

Redes de Computadores

Camada de Redes

Kalinka Regina Lucas Jaquie Castelo Branco

Camada de Enlace

Aplicação
Apresentação
Sessão
Transporte
Rede
Enlace
Físico

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 internet

Sumário:

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

Revisando...

❑ Nível Físico:

- Descreve os procedimentos e características mecânicas, elétricas e funcionais. É responsável pela **transmissão de bits** de um ponto a outro.

❑ Nível de Enlace de Dados:

- Gerencia a transmissão, **detectando e corrigindo erros** na camada física, buscando obter um canal confiável (mais confiável do que ele já é). Separa as **mensagens em quadros**, inserindo aspectos como sincronização, controle de erro e controle de fluxo.

Camada de Rede - Funções

■ Nível de Rede:

- Estabelece, mantém e termina conexões lógicas, é responsável pela tradução de endereços lógicos ou nomes em endereços físicos(rooteamento).

ou

- Provê os meios funcionais e procedurais para a transmissão de dados orientada ou não-orientada à conexão entre entidades do nível de transporte

Camada de Rede

■ Funções:

- multiplexação
- endereçamento
- mapeamento entre endereços de rede e endereços de enlace
- roteamento
- estabelecimento e liberação de conexões de rede
- transmissão de unidades de dados do serviço de rede
- controle de congestionamento

Propriedades

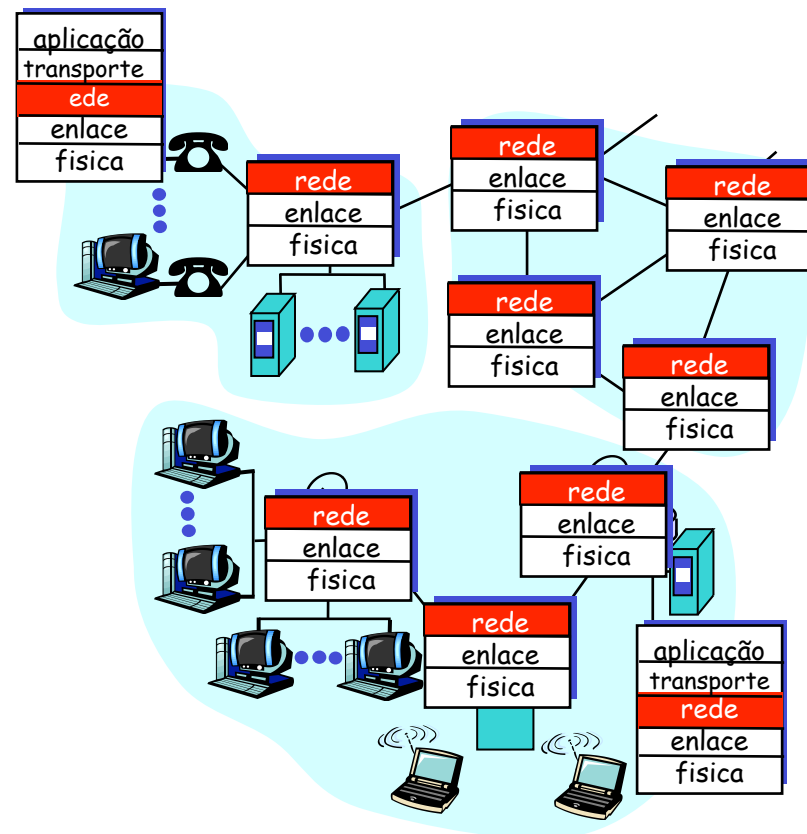
- ❑ Antes de enviar os dados, é ajustado uma conexão entre as partes.
- ❑ Assim que a conexão é estabelecida, as partes negociam os parâmetros, a qualidade e o custo do serviço oferecido.
- ❑ A comunicação flui nas duas direções, e os pacotes são enviados em seqüência.
- ❑ Controle de fluxo é fornecido automaticamente.

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

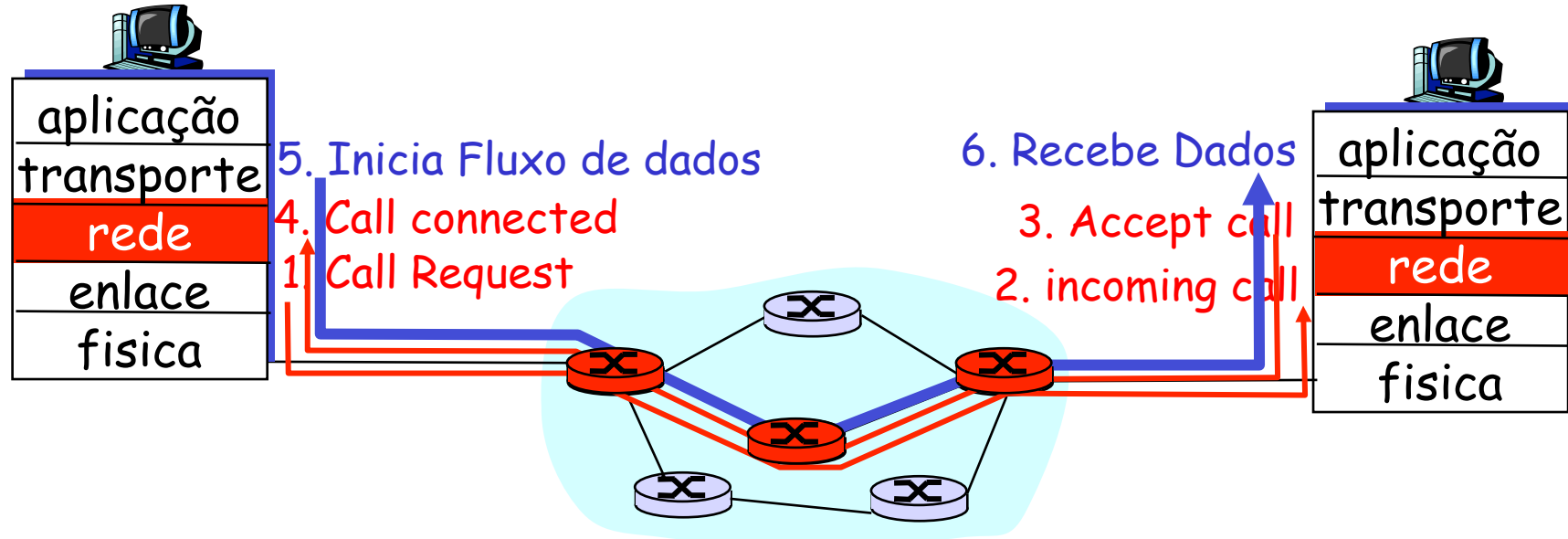
Circuitos Virtuais (VC)

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

- Orientado ao desempenho
 - 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

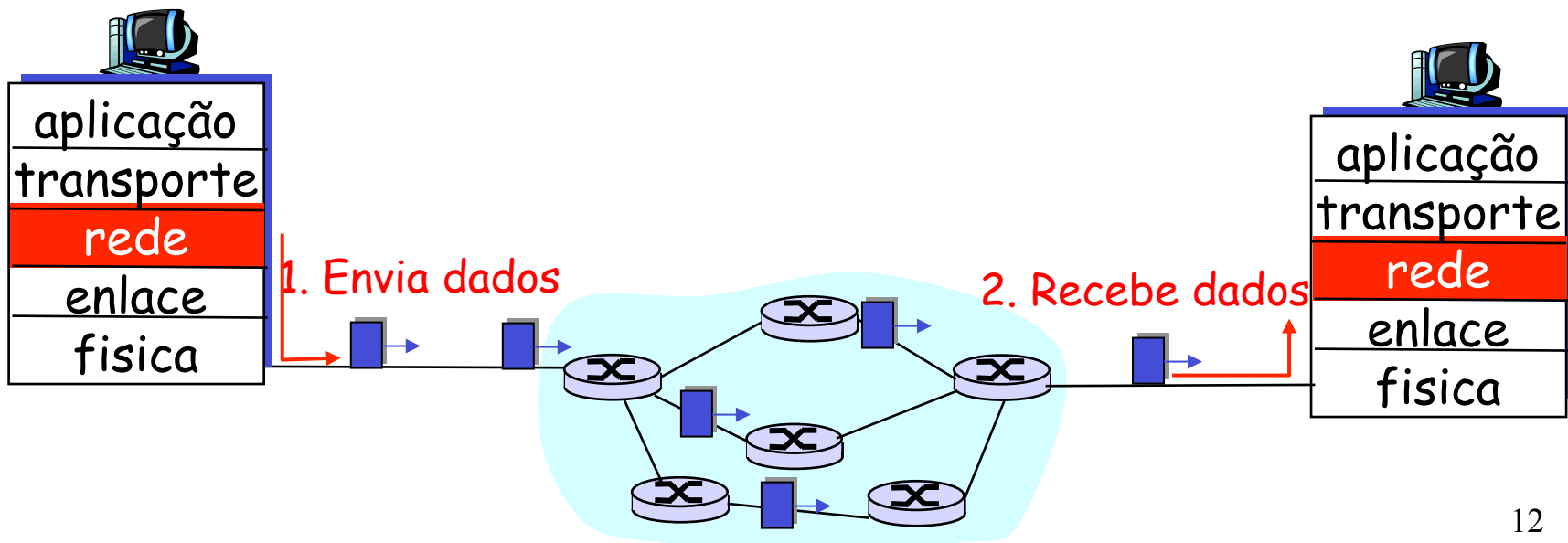
Circuitos Virtuais: Sinalização

- ❑ 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



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

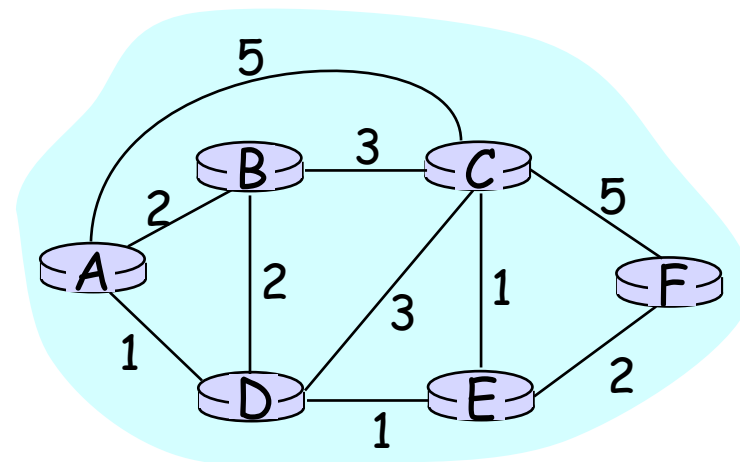
Roteamento

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 custo dos enlaces

- ❑ algoritmos "Link state"

Descentralizada:

- ❑ Roteadores só conhecem informações sobre seus vizinhos e os enlaces para eles
- ❑ Processo de computação iterativo, troca de informações com os vizinhos
- ❑ algoritmos "Distance vector"

Estático ou Dinâmico?

