

# Protocolos em Redes de Dados

## Aula 03

### Introdução ao encaminhamento

©Luís Rodrigues

FCUL

2005-2006

# Sumário

- ▶ Introdução ao encaminhamento.
- ▶ Distance Vector Routing.
- ▶ Link State Routing.
- ▶ RIP.
- ▶ (Nota: a maioria das figuras foram retiradas de A. Tanenbaum ©Prentice-Hall 1996)

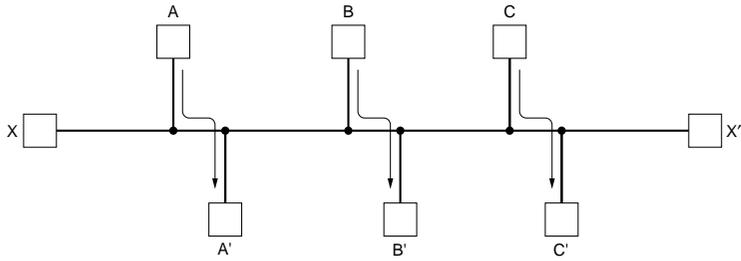
# Algoritmo de encaminhamento

- ▶ Decide qual o caminho que os pacotes devem tomar.
- ▶ Serviços datagrama:
  - ▶ Executado pacote a pacote.
- ▶ Serviços orientados ao circuito:
  - ▶ Executado quando se cria o circuito.

# Dificuldades do encaminhamento

- ▶ Tentar maximizar os recursos da rede (desempenho).
- ▶ Tentar garantir o acesso a todos os participantes (igualdade).
  - ▶ Objectivos em conflito.
- ▶ Que tipo de característica se deve privilegiar?
  - ▶ Débito, latência, preço, fiabilidade, etc.

# Igualdade e desempenho



## Encaminhamento

Encaminhamento estático

Encaminhamento dinâmico

Vectores de Distância  
Estado dos Elos  
Encaminhamento Hierárquico

RIP

RIP V1  
RIP V2

# Árvore de escoamento

## Encaminhamento

Encaminhamento estático

Encaminhamento dinâmico

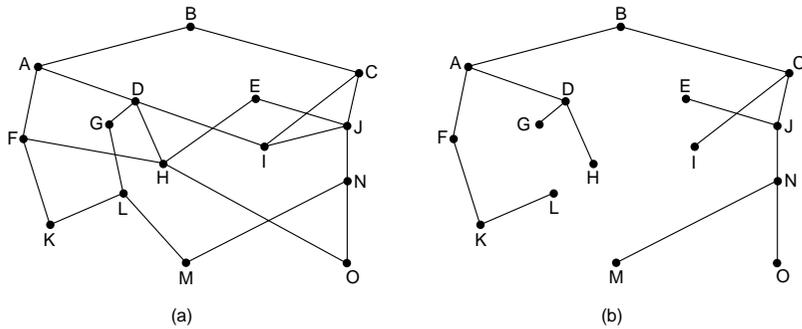
Vectores de Distância  
Estado dos Elos  
Encaminhamento Hierárquico

RIP

RIP V1  
RIP V2

- ▶ O conjunto de caminhos óptimos entre um nó e os restantes forma uma árvore.
- ▶ Designa-se por árvore de escoamento.
- ▶ Dificuldade na definição da árvore:
  - ▶ Os nós podem ter opiniões diferentes acerca de quais os encaminhadores que estão a montante e quais os que estão jusante.

