

Capítulo 4

Camada de rede

Nota sobre o uso destes slides ppt:

Estamos disponibilizando estes slides gratuitamente a todos (professores, alunos, leitores). Eles estão em formato do PowerPoint para que você possa incluir, modificar e excluir slides (incluindo este) e o conteúdo do slide, de acordo com suas necessidades. Eles obviamente representam *muito* trabalho da nossa parte. Em retorno pelo uso, pedimos apenas o seguinte:

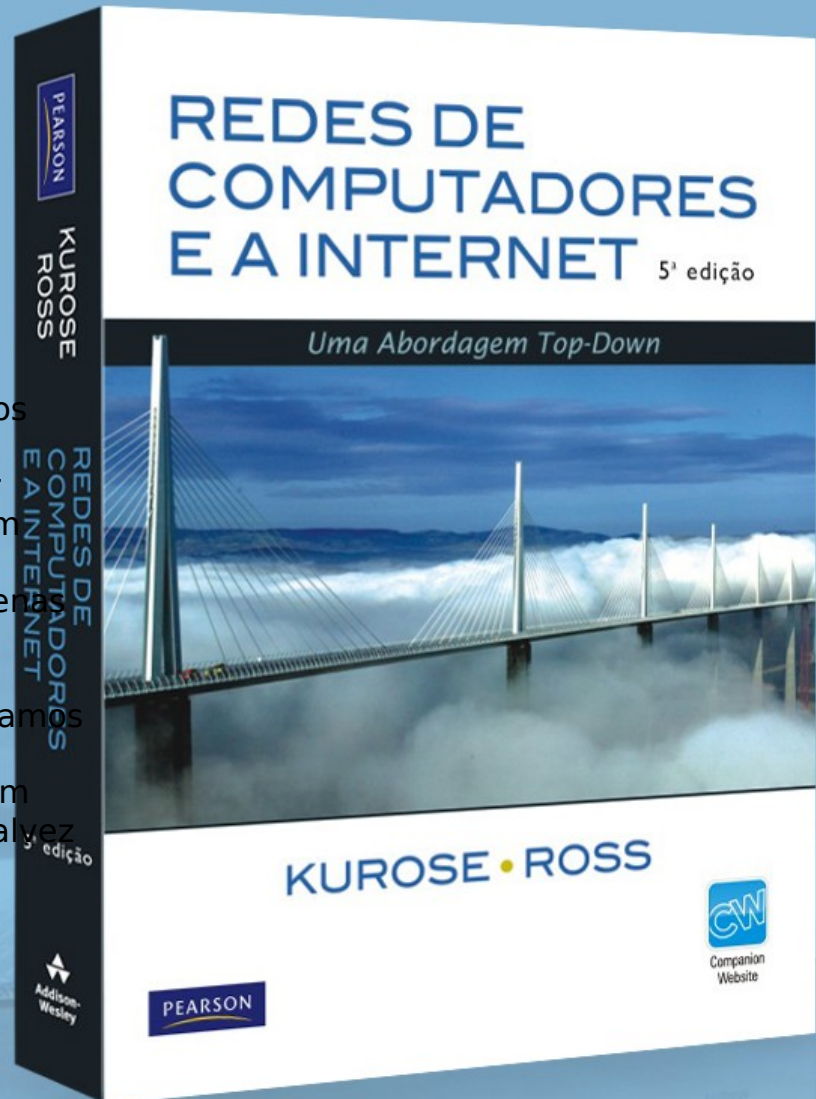
Se você usar estes slides (por exemplo, em sala de aula) sem muita alteração, que mencione sua fonte (afinal, gostamos que as pessoas usem nosso livro!).

Se você postar quaisquer slides sem muita alteração em um site Web, que informe que eles foram adaptados dos (ou talvez idênticos aos) nossos slides, e inclua nossa nota de direito autoral desse material.

Obrigado e divirta-se! JFK/KWR

Todo o material copyright 1996-2009

J. F Kurose e K. W. Ross, Todos os direitos reservados.



Capítulo 4: Camada de rede

Objetivos do capítulo:

- ❑ entender os princípios por trás dos serviços da camada de rede:
 - modelos de serviço da camada de rede
 - repasse *versus* roteamento
 - como funciona um roteador
 - roteamento (seleção de caminho)
 - lidando com escala
 - tópicos avançados: IPv6, mobilidade
- ❑ instanciação, implementação na Internet

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

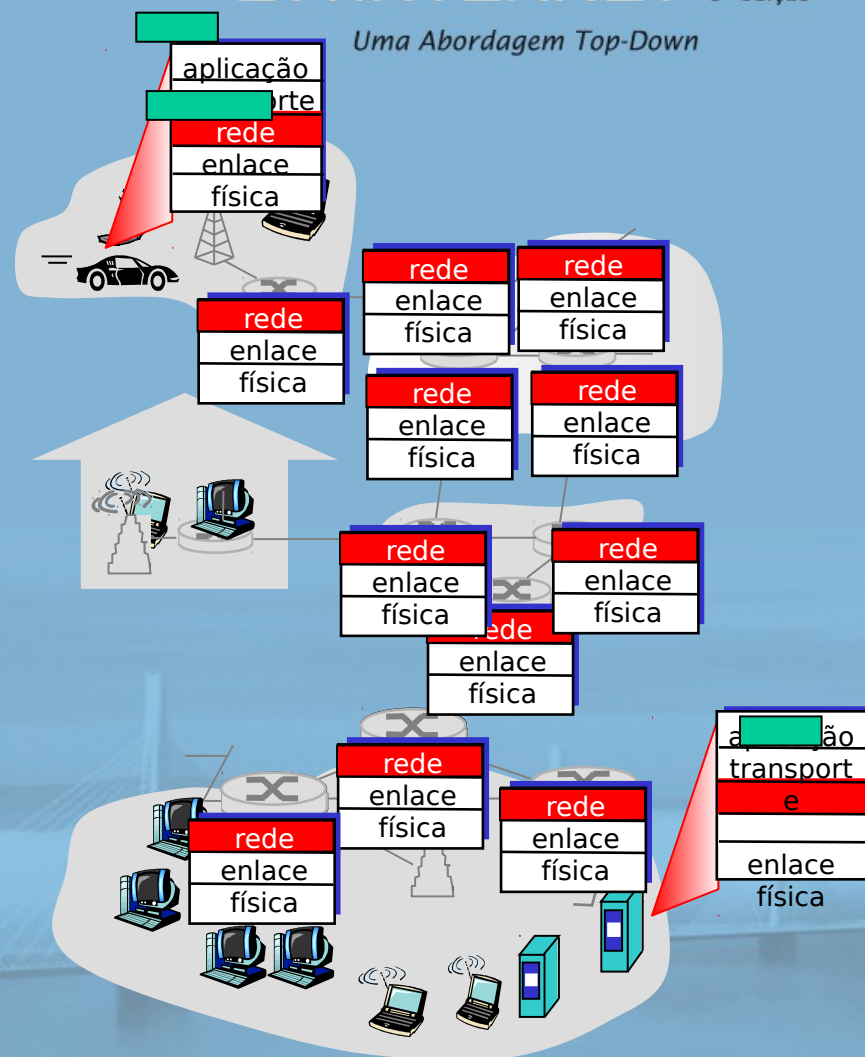
Camada de rede

- ❑ segmento de transporte do hosp. emissor ao receptor
- ❑ o lado emissor encapsula segmentos em datagramas
- ❑ o lado receptor entrega segmentos à camada de transporte
- ❑ protocolos da camada de rede em *cada* hosp., roteador
- ❑ roteador examina campos de cabeçalho em todos os datagramas IP que passam por ele

REDES DE COMPUTADORES E A INTERNET

5ª edição

Uma Abordagem Top-Down



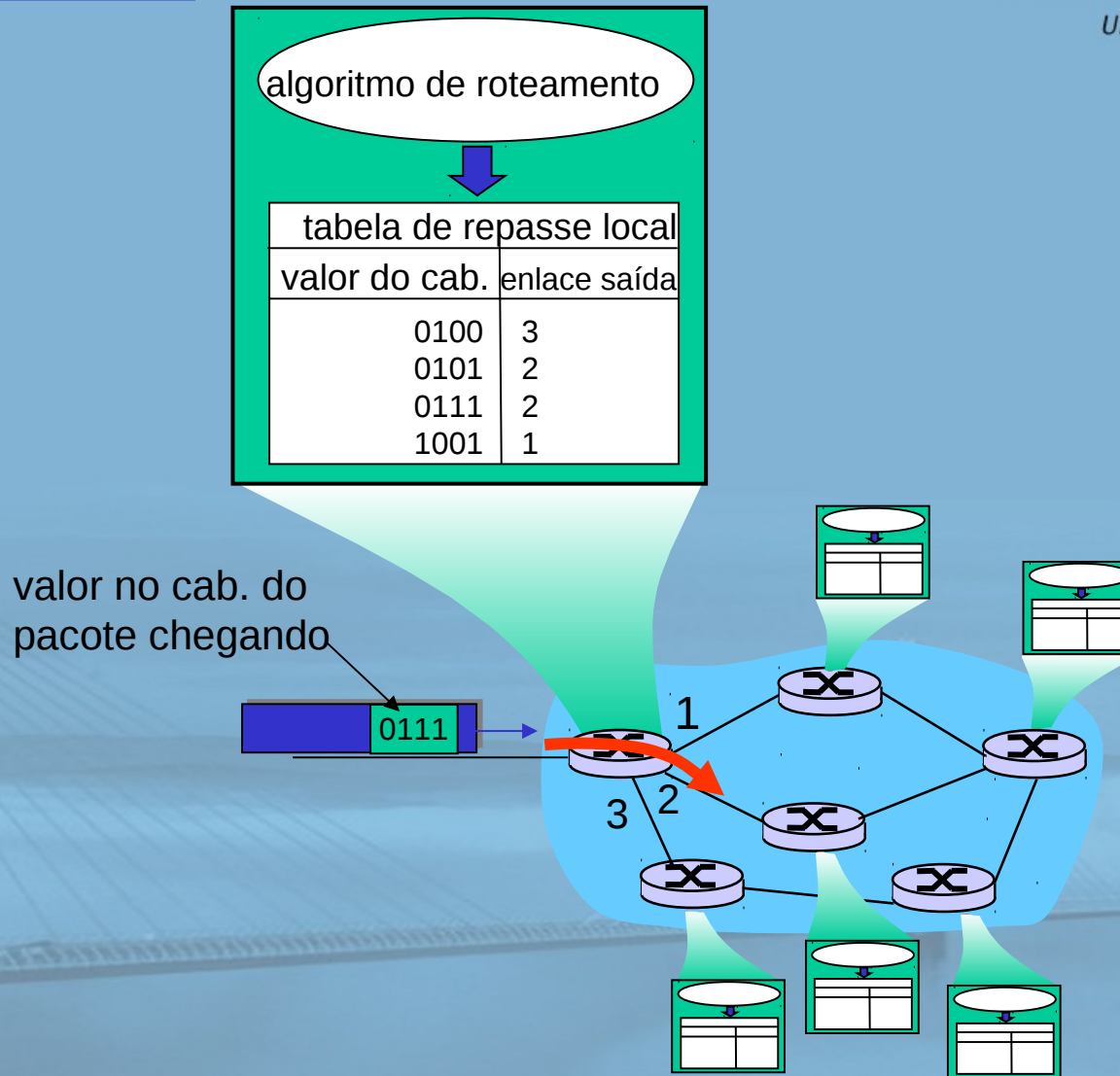
Duas importantes funções da camada de rede

- ❑ *repasse*: mover pacotes da entrada do roteador para a saída apropriada do roteador
- ❑ *roteamento*: determinar rota seguida pelos pacotes da origem ao destino
 - *algoritmos de roteamento*

analogia:

- ❑ *roteamento*: processo de planejamento da viagem da origem ao destino
- ❑ *repasse*: processo de passar por um único cruzamento

Interação entre roteamento e repasse



Estabelecimento de conexão

- ❑ 3ª função importante em *algumas* arquiteturas de rede:
 - ATM, frame relay, X.25
- ❑ antes que os datagramas fluam, dois hospedeiros finais e roteadores entre eles estabelecem conexão virtual
 - roteadores são envolvidos
- ❑ serviço de conexão da camada de rede *versus* transporte:
 - **rede**: entre dois hospedeiros (também pode envolver roteadores entre eles, no caso de VCs)
 - **transporte**: entre dois processos

Modelo de serviço de rede

P: Que *modelo de serviço* é o melhor para o “canal” que transporta datagramas do remetente ao destinatário?

exemplo de serviços para datagramas individuais:

- ❑ entrada garantida
- ❑ entrega garantida com atraso limitado

exemplo de serviços para fluxo de datagramas:

- ❑ entrega de datagrama na ordem
- ❑ largura de banda mínima garantida
- ❑ restrições sobre mudanças no espaçamento entre pacotes

Modelos de serviço da camada de rede:

Arquitetura da rede	Modelo de serviço	Garantia de largura de banda	Garantia contra perda	Ordenamento	Temporização	Indicação de congestionamento
Internet	Melhor esforço	Nenhuma	Nenhuma	Qualquer ordem possível	Não mantida	Nenhuma
ATM	CBR	Taxa constante garantida	Sim	Na ordem	Mantida	Não ocorrerá congestionamento
ATM	ABR	Mínima garantida	Nenhuma	Na ordem	Não mantida	Indicação de congestionamento

CBR (Constant Bit Rate) – não faz checagem de erros, controle de fluxo ou outro tipo de processamento de controle, como exemplo, temos o sistema telefônico;

ABR (Available Bit Rate) – utilizado em tráfego de rajadas cuja banda é fortemente conhecida. É a única categoria de serviço em que a rede provê feedback da taxa para o transmissor, pedindo a ele para diminuir o tráfego quando ocorre congestionamento. Assumindo que o transmissor concorda com tais requisições, a perda de células esperada é baixa, como exemplo temos o tráfego de browsers na Web;

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Serviço com e sem conexão da camada de rede

- ❑ rede de datagrama fornece serviço sem conexão da camada de rede
- ❑ rede VC fornece serviço com conexão da camada de rede
- ❑ análogo aos serviços da camada de transporte, mas:
 - **serviço:** hospedeiro a hospedeiro
 - **sem escolha:** a rede oferece um ou outro
 - **implementação:** no núcleo da rede

Circuitos virtuais

“Caminho da origem ao destino comporta-se como um circuito telefônico”

- com respeito ao desempenho
- ações da rede ao longo do caminho da origem ao destino

- ❑ estabelecimento e término para cada chamada *antes* que os dados possam fluir
- ❑ cada pacote carrega identificador VC (não endereço do hospedeiro de destino)
- ❑ *cada* roteador no caminho origem-destino mantém “estado” para cada conexão que estiver passando
- ❑ recursos do enlace e roteador (largura de banda, buffers) podem ser *alocados* ao VC (recursos dedicados = serviço previsível)

Implementação do VC

um VC consiste em:

1. caminho da origem ao destino
 2. números de VC, um número para cada enlace ao longo do caminho
 3. entradas em tabelas de repasse nos roteadores ao longo do caminho
- ❑ pacote pertencente ao VC carrega número do VC (em vez do endereço de destino)
 - ❑ número do VC pode ser alterado em cada enlace
 - novo número de VC vem da tabela de repasse

Tabela de repasse

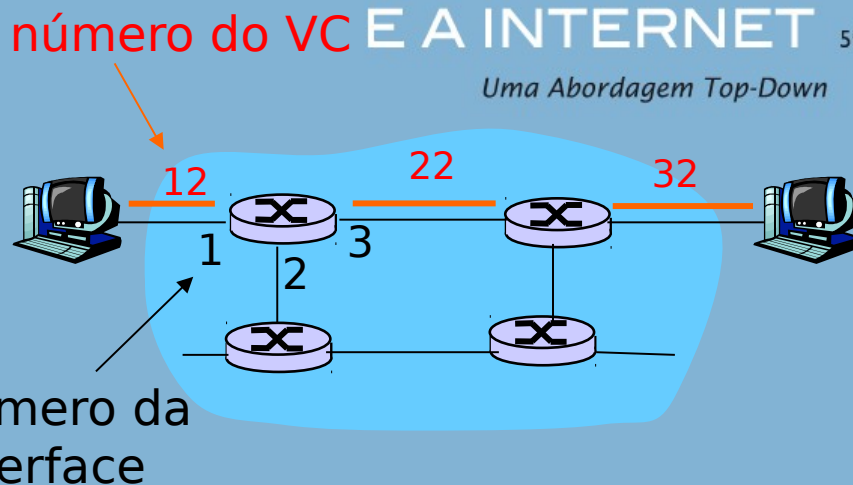


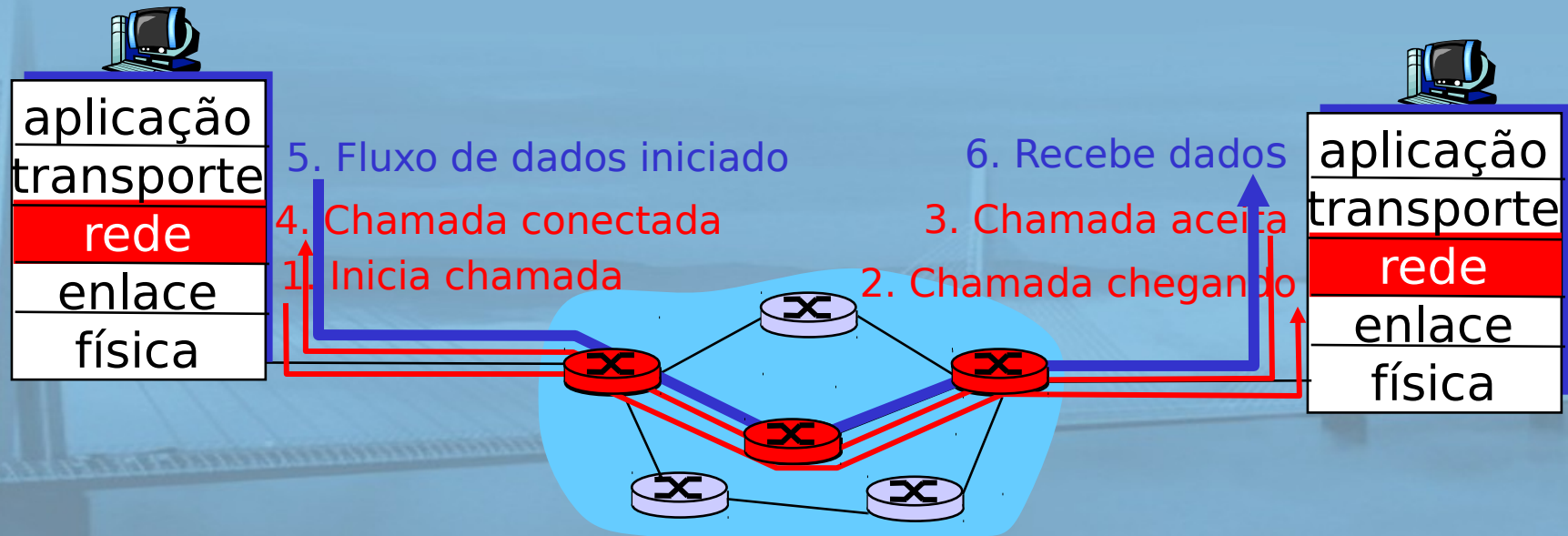
tabela de repasse no roteador noroeste:

Interface de entrada	Nº do CV de entrada	Interface de saída	Nº do CV de saída
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...

Roteadores mantêm informação de estado da conexão!

Circuitos virtuais: protocolos de sinalização

- ❑ usados para estabelecer, manter e terminar VC
- ❑ usados em ATM, frame-relay, X.25
- ❑ não usados na Internet de hoje



Redes de datagrama

- ❑ sem estabelecimento de chamada na camada de rede
- ❑ roteadores: sem estado sobre conexões fim a fim
 - sem conceito em nível de rede da “conexão”
- ❑ pacotes repassados usando endereço do hospedeiro de destino
 - pacotes entre mesmo par origem-destino podem tomar caminhos diferentes

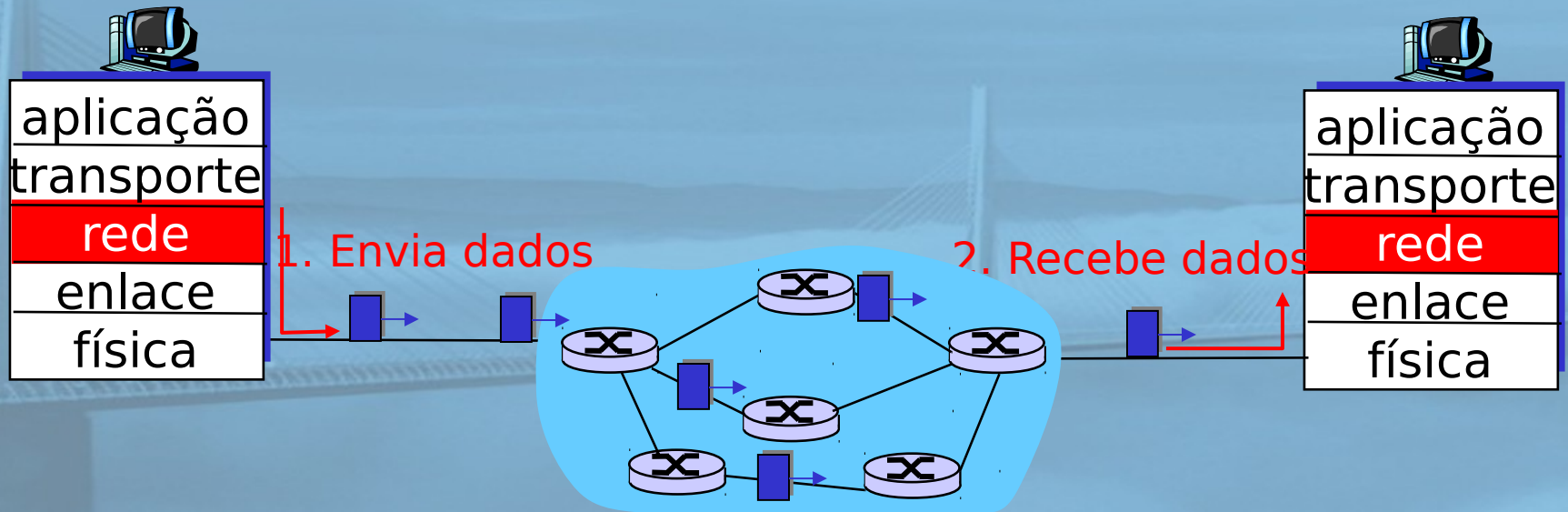


Tabela de repasse

4 bilhões de entradas
possíveis

Faixa de endereços de destino

Interface de enlace

11001000 00010111 00010000 00000000

até

0

11001000 00010111 00010111 11111111

11001000 00010111 00011000 00000000

até

1

11001000 00010111 00011000 11111111

11001000 00010111 00011001 00000000

até

2

11001000 00010111 00011111 11111111

senão

3

Rede de datagramas ou VC: por quê?

