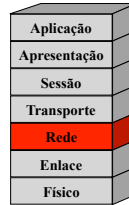


Disciplina: Redes de Computadores

Nível de Rede

Profa. Débora Christina Muchaluat Saade
debora@midia.com.uff.br

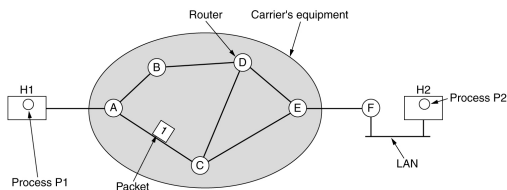
Nível de Rede



- ✓ Endereçamento
- ✓ Roteamento
- ✓ Transmissão de unidades de dados do serviço de rede
- ✓ Segmentação e blocagem
- ✓ Detecção e recuperação de erros
- ✓ Sequenciação
- ✓ Controle de congestionamento
- ✓ Serviços
 - orientado à conexão
 - não orientado à conexão

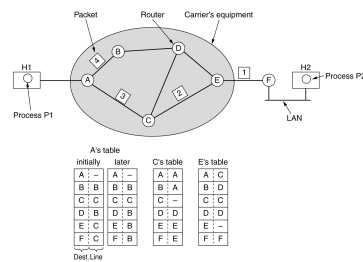
Comutação de Pacotes - Store-and-Forward

Ambiente de um protocolo de rede



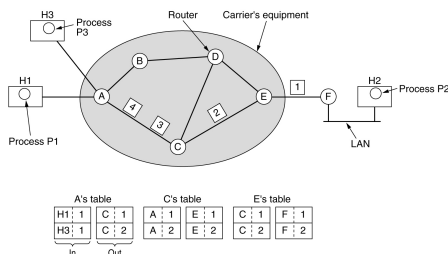
Implementação de um Serviço sem Conexão

Encaminhamento de datagramas



Implementação de um Serviço Orientado à Conexão

Encaminhamento de pacotes em um circuito virtual



Comparação entre Circuito Virtual e Datagrama

Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Endereçamento

Redes de Computadores

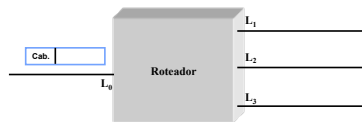
- ✓ Hierárquico
- ✓ Horizontal
- ✓ Mapeamento do SAP de Rede no endereço da estação
 - *Vinculação direta*
 - tabelas de conversão
 - *Vinculação dinâmica*
 - Protocolo de resolução de endereços

10

Algoritmos de Roteamento

Redes de Computadores

- ✓ Para qual linha de saída do roteador deve ser encaminhado um pacote recebido?
- ✓ Conexão de Circuito Virtual x Datagrama



13

Algoritmos de Roteamento

Redes de Computadores

- ✓ Propriedades Desejáveis
 - *correto*
 - *simples*
 - *robusto*
 - *estável*
 - *justo*
 - *ótimo*
- ✓ O que otimizar?
 - *minimizar a distância percorrida por cada pacote*
 - *minimizar o retardo médio dos pacotes*
 - *maximizar a vazão (throughput) total da rede*
 - *minimizar o número de saltos (hops)*

14

Algoritmos de Roteamento

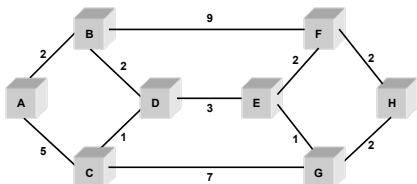
Redes de Computadores

- ✓ Analogia com estudo de grafos
 - *Roteador = Nó*
 - *Enlace = Aresta*
- ✓ Medidas de custo nos enlaces
 - *valor unitário (salto ou hop)*
 - *distância*
 - *tempo médio de espera na fila de transmissão + tempo de transmissão propriamente dito*
 - *capacidade de transmissão*
 - *custo da linha de comunicação*

18

Algoritmos de Roteamento

Redes de Computadores

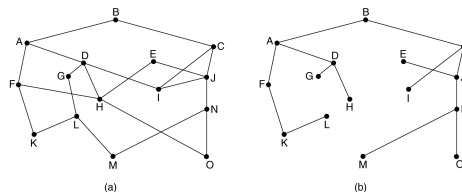


19

The Optimality Principle

Redes de Computadores

- (a) A subnet. (b) A sink tree for router B.



20

Caminho mais Curto (Shortest Path)

Redes de Computadores

- ✓ Diversos algoritmos na literatura para computar o caminho mais curto entre dois nós de um grafo [Tanenbaum 1996, Dijkstra 1959]
- ✓ Baseia-se no uso de uma métrica:
 - número de saltos (hops)
 - distância
 - capacidade de transmissão
 - tempo médio de espera na fila de transmissão + tempo de transmissão propriamente dito
 - custo da linha de comunicação

21

Caminho mais Curto - Dijkstra

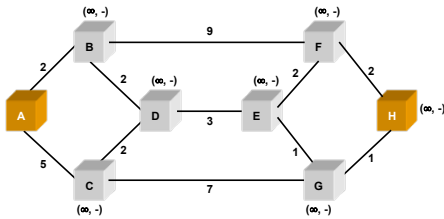
Redes de Computadores

- ✓ Cada nó tem um rótulo que pode ser:
 - *tentativa*
 - *permanente*
- ✓ o rótulo mantém a menor distância ao nó origem usando o melhor caminho conhecido (inicialmente é infinito)
- ✓ quando descobre-se que a distância ao nó origem é a menor dentre as tentativas, faz-se o rótulo permanente

22

Caminho mais Curto

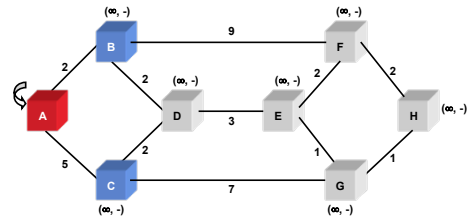
Redes de Computadores



23

Caminho mais Curto

Redes de Computadores



Vizinhos: B, C

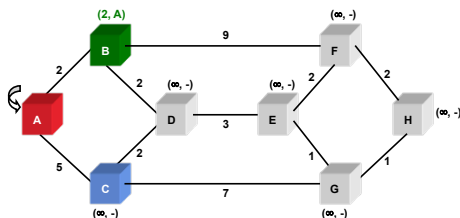
Tentativas:

