

# Redes de Computadores II

# Na aula anterior

- Desenvolvimento do trabalho

# Na aula de hoje

- Revisão sobre rotas estáticas
- Introdução às rotas dinâmicas

# Relembrando: determinação de rotas

- Entrega direta: ocorre quando o roteador está diretamente conectado na rede de destino
- Entrega indireta: o roteador não está diretamente conectado
  - Rotas estáticas (manuais)
  - Rotas dinâmicas

# Configuração de uma rede c/ rotas estáticas

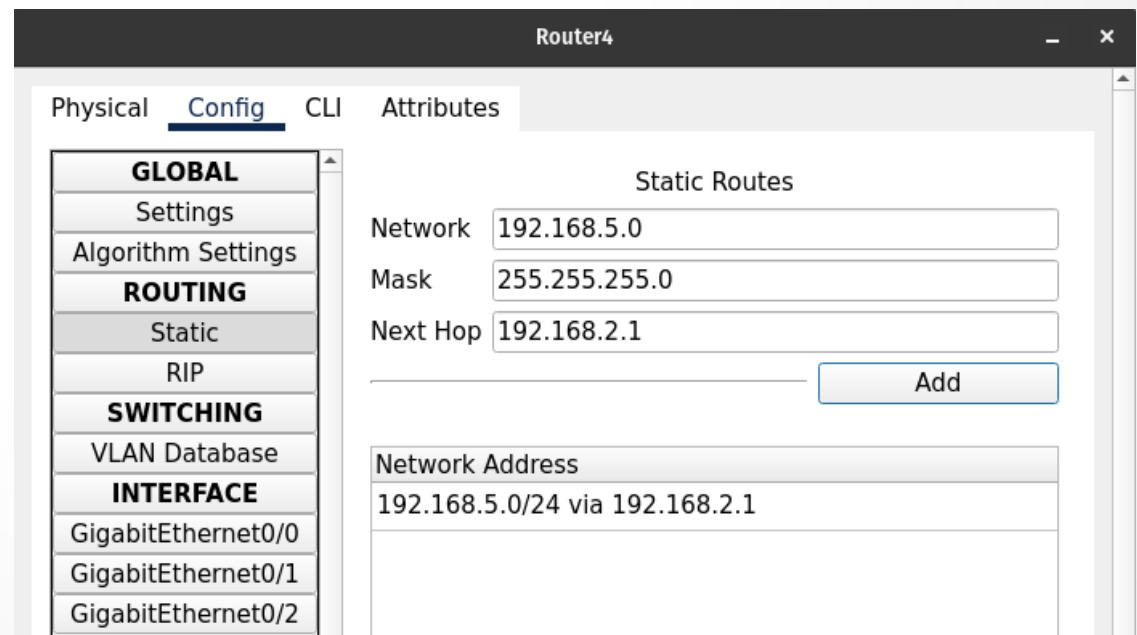
- Definir quais serão as sub-redes, IPs e registrar. Usar o recurso notas
- Nos hosts
  - Configurar os IP's dos hosts
  - Configurar o default gateway de cada host (observar qual o IP do roteador)
- Nos roteadores
  - Ativar a interface (desligada por padrão)
  - Configurar os endereços IP's (cuidado p/ não confundir as interfaces)

# Configuração de uma rede c/ rotas estáticas

- Ainda nos roteadores: configurar as rotas para destinos não conectados diretamente
- A configuração pode ser feita via CLI com o comando `ip route`

