

PROGRAMAÇÃO EM AMBIENTES DE REDES DE COMPUTADORES

Raiza Artemam de Oliveira



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS



Princípios de roteamento IP

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Explicar o que é roteamento.
- Diferenciar roteamento estático de dinâmico.
- Enumerar os principais algoritmos para roteamento.

Introdução

Com a evolução da tecnologia, a expansão de dispositivos tecnológicos e conectados a redes aumentou exponencialmente, tornando necessário o desenvolvimento de estratégias para o aprimoramento da internet, para que todos os pacotes não se perdessem no meio do caminho e fossem entregues ao seu destino. Assim, vários estudos sobre redes de computadores (conceito que hoje abrange diversos outros dispositivos) permitiram o aprimoramento da rede mundial que conhecemos como internet.

Hoje, praticamente todas as áreas dependem do acesso à internet para que os processos funcionem de maneira correta e ágil, o que depende, em redes, do conceito de roteamento. Neste capítulo, você estudará do que se trata o roteamento, analisando seus principais algoritmos e diferenciando os tipos estático e dinâmico.

Roteamento

Para que os dados sejam transmitidos na rede, os dispositivos precisam atuar sob modelos arquiteturais de redes, existindo, atualmente, dois: o *transmission control protocol/internet protocol* (TCP/IP) e o *open systems interconnection* (OSI), definido em meados de 1970 pela *International Organization for Standardization* (ISO). No início, a indústria de computadores enfrentava um grande problema em relação à compatibilidade de comunicação entre os dispositivos, pois cada fabricante queria implantar seu próprio padrão. O modelo OSI é um

sistema aberto que contém um conjunto de protocolos que possibilitam que diferentes sistemas se comuniquem independentemente de suas arquiteturas subjacentes, ou seja, seu objetivo consiste em facilitar a comunicação entre sistemas diferentes sem a necessidade de realizar mudanças na lógica do *software* ou do *hardware* dos dispositivos (FOROUZAN, 2008).

Quando falamos do modelo arquitetural OSI, não podemos confundí-lo com protocolos, uma ideia incorreta: trata-se de um modelo de sete camadas, cada uma das quais responsável por tratar a mensagem, o que possibilita a compreensão e a construção do projeto de uma arquitetura de redes flexível, robusta e interoperável. Na Figura 1, é possível observar, de modo geral, as camadas e como esse modelo trabalha — as camadas dos dispositivos A e B operam sob o mesmo protocolo; quando uma mensagem sai da origem, é montada e empacotada em cada camada, e, quando é recebida no destino, passa pelas mesmas camadas, porém de baixo para cima, sendo desmontada e desempacotada.

