Redes de Computadores I

Prof. Ronaldo T. Oikawa

Camada de Rede

Camada De Rede

Objetivos

- Explicar as funções da camada de rede
  - Roteamento (esc. Caminho)
  - Escalabilidade
  - Como funciona um roteador
  - Tópicos avançados: ipv6, multicast
- Instanciação e implementação na intenet

Sumário:

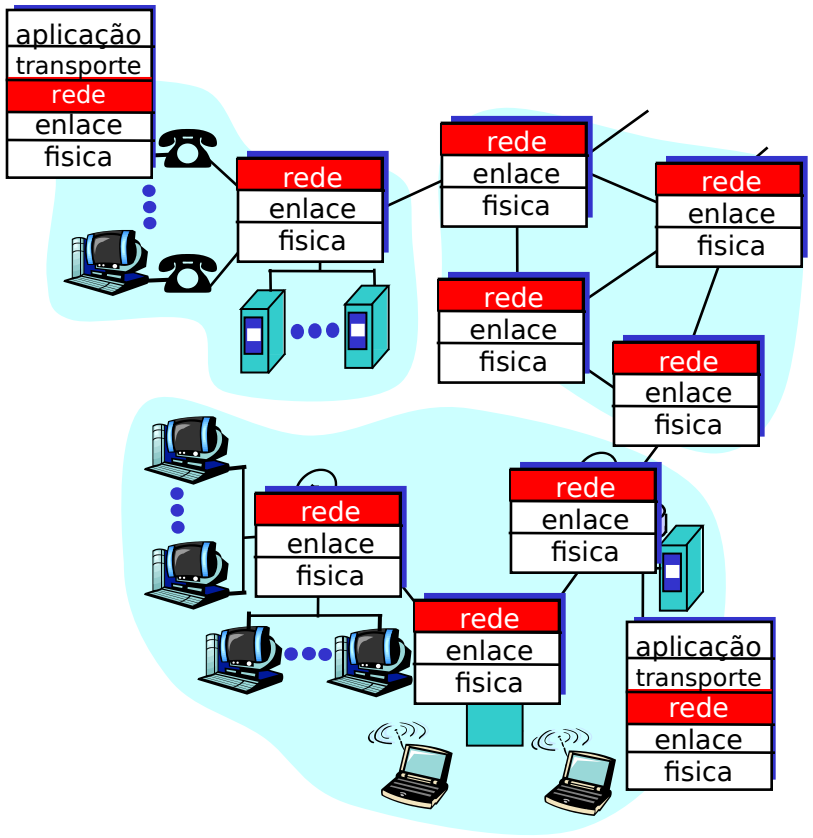
- Serviços da camada de rede
- Routeamento: seleção de rotas
- Roteamento hierárquico
- Ip
- Protocolos de roteamento da internet
  - Intra-domain
  - Inter-domain
- Como funciona um roteador IP
- Ipv6
- Roteamento multicast

Funções Da Camada De Rede

- Transportar pacotes entre os sistemas finais da rede
- A camada de rede deve ter uma entidade em cada sistema final ou roteador da rede

3 funções importantes:

- *Determinação de caminhos:* rota escolhida pelos pacotes entre a origem e o destino. *Algoritmos de roteamento*
- *Comutação:* mover pacotes entre as portas de entrada e de saída dos roteadores
- *Estabelecimento de conexão:* algumas arquiteturas de rede exigem o estabelecimento de circuitos virtuais antes da transmissão de dados



Modelo Do Serviço De Rede

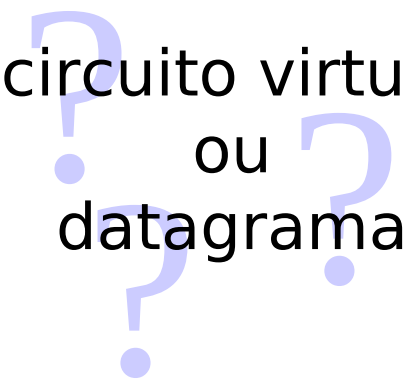
Q: como escolher um *modelo de serviço* para o canal transportando pacotes da origem ao destino?

abstração de serviço

- Banda-passante garantida?
- Preservação dos intervalos entre pacotes?
- Entrega sem perdas?
- Entrega em ordem?
- Realimentação de informação de congestionamento?

Nível mais geral de abstração na camada de rede

circuito virtual
ou
datagrama



Circuitos Virtuais (VC)

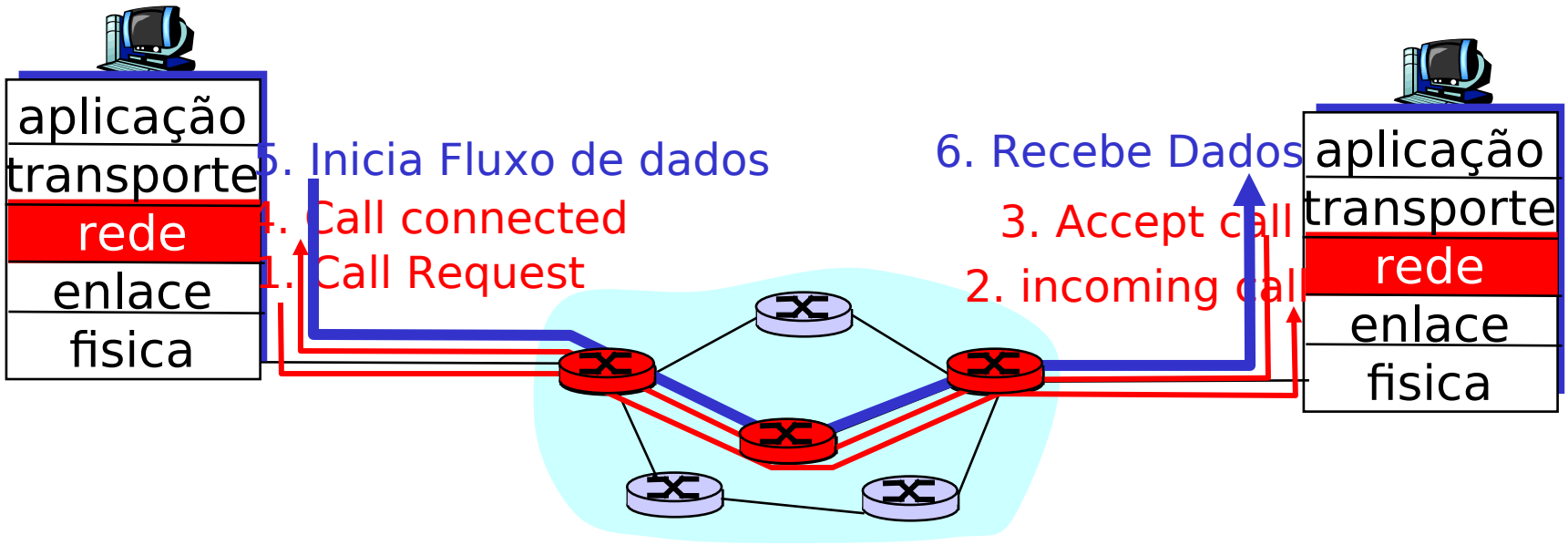
"A ligação entre a origem e o destino emula uma ligação telefônica"

- Orientado ao dsempenho
- A rede controla a conexão entre a origem e o destino

- Estabelecimento da conexão deve proceder o envio de dados. Liberação da conexão após os dados.
- Cada pacote transporte um identificador do CV, não transporta o endereço completo do destino
- Cada roteador na rota mantém informação de estado para conexão que passa por ele.
  - A conexão de camada de transporte envolve apenas os sistemas finais
- A banda passante e os recursos do roteador podem ser alocado por VC
  - Controle de Qualidade de Serviço por VC

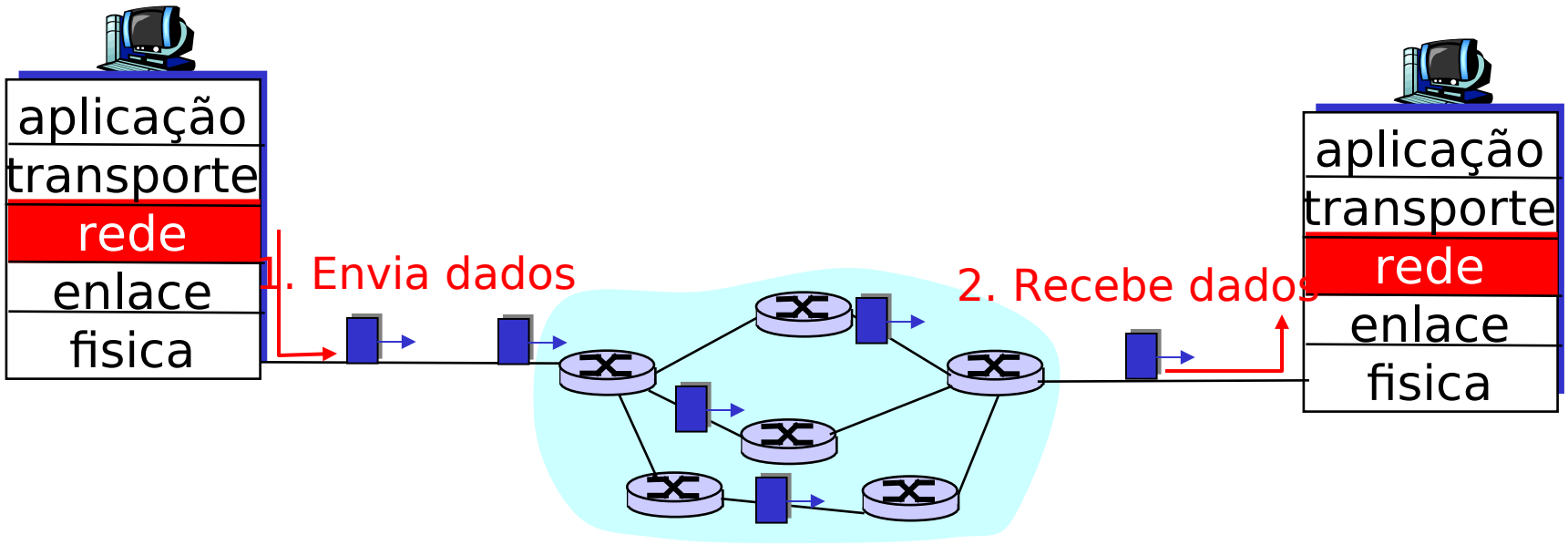
Circuitos Virtuais: Sinalização

- Usado para estabelecer, manter e encerrar Circuitos Virtuais
- Usados em ATM, Frame-Relay e X-25, mas não na Internet