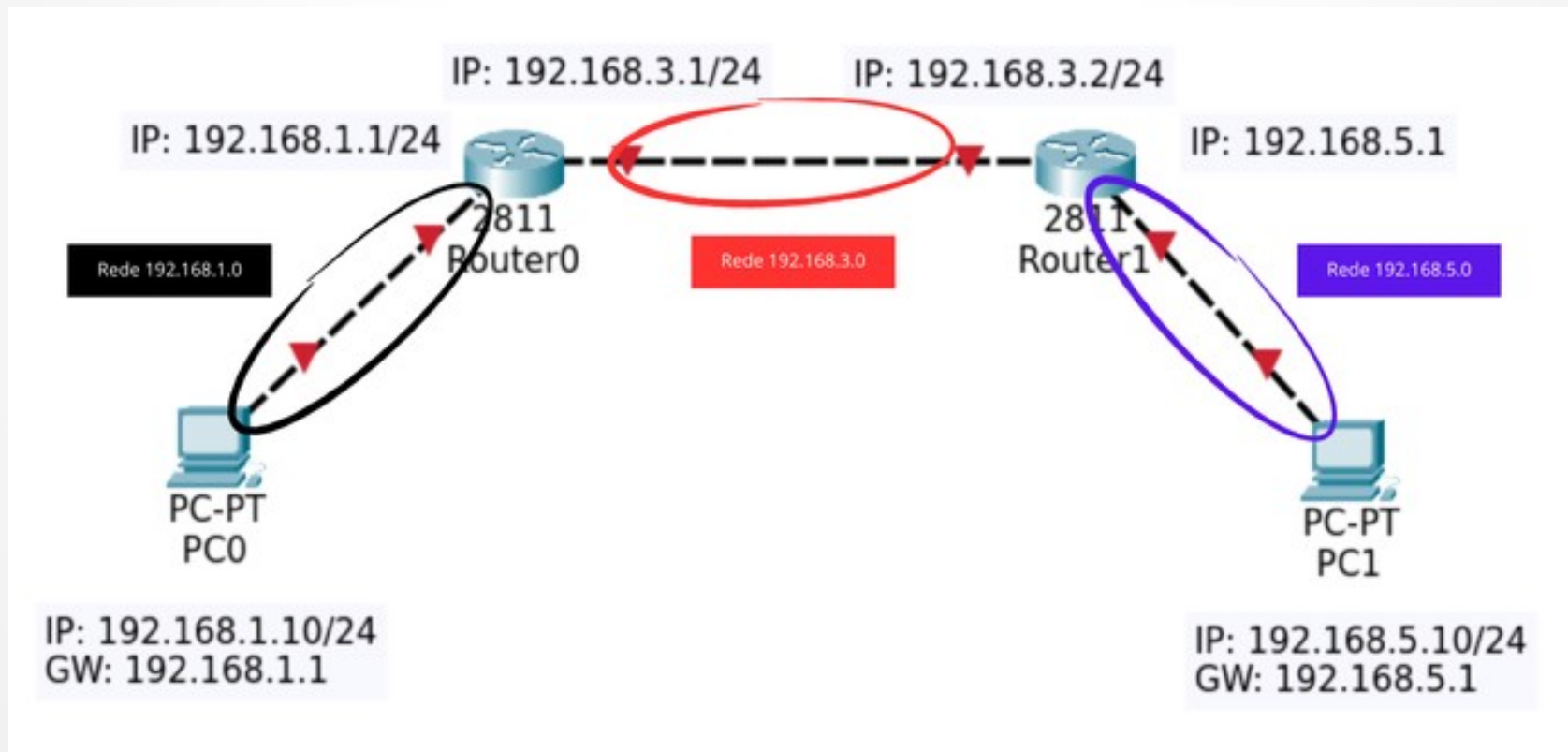
Ex: `ip route 192.168.5.0 255.255.255.0 192.168.3.2`

- Ou via interface gráfica



# Exemplo

- No exemplo o Router0 não conhece nada, a priori, sobre a rede 192.168.5.0
- Igualmente, o Router1 não sabe nada sobre a rede 192.168.1.0



# Testes

- “Unreachable” → Inalcançável
  - Possível causa: A rota de ida não está configurada
- “Timed out” → Sem resposta
  - Possível causa: A rota no sentido do retorno não está configurada



# Rotas dinâmicas

- Existem duas classes de algoritmos que utilizam rotas dinâmicas
  - Algoritmos de vetor de distância
  - Algoritmos de estado de enlace