Permanentes: A

24

Caminho mais Curto

Redes de Computadores



Vizinhos: C

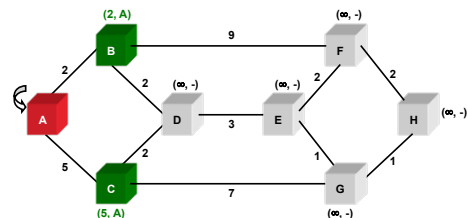
Tentativas: B

Permanentes: A

26

Caminho mais Curto

Redes de Computadores

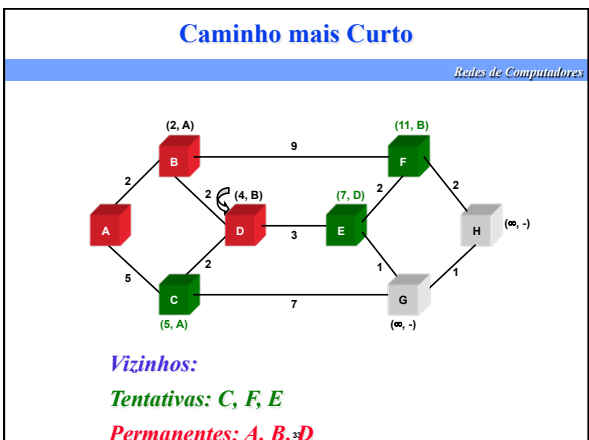
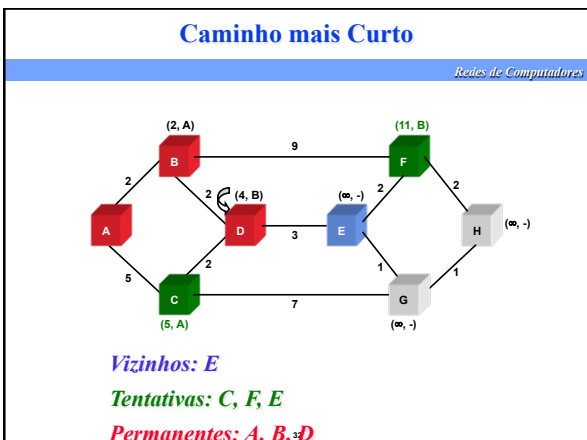
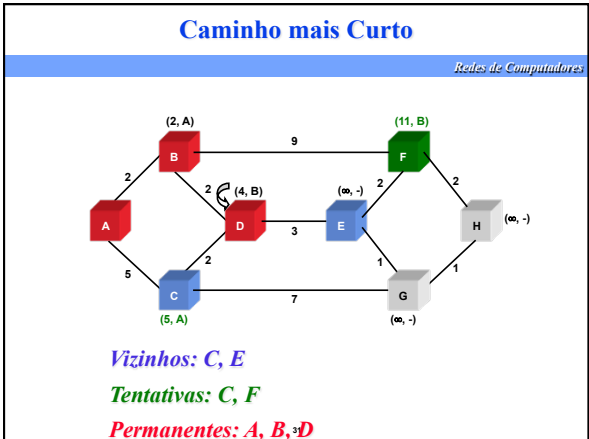
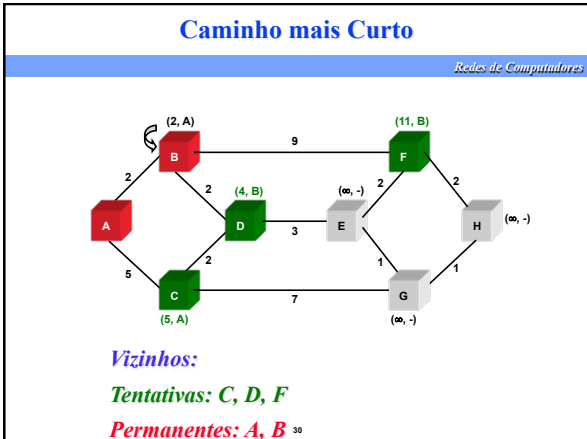
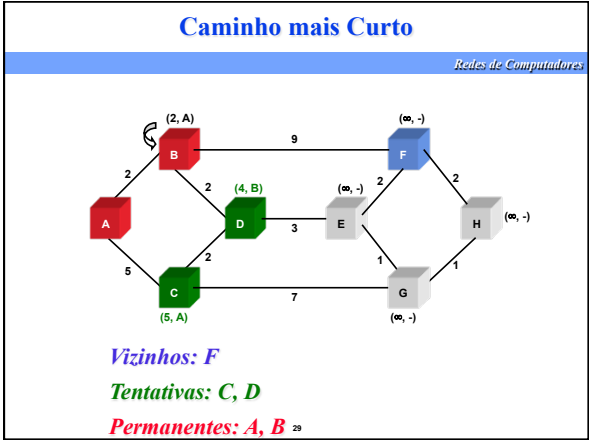
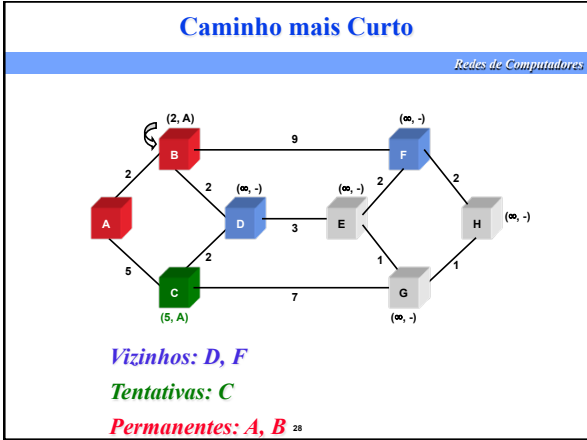