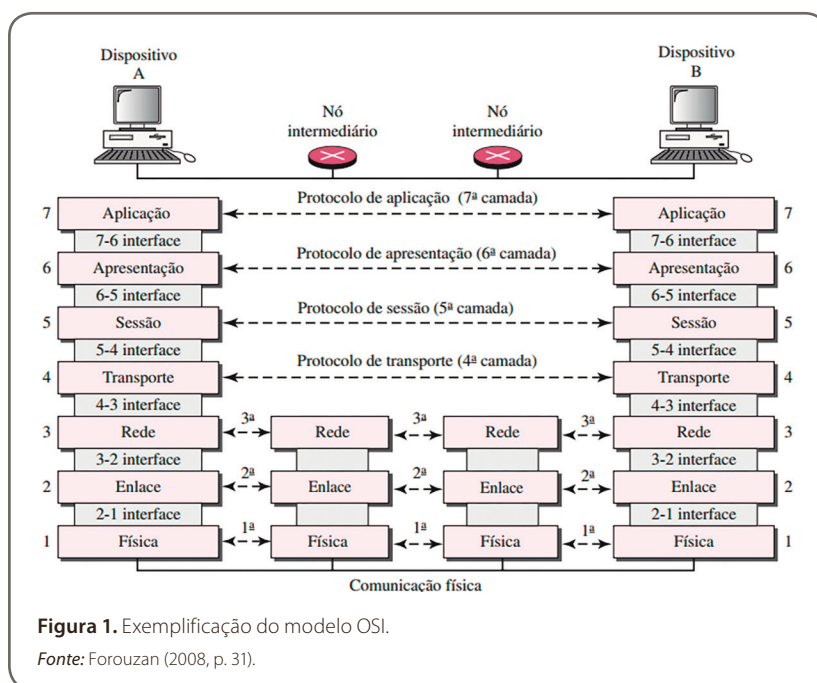
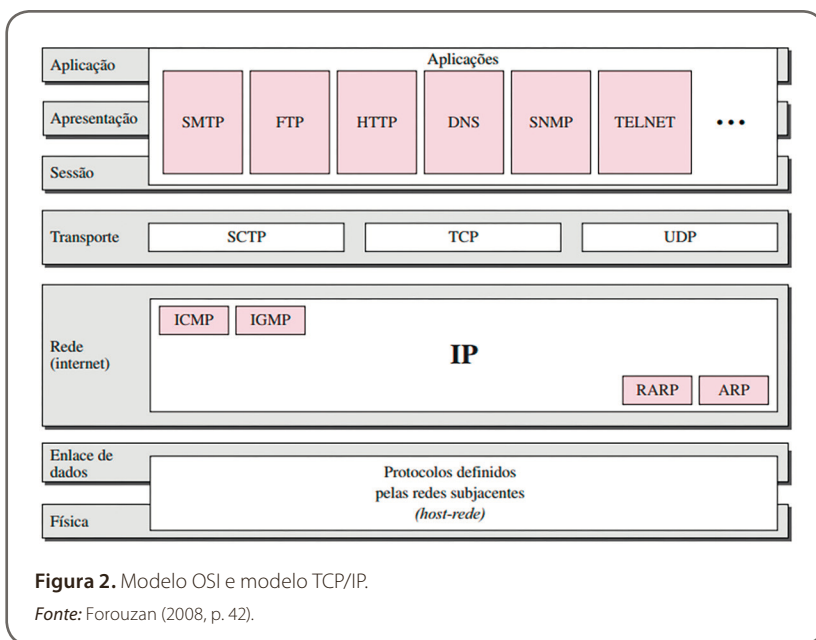


Figura 1. Exemplificação do modelo OSI.

Fonte: Forouzan (2008, p. 31).

O modelo TCP/IP, com somente quatro camadas, foi criado antes do modelo OSI, no início dos estudos sobre construção de redes durante a Segunda Guerra Mundial, sendo atualmente o modelo utilizado na internet. Na Figura 2, você consegue observar a diferença entre as camadas dos dois modelos em que: as três camadas superiores do modelo OSI estão englobadas na camada de aplicação do modelo TCP/IP; as camadas de transporte e rede se mantêm; e as duas últimas camadas do modelo OSI estão englobadas na camada física.



Tendo fixado os conceitos básicos sobre os modelos arquiteturais de rede, podemos agora iniciar o estudo sobre roteamento, que pode, segundo estudos distintos, estar localizado na camada de rede ou na internet. É comum, ainda, a confusão entre os conceitos de roteamento e repasse (ou encaminhamento): segundo Kurose e Ross (2007), repasse é quando um pacote chega à entrada de um roteador e este deve conduzi-lo a uma saída apropriada, e roteamento, por sua vez, se dá quando a camada de rede do modelo arquitetural deve definir a tora ou o caminho tomado pelos pacotes ao fluírem de um remetente a um destinatário. Para isso, utilizam-se algoritmos de roteamento.



Fique atento

Não confunda repasse e roteamento. Repasse está relacionado à transferência de um pacote de um enlace de entrada para o enlace de saída dentro de um único roteador. E roteamento envolve todos os roteadores conectados em uma rede, cenário em que a interação coletiva do protocolo de roteamento determina os caminhos que os pacotes percorrerão do nó de origem para o nó de destino (FOROUZAN, 2008; KUROSE; ROSS, 2007).

