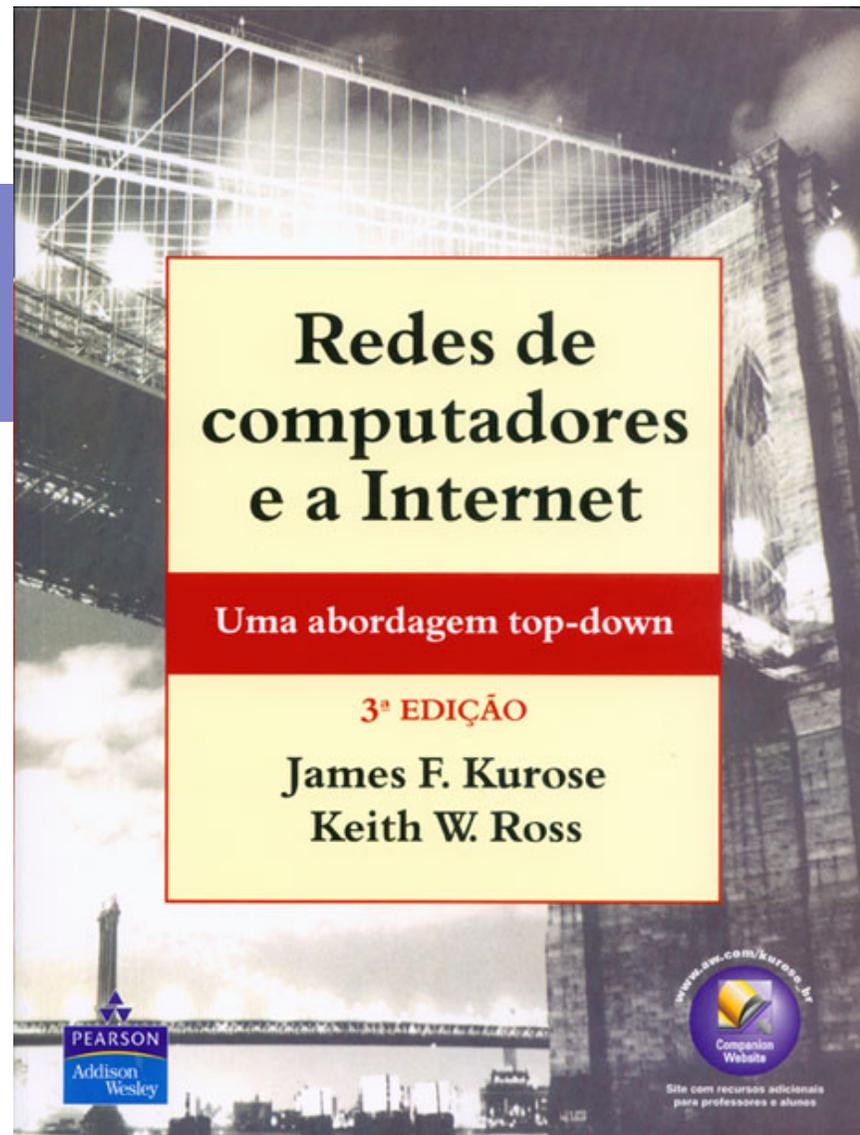


# Redes de computadores e a Internet

## Capítulo 4

### A camada de rede



# 4 A camada de rede

## Objetivos do capítulo:

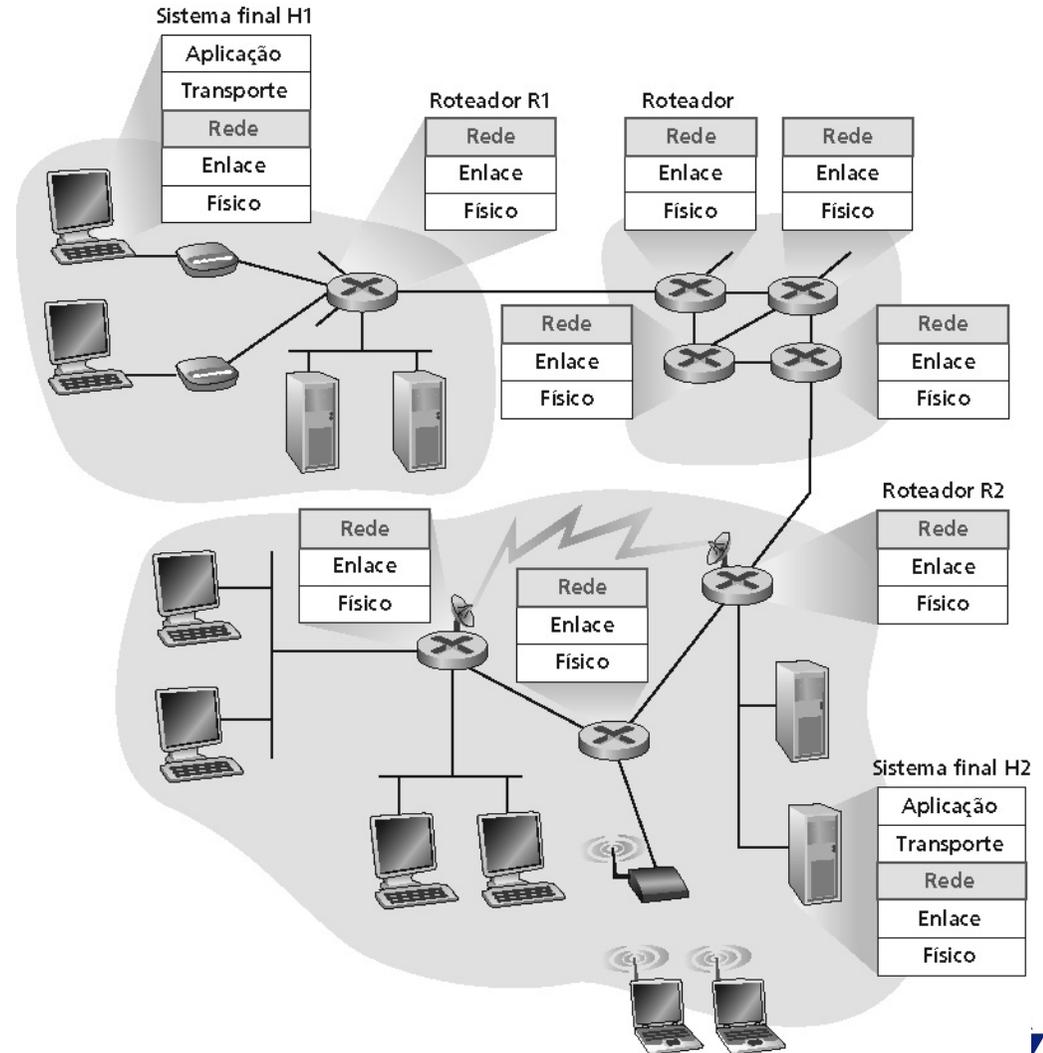
- Entender os princípios dos serviços da camada de rede:
  - Roteamento (seleção de caminho)
  - Escalabilidade
  - Como funciona um roteador
  - Tópicos avançados: IPv6, mobilidade
- Instanciação e implementação na Internet

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 A camada de rede

- Transporta segmentos do hospedeiro transmissor para o receptor
- No lado transmissor, encapsula os segmentos em datagramas
- No lado receptor, entrega os segmentos à camada de transporte
- Protocolos da camada de rede em *cada* hospedeiro, roteador
- Roteador examina campos de cabeçalho em todos os datagramas IP que passam por ele



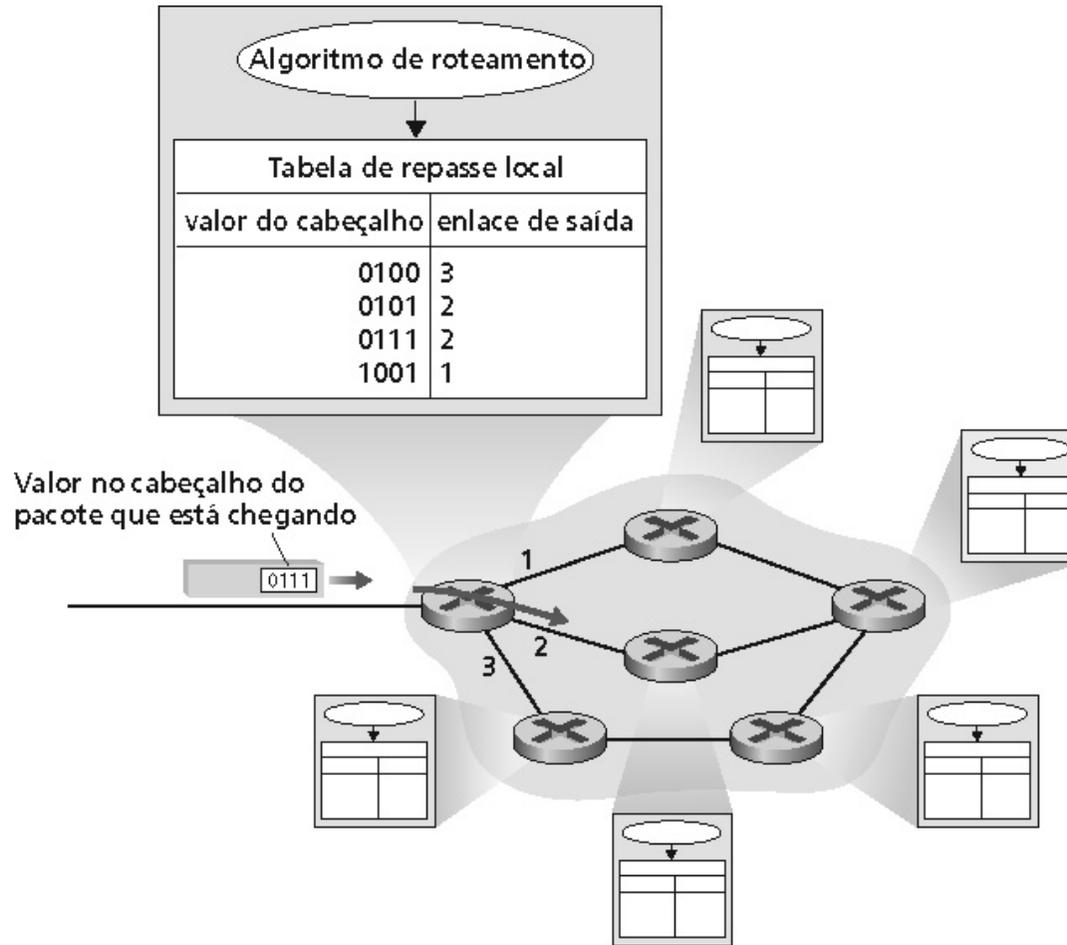
# 4 Funções-chave da camada de rede

- **Comutação:** mover pacotes da entrada do roteador para a saída apropriada do roteador
- **Roteamento:** determinar a rota a ser seguida pelos pacotes desde a origem até o destino.
  - **Algoritmos de roteamento**

## Analogia:

- **Roteamento:** processo de planejar a viagem da origem ao destino
- **Comutação:** processo de passar por um único intercâmbio

# 4 Interação entre roteamento e comutação



# 4 Estabelecimento de conexão

- 3ª função importante em *algumas* arquiteturas de rede:
  - ATM, frame relay, X.25
- Antes do fluxo de datagramas, dois hospedeiros e os devidos roteadores estabelecem uma conexão virtual
  - Roteadores são envolvidos
- Serviço de conexão da camada de rede e de transporte:
  - **Rede:** entre dois hospedeiros
  - **Transporte:** entre dois processos

# 4 Modelo de serviço de rede

P.: Como escolher o **modelo de serviço** para o “canal” de transporte de datagramas do transmissor ao receptor?

**Exemplo de serviços para datagramas individuais:**

- Garantia de entrega
- Garantia de entrega com menos do que 40 msec de atraso

**Exemplo de serviços para um fluxo de datagramas:**

- Entrega em ordem dos datagramas
- Garantia de uma banda mínima para o fluxo
- Restrições em mudanças no espaçamento entre pacotes

# 4 Modelos de serviço da camada de rede

Arquitetura de rede	Modelo de serviço	Parâmetros garantidos				Realim. de congestão
		Banda	Perda	Ordem	Tempo	
Internet	melhor esforço	não	não	não	não	não (examina perdas)
ATM	CBR	taxa constante	sim	sim	sim	não há congestão
ATM	VBR	taxa garantida	sim	sim	sim	não há congestão
ATM	ABR	mínimo garantido	não	sim	não	sim
ATM	UBR	não	não	sim	não	não

- Novos serviços na Internet: Intserv, Diffserv

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast



# 4 Camada de rede: serviços de conexão e sem-conexão

- Redes de datagrama provêem serviços sem-conexão na camada de rede
- Redes de circuito virtual provêem serviços de conexão na camada de rede
- Análogo aos serviços da camada de transporte, mas:
  - **Serviço:** hospedeiro-a-hospedeiro
  - **Sem escolha:** a rede provê ou um ou outro
  - **Implementação:** no núcleo

# 4 Circuitos virtuais (VC)

“A ligação entre a origem e o destino emula uma ligação telefônica”

- Orientado ao desempenho
- A rede controla a conexão entre a origem e o destino
- Estabelecimento da conexão deve preceder o envio de dados. Liberação da conexão após os dados.
- Cada pacote transporta um identificador do CV, não transporta o endereço completo do destino
- Cada roteador na rota mantém informação de estado para conexão que passa por ele.
- O link e os recursos do roteador (banda, buffers) podem ser alocados por VC

# 4 Implementação de VC

Um VC consiste de:

1. Caminho da origem até o destino
  2. Números de VC, um número para cada link ao longo do caminho
  3. Entradas em tabelas de comutação em roteadores ao longo do caminho
- Pacotes pertencentes a um VC carregam um número de VC
  - O número de VC deve ser trocado em cada link
  - Novos números de VC vêm da tabela de comutação

# 4 Tabela de comutação

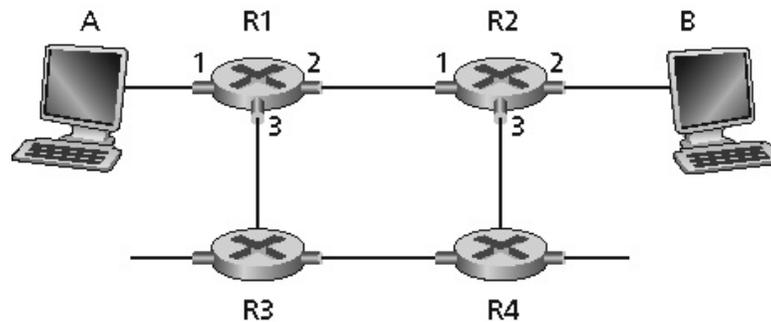


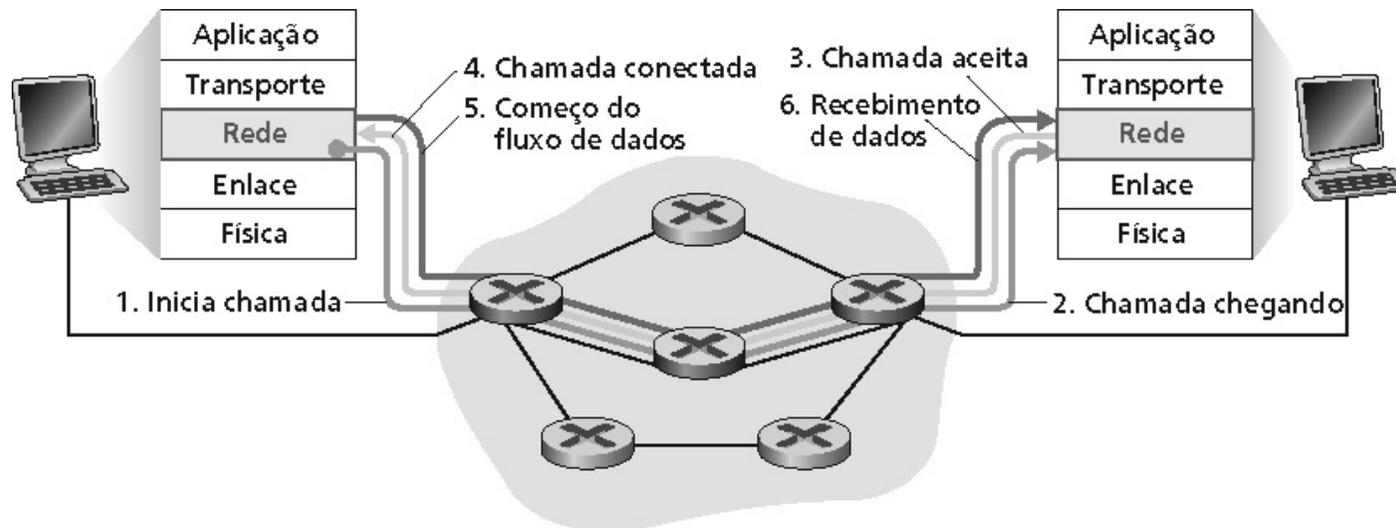
Tabela de comutação no roteador a noroeste:

Interface de entrada	VC # de entrada	Interface de saída	VC # de saída
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Roteadores mantêm informações de estado de conexão

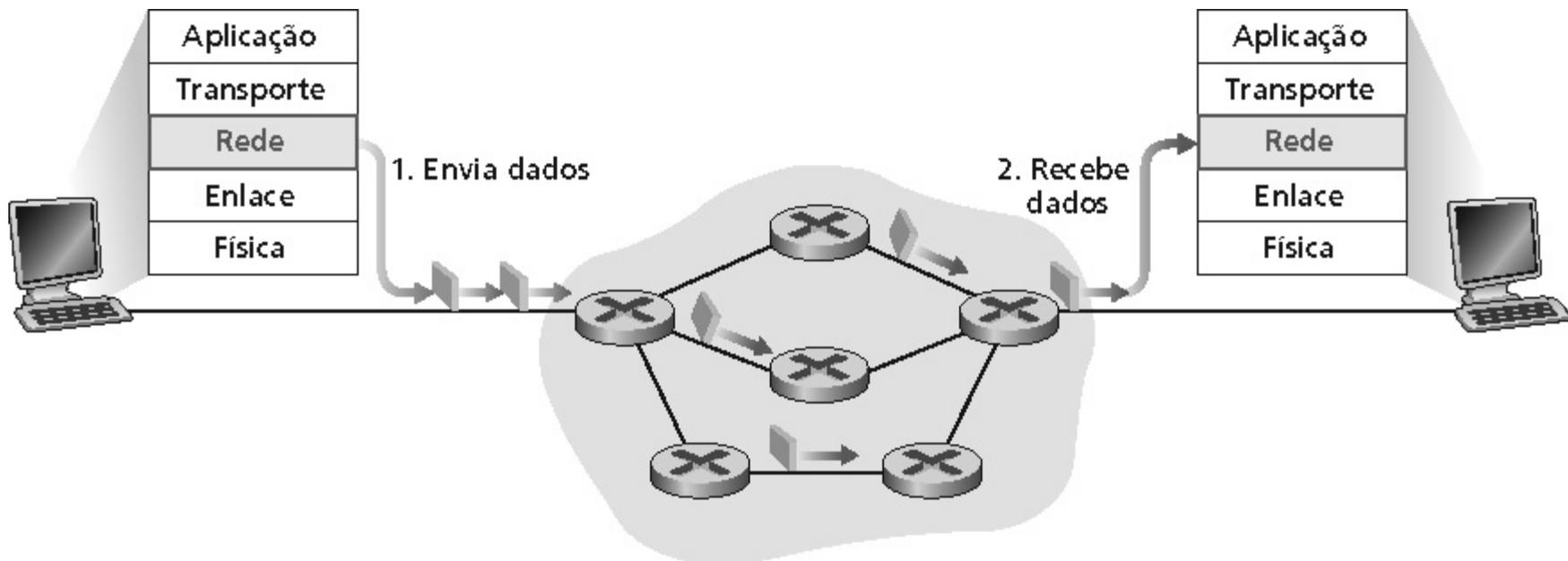
# 4 Circuitos virtuais: protocolos de sinalização

- Usado para estabelecer, manter e encerrar circuitos virtuais
- Usados em ATM, frame-relay e X-25
- Não é usado na Internet atualmente



# 4 Redes de datagrama

- Não existe estabelecimento de conexão na camada de rede
- Roteadores: não existe estado sobre conexões fim-a-fim
  - O conceito “conexão” não existe na camada de rede
- Pacotes são encaminhados pelo endereço do hospedeiro de destino
  - Pacotes para o mesmo destino podem seguir diferentes rotas



# 4 Tabela de comutação

4 bilhões de entradas possíveis

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

# 4 Encontro de prefixos maiores

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

## Exemplos

DA: 11001000 00010111 00010110 10100001

Qual interface?

DA: 11001000 00010111 00011000 10101010

Qual interface?

