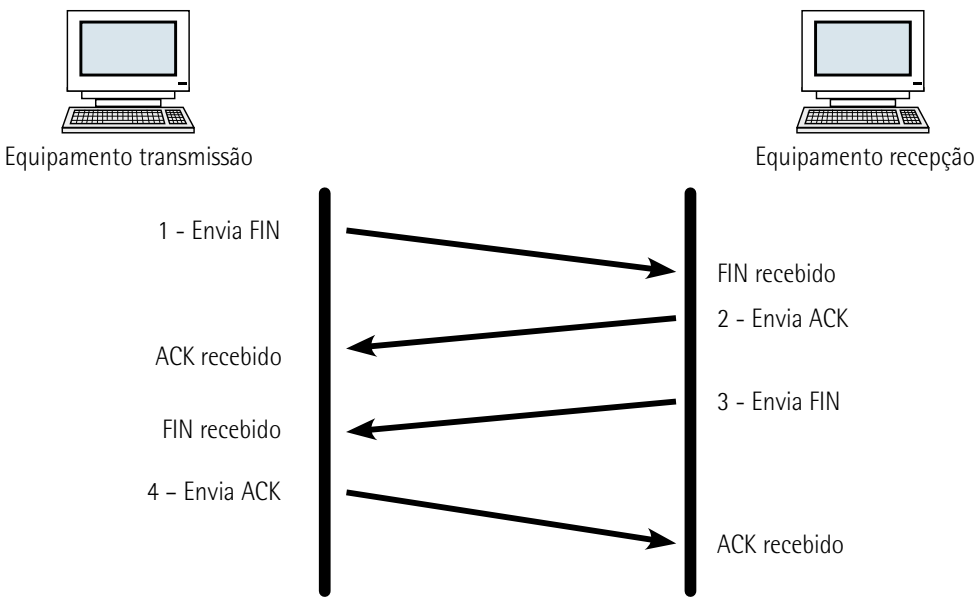


Figura 120 – Encerramento de sessão

Outra atividade desenvolvida pelo TCP é a entrega ordenada. Em uma comunicação, quando diversos segmentos são enviados entre a entidade de origem e a entidade de destino, a chegada deles ao seu destino pode ser encarada de forma desordenada (justamente pelas diversas possibilidades de rota que estão disponíveis em uma comunicação em rede). Para que eles possam ser organizados e ordenados ao

seu destino, cada segmento recebe um número de sequência. Quando esses segmentos chegam fora da ordem original, eles são colocados em um *buffer* para que, depois de organizados e ordenados, possam ser entregues às camadas superiores.

A figura a seguir apresenta a transmissão de três segmentos na forma de três mensagens. Encapasulados em pacotes, eles tomam caminhos diferentes e chegam fora de ordem ao destino. Ao trabalhar com o TCP, o destino faz um reordenamento desses segmentos.

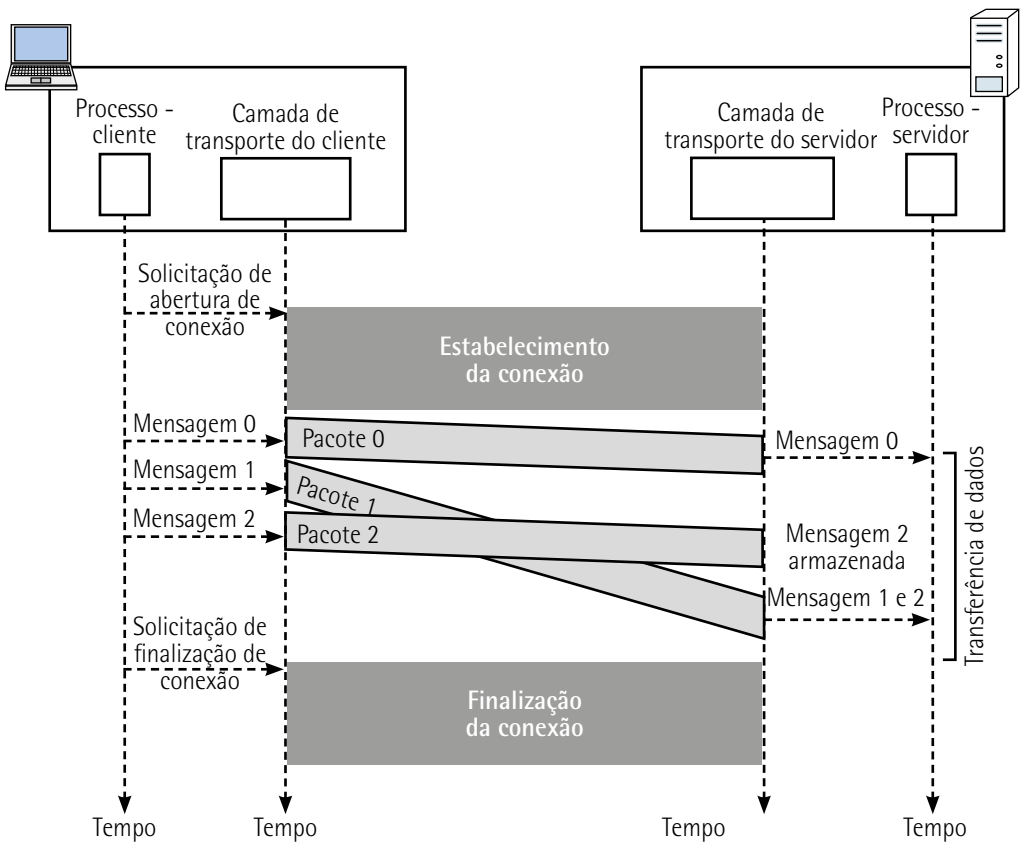


Figura 121 – Reordenamento de segmentos no destino

Adaptado de: Forouzan e Mosharraf (2013, p. 152).

Para garantir a confiabilidade em uma comunicação, a camada de transporte, ao operar com TCP, utiliza o conceito de confirmação positiva ou confirmação esperada. Nesse caso, são usados números sequenciais juntamente com os números de confirmações (ACK). Ao receber esses datagramas que foram enviados pela entidade de origem, a entidade de destino confirma o recebimento e pede pelo próximo datagrama da fila; isso significa que a entidade de origem entende que a entidade de destino recebeu todos os datagramas anteriores. A figura a seguir apresenta essa funcionalidade.

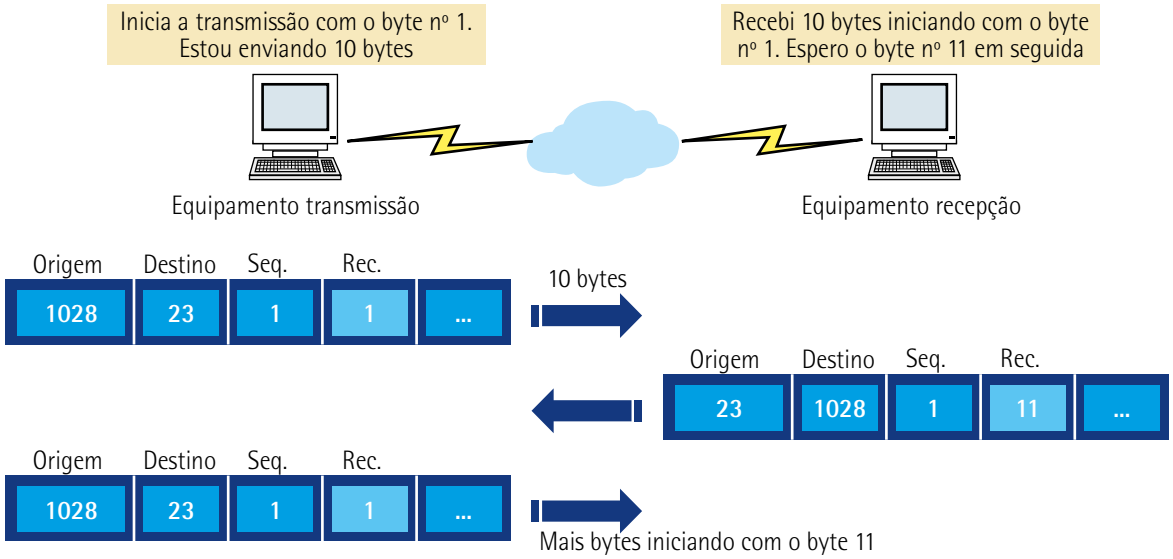


Figura 122 – Confirmação positiva

O controle de fluxo das informações, por meio do janelamento, é uma atribuição da camada de transporte quando o protocolo utilizado é o TCP. Por meio desse controle, indica-se a quantidade de informação que poderá ser transferida antes da confirmação de recebimento por seu destino. A camada de transporte então faz uso do janelamento para essa função.

O janelamento é considerado uma janela móvel, também conhecida como janela deslizante (ou seja, o valor do tamanho da janela não é fixo), em que os valores são alterados durante a transmissão. Assim, o fluxo das informações é gerenciado conforme ocorre o controle de fluxo. A figura a seguir apresenta o controle de fluxo com o janelamento.