Internet (datagrama)

- ❑ troca de dados entre computadores
 - serviço “elástico”, sem requisitos de temporização estritos
- ❑ sistemas finais “inteligentes” (computadores)
 - pode adaptar, realizar controle, recup. de erros
 - simples dentro da rede, complexidade na “borda”
- ❑ muitos tipos de enlace
 - diferentes características
 - serviço uniforme difícil

ATM (VC)

- ❑ evoluída da telefonia
- ❑ conversação humana:
 - requisitos de temporização estritos, confiabilidade
 - necessário para serviço garantido
- ❑ sistemas finais “burros”
 - telefones
 - complexidade dentro da rede

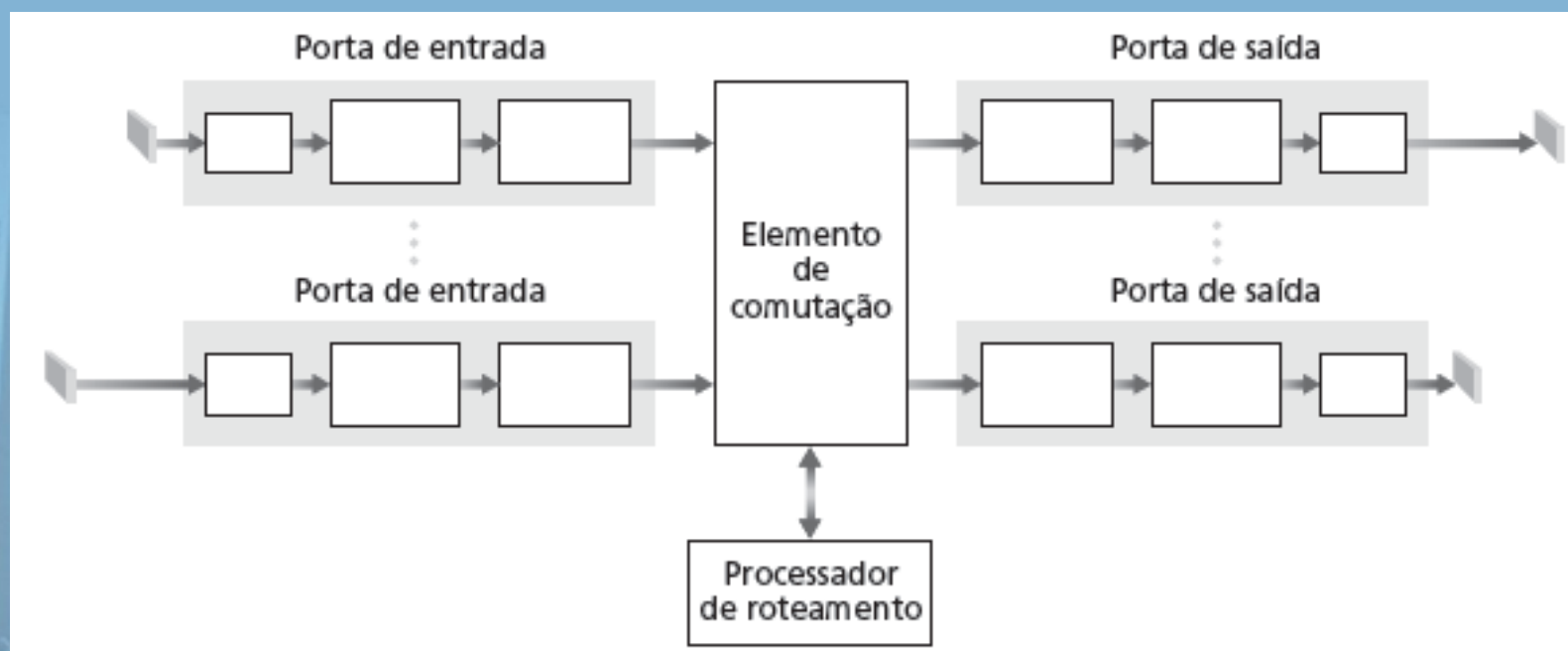
Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

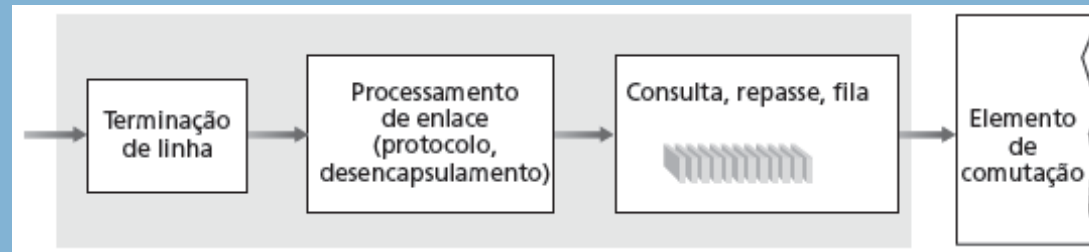
Visão geral da arquitetura do roteador

Duas funções principais do roteador:

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



Funções da porta de entrada



Camada física:
recepção por bit

Camada de enlace
de dados:
p. e., Ethernet
ver Capítulo 5

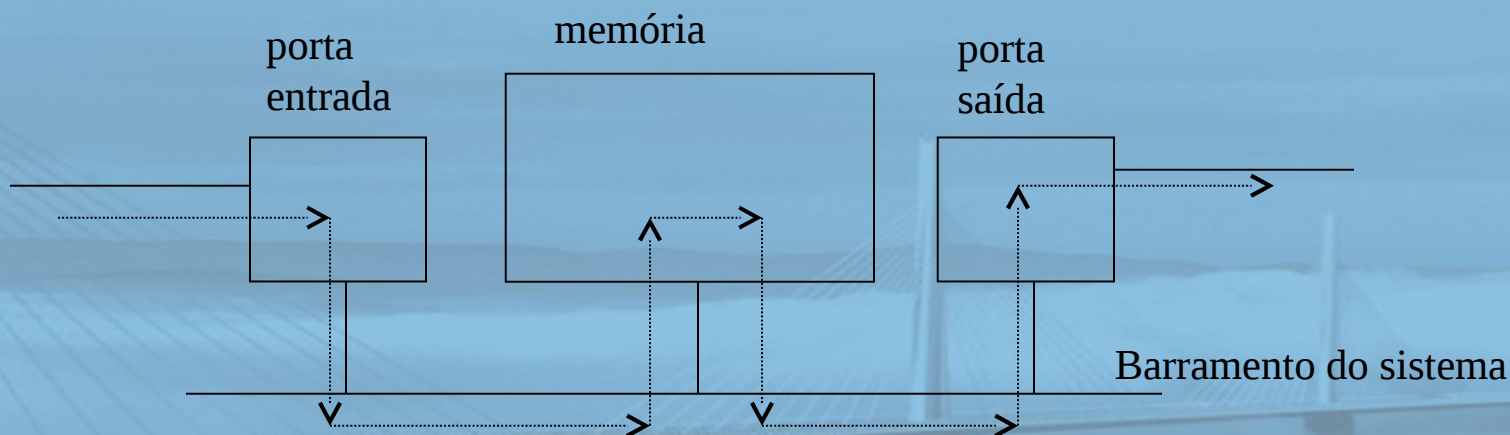
Comutação descentralizada:

- ❑ dado destino do datagrama, porta de saída de pesquisa usando tabela de repasse na memória da porta de entrada
- ❑ objetivo: processamento completo da porta de entrada na 'velocidade de linha'
- ❑ fila: se datagramas chegarem mais rápido que taxa de repasse no elemento de comutação

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



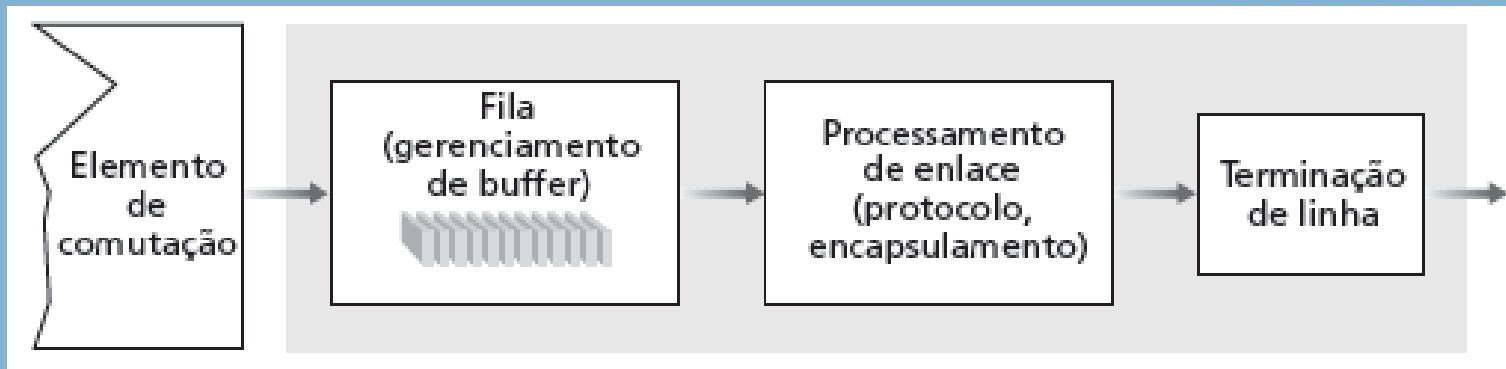
Comutação por um barramento

- ❑ datagrama da memória da porta de entrada à memória da porta de saída por um barramento compartilhado
- ❑ **disputa pelo barramento:** velocidade da comutação limitada pela largura de banda do barramento
- ❑ barramento Cisco 5600 de 32 Gbps: velocidade suficiente para roteadores de acesso e corporativos

Comutação por uma rede de interconexão

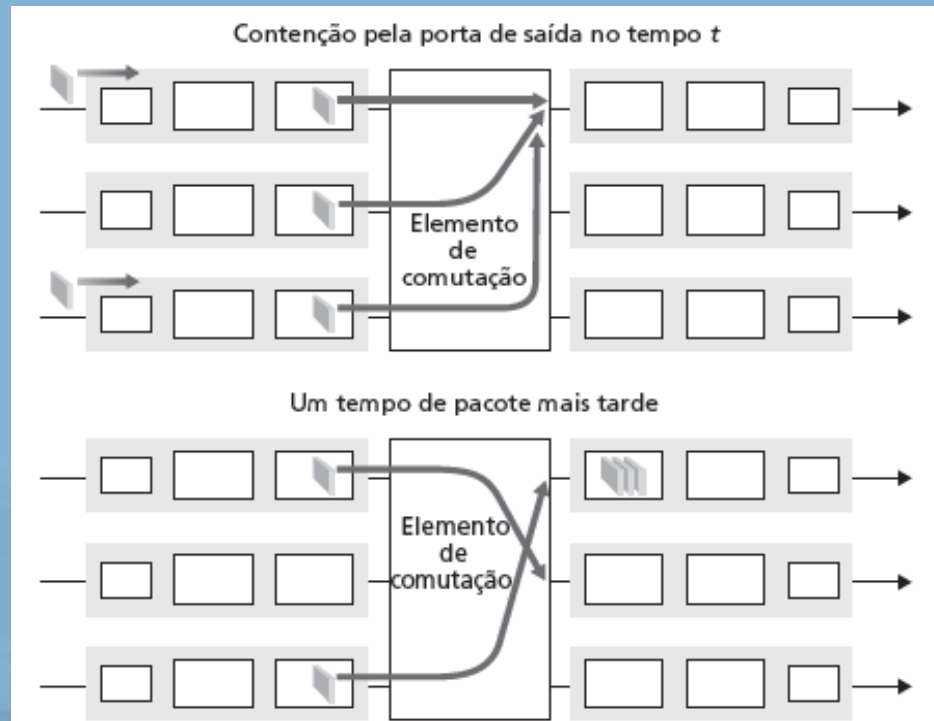
- ❑ contorna limitações de largura de banda do barramento
- ❑ redes Banya, outras redes de interconexão desenvolvidas inicialmente para conectar processadores no multiprocessador
- ❑ projeto avançado: fragmenta datagrama em células de tamanho fixo, comuta células através do elemento de comutação
- ❑ Cisco 12000: comuta 60 Gbps através da rede de interconexão

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

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!*

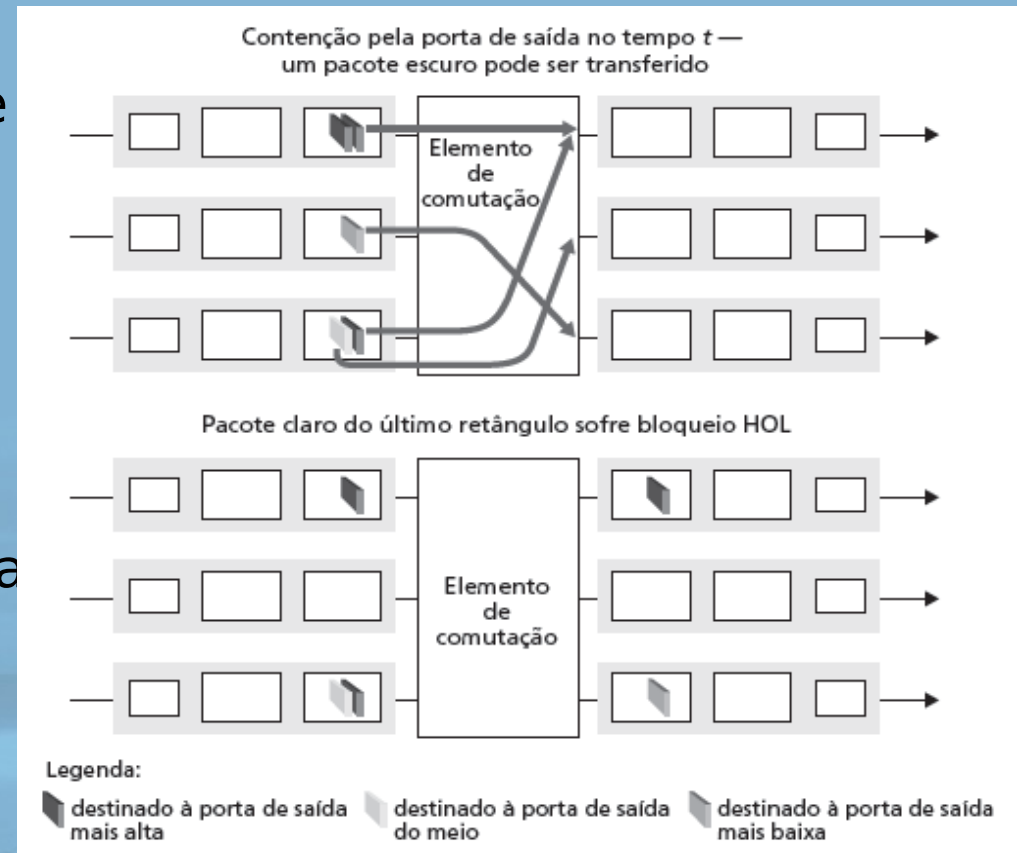
Quanto armazenamento em buffer?

- ❑ regra prática da RFC 3439:
armazenamento médio em buffer igual à RTT “típica” (digamos, 250 ms) vezes capacidade do enlace C
 - p. e., C = enlace de 10 Gps: buffer de 2,5 Gbit
- ❑ recomendação recente: com N fluxos, armazenamento deve ser igual a

$$\frac{RTT \cdot C}{\sqrt{N}}$$

Enfileiramento da 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**

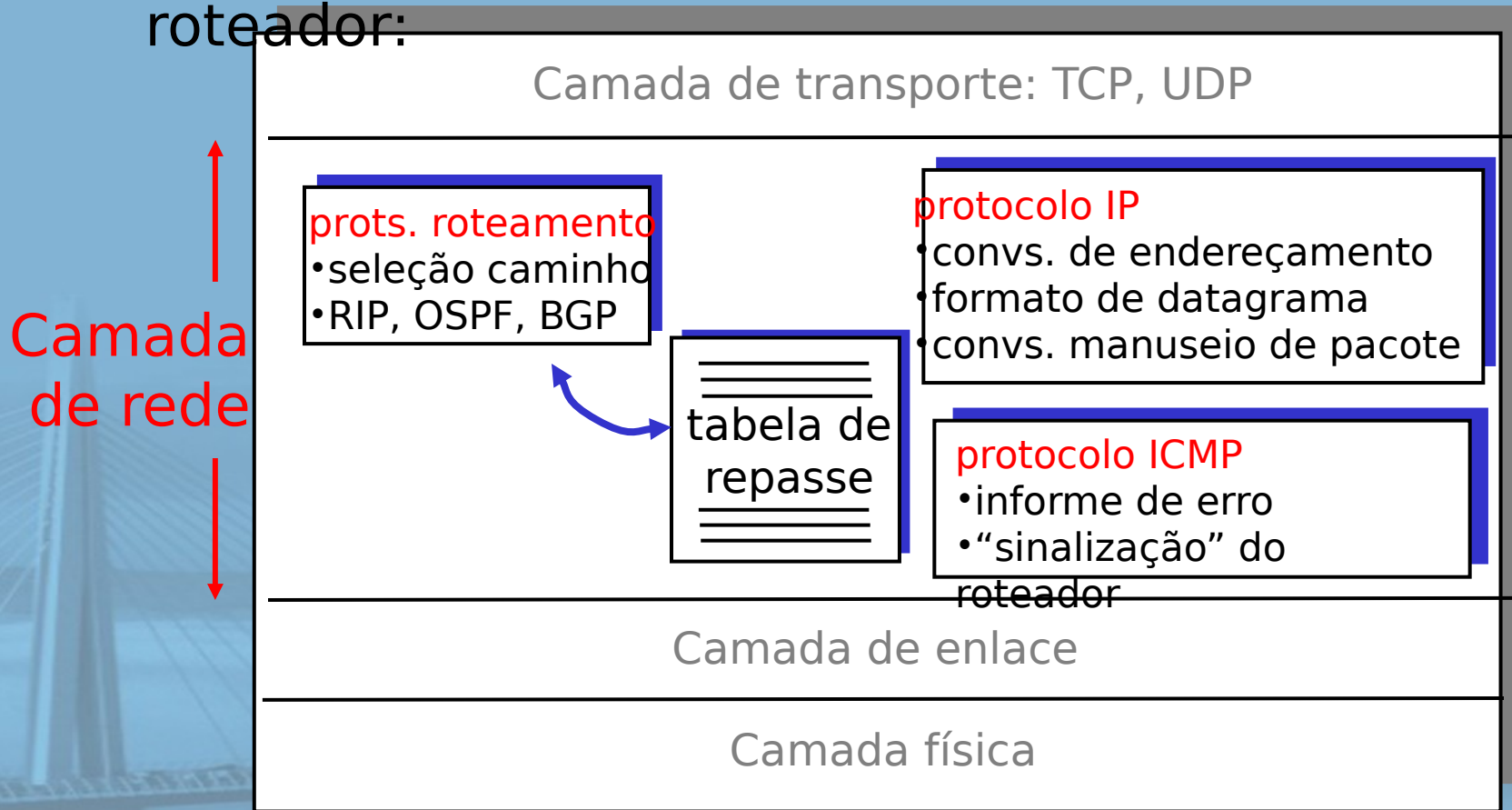


Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

A camada de rede da Internet

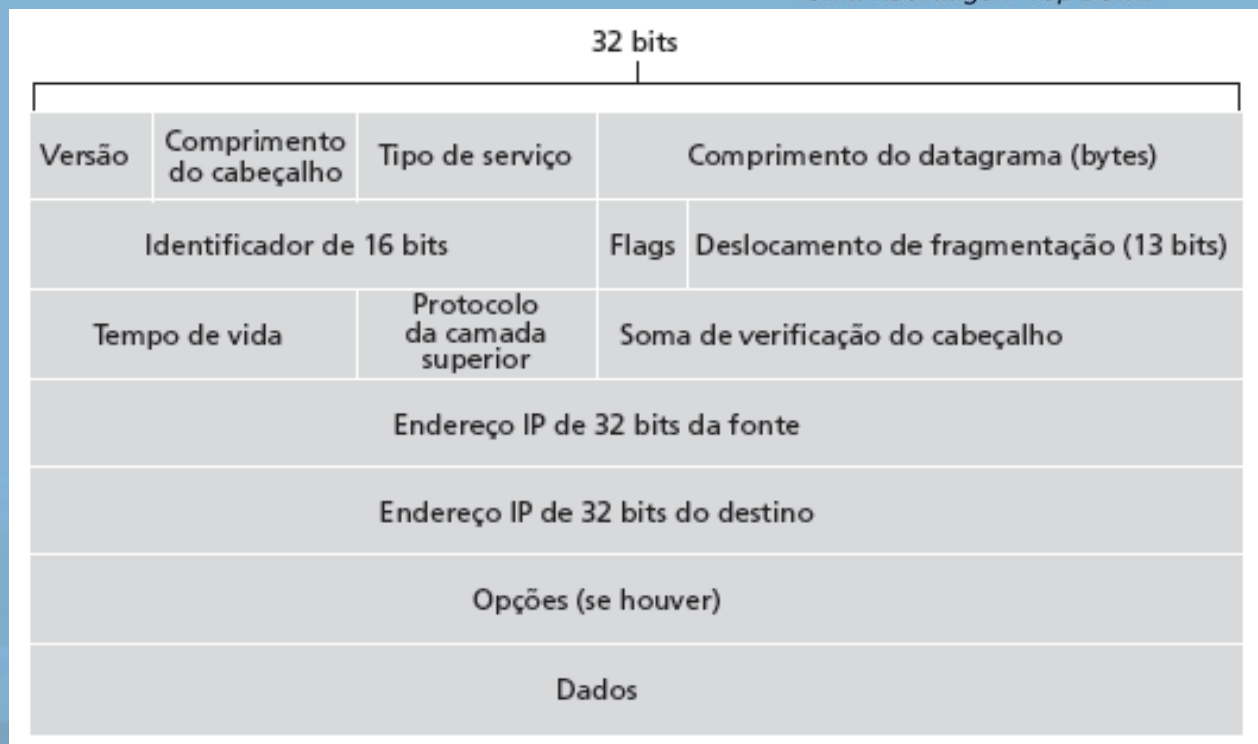
Funções na camada de rede do hospedeiro e roteador:



Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Formato do datagrama IP

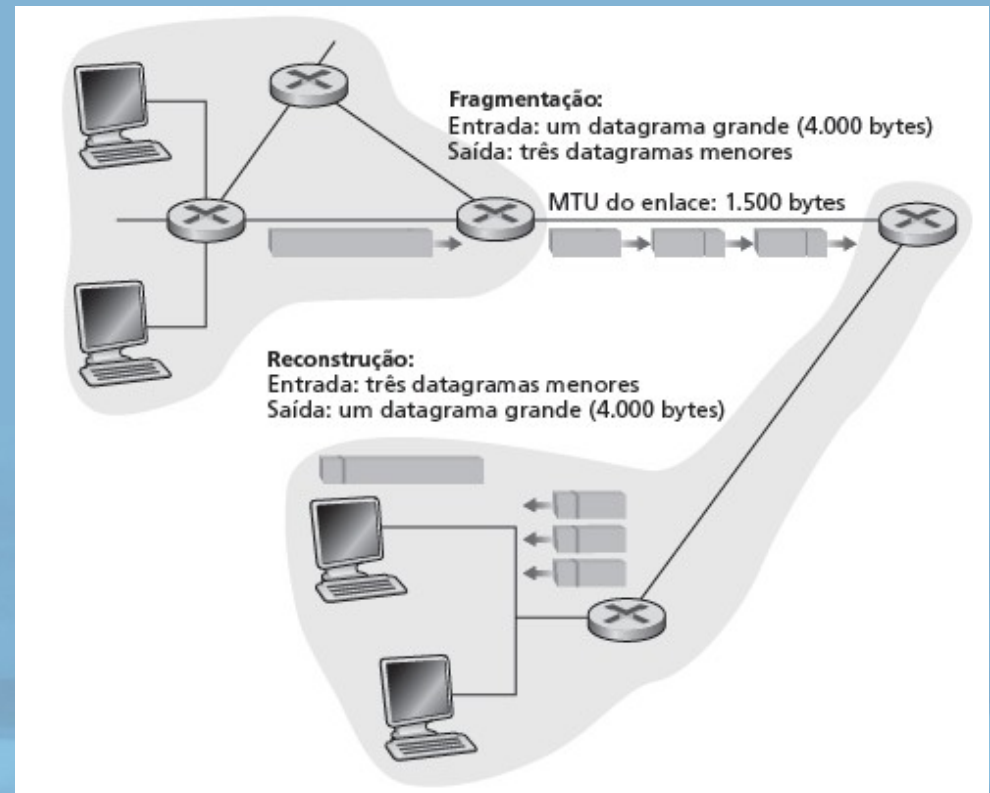


Quanto overhead com TCP?