Vetor de distância

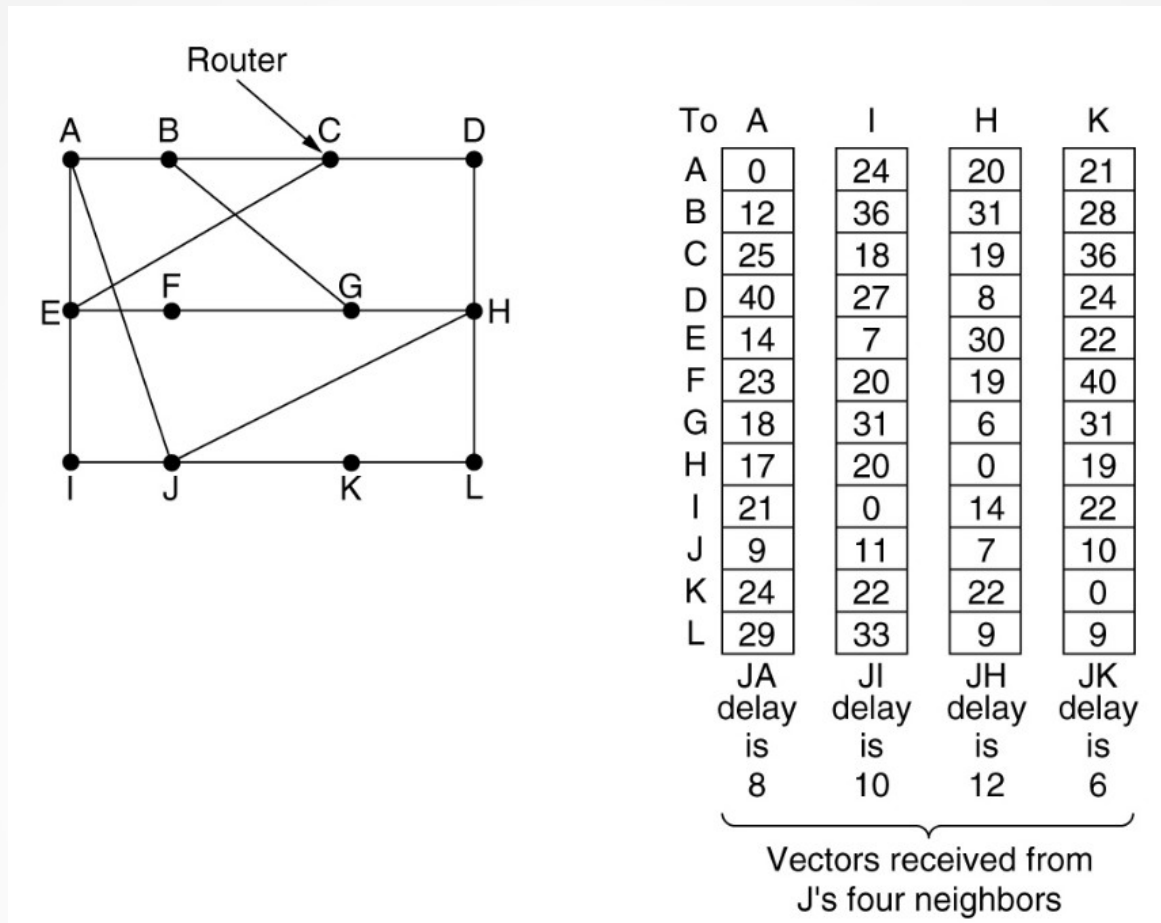
# Algoritmo de vetor de distância

- A proposta desse tipo de algoritmo é manter uma tabela das “distâncias” entre roteadores e periodicamente trocar informações das tabelas com os vizinhos
- Essa distância pode ser a quantidade de hops, o delay entre roteadores, ou alguma medida do congestionamento entre eles

# Algoritmo de vetor de distância

- Era o tipo de algoritmo usado originalmente na ARPANET e depois no começo da Internet
- Esse algoritmo tem o nome de RIP

