

# *Modelo de Referência OSI*

Camada de Rede  
(Pág. 69 da Apostila)



## *Objetivos:*

- Assegurar o transporte de unidades de dados, denominados pacotes, do sistema fonte ao sistema destinatário.
  - A trajetória pode significar a passagem por diversos nós intermediários. Isto requer o conhecimento de todos os aspectos topológicos da rede, e esta informação, ser capaz de escolher o caminho a ser traçado.
- 
-

## *Funções:*

- Endereçamento
  - Roteamento
  - Controle de Congestionamento
    - OBS: Deve-se considerar o tráfego das mensagens, evitando-se a sobrecarga (congestionamento) de certos trechos de linhas de comunicação.
- 
-

# *Endereçamento:*

- Endereçamento Hierárquico:
  - O endereço é constituído conforme a posição de cada entidade na hierarquia da rede, sugerindo o local onde se encontra. Ex: MAN e WAN.
- Endereçamento Horizontal:
  - O endereço não tem relação com a localização da entidade na rede. Ex: LAN.

# ***Roteamento:***

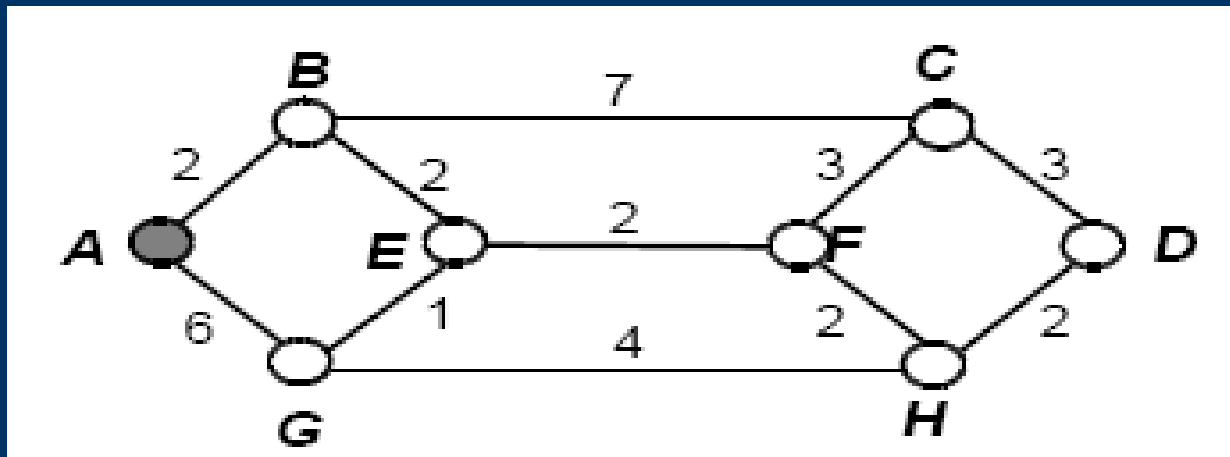
- Algoritmos Estáticos:
  - Simplesmente entregam a mensagem.
- Algoritmos Adaptativos:
  - Levam em conta a situação do tráfego da rede e a topologia utilizada.

# *Roteamento - Algoritmos Estáticos*

- Algoritmo do caminho mais curto:
  - Representa a sub-rede como um grafo. O objetivo é encontrar o caminho mais curto.
  - O conceito de caminho mais curto pode levar em consideração diferentes aspectos:
    - Número de nós
    - Distância geográfica entre pontos
    - Tempo de espera em cada nó da trajetória
    - etc...