- ❑ 20 bytes de TCP
- ❑ 20 bytes de IP
- ❑ = 40 bytes + overhead da camada de aplicação

Fragmentação e reconstrução do IP

- ❑ enlaces de rede têm MTU (tamanho máx. transferência)
 - maior quadro em nível de enlace possível.
 - diferentes tipos de enlace, diferentes MTUs
- ❑ grande datagrama IP dividido (“fragmentado”) dentro da rede
 - um datagrama torna-se vários datagramas
 - “reconstruído” somente no destino final
 - bits de cabeçalho IP usados para identificar, ordenar fragmentos relacionados



Exemplo

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

1480 bytes no campo de dados

deslocamento = $1480/8$

	tam.	ID	fragflag	desloc.	
	= 4000	= x	= 0	= 0	

Um datagrama grande torna-se vários datagramas menores

	tam.	ID	fragflag	desloc.	
	= 1500	=	= 1	= 0	

x

	tam.	ID	fragflag	desloc.	
	= 1500	= x	= 1	= 185	

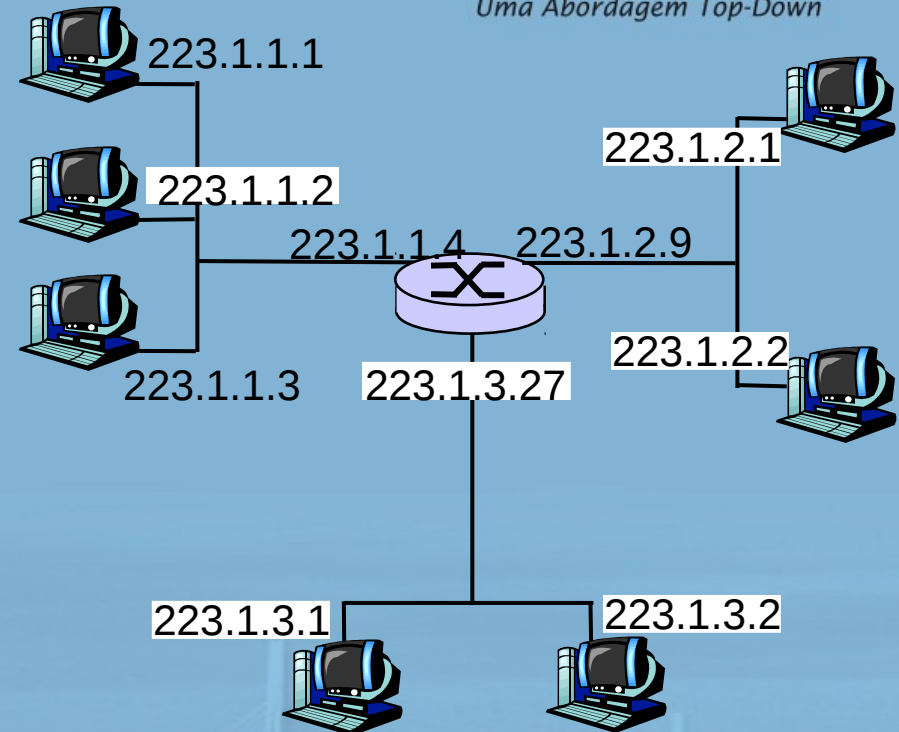
	tam.	ID	fragflag	desloc.	
	= 1040	= x	= 0	= 370	

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Endereçamento IP: introdução

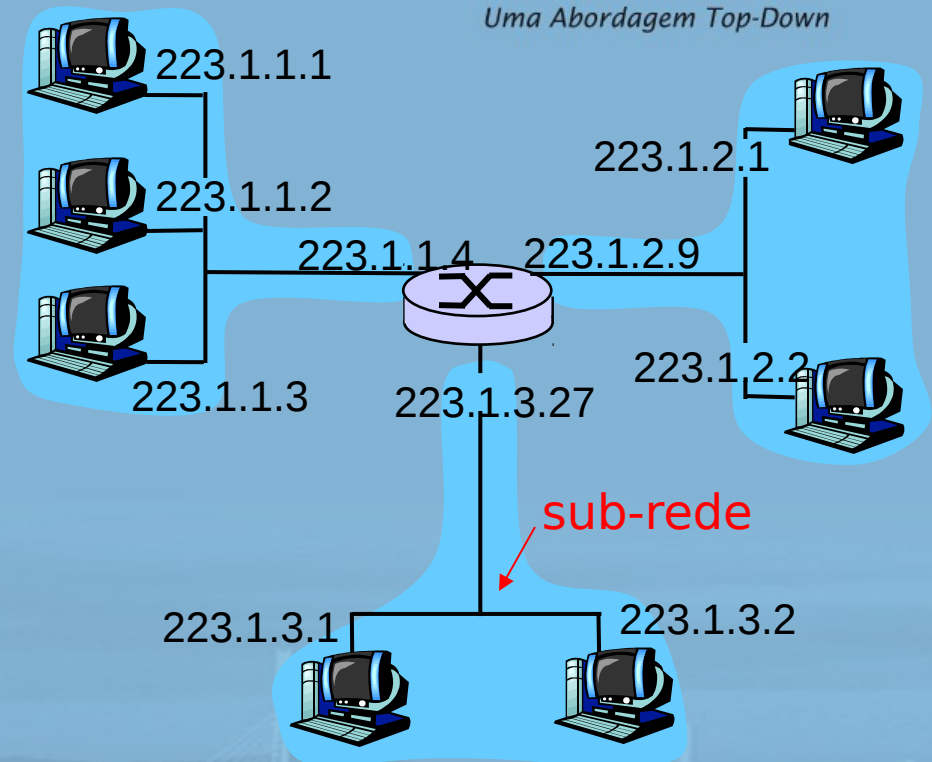
- endereço IP: identificador de 32 bits para *interface de* hospedeiro e roteador
- *interface*: conexão entre hospedeiro/roteador e enlace físico
 - roteadores normalmente têm várias interfaces
 - hospedeiro normalmente tem uma interface



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

Sub-redes

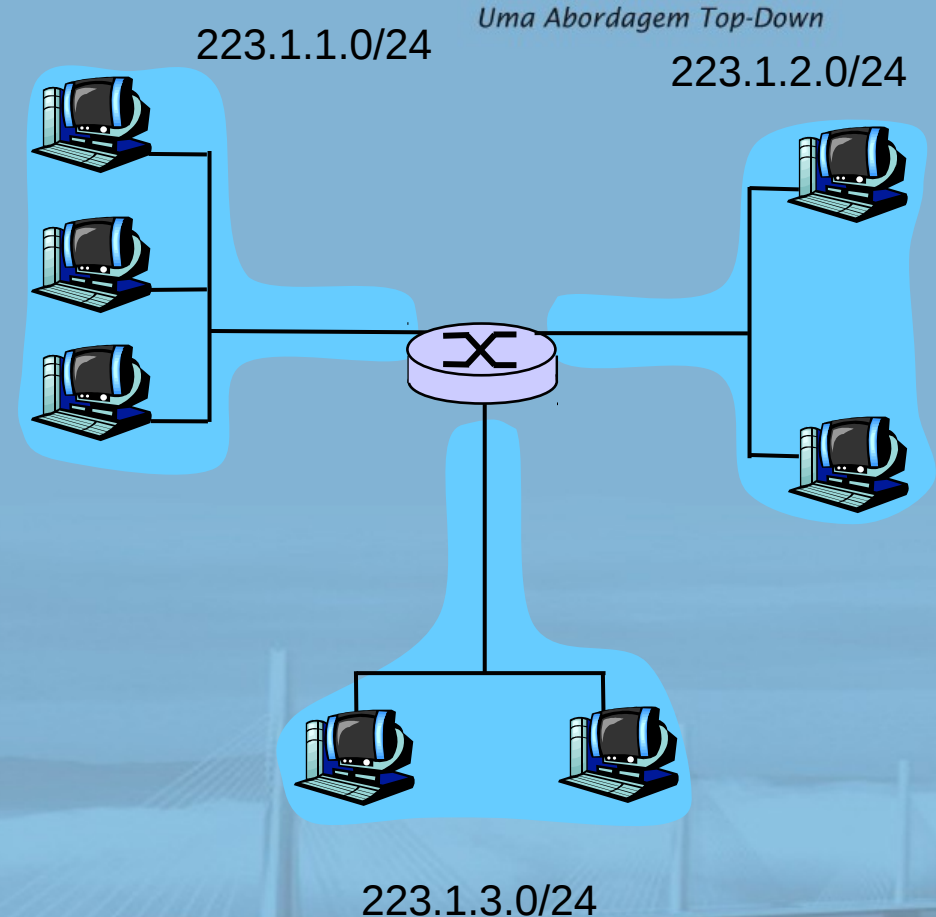
- ❑ endereço IP:
 - parte da sub-rede (bits de alta ordem)
 - parte do host (bits de baixa ordem)
- ❑ *O que é uma sub-rede?*
 - dispositivo se conecta à mesma parte da sub-rede do endereço IP
 - pode alcançar um ao outro fisicamente sem roteador intermediário



rede consistindo em 3 sub-redes

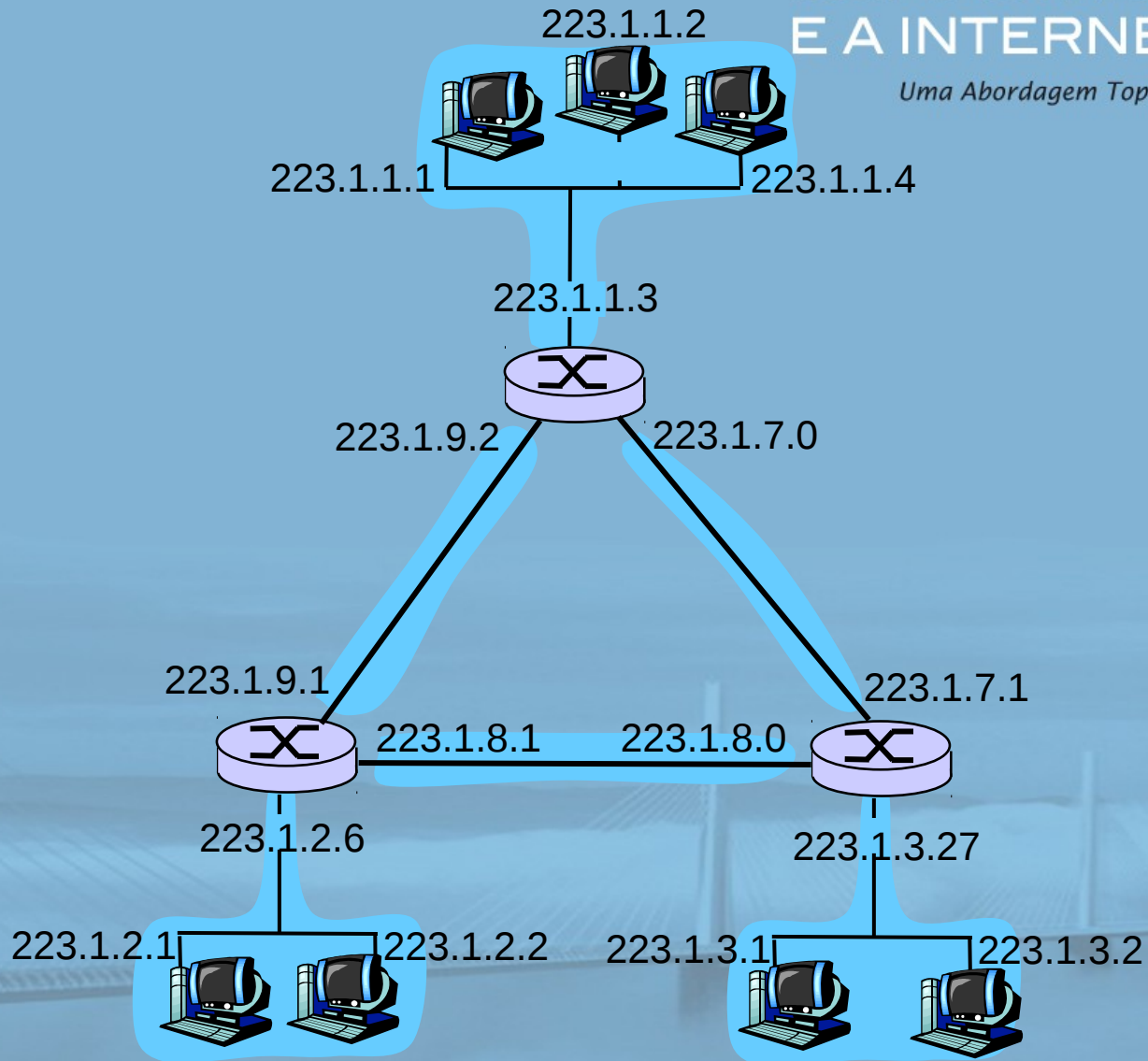
Receita

- para determinar as sub-redes, destaque cada interface de seu hospedeiro ou roteador, criando ilhas de redes isoladas. Cada rede isolada é denominada **sub-red**



Máscara de sub-rede: /24

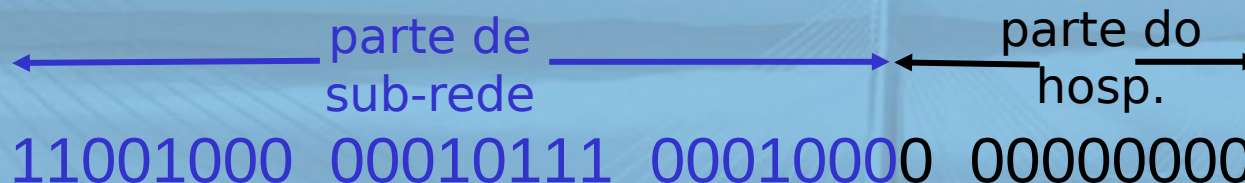
Quantas?



Endereçamento IP: CIDR

CIDR: Classless InterDomain Routing
(roteamento interdomínio sem classes)

- parte de sub-rede do endereço de tamanho arbitrário
- formato do endereço: **a.b.c.d/x**, onde x é # bits na parte de sub-rede do endereço



200.23.16.0/23

Endereços IP: como obter um?

P: Como um *hospedeiro* obtém endereço IP?

- ❑ fornecido pelo administrador do sistema em um arquivo
 - Windows: painel de controle->rede
->configuração->tcp/ip->propriedades
 - UNIX: /etc/rc.config
- ❑ **DHCP: D**ynamic **H**ost **C**onfiguration **P**rotocol:
recebe endereço dinamicamente do servidor
 - “plug-and-play”

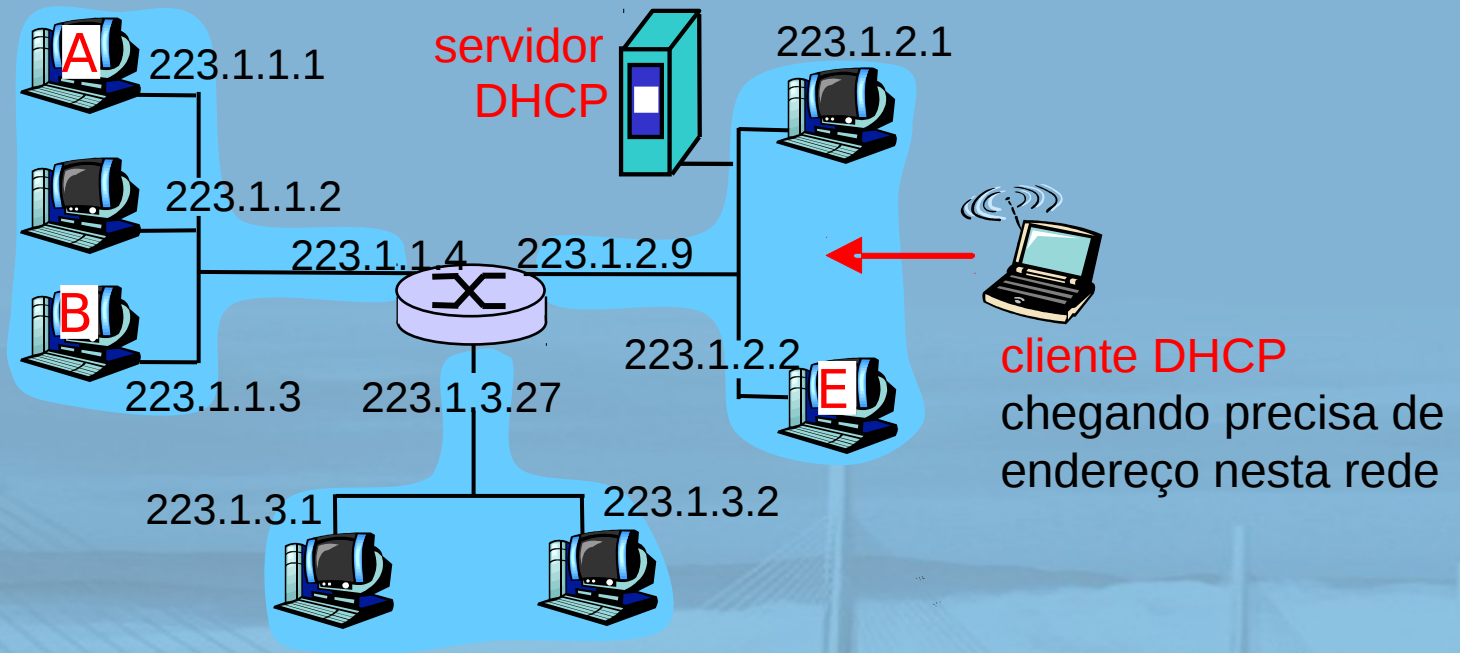
DHCP: Dynamic Host Configuration Protocol

Objetivo: permitir que o hospedeiro obtenha *dinamicamente* seu endereço IP do servidor de rede quando se conectar à rede
pode renovar seu prazo no endereço utilizado
permite reutilização de endereços (só mantém endereço enquanto conectado e “ligado”)
aceita usuários móveis que queiram se juntar à rede (mais adiante)

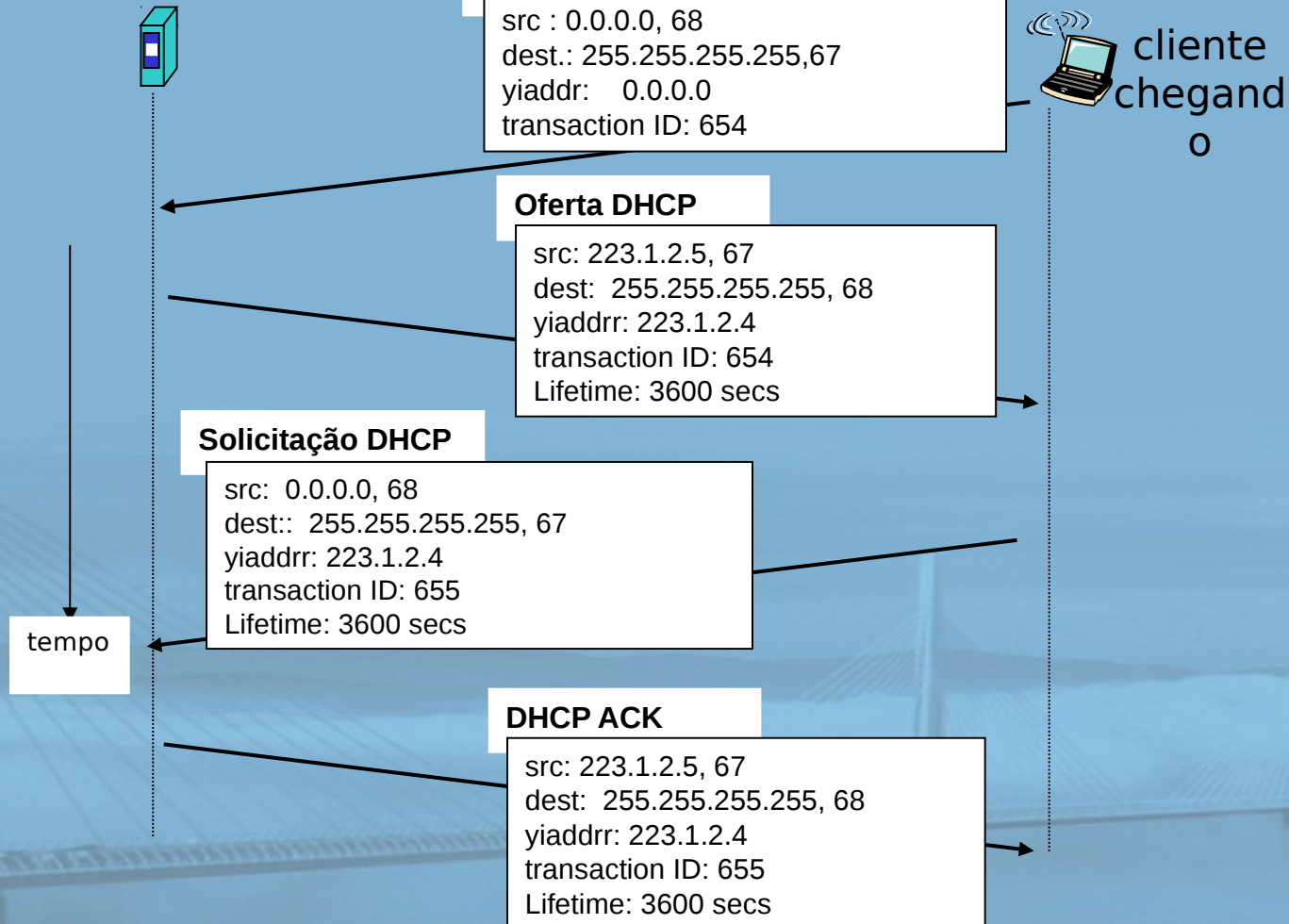
Visão geral do DHCP:

- host broadcasts “**DHCP discover**” msg [optional]
- servidor DHCP responde com msg “**DHCP offer**” [opcional]
- hospedeiro requer endereço IP: msg “**DHCP request**”
- servidor DHCP envia endereço: msg “**DHCP ack**”