# 4 Datagrama *versus* circuito virtual

- **Internet**
- Dados trocados entre computadores
  - Serviço elástico, requisitos de atraso não críticos
- Sistemas finais inteligentes
  - Podem adaptar-se, realizar controle e recuperação de erros
  - A rede é simples; a complexidade fica nas pontas
- Muitos tipos de enlaces
  - Características diferentes
  - Difícil obter um serviço uniforme

## ATM

- Originário da telefonia
- Conversação humana:
  - Tempos estritos, exigências de confiabilidade
  - Necessário para serviço garantido
- Sistemas finais “burros”
  - Telefones
  - Complexidade dentro da rede

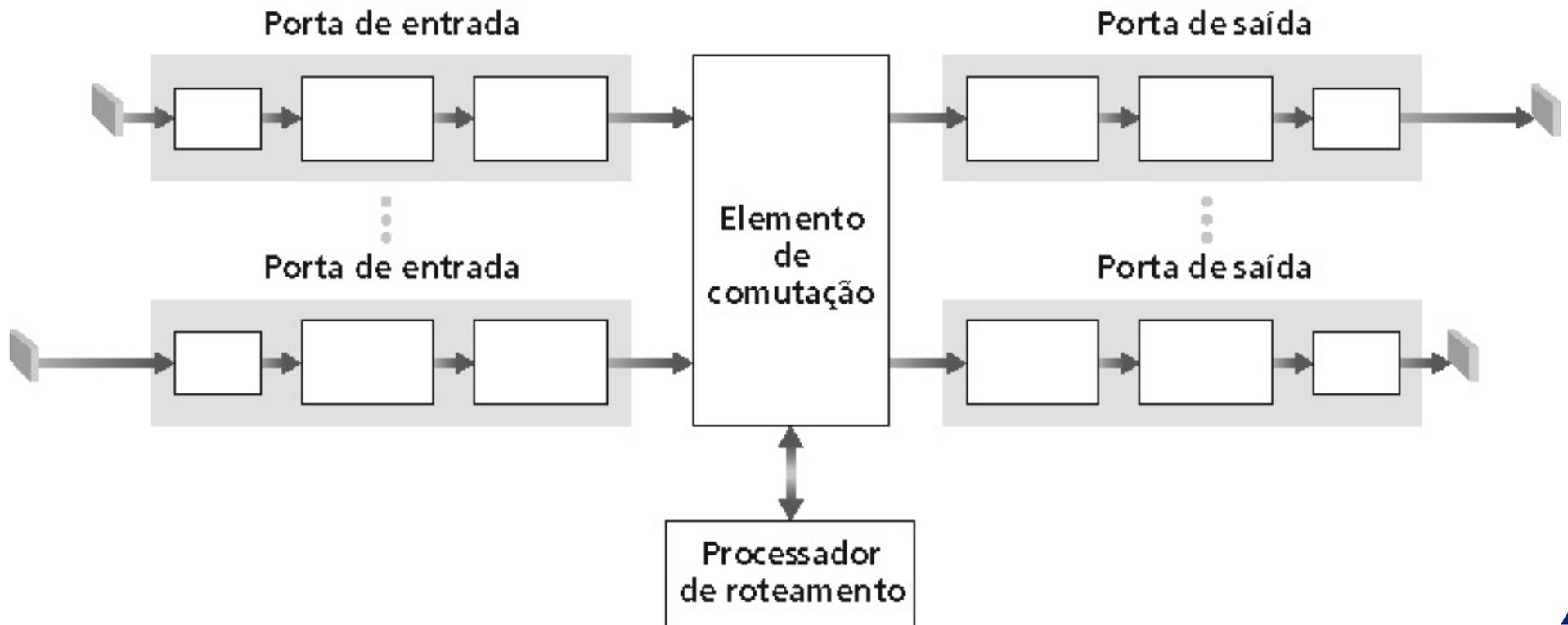
# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

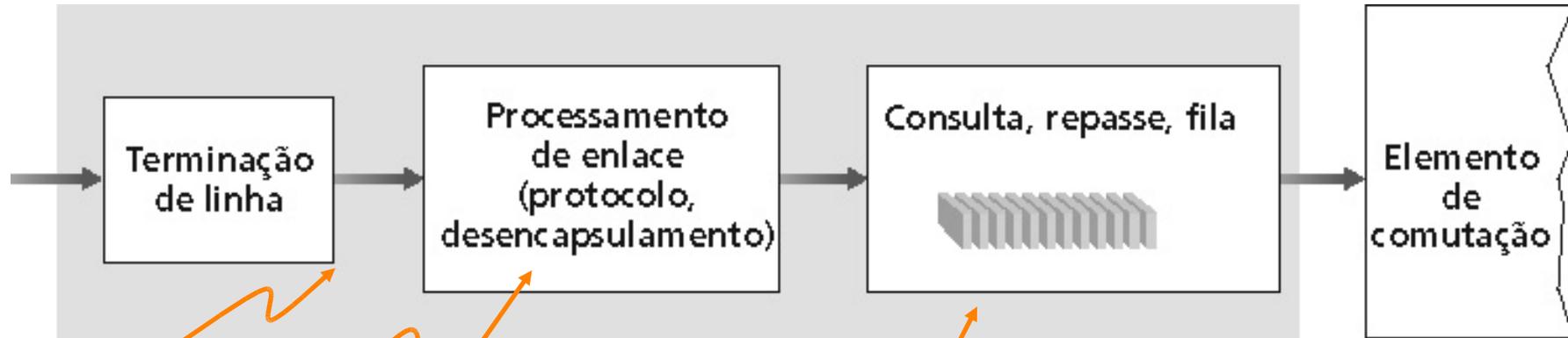
# 4 Visão geral da arquitetura do roteador

Duas funções-chave do roteador:

- Executar algoritmos/protocolos (RIP, OSPF, BGP)
- **Comutar** os datagramas do link de entrada para o link de saída



# 4 Funções da porta de entrada



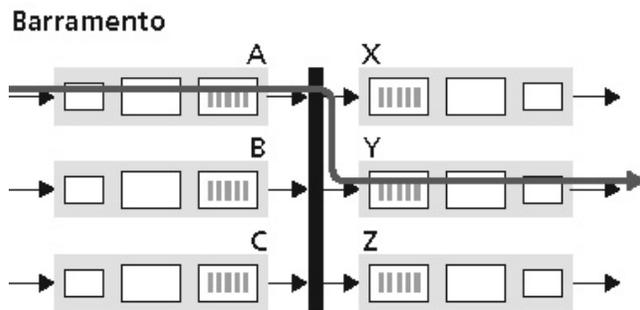
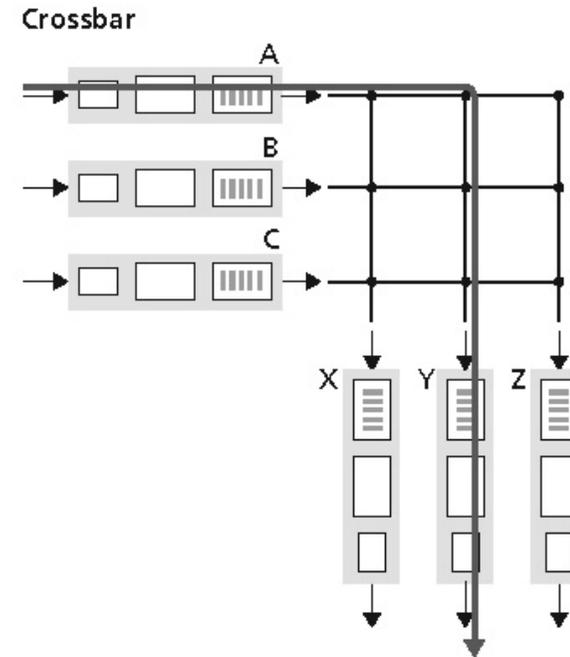
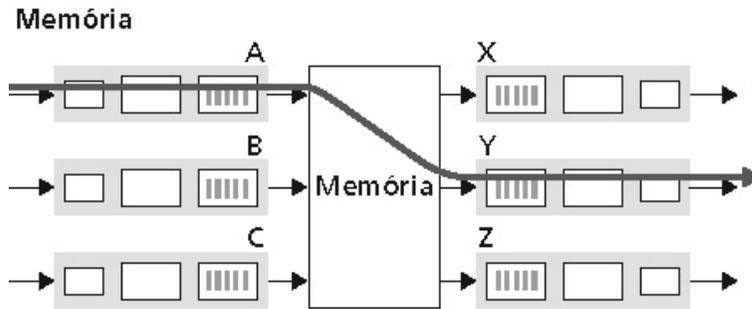
Camada física:  
recepção de bits

Camada de enlace:  
ex.: Ethernet  
(veja capítulo 5)

## Comutação descentralizada:

- Dado o destino do datagrama, procura a porta de saída usando a tabela de comutação na memória da porta de entrada
- Objetivo: completar o processamento da porta de entrada na 'velocidade da linha'
- Fila: se os datagramas chegam mais rápido do que a taxa de comutação para o switch

# 4 Três tipos de estrutura de comutação



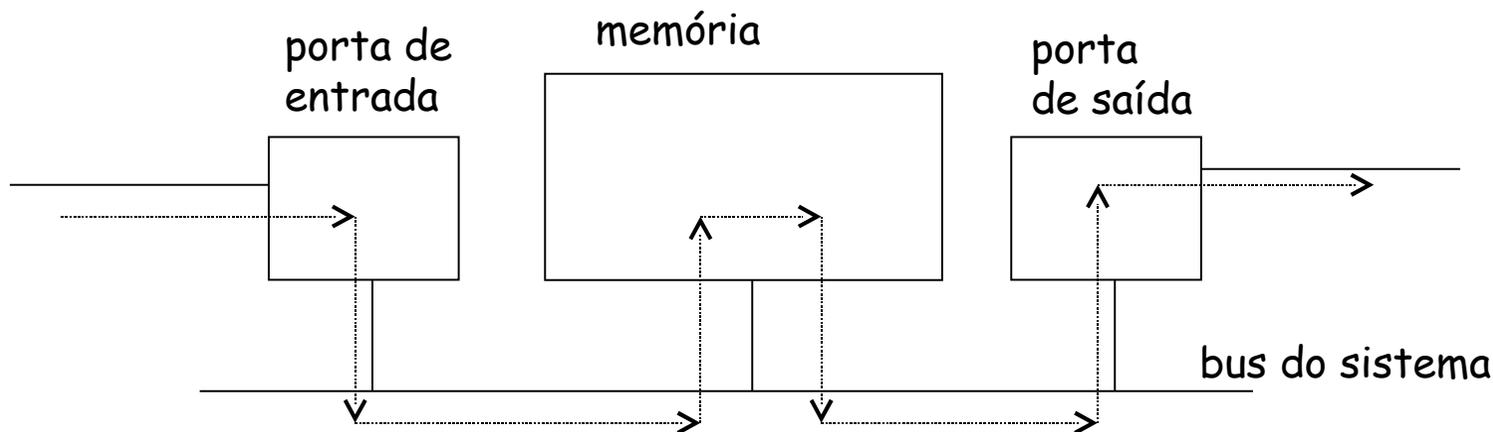
Legenda:



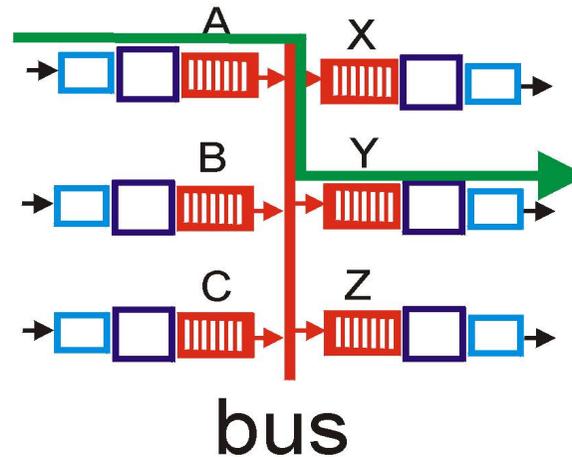
# 4 Comutação via memória

## Primeira geração de roteadores:

- Computadores tradicionais com comutação sob controle direto da CPU
- Pacote copiado para a memória do sistema
- Velocidade limitada pela largura de banda (2 bus cruzados por datagrama)
- Cisco 8500 Catalyst e Acceler 1200 da Bay Networks

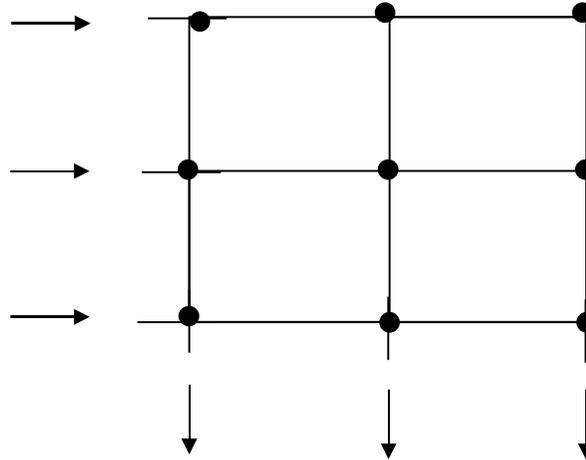


# 4 Comutação via bus



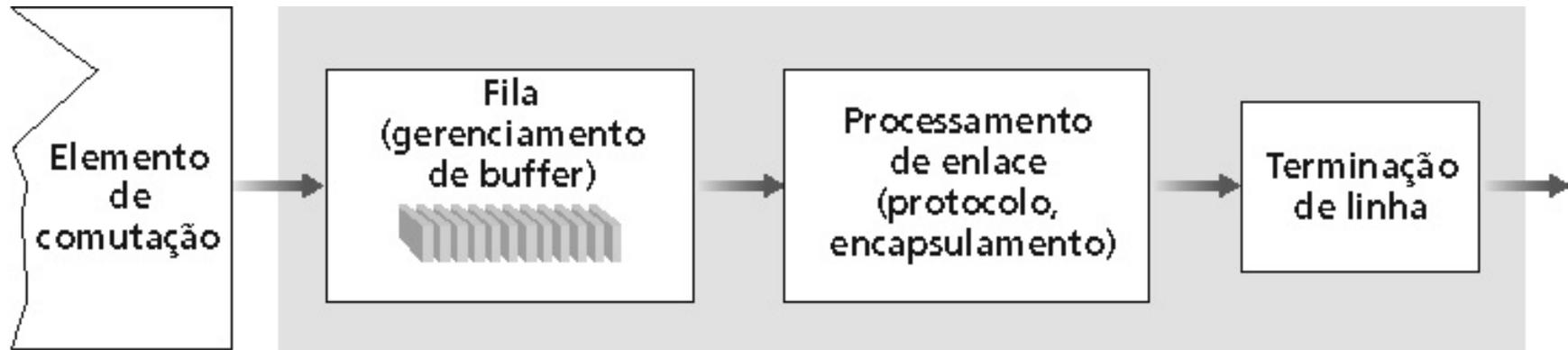
- Datagrama da memória da porta de entrada para a memória da porta de saída através de um bus compartilhado
- **Contenção do bus:** velocidade de comutação limitada pela largura de banda do bus
- Barramento de 1 Gbps, Cisco 1900: velocidade suficiente para roteadores de acesso e de empresas (não para roteadores regionais ou de backbone)

# 4 Comutação via rede de interconexão



- Supera as limitações de largura de banda do bus
- Redes de Banyan, outras redes de interconexão inicialmente desenvolvidas para conectar processadores em multiprocessamento
- Projeto avançado: fragmentar datagramas em células de tamanho fixo, comutar as células através do switch.
- Cisco 12000: comuta até 60 Gbps através da rede de interconexão

# 4 Portas de saída

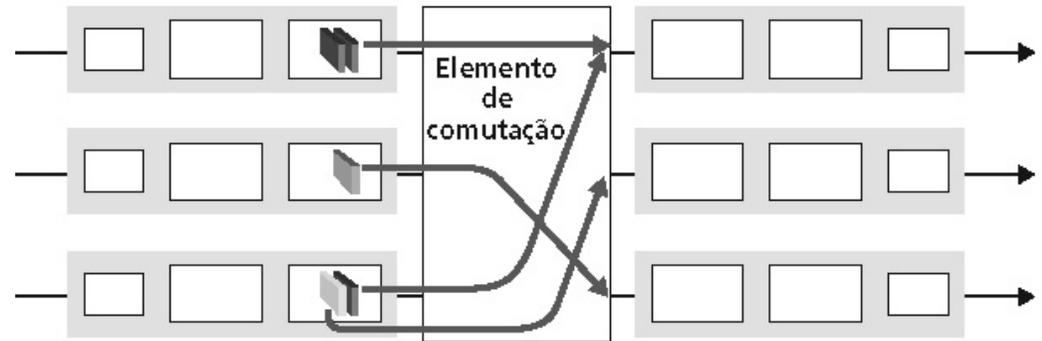


- **Buffering** necessário quando datagramas chegam do switch mais rápido do que a taxa de transmissão
- **Disciplina de agendamento** escolhe entre os datagramas na fila para transmissão

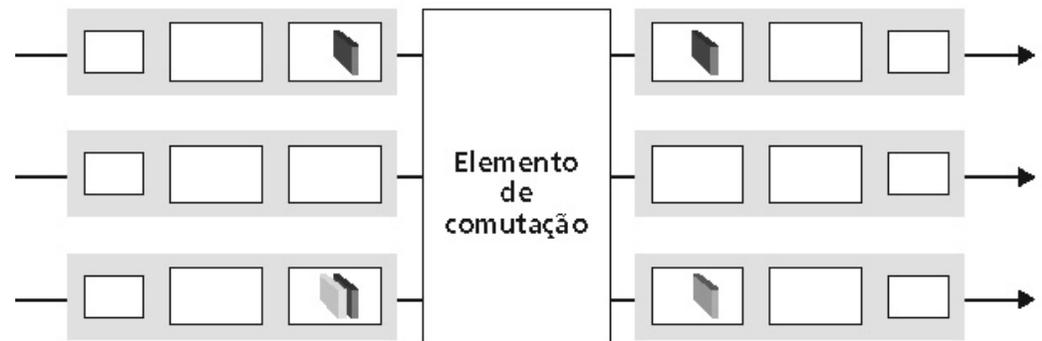
# 4 Enfileiramento na porta de saída

- Buffering: quando a taxa de chegada pelo switch excede a velocidade da linha de saída
- **Queueing (atraso) e perda devido ao buffer overflow da porta de saída!**

Contenção pela porta de saída no tempo  $t$  — um pacote escuro pode ser transferido



Pacote claro do último retângulo sofre bloqueio HOL

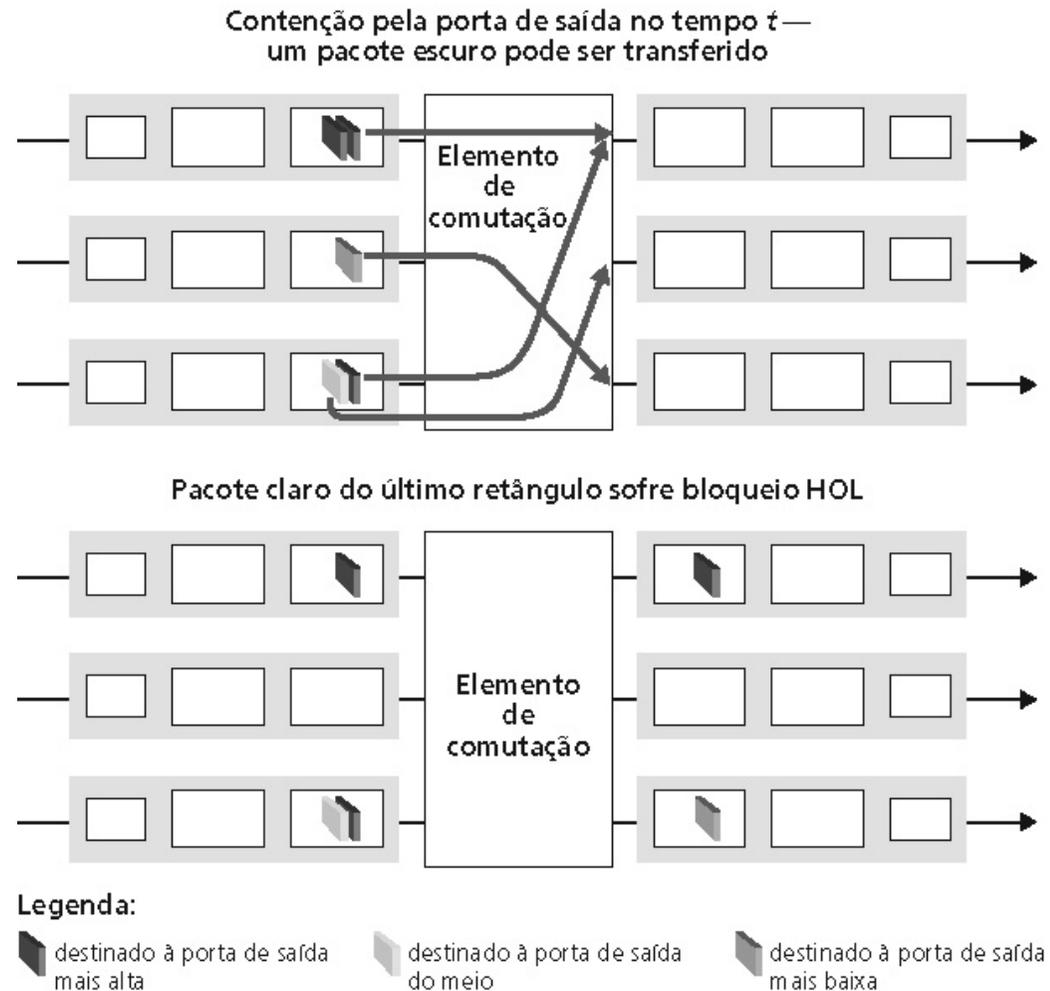


Legenda:

- ▀ destinado à porta de saída mais alta
- ▄ destinado à porta de saída do meio
- ▁ destinado à porta de saída mais baixa

# 4 Enfileiramento na porta de entrada

- Switch mais lento que as portas de entrada combinadas -> pode ocorrer filas na entrada
- **Bloqueio Head-of-the-Line (HOL):** datagrama na frente da fila impede os outros na fila de se moverem para adiante
- **Atraso e perda na fila devido ao overflow no buffer de entrada!**

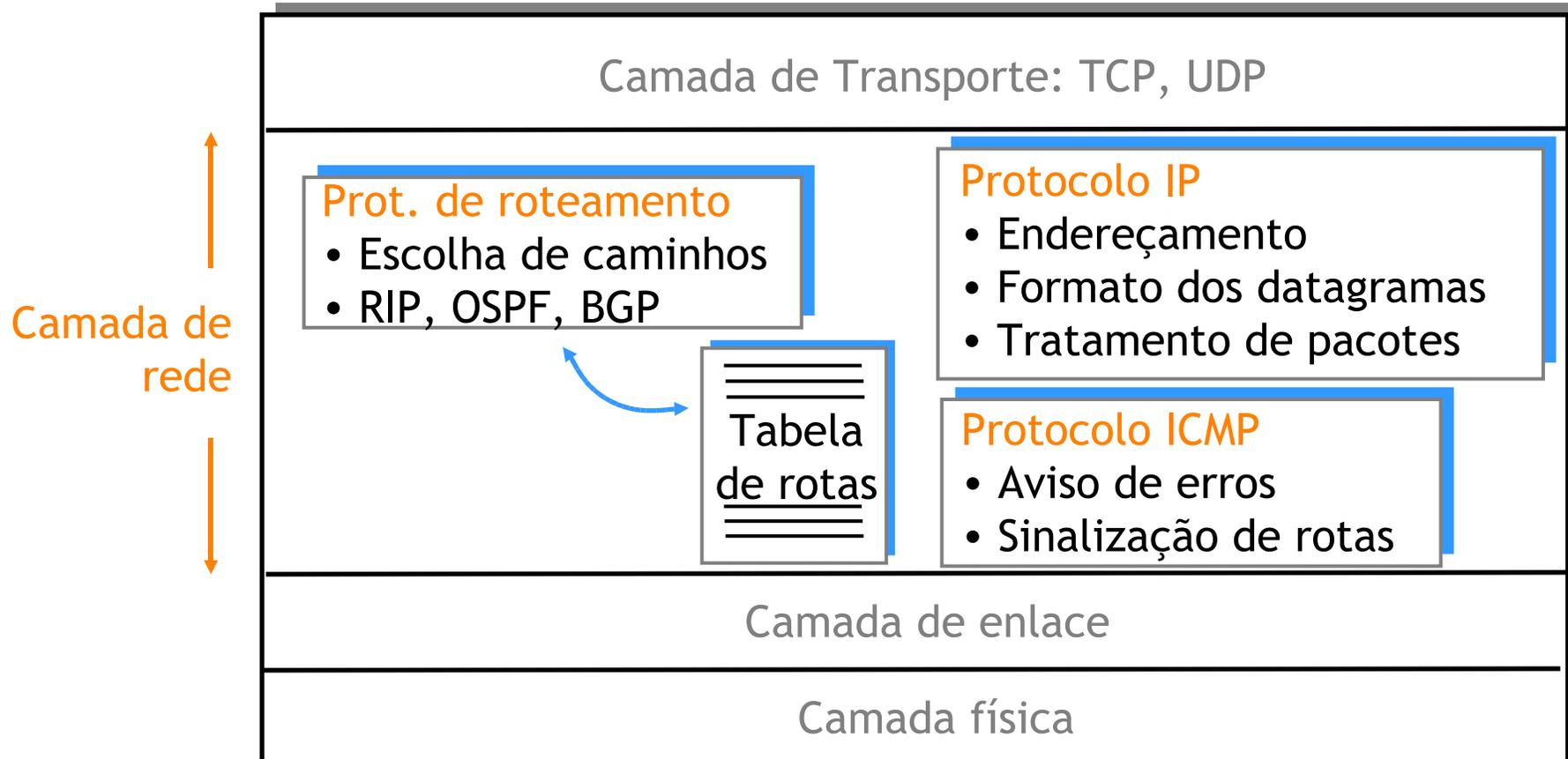


# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 A camada de rede

Entidade de rede em roteadores ou hospedeiros:



# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Formato do datagrama IP

versão do protocolo IP

tamanho do header (bytes)

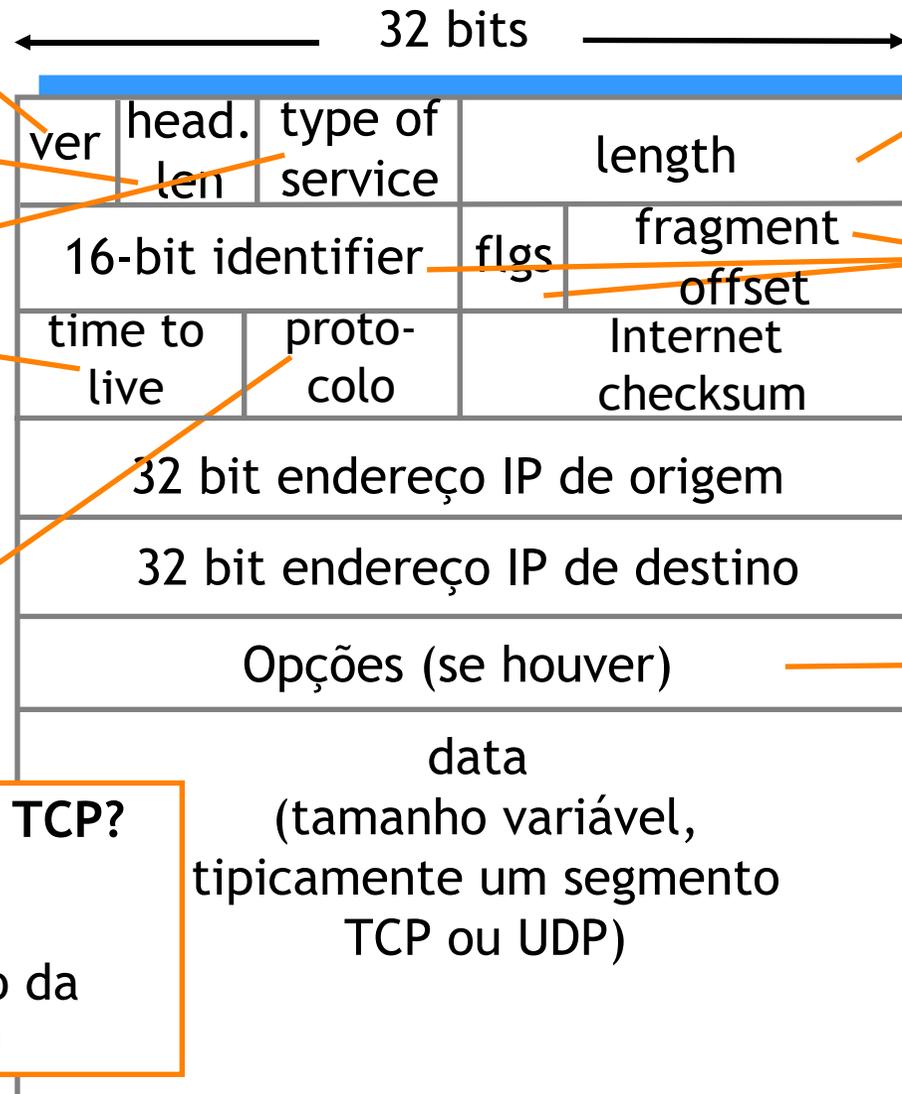
classe de serviço

número máximo de saltos (decrementado em cada roteador)

protocolo da camada superior com dados no datagrama

## Tamanho do cabeçalho TCP?

- 20 bytes do TCP
- 20 bytes do IP
- = 40 bytes + cabeçalho da camada de aplicação



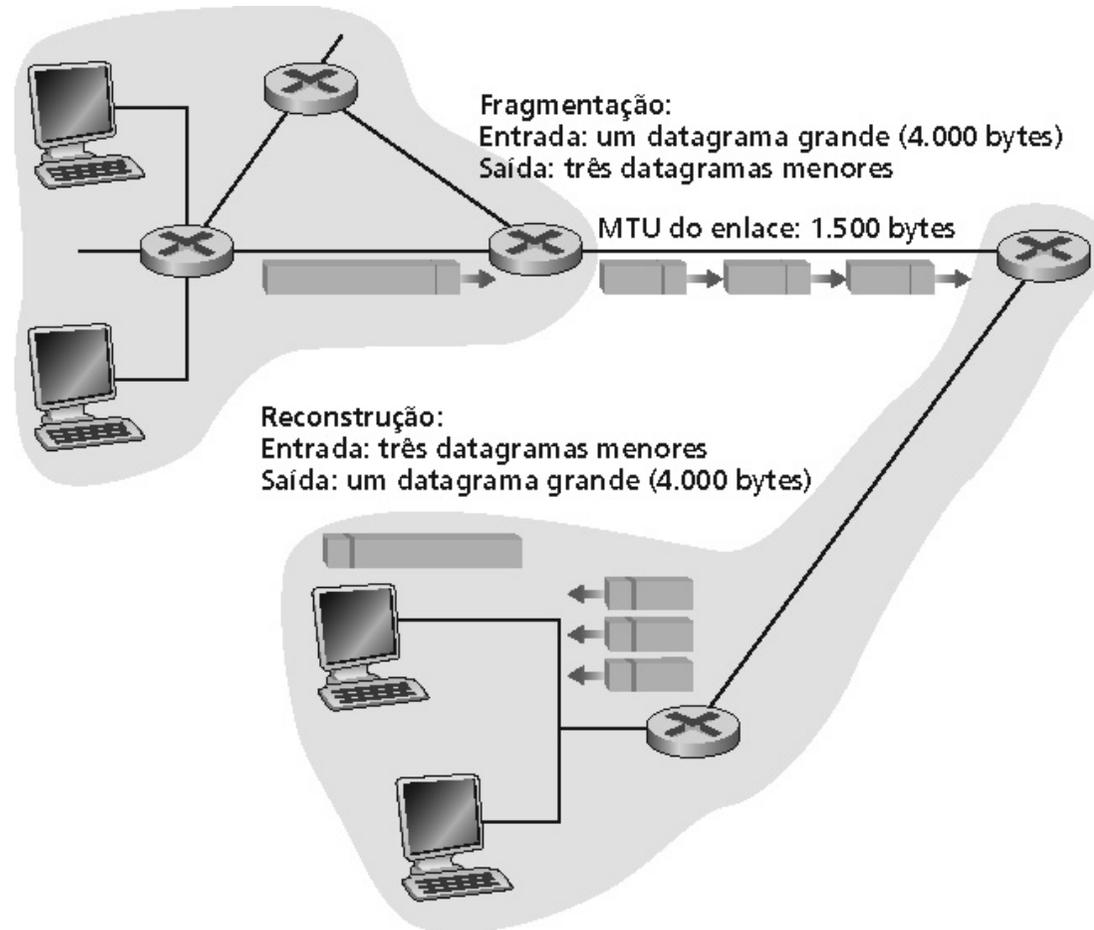
tamanho total do datagrama (bytes)

para fragmentação/remontagem

Ex.: marca de tempo, registro de rota, lista de roteadores a visitar

# 4 IP fragmentação e remontagem

- Enlaces de rede têm MTU (max. transfer size) – corresponde ao maior frame que pode ser transportado pela camada de enlace.
  - Tipos de enlaces diferentes possuem MTU diferentes (Ethernet: 1.518 bytes)
- Datagramas IP grandes devem ser divididos dentro da rede (fragmentados)
  - Um datagrama dá origem a vários datagramas
  - “Remontagem” ocorre apenas no destino final
  - O cabeçalho IP é usado para identificar e ordenar datagramas relacionados



# 4 IP fragmentação e remontagem

- Exemplo
- datagrama de 4000 bytes
- MTU = 1500 bytes

	tamanho	ID	fragflag	offset	
	=4000	=x	=0	=0	

Um grande datagrama se torna vários datagramas menores

1480 bytes no campo de dados

offset =  
1480/8

	tamanho	ID	fragflag	offset	
	=1500	=x	=1	=0	

	tamanho	ID	fragflag	offset	
	=1500	=x	=1	=1480	

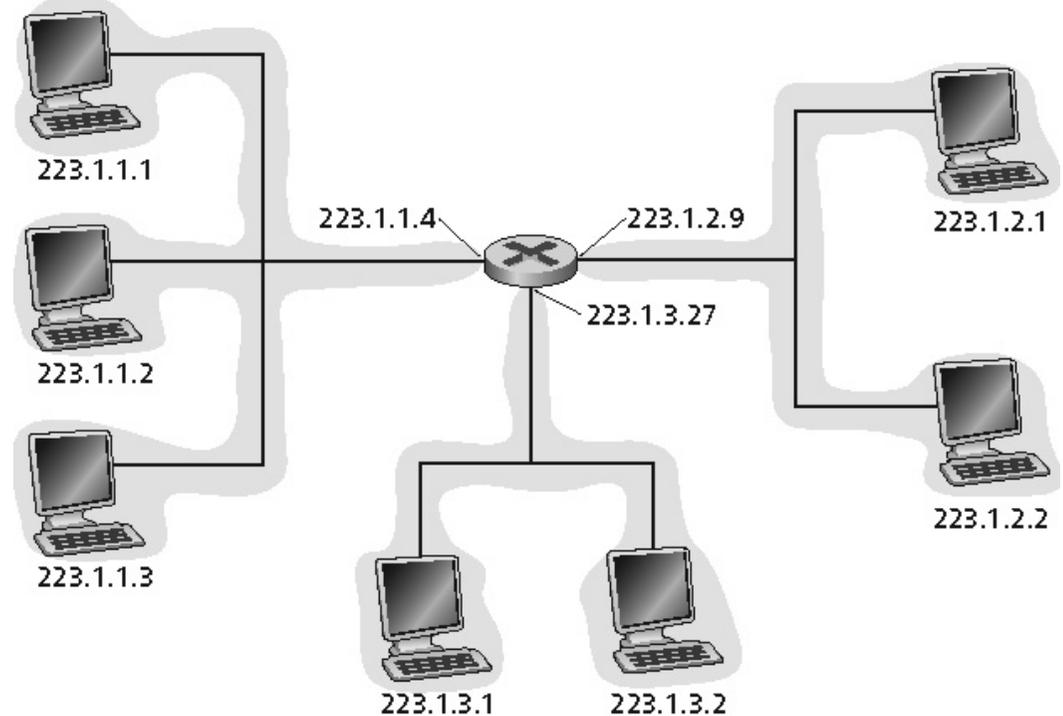
	tamanho	ID	fragflag	offset	
	=1040	=x	=0	=2960	

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Endereçamento IP : Introdução

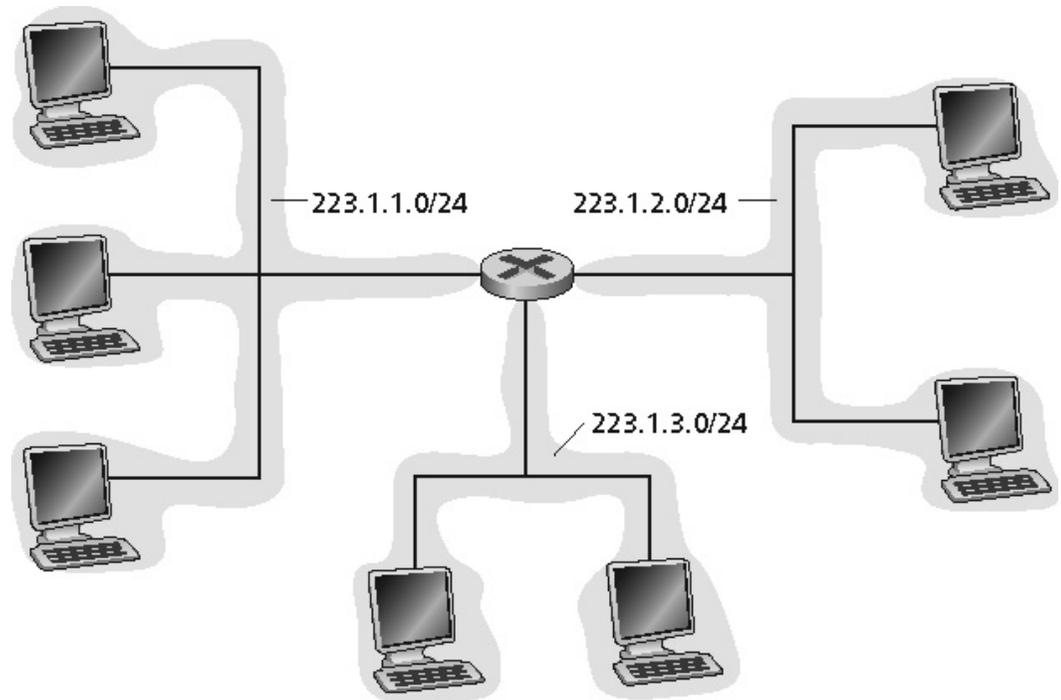
- **Endereço IP:** identificador de 32 bits para **interfaces** de roteadores e hospedeiros
- **Interface:** conexão entre roteador ou hospedeiro e enlace físico
  - Roteador tem tipicamente múltiplas interfaces
  - Hospedeiros podem ter múltiplas interfaces
  - Endereços IP são associados com interfaces, não com o hospedeiro ou com o roteador



$$223.1.1.1 = \underbrace{11011111}_{223} \underbrace{00000001}_1 \underbrace{00000001}_1 \underbrace{00000001}_1$$

# 4 Sub-redes

- **Endereço IP:**
  - Parte da sub-rede (bits de ordem superior)
  - Parte do hospedeiro (bits de ordem inferior)
- **O que é uma sub-rede?**
  - Interfaces de dispositivo com a mesma parte de sub-rede do endereço IP
  - Podem alcançar fisicamente uns aos outros sem intervenção de roteador

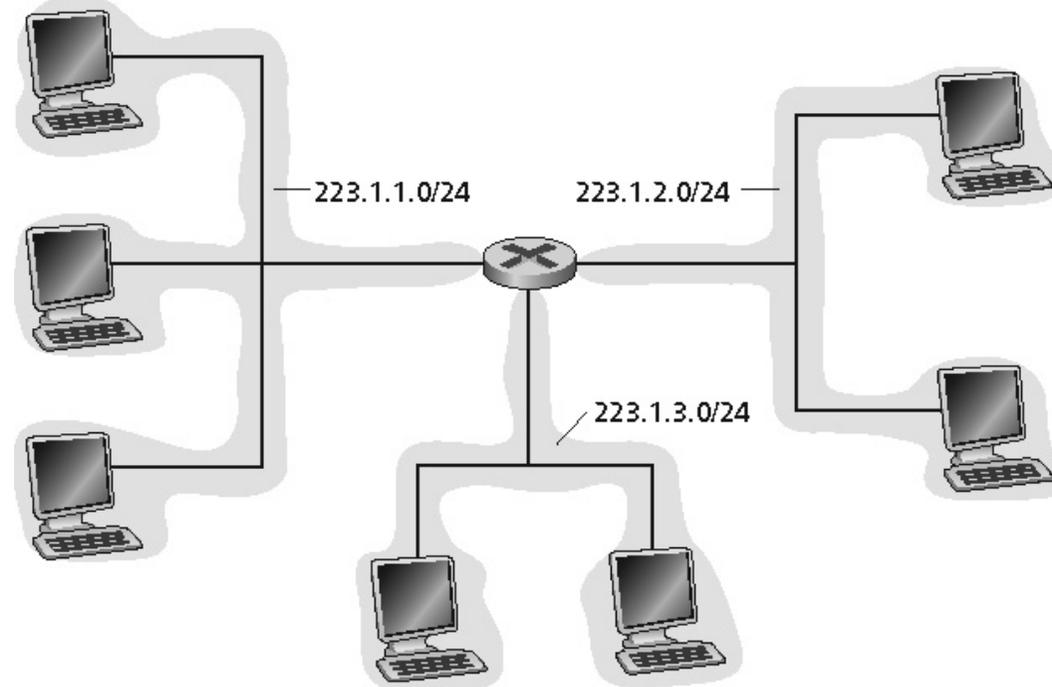


rede consistindo de 3 sub-redes

# 4 Sub-redes

## Receita

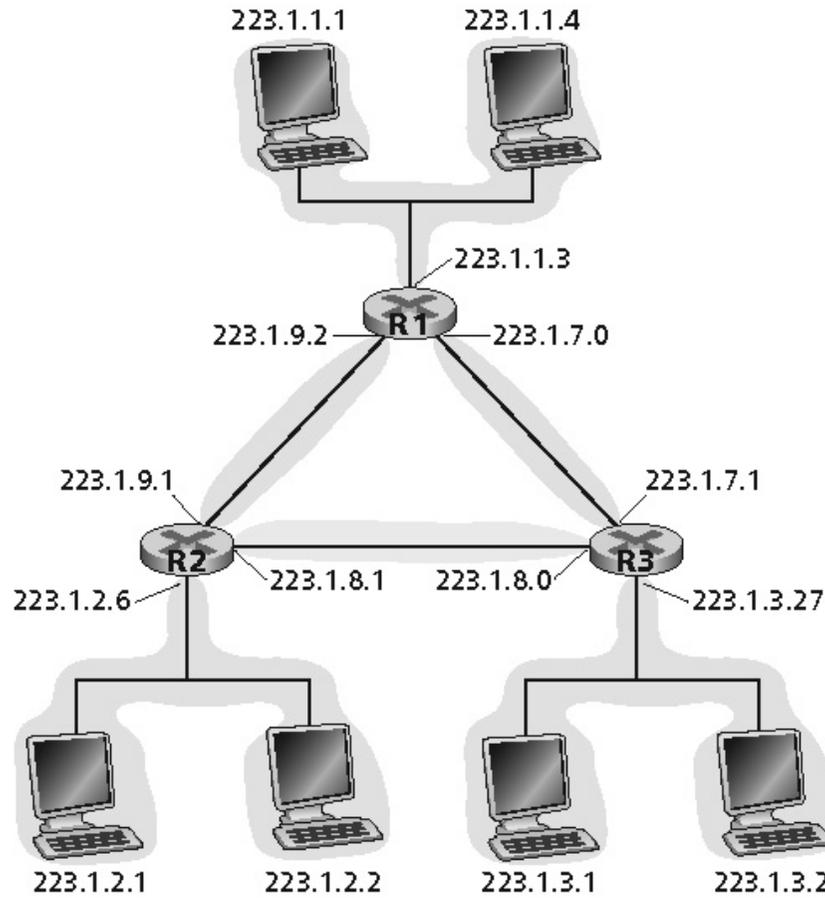
- Para determinar as sub-redes, destaque cada interface de seu hospedeiro ou roteador, criando ilhas de redes isoladas. Cada rede isolada é considerada uma **sub-rede**.



máscara de sub-rede: /24

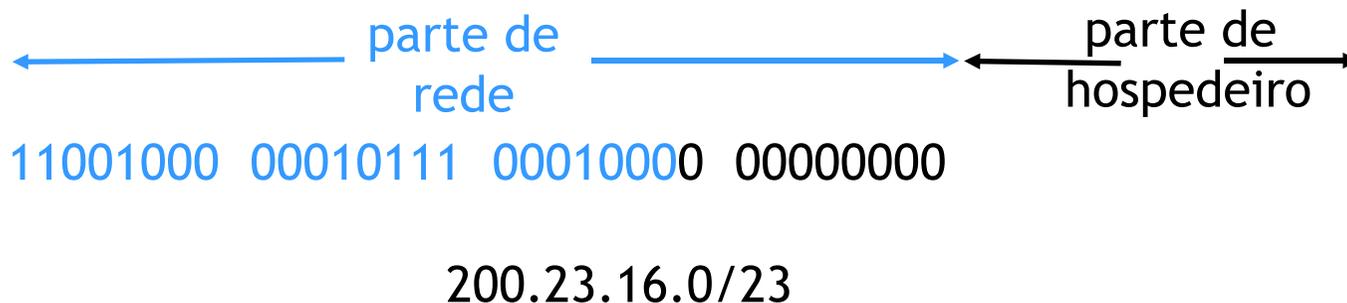
# 4 Sub-redes

Quantas?