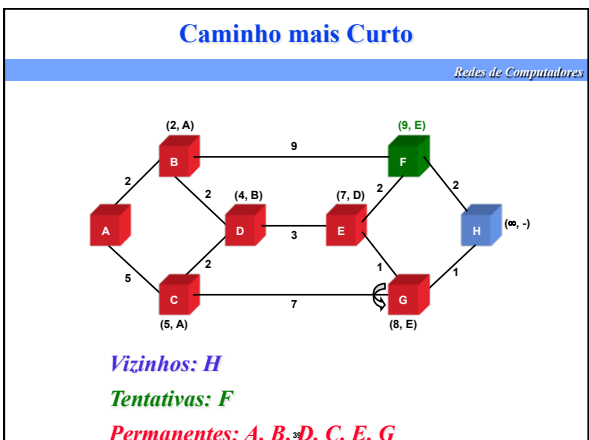
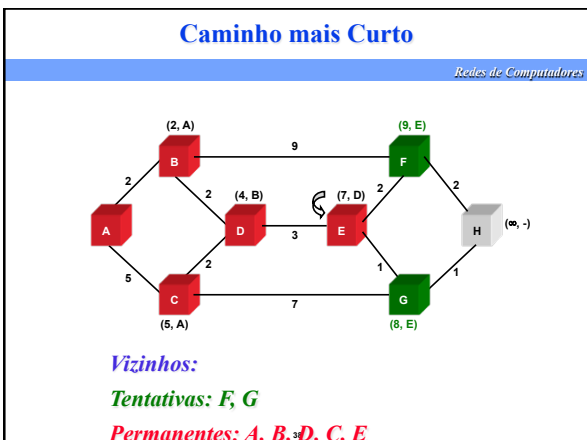
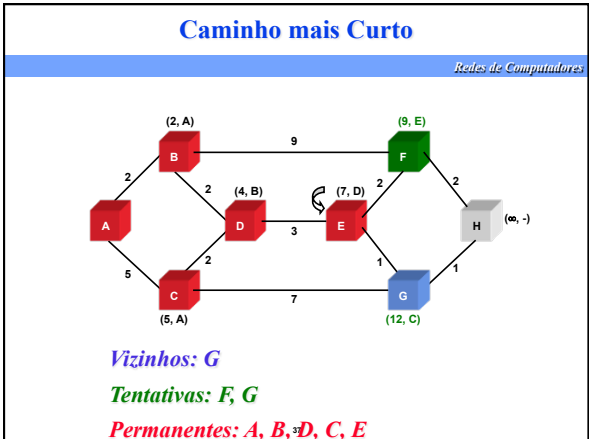
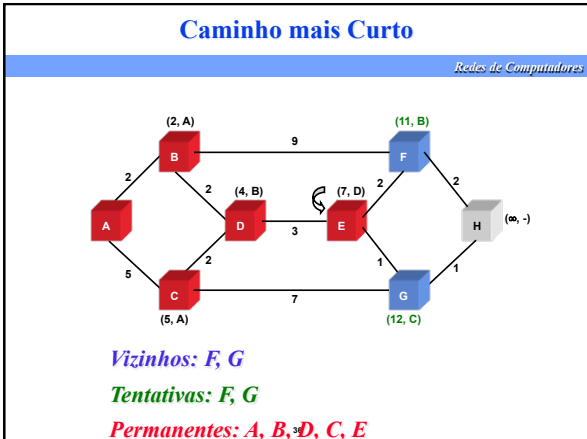
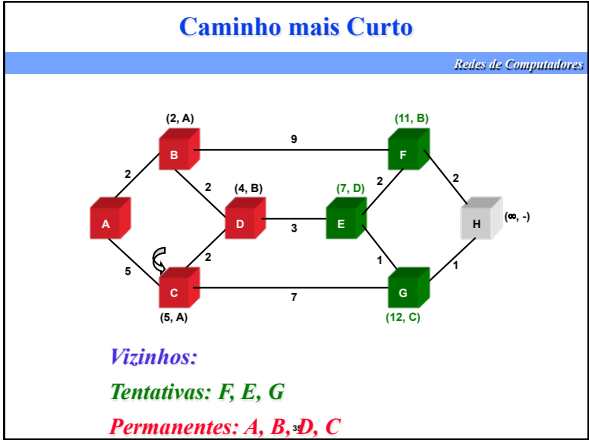
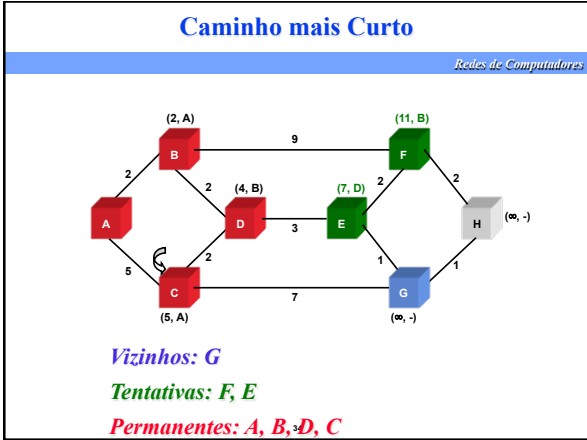
Vizinhos:

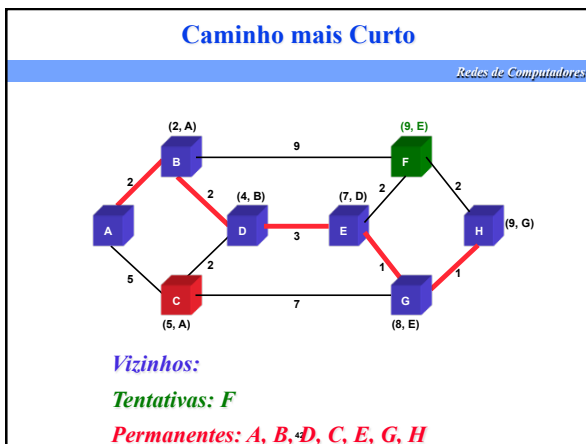
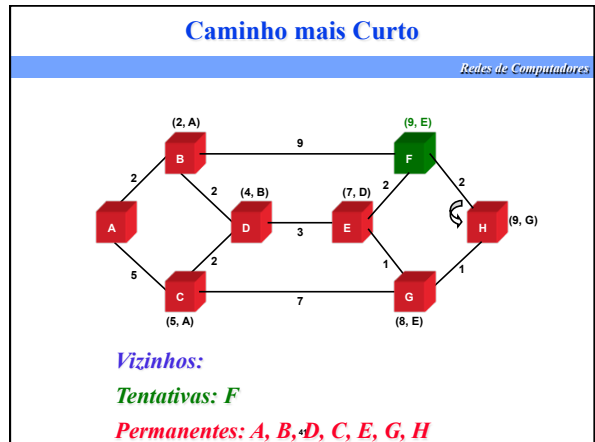
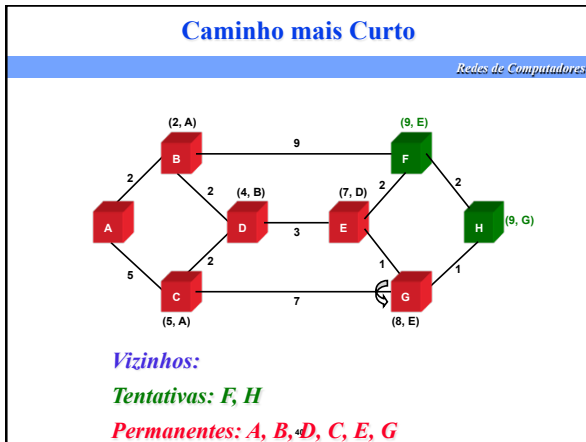
Tentativas: B, C

