Redes de Computadores III --/--/----

Aula __: Algoritmo Vetor de Distância

Professor: Eraldo Silveira e Silva eraldo@ifsc.edu.br

1 Objetivos da Aula

- Apresentar o algoritmo vetor de distâncias;
- Discutir algumas condições de convergência do algoritmo;
- Apresentar a solução de envenenamento reverso.

Esta nota de aula é baseada no livro de Kurose [1].

2 O Algoritmo Vetor de Distância

2.1 Princípios do Algoritmo

O algoritmo vetor de distâncias (DV) é distribuído, assíncrono e iterativo.

No algoritmo DV cada roteador mantém uma Tabela de Distâncias. Nesta tabela, cada linha é associada a um destino na rede e cada coluna associada a um roteador vizinho. Uma elemento na tabela de um dado roteador X pode ser descrito como $D^X(Y,Z)$, que é, do ponto de vista de X, o custo para chegar em Y, passando pelo seu vizinho Z. Este custo é dado por:

$$D^X(Y,Z) = c(X,Z) + min_w\{D^Z(Y,w)\} \tag{1}$$

Ou seja, o custo para chegar em Y, passando pelo vizinho Z, é o custo para ir de X a Z mais o mínimo custo de Z para Y. Note que w representa os nós vizinhos de Z (incluindo o próprio X).

A equação 1 sugere que cada roteador deve informar aos seus vizinhos o seu custo mínimo para chegar até um dado destino. Na realidade, assim que um roteador computar um novo mínimo para um destino ele deve informar aos seus vizinhos.

2.2 Um exemplo inicial

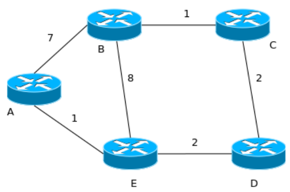
Seja a Fig.1 que mostra a tabela de distâncias para o nó E, após a convergência, para a rede em questão. Considere a linha referente ao destino A. As seguintes considerações podem ser feitas:

- O custo para A a partir de E é 1 ($D^E(A,A)$);
- O custo para chegar a A via D é 5. O caminho para obter este custo passa pelo próprio E;
- O custo para A a partir de E via B é 14. Qual é o caminho de menor custo para A via B? Este custo somado com o $c(E,B)$ resulta em 14.

Na tabela de vetor de distâncias, os menores custo estão marcados com um quadrado. Estas marcações são a base para a construção da *tabela de encaminhamento*.

Construa no espaço abaixo a tabela de encaminhamento para o roteador E baseando na tabela de distâncias acima.

Observe que a análise acima foi realizada olhando-se para todo o grafo da rede. O algoritmo DV (de Bellman-Ford) não tem esta informação pois é decentralizado, conhecendo apenas os custos até os seus vizinhos. Tais algoritmos são usados nos protocolos BGP e RIP.



$D^E()$	A	B	D
A	$\boxed{1}$	14	5
B	7	8	$\boxed{5}$
C	6	9	$\boxed{4}$
D	4	11	$\boxed{2}$

Figura 1: Tabela de Distâncias para o nó E

Figura 2: Tabela de Encaminhamento para o roteador E

3 O Algoritmo

Considere o algoritmo DV executado a partir do nó X conforme mostrado na Tabela.1.

Note que o algoritmo é executado pelo ponto de vista de um nó X. O algoritmo opera sobre um vetor de distâncias a partir do qual deve ser gerada a tabela de encaminhamento. Um procedimento de inicialização é inicialmente realizado (linhas 2 a 6). Os custos de X para todos os nós da rede indo através de seus vizinhos são iniciados com INFINITO, a exceção quando o destino é o próprio vizinho, quando então o custo é o do enlace para o vizinho. Em seguida, o nó X envia para todos os seus vizinhos o menor custo para cada destino.

Na segunda parte, o algoritmo entra em um *loop* eterno e se bloqueia esperando por um evento. Este evento pode ser a mudança de custo de para um determinado vizinho ou a recepção de uma mensagem de um vizinho informando que existe um novo valor de custo para um determinado destino.

Caso o evento seja a mudança de custo do enlace até o vizinho então o nó X atualiza (linhas 9 a 11) todos os custos para os destinos através do vizinho (todos os valores da coluna associada ao vizinho).

Caso o evento seja o recebimento de um novo custo adicional *newval* (positivo ou negativo) provindo de um vizinho então X deve atualizar o custo para este destino, no seu vetor de distâncias, indo pelo vizinho em questão (linhas de 13 e 14).

Finalmente, se for computado um novo mínimo para um novo destino então este novo custo mínimo deve ser enviado para todos os vizinhos (linhas 15 e 16).

É bom ressaltar que o algoritmo é iterativo e assíncrono: cada iteração local é causada por uma mudança do custo do enlace local ou por uma mensagem do vizinho (mudança de caminho de menor custo para algum destino). O termo assincrono indica que isto pode acontecer a qualquer momento, sem sincronismo com outros eventos da rede.