# Algoritmo de vetor de distância



Fonte: Tanenbaum

J recebe as tabelas dos seus vizinhos (A, I, H e K) e a partir da distância mais curta monta a própria tabela. Por exemplo para chegar até G a menor distância seria 18 (6 de H-G + 12 de J-H)

- Esse algoritmo tem dois problemas
  - Demora na convergência
  - Más notícias demoram muito para serem propagadas:
    - O problema da contagem até o infinito: Ao receber uma tabela, o roteador não tem como saber que o caminho passava por ele mesmo
    - Então os roteadores ficam aumentando lentamente a sua distância para nós não conectados diretamente

- Existe uma variação do vetor de distância que é chamado de vetor de caminho. Ao invés de mandar a distância cada roteador manda o caminho completo até outros nós
  - Minimiza o problema da contagem até o infinito
- Principal algoritmo desse tipo é o BGP
- É o algoritmo usado na Internet atualmente



Estado de enlace

# Algoritmos de estado de enlace

- Idéia geral: Cada roteador manda não a distância dos seus vizinhos mas sim um “mapa” dos seus vizinhos
- Exemplo: Algoritmo OSPF
- Passos:
  - Cada roteador elenca seus vizinhos e descobre a “distância” para cada um deles
  - Roteador manda essa informação em modo broadcast para toda a rede
  - Ao receber os pacotes dos demais roteadores cada um monta sua própria topologia completa
  - Para determinar a rota final é usado um algoritmo de menor distância

# Algoritmos de estado de enlace

- Idéia geral: Cada roteador manda não a distância dos seus vizinhos mas sim um “mapa” dos seus vizinhos
- Passos:
  - Cada roteador elenca seus vizinhos e descobre a “distância” para cada um deles
  - Roteador manda essa informação em modo broadcast para toda a rede
  - Ao receber os pacotes dos demais roteadores cada um monta sua própria topologia completa
  - Para determinar a rota final é usado um algoritmo de menor distância

# Algoritmo de menor distância

- O algoritmo mais usado nesse problema é o algoritmo de Dijkstra
- Para o algoritmo a seguir considere:
  - $G$  é o conjunto de vértices total
  - $d$  o vetor que armazena a distância até um destino
  - $p$  é o vetor que armazena o “parente” através do qual se chega em um destino
  - $Q$  é o conjunto de vértices não visitados