# 4 Endereçamento IP : CIDR

- **CIDR**: **C**lassless **I**nter**D**omain **R**outing
  - A porção de endereço de rede tem tamanho arbitrário
  - Formato do endereço: **a.B.C.D/x**, em que **x** é o número de bits na parte de rede do endereço



# 4 Como obter um endereço IP

**P.:** Como um hospedeiro obtém endereço IP ?

- Definido pelo administrador do sistema
  - Wintel: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- **DHCP:** dynamic host configuration protocol: obtém dinamicamente endereços IP de um servidor
  - “plug-and-play”
  - (mais no próximo capítulo)

# 4 Como obter um endereço IP

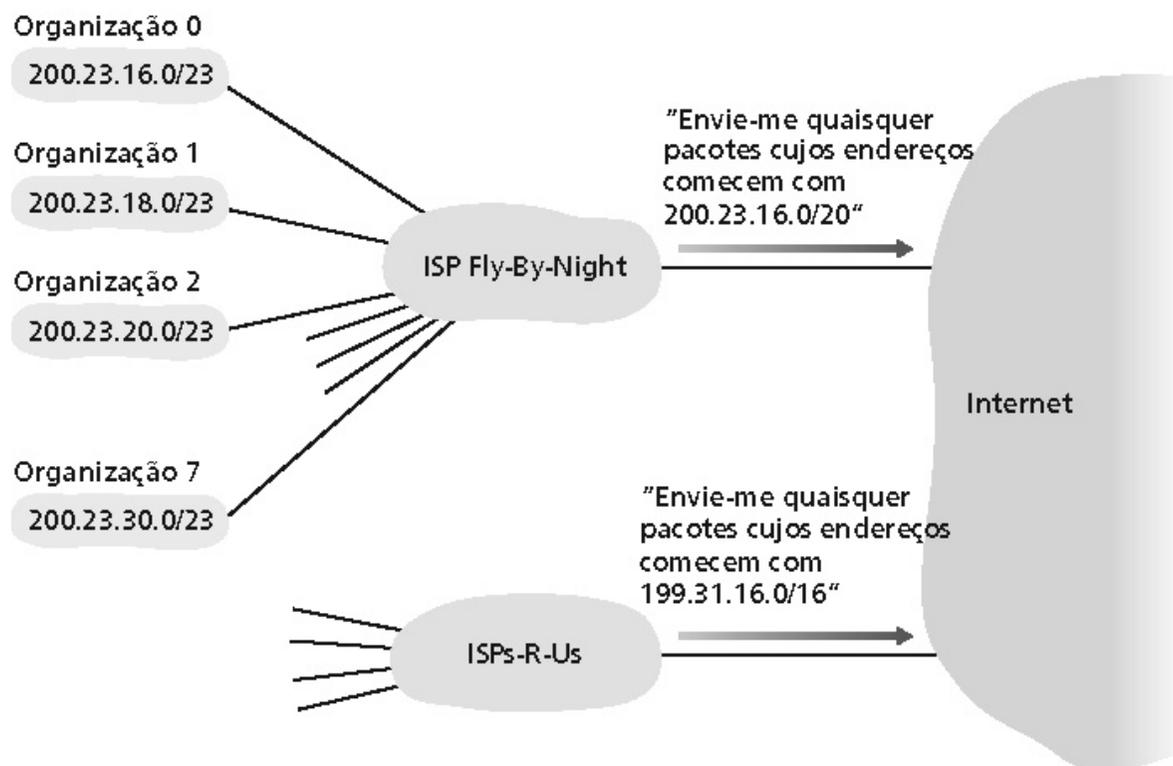
**P.:** Como uma **rede** obtém a parte de sub-rede do endereço IP?

**R.:** Obtém a porção alocada no espaço de endereço do seu provedor ISP

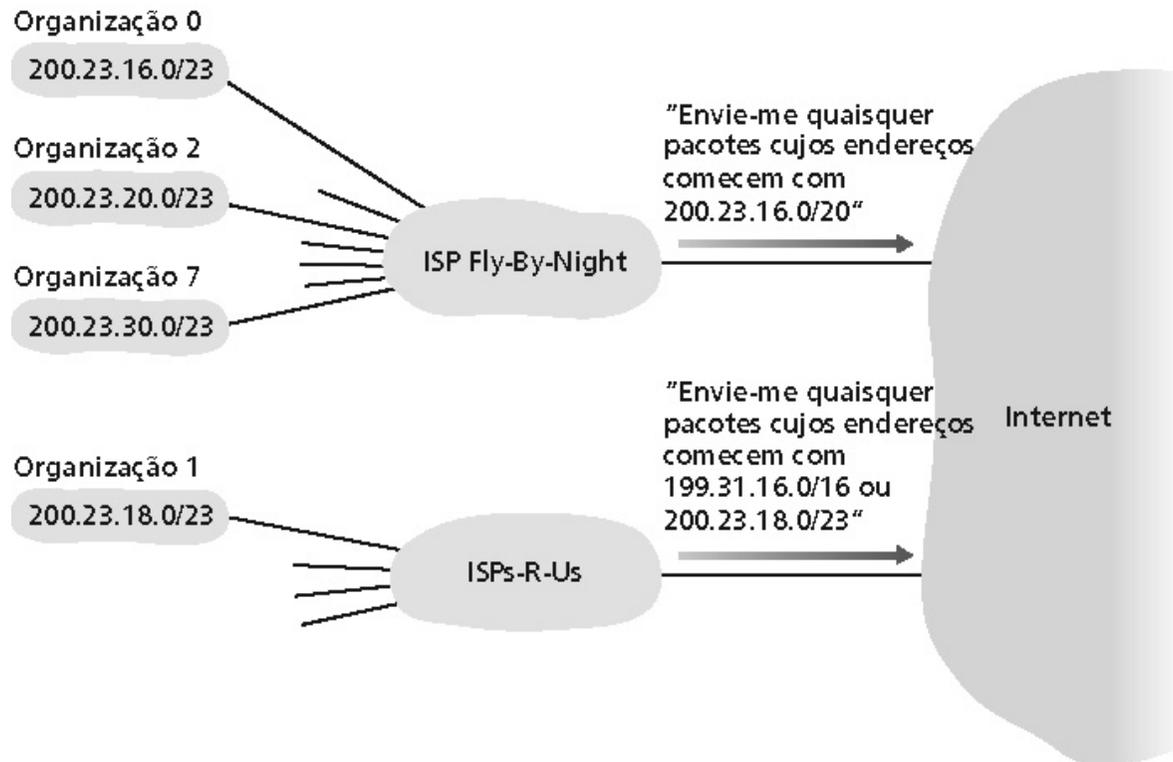
bloco do ISP	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/20
Organização 0	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/23
Organização 1	<u>11001000</u>	<u>00010111</u>	<u>00010010</u>	00000000	200.23.18.0/23
Organização 2	<u>11001000</u>	<u>00010111</u>	<u>00010100</u>	00000000	200.23.20.0/23
...	...	...	...	...	...
Organização 7	<u>11001000</u>	<u>00010111</u>	<u>00011110</u>	00000000	200.23.30.0/23

# 4 Endereçamento hierárquico: agregação de rotas

O endereçamento hierárquico permite uma propagação de rotas mais eficiente:



# 4 Endereçamento hierárquico: rotas mais específicas



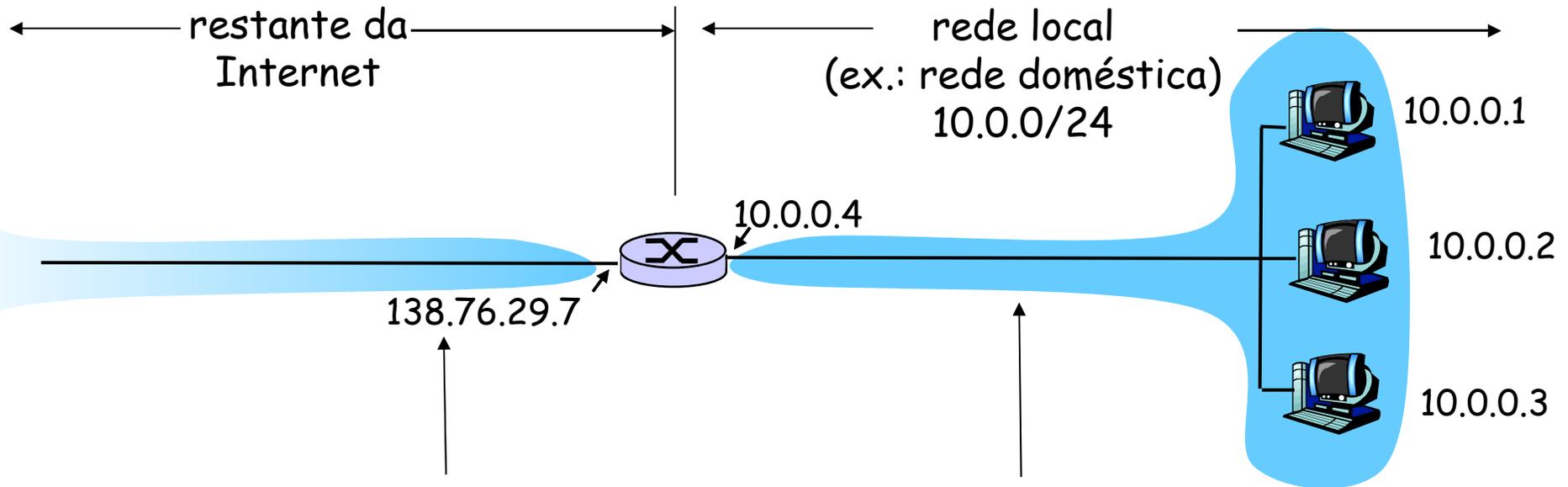
# 4 Endereçamento IP : a última palavra...

P.: Como o ISP obtém seu bloco de endereço?

R.: ICANN: internet corporation for assigned names and numbers

- Aloca endereços
- Gerencia DNS
- Atribui nomes de domínios e resolve disputas

# 4 NAT: Network Address Translation



**todos os** datagramas que **saem** da rede local possuem o **mesmo** e único endereço IP do NAT de origem: 138.76.29.7, números diferentes de portas de origem

datagramas com origem ou destino nesta rede possuem endereço 10.0.0/24 para origem, destino (usualmente)

# 4 NAT: Network Address Translation

- **Motivação:** redes locais podem utilizar apenas um endereço IP:
  - Não é preciso alocar uma gama de endereços do ISP: apenas um endereço IP é usado para todos os dispositivos
  - Podem-se alterar os endereços dos dispositivos na rede local sem precisar notificar o mundo exterior
  - Pode-se mudar de ISP sem alterar os endereços dos dispositivos na rede local
  - Dispositivos da rede local não são explicitamente endereçáveis ou visíveis pelo mundo exterior (um adicional de segurança).

# 4 NAT: Network Address Translation

**Implementação:** o roteador NAT deve:

**Datagramas que saem: substituir** (endereço IP de origem, porta #) de cada datagrama para (endereço IP do NAT, nova porta #)

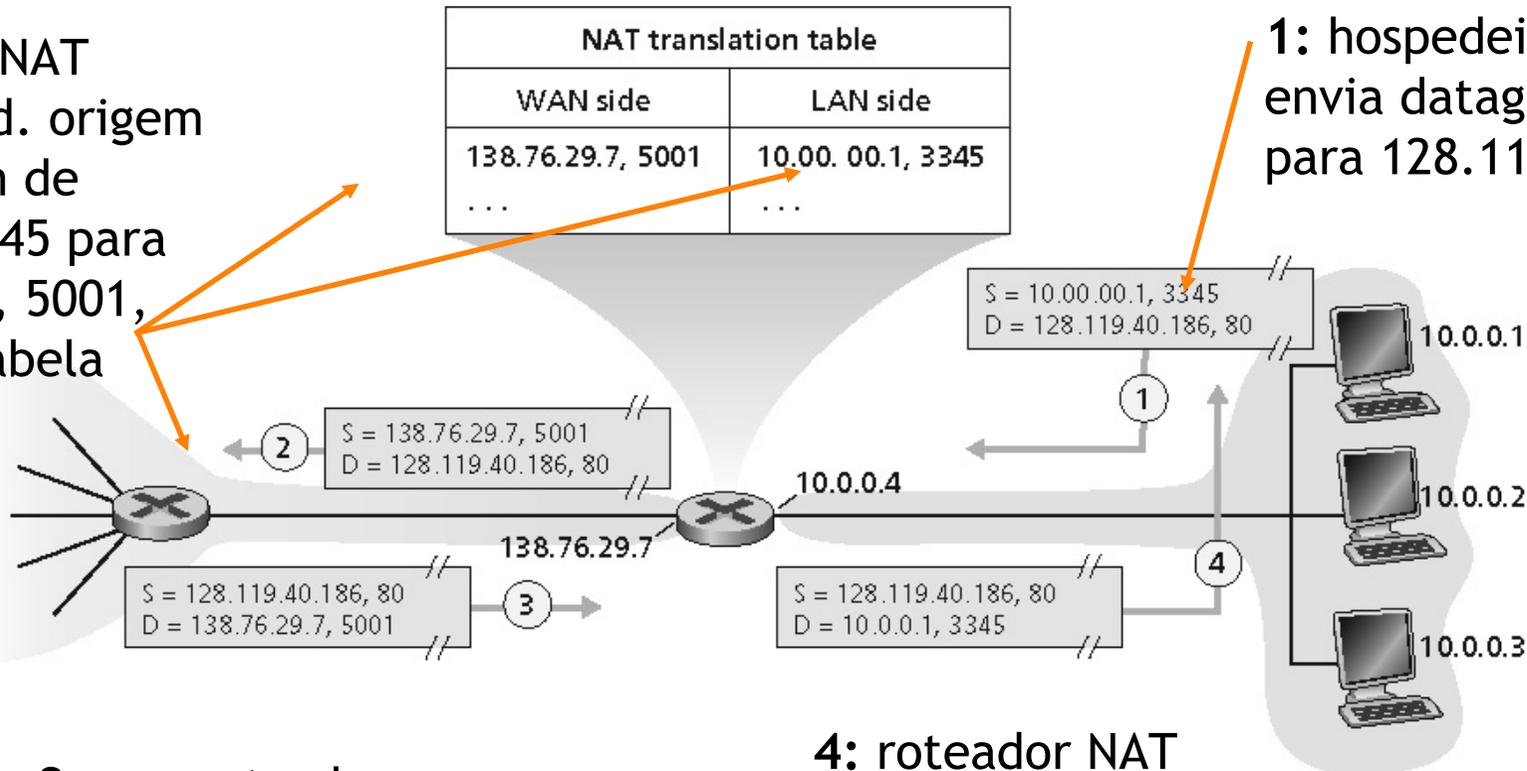
. . . clientes/servidores remotos responderão usando (endereço IP do NAT, nova porta #) como endereço de destino.

- **Lembrar (na tabela de tradução do NAT)** cada (endereço IP de origem, porta #) para o par de tradução (endereço IP do NAT, nova porta #).
- **Datagramas que chegam: substituir** (endereço IP do NAT, nova porta #) nos campos de destino de cada datagrama pelos correspondentes (endereço IP de origem, porta #) armazenados da tabela NAT



# 4 NAT: Network Address Translation

2: roteador NAT substitui end. origem do datagram de 10.0.0.1, 3345 para 138.76.29.7, 5001, atualiza a tabela



1: hospedeiro 10.0.0.1 envia datagrama para 128.119.40, 80

3: resposta chega endereço de destino: 138.76.29.7, 5001

4: roteador NAT substitui o endereço de destino do datagrama de 138.76.29.7, 5001 para 10.0.0.1, 3345

# 4 NAT: Network Address Translation

- Campo número de porta com 16 bits:
  - 60.000 conexões simultâneas com um único endereço de LAN
- NAT é controverso:
  - Roteadores deveriam processar somente até a camada 3
  - Violação do argumento fim-a-fim
  - A possibilidade de NAT deve ser levada em conta pelos desenvolvedores de aplicações, ex., aplicações P2P
  - A escassez de endereços deveria ser resolvida pelo IPv6

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 ICMP: Internet Control Message Protocol

- Usado por computadores e roteadores para troca de informação de controle da camada de rede
  - Error reporting: hospedeiro, rede, porta ou protocolo
  - Echo request/reply (usado pela aplicação ping)
- Transporte de mensagens:
  - Mensagens ICMP transportadas em datagramas IP
- **ICMP message:** tipo, código, mais primeiros 8 bytes do datagrama IP que causou o erro

Tipo	Código	Descrição
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

# 4 Traceroute e ICMP

- O transmissor envia uma série de segmentos UDP para o destino
  - O 1º possui TTL = 1
  - O 2º possui TTL = 2 etc.
  - nº de porta improvável
- Quando o enésimo datagrama chega ao enésimo roteador:
  - O roteador descarta o datagrama
  - E envia à origem uma mensagem ICMP (type 11, code 0)
  - A mensagem inclui o nome do roteador e o endereço IP
- Quando a mensagem ICMP chega, a origem calcula o RTT
- O traceroute faz isso três vezes
- **Critério de interrupção**
- O segmento UDP finalmente chega ao hospedeiro de destino
- O destino retorna o pacote ICMP “hospedeiro unreachable” (type 3, code 3)
- Quando a origem obtém esse ICMP, ela pára.

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Cabeçalho IPv6

- **Motivação inicial:** o espaço de endereços de 32 bits está próximo de ser completamente alocado
- Motivação adicional:
  - Melhorar o formato do header para permitir maior velocidade de processamento e de transmissão
  - Mudanças no header para incorporar mecanismos de controle de QOS
  - **Formato do datagrama IPV:**
  - Cabeçalho fixo de 40 bytes
  - Não é permitida fragmentação

# 4 Cabeçalho IPv6

**Priority:** permitir definir prioridades diferenciadas para vários fluxos de informação

**Flow label:** identifica datagramas do mesmo “fluxo.” (conceito de “fluxo” não é bem definido).

**Next header:** identifica o protocolo da camada superior ou um header auxiliar



# 4 Outras mudanças do IPv4

- **Checksum**: removido inteiramente para reduzir o tempo de processamento em cada salto
- **Options**: são permitidas, mas são alocadas em cabeçalhos suplementares, indicados pelo campo “Next header”
- **ICMPv6**: nova versão de ICMP
  - Tipos de mensagens adicionais, ex.: “Packet Too Big”
  - Funções de gerenciamento de grupos multicast

# 4 Transição do IPv4 para IPv6

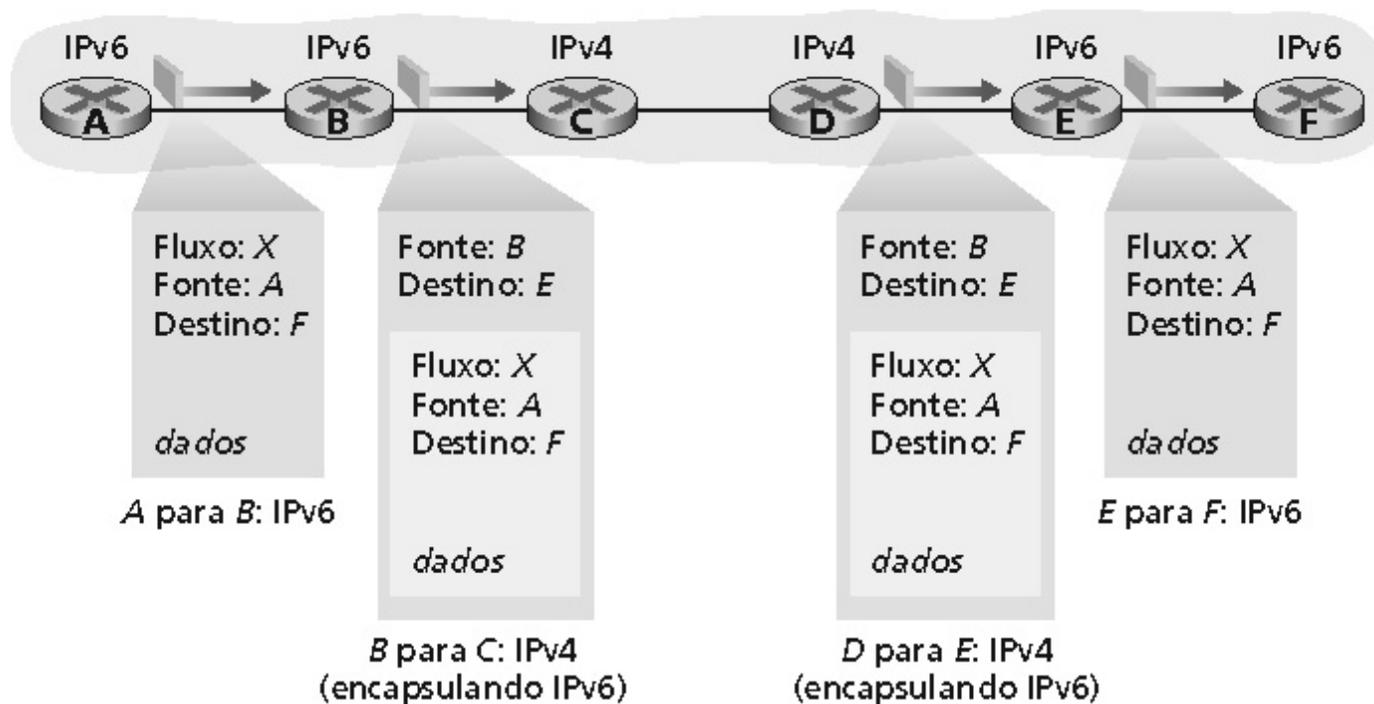
- Nem todos os roteadores poderão ser atualizados simultaneamente
  - Não haverá um dia da vacinação
  - Como a rede irá operar com roteadores mistos de IPV4 e IPV6?
- **Tunelamento**: IPv6 transportado dentro de pacotes IPv4 entre roteadores IPv4

# 4 Tunelamento

Visão lógica



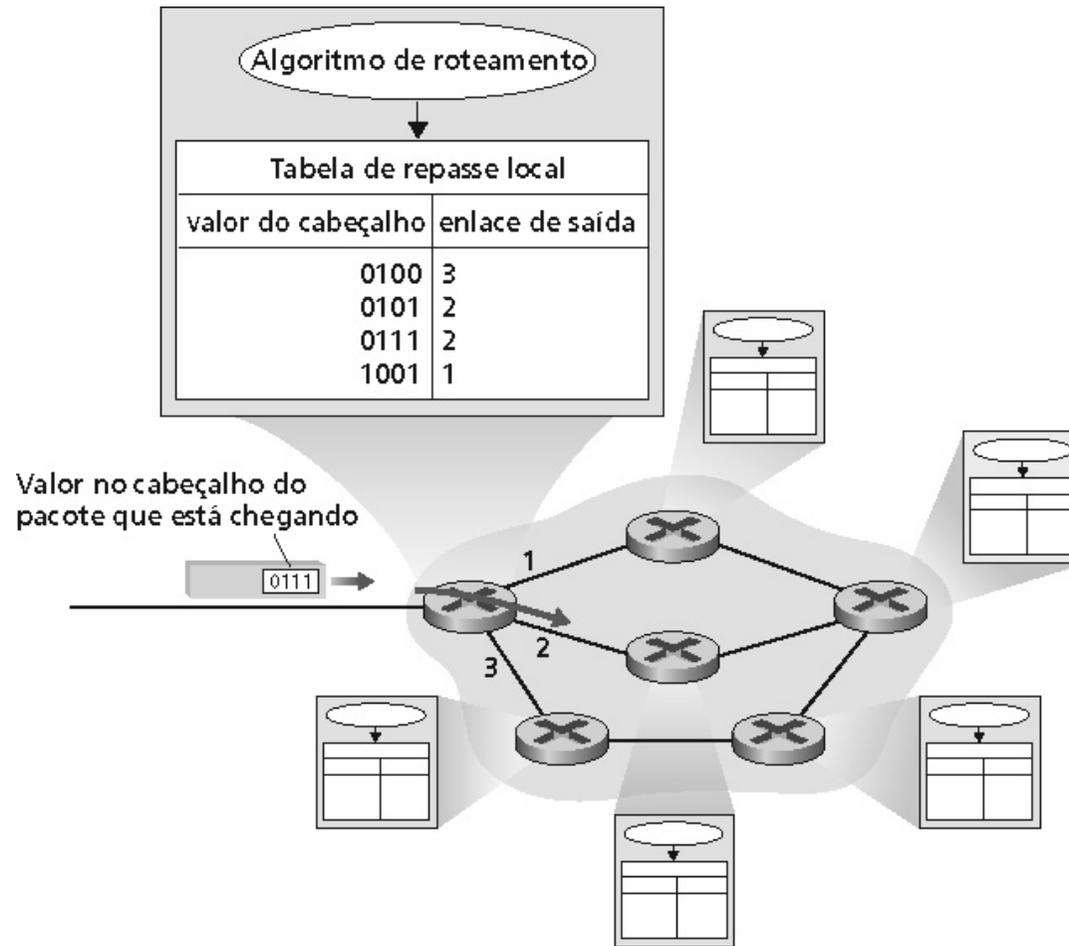
Visão física



# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Interação entre roteamento e comutação