DHCP – cenário cliente/servidor



servidor DHCP: 223.1.2.5

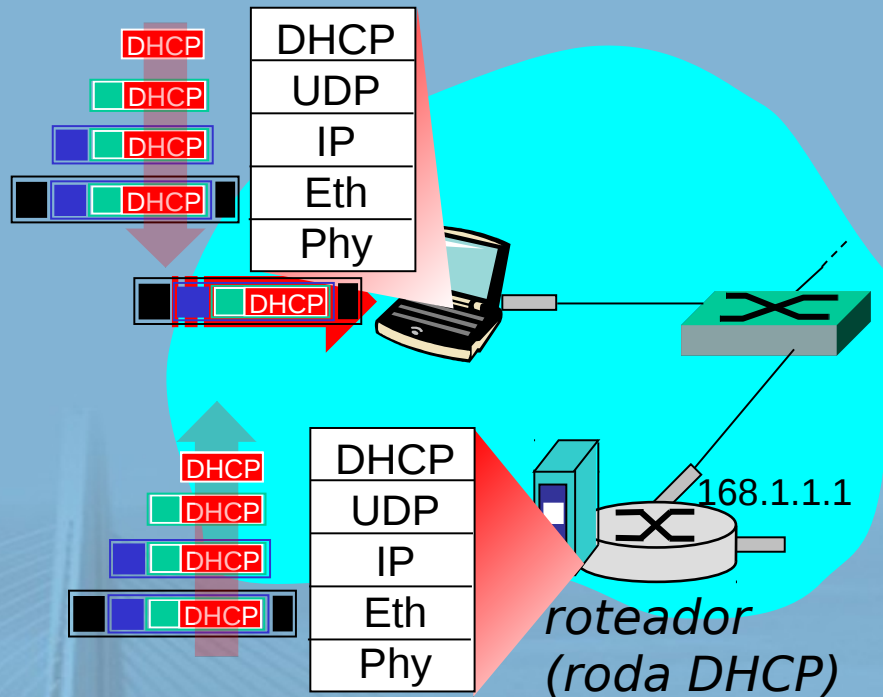


DHCP: mais do que endereço IP

DHCP pode retornar mais do que apenas o endereço IP alocado na sub-rede:

- endereço do roteador do primeiro salto para o cliente
- nome e endereço IP do servidor DNS
- máscara de rede (indicando parte de rede *versus* hospedeiro do endereço)

DHCP: exemplo

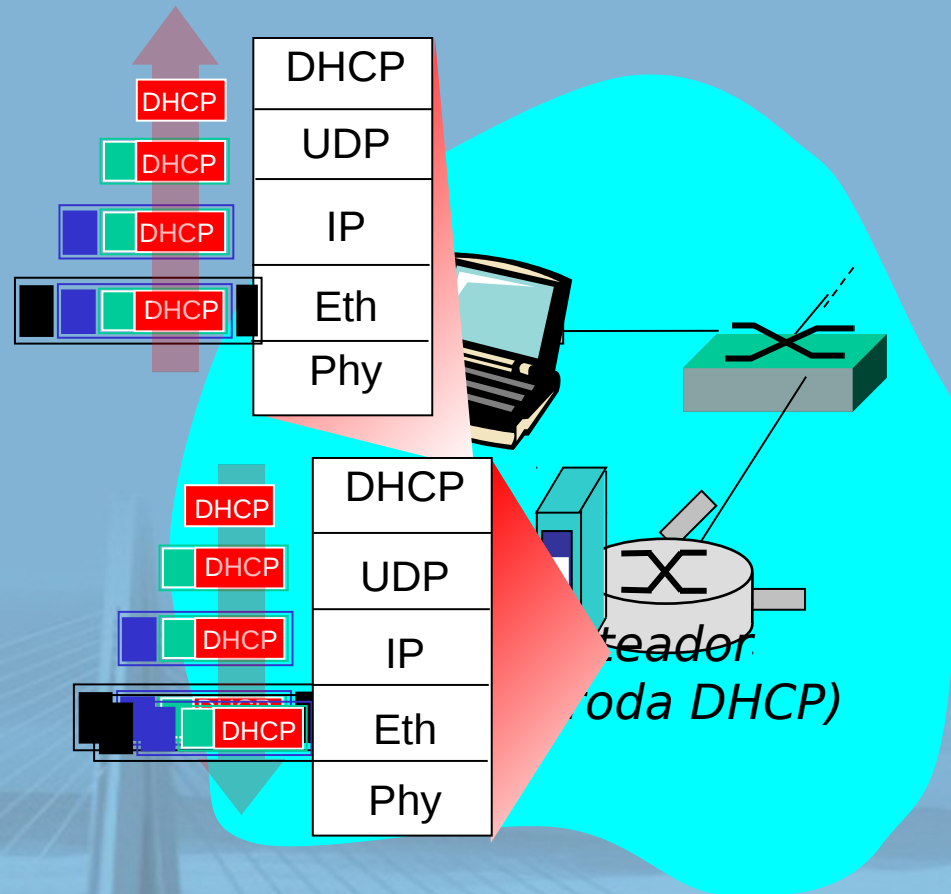


- ❑ conexão de laptop precisa do seu endereço IP, endereço do roteador do primeiro salto, endereço do servidor DNS: use DHCP
- ❑ solicitação DHCP encapsulada no UDP, encapsulada no IP, encapsulado no Ethernet 802.1
- ❑ broadcast de quadro Ethernet (dest: FFFFFFFFFFFFFFFF) na LAN, recebido no roteador rodando DHCP
- ❑ Ethernet demultiplexado para IP demultiplexado, UDP demultiplexado para DHCP

REDES DE COMPUTADORES E A INTERNET

5ª edição

Uma Abordagem Top-Down



- servidor DHCP formula DHCP ACK contendo endereço IP do cliente, endereço IP do roteador do primeiro salto para cliente, nome & endereço IP do servidor DNS
- roteador repassa ao cliente, demultiplexando para DHCP no cliente
- cliente agora sabe seu endereço IP, nome e endereço IP do servidor DNS, endereço IP do seu roteador do primeiro salto

DHCP: Saída wireshark (LAN doméstica)

solicitação

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t = 53,l = 1) **DHCP Message Type = DHCP Request**
Option: (61) Client identifier
 Length: 7; Value: 010016D323688A;
 Hardware type: Ethernet
 Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t = 50,l = 4) Requested IP Address = 192.168.1.101
Option: (t = 12,l = 5) Host Name = "nomad"
Option: (55) Parameter Request List
 Length: 11; Value: 010F03062C2E2F1F21F92B
 1 = Subnet Mask; 15 = Domain Name
 3 = Router; 6 = Domain Name Server
 44 = NetBIOS over TCP/IP Name Server

resposta

Message type: **Boot Reply (2)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t = 53,l = 1) DHCP Message Type = DHCP ACK
Option: (t = 54,l = 4) Server Identifier = 192.168.1.1
Option: (t = 1,l = 4) Subnet Mask = 255.255.255.0
Option: (t = 3,l = 4) Router = 192.168.1.1
Option: (6) Domain Name Server
 Length: 12; Value: 445747E2445749F244574092;
 IP Address: 68.87.71.226;
 IP Address: 68.87.73.242;
 IP Address: 68.87.64.146
Option: (t = 15,l = 20) Domain Name = "hsd1.ma.comcast.net."

Endereços IP: como obter um?

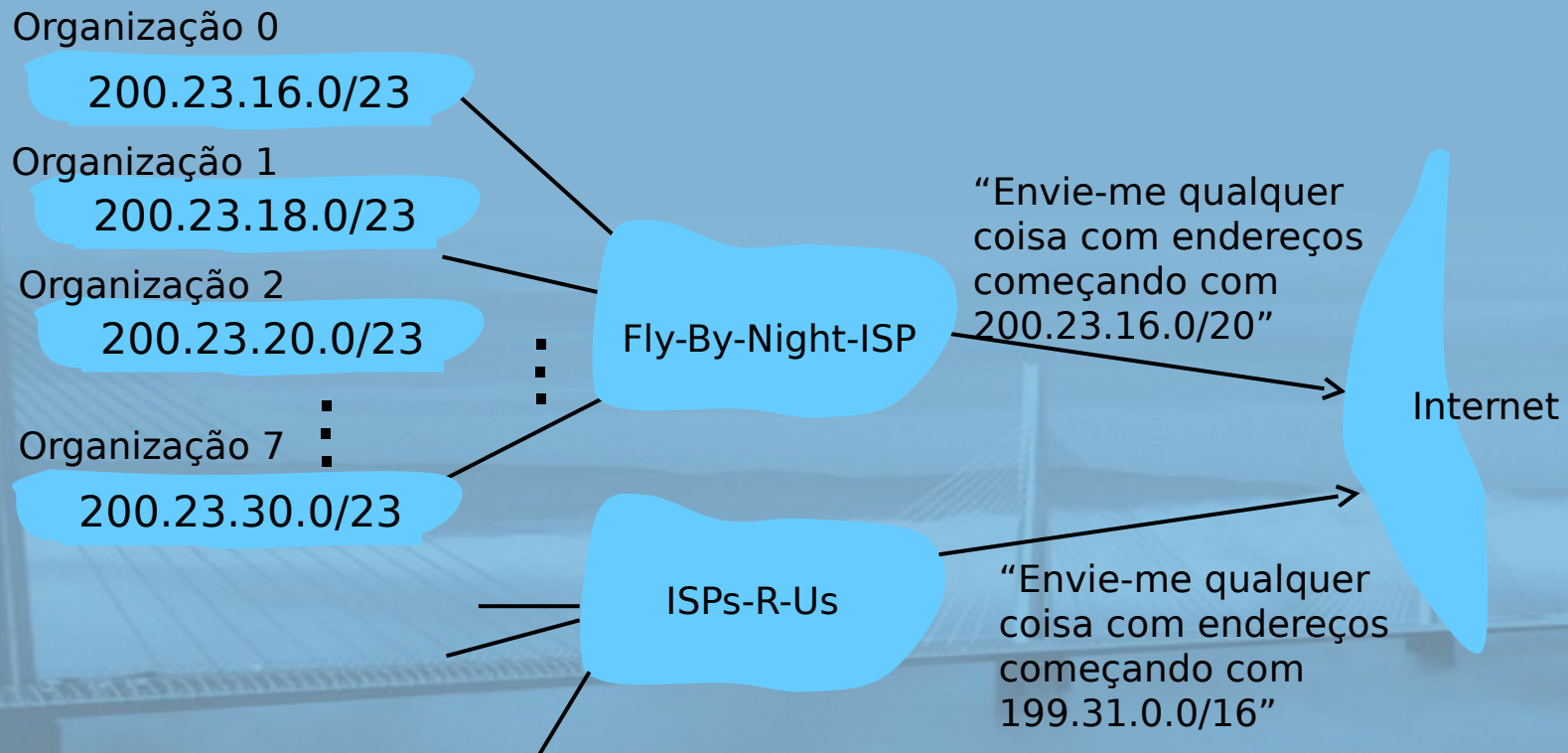
P: Como a *rede* obtém a parte de sub-rede do endereço IP?

R: Recebe parte alocada do espaço de endereços do seu 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

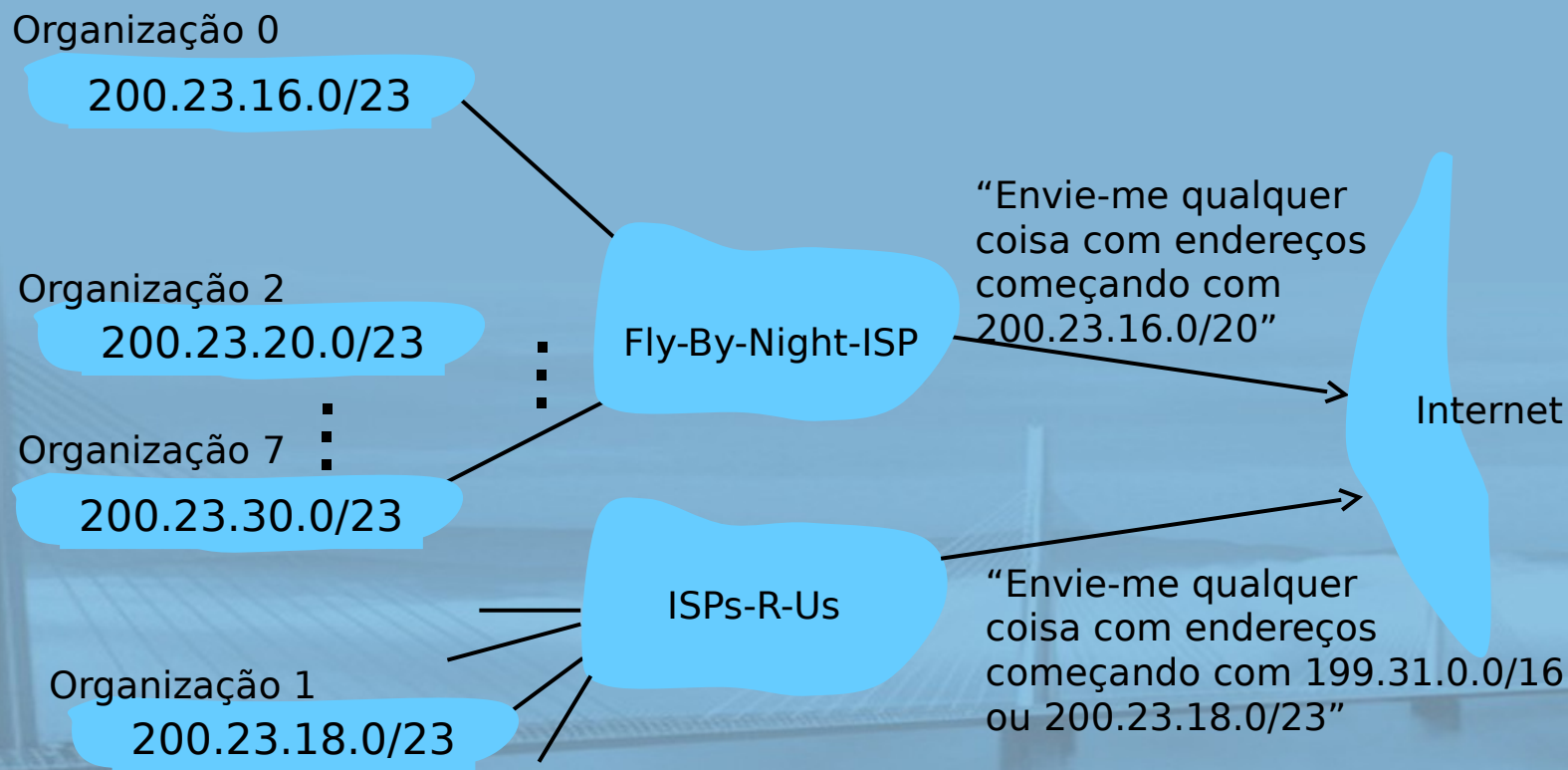
Endereçamento hierárquico: agregação de rota

Endereçamento hierárquico permite anúncio eficiente da informação de roteamento:



Endereçamento hierárquico: rotas mais específicas

ISPs-R-Us tem uma rota mais específica para Organização 1



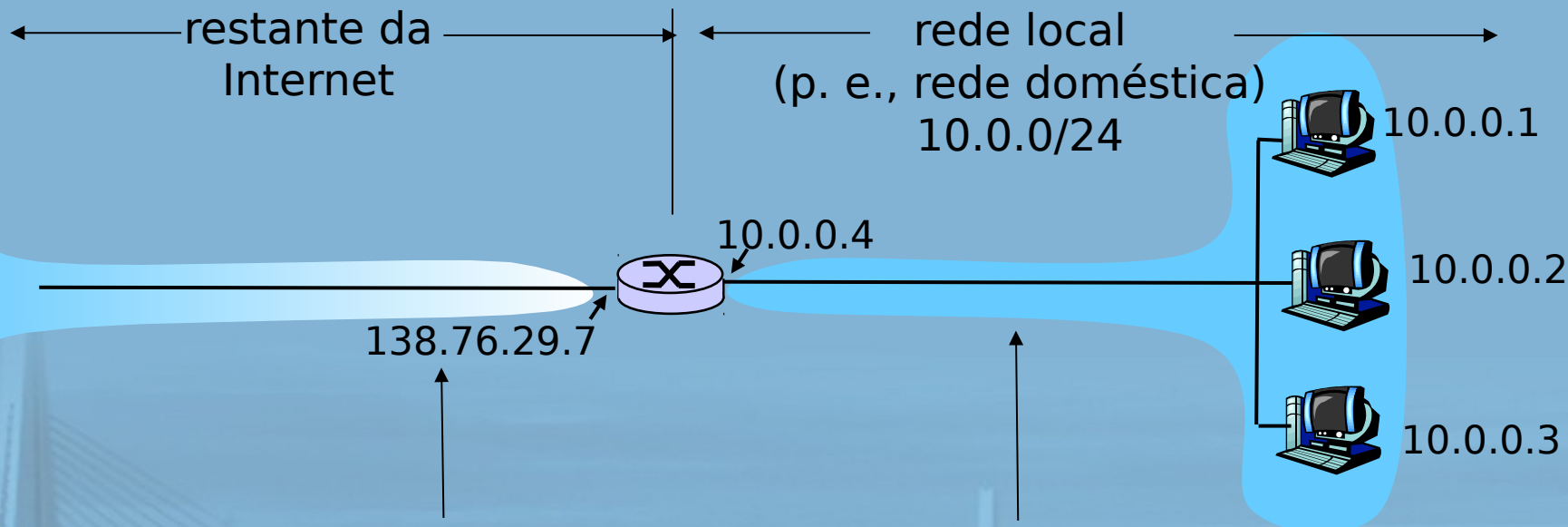
Endereçamento IP: a última palavra...

P: Como um ISP recebe bloco de endereços?

R: **ICANN**: Internet **C**orporation for **A**ssigned **N**ames and **N**umbers

- aloca endereços
- administra o DNS
- atribui nomes de domínio e resolve disputas

NAT: Network Address Translation



todos os datagramas *saindo* da rede local têm *mesmo* endereço IP NAT de origem: 138.76.29.7, mas diferentes números de porta de origem

datagramas com origem ou destino nesta rede têm endereço 10.0.0/24 para origem/destino (como sempre)

- ❑ **motivação:** rede local usa apenas um endereço IP no que se refere ao mundo exterior:
 - intervalo de endereços não necessário pelo ISP: apenas um endereço IP para todos os dispositivos
 - pode mudar os endereços dos dispositivos na rede local sem notificar o mundo exterior
 - pode mudar de ISP sem alterar os endereços dos dispositivos na rede local
 - dispositivos dentro da rede local não precisam ser explicitamente endereçáveis ou visíveis pelo mundo exterior (uma questão de segurança).

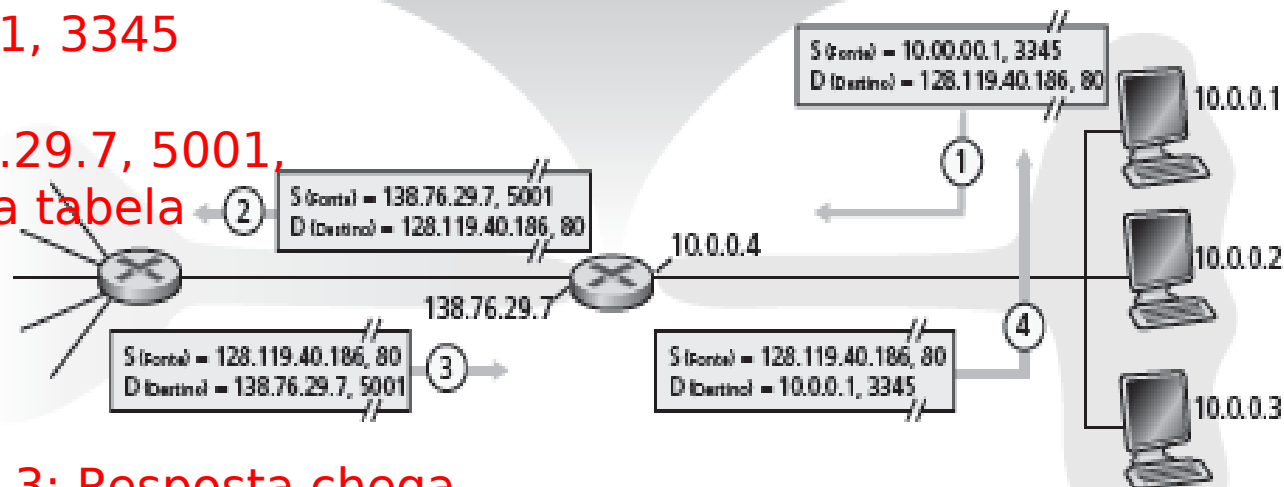
Implementação: roteador NAT deve:

- *enviando datagramas: substituir* (endereço IP de origem, # porta) de cada datagrama saindo por (endereço IP da NAT, novo # porta)
... clientes/servidores remotos responderão usando (endereço IP da NAT, novo # porta) como endereço de destino
- *lembrar (na tabela de tradução NAT)* de cada par de tradução (endereço IP de origem, # porta) para (endereço IP da NAT, novo # porta)
- *recebendo datagramas: substituir* (endereço IP da NAT, novo # porta) nos campos de destino de cada datagrama chegando por (endereço IP origem, # porta) correspondente, armazenado na tabela NAT

2: roteador NAT muda endereço de origem do datagrama de 10.0.0.1, 3345 para 138.76.29.7, 5001, atualiza tabela

Tabela de tradução NAT	
Lado da WAN	Lado da LAN
138.76.29.7, 5001	10.0.0.1, 3345
...	...