Para tornar o conceito mais prático, façamos uma analogia com um viajante. Durante a viagem, o motorista do ônibus passa por muitos cruzamentos de rodovias em sua rota até o destino. Neste exemplo, podemos imaginar o repasse como o processo de passar por um único cruzamento, ou seja, o motorista dirige até o cruzamento e determina qual rodovia pegará para sair do cruzamento. Já o roteamento seria o processo de planejar a viagem da origem até o destino, pois, antes de partir, o motorista do ônibus consulta um mapa e determina as rotas dos diversos caminhos possíveis. Assim, utilizam-se as tabelas de roteamento para auxiliar o roteamento (FOROUZAN, 2008; TANENBAUM, 2003).

Para que todo esse processo seja possível, há no roteador uma tabela de roteamento. Existem várias técnicas para reduzir a quantidade de conteúdo na tabela de roteamento, uma vez que a rede mundial de internet, ou até mesmo redes pequenas, requer muito do processamento dos roteadores, além de tratar de questões de segurança, fator imprescindível para que os dados circulem na rede, como:

- Método do próximo salto: a tabela de roteamento armazena apenas o endereço do próximo salto, ou seja, o próximo *host* em que o pacote será enviado, em vez das informações completas da rota (KUROSE; ROSS, 2007).
- Método da rede específica: em vez de na tabela de roteamento haver uma entrada para cada *host* de destino conectado a uma mesma rede, existe apenas uma entrada responsável por definir o endereço da rede de destino, quando todos os *hosts* conectados a uma rede são tratados como uma única entidade.

- Método do *host* específico: cada *host* tem sua própria identificação na tabela de roteamento, sendo usado em casos nos quais se torna necessário o fornecimento de medidas de segurança específicas (FOROUZAN, 2008; KUROSE; ROSS, 2007).
- Método-padrão: em um caso no qual há um *host* A conectado a uma rede com dois roteadores, em que o R1 direciona os pacotes conectados a D2 (destino), porém para o restante da internet, é usado o roteador R2, em que, em vez de listar todas as redes conectada à rede internet inteira, o *host* A pode ter apenas uma entrada denominada padrão, sendo 0.0.0.0.

Roteamento estático *versus* roteamento dinâmico

Para que os dados circulem na rede, existe o conceito de roteamento, para o qual os roteadores armazenam uma tabela de roteamento, que podem ser do tipo estático ou dinâmico.

Nas tabelas de roteamento do tipo estático, o administrador insere manualmente as informações da rota para cada destino; ainda, quando se cria uma tabela, não pode ser atualizada automaticamente com uma mudança na internet. Do mesmo modo, há a necessidade da alteração manual do administrador. Esse tipo pode ser utilizado em redes pequenas, nas quais não há alterações com frequência.

Em tabelas de roteamento do tipo dinâmico, há uma atualização periódica e automática, por meio do emprego protocolos (algoritmos) de roteamento dinâmico, como *routing information protocol* (RIP), *open shortest path first* (OSPF) e *border gateway protocol* (BGP), ou seja, sempre que houver uma mudança na internet, os algoritmos de roteamento atualizarão automaticamente todas as tabelas nos roteadores. Em grandes redes, como a internet, os roteadores precisam ser atualizados de maneira dinâmica, pois torna-se inviável a configuração manual sempre que houver uma alteração na rede, além da necessidade de que todos os roteadores estejam atualizados para uma entrega eficiente. Na Figura 3, é possível observar o formato de uma tabela de roteamento com os campos comumente utilizados.

Máscara	Endereço de rede	Endereço do próximo salto	Interface	Flags	Contagem de referência	Uso
.....

Figura 3. Campos em uma tabela de roteamento.

Fonte: Forouzan (2008, p. 656).

