



Redes de Computadores I

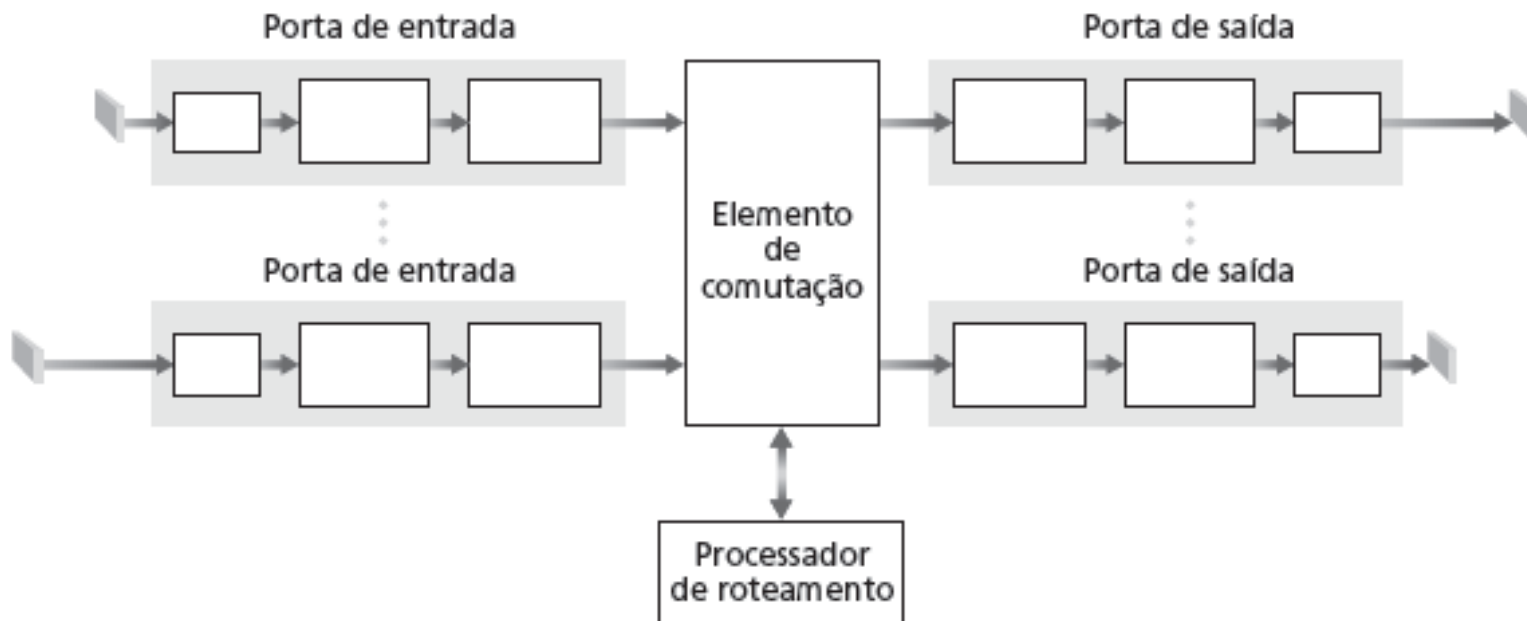
Roteamento

Fernando Parente Garcia

Funcionamento interno de roteadores e switches

Duas funções principais dos roteadores e comutadores:

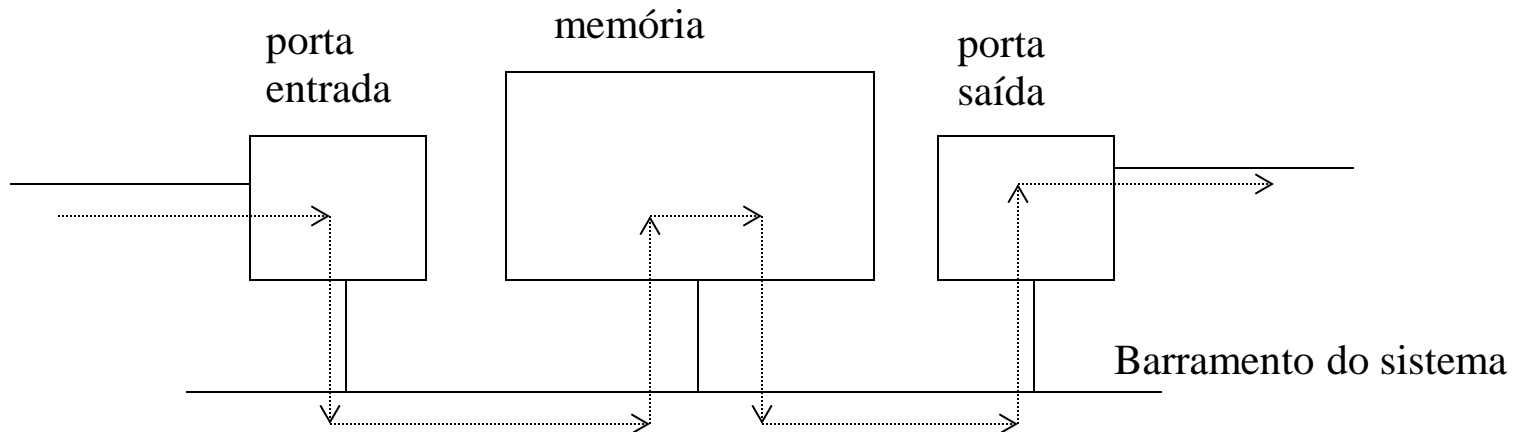
- executar algoritmos/protocolo de roteamento (RIP, OSPF, BGP)
- repassar datagramas do enlace de entrada para saída



Funcionamento interno Comutação por memória

Roteadores de primeira geração:

- computadores tradicionais com a comutação via controle direto da CPU
- pacote copiado para a memória do sistema
- velocidade limitada pela largura de banda da memória (2 travessias de barramento por datagrama)





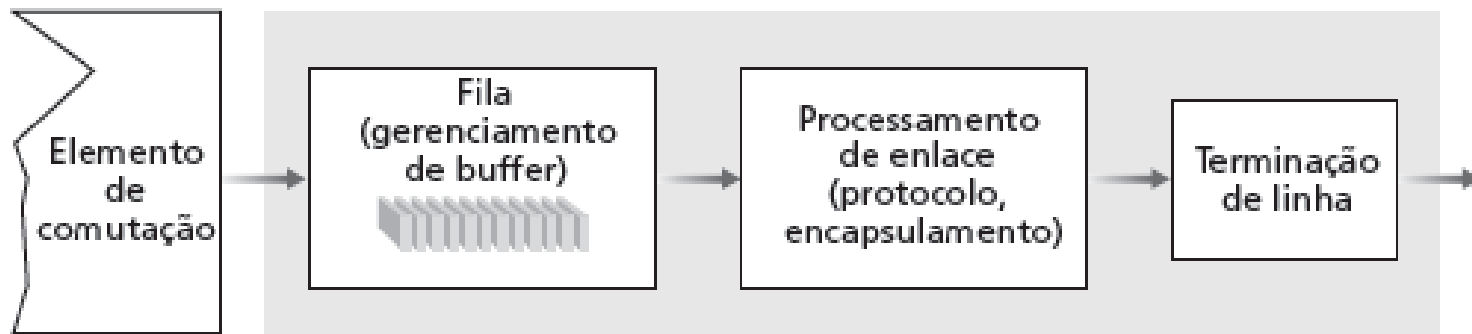
Funcionamento interno

Comutação por uma rede de interconexão

- contorna limitações de largura de banda do barramento
- redes de interconexão desenvolvidas para interligar as entradas às saídas
- projeto avançado: fragmenta datagrama em células de tamanho fixo, comuta células através do elemento de comutação
- Exemplo:
 - Cisco 12000: comuta 60 Gbps através da rede de interconexão