Estático:

- ❑ As rotas mudam lentamente ao longo do tempo

Dinâmico:

- ❑ As rotas mudam mais rapidamente
 - 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 Dijkstra'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 $D(v) = \text{infty}$

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

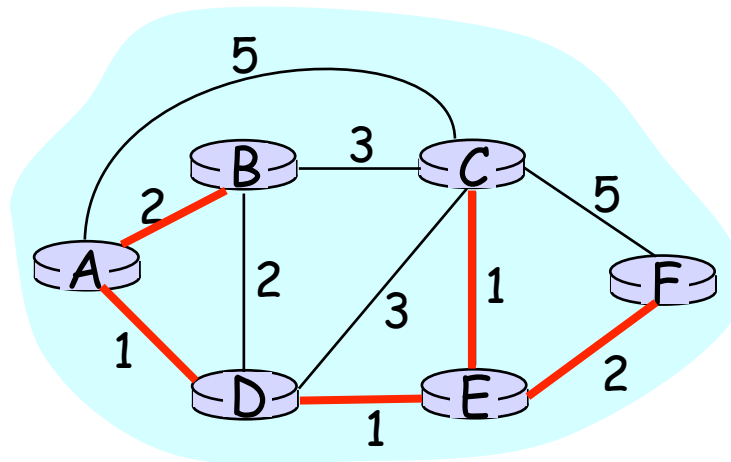
13 /* novo custo para v é ou o custo anterior para v ou o menor

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					



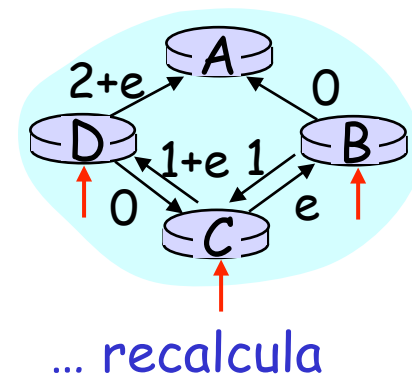
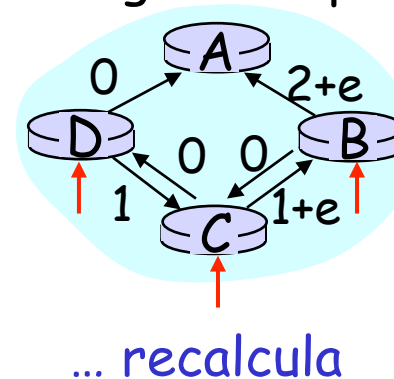
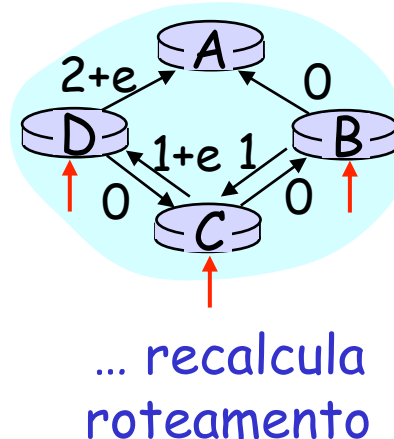
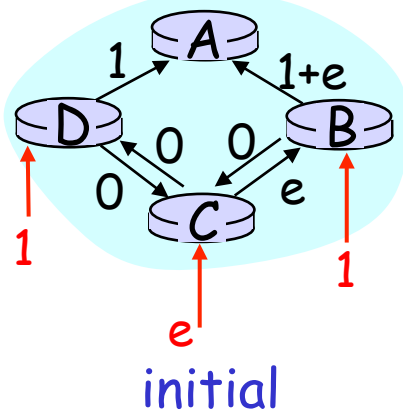
Discussão do Algoritmo de Dijkstra

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: $O(n \log n)$

Oscilações possíveis:

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



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:

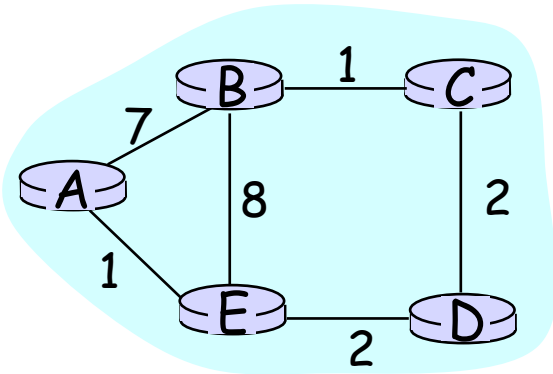
- ❑ Cada nó se comunica apenas com os seus vizinhos, diretamente conectados

Estrutura de Dados da Tabela de Distância

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

$$\begin{aligned} D^X(Y, Z) &= \text{distância de X to Y, via Z como prox. salto} \\ &= c(X, Z) + \min_w \{D^Z(Y, w)\} \end{aligned}$$

Exemplo de Tabela de Distância



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

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

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

		custo via nó vizinho		
$D^E()$		A	B	D
destino	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
	B	7	8	5
	C	6	9	4
	D	4	11	2

destino

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

destino

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:

espera por mudança no custo dos enlaces locais ou mensagem do vizinho

recalcula tabela de distância

se o caminho de menor custo para algum destino mudou, *notifica* vizinhos

Algoritmo Vetor-Distância:

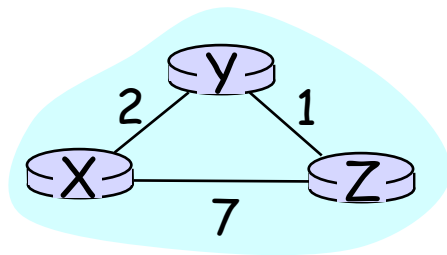
Para todos os nós, X :

- 1 Inicialização:
- 2 para todos os nós adjacentes v :
- 3 $D^X(*,v) = \text{infinito}$ /* 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:  $D^X(Y,V) = c(X,V) + \text{newval}$ 
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



		cost via X	
dest	D	X	
dest	Y	2	∞
	Z	∞	7

		cost via Y	
dest	D	Y	Z
dest	Y	2	8
	Z	3	7

		cost via Z	
dest	D	Y	Z
dest	Y		
	Z		

		cost via X	
dest	D	X	Z
dest	X	2	∞
	Z	∞	1

		cost via Y	
dest	D	X	Z
dest	X	2	8
	Z	9	1

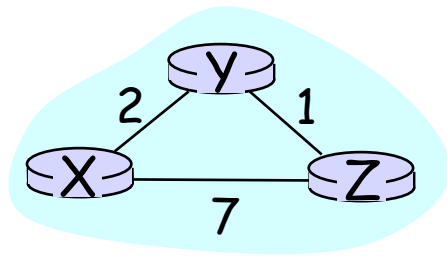
		cost via Z	
dest	D	X	Z
dest	X		
	Z		

		cost via X	
dest	D	X	Y
dest	X	7	∞
	Y	∞	1

		cost via Y	
dest	D	X	Y
dest	X	7	3
	Y	9	1

		cost via Z	
dest	D	X	Y
dest	X		
	Y		

Exemplo: algoritmo vetor-distância



		cost via	
		Y	Z
dest	X		
	D		
dest	Y	2	∞
	Z	∞	7

		cost via	
		X	Z
dest	Y		
	D		
dest	X	2	∞
	Z	∞	1

		cost via	
		X	Y
dest	Z		
	D		
dest	X	7	∞
	Y	∞	1

		cost via	
		Y	Z
dest	X		
	D		
dest	Y	2	8
	Z	3	7

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

$$= 7 + 1 = 8$$

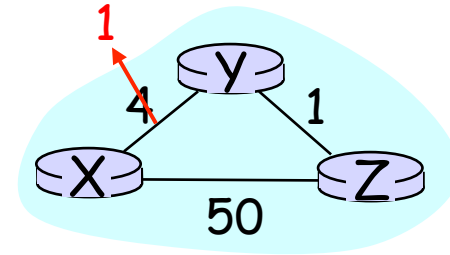
$$D^X(Z,Y) = c(X,Y) + \min_w \{D^Y(Z,w)\}$$

$$= 2 + 1 = 3$$

Vetor-Distância: Mudança no custo do enlace

Mudança no custo do enlace:

- ❑ não 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”