A seguir, conforme Forouzan (2008), Kurose e Ross (2007) e Tanenbaum (2003), são listados os campos em uma tabela de roteamento.

- **Máscara:** define a máscara de entrada utilizada na rede.
- **Endereço de rede:** determina o endereço do roteador do *host* de destino.
- **Endereço do próximo salto:** define o endereço do roteador do próximo nó ao qual o pacote é entregue.
- **Interface:** estabelece a interface utilizada na rede (p. ex., ethernet).
- **Flags:** contém cinco *flags* que indicam presença ou ausência: U (operando), G (*gateway*), H (*host* específico), D (acrescentado por redirecionamento) e M (modificado por redirecionamento).
- **Contagem de referência:** a quantidade de usuários de determinada rota no momento, ou seja, a quantidade de pessoas conectadas a um mesmo *host* de um roteador simultaneamente.
- **Uso:** a quantidade de pacotes transmitidos por meio de determinado roteador para o destino correspondente.

Nos sistemas operacionais, estão disponíveis vários utilitários que permitem que o usuário veja informações de roteamento e conteúdo de uma tabela de roteamento, como o comando `netstat` (Figura 4), que torna possível conhecer todas as conexões TCP ativas no computador em que o comando foi executado.

```
[root@localhost ~]# netstat -rn
Kernel IP routing table
Destination    Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0        10.5.0.1        0.0.0.0         UG      0  0        0 eth0
10.5.0.0       0.0.0.0         255.255.0.0     U        0  0        0 eth0
[root@localhost ~]#
```

Figura 4. Exemplificação de uma tabela de roteamento real.

No exemplo da Figura 4, utilizamos os parâmetros ‘-rn’, ou seja: ‘r’ para exibir o conjunto de encaminhamento e ‘n’ para mostrar os endereços e os números das portas, sem resolução de nomes. Então, é possível observar: o endereço de destino, o *gateway* (no Linux, é utilizado como sinônimo de roteador), que define basicamente o endereço do nó mais próximo; a máscara da rede; e as *flags* de cada endereço, que funcionam como uma “etiqueta” indicando o estado atual de cada endereço — a *flag* do primeiro endereço contém G, que indica que o destino pode ser alcançado por meio de um roteador —; e a interface, que indica que a *host* está conectada à rede por meio da *ethernet* 0. É importante ressaltar que esta é uma tabela de roteamento do *host*, e não de um roteador. Utilizando o comando `ifconfig eth0`, a interface de que queremos saber as informações, temos o resultado, como podemos observar na Figura 5.

```
[root@localhost ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 02:96:FE:AA:55:95
          inet addr:10.5.75.33  Bcast:10.5.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1872 (1.8 KiB)  TX bytes:804 (804.0 B)
```

Figura 5. Exemplificação do comando `ifconfig`.

Com as informações levantadas, pode-se deduzir a configuração do servidor estudado no exemplo, como observado na Figura 6.

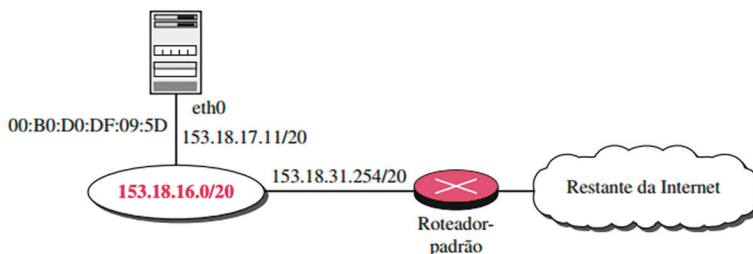


Figura 6. Configuração da rede do exemplo estudado.

Fonte: Forouzan (2008, p. 658).

Há, ainda, o conceito de rota estática flutuante, que nada mais é do que rotas estáticas (implementadas manualmente) para fornecer caminhos alternativos para uma rota, estática ou dinâmica (KUROSE; ROSS, 2007).