Funcionamento interno

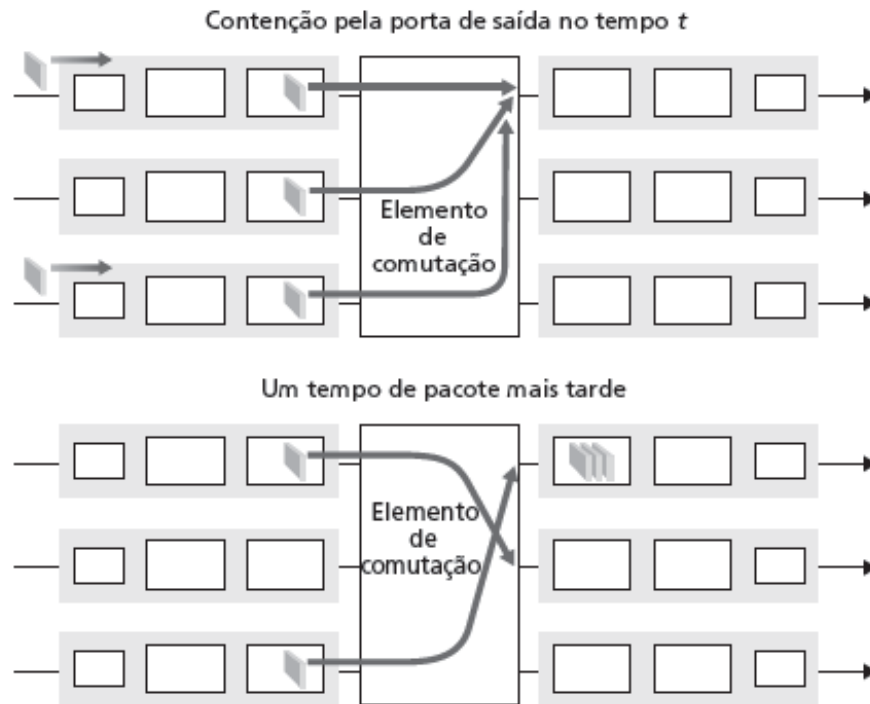
Portas de saída



- *Buffering* exigido quando os datagramas chegam do elemento de comutação mais rápido que a taxa de transmissão
- *Disciplina de escalonamento* escolhe entre os datagramas enfileirados para transmissão

Funcionamento interno

Enfileiramento na porta de saída

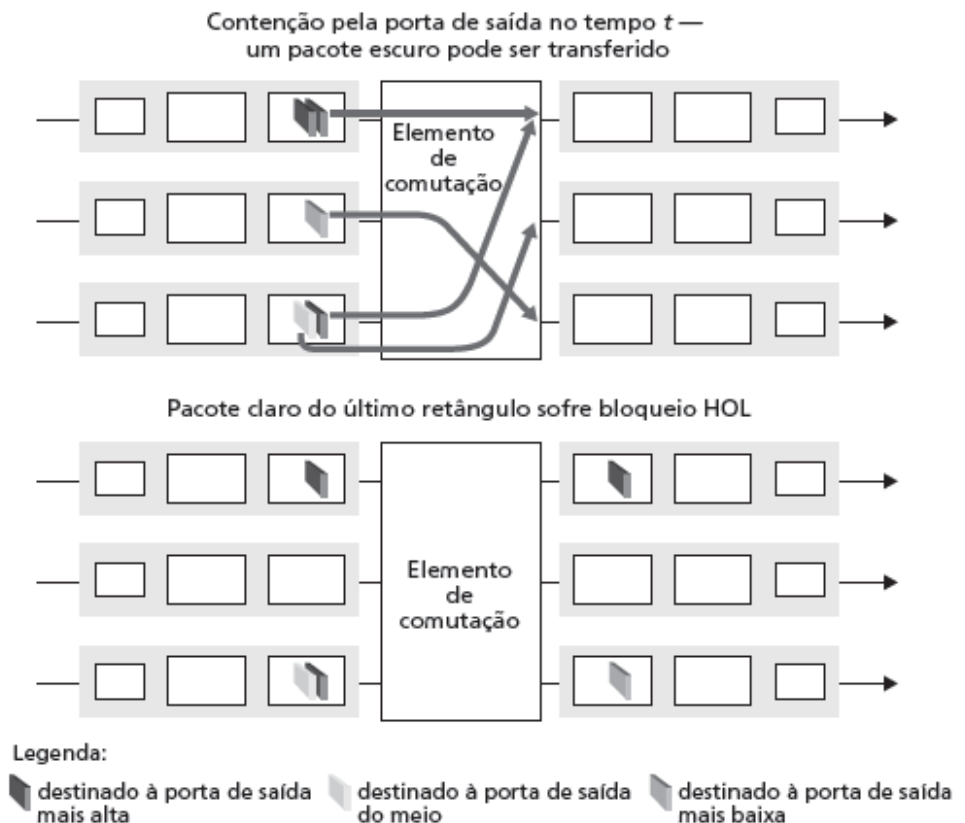


- **buffering** quando a taxa de chegada via comutador excede a velocidade da linha de saída
- **enfileiramento (atraso) e perda** devidos a estouro de buffer na porta de saída!

Funcionamento interno

Enfileiramento na porta de entrada

- elemento de comutação mais lento que portas de entrada combinadas -> enfileiramento possível nas filas de entrada
- **bloqueio de cabeça de fila (HOL):** datagrama enfileirado na frente da fila impede que outros na fila sigam adiante
- **atraso de enfileiramento e perda** devidos a estouro no buffer de entrada



Roteamento

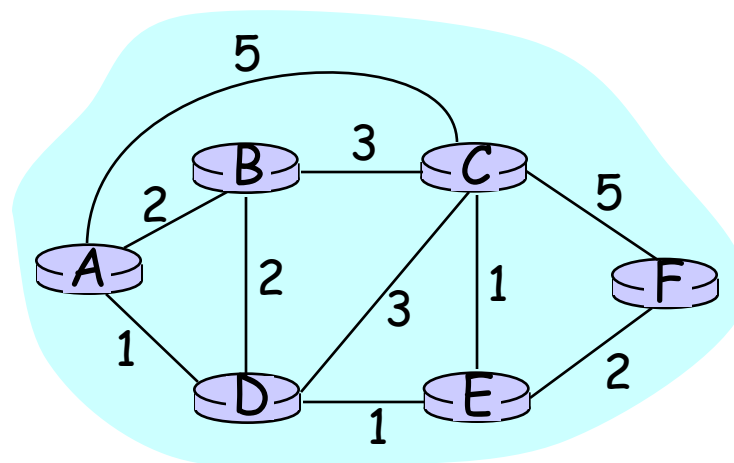
Introdução

Protocolo de Roteamento

OBJ: determinar “bons” caminhos (seqüência de roteadores) através da rede da fonte ao destino.

Algoritmos de roteamento são descritos por grafos:

- Nós do grafo são roteadores
- Arestas do grafo são enlaces
 - Custo do enlace: atraso, número de hops, congestionamento...
- “bons” caminhos:
 - tipicamente correspondem aos caminhos de menor custo
 - caminhos redundantes





Roteamento

Protocolos e Algoritmos

- Os protocolos de roteamento implementam um ou mais algoritmos de roteamento
 - Exemplos de Algoritmos
 - Distance Vector
 - Flooding
 - SPF (Shortest Path First)
 - Link State
 - Rota Fixa
 - Exemplos de protocolos
 - RIP
 - OSPF
 - IGRP
 - BGP



Algoritmos de roteamento

Características desejáveis

- Otimização
- Robustez
- Estabilidade
- Correção
- Simplicidade



Algoritmos de roteamento

Classificação

- **Informação global ou descentralizada**

- **Global:**

- Todos os roteadores tem informações completas da topologia e dos custos dos enlaces;
- Ex: algoritmo **Link State**.

- **Descentralizada:**

- Roteadores só conhecem informações sobre seus vizinhos e os enlaces para eles;
- Processo de computação iterativo, troca de informações com os vizinhos;
- Ex: algoritmo **Distance vector**.