# 4 Abstração do gráfico

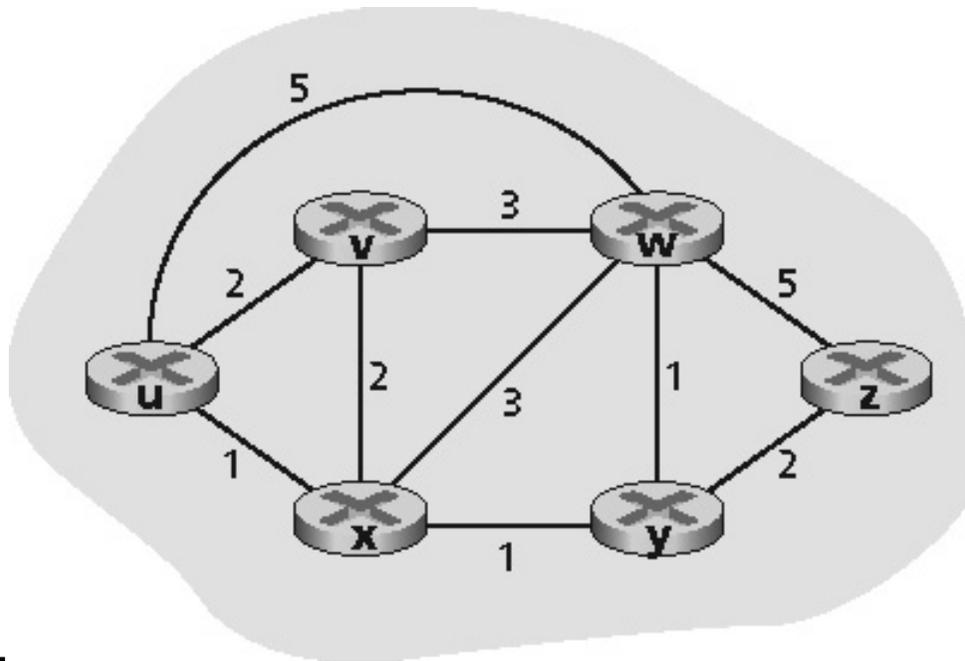


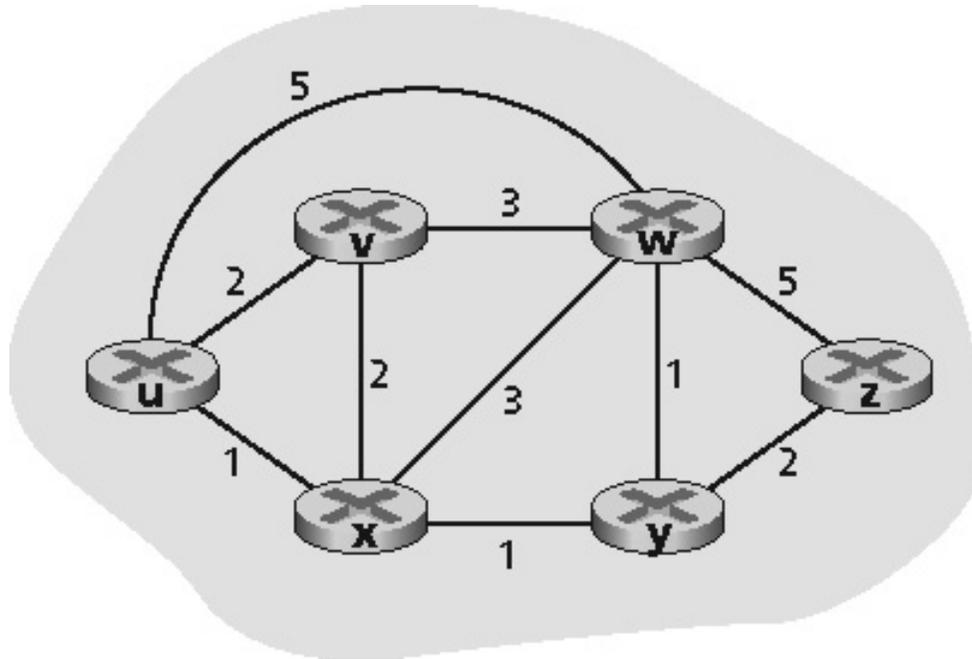
Gráfico:  $G = (N,E)$

$N =$  conjunto de roteadores =  $\{ u, v, w, x, y, z \}$

$E =$  conjunto de links =  $\{ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) \}$

**Lembrete: A abstração de gráfico é útil em outros contextos de rede. Exemplo: P2P, em que  $N$  é o conjunto de peers e  $E$  é o conjunto de conexões TCP**

# 4 Abstração do gráfico: custo



- $c(x, x')$  = custo do link  $(x, x')$ 
  - ex.,  $c(w, z) = 5$
- Custo poderia ser sempre 1, ou inversamente relacionado à largura de banda ou ao congestionamento

Custo do caminho  $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

**Questão: Qual é o caminho de menor custo entre u e z ?**

**Algoritmo de roteamento: algoritmo que encontra o caminho de menor custo**

# 4 Classificação dos algoritmos de roteamento

## Informação global ou descentralizada

### Global:

- Todos os roteadores têm informações completas da topologia e do custos dos enlaces
- Algoritmos “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
- Algoritmos “distance vector”

## Estático ou dinâmico?

### Estático:

- As rotas mudam lentamente ao longo do tempo

### Dinâmico:

- As rotas mudam mais rapidamente
  - Podem responder a mudanças no custo dos enlaces
  - Atualizações periódicas

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Algoritmo de roteamento link-state

- **Algoritmo de Dijkstra**
- Topologia de rede e custo dos enlaces são conhecidos por todos os nós
  - Implementado via “link state broadcast”
  - Todos os nós têm a mesma informação
- Computa caminhos de menor custo de um nó (fonte) para todos os outros nós
  - Fornece uma **tabela de roteamento** para aquele nó
- Convergência: após  $k$  iterações, conhece o caminho de menor custo para  $k$  destinos

## Notação:

- $C(i,j)$ : custo do enlace do nó  $i$  ao nó  $j$ . Custo é infinito se não houver ligação entre  $i$  e  $j$
- $D(v)$ : valor atual do custo do caminho da fonte ao destino  $V$
- $P(v)$ : nó predecessor ao longo do caminho da fonte ao nó  $v$ , isto é, antes do  $v$
- $N'$ : conjunto de nós cujo caminho de menor custo é definitivamente conhecido

# 4 Algoritmo de Dijkstra

1 **Inicialização:**

2  $N' = \{u\}$

3 para todos os nós  $v$

4 se  $v$  é adjacente a  $u$

5 então  $D(v) = c(u,v)$

6 senão  $D(v) = \infty$

7

8 **Loop**

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

10 acrescente  $w$  a  $N'$

11 atualize  $D(v)$  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 ou o custo anterior para  $v$  ou o menor

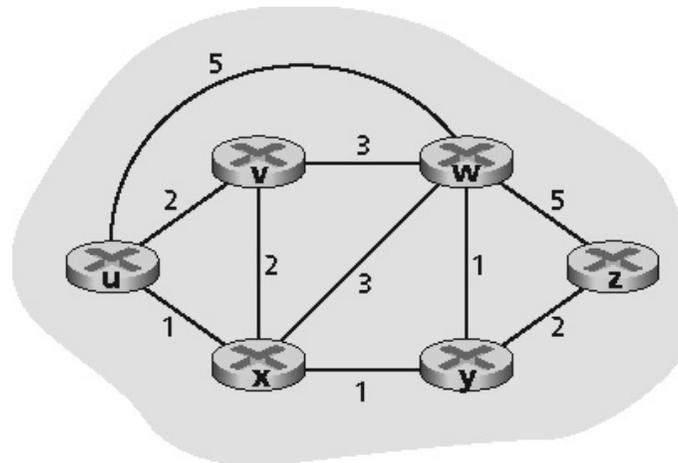
14 custo de caminho conhecido para  $w$  mais o custo de  $w$  a  $v$  \*/

15 **até que todos os nós estejam em  $N'$**



# 4 Exemplo: Algoritmo de Dijkstra

Passo	Início 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	$\infty$	$\infty$
→ 1	ux	2,u	4,x		2,x	$\infty$
→ 2	uxy	2,u	3,y			4,y
→ 3	uxyv		3,y			4,y
→ 4	uxyvw					4,y
5	uxyvwz					



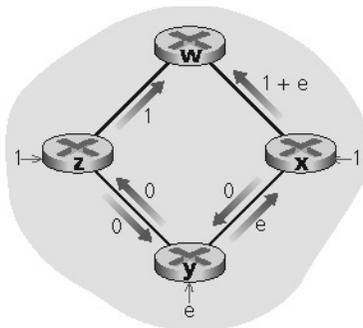
# 4 Discussão do algoritmo de Dijkstra

**Complexidade do algoritmo:**  $n$  nós

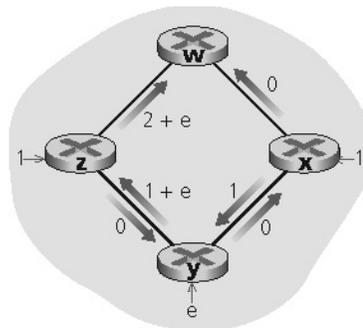
- Cada iteração: precisa verificar todos os nós  $w$ , que não estão em  $N$
- $n(n+1)/2$  comparações:  $O(n^2)$
- Implementações mais eficientes:  $O(n \log n)$

**Oscilações possíveis:**

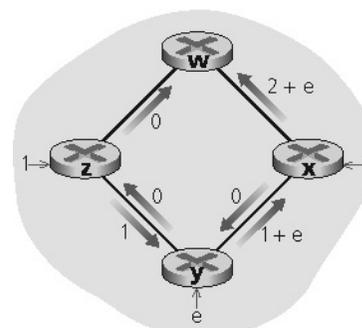
- Ex.: custo do link = quantidade de tráfego transportado



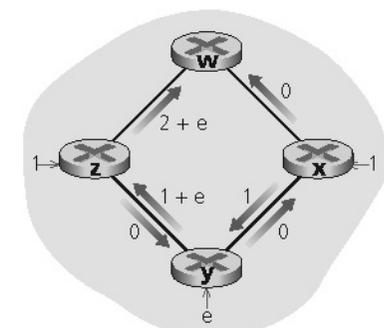
a. Roteamento inicial



b.  $x, y$  detectam melhor caminho até  $w$  em sentido horário



c.  $x, y, z$  detectam melhor caminho até  $w$  em sentido anti-horário



d.  $x, y, z$  detectam melhor caminho até  $w$  em sentido horário

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - **Distance vector**
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Algoritmo vetor de distância (1)

Equação de Bellman-Ford (programação dinâmica)

Define

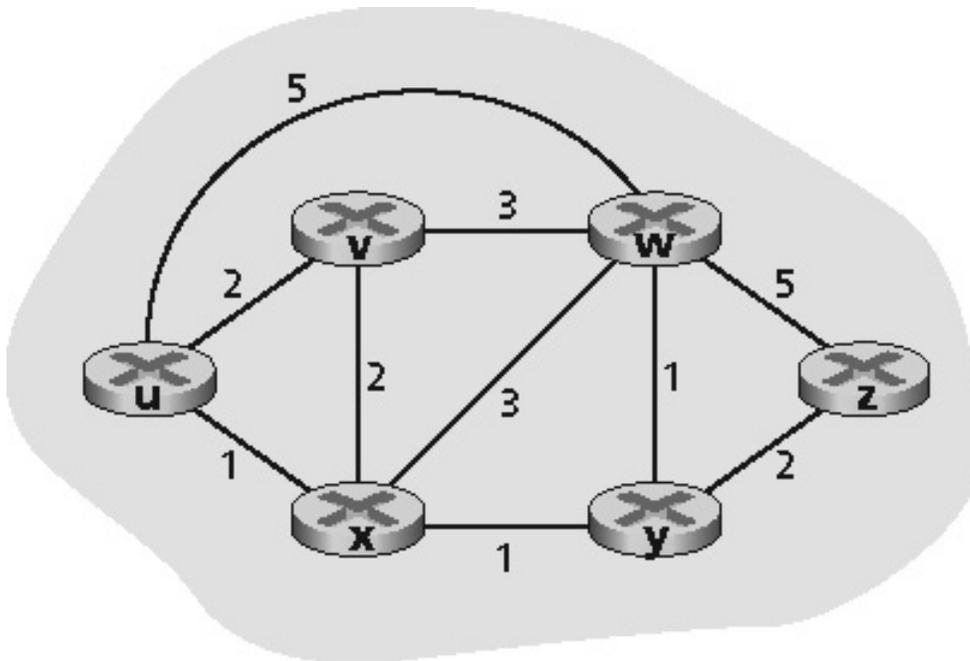
$d_x(y)$  = custo do caminho de menor custo de  $x$  para  $y$

Então

$$d_x(y) = \min \{c(x,v) + d_v(y)\}$$

Em que min é calculado sobre todos os vizinhos de  $x$

# 4 Exemplo: Bellman-Ford (2)



Claramente,  $d_v(z) = 5$ ,  $d_x(z) = 3$ ,  
 $d_w(z) = 3$

A equação B-F diz que:

$$\begin{aligned} d_u(z) &= \min \{ c(u,v) + d_v(z), \\ &\quad c(u,x) + d_x(z), \\ &\quad c(u,w) + d_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

O nó que atinge o mínimo é o próximo salto no caminho mais curto □  
tabela de roteamento

# 4 Algoritmo vetor de distância (3)

- $D_x(y)$  = estima o menor custo de x para y
- Vetor de distância:  $D_x = [D_x(y): y \in N]$
- O nó x conhece o custo para cada vizinho v:  $c(x,v)$
- O nó x mantém  $D_x = [D_x(y): y \in N]$
- O nó x também mantém os vetores de distância de seus vizinhos
  - Para cada vizinho v, x mantém  $D_v = [D_v(y): y \in N]$

# 4 Algoritmo vetor de distância (4)

## Idéia básica:

- Cada nó envia periodicamente sua própria estimativa de vetor de distância aos vizinhos
- Quando o nó  $x$  recebe nova estimativa de DV do vizinho, ele atualiza seu próprio DV usando a equação B-F:

$$D_x(y) = \min_v \{c(x,v) + D_v(y)\} \quad \text{para cada nó } y \in N$$

- Ao menos em condições naturais, a estimativa  $D_x(y)$  converge para o menor custo atual  $d_x(y)$

# 4 Algoritmo vetor de distância (5)

**Iterativo, assíncrono:** cada

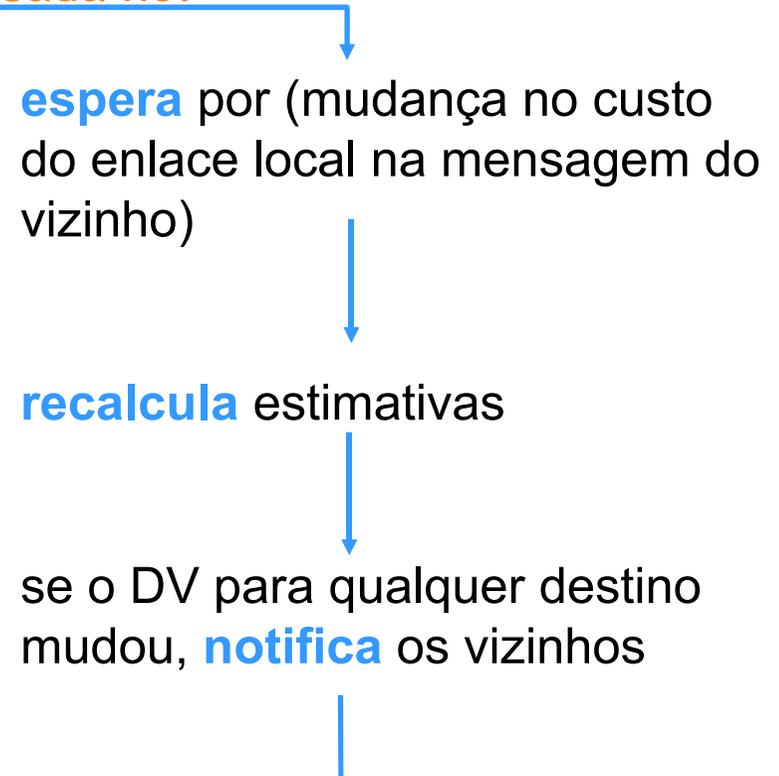
iteração local é causada por:

- Mudança no custo do enlace local
- Mensagem de atualização DV do vizinho

**Distribuído:**

- Cada nó notifica os vizinhos **apenas** quando seu DV mudar
  - Os vizinhos então notificam seus vizinhos, se necessário

**Cada nó:**



# 4 A camada de rede

Tabela do nó x

		Custo até		
		x	y	z
De	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

		Custo até		
		x	y	z
De	x	0	2	3
	y	2	0	1
	z	7	1	0

		Custo até		
		x	y	z
De	x	0	2	3
	y	2	0	1
	z	3	1	0

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} = \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} = \min\{2+1, 7+0\} = 3$$

Tabela do nó y

		Custo até		
		x	y	z
De	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

		Custo até		
		x	y	z
De	x	0	2	7
	y	2	0	1
	z	7	1	0

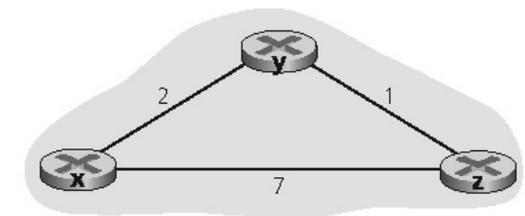
		Custo até		
		x	y	z
De	x	0	2	3
	y	2	0	1
	z	3	1	0

Tabela do nó z

		Custo até		
		x	y	z
De	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

		Custo até		
		x	y	z
De	x	0	2	7
	y	2	0	1
	z	3	1	0

		Custo até		
		x	y	z
De	x	0	2	3
	y	2	0	1
	z	3	1	0

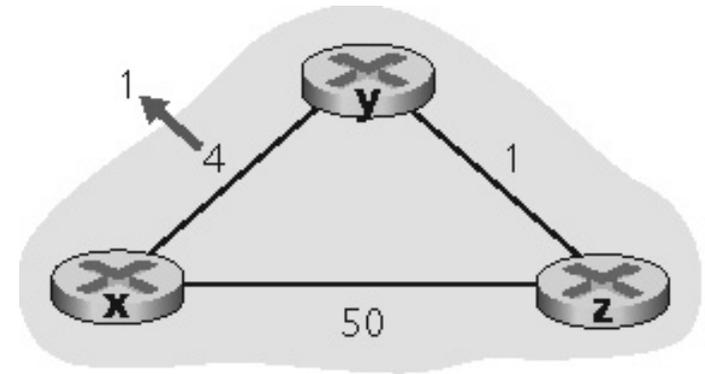


.....> Tempo

# 4 Vetor de distância: mudanças no custo do enlace

## Mudanças no custo do enlace:

- Nó detecta mudança no custo do enlace local
- Atualiza informações de roteamento, recalcula o vetor de distância
- Se o DV muda, notifica vizinhos



a.

No tempo  $t_0$ , y detecta a mudança no custo do enlace, atualiza seu DV e informa seus vizinhos.

No tempo  $t_1$ , z recebe a atualização de y e atualiza sua tabela.

Ele calcula o menor custo novo para x e envia seu DV para os vizinhos.

No tempo  $t_2$ , y recebe a atualização de z a atualiza sua tabela de distância. O menor custo de y's não muda e então y *não* envia nenhuma mensagem para z.

“boas notícias viajam depressa”

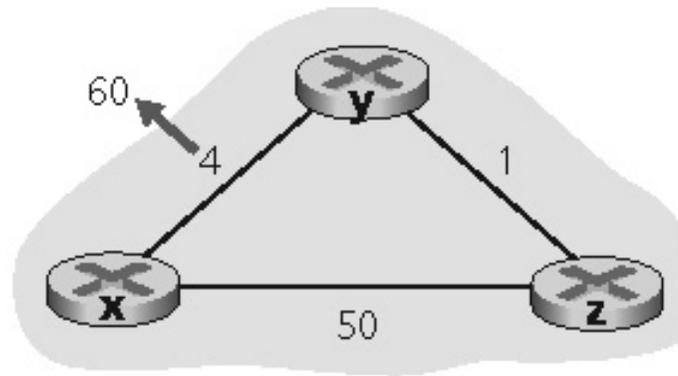
# 4 Vetor de distância: mudanças no custo do enlace

## Mudanças no custo do enlace:

- Boas notícias viajam rápido
- Más notícias viajam devagar – problema da “contagem ao infinito”!
- 44 iterações antes de o algoritmo estabilizar: veja o texto

## Reversão envenenada:

- Se Z roteia por Y para alcançar X :
  - Z diz a Y que sua distância (de Z) para X é infinita (então Y não roteará até X via Z)
- Isso resolverá completamente o problema da contagem ao infinito?



b.

# 4 Comparação dos algoritmos LS e VD

## Complexidade

- **LS:** com  $n$  nós,  $E$  links,  $O(NE)$  mensagens enviadas
- **DV:** trocas somente entre vizinhos
  - Tempo de convergência varia

## Tempo de convergência

- **LS:** algoritmo  $O(N^2)$  exige mensagens  $O(NE)$ 
  - Pode ter oscilações
- **DV:** tempo de convergência varia
  - Pode haver loops de roteamento
  - Problema da contagem ao infinito

**Robustez:** o que acontece se um roteador funciona mal?

**Ls:**

- Nós podem informar custos de **link** incorretos
- Cada nó calcula sua própria tabela de roteamento

**Dv:**

- Nó DV pode informar custo de **caminho** incorreto
- Tabela de cada nó é usada por outros
  - Propagação de erros pela rede

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Roteamento hierárquico

Nosso estudo é uma idealização

- Roteadores são todos idênticos
- Redes “flat”
- ... na prática, isso não é verdade

**Escala:** com 200 milhões de destinos:

- Não é possível armazenar todos os destinos numa única tabela de rotas!
- As mudanças na tabela de rotas irão congestionar os enlaces!