# Algoritmo de menor distância (Dijkstra)

para todo  $v \in G$

$d[v] \leftarrow \infty$

$p[v] \leftarrow -1$

enquanto  $Q \neq \emptyset$

$u \leftarrow \text{mín}(Q)$

$Q \leftarrow Q - \{u\}$

para cada  $v$  adjacente a  $u$

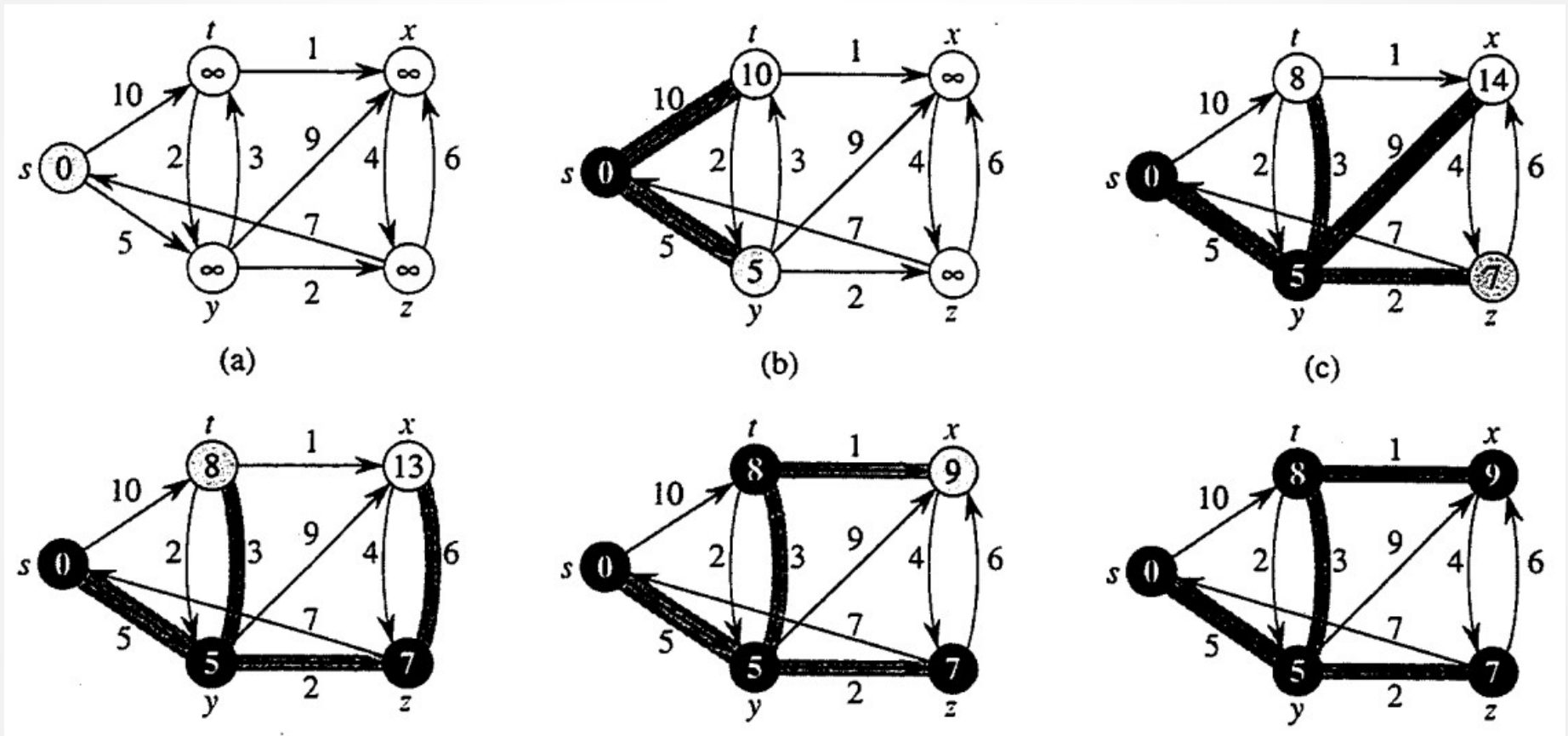
se  $d[u] + \text{peso}(u, v) < d[v]$       //relaxe ( $u, v$ )

$d[v] \leftarrow d[u] + \text{peso}(u, v)$

$p[v] \leftarrow u$

# Algoritmo de menor distância (Dijkstra)

- Exemplo de execução do Dijkstra (fonte: Cormen)



# Contexto

- Protocolos podem ser classificados em dois contextos
  - Uso interno (dentro de organizações)
  - Uso externo (fora de organização, ou seja, na Internet)
- Às vezes são usados os termos Interior Gateway Protocol (IGP) e Exterior Gateway Protocol (EGP)

# Resumo sobre algoritmos de roteamento

Algoritmo	Dono	Tipo	Contexto
RIP	Público	Vetor de distância	Interno
OSPF	Público	Estado de enlace	Interno
IGRP	Cisco	Vetor de distância	Interno
EIGRP	Cisco	Vetor de distância	Interno
BGP	Público	Vetor de caminho	Externo

Fonte: Peres et al



# Referências

- Algoritmo de Dijkstra. Disponível em:  
[https://pt.wikipedia.org/wiki/Algoritmo\\_de\\_Dijkstra](https://pt.wikipedia.org/wiki/Algoritmo_de_Dijkstra)
- CORMEN, Thomas H. et al. Algoritmos: teoria e prática. Rio de Janeiro: Campus, 2002. 916 p.
- PERES, André; LOUREIRO, César Augusto Hass; SCHMITT, Marcelo Augusto Rauh. Redes de Computadores II: Níveis de Transporte e Rede. 2014.
- TANENBAUM, Andrew S. Redes de Computadores. 4ª. Edição. Editora Campus, 2003.