Redes Datagrama: o modelo da Internet

- Não existem conexões na camada de transporte
- Não há informação de estado de conexão nos roteadores
  - Não existe conexão na camada de rede
- Pacotes tipicamente transportam o endereço de destino
  - Pacotes para o mesmo destino podem seguir diferentes rotas



Modelos de Serviço da Camada de Rede:

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

- Novos serviços na Internet: Intserv, Diffserv

Datagrama versus Circuito Virtual

Internet

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

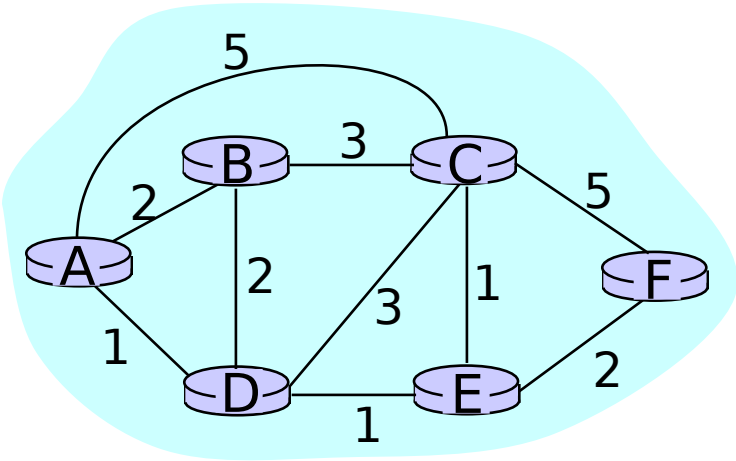
ATM

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

Roteamento

Protocolo de Roteamento

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



Algoritmos de roteamento são descritos por grafos:

- Nós do gráfico são roteadores
- Arestas do gráfico são enlaces
  - Custo do enlace: atraso, preço ou nível de congestão

- “bons” caminhos:
  - tipicamente corresponde aos caminhos de menor custo
  - caminhos redundantes

Classificação dos Algoritmos de Roteamento

Informação global ou descentralizada

Global:

- Todos os roteadores tem informações completas da topologia e do custos dos enlaces
- algoritmos “Link state”

Descentralizada:

- Roteadores só conhecem informações sobre seus vizinhos e os enlaces para eles
- Processo de computação interativo, troca de informações com os vizinhos
- algoritmos “Distance vector”

Estático ou Dinâmico?

Estático:

- As rotas mudam lentamente ao longo do tempo

Dinâmico:

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

Algoritmo Link-state

Algoritmo de Dijkstra's

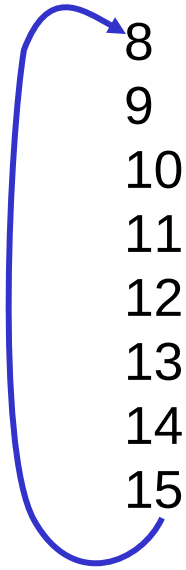
- Topologia de rede e custo dos enlaces são conhecidos por todos os nós.
  - Implementado via "link state broadcast"
  - Todos os nós têm a mesma informação
- Computa caminhos de menor custo de um nó (fonte) para todos os outros nós
  - Fornece uma tabela de roteamento para aquele nó
- Convergência: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

Notação:

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

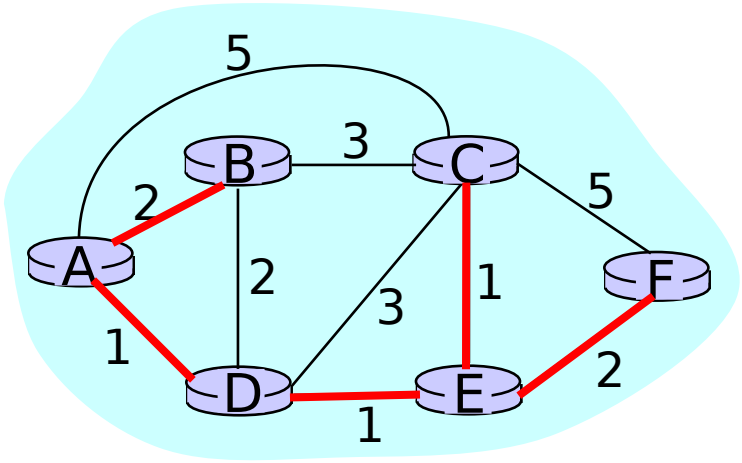
Algoritmo de Dijsktra's

```
1  Inicialização:
2    N = {A}
3    para todos os nós v
4      se v é adjacente a  A
5        então D(v) = c(A,v)
6        senão D(v) = infty
7
8   Loop
9     ache w não em N tal que D(w) é um mínimo
10    acrescente w  a N
11    atualize D(v) para todo v adjacente a w e não em N:
12       D(v) = min( D(v), D(w) + c(w,v) )
13    /* novo custo para v é ou o custo anterior para v ou o menor
14     custo de caminho conhecido para w mais o custo de w a v */
15  até que todos os nós estejam em N
```



Exemplo: Algoritmo de Dijkstra's

Passo	início N	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					



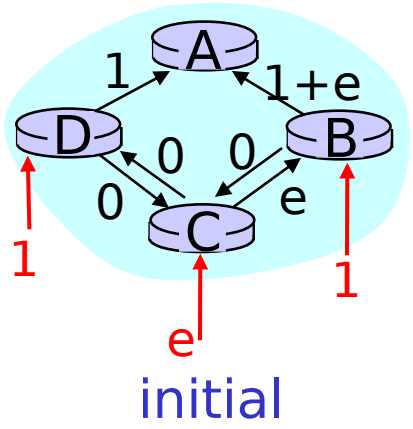
Discussão do Algoritmo de Dijkstra

Complexidade do Algoritmo: n nós

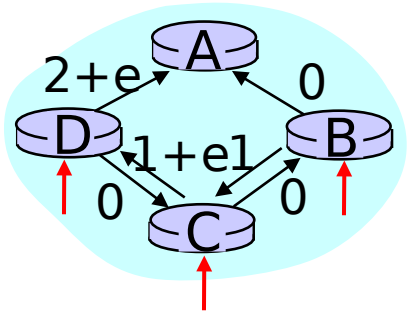
- Cada iteração: precisa verificar todos os nós w, que não estão em N
- N*(n+1)/2 comparações: o(n**2)
- Implementações mais eficientes: o(nlogn)

Oscilações possíveis:

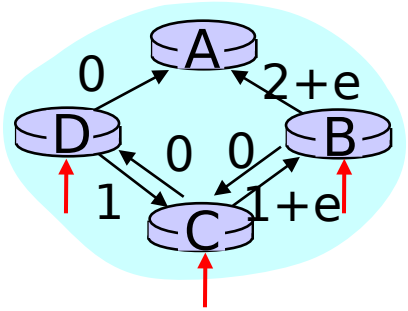
- E.G., custo do enlace = total de tráfego transportado



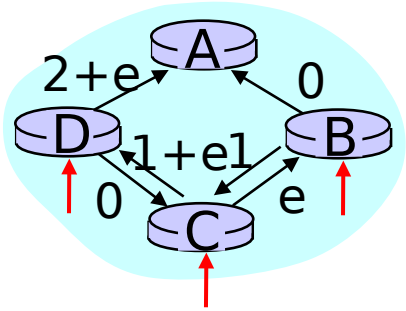
initial



... recalcula roteamento



... recalcula



... recalcula

Algoritmo “Distance Vector”

Iterativo:

- Continua até que os nós não troquem mais informações.
- *Self-terminating*: Não há sinal de parada

Assíncrono:

- Os nós não precisam trocar informações simultaneamente!

Distribuído:

- Cada nós se comunica apenas com os seus vizinhos, diretamente conectados

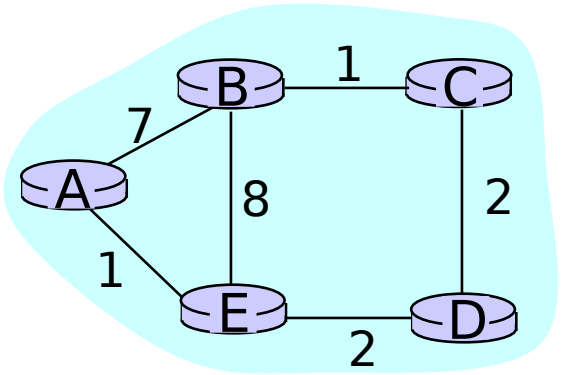
Estrutura de Dados da Tabela de Distância

- Cada nó tem sua própria tabela
- Linha para cada possível destino
- Coluna para cada roteador vizinho
- Exemple: no nó X, para destino Y via vizinho Z:

$$D^{X}(Y,Z) = \text{distância } de \text{ X } to \text{ Y, } via \text{ Z como prox. salto}$$