**Autonomia administrativa**

- Internet = rede de redes
- Cada administração de rede pode querer controlar o roteamento na sua própria rede

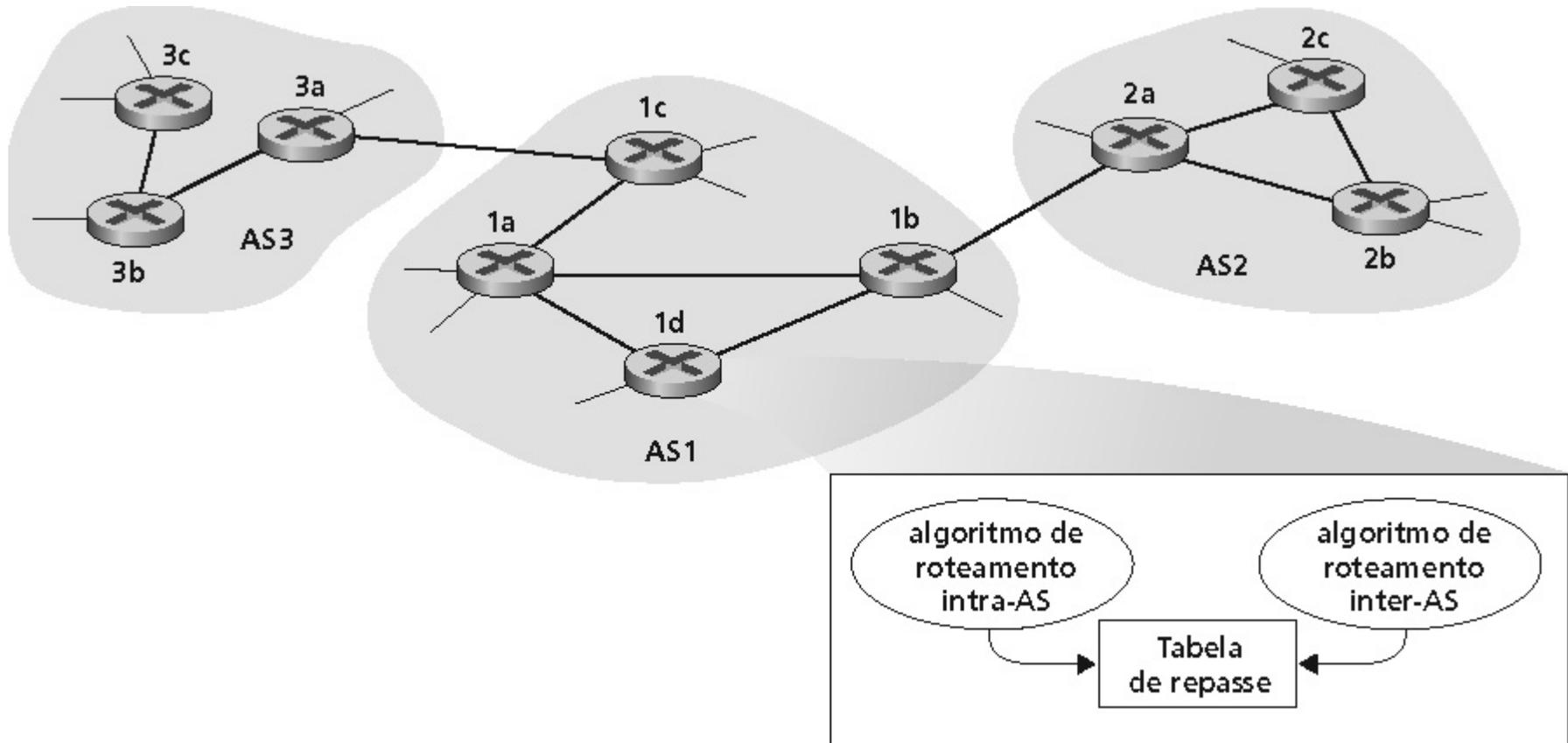
# 4 Roteamento hierárquico

- Agrega roteadores em regiões, “**sistemas autônomos**” (AS)
- Roteadores no mesmo AS rodam o mesmo protocolo de roteamento
  - Protocolo de roteamento “**intra-AS**”
  - Roteadores em diferentes AS podem rodar diferentes protocolos de roteamento

## **Roteador Gateway**

- Link direto para um roteador em outro AS

# 4 AS s interconectadas



- Tabela de roteamento é configurada por ambos os algoritmos, intra e inter-AS
  - Intra-AS estabelece entradas para destinos internos
  - Inter-AS e intra-As estabelecem entradas para destinos externos

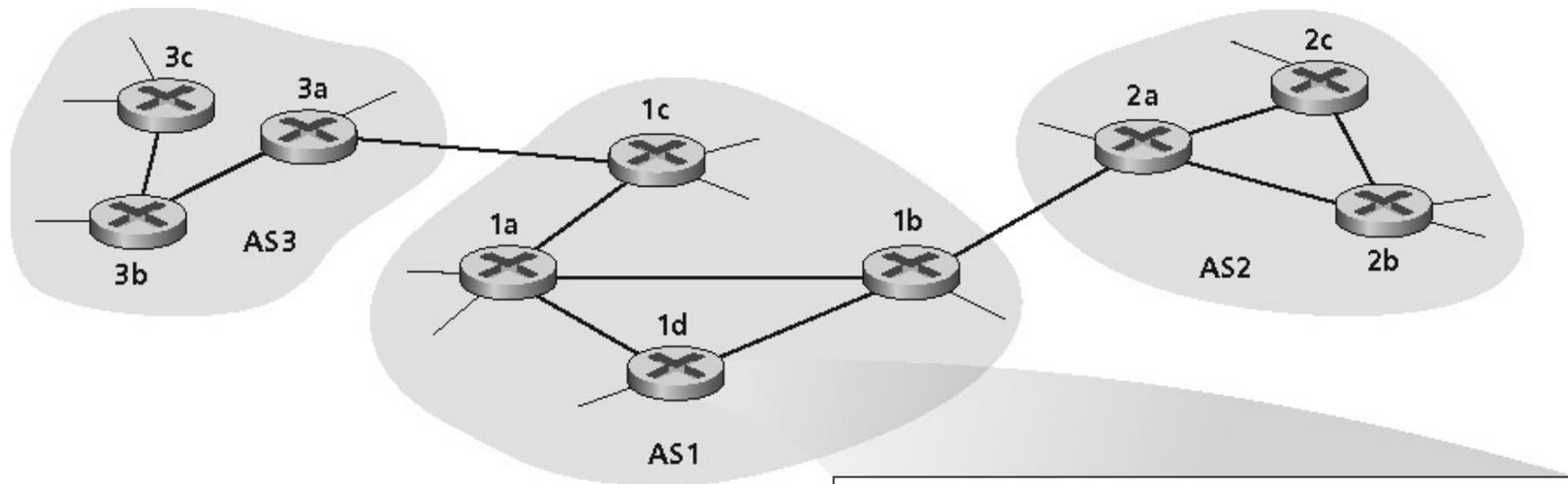
# 4 Tarefas Inter-AS

- Suponha que um roteador no AS1 receba um datagrama cujo destino seja fora do AS1
  - O roteador deveria encaminhar o pacote para os roteadores gateway, mas qual deles?

## AS1 precisa:

1. Aprender quais destinos são alcançáveis através de AS2 e através de AS3.
2. Propagar suas informações de alcance para todos os roteadores em AS1.

Tarefa para o roteamento inter-AS routing!

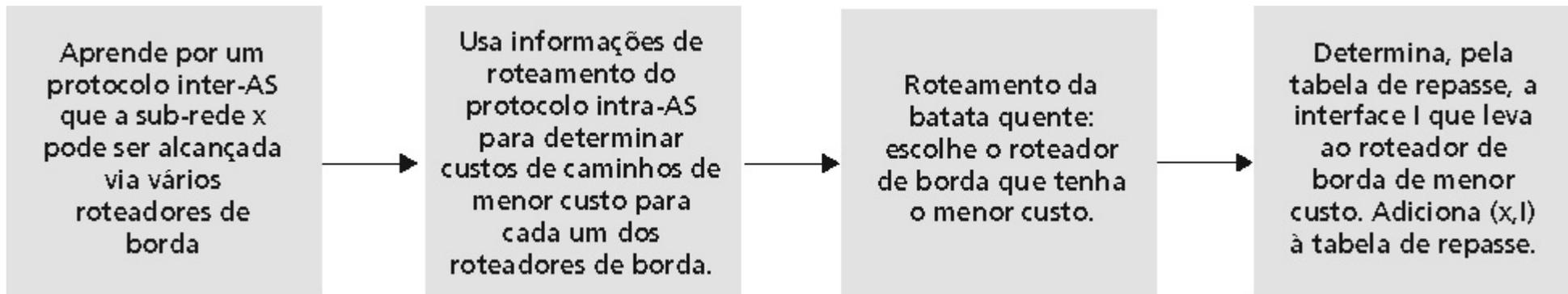


# 4 Exemplo: Ajustando a tabela de roteamento no roteador 1d

- Suponha que AS1 aprende pelo protocolo inter-AS protocol que a sub-rede **x** é alcançável através de AS3 (gateway 1c) mas não através de AS2
- O protocolo inter-AS propaga informações de alcance para todos os roteadores internos
- Baseado nas informações de roteamento intra-AS, o roteador 1d determina que sua interface **I** está no caminho de menor custo para 1c
- Coloca na tabela de roteamento a entrada **(x,I)**

# 4 Exemplo: Escolhendo entre múltiplas AS s

- Agora suponha que AS1 aprende pelo protocolo inter-AS que a sub-rede **x** é alcançavel através de AS3 e através de AS2.
- Para configurar a tabela de roteamento, o roteador 1d deve determinar por qual gateway ele deve encaminhar os pacotes para o destino **x**.
- Isso também é tarefa para o protocolo de roteamento inter-AS.
- **Roteamento de “batata quente”**: envia o pacote para o mais próximo de dois roteadores.



# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Roteamento intra-AS

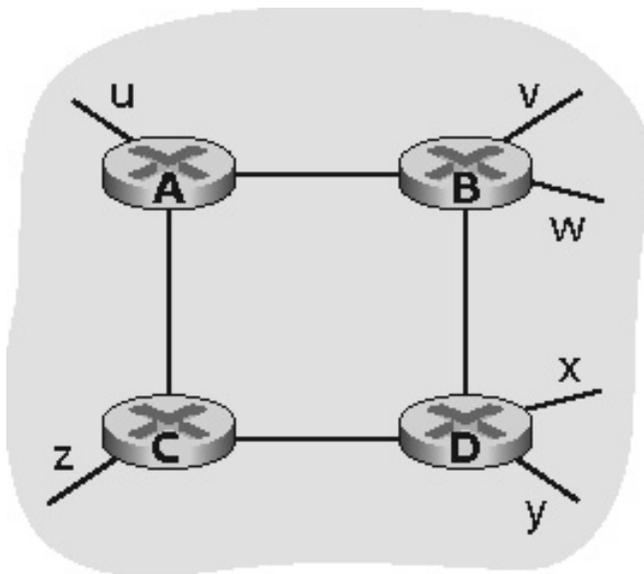
- Também conhecido como **Interior Gateway Protocols (IGP)**
- Protocolos de roteamento intra-AS mais comuns:
  - RIP: Routing Information Protocol
  - OSPF: Open Shortest Path First
  - IGRP: Interior Gateway Routing Protocol (proprietário da Cisco)

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 RIP (Routing Information Protocol)

- Algoritmo do tipo vetor distância
- Incluso na distribuição do BSD-UNIX em 1982
- Métrica de distância: # de saltos (máx. = 15 saltos)

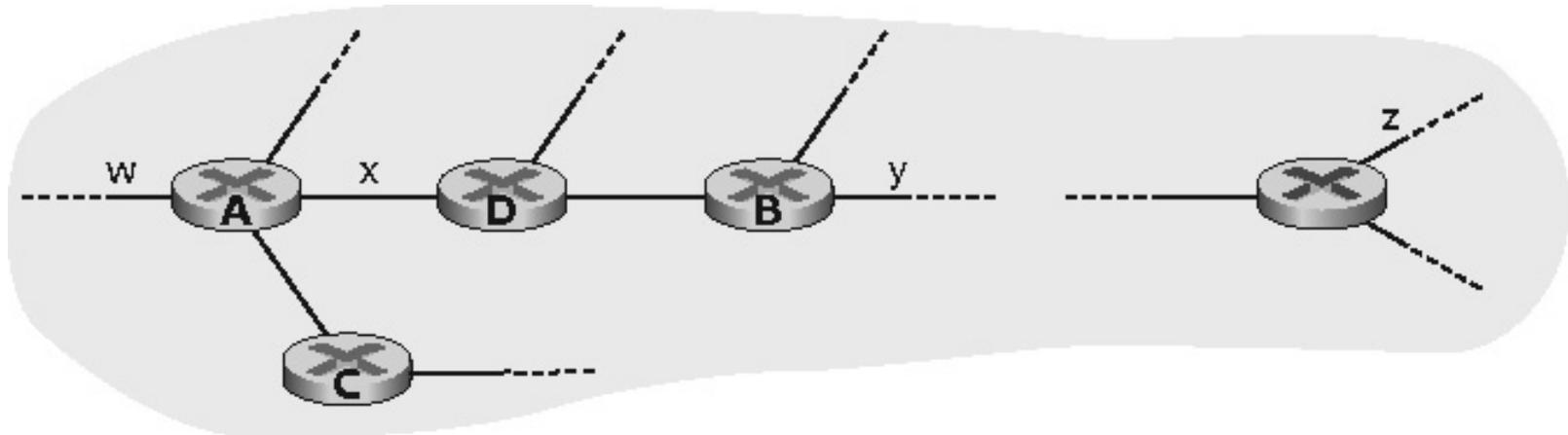


Destino	Saltos
u	1
v	2
w	2
x	3
y	3
z	2

# 4 Anúncio RIP

- Vetores de distância: trocados a cada 30 s via Response Message (também chamado **advertisement**, ou anúncio)
- Cada anúncio indica rotas para até 25 redes de destino

# 4 RIP : Exemplo



rede de destino	roteador seguinte	núm. de saltos para dest.
w	A	2
y	B	2
z	B	7
x	--	1
....	....	....

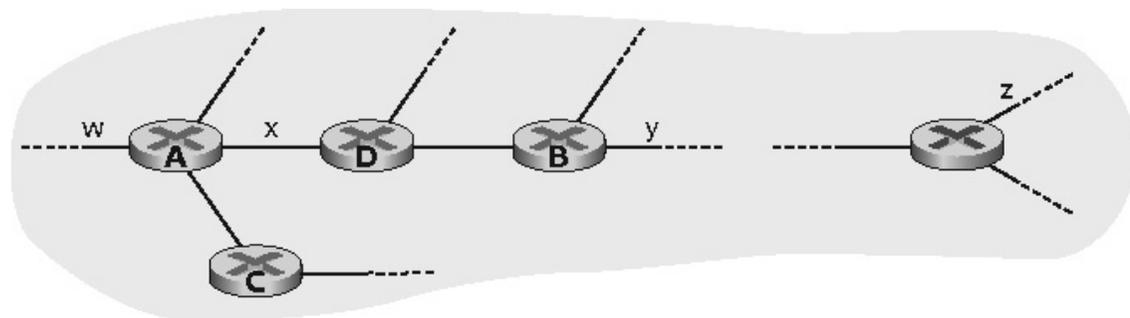
Tabela de roteamento em D

# 4 RIP : Exemplo

dest. próximos saltos

w	-	-
x	-	-
z	C	4
....	...	...

Anúncio de A para D



rede de destino	roteador seguinte	núm. de saltos até dest.
w	A	2
y	B	2
z	<del>B A</del>	<del>7 5</del>
x	--	1
....	....	....

Routing table in D

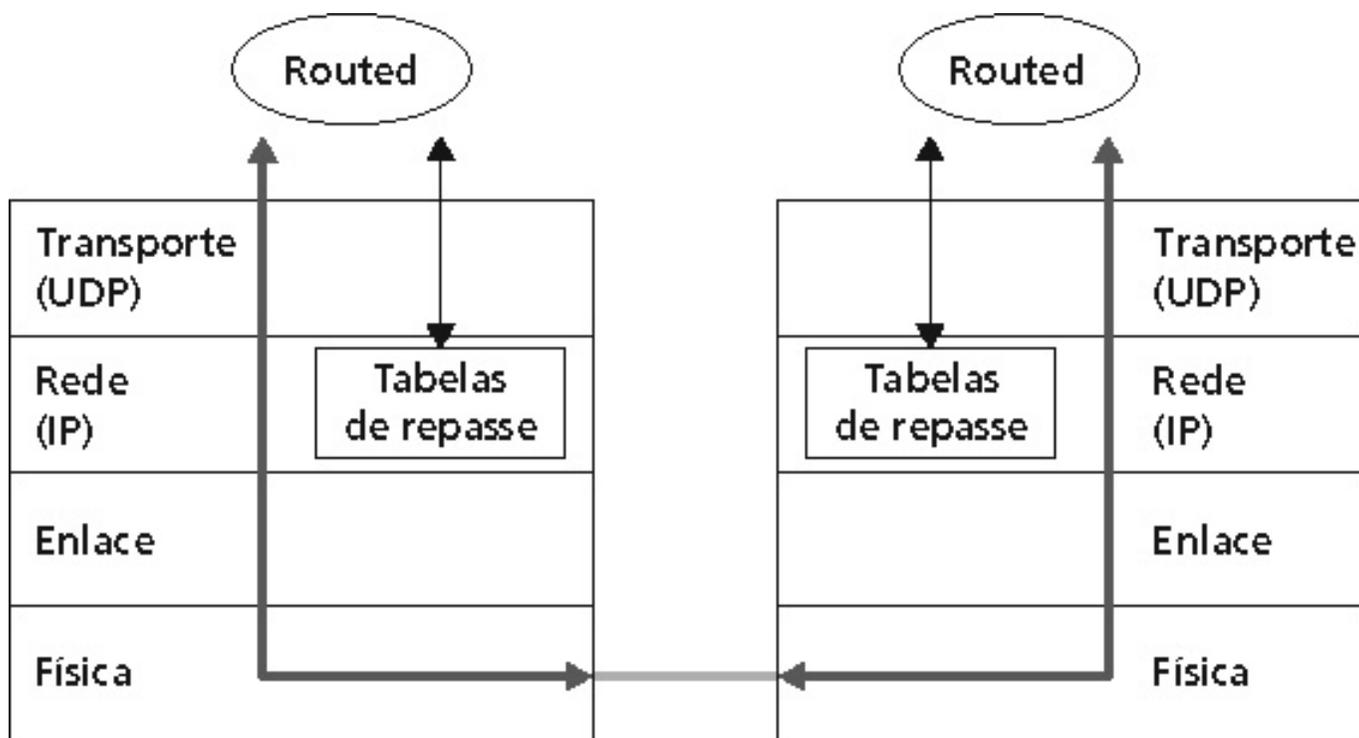
# 4 RIP : falha de enlaces e recuperação

Se não há um aviso depois de 180 s, --> 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
- Reversão envenenada é usada para prevenir loops, (distância infinita = 16 saltos)

# 4 RIP Processamento da tabela de rotas

- As tabelas de roteamento do RIP são manipuladas por um processo de aplicação chamado route-d (daemon)
- Anúncios são enviados em pacotes UDP com repetição periódica



# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

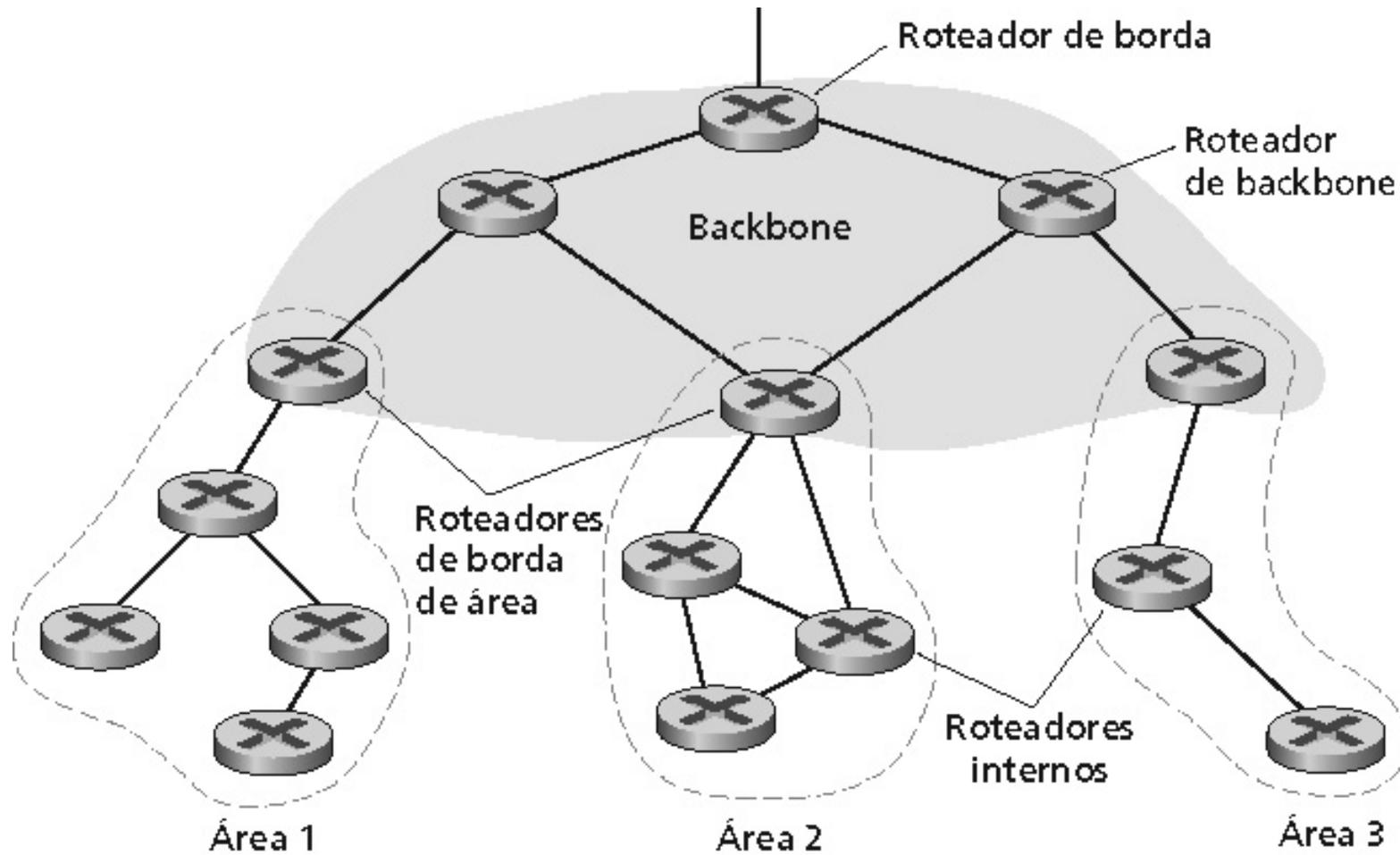
# 4 OSPF (Open Shortest Path First)

- “Open”: publicamente disponível
- Usa algoritmo do tipo link state
  - Disseminação de pacotes LS
  - Mapa topológico em cada nó
  - Usa algoritmo de Dijkstra para cálculo de rotas
- Anúncios do OSPF transportam um registro para cada roteador vizinho
- Anúncios são distribuídos para **todo** o AS (via flooding)
  - Transportado por mensagens OSPF diretamente sobre IP (melhor do que TCP ou UDP)

# 4 OSPF características “avançadas”

- **Segurança:** todas as mensagens do OSPF são autenticadas (para prevenir intrusões maliciosas)
- Múltiplos caminhos de mesmo custo são permitidos (o RIP só permite um caminho)
- Para cada link, múltiplas métricas de custo para **TOS** diferentes (ex.: custo de enlace por satélite definido baixo para tráfego de “melhor esforço” e alto para serviços de tempo real)
- Integra tráfego uni- e **multicast:**
  - Multicast OSPF (MOSPF) usa a mesma base de dados de topologia do OSPF
- **OSPF hierárquico:** OSPF para grandes domínios

# 4 OSPF hierárquico



# 4 OSPF hierárquico

- **Hierarquia de dois níveis:** área local e backbone.
  - Anúncios de link state apenas na área
  - Cada nó tem a topologia detalhada da área, mas somente direções conhecidas (caminhos mais curtos) para redes em outras áreas
- **Roteadores de borda de área:** “resumem” distâncias para redes na própria área e enviam para outros roteadores de borda de área
- **Roteadores de backbone:** executam o roteamento OSPF de forma limitada ao backbone.
- **Roteadores de borda:** conectam-se a outras AS's.