Algoritmos de roteamento

Classificação

- **Estático ou Dinâmico**

- **Estático:**

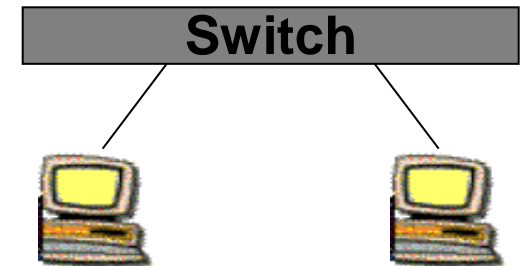
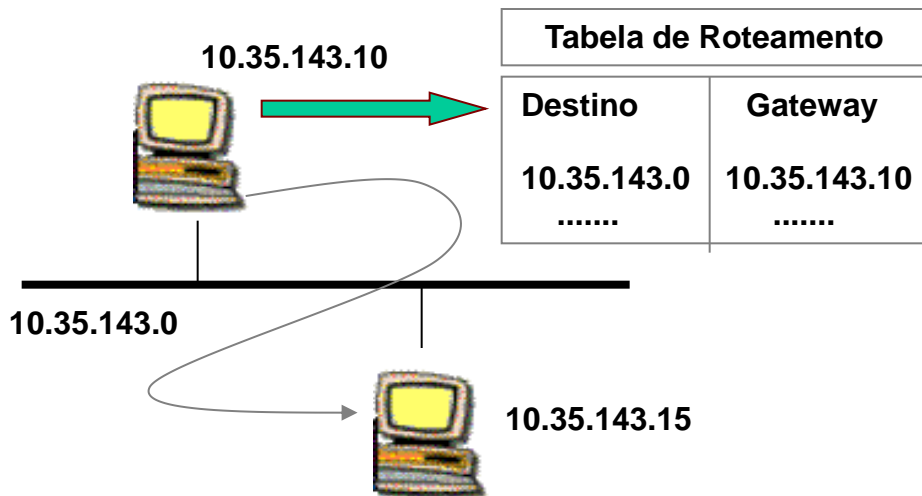
- As rotas são previamente definidas pelo administrador da rede;
- Ex: Rota Fixa.

- **Dinâmico:**

- As rotas são mantidas e atualizadas pelos roteadores;
- Atualizações periódicas;
- Roteadores podem responder a mudanças no custo dos enlaces;
- Ex: Vetor de Distância

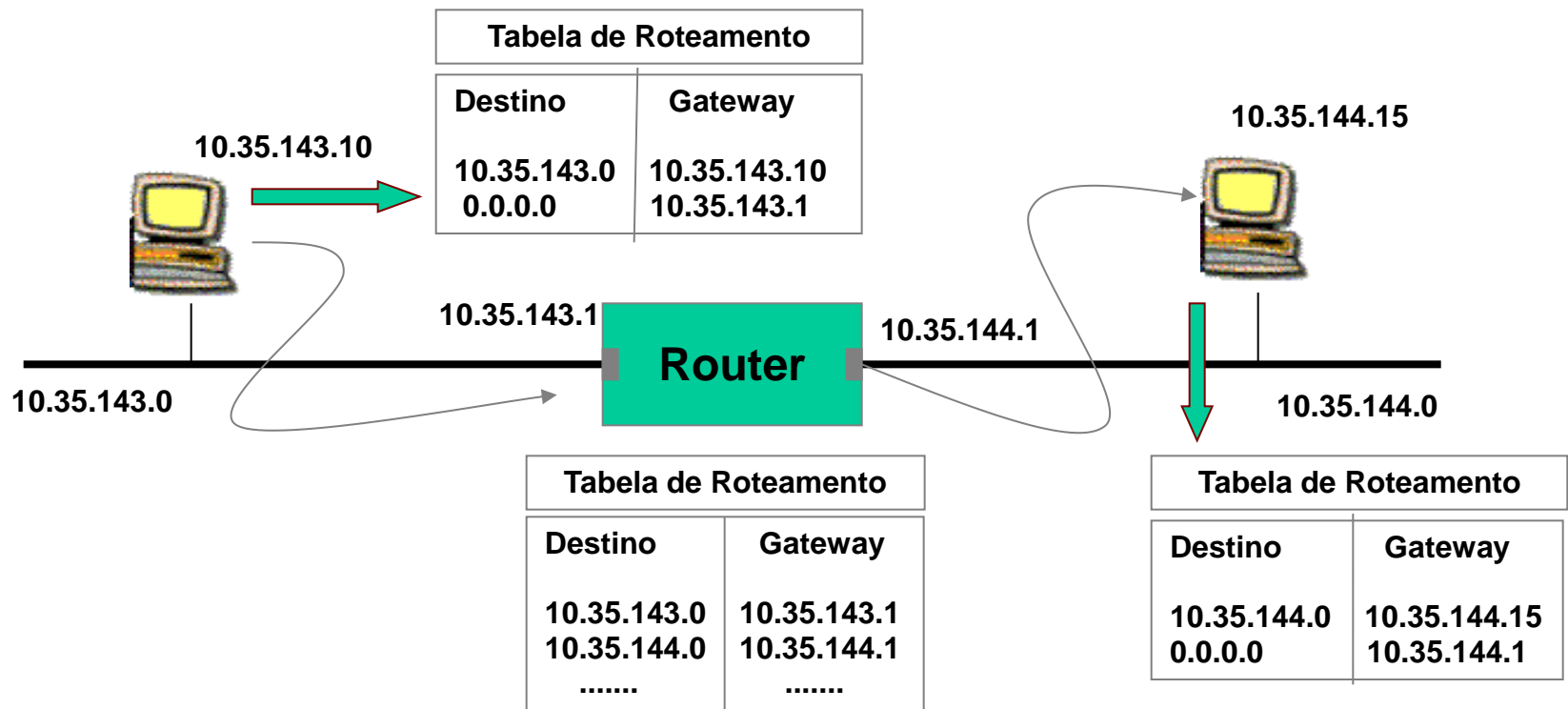
Roteamento Direto

- Origem e Destino na mesma rede



Roteamento Indireto

- Origem e Destino estão em redes diferentes.





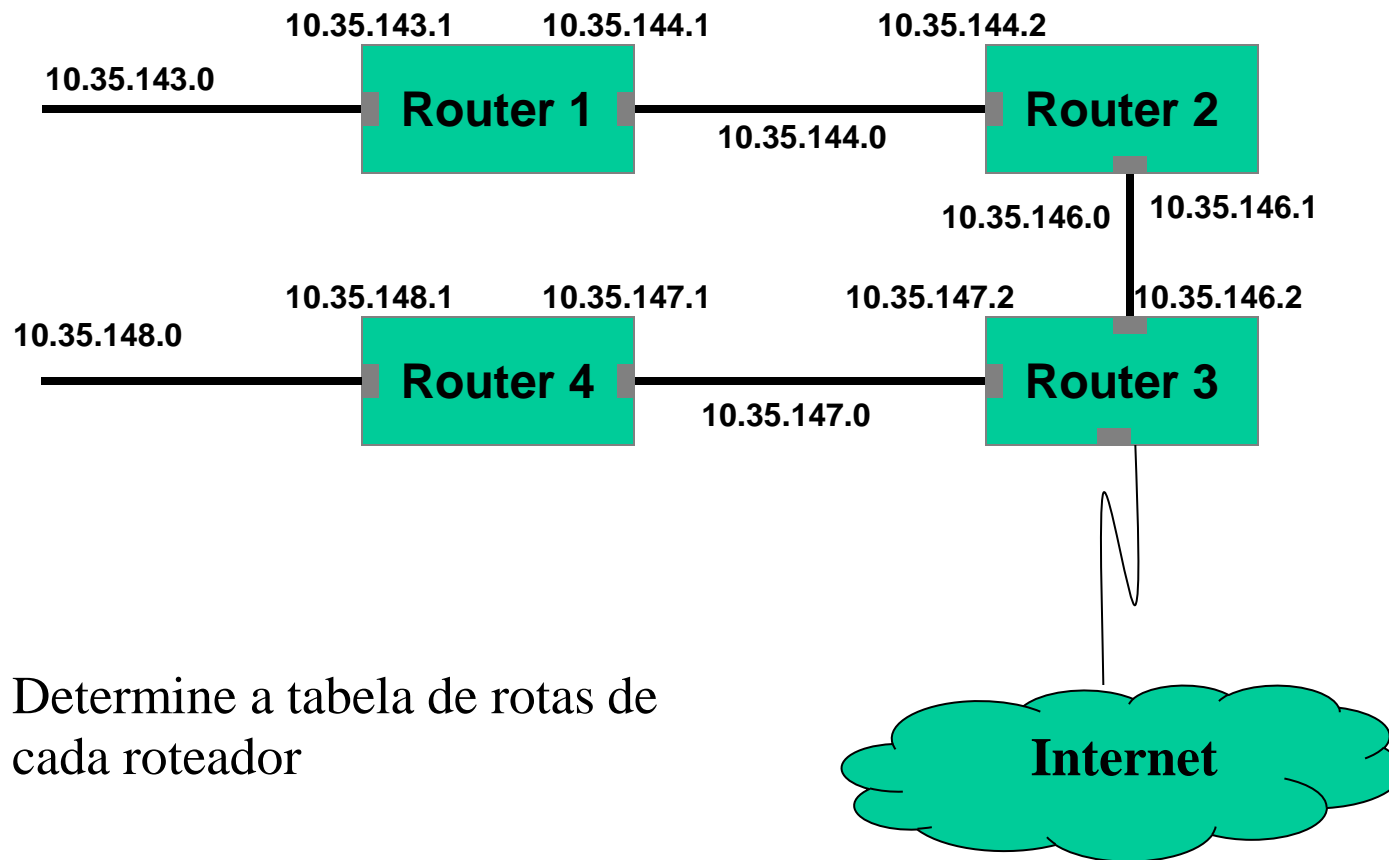
Roteamento Estático

Características

- Configurado manualmente pelo administrador da rede;
- A tabela de roteamento é estática;
- As rotas não se alteram dinamicamente de acordo com as alterações da topologia da rede;
- Custo de manutenção cresce de acordo com a complexidade e tamanho da rede;
- Sujeito a falhas de configuração.