Permanentes: A

27

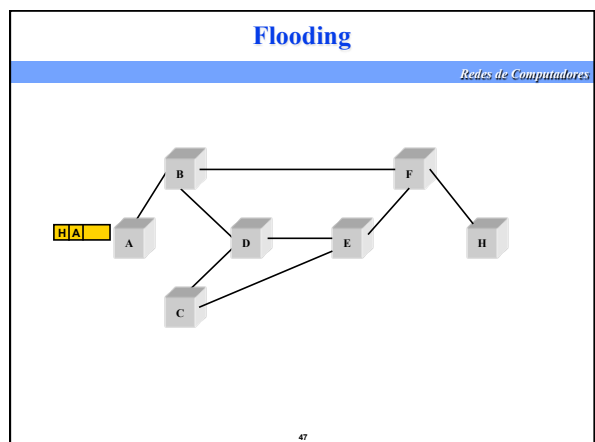


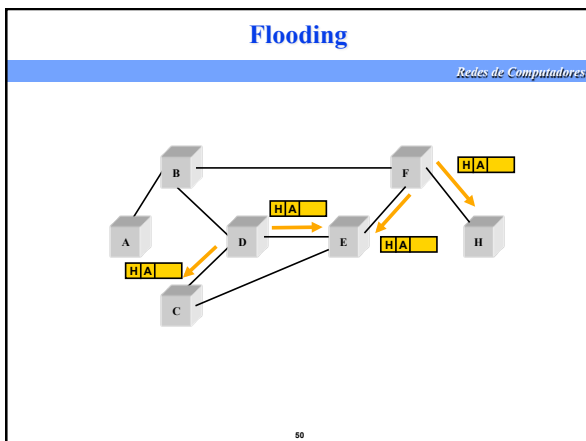
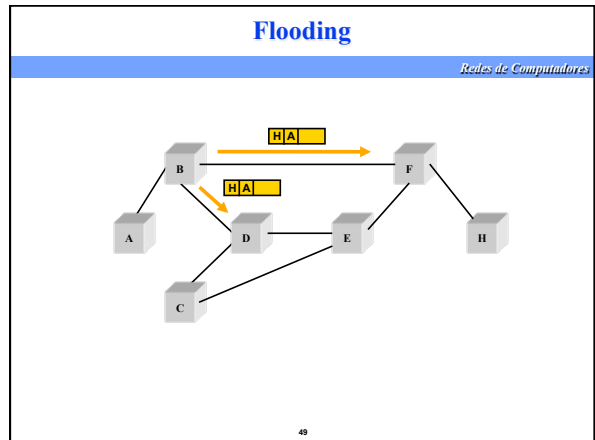
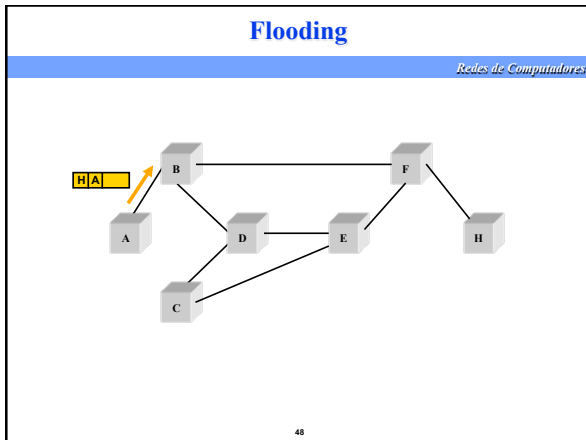




- ### Algoritmos de Roteamento
- Redes de Computadores
- ✓ Baseiam-se em uma estrutura de dados chamada **TABELA de ROTAS** que possui os campos:
 - endereço de destino
 - custo do caminho (mais curto) até o destino
 - linha de saída
 - ✓ Classificam-se de acordo com a forma de atualização da tabela:
 - Não adaptativos (Estáticos ou por Rota Fixa)
 - Adaptativos (Dinâmicos)
- 43

- ### Algoritmos de Roteamento Não Adaptativos (Roteamento Estático ou por Caminho Fixo)
- Redes de Computadores
- ✓ Uma vez criadas as tabelas, as mesmas não são mais alteradas
 - ✓ Caminhos alternativos são utilizados apenas em caso de falhas
 - ✓ Aspectos negativos
 - Probabilidade alta de uma má utilização dos meios de comunicação
 - Dificuldade para manutenção e gerência
- 46





- ### Flooding
- Redes de Computadores
- ✓ Crescimento exponencial do número de pacotes na rede
 - ✓ Exige a utilização de algum mecanismo de descarte dos pacotes para evitar que pacotes fiquem eternamente circulando na rede
 - Contador de saltos no cabeçalho de cada pacote
 - Não guarda a informação dos pacotes que já foram repassados por ele
 - Selective flooding
 - ✓ Pouco prático, porém útil em algumas aplicações
 - Aplicações militares (robustez do flooding é desejável)
 - Atualização de bancos de dados distribuídos
 - Métrica de avaliação para outros algoritmos de roteamento (sempre utiliza o caminho mais curto, pois usa todos os caminhos em paralelo)
- 52

- ### Algoritmos de Roteamento Adaptativos (Roteamento Dinâmico)
- Redes de Computadores
- ✓ Rota escolhida de acordo com o arranjo topológico da rede, ou mesmo com a carga na rede
 - ✓ Atualização periódica das tabelas de rota
 - Modo centralizado
 - Modo distribuído
- 55

- ### Algoritmos de Roteamento Dinâmicos Modo Distribuído
- Redes de Computadores
- ✓ Os nós trocam entre si suas respectivas informações locais para cálculo das tabelas de rota.
 - Envio de informações pode ser feito
 - Periodicamente
 - Em resposta a alterações nas informações dos enlaces
 - ✓ Exemplos de algoritmos
 - Distance Vector
 - Link State
- 60

Redes de Computadores

Distance Vector

61

Redes de Computadores

Distance Vector

- ✓ Também conhecido pelos nomes de
 - *Algoritmo de roteamento de Bellman-Ford*
 - *Algoritmo de roteamento de Ford-Fulkerson*
- ✓ Algoritmo de roteamento utilizado pelos primeiros protocolos de nível de rede
 - *“Antigo” algoritmo de roteamento da ARPANET (até 1979)*
 - *RIP (Routing Information Protocol, RFC 1058)*
 - *IPX (primeiras versões)*
 - *DECnet (primeiras versões)*

62

Redes de Computadores

Distance Vector

Des	C	Lin
A	0	-
B	∞	-
C	∞	-
D	∞	-

Des	C	Lin
A	∞	-
B	0	-
C	∞	-
D	∞	-

Des	C	Lin
A	∞	-
B	∞	-
C	0	-
D	∞	-

Des	C	Lin
A	∞	-
B	∞	-
C	0	-
D	∞	-

64

Redes de Computadores

Distance Vector

- ✓ Cada nó possui:
 - *identificador único*
 - *custo de cada enlace*
- ✓ O nó transmite o seu vetor de distâncias (com destino/custo) para cada um dos seus vizinhos sempre que o seu vetor se modifica (ou periodicamente)
- ✓ Cada nó mantém o vetor de distâncias mais recente enviado por cada um de seus vizinhos
- ✓ Cada nó calcula o seu próprio vetor de distâncias, minimizando o custo para cada destino
- ✓ O vetor de distâncias é recalculado sempre que:
 - *um vizinho enviar um vetor de distâncias contendo informações diferentes das anteriores*
 - *houver a queda em um enlace para um vizinho. Nesse caso, o vetor de distâncias desse vizinho é descartado para que o vetor de distâncias do nó seja recalculado*