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

Exemplo de Tabela de Distância



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

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

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

custo via nó vizinho

destino	$D^{E}()$ A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

A Tabela de Distâncias Gera a Tabela de Roteamento

custo através de

$D^E()$	A	B	D
destino A	1	14	5
destino B	7	8	5
destino C	6	9	4
destino D	4	11	2

	Enlace de saída, cost
destino A	A,1
destino B	D,5
destino C	D,4
destino D	D,4

Tabela de distância → Tabela de Roteamento

Roteamento Vetor-Distância: Resumo

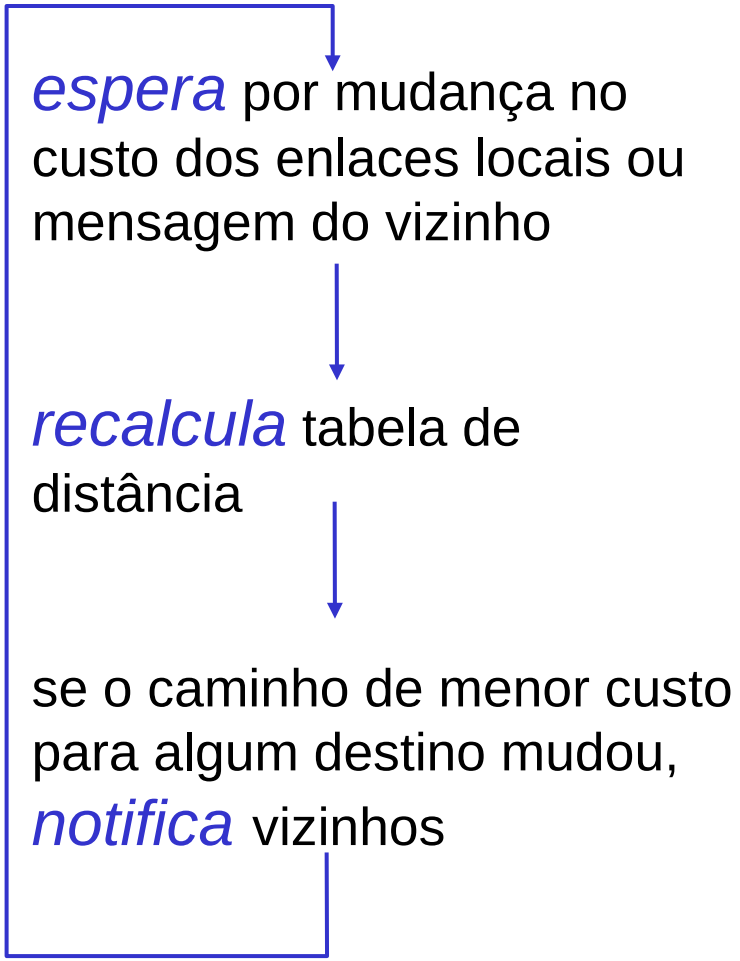
Iterativo, assíncrono:
cada iteração local é causada por:

- Mudança de custo dos enlaces locais
- Mensagem do vizinho: seu caminho de menor custo para o destino mudou

Distribuído:

- Cada nó notifica seus vizinhos apenas quando seu menor custo para algum destino muda
  - Vizinhos notificam seus vizinhos e assim por diante

Cada nó:



Algoritmo Vetor-Distância:

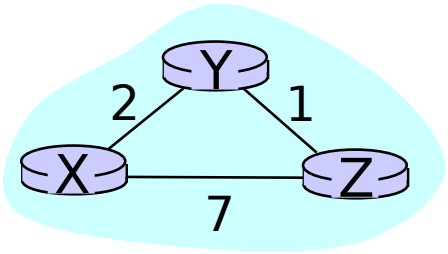
Para todos os nós, X:

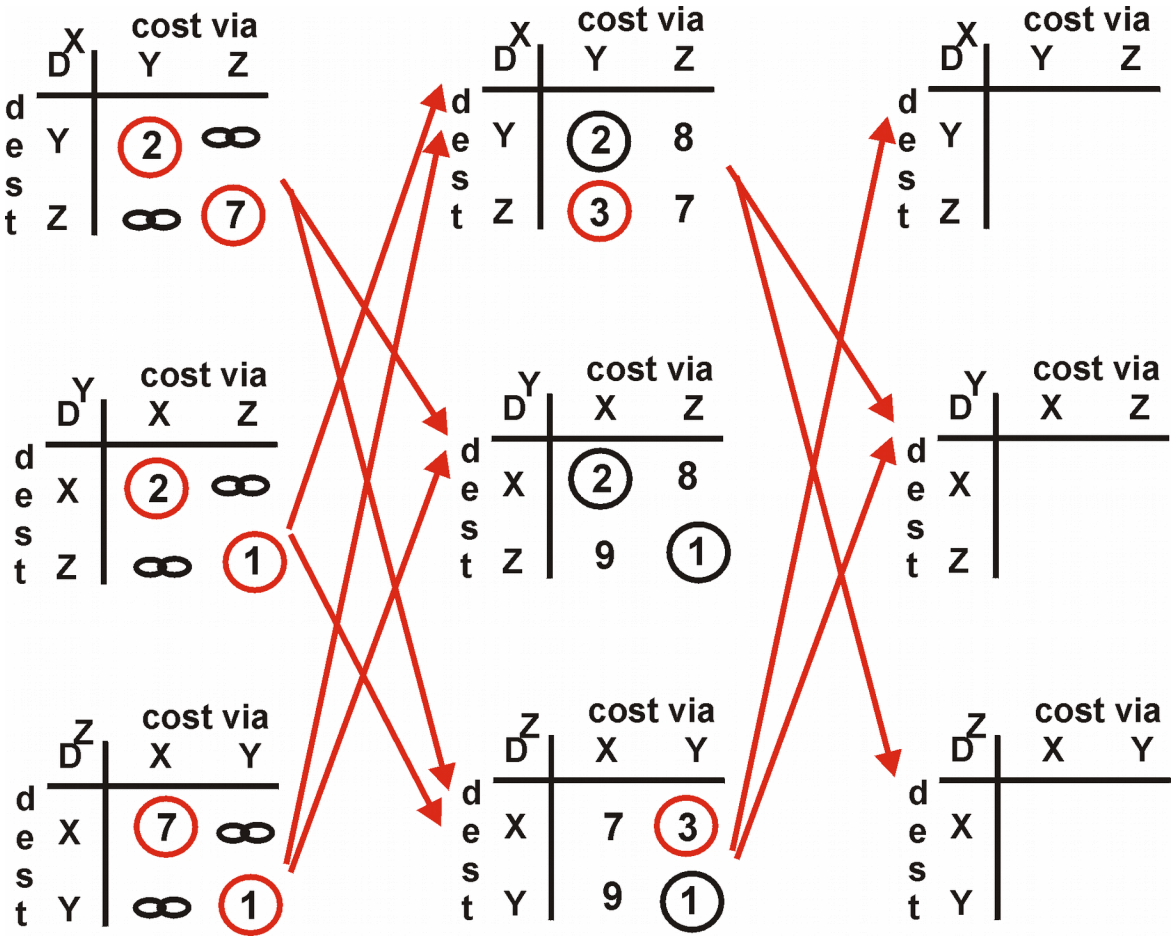
1 Initialização:
2 para todos os nós adjacentes v:
3 $D^X(*,v) = infinito$ /* o operador * significa ”para todas as colunas" */
4 $D^X(v,v) = c(X,v)$
5 para todos os destinos, y
6 envia $\min_w D^X(y,w)$ para cada vizinho /* w sobre todos vizinhos de X*/

Algoritmo Vetor-Distância (Cont.):

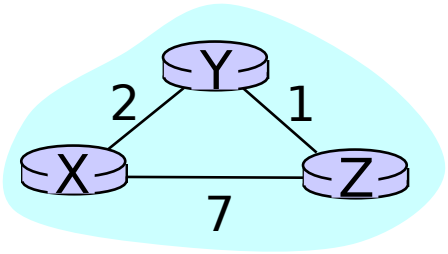
```
8  loop
9    wait (até ocorrer uma mudança no custo do enlace para vizinho V
10         ou até receber atualização do vizinho V)
11
12   if (c(X,V) muda por d)
13     /* muda o custo para todos os destinos via vizinho v por d */
14     /* nota: d pode ser positivo ou negativo */
15     para todos os destinos y:  D^X(y,V) =  D^X(y,V) + d
16
17   else if (atualização recebida de  V sobre destino  Y)
18     /* caminho mais curto de V para algum Y mudou  */
19     /* V enviou um novo valor para seu min_w D^v(Y,w) */
20     /* chame este novo valor recebido  "newval"     */
21     para o único destino y: D^X(Y,V) = c(X,V) + newval
22
23   if nós temos um novo min_w D^X(Y,w) para algum destino Y
24      envie novo valor de  min_w D^X(Y,w) para todos os vizinhos
25
26 forever
```

Exemplo: algoritmo vetor-distância





Exemplo: algoritmo vetor-distância



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

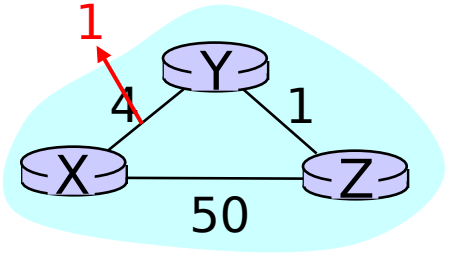
$$D^X(Y,Z) = c(X,Z) + \min_w\{D^Z(Y,w)\}$$
$$= 7+1 = 8$$

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

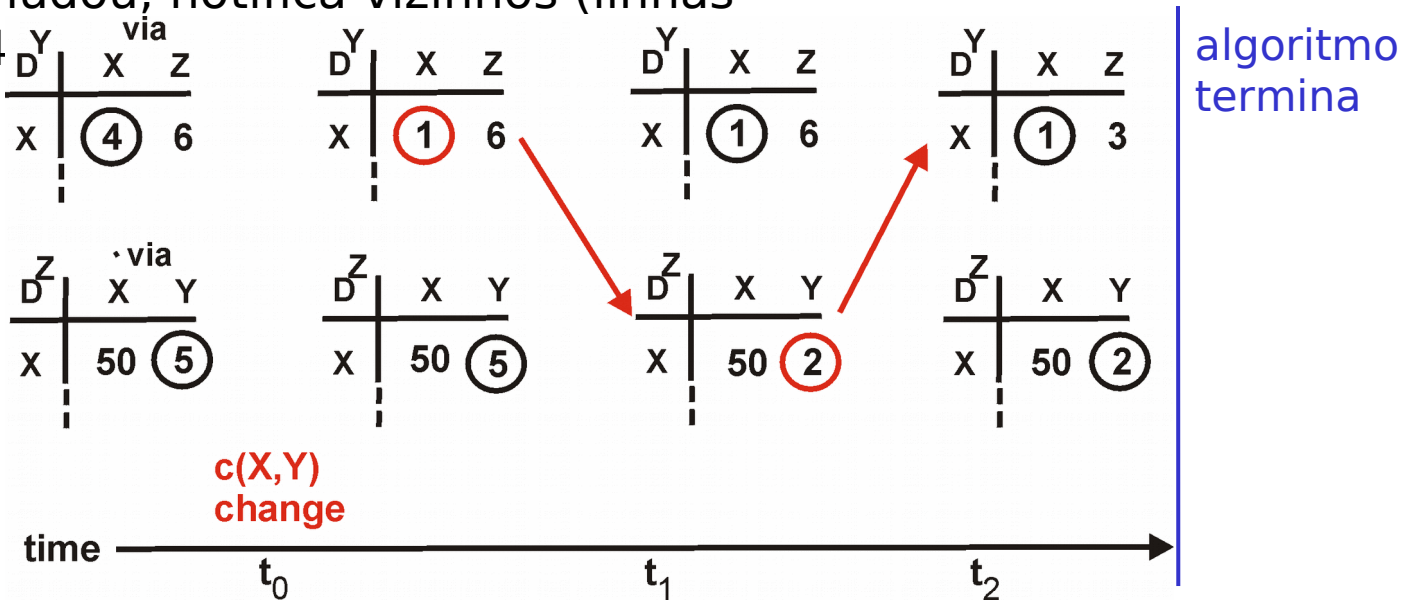
Vetor-Distância: Mudança no custo do enlace