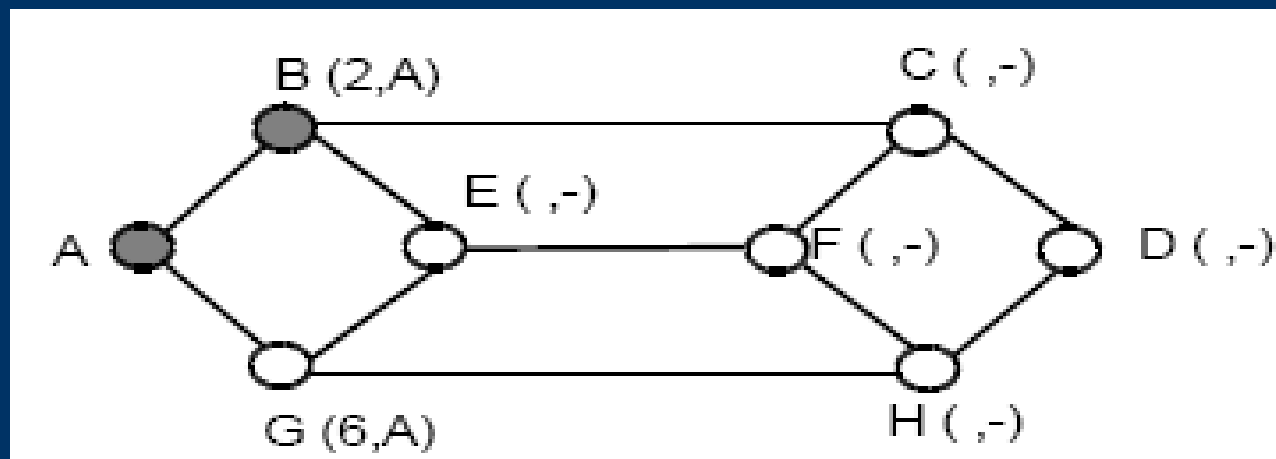
## *Ex: Algoritmo do caminho mais curto*

1. Nenhum caminho é conhecido e os nós são identificados como infinito.
2. O objetivo será encontrar o caminho mais curto entre A e D.
3. O nó A será marcado como referência.



## Ex: Algoritmo do caminho mais curto

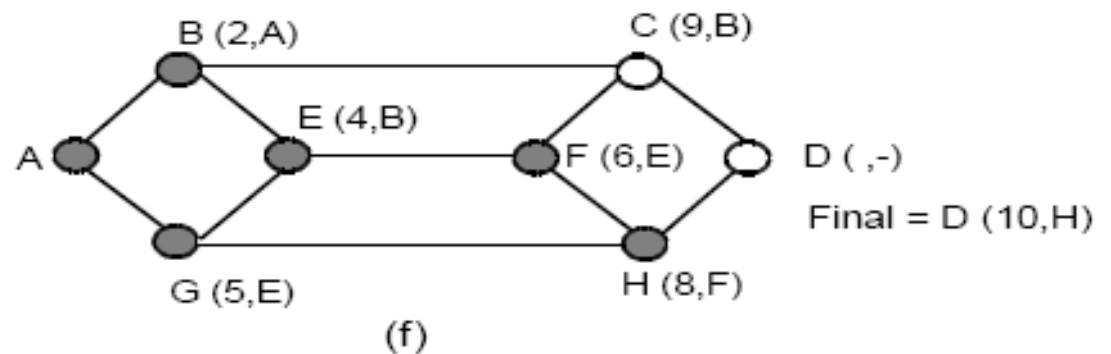
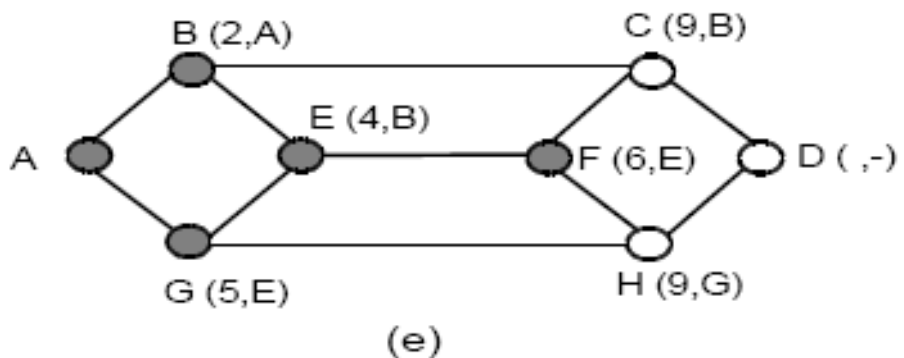
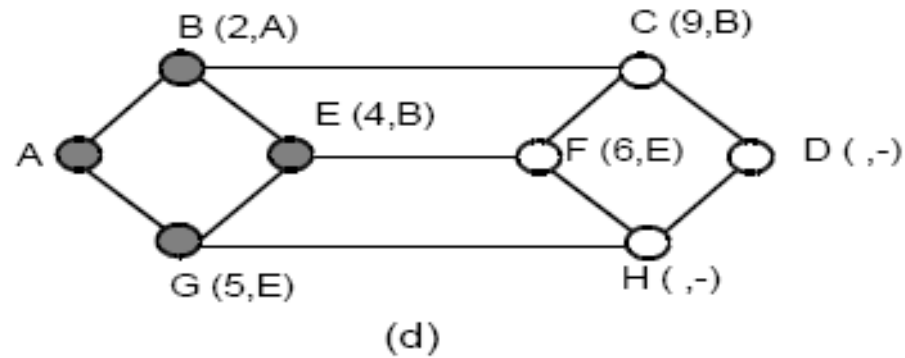
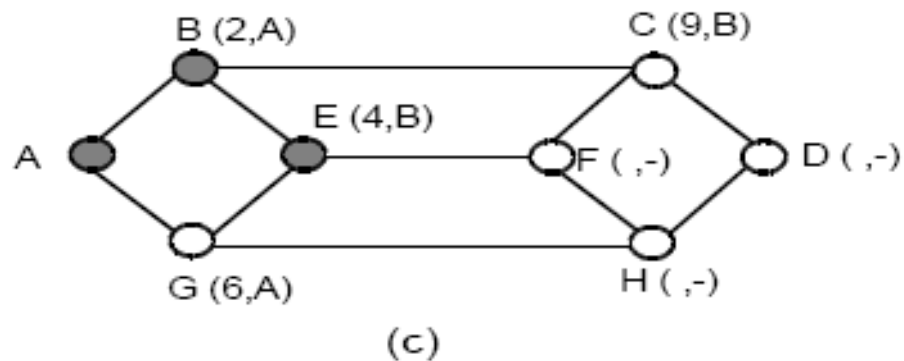
1. Os nós adjacentes a “A” serão analisados.
2. Estes serão etiquetados com a distância que os separa de “A”.
3. Ainda, será marcado o último nó a partir do qual foi realizado o cálculo.