65

Redes de Computadores

Distance Vector

Des	C	Lin
A	0	-
B	∞	-
C	∞	-

Des	C	Lin
A	∞	-
B	0	-
C	∞	-

Des	C	Lin
A	∞	-
B	∞	-
C	0	-

66

Redes de Computadores

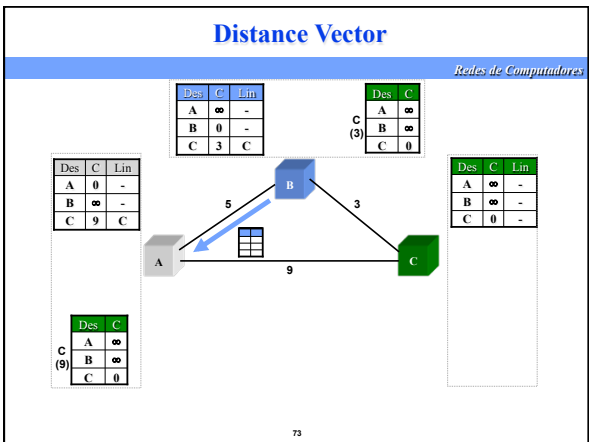
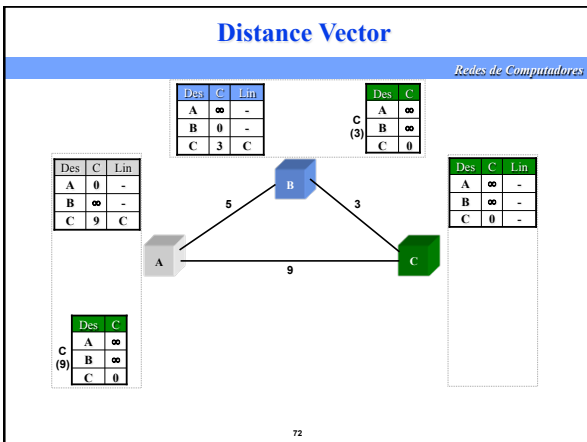
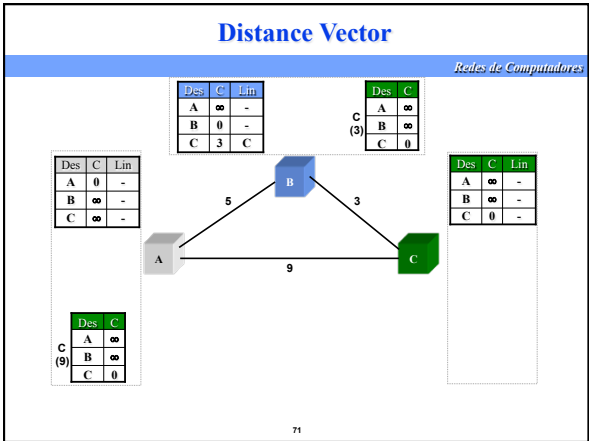
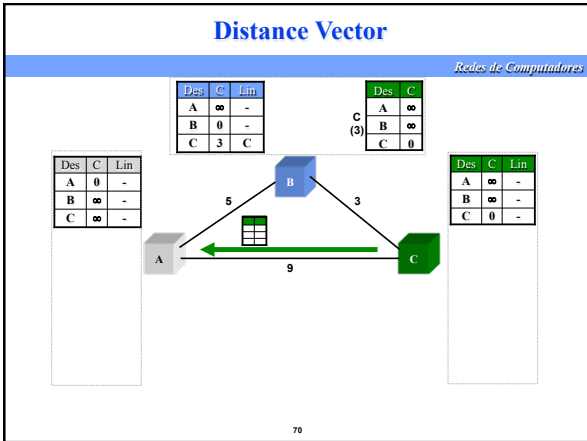
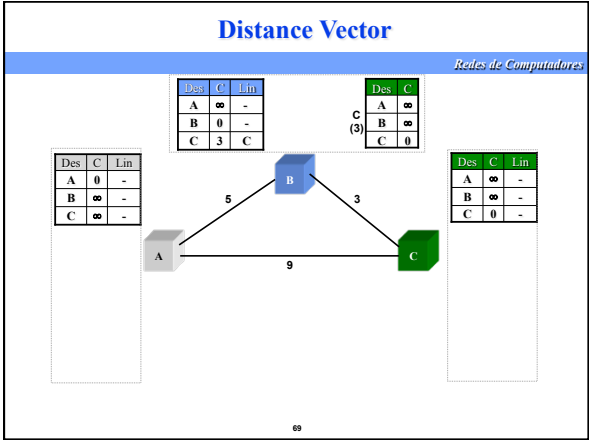
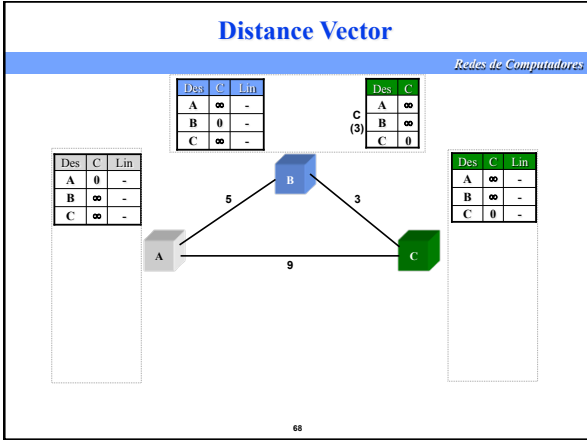
Distance Vector

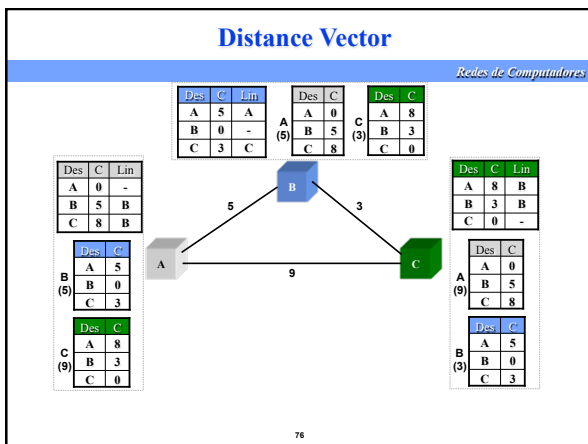
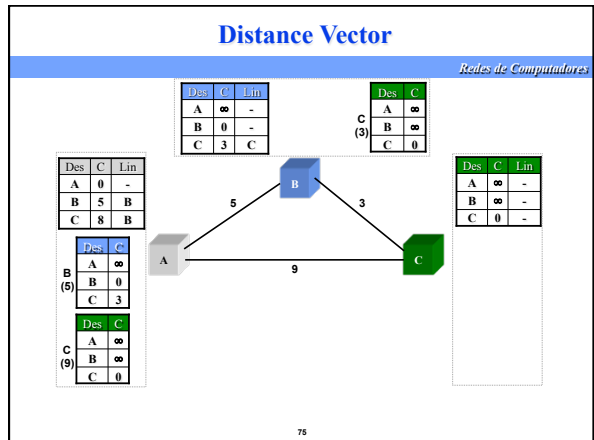
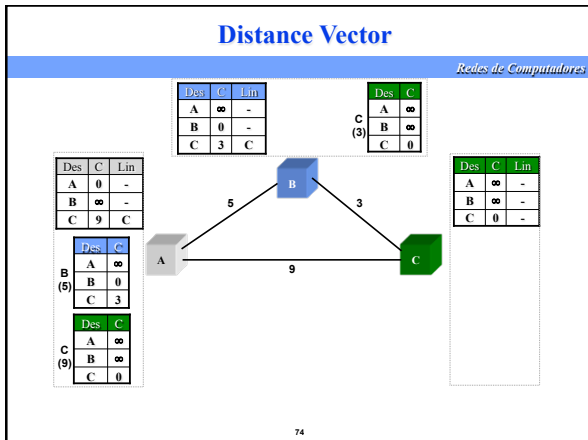
Des	C	Lin
A	0	-
B	∞	-
C	∞	-