Roteamento Estático

Exemplo





Roteamento Dinâmico

Características

- Divulgação e alteração das tabelas de roteamento de forma dinâmica pelos próprios roteadores;
 - Não há intervenção constante do administrador da rede;
- Alteração das tabelas dinamicamente de acordo com a alteração da topologia da rede;
 - **Adaptativo:** a tabela de roteamento pode ser alterada de acordo com as alterações dos custos dos enlaces;
- Diminui consideravelmente o tempo de manutenção das tabelas em grandes redes;
- Também está sujeito a falhas.



Roteamento Dinâmico

Vetor de Distância (distance vector)

- Bellman-Ford
- É um algoritmo simples
 - Um roteador mantém uma lista de todas as rotas conhecidas em uma tabela;
 - Cada roteador divulga para os seus vizinhos as rotas que conhece;
 - Cada roteador seleciona dentre as rotas conhecidas e as divulgadas os melhores caminhos.



Algoritmo Vetor de Distância Métrica

- A escolha do melhor caminho é baseada na comparação da métrica do enlace
 - Normalmente: **Melhor = menor caminho**
- A métrica é o custo de envio em um enlace
 - Pode ser calculada utilizando vários parâmetros:
 - Taxa de transmissão em bps
 - Vazão
 - Atraso
 - Tráfego
 - Congestionamento
 - Número de saltos (no. de *hops*) (+ usado)



Algoritmo Vetor de Distância

1. Quando o roteador executa o “boot” ele armazena na tabela informações sobre cada uma das redes que estão diretamente conectadas a ele.
 - Cada entrada na tabela indica uma rede destino, o gateway para a rede e a sua métrica.
2. Periodicamente cada roteador envia uma cópia da sua tabela para todos os roteadores que sejam diretamente alcançáveis.
3. Cada roteador que recebe uma cópia da tabela, verifica as rotas divulgadas e suas métricas.
 - O roteador soma à métrica divulgada o custo do enlace entre ele e o roteador que fez a divulgação.
 - Em seguida, compara cada uma das entradas da tabela divulgada com as da sua tabela de roteamento.
 - Rotas novas são adicionadas, rotas existentes são selecionadas pela sua métrica.



Algoritmo Vetor de Distância

3.1 Se a rota já existe na tabela e a métrica calculada é menor do que a da rota conhecida

- Remove a entrada anterior e adiciona a nova rota divulgada.

3.2 Se a rota já existe na tabela e a métrica calculada é igual a da rota conhecida

- Não altera a entrada.

3.3. Se a rota já existe na tabela e a métrica divulgada é maior do que a da rota conhecida, então verifica se o gateway desta rota é o mesmo que está fazendo nova divulgação

- Se o gateway é o mesmo altera a métrica para esta rota
- Se o gateway não é o mesmo não altera a rota conhecida



Routing Information Protocol (RIP)

- Implementa o algoritmo Vetor Distância;
- A métrica utilizada é o número de estações intermediárias (no. de *hops*);
- Não permite o balanceamento de tráfego;
- Cada roteador divulga sua tabela periodicamente a cada **30 segundos** através de um **broadcast** para todos os roteadores conectados diretamente à ele;
 - Se uma rota não for atualizada em 180 segundos é considerada inatingível;
 - A informação de rota inatingível é repassada aos roteadores “vizinhos” (diretamente alcançáveis);
- As mensagens divulgadas levam *n tuplas* contendo
<redes destino, métrica>



RIP

Problemas

- Não tem mecanismos de segurança
 - É suscetível a *spoofing*;
- Não tem controle de “idade” das mensagens
 - Mensagens “velhas” podem ser processadas após mensagens “novas”
 - Inconsistência nas tabelas de roteamento
- Problemas de loops na divulgação das rotas
- Limitação de número de roteadores intermediários
 - Métrica = 16, indica rota inalcançável
- Não suporta máscara de subrede



RIP

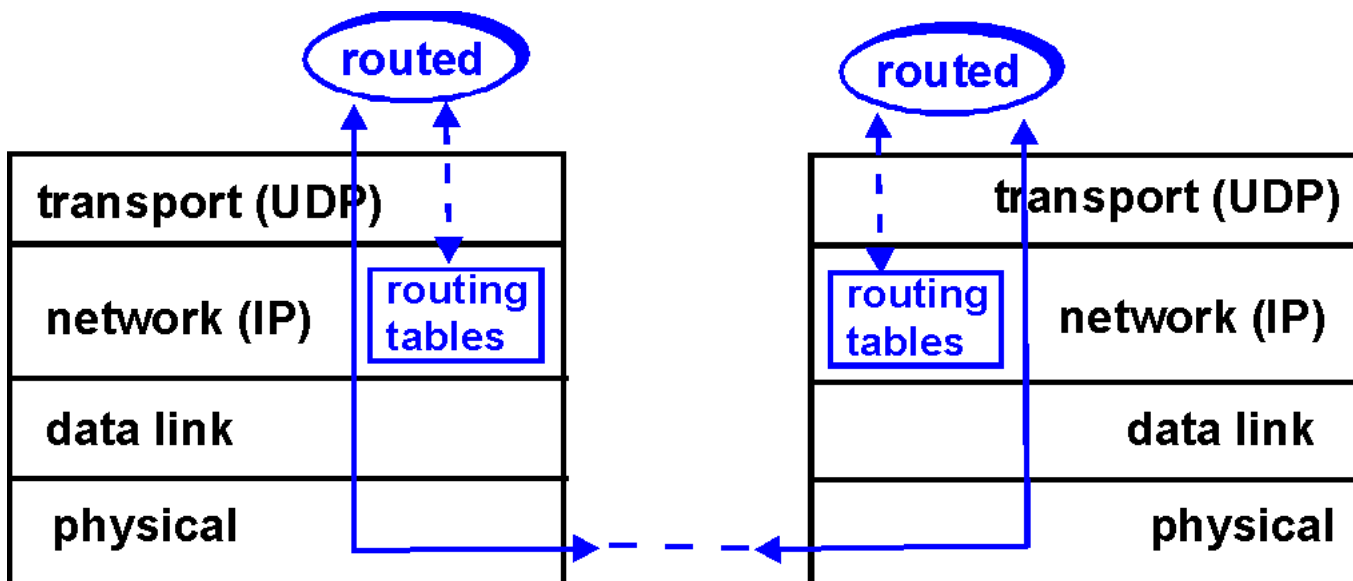
Falha de Enlaces e Recuperação

- Se não há comunicação depois de 180 segundos, o vizinho e o enlace são declarados mortos
 - rotas através do vizinho são anuladas
 - novos anúncios são enviados aos vizinhos
 - os vizinhos por sua vez devem enviar novos anúncios (se suas tabelas de rotas foram alteradas)
 - a falha de um enlace se propaga rapidamente para a rede inteira
 - poison reverse é usado para prevenir loops, isto é, evitar que a rota para um destino passe pelo próprio roteador que está enviando a informação de distância (distância infinita= 16 hops)

