Capítulo 4: Camada de Rede

Metas do capítulo:

- entender os princípios em que se fundamentam os serviços de rede:
  - roteamento (seleção de caminhos)
  - escalabilidade
  - como funciona um roteador
  - tópicos avançados: IPv6, multiponto
- instanciação e implementação na Internet

Resumo:

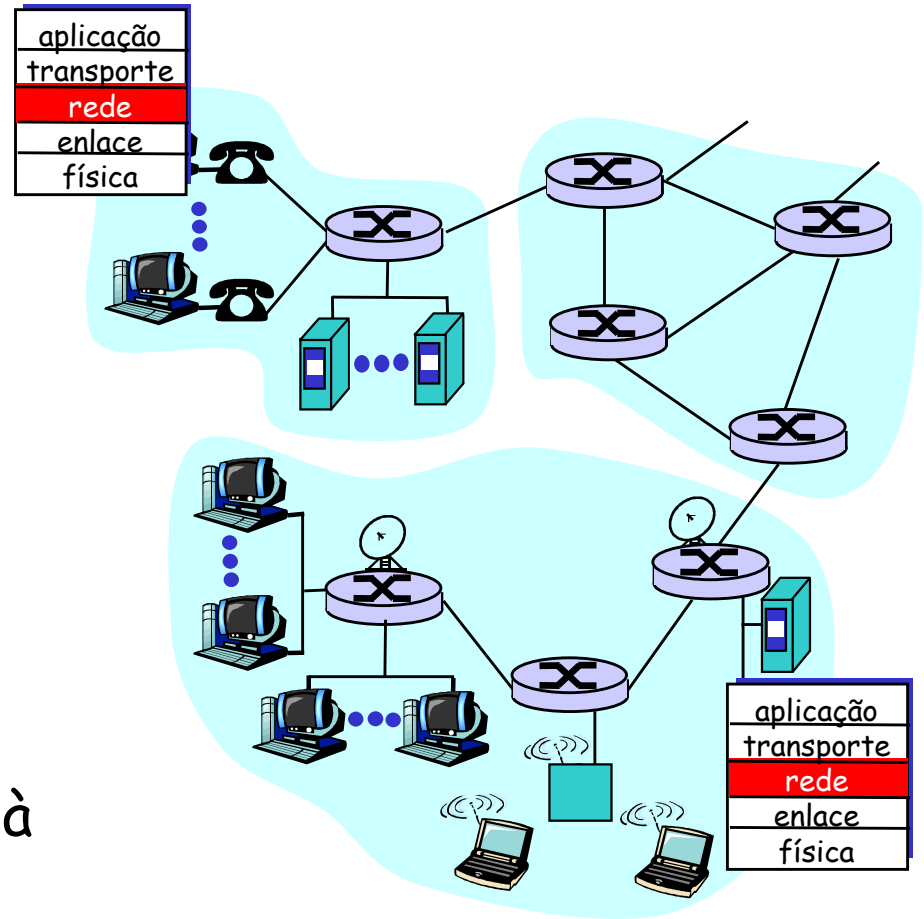
- serviços da camada de rede
- princípio de roteamento: seleção de caminhos
- roteamento hierárquico
- IP
- Protocolos de roteamento da Internet
  - dentro de um domínio
  - entre domínios
- como funciona um roteador?
- roteamento multiponto
- IPv6

Funções da camada de rede

- transporta pacote da estação remetente à receptora
- protocolos da camada de rede em *cada* estação, roteador

três funções importantes:

- *determinação do caminho:* rota seguida por pacotes da origem ao destino. *Algoritmos de roteamento*
- *comutação:* mover pacotes dentro do roteador da entrada à saída apropriada
- *estabelecimento da chamada:* algumas arquiteturas de rede requerem determinar o caminho antes de enviar os dados

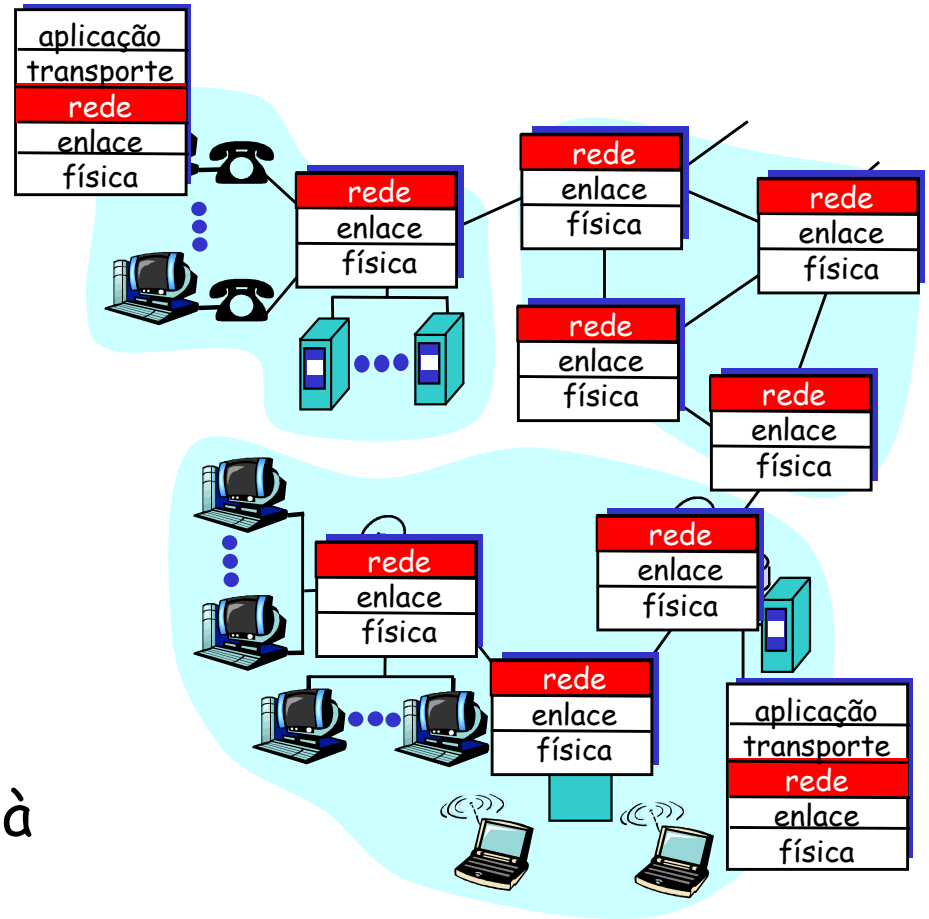


Funções da camada de rede

- transporta pacote da estação remetente à receptora
- protocolos da camada de rede em *cada* estação, roteador

três funções importantes:

- *determinação do caminho:* rota seguida por pacotes da origem ao destino. *Algoritmos de roteamento*
- *comutação:* mover pacotes dentro do roteador da entrada à saída apropriada
- *estabelecimento da chamada:* algumas arquiteturas de rede requerem determinar o caminho antes de enviar os dados



Modelo de serviço de rede

Q: Qual é o *modelo de serviço* para o "canal" que transporta pacotes do remetente ao receptor?

abstração do serviço

- largura de banda garantida?
- preservação de temporização entre pacotes (sem *jitter*)?
- entrega sem perdas?
- entrega ordenada?
- realimentar informação sobre congestionamento ao remetente?

A abstração mais importante provida pela camada de rede:

circuito virtual
ou
datagrama?

Modelos de serviço da camada de rede:

Arquitetura de Rede	Modelo de serviço	Garantias ?				Informa s/ congestion.?
		Banda	Perdas	Ordem	Tempo	
Internet	melhor esforço	nenhuma	não	não	não	não (inferido via perdas)
ATM	CBR	taxa constante	sim	sim	sim	sem congestion.
ATM	VBR	taxa garantida	sim	sim	sim	sem congestion.
ATM	ABR	mínima garantida	não	sim	não	sim
ATM	UBR	nenhuma	não	sim	não	não

- Modelo Internet está sendo estendido: Intserv, Diffserv
  - Capítulo 7

Circuitos virtuais

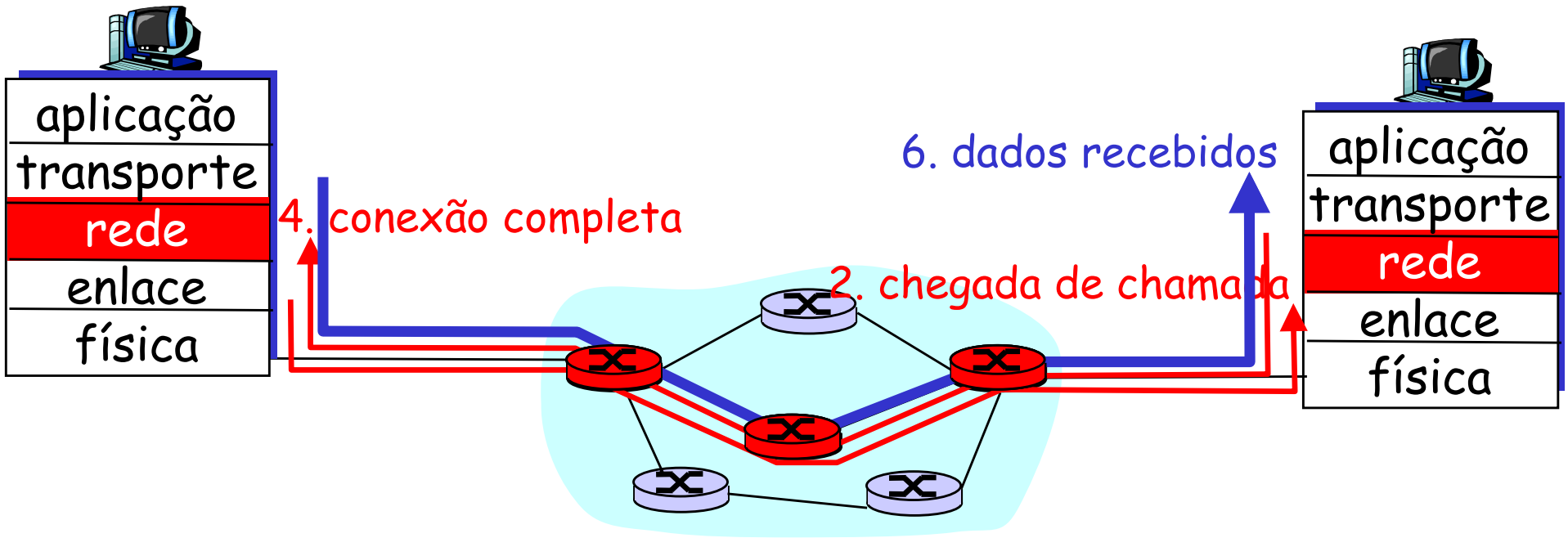
"caminho da-origem-ao-destino se comporta como um circuito telefônico"

- em termos de desempenho
- em ações da rede ao longo do caminho da-origem-ao-destino

- estabelecimento de cada chamada *antes* do envio dos dados
- cada pacote tem ident. de CV (e não endereços origem/dest)
- *cada* roteador no caminho da-origem-ao-destino mantém "estado" para cada conexão que o atravessa
  - conexão da camada de transporte só envolve os 2 sistemas terminais
- recursos de enlace, roteador (banda, *buffers*) podem ser *alocados* ao CV
  - para permitir desempenho como de um circuito

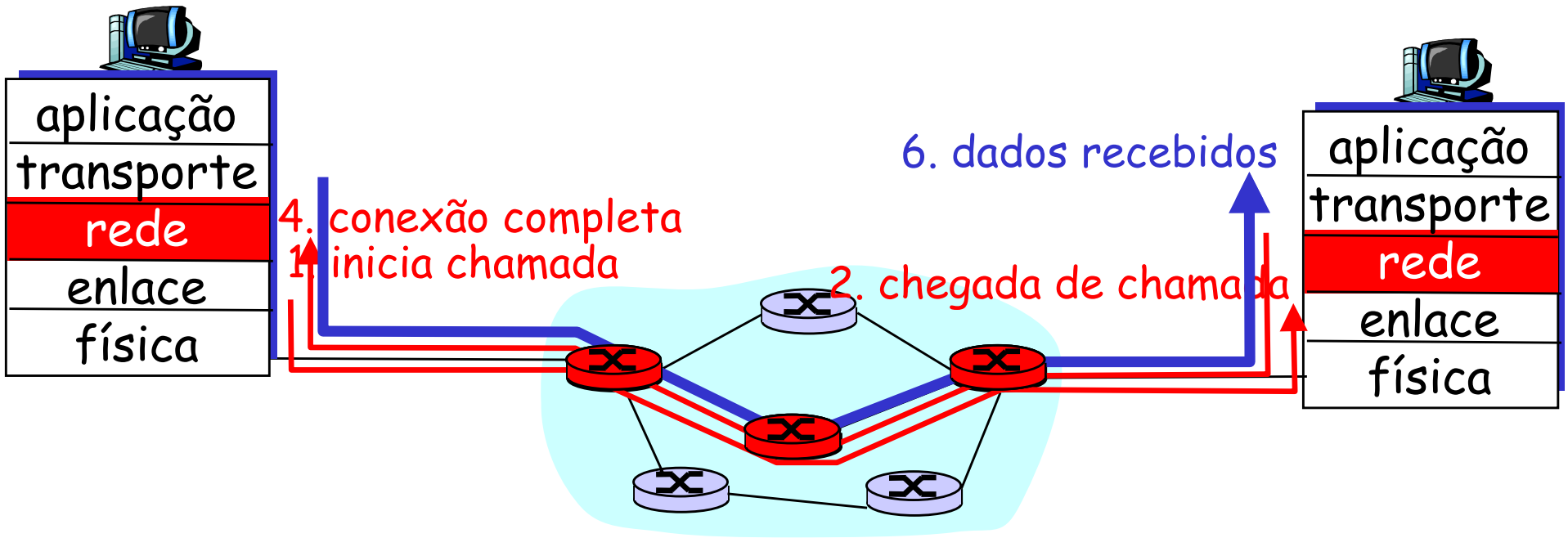
Circuitos virtuais: protocolos de sinalização

- usados para estabelecer, manter, destruir CV
- usados em ATM, frame-relay, X.25
- não usados na Internet de hoje



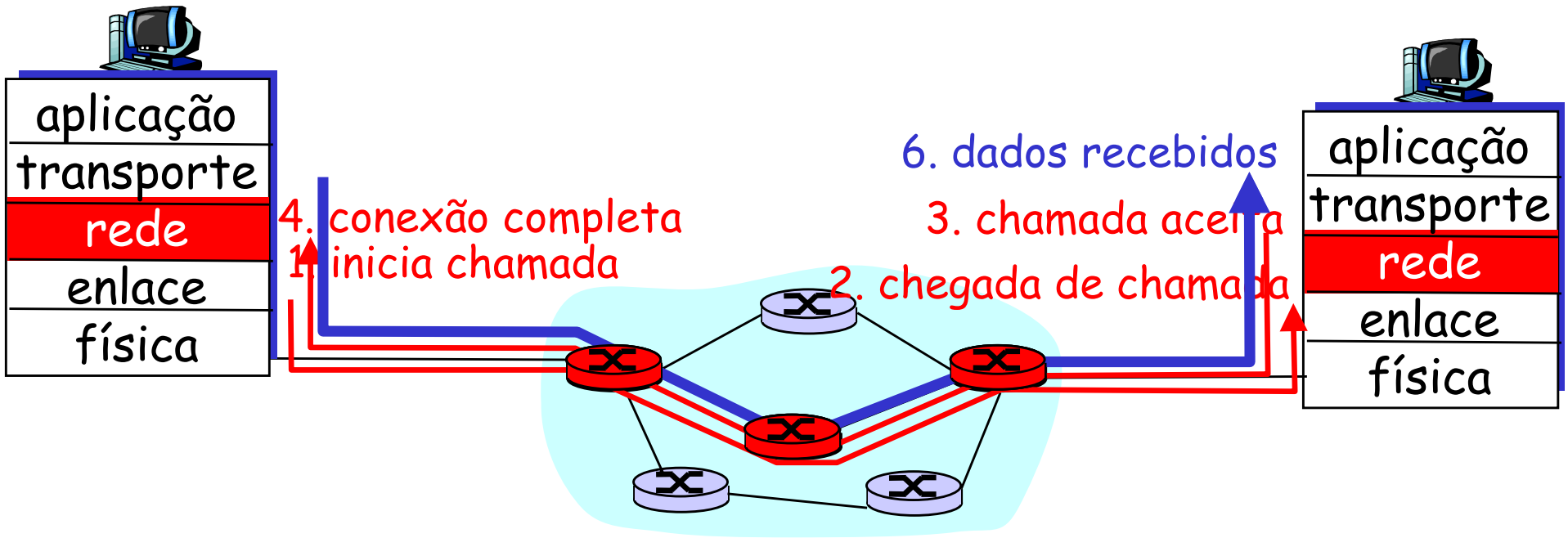
Circuitos virtuais: protocolos de sinalização

- usados para estabelecer, manter, destruir CV
- usados em ATM, frame-relay, X.25
- não usados na Internet de hoje



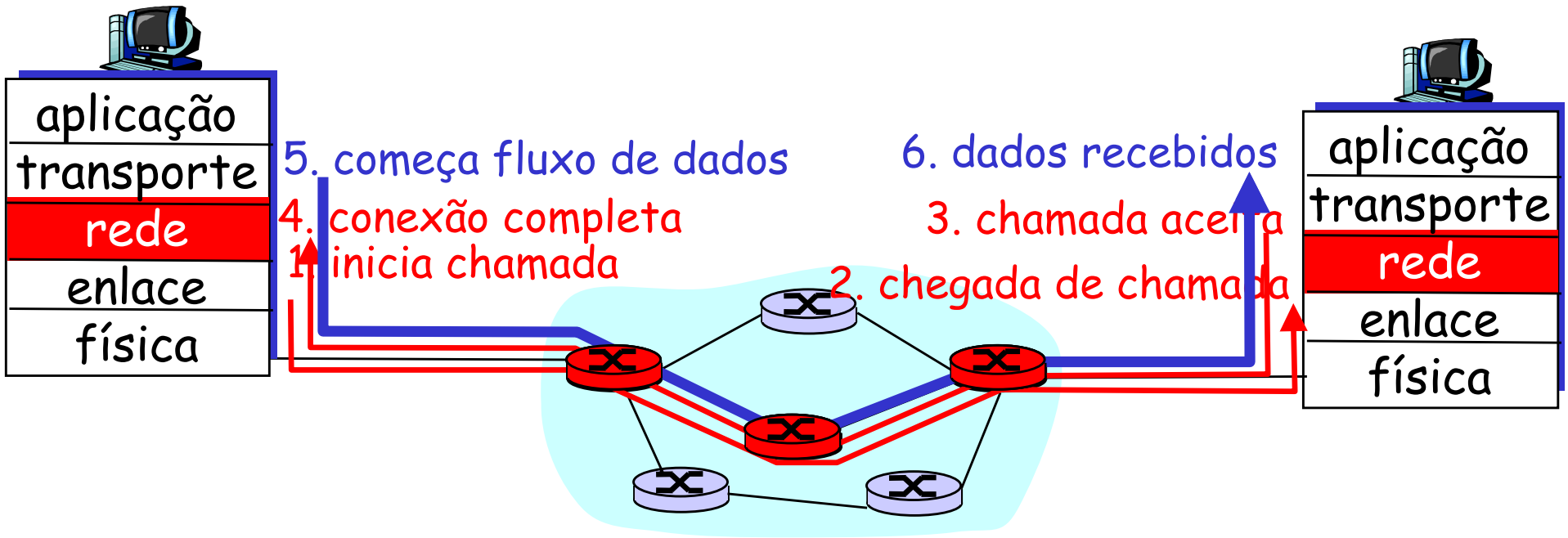
Circuitos virtuais: protocolos de sinalização

- usados para estabelecer, manter, destruir CV
- usados em ATM, frame-relay, X.25
- não usados na Internet de hoje



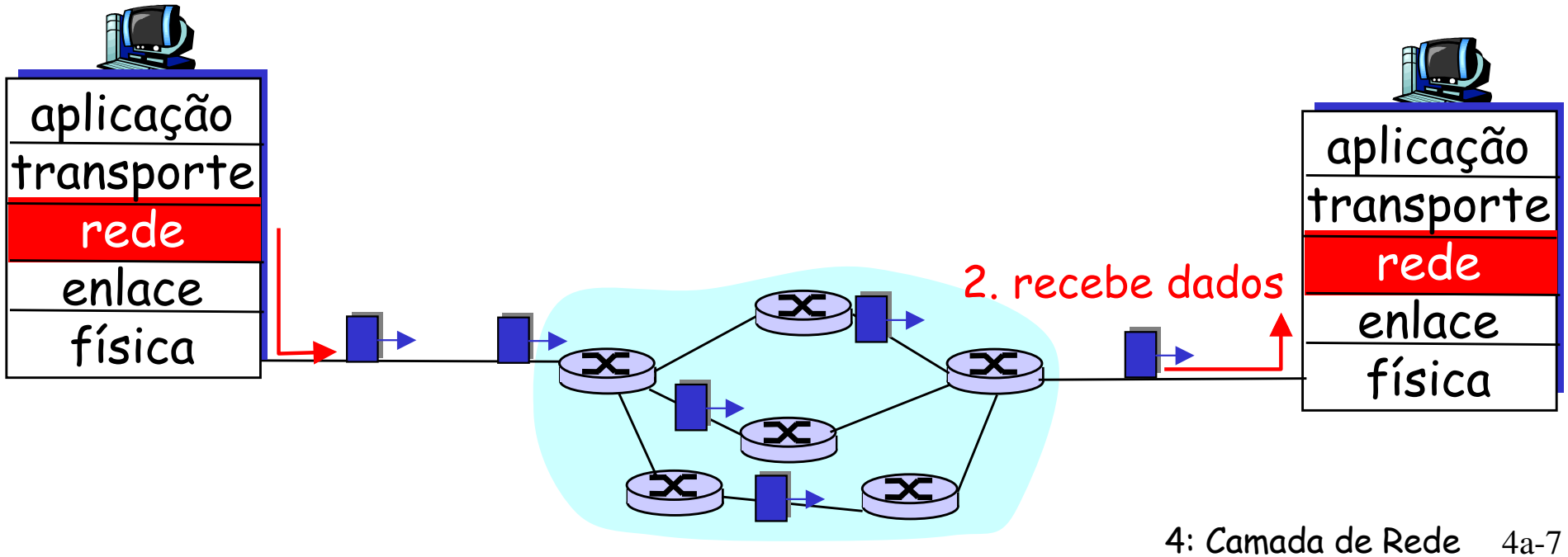
Circuitos virtuais: protocolos de sinalização

- usados para estabelecer, manter, destruir CV
- usados em ATM, frame-relay, X.25
- não usados na Internet de hoje



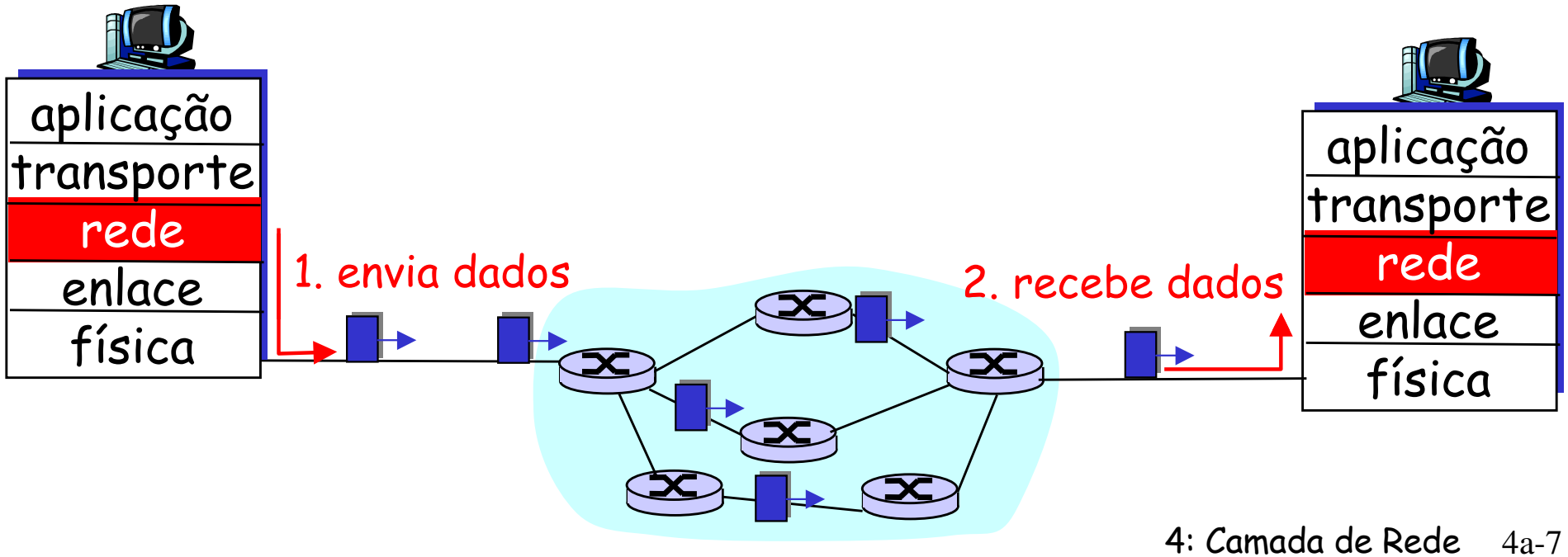
Rede de datagramas: o modelo da Internet

- não requer estabelecimento de chamada na camada de rede
- roteadores: não guardam estado sobre conexões fim a fim
  - não existe o conceito de "conexão" na camada de rede
- pacotes são roteados tipicamente usando endereços de destino
  - pacotes entre o mesmo par origem-destino podem seguir caminhos diferentes



Rede de datagramas: o modelo da Internet

- não requer estabelecimento de chamada na camada de rede
- roteadores: não guardam estado sobre conexões fim a fim
  - não existe o conceito de "conexão" na camada de rede
- pacotes são roteados tipicamente usando endereços de destino
  - pacotes entre o mesmo par origem-destino podem seguir caminhos diferentes



Rede de datagramas ou CVs: por quê?

Internet

- troca de dados entre computadores
  - serviço "elástico", sem reqs. temporais estritos
- sistemas terminais "inteligentes" (computadores)
  - podem se adaptar, exercer controle, recuperar de erros
  - núcleo da rede simples, complexidade na "borda"
- muitos tipos de enlaces
  - características diferentes
  - serviço uniforme difícil

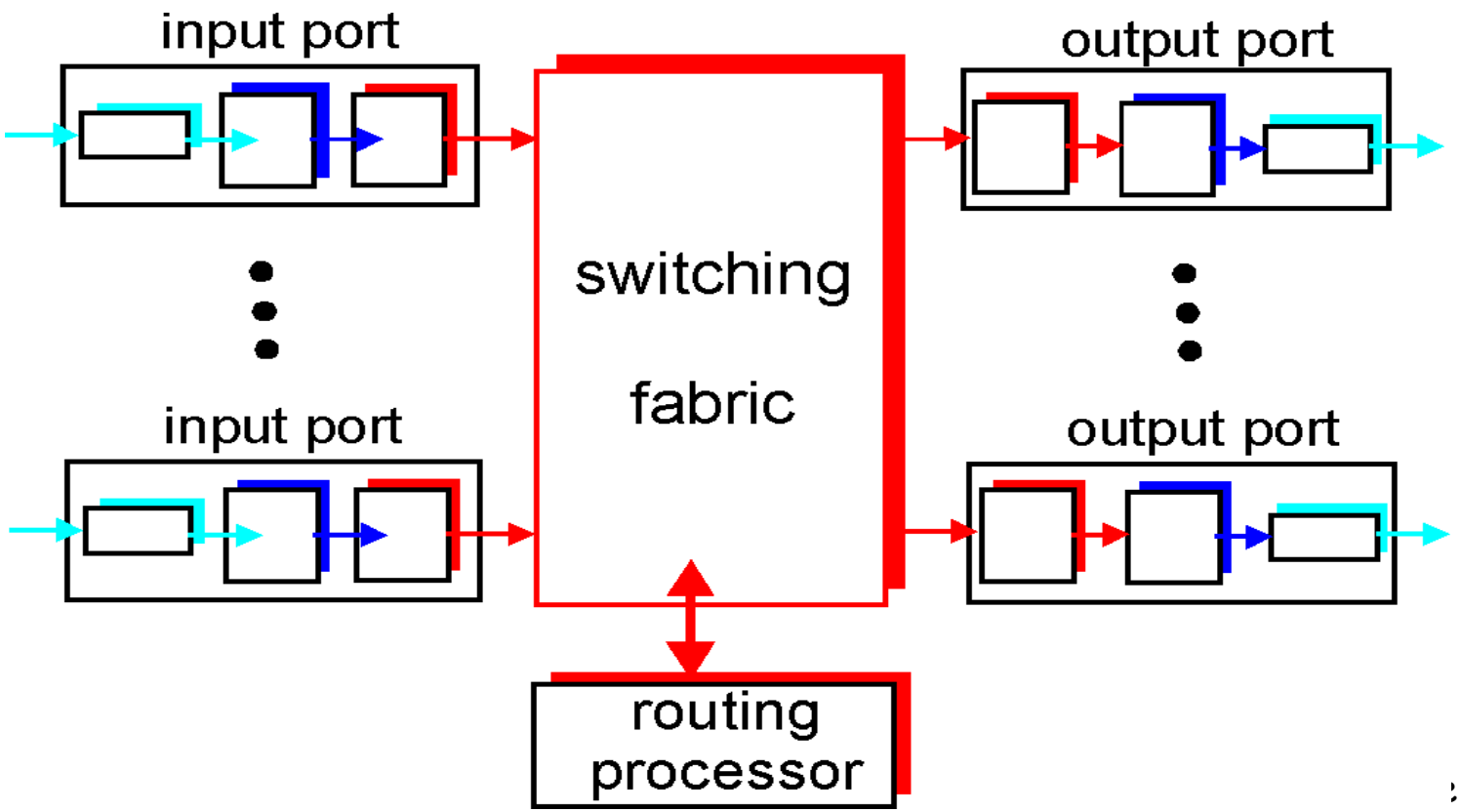
ATM

- evoluiu da telefonia
- conversação humana:
  - temporização estrita, requisitos de confiabilidade
  - requer serviço garantido
- sistemas terminais "burros"
  - telefones
  - complexidade dentro da rede

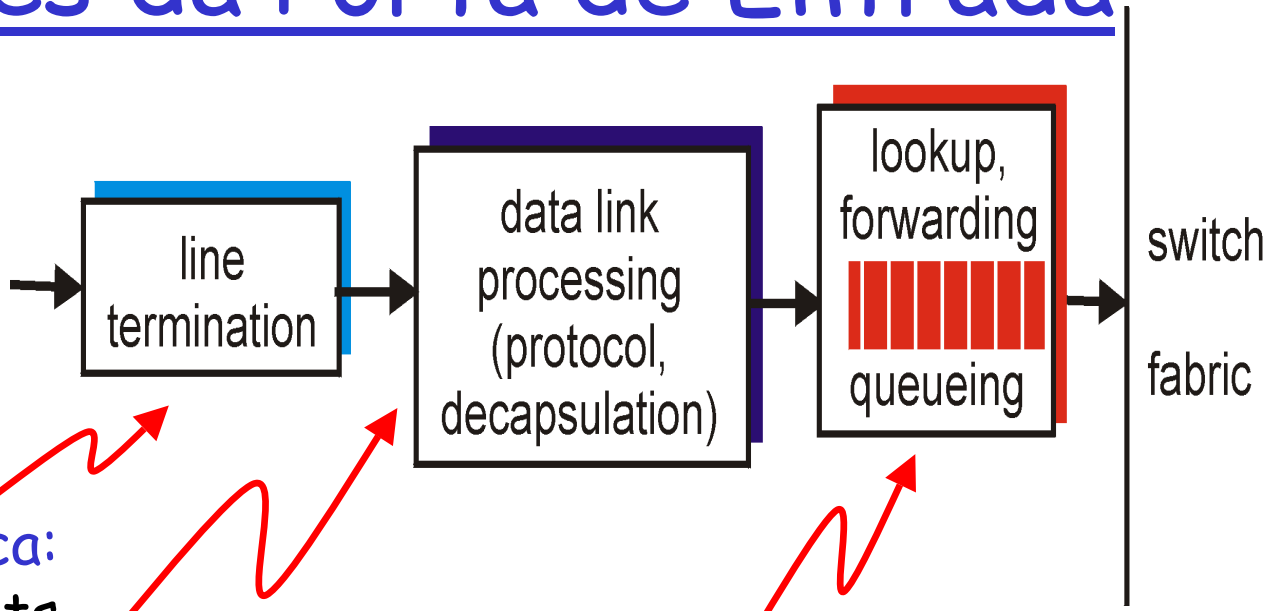
Sumário de Arquitetura de Roteadores

Duas funções chave de roteadores:

- usam algoritmos/protocolos de roteamento (RIP, OSPF, BGP)
- *comutam* datagramas do enlace de entrada para a saída



Funções da Porta de Entrada



Camada f'ísica:
recepção de bits

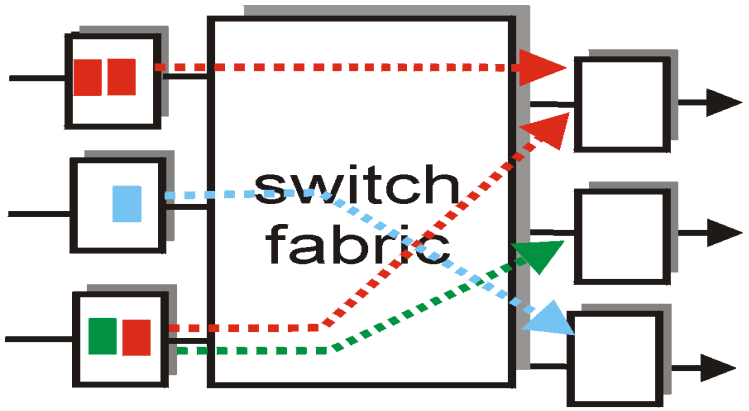
Camada de enlace:
p.ex., Ethernet
veja capítulo 5

Comutação descentralizada:

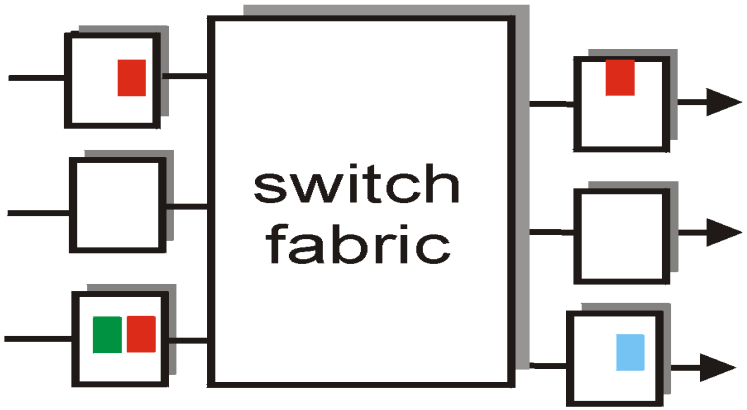
- dado o dest do datagrama, procura porta de saída usando tab. de rotas na memória da porta de entrada
- meta: completar processamento da porta de entrada na 'velocidade da linha'
- filas: se datagramas chegam mais rápido que taxa de re-envio para matriz de comutação

Filas na Porta de Entrada

- Se matriz de comutação for mais lenta do que a soma das portas de entrada juntas -> pode haver filas nas portas de entrada
- Bloqueio cabeça-de-linha (Head-of-the-Line - HOL): datagrama na cabeça da fila impede outros na mesma fila de avançarem
- *retardo de enfileiramento e perdas devido ao transbordo do buffer de entrada!*