# Árvore de escoamento



## Encaminhamento

Encaminhamento estático

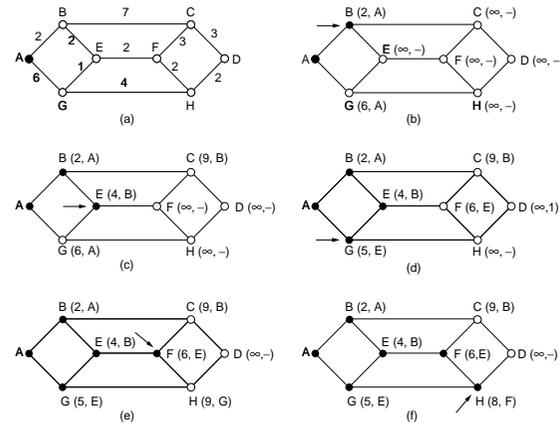
Encaminhamento dinâmico

Vectores de Distância  
Estado dos Elos  
Encaminhamento Hierárquico

RIP

RIP V1  
RIP V2

# Cálculo do caminho mais curto



## Encaminhamento

Encaminhamento estático

Encaminhamento dinâmico

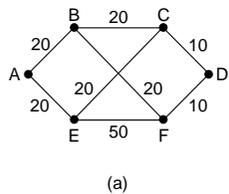
Vectores de Distância  
Estado dos Elos  
Encaminhamento Hierárquico

RIP

RIP V1  
RIP V2

- ▶ Enviar um pacote por todos os caminhos possíveis.
- ▶ Dispendioso e induz ciclos infinitos.
  - ▶ É necessário acrescentar um mecanismo de descarte de pacotes.
  - ▶ “Tempo-de-vida” do pacote.

## Matriz de encaminhamento



| Source | Destination |          |          |           |          |          |
|--------|-------------|----------|----------|-----------|----------|----------|
|        | A           | B        | C        | D         | E        | F        |
| A      |             | 9<br>AB  | 4<br>ABC | 1<br>ABFD | 7<br>AE  | 4<br>AEF |
| B      | 9<br>BA     |          | 8<br>BC  | 3<br>BFD  | 2<br>BFE | 4<br>BF  |
| C      | 4<br>CBA    | 8<br>CB  |          | 3<br>CD   | 3<br>CE  | 2<br>CEF |
| D      | 1<br>DFBA   | 3<br>DFB | 3<br>DC  |           | 3<br>DCE | 4<br>DF  |
| E      | 7<br>EA     | 2<br>EFB | 3<br>EC  | 3<br>ECD  |          | 5<br>EF  |
| F      | 4<br>FEA    | 4<br>FB  | 2<br>FEC | 4<br>FD   | 5<br>FE  |          |

(b)

### Encaminhamento

Encaminhamento estático

Encaminhamento dinâmico

Vectores de Distância  
Estado dos Elos  
Encaminhamento Hierárquico

RIP

RIP V1  
RIP V2

# Encaminhamento baseado no fluxo

### Encaminhamento

Encaminhamento estático

Encaminhamento dinâmico

Vectores de Distância  
Estado dos Elos  
Encaminhamento Hierárquico

RIP

RIP V1  
RIP V2

- ▶ Suponha-se que se conhecem de antemão os parâmetros da rede:
  - ▶ Padrões de tráfego, topologia, débito dos elos.
- ▶ É possível prever o atraso em cada caminho, para uma determinada configuração de encaminhamento:
  - ▶ Tamanho médio dos pacotes ( $1/\mu$ ), capacidade ( $C$ ) e taxa de transmissão ( $\lambda$ ).
  - ▶  $T = (\mu C - \lambda)^{-1}$
- ▶ Pode-se calcular o encaminhamento que minimiza o atraso médio na rede.

## Cálculo do atraso médio

- ▶ Considerando  $1/\mu = 800\text{bits}$ ,  $\mu C$  indica a capacidade do elo em pacotes por segundo e a variável “weight” indica a percentagem de pacotes a percorrer o elo em relação ao número total de pacotes trocados.

| i | Line | $\lambda_i$ (pkts/sec) | $C_i$ (kbps) | $\mu C_i$ (pkts/sec) | $T_i$ (msec) | Weight |
|---|------|------------------------|--------------|----------------------|--------------|--------|
| 1 | AB   | 14                     | 20           | 25                   | 91           | 0.171  |
| 2 | BC   | 12                     | 20           | 25                   | 77           | 0.146  |
| 3 | CD   | 6                      | 10           | 12.5                 | 154          | 0.073  |
| 4 | AE   | 11                     | 20           | 25                   | 71           | 0.134  |
| 5 | EF   | 13                     | 50           | 62.5                 | 20           | 0.159  |
| 6 | FD   | 8                      | 10           | 12.5                 | 222          | 0.098  |
| 7 | BF   | 10                     | 20           | 25                   | 67           | 0.122  |
| 8 | EC   | 8                      | 20           | 25                   | 59           | 0.098  |