		via	
Y	D	X	Z
	x	4	6

	Y	
D	X	Z
X	1	6

Y	X	Z
X	1	6

	Y	X	Z
D			
X		1	3

Z	via	
D	X	Y
x	50	5

D ^Z	X	Y
x	50	5

D ^Z	X	Y
X	50	2

D ^Z	X	Y
x	50	2

algoritmo
termina

c(X,Y)
change

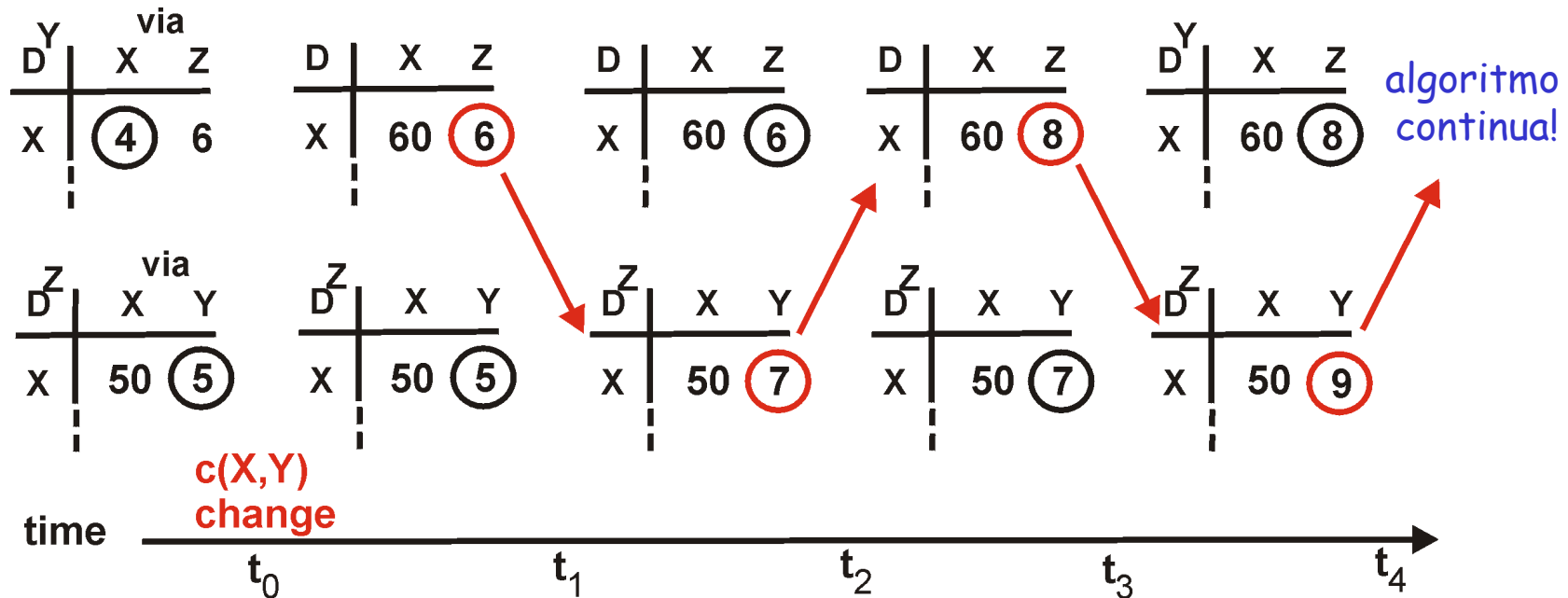
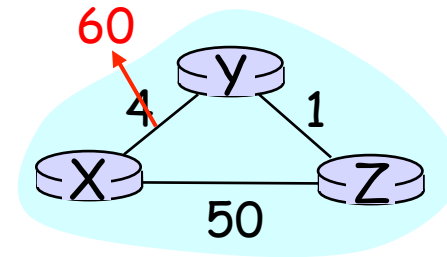
time

 t_0 t_1 t_2

Vetor Distância: Mudança no custo do enlace

Mudança no custo do enlace:

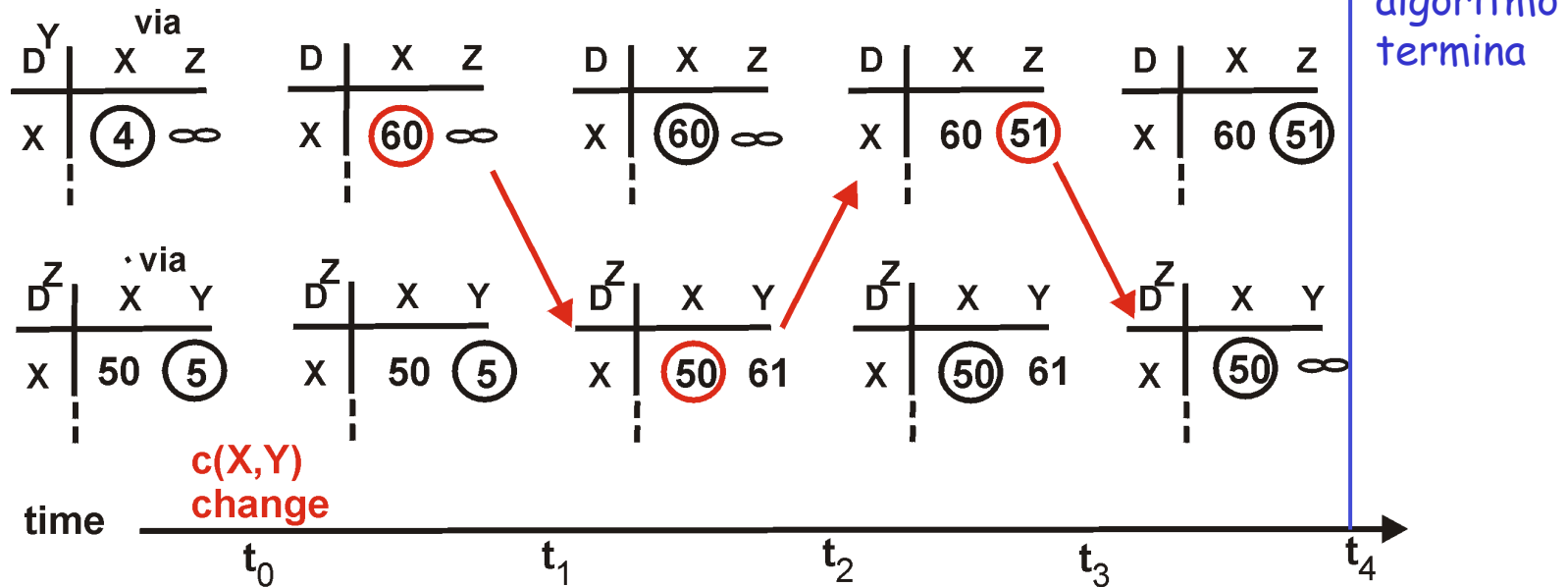
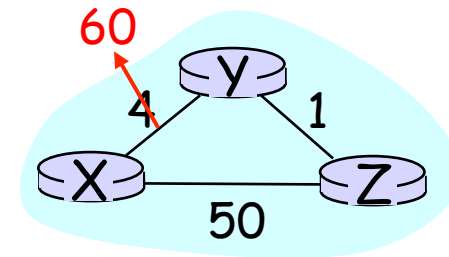
- ❑ más notícias viajam devagar - problema da contagem ao infinito



