

Algoritmos de roteamento em redes de computadores

Prof. Robson de Souza

A principal função da camada de rede é rotear os pacotes da máquina de origem até a máquina de destino.

O roteamento depende do fato de a origem e o destino não estarem na mesma rede.

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. Em circuitos virtuais não é necessário o roteamento, a não ser no momento em que o circuito é estabelecido, isso se chama roteamento por sessão.

O **roteamento** é diferente do **encaminhamento**. O roteamento é a tomada de decisão sobre quais rotas utilizar (exemplo: Preenchimento e atualização das tabelas de roteamento). O encaminhamento é o que acontece quando um pacote chega (exemplo: Procurar a linha de saída com base nas tabelas).

Os algoritmos de roteamento podem ser **não adaptativos**, nesse caso, a escolha das rotas é previamente calculada off-line quando a rede é inicializada, às vezes, esse procedimento é chamado **roteamento estático**.

Os algoritmos **adaptativos** mudam suas decisões de roteamento para refletir mudanças na topologia e no tráfego. Esses algoritmos diferem em termos do lugar em que obtém suas informações, momento em que alteram as rotas, e da unidade métrica usada para otimização.

Princípio da otimização → Se **J** está no caminho ótimo entre **I** e **K**, o caminho ótimo de **J** até **K** está na mesma rota.

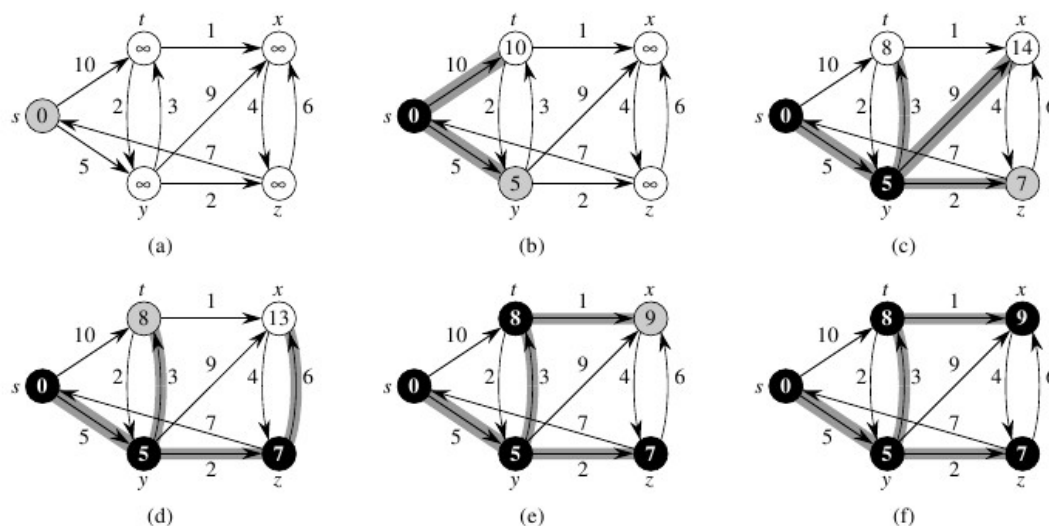
O conjunto de rotas ótimas de todas as origens para um determinado destino forma uma árvore com raiz no destino, chamada **árvore de escoamento**. Uma árvore de escoamento não é necessariamente exclusiva, podem existir outras com caminhos de mesmo tamanho.

Roteamento pelo caminho mais curto

Consiste em criar um grafo da sub rede, cada nó representa um roteador e cada arco uma linha de comunicação (enlace), o algoritmo encontra o caminho mais curto entre um par de roteadores.

Dijkstra

O algoritmo de Dijkstra trabalha de forma que cada nó é identificado por sua distância a partir do nó de origem ao longo do melhor caminho conhecido. Inicialmente nenhum caminho é conhecido, todos os nós são rotulados com infinito. Os rótulos podem mudar à medida que o algoritmo prossegue e os caminhos são encontrados, refletindo melhores caminhos. Quando se descobre que um rótulo representa um caminho mais curto, ele se torna permanente e nunca mais é alterado.