Mudança no custo do enlace:

- nó detecta que o custo do enlace local mudou
- atualiza tabela de distâncias (linha 15)
- se o custo do caminho de menor custo mudou, notifica vizinhos (linhas 23 e 24



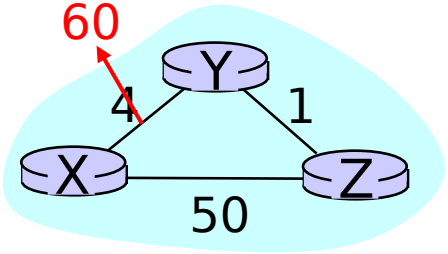
"boas notícias viajam depressa"

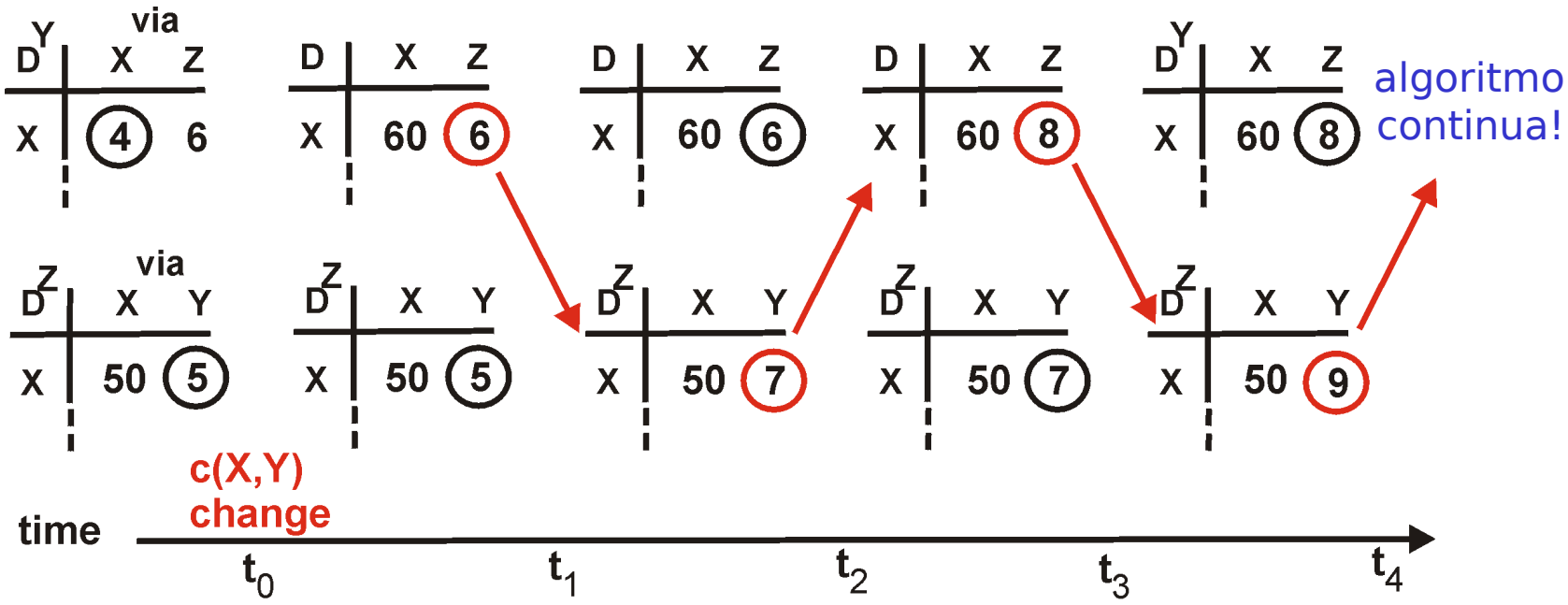


Vetor Distância: Mudança no custo do enlace

Mudança no custo do enlace:

- más notícias viajam devagar - problema da contagem ao infinito

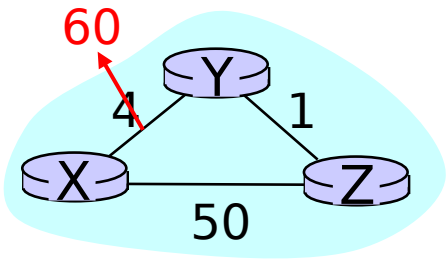


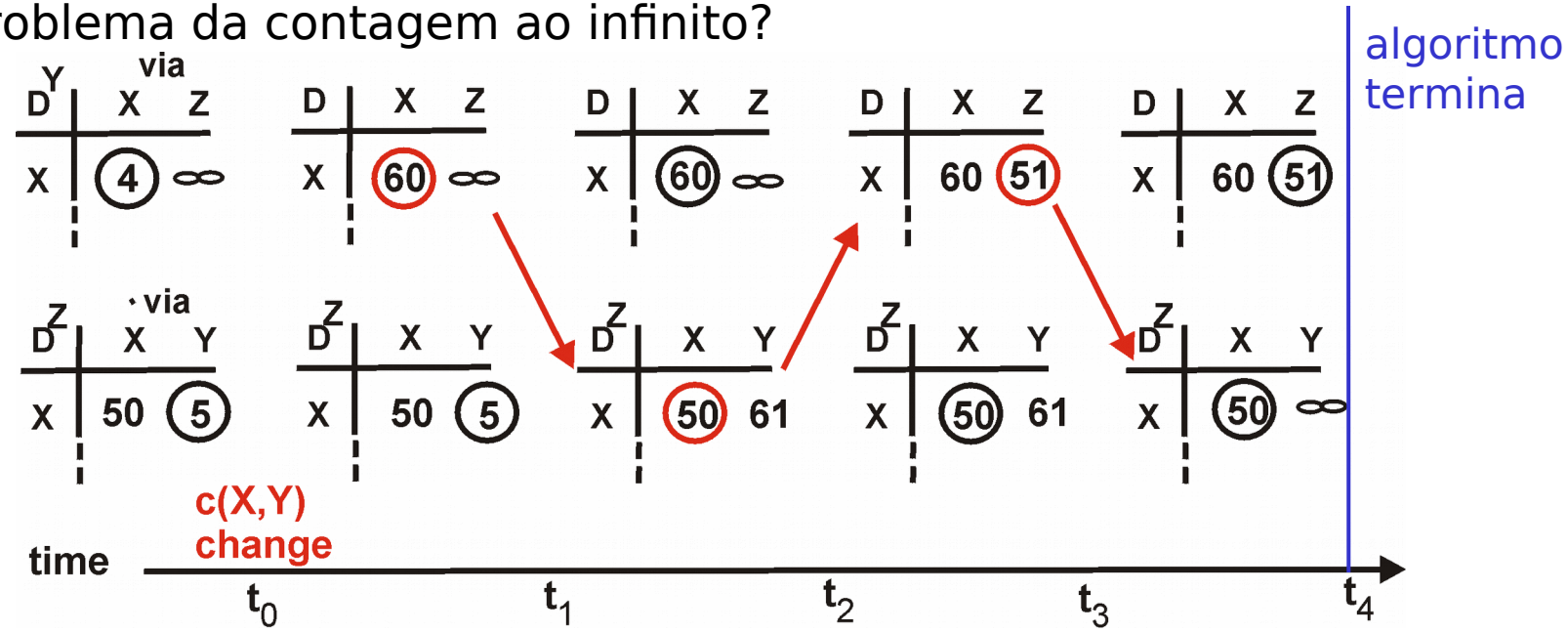


Vetor Distância: Poisoned Reverse

Se Z roteia através de Y para chegar a X :

- Z diz a Y que sua (de Z) distância para X é infinita (assim Y não roteará para X via Z)
- será que isso resolve completamente o problema da contagem ao infinito?





Comparação dos Algoritmos LS e VD

Complexidade

- LS: com n nós, E links, o(ne) mensagens enviadas
- DV: trocas somente entre vizinhos
  - Tempo de convergência varia

Tempo de convergência

- LS: algoritmo o(n**2) exige o(ne) msgs
  - Pode ter oscilações
- DV: tempo de convergência varia
  - Podem haver loops de roteamento
  - Problema da contagem ao infinito

Robustez: o que acontece se um roteador funciona mal?