## Ex: Algoritmo do caminho mais curto

A seguir é apresentado a progressão do algoritmo.



Resultado: A-B-E-F-H-D

# *Roteamento Multicaminhos*

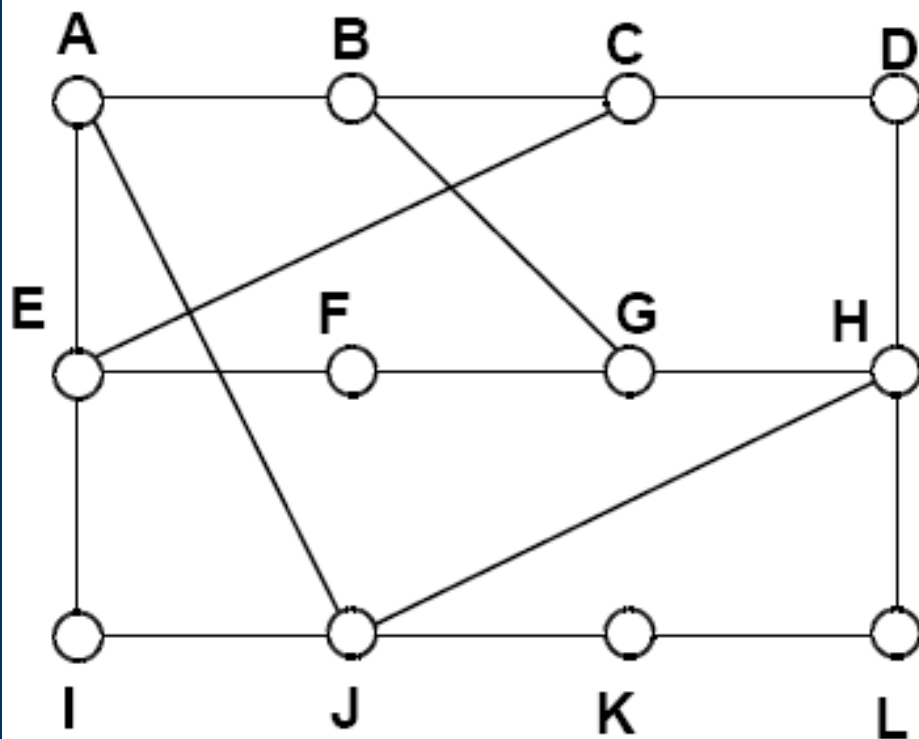
- Leva em conta a possibilidade de existência de diversos caminhos entre 2 nós.
  - Cada nó mantém atualizada uma tabela com uma linha para cada destino da rede.
  - Cada nó terá diferentes linhas de saída para um mesmo destino, classificadas em ordem decrescente, do melhor ao menos eficiente, com um peso relativo.
- 
-