RIP

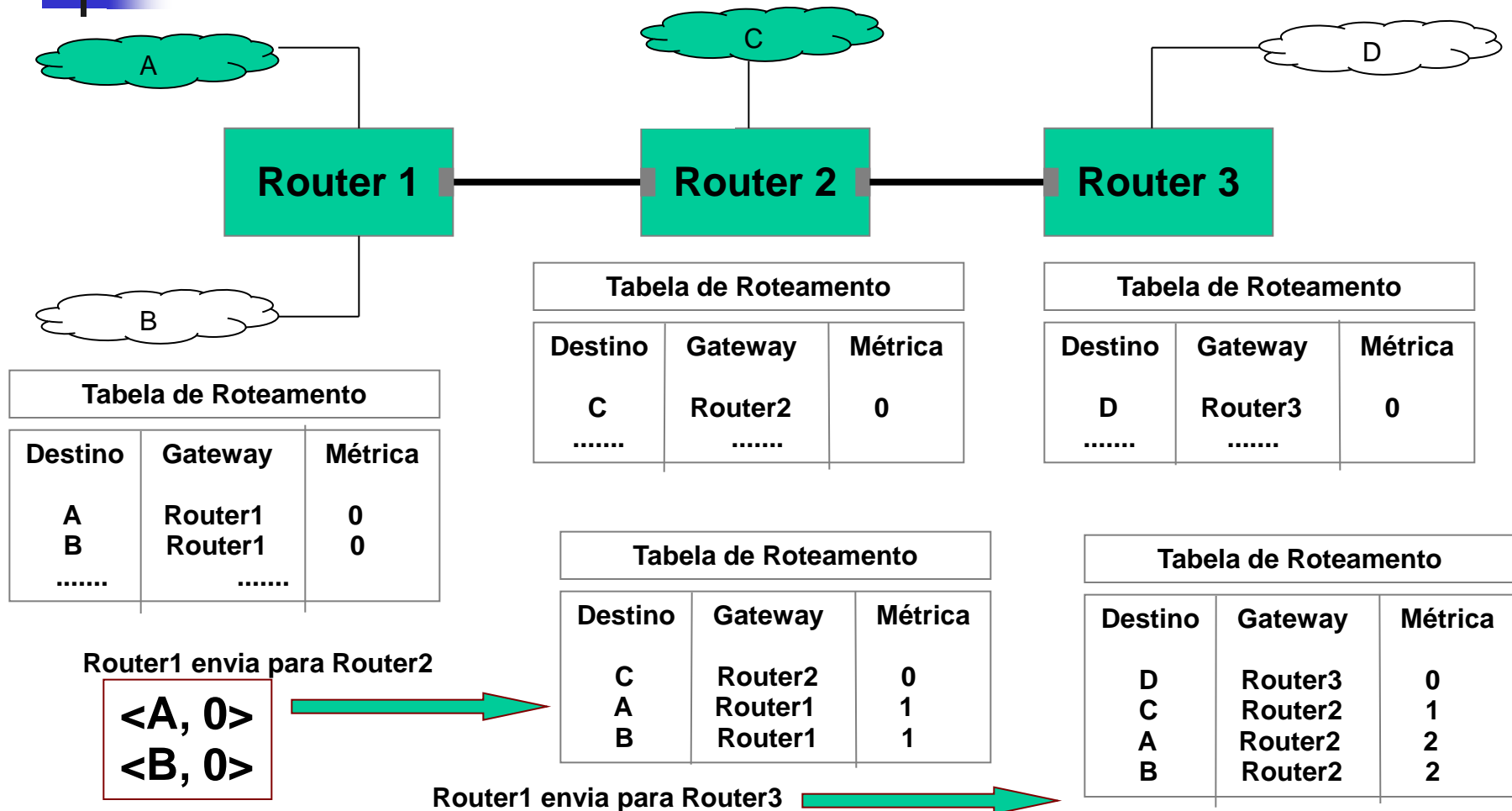
Processamento da tabela de rotas

- As tabelas de roteamento do RIP são manipuladas por um processo de aplicação chamado routed (daemon).
- Anúncios são enviados periodicamente em segmentos UDP.



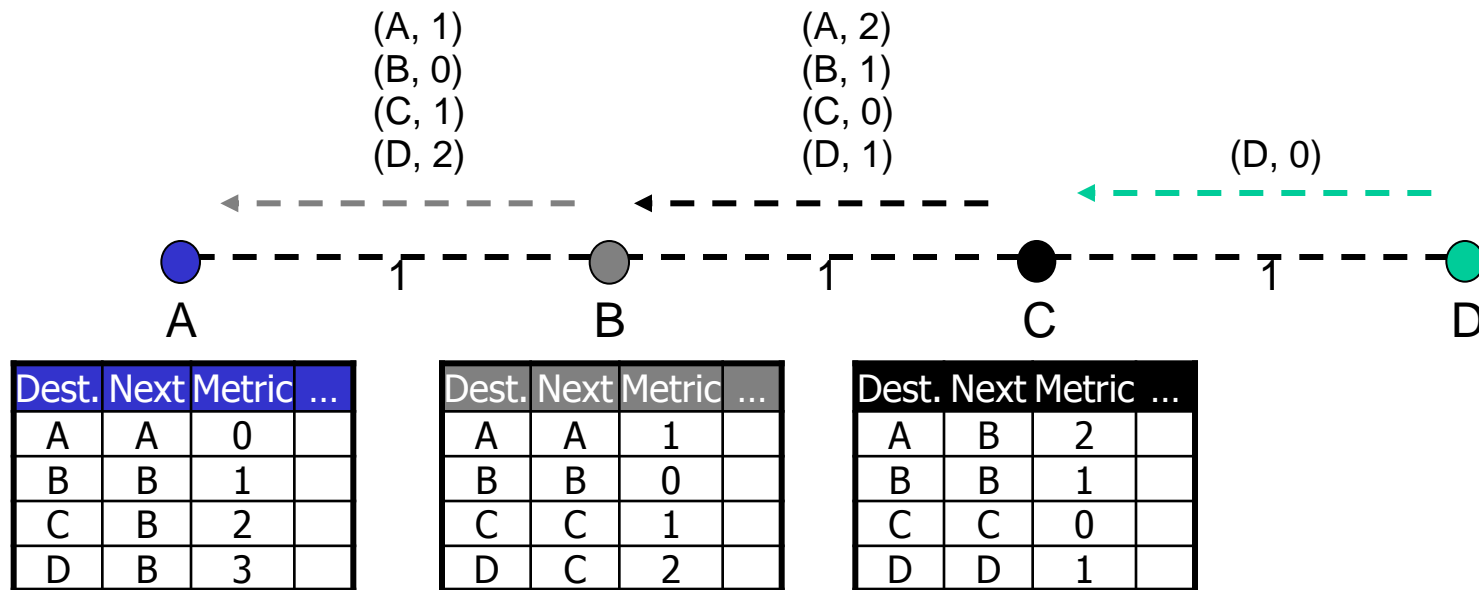
RIP

Exemplo



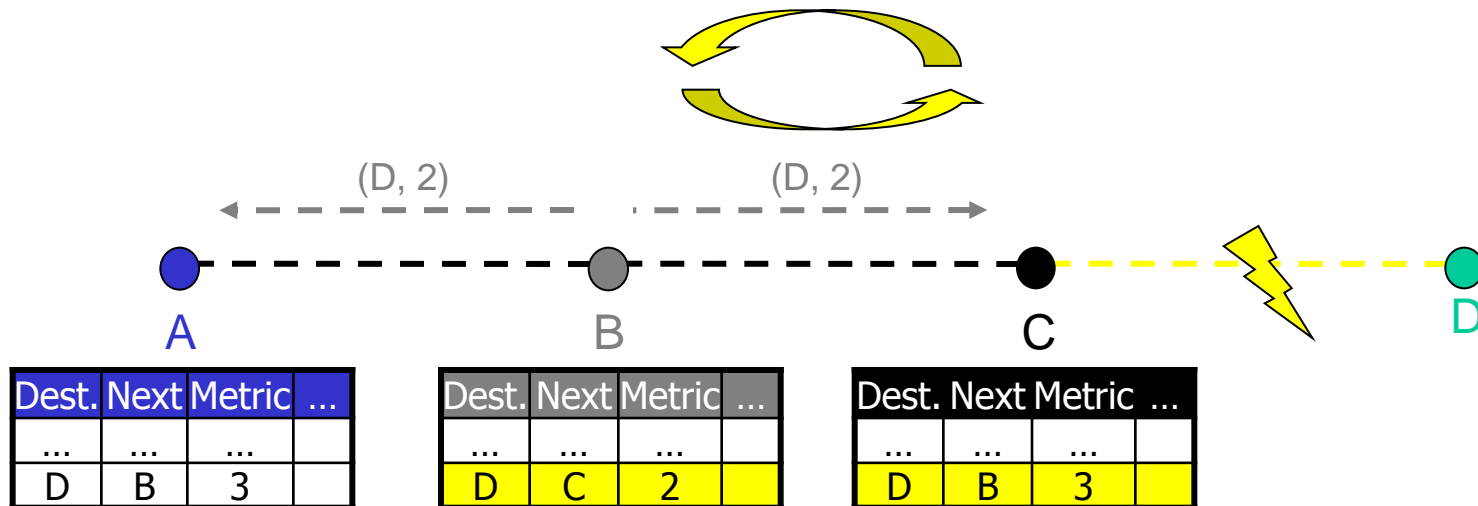
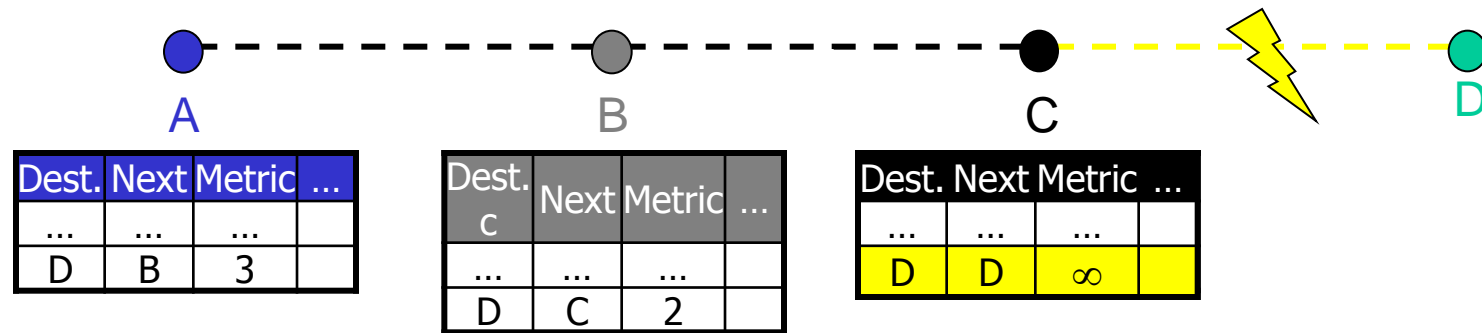
RIP