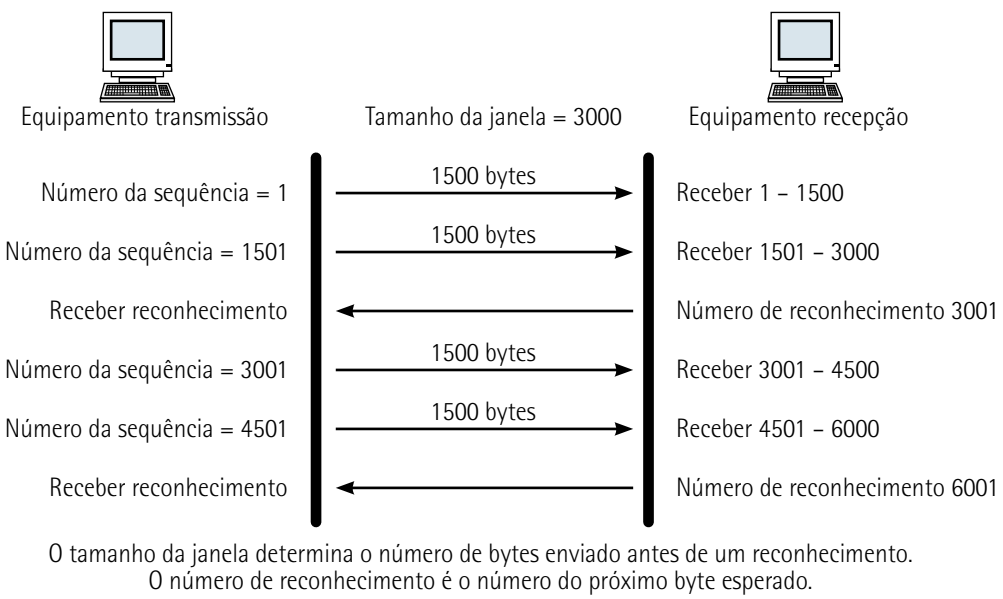


Figura 123 – Controle de fluxo com janelamento

7.2.2 Segmento TCP

Para prover a orientação à conexão, o segmento TCP é dotado de um cabeçalho relativamente grande se comparado ao utilizado pelo UDP. A figura a seguir apresenta o segmento TCP.

Bit 0 — Bit 15	Bit 16 — Bit 31
Porta de origem (16)	Portal de destino (16)
Número de sequência (32)	
Número de reconhecimento (32)	
Comprimento do cabeçalho (4) Reservado (6) bits de código (6)	Janela (16)
Checksum (16)	Urgente (16)
Opções (0 ou 32, se houver)	
Dados da camada de aplicativos (tamanho varia)	

Figura 124 – Segmento TCP

Os campos que constam no segmento TCP são:

- **Porta de origem**: campo de 16 bits que contém o número da porta de origem.
- **Porta de destino**: campo de 16 bits que contém o número da porta de destino.
- **Número de sequência**: campo de 32 bits utilizado para ordenar os datagramas.
- **Número de reconhecimento**: campo de 32 bits com o número de confirmação que indica o próximo segmento TCP esperado.
- **Comprimento do cabeçalho**: campo de 4 bits que indica o tamanho do cabeçalho do datagrama.
- **Bits de código**: bits utilizados para determinar o tipo de segmento.
- **Janela**: campo de 16 bits com o número de segmentos que poderão ser transmitidos antes de aguardar uma confirmação.
- **Checksum**: campo de 16 bits para o cálculo de verificação de erros.
- **Dados**: campo com os dados das camadas superiores.

7.2.3 UDP

A camada de transporte nem sempre precisa oferecer um serviço confiável. Em alguns casos, em que a confiabilidade da comunicação não é necessária, um protocolo não orientado à conexão pode ser usado. O protocolo de camada de transporte que pode fornecer o serviço não orientado à conexão é o UDP (*User Datagram Protocol* – Protocolo de Datagrama do Usuário).

A utilização do UDP, que deixa de lado as questões relacionadas à confiabilidade, é mais rápido na transmissão de dados justamente por não executar tarefas como janelamento, entrega ordenada, correção de erros, retransmissão automática e outros.

A figura a seguir apresenta a entrega de datagramas feita pelo UDP ao destino sem qualquer ordenamento.

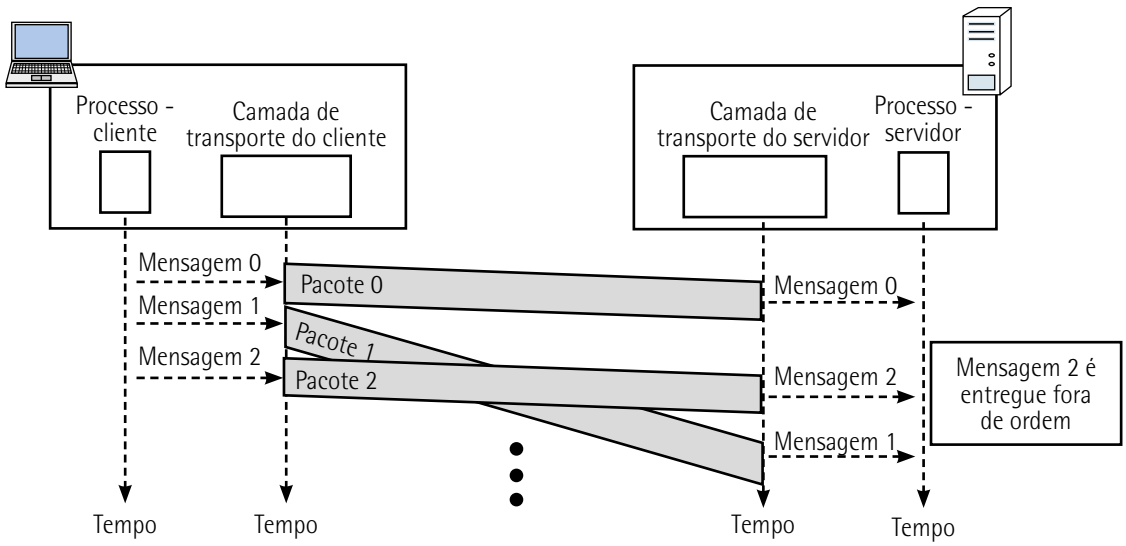


Figura 125 – Entrega de datagramas UDP

Adaptado de: Forouzan e Mosharraf (2013, p. 151).

Observação

Ao trabalhar com o UDP, a PDU (*Protocol Data Unit* – Unidade de Dados de Protocolo) da camada de transporte é chamada de datagrama.

Tudo isso faz com que o UDP seja um protocolo "mais leve", que não tem a necessidade de um datagrama com um cabeçalho grande. A figura a seguir apresenta o datagrama UDP.

Bit 0	Bit 15	Bit 16	Bit 31
Porta de origem (16)		Portal de destino (16)	
Comprimento (16)		Checksum (16)	
Dados da camada de aplicativos (tamanho varia)			

Figura 126 – Datagrama UDP

Os campos que constam no datagrama UDP são:

- **Porta de origem**: campo de 16 bits que contém o número da porta de origem.
- **Porta de destino**: campo de 16 bits que contém o número da porta de destino.
- **Comprimento**: campo de 16 bits que indica o tamanho do datagrama, incluindo os dados.
- **Checksum**: campo de 16 bits para o cálculo de verificação de erros.
- **Dados**: campo com os dados das camadas superiores.

7.2.4 Diferenças entre o TCP e o UDP

Podemos observar que os protocolos TCP e UDP têm semelhanças e diferenças. Em primeiro lugar, vale lembrar que a função deles é basicamente a mesma – ou seja, o transporte de dados das camadas superiores entre os dispositivos finais e a diferenciação das diversas conversações em formato simultâneo por meio de números de portas. Os dois protocolos dispõem de campos de números de portas e de checksum, e também campos de dados com funções equivalentes.

As semelhanças, porém, param por aí. Podemos observar que o protocolo TCP tem mais campos do que o UDP – exatamente pelo fato de o TCP oferecer serviços orientados à conexão com confiabilidade.

Além de o TCP dispor de um cabeçalho muito maior do que o UDP, são 20 bytes para o TCP e 8 bytes para o UDP: o *overhead* que o protocolo TCP impõe é bem maior, ou seja, o protocolo UDP é bem mais leve.

Por essas razões, o protocolo UDP pode ser usado em princípios de comunicação nos quais não é necessário existir a confiabilidade, embora isso não seja recomendado.

8 PADRÕES E PROTOCOLOS DE CAMADA DE APLICAÇÃO

Depois de ter trilhado um caminho passando por quase todas as camadas do nosso modelo híbrido, chegamos à última: a camada de aplicação. Ela é considerada a camada mais próxima do usuário, fornecendo serviços e protocolos para as tarefas do cotidiano envolvendo as redes de computadores.

A camada de aplicação oferece o serviço para que as aplicações dos usuários possam interagir com elementos da rede. A camada é composta por protocolos que possibilitam a comunicação entre as aplicações; então, quando uma aplicação precisa de um protocolo específico dessa camada, ela usará esse protocolo para codificar os dados e encaminhá-los à camada subsequente: a camada de transporte.

Dentre os vários protocolos que fazem uso dessa camada, podemos citar o SMTP (*Simple Mail Transfer Protocol*) para serviços de entrega de mensagens de e-mail, o DNS (*Domain Name System*) para resolução de nomes de internet, o FTP (*File Transfer Protocol*) para transferência de arquivos, o HTTP (*Hyper Text Transfer Protocol*) para navegação em páginas web, entre outros.

8.1 Protocolos e serviços da camada de aplicação

8.1.1 Protocolo de Transferência Hipertexto

O HTTP ((*Hyper Text Transfer Protocol* – Protocolo de Transferência de Hipertexto) é um dos mais importantes e conhecidos protocolos da camada de aplicação. Ele está associado aos serviços web oferecidos aos usuários e define como o cliente web (browser) requisita uma página web a um servidor, e como esse servidor transfere a página para o cliente.

O HTTP utiliza o protocolo TCP como protocolo de transporte: a mensagem sai de suas mãos e passa para as mãos do TCP. Com essa ajuda, o TCP provê ao HTTP um serviço confiável de transferência de dados, que garante que todas as mensagens de requisição HTTP emitidas por um processo do cliente chegarão intactas ao servidor.

Da mesma forma, todas as mensagens emitidas pelo servidor chegarão intactas ao cliente. O HTTP não precisa se preocupar com eventuais dados perdidos, nem com os detalhes de como o TCP os recupera.

Essas informações que são enviadas entre clientes e servidores não são armazenadas. Por isso, se um cliente solicitar o mesmo objeto duas vezes, o servidor não informará que esse objeto já foi enviado; ele o enviará novamente.