1: hospedeiro 10.0.0.1 envia datagrama para 128.119.40.186, 80



3: Resposta chega endereço destino: 138.76.29.7, 5001

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

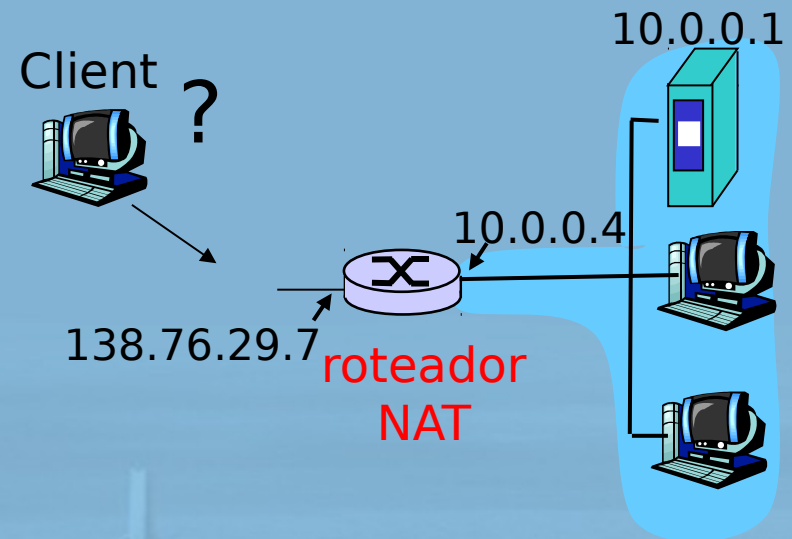
- ❑ campo de número de porta de 16 bits:
 - 60.000 conexões simultâneas com um único endereço no lado da LAN!
- ❑ NAT é controverso:
 - roteadores só devem processar até a camada 3
 - viola argumento de fim a fim
 - a possibilidade de NAT deve ser levada em conta pelos projetistas da aplicação, p. e., aplicações P2P
 - a falta de endereços deverá ser resolvida pelo IPv6

Problema da travessia da NAT

- ❑ cliente quer se conectar ao servidor com endereço 10.0.0.1

- endereço do servidor 10.0.0.1 local à LAN (cliente não pode usá-lo como endereço destino)
- apenas um endereço NAT visível externamente: 138.76.29.7

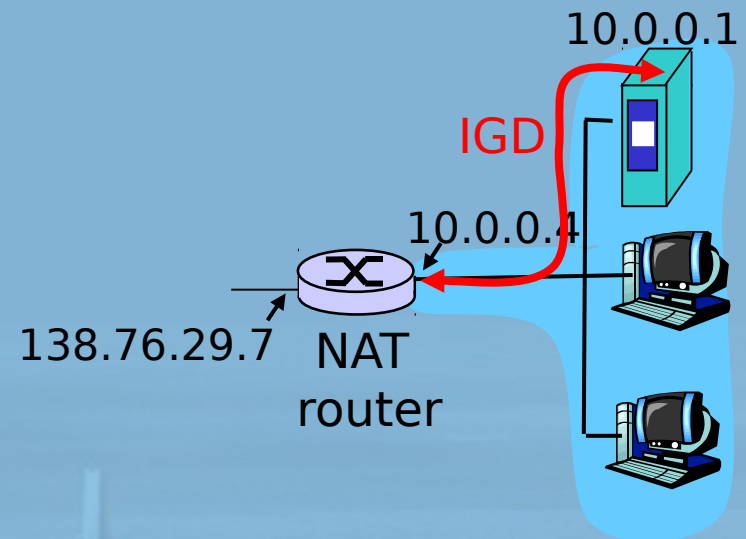
- ❑ solução 1: configure a NAT estaticamente para repassar as solicitações de conexão que chegam a determinada porta ao servidor



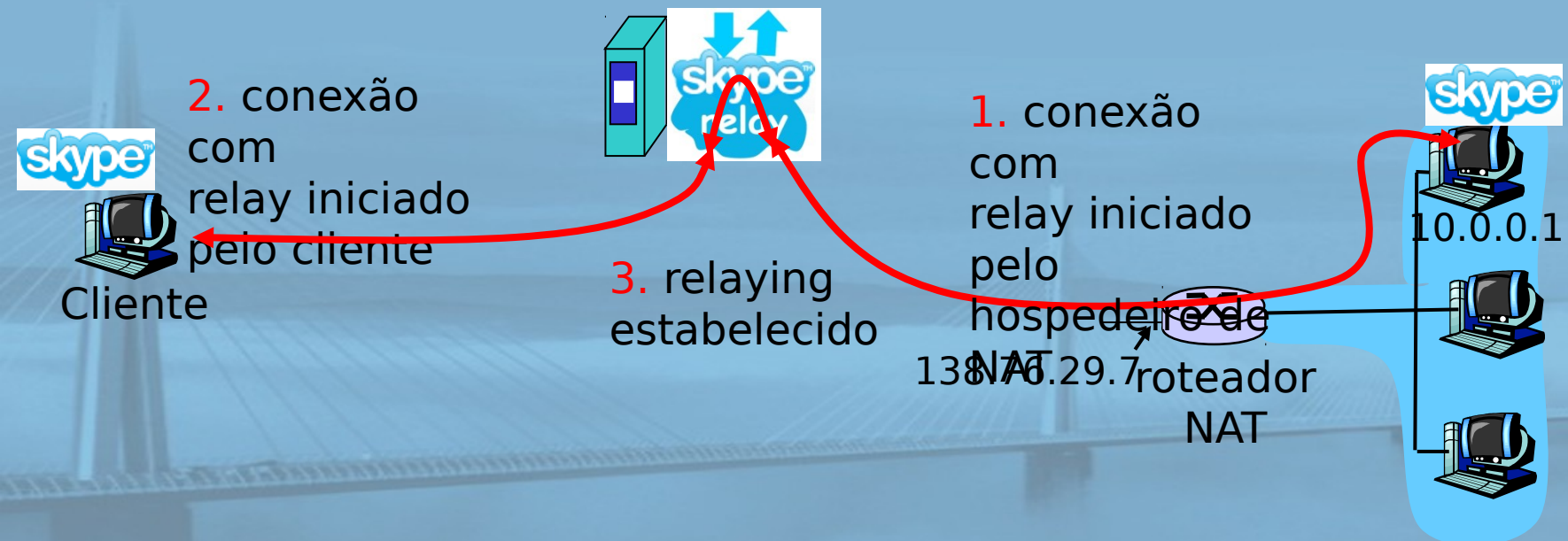
- ❑ solução 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Permite que o hospedeiro com NAT:

- ❖ descubra endereço IP público (138.76.29.7)
- ❖ inclua/remova mapeamentos de porta (com tempos de posse)

ou seja, automatizar configuração estática do mapa de porta NAT



- ❑ solução 3: repasse (usado no Skype)
 - cliente com NAT estabelece conexão com repasse
 - cliente externo se conecta ao repasse
 - repasse liga pacotes entre duas conexões



Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

ICMP: Internet Control Message Protocol

- ❑ usado por hospedeiros & roteadores para comunicar informações em nível de rede
 - relato de erro: hospedeiro, rede, porta, protocolo inalcançável
 - eco de solicitação/resposta (usado por ping)
- ❑ camada de rede “acima” do IP:
 - msgs ICMP transportadas em datagramas IP
- ❑ **mensagem ICMP:** tipo, código mais primeiros 8 bytes do datagrama IP

<u>Tipo</u>	<u>Cód.</u>	<u>Descrição</u>
0	0	resposta de eco (ping)
3	0	rede de destino inalcançável
3	1	hosp. de destino inalcançável
3	2	protocolo de destino inalcançável
3	3	porta de destino inalcançável
3	6	rede de destino desconhecida
3	7	hosp. de destino desconhecido
4	0	redução da fonte (controle de congestionamento – não usado)
8	0	solicitação de eco (ping)
9	0	anúncio de rota
10	0	descoberta do roteador
11	0	TTL expirado
12	0	cabeçalho IP inválido

Traceroute e ICMP

- ❑ origem envia série de segmentos UDP ao destino
 - primeiro tem TTL = 1
 - segundo tem TTL = 2 etc.
 - número de porta improvável
- ❑ quando n^o datagrama chegar no n^o roteador:
 - roteador descarta datagrama
 - e envia à origem uma msg ICMP (tipo 11, código 0)
 - mensagem inclui nome do roteador & endereço

- ❑ quando a mensagem ICMP chega, origem calcula RTT
- ❑ traceroute faz isso 3 vezes

Critério de término

- ❑ segmento UDP por fim chega no hospedeiro de destino
- ❑ destino retorna pacote ICMP “host inalcançável” (tipo 3, código 3)
- ❑ quando origem recebe esse ICMP, termina.

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

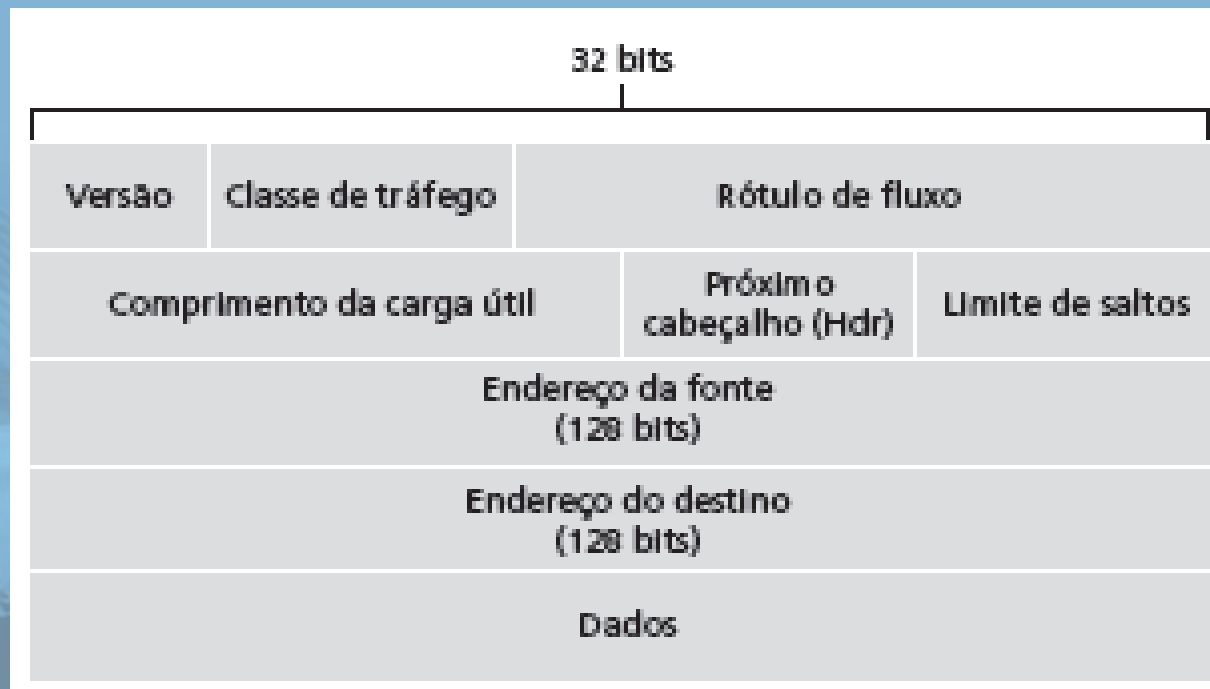
- ❑ **motivação inicial:** espaço de endereço de 32 bit logo estará completamente alocado
- ❑ **motivação adicional:**
 - formato de cabeçalho ajuda a agilizar processamento e repasse
 - mudanças no capítulo para facilitar QoS
- formato de datagrama IPv6:**
 - cabeçalho de 40 bytes de tamanho fixo
 - fragmentação não permitida

Cabeçalho IPv6

prioridade: identificar prioridade entre datagramas no fluxo

rótulo de fluxo: identificar datagramas no mesmo “fluxo.”
(conceito de “fluxo” não bem definido)

próximo cabeçalho: identificar protocolo da camada superior
para dados



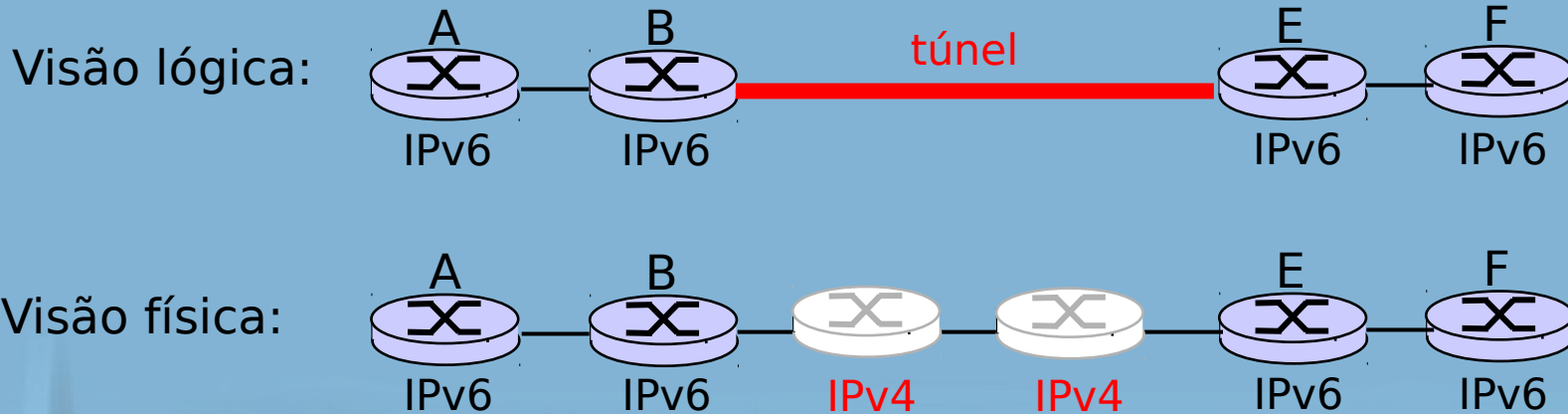
Outras mudanças do IPv4

- ❑ *soma de verificação*: removida inteiramente para reduzir tempo de processamento em cada salto
- ❑ *opções*: permitidas, mas fora do cabeçalho, indicadas pelo campo de “Próximo Cabeçalho”
- ❑ *ICMPv6*: nova versão do ICMP
 - tipos de mensagem adicionais, p. e. “Pacote Muito Grande”
 - funções de gerenciamento de grupo multicast

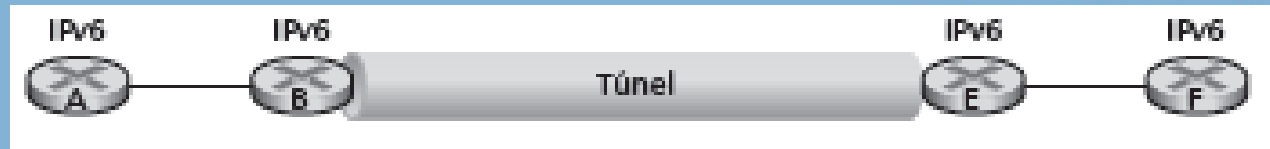
Transição de IPv4 para IPv6

- ❑ nem todos os roteadores podem ser atualizados simultaneamente
 - sem “dia de conversão”
 - como a rede operará com roteadores IPv4 e IPv6 misturados?
- ❑ *implantação de túnel*: IPv6 transportado como carga útil no datagrama IPv4 entre roteadores IPv4

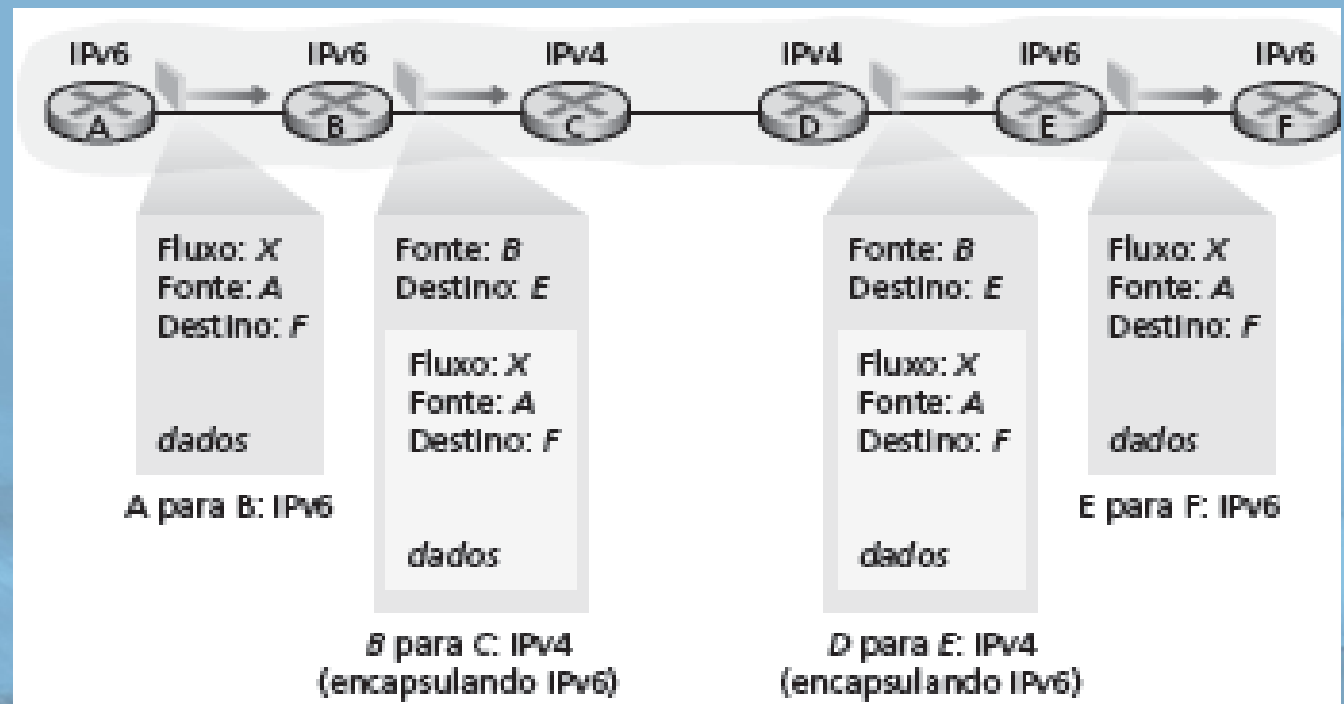
Implantação de túnel



Visão lógica:



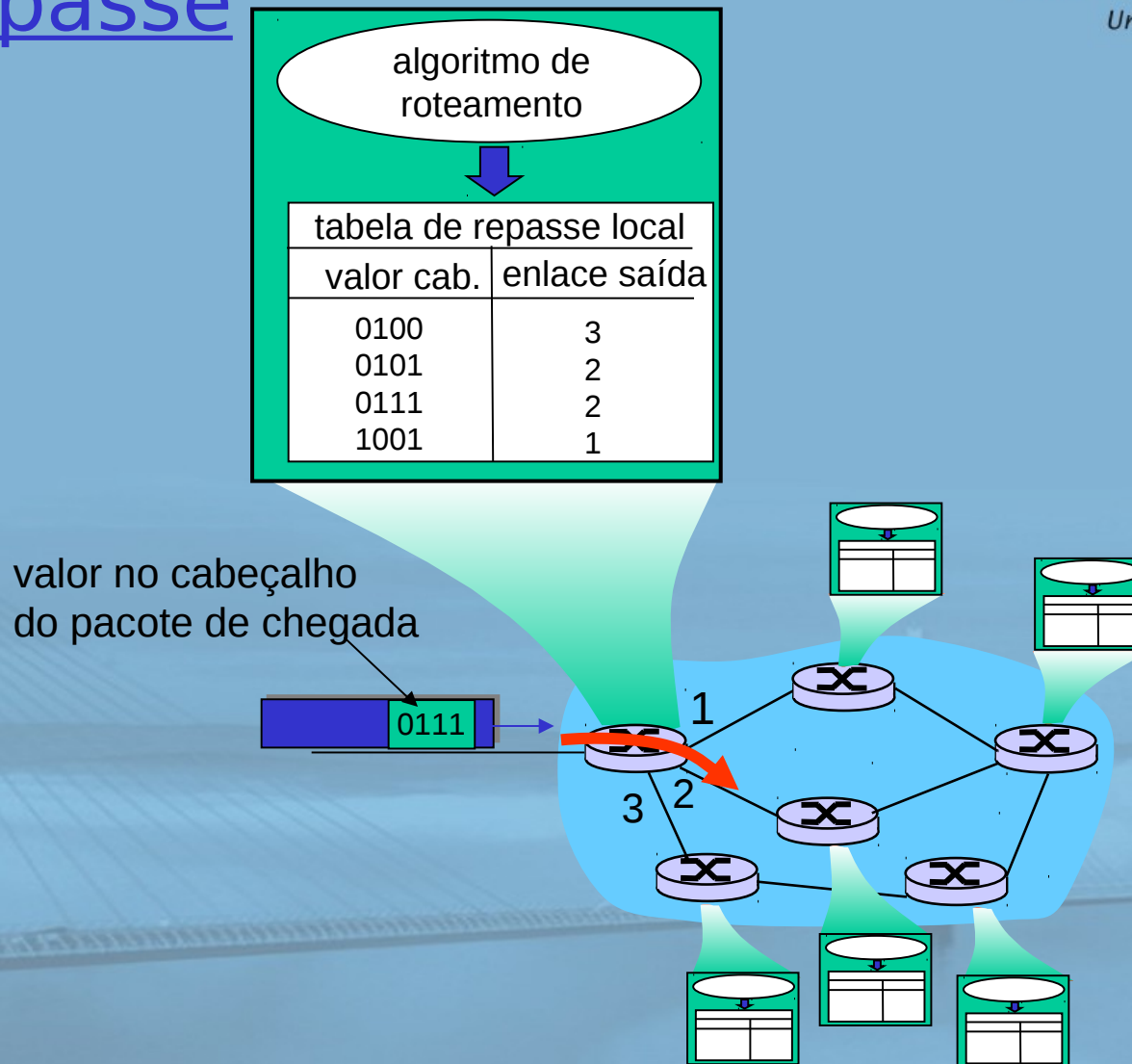
Visão física:



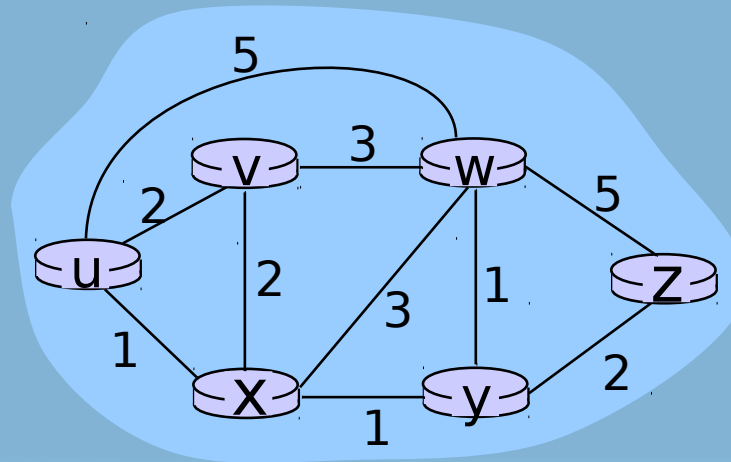
Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Interação entre roteamento e repasse



Abstração de grafo



Grafo: $G = (N, E)$

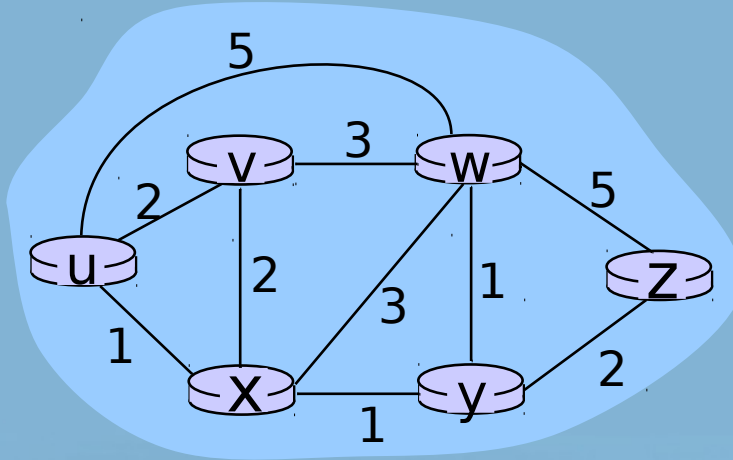
N = conjunto de roteadores = $\{ u, v, w, x, y, z \}$

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

Comentário: Abstração de grafo é útil em outros contextos de rede

Exemplo: P2P, onde N é conj. de pares e E é conj. de conexões TCP

Abstração de grafo: custos



- $c(x, x') =$ custo do enlace (x, x')
 - p. e., $c(w, z) = 5$
- custo poderia ser sempre 1, ou inversamente relacionado à largura ou inversamente relacionado ao congestionamento

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

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

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

Classificação do algoritmo de roteamento

informação global ou descentralizada?

global:

- ❑ todos os roteadores têm topologia completa, informação de custo do enlace
- ❑ algoritmos de “estado do enlace”

descentralizada:

- ❑ roteador conhece vizinhos conectados fisicamente, custos de enlace para vizinhos
- ❑ processo de computação iterativo, troca de informações com vizinhos

Estático ou dinâmico?

estático:

- ❑ rotas mudam lentamente com o tempo

dinâmico:

- ❑ rotas mudam mais rapidamente
 - atualização periódica
 - em resposta a mudanças no custo do enlace

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Algoritmo de roteamento de estado do enlace

algoritmo de Dijkstra

- ❑ nova topologia, 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
 - da **tabela de repasse** para esse nó
- ❑ iterativo: após k iterações, sabe caminho de menor custo para k destinos