Link

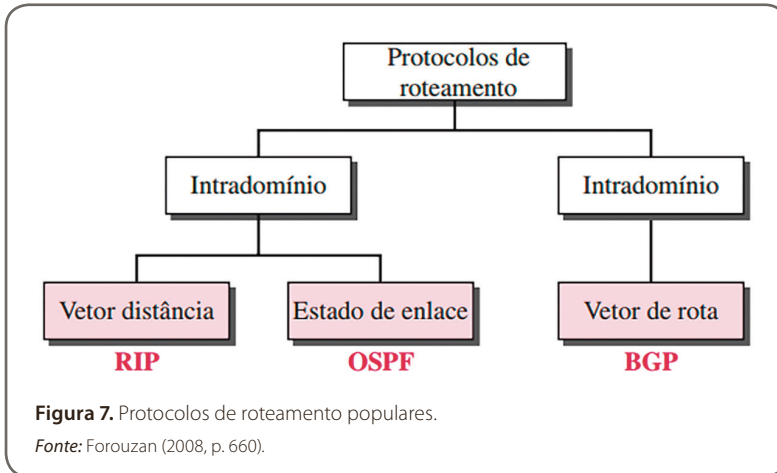
Acesse o link a seguir para testar seus comandos em distribuições de Linux diferentes sem precisar instalá-las em seu computador.

<https://qrigo.page.link/VtbDM>

Algoritmos de roteamento

Assim como tudo na computação, para todo *hardware* há um *software* referente ao funcionamento correto da finalidade do dispositivo. Para que os roteadores consigam efetuar o roteamento dos dados, há neles algoritmos de roteamento, os quais estão disponíveis em roteadores que utilizam o roteamento dinâmico. Nesta seção, estudaremos os algoritmos RIP (protocolo de informações de roteamento), OSPF (primeira rota aberta mais curta) e BGP (protocolo *gateway* de fronteira).

Contudo, antes de falarmos a respeito dos algoritmos de roteamento, precisamos conhecer o conceito de roteamento interdomínio e intradomínio. Atualmente, sabemos que a internet pode ser tão grande que um algoritmo de roteamento sozinho não seria capaz de lidar com a tarefa de atualizar as tabelas de roteamento de todos os roteadores, o que a leva ser dividida em sistemas autônomos (SA), grupos de redes e roteadores sob a regência de uma única administração (FOROUZAN, 2008). Dentro de um sistema autônomo, há um roteamento, denominado roteamento intradomínio, um entre os sistemas autônomos, chamado roteamento interdomínio. É interessante observar que, dentro de um sistema autônomo, pode-se executar algoritmos de roteamento distintos, mas, entre sistemas autônomos, somente um protocolo de roteamento interdomínio é capaz de ser executado. Existem vários outros algoritmos de roteamento, porém, como já dito, estudaremos os mais populares. Na Figura 7, podemos observar os algoritmos e suas classificações entre os sistemas autônomos.



Os algoritmos de roteamento implementam os seguintes protocolos:

- **RIP** — *routing information protocol* (protocolo de informações de roteamento): implementação do protocolo vetor distância;
- **OSPF** — *open shortest path first* (primeira rota aberta mais curta): implementação do protocolo estado de enlace;
- **BGP** — *border gateway protocol* (protocolo gateway de fronteira): implementação do protocolo vetor rota.

O algoritmo de vetor distância armazena em cada nó uma tabela (vetor) de distâncias mínimas para todos os nós, em que as tabelas são responsáveis por orientar os pacotes para o nó desejado, apresentando uma próxima parada na rota e sempre priorizando a rota de menor custo. Um exemplo de RIP, que implementa esse protocolo, pode ser observado na Figura 8, na qual há as tabelas de roteamento em cada nó, com o custo (distância) e o próximo nó.