### Encaminhamento

Encaminhamento estático

Encaminhamento dinâmico

Vectores de Distância  
Estado dos Elos  
Encaminhamento Hierárquico

RIP

RIP V1  
RIP V2

### Encaminhamento

Encaminhamento estático

Encaminhamento dinâmico

Vectores de Distância  
Estado dos Elos  
Encaminhamento Hierárquico

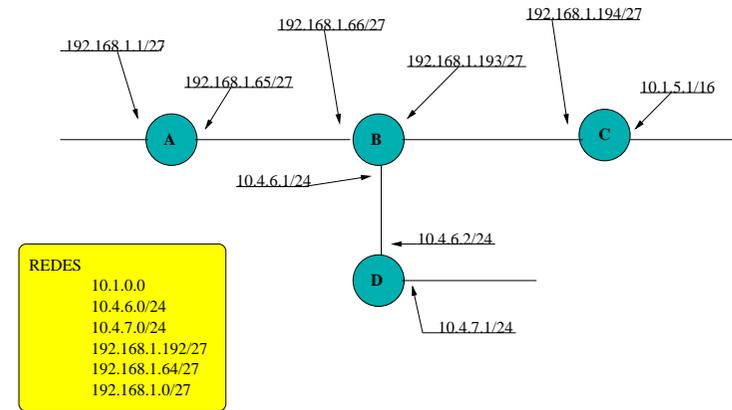
RIP

RIP V1  
RIP V2

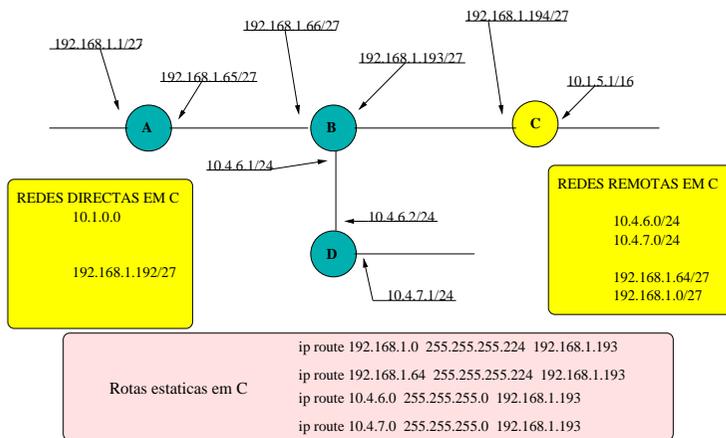
# Encaminhamento estático

- ▶ Cada encaminhador é configurado estaticamente com as rotas correctas.
  - ▶ Qualquer reconfiguração da rede obriga à reconfiguração dos encaminhadores.
  - ▶ A possibilidade de adaptar dinamicamente o comportamento da rede em função de falhas é bastante limitada.

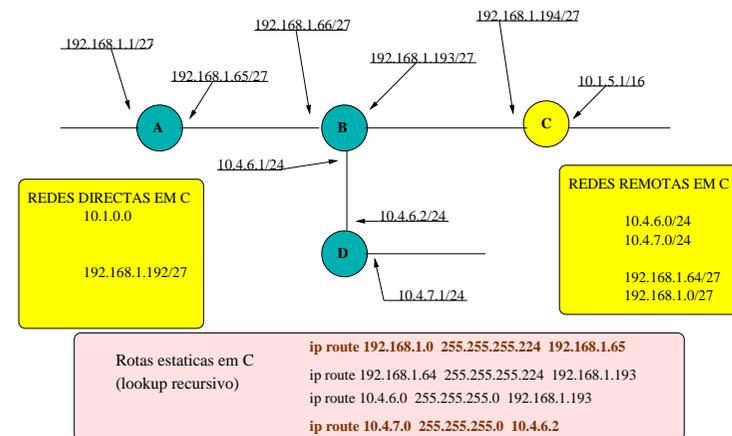
# Rede alvo



# Encaminhamento estático



# Lookup recursivo



# Lookup recursivo

- ▶ Obriga a um maior processamento
  - ▶ Embora o encaminhador possa fazer manter as traduções em cache.
- ▶ Pode simplificar a reconfiguração.