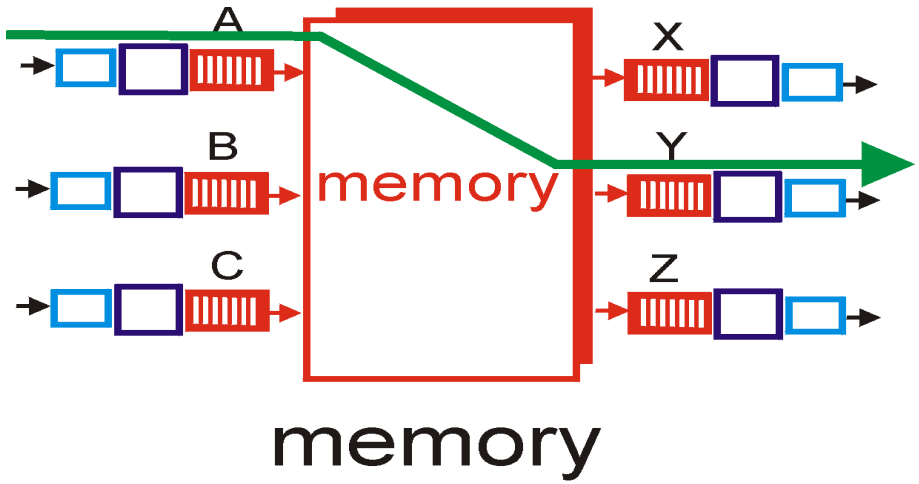


output port contention
at time t - only one red
packet can be transferred

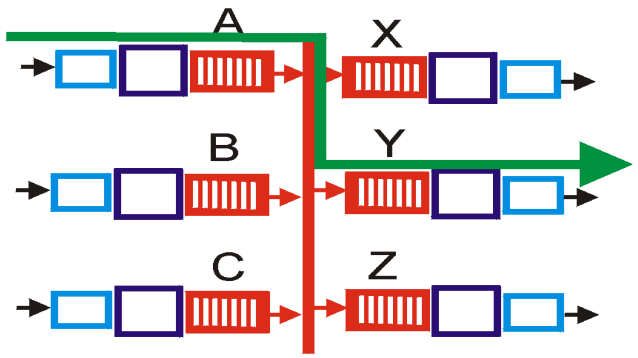


green packet
experiences HOL blocking

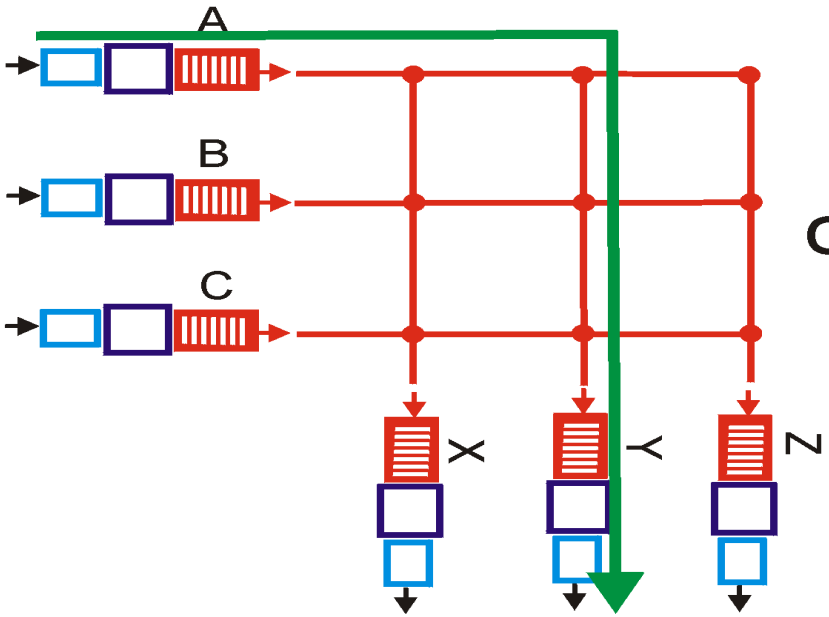
Três tipos de matriz de comutação



memory



bus

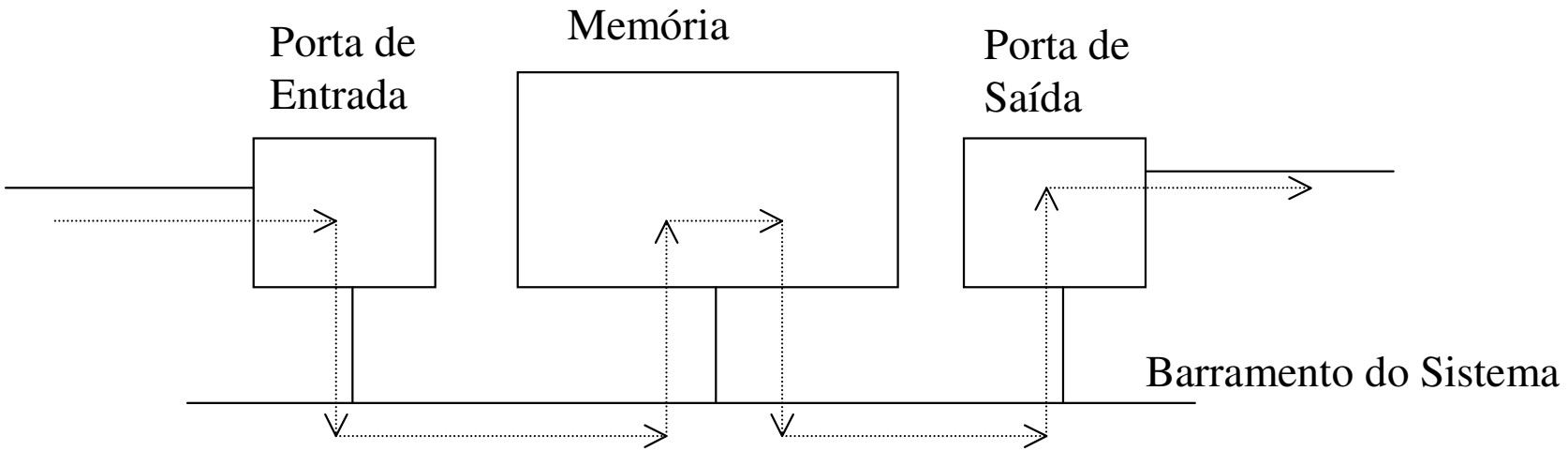


crossbar

Comutação via Memória

Roteadores da primeira geração:

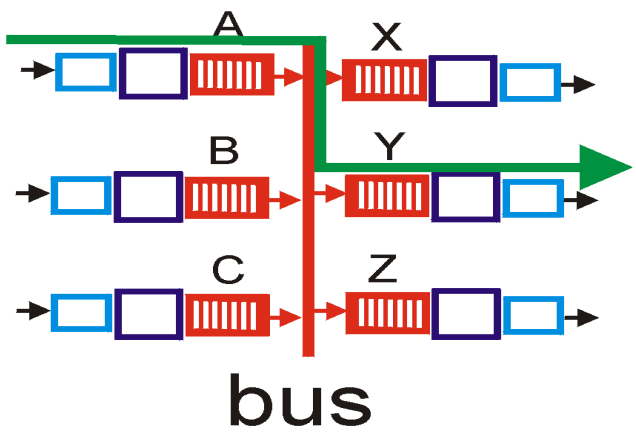
- pacote copiado pelo processador (único) do sistema
- velocidade limitada pela largura de banda da memória (2 travessias do barramento por datagrama)



Roteadores modernos:

- processador da porta de entrada consulta tabela, copia para a memória
- Cisco Catalyst 8500

Comutação via Barramento

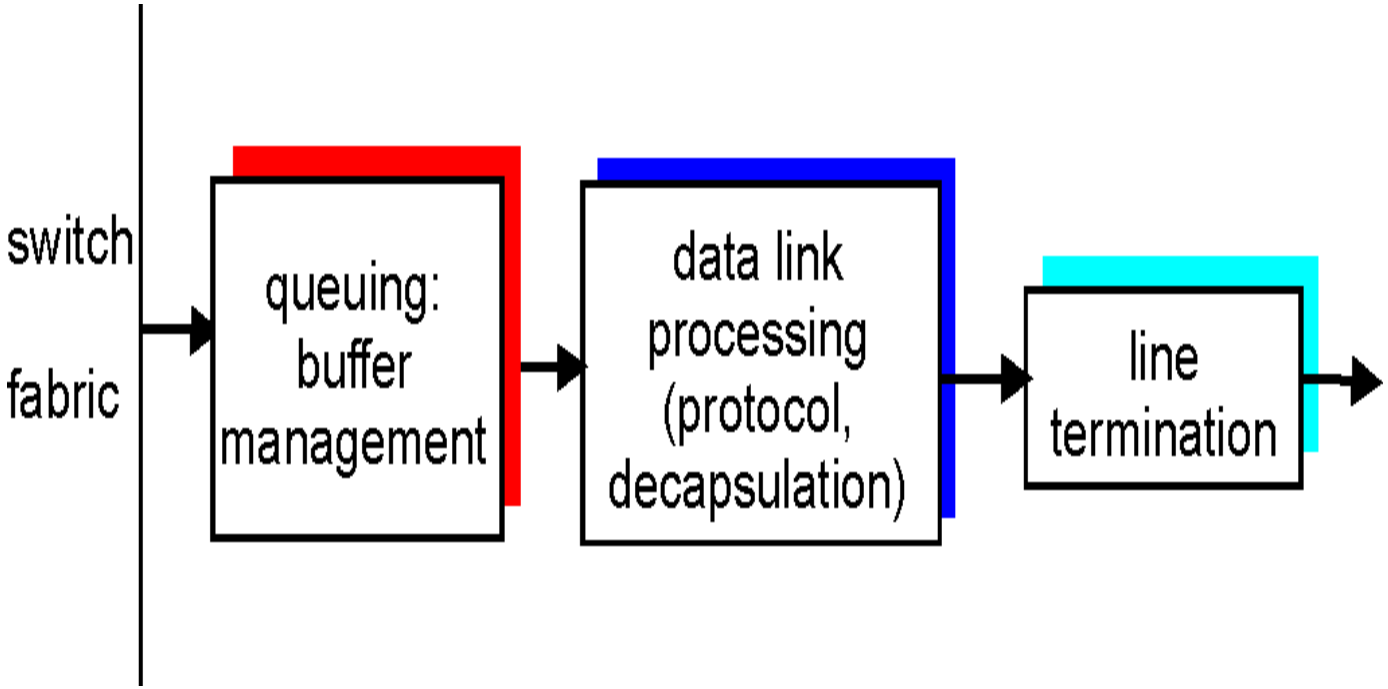


- datagrama da memória da porta de entrada à memória da porta de saída via um barramento compartilhado
- contenção pelo barramento: taxa de comutação limitada pela largura de banda do barramento
- Barramento de 1 Gbps, Cisco 1900: velocidade suficiente para roteadores de acesso e corporativos (mas não regionais ou de backbone)

Comutação via uma Rede de Interconexão

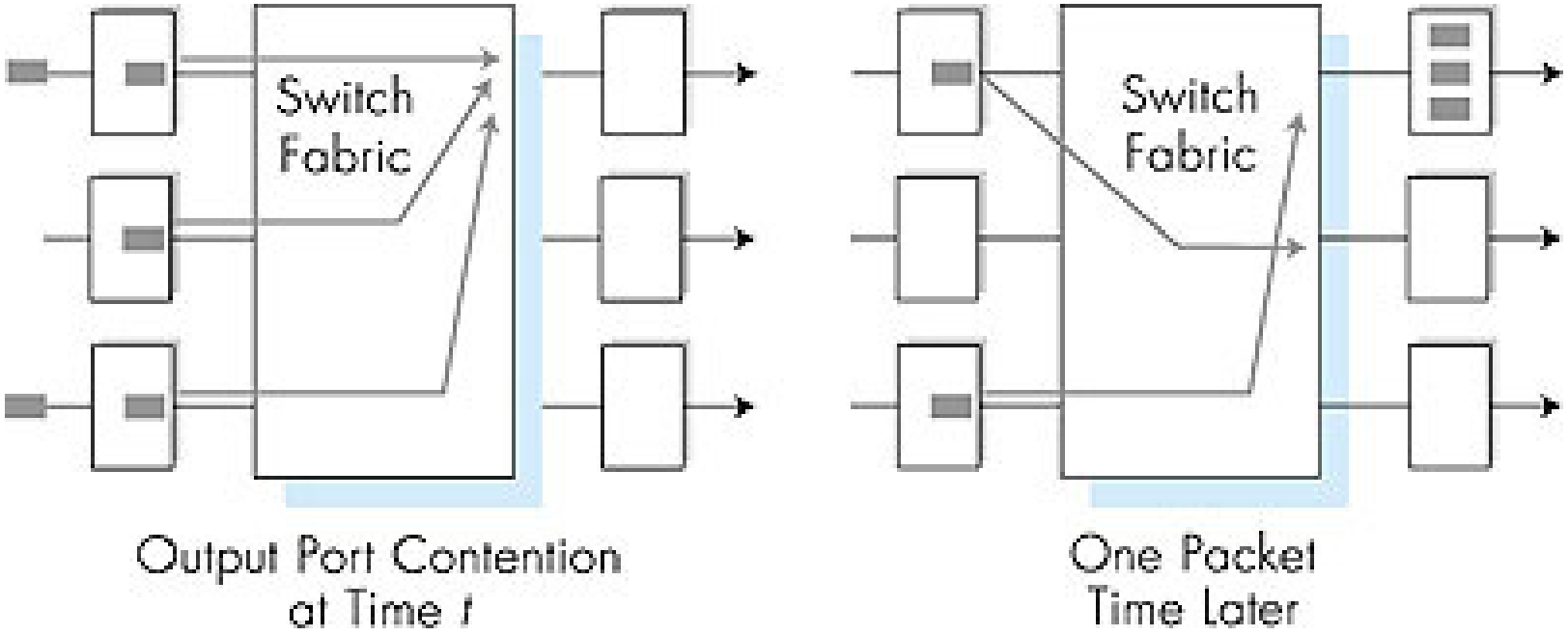
- supera limitações de banda dos barramentos
- Redes Banyan, outras redes de interconexão desenvolvidas inicialmente para interligar processadores num multiprocessador
- Projeto avançado: fragmentar datagrama em células de tamanho fixo, comutar células através da matriz de comutação.
- Cisco 12000: comuta N Gbps pela rede de interconexão.

Porta de Saída



- *Buffers* necessários quando datagramas chegam da matriz de comutação mais rapidamente que a taxa de transmissão
- *Disciplina de escalonamento* escolhe um dos datagramas enfileirados para transmissão

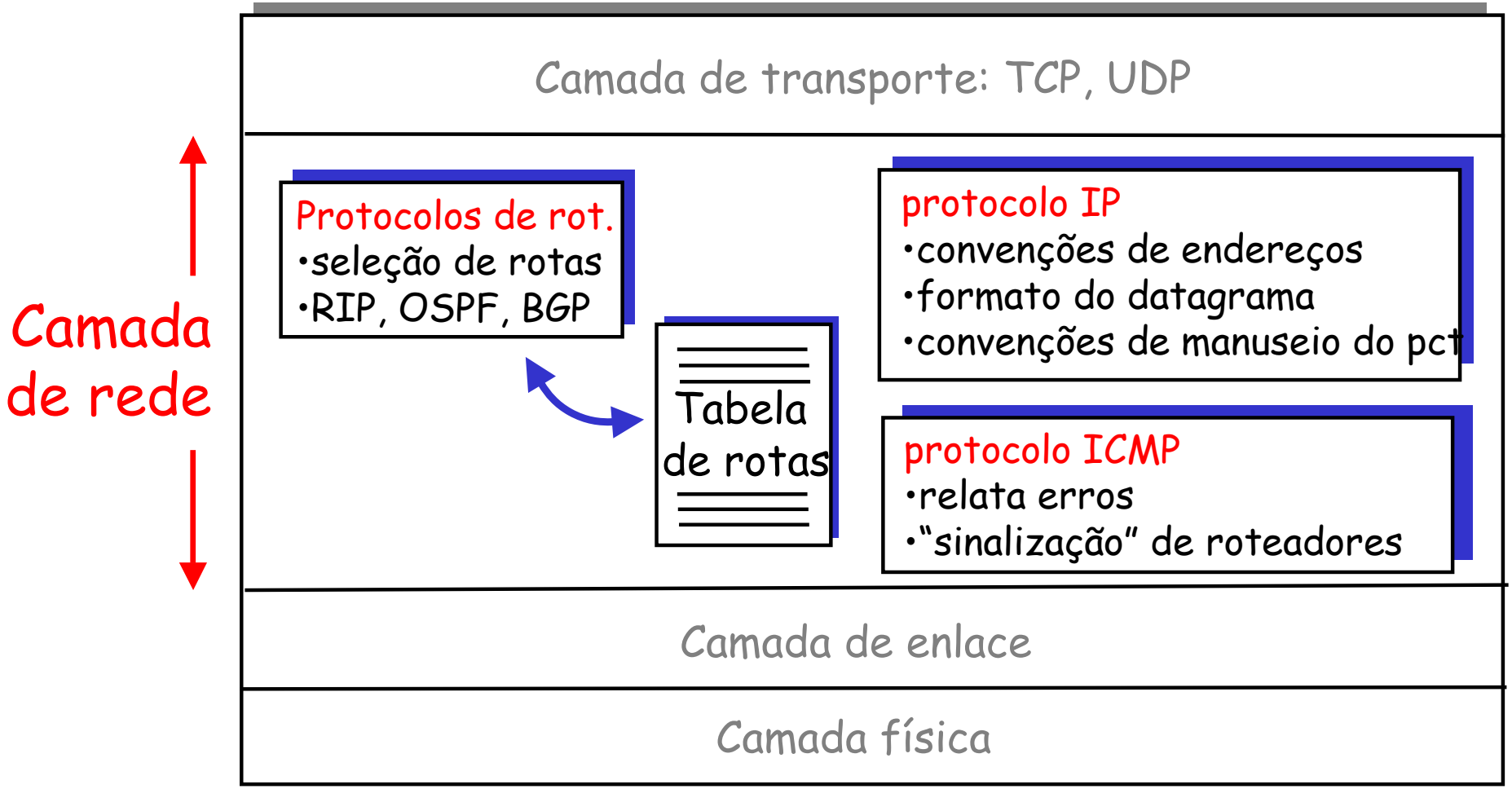
Filas na Porta de Saída



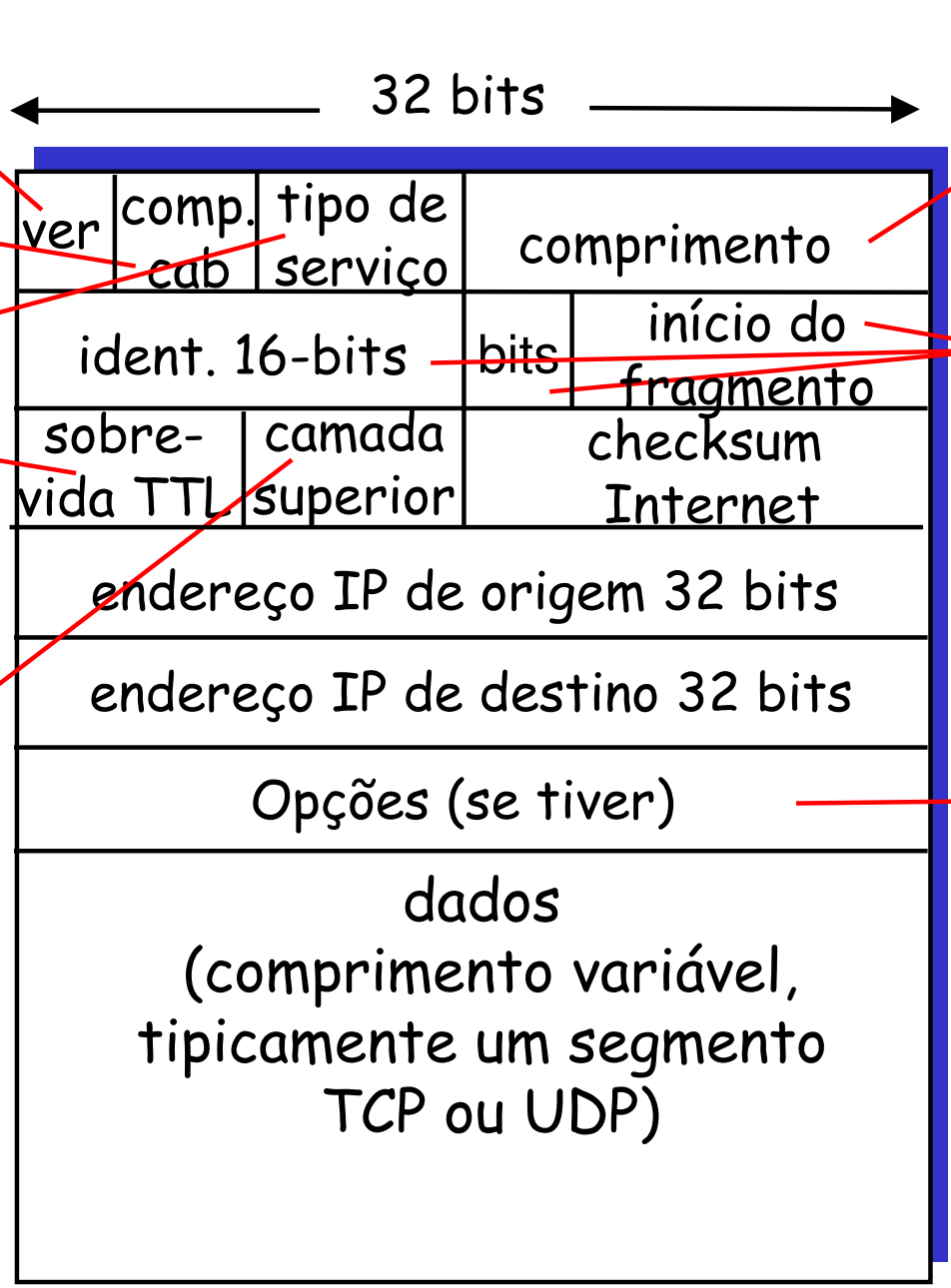
- usa buffers quando taxa de chegada através do comutador excede taxa de transmissão de saída
- *enfileiramento (retardo), e perdas devidas ao transbordo do buffer da porta de saída!*

A Camada de Rede na Internet

Funções da camada de rede em estações, roteadores:



Formato do datagrama IP



número da versão do protocolo IP

comprimento do cabeçalho (bytes)

"tipo" dos dados (DS)

número máximo de enlaces restantes (decrementado a cada roteador)

protocolo da camada superior ao qual entregar os dados

comprimento total do datagrama (bytes)

para fragmentação/ remontagem

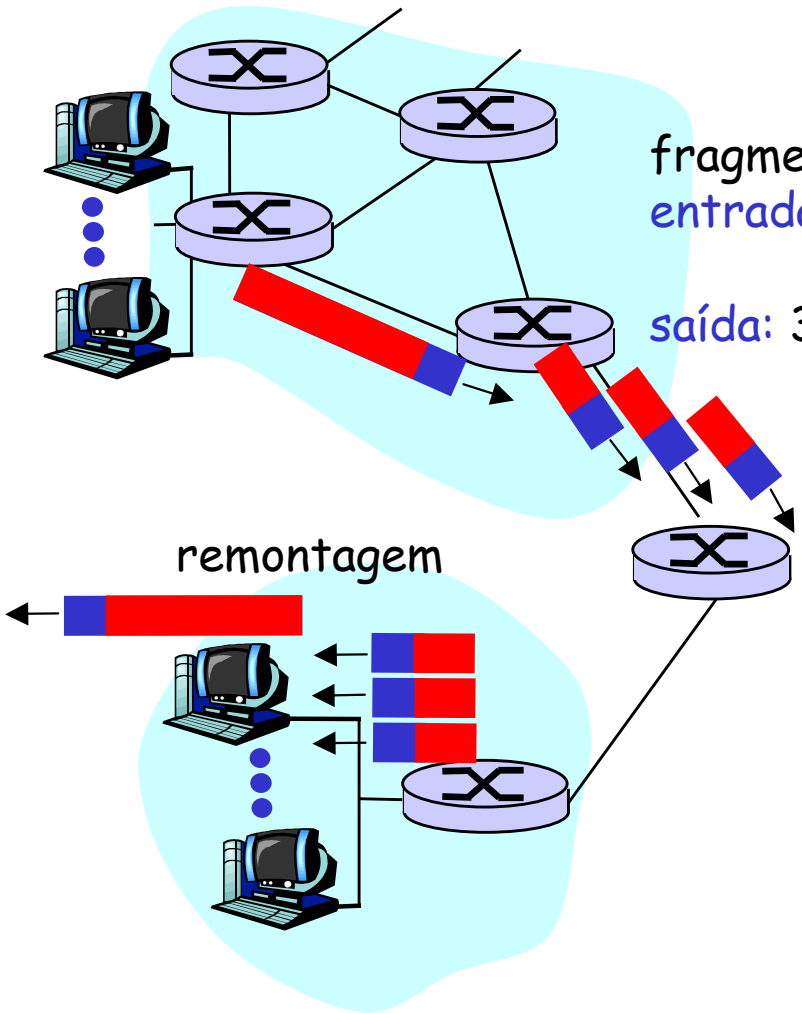
p.ex. temporizador, registrar rota seguida, especificar lista de roteadores a visitar.

Qual o overhead com TCP?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + overhead aplic.

IP: Fragmentação & Remontagem

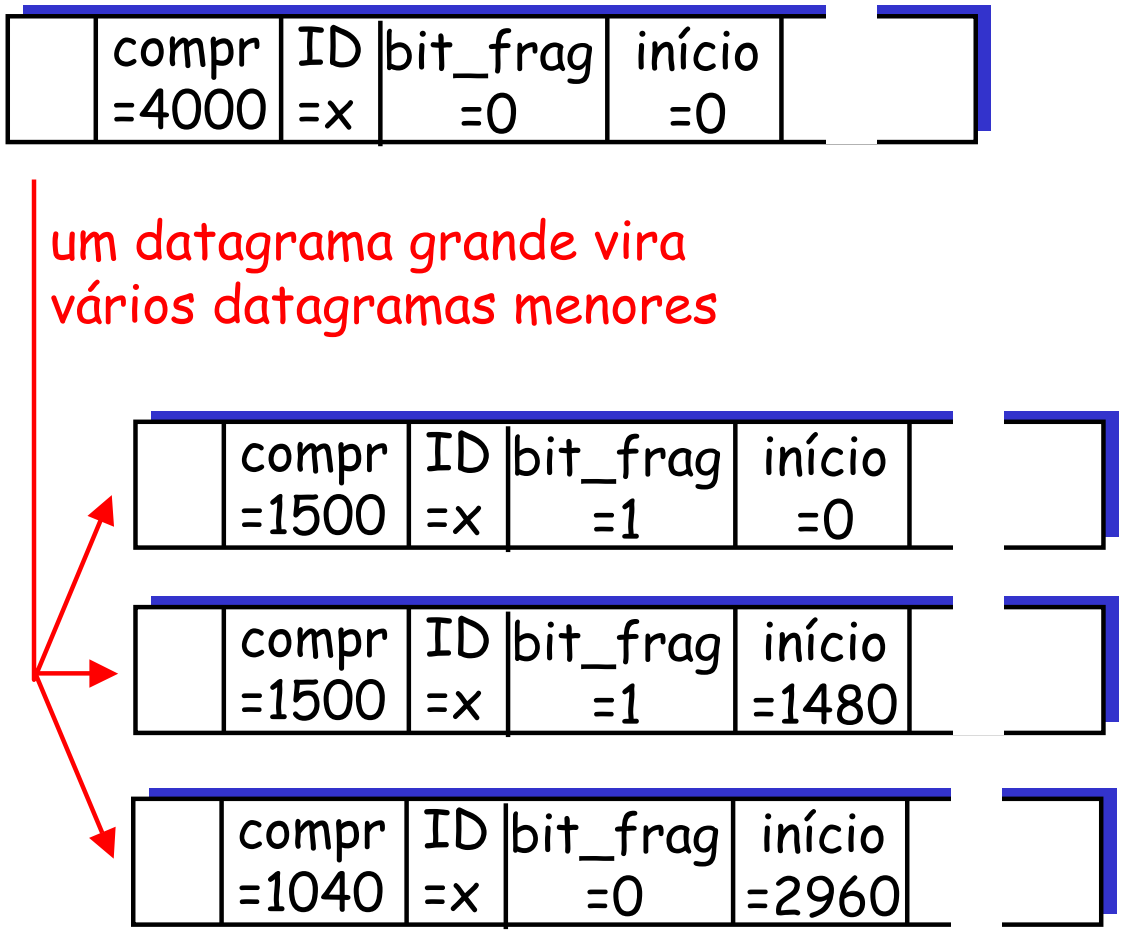
- cada enlace de rede tem MTU (max.transmission unit) - maior tamanho possível de quadro neste enlace.
  - tipos diferentes de enlace têm MTUs diferentes
- datagrama IP muito grande dividido ("fragmentado") dentro da rede
  - um datagrama vira vários datagramas
  - "remontado" apenas no destino final
  - bits do cabeçalho IP usados para identificar, ordenar fragmentos relacionados



IP: Fragmentação & Remontagem

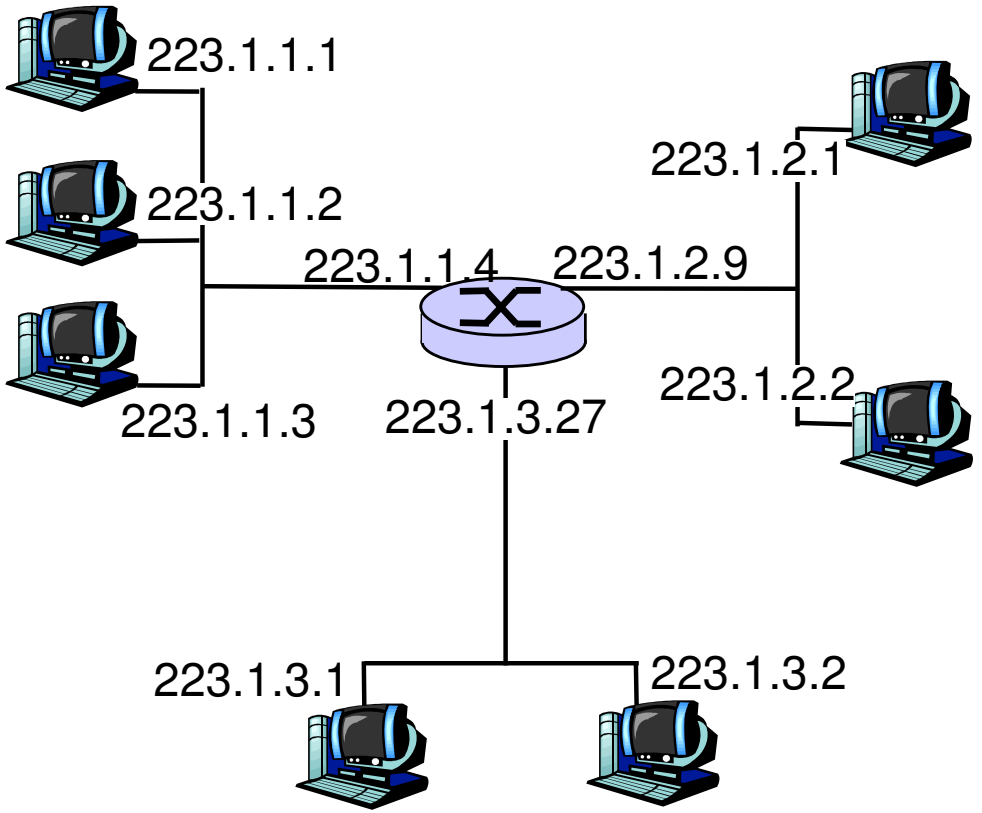
Exemplo

- Datagrama com 4000 bytes
- MTU = 1500 bytes



Endereçamento IP: introdução

- endereço IP: ident. de 32-bits para *interface* de estação, roteador
- *interface:* conexão entre estação, roteador e enlace físico
  - roteador típico tem múltiplas interfaces
  - estação pode ter múltiplas interfaces
  - endereço IP associado à interface, não à estação ou roteador

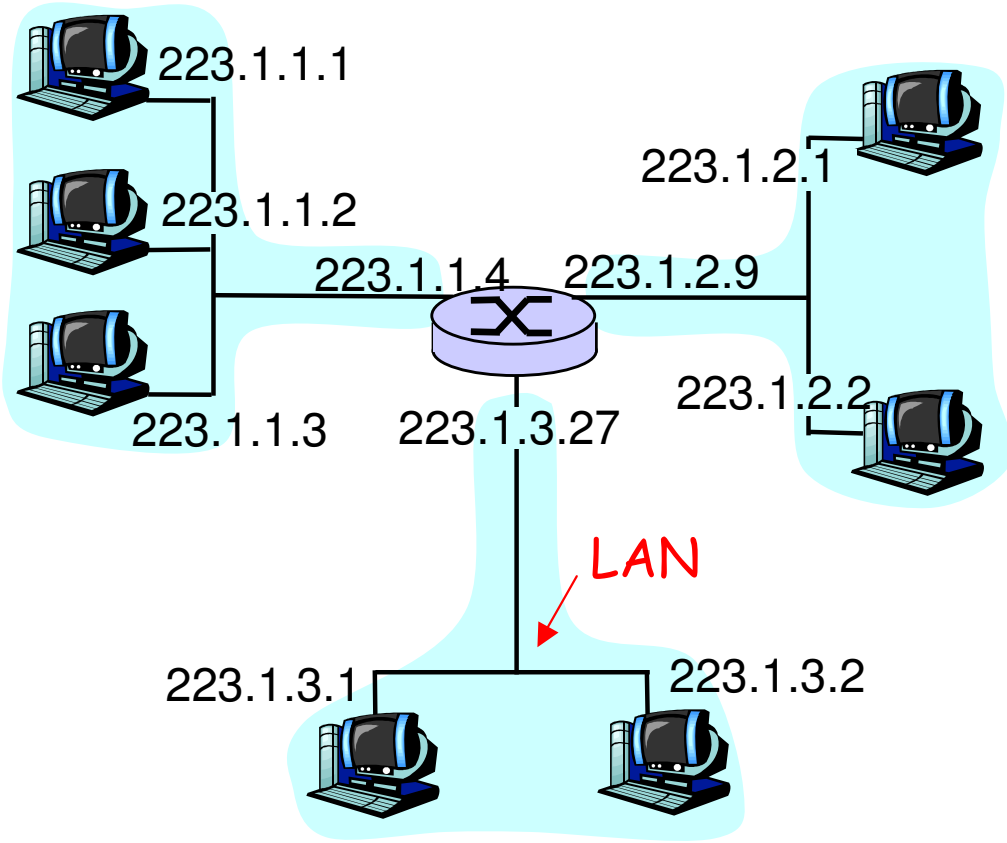


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Endereçamento IP

- endereço IP:
  - parte de rede (bits de mais alta ordem)
  - parte de estação (bits de mais baixa ordem)
- *O que é uma rede IP?* (da perspectiva do endereço IP)
  - interfaces de dispositivos com a mesma parte de rede nos seus endereços IP
  - podem alcançar um ao outro sem passar por um roteador

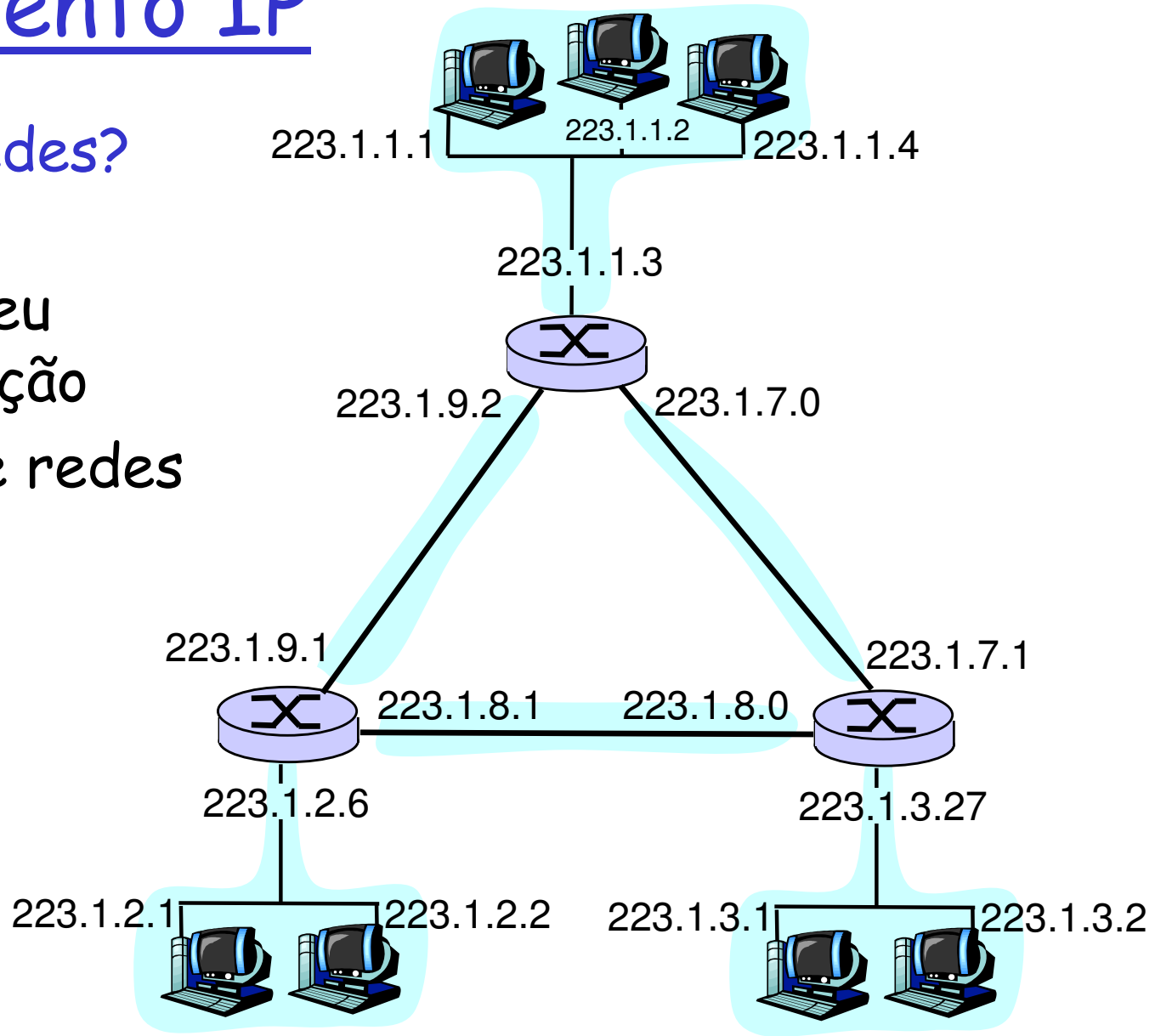


Esta rede consiste de 3 redes IP (para endereços IP começando com 223, os primeiros 24 bits são a parte de rede)

Endereçamento IP

Como achar as redes?