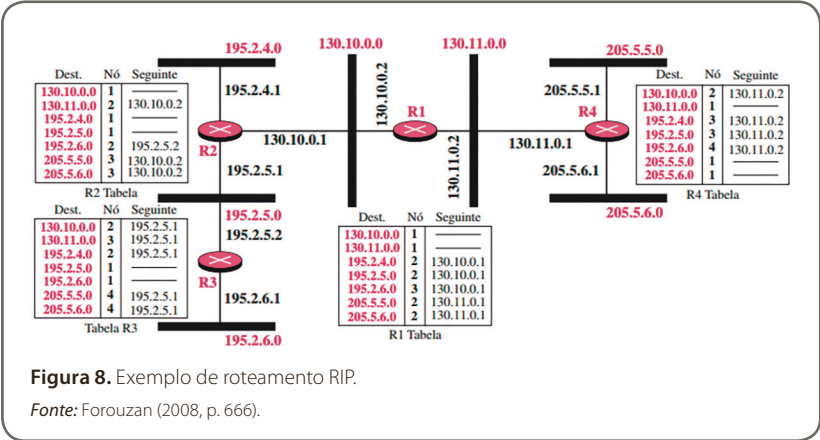


Figura 8. Exemplo de roteamento RIP.
Fonte: Forouzan (2008, p. 666).

No estado de enlace, se cada nó conter toda a topologia do domínio, como custo, tipo e condições dos enlaces (ativo ou inativo), o nó poderá utilizar o algoritmo de *Dijkstra* para construir a tabela de roteamento, sem levar em consideração a quantidade de nós na rota, mas sim as métricas, pois cada nó terá uma cópia da rota inteira. O OSPF implementa esse protocolo, cujo funcionamento pode ser observado na Figura 9.

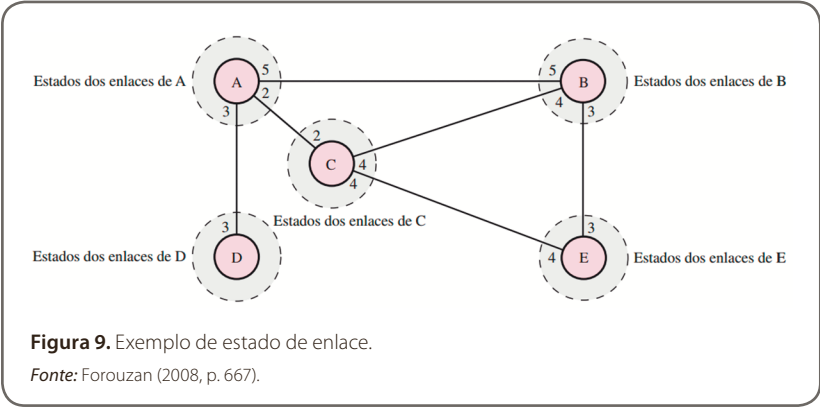
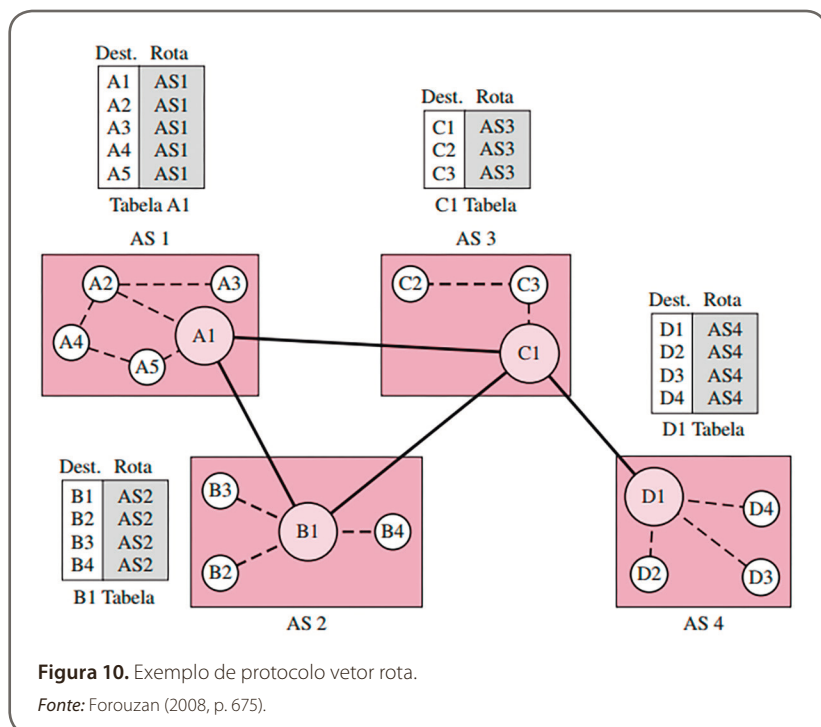