Ls:

- Nós podem advertir custos incorretos para os enlaces.
- Cada nó calcula sua própria tabela de roteamento

Dv:

- Nó pode advertir caminhos com custo incorreto
- Tabela de cada nó é usada por outros
  - Propagação de erros pela rede

Roteamento Hierárquico

Problemas do mundo real

- roteadores não são todos idênticos
- as redes não são “flat” na prática

Escala: com 50 milhões de destinos:

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

Autonomia Administrativa

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

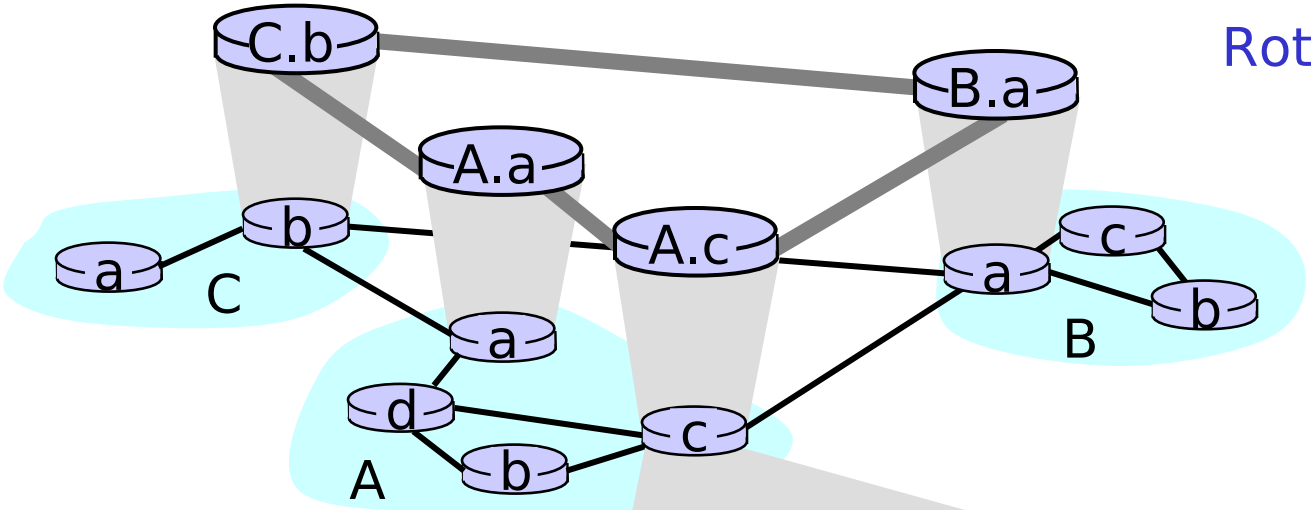
Roteamento Hierárquico

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

roteadores de borda

- Roteadores de interface de um AS
- Rodam protocolos de roteamento intra-as com os outros roteadores do AS
- *Também* responsáveis por enviar mensagens para fora do AS
  - Rodam *protocolo de roteamento inter-as* com outros rotea-dores de borda