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

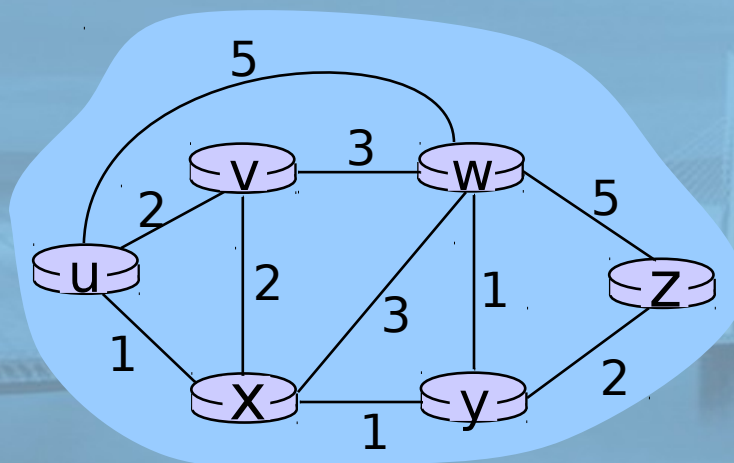
13 /* novo custo para v é custo antigo para v ou custo conhecido

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 (2)

árvore resultante do caminho mais curto a partir de u:

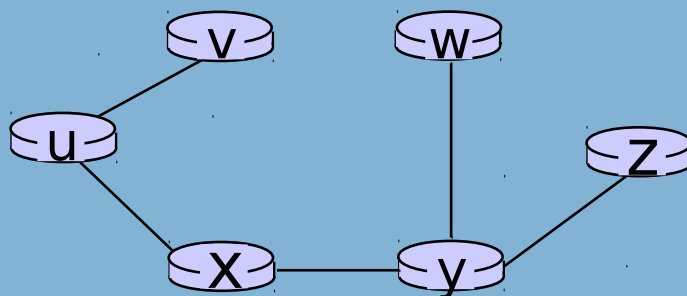


tabela de repasse resultante em u:

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

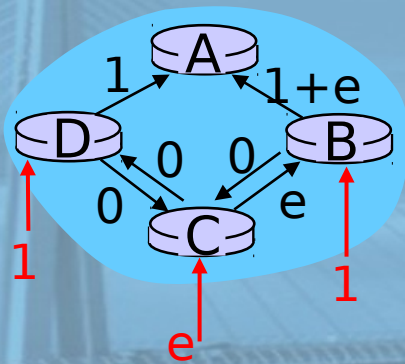
Algoritmo de Dijkstra, discussão

complexidade do algoritmo: n nós

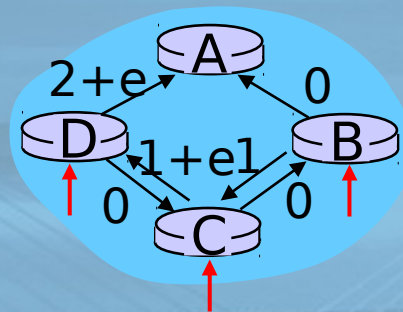
- ❑ cada iteração: precisa verificar todos os nós, w , não em N
- ❑ $n(n+1)/2$ comparações: $O(n^2)$
- ❑ implementações mais eficientes possíveis: $O(n \log n)$

oscilações possíveis:

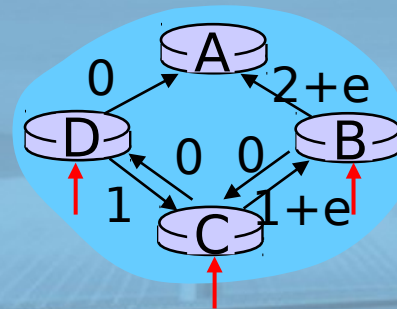
- ❑ p. e., custo do enlace = quantidade de tráfego transportado



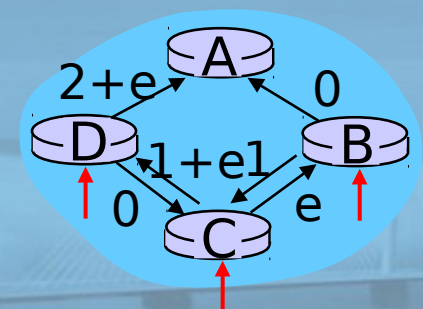
inicialmente



... recalcula
roteamento



... recalcula



... recalcula

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - **vetor de distâncias**
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Algoritmo de vetor de distância

Equação de Bellman-Ford (programação dinâmica)

defina

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

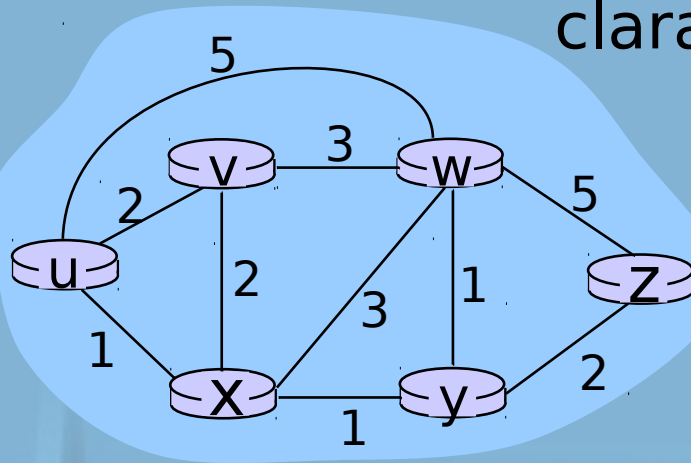
depois

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

onde min assume todos os vizinhos v de x

Exemplo de Bellman-Ford

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



equação B-F diz:

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

nó que alcança mínimo é o próximo salto
no caminho mais curto → tabela de repasse

Algoritmo de vetor de distância

- $D_x(y)$ = estimativa do menor custo de x para y
- nó x sabe custo de cada vizinho v: $c(x,v)$
- nó x mantém vetor de distância $D_x = [D_x(y): y \in N]$
- nó x também mantém vetor de distância de seus vizinhos
 - para cada vizinho v, x mantém $D_v = [D_v(y): y \in N]$

Algoritmo de vetor de distância (4)

ideia básica:

- de tempos em tempos, cada nó envia sua própria estimativa de vetor de distância aos vizinhos
- assíncrono
- quando um nó x recebe nova estimativa DV do vizinho, ele atualiza seu próprio DV usando a equação de Bellman-Ford:
$$D_x(y) \leftarrow \min_v \{c(x,v) + D_v(y)\} \quad \text{para cada nó } y \in N$$
- sob condições modestas, naturais, a estimativa $D_x(y)$ converge para o menor custo real $d_x(y)$

Algoritmo de vetor de distância (5)

iterativo, assíncrono: cada iteração local causada por:

- ❑ mudança de custo do enlace local
- ❑ mensagem de atualização do DV do vizinho

distribuído:

- ❑ cada nó notifica os vizinhos *apenas* quando seu DV muda
 - vizinhos, então, notificam seus vizinhos, se necessário

Cada nó:

espera (mudança no custo do enlace local ou msg do vizinho)

recalcula estimativas

se DV a qualquer destino tiver mudado, notifica vizinhos

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

$$= \min\{2+0, 7+1\} = 2$$

tabela nó x

	custo para		
	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

tabela nó y

	custo para		
	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

tabela nó z

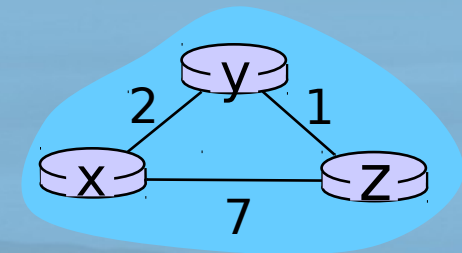
	custo para		
	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

custo para

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

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

$$= \min\{2+1, 7+0\} = 3$$



► tempo

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

$$= \min\{2+0, 7+1\} = 2$$

tabela nó x

custo para

	x	y	z
x	0	2	7
y	∞	0	∞
z	∞	∞	0

tabela nó y

custo para

	x	y	z
x	∞	0	∞
y	2	0	1
z	∞	∞	0

tabela nó z

custo para

	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

custo para

	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

custo para

	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

custo para

	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

custo para

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

custo para

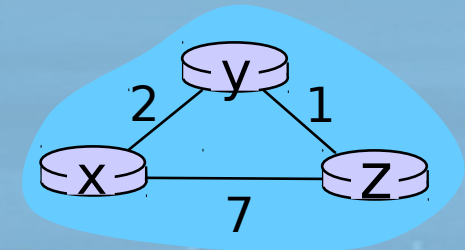
	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

custo para

	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

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

$$= \min\{2+1, 7+0\} = 3$$

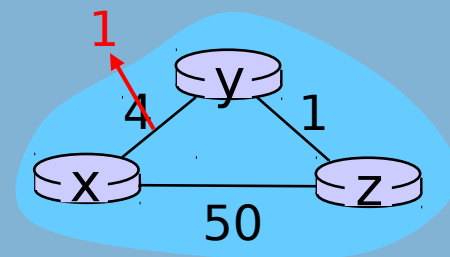


tempo

Vetor de distância: mudanças de custo do enlace

mudanças de custo do enlace:

- ❑ nó detecta mudança de custo no enlace local
- ❑ atualiza informação de roteamento, recalcula vetor de distância
- ❑ se DV mudar, notifica vizinhos



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

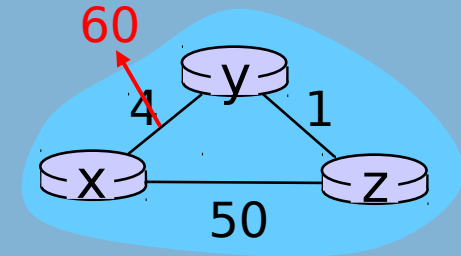
no tempo t_1 , z recebe a atualização de y e atualiza sua tabela. Calcula um novo custo mínimo para x e envia seu DV aos vizinhos.

no tempo t_2 , y recebe a atualização de z e atualiza sua tabela de distância. Menores custos de y não mudam, e daí y *não* envia qualquer mensagem a z.

“boas
notícias
correm
rápido”

mudanças de custo do enlace:

- ❑ boas notícias correm rápido
- ❑ más notícias correm lento – problema da “contagem até o infinito”!
- ❑ 44 iterações antes que o algoritmo estabilize: ver texto



reverso envenenado:

- ❑ se Z passa por Y para chegar a X:
 - Z diz a Y que sua distância (de Z) até X é infinita (de modo que Y não roteará para X passando por Z)
- ❑ isso solucionará completamente o problema da contagem até o infinito?

Comparação dos algoritmos LS e DV

complexidade da mensagem

- ❑ LS: com n nós, E enlaces, $O(nE)$ mensagens enviadas
- ❑ DV: troca apenas entre vizinhos
 - tempo de convergência varia

velocidade de convergência

- ❑ LS: algoritmo $O(n^2)$ requer $O(nE)$ mensagens
 - pode ter oscilações
- ❑ DV: tempo de convergência varia

robustez: o que acontece se roteador der defeito?

LS:

- nó pode anunciar custo do *enlace* incorreto
- cada nó calcula apenas sua *própria* tabela

DV:

- nó DV pode anunciar custo do *caminho* incorreto
- tabela de cada nó usada por outros

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Roteamento hierárquico

nosso estudo de roteamento até aqui – o ideal:

- ❑ todos os roteadores idênticos
- ❑ rede “achatada”

... *não* acontece na prática

escala: com 200
milhões de destinos:

- ❑ não pode armazenar todos os destinos nas tabelas de roteamento!
- ❑ troca de tabela de roteamento atolaria os enlaces!

**autonomia
administrativa**

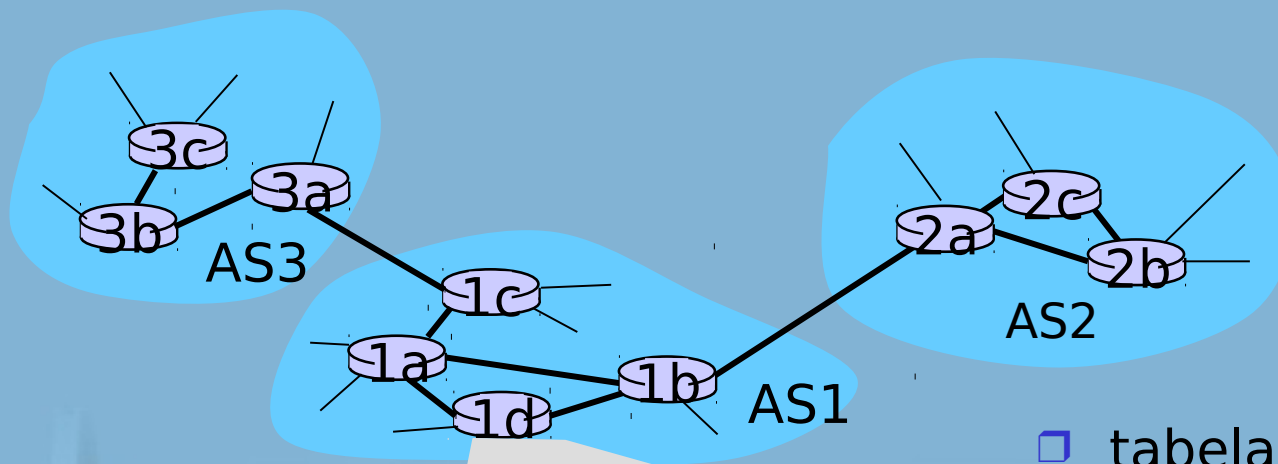
- ❑ Internet = rede de redes
- ❑ cada administrador de rede pode querer controlar o roteamento em sua própria rede

- ❑ roteadores agregados em regiões, “**sistemas autônomos**” (AS)
- ❑ roteadores no mesmo AS rodam o mesmo protocolo de roteamento
 - protocolo de **roteamento “intra-AS”**
 - roteadores em ASes diferentes podem executar protocolo de roteamento intra-AS diferente

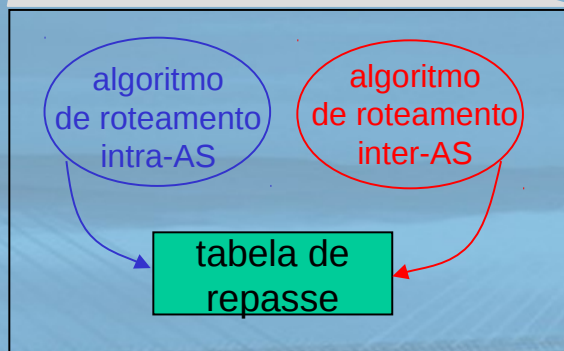
roteador de borda

- ❑ Enlace direto com roteador em outro AS

ASes interconectados

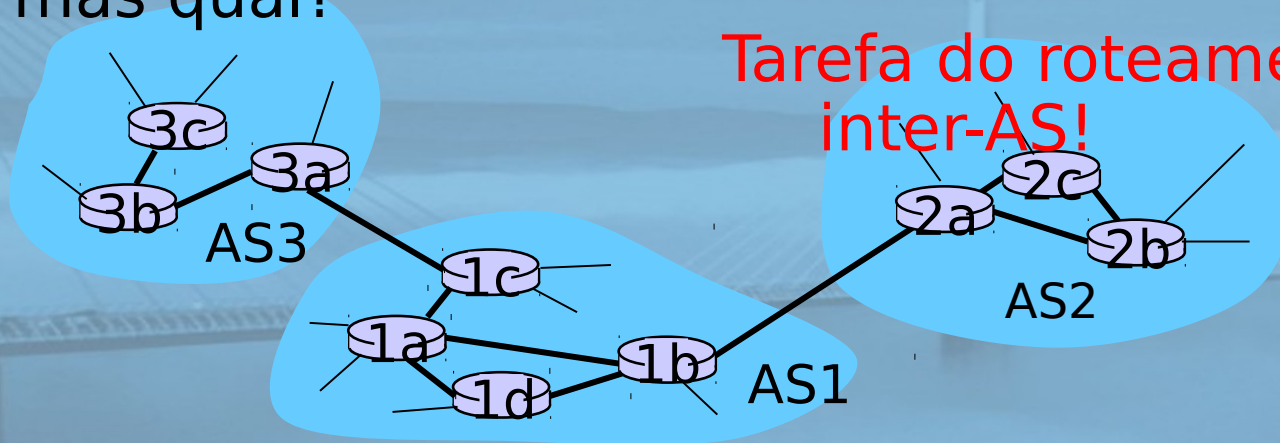


- tabela de repasse configurada por algoritmo de roteamento intra e inter-AS
 - intra-AS define entradas para destinos internos
 - inter-AS & intra-AS definem entradas para destinos externos



Tarefas inter-AS

- suponha que roteador no AS1 recebe datagrama destinado para fora do AS1:
 - roteador deve encaminhar pacote ao roteador de borda, mas qual?



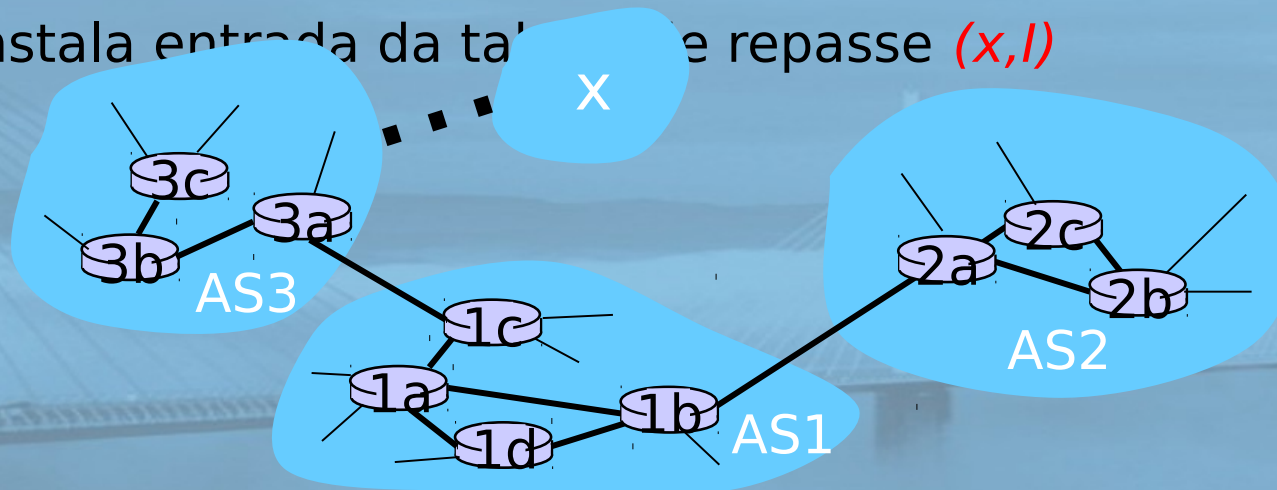
AS1 deve:

1. descobrir quais destinos são alcançáveis por AS2 e quais por AS3
2. propagar essa informação de acessibilidade a todos os roteadores no AS1

Tarefa do roteamento inter-AS!

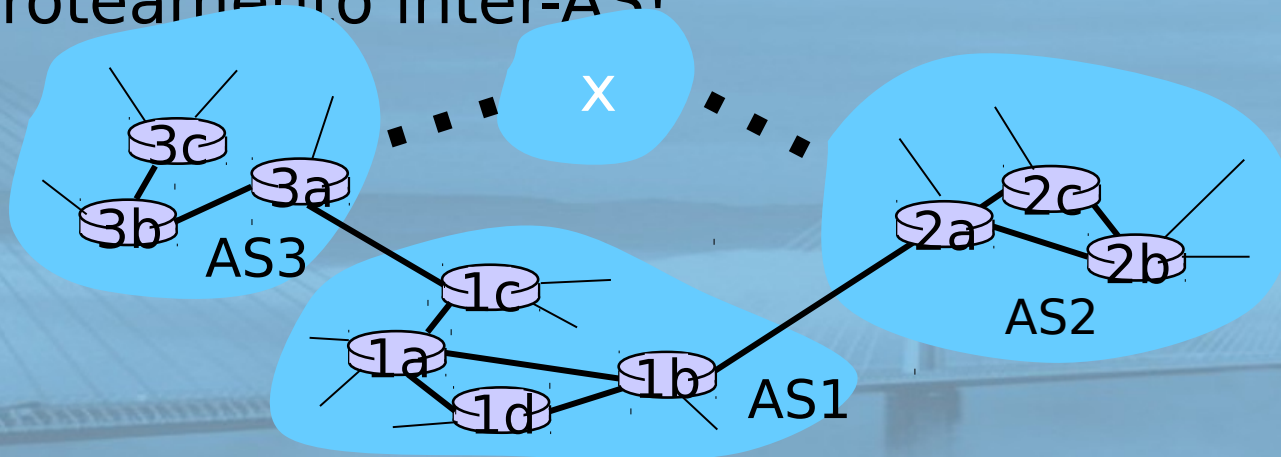
Exemplo: definindo tabela de repasse no roteador 1d

- suponha que AS1 descubra (pelo protocolo inter-AS) que a sub-rede **x** é alcançável via AS3 (gateway 1c), mas não via AS2.
- protocolo inter-AS propaga informação de acessibilidade a todos os roteadores internos.
- roteador 1d determina pelo roteamento intra-AS informação de que sua interface **f** está no caminho de menor custo para 1c.
 - instala entrada da tabela de repasse **(x,f)**

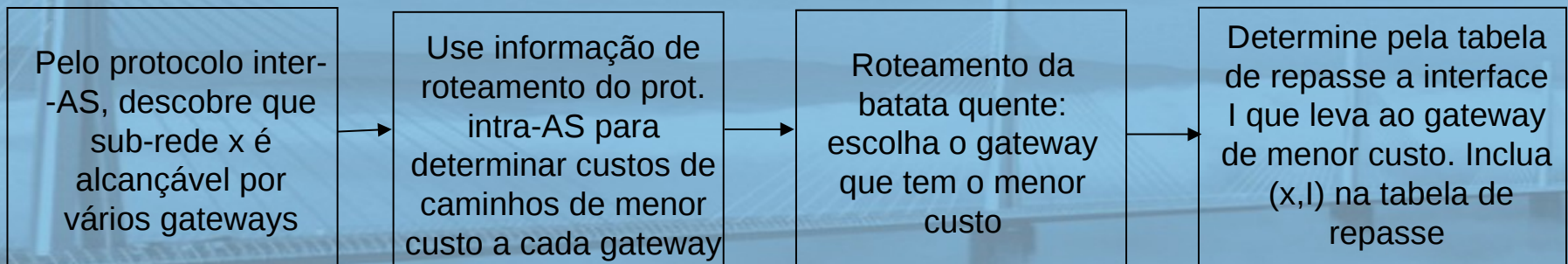


Exemplo: escolhendo entre múltiplos ASes

- agora suponha que o AS1 descubra pelo protocolo inter-AS que a sub-rede **x** pode ser alcançada por AS3 e por AS2.
- para configurar a tabela de repasse, roteador 1d deve determinar para que gateway ele deve repassar os pacotes para o destino **x**.
 - isso também é tarefa do protocolo de roteamento inter-AS!



- ❑ agora suponha que AS1 descubra pelo protocolo inter-AS que sub-rede **x** pode ser alcançada por AS3 e por AS2.
- ❑ para configurar a tabela de repasse, o roteador 1d deve determinar para qual gateway deve repassar pacotes para destino **x**.
 - isso também é tarefa do protocolo de roteamento inter-AS!
- ❑ **roteamento da batata quente:** envia pacote para o mais próximo dos dois roteadores.



Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Roteamento intra-AS

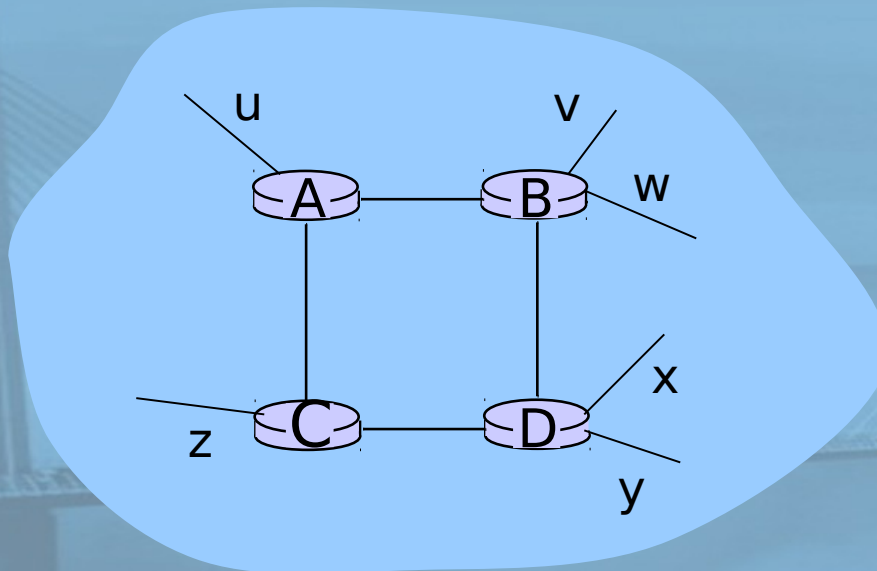
- ❑ também conhecido como **Interior Gateway Protocols (IGP)**
- ❑ protocolos de roteamento intra-AS mais comuns:
 - RIP: Routing Information Protocol
 - OSPF: Open Shortest Path First
 - IGRP: Interior Gateway roteamento Protocol (proprietário da Cisco)

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

RIP (Routing Information Protocol)

- ❑ algoritmo de vetor de distância
- ❑ incluído na distribuição BSD-UNIX em 1982
- ❑ métrica de distância: # de saltos (máx. = 15 saltos)



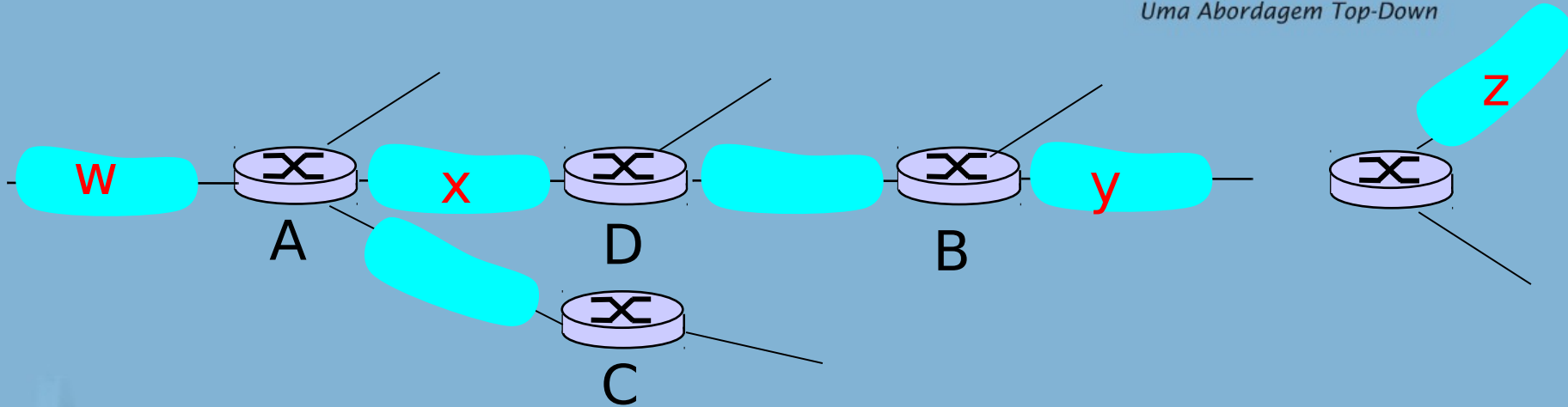
Do roteador A às sub-redes:

<u>destino</u>	<u>saltos</u>
u	1
v	2
w	2
x	3
y	3
z	2

Anúncios RIP

- ❑ vetores de distância: trocados entre vizinhos a cada 30 s por meio de mensagem de resposta (também conhecida como **anúncio**)
- ❑ cada anúncio: lista de até 25 sub-redes de destino dentro do AS

RIP: Exemplo

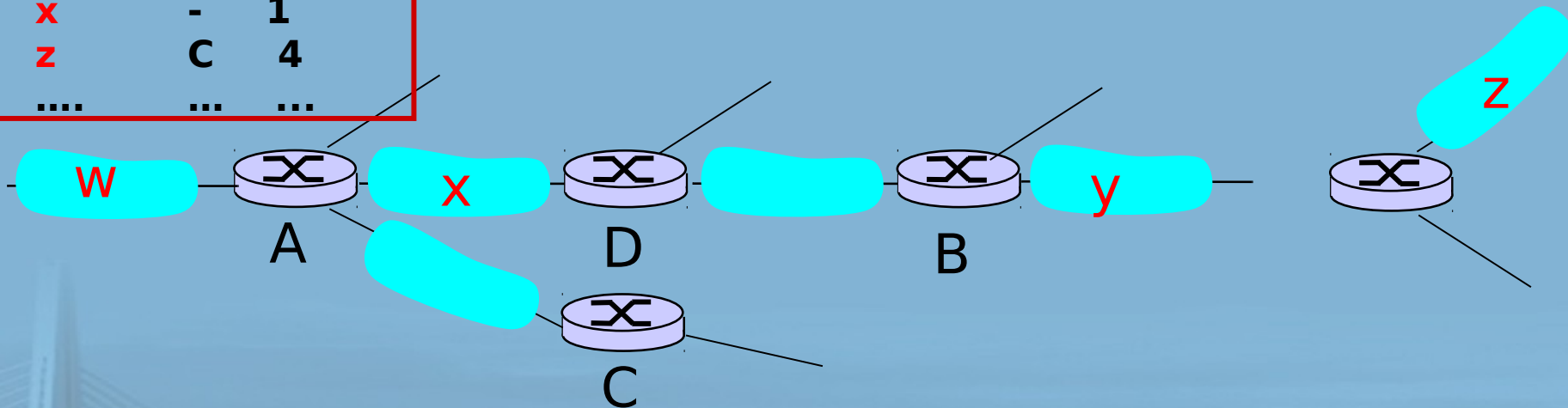


Rede de destino	Roteador seguinte	Núm. saltos até
dest. w	A	2
y	B	2
z	B	7
x	--	1
....

tabela de roteamento/repassse em D

Destino	Próx. saltos
w	- 1
x	- 1
z	C 4
....

anúncio de A para D



Rede de destino saltos até dest.	Roteador seguinte	Núm.
w	A	2
y	B	2
z	B A	7 5
x	--	1
....

tabela de roteamento/repassse em D

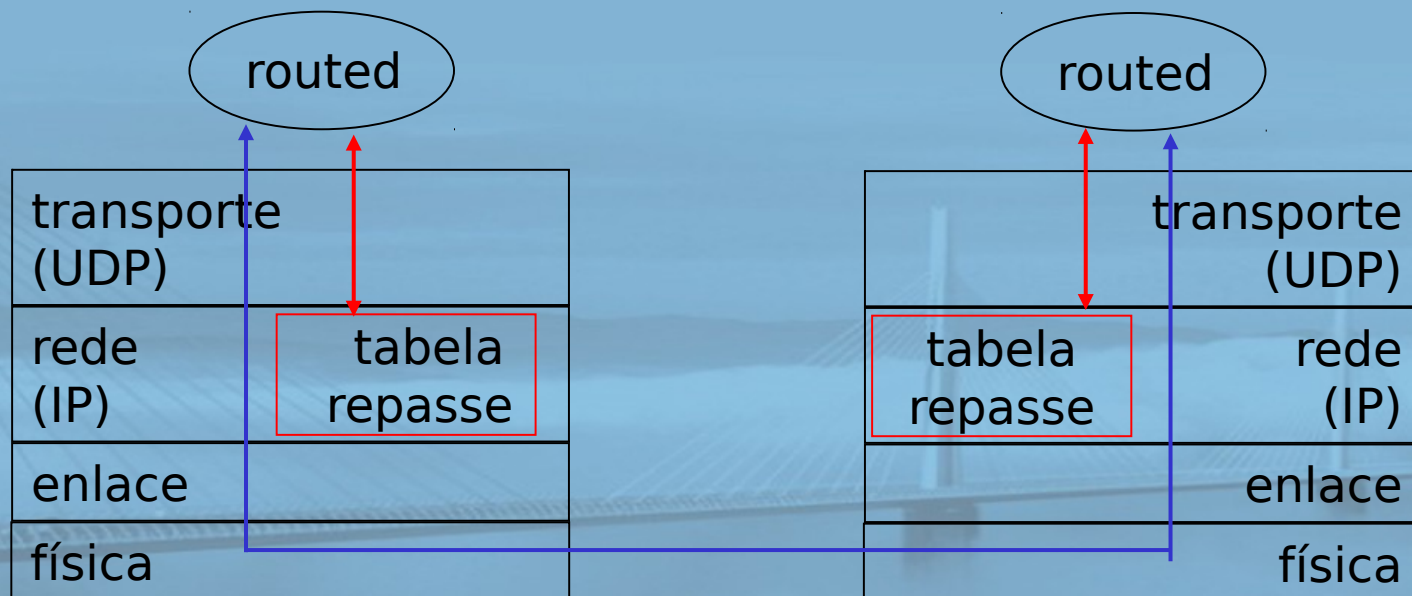
RIP: falha e recuperação do enlace

se nenhum anúncio for ouvido após 180 s -->
vizinho/enlace declarado morto

- rotas via vizinho invalidadas
- novos anúncios enviados aos vizinhos
- vizinhos por sua vez enviam novos anúncios (se não houver tabelas alteradas)
- informação de falha do enlace rapidamente (?) se propaga para rede inteira
- *reversão envenenada* usada para impedir loops de pingue-pongue (distância infinita = 16 saltos)

Processamento de tabela RIP

- ❑ tabelas de roteamento RIP controladas por processo em **nível de aplicação** chamado routed (daemon)
- ❑ anúncios enviados em pacotes UDP, repetidos periodicamente



Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

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
- ❑ anúncios disseminados ao AS **inteiro** (com inundação)
 - transportados nas mensagens OSPF diretamente por IP (em vez de TCP ou UDP)

Recursos “avancados” do OSPF (não no RIP)

- ❑ **segurança**: todas as mensagens OSPF autenticadas (para impedir intrusão maliciosa)
- ❑ **múltiplos caminhos** de mesmo custo permitidos (apenas um caminho no RIP)
- ❑ para cada enlace, múltiplas métricas de custo para diferentes **TOS** (p. e., custo de enlace de satélite definido “baixo” para melhor esforço; alto para tempo real)
- ❑ suporte integrado para uni e **multicast**:
 - Multicast OSPF (MOSPF) usa mesma base de dados de topologia que o OSPF
- ❑ OSPF **hierárquico** em grandes domínios

- ❑ **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.

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Roteamento inter-AS da Internet: BGP

- ❑ **BGP (Border Gateway Protocol):** o padrão de fato
- ❑ BGP oferece a cada AS um meio de:
 1. obter informação de acessibilidade da sub-rede a partir de ASs vizinhos.
 2. propagar informação de acessibilidade a todos os roteadores internos ao AS.
 3. determinar rotas “boas” para sub-redes com base na informação e política de acessibilidade.
- ❑ permite que a sub-rede anuncie sua existência ao resto da Internet: *“Estou aqui”*

Fundamentos do BGP

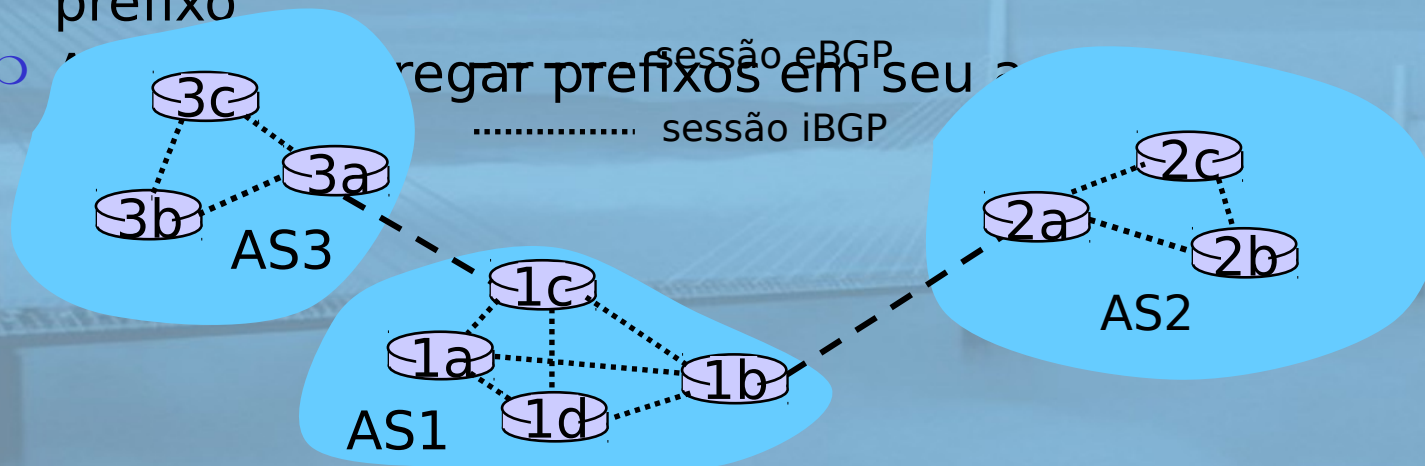
- pares de roteadores (pares BGP) trocam informações de roteamento nas conexões TCP semipermanentes: sessões **BGP**

- sessões BGP não precisam corresponder a enlaces físicos

- quando AS2 anuncia um prefixo para AS1:

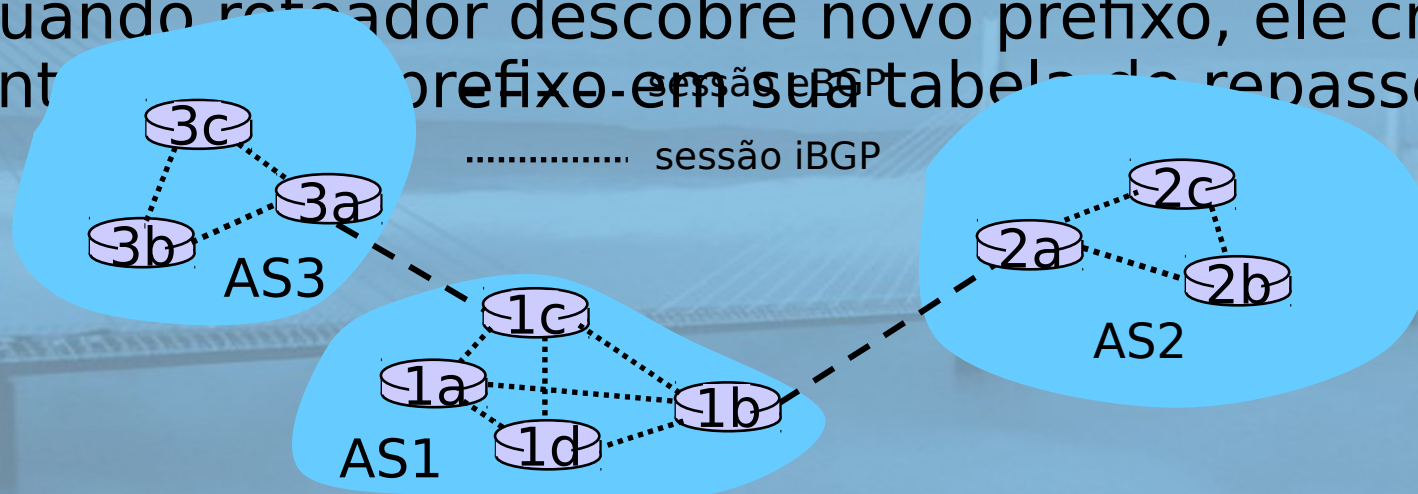
- AS2 **promete** que repassará datagramas para esse prefixo

- AS1 promete repassar prefixos em seu nome



Distribuindo informações de atingibilidade

- usando sessão eBGP entre 3a e 1c, AS3 envia informação de atingibilidade do prefixo a AS1.
 - 1c pode então usar iBGP para distribuir nova informação de prefixo a todos os roteadores em AS1
 - 1b pode então reanunciar nova informação de atingibilidade para AS2 por sessão 3BGP 1b-para-2a
- quando roteador descobre novo prefixo, ele cria entrada para esse prefixo em sua tabela de roteamento.



Atributos de caminho & rotas BGP

- ❑ prefixo anunciado inclui atributos BGP.
 - prefixo + atributos = “rota”
- ❑ dois atributos importantes:
 - **AS-PATH**: contém ASs através dos quais o anúncio do prefixo passou: p. e., AS 67, AS 17
 - **NEXT-HOP**: indica roteador específico do AS interno para AS do próximo salto (podem ser múltiplos enlaces para AS atual até AS do próximo salto)
- ❑ quando o roteador de borda recebe anúncio de rota, usa **política de importação** para aceitar/declinar.

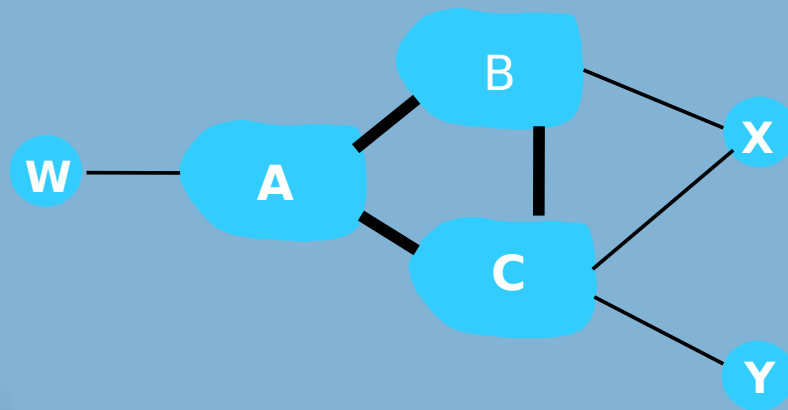
Seleção de rota BGP

- ❑ roteador pode aprender sobre mais de 1 rota para algum prefixo. Roteador deve selecionar rota
- ❑ regras de eliminação:
 1. atributo do valor de preferência local: decisão política
 2. AS-PATH mais curto
 3. roteador NEXT-HOP mais próximo: roteamento batata quente
 4. critérios adicionais

Mensagens BGP

- ❑ Mensagens BGP trocadas usando TCP.
- ❑ Mensagens BGP:
 - **OPEN**: abre conexão TCP com par e autentica remetente
 - **UPDATE**: anuncia novo caminho (ou retira antigo)
 - **KEEPALIVE** mantém conexão viva na ausência de UPDATES; também envia ACK para solicitação OPEN
 - **NOTIFICATION**: informa erros na msg anterior; também usada para fechar conexão

Política de roteamento BGP



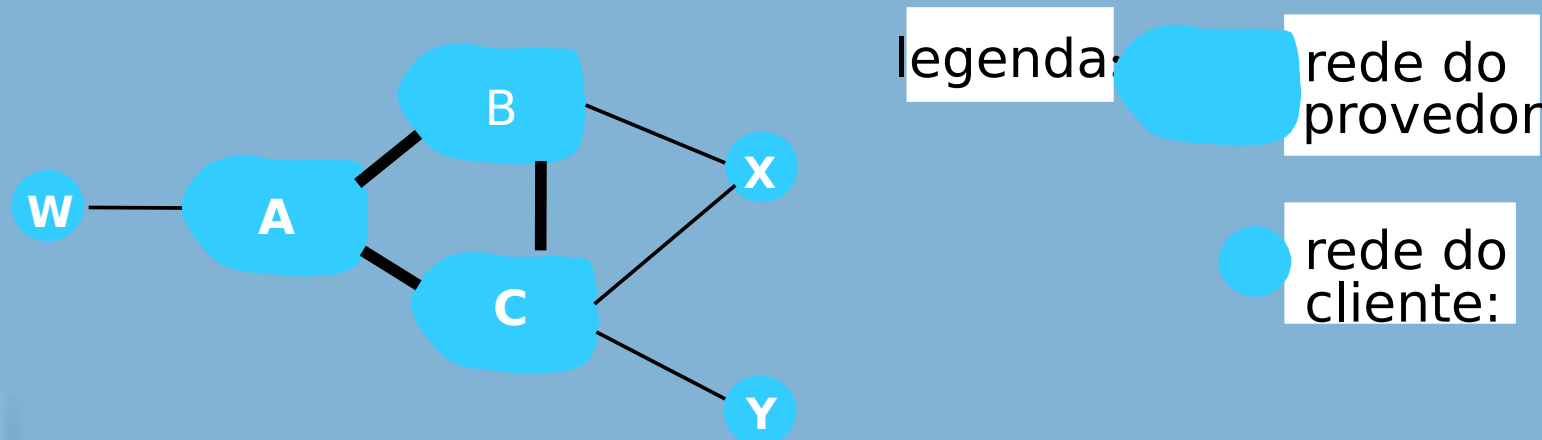
legenda:

- rede do provedor
- rede do cliente

- ❑ A, B, C são **redes do provedor**
- ❑ X, W, Y são clientes (de redes do provedor)
- ❑ X é **dual-homed**: conectada a duas redes
 - X não quer rotear a partir de B por meio de X para C
 - .. logo, X não anunciará a B uma rota para C

Política de roteamento

BGP (2)



- A anuncia caminho AW para B
- B anuncia caminho BAW para X
- B deve anunciar caminho BAW para C?
 - de forma alguma! B não recebe “retorno” para roteamento CBAW; nem W nem C são clientes de B
 - B quer forçar C a rotear para W por meio de A
 - B quer rotear *apenas* para/de seus clientes!