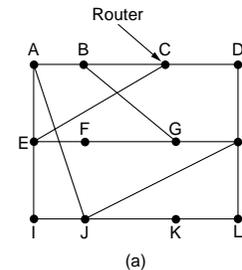
# Encaminhamento dinâmico

- ▶ Duas alternativas principais:
  - ▶ Vectores de distância.
  - ▶ Estados dos elos.

# Encaminhamento baseado na distância

- ▶ Distance Vector Routing.
  - ▶ Cada encaminhador mantém uma tabela que indica a “distância” em relação aos restantes nós e qual o caminho a seguir.
- ▶ A métrica usada para medir a distância pode variar.
  - ▶ Número de encaminhadores, tempo de espera, etc.
- ▶ Cada nó propaga para os seus vizinhos o conteúdo da sua tabela.
  - ▶ A tabela local é actualizada após receber informação dos vizinhos.

# Actualização da informação de custos



| To | A  | I  | H  | K  | New estimated delay from J |
|----|----|----|----|----|----------------------------|
| A  | 0  | 24 | 20 | 21 | 8 A                        |
| B  | 12 | 36 | 31 | 28 | 20 A                       |
| C  | 25 | 18 | 19 | 36 | 28 I                       |
| D  | 40 | 27 | 8  | 24 | 20 H                       |
| E  | 14 | 7  | 30 | 22 | 17 I                       |
| F  | 23 | 20 | 19 | 40 | 30 I                       |
| G  | 18 | 31 | 6  | 31 | 18 H                       |
| H  | 17 | 20 | 0  | 19 | 12 H                       |
| I  | 21 | 0  | 14 | 22 | 10 I                       |
| J  | 9  | 11 | 7  | 10 | 0 -                        |
| K  | 24 | 22 | 22 | 0  | 6 K                        |
| L  | 29 | 33 | 9  | 9  | 15 K                       |

JA delay is 8    JI delay is 10    JH delay is 12    JK delay is 6  
 Vectors received from J' s four neighbors  
 New routing table for J

# Lentidão na actualização de falhas

| A        | B        | C        | D        | E        |                   |
|----------|----------|----------|----------|----------|-------------------|
| $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ | Initially         |
| 1        | $\infty$ | $\infty$ | $\infty$ | $\infty$ | After 1 exchange  |
| 1        | 2        | $\infty$ | $\infty$ | $\infty$ | After 2 exchanges |
| 1        | 2        | 3        | $\infty$ | $\infty$ | After 3 exchanges |
| 1        | 2        | 3        | 4        | $\infty$ | After 4 exchanges |

(a)

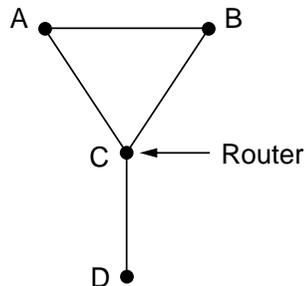
| A        | B        | C        | D        | E        |                   |
|----------|----------|----------|----------|----------|-------------------|
| 1        | 2        | 3        | 4        | 4        | Initially         |
| 3        | 2        | 3        | 4        | 4        | After 1 exchange  |
| 3        | 4        | 3        | 4        | 4        | After 2 exchanges |
| 5        | 4        | 5        | 4        | 4        | After 3 exchanges |
| 5        | 6        | 5        | 6        | 6        | After 4 exchanges |
| 7        | 6        | 7        | 6        | 6        | After 5 exchanges |
| 7        | 8        | 7        | 8        | 8        | After 6 exchanges |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |                   |
| $\infty$ | $\infty$ | $\infty$ | $\infty$ | $\infty$ |                   |

(b)

# Contagem até ao infinito e "split horizon"

- ▶ O problema anterior é designado por problema da contagem até ao infinito.
- ▶ Uma técnica que tenta resolver este problema consiste em dar informação acerca da direcção dos pacotes.
  - ▶ Se A encaminha pacotes para C através de B, não indica a B que possui um caminho alternativo para C.
  - ▶ Põe essa distância a "infinito".
  - ▶ Designa-se por "split horizon".