Inserção de um nó na rede



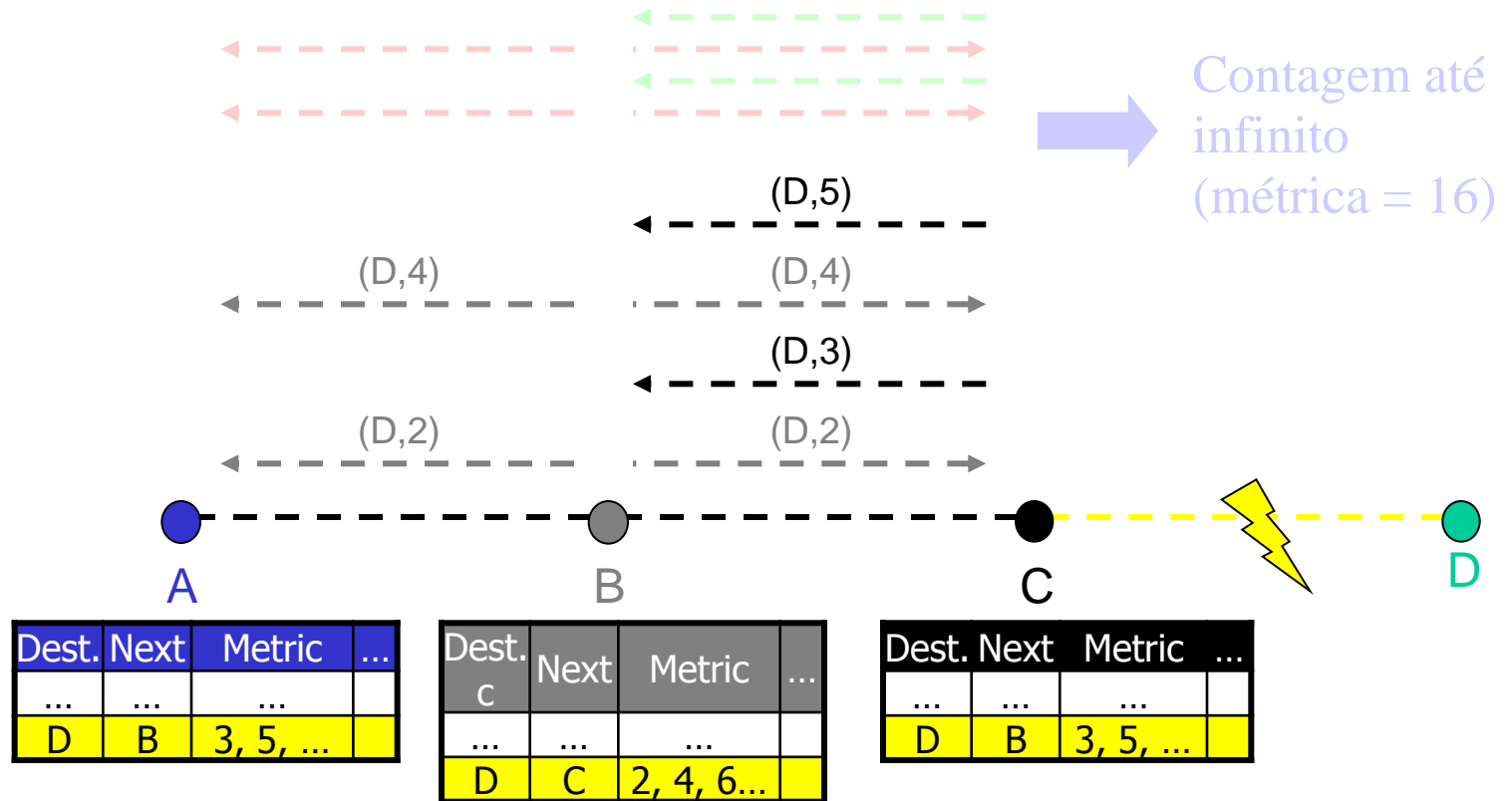
RIP

Queda de um enlace



RIP

Queda de um enlace





IGRP (Interior Gateway Routing Protocol)

- Protocolo proprietário da CISCO;
- Sucessor do RIP (meados dos anos 80);
- Algoritmo vetor de distância;
- Várias métricas de custo:
 - atraso, largura de banda, taxa de erro, tráfego, etc.
- Usa o TCP para trocar informações de novas rotas;
- Loop-free routing
 - Distributed Updating Algorithm (DUAL) baseado em técnicas de *computação difusa*.



Algoritmo de estado do enlace (Link State)

algoritmo de Dijkstra

- Topologia e custos de enlace conhecidos de todos os nós
 - realizado por “broadcast de estado do enlace”
 - todos os nós têm a mesma informação
- calcula caminhos de menor custo de um nó (“origem”) para todos os outros nós

notação:

- $c(x,y)$: custo do enlace do nó x até y ; $= \infty$ se não forem vizinhos diretos
- $D(v)$: valor atual do custo do caminho da origem ao destino v
- $p(v)$: nó predecessor ao longo do caminho da origem até v
- N' : conjunto de nós cujo caminho de menor custo é definitivamente conhecido



Algoritmo de Dijkstra

1 **Inicialização:**

2 $N' = \{u\}$

3 para todos os nós v

4 se v adjacente a u

5 então $D(v) = c(u, v)$

6 senão $D(v) = \infty$

7

8 **Loop**

9 acha w não em N' tal que $D(w)$ é mínimo

10 acrescenta w a N'

11 atualiza $D(v)$ para todo v adjacente a w e não em N' :

12 $D(v) = \min(D(v), D(w) + c(w, v))$

13 /* novo custo para v é custo antigo para v ou custo conhecido

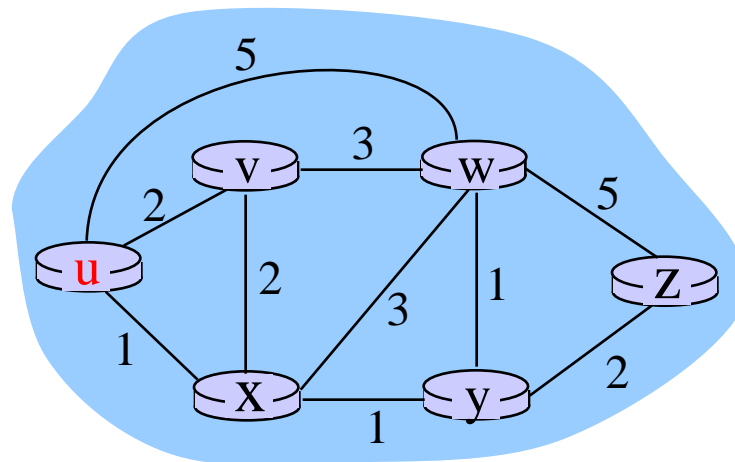
14 do caminho mais curto para w + custo de w para v */