Des	C	Lin
A	∞	-
B	0	-
C	∞	-

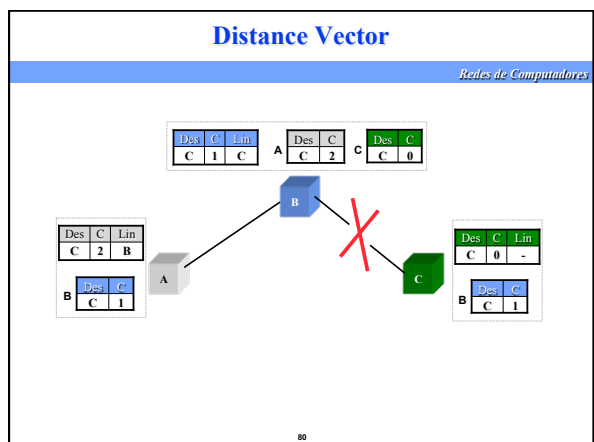
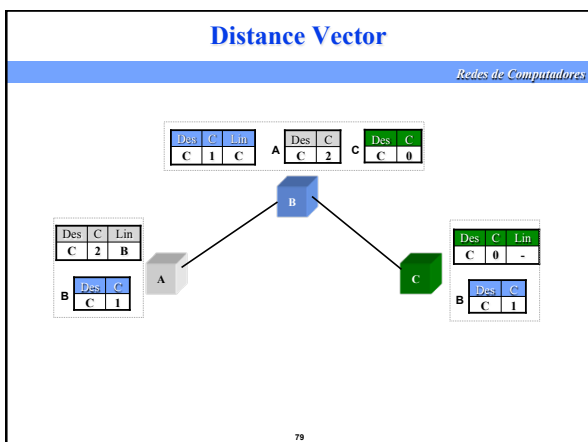
Des	C	Lin
A	∞	-
B	∞	-
C	0	-

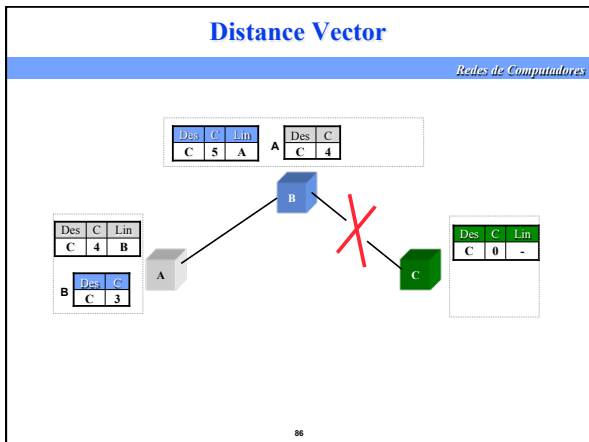
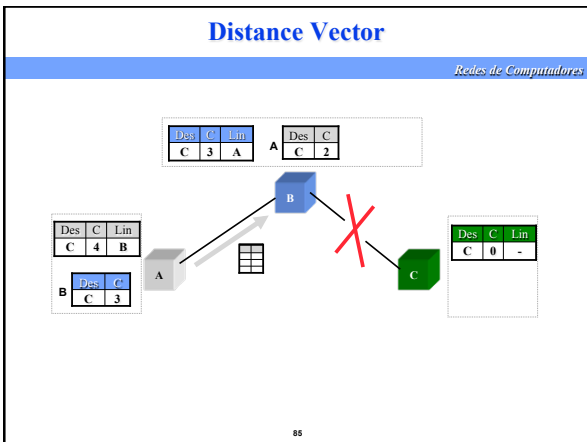
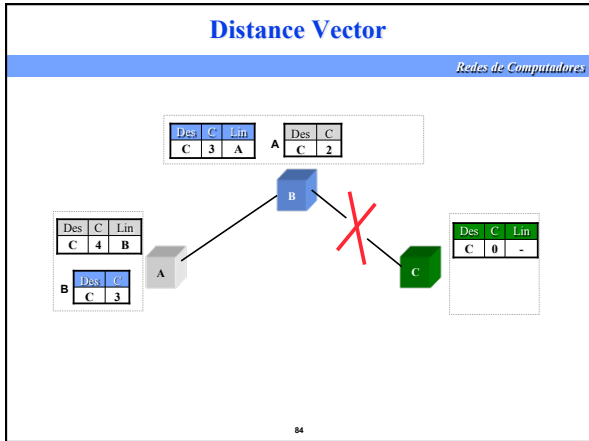
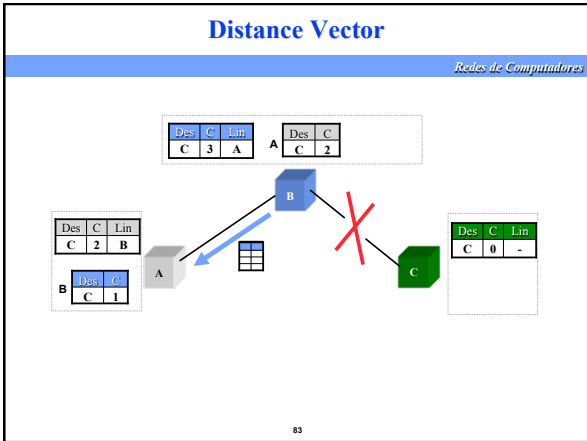
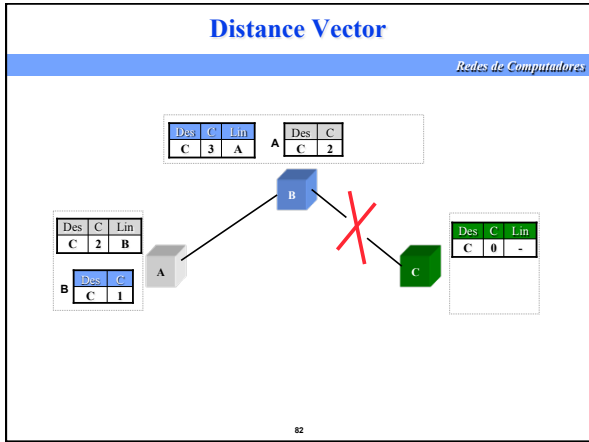
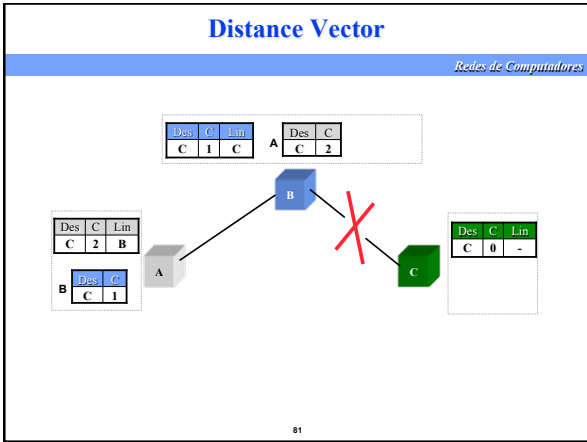
67