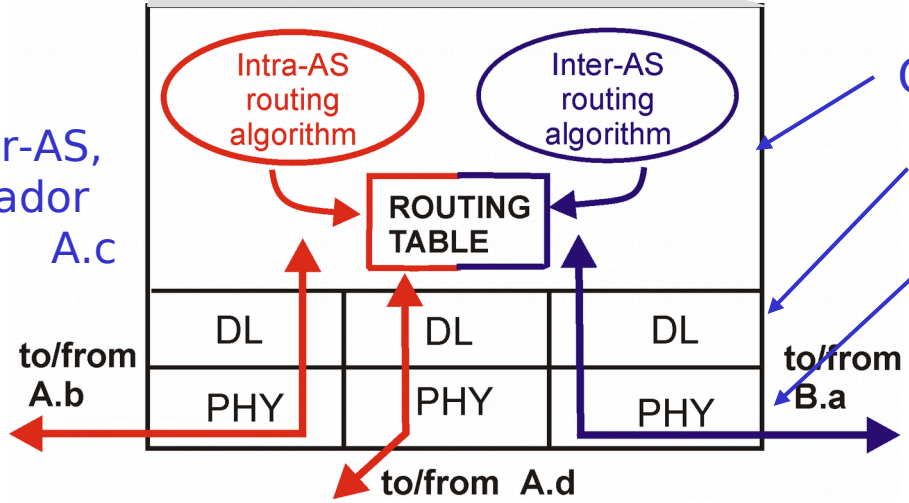
Roteamento Intra-as and Inter-as



Roteadores de Borda

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

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

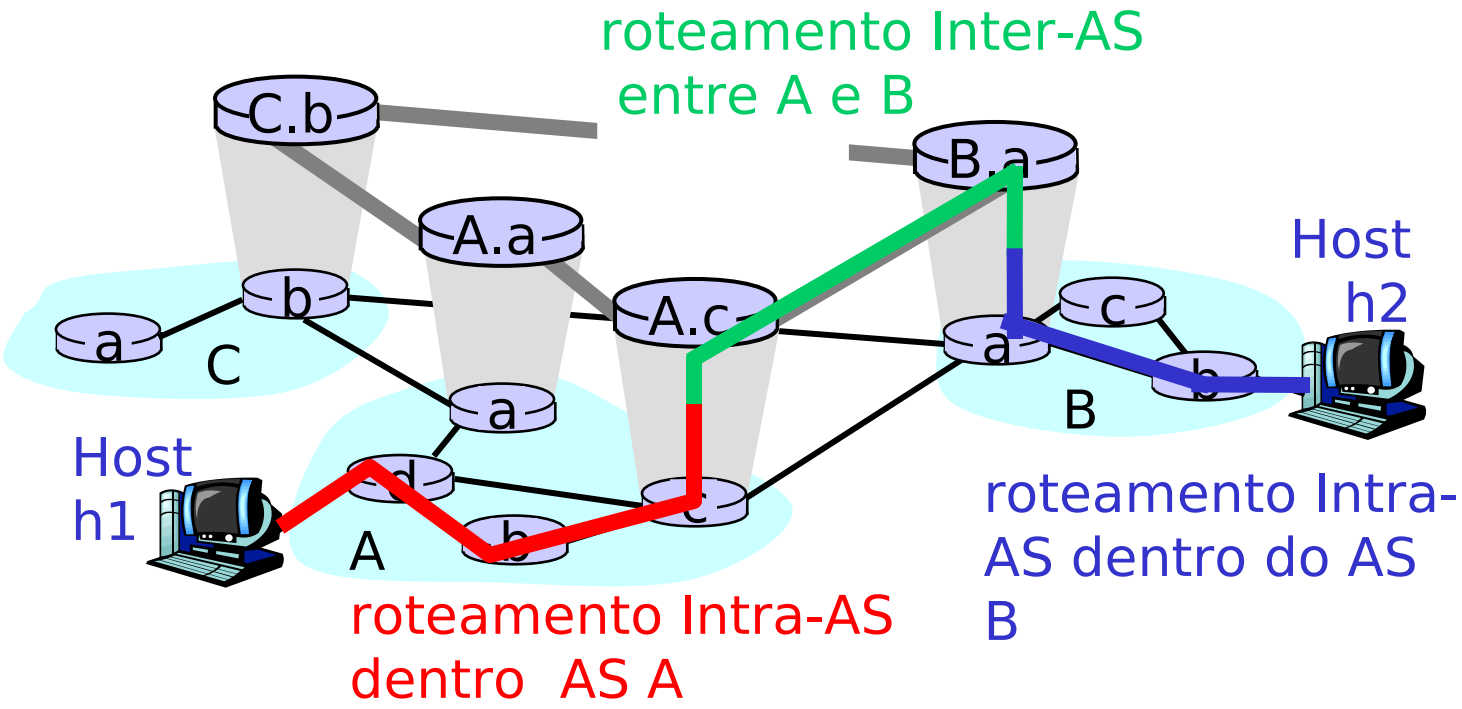


Camada de rede

Camada de enlace

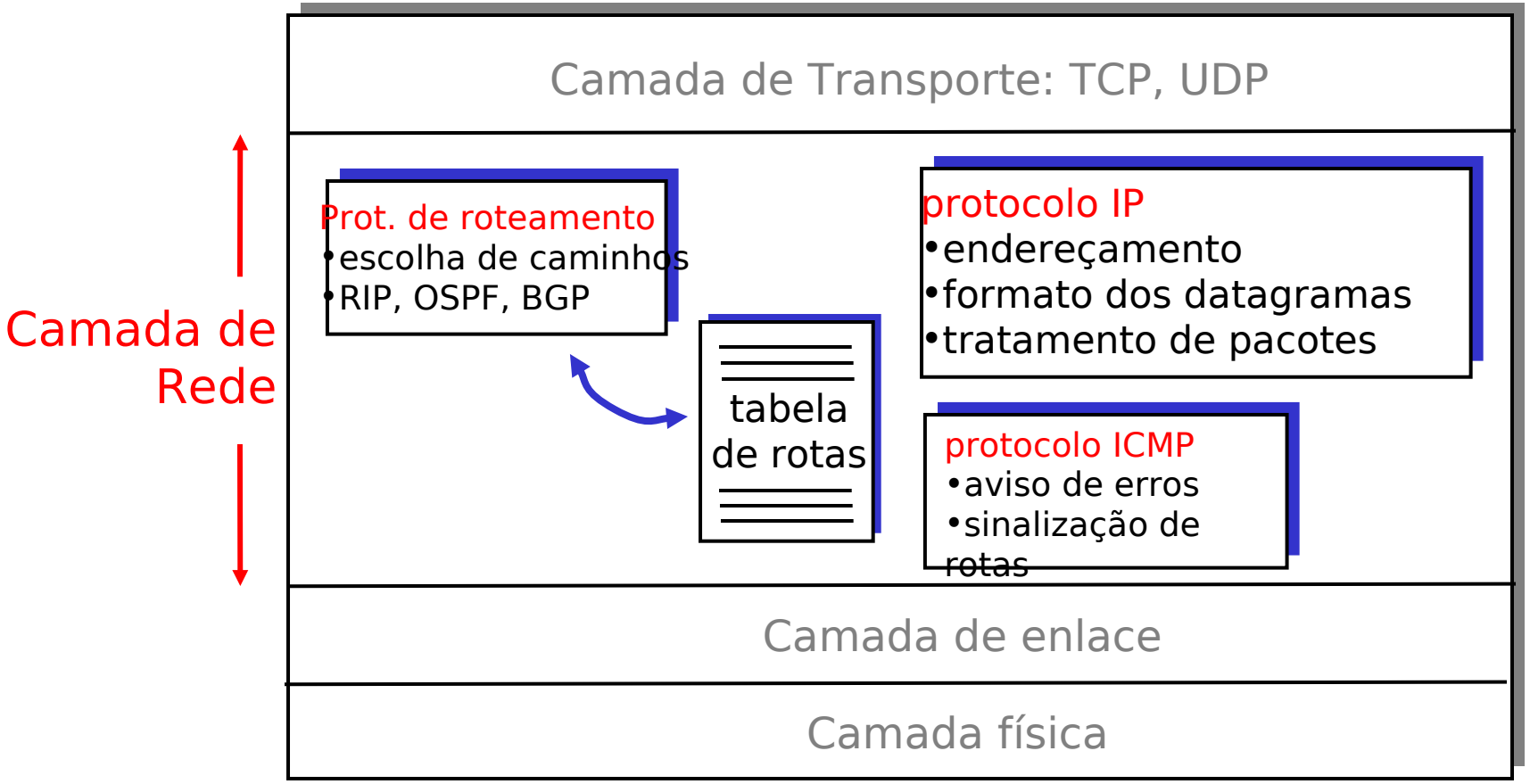
Camada fisica

Roteamento Intra-AS e Inter-AS



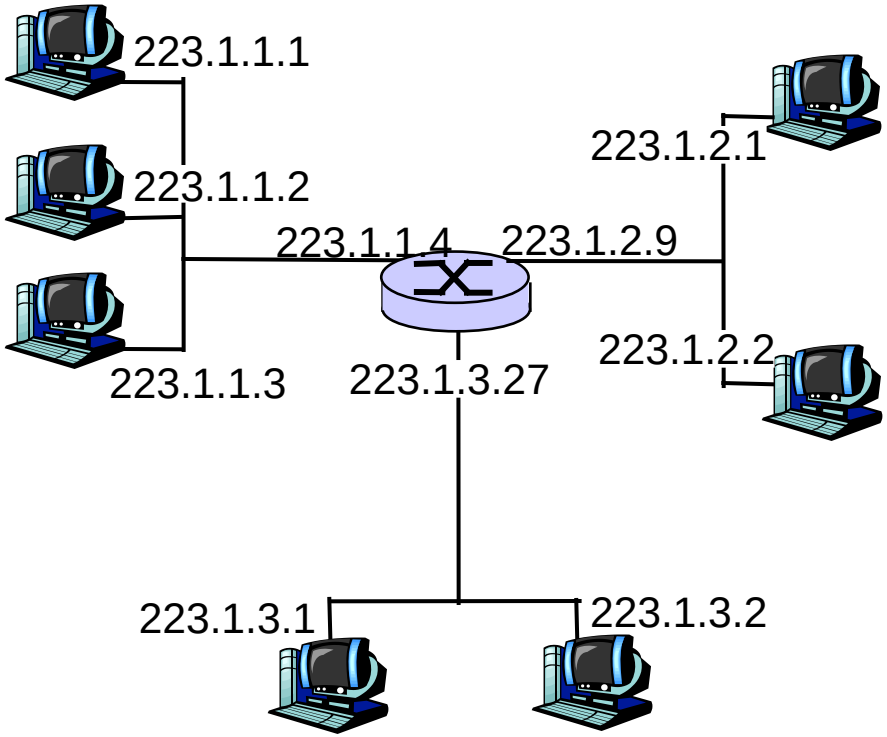
A camada de rede da Internet

Entidade de rede em roteadores ou hosts:



Endereçamento IP: Introdução

- endereço IP: identificador de 32-bits para *interfaces* de roteadores e hosts
- *Interface:* conexão entre roteador ou host e enlace físico
  - Roteador tem tipicamente múltiplas interfaces
  - Hosts podem ter múltiplas interfaces
  - endereços IP são associados com interfaces, não com o host ou com o roteador

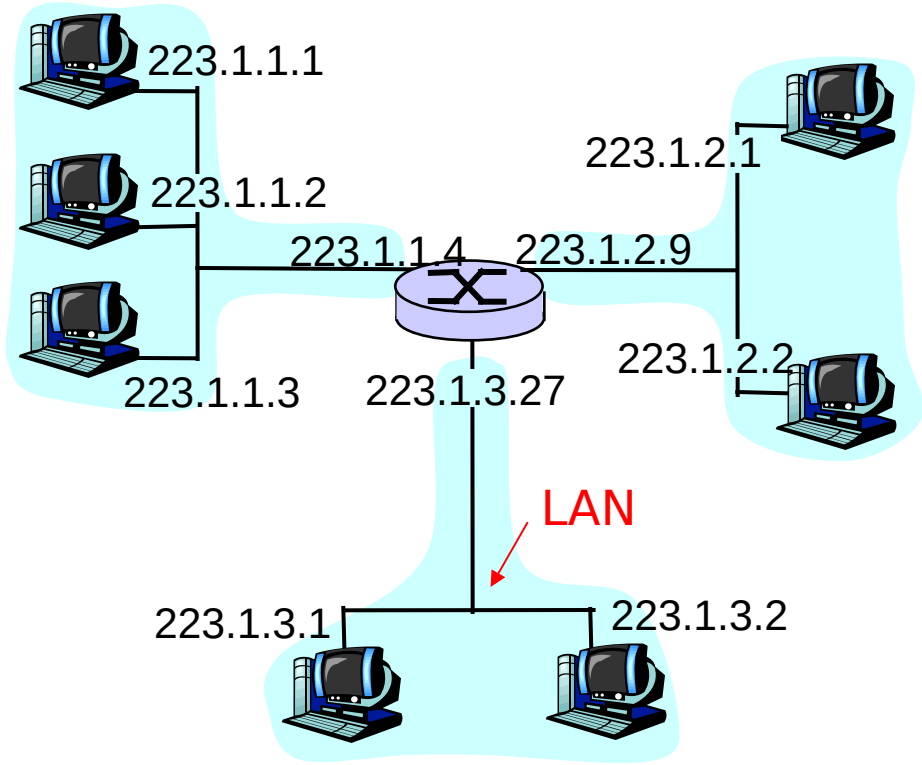


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Endereçamento IP

- Endereço IP:
  - parte de rede (bits mais significativos)
  - parte de Host part (bits menos significativos)
- *O que é uma rede?* (na prespectiva do endereço)
  - Interfaces de dispositivos com a mesma parte de rede no endereço IP
  - Podem fisicamente se comunicar sem o auxílio de um rotedor

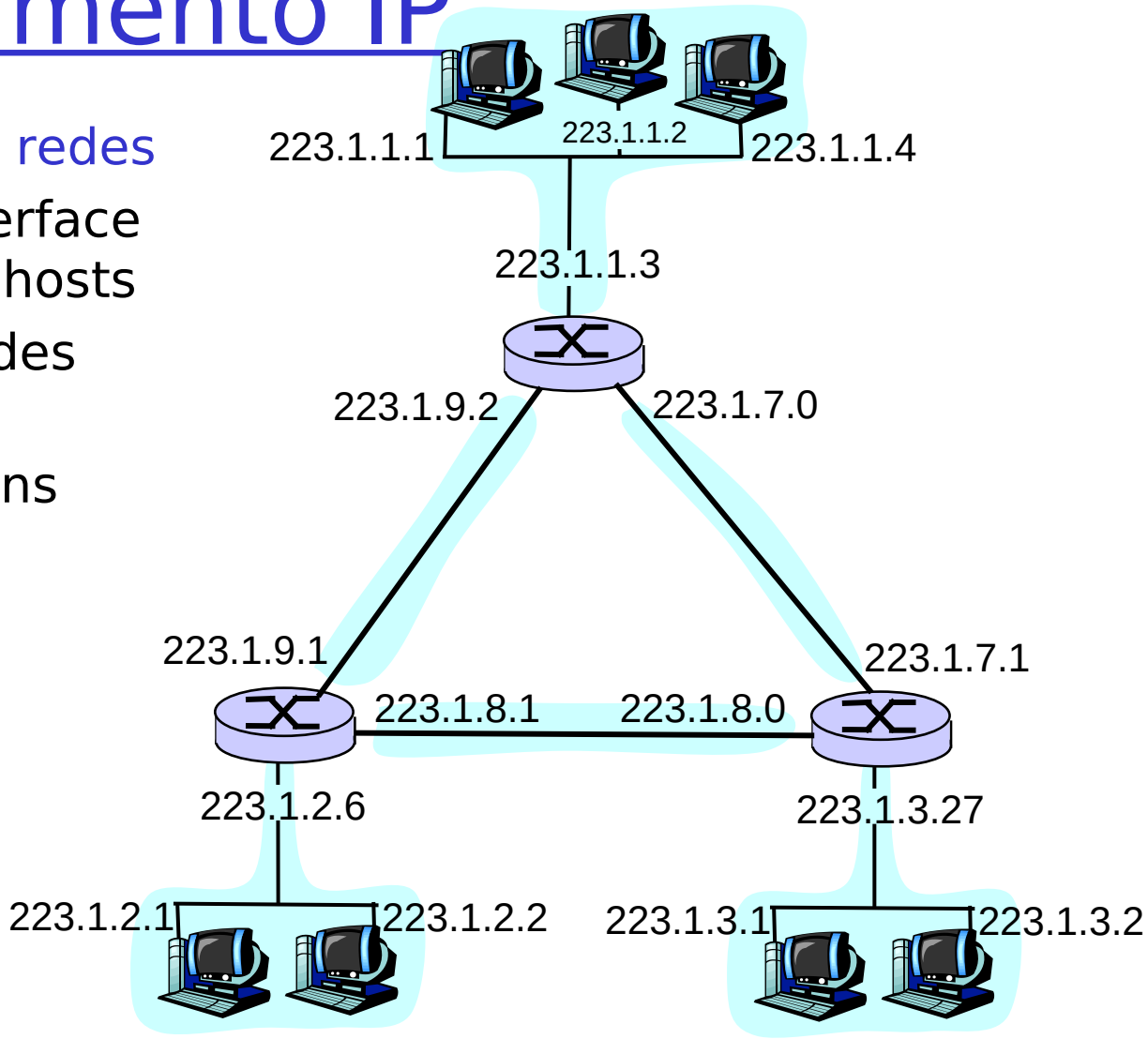


rede consistindo de de 3 redes IP (para endereços IP começando com 223 os primeiros 24 bits são o endereço de rede)

Endereçamento IP

Como encontrar as redes

- Separe cada interface de roteadores e hosts
- Criar ilhas de redes isoladas
- Técnica de nuvens