Como o HTTP não mantém nenhuma informação sobre o cliente, ele é identificado como um protocolo sem estado. A primeira versão, conhecida como HTTP 1.0 e criada por volta da década de 1990, deu rapidamente lugar à versão 1.1 em 1997, colocando o HTTP na posição de protocolo mais popular no início da internet. A versão 2 (HTTP/2) surge em 2015, e logo é substituída pela versão 3 (HTTP/3) por volta de 2021, compatível com o novo protocolo de transporte QUIC.

A ideia principal do HTTP é a definição de regras para o carregamento de páginas da internet em um navegador e oriundas de um servidor. A figura a seguir ilustra a operação do HTTP.

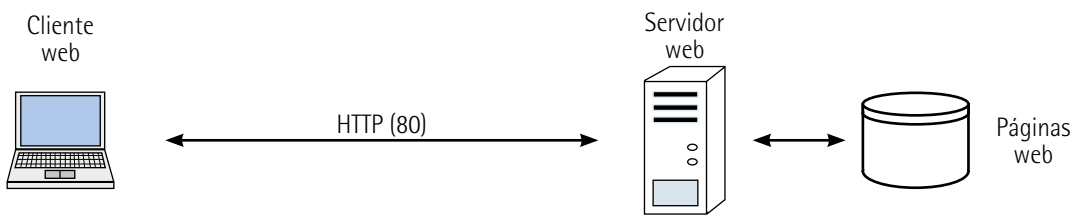


Figura 127 – Operação do HTTP

Adaptado de: Maia (2013, p. 245).

Sabemos que os servidores HTTP são classificados como serviço, mas seria interessante que os sites web identificassem seus usuários. Para que isso aconteça, é necessária a utilização dos cookies, que permitem que os sites monitorem seus usuários.

Grande parte dos portais (por exemplo, <www.google.com> e <www.msn.com>) e sites de comércio eletrônico (por exemplo, <www.ebay.com>) faz uso intensivo de cookies.

O cookie é formado pelos seguintes componentes:

- Uma linha de cabeçalho de cookie na mensagem de resposta do HTTP.
- Uma linha de cabeçalho de cookie na mensagem de requisição do HTTP.
- Um arquivo de cookie mantido no computador do usuário e gerenciado pelo browser.
- Um banco de dados de apoio no site web.

O funcionamento dos cookies se dá da seguinte forma: vamos supor que você deseja comprar algum produto na loja on-line Submarino. Quando você acessa o site pela primeira vez, é criado um número de identificação exclusivo que será armazenado no seu computador e criará uma entrada no banco de dados do servidor da loja. Esse número o identificará.

Toda vez que você acessar o site, seu browser vai consultar a identificação no arquivo de cookies e inseri-la no cabeçalho HTTP de requisição. Com isso, o site web pode monitorar se é você mesmo que o está acessando novamente.

Os sites de comércio eletrônico utilizam bastante os cookies em razão dos carrinhos de compra. Eles podem recomendar produtos com base em suas buscas na última visita ou armazenar os produtos que você adicionou no carrinho e não comprou.

Os cookies podem ser utilizados para criar uma camada de sessão de usuário sobre o HTTP, que é sem estado. Por exemplo: quando você acessa uma aplicação de webmail, o browser envia suas informações de cookie ao servidor, que, por sua vez, o identifica por meio da sessão do usuário com a aplicação.

8.1.2 Telnet

O Telnet é comumente utilizado para estabelecer uma conexão on-line com uma máquina remota, é suportado por inúmeras aplicações de rede e também é entendido como uma aplicação auxiliar.

Trata-se de um software de emulação de terminal que permite o acesso de forma remota a outro computador. Ele permite que um comando de logon seja executado em uma máquina da internet e efetue comandos usando a sintaxe adequada. O cliente Telnet é chamado de máquina local, e um servidor Telnet é chamado de máquina remota.

Ao fazer uma conexão de um cliente Telnet, é preciso escolher uma opção de conexão. Uma caixa de diálogo solicita um nome de host e um tipo de terminal. O nome do host, ou máquina remota, é o endereço IP (ou solução de nome correspondente) do computador remoto ao que se deseja conectar. O tipo de terminal descreve o modo de emulação terminal que se deseja executar pela máquina local.

A operação Telnet não usa nenhuma capacidade de processamento da máquina local. Em vez disso, ela transmite as teclas pressionadas à máquina remota e envia a saída de tela resultante de volta ao monitor local. Todo o processamento e todo o armazenamento ocorrem na máquina remota.

O Telnet é iniciado como um processo de correio eletrônico. Quando é inserido um nome de DNS para um local do Telnet, o nome deverá ser convertido em seu endereço IP associado antes de estabelecer uma conexão (o que resulta na resolução de nome-para-número, ou URL correspondente).

A aplicação Telnet trabalha principalmente nas três camadas superiores do modelo OSI (*Open System Interconnection*), a camada de aplicação (comandos), a camada de apresentação (formatos, normalmente ASCII) e a camada de sessão (transmissões). Seus dados passam para a camada de transporte, onde são segmentados e lhe são acrescidos o endereço da porta e a verificação de erros. Os dados passam, então, para a camada de rede, em que o cabeçalho IP (contendo o endereço IP de origem e de destino) é adicionado. Depois, o pacote trafega para a camada de enlace, que encapsula o pacote em um quadro de dados, adiciona o endereço MAC (*Media Access Control* – Controle de Acesso de Mídia) de origem e de destino e um trailer de quadro.

Se o computador de origem não tiver o endereço MAC do computador de destino, ele executará uma solicitação ARP. Após a identificação do endereço MAC, o quadro trafegará pelo meio físico (na forma binária) para o próximo dispositivo.

Quando os dados chegarem à máquina remota, as camadas de enlace, de rede e de transporte passarão pelo reagrupamento dos comandos de dados originais. A máquina remota, então, executa os comandos e transmite os resultados de volta para a tela da máquina local, usando o mesmo processo de encapsulamento que entregou os comandos originais. Todo esse processo se repete, enviando comandos e recebendo resultados, até que o usuário local tenha concluído o trabalho que precisava ser executado. Após a conclusão do trabalho, o usuário local terminará a sessão.

8.1.3 DNS

Existem várias maneiras de identificar pessoas: por meio do nome, dos números de CPF, RG etc. Cada uma dessas maneiras se enquadra em um contexto apropriado. A universidade, por exemplo, identifica seus estudantes pelo número de matrícula em vez do número de documento (RG ou CPF); já as pessoas preferem identificar seus amigos pelo nome, que é bem mais fácil de ser lembrado do que o RG. Imagine alguém sendo chamado pelo número do RG; ninguém se entenderia.

Da mesma maneira que podemos ser identificados de formas diferentes, os hosts conectados pela internet também podem. Nomes como www.google.com, www.globo.com, www.unip.br etc. são fáceis de lembrar, e por isso são bem usados pelos usuários. Esse tipo de identificação, porém, fornece poucas informações sobre a localização desses hosts. Como os caracteres utilizados nos nomes são variáveis, torna-se complexo o processamento deles pelos roteadores, e, por esse motivo, os hosts também podem ser identificados por endereços IP.

Para que ocorra uma forma fluida de solução de endereço IP e identificação através dos nomes com caracteres, é necessário um serviço de diretório que execute a tradução dos nomes para os endereços IP. Essa é a função do DNS (*Domain Name System* – Sistema de Nome de Domínios).

O DNS pode ser visto como um grande banco de dados distribuído e integrado por meio de uma hierarquização de servidores de nomes, chamados de servidores DNS. Ele tem a assistência de um protocolo da camada de aplicação que permite que os hosts consultem o banco de dados de informações.

Os serviços, entidades e protocolos da camada de aplicação que utilizam o DNS são: HTTP, SMTP, FTP etc. Elas fazem uso do DNS para traduzir nomes de hosts fornecidos por usuários para o endereço IP.

Por exemplo: quando você digita no navegador a URL www.yahoo.com, acontecem os seguintes passos:

- **Passo 1**: sua máquina executa o lado cliente da aplicação DNS.
- **Passo 2**: o navegador passa o nome do host <www.yahoo.com> para o lado cliente da aplicação.
- **Passo 3**: o cliente DNS envia uma consulta para o servidor DNS contendo o endereço <www.yahoo.com>.
- **Passo 4**: o servidor DNS envia uma resposta para o cliente contendo o IP do host desejado.
- **Passo 5**: depois de receber o endereço, o navegador abre uma conexão TCP com um processo HTTP localizado naquele endereço IP resolvido.

Como vemos, acontece uma troca de mensagens entre o servidor e o cliente DNS, mas existe algum atraso para as aplicações de internet que utilizam os serviços de DNS. Para mitigar esse problema, os endereços IP que são procurados com frequência são armazenados no cache de servidores de DNS mais próximos, o que ajuda a diminuir o tráfego e o atraso.

Da mesma forma, como os protocolos HTTP, FTP e SMTP, o DNS também é um protocolo da camada de aplicação, só que seu papel é diferente dos demais, dado que ele não é uma aplicação com a qual os usuários atuam diretamente. Em vez disso, ele fornece uma ação interna da internet, que é a tradução de nome-para-número IP.

Existem outros serviços pelos quais o DNS é o responsável:

- **Apelidos dos hosts**: por vezes, os hosts possuem algum nome complexo ou complicado, ou mais de um nome. Um nome como zonaX.setor-Y.empresa.com.br pode ainda dispor de dois ou mais apelidos, como www.empresa.com.br e empresa.com.br. Os apelidos são bem mais fáceis de lembrar, e, por isso, o DNS pode ser solicitado para obter o nome real do host a partir de seu apelido.

- **Apelido do servidor de correio**: da mesma forma que os apelido dos hosts, o importante é que o nome de um e-mail seja simples de ser memorizado. Aqueles que possuem uma conta no Yahoo, por exemplo, podem ter o seguinte e-mail: usuario@yahoo.com.br. O servidor de hospedagem do Yahoo, porém, pode ter um nome complicado, como zona99.setor-y.yahoo.com.br. O DNS é acionado pela aplicação de correio eletrônico para receber o nome real a partir do apelido que é fornecido e do endereço IP do servidor.