15 **até todos os nós em N'**



Algoritmo de Dijkstra: exemplo

Etapa	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					



Algoritmo de Dijkstra: exemplo

árvore resultante do caminho
mais curto a partir de u:

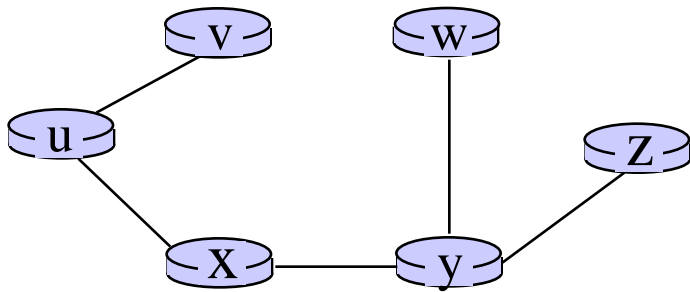


tabela de repasse resultante em u:

destino	enlace
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)



OSPF

(Open Shortest Path First)

- “open”: publicamente disponível
- usa algoritmo Link State
 - disseminação de pacote LS
 - mapa de topologia em cada nó
 - cálculo de rota usando algoritmo de Dijkstra
- anúncio OSPF transporta uma entrada por roteador vizinho



Recursos “avanzados” do OSPF (nãõ no RIP)

- **segurança**: todas as mensagens OSPF autenticadas (para impedir intrusão maliciosa)
- **múltiplos caminhos** de mesmo custo permitidos (apenas um caminho no RIP)
- suporte integrado para uni e **multicast**:
 - Multicast OSPF (MOSPF) usa mesma base de dados de topologia que o OSPF
- OSPF **hierárquico** em grandes domínios



OSPF

(Open Shortest Path First)

- **hierarquia em dois níveis:** área local, backbone.
 - anúncios de estado do enlace somente na área
 - cada nó tem topologia de área detalhada; somente direção conhecida (caminho mais curto) para redes em outras áreas.
- *roteadores de borda:* “resumem” distâncias às redes na própria área, anunciam para outros roteadores de borda.
- *roteadores de backbone:* executam roteamento OSPF limitado ao backbone.
- *roteadores de fronteira:* conectam-se a outros AS's.



Sistemas Autônomos

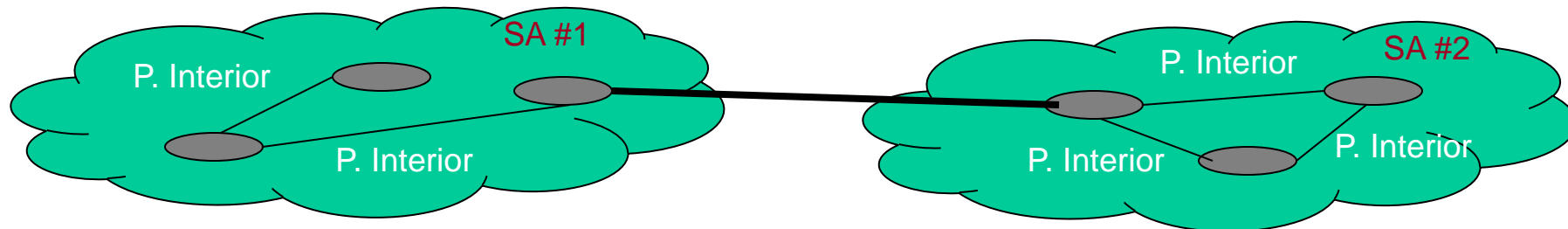
Introdução

- Um SA (Sistema Autônomo) pode ser definido como
"Um grupo de redes e roteadores controlados por uma única autoridade administrativa."
- Roteadores em um sistema autônomo seguem as mesmas "regras" de roteamento.
- Protocolos de roteamento são classificados de acordo com sua atuação.

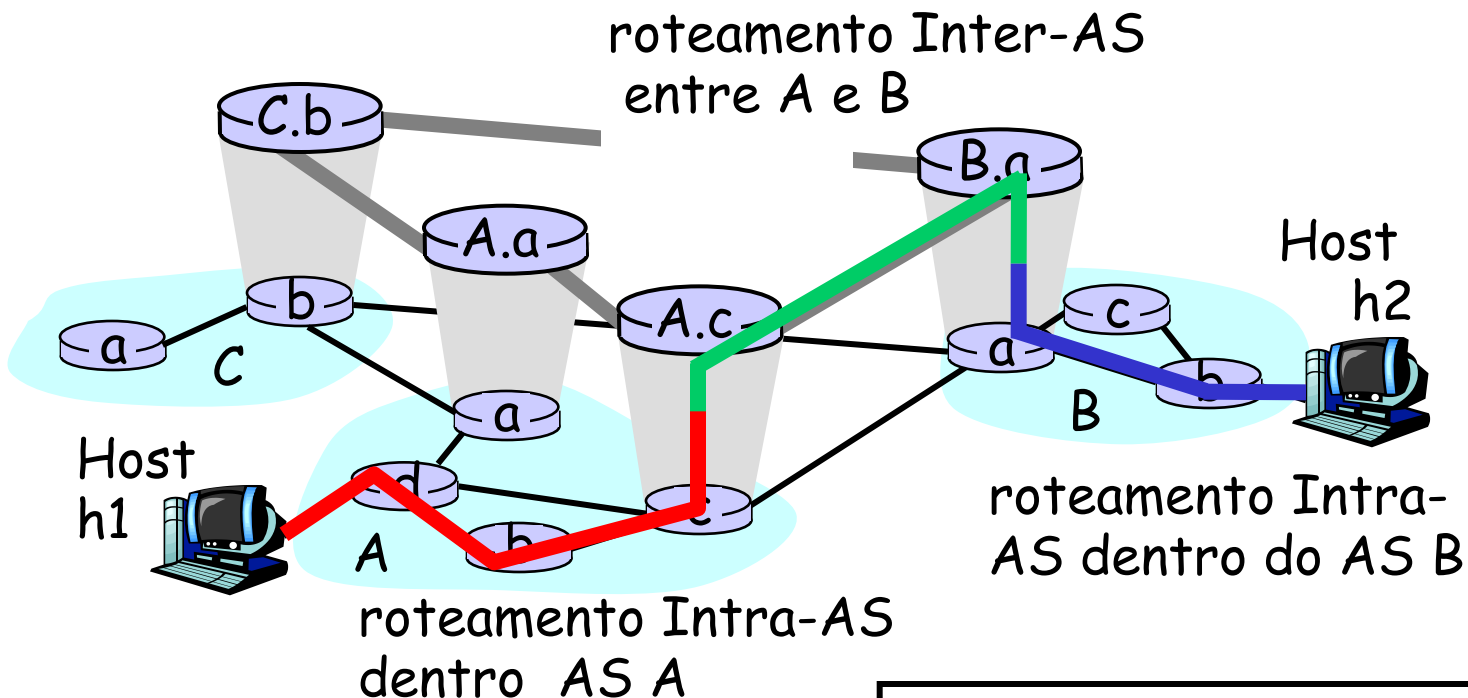
Sistemas Autônomos

Protocolo Interiores e Exteriores

- Protocolos Interiores (Intra-AS)
 - São aqueles utilizados para comunicação entre roteadores de um mesmo sistema autônomo.
- Protocolos Exteriores (Inter-AS)
 - São aqueles utilizados para comunicação entre roteadores de sistemas autônomos diferentes.