- dissociar cada interface do seu roteador, estação
- criar "ilhas" de redes isoladas

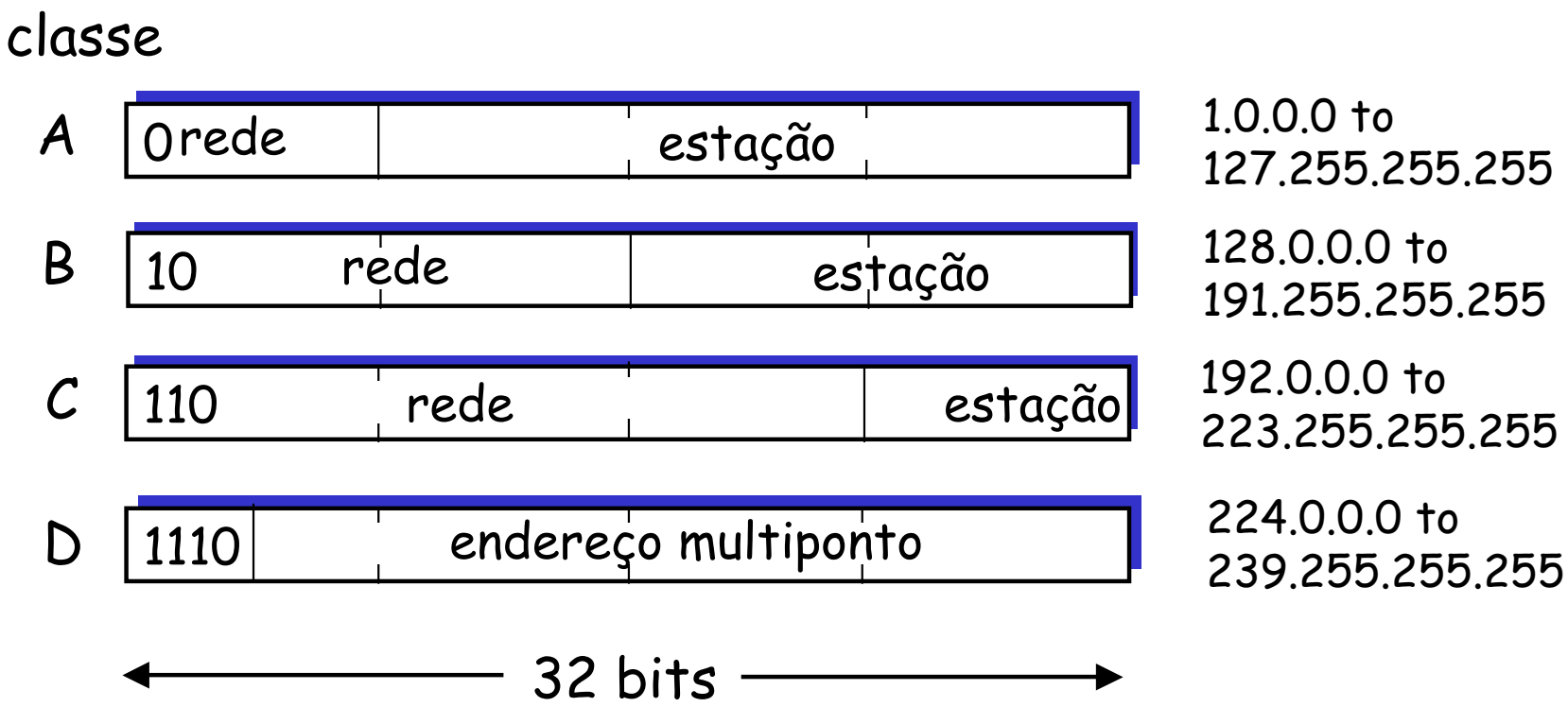


Sistema interligado consistindo de seis redes

Endereços IP

dada a noção de "rede", vamos reexaminar endereços IP:

endereçamento "baseado em classes":



Endereçamento IP: CIDR

- Endereçamento baseado em classes:
  - uso ineficiente e esgotamento do espaço de endereços
  - p.ex., rede da classe B aloca endereços para 65K estações, mesmo se houver apenas 2K estações nessa rede
- CIDR: Classless InterDomain Routing
  - parte de rede do endereço de comprimento arbitrário
  - formato de endereço: a.b.c.d/x, onde x é no. de bits na parte de rede do endereço

200.23.16.0/23

Endereços IP: como conseguir um?

Estações (parte de estação):

- codificado pelo administrador num arquivo
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
- DHCP: Dynamic Host Configuration Protocol: obtém endereço dinamicamente: "plug-and-play"
  - estação difunde mensagem "DHCP discover"
  - servidor DHCP responde com "DHCP offer"
  - estação solicita endereço IP: "DHCP request"
  - servidor DHCP envia endereço: "DHCP ack"

Endereços IP: como conseguir um?

Rede (parte de rede):

- conseguir alocação a partir do espaço de endereços do seu provedor IP

Bloco do provedor	11001000 00010111 00010000	00000000	200.23.16.0/20
Organização 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organização 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organização 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organização 7	11001000 00010111 00011110	00000000	200.23.30.0/23

DHCP: Dynamic Host Configuration Protocol

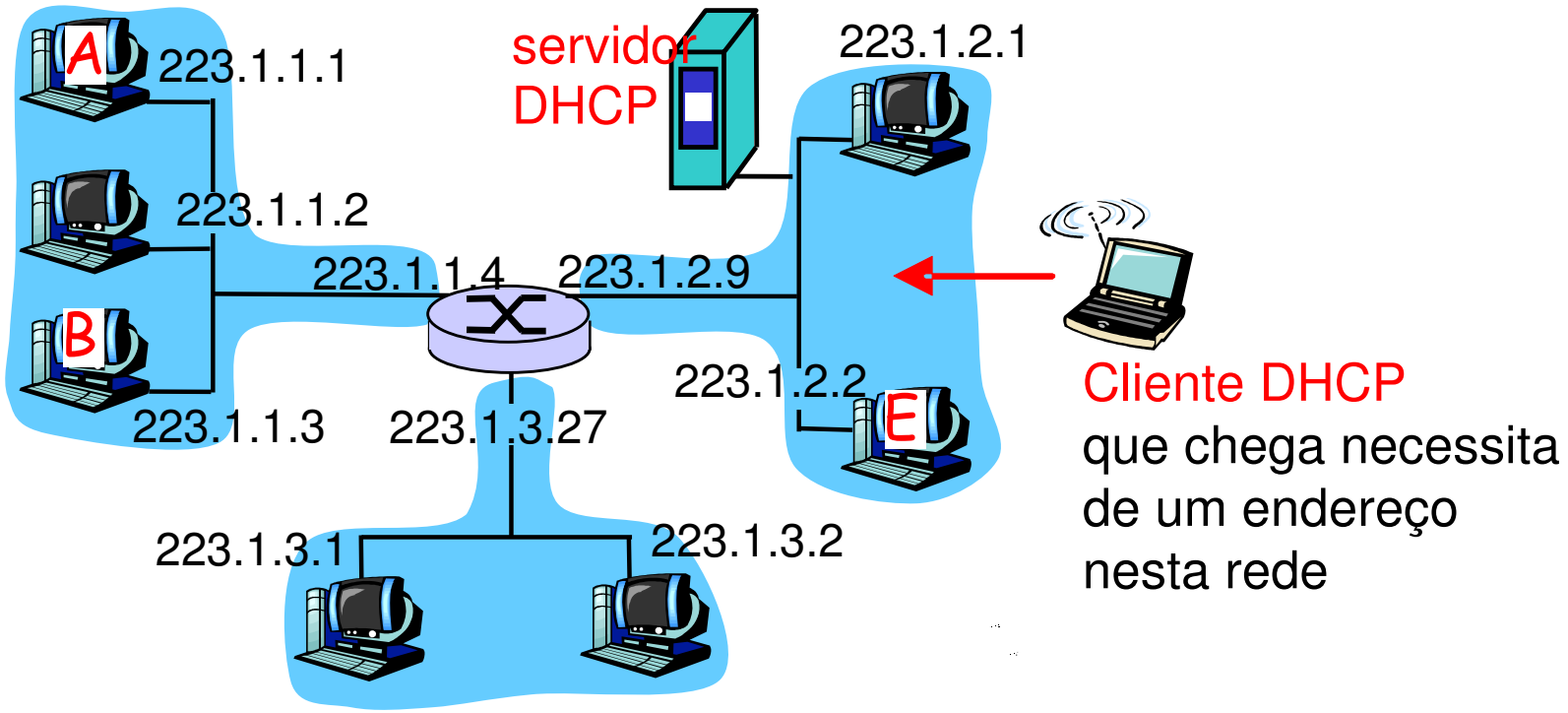
Objetivo: permite que endereços IPs sejam dinamicamente atribuídos pelos servidores de rede aos hosts quando estes se conectam a rede

- Permite a reutilização de endereços (os endereços são mantidos enquanto a máquina está ligada)
- Dá suporte a usuários móveis que desejem conectar-se a rede

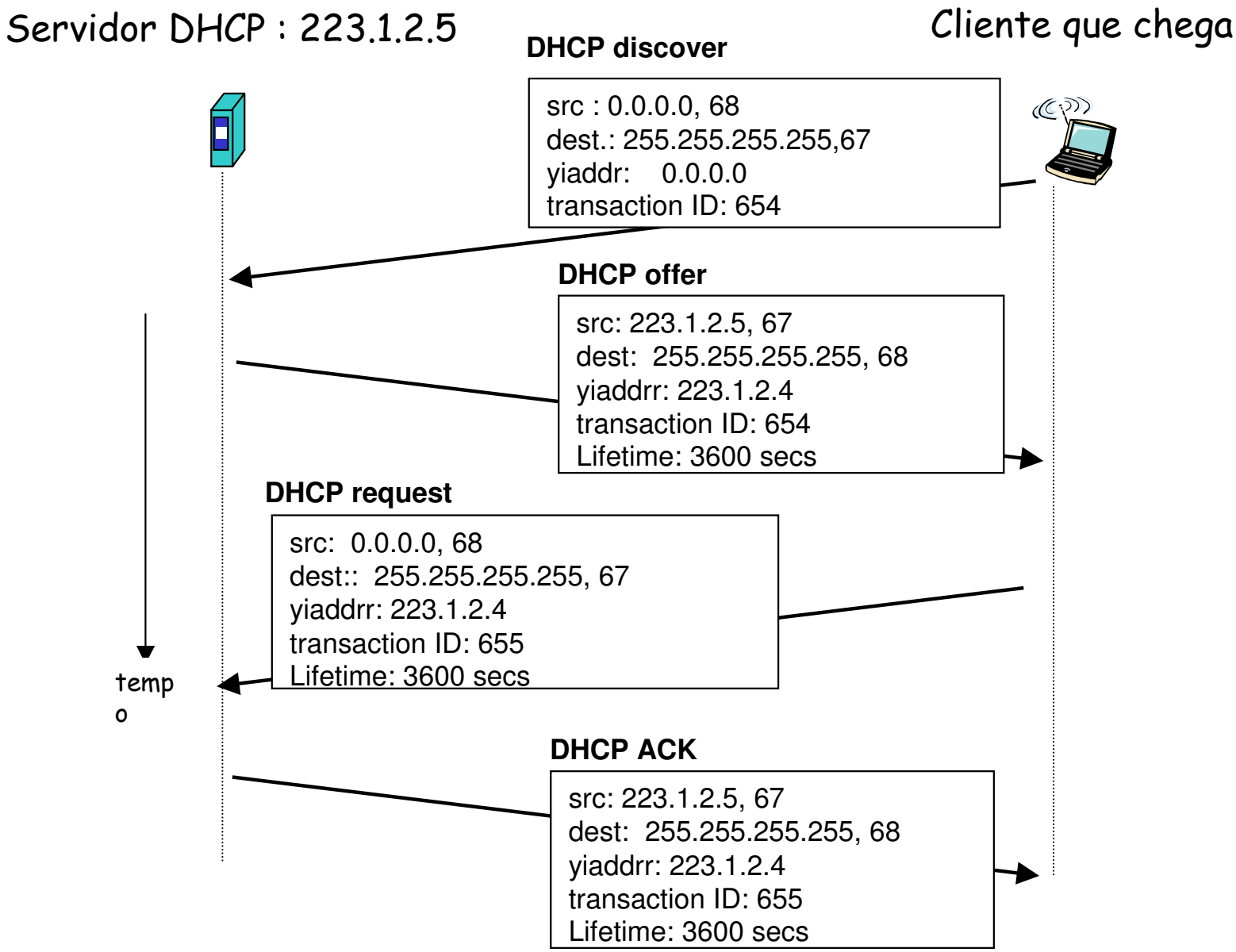
Visão geral DHCP:

- host envia msg "DHCP discover" via broadcast
- Servidor DHCP responde com msg "DHCP offer"
- host requisita endereço IP: msg "DHCP request"
- Servidor DHCP envia endereço: msg "DHCP ack"

DHCP: cenário cliente-servidor

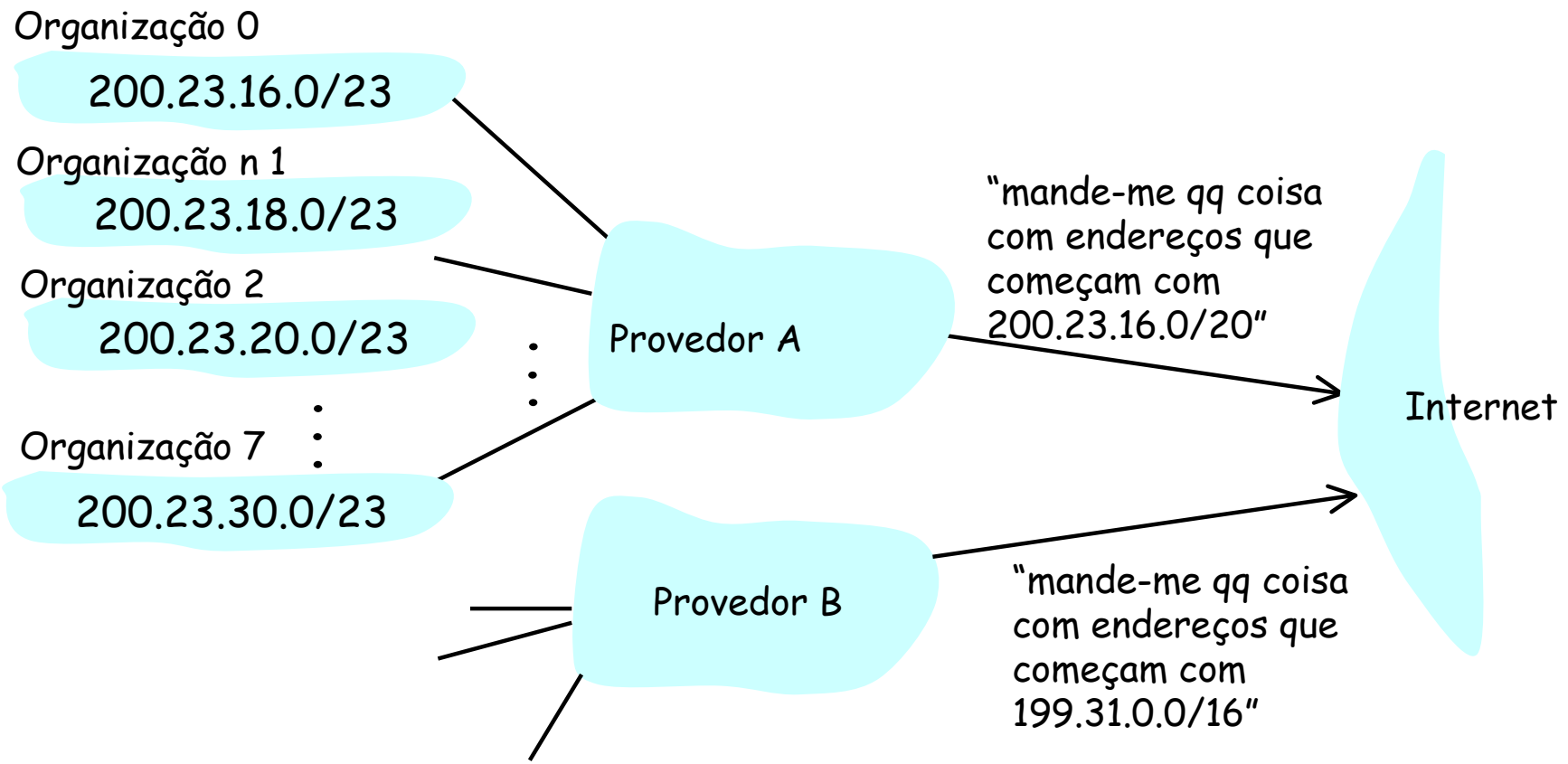


DHCP: cenário cliente-servidor



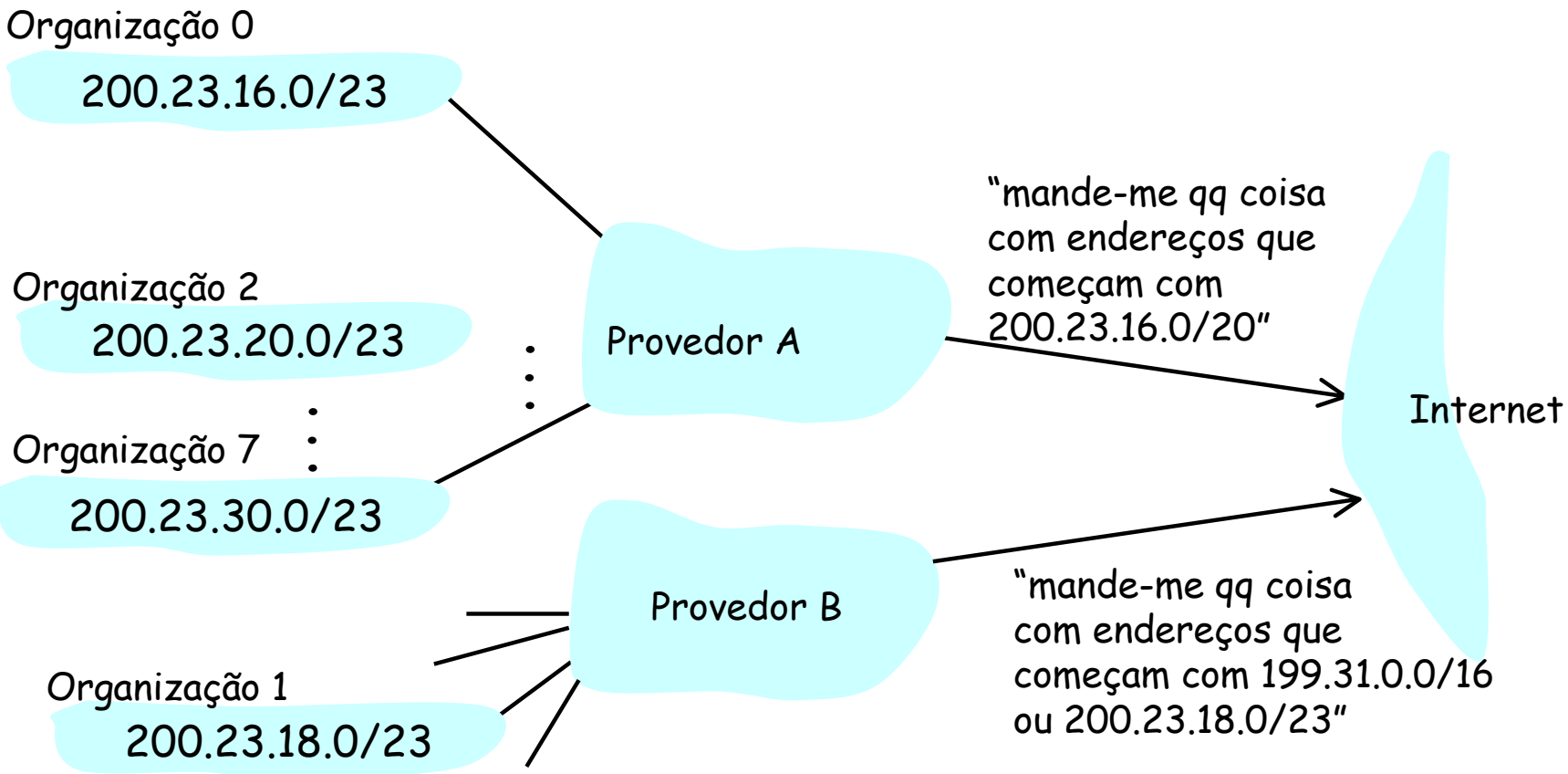
Endereçamento hierárquico: agregação de rotas

Endereçamento hierárquico permite anunciar eficientemente informação sobre rotas:



Endereçamento hierárquico: rotas mais específicas

Provedor B tem uma rota mais específica para a Organização 1



Endereçamento IP: a última palavra...

P: Como um provedor IP consegue um bloco de endereços?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- aloca endereços
- gerencia DNS
- aloca nomes de domínio, resolve disputas

(no Brasil, estas funções foram delegadas ao Registro nacional, sediado na FAPESP (SP), e comandado pelo Comitê Gestor Internet BR)

Enviando um datagrama da origem ao destino

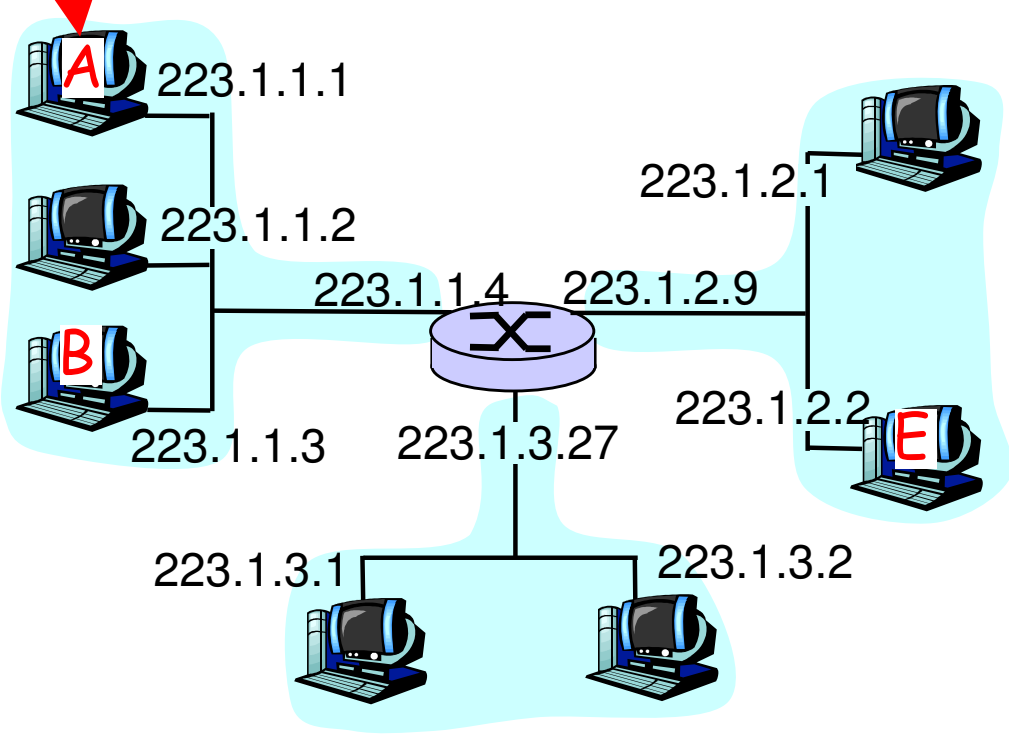
datagrama IP:

aampos misc	end. IP origem	end. IP dest	dados

- datagrama permanece inalterado, enquanto passa da origem ao destino
- campos de endereços de interesse aqui

tabela de rotas em *A*

rede dest.	próx. rot.	Nenlaces
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



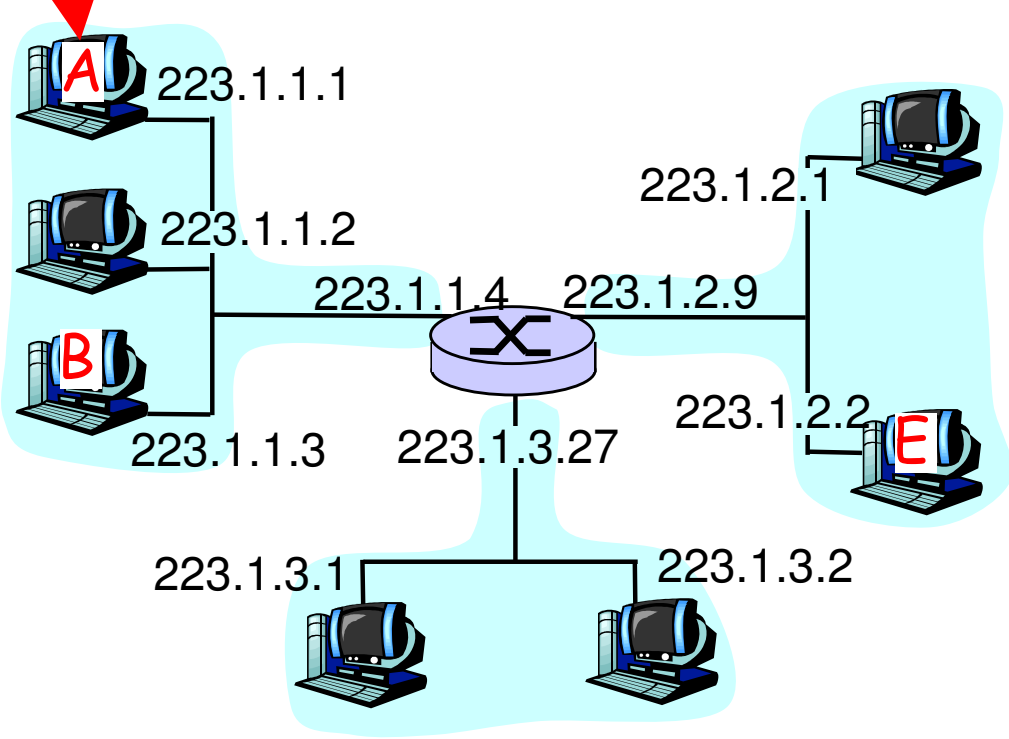
Enviando um datagrama da origem ao destino

campos div.	223.1.1.1	223.1.1.3	dados

Supomos um datagrama IP originando em A, e endereçado a B:

- procura endereço de rede de B
- descobre que B é da mesma rede que A
- camada de enlace remeterá datagrama diretamente para B num quadro da camada de enlace
  - B e A estão diretamente ligados

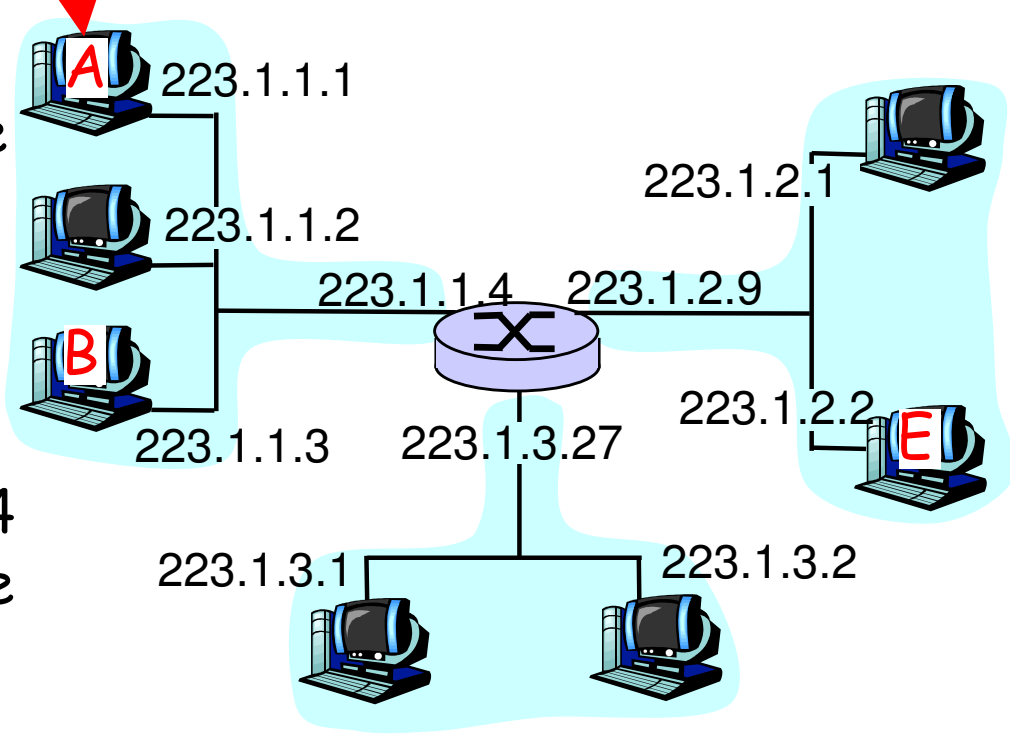
rede dest.	próx. rot.	Nenlaces
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



Enviando um datagrama da origem ao destino

campos div.	223.1.1.1	223.1.2.2	dados

rede dest.	próx. rot.	Nenlaces
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



OrigemA, destino E:

- procura endereço de rede de E
- E numa rede *diferente*
  - A, E não ligados diretamente
- tabela de rotas: próximo roteador na rota para E é 223.1.1.4
- camada de enlace envia datagrama ao roteador 223.1.1.4 num quadro da camada de enlace
- datagrama chega a 223.1.1.4
- continua...

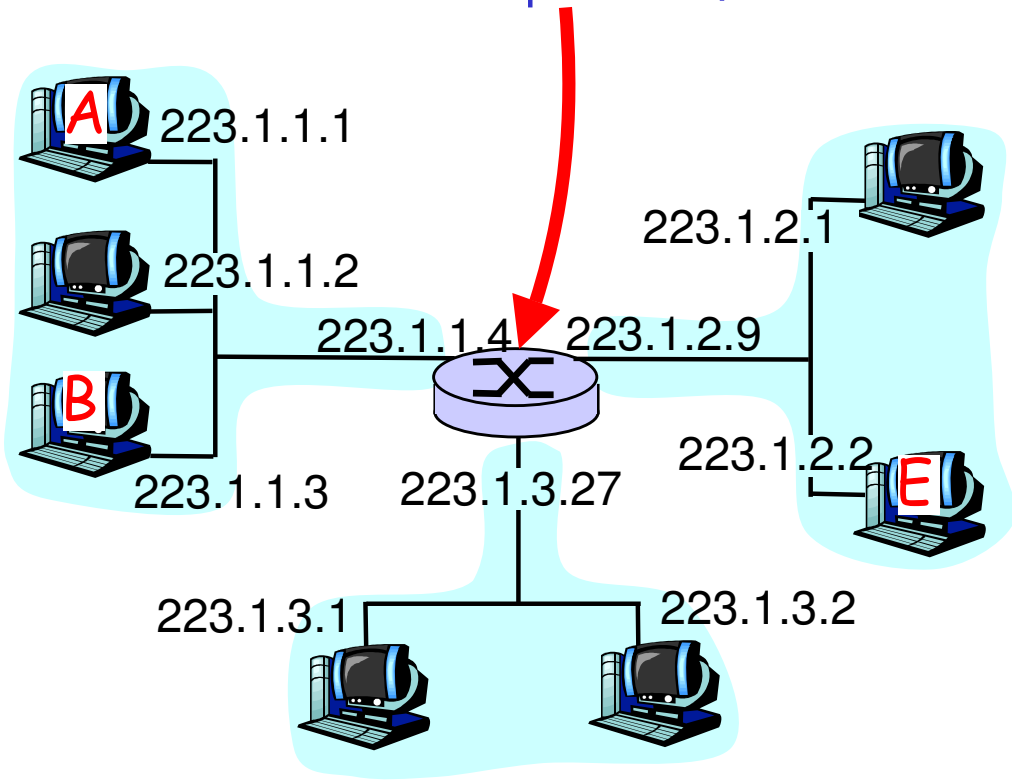
Enviando um datagrama da origem ao destino

campos div.	223.1.1.1	223.1.2.2	dados

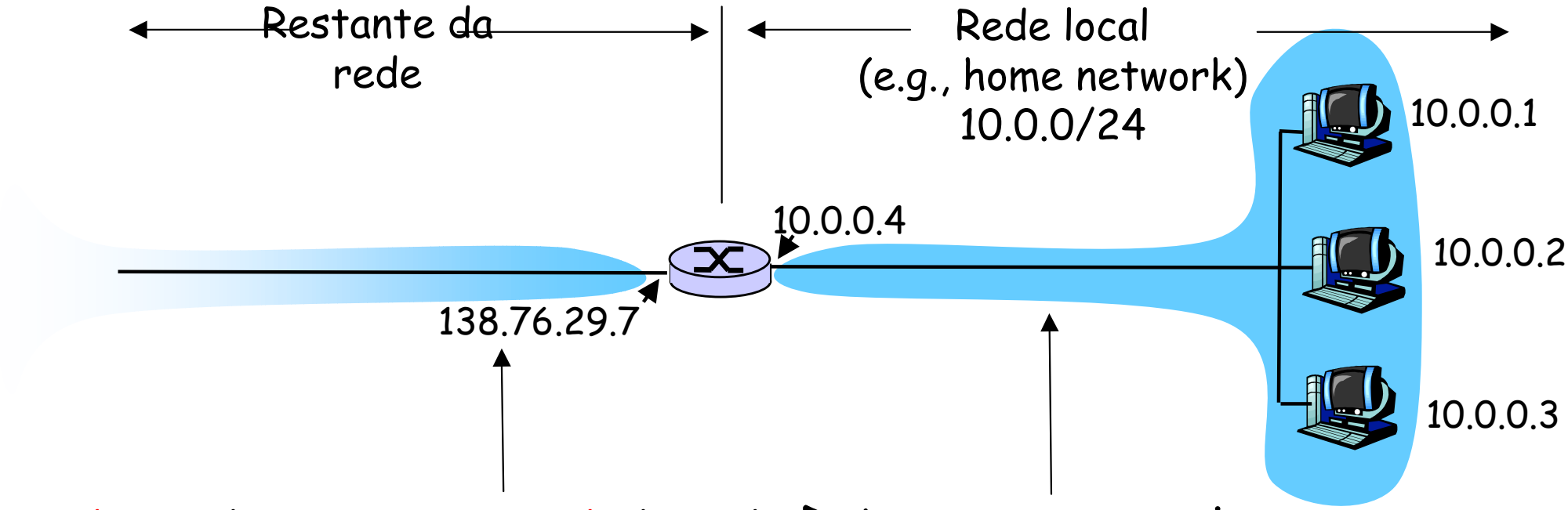
Chegando a 223.1.1.4, destinado a 223.1.2.2

- procura endereço de rede de E
- E fica na mesma rede que a interface 223.1.2.9 do roteador
  - roteador, E estão diretamente ligados
- camada de enlace envia datagrama p/ 223.1.2.2 dentro de quadro de camada de enlace via interface 223.1.2.9
- datagrama chega a 223.1.2.2!!! (oba!)

rede dest.	próx. rot.	Nenl.	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27



NAT: Network Address Translation



Todos os datagramas *saindo* da rede local tem o mesmo endereço NAT IP: 138.76.29.7, diferentes números de portas fontes

Datagramas com origem ou destino nesta rede tem endereço 10.0.0/24 para fonte, e de destino o usual

NAT: Network Address Translation

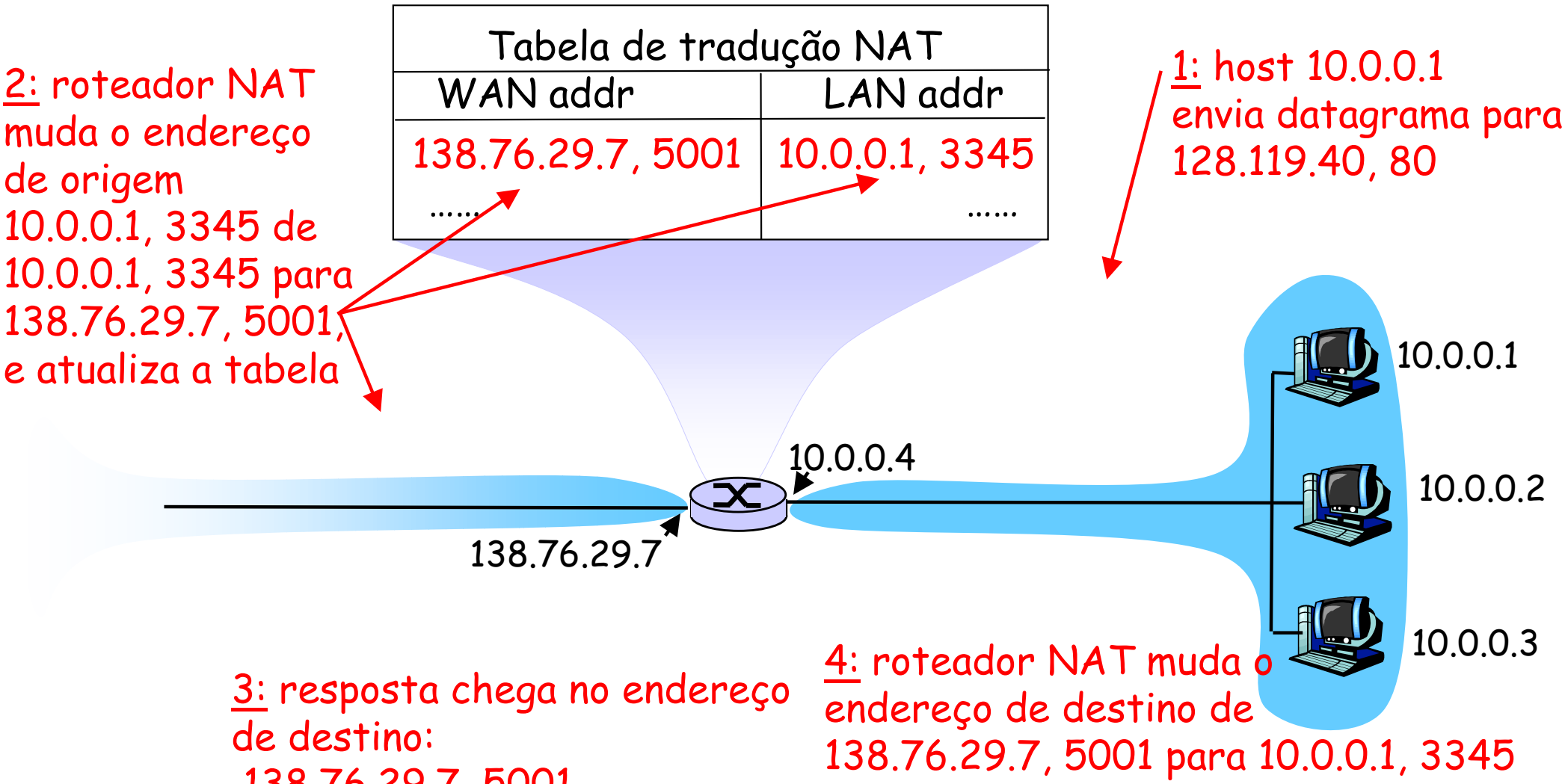
- Motivação: rede local usa apenas um endereço IP:
  - Não há necessidade de alocar faixas de endereços de um ISP
    - apenas um endereço IP é usado para todos os dispositivos
  - Permite mudar o endereço dos dispositivos internos sem necessitar notificar o mundo externo;
  - Permite a mudança de ISPs sem necessitar mudar os endereços dos dispositivos internos da rede local
  - Dispositivos internos a rede, não são visíveis nem endereçaveis pelo mundo externo (melhora segurança);