- **Distribuição de cargas**: o serviço DNS é requisitado para distribuir cargas em sites que são muito utilizados, como o Google. Esse balanceamento é feito com o uso de vários servidores que usam IPs diferentes. Todo o conjunto de IPs desses servidores é associado ao nome real do site e armazenado na base de dados do DNS. Quando a máquina cliente do DNS solicita o endereço do site, o servidor de DNS oferece um conjunto de endereços IP a ele associado, mas efetua um balanceamento na ordem dos endereços a cada solicitação. Esse balanceamento força a distribuição de tráfego pelos vários servidores replicados ao serviço.

8.1.4 Outros protocolos de aplicação

Existem diversos protocolos na camada de aplicação. Especificamente para a transferência de arquivos, existem o FTP e o TFTP.

O FTP (*File Transfer Protocol*) é um protocolo que tem como finalidade transferir arquivos de um computador para o outro, copiando e movendo arquivos dos servidores para os clientes e vice-versa. Por ser um protocolo confiável e orientado à conexão, o FTP garante que as informações serão entregues ao destino.

O TFTP (*Trivial File Transfer Protocol*) é uma variante do protocolo FTP de mesma finalidade, ou seja, transferir arquivos. A principal diferença entre esses protocolos é que o TFTP não é confiável, e também não é orientado à conexão – ou seja, não existe garantia na entrega da informação.

Por essa razão, o TFTP é mais rápido do que o FTP por não utilizar recursos que garantam a entrega dos dados. Por outro lado, o FTP é muito mais seguro e confiável.

Partindo para os serviços de correio eletrônico ou e-mail, encontramos mais três protocolos: SMTP, POP e IMAP. O SMTP (*Simple Mail Transfer Protocol*) é o protocolo usado para transferir e-mails entre servidores e também pelo aplicativo cliente para enviar e-mails. Os protocolos POP (*Post Office Protocol*) e IMAP (*Internet Message Access Protocol*) são usados pelo aplicativo cliente para baixar um e-mail do servidor local.

Para o gerenciamento de redes, encontramos um protocolo chamado SNMP (*Simple Network Management Protocol*), que tem a função de trocar informações de gerenciamento entre os dispositivos de uma determinada rede. O SNMP ajuda os administradores de rede a gerenciá-la de forma otimizada: mensagens de alerta são enviadas para o computador que gerencia a rede, e são armazenadas em base de dados de coleta de informações para registro histórico de atividade dos ativos e serviços gerenciados pelo protocolo.

Os elementos que compõem a operação do SNMP são:

- **Entidade de gerenciamento**: também chamada de NMS (*Network Management Systems*), é a responsável pela aplicação principal, ou seja, é a que gerencia a rede. Ela geralmente é instalada em um servidor dedicado.

- **Dispositivos gerenciados**: são os dispositivos que são gerenciados pelo protocolo SNMP. Exemplos de dispositivos gerenciados são: roteadores, switches, servidores, impressoras, estações de trabalho etc.

- **Agentes**: são módulos de software de gerenciamento de rede que residem em dispositivos gerenciados. Um agente tem conhecimento local das informações de gerenciamento e as converte para uma forma compatível com o SNMP.

Saiba mais

Para conhecer um pouco mais sobre os protocolos da camada de aplicação, leia o capítulo 6, "A camada de transporte", do livro a seguir:

TANENBAUM, A.; FEAMSTER, N.; WETHERALL, D. *Redes de computadores*. Tradução: Daniel Vieira. 5. ed. Rio de Janeiro: Pearson Prentice Hall, 2021.

8.2 Segurança em redes de computadores

8.2.1 Fundamentos de segurança em redes de computadores

Muito antes de trafegar nas redes, as informações têm características que devemos conhecer. Elas são constituídas por um conjunto de dados agrupados de maneira lógica e que, de alguma forma, agregam valor a pessoas, processos ou organizações.

As redes são vitais no processo de transporte e troca de informações, adicionando mais valor a essas informações de forma direta ou indireta. Quando nos referimos à segurança, a transição de um ponto a outro das informações é o seu maior risco.

Os desafios dos profissionais que trabalham no ambiente tecnológico de redes são inúmeros, como proteger as informações que trafegam na rede sob sua gestão a fim de evitar vazamentos, furtos e falhas. Assim, é necessário atentar aos fatores humanos, aos processos e às tecnologias.

Todas as ameaças que rondam a segurança das informações visam explorar as vulnerabilidades desses elementos. À primeira vista, podemos imaginar que devemos nos preocupar apenas com as ameaças que exploram os elementos tecnológicos, mas essa é uma meia-verdade. Dado que os elementos que formam o conjunto devem sempre ser analisados na implantação dos mecanismos

de proteções em redes, deixar de avaliar um desses elementos pode dar uma visão incorreta e falsa sensação de segurança, o que pode levar a incidentes de segurança da informação.

Mas o que é mesmo a segurança da informação?

Moraes (2020, p. 19) afirma que:

> a segurança da informação pode ser definida como um processo de proteger a informação do mau uso tanto acidental como intencional, por pessoas internas ou externas à informação, incluindo empregados, consultores e hackers.

Para compreender melhor os processos de proteção da informação, é preciso entender os agentes diretamente envolvidos que influenciam todas as decisões na segurança das informações.

O valor da informação é definido pelo seu grau de importância para a organização. Isso parece ser lógico e simples, mas não é: o valor da informação deve ser definido por um conjunto de fatores que inclui seu valor financeiro, de imagem, sua importância estratégica, o atendimento às leis locais e internacionais etc. Isso necessariamente levará à classificação das informações e da análise de riscos e, posteriormente, à priorização de investimentos em mecanismos de proteção.

As proteções são definidas a partir do valor e da relevância do ativo de informação para a organização. Elas podem ser desenvolvidas para processos (política e normas), para pessoas (portas, alarmes e treinamento) ou para tecnologia (permissão de acesso, firewalls).

As proteções sempre são implantadas sobre três aspectos: lógica (influenciado pela tecnologia); física (influenciado pelas pessoas) e administrativa (influenciado pelos processos).

Os mecanismos de proteção implantados de forma isolada são pouco eficazes, e devemos atentar aos aspectos físicos e lógicos que os envolvem no ambiente de rede.

As ameaças são caracterizadas por algum evento potencialmente prejudicial aos ativos de informação. Elas podem explorar vulnerabilidades e assim se concretizarem, e podem ser classificadas como: natural, acidental ou intencional.

Os agentes das ameaças que exploram as vulnerabilidades, como crackers, atacam as organizações de fora para dentro. Há ataques internos à organização, como os feitos por funcionários insatisfeitos, por exemplo, que são difíceis de prever e conter.

Para compreender os cenários de ameaça, é necessário entender três conceitos fundamentais. São eles:

- **Vulnerabilidades**: são brechas que podem representar portas de entrada para a concretização de um incidente ou ataque à segurança da informação, possivelmente causando impactos ao negócio.

- **Risco**: é matemático, ou seja, é a probabilidade de uma ameaça explorar alguma vulnerabilidade, também causando impactos à segurança da informação. O risco pode ser positivo ou negativo, mas, quando o assunto é segurança da informação, enfatizamos o risco negativo.

- **Impacto**: geralmente é retratado pelo dano causado pela concretização do risco. Quando é representado por prejuízos financeiros, fica fácil mensurar o impacto. Todavia, em danos causados à imagem ou aos controles regulatórios, não é tarefa simples: o impacto pode afetar tanto o negócio, como os acionistas, os fornecedores e terceiros.

8.2.2 Mecanismos e estratégias de segurança em redes de computadores

A proteção de redes e dispositivos de telecomunicações é, sem sombra de dúvida, uma das áreas mais importantes relacionadas à segurança da informação. Em razão dos altíssimos níveis de automatização existentes hoje na grande maioria das organizações e do papel que a internet assumiu como principal meio de comunicação empresarial, desbancando em alguns anos até as redes de telefonia convencional, grande parte das informações que precisamos proteger se encontra armazenada em computadores ou trafegando por diversos tipos de tecnologia de rede e comunicação remota.

Proteger essas informações requer conhecimento especializado e abordagens estruturadas de avaliação das reais condições de segurança existentes. Além disso, os profissionais que se especializam em segurança de redes costumam ter, antes de tudo, um amplo conhecimento sobre o seu funcionamento e sobre as tecnologias que as suportam. Já para a maioria dos gestores, não há a necessidade de se aprofundar nos detalhes técnicos. Porém, entender a finalidade das diversas tecnologias, seus problemas de segurança e as soluções disponíveis para resolvê-los é fundamental, o que demanda preparo e capacitação.

Adentrando ainda mais as estratégias de segurança, podemos começar pela segurança física. Ela cuida da proteção de todos os ativos valiosos da organização e, por isso, abrange as instalações físicas, internas e externas em todas as localidades da organização, além de também cuidar da proteção dos ativos enquanto são transportados como valores ou fitas de backup.

Quando nos referimos à segurança física, a palavra prevenção vem em primeiro lugar. As medidas preventivas que devem ser tomadas podem ser chamadas de barreiras de segurança. Beal (2008) representa uma barreira de segurança como um obstáculo que é colocado para prevenir um ataque.

Quando se trata de segurança física, destacam-se como exemplos uma cerca elétrica e uma parede; em segurança lógica, um processo de logon para acesso a uma rede. Combinadas, ambas formam o perímetro de segurança.

A ISO/IEC 27001 define perímetro de segurança como quaisquer elementos que estabeleçam uma barreira ao acesso indevido. Um perímetro de segurança seria como uma linha delimitadora que define uma área separada, protegida por um conjunto de barreiras físicas e lógicas.

