

Unidade IV – Roteamento

TP308 – Introdução às Redes de Telecomunicações

© Antônio M. Alberti 2007

Tópicos

- ✓ Serviços Providos pela Camada de Rede
- ✓ Classificação dos Algoritmos de Roteamento
- ✓ Roteamento Centralizado
- ✓ Roteamento Isolado
- ✓ Roteamento por Inundação
- ✓ Roteamento Distribuído
- ✓ Roteamento pelo Caminho mais Curto
- ✓ Algoritmo de *Dijkstra*
- ✓ Roteamento por Vetor de Distância
- ✓ Roteamento por Estado de Enlace



© Antônio M. Alberti 2007

- Roteamento**
- ## Serviços Providos pela Camada de Rede
- ✓ O papel da **camada de rede** é **aparentemente** simples: transportar pacotes de uma estação remetente a uma estação destinatária.

 - ✓ Para realizar esta tarefa, **três** importantes **funções** podem ser identificadas:
 - **Determinação do Trajeto**
 - **Estabelecimento de Conexão**
 - **Comutação dos Pacotes**
- Unidade 4**

© Antônio M. Alberti 2007

- Roteamento**
- ## Classificação dos Algoritmos de Roteamento
- ✓ Podemos classificar os algoritmos de roteamento de acordo com:
 - A **dinâmica** das **tabelas de roteamento**:
 - **Estáticos**
 - Trajetos são definidos e não mais alterados, a menos que haja alteração topológica da rede.
 - **Dinâmicos**
 - Os trajetos são alterados periodicamente para refletir mudanças no estado da rede, tais como congestionamentos, falhas, carga, etc.

 - A **forma** como a **decisão de roteamento** é tomada:
 - **Centralizado**
 - **Isolado**
 - **Distribuído**
- Unidade 4**

Cortesia: Prof. Dr. José Marcos Câmara Brito

© Antônio M. Alberti 2007

Classificação dos Algoritmos de Roteamento

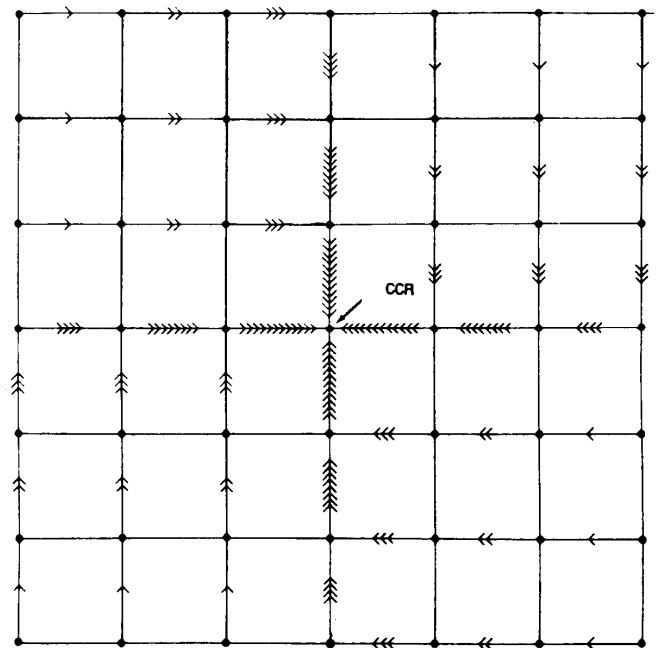
- O **número de caminhos** utilizados:
 - **Caminho Simples**
 - Um único trajeto deve ser utilizado entre dois pontos.
 - **Múltiplos Caminhos**
 - Mais de um trajeto pode ser utilizado entre dois pontos.

Roteamento Centralizado

- ✓ Um **único nó** é responsável pela **tomada de decisões de roteamento** na rede e pelo **encaminhamento** dessas decisões para os demais nós da rede.
- ✓ No roteamento centralizado é comum a utilização de um **Centro de Controle de Roteamento (CCR)**.
- ✓ Periodicamente, cada nó envia informações de estado para o CCR, como por exemplo, uma lista de seus vizinhos que estão em atividade.
- ✓ O CCR coleta todas estas informações e, conhecendo agora o comportamento global da rede, determina o melhor caminho para todas as comunicações possíveis.

Roteamento Centralizado

- ✓ Deste processamento resultam novas **tabelas de roteamento** que são distribuídas para todos os nós da rede.