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

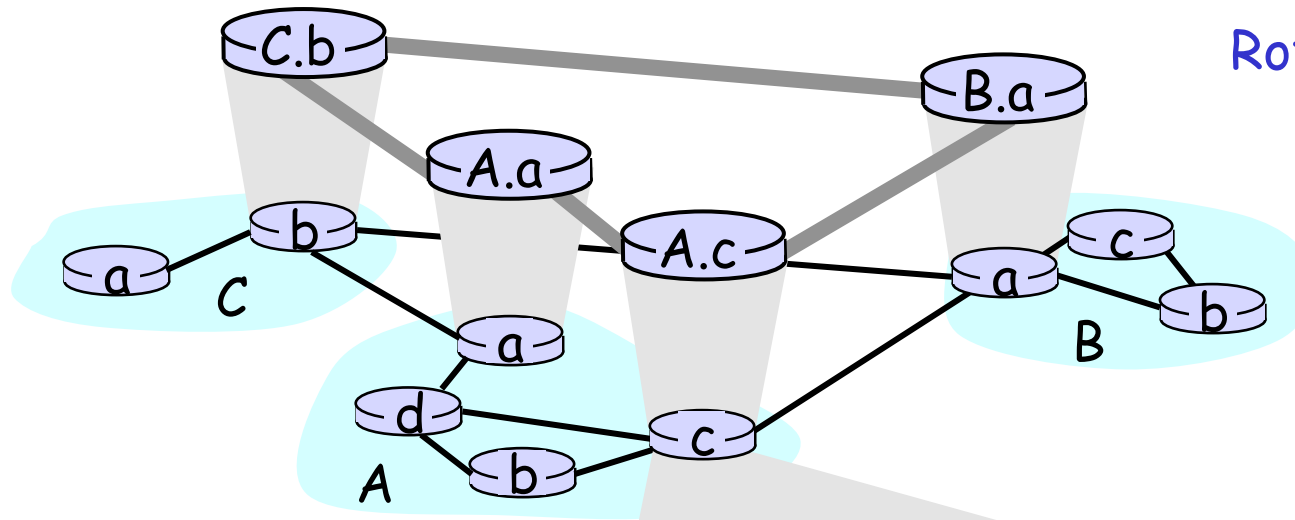
Roteamento Hierárquico

- ❑ 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 roteadores de borda

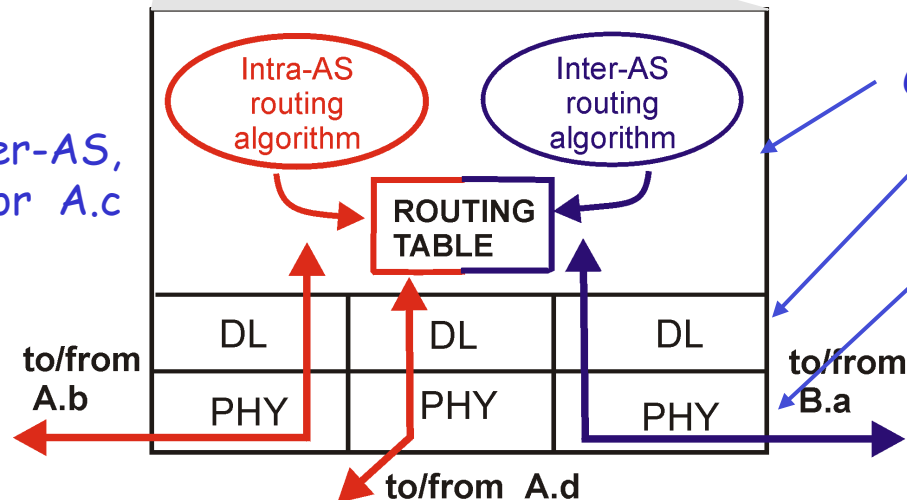
Roteamento Intra-as and Inter-as



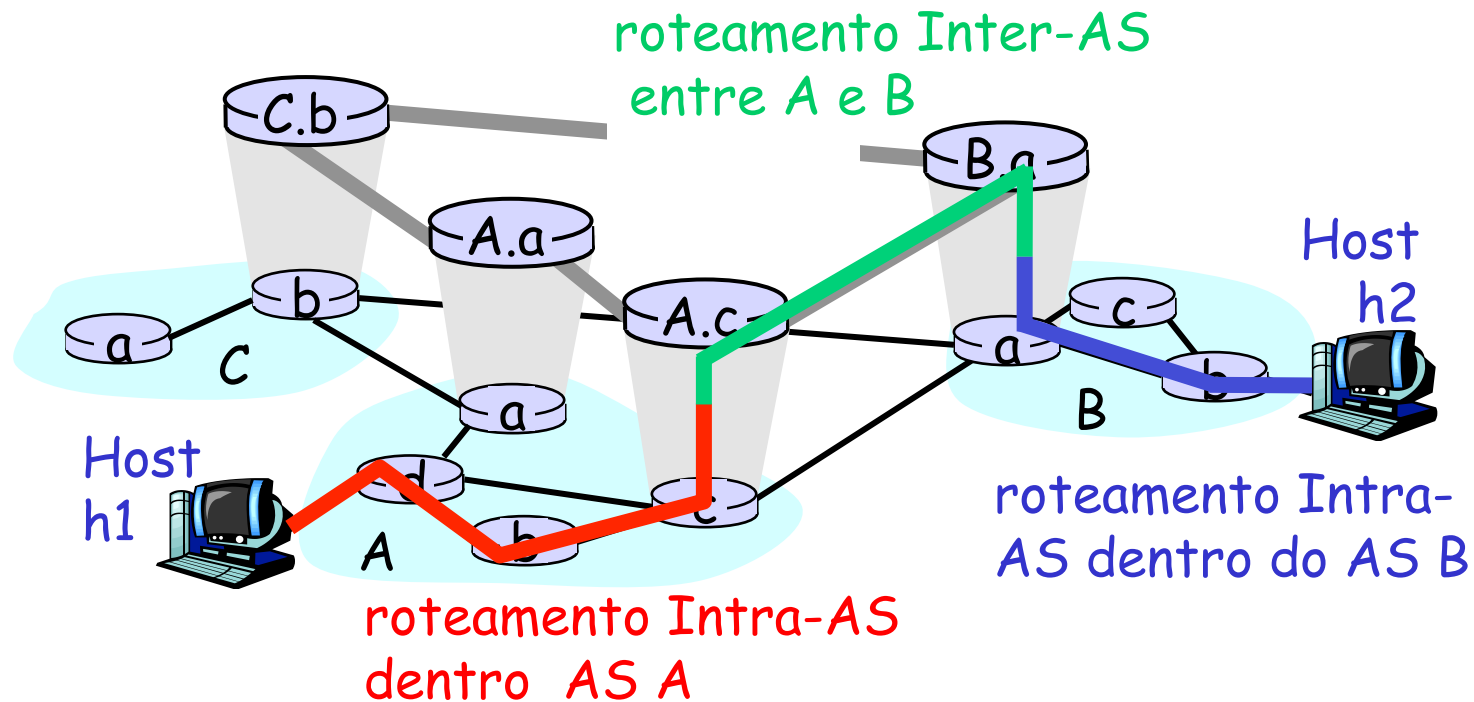
Roteadores de Borda

- 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

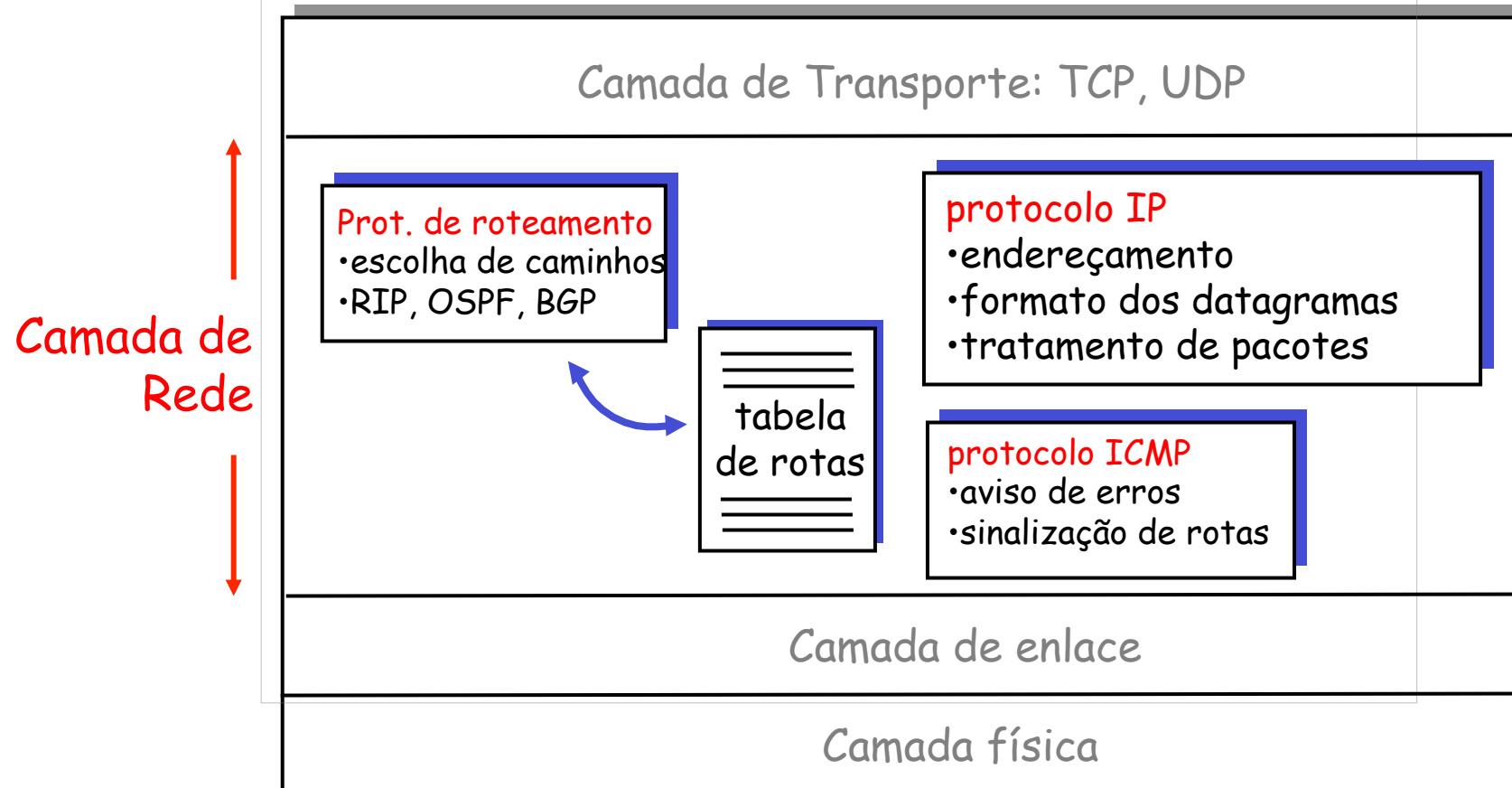


Roteamento Intra-AS e Inter-AS



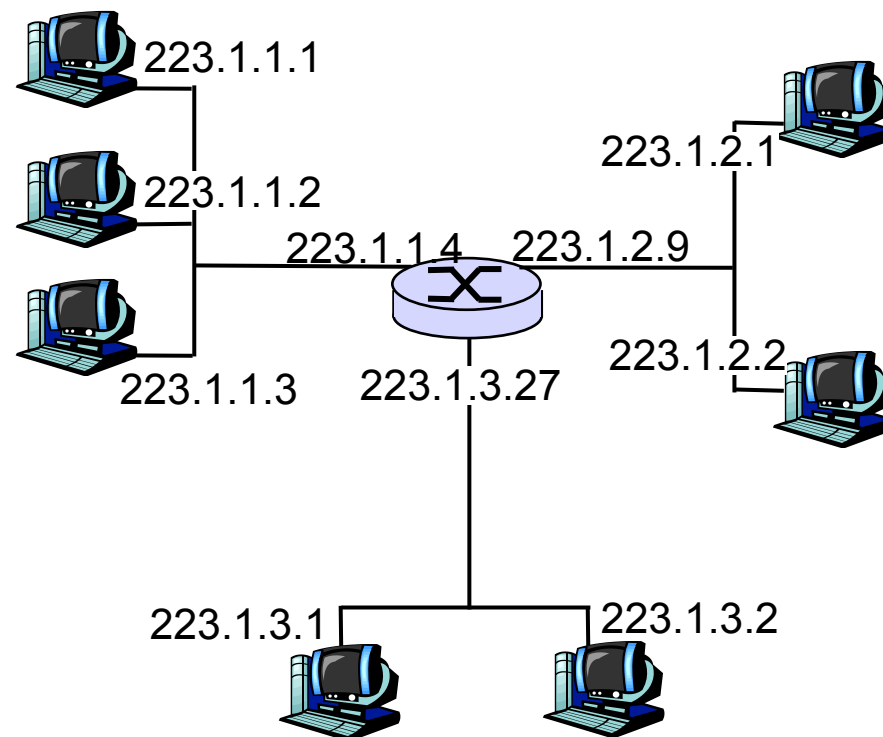
A camada de rede da Internet

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



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

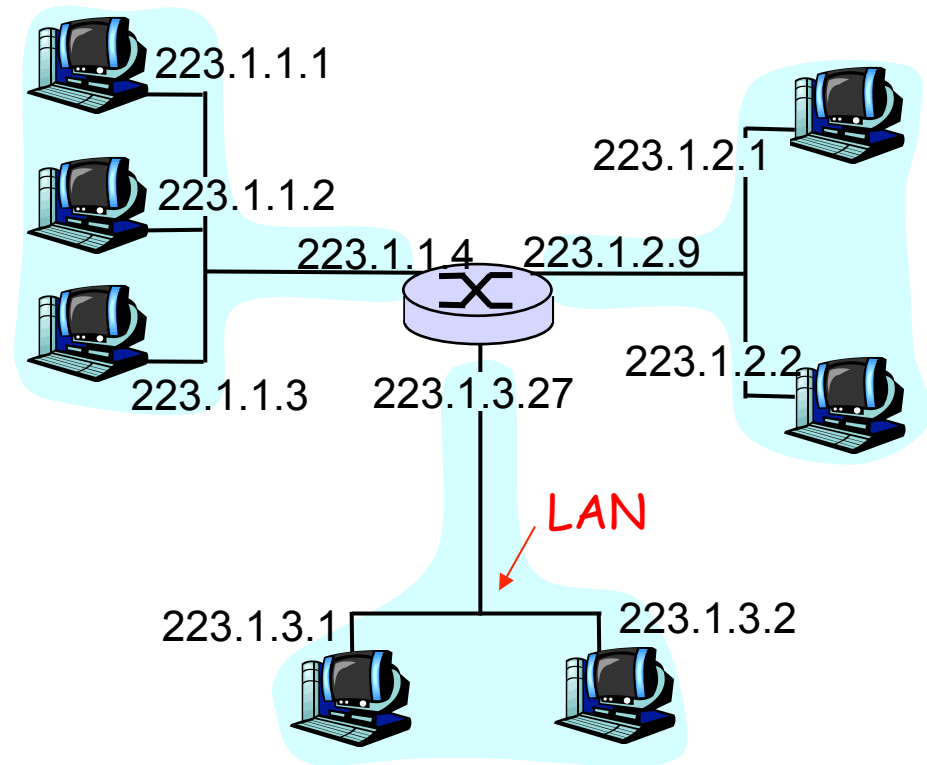
Endereçamento IP

❑ Endereço IP:

- parte de rede (bits mais significativos)
- parte de Host part (bits menos significativos)

❑ *O que é uma rede?* (na perspectiva do endereço)

- Interfaces de dispositivos com a mesma parte de rede no endereço IP
- Podem fisicamente se comunicar sem o auxílio de um roteador



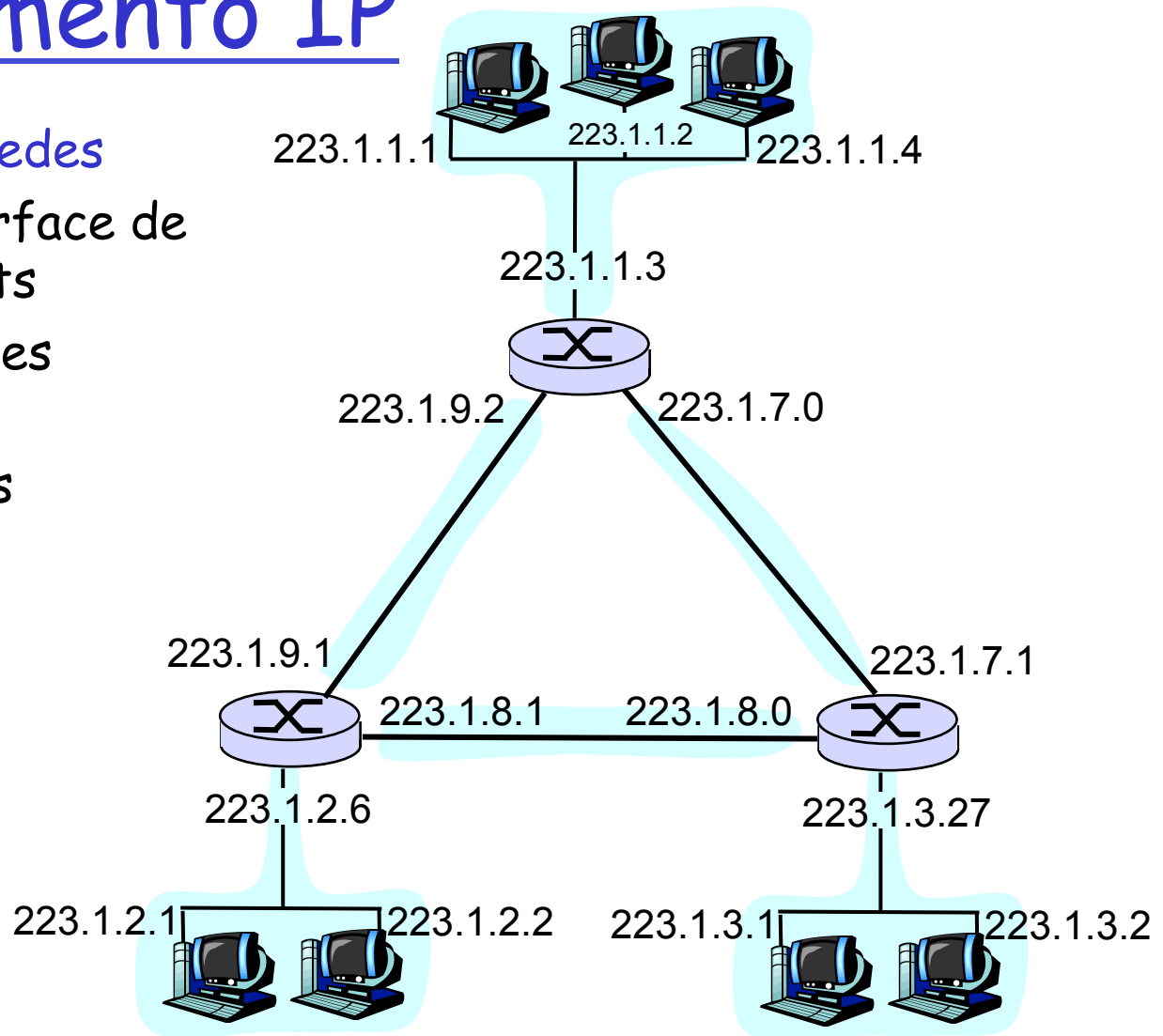
rede consistindo de de 3 redes IP
(para endereços IP começando com 223,
os primeiros 24 bits são o endereço de
rede)

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

A	0	rede		host		1.0.0.0 to 127.255.255.255	
B	10		rede		host	128.0.0.0 to 191.255.255.255	
C	110		rede			host	192.0.0.0 to 223.255.255.255
D	1110				multicast address		224.0.0.0 to 239.255.255.255

← 32 bits →

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

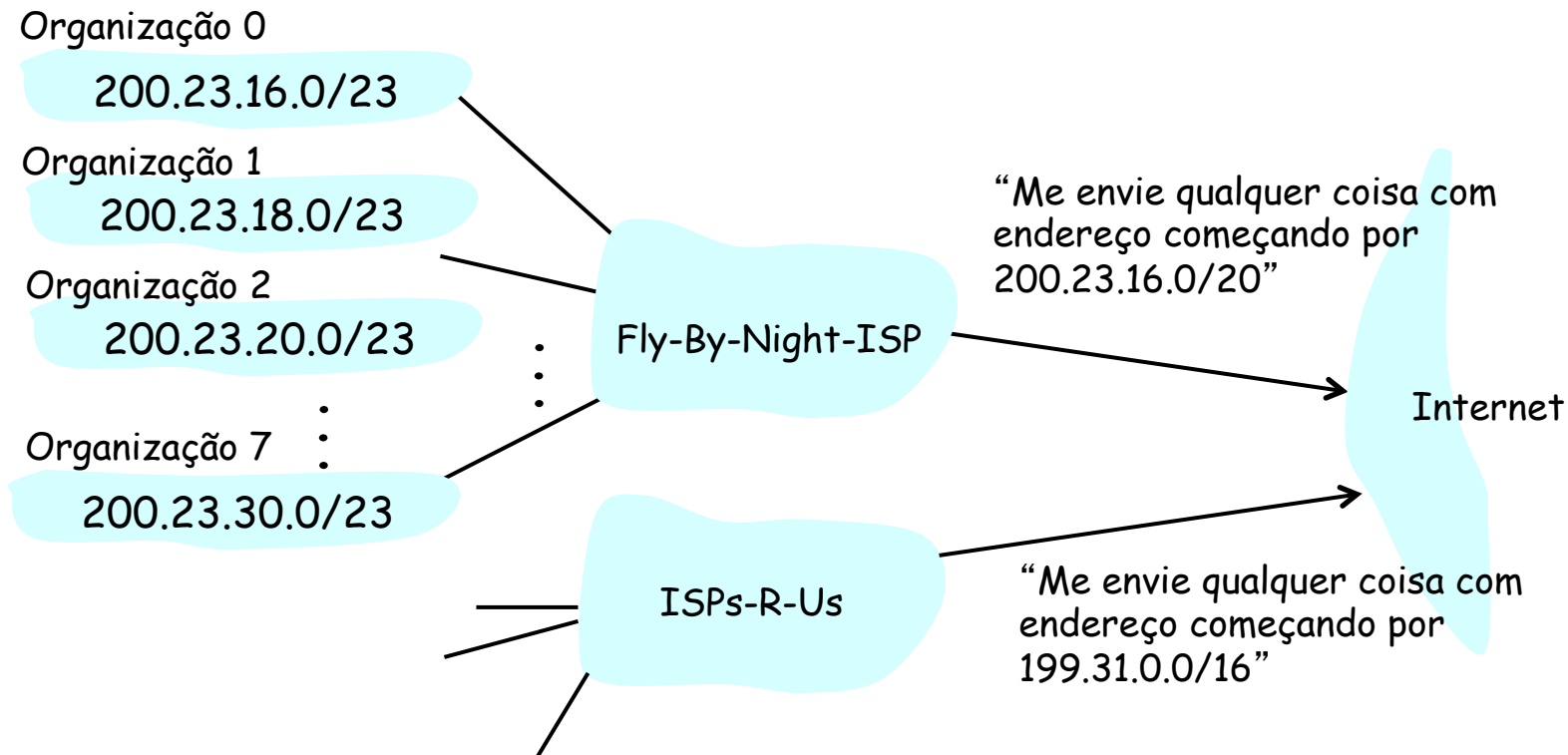
Rede (porção de rede)

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

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

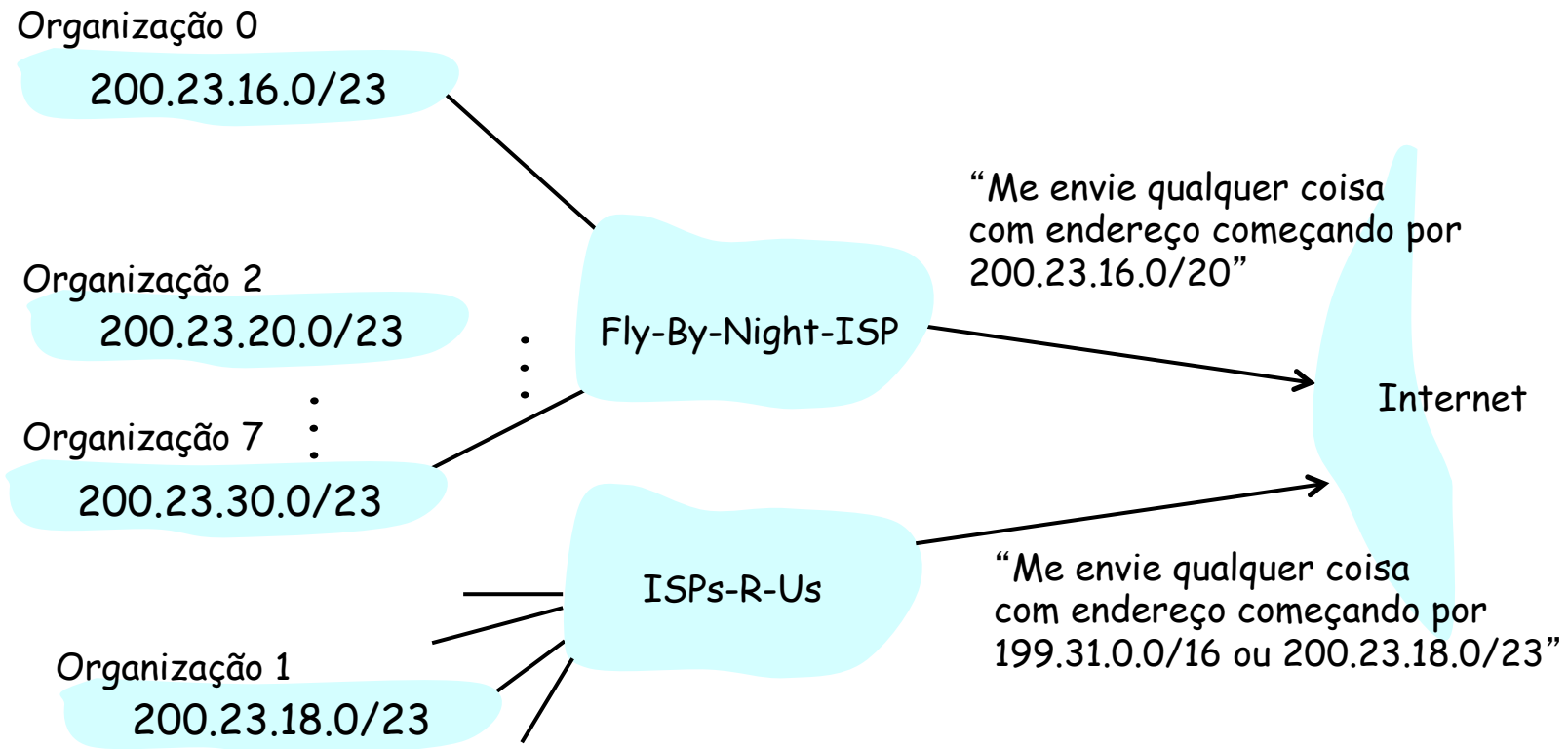
Endereçamento Hierárquico: agregação de rotas

