

Algoritmos de roteamento em redes de computadores – Continuação...

Prof. Robson de Souza

Algoritmo de roteamento por estado de enlace (Link State)

O algoritmo com vetor de distância não levava em conta a largura de banda, no início não tinha muito problema, pois todas as linhas tinham 56 Kbps, mas depois esse problema passou a ser importante.

Outro problema era a contagem até o infinito, o roteamento com vetor de distância foi substituído na ARPANET pelo roteamento por estado de enlace.

O roteamento por estado de enlace é composto por 5 partes:

- Descobrir seus vizinhos e aprender seus endereços de rede.
- Medir o retardo ou o custo até cada um dos seus vizinhos.
- Criar um pacote que informe tudo que ele acabou de aprender.
- Enviar esse pacote a todos os outros roteadores.
- Calcular o caminho mais curto até cada um dos outros roteadores.

A topologia e todos os retardos podem ser medidos experimentalmente e distribuídos para os outros roteadores. Em seguida, o algoritmo de Dijkstra pode ser utilizado para encontrar o caminho mais curto até cada um dos outros roteadores.

Conhecendo os vizinhos

A primeira tarefa quando um roteador é inicializado é conhecer os seus vizinhos, para isso, pode-se enviar um pacote HELLO em cada linha de saída, assim, o roteador que estiver na outra extremidade deve enviar uma resposta informando quem ele é.

Os nomes dos roteadores devem ser globalmente exclusivos para que os roteadores possam identificar os outros corretamente.

Medir o custo da linha

Para medir o retardo, ou pelo menos estimá-lo, a forma mais simples seria enviar um pacote ECHO pela linha que o outro lado deve transmitir de volta. Com isso, pode-se medir o tempo de ida e volta e dividir por 2. Também é interessante fazer isso várias vezes e obter uma média.

Um ponto importante nessa questão é se a carga deve ser levada em consideração, ou seja, se é pra contabilizar o tempo que o pacote fica na fila de espera também. Se for o caso de medir apenas o custo sem a carga, o timer deve ser inicializado quando o pacote ECHO atingir o início da fila. Para contabilizar a carga, o timer deve ser inicializado quando o pacote ECHO for enfileirado.

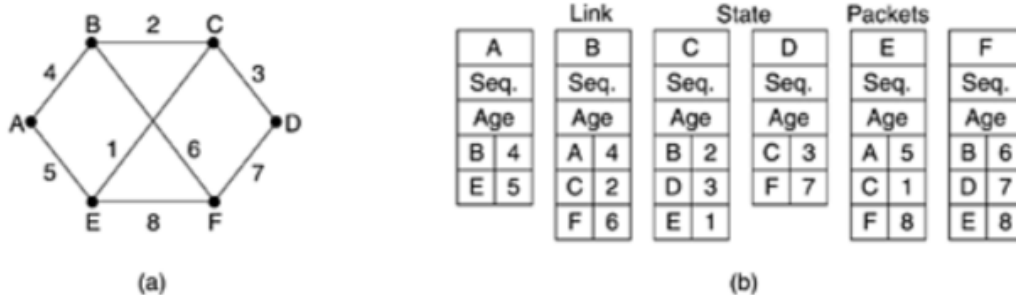
A vantagem em se contabilizar a carga na medição é que quando um roteador precisar escolher entre duas linhas com a mesma largura de banda, sendo que uma delas está pesadamente carregada, vai considerar a rota sobre a linha não carregada um caminho mais curto.

O problema disso é que, por exemplo, se existirem duas linhas e uma delas estiver fortemente carregada, o algoritmo vai enviar os pacotes para a outra linha, sobrecarregando-a e depois vai voltar para a primeira linha, ou seja, as tabelas de roteamento poderão oscilar muito.

Se a carga for ignorada e considerarmos só a largura de banda, esse problema não ocorrerá. Como alternativa, a carga pode ser distribuída entre as duas linhas, mas essa solução não utiliza totalmente o melhor caminho. Para evitar oscilações na escolha do melhor caminho, talvez seja sensato distribuir a carga por várias linhas, com alguma fração conhecida dessa carga sendo transmitida através de cada linha.

Criar pacotes de estado de enlace

O pacote começa com a identidade do transmissor, seguida por um número de sequência, idade e uma lista de vizinhos com o retardo referente a cada um deles.



Criar os pacotes é fácil, difícil é determinar quando criar. Pode ser periodicamente ou durante algum evento significativo, como uma linha ou um vizinho que sai do ar, entra em atividade novamente ou altera suas propriedades significativamente.

Distribuição dos pacotes de estado de enlace

A parte mais complicada desse algoritmo é distribuir os pacotes de forma confiável. Os roteadores que obtiverem os primeiros pacotes mudarão suas rotas, mas os diferentes roteadores talvez estejam usando diferentes versões da topologia, o que poderá levar a inconsistências, loops, máquinas inacessíveis e outros problemas.