Inundação (Flooding)

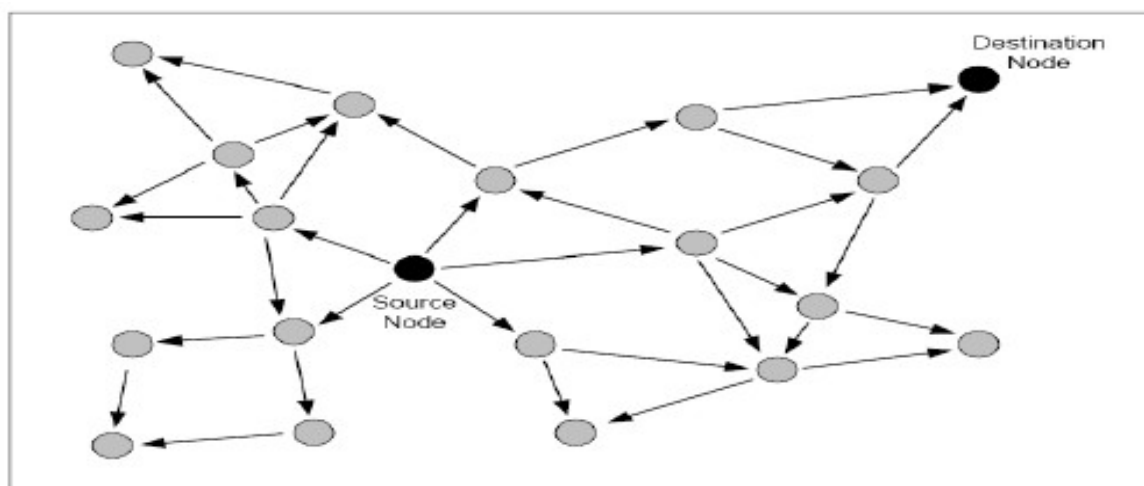
O algoritmo de inundação é um algoritmo estático, no qual cada pacote de entrada é enviado para toda linha de saída, exceto para aquela em que chegou. Esse algoritmo gera uma vasta quantidade de pacotes duplicados, pode ser um número infinito se algumas medidas não forem tomadas.

Essas medidas podem ser diversas, por exemplo, pode-se inserir um contador de hops no cabeçalho de cada pacote, a cada hop o contador é decrementado e o pacote é descartado quando o contador chegar a zero.

Outra medida poderia ser controlar quais pacotes foram transmitidos por inundação para não transmiti-los uma segunda vez, isso pode ser feito utilizando um número de sequência em cada pacote.

Outra opção é fazer uma **inundação seletiva**, onde os roteadores enviam cada pacote de entrada apenas para as linhas que provavelmente estão na direção certa.

O algoritmo de inundação é útil em aplicações militares, bancos de dados distribuídos e redes sem fio. Esse algoritmo sempre escolhe o caminho mais curto pois todos os caminhos possíveis são testados em paralelo.



Roteamento com vetor de distância

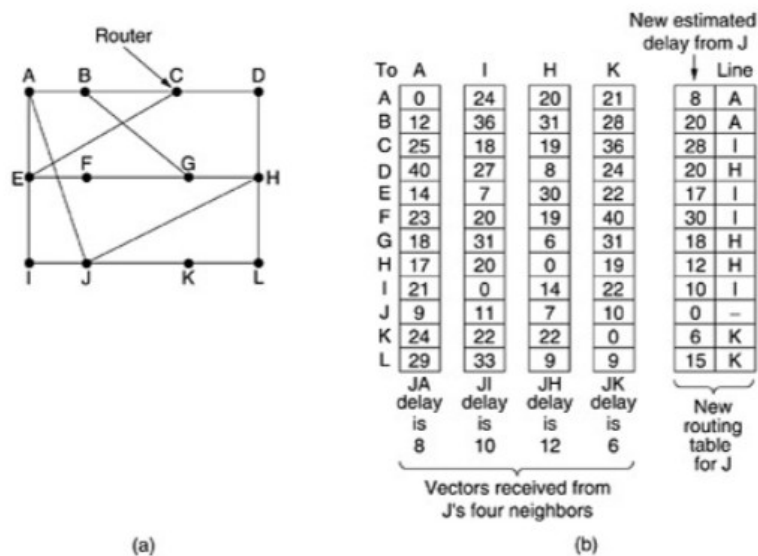
Algoritmos estáticos não levam em conta a carga atual da rede, os algoritmos de roteamento com vetor de distância são dinâmicos. Esses algoritmos operam fazendo cada roteador manter uma tabela (vetor), que fornece:

- A melhor distância conhecida até cada destino
- Qual linha deve ser utilizada para se chegar lá

Essas tabelas são atualizadas através da troca de informações. O algoritmo com vetor de distância era o algoritmo original da ARPANET e foi utilizado na Internet com o nome RIP, logo, o protocolo RIP utiliza o algoritmo de roteamento com vetor de distância.

Presume-se que o roteador conheça a distância até cada um dos seus vizinhos. A unidade métrica pode ser o hop, comprimento da fila, retardo, etc. O roteador pode usar esses dados para descobrir qual estimativa parece ser a melhor.

A figura abaixo mostra um exemplo onde se tem (a) uma sub rede com 12 roteadores, (b) as tabelas de retardo dos quatro vizinhos do roteador J e a tabela mais a direita mostra qual linha deve ser utilizada para se chegar ao destino a partir do roteador J:

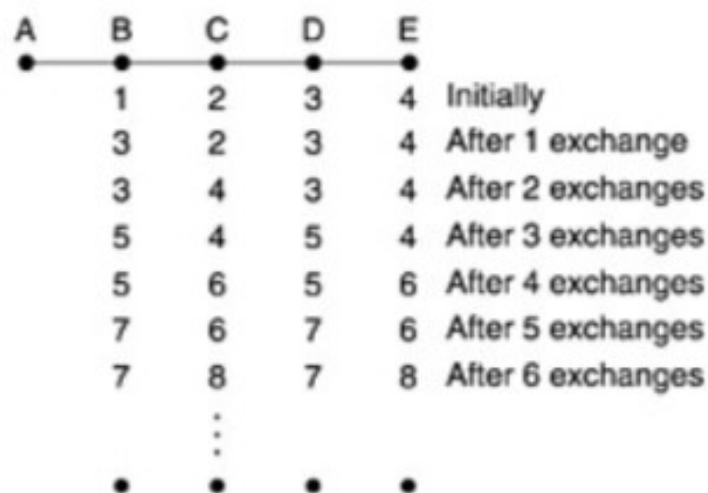


Embora o roteamento com vetor de distância encontre uma resposta correta, ele pode fazer isso muito lentamente. Ele reage com rapidez a boas notícias, mas reage devagar a más notícias. Ele também pode deixar de usar uma determinada linha dependendo do retardo.

Um problema que é muito grave em um roteamento com vetor de distância é o problema da contagem ao infinito, onde caso um roteador caia, os outros roteadores vão atualizar as suas tabelas infinitamente sem perceber que o roteador caiu.

Por exemplo, considere a figura abaixo, que mostra a distância entre os roteadores B, C, D e E até o roteador A, em hops. Inicialmente, os roteadores B, C, D e E contêm respectivamente, caminhos que levam 1, 2, 3 e 4 hops até A. Também vamos supor que nesse caso, os roteadores também estão ligados exatamente nessa ordem, ou seja, o A está ligado diretamente ao B, que está ligado diretamente ao C e assim por diante. Caso o roteador A caia, quando o roteador B descobrir essa informação, o roteador C indica que consegue acessar o roteador A com uma distância de 2 hops, o roteador B não sabe que esse caminho de C até A passa por ele mesmo, afinal, o roteador C pode ter várias outras saídas que também levam ao A (o que não é o caso aqui), logo, o B atualiza a sua distância até A para 3, passando pelo roteador C. Na próxima atualização, o roteador C vai perceber que o B aumentou a distância até A e vai aumentar a sua distância em mais 1, gerando 4, e assim por diante com todos os roteadores.

Ou seja, eles vão ficar contando infinitamente sem descobrir que o roteador A, na verdade, caiu.



Referências bibliográficas:

TANENBAUM, Andrew. S. Redes de Computadores. São Paulo: Pearson, 5ª Ed. 2011.

KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet – Uma Abordagem Top-Down. São Paulo: Pearson, 6ª Ed. 2013.