Roteamento Centralizado

- ✓ **Vantagens:**
 - Como o CCR possui todas as informações da rede, o roteamento centralizado melhora tomada de decisões.
 - Como o CCR processa a tabela de roteamento e a distribui para os demais nós da rede, ocorre uma diminuição de processamento.
- ✓ **Desvantagens:**
 - Possui um tempo grande de adaptação às mudanças, de forma que uma nova tabela gerada pode já não refletir a melhor condição de roteamento.
 - Se o CCR falhar a rede toda pára.
 - Os diversos nós não recebem a nova tabela ao mesmo tempo, causando problemas de inconsistência no roteamento.
 - Alto tráfego de controle nas proximidades do CCR.

- Roteamento
- Unidade 4
- ## Roteamento Isolado
- ✓ Nesta técnica o nó toma a decisão de roteamento baseado **apenas** em **informação local**, sem trocar informações com os demais nós da rede.
 - ✓ Trata-se de uma técnica **descentralizada**.
 - ✓ Um algoritmo simples do tipo isolado adaptativo é o algoritmo conhecido como **Hot Potato** (**Batata Quente**).
 - ✓ Neste algoritmo quando um pacote de mensagem chega a um nó, ele procura se livrar do pacote **o mais rápido possível**, colocando-o na fila de saída que apresentar menor tamanho naquele momento.

- Roteamento
- Unidade 4
- ## Roteamento por Inundação
- ✓ Uma **forma extrema** de roteamento isolado é o chamado **roteamento por inundação**.
 - ✓ Neste tipo de roteamento, um pacote é **enviado** para **todas** as **portas de saída** de um roteador, com exceção da porta por onde o pacote foi recebido.
 - ✓ O roteamento por inundação gera uma **enorme quantidade de cópias** de um pacote na rede.
 - ✓ Para eliminar as cópias não mais necessárias, **mecanismos de enxugamento** devem ser utilizados.

- Roteamento
- Unidade 4
- ## Roteamento por Inundação
- ✓ Um **exemplo** de um mecanismo de enxugamento é um **contador de hops** inicializado com a distância entre fonte e destino ou com a distância máxima da rede, e decrementado em cada roteador.
 - ✓ As principais **aplicações** do roteamento por inundação são:
 - **Medição** de **características** da rede.
 - **Atualização simultânea** de **bancos de dados distribuídos**.
 - Transmissão de pacotes com o menor tempo possível.

- Roteamento
- Unidade 4
- ## Roteamento Distribuído
- ✓ Nesta classe de algoritmos de roteamento cada nó **troca**, **periodicamente**, **informações de roteamento** com outros nós da rede (por exemplo, com os nós vizinhos).
 - ✓ **Baseado** então em informações **locais** e informações provenientes **de outros nós** da rede, cada nó toma suas decisões de roteamento.

- Roteamento**
- Unidade 4**
- ## Roteamento pelo Caminho mais Curto
- ✓ Baseado em informações de custo de cada enlace.
 - ✓ Tentam encontrar o caminho de menor custo, ou seja um caminho que minimize a soma dos custos de cada enlace (custo total).
 - ✓ Parâmetros de custo possíveis:
 - Comprimento do enlace
 - Atraso estimado
 - Confiabilidade
 - Taxa de transmissão

- Roteamento**
- Unidade 4**
- ## Roteamento pelo Caminho mais Curto
- ✓ Dentre os principais algoritmos de roteamento pelo caminho mais curto estão:
 - Algoritmo de Dijkstra
 - Algoritmo de Bellman-Ford
 - ✓ Estes algoritmos e suas variações são amplamente utilizados em diversas redes de comunicações, incluindo a Internet baseada em TCP/IP.

Algoritmo de Dijkstra

- ✓ Foi desenvolvido por *Edsger Wybe Dijkstra* em 1959.



- ✓ **Variáveis:**

- $D(v)$ = distância do nó fonte (1) para o nó (v).
- $I(i,j)$ = custo entre o nó (i) e o nó (j).
- N = conjuntos de nós com distância definida.

Algoritmo de Dijkstra

- ✓ **Inicialização:**

- **Faça** $N = \{1\}$.
- Para cada nó (v) fora de N, **faça** $D(v) = I(1,v)$. Se (v) não está conectado a N, faça $D(v) = \text{infinito}$.