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
- Atribui nomes de domínios e resolve disputas

Levando um Datagrama da Fonte ao Destino

Destino

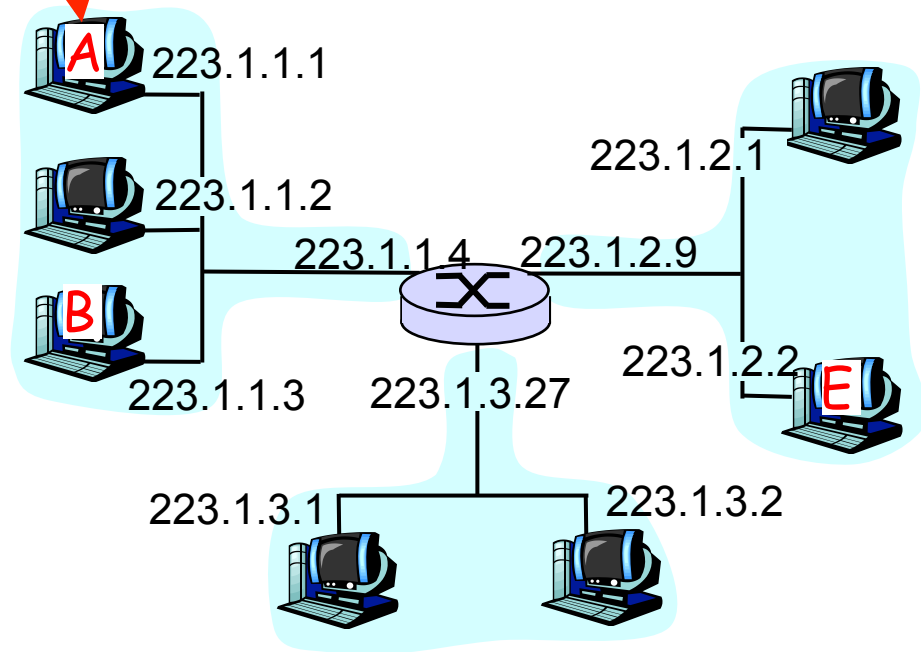
tabela de roteamento
em A

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

datagrama IP:

outros campos	endereço IP origem	endereço IP destino	dados
---------------	--------------------	---------------------	-------

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

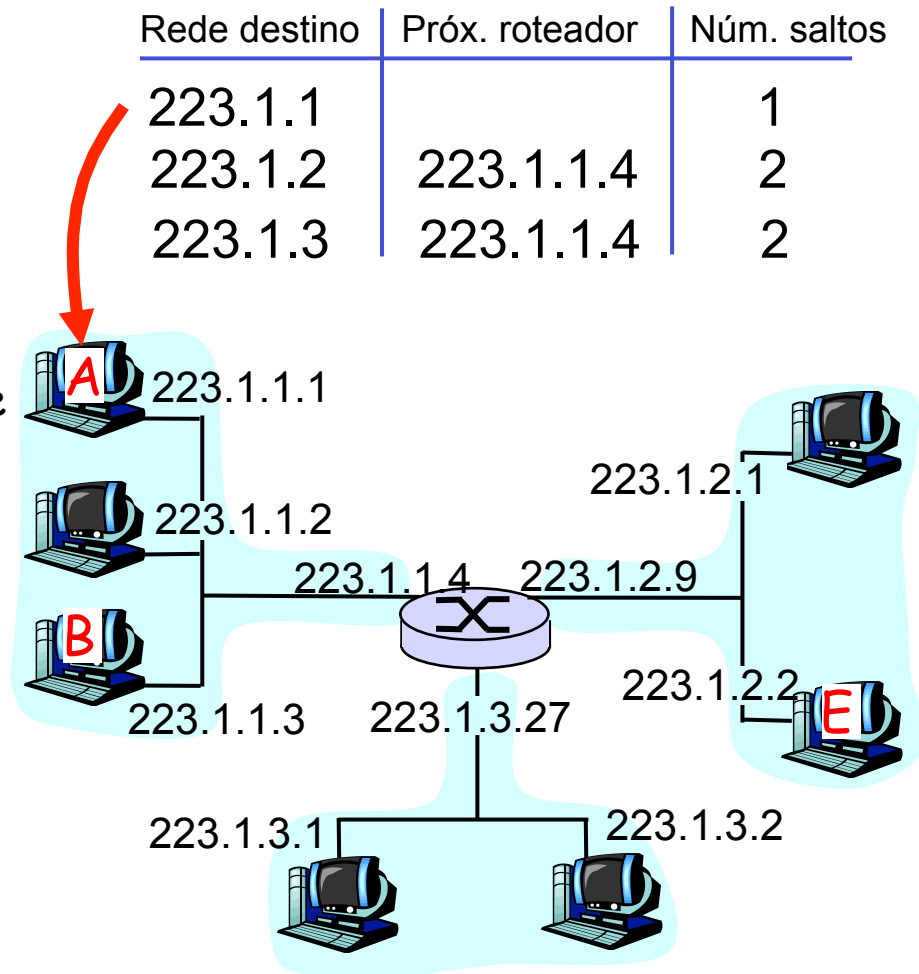


Levando um Datagrama da Fonte ao Destino

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
- descubra 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



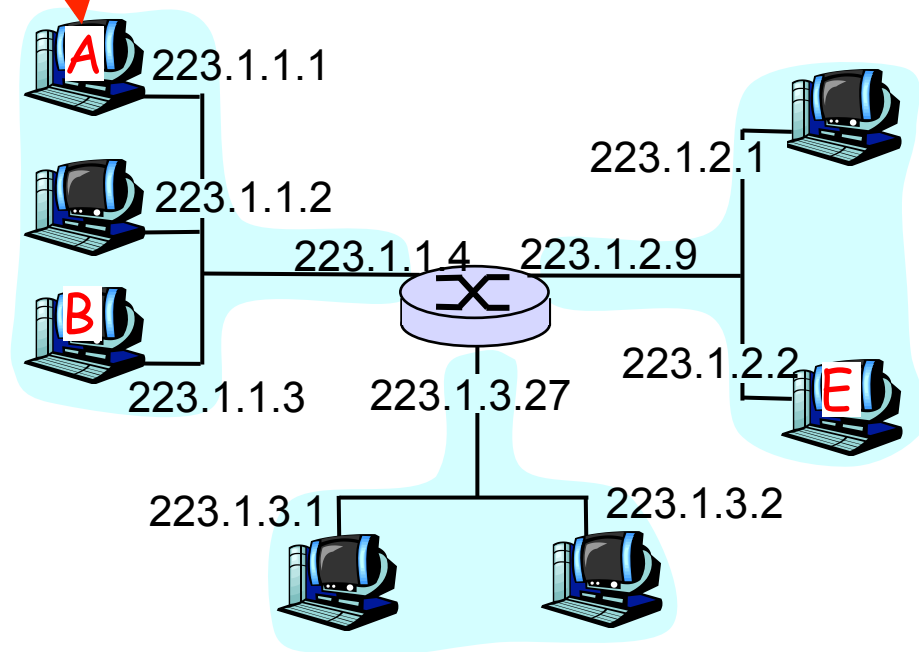
Levando um Datagrama da Fonte ao Destino

outros campos	223.1.1.1	223.1.2.3	dados
---------------	-----------	-----------	-------

Começando em A, dest. E:

- ❑ examina endereço de rede de E
- ❑ E está numa 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



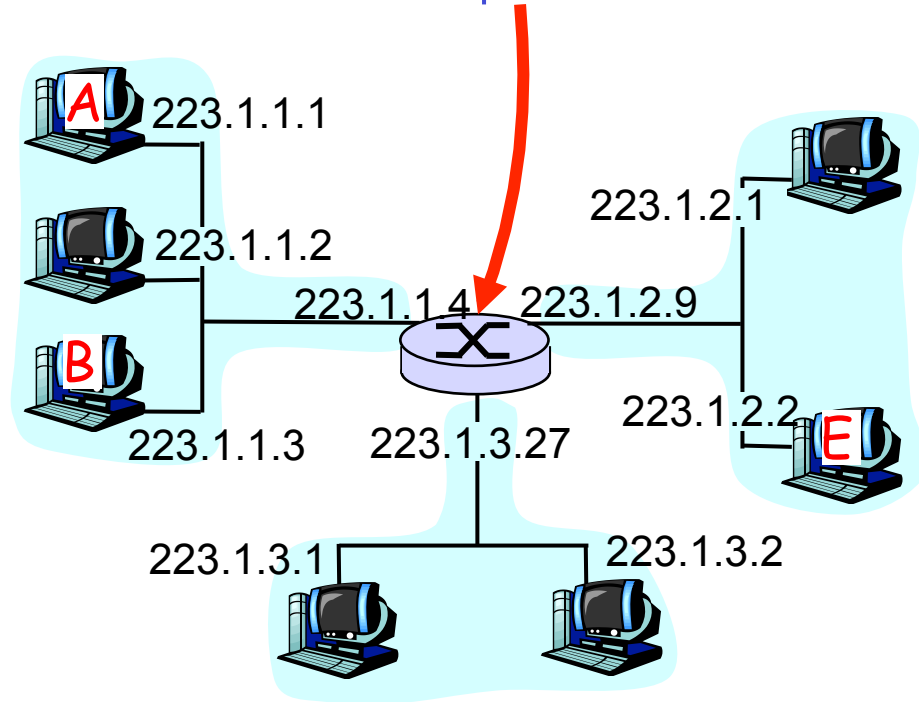
Levando um Datagrama da Fonte ao Destino

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



Endereçamento IP

- ❑ É um esquema de endereçamento lógico onde cada dispositivo conectado em rede necessita usar pelo menos um endereço IP.
- ❑ Esse endereço permite identificar o dispositivo e a rede na qual ele pertence.

