

REDES PARA INTERNET

Algoritmos de Roteamento – Link State

4.3

Universidade Estácio de Sá
Prof. Welsing Moreira Pereira
<http://www.professorwelsing.webnode.com>

O que sabemos é uma gota, o que ignoramos é um oceano.
Isaac Newton (1642-1727) – Físico e Matemático Inglês.



Estácio

1

Link State



- A topologia completa e todos os retardos medidos experimentalmente são distribuídos para cada roteador.
- Depois de possuir o mapa completo da topologia, cada roteador pode rodar o algoritmo de Dijkstra para encontrar o caminho mais curto para cada outro roteador

2

Link State



- Cada roteador deve:
 1. Descobrir os seus vizinhos e aprender seus endereços de rede
 2. Determinar o custo para alcançar cada um deles.
 3. Construir um pacote, denominado *link state packet* –LSP, contendo uma lista dos vizinhos com os respectivos custos.
 4. Difundir seu LSP para todos os outros roteadores da rede, armazenando os mais recentes LSPs recebidos dos demais roteadores.
 5. Computar a melhor rota (caminho mais curto) para cada um dos roteadores.

3

Link State



- Identificação dos Vizinhos
 - Enlaces ponto-a-ponto:
 - Troca de pacotes especiais de identificação (pacotes *HELLO*)
- Determinando o custo de cada linha
 - Exemplo com custo=retardo:
 - Pacote especial ECHO é enviado em cada linha
 - Outra ponta deve respondê-lo imediatamente
 - retardo medido é dividido por 2
 - teste realizado várias vezes, e depois a média é computada
 - considera ou não o tempo de espera na fila (carga do enlace)

4

- Construção do LSP

Formato do LSP

Endereço de Origem (Identificação do nó)	
Número de seqüência	
Tempo de vida	
Endereço do Vizinho	Custo
...	...
Endereço do Vizinho	Custo

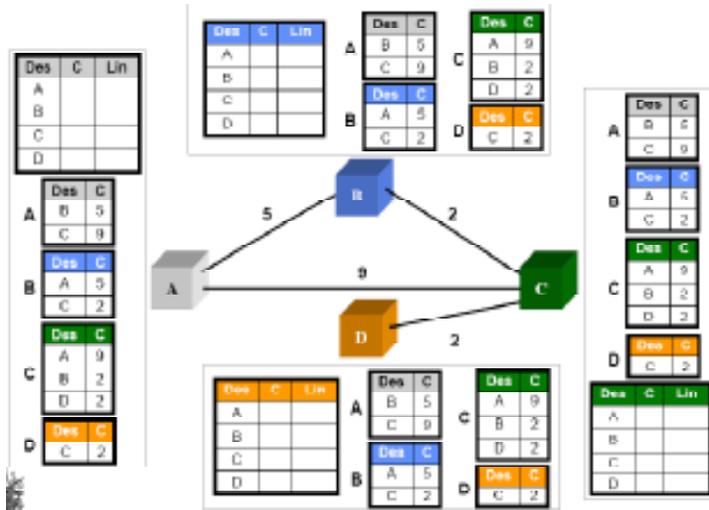
- Um nó gera um novo LSP em várias situações:
 - Periodicamente
 - Logo que descobre um novo vizinho
 - Logo que “cai” um enlace de comunicação com um vizinho
 - Logo que o custo de um enlace para um vizinho existente é alterado

- Difusão do LSP
 - Parte mais complexa e crítica do algoritmo
 - Garantir uma consistência nos conjuntos de LSPs de cada nó
 - Evitar que os LSPs se multipliquem de uma forma desordenada
 - Garantir a confiabilidade (correção e ordenação) na entrega dos LSPs
 - Evitar a propagação dos LSPs utilizando
 - informações de roteamento da rede para não criar um problema recursivo
 - Flooding (adaptado)

- Difusão do LSP
 - Flooding para distribuir LSPs
 - cada pacote contém um número de seqüência (32bits) incrementado a cada novo LSP enviado
 - roteadores guardam pares (roteador_origem,número_seqüência) que recebem
 - quando chega um LSP:
 - se for novo (número de seqüência maior que o armazenado) , é encaminhado em todas as portas exceto a de origem
 - se for duplicata, é descartado
 - se tiver número de seqüência menor que o armazenado para aquela origem, é descartado como obsoleto.

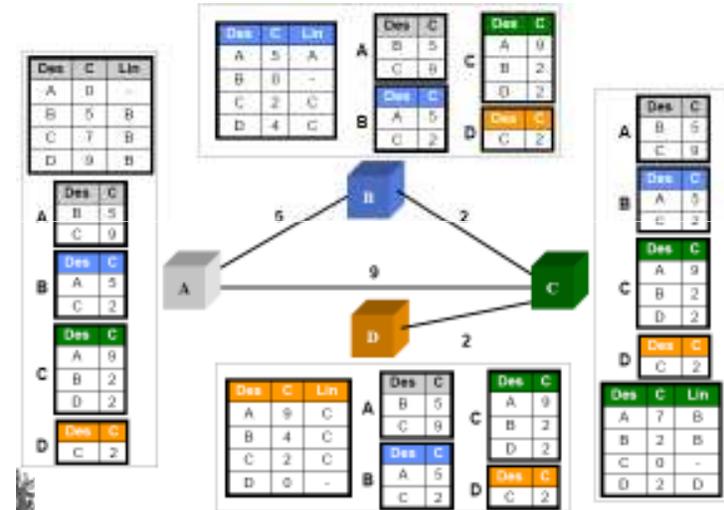
- Cálculo das Rotas
 - Feito baseando-se no conjunto de LSPs recebidos
 - Cada roteador constrói a topologia completa
 - Algoritmo normalmente utilizado é o algoritmo de Dijkstra, executado localmente, para descobrimento do caminho mais curto

Link State



9

Link State



10

Link State



- Exemplos de protocolo de roteamento baseados em *link state*
 - “Novo” protocolo de roteamento da ARPANET
 - IS-IS (Intermediate System-Intermediate System – originado do DECnet fase V e adotado como padrão pela ISO - CLNP)
 - OSPF (Open Shortest Path First – RFC 2328)
 - NLSP (NetWare Link Services Protocol) (Novell IPX)
 - PNNI (Private Network-to-Network Interface)

11

Distance Vector x Link State



- Problemas com software ou hardware dos roteadores podem causar sérios danos ao roteamento. No *link state*, por exemplo, exemplos de mau funcionamento de um nó podem ser:
 - Anunciar que possui um enlace que na verdade não possui
 - Não divulgar um enlace que possui
 - Não propagar corretamente o LSP ou corromper o seu conteúdo
 - Efetuar o cálculo das rotas de forma incorreta

12

Distance Vector x Link State



- Principais vantagens do algoritmo *link state*
 - Menor tempo de convergência
 - Mais fácil inferir a topologia da rede
 - Mais fácil identificar falhas em enlaces da rede
- Principal vantagem do algoritmo *distance vector*
 - Simplicidade

Exercícios



- Descreva o funcionamento do algoritmo de roteamento Link State.
- Compare Distance Vector e Link State.