Roteamento Intra-AS e Inter-AS

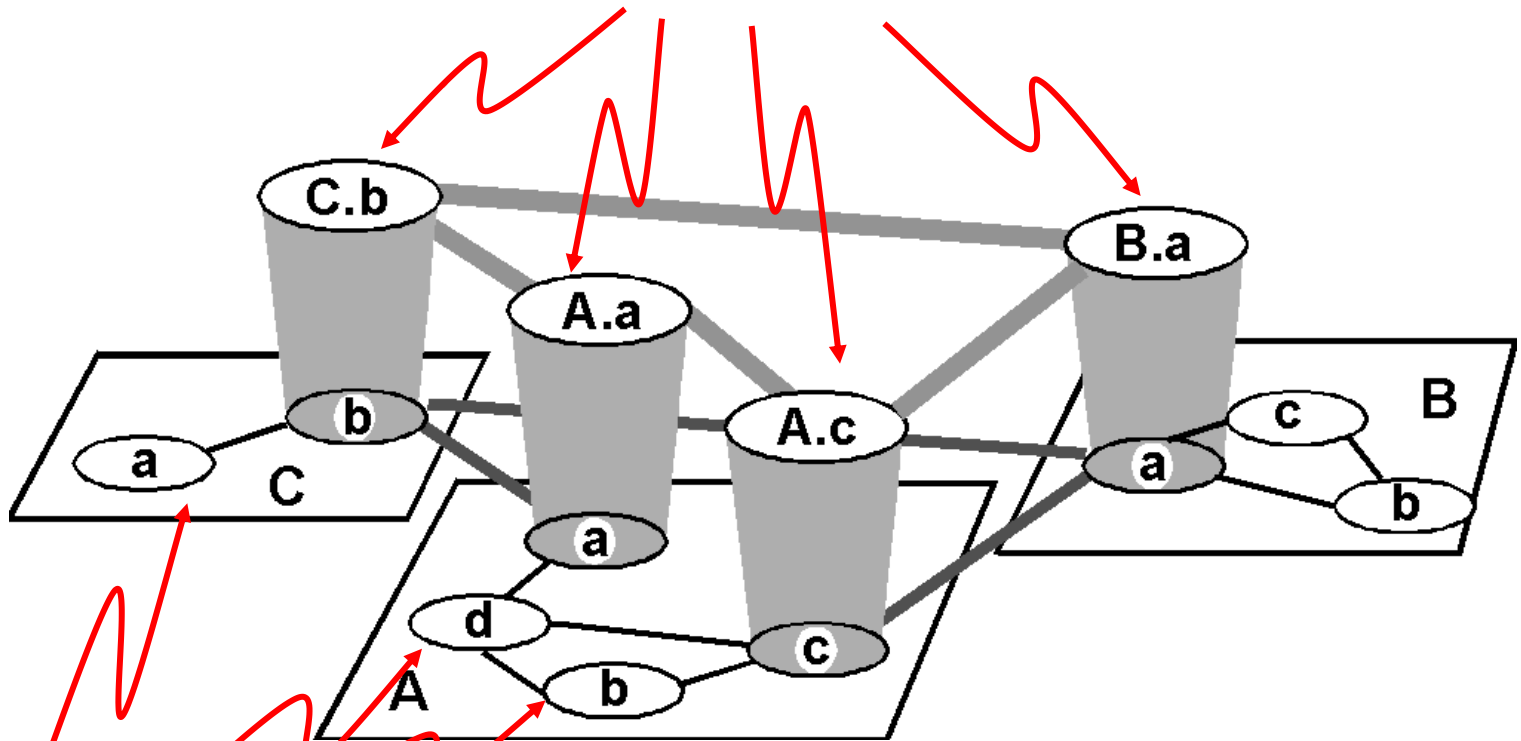


Roteadores de Borda

- realizam roteamento inter-AS entre si
- realizam roteamento intra-AS com outros roteadores do mesmo AS

Hierarquia de AS

Roteador de borda Intra-AS (exterior gateway)



Roteador interno Inter-AS (gateway)



Hierarquia de AS

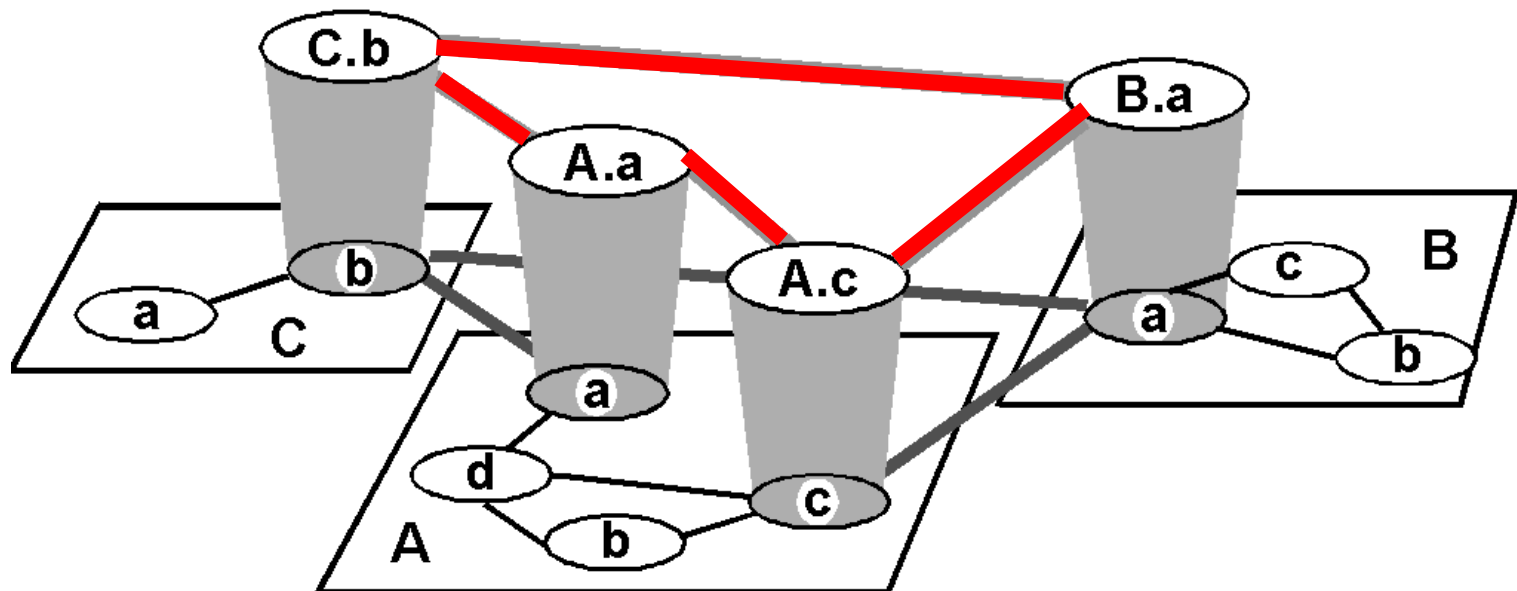
- Roteadores no mesmo AS rodam o mesmo protocolo de roteamento
 - Protocolo de roteamento "Intra-AS"
- Roteadores em diferentes AS's podem rodar diferentes protocolos de roteamento



Roteamento Intra-AS

- Também conhecido como **Interior Gateway Protocols (IGP)**;
- IGP's mais comuns:
 - **RIP**: Routing Information Protocol
 - **IGRP**: Interior Gateway Routing Protocol
 - proprietário da Cisco
 - **OSPF**: Open Shortest Path First

Roteamento Inter-AS



Ex.: BGP (Border Gateway Protocol)



Roteamento Inter-AS

BGP

- BGP (Border Gateway Protocol)
 - E o padrão de fato para uso na Internet;
- Algoritmo **Path Vector**
 - Similar ao algoritmo vetor de distância;
 - Cada roteador de borda envia em broadcast aos seus vizinhos o caminho inteiro (a seqüência de ASs) até o destino;



Roteamento Inter-AS

BGP

- **Exemplo:** Roteador **X** envia seu caminho até o destino **Z** para o roteador vizinho **W**

$$\text{Path (X,Z)} = \text{X,Y1,Y2,Y3,...,Z}$$

- O roteador **W** pode escolher ou não o caminho oferecido por **X**
 - critérios de escolha:
 - Custo
 - Regras (não rotear através de AS rivais)
- Se o roteador **W** seleciona o caminho oferecido por **X**, então:
$$\text{Path (W,Z)} = \text{W, Path (X,Z)}$$
- Nota: O roteador **X** pode controlar o tráfego de entrada controlando as rotas que ele informa aos seus vizinhos:
 - Se **X** não quer rotear tráfego para **Z**, então **X** não informa nenhuma rota para **Z**.



Roteamento Inter-AS

BGP

- As mensagens do BGP são trocadas encapsuladas no TCP.
- Mensagens BGP:
 - **OPEN**
 - Inicia a conexão TCP com um roteador vizinho e autentica o transmissor.
 - **UPDATE**
 - Anuncia novo caminho (ou retira um velho).
 - **KEEPALIVE**
 - Mantém a conexão viva em caso de ausência de atualizações;
 - **NOTIFICATION:**
 - Reporta erros nas mensagens anteriores;
 - Também usado para encerrar uma conexão.