Endereçamento IP

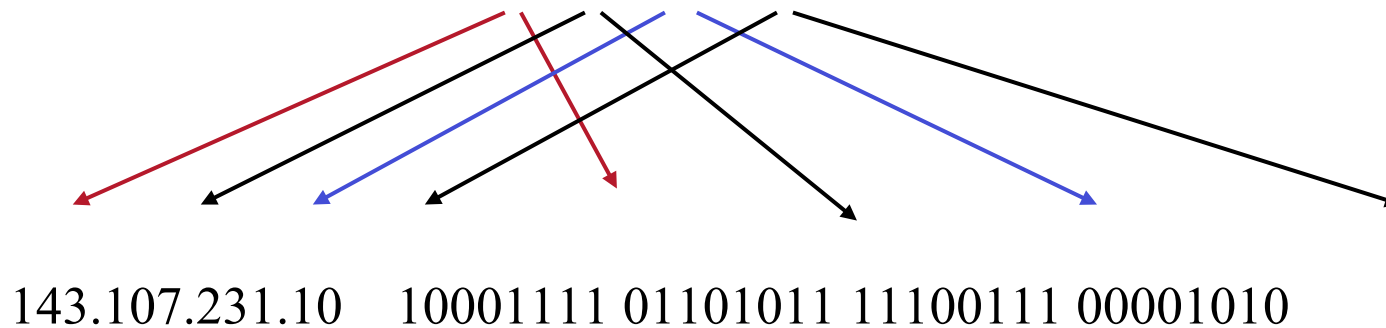
- ❑ Um endereço IP possui basicamente duas partes:
 - Indicação da rede e;
 - Indicação do dispositivo

Identificação da Rede	Identificação do Dispositivo
-----------------------	------------------------------

Composição do Endereço

- ❑ O endereço IP é um número de 32 bits, representado em decimal em forma de oito bits separados, no formato:

a.b.c.d



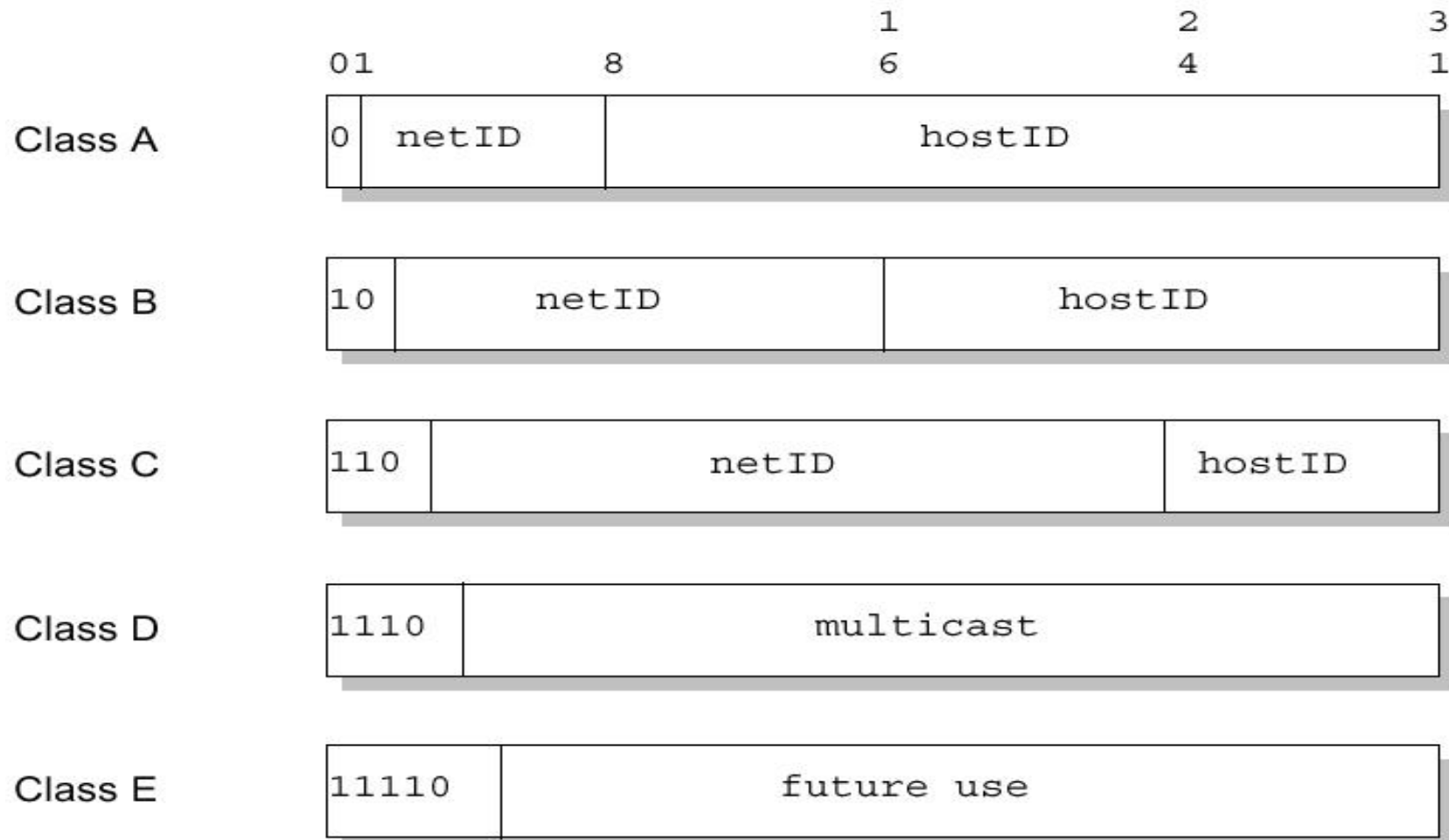
Composição do Endereço IP

- ❑ Todos os computadores dentro da mesma rede devem possuir o mesmo número de rede e o número do próprio computador, diferenciado.
- ❑ O mapeamento entre o formato numérico e um nome simbólico, como icmc.usp.br, é feito pelo servidor de nomes (*Domain Name Server - DNS*).

Endereçamento IP

- ❑ Cada dispositivo em uma rede TCP/IP precisa ter um endereço IP único.
- ❑ Para facilitar a distribuição de endereços IP, foram especificadas 5 classes:
 - Classe A
 - Classe B
 - Classe C
 - Classe D
 - Classe E

Classes de Endereços IP



Classe A

- ❑ Nos endereços de Classe A, o primeiro número identifica a rede e os outros três números identificam o próprio computador (nó).
- ❑ O bit de maior grau em uma classe A é sempre zero.
- ❑ Os próximos 7 bits (*preenchendo o primeiro octeto*) completam a identificação de rede.
- ❑ Os 24 bits restantes (*os últimos 3 octetos*) representam a identificação do host.

Classe A

- ❑ Permite 2^7-2 redes e $2^{24}-2$ hosts em cada
- ❑ Um endereço é classificado como Classe A, quando o primeiro número (byte) do endereço estiver entre 1 e 127.
- ❑ Exemplo:
 - 13.0.0.1
 - 80.10.69.12
 - 37.25.10.99

13.0.0.1



13.0.0.2

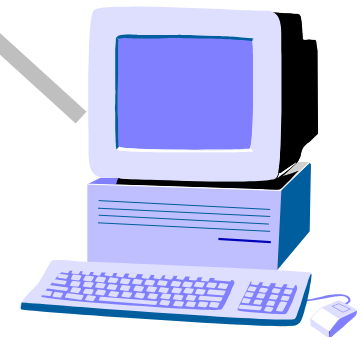


TCP/IP

13.0.0.3



13.0.0.4



Classe B

- ❑ Nos endereços de Classe B, os dois primeiros números identificam a rede e os outros dois números identificam o próprio computador (nó).
- ❑ Os 2 bits de maior grau em uma classe B são sempre os valores binários 10.
- ❑ Os próximos 14 bits (*preenchendo primeiro e o segundo octeto*) completam a identificação de rede.
- ❑ Os 16 bits restantes (os últimos 2 octetos) representam a identificação do host.

Classe B

- ❑ Permite $2^{14}-2$ redes e $2^{16}-2$ hosts em cada.
- ❑ Um endereço é classificado como Classe B, quando o primeiro número (byte) do endereço estiver entre 128 e 191.
- ❑ Exemplo:
 - 133.0.0.1
 - 140.10.69.12
 - 190.25.10.99

140.249.238.2

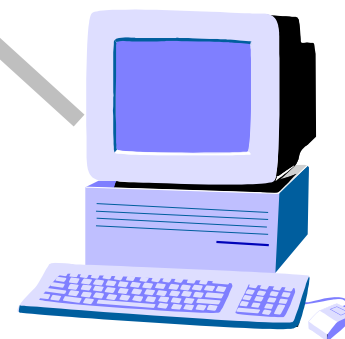
140.249.238.3



TCP/IP

140.249.238.5

140.249.238.7



Classe C

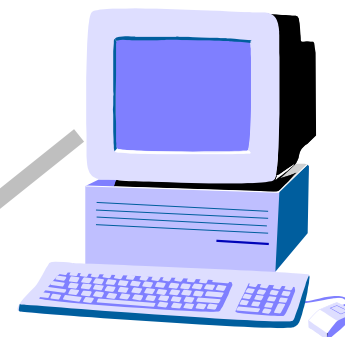
- ❑ Nos endereços de Classe C, os três primeiros números identificam a rede e o último número identifica o próprio computador (nó).
- ❑ Os 3 bits de maior grau em uma classe C são sempre os valores binários 110.
- ❑ Os próximos 21 bits (*preenchendo os 3 primeiros octetos*) completam a identificação de rede.
- ❑ Os oito bits restantes (*o último octeto*) representam a identificação do host.