INICIALIZAÇÃO:
  PARA todos nós adjacentes v **FAZ**
    $D^X(*, v) = \infty$
    $D^X(v, v) = c(X, v)$
  PARA todos os destinos y **FAZ**
    enviar $min_w D^X(y, w)$ para cada vizinho
REPETE
  Espera por evento
  SE $c(X, V)$ muda em d unidades **ENTÃO**
    Para todos destinos y faz
      $D^X(y, V) = D^X(y, V) + d$
  SENÃO
    SE recebeu atualização $newval$ de custo, de um vizinho V para um destino Y **ENTÃO**
      Para este destino Y faz $D^X(Y, V) = c(X, V) + newval$
  SE existe um novo $min_w D^X(Y, w)$ para qualquer destino Y **ENTÃO**
    envia novo mínimo para todos os vizinhos
PARA SEMPRE

Tabela 1: Algoritmo Vetor de Distâncias executado no nó X

4 Executando o algoritmo

Seja a rede (grafo) mostrada na Fig.3 A execução do algoritmo leva a sequência de tabelas mostrada na Fig.4.

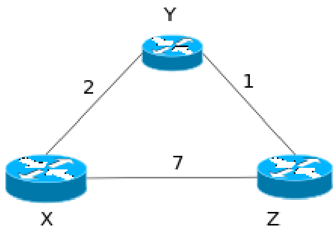


Figura 3: Rede a ser aplicada o algoritmo DV

Seja a rede da figura 5. Uma mudança no custo de um enlace, por exemplo, de 4 para 1, entre os roteadores X e Y leva a uma sequência de atualizações tal como na Fig.6 No exemplo, no tempo t_0 o nó Y detecta a mudança de custo, atualiza a sua tabela de distâncias e repassa o novo valor aos seus vizinhos. Este exemplo mostra que boas notícias são rapidamente propagadas.

Se o custo entre X e Y passar de 4 para 60 então o algoritmo se comporta como mostrado na Fig.7. Neste caso, as más notícias fazem com que o algoritmo demore para convergir. É o problema de **contagem ao infinito**.

O problema acima pode ser contornado em parte pelo "**envenenamento de rotas reversas**" (Fig.8). Nesta abordagem, se Z roteia via Y p/ chegar a X então Z informa para Y que sua distância para X é infinita (isto para que Y não roteie para X via Z).