NAT: Network Address Translation

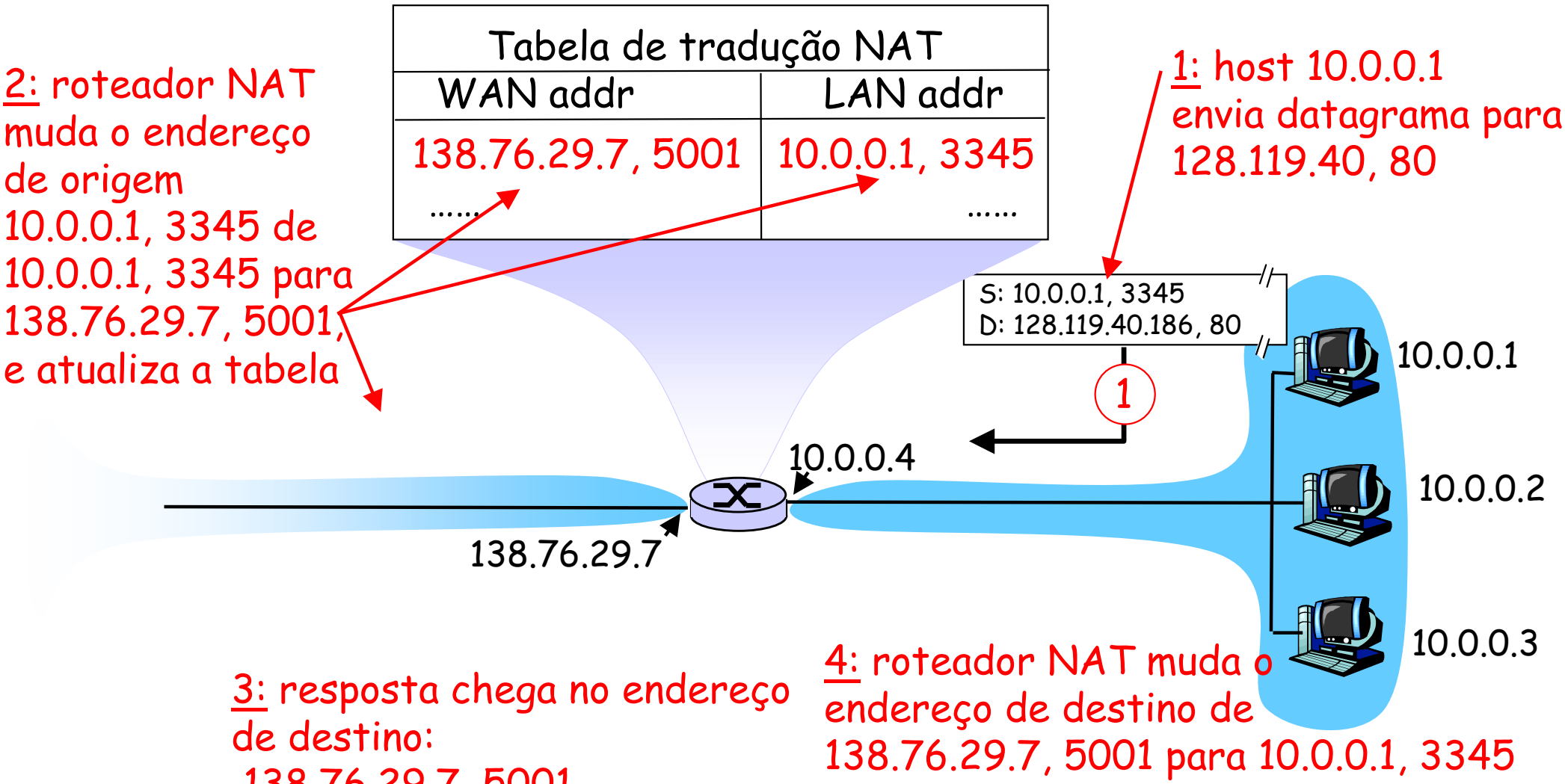
Implementação: roteador NAT deve;

- *Datagramas que saem: trocar* (endereço IP fonte, porta #) de cada datagrama de saída para (endereço NAT IP, nova porta #)
  - . . . clientes/servidores remotos irão responder usando (endereço NAT IP, nova porta #) como endereço destino.
- *guardar (na tabela de tradução de endereços NAT): os pares de tradução de endereços* (endereço IP fonte, porta #) para (endereços NAT IP, nova porta #)
- *Datagramas qeu chegam: trocar* (endereço NAT IP, nova porta #) no campo de destino de cada datagrama que chega com o correspondente (endereço IP fonte, porta #) armazenado na tabela NAT

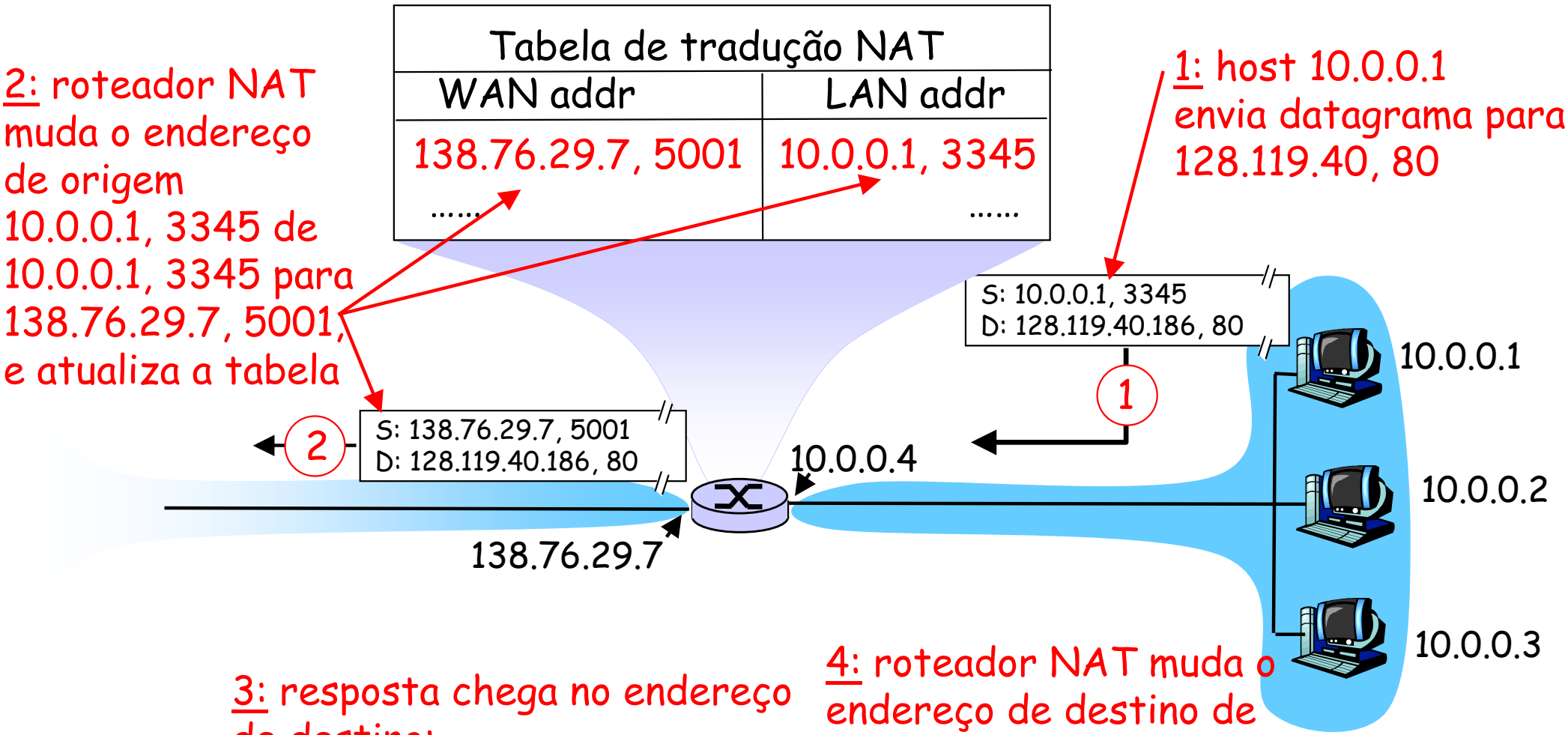
NAT: Network Address Translation



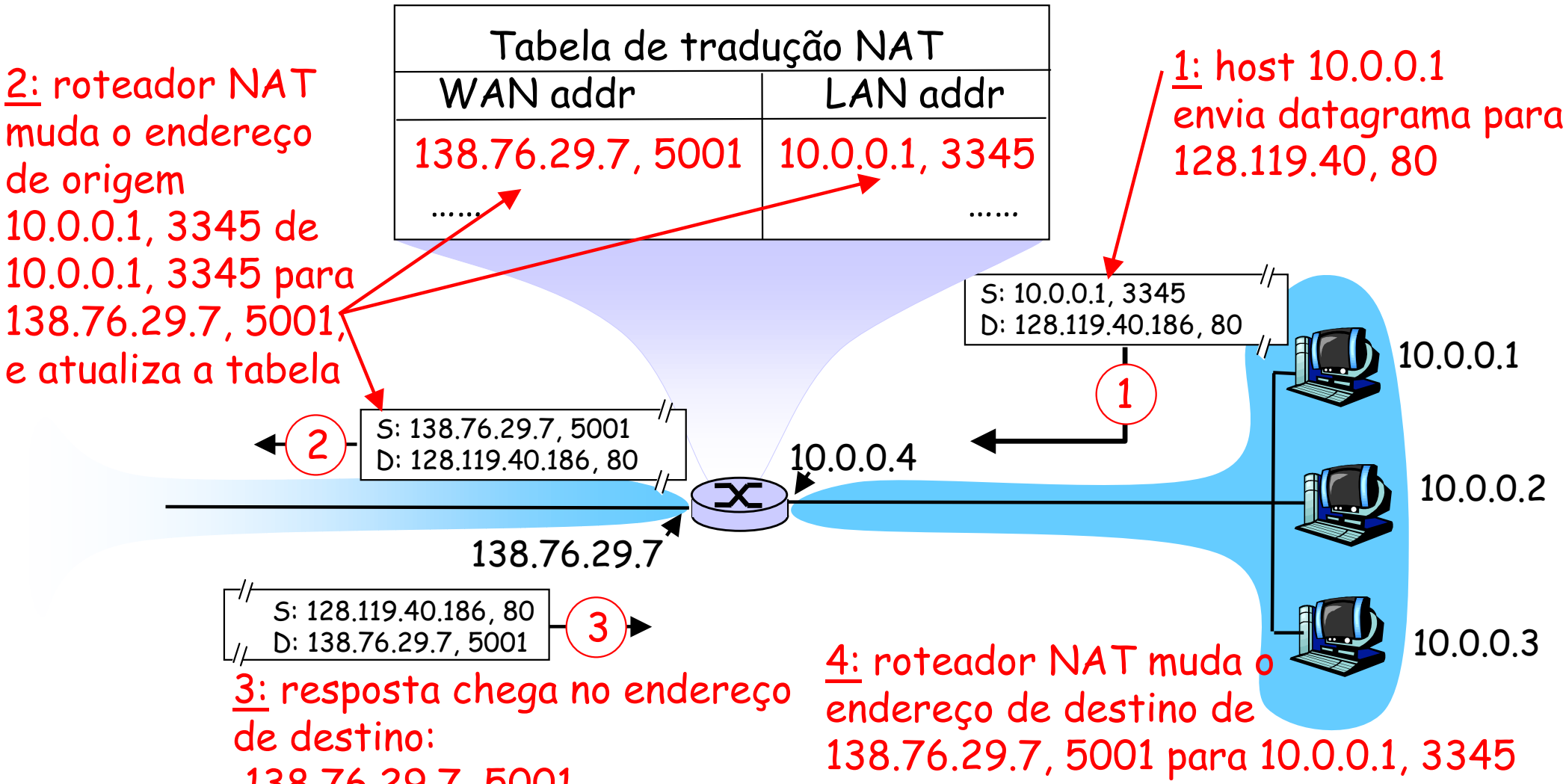
NAT: Network Address Translation



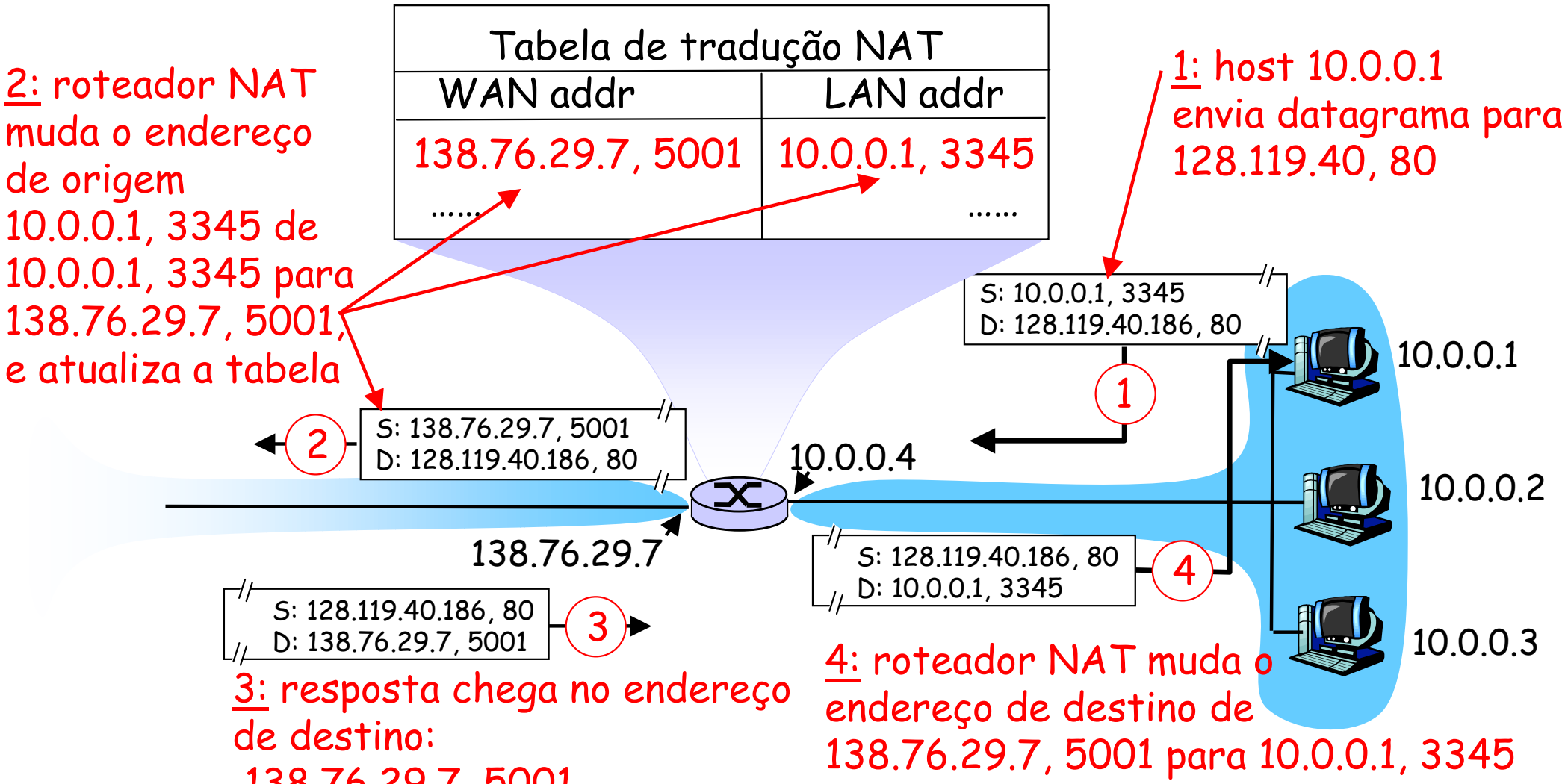
NAT: Network Address Translation



NAT: Network Address Translation



NAT: Network Address Translation



NAT: Network Address Translation

- Campo de porta de 16-bit :
  - 60,000 conexões simultâneas com um único endereço de rede;
- NAT é controverso:
  - Roteadores devem fazer processamentos até no máximo a camada 3;
  - Viola o "conceito fim-a-fim"
    - A possibilidade de suporte a NAT deve ser levado em consideração pelos desenvolvedores de aplicações;
  - O problema de diminuição do número de endereços deveria ser tratada por IPv6;

ICMP: Internet Control Message Protocol

- usado por estações, roteadores para comunicar informação s/ camada de rede
  - relatar erros: estação, rede, porta, protocolo inalcançáveis
  - pedido/resposta de eco (usado por ping)
- camada de rede "acima de" IP:
  - msgs ICMP transportadas em datagramas IP
- mensagem ICMP: tipo, código mais primeiros 8 bytes do datagrama IP causando erro

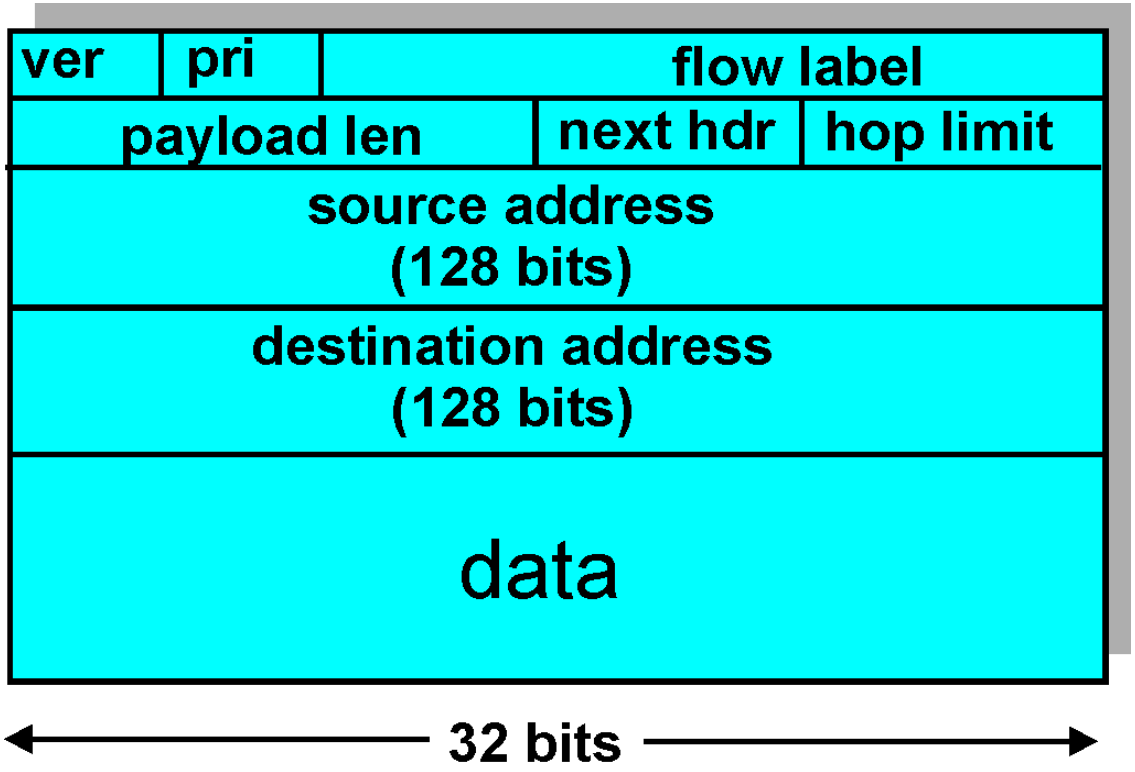
Tipo	Código	descrição
0	0	resposta de eco (ping)
3	0	rede dest. inalcançável
3	1	estação dest inalcançável
3	2	protocolo dest inalcançável
3	3	porta dest inalcançável
3	6	rede dest desconhecida
3	7	estação dest desconhecida
4	0	abaixar fonte (controle de congestionamento - ñ usado)
8	0	pedido eco (ping)
9	0	anúncio de rota
10	0	descobrir roteador
11	0	TTL (sobrevida) expirada
12	0	erro de cabeçalho IP

IPv6

- Motivação inicial: espaço de endereços de 32-bits completamente alocado até 2008.
- Motivação adicional :
  - formato do cabeçalho facilita acelerar processamento/re-encaminhamento
  - mudanças no cabeçalho para facilitar QoS
  - novo endereço "anycast": rota para o "melhor" de vários servidores replicados
- formato do datagrama IPv6:
  - cabeçalho de tamanho fixo de 40 bytes
  - não admite fragmentação

Cabeçalho IPv6

Prioridade: identifica prioridade entre datagramas no fluxo
Rótulo do Fluxo: identifica datagramas no mesmo "fluxo" (conceito de "fluxo" mal definido).
Próximo cabeçalho: identifica protocolo da camada superior para os dados



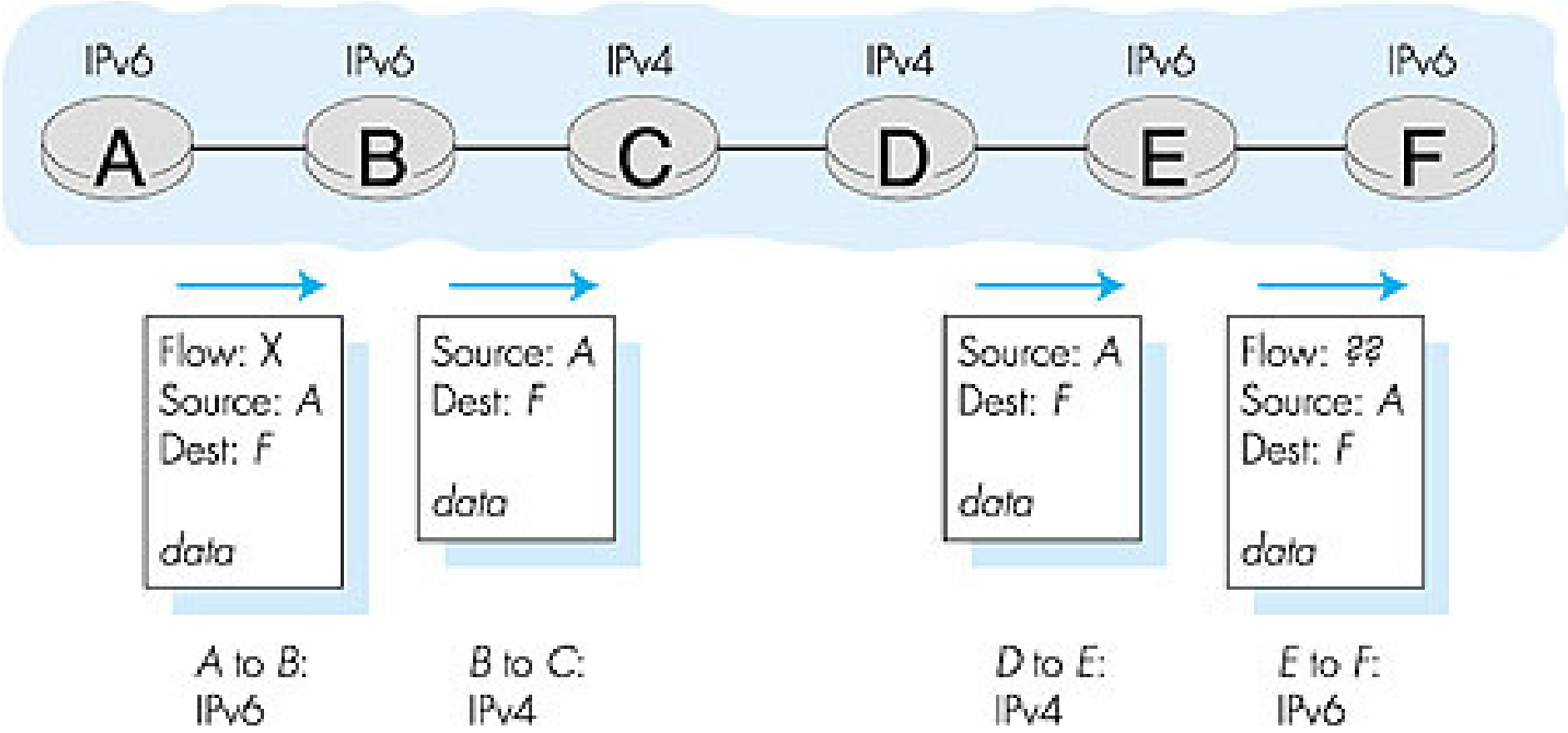
Outras mudanças de IPv4

- *Checksum*: removido completamente para reduzir tempo de processamento a cada roteador
- *Opções:* permitidas, porém fora do cabeçalho, indicadas pelo campo "Próximo Cabeçalho"
- *ICMPv6:* versão nova de ICMP
  - tipos adicionais de mensagens, p.ex. "Pacote Muito Grande"
  - funções de gerenciamento de grupo multiponto

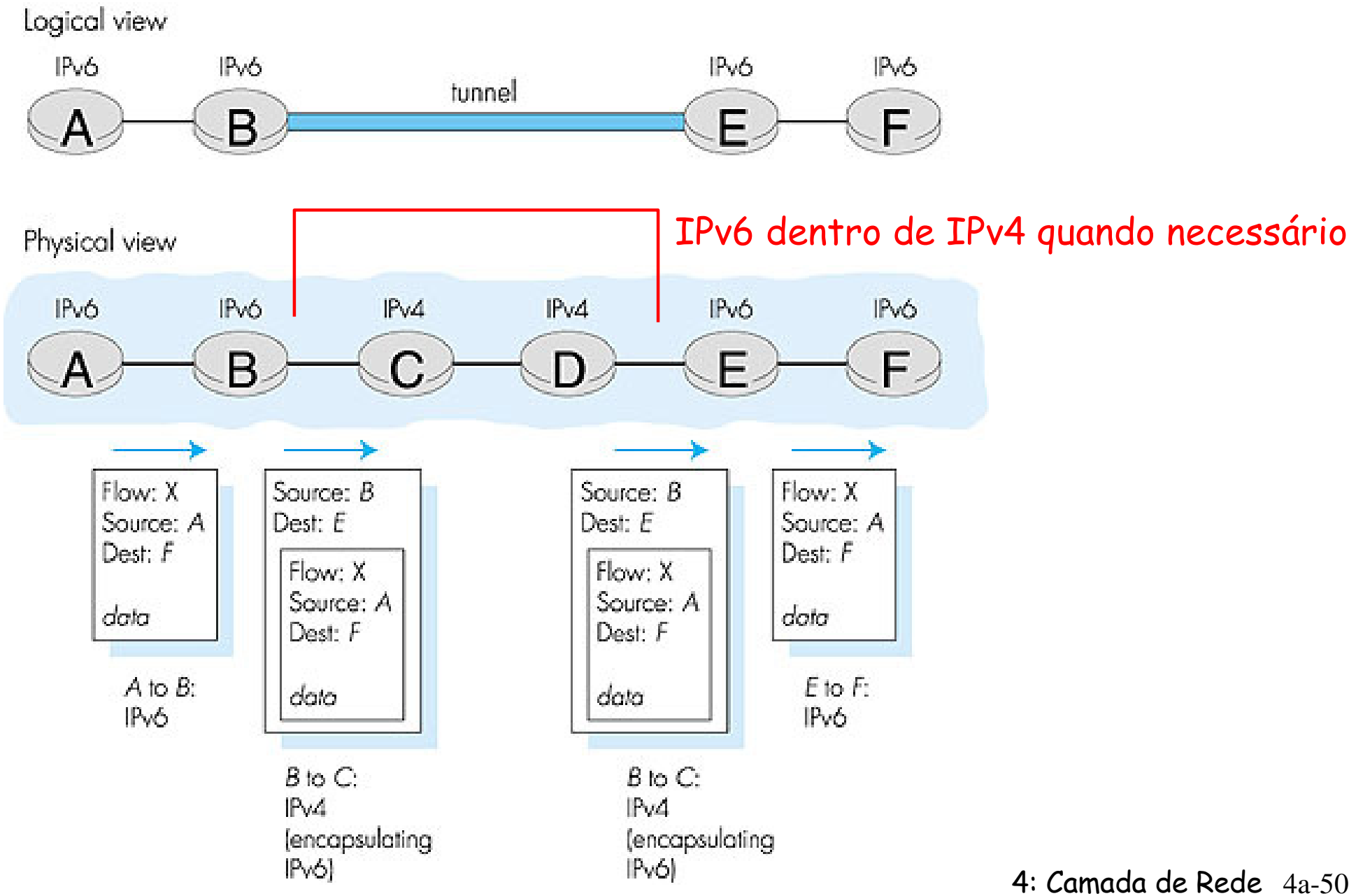
Transição de IPv4 para IPv6

- Não todos roteadores podem ser atualizados simultaneamente
  - "dias de mudança geral" inviáveis
  - Como a rede pode funcionar com uma mistura de roteadores IPv4 e IPv6?
- Duas abordagens propostas:
  - *Pilhas Duais*: alguns roteadores com duas pilhas (v6, v4) podem "traduzir" entre formatos
  - *Tunelamento:* datagramas IPv6 carregados em datagramas IPv4 entre roteadores IPv4

Abordagem de Pilhas Duais



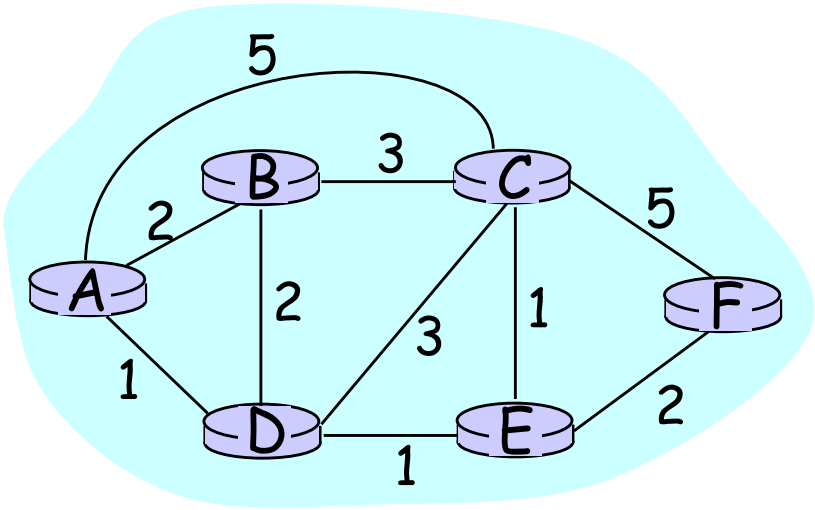
Tunelamento



Roteamento

protocolo de roteamento

meta: determinar caminho (seqüência de roteadores) "bom" pela rede da origem ao destino



Abstração de grafo para algoritmos de roteamento:

- nós do grafo são roteadores
- arestas do grafo são os enlaces físicos
  - custo do enlace: retardo, financeiro, ou nível de congestionamento

- caminho "bom":
  - tipicamente significa caminho de menor custo
  - outras definições são possíveis

Classificação de Algoritmos de Roteamento

Informação global ou descentralizada?

Global:

- todos roteadores têm info. completa de topologia, custos dos enlaces
- algoritmos "estado de enlaces"

Decentralizada:

- roteador conhece vizinhos diretos e custos até eles
- processo iterativo de cálculo, troca de info. com vizinhos
- algoritmos "vetor de distâncias"

Estático ou dinâmico?

Estático:

- rotas mudam lentamente com o tempo

Dinâmico:

- rotas mudam mais rapidamente
  - atualização periódica
  - em resposta a mudanças nos custos dos enlaces

Um algoritmo de roteamento de "estado de enlaces" (EE)

Algoritmo de Dijkstra

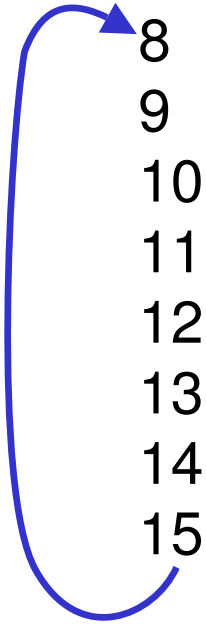
- topologia da rede, custos dos enlaces conhecidos por todos os nós
  - realizado através de "difusão do estado dos enlaces"
  - todos os nós têm mesma info.
- calcula caminhos de menor custo de um nó ("origem") para todos os demais
  - gera tabela de rotas para aquele nó
- iterativo: depois de k iterações, sabemos menor custo p/ k destinos

Notação:

- c(i,j): custo do enlace do nó i ao nó j. custo é infinito se não forem vizinhos diretos
- D(V): valor corrente do custo do caminho da origem ao destino V
- p(V): nó antecessor no caminho da origem ao nó V, imediatamente antes de V
- N: conjunto de nós cujo caminho de menor custo já foi determinado

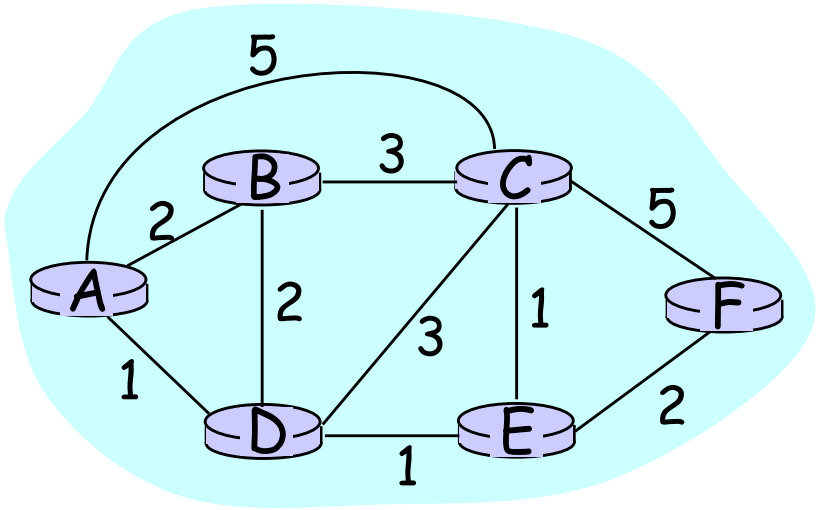
O algoritmo de Dijkstra

```
1  Inicialização:
2    N = {A}
3    para todos os nós V
4      se V for adjacente ao nó A
5        então D(V) = c(A,V)
6        senão D(V) = infinito
7
8  Repete
9     determina W não contido em N tal que D(W) é o mínimo
10    adiciona W ao conjunto N
11    atualiza D(V) para todo V adjacente ao nó W e ainda não em N:
12       D(V) = min( D(V), D(W) + c(W,V) )
13    /* novo custo ao nó V ou é o custo velho a V ou o custo do
14     menor caminho ao nó W, mais o custo de W a V */
15 até que todos nós estejam em N
```



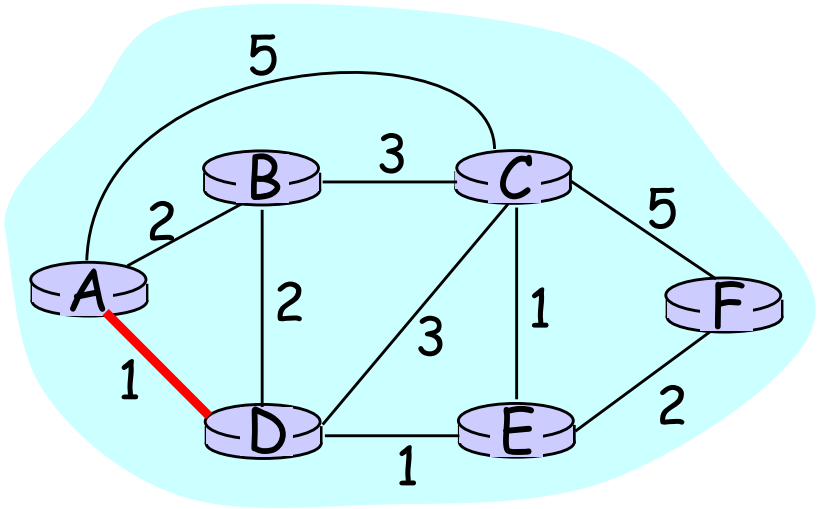
Algoritmo de Dijkstra: exemplo

Passo	N inicial	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
→ 0	A	2,A	5,A	1,A	infinito	infinito
→ 1	AD	2,A	4,D		2,D	infinito
→ 2	ADE	2,A	3,E			4,E
→ 3	ADEB		3,E			4,E
→ 4	ADEBC					4,E
5	ADEBCF					



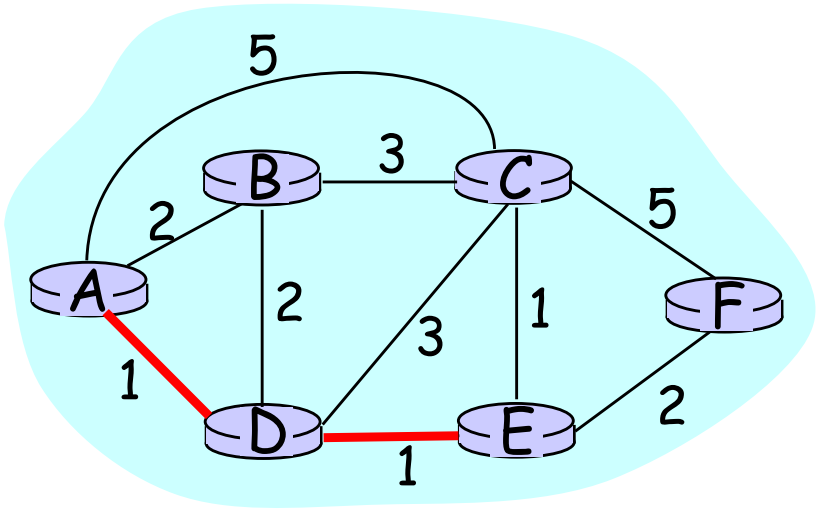
Algoritmo de Dijkstra: exemplo

Passo	N inicial	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
→ 0	A	2,A	5,A	1,A	infinito	infinito
→ 1	AD	2,A	4,D		2,D	infinito
→ 2	ADE	2,A	3,E			4,E
→ 3	ADEB		3,E			4,E
→ 4	ADEBC					4,E
5	ADEBCF					