Alguns exemplos de barreiras que agregadas podem formar um perímetro de segurança são: salas-cofre; roletas de controle de acesso físico e uso de token ou dispositivo biométrico para autenticação de pessoas antes da liberação da passagem; circuitos internos de TV; detectores de fumaça; sirenes, alarmes e acionadores de água para combate a incêndios.

A análise de risco para os aspectos físicos auxilia na identificação das instalações físicas e dos ativos de negócio que estão associados à proteção física. Assim, é possível adotar mecanismos que compensam investimento e benefício – uma vez que controles físicos requerem investimentos razoavelmente altos.

A segurança da informação deve interferir o mínimo possível na rotina de uma organização. Entretanto, quando o assunto é segurança física, essa intervenção é inevitável e necessária para tornar o ambiente seguro. Desse modo, devem ser adotados padrões mínimos para a proteção física da organização, de seus ativos de informação e de seus colaboradores.

Quanto maior for a necessidade de proteção, maior será a intervenção da segurança da informação na rotina da organização, que, consequentemente, causará desconforto aos colaboradores. O grande desafio da segurança da informação, em especial da segurança física, é adotar um mecanismo de proteção que equilibre as dificuldades e as proteções para evitar o excesso de procedimentos de controle.

O uso de planos de conscientização ajuda a dividir as responsabilidades, reduzindo a sensação de desconforto causada pela implantação de mecanismos de proteção física no ambiente da empresa. A padronização dos mecanismos de proteção física nas áreas comuns da organização é uma preocupação que transcende a segurança da informação, porque envolve a própria edificação da estrutura de construção da empresa.

O conceito de prevenção criminal por meio do desenho ambiental vem sendo desenvolvido faz 35 anos, e ainda está em evolução. Após a sua elaboração inicial na década de 1960, Oscar Newman estabeleceu as bases do assunto no livro *Defensible space: crime prevention through urban design* (1972). A teoria pauta-se na possibilidade de reduzir o crime e o medo por meio de certas medidas de planejamento e de projeto de áreas. Para isso, são utilizadas duas abordagens básicas. A primeira frisa que é possível desenhar ambientes que reduzem as oportunidades para que um crime possa ocorrer. A segunda tem foco na redução do medo do crime e no aumento da sensação de segurança pessoal, melhorando a qualidade de vida das pessoas e o seu relacionamento com medidas necessárias de segurança.

Alguns princípios básicos são dispostos pela teoria e devem ser seguidos por pessoas responsáveis pelo projeto de construções e espaços urbanos:

- A iluminação deve ser adequada (diurna e noturna) para melhorar a sensação de segurança e desencorajar a prática do crime, porque inibirá a ação do criminoso e aumentará a capacidade de vigilância.

- A definição do campo de visão visa projetar áreas em que os usuários possam ter um largo campo de visão que lhes permita antecipar as proximidades, dando-lhes a sensação de

segurança e de antecipação. Um bom campo de visão de uma área aumenta naturalmente a quantidade de pessoas vigiando o ambiente.

- Os pontos de esconderijo devem ser evitados, como vielas ou reentrâncias em construções, porque podem servir de refúgio para criminosos.

- Construções que predizem por onde as pessoas devem necessariamente passar, como túneis para pedestres ou passarelas, devem ser evitadas por facilitar a ação de criminosos.

- Ambientes que permitam um bom campo de visão aumentam a vigilância natural e a exposição de alguém que deseja cometer um crime.

- O uso misto de áreas (residenciais e comerciais) promove uma boa distribuição de pessoas em diversos horários, garantindo a vigilância a qualquer hora do dia.

- Espaços que geram atividades e trazem pessoas para ocupá-los devem ser criados, como parques e praças de alimentação em ambientes abertos.

- As áreas devem criar um senso de propriedade nas pessoas que passam por lá. Assim, é preciso haver manutenções constantes, que devem ser programadas – além de manter a limpeza, desencorajam a ação de vândalos.

- As localidades e os caminhos possíveis devem estar sinalizados claramente. Isso transmite a sensação de segurança aos usuários, que podem identificar facilmente pontos de apoio ou rotas de fuga alternativas.

Outro aspecto importante está relacionado à localidade das instalações. A localização geográfica interfere diretamente na segurança da informação e é crucial na hora de identificar as ameaças, vulnerabilidades e os riscos para dado ambiente. Por meio da localização geográfica, é possível analisar a probabilidade de ocorrerem problemas de ordem natural ou climática.

Algumas perguntas devem ser respondidas quando analisamos a localização física:

- O local pode ser alvo de ataques terroristas?

- Manifestações públicas são constantes ou esperadas?

- Existe histórico de problemas recorrentes no fornecimento dos serviços básicos?

- A área apresenta riscos inerentes, como proximidade com aeroportos, bases militares ou zonas com alto índice de incidência de crimes?

Os projetos de construção devem pensar nos aspectos de segurança relacionados à entrada de veículos, colaboradores, visitantes, prestadores de serviço e entregadores, bem como logística,

fornecimento de serviços básicos, sistema de ar-condicionado e ventilação – tudo para garantir que a operação de organização não seja interrompida. Os projetos de construção devem pensar na implantação de mecanismos de proteção em sua concepção; a primeira forma de proteção é a chamada perimetral ou periférica, e as barreiras que compõem a segurança periférica visam ser a primeira proteção física de uma organização.

8.2.3 Criptografia e infraestrutura de chaves

A critografia é uma ciência fundamental para a segurança, servindo de base para diversas tecnologias e protocolos. Suas propriedades de confidencialidade, autenticidade, integridade, autenticação e não repúdio garantem o armazenamento, a comunicação e as transmissões de dados de forma segura.

Para saber como chegamos aos conceitos aplicados hoje em dia, é preciso voltar ao passado e entender onde tudo começou. Aproximadamente 1900 a.C., escribas hebreus utilizaram um sistema de substituição do alfabeto de forma reversa. Esse método foi denominado Atbash.

Chamamos essa técnica de cifras de substituição, em que os caracteres da mensagem original são substituídos pelos da mensagem cifrada. Tempos mais tarde, por volta de 100 e 44 a.C., Júlio César utilizou um sistema chamado de cifra de César, que consiste em deslocar as letras do alfabeto em algumas posições. Por exemplo, utilizado a chave 3, o alfabeto cifrado seria:

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

Usando essa técnica para escrever "BOM DIA", o resultado seria "ERP GLD". Até os dias atuais, muitas coisas aconteceram e os métodos de criptografia evoluíram cada vez mais – a exemplo de novos algoritmos, como DES, 3 DES, Twofish, Blowfish, entre outros.

Utilizando os mecanismos disponíveis para o processo de criptografia, descrevemos a seguir um resumo dos serviços de segurança que a criptografia pode nos oferecer:

- **Confidencialidade**: protege o sigilo das informações contra o acesso de terceiros não autorizado.
- **Autenticação**: verifica a identidade de um indivíduo ou de um sistema.
- **Autenticidade**: serve para assegurar que a mensagem foi gerada por quem realmente alega ser.
- **Integridade**: garante que as informações não foram alteradas desde a sua geração.
- **Não repúdio**: impede que uma pessoa ou um sistema negue sua responsabilidade sobre seus atos.

Para usar cada um desses serviços, diversas técnicas são empregadas. A respeito do canal de transmissão dessas informações (chave e mensagens), e considerando que toda a segurança desse sistema depende do sigilo da chave, ela não pode ser transmitida no mesmo canal da mensagem.

Para isso, são utilizados canais seguros para a troca das chaves e, devido ao alto custo dos canais seguros, canais não tão seguros para a transmissão das mensagens.

Há dois tipos principais de criptografia: a simétrica e a assimétrica. A criptografia simétrica possui dois elementos fundamentais: um algoritmo e uma chave, que deve ser compartilhada entre os participantes na comunicação; a mesma chave é utilizada tanto para codificar como para decodificar as mensagens.

A figura a seguir mostra um exemplo simples de como funciona a criptografia simétrica.

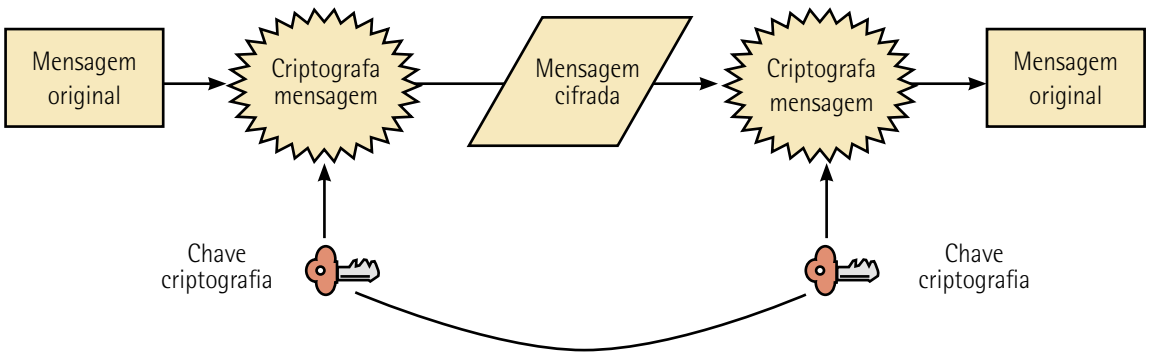


Figura 128 – Criptografia simétrica

Com base nesses conceitos, dá para perceber que esse mecanismo tem algumas desvantagens aparentes, como escalabilidade na troca das chaves e garantia de não repúdio e falta de mecanismos seguros de autenticação. Como vantagens, acentua-se a baixa demanda de processamento e memória.

As funções de hash têm por objetivo macro garantir que a mensagem não seja alterada durante o caminho. Para isso, a mensagem a ser criptografada recebe uma chave simétrica que é utilizada para gerar o MAC (*Message Autentication Code*). Esse processo é bastante complexo, pois são empregadas funções fáceis de serem calculadas em uma direção, mas extremamente difíceis na direção contrária.