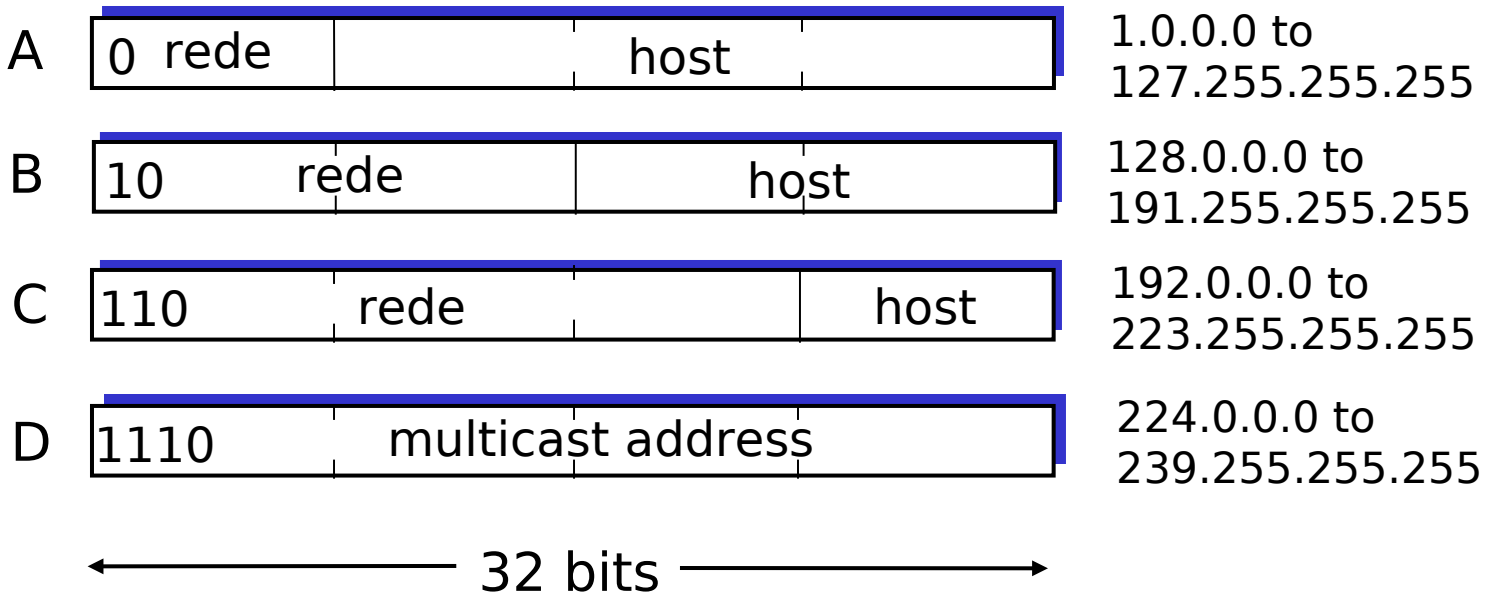


Sistema com seis redes interconectadas

Endereços IP

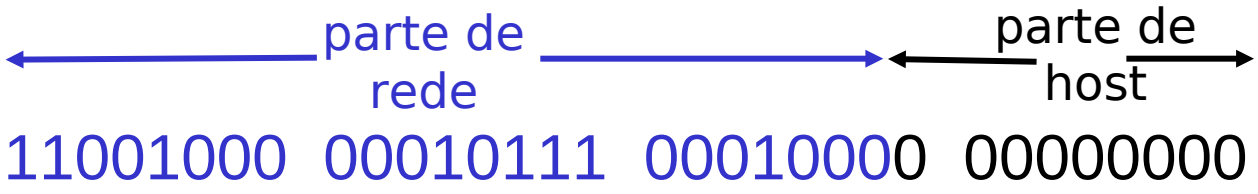
endereçamento “class-full”:

class



Endereçamento IP: CIDR

- Endereçamento “Classful”:
  - Uso ineficiente do espaço de endereçamento, exaustão do espaço de endereços
  - E.G., rede de Classe B aloca endereços para 65K hosts, mesmo se só existem 2000 hosts naquela rede
- CIDR: classless interdomain routing
  - A porção de endereço de rede tem tamanho arbitrário
  - Formato do endereço: a.B.C.D/x, onde x é o número de bits na parte de rede do endereço



200.23.16.0/23

Como obter um endereço IP

Hosts :

- Endereço fixo: definido pelo administrador
- DHCP: dynamic host configuration protocol: permite a atribuição dinâmica de endereços IP
  - Host envia (broadcast) mensagem "DHCP discover"
  - DHCP server responde com mensagem "DHCP offer"
  - Host pede endereço IP com mensagem : "DHCP request"
  - DHCP server envia endereço com a mensagem: "DHCP ack"

Como obter um endereço IP

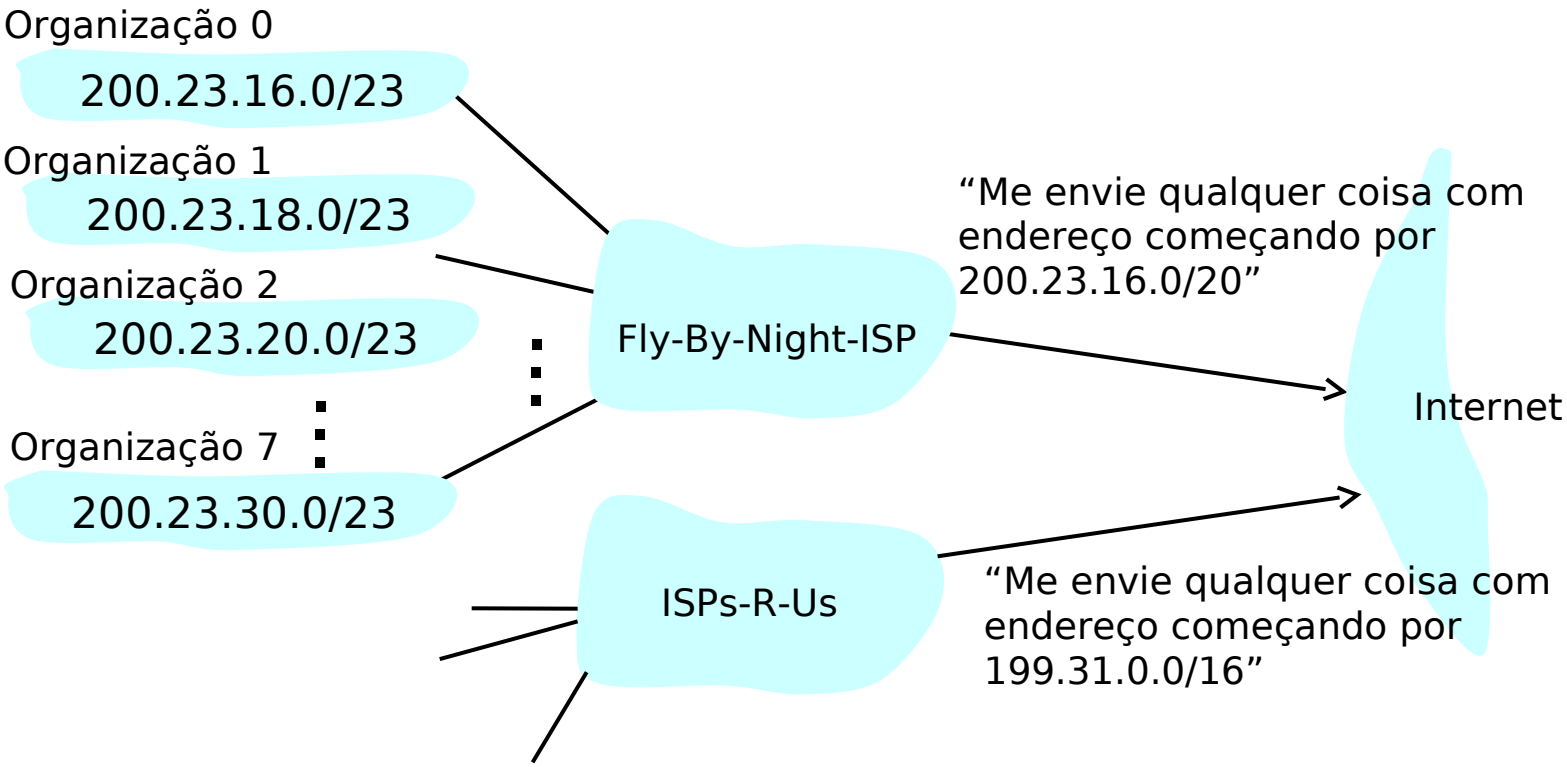
Rede (porção de rede)

- Obter uma parte do espaço de endereços do seu ISP:

bloco do ISP	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

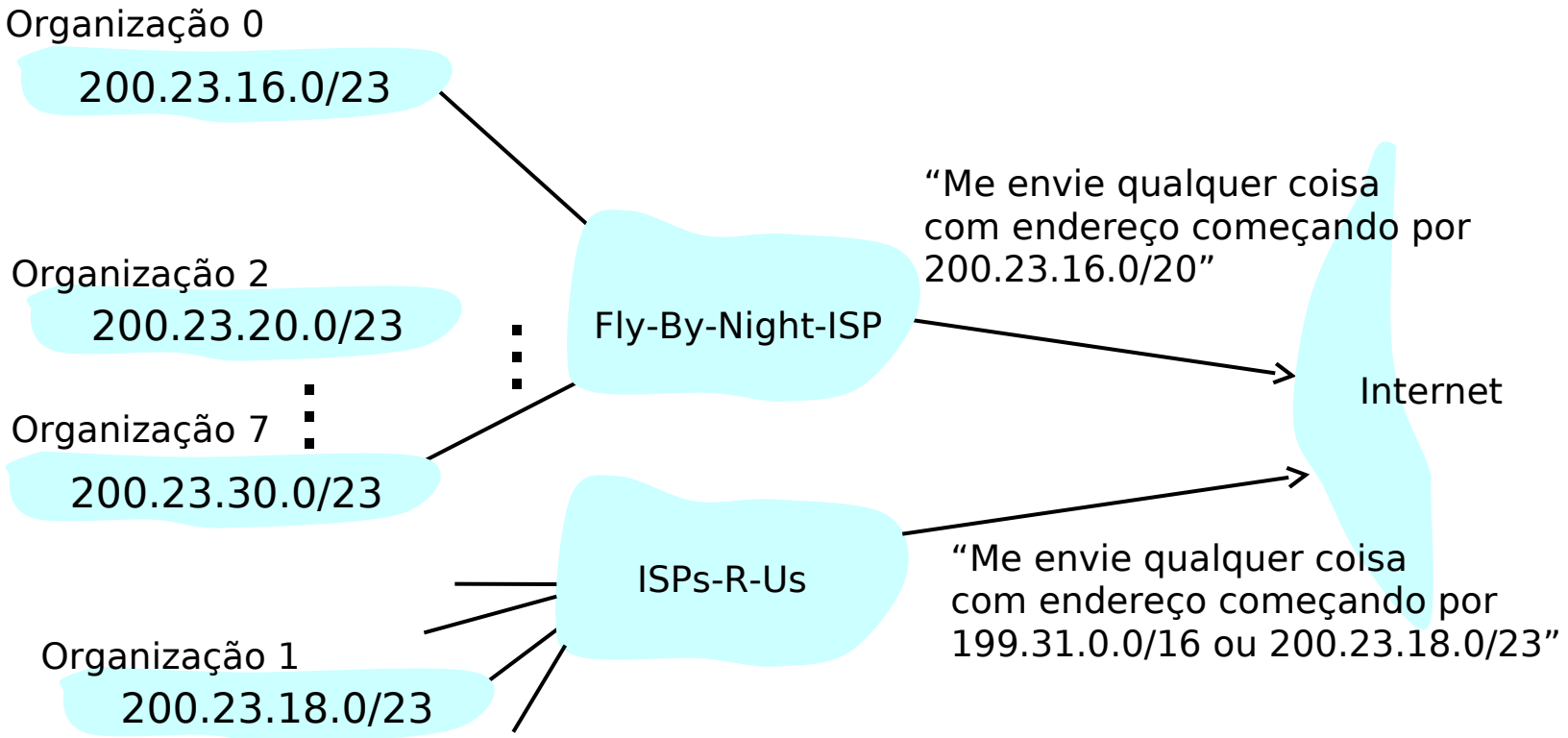
Endereçamento Hierárquico: agregação de rotas

O endereçamento hierárquico permite uma propagação de rotas mais eficiente:



Roteamento Hierárquico:rotas mais específicas

ISPs-R-Us tem uma rota mais específica para a organização 1



Como obter um endereço IP...

Q: Como o ISP obtém seu bloco de endereço?

A: ICANN: internet corporation for assigned names and numbers

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

Levando um Datagrama da Fonte ao Destino

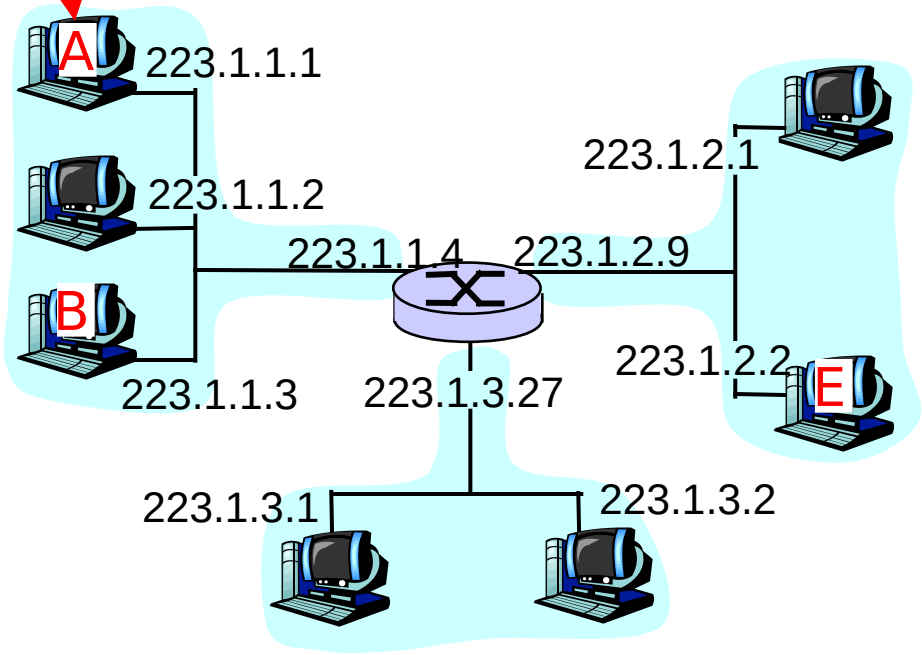
datagrama IP:

outros campos	endereço IP origem	endereço IP destino	dados

- os endereços do datagrama não mudam ao viajar da fonte ao destino

tabela de roteamento em A

Rede destino	próx. roteador	Núm. saltos
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



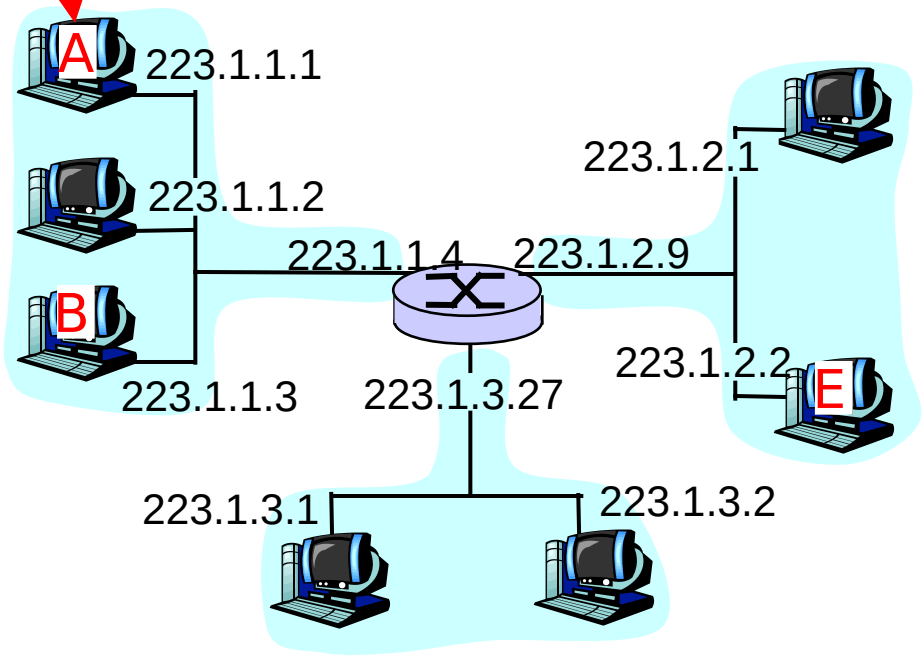
Levando um Datagrama da Fonte ao Destino

outros campos	223.1.1.1	223.1.1.3	dados

Começando em A, levar datagrama IP para B:

- examine endereço de rede de B
- descobre que B está na mesma rede de A
- camada de enlace envia datagrama diretamente para B num quadro da camada de enlace
- Se necessário descobre endereço físico de B
  - B e A são diretamente conectados

Rede destino	Próx. roteador	Núm. saltos
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



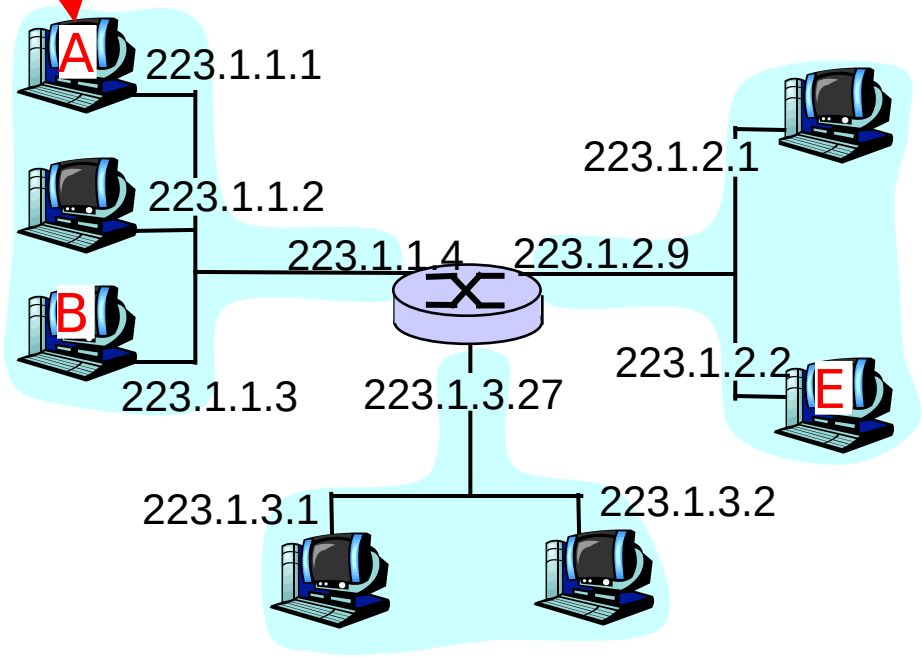
Levando um Datagrama da Fonte ao Destino

outros campos	223.1.1.1	223.1.2.3	dados

Começando em A, dest. E:

- examina endereço de rede de E
- E está num rede diferente
  - A, E não estão diretamente conectados
- tabela de roteamento: próximo roteador para E é 223.1.1.4
- encontra endereço físico de 223.1.1.4 e envia o datagrama num quadro de enlace
- datagrama chega em 223.1.1.4
- continua.....

Rede destino	Próx. roteador	Núm. saltos
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



Levando um Datagrama da Fonte ao Destino

outro campos	223.1.1.1	223.1.2.3	dados

Chegando em 223.1.1.4, destined for 223.1.2.2

- examina endereço de rede de E
- E está na mesma rede da interface 223.1.2.9 do roteador
  - roteador e E estão diretamente ligados
- descobre endereço físico de 223.1.2.2 e envia o datagrama num quadro da camada de enlace
- datagrama chega em 223.1.2.2!!! (ufa!)

Rede destino	Próx. roteador	Núm. saltos	Endereço 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