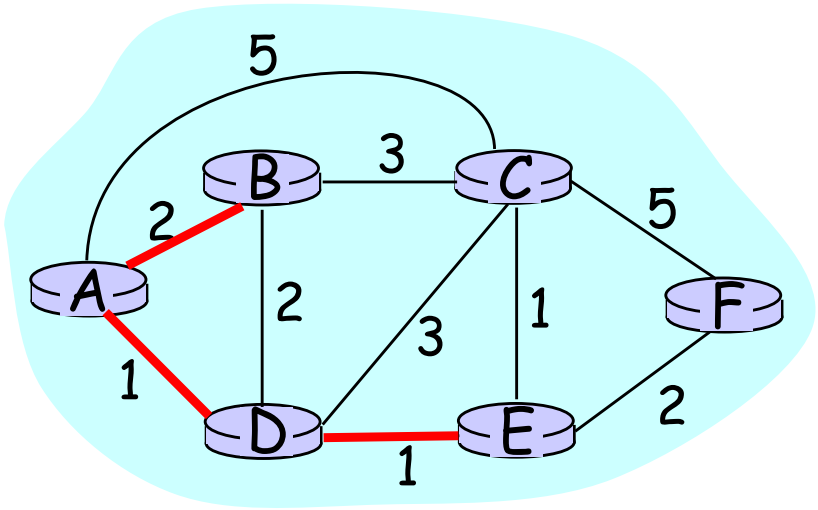
Algoritmo de Dijkstra: exemplo

Passo	N inicial	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinito	infinito
1	AD	2,A	4,D		2,D	infinito
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



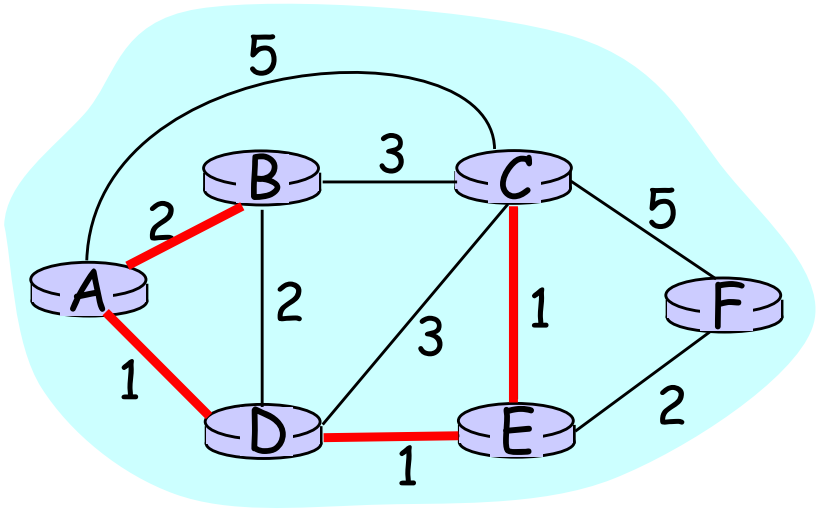
Algoritmo de Dijkstra: exemplo

Passo	N inicial	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinito	infinito
1	AD	2,A	4,D		2,D	infinito
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



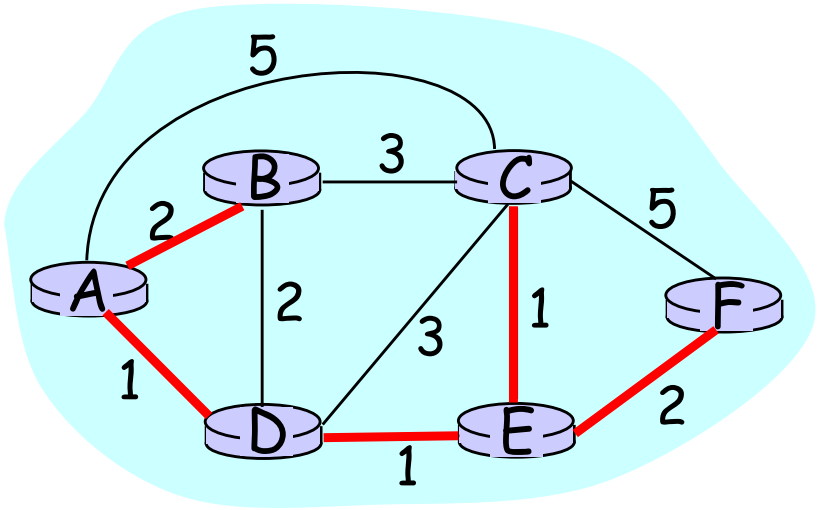
Algoritmo de Dijkstra: exemplo

Passo	N inicial	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	2,A	5,A	1,A	infinito	infinito
1	AD	2,A	4,D		2,D	infinito
2	ADE	2,A	3,E			4,E
3	ADEB		3,E			4,E
4	ADEBC					4,E
5	ADEBCF					



Algoritmo de Dijkstra: exemplo

Passo	N inicial	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
→ 0	A	2,A	5,A	1,A	infinito	infinito
→ 1	AD	2,A	4,D		2,D	infinito
→ 2	ADE	2,A	3,E			4,E
→ 3	ADEB		3,E			4,E
→ 4	ADEBC					4,E
5	ADEBCF					



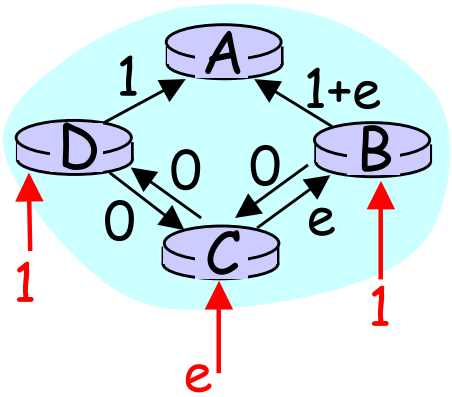
Algoritmo de Dijkstra, discussão

Complexidade algoritmica: n nós

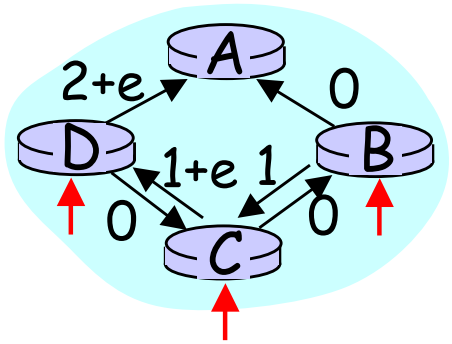
- a cada iteração: precisa checar todos nós, W, não em N
- n*(n+1)/2 comparações => O(n**2)
- implementações mais eficientes possíveis: O(nlogn)

Oscilações possíveis:

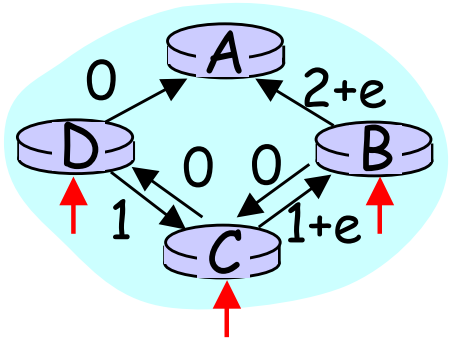
- p.ex., custo do enlace = carga do tráfego carregado



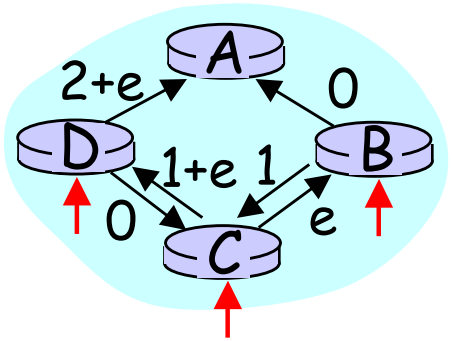
inicialmente



... recalcula rotas



... recalcula



... recalcula

Um algoritmo de roteamento de "vetor de distâncias" (VD)

iterativo:

- continua até que não haja mais troca de info. entre nós
- *se auto-termina*: não há "sinal" para parar

assíncrono:

- os nós *não* precisam trocar info./iterar de forma sincronizada!

distribuído:

- cada nó comunica *apenas* com seus vizinhos diretos

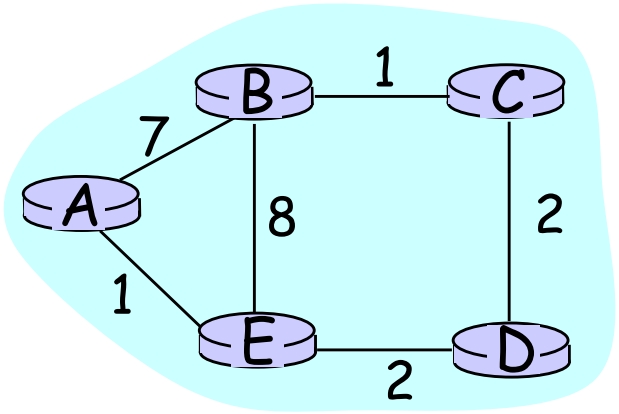
Estrutura de dados: Tabela de Distâncias

- cada nós possui sua própria TD
- 1 linha para cada destino possível
- 1 coluna para cada vizinho direto
- exemplo: no nó X, para destino Y através do vizinho Z:

$$D^{X}(Y,Z) = \text{distância } \textit{de } X \textit{ para } Y, \textit{ usando } Z \text{ como caminho}$$

$$= c(X,Z) + \min_{w}\{D^{Z}(Y,w)\}$$

Tabela de Distâncias: exemplo



$$D^{E}(C,D) = c(E,D) + \min_{w}\{D^{D}(C,w)\}$$
$$= 2+2 = 4$$

$$D^{E}(A,D) = c(E,D) + \min_{w}\{D^{D}(A,w)\}$$
$$= 2+3 = 5$$ ciclo!

$$D^{E}(A,B) = c(E,B) + \min_{w}\{D^{B}(A,w)\}$$
$$= 8+6 = 14$$ ciclo!

$D^{E}()$ — custo ao destino via

destino	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

Tabela de distâncias gera tabela de rotas

$D^E()$

destino	custo ao destino via A	via B	via D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destino	enlace de saída a usar, custo
A	A,1
B	D,5
C	D,4
D	D,4

Tabela de distâncias → Tabela de rotas

Roteamento vetor de distâncias: sumário

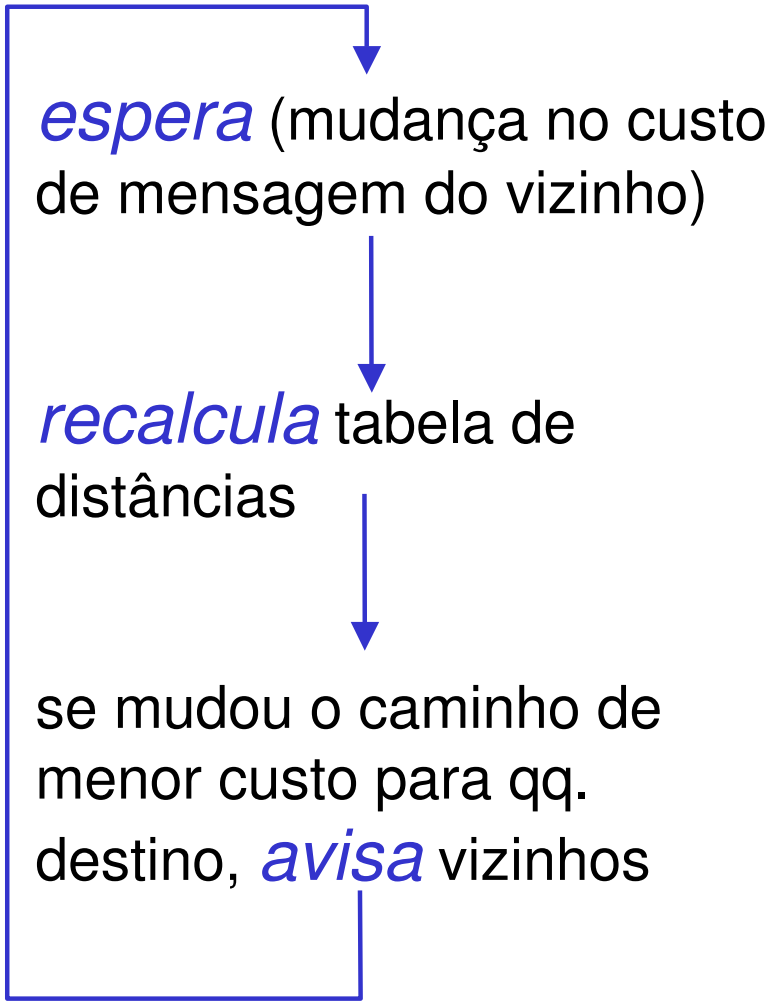
Iterativo, assíncrono: cada iteração local causada por:

- mudança do custo do enlace local
- mensagem do vizinho: mudança de caminho de menor custo para algum destino

Distribuído:

- cada nó avisa a seus vizinhos *apenas* quando muda seu caminho de menor custo para qualquer destino
  - os vizinhos então avisam a seus vizinhos, se for necessário

Cada nó:



Algoritmo Vetor de Distâncias:

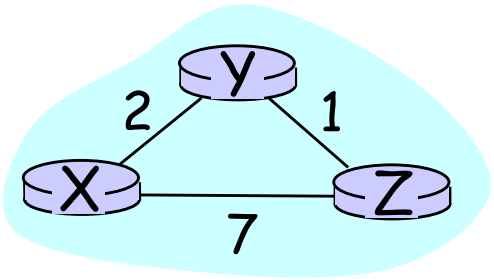
Em todos nós, X:

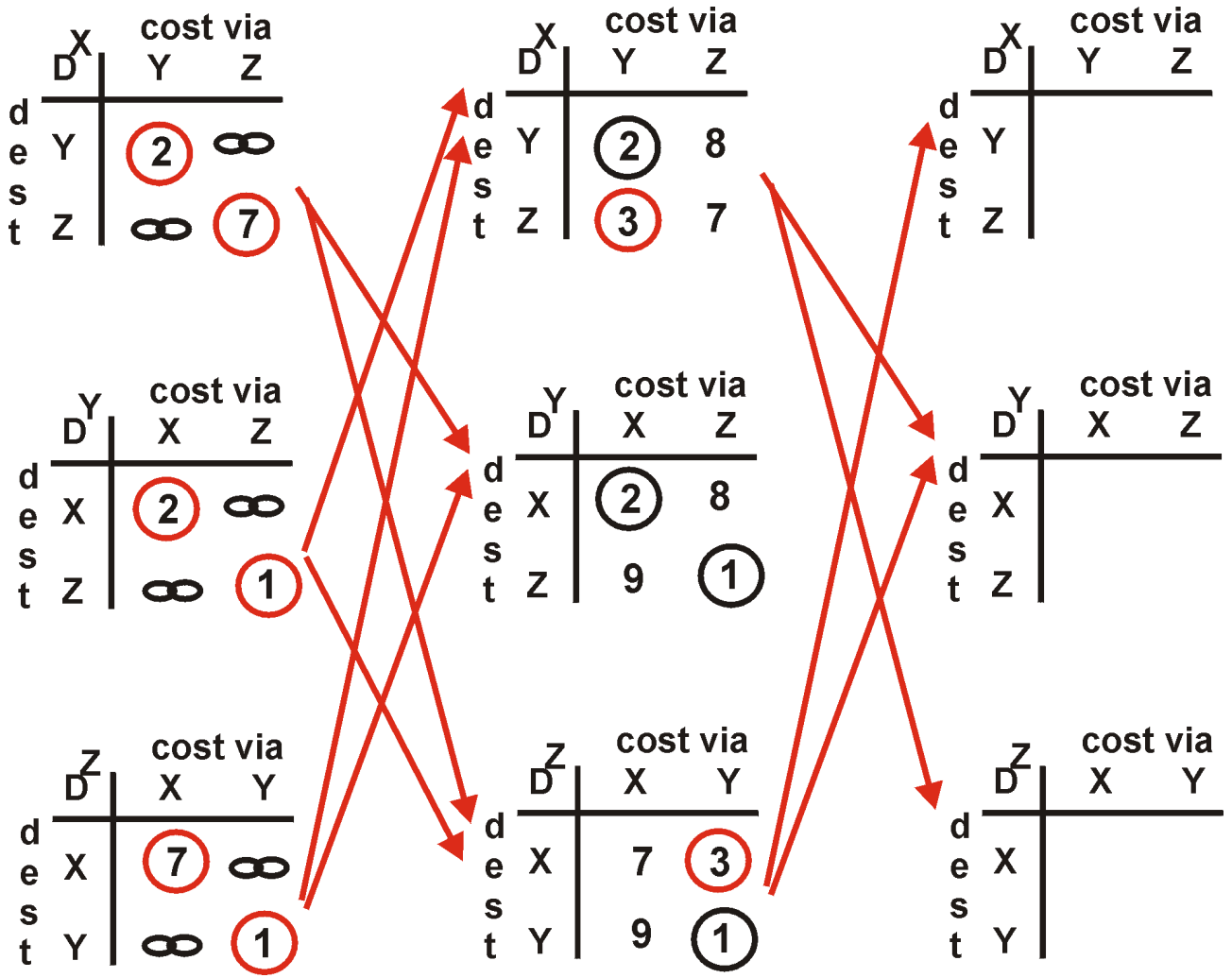
1 Inicialização:
2 para todos nós adjacentes V:
3 $D^X(*,V)$ = infinito /* o operador * significa "para todas linhas" */
4 $D^X(V,V)$ = c(X,V)
5 para todos destinos, Y
6 envia $\min_w D^X(Y,W)$ para cada vizinho /* W indica vizinhos de X */

Algoritmo Vetor de Distâncias (cont.):

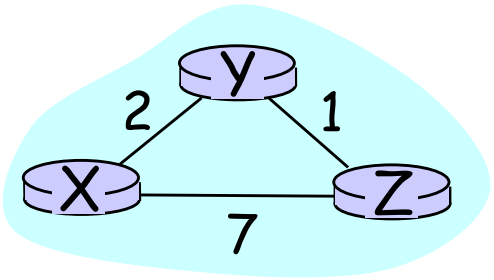
8 **repete**
9 **espera** (até observar mudança de custo do enlace ao vizinho V,
10 ou até receber atualização do vizinho V)
11
12 **se** (c(X,V) muda por d unidades)
13 /* altera custo para todos destinos através do vizinho V por d */
14 /* note: d pode ser positivo ou negativo */
15 para todos destinos Y: $D^X(Y,V) = D^X(Y,V) + d$
16
17 **senão, se** (atualização recebido de V para destino Y)
18 /* mudou o menor caminho de V para algum Y */
19 /* V enviou um novo valor para seu $\text{mín}_w D^V(Y,w)$ */
20 /* chamamos este novo valor de "val_novo" */
21 para apenas o destino Y: $D^X(Y,V) = c(X,V) + \text{val\_novo}$
22
23 **se** temos um novo $\text{mín}_w D^X(Y,W)$ para qq destino Y
24 envia novo valor de $\text{mín}_w D^X(Y,W)$ para todos vizinhos
25
26 **para sempre**

Algoritmo Vetor de Distâncias: exemplo





Algoritmo Vetor de Distâncias: exemplo



D^X

dest	via Y	via Z
Y	2	∞
Z	∞	7

D^Y

dest	via X	via Z
X	2	∞
Z	∞	1

D^Z

dest	via X	via Y
X	7	∞
Y	∞	1

D^X

dest	via Y	via Z
Y	2	8
Z	3	7

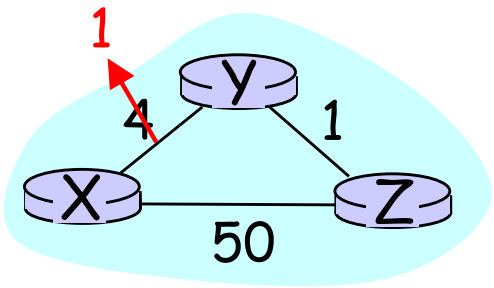
$$D^X(Y,Z) = c(X,Z) + \min_w\{D^Z(Y,w)\}$$
$$= 7+1 = 8$$

$$D^X(Z,Y) = c(X,Y) + \min_w\{D^Y(Z,w)\}$$
$$= 2+1 = 3$$

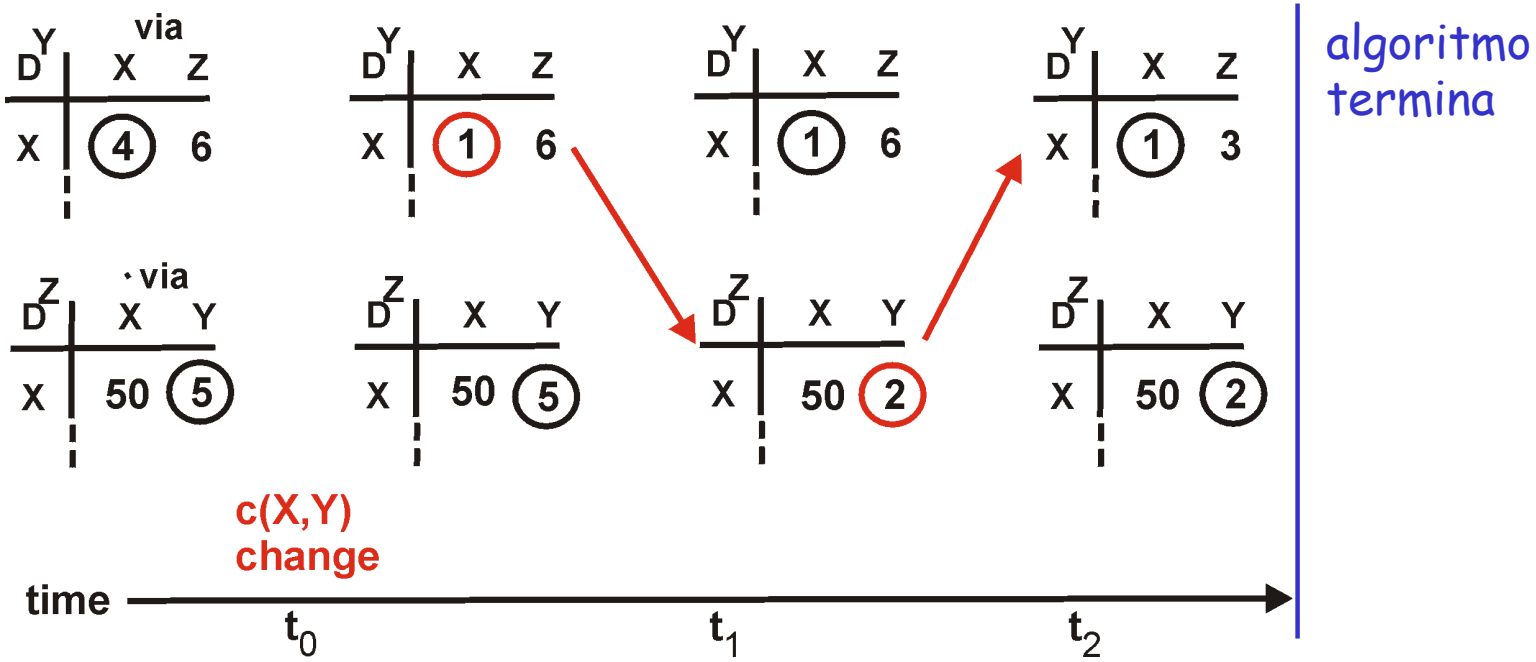
Vetor de Distâncias: mudança no custo dos enlaces

Mudança no custo dos enlaces:

- nó detecta mudança no custo do enlace local
- atualiza tabela de distâncias (linha 15)
- se mudou custo do menor caminho, avisa aos vizinhos (linhas 23,24)



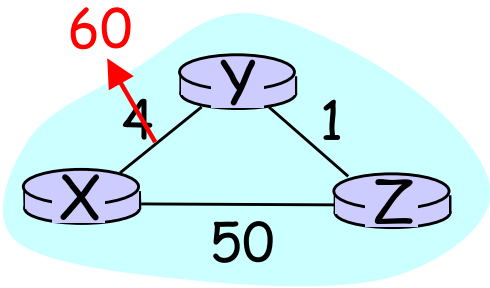
"boas notícias chegam logo"

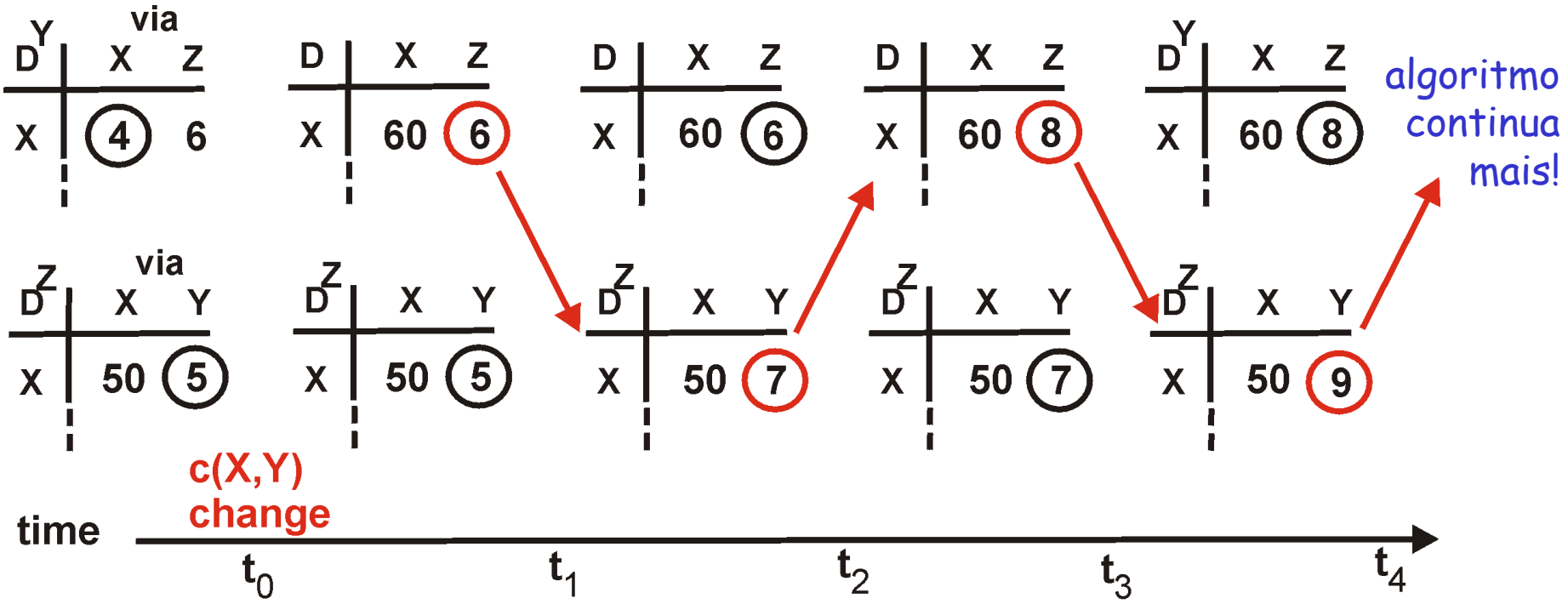


Vetor de Distâncias: mudança no custo dos enlaces

Mudança no custo dos enlaces:

- boas notícias chegam logo
- más notícias demoram para chegar - problema da "contagem ao infinito"!

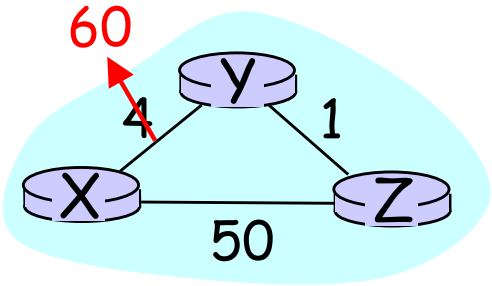


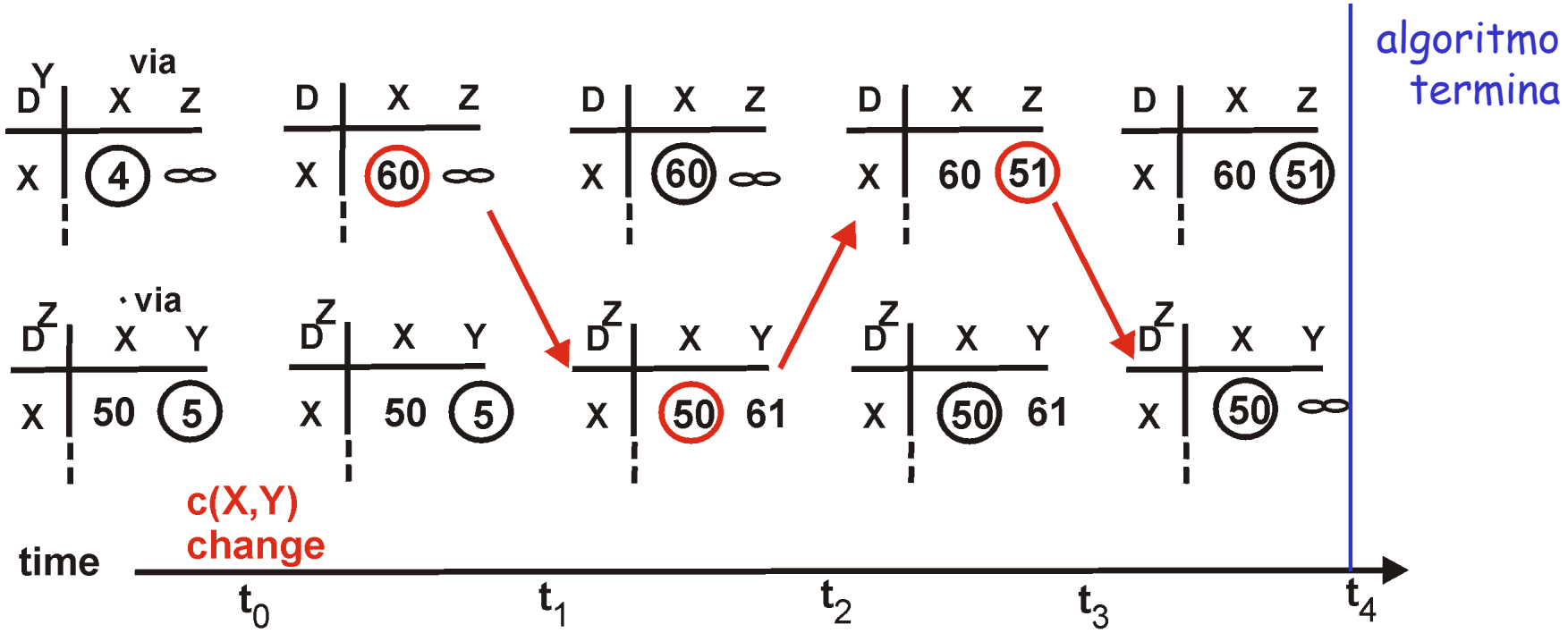


Vetor de Distâncias: reverso envenenado

Se Z roteia via Y p/ chegar a X :

- Z informa p/ Y que sua distância p/ X é infinita (p/ que Y não roteie p/ X via Z)
- P: será que isto resolve completamente o problema da contagem ao infinito?





Comparação dos algoritmos EE e VD

Complexidade de mensagens

- EE: com n nós, E enlaces, O(nE) mensagens enviadas
- VD: trocar mensagens apenas entre vizinhos
  - varia o tempo de convergência

Rapidez de Convergência

- EE: algoritmo O(n**2) requer O(nE) mensagens
  - podem ocorrer oscilações
- VD: varia tempo para convergir
  - podem ocorrer rotas cíclicas
  - problema de contagem ao infinito

Robustez: o que acontece se houver falha do roteador?

EE:

- nó pode anunciar valores incorretos de custo de *enlace*
- cada nó calcula sua *própria* tabela

VD:

- um nó VD pode anunciar um custo de *caminho* incorreto
- a tabela de cada nó é usada pelos outros nós
  - erros se propagam pela rede

Roteamento Hierárquico

Neste estudo de roteamento fizemos uma idealização:

- todos os roteadores idênticos
- rede "não hierarquizada" ("flat")
- ... *não* é verdade, na prática

escala: com > 100 milhões de destinos:

- impossível guardar todos destinos na tabela de rotas!
- troca de tabelas de rotas afogaria os enlaces!

autonomia administrativa

- internet = rede de redes
- cada administrador de rede pode querer controlar roteamento em sua própria rede

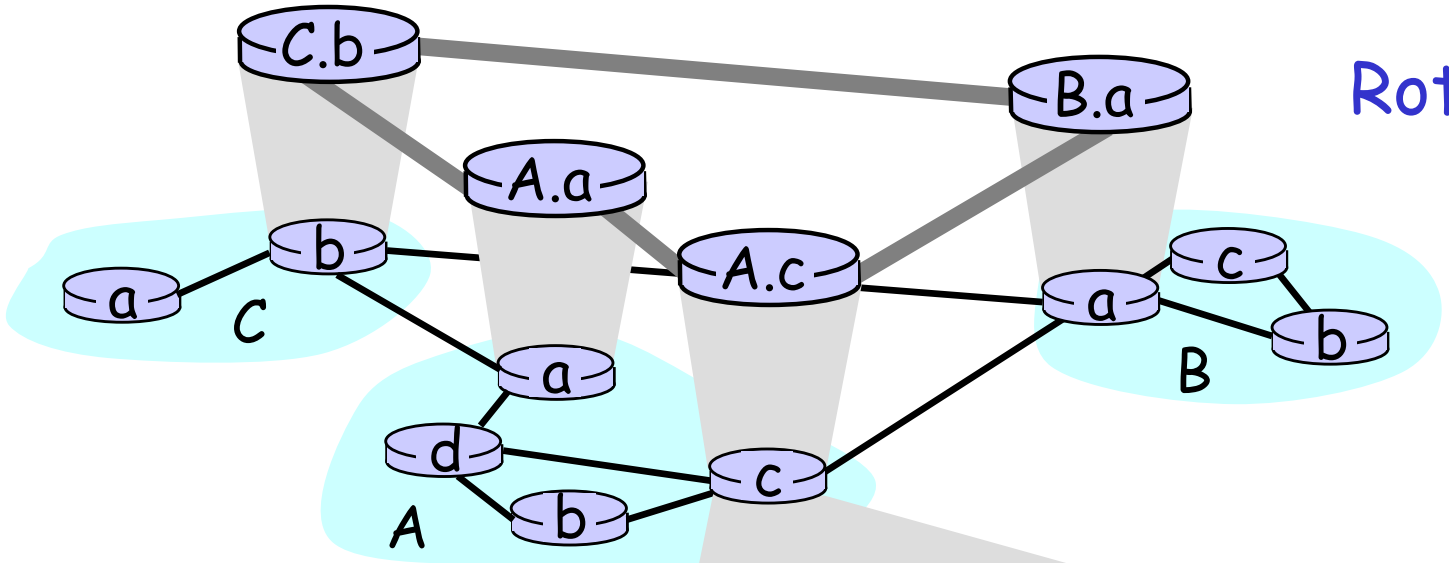
Roteamento Hierárquico

- agregar roteadores em regiões, "sistemas autônomos" (SAs)
- roteadores no mesmo SA usam o mesmo protocolo de roteamento
  - protocolo de roteamento "intra-SA"
  - roteadores em SAs diferentes podem usar diferentes protocolos de roteamento intra-SA

roteadores de borda

- roteadores especiais no SA
- usam protocolo de roteamento intra-SA com todos os demais roteadores no SA
- *também* responsáveis por rotear para destinos fora do SA
  - usam protocolo de roteamento "inter-SA" com outros roteadores de borda

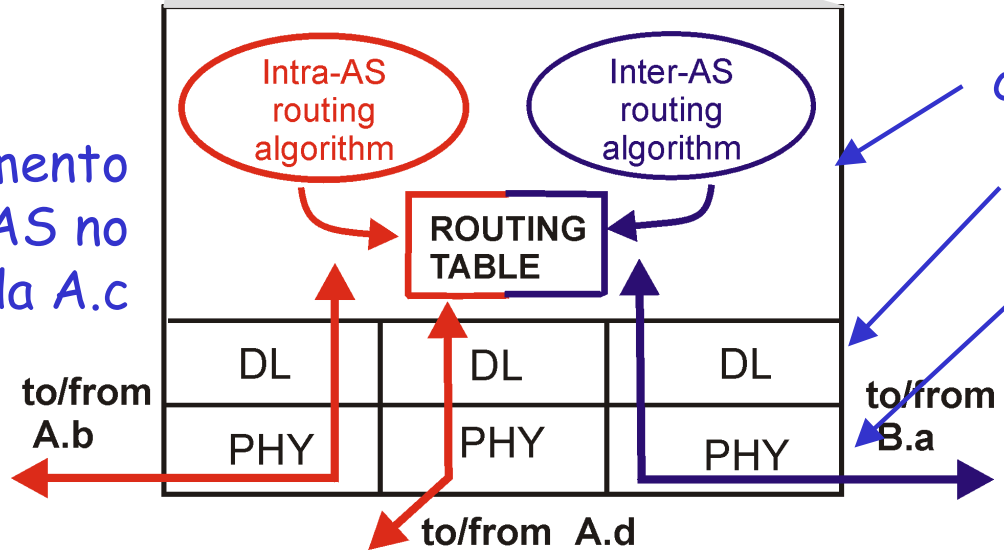
Roteamento Intra-SA e Inter-SA



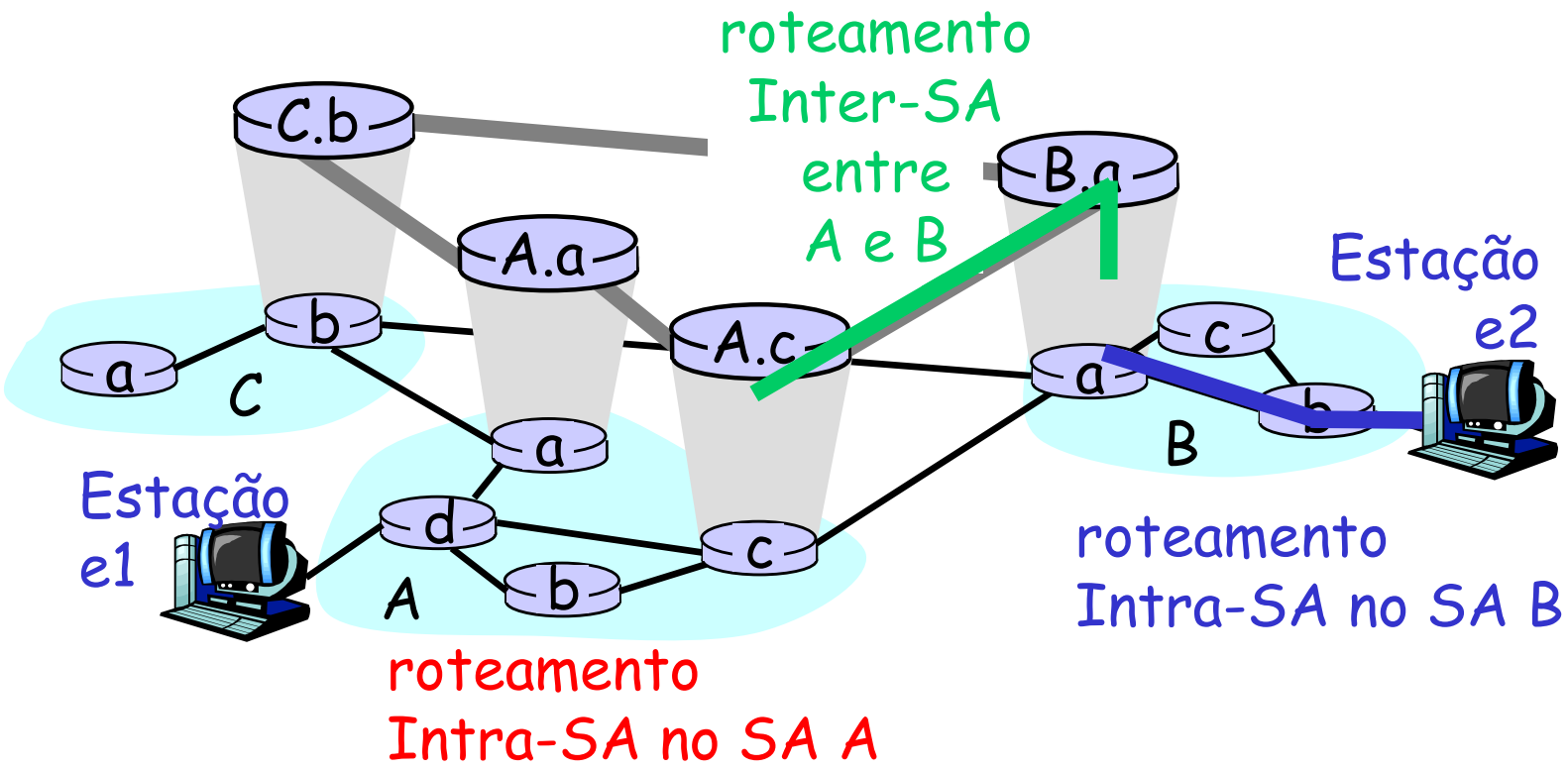
Roteadores de borda:

- fazem roteamento inter-SA entre si
- fazem roteamento intra-SA com outros roteadores do seu próprio SA

Roteamento inter-AS, intra-AS no roteador de borda A.c

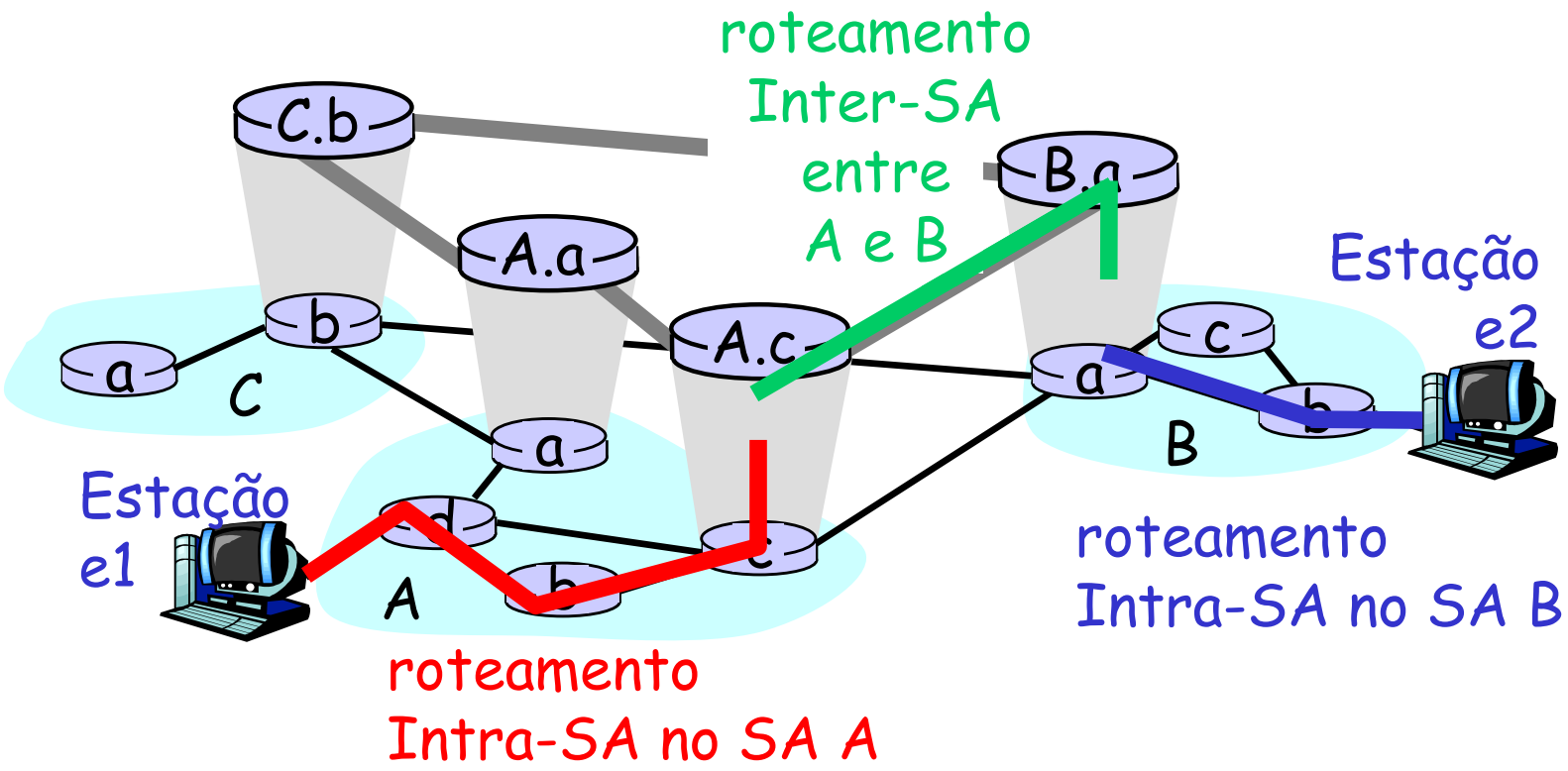


Roteamento Intra-SA e Inter-SA



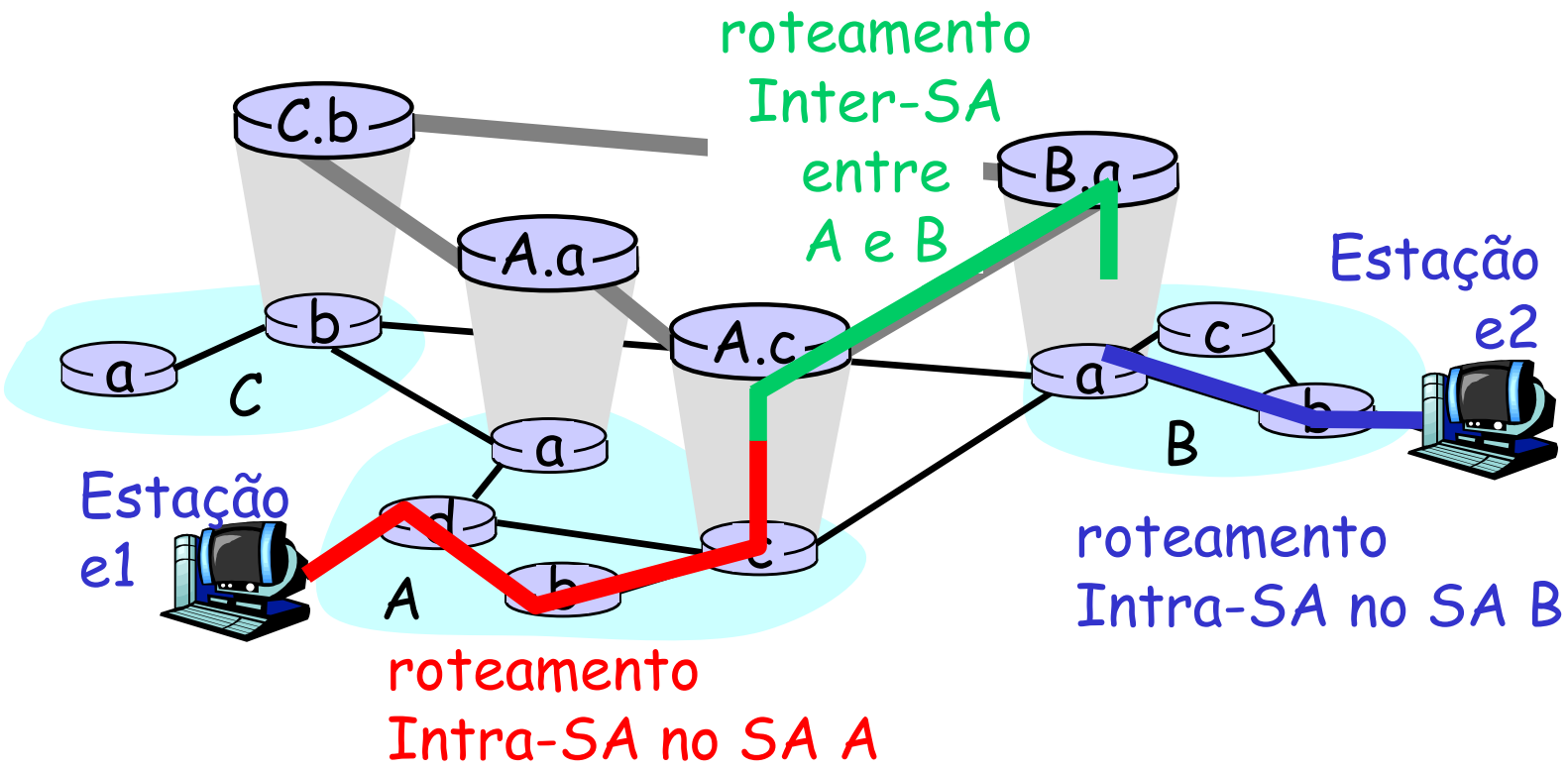
- Em breve veremos protocolos de roteamento inter-SA e intra-SA específicos da Internet

Roteamento Intra-SA e Inter-SA



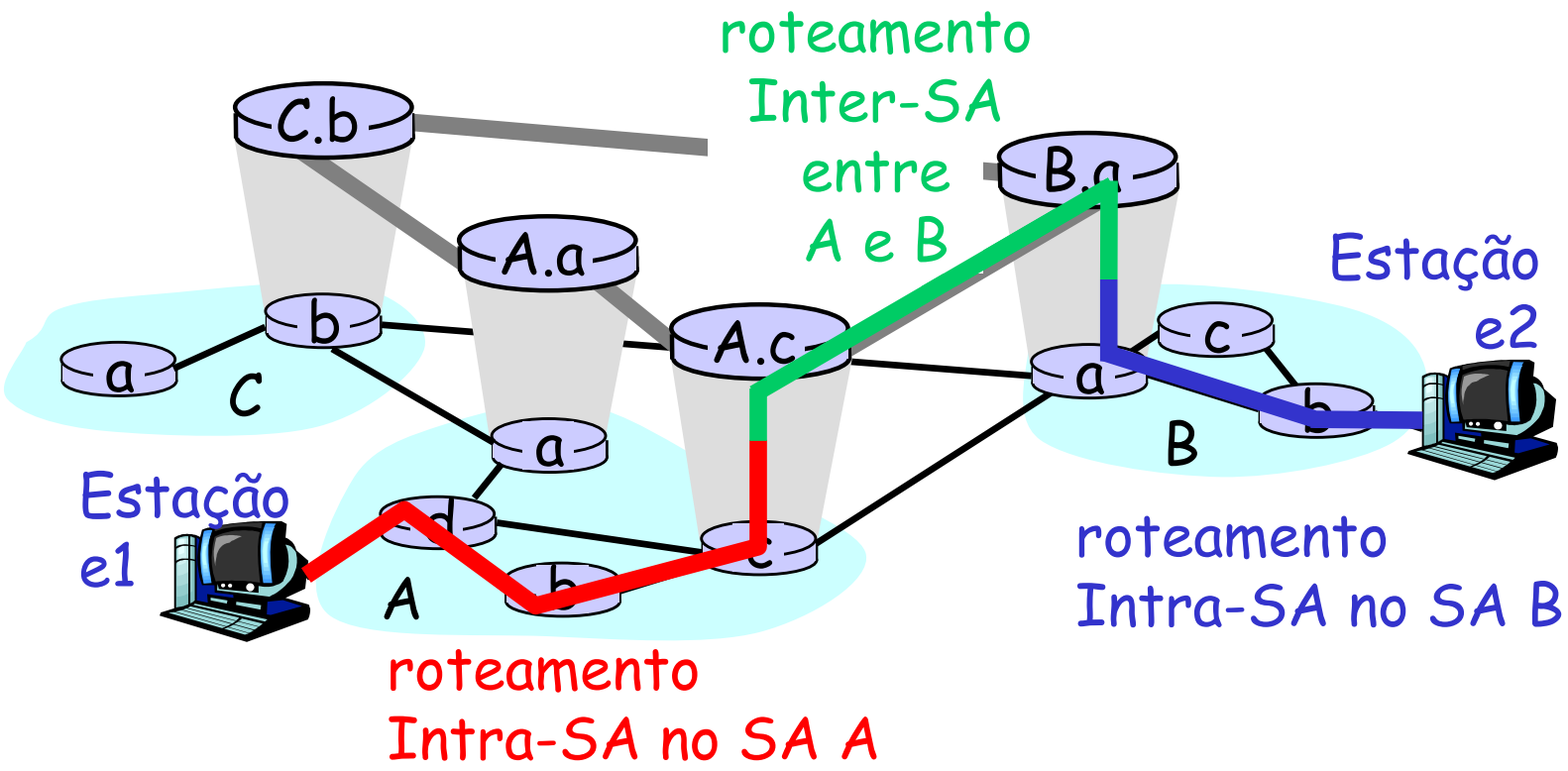
- Em breve veremos protocolos de roteamento inter-SA e intra-SA específicos da Internet

Roteamento Intra-SA e Inter-SA



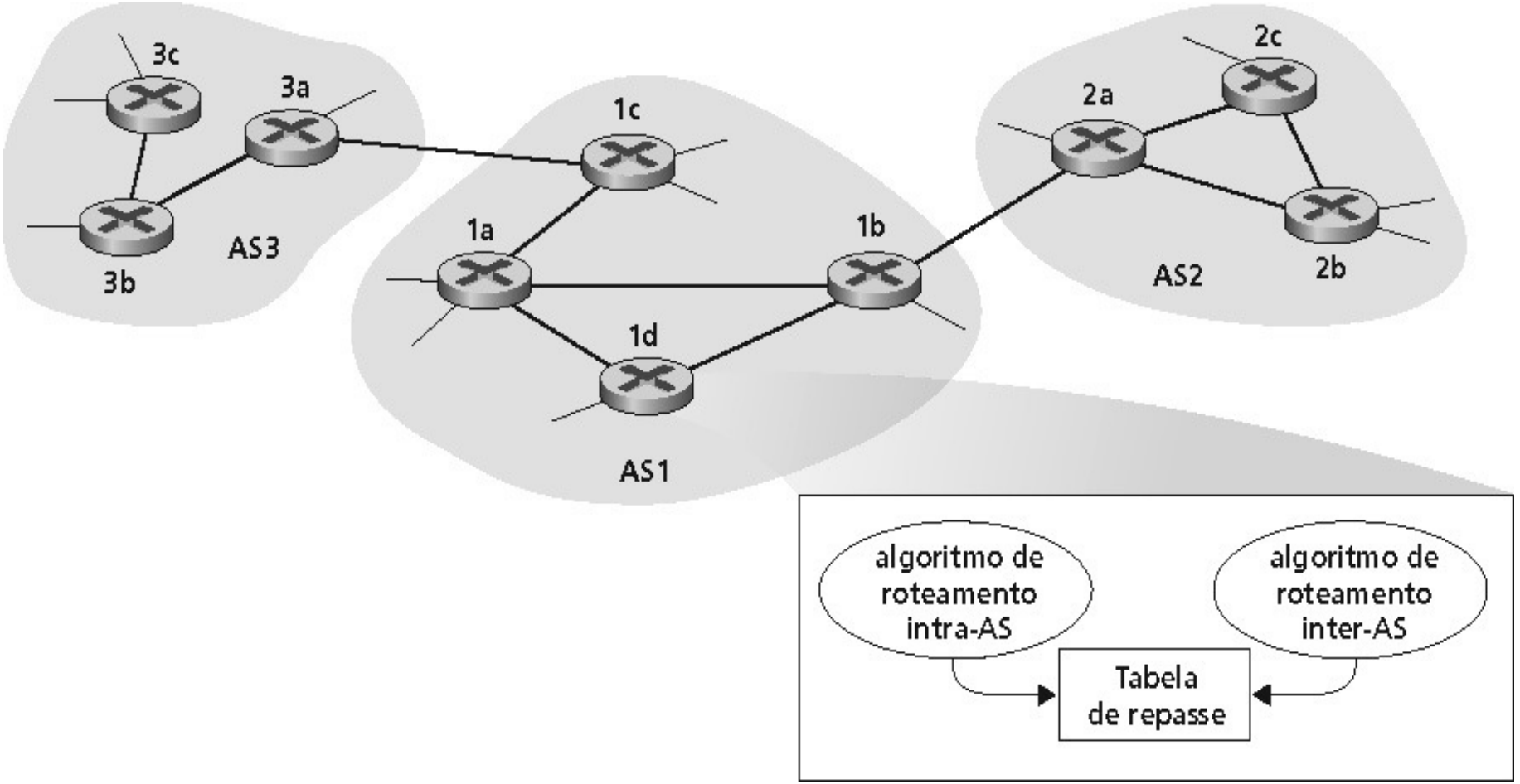
- Em breve veremos protocolos de roteamento inter-SA e intra-SA específicos da Internet

Roteamento Intra-SA e Inter-SA



- Em breve veremos protocolos de roteamento inter-SA e intra-SA específicos da Internet

ASs interconectadas



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

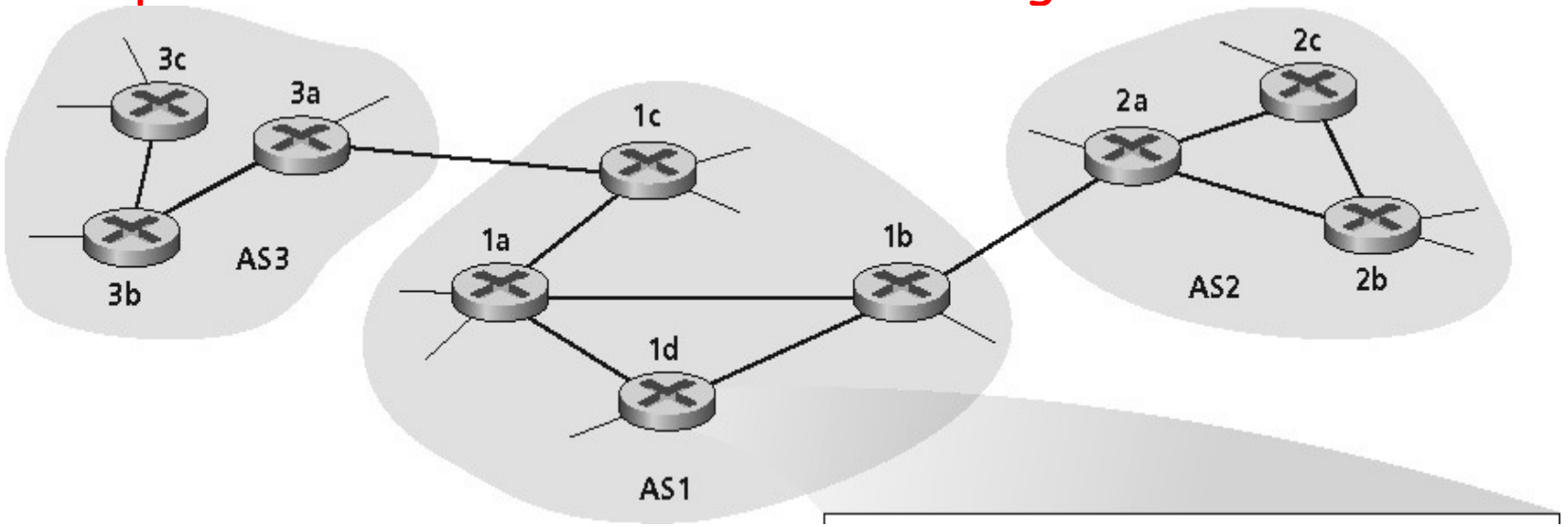
Tarefas Inter-AS

- 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 alcancáveis 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!

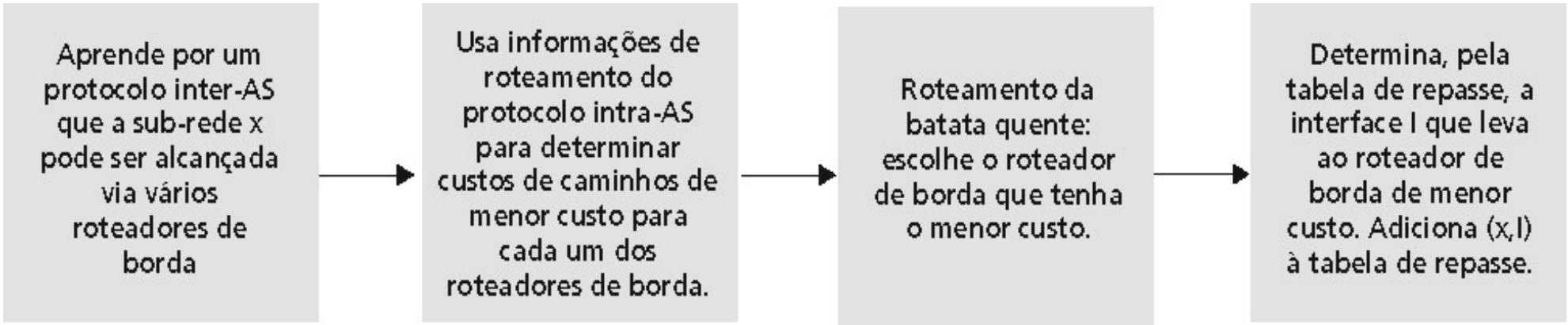


Exemplo: Ajustando a tabela de roteamento no roteador 1d

- Suponha que AS1 aprende pelo protocolo inter-AS protocol que a sub-rede **x** é alcancável 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)**.

Exemplo: Escolhendo entre múltiplas ASs

- 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.



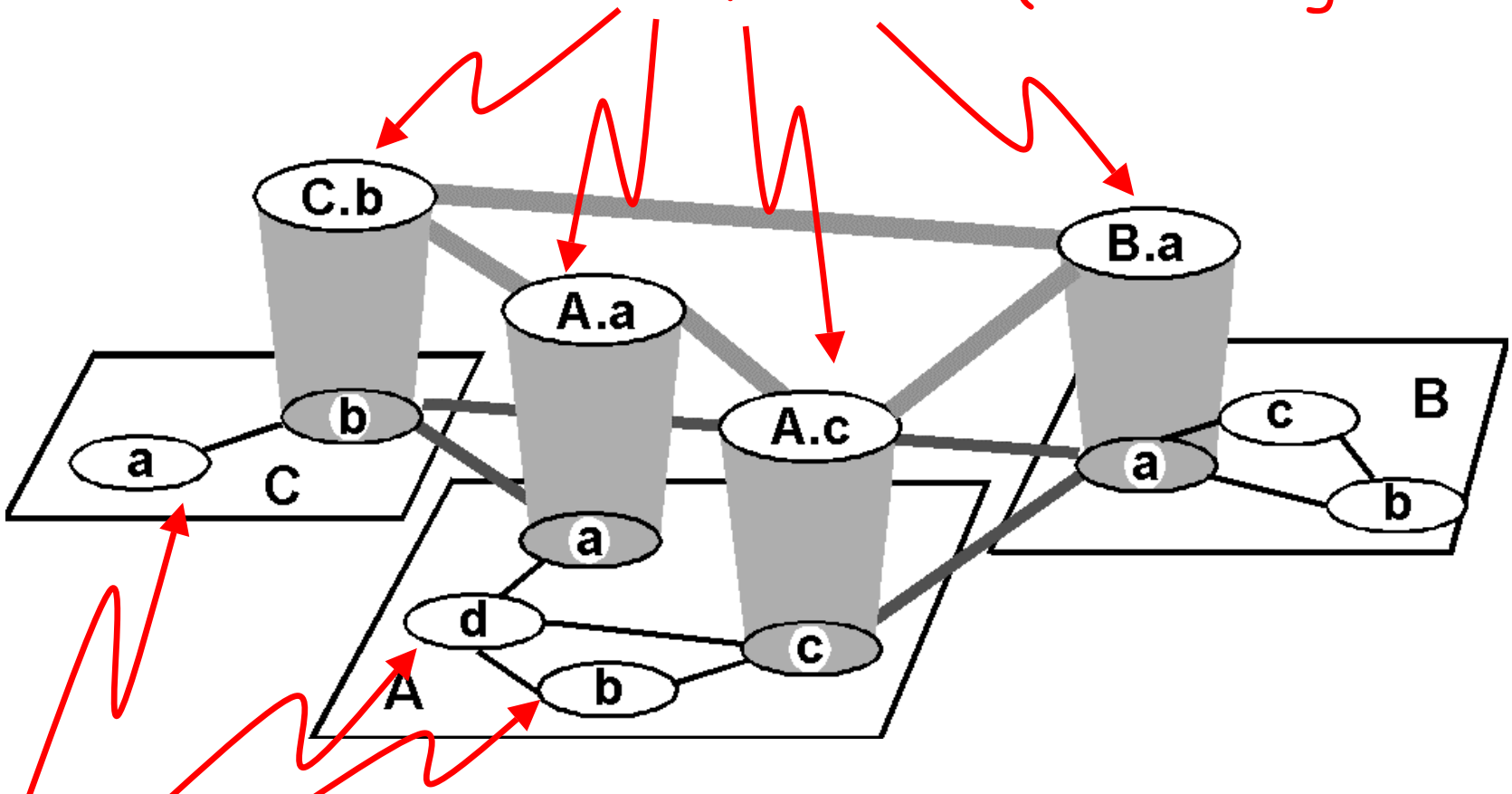
Roteamento na Internet

- A Internet Global consiste de Sistemas Autonômos (SAs) interligados entre si:
  - **SA Folha**: empresa pequena
  - **SA com Múltipla Conectividade**: empresa grande (sem trânsito)
  - **SA de Trânsito**: provedor

- Roteamento em dois níveis:
  - **Intra-SA**: administrador é responsável pela escolha
  - **Inter-SA**: padrão único

Hierarquia de SAs na Internet

Inter-SA: roteadores de fronteira (exterior gateways)



Intra-SA: roteadores internos (interior gateways)

Roteamento Intra-SA

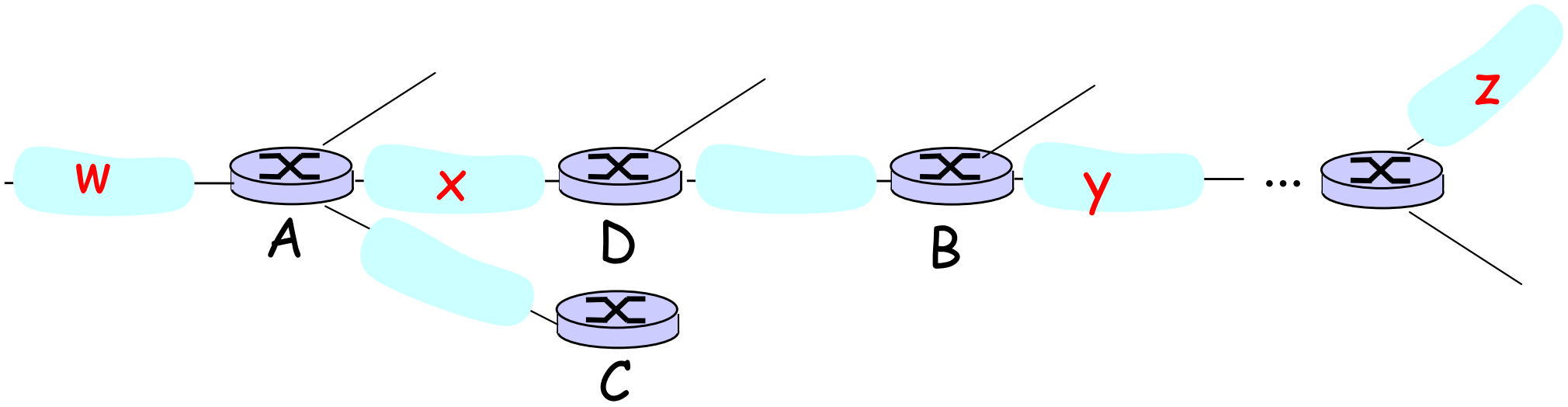
- Também conhecido como Interior Gateway Protocols (IGP) (protocolos de roteamento interno)
- Os IGPs mais comuns são:
  - RIP: *Routing Information Protocol*
  - OSPF: *Open Shortest Path First*
  - IGRP: *Interior Gateway Routing Protocol* (proprietário da Cisco)

RIP (*Routing Information Protocol*)

- Algoritmo do tipo vetor de distâncias
- Incluído na distribuição do BSD-UNIX em 1982
- Métrica de distância: # de enlaces (máx = 15 enlaces)
  - *Você pode adivinhar porquê?*

- Vetores de distâncias: trocados a cada 30 seg via Mensagem de Resposta (tb chamada de **anúncio**)
- Cada anúncio: rotas para 25 redes destino

RIP: exemplo



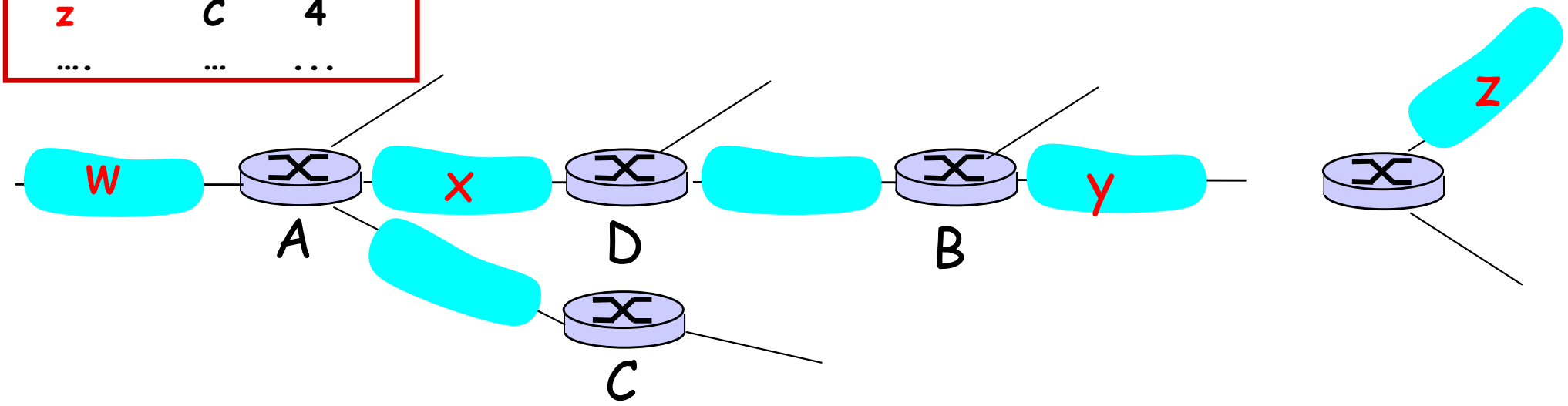
Rede Destino	Próximo Roteador	No. de enlaces ao destino
w	A	2
y	B	2
z	B	7
x	--	1
....	....	....

Tabela de rotas em D

RIP: Exemplo

Dest	Prox	hops
w	-	-
x	-	-
z	C	4
....	...	...

Anúncio de A para D



Rede Destino	Próximo Roteador	No. de enlaces ao destino
w	A	2
y	B	2
z	~~B~~ A	~~7~~ 5
x	--	1
....	....	....

Tabela de Roteamento em D

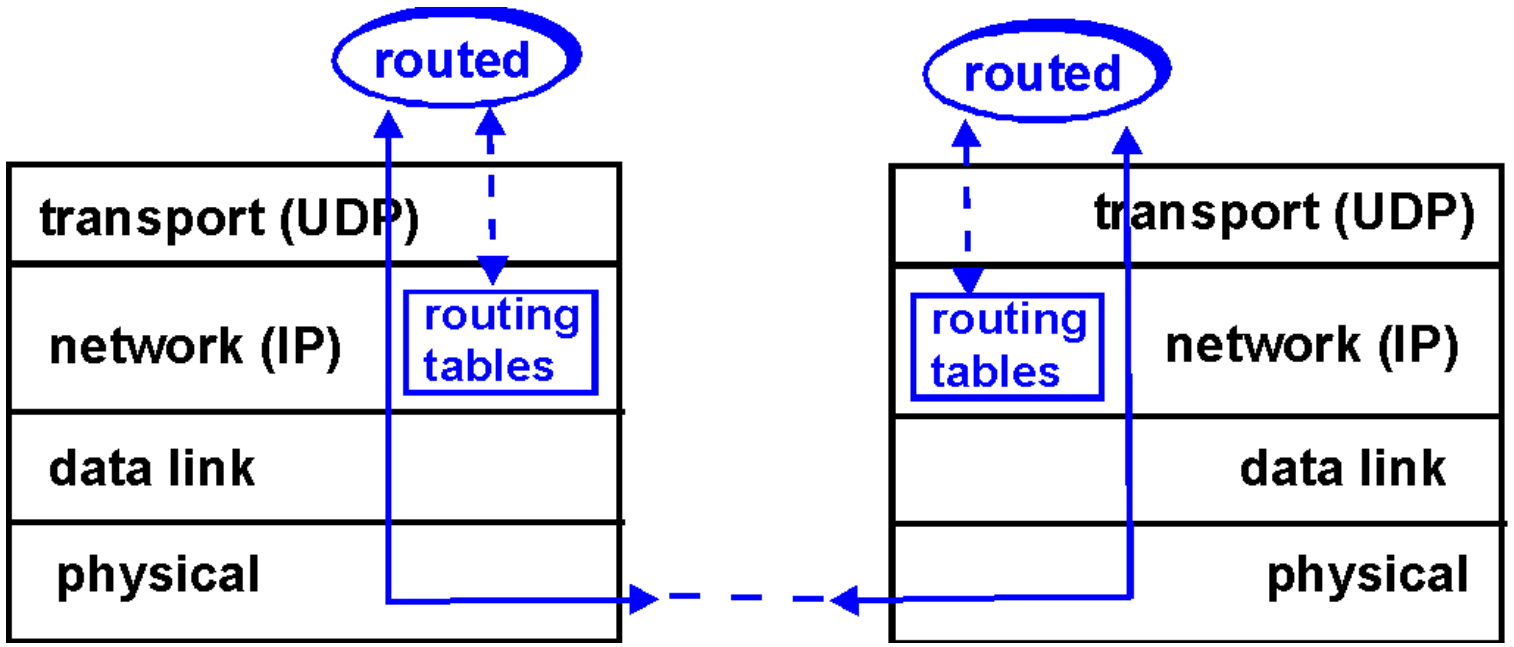
RIP: Falha e Recuperação de Enlaces

Se não for recebido anúncio novo durante 180 seg --> vizinho/enlace declarados mortos

- rotas via vizinho invalidadas
- novos anúncios enviados aos vizinhos
- na sua vez, os vizinhos publicam novos anúncios (se foram alteradas as suas tabelas)
- informação sobre falha do enlace rapidamente propaga para a rede inteira
- reverso envenenado usado para impedir rotas cíclicas (ping-pong) (distância infinita = 16 enlaces)

RIP: Processamento de tabelas

- Tabelas de roteamento RIP gerenciadas por processo de **nível de aplicação** chamado routed (routing daemon)
- anúncios enviados em pacotes UDP, repetidos periodicamente



RIP: exemplo de tabela de rotas (cont)

Router: *giroflee.eurocom.fr*

Destination	Gateway	Flags	Ref	Use	Interface
127.0.0.1	127.0.0.1	UH	0	26492	lo0
192.168.2.	192.168.2.5	U	2	13	fa0
193.55.114.	193.55.114.6	U	3	58503	le0
192.168.3.	192.168.3.5	U	2	25	qaa0
224.0.0.0	193.55.114.6	U	3	0	le0
default	193.55.114.129	UG	0	143454	

- Três redes vizinhas diretas da classe C (LANs)
- Roteador apenas sabe das rotas às LANs vizinhas
- Roteador "default" usado para "subir"
- Rota de endereço multiponto: 224.0.0.0
- Interface "loopback" (para depuração)

OSPF (*Open Shortest Path First*)

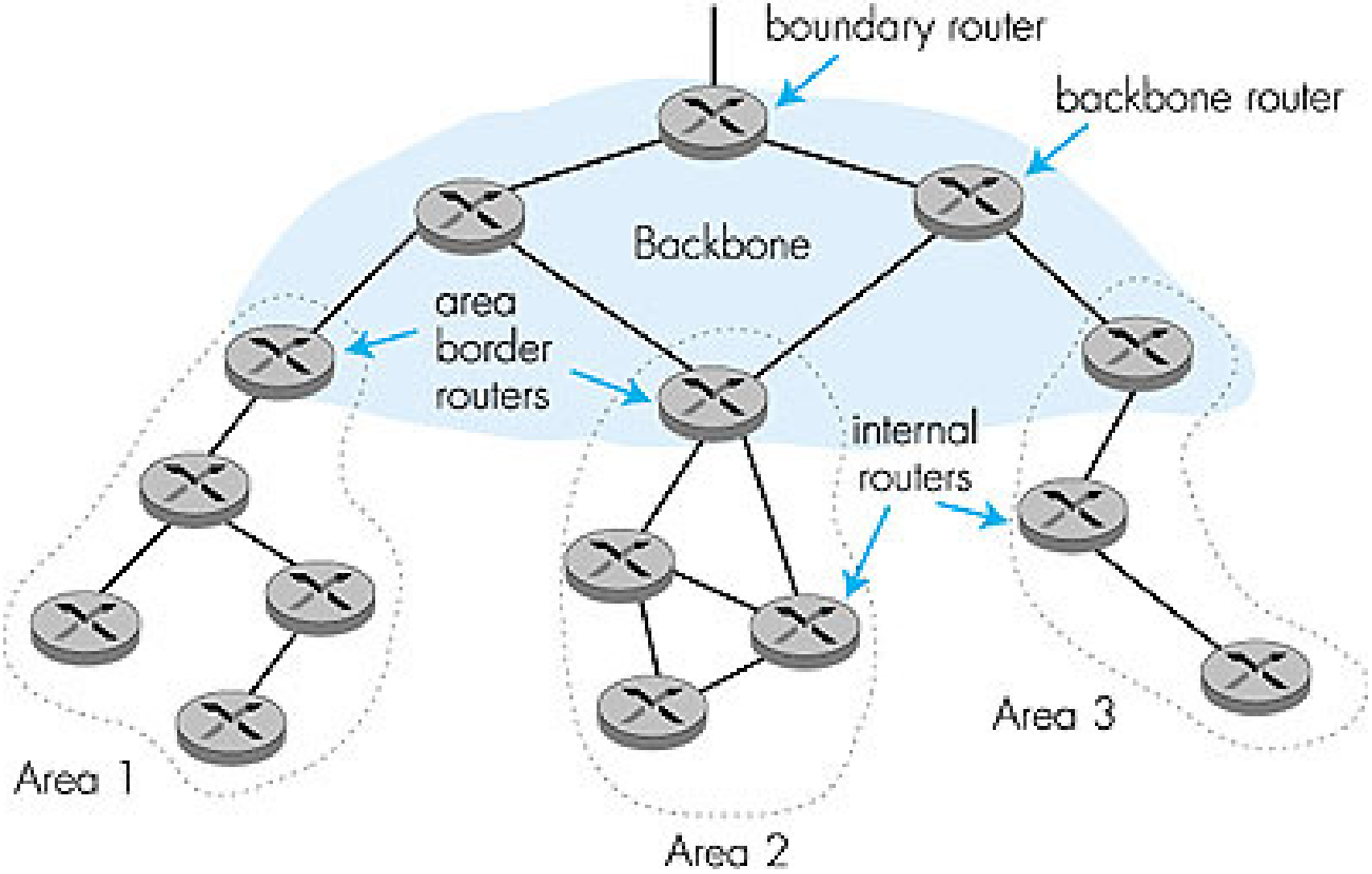
- "open" (aberto): publicamente disponível
- Usa algoritmo do Estado de Enlaces
  - disseminação de pacotes EE
  - Mapa da topologia a cada nó
  - Cálculo de rotas usando o algoritmo de Dijkstra

- Anúncio de OSPF inclui uma entrada por roteador vizinho
- Anúncios disseminados para SA inteiro (via inundação)

OSPF: características "avançadas" (não existentes no RIP)

- Segurança: todas mensagens OSPF autenticadas (para impedir intrusão maliciosa); conexões TCP usadas
- Caminhos Múltiplos de custos iguais permitidos (o RIP permite e usa apenas uma rota)
- Para cada enlace, múltiplas métricas de custo para TOS diferentes (p.ex, custo de enlace de satélite colocado como "baixo" para melhor esforço; "alto" para tempo real)
- Suporte integrado para ponto a ponto e multiponto:
  - OSPF multiponto (MOSPF) usa mesma base de dados de topologia usado por OSPF
- OSPF hierárquico em domínios grandes.