Dessa forma, a chave empregada para gerar o MAC não pode ser a mesma que é usada para criptografia da mensagem; ou seja, é preciso um canal seguro para fazer a troca das chaves. Entretanto, para as chaves simétricas, é necessário usar duas chaves – uma para criptografar as mensagens e outra para gerar o MAC. Assim, quanto mais usuários utilizarem esse mecanismo, mais chaves serão necessárias, o que torna o processo de gestão extremamente complexo. Para resolver esse problema, foi criado o conceito de chaves assimétricas.

A criptografia assimétrica surgiu na década de 1970 junto com o conceito de chaves assimétricas. Nele, usam-se duas chaves matematicamente relacionadas, sendo uma pública e outra privada. O objetivo principal de seu desenvolvimento foi resolver os problemas de troca segura de chaves e escalabilidade encontrados nas cifras simétricas.

De forma básica, quando uma mensagem é codificada com uma chave pública, ela somente pode ser interpretada utilizando a chave privada e vice-versa. Tecnicamente, esse mecanismo parte do

princípio de que é fácil fazer a multiplicação de dois números primos, mas sua fatoração (processo inverso) para descobrir quais foram os números iniciais é um problema ainda muito difícil de ser resolvido. Um exemplo de aplicação dessa metodologia é o algoritmo RSA, proposto por Rivest, Shamir e Adleman em 1977.

A figura a seguir apresenta um exemplo de criptografia assimétrica.

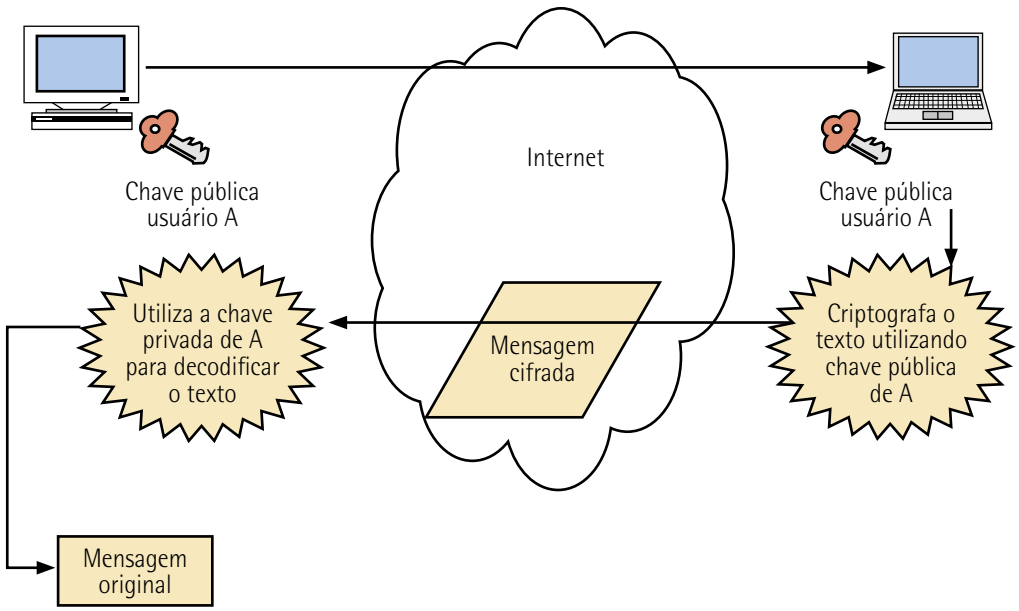


Figura 129 – Criptografia assimétrica

Até hoje, a grande complexidade desse sistema e o tamanho das chaves utilizadas são fatores que garantem o sucesso desse algoritmo; entretanto, devido à quantidade de cálculos processados, o modelo exige uma grande capacidade de processamento.

Para resolver problemas de verificação de autenticidade de uma mensagem, os sistemas de chaves públicas utilizam as chamadas assinaturas digitais, que funcionam da seguinte forma:

- O remetente gera um hash da mensagem a ser envidada.
- O remetente codifica com sua chave privada, gerando uma assinatura digital.
- O remetente adiciona a mensagem, que é cifrada com a chave pública do destinatário.

Dessa forma, se o hash da assinatura for igual ao gerado a partir da mensagem recebida, a assinatura será confirmada e a mensagem será autêntica e íntegra.

Com base nesses conceitos, nota-se que tanto os algoritmos simétricos como os assimétricos têm vantagens e desvantagens, que estão descritas no quadro a seguir.

Quadro 17 – Vantagens e desvantagens das criptografias simétrica e assimétrica

Sistema	Criptografia assimétrica	Criptografia simétrica
Vantagens	Chaves podem ser negociadas por meio de um canal inseguro	Velocidade
	Maior escalabilidade	Elevado nível de segurança
	Utilizada em outros serviços, como assinatura digital	
Desvantagens	Lentidão	A troca das chaves deve ocorrer em um canal seguro
	Necessita de uma chave maior para obter segurança	Baixa escalabilidade
		Não proporciona outros serviços

Para melhor aproveitar essas tecnologias, alguns produtos utilizam uma abordagem mista, conhecida como criptografia híbrida, aproveitando a vantagem de cada sistema.

Nesse modelo, a mensagem seria codificada através da criptografia simétrica com uma chave gerada de forma pseudorrandômica, e sua transmissão ocorreria através de algoritmos assimétricos. Um exemplo disso é o protocolo SSL, em que uma das partes (cliente) gera uma chave simétrica e a codifica com uma chave pública do servidor; quando essa chave é recepcionada, o servidor usa um algoritmo simétrico para criptografar as mensagens em si. Nesse exemplo, são utilizados o desempenho da criptografia simétrica e a segurança na troca das chaves através da criptografia assimétrica.

8.2.4 Certificados digitais

Os certificados digitais, os sistemas de chaves públicas e seus conceitos resolveram uma série de problemas, e o mesmo ocorreu com o uso de sistemas híbridos; entretanto, destaca-se a seguinte pergunta: como é possível garantir a propriedade de uma chave pública em todos esses processos? Para isso, é necessário que exista uma entidade terceira responsável por verificar a identidade do proprietário de uma chave pública, assinando digitalmente sua comprovação. Por essa razão é que foram criadas as ACs (Autoridades Certificadoras), cujo objetivo é atestar a propriedade de sua chave pública – assim, na troca de informações, cada uma das partes solicita seus respectivos certificados digitais para as ACs.

Todavia, esse modelo não seria viável devido às diferenças de localização e à quantidade de solicitações para uma única AC. Essa demanda foi resolvida com o relacionamento entre as ACs, que pode ser dividido em três formatos:

- **Hierárquico**: uma AC raiz tem a função de assinar o certificado de outras ACs, sendo que essas ACs podem ter outras ACs subordinadas e assim por diante.

- **Certificação cruzada**: a AC raiz de uma cadeia assina o certificado de outra AC raiz de uma nova cadeia.

- **Híbrido**: são utilizados os dois conceitos. Por exemplo, se a AC raiz "x" assina o certificado da AC raiz "y", consequentemente todas as ACs abaixo dessas cadeias confiam umas nas outras.

Embora as ACs tenham diversos problemas resolvidos quanto à comprovação da identidade do responsável por uma chave pública, algumas questões de gestão dos certificados ainda precisam ser tratadas.

Assim, adota-se a estrutura chamada de ICP (Infraestrutura de Chaves Públicas), que deve combinar a utilização de software, hardware, protocolos, padrões e processos para fornecer seus serviços. Uma ICP é um conjunto de tecnologias e processos desenhados para prover diversos serviços de segurança.

Uma ICP tem, entre seus componentes básicos: usuários, aplicações, ACs, certificados digitais, além de ARs (Autoridades Registradoras) e diretórios/repositórios de dados. As ARs têm por objetivo interagir com o usuário e repassar as solicitações de, por exemplo, emissão ou renovação de certificados digitais, para o processamento das ACs, garantindo a proteção das ACs contra ações externas.

Os diretórios/repositórios de dados, por sua vez, fornecem um local de fácil acesso para os terceiros acessarem os certificados emitidos pelas ACs. A respeito da infraestrutura de chaves públicas, para além de tecnologia, políticas e processos estabelecidos para garantir segurança, destaca-se como último elemento nesse processo a legislação, cujo objetivo é validar legalmente os mecanismos criptográficos utilizados. Confidencialidade, autenticidade e o não repúdio são os principais serviços oferecidos pela criptografia; assim, a assinatura digital e os certificados digitais possibilitam a criação de leis que reconheçam esses registros. Diante disso, em agosto de 2001, foi editada a Medida Provisória n. 2.200-2 (BRASIL, 2001), que criou a ICP-Brasil, dando validade a documentos assinados digitalmente. A MP tem por objetivo:

> [...] garantir a autenticidade, a integridade e a validade jurídica de documentos em forma eletrônica, das aplicações de suporte e das aplicações habilitadas que utilizem certificados digitais, bem como a realização de transações eletrônicas seguras (BRASIL, 2001, art. 1º).

Essa MP também instituiu a formação de um Comitê Gestor da ICP-Brasil, cuja missão é "adotar as medidas necessárias e coordenar a implantação e o funcionamento da ICP-Brasil [...]". Também foi instituída a AC raiz como primeira entidade da cadeia de certificação da ICP-Brasil, além das ACs e ARs, em conformidade com as normas do Comitê Gestor (BRASIL, 2001, art. 4º).

A segurança contra ataques criptográficos e a avaliação da segurança de algoritmos representam a facilidade ou não com que uma pessoa consegue decifrar as mensagens. A forma mais simples de ataque a algoritmos é o ataque de força bruta, mas também é o menos eficiente e, às vezes, impossível de ser implementado devido ao tamanho das chaves. Existem outras técnicas que podem ser utilizadas pelos criptoanalistas, que são:

- **COA** (*Ciphertext Only Attack*): o foco é o comprometimento da confidencialidade das informações. O criptoanalista tem acesso apenas a uma ou mais mensagens codificadas e seu objetivo é descobrir as mensagens originais.