# Falha do algoritmo "split horizon".



# Encaminhamento baseado no estado dos elos

- ▶ Cada nó descobre quais os encaminhadores que estão na sua vizinhança e qual o estado dos elos.
- ▶ Esta informação é propagada na rede para todos os encaminhadores.
- ▶ Cada encaminhador calcula uma tabela de encaminhamento localmente.

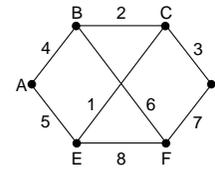
# Reconhecimento dos vizinhos

- ▶ Pacotes dedicados para saber a identidade dos vizinhos (hello).
- ▶ Pacotes de eco para medir os atrasos na rede:
  - ▶ Podem ou não ter em conta o tráfego (usando as filas normais para transmitir o eco).
- ▶ A informação recolhida é enviada aos restantes nós.

# Distribuindo o estado das linhas

- ▶ Tenta-se que todos os encaminhadores usem a mesma informação.
- ▶ Os pacotes são numerados e encaminhados por inundação.
- ▶ Pacotes já encaminhados/recebidos são descartados.
- ▶ Cada entrada é apagada ao fim de algum tempo.
- ▶ A recepção dos pacotes é confirmada ao receptor.

# Pacotes do "link-state"



(a)

| Link |      | State |      | Packets |      |
|------|------|-------|------|---------|------|
| A    | B    | C     | D    | E       | F    |
| Seq. | Seq. | Seq.  | Seq. | Seq.    | Seq. |
| Age  | Age  | Age   | Age  | Age     | Age  |
| B 4  | A 4  | B 2   | C 3  | A 5     | B 6  |
| E 5  | C 2  | D 3   | F 7  | C 1     | D 7  |
|      | F 6  | E 1   |      | F 8     | E 8  |

(b)

# Cálculo de novas tabelas de encaminhamento

- ▶ Feito localmente usando o estado da rede recebido.
- ▶ Algoritmos semelhantes aos estudados anteriormente.
- ▶ Variantes deste algoritmo muito utilizadas em redes actuais.
  - ▶ OSPF.

# Encaminhamento hierárquico

- ▶ Não é exequível manter tabelas com todos os encaminhadores que existem no mundo.
- ▶ Dividir o espaço de encaminhamento em regiões.
- ▶ Encaminhar entre as regiões sem conhecer a sua estrutura interna.
  - ▶ Aplicar este princípio recursivamente.

# Duas grandes classes de protocolos

- ▶ Interior Routing Protocols:
  - ▶ Usados para encaminhar pacotes dentro de um sistema autónomo.
- ▶ Exterior Routing Protocols:
  - ▶ Usados para encaminhar pacotes entre sistemas autónomos.

# Sistemas autónomos

- ▶ A rede é vista como um conjunto de sistemas autónomos interligados.
- ▶ A estrutura interna do sistema autónomo não é vista do exterior.
- ▶ Protocolos diferentes dentro e fora do sistema autónomo.

# Algumas siglas (Protocolos interiores)

- ▶ RIP:
  - ▶ Routing Information Protocol.
- ▶ OSPF:
  - ▶ Open Shortest Path First.
- ▶ IGRP, EIGRP:
  - ▶ Proprietário da Cisco.

# Algumas siglas (Protocolos exteriores)

- ▶ EGP:
  - ▶ Exterior Gateway Protocol.
- ▶ BGP:
  - ▶ Border Gateway Protocol.
- ▶ CIDR:
  - ▶ Classless Inter-Domain Routing.