Por que roteamento intra e inter-AS diferente?

política:

- ❑ inter-AS: admin deseja controle sobre como seu tráfego é roteado, quem roteia através de sua rede
- ❑ intra-AS: único admin, de modo que nenhuma decisão política é necessária

escala:

- ❑ roteamento hierárquico salva tamanho de tabela, tráfego de atualização reduzido

desempenho:

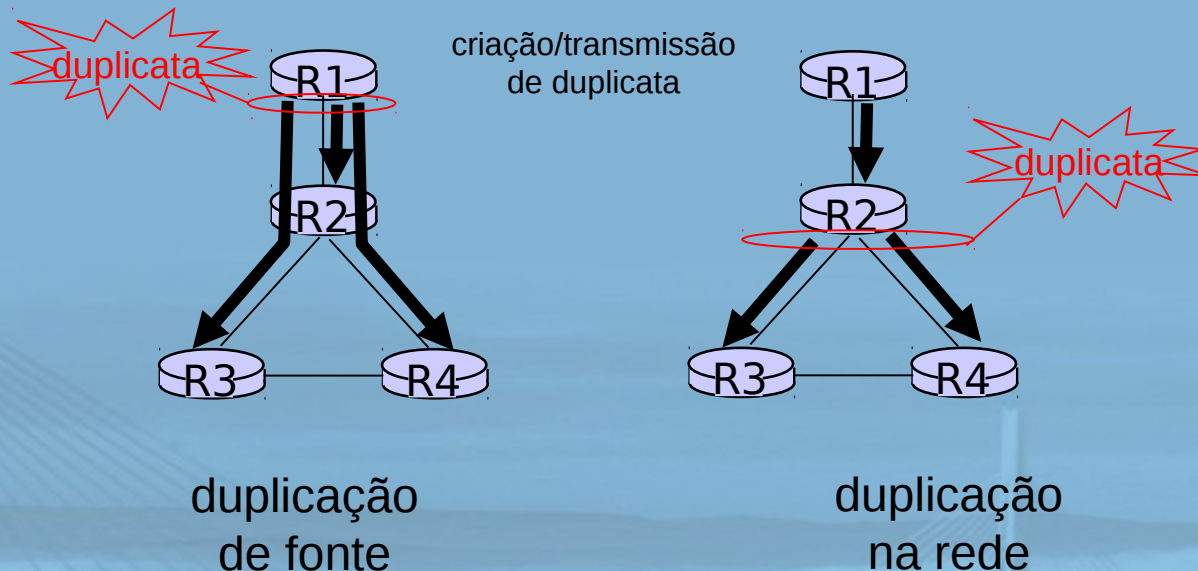
- ❑ intra-AS: pode focalizar no desempenho
- ❑ inter-AS: política pode dominar sobre desempenho

Capítulo 4: Camada de rede

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast

Roteamento broadcast

- entrega pacotes da fonte para todos os outros nós
- duplicação de fonte é ineficaz:



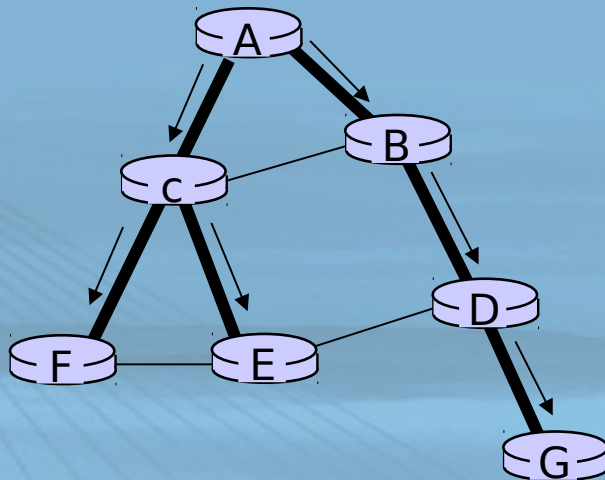
- duplicação de fonte: como a fonte determina endereços de destinatário?

Duplicação dentro da rede

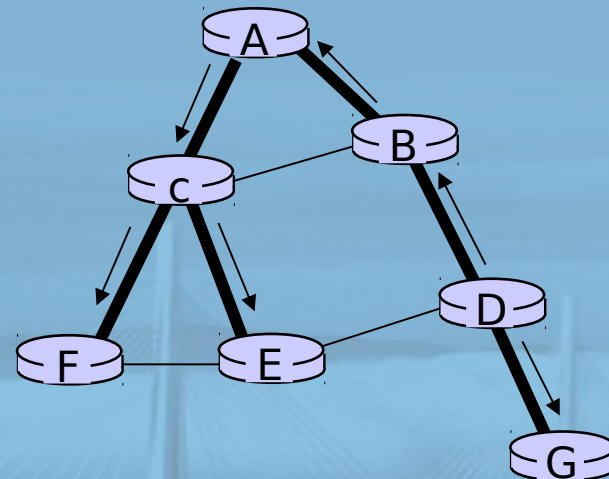
- ❑ inundação: quando o nó recebe pacote de broadcast, envia cópia para todos os vizinhos
 - problemas: ciclos & tempestade de broadcast
- ❑ inundação controlada: nó só transmite pacote se não tiver transmitido algum pacote antes
 - nó registra ids de pacote já transmitidos por broadcast
 - ou repasse pelo caminho inverso (RPF): só repassa pacote se chegasse no caminho mais curto entre nó e fonte
- ❑ spanning tree
 - nenhum pacote redundante recebido por qualquer nó

Spanning Tree

- primeiro construa uma spanning tree
- nós repassam cópias apenas ao longo da spanning tree



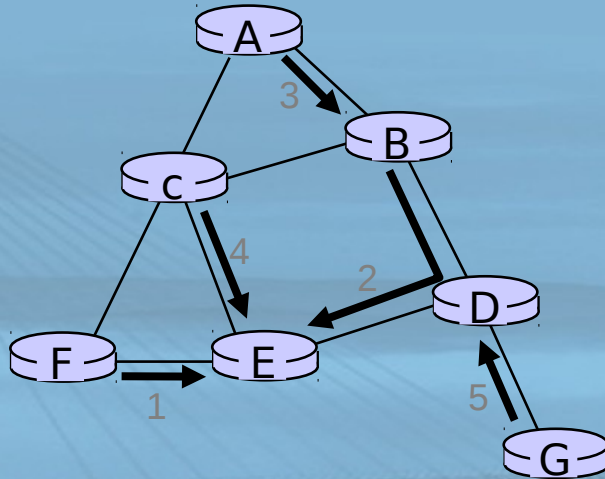
(a) broadcast iniciado em A



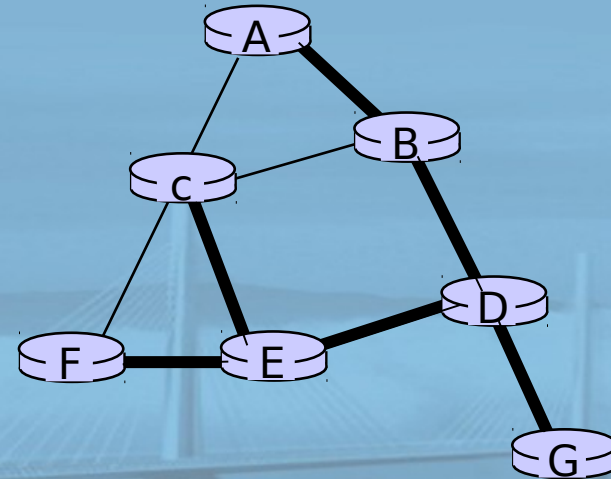
(b) broadcast iniciado em D

Spanning Tree: criação

- nó central
- cada nó envia mensagem de unicast conjunto para nó central
 - mensagem encaminhada até que chegue a um nó já pertencente à spanning tree



(a) construção passo a passo da spanning tree



(b) spanning tree construída

Roteamento multicast: declaração do problema

- **objetivo:** achar uma árvore (ou árvores) conectando roteadores que têm membros do grupo mcast local
 - **árvore:** nem todos os caminhos entre roteadores são usados
 - **baseado em fonte:** árvore diferente de cada emissor aos receptores
 - **árvore compartilhada:** mesma árvore usada por todos os membros do grupo



árvore compartilhada



árvores baseadas na fonte

Técnicas para criação de árvores mcast

técnicas:

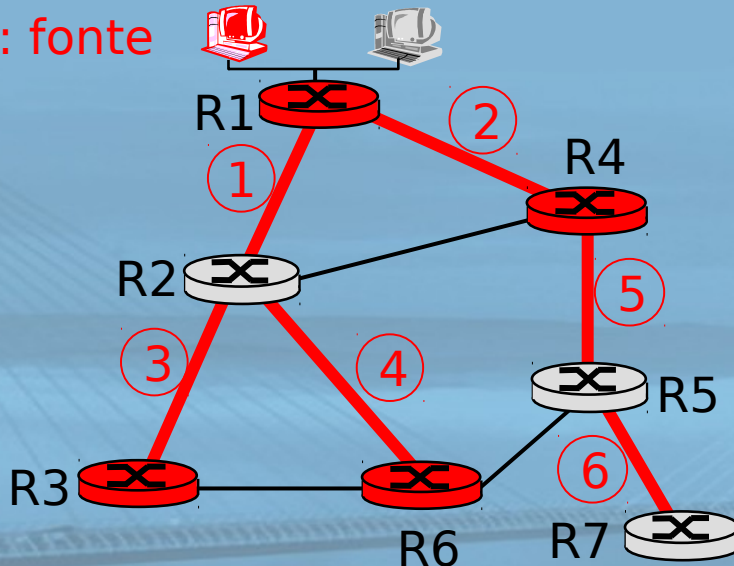
- ❑ **árvore baseada na fonte:** uma árvore por fonte
 - árvores de caminho mais curto
 - repasse pelo caminho inverso
- ❑ **árvore compartilhada pelo grupo:** grupo usa uma árvore
 - spanning mínimo (Steiner)
 - árvores baseadas no centro

...primeiro vemos as técnicas básicas, depois protocolos específicos que adotam essas técnicas

Árvore de caminho mais curto

- árvore de repasse multicast: árvore de rotas do caminho mais curto da fonte a todos os receptores
 - algoritmo de Dijkstra

S: fonte



LEGENDA

- roteador com membro de grupo conectado
- roteador sem membro de grupo conectado
- enlace usado para repasse, i indica enlace de ordem acrescentado pelo algoritmo

Repasse de caminho mais curto

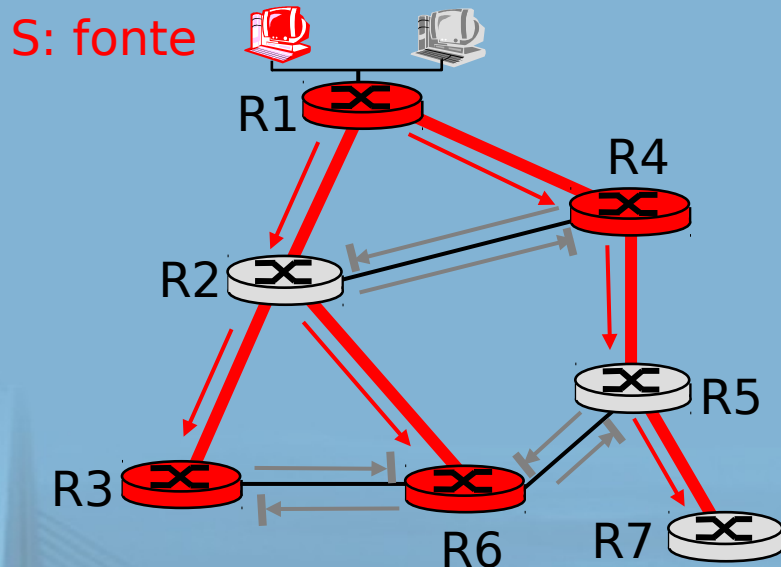
- conta com conhecimento do roteador do caminho de unicast mais curto dele para o remetente
- cada roteador tem comportamento de repasse simples:

se (datagrama mcast recebido no enlace de chegada no caminho mais curto de volta ao centro)

então inunda datagrama em todos os enlaces saindo

senão ignora datagrama

Repasse de caminho inverso: exemplo



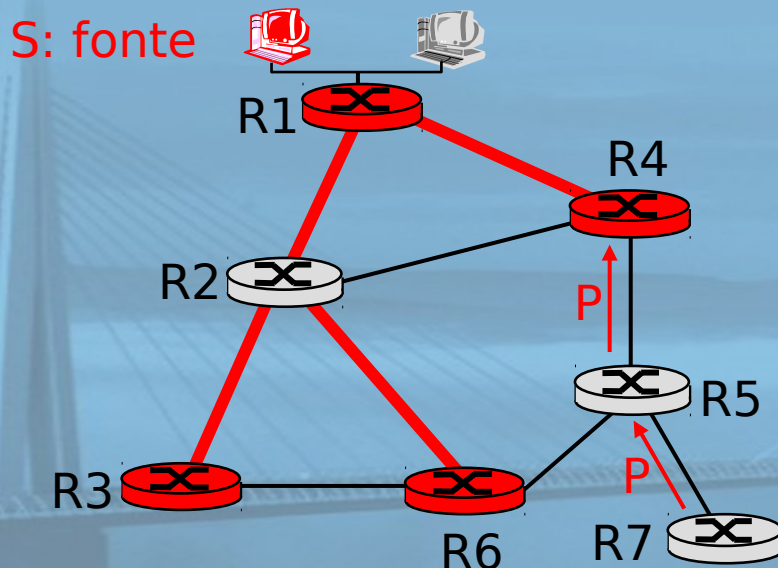
LEGENDA

- roteador com membro de grupo conectado
- roteador sem membro de grupo conectado
- datagrama será repassado
- datagrama não será repassado

- resultado é uma SPT *inversa* específica da fonte
 - pode ser uma escolha ruim com enlaces assimétricos

Reverse Path repasse: poda

- árvore de repasse contém subárvores sem membros de grupo de multicast
 - não precisa repassar datagramas adiante dele na subárvore
 - msgs de “poda” enviadas antes dele pelo roteador sem membros de grupo adiante dele



LEGENDA

- roteador com membro de grupo conectado
- roteador sem membro de grupo conectado
- mensagem de poda
- enlaces com repasse multicast

Árvore compartilhada: árvore de Steiner

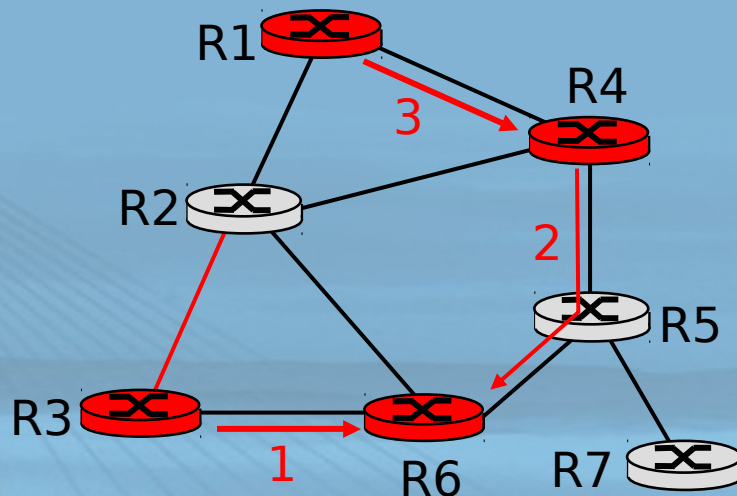
- ❑ **árvore de Steiner:** árvore de custo mínimo conectando todos os roteadores com membros de grupo conectados
- ❑ Problema sendo NP-completo
- ❑ existe excelente heurística
- ❑ não usada na prática:
 - complexidade computacional
 - necessário informações sobre a rede inteira
 - monolítica: executada novamente sempre que um roteador precisa se juntar/sair

Árvores baseadas no centro

- ❑ uma árvore de distribuição compartilhada por todos
- ❑ um roteador identificado como “centro” da árvore
- ❑ para se juntar:
 - roteador de borda envia *msg de ingresso* unicast endereçada ao roteador do centro
 - *msg de ingresso* “processada” por roteadores intermediários e repassada para o centro
 - *msg de ingresso* ou alcança rama da árvore existente para este centro ou chega no centro
 - caminho tomado pela *msg de ingresso* torna-se novo ramo da árvore para este roteador

Árvores baseadas no centro: exemplo

considere R6 escolhido como centro:



LEGENDA

- roteador com membro de grupo conectado
- roteador sem membro de grupo conectado
- ordem do caminho em que mensagens de ingresso são geradas

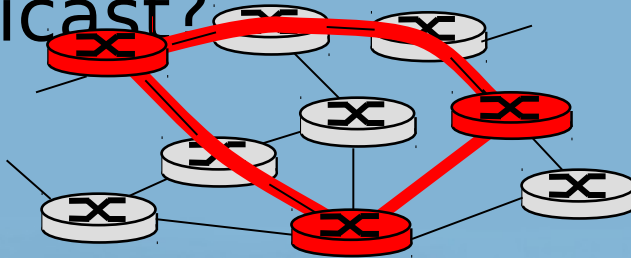
Roteamento multicasting da Internet: DVMRP

- ❑ **DVMRP:** Distance Vector Multicast Routing Protocol, RFC1075
- ❑ **inundação e poda:** repasse de caminho inverso, árvore baseada na fonte
 - árvore RPF baseada nas próprias tabelas de roteamento do DVMRP construídas pela comunicação de roteadores DVMRP
 - sem suposições sobre unicast subjacente
 - datagrama inicial para grupo multicast inundado para toda parte por meio de RPF
 - roteadores não querendo agrupar: enviam mensagens de poda para roteadores antes dele

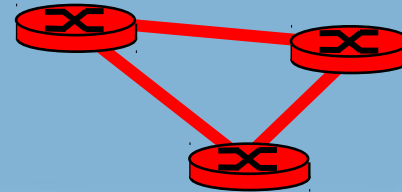
- ❑ estado soft: roteador DVMRP periodicamente (1 min.) “esquece” que os ramos são podados:
 - dados multicast novamente fluem pelo ramo não podado
 - roteador adiante: poda novamente ou então continua a receber dados
- ❑ roteadores podem rapidamente ser enxertados à árvore
 - seguindo ingresso IGMP na folha
- ❑ alguns detalhes
 - normalmente implementado em roteadores comerciais
 - roteamento Mbone é feito usando DVMRP

Tunelamento

P: Como conectar “ilhas” de roteadores multicast em um “mar” de roteadores unicast?



topologia física



topologia lógica

- ❑ datagrama multicast encapsulado dentro do datagrama “normal” (não endereçado por multicast)
- ❑ datagrama IP normal enviado por “túnel” via unicast IP regular ao roteador multicast receptor
- ❑ roteador multicast receptor encapsula para receber datagrama multicast

PIM: Protocol Independent Multicast

- ❑ não depende de qualquer algoritmo de roteamento unicast específico (funciona com todos)
- ❑ dois cenários de distribuição multicast diferentes:
 - denso:
 - ❑ membros do grupo densamente compactados, muito próximos
 - ❑ largura de banda mais farta
 - esparso:
 - ❑ # redes com membros do grupo pequeno em relação ao # total de redes
 - ❑ membros do grupo “bastante dispersos”
 - ❑ largura de banda não farta

Consequências da dicotomia esparso-denso:

denso

- ❑ inclusão no grupo pelos roteadores *assumida* até que roteadores se juntem explicitamente
- ❑ construção sobre árvore multicast *controlada por dados* (p. e., RPF)
- ❑ largura de banda e processamento de roteador não no grupo *desperdicadores*

esparso:

- ❑ membros não incluídos até que roteadores se juntem explicitamente
- ❑ construção de árvore multicast *controlada pelo destinatário* (p. e., baseada no centro)
- ❑ largura de banda e processamento de roteador não no grupo *conservadores*

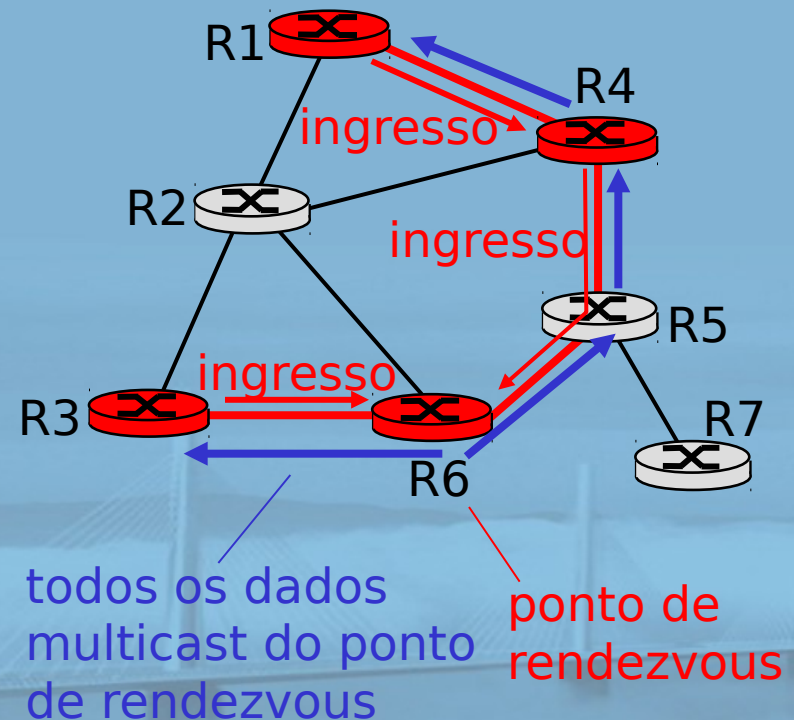
PIM - modo denso

RPF inundar e podar, semelhante ao DVMRP mas

- ❑ protocolo unicast subjacente oferece informação de RPF para datagrama que chega
- ❑ fluxo adiante menos complicado (menos eficiente) que DVMRP reduz dependência do algoritmo de roteamento subjacente
- ❑ tem mecanismo de protocolo para o roteador detectar que é um roteador de nó folha

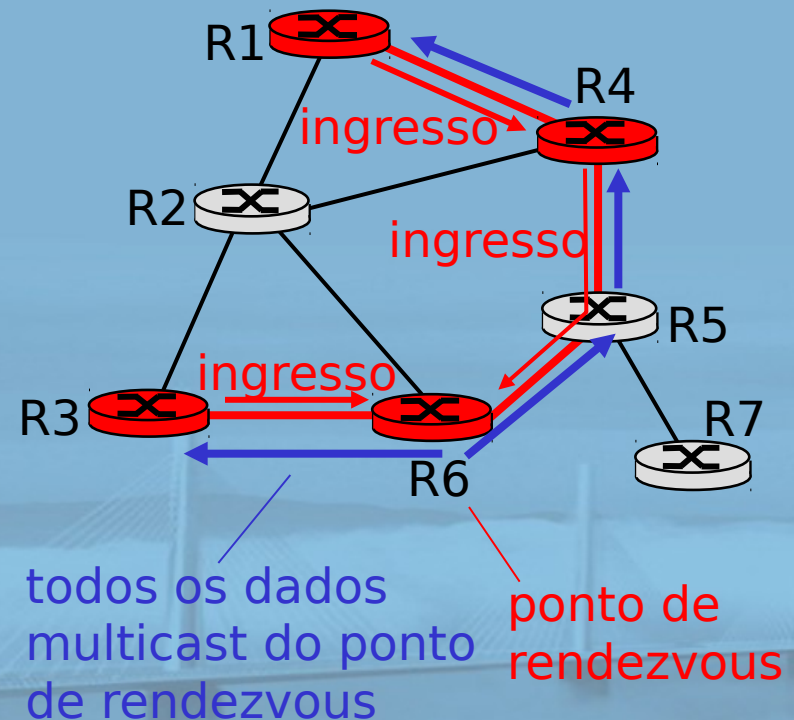
PIM – modo esparso

- enfoque baseado em centro
- roteador envia msg de *ingresso* ao Rendezvous Point (RP)
 - roteadores intermediários atualizam estado e encaminham *ingresso*
- após ingressar via RP, roteador pode passar para árvore específica da fonte
 - maior desempenho: menor concentração, caminhos mais curtos



remetente(s):

- ❑ dados unicast ao RP, que distribui pela árvore com raiz no RP
- ❑ RP pode estender árvore multicast para antes dele, até a fonte
- ❑ RP pode enviar msg *parar* se não houver destinatários conectados
 - “ninguém está ouvindo!”



Capítulo 4: Resumo

- ❑ 4.1 Introdução
- ❑ 4.2 Redes de circuitos virtuais e de datagramas
- ❑ 4.3 O que há dentro de um roteador?
- ❑ 4.4 IP: Internet Protocol
 - formato do datagrama
 - endereçamento IPv4
 - ICMP
 - IPv6
- ❑ 4.5 Algoritmos de roteamento
 - estado de enlace
 - vetor de distâncias
 - roteamento hierárquico
- ❑ 4.6 Roteamento na Internet
 - RIP
 - OSPF
 - BGP
- ❑ 4.7 Roteamento broadcast e multicast