# 4 A camada de rede

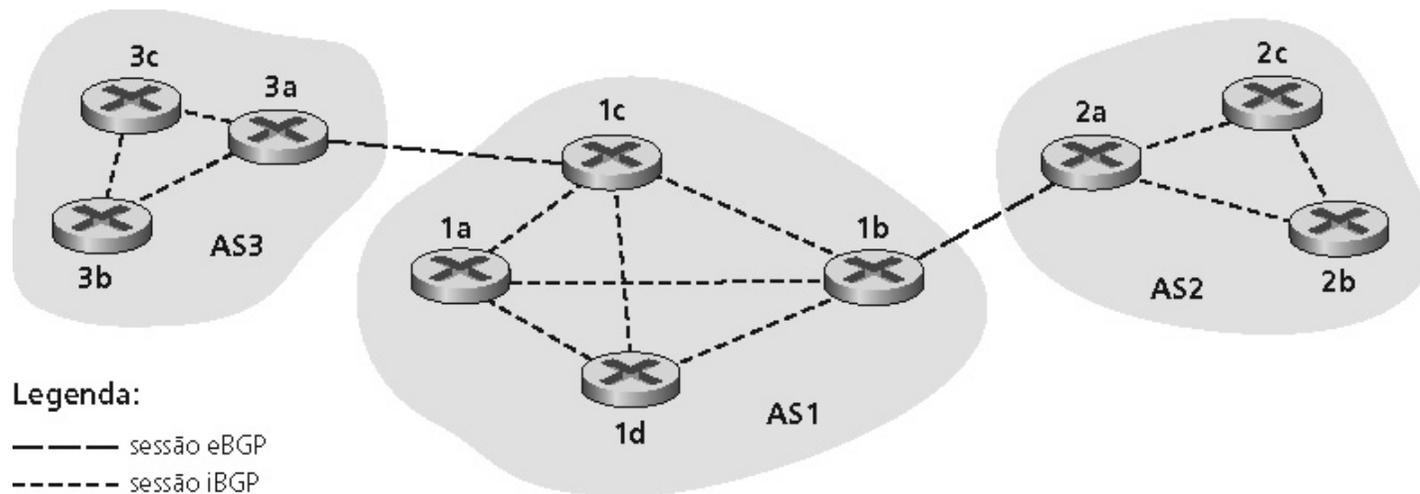
- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

# 4 Roteamento inter-AS da Internet: BGP

- **BGP (Border Gateway Protocol):** é o padrão de fato para uso na Internet
- BGP provê cada AS dos meios para:
  1. Obter informações de alcance de sub-rede dos ASs vizinhos
  2. Propagar informações de alcance para todos os roteadores internos ao AS
  3. Determinar “boas” rotas para as sub-redes baseado em informações de alcance e política
- Permite que uma subnet comunique sua existência para o resto da Internet:  
“Estou aqui”

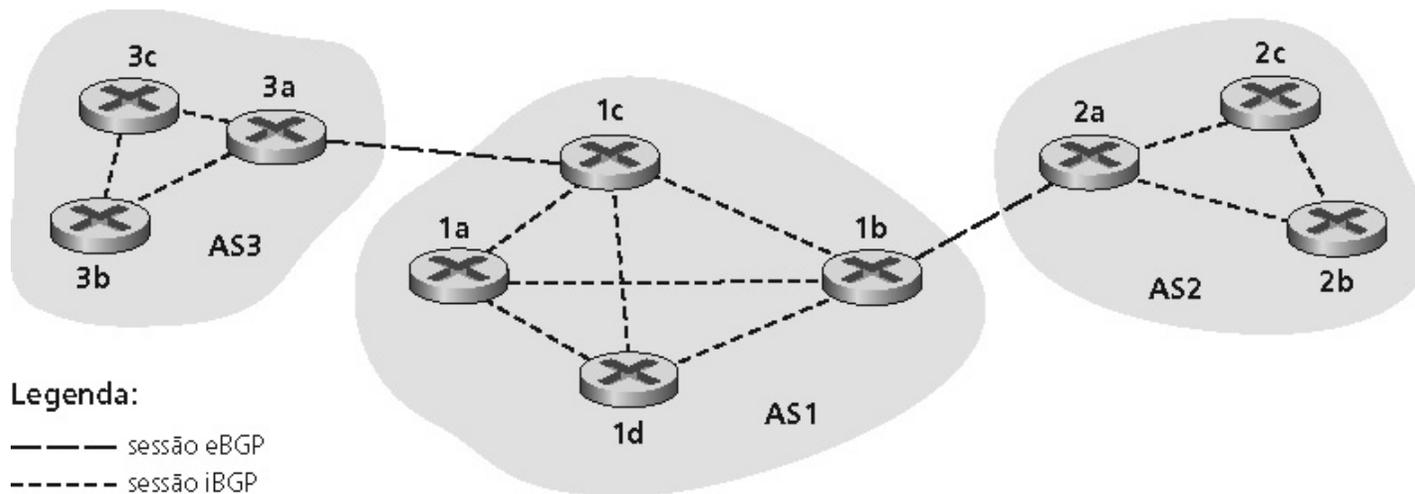
# 4 BGP: conceitos básicos

- Pares de roteadores (BGP peers) trocam informações de roteamento por conexões TCP semipermanentes: **sessões BGP**
- Note que as sessões BGP não correspondem aos links físicos
- Quando AS2 comunica um prefixo ao AS1, AS2 está **prometendo** que encaminhará todos os datagramas destinados a esse prefixo em direção ao prefixo
  - AS2 pode agregar prefixos em seu comunicado



# 4 Distribuindo informações de alcance

- Em cada sessão eBGP entre 3a e 1c, AS3 envia informações de alcance de prefixo para AS1.
- 1c pode então usar iBGP para distribuir essa nova informação de alcance de prefixo para todos os roteadores em AS1
- 1b pode recomunicar essa nova informação para AS2 por meio da sessão eBGP 1b-para-2a.
- Quando um roteador aprende um novo prefixo, ele cria uma entrada para o prefixo em sua tabela de roteamento.



# 4 Atributos de caminho e rotas BGP

- Quando se comunica um prefixo, o comunicado inclui os atributos do BGP.
  - Prefixo + atributos = “rota”
- Dois atributos importantes:
  - **AS-PATH**: contém os ASs pelos quais o comunicado para o prefixo passou: AS 67 AS 17
  - **NEXT-HOP**: Indica o roteador específico interno ao AS para o AS do próximo salto (next-hop). (Pode haver múltiplos links do AS atual para o AS do próximo salto.)
- Quando um roteador gateway recebe um comunicado de rota, ele usa **política de importação** para aceitar/rejeitar.

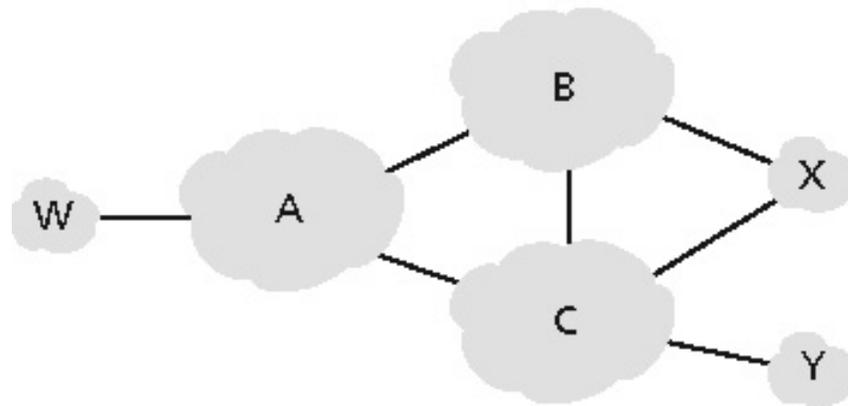
# 4 BGP: seleção de rota

- Um roteador pode aprender mais do que 1 rota para o mesmo prefixo. O roteador deve selecionar uma rota
- Regras de eliminação:
  - Atributo de valor de preferência local: decisão de política
  - AS-PATH (caminho) mais curto
  - Roteador do NEXT-HOP (próximo salto) mais próximo: roteamento da “batata quente”
  - Critérios adicionais

# 4 Mensagens BGP

- Mensagens BGP são trocadas usando o TCP
- Mensagens BGP:
  - **OPEN**: abre conexão TCP para o peer e autentica o transmissor
  - **UPDATE**: comunica novo caminho (ou retira um antigo)
  - **KEEPALIVE** mantém a conexão ativa na ausência de atualizações (updates); também ACKs OPEN request
  - **NOTIFICATION**: reporta erros em mensagens anteriores; também usado para fechar a conexão

# 4 BGP: política de rotear

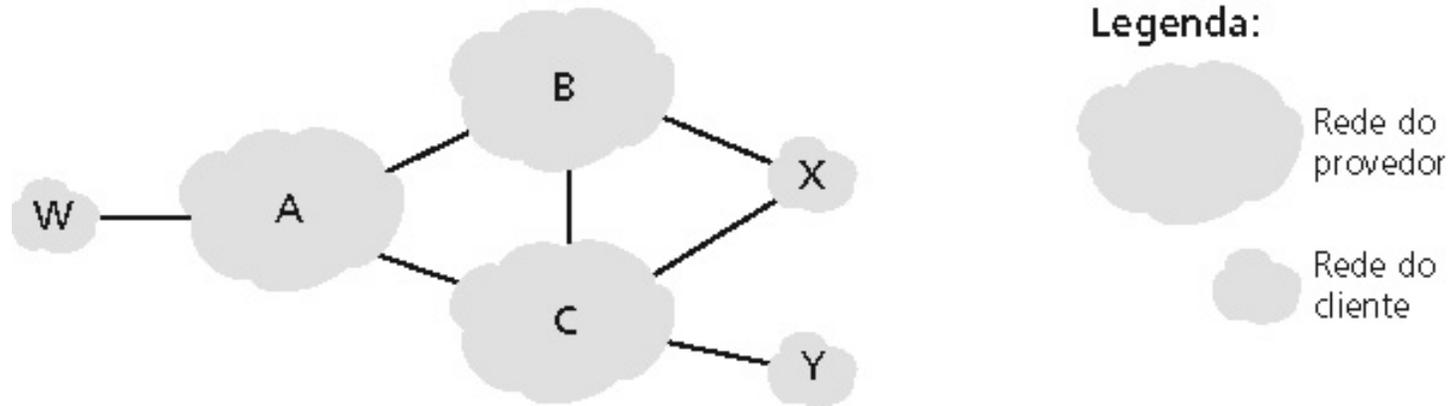


Legenda:



- A, B, C são **redes do provedor**
- X, W, Y são clientes (das redes do provedor)
- X é **dual-homed**: anexados a duas redes
  - X não quer rotear de B via X para C
  - ... então X não comunicará ao B uma rota para C

# 4 BGP: política de roteamento (2)



- A comunica ao B o caminho AW
- B comunica ao X o caminho BAW
- B deveria comunicar ao C o caminho BAW?
  - De jeito nenhum! B não obtém nenhum “rendimento” em rotear CBAW pois nem W nem C são seus clientes
  - B quer forçar C a rotear para W via A
  - B quer rotear **somente** de/para seus clientes!

# 4 Por que os protocolos intra e inter-AS são diferentes?

## Políticas:

- Inter-AS: a administração quer ter controle sobre como seu tráfego é roteado e sobre quem roteia através da sua rede.
- Intra-AS: administração única, então não são necessárias políticas de decisão

## Escalabilidade

- O roteamento hierárquico poupa espaço da tabela de rotas e reduz o tráfego de atualização

## Desempenho:

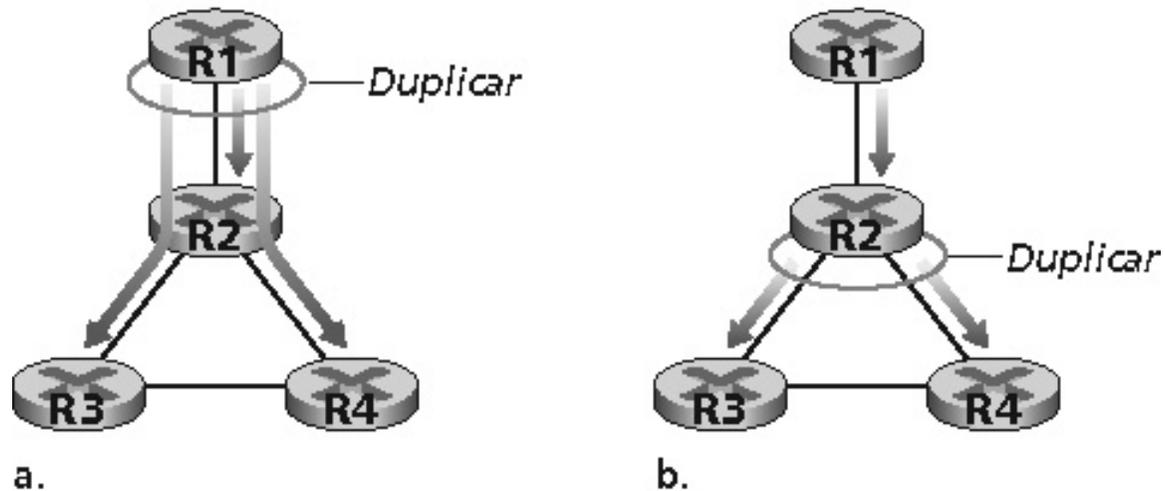
- Intra-AS: preocupação maior é desempenho
- Inter-AS: políticas podem ser dominantes em relação ao desempenho

# 4 A camada de rede

- 4.1 Introdução
- 4.2 Circuito virtual e redes de datagrama
- 4.3 O que há dentro de um roteador
- 4.4 IP: Protocolo da Internet
  - Formato do datagrama
  - Endereçamento IPv4
  - ICMP
  - IPv6
- 4.5 Algoritmos de roteamento
  - Link state
  - Distance vector
  - Roteamento hierárquico
- 4.6 Roteamento na Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Roteamento de broadcast e multicast

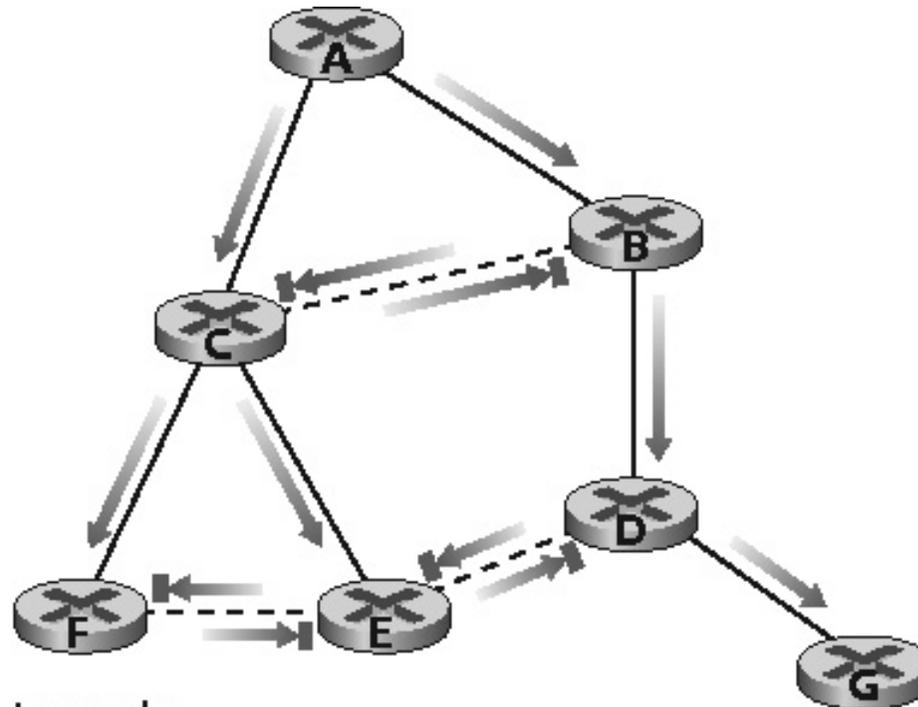
# 4 Duplicação na origem *versus* duplicação na rede

Criação/transmissão de duplicatas



(a) duplicação na origem, (b) duplicação na rede

# 4 Repasse pelo caminho reverso

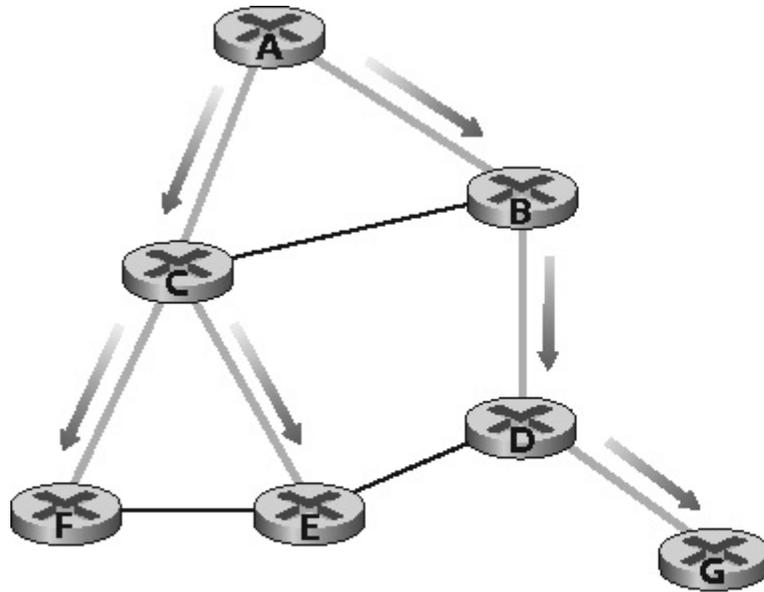


Legenda:

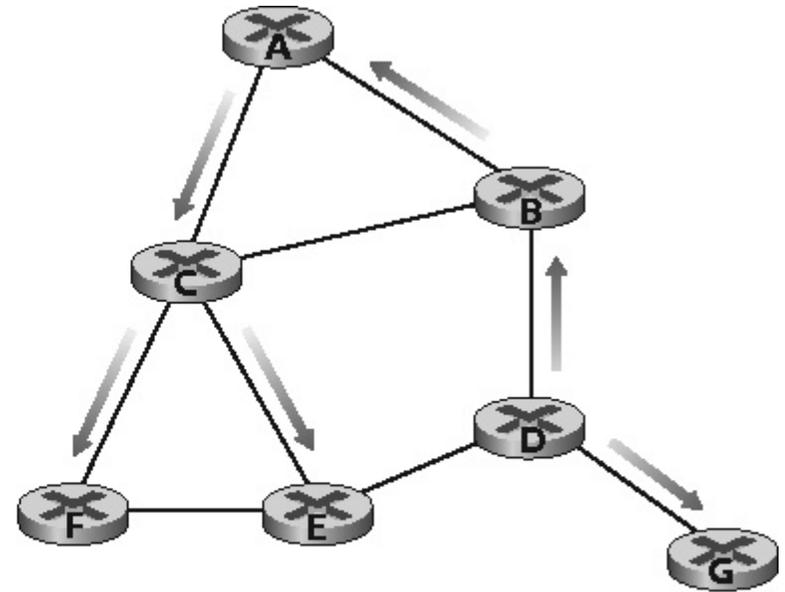
→ pacote (pkt) será repassado

→ pacote (pkt) não será repassado além do roteador receptor

# 4 Broadcast ao longo de uma spanning tree

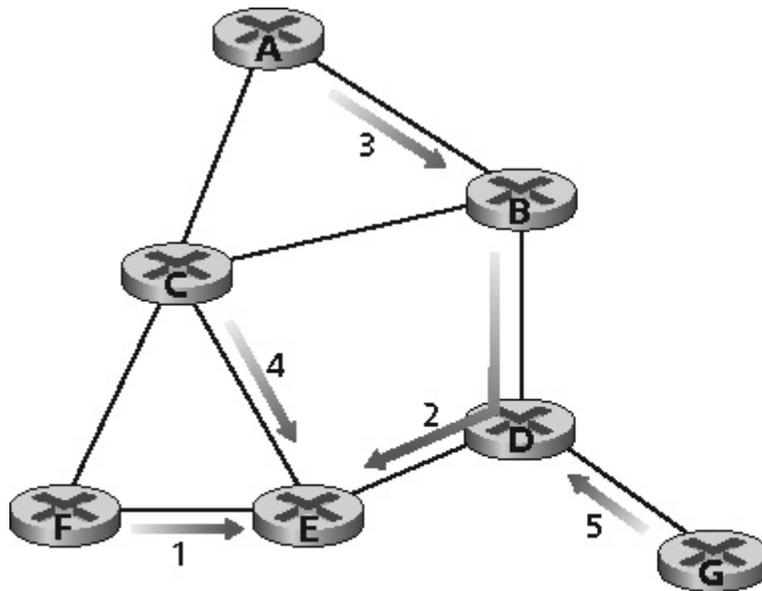


a. Broadcast iniciado em A

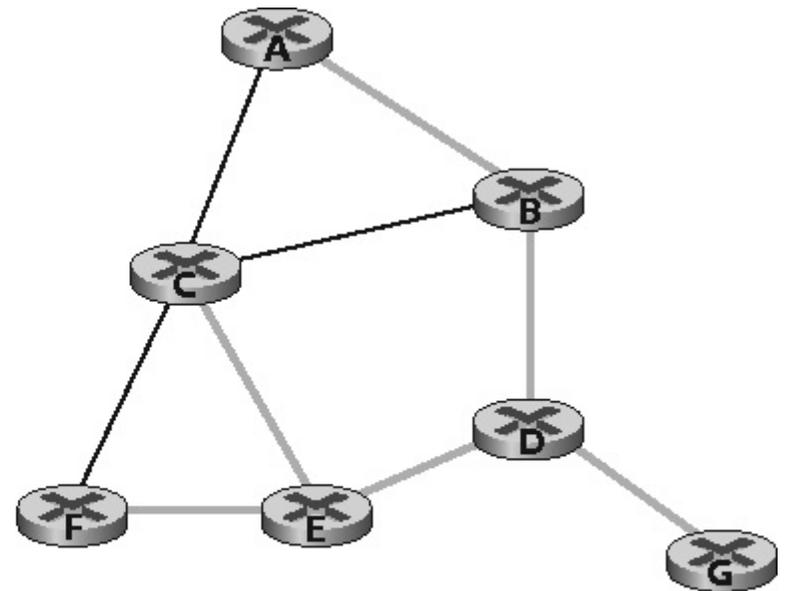


b. Broadcast iniciado em D

# 4 Construção centro-baseada de uma spanning tree



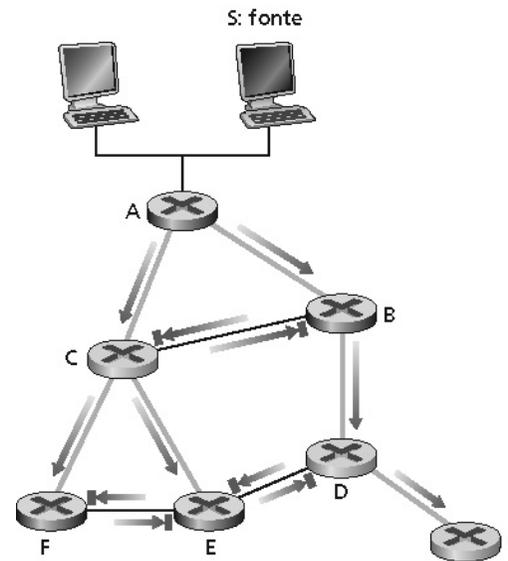
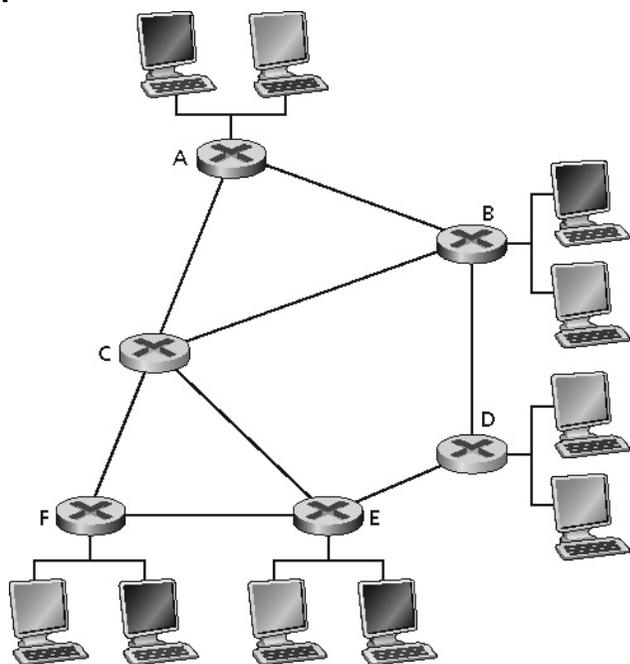
a. Construção da spanning tree passo a passo



b. Spanning tree construída

# 4 Roteamento multicast: indicação do problema

- **Objetivo:** encontrar uma árvore (ou árvores) conectando roteadores que possuem membros de grupo multicast local
  - **Árvore:** não são todos os caminhos entre os roteadores usados
  - **Baseada na fonte:** uma árvore diferente de cada transmissor para os receptores
  - **Árvore compartilhada:** a mesma árvore é usada por todos o membros do grupo



Legenda:

→ pacote (pkt) que será repassado

→ pacote (pkt) que não será repassado além do roteador destinatário

# 4 Métodos para construir multicast trees

Métodos:

- **Árvore baseada na fonte:** uma árvore por origem
  - Shortest path trees
  - Repasse pelo caminho reverso
- **Árvore compartilhada pelo grupo:** grupo usa uma árvore
  - Minimal spanning (Steiner)
  - Center-based trees

... primeiro veremos métodos básicos e, então, os protocolos específicos que adotam estes métodos

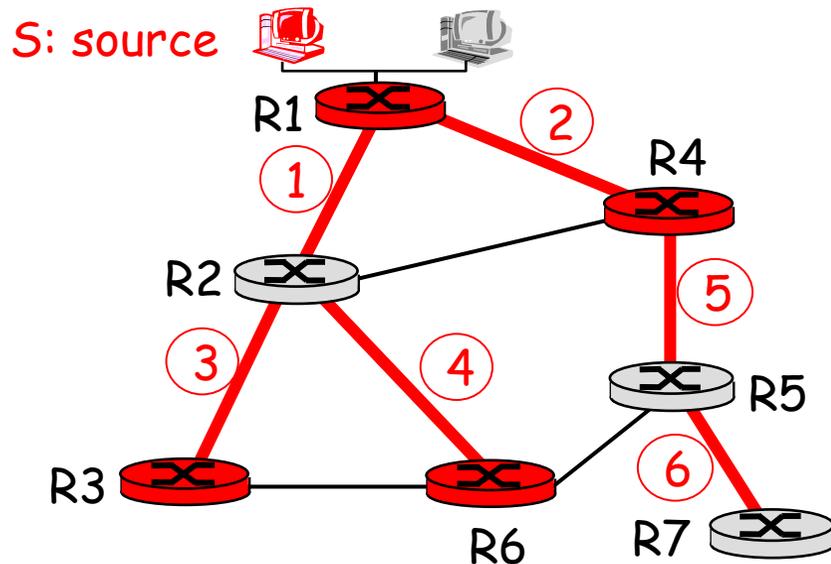


PEARSON

Addison  
Wesley

# 4 Shortest Path Tree

- mcast forwarding tree: árvore de rotas de caminho mais curto da origem para todos os receptores
- Algoritmo de Dijkstra



## LEGENDA

-  roteador com membro de grupo anexado
-  roteador sem nenhum membro de grupo anexado
-  link usado para encaminhamento, i indica link de ordem adicionado por algoritmo

# 4 Reverse Path Forwarding

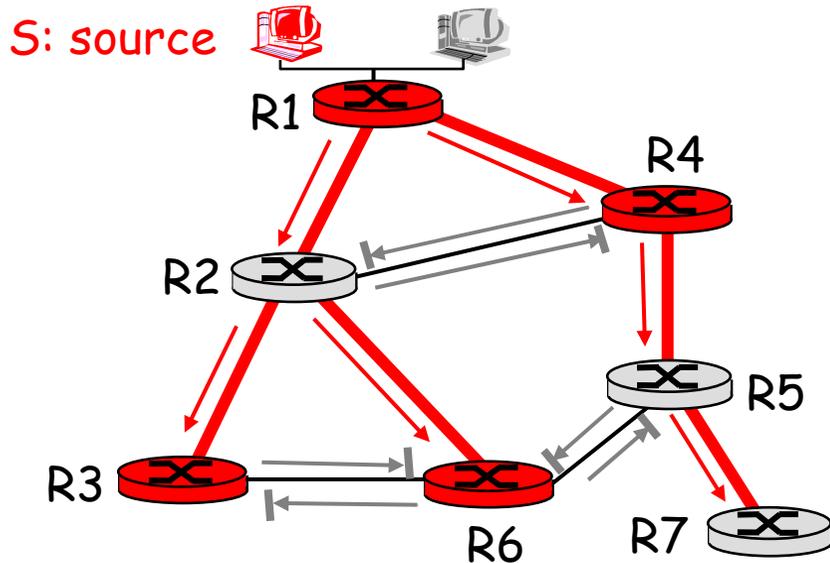
- Baseia-se no conhecimento dos roteadores sobre caminhos de unicast mais curtos dele até o transmissor
- Cada roteador possui comportamento de encaminhamento simples:

*if* (datagrama mcast recebido no link de entrada no menor caminho  
retorna ao centro)

*then* dispara datagramas para todos os links de saída

*else* ignora datagrama

# 4 Reverse Path Forwarding: exemplo



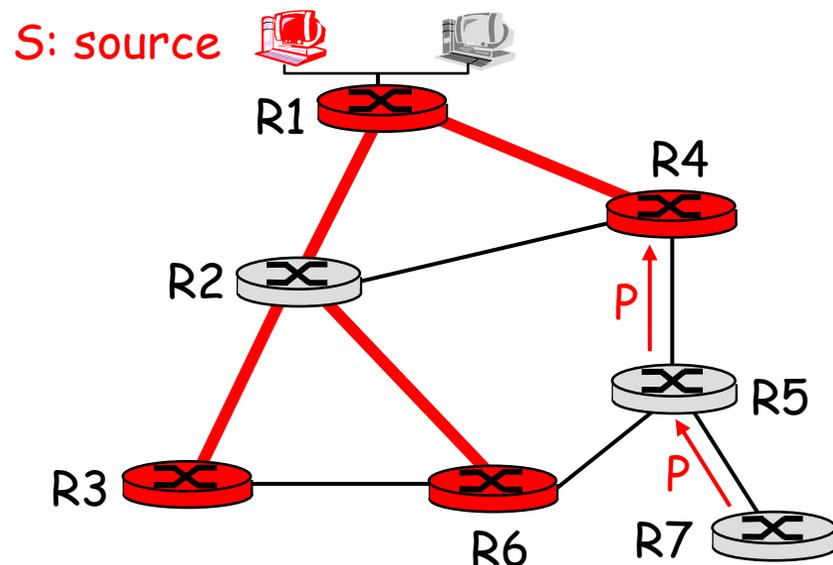
## LEGENDA

-  roteador com membro de grupo anexado
-  roteador sem nenhum membro de grupo anexado
-  datagrama será encaminhado
-  datagrama não será encaminhado

- Resultado é um reverse SPT de origem específica.
- Pode ser uma má escolha com links assimétricos

# 4 Reverse Path Forwarding: pruning

- Árvores de encaminhamento contêm subárvores com membros de grupo sem multicast
- Não necessita encaminhar datagramas por subárvores abaixo
- Mensagens “prune” são enviadas por upstream pelo roteador com membros de grupo sem nenhum downstream



## LEGENDA

-  roteador com membro de grupo anexado
-  roteador sem nenhum membro de grupo anexado
-  mensagem prune
-  links com encaminhamento multicast

# 4 Shared-Tree Steiner Tree

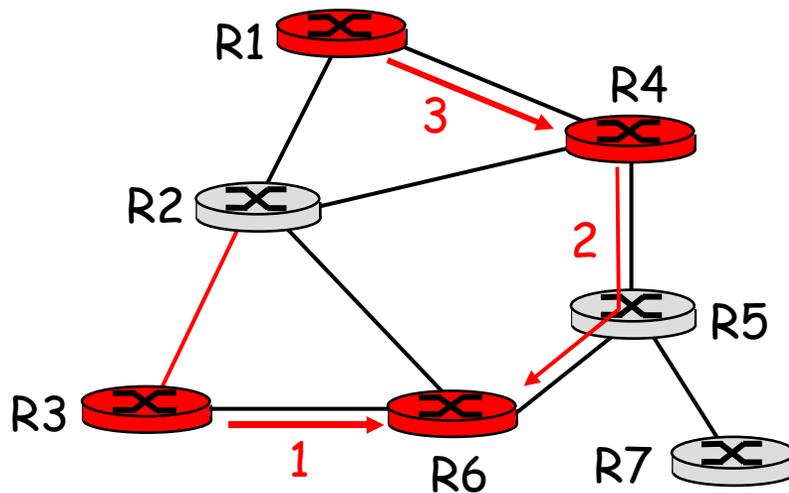
- **Steiner Tree:** árvore de custo mínimo conectando todos os roteadores com membros de grupo anexados
- Problema é NP-completo
- Existe uma heurística excelente
- Não é usado na prática:
  - Complexidade computacional
  - Informação sobre toda a rede é necessária
  - Monolítica: reexecuta sempre que um roteador precisa se juntar/deixar.

# 4 Center-based trees

- Única árvore de entrega compartilhada por todos
- Um roteador é identificado como “centro” da árvore para se juntar:
  - Roteador de borda envia uma **join-msg** unicast endereçada ao roteador de centro
  - **join-msg** “processada” pelos roteadores intermediários e encaminhada rumo ao centro
  - **join-msg** ou encontra um ramo da árvore para seu centro, ou chega até o centro
  - O caminho tomado pela **join-msg** torna-se um novo ramo da árvore para esse roteador

# 4 Center-based trees: um exemplo

Suponha que R6 escolheu como centro:



## LEGENDA

-  roteador com membro de grupo anexado
-  roteador sem nenhum membro de grupo anexado
-  ordem de caminho onde são geradas mensagens join

# 4 Roteamento de multicast da Internet: DVMRP

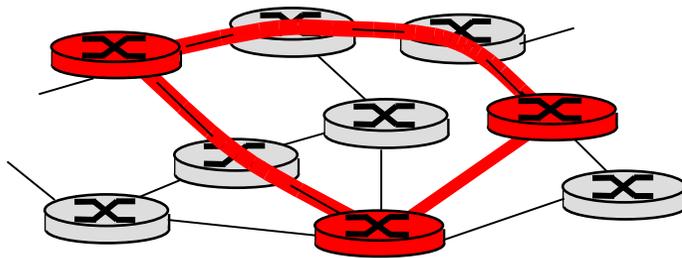
- **DVMRP**: distance vector multicast routing protocol, RFC1075
- **flood and prune**: reverse path forwarding, source-based tree
  - Árvore RPF baseada em tabelas de roteamento do DVMRP construídas pelos roteadores de comunicação do DVMRP
  - Sem suposições sobre unicast subjacente
  - Datagrama inicial para grupo de multicast disparado para todos os lados via RPF
  - Roteadores não querem grupo: enviam mensagens “prune” de upstream

# 4 DVMRP

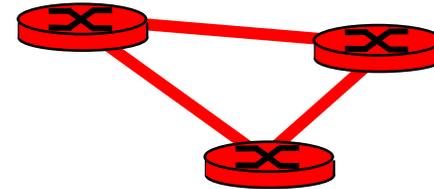
- **soft state:** o roteador DVMRP periodicamente (1 min) “esquece” os ramos que estão podados:
  - Dados mcast fluem novamente por ramos não podados
  - Roteador de downstream: poda novamente ou senão continua a receber dados
- Roteadores podem rapidamente se reintegrar à árvore
  - IGMP seguinte se junta na folha
- Probabilidades e extremidades
  - Comumente implementado em roteadores comerciais
  - Roteamento Mbone é feito usando DVMRP

# 4 Tunelamento

**P.:** Como conectar “ilhas” de roteadores multicast num “mar” de roteadores unicast?



topologia física



topologia lógica

- Datagrama mcast encapsulado dentro de um datagrama “normal” (sem endereço mcast)
- O datagrama IP normal é enviado pelo “túnel” via unicast IP regular para o roteador mcast receptor
- O roteador mcast receptor desencapsula para obter o datagrama mcast

# 4 PIM: Protocol Independent Multicast

- Não depende de nenhum algoritmo de roteamento unicast específico (trabalha com todos)
- Dois cenários diferentes de distribuição de multicast:

## Denso

- Membros de grupo densamente empacotados, em proximidade.
- Largura de banda mais abundante

## Esparso

- # redes com membros de grupo pequeno wrt # redes interconectadas
- Membros “extensamente dispersados”
- Largura de banda não abundante

# 4 Consequência de dicotomia esparsa-densa

## Densa

- Sociedade de grupo de roteadores **assumida** até que os roteadores sejam explicitamente podados
- Construção dirigida ao dado (**data-driven**) na árvore multicast (ex., RPF)
- Largura de banda e processamento de roteador sem grupo **deliberados**

## Esparsa

- Não há sociedade até que os roteadores se juntem explicitamente
- Construção dirigida ao receptor (**receiver-driven**) da árvore multicast (ex., center-based)
- Largura de banda e processamento de roteador sem grupo **conservativos**

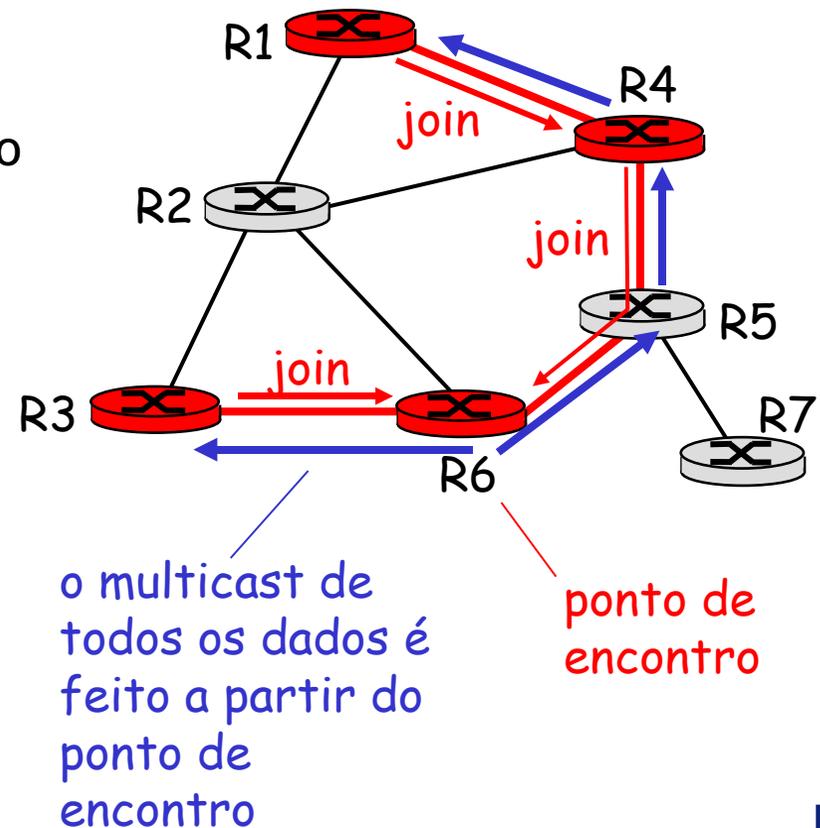
# 4 PIM-Modo denso

**Flood and prune RPF**, similar ao DVMRP mas

- Protocolo unicast em questão provê informações RPF para os datagramas que chegam
- Flood de downstream menos complicado (menos eficiente) do que o DVMRP reduz a confiança no algoritmo de roteamento em questão
- Possui mecanismo de protocolo para o roteador detectar que ele é um roteador **leaf-node**

# 4 PIM-Modo esparso

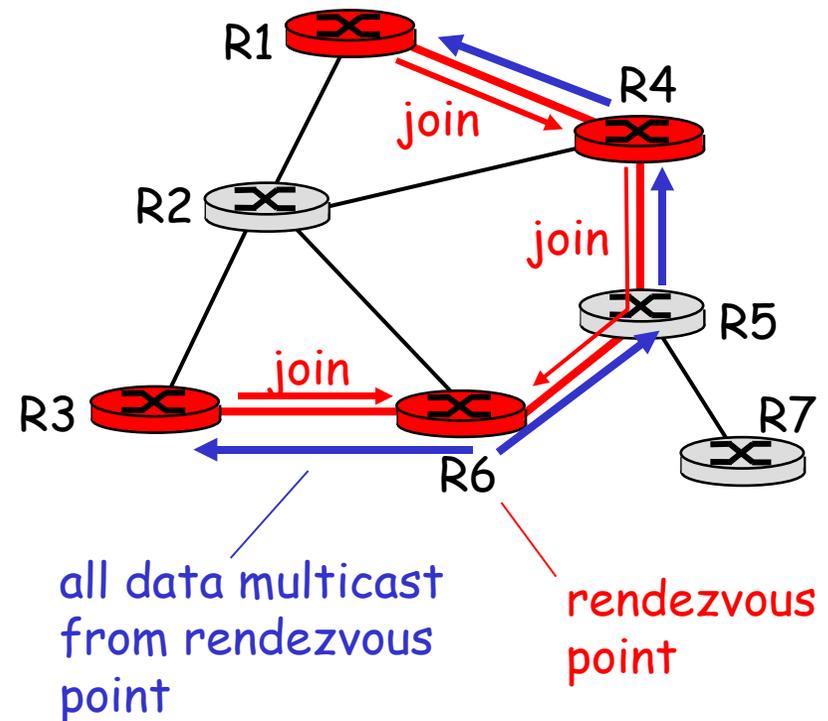
- Método **center-based**
- Roteador envia mensagens *join* para o ponto de encontro RP (**rendezvous point**)
  - Roteadores intermediários atualizam o estado e encaminham **join**
- Após juntar via RP, o roteador pode mudar para uma árvore de origem específica
  - Aumento de desempenho: menos concentração, caminhos menores



# 4 PIM-Modo esparso

## Transmissor(es):

- Unicast de dados para o RP, que distribui pela árvore **RP-roteada**
- RP pode estender o upstream da árvore multicast para a origem
- RP pode enviar a mensagem **stop** se não houver receptores anexados
  - “Ninguém está escutando!”



# 4 Camada de rede: resumo

## O que foi coberto:

- Serviços da camada de rede
- Princípios de roteamento: link state e distance vector
- Roteamento hierárquico
- IP
- Protocolos de roteamento da Internet: RIP, OSPF, BGP
- O que há dentro de um roteador?
- IPv6

**Próxima etapa: a camada de enlace de dados!**