- ### Distance Vector
- Redes de Computadores
- ✓ **Vantagens**
 - *Simplicidade*
 - *Tempo de convergência baixo quando a rede opera bem*
 - ✓ **Principal desvantagem**
 - *Tempo de convergência muito alto quando ocorrem problemas na rede*
 - Problema da contagem para infinito (*count-to-infinity problem*)
- 77





Distance Vector

Redes de Computadores

- ✓ Diversas propostas para melhorar a questão do tempo de convergência [Perlman 1999]

- **Split Horizon:** a distância ao nó X não é reportada na linha por onde pacotes com destino X são encaminhados

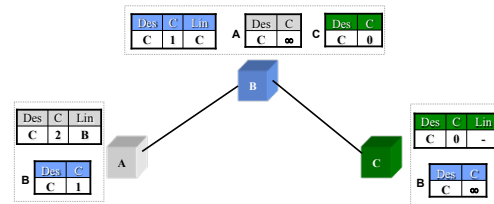
– *poison reverse:* a distância é reportada como sendo infinito

- ✓ Nenhuma das propostas resolve a questão satisfatoriamente

87

Distance Vector – Split Horizon & Poison Reverse

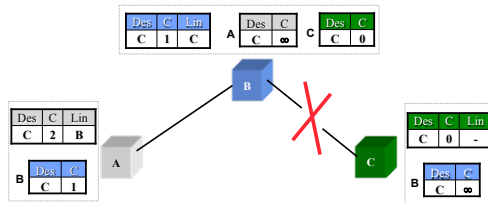
Redes de Computadores



92

Distance Vector – Split Horizon & Poison Reverse

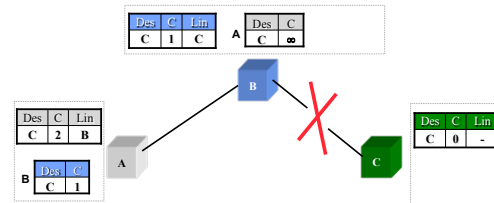
Redes de Computadores



93

Distance Vector – Split Horizon & Poison Reverse

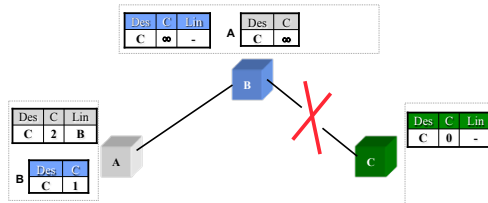
Redes de Computadores



94

Distance Vector – Split Horizon & Poison Reverse

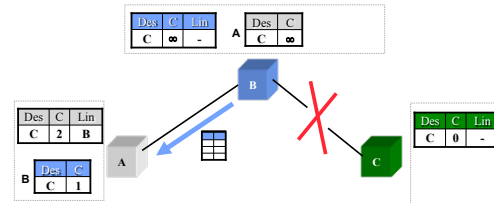
Redes de Computadores



95

Distance Vector – Split Horizon & Poison Reverse

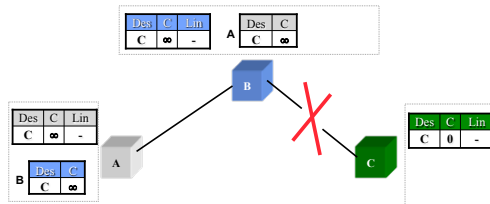
Redes de Computadores



96

Distance Vector – Split Horizon & Poison Reverse

Redes de Computadores



97

Redes de Computadores

Link State

98

Link State

Redes de Computadores

- ✓ A topologia completa e todos os custos (p.e., 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

99

Link State

Redes de Computadores

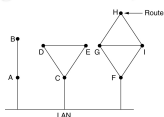
- ✓ Cada roteador deve
 1. Descobrir os seus vizinhos e aprender seus endereços de rede
 2. Determinar o custo para alcançar cada um deles.
 3. Construir um pacote, denominado link state packet – LSP, contendo uma lista dos vizinhos com os respectivos custos.
 4. Difundir seu LSP para todos os outros 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.

100

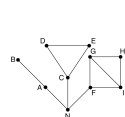
Link State Identificação dos Vizinhos

Redes de Computadores

- ✓ Enlaces ponto-a-ponto
 - Troca de pacotes especiais de identificação (pacotes HELLO)
- ✓ LANs conectando roteadores
 - Considerar a LAN como um nó artificial (LAN designated router)



(a) Nine routers and a LAN.



(b) A graph model of (a).

101

Link State Determinando o custo de cada linha

Redes de Computadores

- ✓ 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)

103

Link State Construção do LSP

Redes de Computadores

Formato do LSP

Endereço de Origem (Identificação do nó)	
Número de sequência	
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*

105

Link State Difusão do LSP

Redes de Computadores

- ✓ 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)*

106

Link State Difusão do LSP

Redes de Computadores

- ✓ Flooding para distribuir LSPs
- ✓ cada pacote contém um número de sequência (32 bits) incrementado a cada novo LSP enviado
- ✓ roteadores guardam pares (roteador_origem, número_sequência) 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 sequência menor que o armazenado para aquela origem, é descartado como obsoleto.*

108

Link State Difusão do LSP

Redes de Computadores

- ✓ Problemas:
 - *reiniciar número de seqüência*
 - *se um roteador falhar, perderia a seqüência e pacotes seriam encarados como duplicata*
 - *erro de transmissão no número de seqüência*
- ✓ Solução:
 - *tempo de vida no pacote, decrementado por cada roteador que o repassa ou o está armazenando*
 - *quando a idade chegar a zero, o roteador deleta os dados associados ao LSP (vizinhos e custos) e retransmite o cabeçalho do LSP para cada um de seus vizinhos*
 - *todo LSP deve ser reconhecido (ack), para isso, cada roteador guarda dois flags para cada linha de saída para cada LSP (um indica que o LSP deve ser retransmitido e o outro indica que um ack deve ser enviado)*