- **KPA** *(Know Plaintext Attack): o* criptoanalista tem acesso ao texto cifrado e ao texto plano que o originou, e, ao usar, essas informações, tenta obter a chave em uso.

- **CPA** (*Choose Plaintext Attack***)**: o adversário é capaz de escolher o texto plano que será cifrado.

- **ACPA** (*Adaptative Choosen Plaintext Attack*): são enviados diversos pequenos blocos de dados, que são adaptados conforme o criptoanalista coleta informações. Seu objetivo é descobrir a chave em uso.

Todos os sistemas criptográficos têm níveis diferentes de segurança, dependendo da facilidade ou dificuldade com que eles são quebrados.

A segurança de um criptossistema não deve ser baseada nos algoritmos que cifram as mensagens, mas sim no tamanho das chaves usadas. Pode-se dizer que um algoritmo é considerado forte quando é praticamente impossível quebrá-lo no determinado espaço de tempo em que as informações ainda sejam relevantes, isto é, ainda podem ser utilizadas por pessoas não autorizadas.

Geralmente, a maneira mais fácil de afirmar se um algoritmo é considerado forte é publicando sua descrição, fazendo com que várias pessoas possam testar e avaliar sua eficiência. Programas que usam algoritmos proprietários não divulgam sua especificação, pois a simples divulgação do método revelará também seus pontos fracos.

8.2.5 Dispositivos de segurança para as redes

Os elementos básicos para proteção de rede incluem dispositivos como roteadores de borda, firewalls, NAT (*Network Address Translation*), VPN (*Virtual Private Networks* – Redes Virtuais Privadas), bastion host, perímetro lógico, IDS (*Intrusion Detection System*), IPS (*Intrusion Prevention System*) e políticas de segurança.

O roteador de borda é o último gateway, responsável por interconectar redes diferentes, que conecta a rede interna da empresa à internet. Ele é a primeira linha de defesa da empresa contra ameaças externas, elemento fundamental para a composição de diversos sistemas firewall. Contudo, é muito comum também que não seja utilizado para funções de segurança por dois simples motivos: o primeiro é porque, muitas vezes, o roteador não pertence à organização, e sim a empresas que provêm a conexão com a internet; o segundo motivo diz respeito à dificuldade de configuração, normalmente feita via linha de comando, o que faz com que muitos administradores, por questão de praticidade, abram mão de utilizar esse recurso essencial.

O roteador de borda permite a criação de controle de acesso. Como roteador, desempenha papel de filtro, gerando as ACLs (*Access Control Lists* – Listas de Controle de Acesso), que muitas vezes classificam

a designação do roteador – dessa forma, toma a atuação de roteador de perímetro como a primeira camada de firewall.

As listas de acesso de um roteador são proporcionalmente mais simples e normalmente são classificadas como filtros de pacotes simples. Por serem equipamentos que trabalham em camada 3, os roteadores provocam pouca ou nenhuma interferência quanto aos protocolos de camada 7 de aplicação.

Geralmente, sua lista de acesso é configurada para permitir ou negar determinado tipo de tráfego com base nas informações de endereço de origem, destino, portas de passagem e de destino.

O firewall é um conjunto de hardware e software que permite a criação de regras definindo que tipos de serviço e tráfegos são permitidos entre as redes que ele conecta. São dispositivos de controle de acesso cuja função principal é proteger as estações e a segmentação de perímetros, impedindo que estranhos acessem a rede. Esse dispositivo normalmente é um computador independente (standalone), um roteador ou um firewall em uma caixa (dispositivo de hardware proprietário). A unidade serve como o único ponto de entrada para seu site e avalia cada solicitação de conexão quando é recebida. Somente solicitações de conexão de equipamentos autorizados são processadas; as demais solicitações de conexão são descartadas. A maioria dos firewalls realiza isso verificando o endereço de origem, e em geral é fixado na junção de duas redes com níveis de confiança distintos.

Os firewalls podem analisar pacotes recebidos de vários protocolos. Com base nessa análise, um firewall pode empreender várias ações. Portanto, eles são capazes de realizar avaliações condicionais. Por exemplo: "se esse tipo de pacote for encontrado, farei isso".

Essas construções condicionais são chamadas regras. Geralmente, quando é configurado, o firewall é equipado com as regras que espelham as diretivas de acesso em sua própria organização.

Entretanto, essa verificação de acesso é apenas uma parte do que os firewalls modernos podem fazer. A maioria dos firewalls comerciais permite verificar o conteúdo – pode-se explorar essa capacidade para bloquear Java, JavaScript, VBScript e scripts ActiveX e cookies no firewall. De fato, é possível criar regras para bloquear determinadas assinaturas de ataque.

Quanto à sua construção, os componentes de um firewall estão baseados na mente das pessoas que o desenvolvem. Em essência, é um conceito, e não um produto; é uma ideia de quem terá permissão para acessar seu site.

O software de um firewall pode ser proprietário (shareware), e o hardware pode ser qualquer hardware que suporta o software. Os firewalls dividem-se em duas categorias básicas: firewalls de nível de rede e firewalls de gateway de aplicativo.

Os firewalls de nível de rede são geralmente roteadores com capacidades poderosas de filtragem de pacote. Utilizando um firewall de nível de rede, é possível conceder ou negar acesso a um site com base em diversas variáveis, incluindo: endereço de origem, protocolo, número de porta e conteúdo.

Os firewalls baseados em roteadores são populares porque são facilmente implementados; para conectar um, basta fornecer algumas regras e está pronto. A maioria dos roteadores novos faz um trabalho muito bom de tratamento de interfaces dúbias, em que IPs de fora devem ser traduzidos por algum outro protocolo interno. Adicionalmente, um firewall baseado em roteador é uma solução de perímetro – isto é, os roteadores são dispositivos externos, então eles eliminam a necessidade de interromper a operação normal da rede.

Quando se utiliza um firewall baseado em roteador, não se pode configurar várias máquinas ou serviços para interagir com ele. Os roteadores podem também oferecer uma solução integrada: se sua rede está permanentemente conectada à internet, será necessário um roteador, então é possível unir duas utilidades em uma. Porém, firewalls baseados em roteador têm várias deficiências: uma delas é que muitos são vulneráveis a ataques de personificação, ou spoffing, embora os fornecedores de roteador estejam desenvolvendo soluções para evitar esse ataque. Outra questão é puramente prática: o desempenho do roteador cai dramaticamente quando você impõe procedimentos de filtragem excessivamente rigorosos.

Vejamos como operam os firewalls de aplicativo proxy: quando um usuário remoto entra em contato com uma rede executando um gateway de aplicativo, o gateway gerencia proxies para a conexão. Nesse caso, pacotes de IP não são encaminhados à rede interna. Em vez disso, um tipo de tradução ocorre, com o gateway agindo como canal e intérprete.

A vantagem de gateways de aplicativo é que eles impedem o tunelamento de pacotes IP em sua rede. A desvantagem é que eles exigem overheads altos e envolvem grande parte da rede. Um aplicativo proxy deve ser configurado para cada serviço na rede, incluindo FTP, Telnet, HTTP, correio, notícia etc.

Além disso, usuários internos devem utilizar clientes cientes de proxy. Se eles utilizarem, você terá de adotar novas diretivas e procedimentos. A desvantagem de gateways de aplicativo é que, no caso de protocolos cliente-servidor como Telnet, são necessários dois passos para enviar ou receber uma conexão. Alguns gateways de aplicativo exigem clientes modificados, que podem ser vistos como uma desvantagem ou uma vantagem dependendo de os clientes modificados tornarem ou não mais fácil utilizar o firewall.

Um gateway de aplicativo de Telnet necessariamente não exigiria um cliente de Telnet modificado, mas exigiria uma alteração no comportamento dos usuários: eles teriam de se conectar, mas não efetuar logon com o firewall em oposição a se conectar diretamente com o host. Contudo, um cliente modificado de Telnet pode tornar o firewall transparente, permitindo que um usuário especifique o sistema alvo (em oposição ao firewall) no comando Telnet. O firewall serviria como a rota para o sistema de destino e, portanto, interceptaria a conexão e realizaria os passos adicionais conforme necessário – por exemplo, consultar uma senha de uma única vez. O comportamento do usuário permaneceria o mesmo, mas exigiria um cliente modificado em cada sistema.

Resumo

Nesta unidade avançamos ainda mais nas camadas da arquitetura de redes de computadores. Começamos pela camada de rede, chamada de camada 3, em que mencionamos o seu funcionamento e suas funcionalidade, destacando o papel dos protocolos roteados e de roteamento para prover a determinação do melhor caminho para o pacote.

Depois da camada de rede, que abrangeu os dois primeiros tópicos, adentramos os conceitos de camada de transporte. Apresentamos os dois tipos de transporte (orientado e não orientado a conexões), enfatizando os protocolos TCP e UDP como ferramentas fundamentais para conseguir confiabilidade ou velocidade no processo de comunicação de dados.

No último tópico chegamos até a camada de aplicação, apresentando os protocolos desta camada, com ênfase para o HTTP, DNS, FTP, TFTP, SMNP, SMTP, POP e IMAP.

Finalizamos esta unidade apresentando os conceitos e bases da segurança em redes de computadores.

Exercícios

Questão 1. (PR-4 UFRJ/2021, adaptada) No nível da camada de rede do modelo híbrido, a criação de protocolos de roteamento permitiu a construção e a atualização de tabelas de roteamento entre gateways. Com o crescimento da rede e das tabelas de roteamento, foi necessária a implantação de protocolos de roteamento hierárquicos. Assim, os roteadores foram divididos em regiões chamadas de *Autonomous System* (AS), em que cada roteador conhecia todos os detalhes de sua própria região e não conhecia a estrutura interna de outras regiões.

Nesse sentido, avalie as afirmativas a seguir sobre conceitos e protocolos de roteamento.

I – O protocolo RIP (*Routing Information Protocol*) utiliza o algoritmo vetor-distância e é responsável pela construção de uma tabela que informa as rotas possíveis dentro do AS.