OSPF Hierárquico



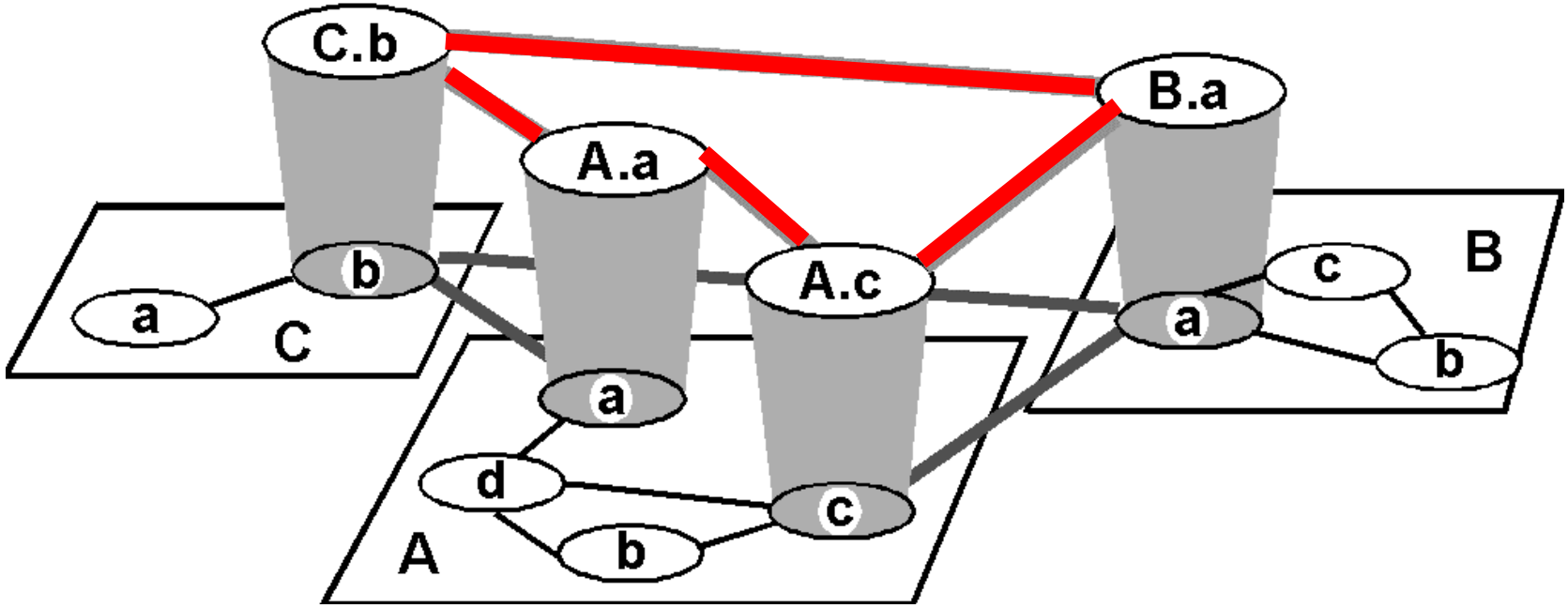
OSPF Hierárquico

- **Hierarquia de dois níveis:** área local, backbone.
  - Anúncios de EE disseminados apenas na mesma área
  - cada nó possui topologia detalhada da área; apenas sabe a direção (caminho mais curto) para redes em outras áreas (alcançadas através do backbone).
- **Roteador de fronteira de área:** "sumariza" distâncias às redes na sua própria área, anuncia a outros roteadores de fronteira de área.
- **Roteadores do backbone:** realizam roteamento OSPF limitado ao backbone.
- **Roteadores de fronteira:** ligam a outros SAs.

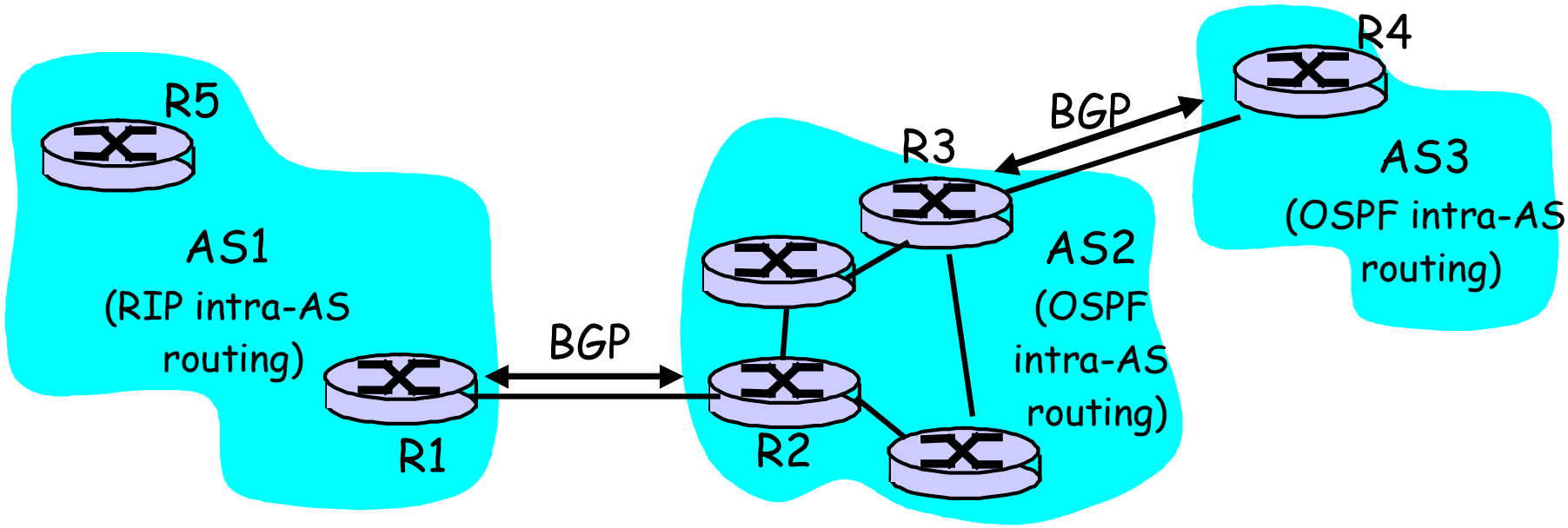
IGRP (*Interior Gateway Routing Protocol*)

- Proprietário da CISCO; sucessor do RIP (anos 80)
- Vetor de Distâncias, como RIP
- Diversas métricas de custo (retardo, largura de banda, confiabilidade, carga, etc)
- usa TCP para trocar mudanças de rotas
- Roteamento sem ciclos via *Distributed Updating Algorithm* (DUAL) baseado em *computação difusa*

Roteamento Inter-SA



Roteamento Inter-AS na Internet: BGP



Roteamento inter-SA na Internet: BGP

- BGP (*Border Gateway Protocol*): *o padrão* de fato
- Protocolo **Vetor de Caminhos** :
  - semelhante ao protocolo de Vetor de Distâncias
  - cada Border Gateway (roteador de fronteira) difunde aos vizinhos (pares) *caminho inteiro* (i.é., seqüência de SAs) ao destino
  - p.ex., roteador de fronteira X pode enviar seu caminho ao destino Z:

Caminho (X,Z) = X,Y1,Y2,Y3,…,Z

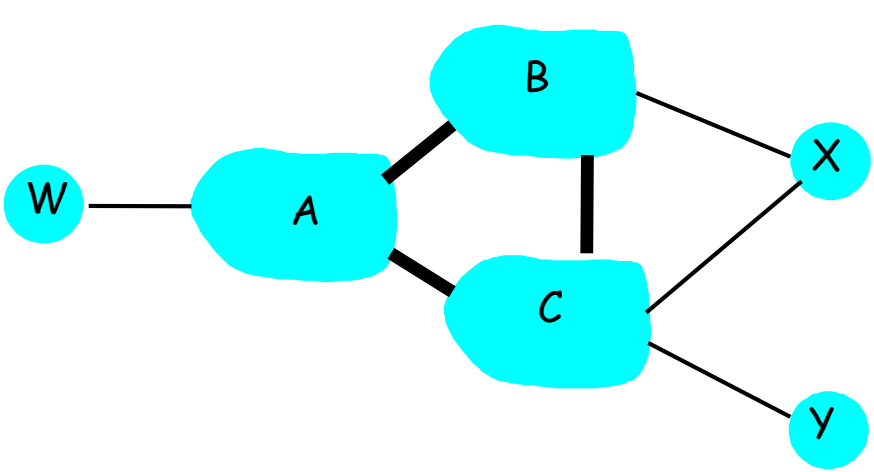
Roteamento inter-SA na Internet: BGP

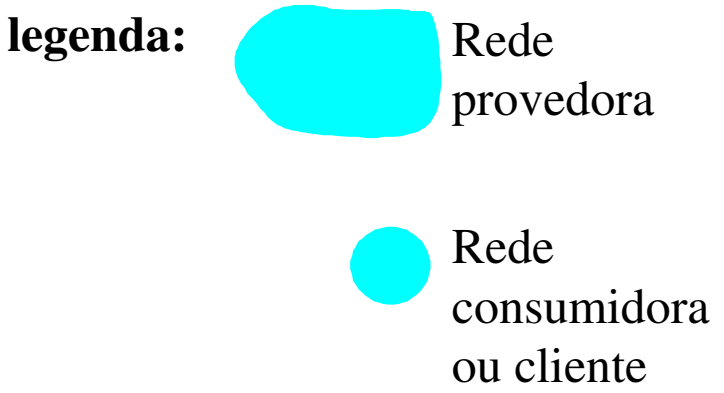
Suponha: roteador X envia seu caminho para roteador para W

- W pode ou não selecionar o caminho oferecido por X
  - razões de custo, políticas (não roteia via o SA de um concorrente), evitar ciclos.
- Se W seleciona caminho anunciado por X, então:

  Caminho (W,Z) = W, Caminho (X,Z)
- Note: X pode controlar tráfego de chegada através do controle dos seus anúncios de rotas aos seus pares:
  - p.ex., ex., se X não quer rotear tráfego para Z, X não informa nenhuma rota para Z

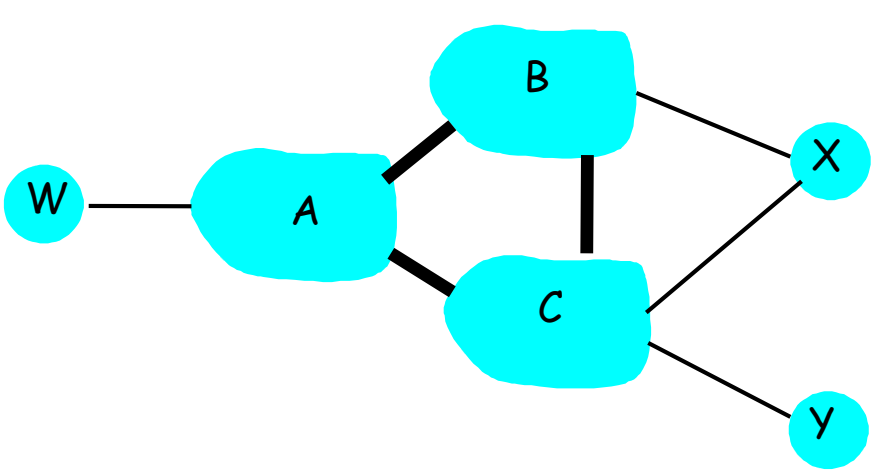
BGP: controlando quem roteia para você

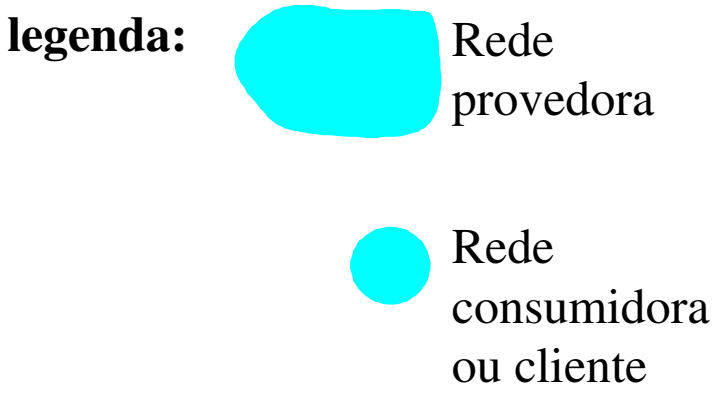




- A,B,C são redes provedoras
- X,W,Y são redes clientes (das redes provedoras)
- X é dual-homed: conectada a duas redes
  - X não deseja roteadar de B via X para C
  - .. assim X não anuncia para B a rota para C C

BGP: controlando quem roteia para você





- A anuncia para B o caminho AW
- B anuncia para X o caminho BAW
- B deve anunciar para C o caminho BAW?
  - De forma alguma! B não ganha nada para rotear CBAW dado que nem W nem C são clientes de B
  - B quer forçar C a rotear para W via A
  - B quer rotear *apenas* de/para seus clientes!

Operação BGP

Q: O que um roteador BGP faz?

- Envia anúncio de rotas para seus vizinhos;
- Recebe e filtra anúncios de rotas dos seus vizinhos diretamente conectados
- Escolha da rota .
  - Para rotear para o destino X, qual caminho (entre tantos anunciados) deve ser seguindo?

Mensagens BGP

- mensagens BGP trocadas usando TCP.
- mensagens BGP:
  - OPEN: abre conexão TCP ao roteador par e autentica remetente
  - UPDATE: anuncia caminho novo (ou retira velho)
  - KEEPALIVE mantém conexão viva na ausência de UPDATES; também reconhece pedido OPEN
  - NOTIFICATION: reporta erros na mensagem anterior; também usada para fechar conexão

Porque protocolos Intra- e Inter-AS diferentes ?

Políticas:

- Inter-SA: administração quer controle sobre como tráfego roteado, quem transita através da sua rede.
- Intra-AS: administração única, logo são desnecessárias decisões políticas

Escalabilidade:

- roteamento hierárquico economiza tamanho de tabela de rotas, reduz tráfego de atualização

Desempenho:

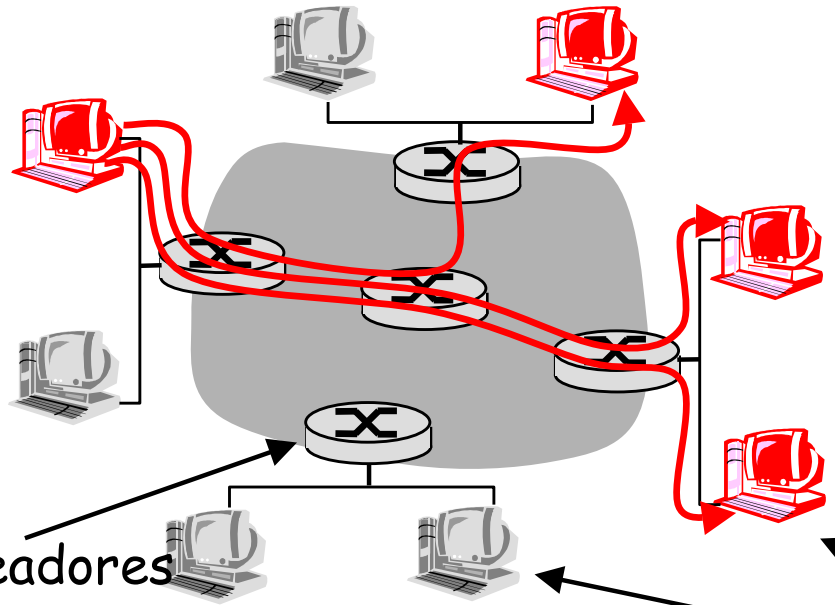
- Intra-AS: pode focar em desempenho
- Inter-AS: políticas podem ser mais importantes do que desempenho

Multicast: um emissor para vários receptores

- Multicast: envia datagramas para múltiplos receptores com uma **única operação de transmissão**
  - analogia: um professor para vários estudantes,
  - alimentação de dados: cotações da bolsa de valores;
  - atualização de cache WWW;
  - ambientes virtuais interativos distribuídos, etc.
- Questão: como garantir multicast?

Multicast: um emissor para vários receptores

- Multicast: envia datagramas para múltiplos receptores com uma **única operação de transmissão**
  - analogia: um professor para vários estudantes,
  - alimentação de dados: cotações da bolsa de valores;
  - atualização de cache WWW;
  - ambientes virtuais interativos distribuídos, etc.
- Questão: como garantir multicast?



Roteadores encaminham datagramas unicast

Receptores multicast (vermelho)

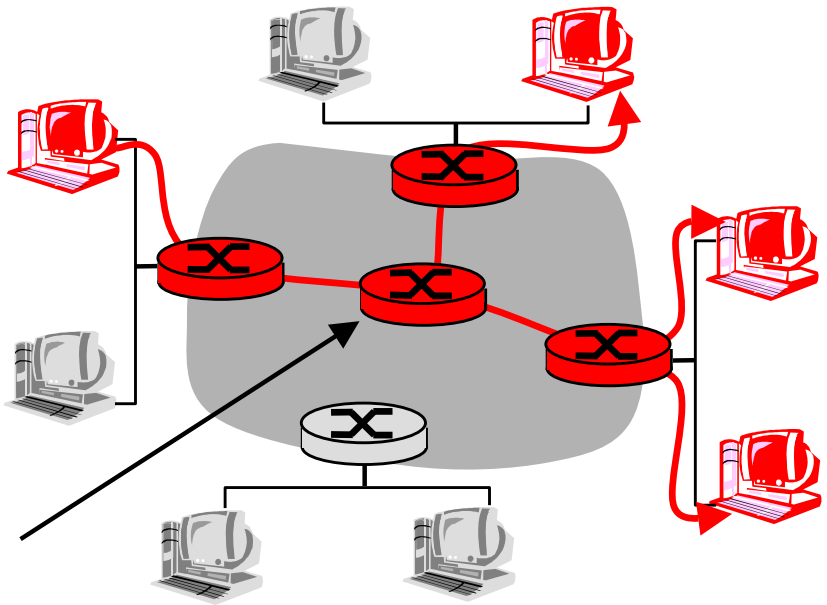
Não é um receptor multicast

Multicast via unicast

- Fonte envia N datagrams unicast, um para cada um dos N receptores

Multicast: um emissor para vários receptores

- Multicast: envia datagramas para múltiplos receptores com uma **única operação de transmissão**
- Questão: como garantir multicast?



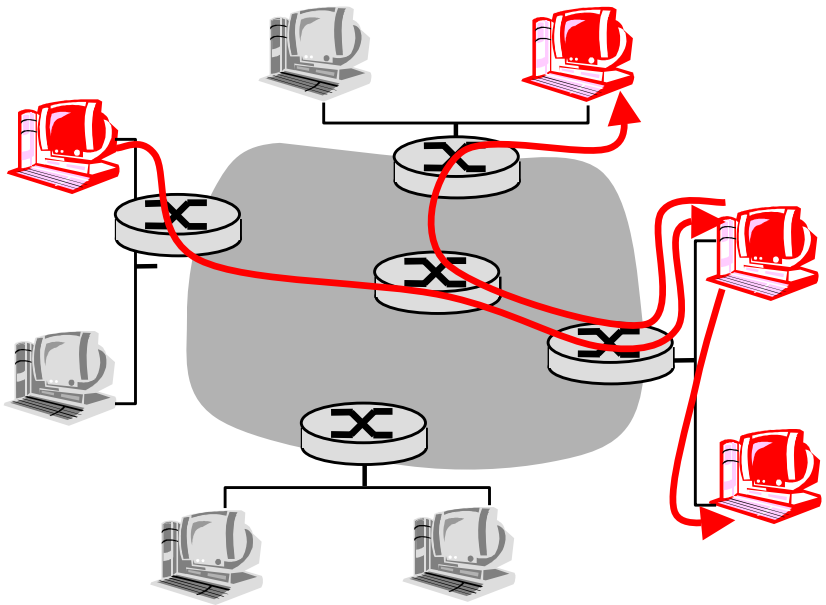
Roteadores multicast (vermelho) duplicam e encaminham os datagramas multicast

Rede multicast

- Roteadores participam ativamente do multicast, fazendo cópias dos pacotes e os encaminhando para os receptores multicast

Multicast: um emissor para vários receptores

- Multicast: envia datagramas para múltiplos receptores com uma **única operação de transmissão**
- Questão: como garantir multicast?



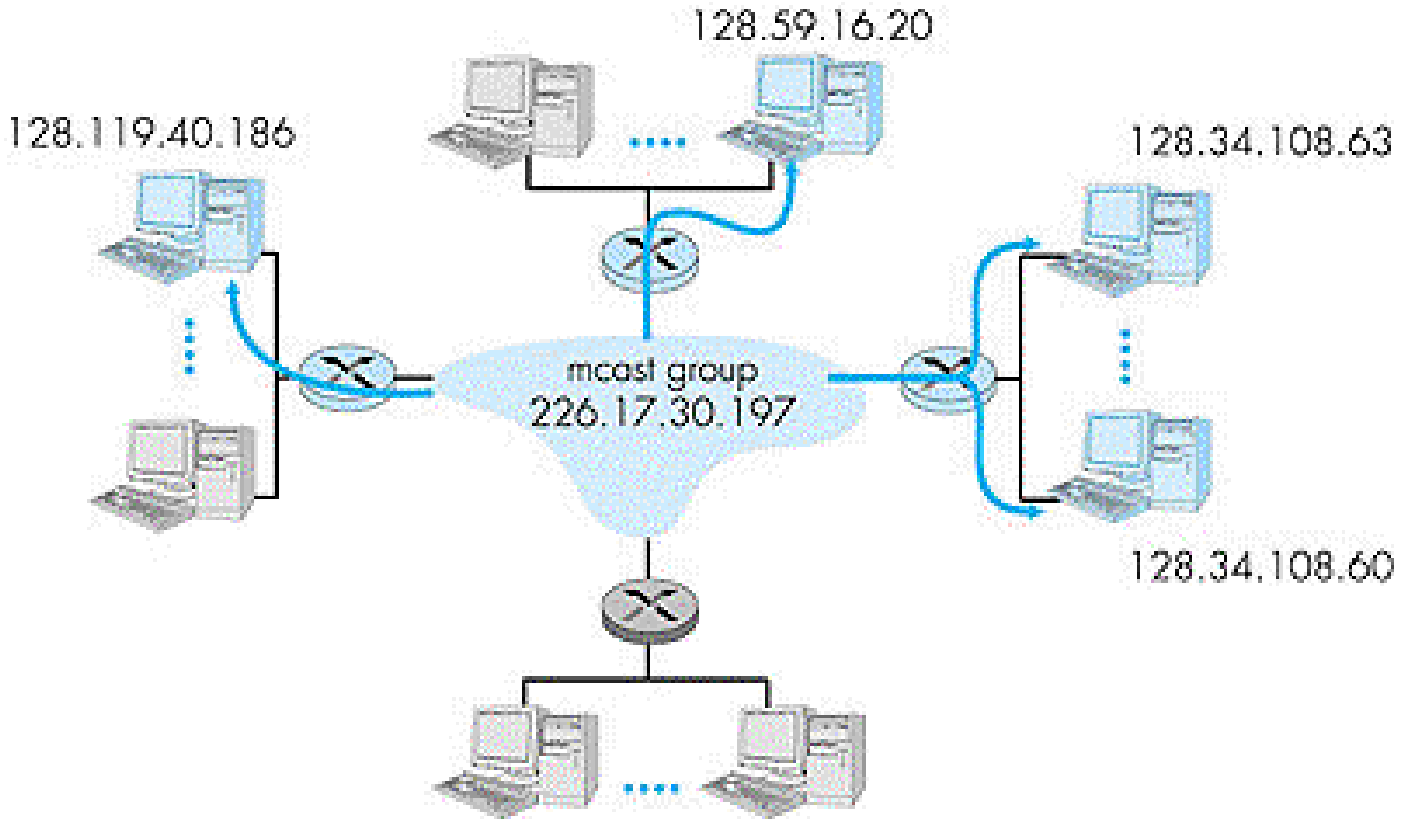
Multicast na camada de Aplicação

- Sistemas finais envolvidos no multicast copiam e encaminham datagrams unicast entre eles

Desafios do Suporte a Multicast na Camada de Rede

- Como identificar os receptores de um datagrama multicast?
- Como endereçar um datagrama a ser enviado para estes receptores.
- Não dá para incluir o endereço IP de cada um dos destinos no cabeçalho do datagrama!
  - Não funciona para um grande número de receptores;
  - requer que o transmissor conheça a identidade e endereços de cada um dos destinatários.
- Endereço indireto: é usado um identificador único para um grupo de usuários.
- Grupo Multicast associado a um endereço classe D.

Modelo de Serviço Multicast da Internet

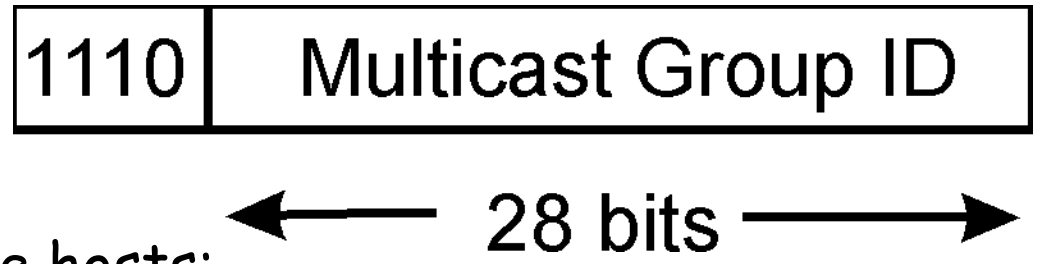


Conceito de grupo Multicast: uso de indireção

- Hosts endereçam os datagramas IP para o grupo multicast
- Roteadores encaminham os datagramas multicast para os hosts que se "juntaram" ao grupo multicast

Grupos Multicast

- Endereços classe D na Internet são reservados para multicast:



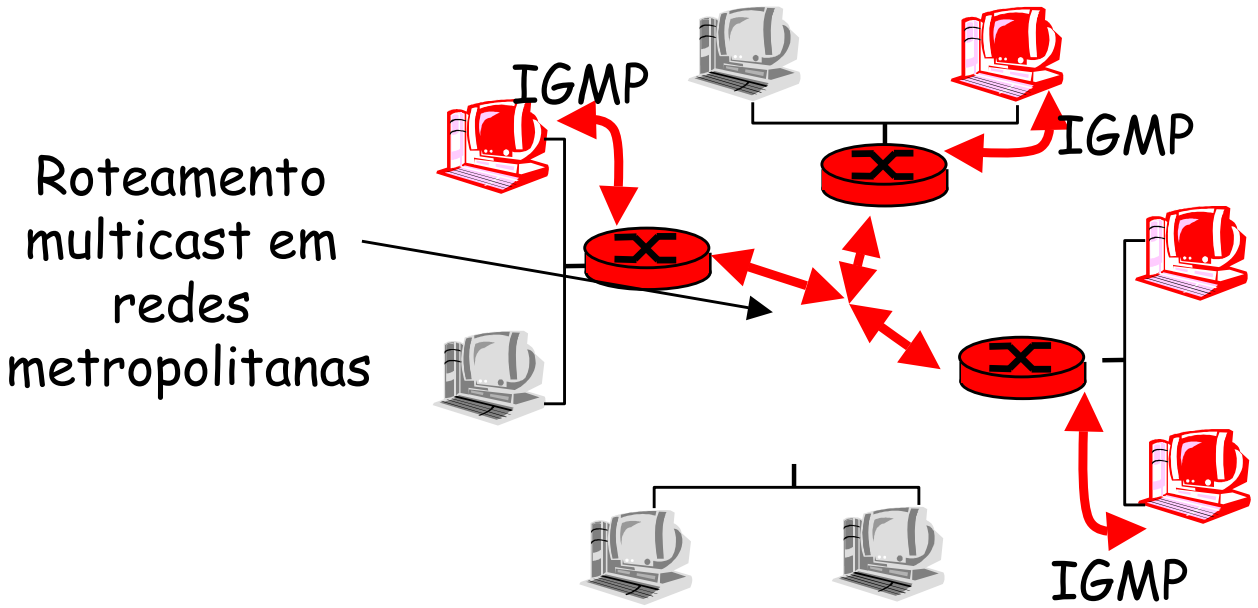
- Semântica de grupo de hosts:
  - ✓ qualquer um pode se "juntar" (receber) a um grupo multicast
  - ✓ qualquer um pode enviar para um grupo multicast
  - ✓ nenhuma identificação na camada de rede para os hosts membros
- *necessário:* infraestrutura para enviar datagramas multicast para todos os hosts que se juntaram ao grupo

Grupos Multicast: questões

- Como um grupo é iniciado e como ele é encerrado?
- Como é escolhido o endereço do grupo?
- Como são adicionados novos *hosts* ao grupo?
- Qualquer um pode fazer parte (ativa) do grupo ou a participação é restrita?
- Caso seja restrita, quem determina a restrição?
- Os membros do grupo têm conhecimento das identidades dos demais membros do grupo na camada de rede?
- Como os roteadores interoperam para entregar um datagrama multicast a todos os membros do grupo?

Juntando-se a um grupo Multicast: processo em dois passos

- *Rede local:* host informa ao roteador multicast local que deseja fazer parte do grupo: IGMP (Internet Group Management Protocol)
- *Rede metropolitanta:* roteador local interage com outros roteadores para receber os fluxos multicast
  - Vários protocolos (e.g., DVMRP, MOSPF, PIM)

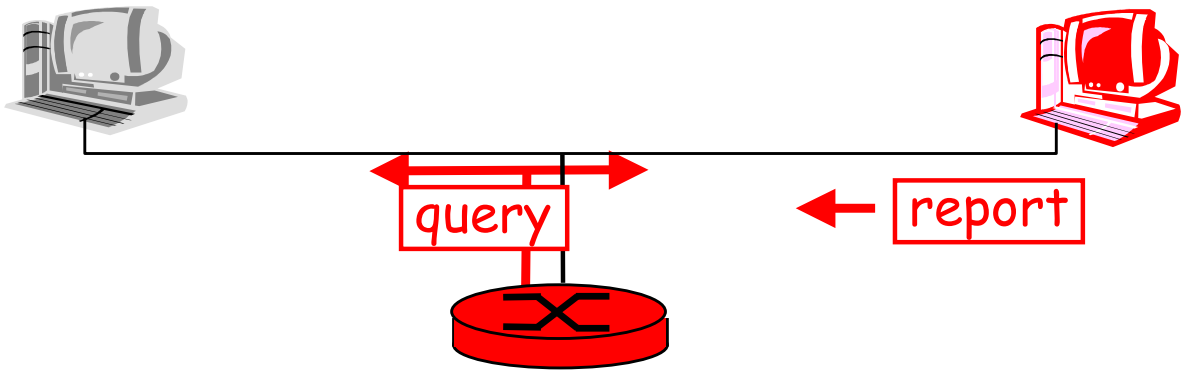


Multicast: aspectos da camada de rede

- Algoritmos de roteamento
- Multicast na Internet não é um serviço sem conexão:
  - devem ser estabelecidas conexões multicast
  - devem ser mantidas informações de estado das conexões multicast em cada roteador participante da mesma.
  - Necessita de uma combinação de protocolos de sinalização e de roteamento.

IGMP: Internet Group Management Protocol – RFC 2236

- Opera entre o host e o roteador ao qual ele está conectado diretamente:
- *host:* envia notificação IGMP quando a aplicação se junta a um grupo multicast
  - IP_ADD_MEMBERSHIP opção de socket
  - host não necessita fazer uma notificação quando sai de um grupo
- *roteador:* envia requisição IGMP a intervalos regulares
  - host pertencente a um grupo multicast deve responder a requisição



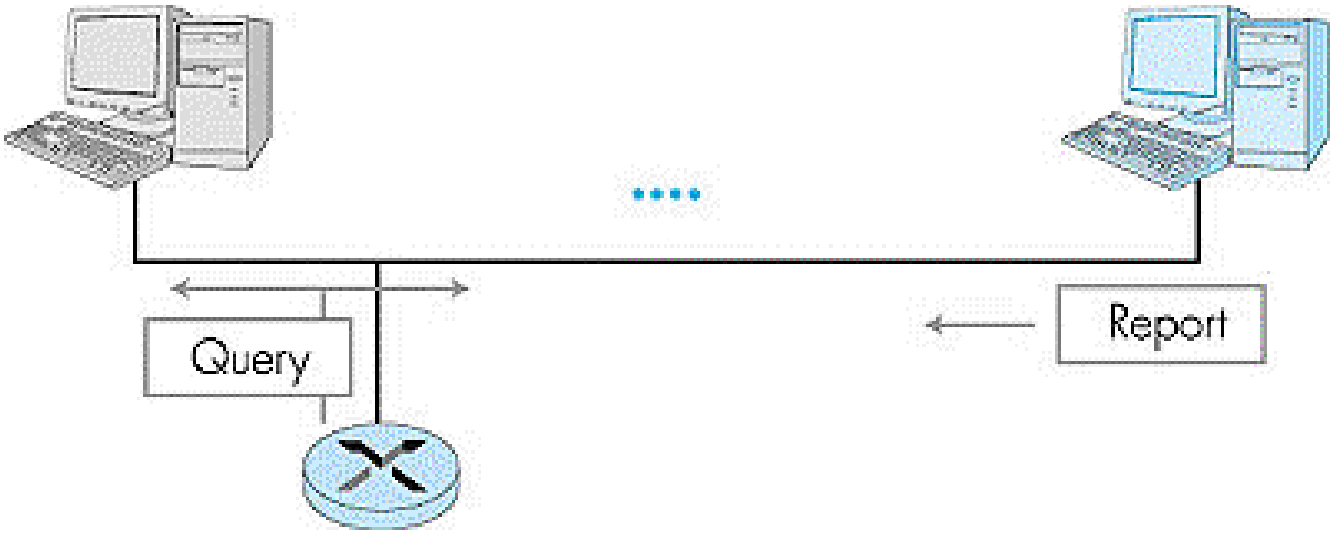
O Protocolo IGMP

- O IGMP fornece meios para que o host informe ao roteador ao qual está conectado que uma aplicação deseja ser incluída em um grupo multicast.
- Apesar do nome ele não é um protocolo que opera entre todos os *hosts* que tenham formado um grupo multicast.
- É necessário um outro protocolo para coordenar os roteadores multicast, de modo que os datagramas multicast sejam roteados até seus destinos: algoritmos de roteamento multicast da camada de rede.
  - Ex: PIM, DVMRP e MOSPF.