5 Exercícios

1. Faça uma pesquise e discuta as vantagens e desvantagens no uso dos protocolos DV e Estado de Enlace.

2. Explique o que é o problema de contagem ao infinito e como pode ser contornado no algoritmo DV.

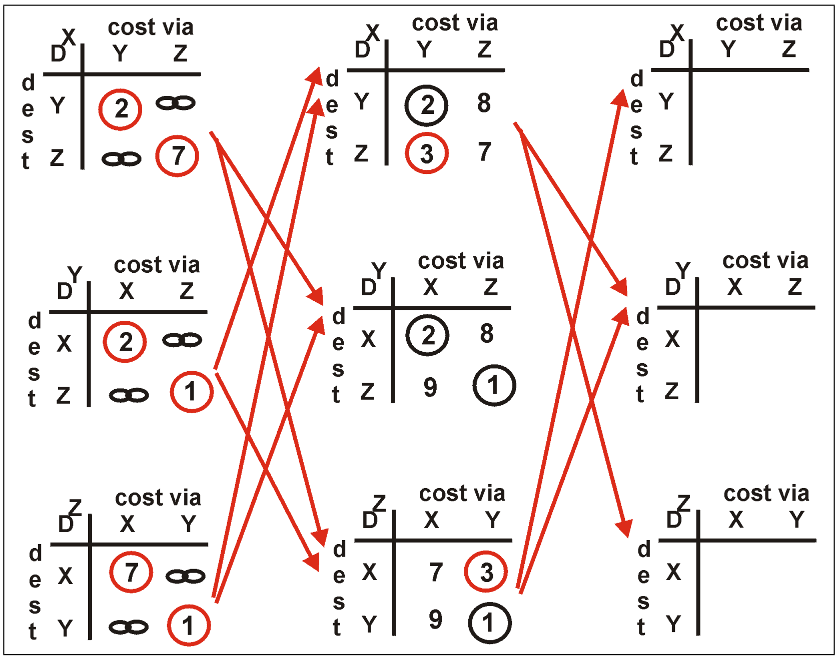


Figura 4: Sequência de tabelas de distância

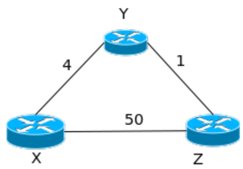


Figura 5: Rede do exemplo de convergência

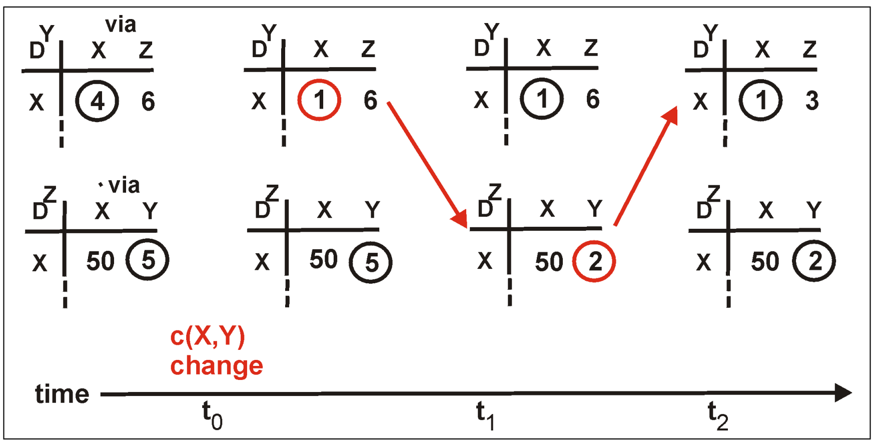


Figura 6: Mudança no custo de um enlace para um valor mais baixo

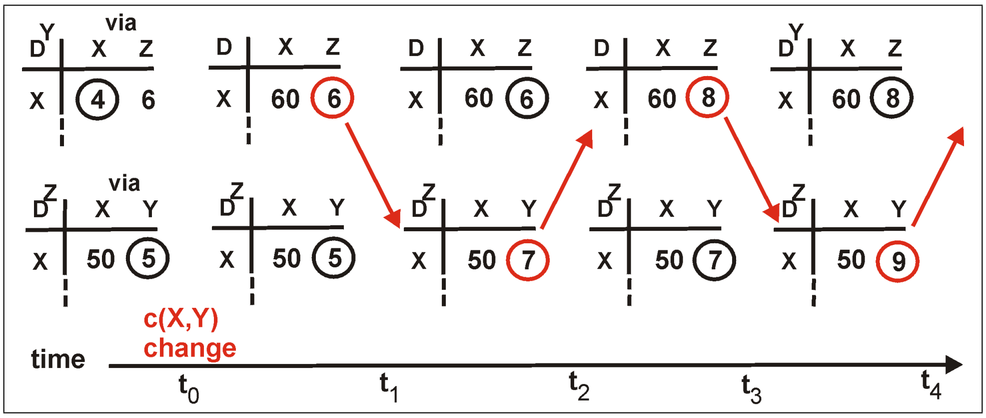


Figura 7: Mudança no custo de um enlace para um valor mais alto

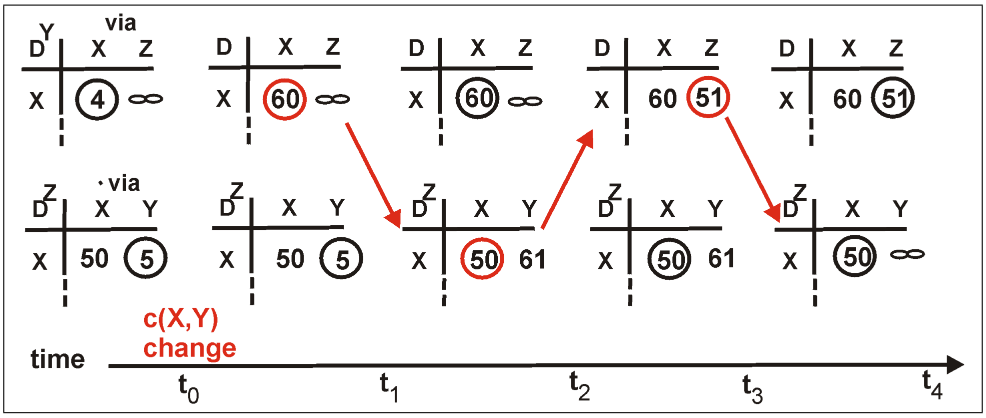


Figura 8: Reverso envenenado

3. Descreva uma situação em que o envenenamento de rota pode não resolver o problema do alto tempo de convergência do algoritmo DV sob condições de más notícias (aumento de custo de um enlace).

4. Compute as tabelas de distância para os roteadores do grafo da Fig.9. Qual a tabela de roteamento de X?

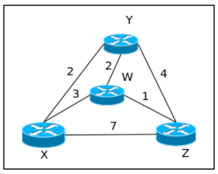


Figura 9: Grafo de Rede

Referências

[1] J. Kurose e K. Ross. *Redes de Computadores e a Internet: Uma abordagem top-down. Tradução da 3ª edição.* Addison Wesley, 2003.