A ideia básica é utilizar o algoritmo de inundação para distribuir os pacotes. Para manter o controle do algoritmo de inundação, cada pacote contém um número de sequência que é incrementado para cada novo pacote enviado. Os roteadores controlam todos os pares que vêm. Quando é recebido, o novo pacote é conferido, se for novo, ele será encaminhado para todas as linhas exceto a linha pela qual ele chegou. Se for uma cópia, ele será descartado.

Se um pacote recebido tiver um número de sequência mais baixo do que o mais alto número de sequência recebido até o momento, ele será rejeitado e considerado obsoleto, pois o roteador terá dados mais recentes.

Isso gera alguns problemas:

- Números de sequência podem se repetir se usarem poucos bits, uma possível solução é usar números de 32 bits.
- O roteador pode falhar, perder o número de sequência e começar de novo em 0, com isso, o novo pacote sempre será rejeitado, pois será considerado uma cópia.

- O número de sequência pode ser adulterado, por exemplo, o número 0000001 (1) sofrer um problema no bit mais significativo e virar 1000001 (65), desse modo, todos os pacotes de 1 a 65 serão rejeitados, pois serão considerados obsoletos.

Para solucionar esses problemas, pode-se incluir a idade de cada pacote após o número de sequência e decrementá-la uma vez por segundo. Quando a idade atingir zero, as informações desse roteador serão descartadas. O campo Idade também é decrementado por cada roteador durante o processo inicial de inundação para garantir que nenhum pacote será perdido e irá durar um período de tempo indefinido.

Alguns aprimoramentos podem ser feitos, por exemplo, pode-se não enfileirar imediatamente o pacote para transmissão, com isso, ele aguarda um pouco em uma área de retenção. Se um outro pacote da mesma origem chegar antes da transmissão do primeiro, os números de sequência serão comparados, se for cópia será descartada, se forem diferentes, o mais antigo será descartado.

Para evitar erros nas linhas entre dois roteadores, todos os pacotes são confirmados. Quando uma linha ficar ociosa, a área de retenção será varrida sequencialmente, afim de selecionar um pacote ou uma confirmação a enviar.

Calcular novas rotas

Com os pacotes de estado de enlace, pode-se criar o grafo completo da sub rede. Todo enlace será representado duas vezes (ida e volta). Pode-se utilizar a média dos valores ou utilizá-los separadamente. Em seguida, o algoritmo de Dijkstra pode ser usado para encontrar os caminhos mais curtos.

Os resultados desse algoritmo podem ser instalados nas tabelas de roteamento e a operação normal pode ser retomada. Podem ocorrer problemas com relação à quantidade de roteadores (memória gasta) e tempo de cálculo, mas no geral, o roteamento por estado de enlace funciona bem.

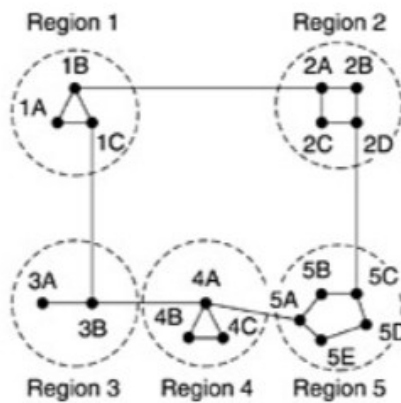
O roteamento por estado de enlace é amplamente utilizado em redes reais. O protocolo OSPF emprega um algoritmo de estado de enlace. Outro protocolo que usa esse algoritmo é o IS-IS.

Roteamento hierárquico

No roteamento hierárquico, os roteadores serão divididos em regiões, com cada roteador sabendo detalhes de como rotear para destinos dentro da sua própria região, mas sem conhecer nada sobre a estrutura interna de outras regiões.

Quando diferentes redes estão interconectadas, cada uma é vista como uma região separada. Em redes muito grandes, uma hierarquia de apenas 2 níveis pode ser insuficiente, para isso, é necessário alocar essas regiões em agrupamentos, os agrupamentos em zonas, as zonas em grupos e assim por diante.

O preço a ser pago é que mesmo que exista um caminho mais curto até um roteador passando por outra região, o pacote vai seguir levando em conta o caminho mais curto até a região daquele roteador.



Roteamento por difusão

O envio do pacote a todos os destinos simultaneamente é chamado difusão (broadcasting). Um método simples seria permitir que a origem enviasse um pacote específico a cada destino, isso não exigiria recursos especiais da sub rede mas desperdiçaria largura de banda e exigiria que a origem tivesse uma lista completa de todos os destinos. Outra opção seria a inundação, porém esse algoritmo tem uma série de problemas.

Pode-se utilizar o roteamento para vários destinos, nesse caso, cada pacote terá uma lista de destinos ou um mapa de bits indicando os destinos desejados, quando um pacote chega a um roteador, este verifica os destinos para determinar o conjunto de linhas de saída que serão necessárias (uma linha será necessária se for a menor rota para pelo menos um dos destinos).

O roteador gera uma cópia do pacote para cada linha de saída a ser utilizada e inclui em cada pacote somente os destinos que vão usar a linha.

Após um número suficiente de hops, cada pacote vai transportar somente um destino e poderá ser tratado como um pacote normal.

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.