Classe C

- ❑ Permite $2^{21}-2$ redes e $2^8 - 2$ hosts em cada.
- ❑ Um endereço é classificado com o Classe C, quando o primeiro número (byte) do endereço estiver entre 192 e 223.
- ❑ Exemplo:
 - 200.0.0.1
 - 220.10.69.12
 - 195.25.10.99

220.255.11.1

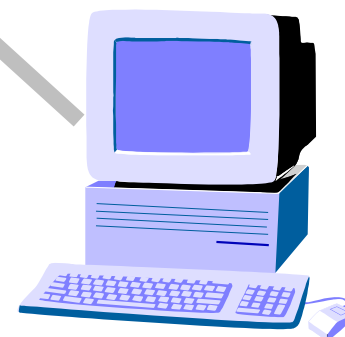
220.255.11.2



TCP/IP

220.255.11.3

220.255.11.4



Classes D

- ❑ Um endereço é classificado como Classe D, quando o primeiro número (byte) do endereço for superior a 224.
- ❑ Exemplo:
 - 225.0.0.1
 - 239.10.69.12
 - 226.25.10.99

Classes D

- ❑ Esta classe está reservado para criar agrupamentos de computadores para o uso de Multicast.
- ❑ Não podemos utilizar esta faixa de endereços para endereçar os computadores de usuários na rede TCP/IP.

Classes E

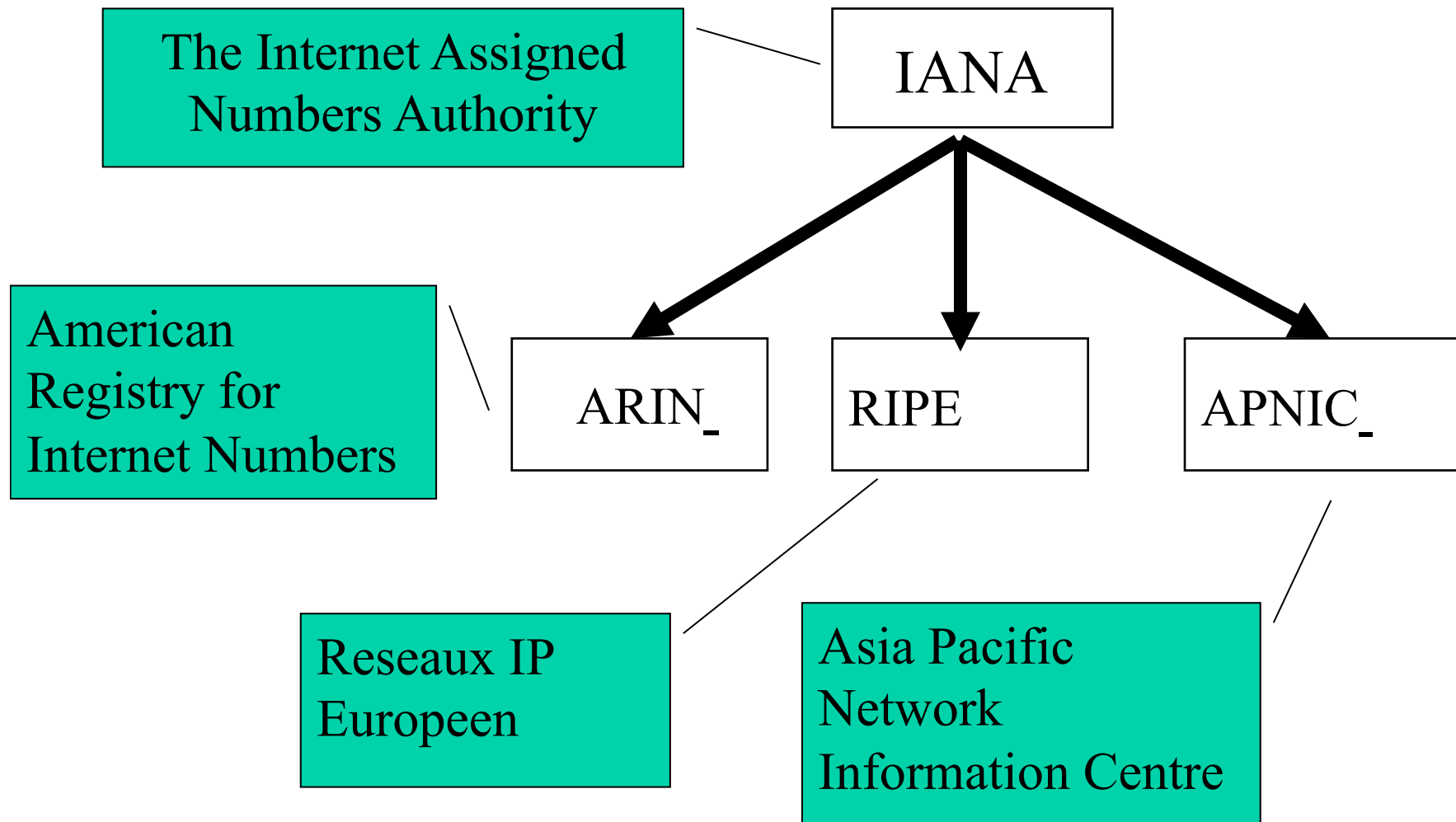
- ❑ A Classe E é um endereço reservado e utilizado para testes e novas implementações e controles do TCP/IP.
- ❑ São endereços IP com valores iniciais acima de 240.0.0.0
- ❑ Não podemos utilizar esta faixa de endereços para endereçar os computadores na rede TCP/IP.

Classes de Endereços IP - Resumo

Classe	Menor Endereço	Maior Endereço
A	1 . 0 . 0 . 0	126 . 255 . 255 . 255
B	128 . 0 . 0 . 0	191 . 255 . 255 . 255
C	192 . 0 . 0 . 0	223 . 255 . 255 . 255
D	224 . 0 . 0 . 0	239 . 255 . 255 . 255
E	240 . 0 . 0 . 0	247 . 255 . 255 . 255

	Número de redes	Hosts por rede	1º Octeto
Classe A	126	16.777.214	1 – 126
Classe B	16.384	65.534	128 – 191
Classe C	2.097.152	254	192 - 223

Alocação de números da Internet



Máscara de Rede

- ❑ É formada por 32 bits no mesmo formato que o endereçamento IP.
- ❑ Utilizado para definir a rede a qual pertence o computador.
- ❑ A rede do computador é obtida a partir de um AND entre o endereço do computador e a máscara.

Máscara de Rede

- ❑ Se a rede do computador destino for a mesma do computador origem o dado é enviado diretamente para o computador destino através da sub-rede
- ❑ Se a rede for diferente os pacotes são enviados para o roteador.

Máscaras comuns

- ❑ Cada bit 1 - parte do endereço usado para a rede.
- ❑ Cada bit 0 - parte do endereço usado para as máquinas.
- ❑ Classe A: 255.0.0.0
- ❑ Classe B: 255.255.0.0
- ❑ Classe C: 255.255.255.0

Encontrando a rede...

Endereço IP:	200	.	237	.	190	.	21	
								AND
Máscara de rede:	255	.	255	.	255	.	0	
	=====							
	200	.	237	.	190	.	0	Endereço de rede

Em binário...

(200.237.190.21)

11001000.11101101.10111110.10101

(255.255.255.0)

11111111.11111111.11111111.0

11001000.11101101.10111110.0

(200.237.190.0)

Encontrando a rede...

200.145.31.34	→	200.145.31.3	
<u>255.255.255.0</u>		<u>255.255.255.0</u>	
200.145.31.0		200.145.31.0	Mesma Rede!!

Encontrando a máquina...

Endereço IP:	200 . 237 . 190 . 21	
		AND
NOT Máscara de rede:	0 . 0 . 0 . 255	
	=====	
	0 . 0 . 0 . 21	Endereço de máquina

Sub-redes (subnet)

- ❑ A estrutura de endereçamento IP pode ser mudada localmente (a critério do administrador de rede), usando-se bits de endereçamento de máquina como um adicional para endereçamento de rede;

Sub-redes

- ❑ O número do host é dividido em número da sub-rede e número do host. O número IP é agora interpretado como:

<network number><subnet number><host number>

- ❑ A divisão é feita usando uma máscara de rede "não padrão" que permita extrair os endereços de rede e de máquina corretamente.

Sub-redes - Exemplo

- Por exemplo: uma rede classe B

16 bits 16 bits
<network number><host number>

- Os 16 bits do número do host podem ser usados da seguinte forma:
 - o primeiro byte é o número da subnet e o segundo byte é o número do host.
 - 2^8-2 (254, valores 0 e 255 são reservados) subnets possíveis
 - 2^8-2 hosts em cada subnet
 - máscara = 255.255.255.0

Sub-redes - Exemplo

- ❑ Rede classe C 165.214.32.0 precisa ser dividida em:
 - 3 subnets com 50 hosts e;
 - 2 subnets com 30 hosts.
- ❑ Pode-se usar a máscara 255.255.255.192 para dividir a rede em 4 sub-redes de 64 máquinas.

Sub-redes - Exemplo

❑ Rede 1:

Sub-rede A - 165.214.32.0 até 165.214.32.63

Sub-rede B - 165.214.32.64 até 165.214.32.127

Sub-rede C - 165.214.32.128 até 165.214.32.191

❑ Rede 2:

Sub-rede D - 165.214.32.192 até 165.214.32.255

Sub-redes - Exemplo

- ❑ A última sub-rede (Sub-rede D) pode ser segmentada com a máscara 255.255.255.224, resultando em duas sub-redes de 32 hosts

- ❑ Rede 2:

Sub-rede D - 165.214.32.192 até 165.214.32.223

- ❑ Rede 3:

Sub-rede E - 165.214.32.224 até 165.214.32.255

Resolvendo do endereço

- Por exemplo, o IP 165.214.32.68 pertence à rede 1:

10100101 11010110 10000000 1000100 (IP)

11111111 11111111 11111111 11000000 (máscara)

----- AND Lógico

10100101 11010110 10000000 1000000 (165.214.32.64 = Rede 1)

Resolvendo do endereço

- O IP 165.214.32.201 pertence à rede 2:

10100101 11010110 10000000 11001001 (IP)

11111111 11111111 11111111 11000000 (máscara)

----- AND Lógico

10100101 11010110 10000000 11000000 (165.214.32.192 = Rede 2)

Resolvendo do endereço

- O IP 165.214.32.251 pertence à rede 3:

10100101 11010110 10000000 11111011 (IP)

11111111 11111111 11111111 11100000 (máscara)

----- AND Lógico

10100101 11010110 10000000 11100000 (165.214.32.224 = Rede 3)