110

Difusão de LSPs

Redes de Computadores

LSP	Flags	Vizinho_1	Vizinho_2	Vizinho_3
Roteador_origem	Send	1	0	0
n_seq	Ack	0	1	0
tempo_de_vida				

Tabela em cada roteador para manter o controle de envio e recebimentos de LSPs

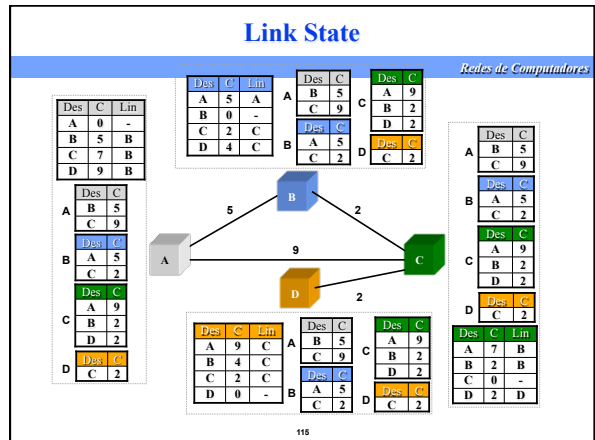
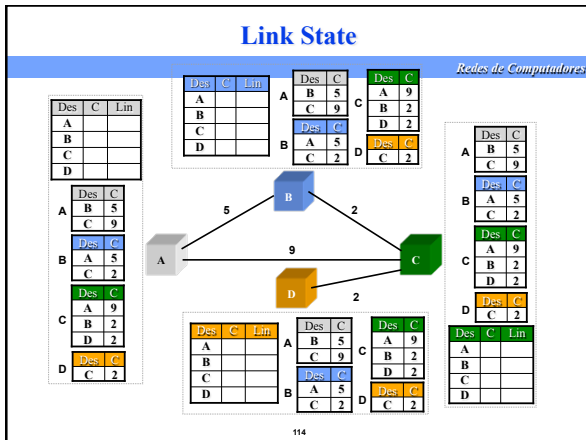
112

Link State Cálculo das Rotas

Redes de Computadores

- ✓ 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

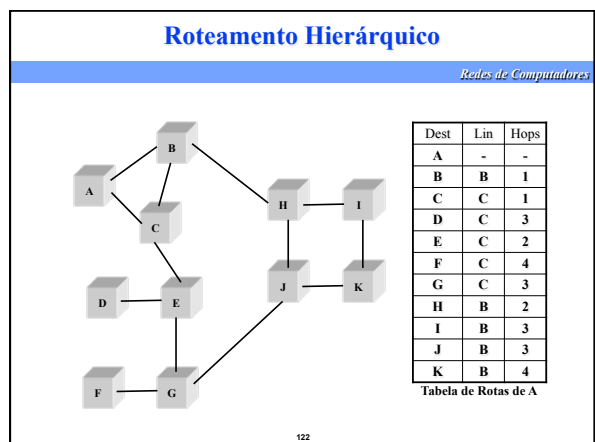
113



- ### Link State
- Redes de Computadores
- ✓ 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)
- 116

- ### Distance Vector x Link State
- Redes de Computadores
- ✓ 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
 - Permite de forma trivial que a rede opere com roteamento na origem
 - ✓ Principal vantagem do algoritmo *distance vector*
 - Simplicidade
- 119

- ### Roteamento Hierárquico
- Redes de Computadores
- ✓ Crescimento das redes pode levar a um crescimento das tabelas que as tornam intratáveis
 - ✓ Dividir os nós em regiões, com cada nó capaz de manter as informações de rotas das regiões a que pertence
 - ✓ O agrupamento pode ser feito utilizando um número arbitrário de níveis
 - regiões, clusters, zonas, grupos, etc.
- 121



Roteamento Hierárquico

Redes de Computadores

Dest	Lin	Hops
A	-	-
B	B	1
C	C	1
Reg. 2	C	2
Reg. 3	C	3
Reg. 4	B	2

Tabela de Rotas de A

123

Roteamento Hierárquico

Redes de Computadores

Full table for 1A			Hierarchical table for 1A		
Dest	Line	Hops	Dest	Line	Hops
1A	-	-	1A	-	-
1B	1B	1	1B	1B	1
1C	1C	1	1C	1C	1
2A	1B	2	2	1B	2
2B	1B	3	3	1C	2
2C	1B	3	4	1C	3
2D	1B	4	5	1C	4
3A	1C	3			
3B	1C	2			
4A	1C	3			
4B	1C	4			
4C	1C	4			
5A	1C	4			
5B	1C	5			
5C	1B	5			
5D	1C	6			
5E	1C	5			

(a) (b) (c)

124

Roteamento por Difusão (Broadcasting)

Redes de Computadores

- ✓ Um pacote para cada nó de destino (origem deve ter lista completa de destinos)
- ✓ Flooding (gera muitos pacotes e consome muita largura de banda)
- ✓ Pacotes contendo múltiplos destinos (cada roteador encaminha o pacote na melhor rota, adaptando a lista de destinos conforme necessário)
- ✓ Spanning tree contendo todos os roteadores (às vezes a spanning tree é conhecida, LSR, às vezes não, DVR)
- ✓ Reverse Path Forwarding

127

Reverse Path Forwarding

Redes de Computadores

➔ Um nó só propaga o pacote de difusão recebido de um nó X, se o pacote chegar pela linha que o nó utilizaria para transmitir dados para o nó X, caso contrário, o pacote é descartado como sendo duplicata.

128

Reverse Path Forwarding

Redes de Computadores

Spanning Tree

129

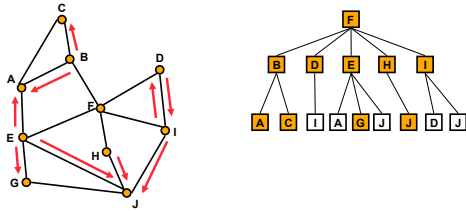
Reverse Path Forwarding

Redes de Computadores

130

Reverse Path Forwarding

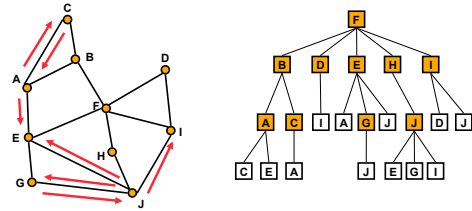
Redes de Computadores



131

Reverse Path Forwarding

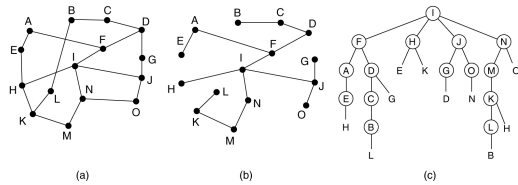
Redes de Computadores



132

Broadcast Routing

Redes de Computadores



Reverse path forwarding. (a) A subnet. (b) a Sink tree. (c) The tree built by reverse path forwarding.

133