# *Roteamento Multicaminhos*

- Antes do envio de um pacote, o nó transmissor gera um número aleatório para definir o caminho, utilizando os pesos como probabilidade.
- As tabelas são criadas de maneira estática pelo administrador do sistema e carregadas em cada unidade na inicialização da rede.

# Roteamento Multicaminhos

Exemplo de tabela para o nó J. Gera-se um número aleatório entre 0 e 0,99 para determinar o caminho.



A	A	0,63	I	0,21	H	0,16
B	A	0,46	H	0,31	I	0,23
C	A	0,34	I	0,33	H	0,33
D	H	0,50	A	0,25	I	0,25
E	A	0,40	I	0,40	H	0,20
F	A	0,34	H	0,33	I	0,33
G	H	0,46	A	0,31	K	0,23
H	H	0,63	K	0,21	A	0,16
I	I	0,65	A	0,22	H	0,13
.	.	.	.	.	.	.
K	K	0,67	H	0,22	A	0,11
L	K	0,42	H	0,42	A	0,16

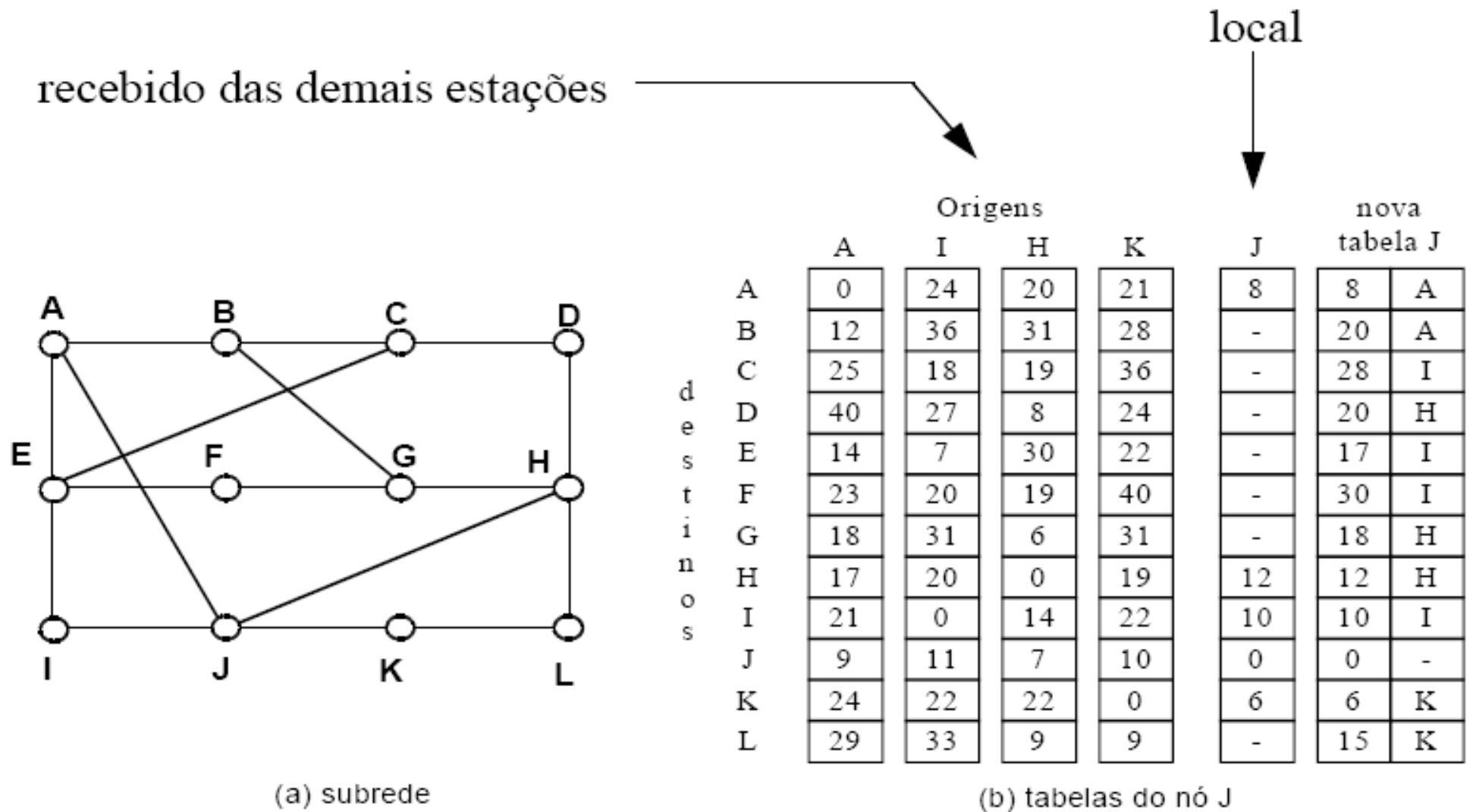
# *Roteamento Multicaminhos*

- Vantagens sobre o algoritmo do menor caminho:
  - Definir diferentes classes de tráfego
  - Confiabilidade. Várias linhas podem ser perdidas sem que se perca sua conectividade.

# *Roteamento Dinâmico Distribuído*

- As estações trocam informações de roteamento com seus vizinhos imediatos, contendo custos de transmissão a partir dela (gera-se uma tabela).
  - Esta tabela contém, para cada destino possível, o nó preferencial de saída e o custo estimado de transmissão para este nó.
  - Para decidir a rota, a estação emissora soma o custo de transmissão até o vizinho imediato com o custo estimado dali até o destino final.