# RIP (V1)

- ▶ Define o formato das mensagens usadas para trocar o conteúdo das tabelas.
- ▶ A métrica usada é o número de "hops", e o infinito está definido como 16.
- ▶ Esta convenção para o infinito impede o uso de métricas sofisticadas.

# RIP

- ▶ Protocolo "simples" da família dos protocolos baseados em vectores de distância.
- ▶ Inspirado nos primeiros protocolos de encaminhamento usados na Internet.

# RIP (V1)

- ▶ Originalmente, concretizado por um programa chamado "routed".
- ▶ Distribuído com o Unix de Berkeley.
- ▶ Hoje substituído por outro chamado "gated".
  - ▶ "gated" disponível para a maioria dos Unix.

# RIP (V1)

- ▶ Configuração:
  - ▶ O servidor deve obter a lista das suas interfaces e quais as redes a que está ligado através de ficheiros de configuração.
- ▶ É necessário também indicar quais são os vizinhos e quais os seus endereços.

# RIP (V1): envio de actualizações

- ▶ Actualizações (response) periódicas.
  - ▶ Em intervalos de 30 s.
- ▶ Estimuladas por alterações na rede:
  - ▶ Propagadas de imediato mas com controlo da taxa máxima.
- ▶ Pedidos:
  - ▶ Utilizados quando um encaminhador se inicia, para solicitar actualizações dos seus vizinhos.
  - ▶ É possível solicitar uma actualização completa ou pedir só o refrescamento de uma rota concreta.

# RIP (V1): formato do pacote

|                           |         |   |
|---------------------------|---------|---|
| Comand                    | Version | 0 |
| Address family identifier |         | 0 |
| IP address                |         |   |
| 0                         |         |   |
| 0                         |         |   |
| Metric                    |         |   |

- ▶ Command: request/response.
- ▶ Version: 1
- ▶ Family: originalmente pensado para suportar XNS, X.25, etc. Na prática IP.

# RIP (V2)

- ▶ Uma segunda versão do RIP foi definida em 1993.
- ▶ Permite trocar máscaras de sub-rede, de modo a criar encaminhamentos diferentes para cada uma das sub-redes.
- ▶ Permite trocar um campo de autenticação.

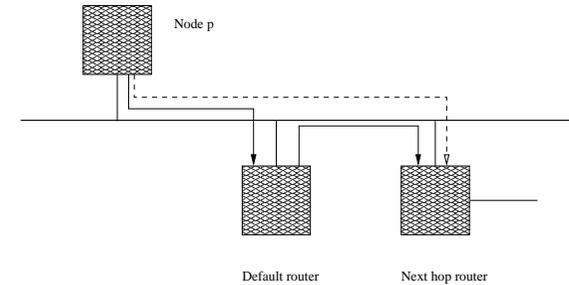
# RIP (V2): formato do pacote

|                           |         |           |
|---------------------------|---------|-----------|
| Comand                    | Version | 0         |
| Address family identifier |         | Route tag |
| IP address                |         |           |
| Subnet mask               |         |           |
| Next hop                  |         |           |
| Metric                    |         |           |

- ▶ Route tag: usado para identificar rotas externas.

# Next hop: ilustração

- ▶ Quando duas redes lógicas partilham a mesma rede física.
- ▶ Permite enviar o pacote directamente para o próximo encaminador.



# RIP (V2): autenticação simples

|                           |         |                     |
|---------------------------|---------|---------------------|
| Comand                    | Version | 0                   |
| 0xFFFF                    |         | Authentication type |
| Authentication            |         |                     |
| Address family identifier |         | Route tag           |
| IP address                |         |                     |
| Subnet mask               |         |                     |
| Next hop                  |         |                     |
| Metric                    |         |                     |

- ▶ Posteriormente reforçado para suportar outras formas de autenticação.

# Resumo

- ▶ Encaminhamento estático.
- ▶ Encaminhamento dinâmico.
  - ▶ Encaminhamento baseado em vectores de distância.
  - ▶ Encaminhamento baseado em "estados-dos-elos".
- ▶ RIP.