Figura 9. Exemplo de estado de enlace.
Fonte: Forouzan (2008, p. 667).

Basicamente, esse protocolo busca descobrir seus vizinhos e aprender seus endereços de rede, medir o retardo ou o custo até cada um de seus vizinhos, criar um pacote que informe tudo o que ele acabou de aprender, enviar pacotes a todos os outros roteadores e calcular o caminho mais curto até cada um dos outros roteadores (KUROSE; ROSS, 2007), trabalhando, assim, com base na ideia de que cada roteador dispõe de informações sobre as redes conectadas a ela.

Em sistemas autônomos interdomínios, os dois algoritmos citados não se enquadram, uma vez que o RIP poderia sofrer instabilidades caso exista um número um pouco maior de nós no domínio de operação e o OSPF demandaria de muito recurso para calcular todas as rotas. Desse modo, para esse tipo de sistema autônomo, há o protocolo vetor rota, que atua sob o princípio de utilizar nós específicos para comunicar-se com nós específicos de outras redes, em que cada nó anuncia para rede somente as rotas intradomínio, e não a métrica dos nós, como é possível observar na Figura 10, em que os nós representados por círculos maiores são os responsáveis por comunicar-se com as demais redes (KUROSE; ROSS, 2007). O algoritmo BGP implementa esse protocolo.



Nesse protocolo, roteadores com o protocolo BGP formam uma conexão TCP ponto a ponto, trabalhando sempre em pares. Cada rede dela é considerada um SA diferente. A função primária desse tipo de protocolo consiste em trocar informações sobre as rotas entre os sistemas BGP; assim, um SA é responsável por informar a rota que ela própria faz ao SA vizinho, e assim sucessivamente.

O algoritmo de roteamento BGP troca mensagens com informações sobre as rotas, utilizando-se do protocolo de transporte TCP para enviar e receber mensagens. Uma de suas principais características é permitir a seleção de rotas por definições políticas, como o fato de poder escolher uma rota mais custosa para atender questões políticas e evitar que pacotes passem por determinados territórios.



Link

A Cisco é uma líder mundial na área de redes, oferecendo diversos conteúdos em seu *site* oficial para que a comunidade consiga estudar tais conceitos. Veja no *link* a seguir o passo a passo da configuração do algoritmo RIP, a qual, para fins educacionais, pode ser implementada no *software* Cisco Packet Tracer, oferecida pela empresa gratuitamente.

<https://qrqo.page.link/CiPP7>



Referências

FOROUZAN, B. A. *Comunicação de dados e redes de computadores*. 4. ed. Porto Alegre: AMGH; Bookman, 2008. 1134 p.

KUROSE, J. F.; ROSS, K. W. *Redes de computadores e a Internet: uma abordagem top-down*. 3. ed. São Paulo: Pearson Addison Wesley, 2007. 634 p.

TANENBAUM, A. S. *Redes de computadores*. 4. ed. Rio de Janeiro: Campus, 2003. 945 p.



Fique atento

Os *links* para *sites* da Web fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:



SOLUÇÕES
EDUCACIONAIS
INTEGRADAS