II – O OSPF é um protocolo que usa a busca pelo melhor caminho exatamente como o RIP, utilizando o algoritmo do vetor-distância, mas, para redes maiores, seu tempo de convergência é muito mais elevado que o tempo do RIP.

III – Tanto o RIP quanto o OSPF utilizam protocolos de roteamento interno a um AS.

É correto o que se afirma em:

A) I, apenas.

B) III, apenas.

C) I e III, apenas.

D) II e III, apenas.

E) I, II e III.

Resposta correta: alternativa C.

Análise das afirmativas

I – Afirmativa correta.

Justificativa: o RIP é um dos protocolos de roteamento mais conhecidos na área de redes de computadores, sendo um dos principais protocolos de roteamento por vetor de distância. Ele utiliza como métrica de menor custo a contagem de saltos, que representa a quantidade de redes pelas quais

um pacote transitará até chegar ao seu destino. As tabelas de roteamento em roteadores que operam com o RIP apresentam o número de saltos a serem dados até que se chegue à rede de destino.

II – Afirmativa incorreta.

Justificativa: o protocolo OSPF não busca o melhor caminho exatamente como o RIP. O OSPF é muito mais eficiente do que o RIP, porque trabalha com um custo que combina banda passante e confiabilidade, entre outras métricas.

III – Afirmativa correta.

Justificativa: tanto o RIP quanto o OSPF são classificados como protocolos de gateway interno (IGPs), caracterizados pela atuação do algoritmo de roteamento dentro de um único AS.

Questão 2. No nível da camada de transporte do modelo híbrido, a depender do protocolo utilizado, trabalhamos com o serviço orientado à conexão ou com o serviço não orientado à conexão.

Considerando esse contexto, avalie as afirmativas a seguir e a relação proposta entre elas.

I – O TCP (*Transport Control Protocol*) é o protocolo requisitado por aplicações que necessitam de confiabilidade no transporte de segmentos.

porque

II – O TCP é um protocolo que não é orientado à conexão, o que garante uma alta velocidade de transporte.

A respeito dessas afirmativas, assinale a opção correta.

A) As afirmativas I e II são proposições verdadeiras, e a afirmativa II é uma justificativa correta da I.

B) As afirmativas I e II são proposições verdadeiras, e a afirmativa II não é uma justificativa correta da I.

C) A afirmativa I é uma proposição verdadeira, e a II é uma proposição falsa.

D) A afirmativa I é uma proposição falsa, e a II é uma proposição verdadeira.

E) As afirmativas I e II são proposições falsas.

Resposta correta: alternativa C.

Análise das afirmativas

I – Afirmativa verdadeira.

Justificativa: na camada de transporte, o TCP é o protocolo requisitado por aplicações que necessitam de confiabilidade no transporte de segmentos. Isso se dá pelo estabelecimento de sessão entre o destino e a origem antes de transmitir dados, feito por meio do *handshake* triplo, de forma a garantir que a troca de dados entre o destino e a origem seja efetuada com sucesso.

II – Afirmativa falsa.

Justificativa: o TCP é um protocolo voltado à conexão, que garante confiabilidade, em detrimento da velocidade. O protocolo UDP (*User Datagram Protocol*), quando comparado ao TCP, apresenta alta velocidade, mas confiabilidade reduzida, pois não é orientado à conexão, de modo que não há a utilização do *handshake*.

REFERÊNCIAS

Textuais

ABNT. ISO 31000 – Gestão de riscos – Diretrizes. Rio de Janeiro: ABNT, 2009a.

ABNT. ISO Guia 73 – Gestão de risco – Vocabulário. Rio de Janeiro: ABNT, 2009b.

ABNT. NBR ISO/IEC 27001 – Tecnologia da informação – Técnicas de segurança – Sistemas de gestão de segurança da informação – Requisitos. Rio de Janeiro: ABNT, 2006.

ABNT. NBR ISO/IEC 27002 – Tecnologia da informação – Técnicas de segurança – Código de prática para a gestão da segurança da informação. Rio de Janeiro: ABNT, 2005.

AMAZONAS, J. R. A. *Projetos de sistemas de comunicações ópticas*. Barueri: Manole, 2005.

BEAL, A. *Gestão estratégica da informação*: como transformar a informação e a tecnologia da informação em fatores de crescimento e de alto desempenho nas organizações. São Paulo: Atlas, 2004.

BEAL, A. *Segurança da informação*: princípios e melhores práticas para proteção de ativos de informação nas organizações. São Paulo: Atlas, 2008.

BERNAL, P. S. M. *Voz sobre protocolo IP*: a nova realidade da telefonia. São Paulo: Érica, 2007.

BRASIL. *Medida Provisória n. 2.200-2, de 24 de agosto de 2001*. Brasília, 2001. Disponível em: https:// cutt.ly/h2T91hz. Acesso em: 9 jan. 2023.

CAMPOS, A. *Sistema de segurança da informação*: controlando os riscos. 2. ed. Florianópolis: Visual Books, 2007.

CARVALHO, L. P. *Introdução a sistemas de telecomunicações*: abordagem histórica. Rio de Janeiro: LTC, 2014.

CICCARELLI, P. *et al. Princípios de redes*. Tradução: Claudio Coutinho de Biasi. Rio de Janeiro: LTC, 2009.

COMER, D. E. *Internetworking with TCP/IP*. 4. ed. New Jersey: Prentice Hall, 2000. v. 1.

DIMARZIO, J. F. *Projeto e arquitetura de redes*: um guia de campo para profissionais de TI. Rio de Janeiro: Elsevier, 2001.

FERREIRA, F. N. F.; ARAUJO, M. T. *Política de segurança da informação*: guia prático para elaboração e implementação. Rio de Janeiro: Ciência Moderna Ltda., 2006.

FILIPPETTI, M. A. *CCNA 6.0*: guia completo de estudo. Florianópolis: Visual Books, 2017.

FITZGERALD, J.; DENNIS, A. *Comunicações de dados empresariais e redes*. Rio de Janeiro: LTC, 2010.

FOROUZAN, B. A. *Comunicação de dados e redes de computadores*. 4. ed. Porto Alegre: AMGH, 2010.

FOROUZAN, B. A.; MOSHARRAF, F. *Redes de computadores*: uma abordagem topdown. São Paulo: McGraw Hill, 2013.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a internet*. Porto Alegre: Bookman, 2021.

LEE, V. *Aplicações móveis*: arquitetura, projeto e desenvolvimento. São Paulo: Pearson, 2005.

LIMA FILHO, E. C. *Fundamentos de redes e cabeamento estruturado*. São Paulo: Pearson, 2014.

MAIA, L. P. *Arquitetura de redes de computadores*. 2. ed. Rio de Janeiro: LTC, 2013.

MARÇULA, M.; BENINI FILHO, P. A. *Informática*: conceitos e aplicações. São Paulo: Érica, 2013.

MARIN, P. S. *Cabeamento estruturado*: desvendando cada passo – do projeto à instalação. 4. ed. São Paulo: Érica, 2013.

MEDEIROS, J. C. O. *Princípios de telecomunicações*: teoria e prática. São Paulo: Érica, 2016.

MIYOSHI, E. M.; SANCHES, C. A. *Projetos de sistemas de rádio*. São Paulo: Érica, 2008.

MORAES, A. F. *Redes de computadores*: fundamentos. 8. ed. São Paulo: Saraiva, 2020.

MORAES, A. F. *Segurança em redes*: fundamentos. São Paulo: Érica, 2010.

NEWMAN, O. *Defensible spaces:* crime prevention through urban design. New York: Macmillan, 1972.

OLIVEIRA, A. V.; MELO, J. L. *Certificação CCNA*: guia preparatório para o exame 200-301. Rio de Janeiro: SF Editorial, 2021.

PINHEIRO, J. M. S. *Guia completo de cabeamento de redes*. 2. ed. Rio de Janeiro: Elsevier, 2015.

RIBEIRO, M. P. *Redes de telecomunicações e teleinformática*: um exercício conceitual com ênfase e modelagem. Rio de Janeiro: Interciência, 2012.

SANCHES, C. A. *Projetando redes WLAN*: conceitos e práticas. São Paulo: Érica, 2007.

SCRIMGER, R. *et al. TCP/IP*: a bíblia. Rio de Janeiro: Campus, 2002.

SHIMONSKI, R. J.; STEINER R.; SHEEDY, S. *Cabeamento de rede*. Rio de Janeiro: LTC, 2014.

SOARES NETO, V. *Redes de telecomunicações*: sistemas avançados. São Paulo: Érica, 2015.

SOARES NETO, V. *Telecomunicações avançadas e as tecnologias aplicadas*. São Paulo: Érica, 2018.

SOUZA, L. B. *Redes de computadores*: guia total. São Paulo: Érica, 2011.

STAIR, R. M.; REYNOLDS, G. W. *Princípios de sistemas de informação*. Tradução: Harue Avritscher. São Paulo: Cengage Learning, 2011.

SUPPI, G. M. *et al.* Uma visão geral sobre a internet das coisas. *Revista Univap*, São José dos Campos, v. 22, n. 40, p. 586-599, 2016.

TANENBAUM, A.; FEAMSTER, N.; WETHERALL, D. *Redes de computadores*. Tradução: Daniel Vieira. 5. ed. Rio de Janeiro: Pearson Prentice Hall, 2021.

TEIXEIRA FILHO, S. A. *Segurança da informação descomplicada*. Santa Catarina: Clube de Autores, 2015.

TORRES, G. *Redes de computadores*. 2. ed. Rio de Janeiro: Novaterra, 2016.

WEIDMAN, G. *Testes de invasão*: uma introdução prática ao hacking. São Paulo: Novatec, 2014.

WHITE, C. *Redes de computadores e comunicação de dados*. São Paulo: Cengage Learning, 2012.

Informações:
www.sepi.unip.br ou 0800 010 9000