Tipos de Mensagens IGMP v2

Tipos das Mensagens IGMP	Enviada por	Finalidade
Consulta sobre participação em grupos:geral	Roteador	Consultar quais os grupos multicast em que os hosts associados estão incluídos.
Consulta sobre participação em grupos:específica	Roteador	Consultar se os hosts associados estão incluídos em um grupos multicast especificado.
Relato de participação	Host	Relatar que o host quer ser ou já está incluído num dado grupo multicast.
Saída de grupo	Host	Relata que está saindo de um determinado grupo multicast.

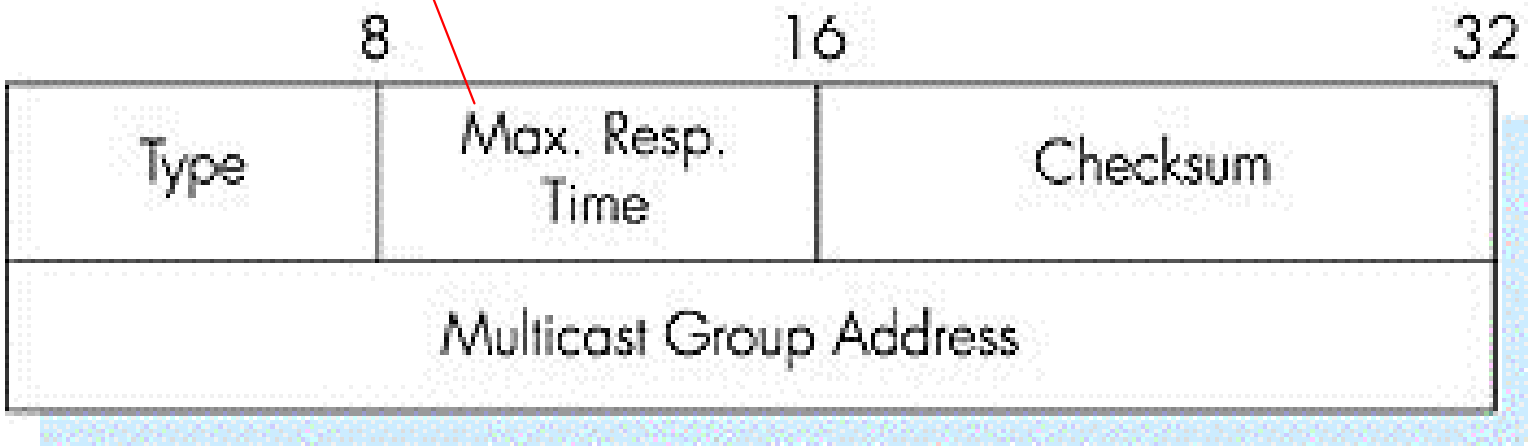
Consulta sobre participação e resposta



- As mensagens de relato também podem ser enviadas por iniciativa do host quando uma aplicação deseja ser incluída num grupo multicast.
- Para o roteador não importa quais nem quantos hosts fazem parte do mesmo grupo multicast.

Formato das Mensagens IGMP

Usado para suprimir relatos duplicados: cada host espera um tempo aleatório entre 0 e este valor máximo antes de enviar o seu relato. Se antes disto este host escutar o relato de algum outro host, ele descarta a sua mensagem.



Encapsuladas em datagramas IP com número de protocolo 2.

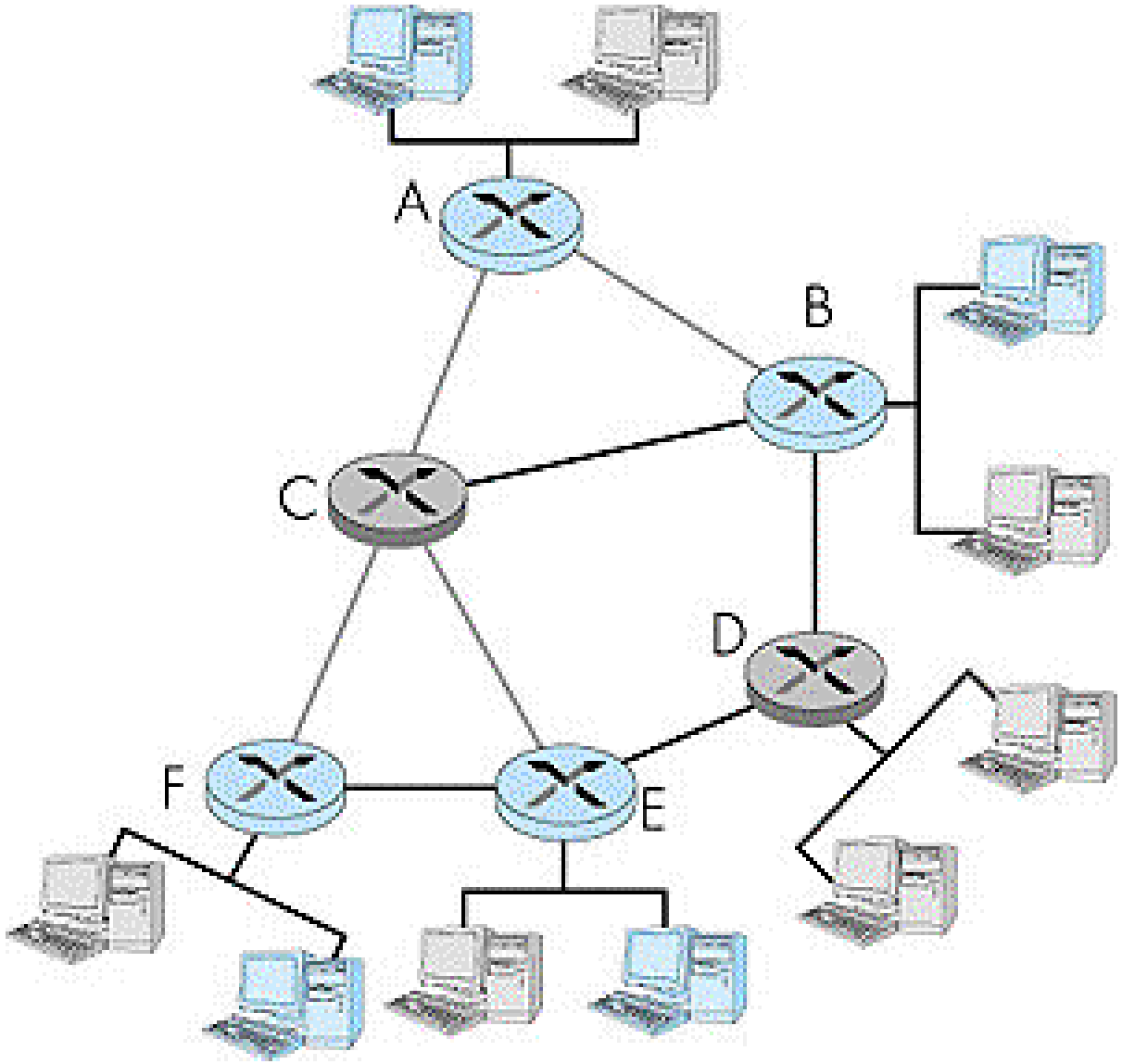
Modelo do Serviço Multicast da Internet

- Qualquer host pode ser incluído no grupo multicast na camada de rede.
  - O host simplesmente envia uma mensagem IGMP de relato de participação para o roteador ao qual está conectado.
- Em pouco tempo o roteador agindo em conjunto com os demais roteadores começará a entregar datagramas multicast para este host.
- Portanto, a adesão a um grupo é uma iniciativa do receptor.

Modelo do Serviço Multicast da Internet

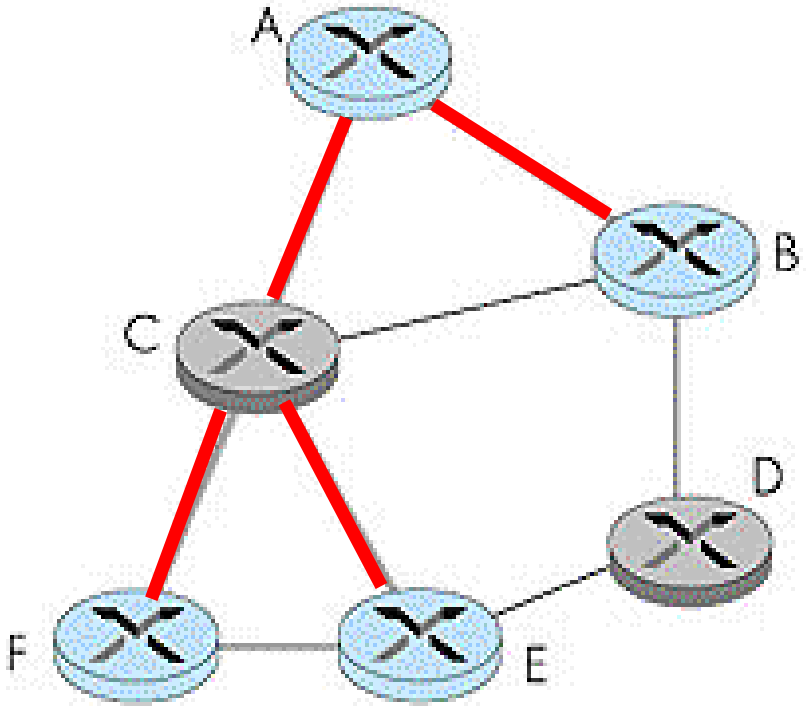
- O transmissor não precisa se preocupar em adicionar receptores e nem controla quem é incluído no grupo.
- Também não há nenhum controle de coordenação a respeito de quem e quando pode transmitir para o grupo multicast.
- Não há nem mesmo uma coordenação na camada de rede sobre a escolha de endereços multicast: dois grupos podem escolher o mesmo endereço!
- Todos estes controles podem ser implementados na camada de aplicação. Alguns deles podem vir a ser incluídos na camada de rede.

Roteamento Multicast: Exemplo

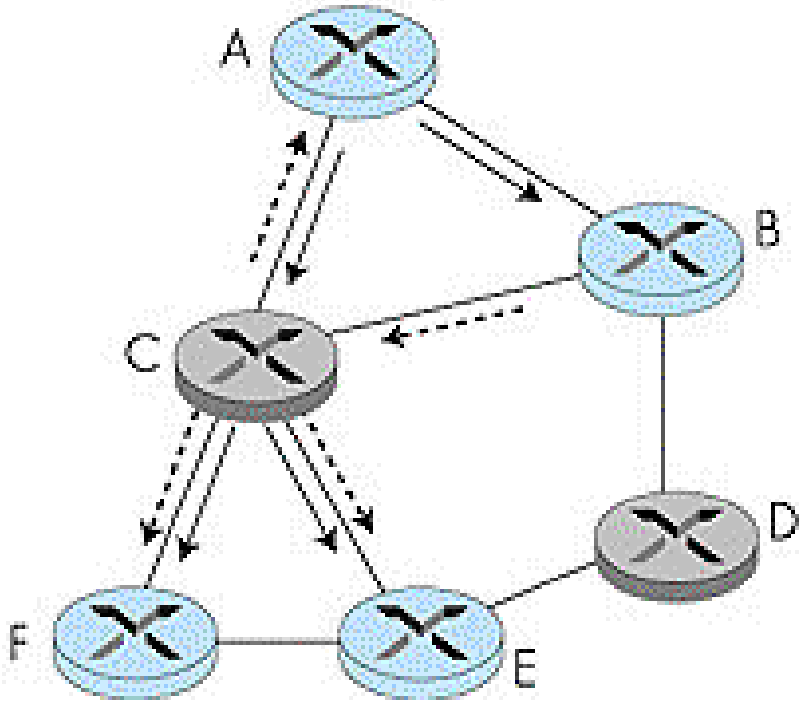


- Um único grupo multicast.
- Estão coloridos os hosts que pertencem ao grupo e os roteadores aos quais eles estão conectados.
- Apenas estes roteadores (A, B, E e F) necessitam receber este tráfego multicast.

Árvores de Roteamento Multicast



Árvore única compartilhada pelo grupo.

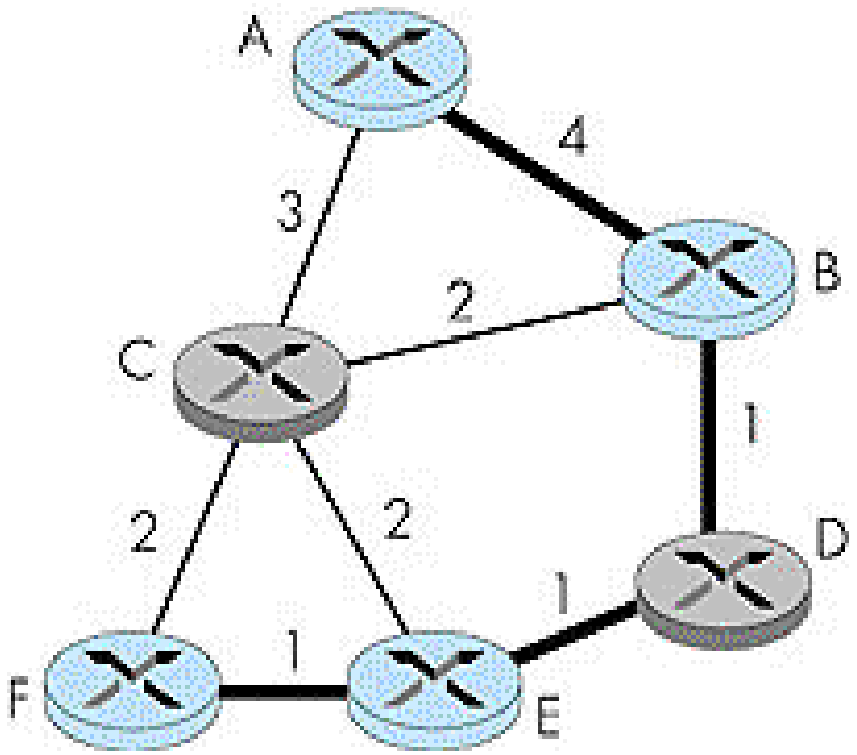


Árvores baseadas nas origens.

Roteamento Multicast usando uma árvore compartilhada

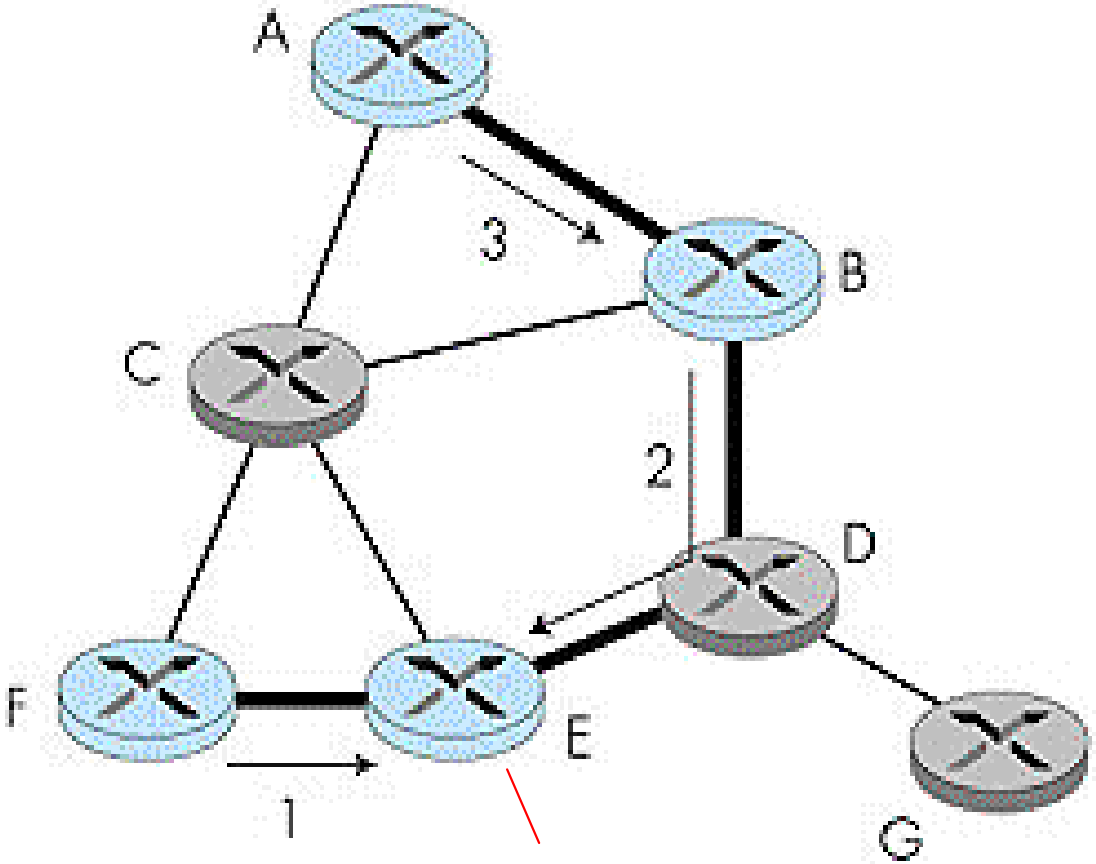
- Encontrar uma árvore que contenha todos os roteadores que tenham conectados a si todos os *hosts* pertencentes a um dado grupo.

- O problema de encontrar uma árvore com custo mínimo é conhecido como o problema da árvore de Steiner.
- Este é um problema NP-completo, mas há diversos algoritmos de aproximação que dão bons resultados.
- Nenhum algoritmo de roteamento multicast da



Árvore ótima com custo 7.

Construindo uma árvore baseada no centro



Legenda

roteador sem conexão com nenhum membro do grupo

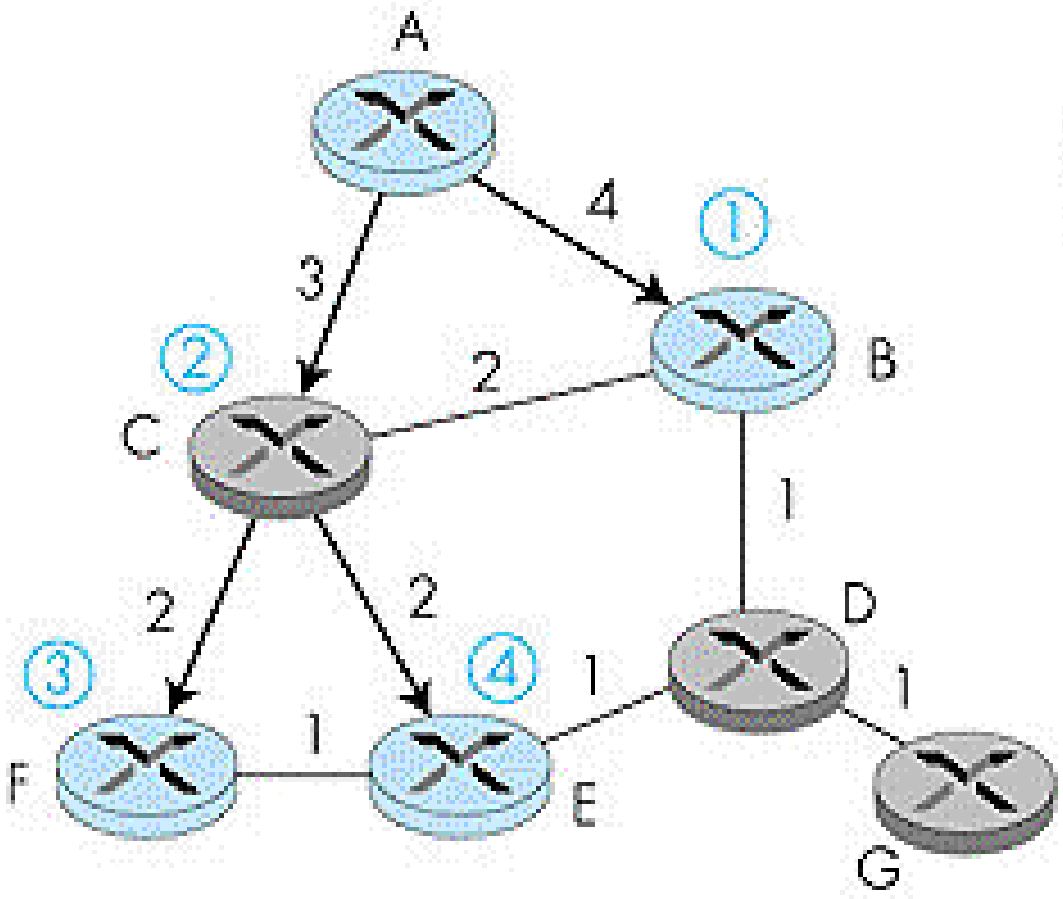
roteador com conexão a algum membro do grupo

Caminho/ordem na qual são geradas as mensagens de adesão.

Os caminhos são enxertados na árvore existente.

Como escolher o centro?

Roteamento Multicast usando árvores baseadas nas origens



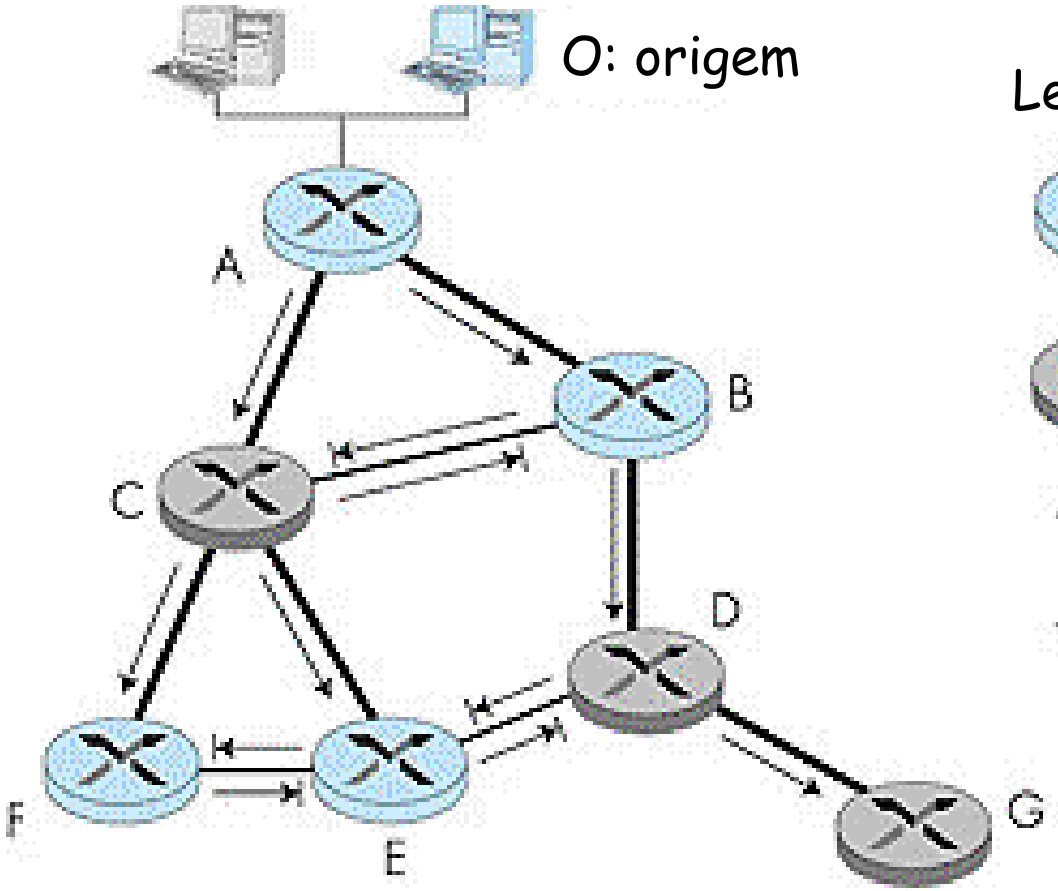
i-ésimo caminho a ser adicionado

- Árvores de caminho mais curto a partir de cada origem.
- Este é um algoritmo de EE (cada roteador deve conhecer o estado de cada enlace na rede).
- Mais simples: envio pelo caminho reverso (RPF – *Reverse Path Forwarding*)

Envio pelo Caminho Reverso

- Idéia simples, mas elegante.
- Quando um roteador recebe um pacote multicast, ele transmite o pacote em todos os seus enlaces de saída (exceto por aquele em que recebeu o pacote) apenas se o pacote tiver sido recebido através do enlace que está no seu caminho mais curto até o transmissor (origem).
- Note que o roteador não precisa conhecer o caminho mais curto até a origem, mas apenas o próximo roteador no seu caminho mais curto unicast até a origem.

Envio pelo Caminho Reverso



Legenda

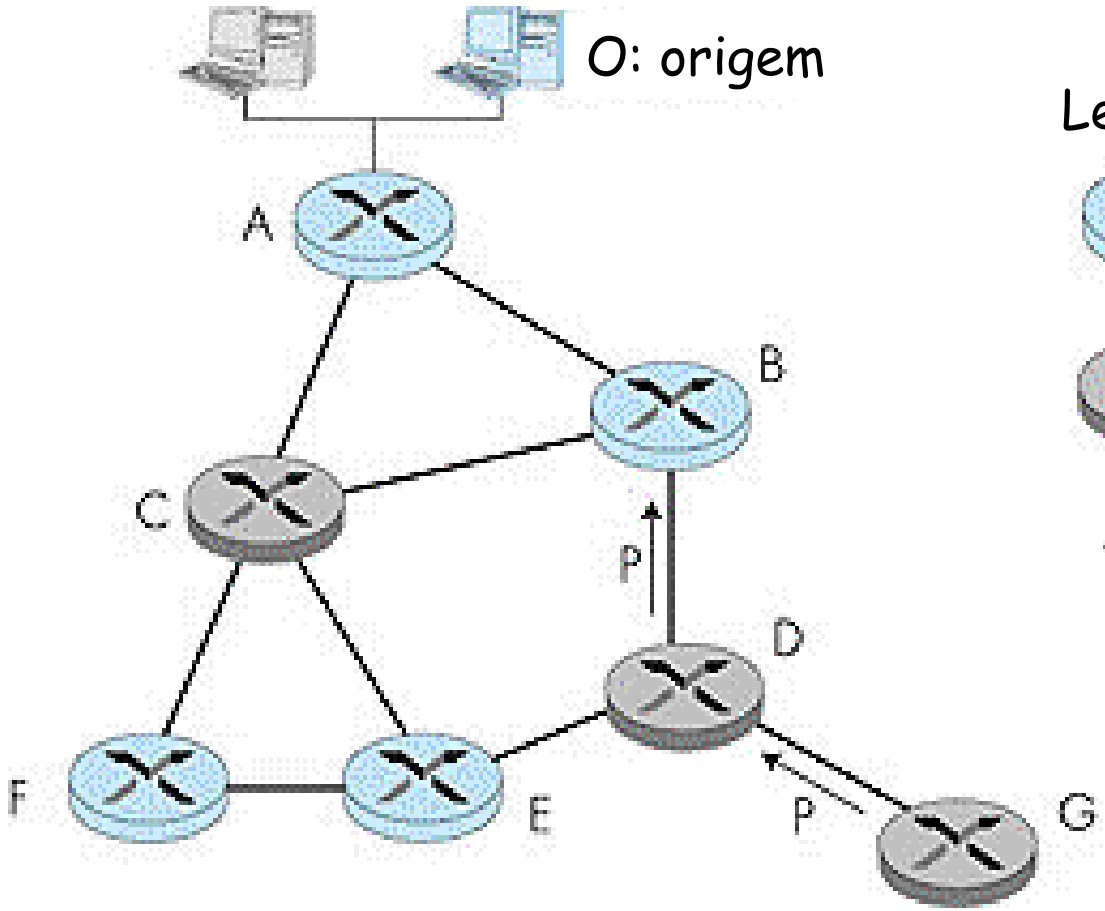
- roteador com conexão a algum membro do grupo
- roteador sem conexão com nenhum membro do grupo
- pacote a ser enviado
- pacote que não será enviado além do roteador receptor

Problema: G e outros roteadores a partir dele receberiam pacotes multicast apesar de não terem conexão com nenhum *host* participante do grupo!
Solução: Podar a árvore!

Poda da árvore de envio pelo caminho reverso

- Um roteador multicast que receba pacotes multicast e não possua conectado a ele nenhum *host* participante daquele grupo, enviará uma mensagem de poda para o roteador que estiver anterior a ele na árvore até a origem.
- Se um roteador receber mensagens de poda de todos os roteadores que estão abaixo dele, ele poderá enviar uma mensagem de poda para o roteador anterior a ele.

Poda da árvore de envio pelo caminho reverso



Legenda

roteador com conexão a algum membro do grupo

roteador sem conexão com nenhum membro do grupo

P mensagem de poda

Poda: questões sutis

- Requer que o roteador conheça quais roteadores abaixo dele dependem dele para receber pacotes multicast.
- Após o envio de uma mensagem de poda o que acontece se ele necessitar fazer parte do grupo?
  - Pode ser inserida uma mensagem de **enxerto** que permitiria desfazer a poda.
  - Os galhos podados seriam reincorporados à arvore após o estouro de um temporizador. O roteador poderia refazer a poda caso ainda não tivesse interesse no tráfego multicast.

Protocolos de Roteamento Multicast na Internet

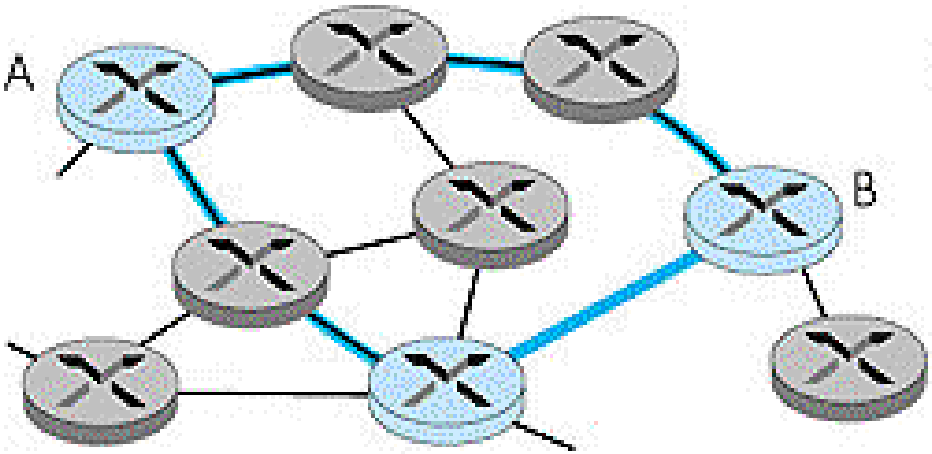
- **DVMRP:** *Distance Vector Multicast Routing Protocol*
- **MOSPF:** *Multicast Open Shortest Path First*
- **CBT:** *Core-Based Trees*
- **PIM:** *Protocol Independent Multicast*

DVMRP – *Distance Vector Multicast Routing Protocol*

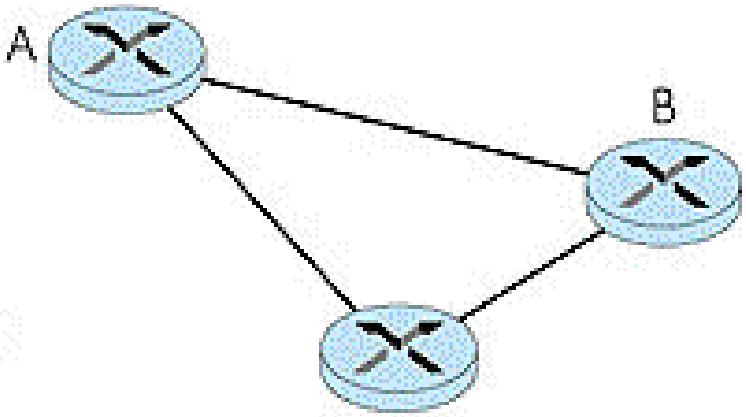
- Primeiro e o mais difundido.
- Implementa árvores baseadas nas origens com envio pelo caminho reverso, poda e enxerto.
- Utiliza o algoritmo de vetor de distância para permitir que o roteador calcule o enlace de saída que se encontra no caminho mais curto até cada uma das origens possíveis.
- Também calcula a lista dos roteadores que estão abaixo dele para questões de poda.
- A mensagem de poda contém a duração da poda (com valor default de 2 horas) após o qual o ramo é automaticamente enxertado na árvore.
- Uma mensagem de enxerto força a reinclusão de um ramo que tenha sido podado anteriormente da árvore multicast

Implantação de roteamento Multicast na Internet

- O ponto crucial é que apenas uma pequena fração dos roteadores estão aptos ao Multicast.
- Tunelamento pode ser usado para criar uma rede virtual de roteadores com multicast.
  - Esta abordagem foi utilizada no Mbone



Topologia física



Topologia lógica

PIM - *Protocol Independent Multicast*

- Considera dois tipos de cenários:
  - **Modo denso**: os membros de um grupo estão concentrados numa dada região. A maior parte dos roteadores devem se envolver com o roteamento dos datagramas de multicast.
  - **Modo esparso**: os membros de um grupo estão muito dispersos geograficamente.
- Conseqüências:
  - No modo denso: todos os roteadores devem ser envolvidos com o multicast. Uma abordagem como a de encaminhamento pelo caminho reverso é adequada.
  - No modo esparso: o default é que o roteador não se envolva com multicast. Os roteadores devem enviar mensagens explicítas solicitando a sua inclusão.

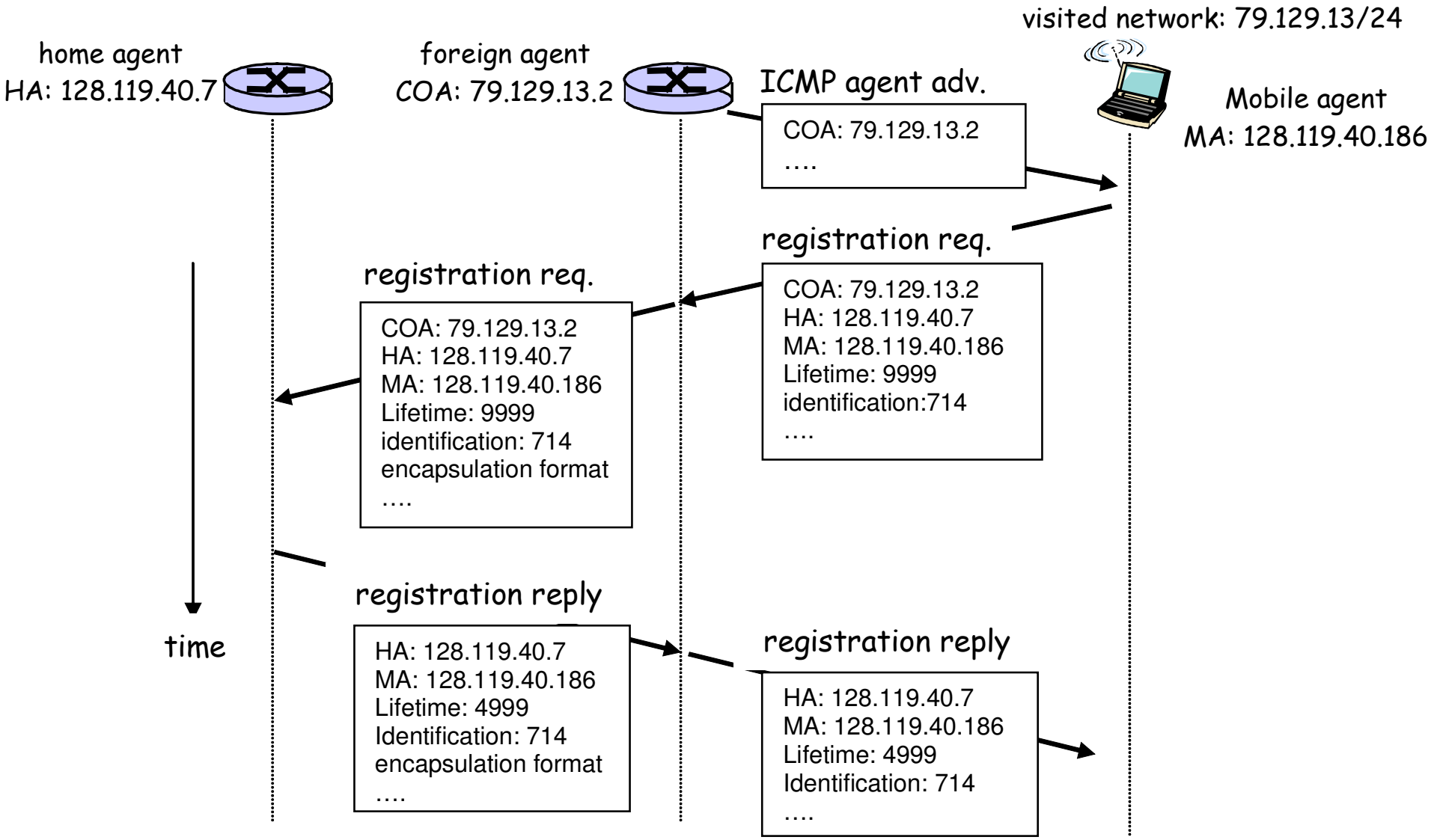
Roteamento Multicast entre Sistemas Autônomos

- Cada SA pode utilizar um protocolo de roteamento multicast diferente.
- Ainda não existe um padrão para o roteamento multicast inter-SA.
- O padrão *de fato* tem sido o DVMRP que não é adequado por ser um protocolo do tipo modo denso, enquanto que os roteadores multicast atuais estão espalhados.

Fatores de avaliação de protocolos multicast

- Escalabilidade: como cresce a quantidade de info de estados com o crescimento do número de grupos e dos transmissores de um grupo?
- Dependência do roteamento unicast: Ex.: MOSPF x PIM.
- Recepção excessiva (não necessária) de tráfego.
- Concentração de tráfego: a árvore única concentra tráfego em poucos enlaces.
- Optimalidade dos caminhos de envio.

Mobile IP: exemplo de registro



Capítulo 4: Resumo

- Iniciamos a nossa jornada rumo ao núcleo da rede.
- Roteamento dos datagramas: um dos maiores desafios da camada de rede.
  - Particionamento das redes em SAs.
  - Problema de escala pode ser resolvido com a hierarquização.
- Capacidade de processamento dos roteadores:
  - As tarefas dos roteadores devem ser as mais simples possíveis.
- Princípios dos alg. de roteamento:
  - Abordagem centralizada
  - Abordagem descentralizada
- Assuntos avançados:
  - IPv6
  - Roteamento multicast
- Próximo capítulo:
  - Camada de Enlace: transferência de pacotes entre nós no mesmo enlace ou LAN.