- 
-

# Roteamento Dinâmico Distribuído



# *Roteamento Dinâmico Distribuído*

A seguir temos um exemplo de rotina para varredura das tabelas de roteamento. Consideremos  $i$  = linhas,  $j$  = colunas e ainda uma matriz com 12 roteadores:

```
int i, j, x;
int matriz[12][12]; // esta matriz contem as tabelas recebidas
for (i=0; i<=11; i++) {
    for (j=0; j<=11; j++) {
        x = matriz[i][j] + matriz[j][9];
        if (x < matriz[i][9])
            matriz[i][9] = x;
    }
} // onde 9 = coluna da matriz que se deseja calcular (J)
```

---

---



# *Histórico dos Algoritmos de Roteamento na Internet*

- Um algoritmo baseado neste princípio dinâmico distribuído, denominado RIP (*Routing Information Protocol*) foi implementado originalmente na ARPANET e posteriormente na Internet (parte do protocolo IP até 1990).
  - Após 1990 foi substituído por um protocolo denominado OSPF (*Open shortest Path First*).
- 
-

# *Controle de Congestionamento*

- Pré-alocação de Buffers
  - Destruição de pacotes
  - Controle de fluxo
  - Controle Isarítmico
  - Pacotes de Estrangulamento
- 
-

# *Exemplos de tempo de resposta*

- Resposta rápida (interno):
  - ping [www.pb.cefetpr.br](http://www.pb.cefetpr.br)
- Resposta média (dentro do país):
  - ping [terra.com.br](http://terra.com.br)
- Resposta lenta (exterior):
  - ping [www.msi.com.tw](http://www.msi.com.tw)