- ✓ **Passo Principal:**

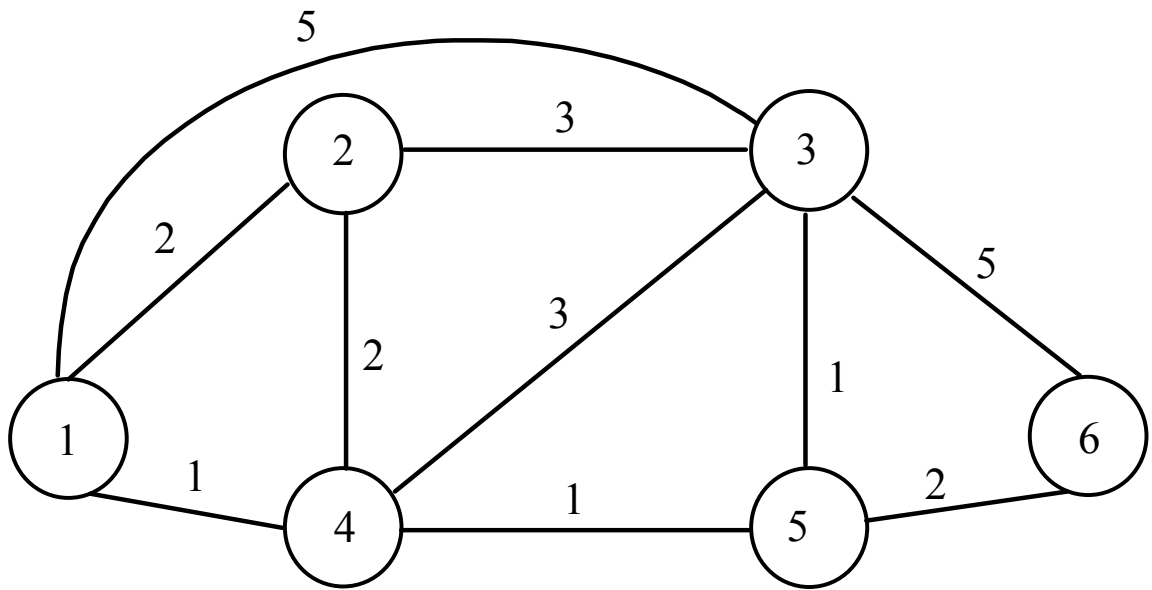
- **Encontre** um nó (w) fora de N tal que $D(w)$ seja mínimo.
- **Adicione** este nó (w) a N.
- **Atualize** $D(v)$ para todos os nós restantes que ainda não estão em N, fazendo:

$$D(v) = \text{Min} [D(v), D(w)+I(w,v)]$$

- O algoritmo pára quando todos os nós fizerem parte de N.

Algoritmo de Dijkstra

✓ Rede exemplo:

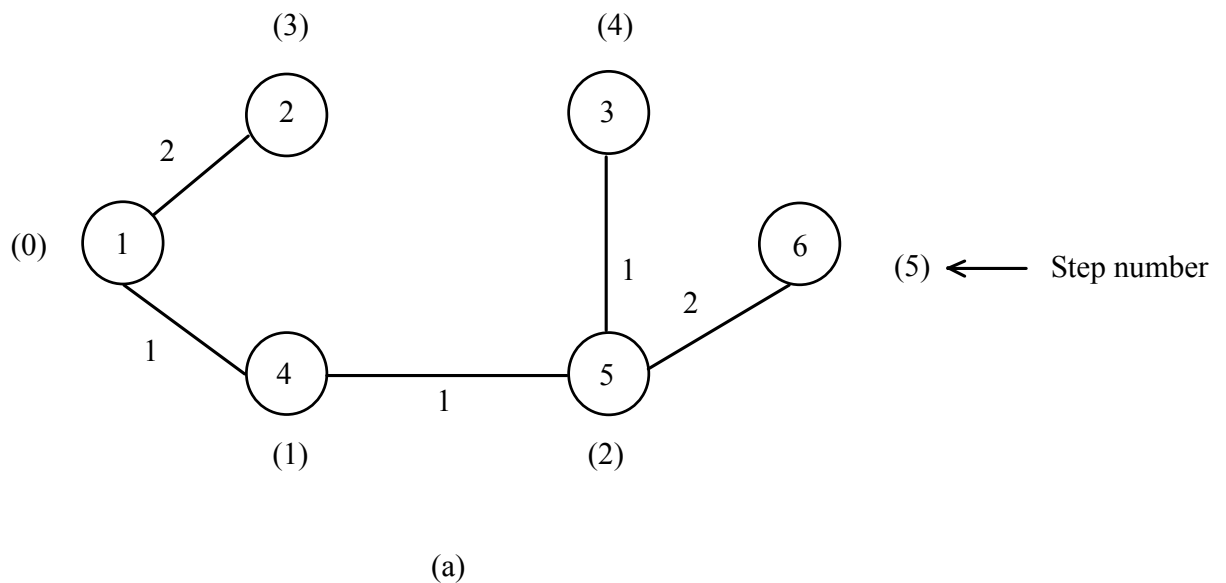


Cortesia: Prof. Dr. José Marcos Câmara Brito

© Antônio M. Alberti 2007

Algoritmo de Dijkstra

✓ Roteamento Resultante:



Cortesia: Prof. Dr. José Marcos Câmara Brito

© Antônio M. Alberti 2007

- Roteamento
- ## Roteamento por Vetor de Distância
- ✓ O algoritmo de **roteamento por vetor de distância** opera mantendo uma **tabela** (isto é, um **vetor**) que fornece:
 - A **melhor distância** conhecida até cada destino.
 - **Qual saída** deve ser tomada para atingir cada destino (**next hop**).
 - ✓ Cada nó faz uma **estimativa** do **custo** necessário para atingir os seus vizinhos.

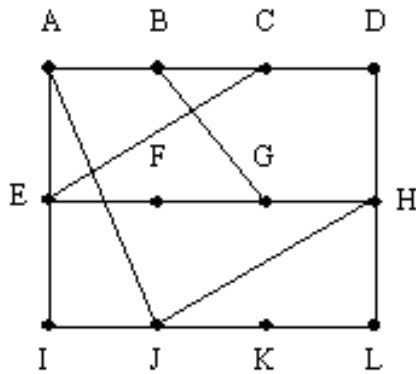
Unidade 4

- Roteamento
- ## Roteamento por Vetor de Distância
- ✓ Periodicamente os nós enviam estas estimativas para os seus vizinhos.
 - ✓ A **melhor distância** é **calculada** em função das **estimativas de custos** levantadas pelo próprio nó e recebidas dos seus vizinhos.

Unidade 4

Roteamento por Vetor de Distância

✓ Rede Exemplo:



Fonte: Tanenbaum

- Considere o nó **j**.
- As estimativas de atraso de **j** até os seus vizinhos são:

$J_A \text{ Delay} = 8$
 $J_I \text{ Delay} = 10$
 $J_H \text{ Delay} = 12$
 $J_K \text{ Delay} = 6$

Cortesia: Prof. Dr. José Marcos Câmara Brito

© Antônio M. Alberti 2007

Roteamento por Vetor de Distância

✓ Rede Exemplo:

- O nó **j** também possui os **vetores de distância** dos seus nós vizinhos:

	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9

Fonte: Tanenbaum

Cortesia: Prof. Dr. José Marcos Câmara Brito

© Antônio M. Alberti 2007

Roteamento por Vetor de Distância

✓ **Rede Exemplo:**

- De posse destas informações o nó *j* executa o algoritmo de *Bellman-Ford* para determinar qual é a **menor distância** e o **próximo hop** na direção de cada um dos nós da rede.

8	A
20	B
28	C
20	D
17	E
30	F
18	G
12	H
10	I
0	J
6	K
15	L

- Desta forma o nó *J* atualiza a sua tabela.

Fonte: Tanenbaum

Roteamento por Vetor de Distância

- ✓ Os nós **passam mensagens de controle** para outros nós levando os seus **vetores de distância** até que o algoritmo esteja completo em cada nó, com convergência para o caminho mais curto.
- ✓ Durante a **fase de convergência** o roteamento por vetor de distância é propenso a **formação de loops**.
- ✓ Outra desvantagem do algoritmo é que o **tempo de convergência** pode ser **longo** quando ocorre a perda de um roteador ou de um enlace.

Roteamento por Estado de Enlace

- ✓ Cada roteador que utiliza o algoritmo de **roteamento por estado de enlace** realiza as seguintes tarefas:
 - **Descobre** seus **vizinhos** e aprende os seus **endereços**.
 - **Faz estimativas de custos** para cada um dos seus vizinhos e **armazena** estas informações.
 - **Cria** um **pacote** para difundir estas informações para **todos os outros roteadores**.
 - **Calcula** o **caminho mais curto** para cada um dos outros roteadores utilizando o algoritmo de **Dijkstra**.

Exercícios: