

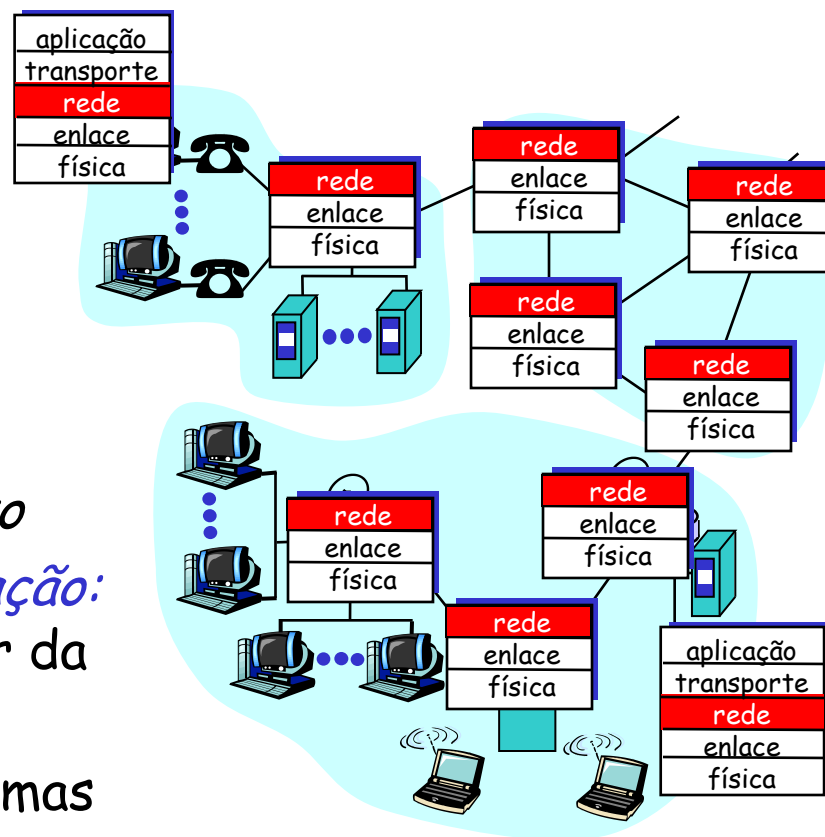
Camada de Rede/Endereçamento IP e DNS

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*
- *Encaminhamento/Repasse/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



Circuitos virtuais

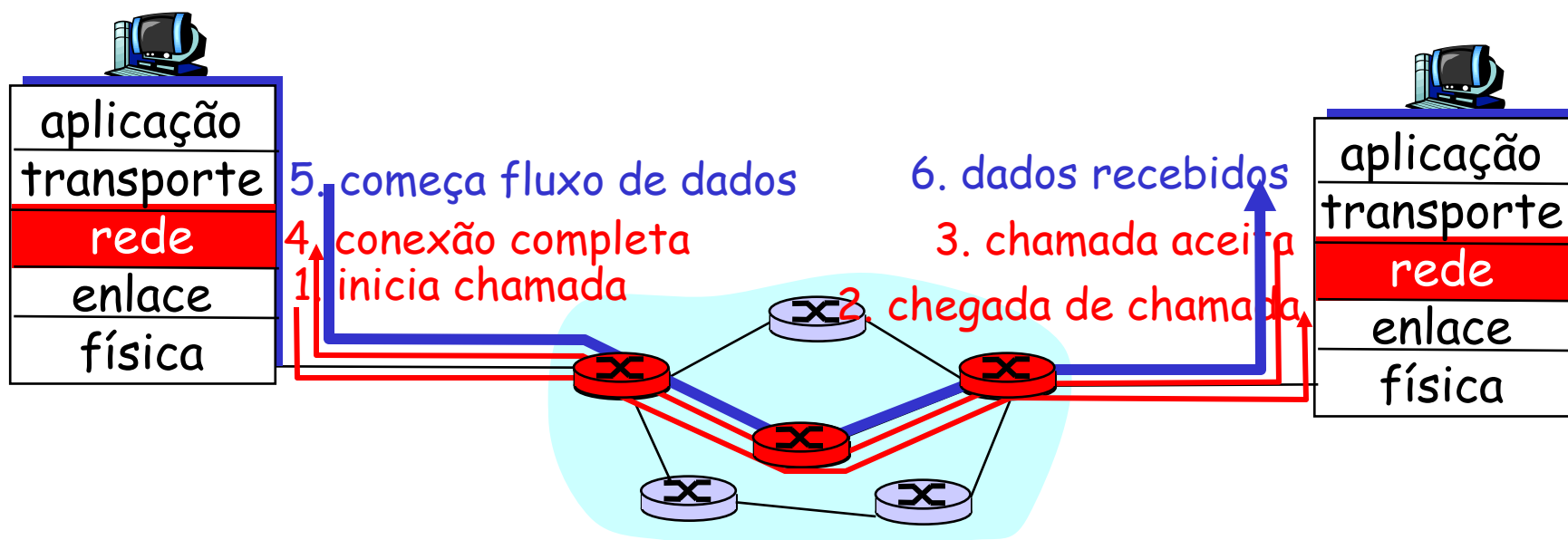
“caminho da-origem-ao-destino se comporta como um circuito telefônico”

- 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

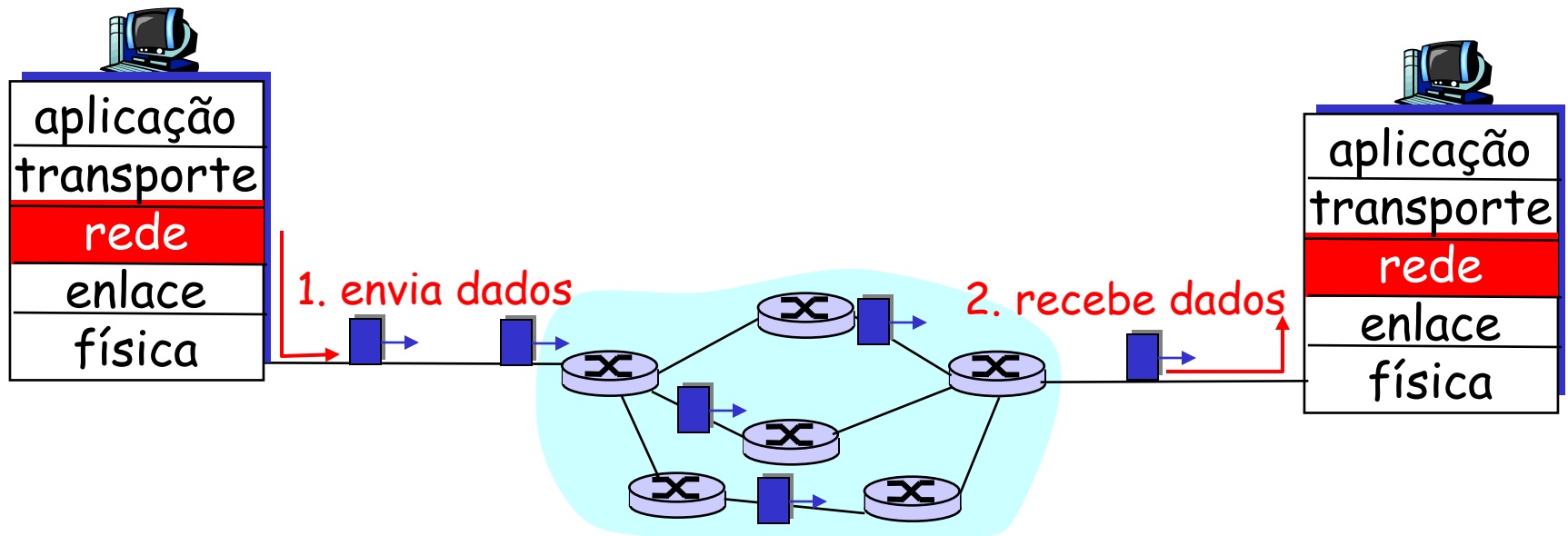
Circuitos virtuais: protocolos de sinalização

- usados para estabelecer, manter, destruir CV
- usados em ATM, frame-relay, X.25



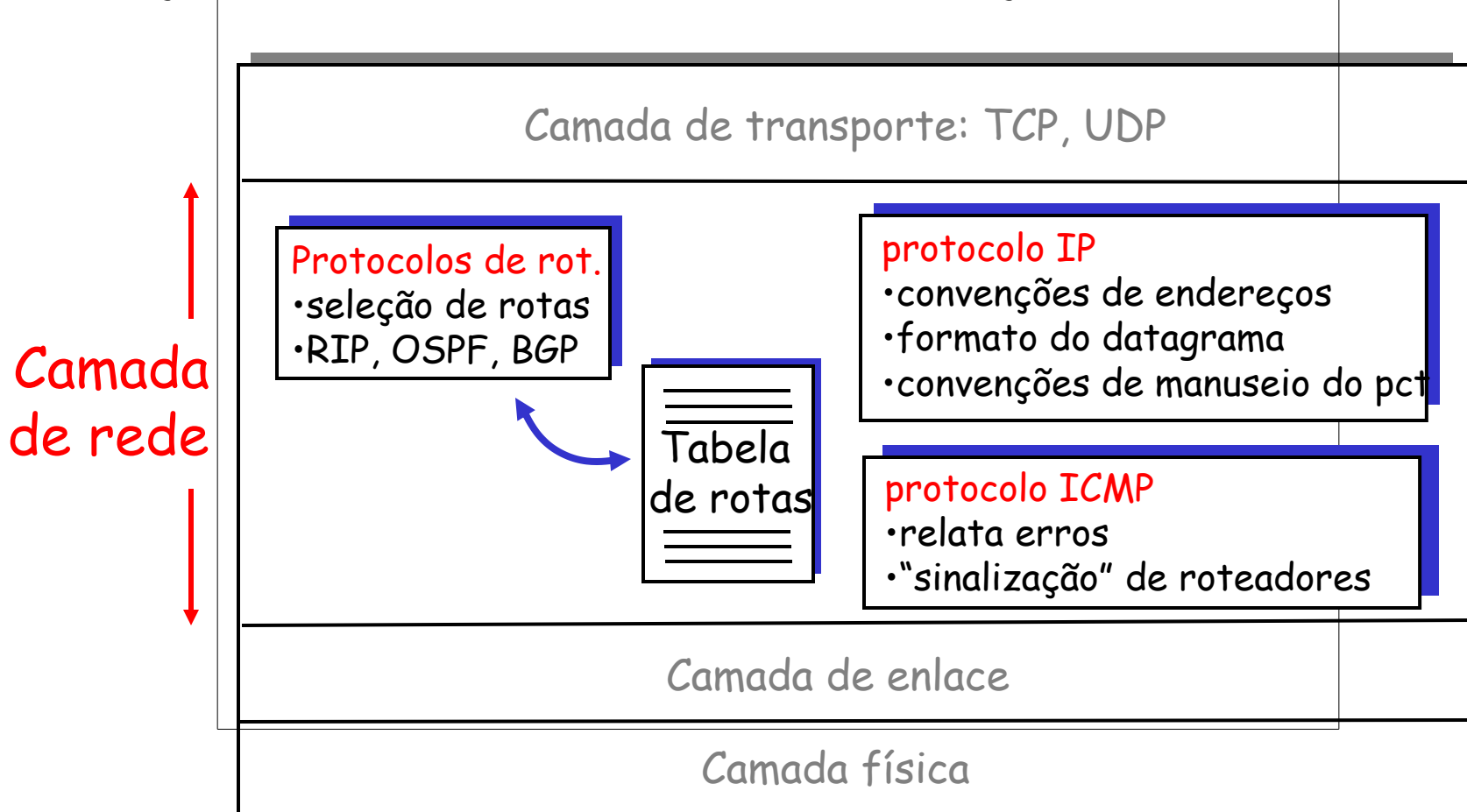
Rede de datagramas: o modelo da Internet

- ❑ 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
 - 2 pacotes entre o mesmo par origem-destino podem seguir caminhos diferentes



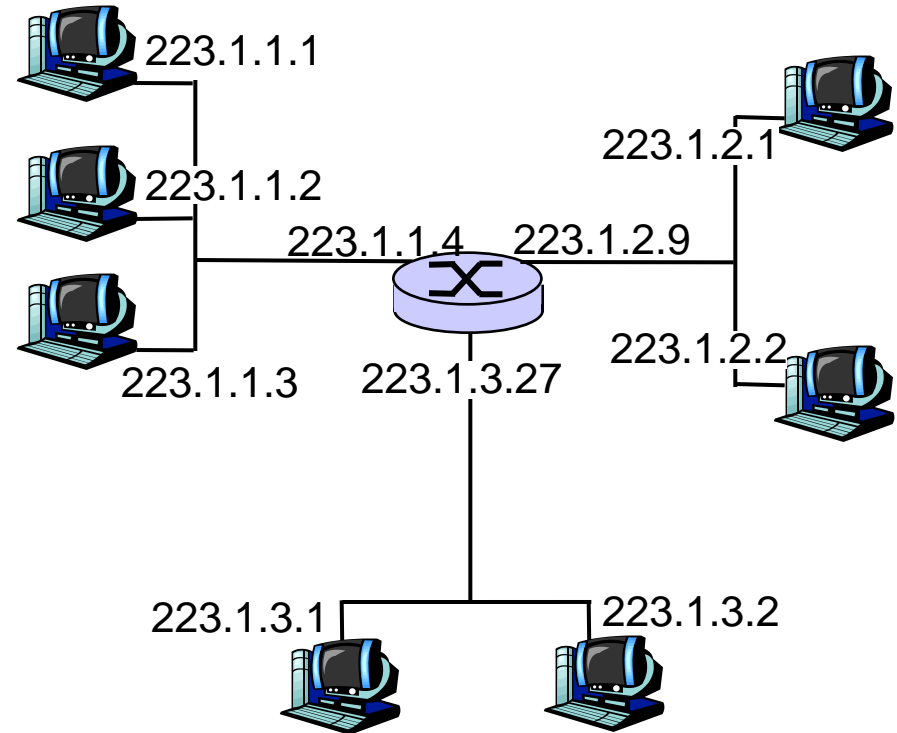
A Camada de Rede na Internet

Funções da camada de rede em estações, roteadores:



Endereçamento IP: introdução

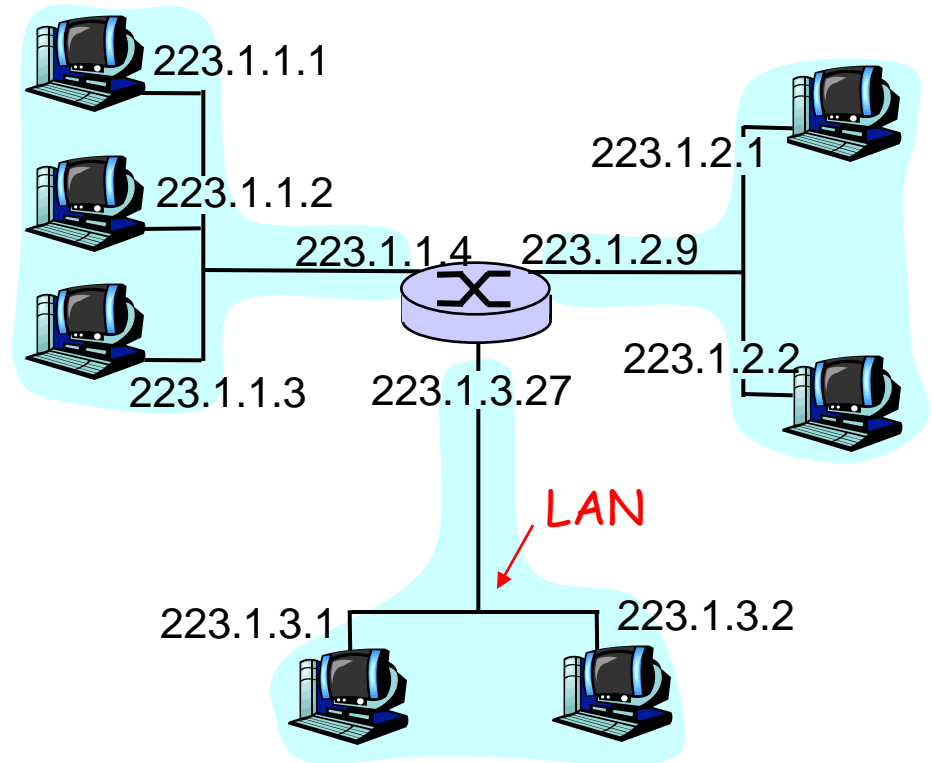
- endereço IP: ident. de 32-bits para *interface* de estação, roteador
- *interface*: conexão entre estação, roteador e enlace físico
 - roteador típico tem múltiplas interfaces
 - estação pode ter múltiplas interfaces
 - endereço IP associado à interface, não à estação ou roteador



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

Endereçamento IP

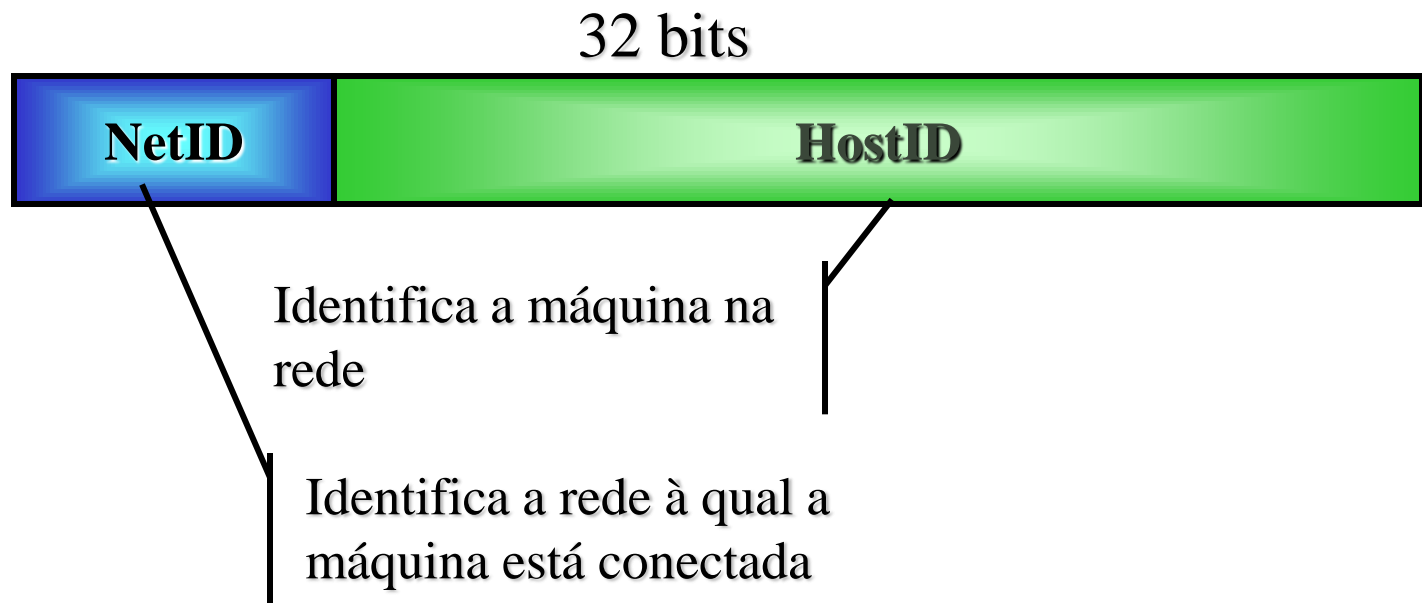
- endereço IP:
 - parte de rede (bits de mais alta ordem)
 - parte de estação (bits de mais baixa ordem)
- *O que é uma rede IP?*
(da perspectiva do endereço IP)
 - interfaces de dispositivos com a mesma parte de rede nos seus endereços IP
 - podem alcançar um ao outro sem passar por um roteador



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

Endereçamento IP

□ Endereçamento hierárquico

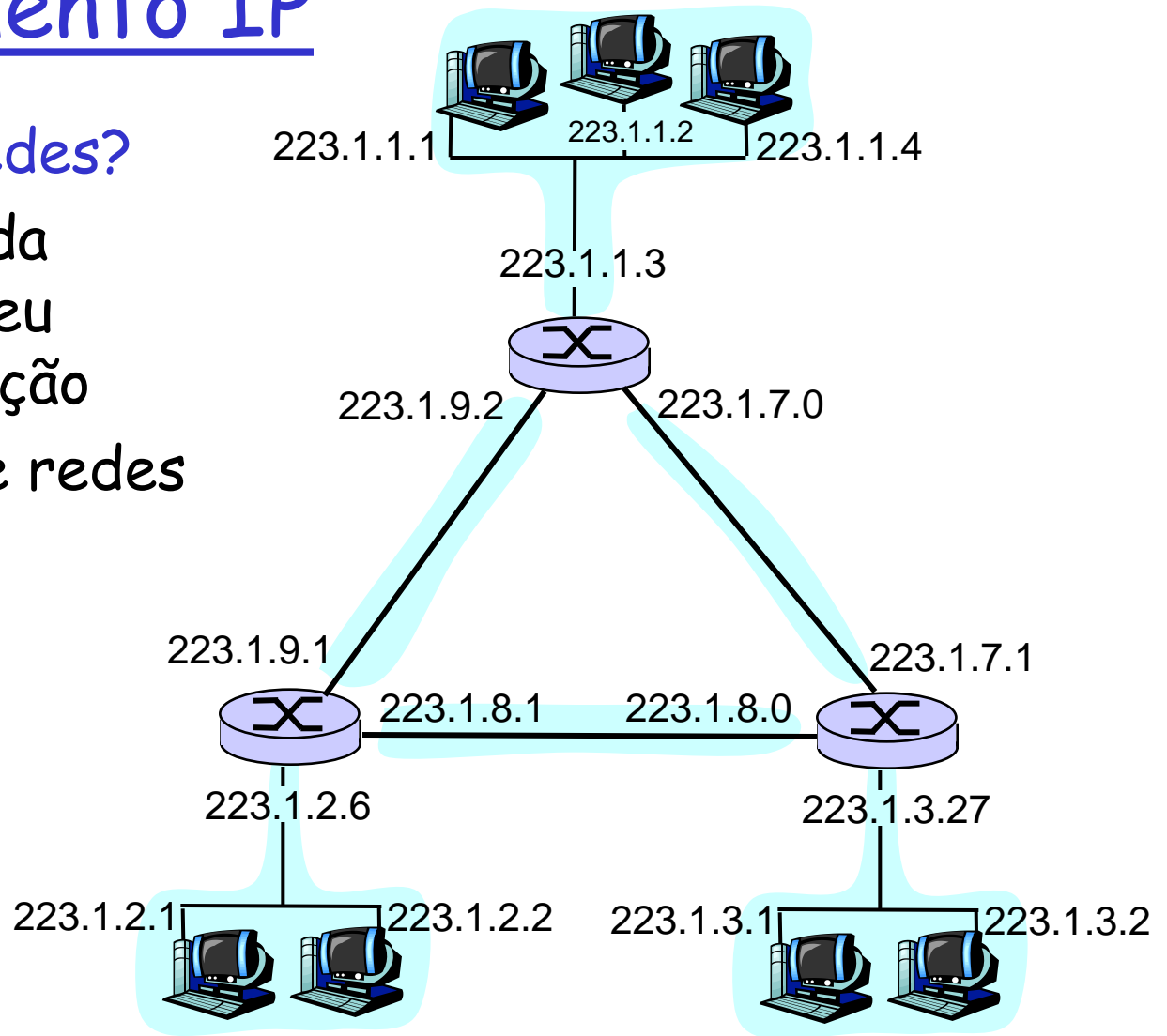


223.1.1.1 = $\underbrace{11011111}_{223}$ $\underbrace{00000001}_1$ $\underbrace{00000001}_1$ $\underbrace{00000001}_1$

Endereçamento IP

Como achar as redes?

- ❑ disassociar 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":

classe

A	0	rede		estação		1.0.0.0 to 127.255.255.255
B	10		rede		estação	128.0.0.0 to 191.255.255.255
C	110		rede		estação	192.0.0.0 to 223.255.255.255
D	1110			endereço multiponto		224.0.0.0 to 239.255.255.255

← 32 bits →

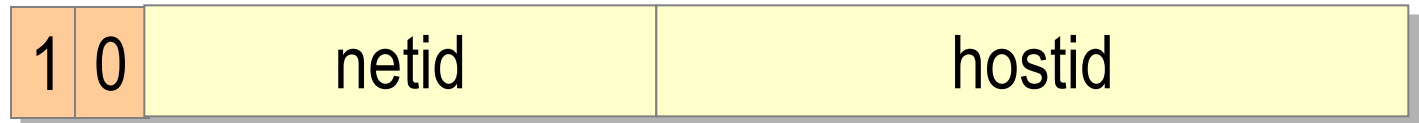
Endereçamento IP

□ Classes de endereços IP

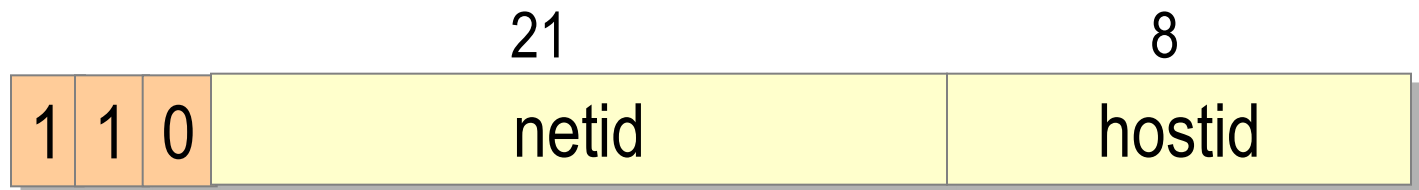
- Classe A: 7 24



- Classe B: 14 16



- Classe C:

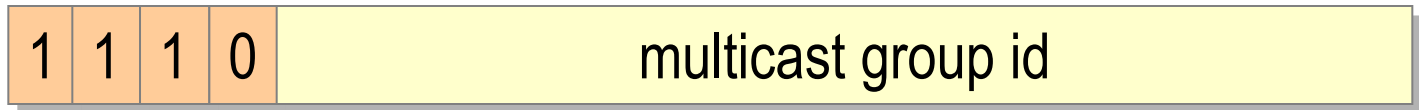


Endereçamento IP

□ Classes de endereços IP

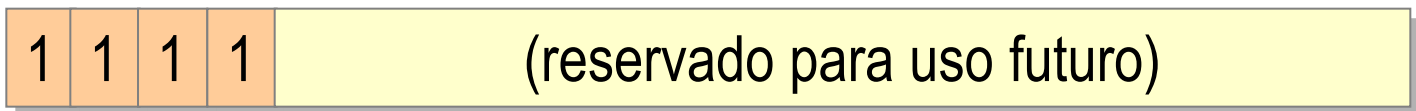
- Classe D (multicast):

28



- Classe E (reservado):

28



Endereçamento IP

□ Classes de endereços IP

Classe	Faixa
A	0.0.0.0 a 127.255.255.255
B	128.0.0.0 a 191.255.255.255
C	192.0.0.0 a 223.255.255.255
D	224.0.0.0 a 239.255.255.255
E	240.0.0.0 a 255.255.255.255

Endereçamento IP

□ Classes de endereços IP

- Classe A: 126 redes com aprox. 16 milhões de hosts
- Classe B: 16 mil redes com aprox. 65 mil hosts
- Classe C: 2 milhões de redes com 256 hosts

□ Endereços reservados (privados):

- Classe A: 10.X.X.X
- Classe B: 172.16.X.X a 172.31.X.X
- Classe C: 192.168.X.X

Endereçamento IP

- Endereços especiais:
 - **127.X.X.X**: Endereço de *loopback*
 - Usado para testes na própria máquina
 - **hostid"todo zero"**: Endereço da rede
 - Identifica a rede como um todo (nenhum host em especial)
 - Utilizado para simplificar as tabelas de rotas dos roteadores
 - Exemplo: 200.129.68.**0** → rede do CEFET-PB
 - **hostid"todo um"**: Endereço de *broadcast*
 - Usado para enviar um pacote para todos os hosts da rede
 - Exemplo: 200.129.68.**255** → todos os hosts do CEFET-PB

Formato do Datagrama IP

versão do Protocolo IP

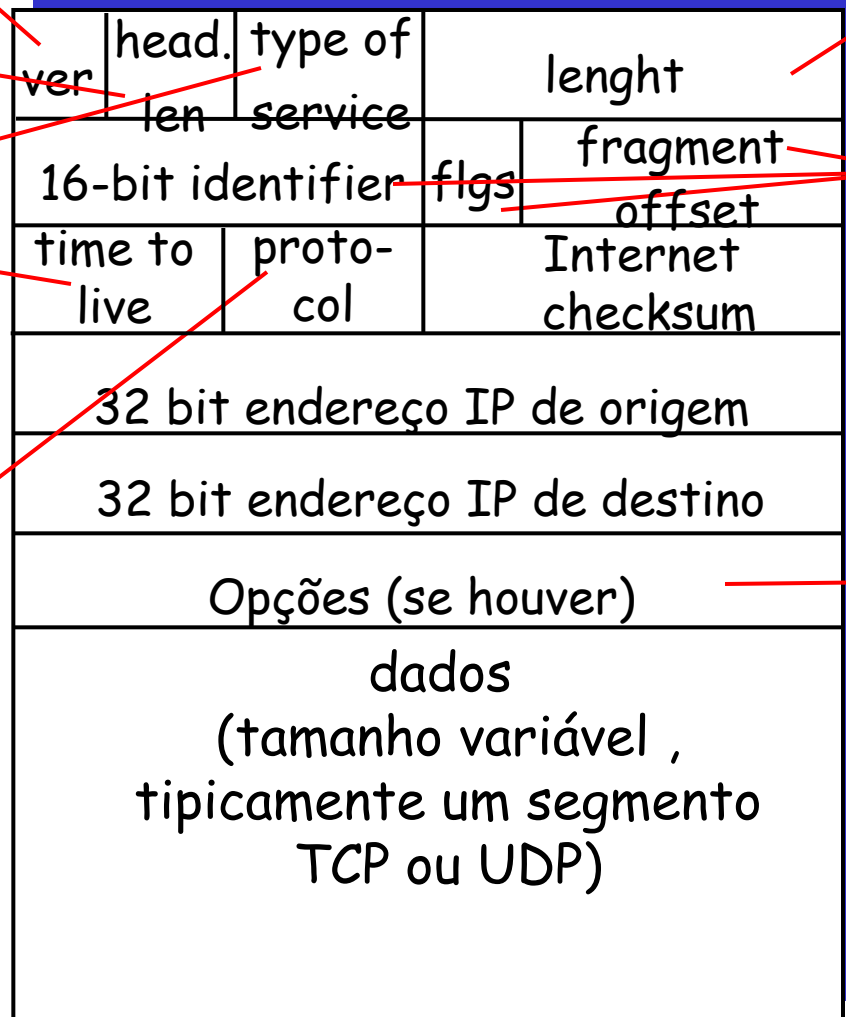
tamanho do header
(bytes)

Classe de serviço

número máximo
de saltos
(decrementado em
cada roteador)

Protocolo da camada
superior com dados no
datagrama

32 bits



tamanho total
do datagrama
(bytes)

para
fragmentação/
remontagem

Ex. timestamp,
registro de rota
lista de rotea-
dores a visitar.

Levando um Datagrama da Fonte ao Destino

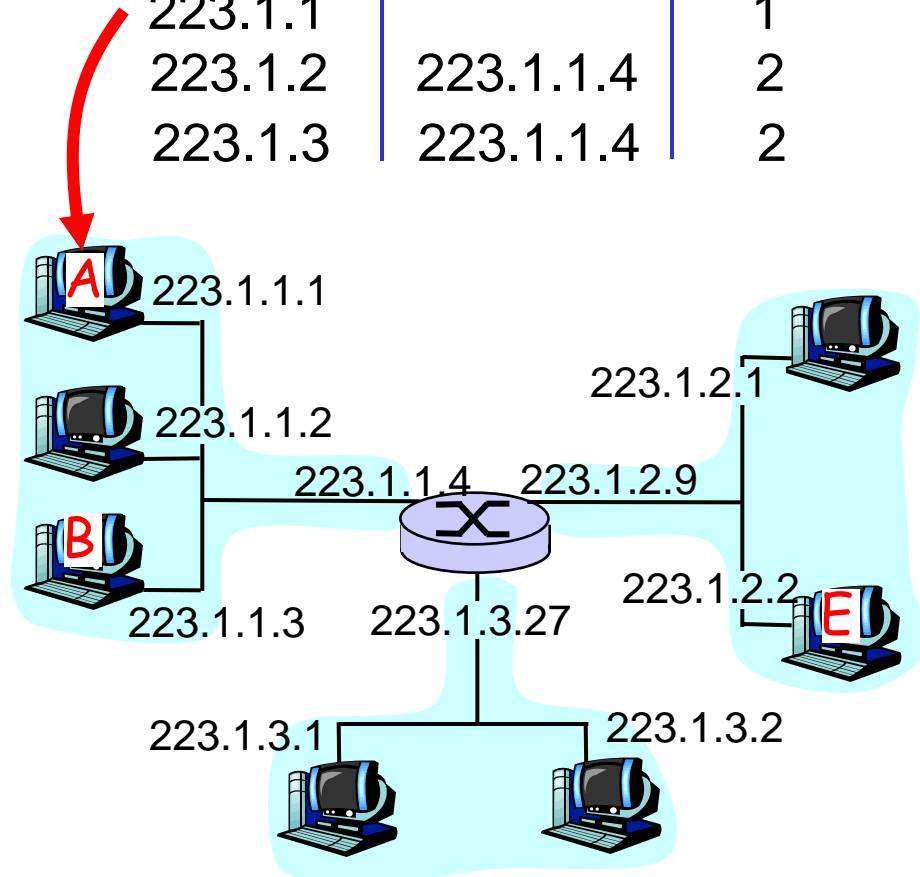
datagrama IP:

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



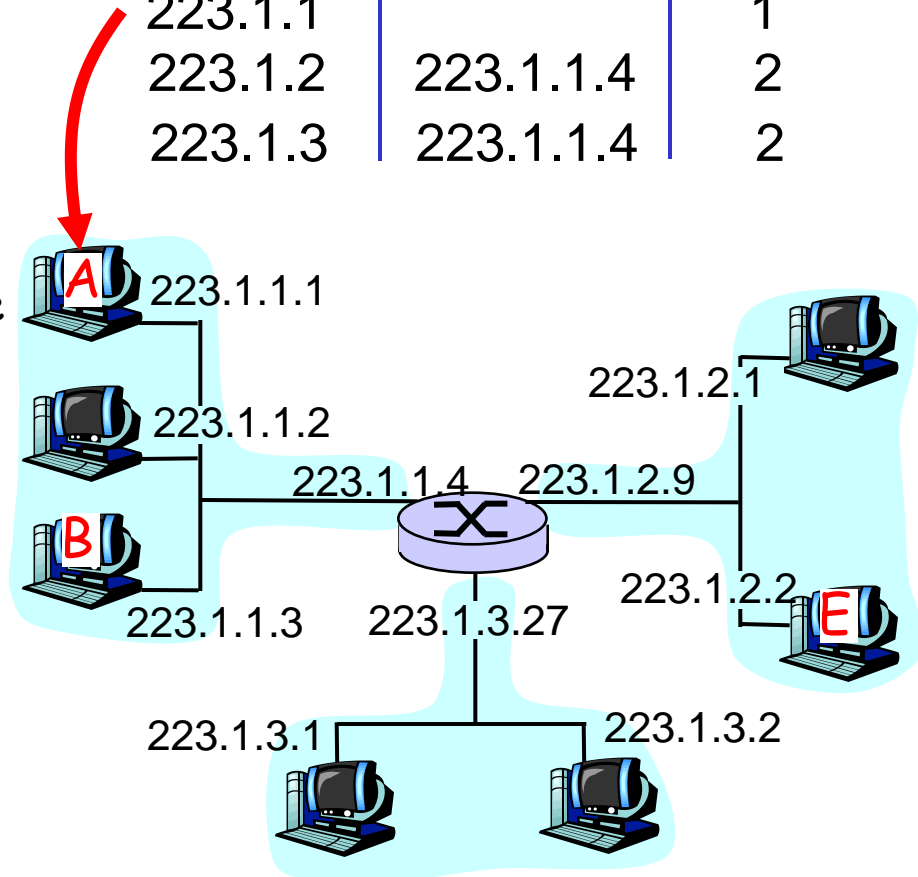
Levando um Datagrama da Fonte ao Destino

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
- ❑ 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

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



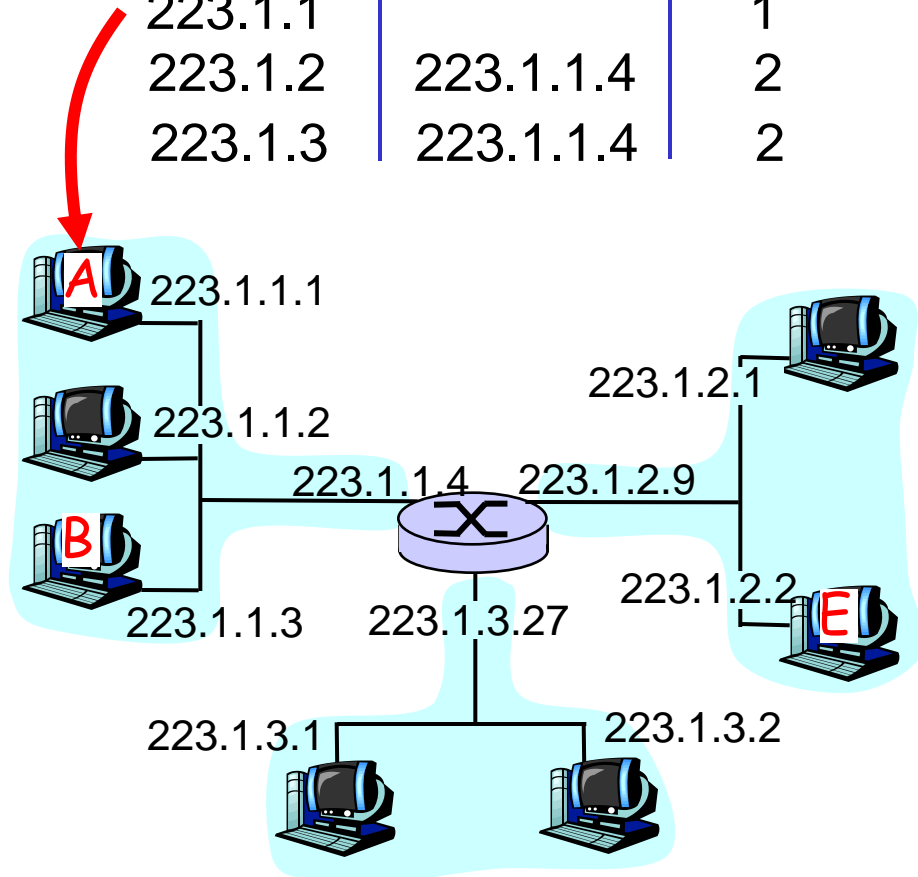
Levando um Datagrama da Fonte ao Destino

outros campos	223.1.1.1	223.1.2.2	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



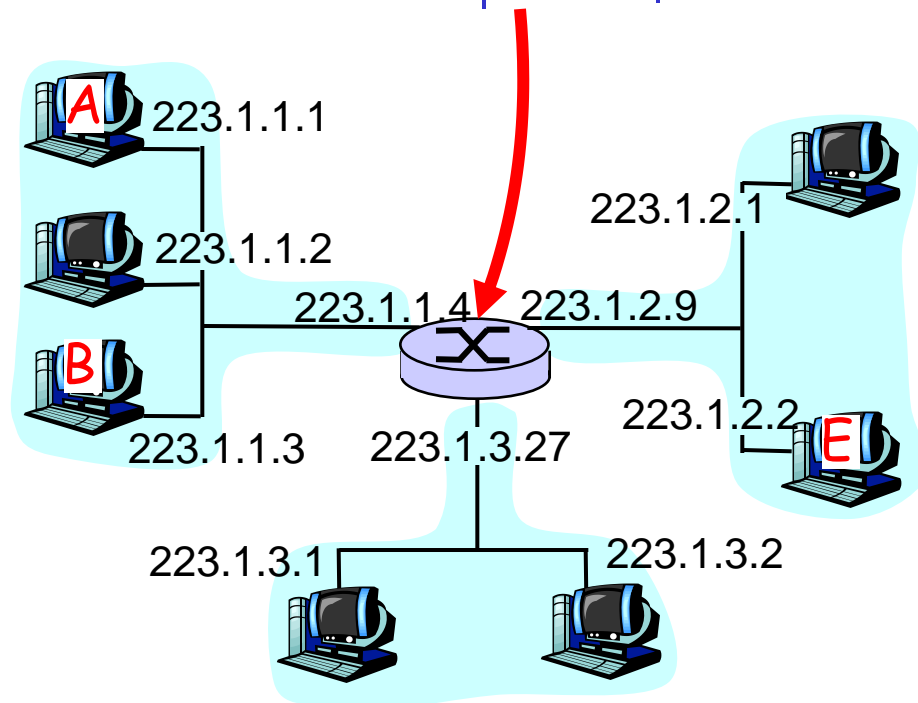
Levando um Datagrama da Fonte ao Destino

outro campos	223.1.1.1	223.1.2.2	dados
--------------	-----------	-----------	-------

Chegando em 223.1.1.4,
destinado para 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



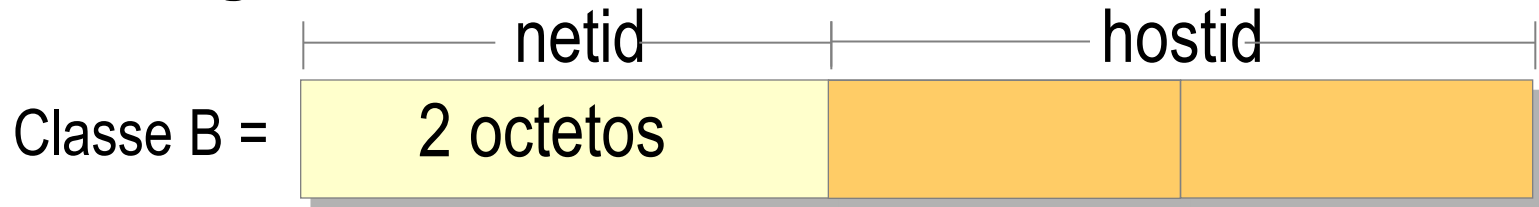
Sub-rede

- ❑ Os endereços classe A e B são mal dimensionados (muitos hosts na rede)
- ❑ O que é subnetting?
 - Forma de aproveitar melhor os endereços IP
 - Forma de segmentar a rede internamente, enquanto mantém sua unidade externamente
 - Forma de criar domínios de *broadcast* numa rede IP

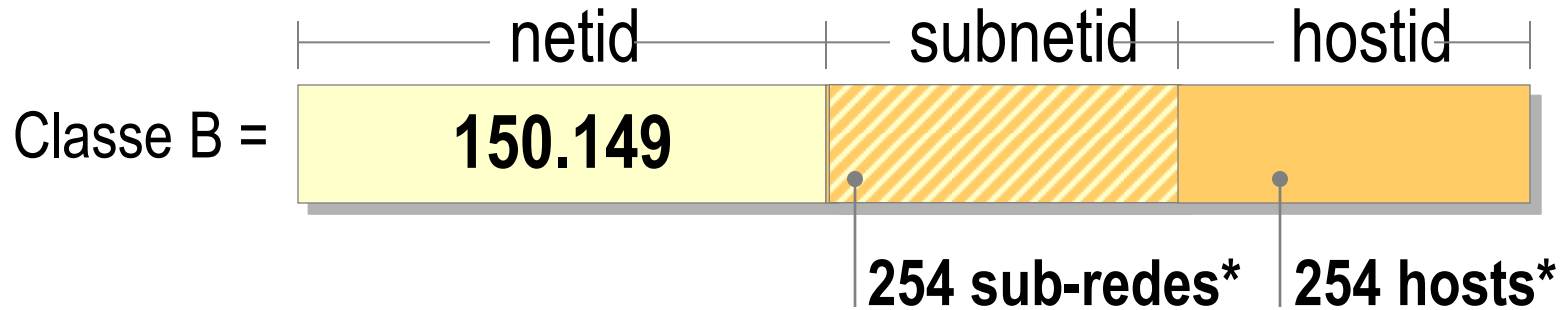
Sub-rede

❑ Exemplo: um endereço classe B

- Original: **150.149.X.X**



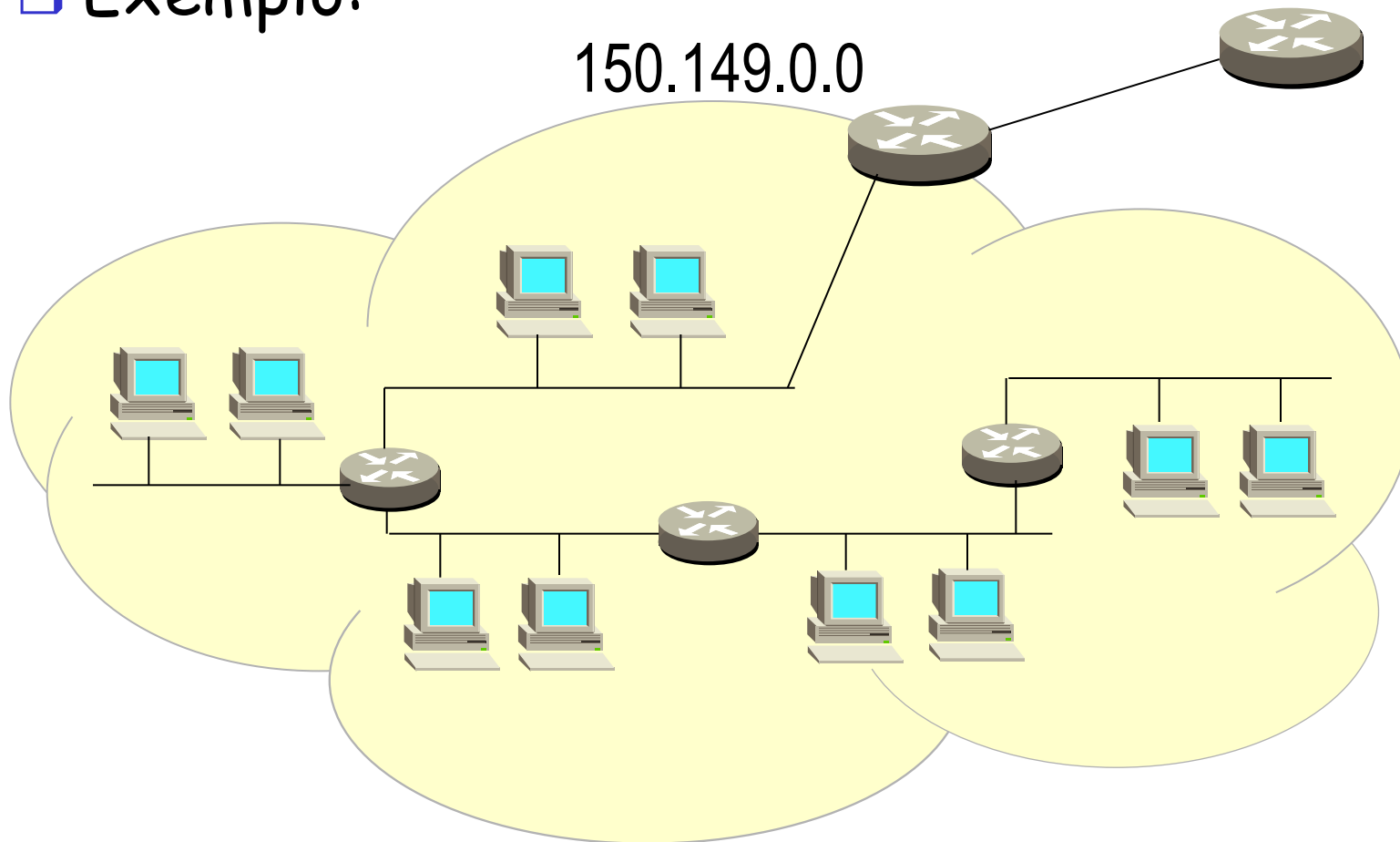
- Subnetado:



*Os endereços 0 e 255 não são utilizados.

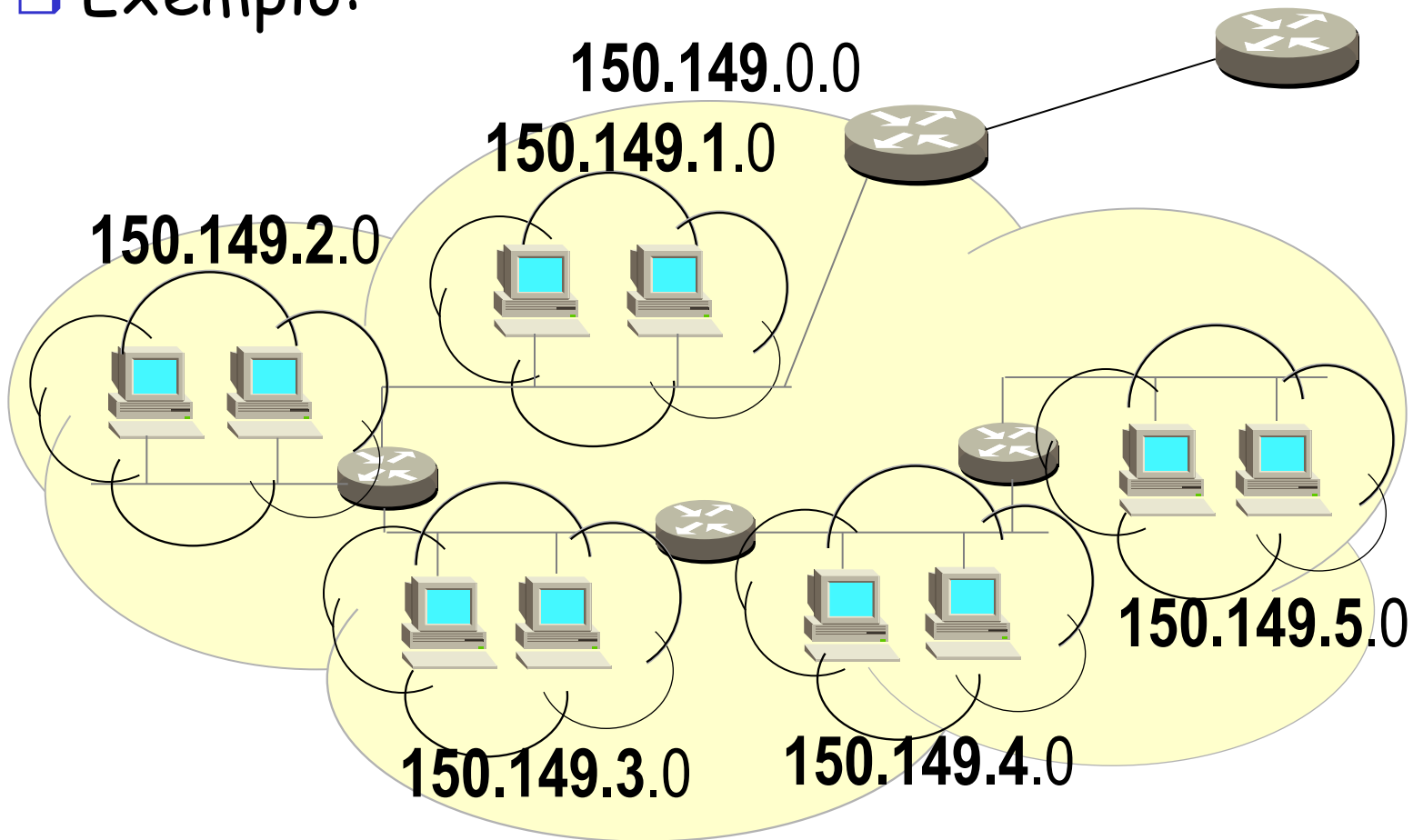
Sub-rede

□ Exemplo:



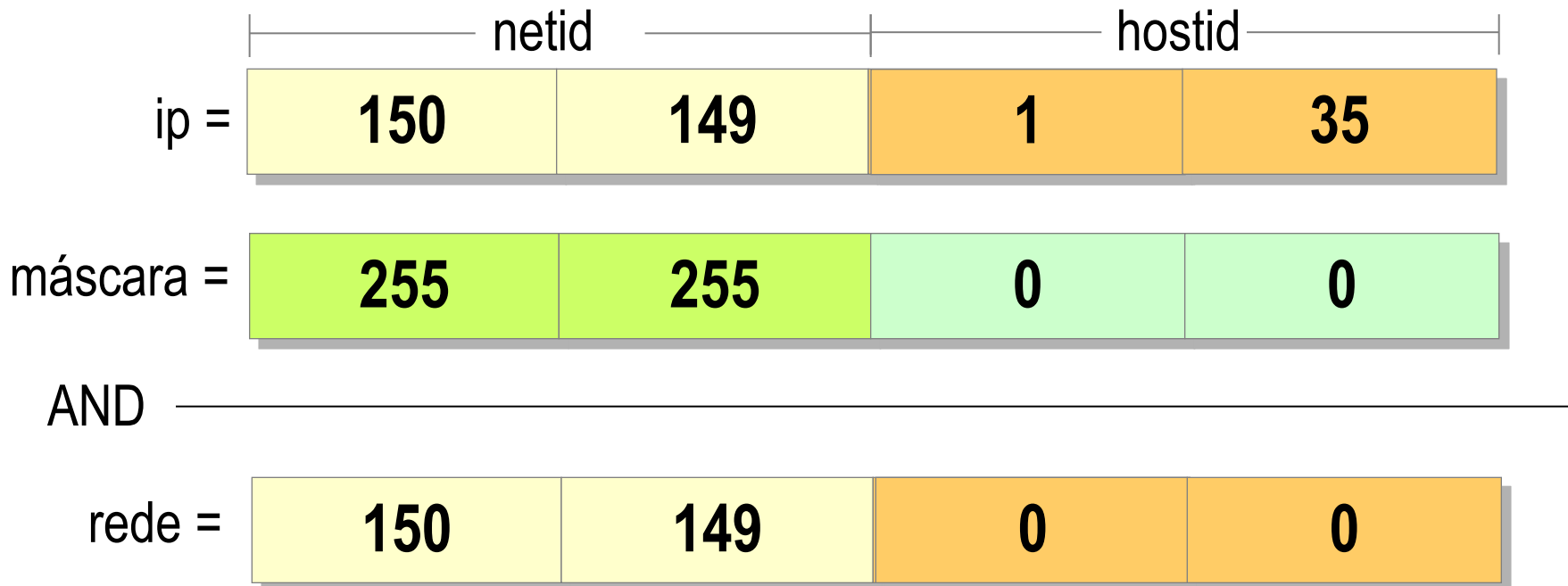
Sub-rede

□ Exemplo:



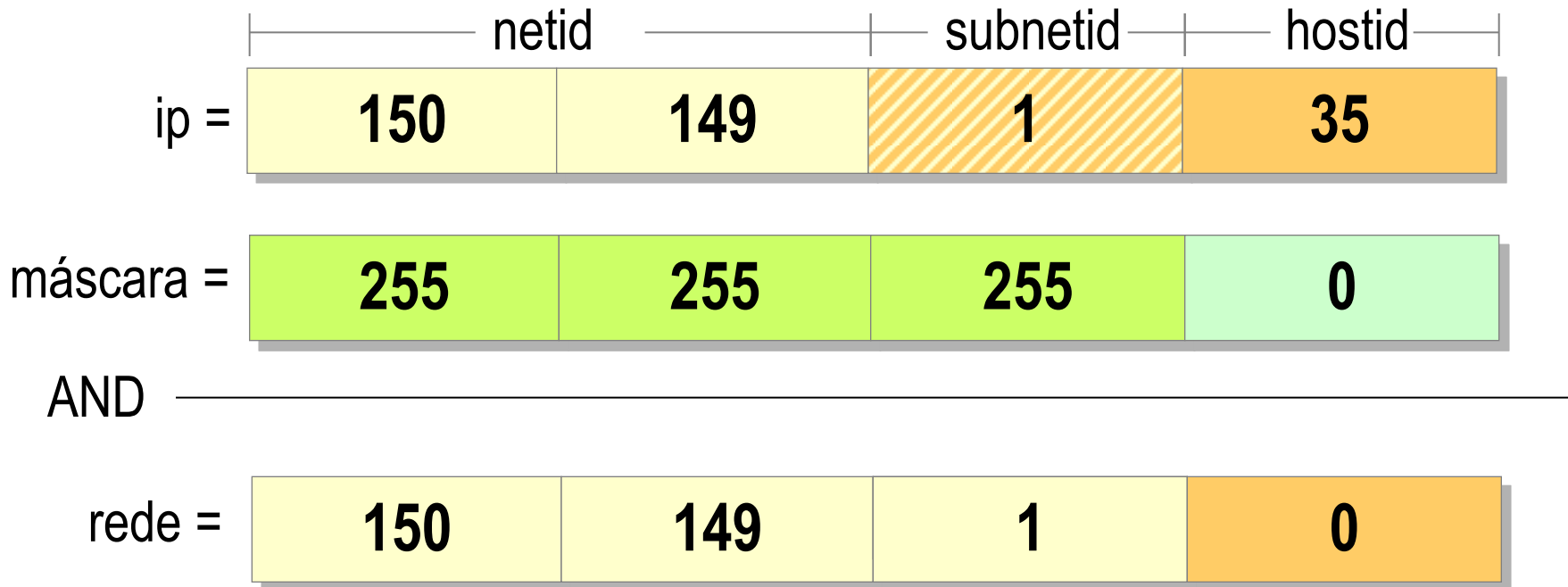
Sub-rede

- ❑ Como fazer para identificar hostid e netid?
 - Máscara de subrede de um classe B comum:



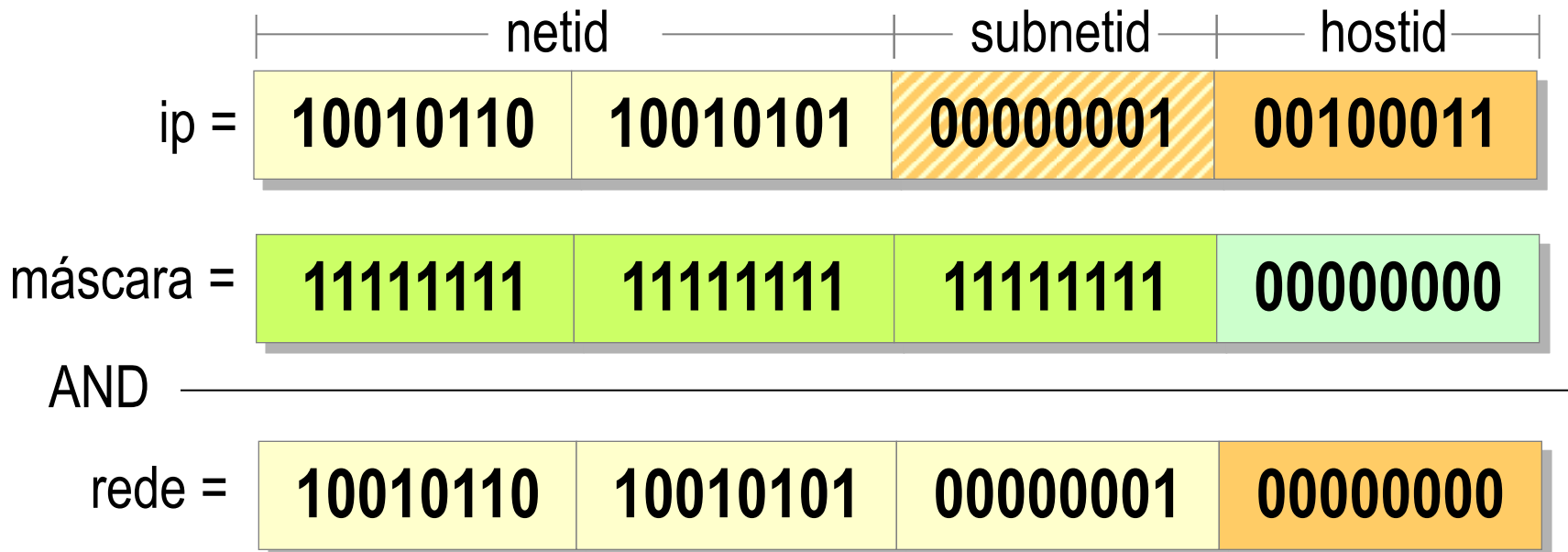
Sub-rede

- ❑ Como fazer para identificar hostid e netid?
 - Máscara de subrede de um classe B subnetado:



Sub-rede

- ❑ Como fazer para identificar hostid e netid?
 - Mesma coisa, só que em binário:



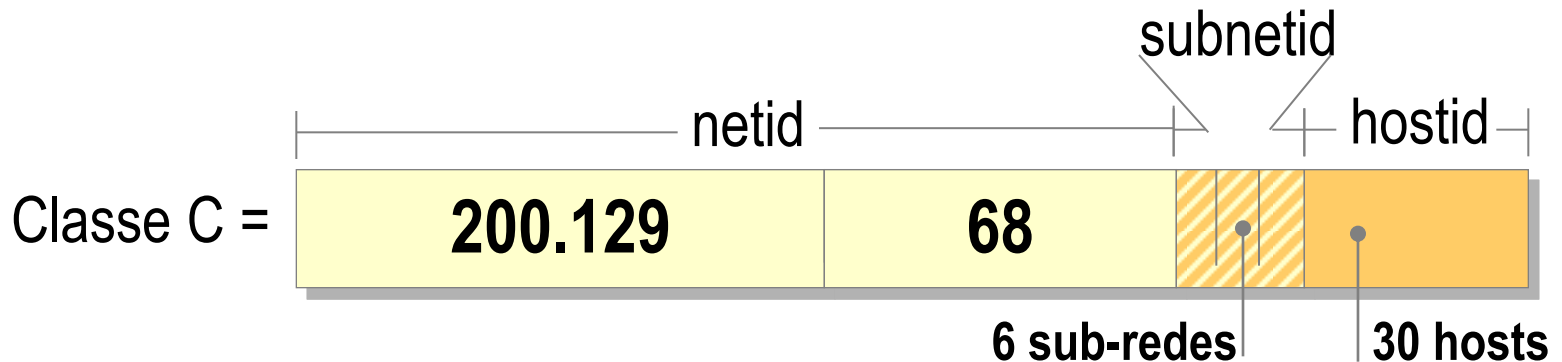
Sub-rede

- ❑ A máscara indica quais bits são parte do netid (1) e quais são parte do hostid (0)
- ❑ Toda máquina executa um AND entre o endereço IP e a máscara para saber como rotear um datagrama (saber qual é a rede de destino)
- ❑ Roteadores externos à rede 150.149.0.0 utilizam a máscara 255.255.0.0 enquanto que os internos utilizam a máscara 255.255.255.0

Sub-rede

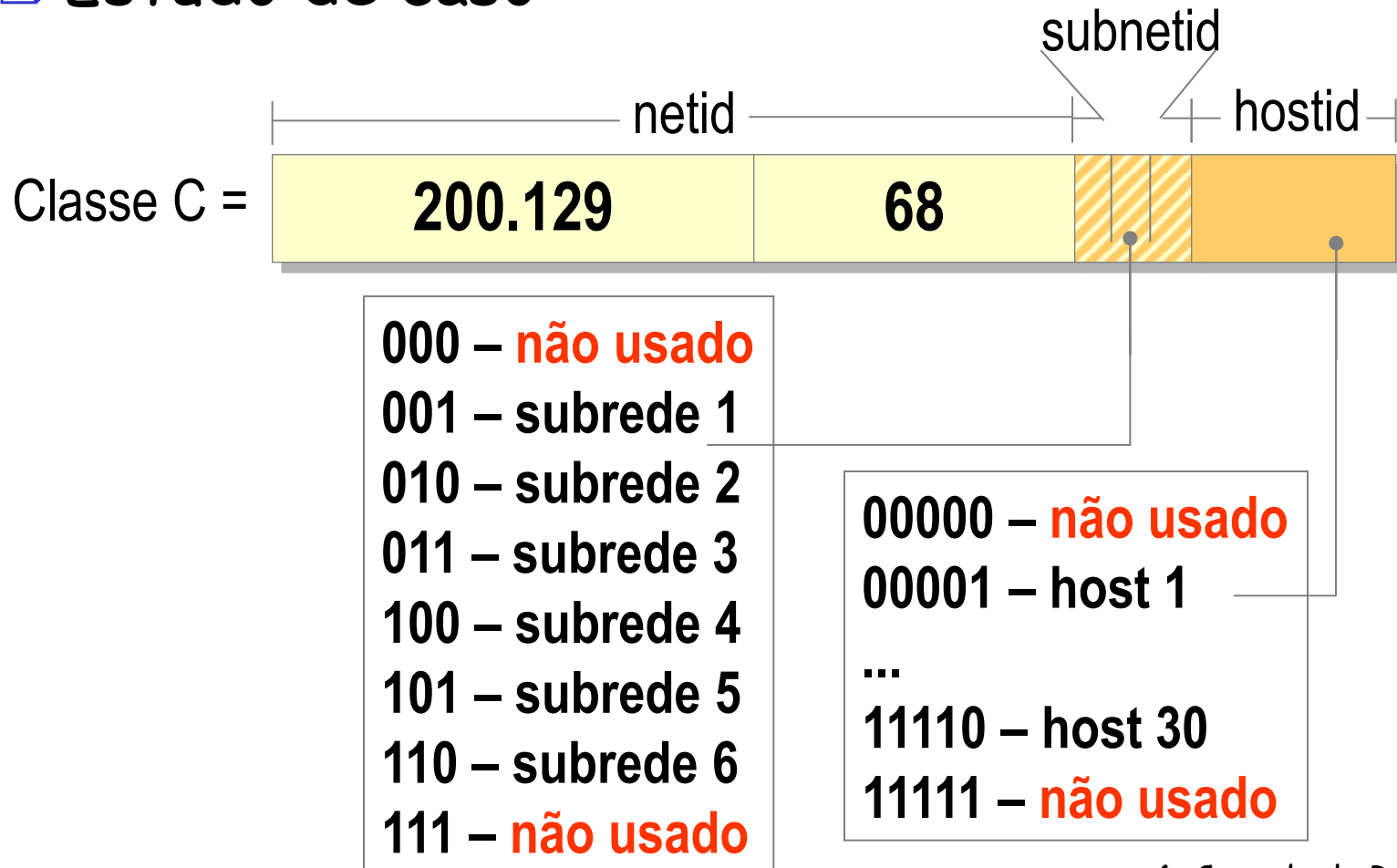
□ Estudo de caso:

- Uma empresa possui um endereço classe C (200.129.68.0) e deseja criar 6 sub-redes com 30 hosts cada. Como fazê-lo?



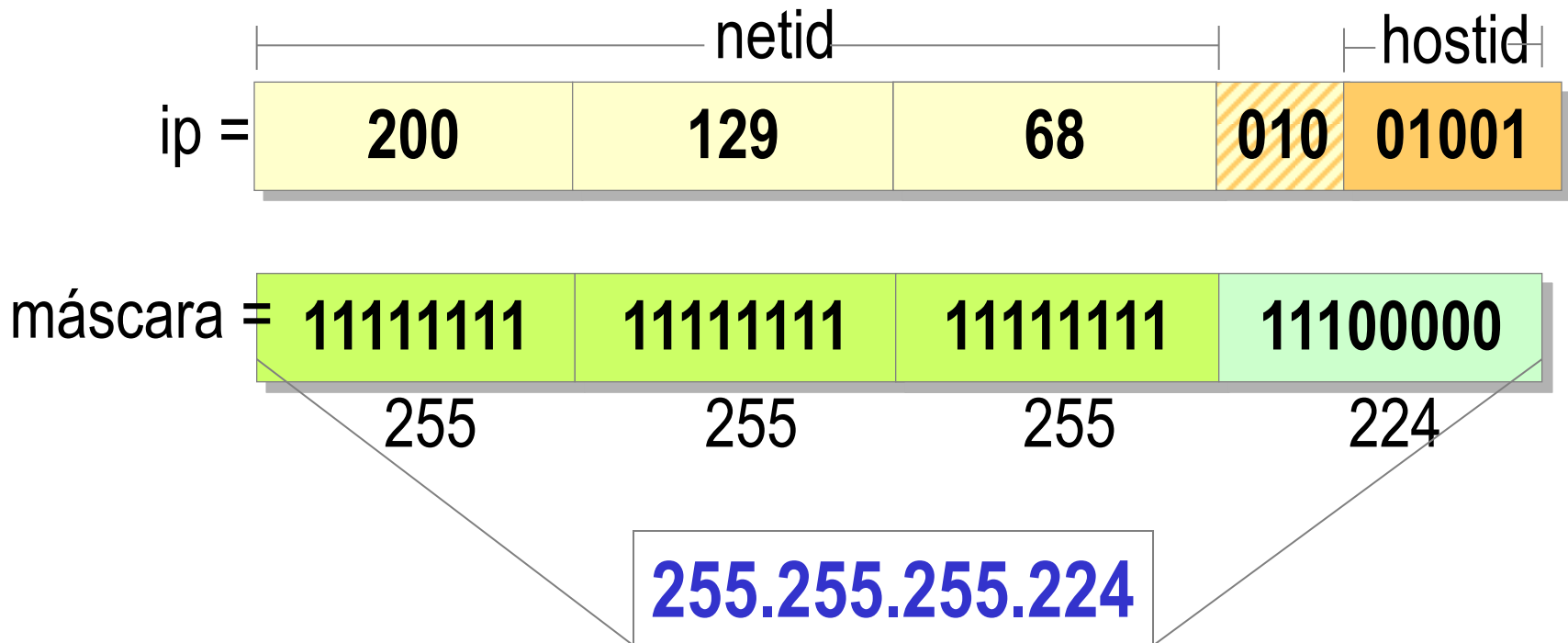
Sub-rede

Estudo de caso:



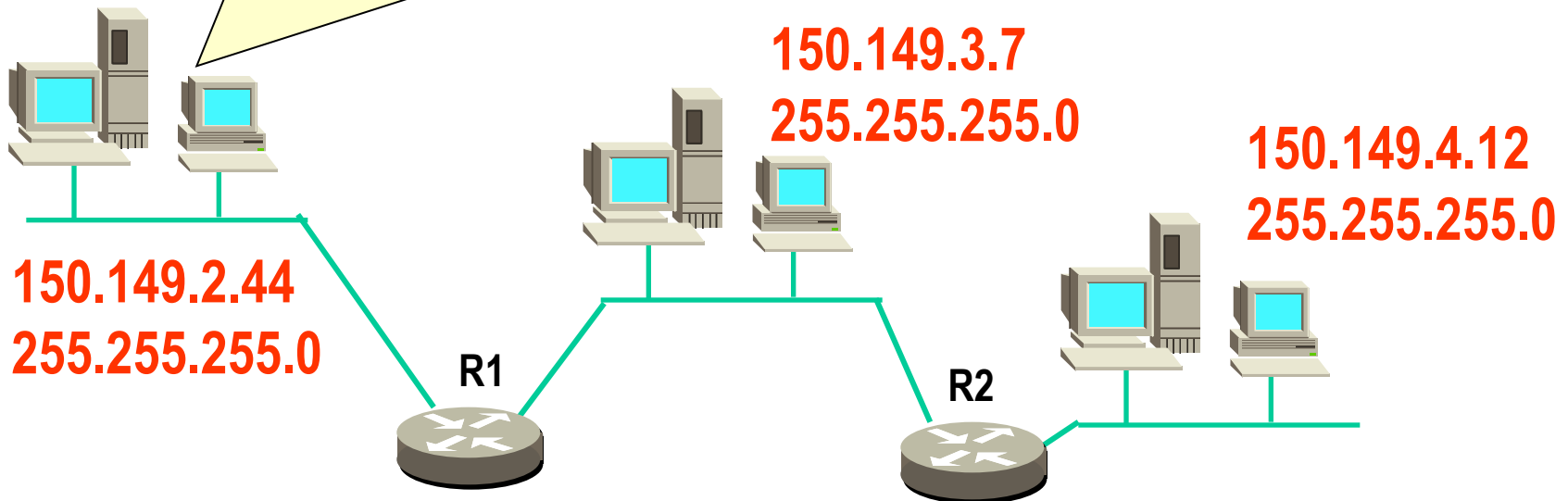
Sub-rede

- Como fica a máscara de rede?

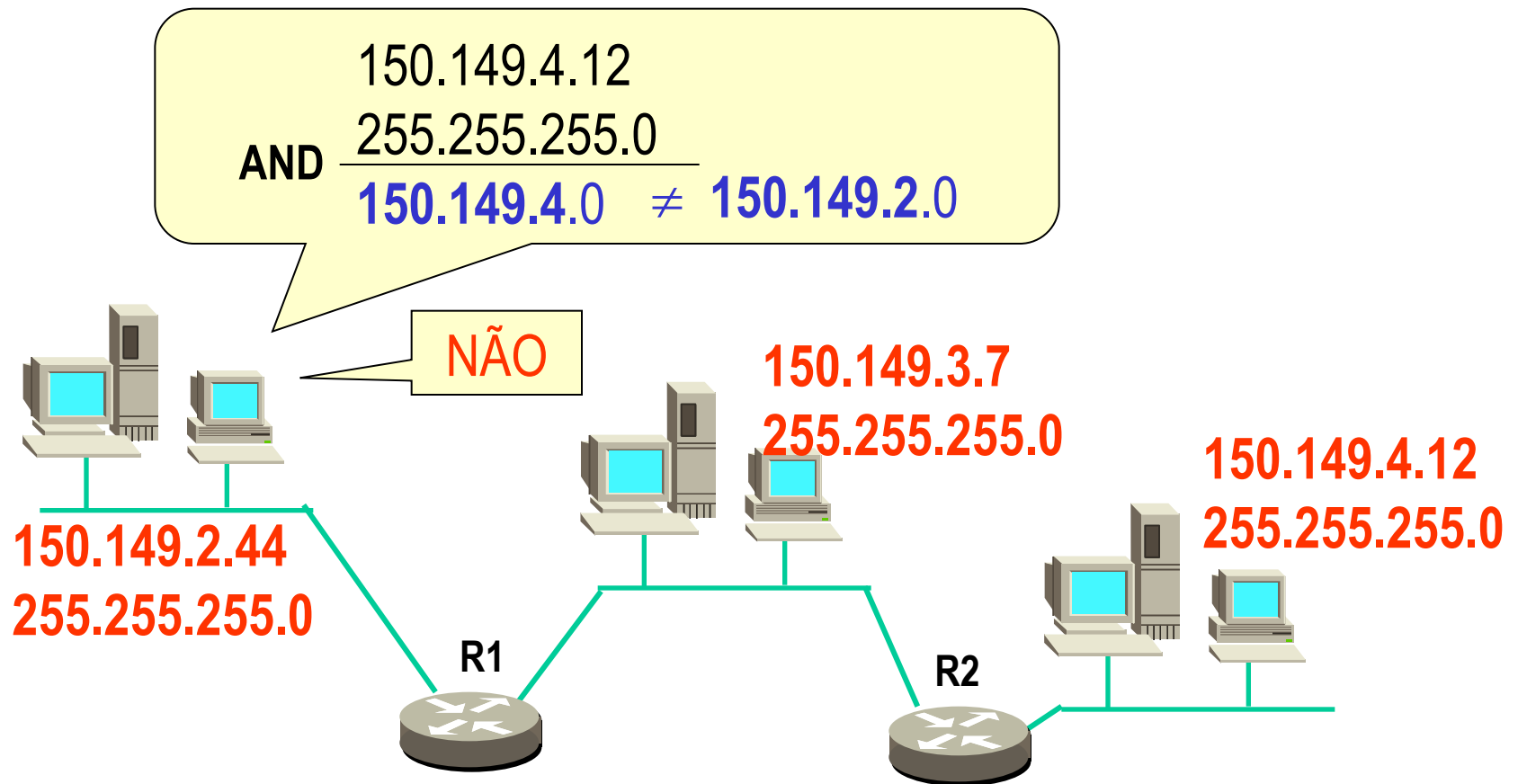


Roteamento

Tenho que enviar um datagrama para 150.149.4.12. Esta máquina pertence à minha rede IP?

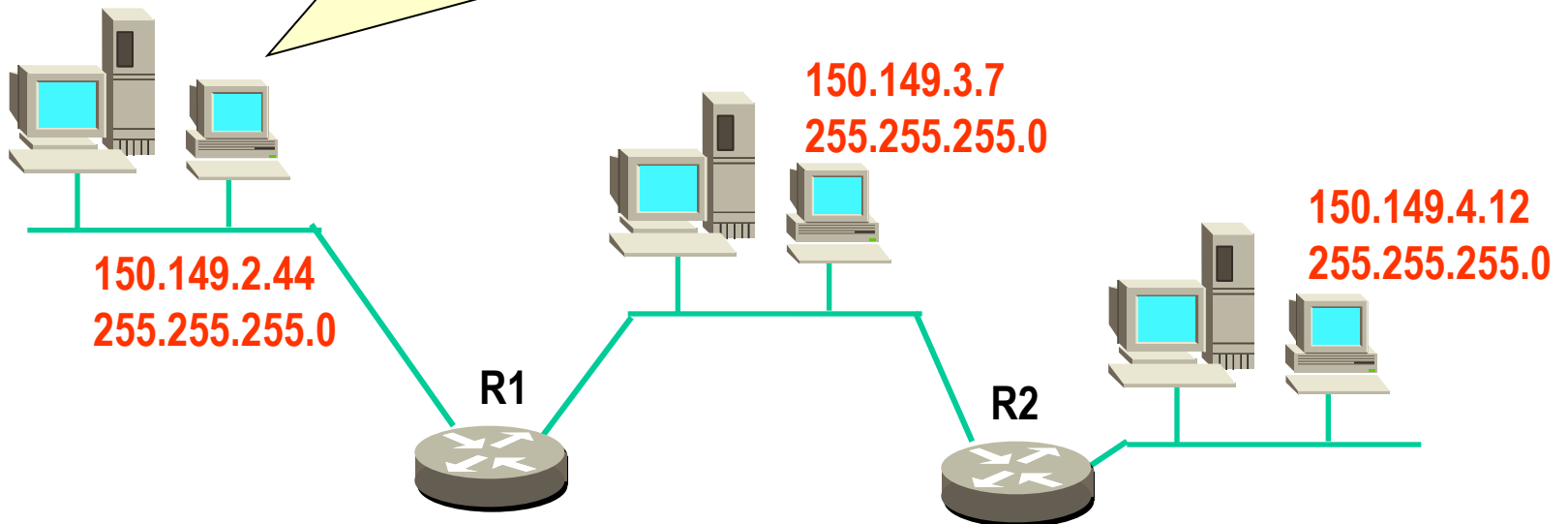


Roteamento

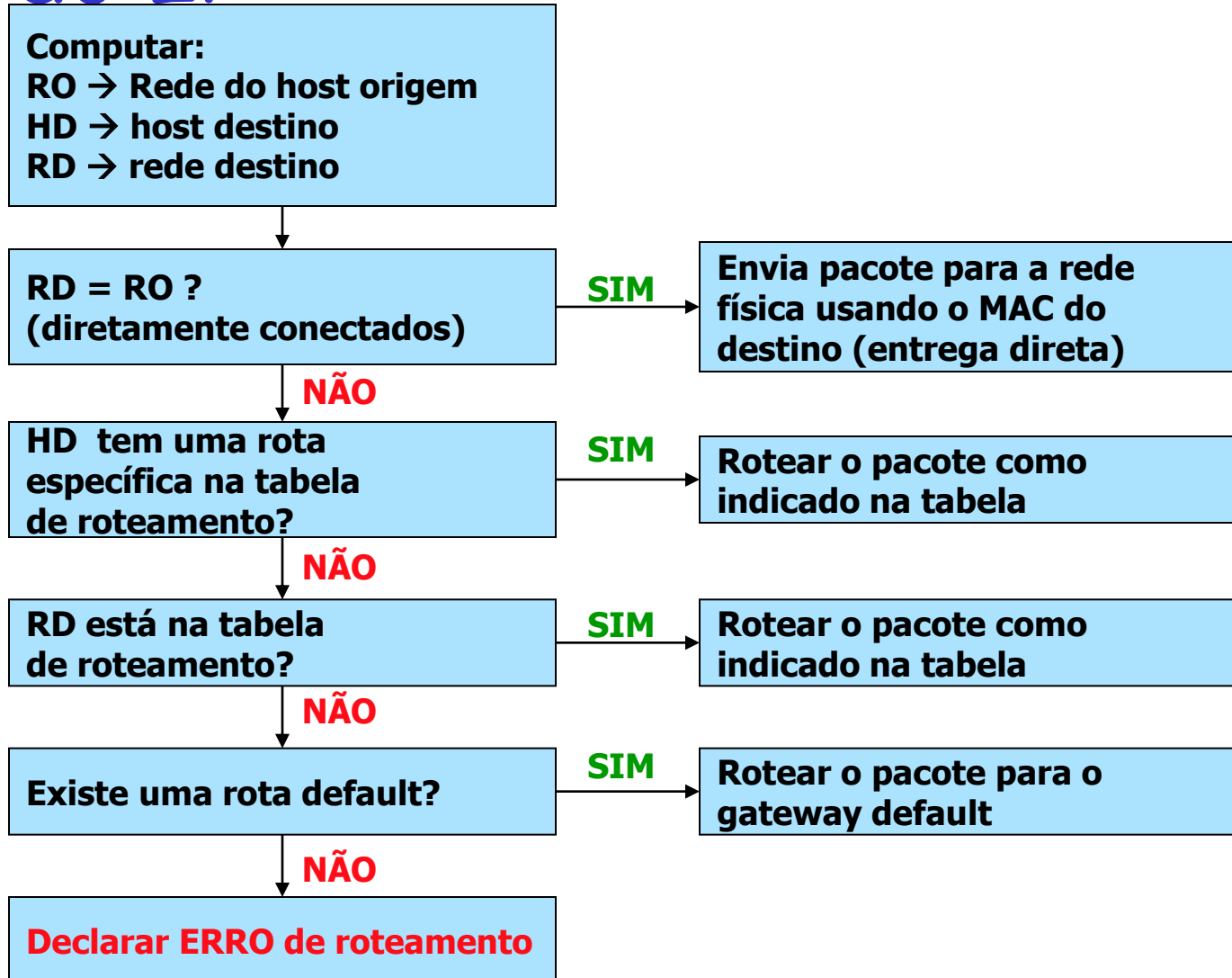


Roteamento

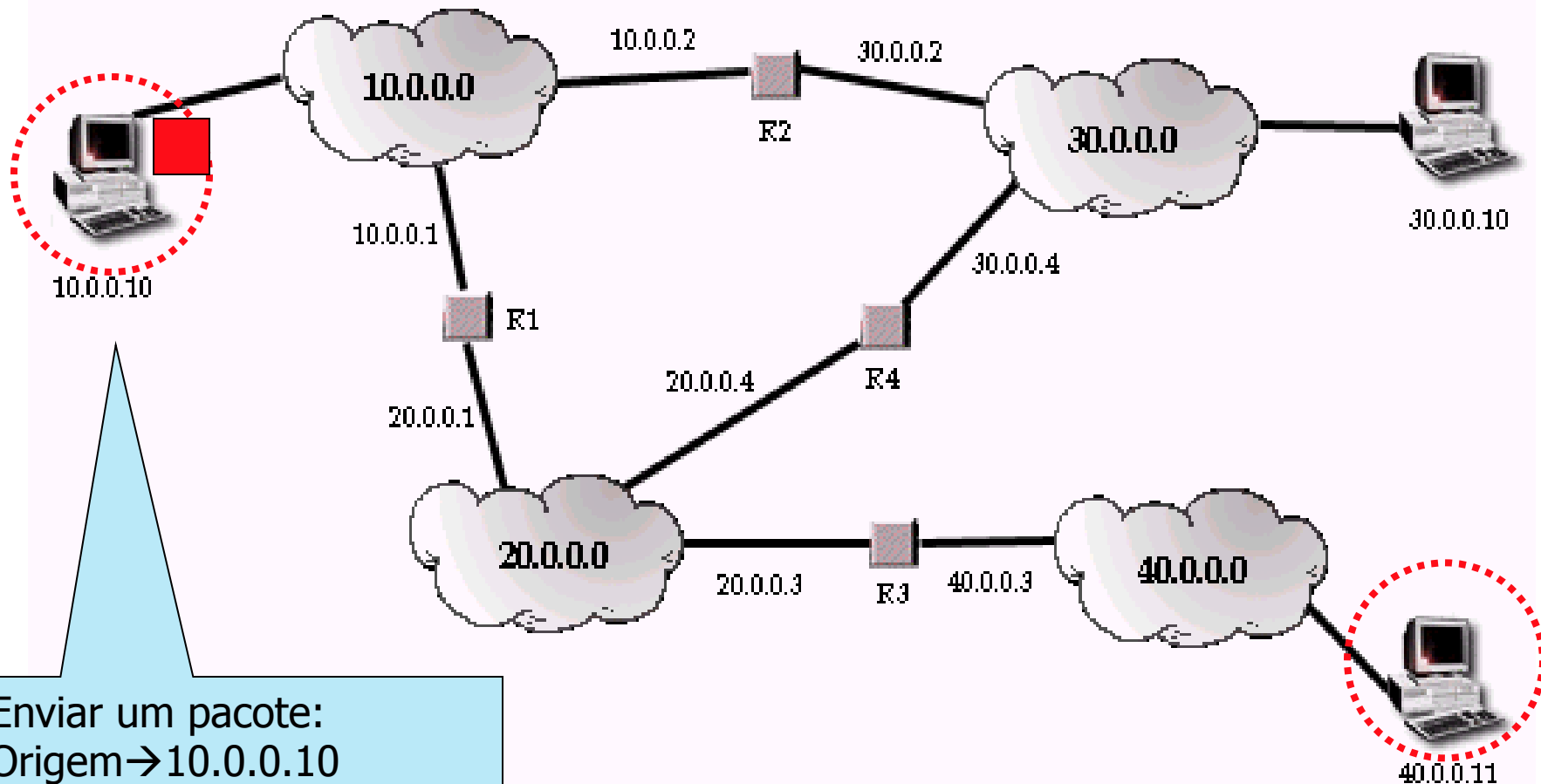
Envie o datagrama para o roteador R1, ele saberá encaminhá-lo ao seu destino!



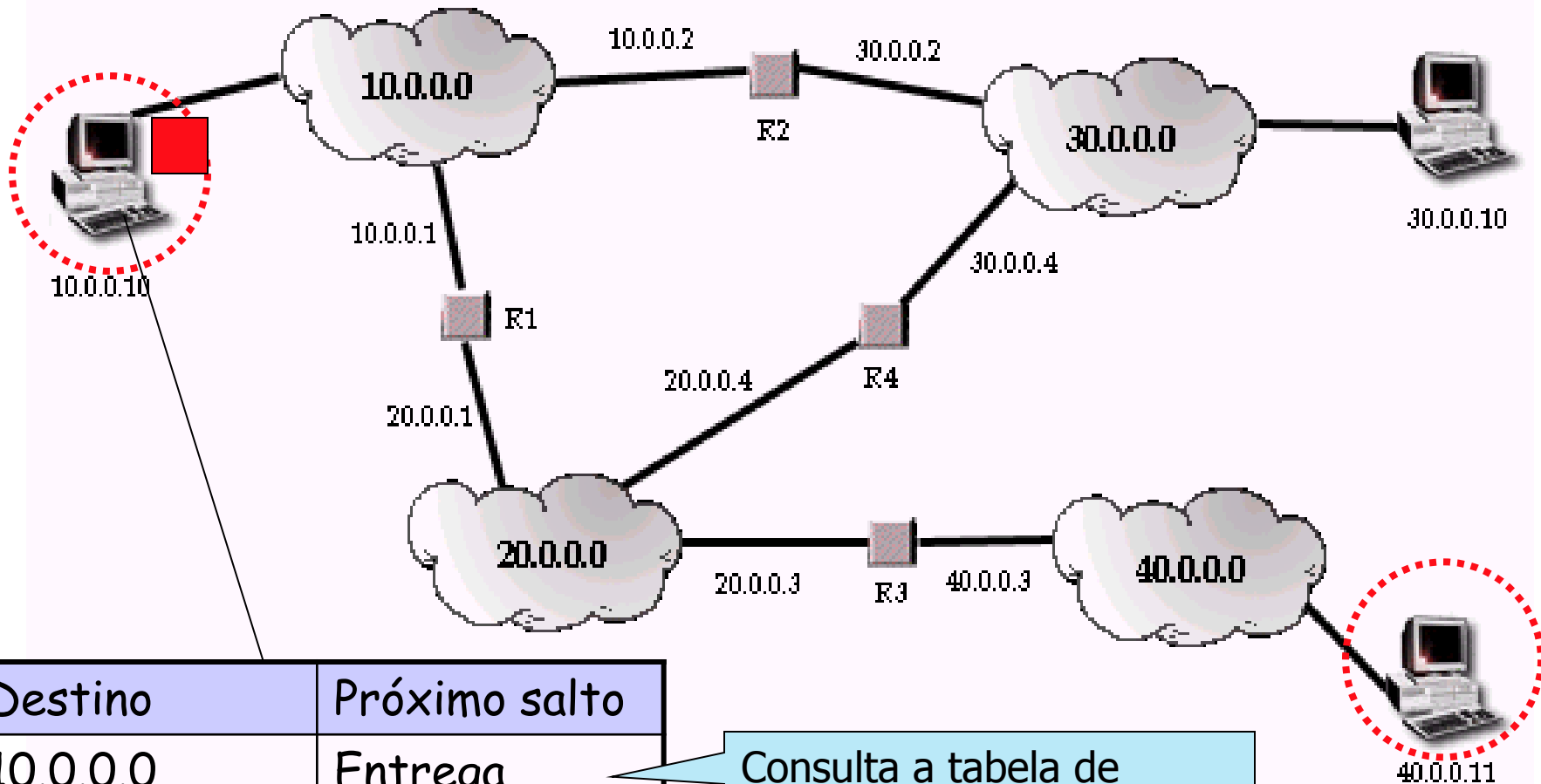
Algoritmo de Encaminhamento do IP



Encaminhamento IP - um exemplo prático



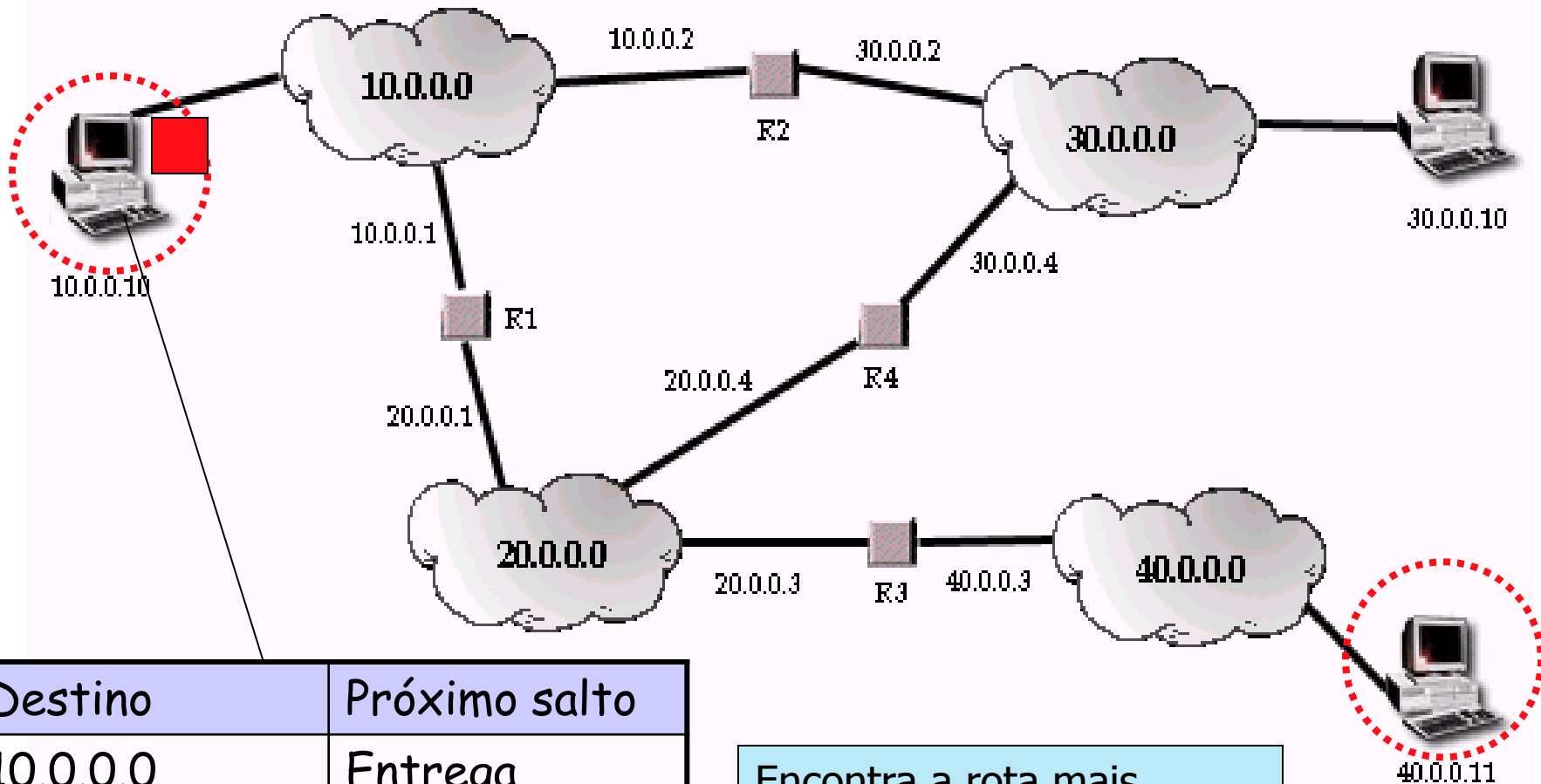
Encaminhamento IP



Destino	Próximo salto
10.0.0.0	Entrega Direta
Default	10.0.0.1

Consulta a tabela de roteamento

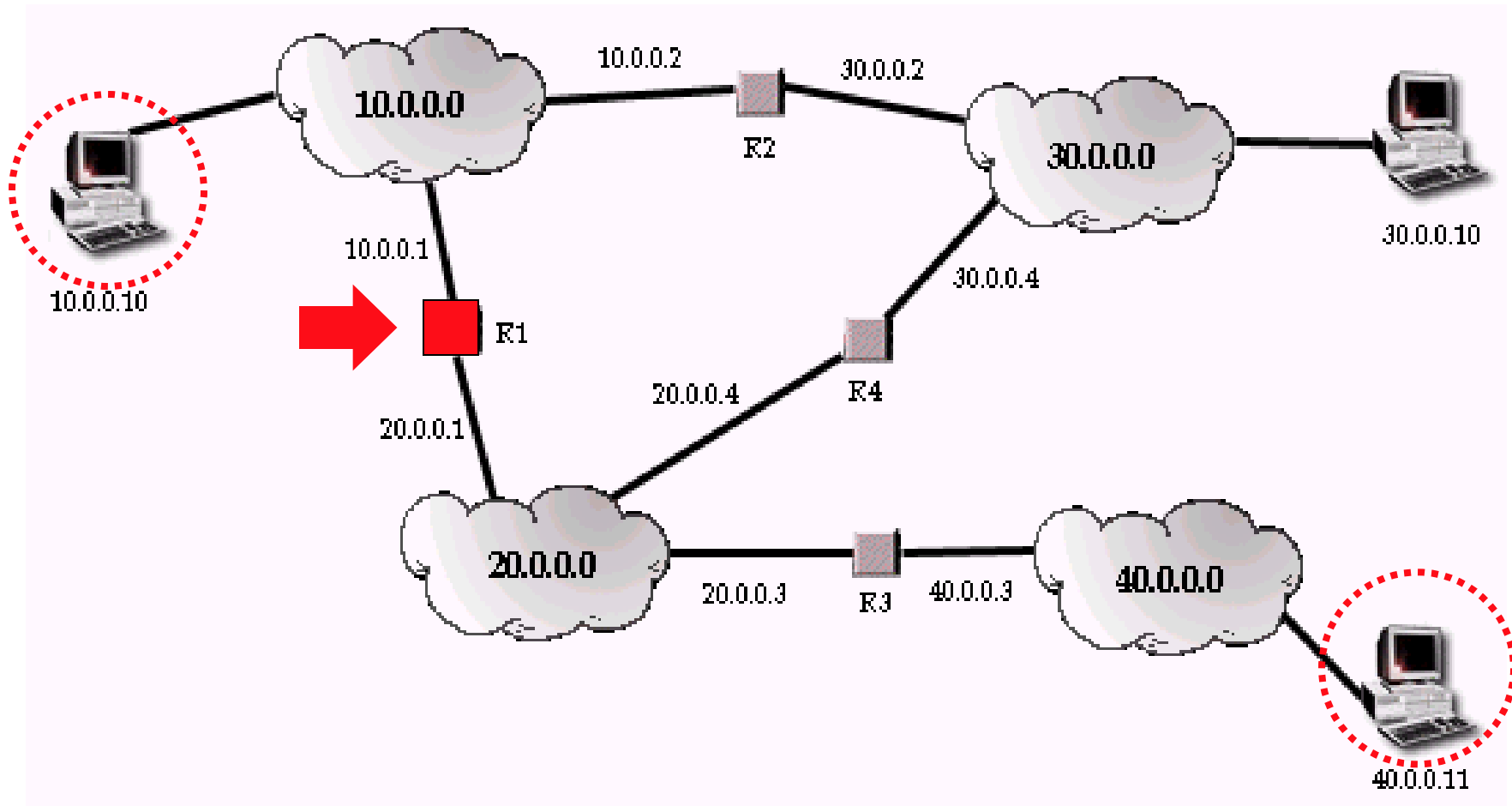
Encaminhamento IP



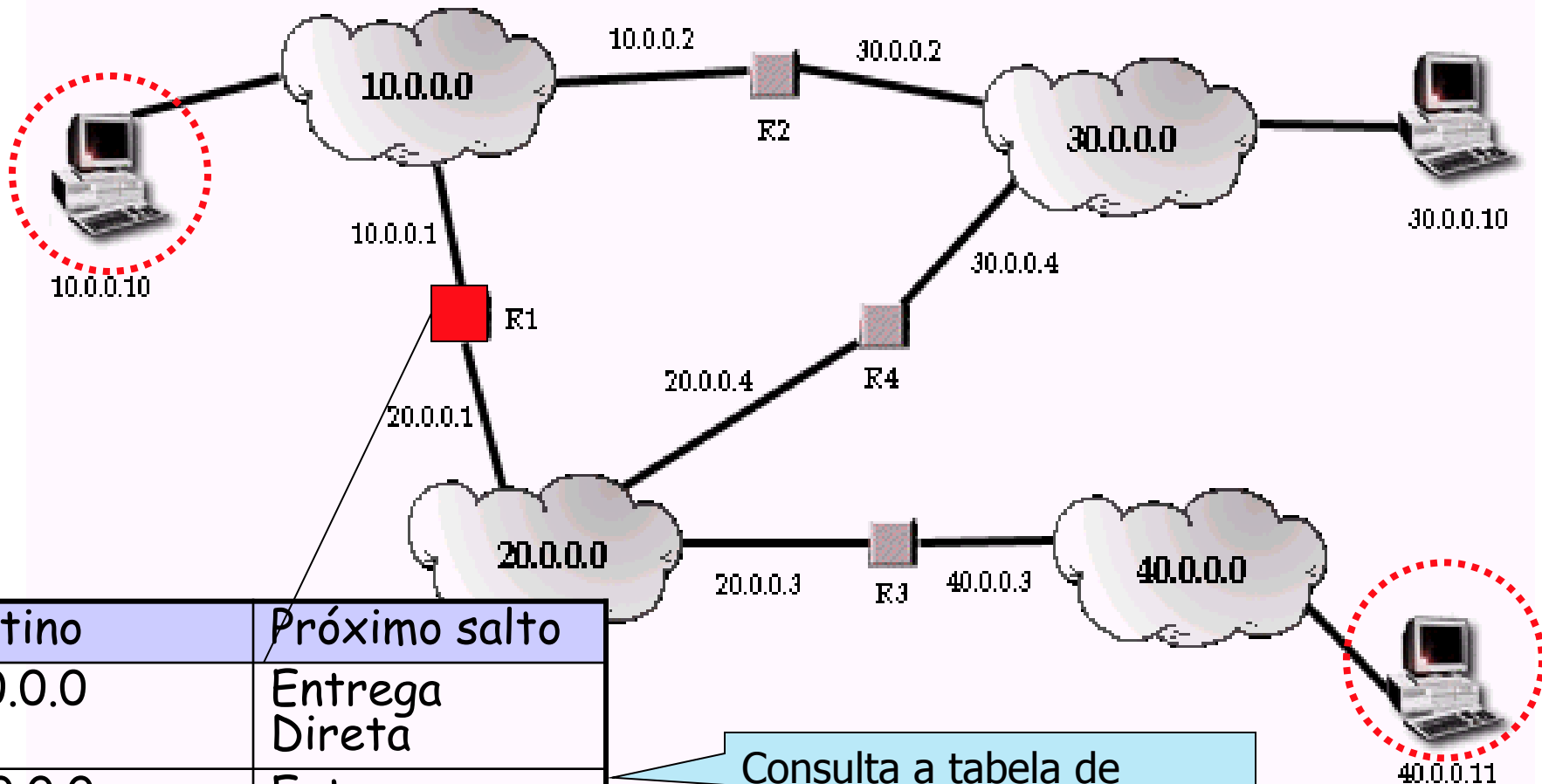
Destino	Próximo salto
10.0.0.0	Entrega Direta
Default	10.0.0.1

Encontra a rota mais apropriada e encaminha para a interface

Encaminhamento IP



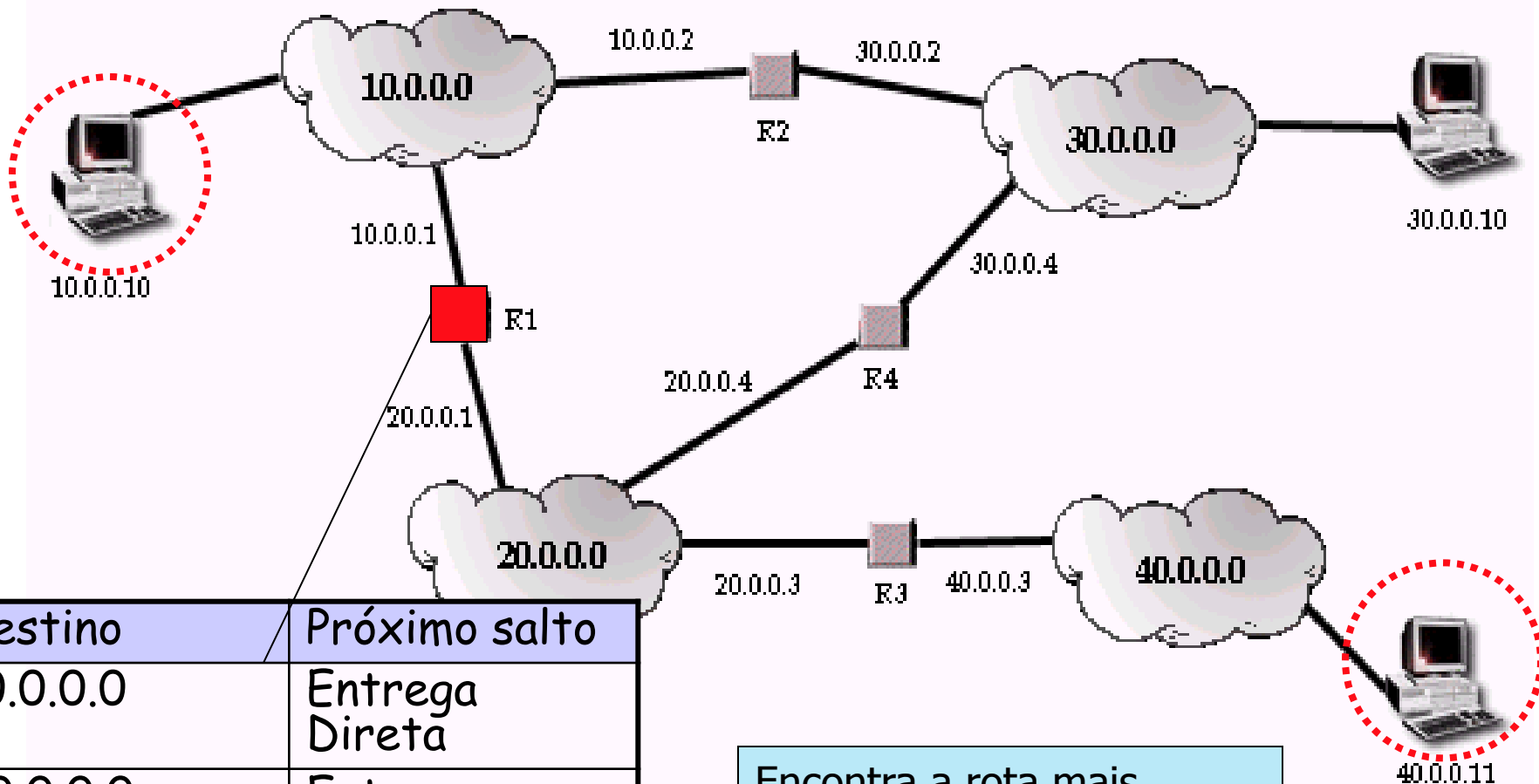
Encaminhamento IP



Destino	Próximo salto
10.0.0.0	Entrega Direta
20.0.0.0	Entrega Direta
30.0.0.0	10.0.0.2
40.0.0.0	20.0.0.3

Consulta a tabela de roteamento

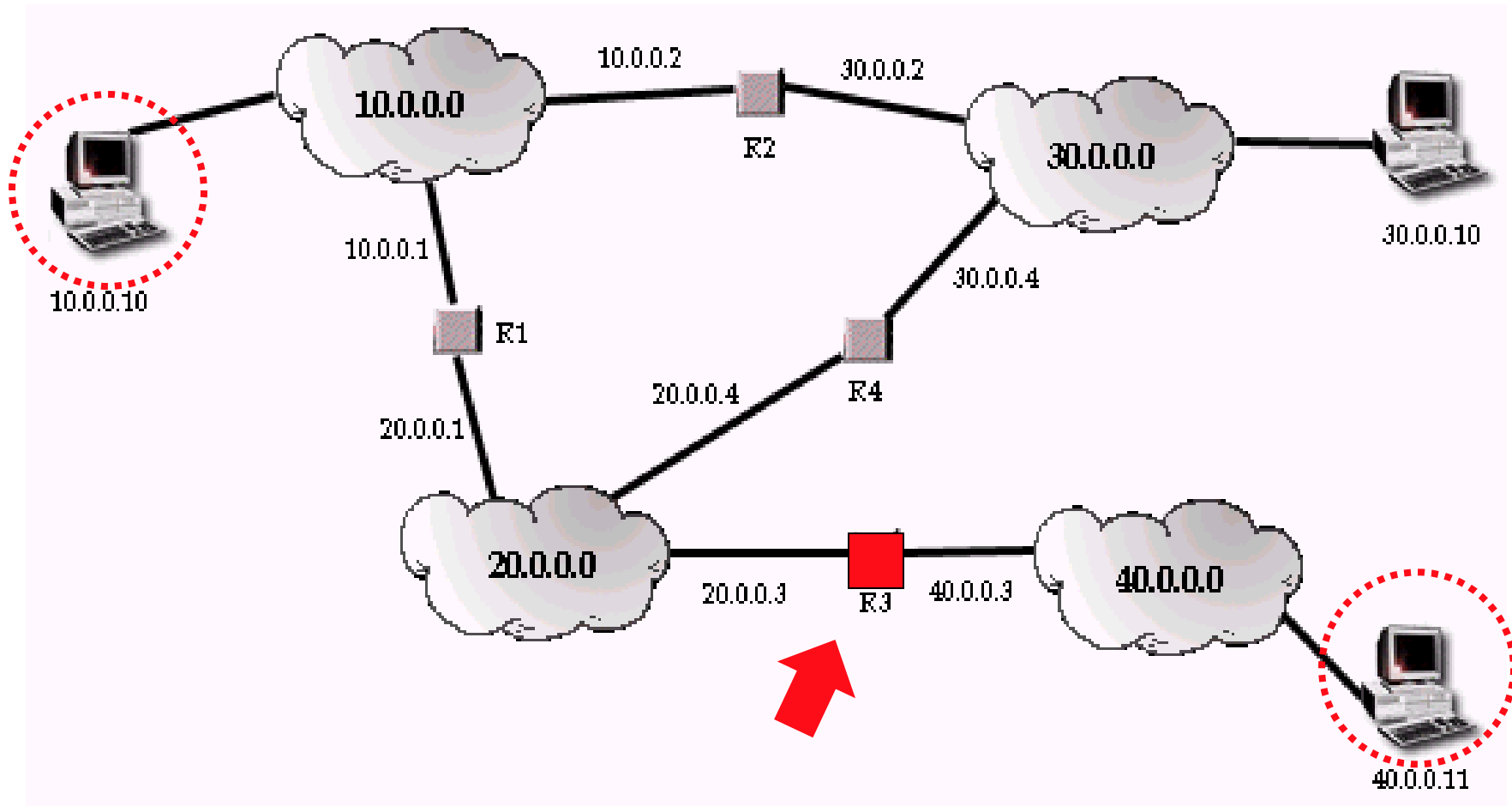
Encaminhamento IP



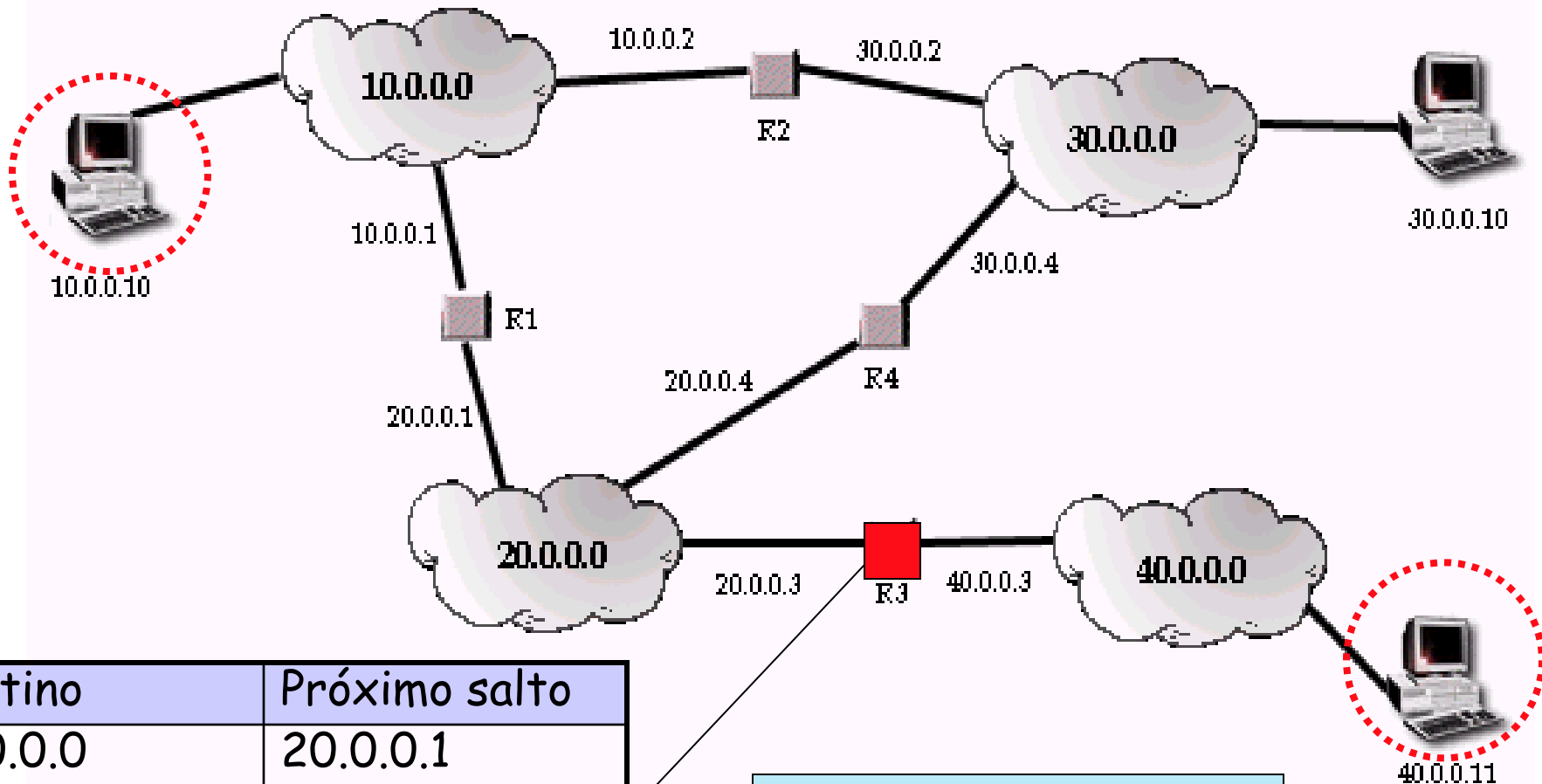
Destino	Próximo salto
10.0.0.0	Entrega Direta
20.0.0.0	Entrega Direta
30.0.0.0	10.0.0.2
40.0.0.0	20.0.0.3

Encontra a rota mais apropriada e encaminha para a interface

Encaminhamento IP



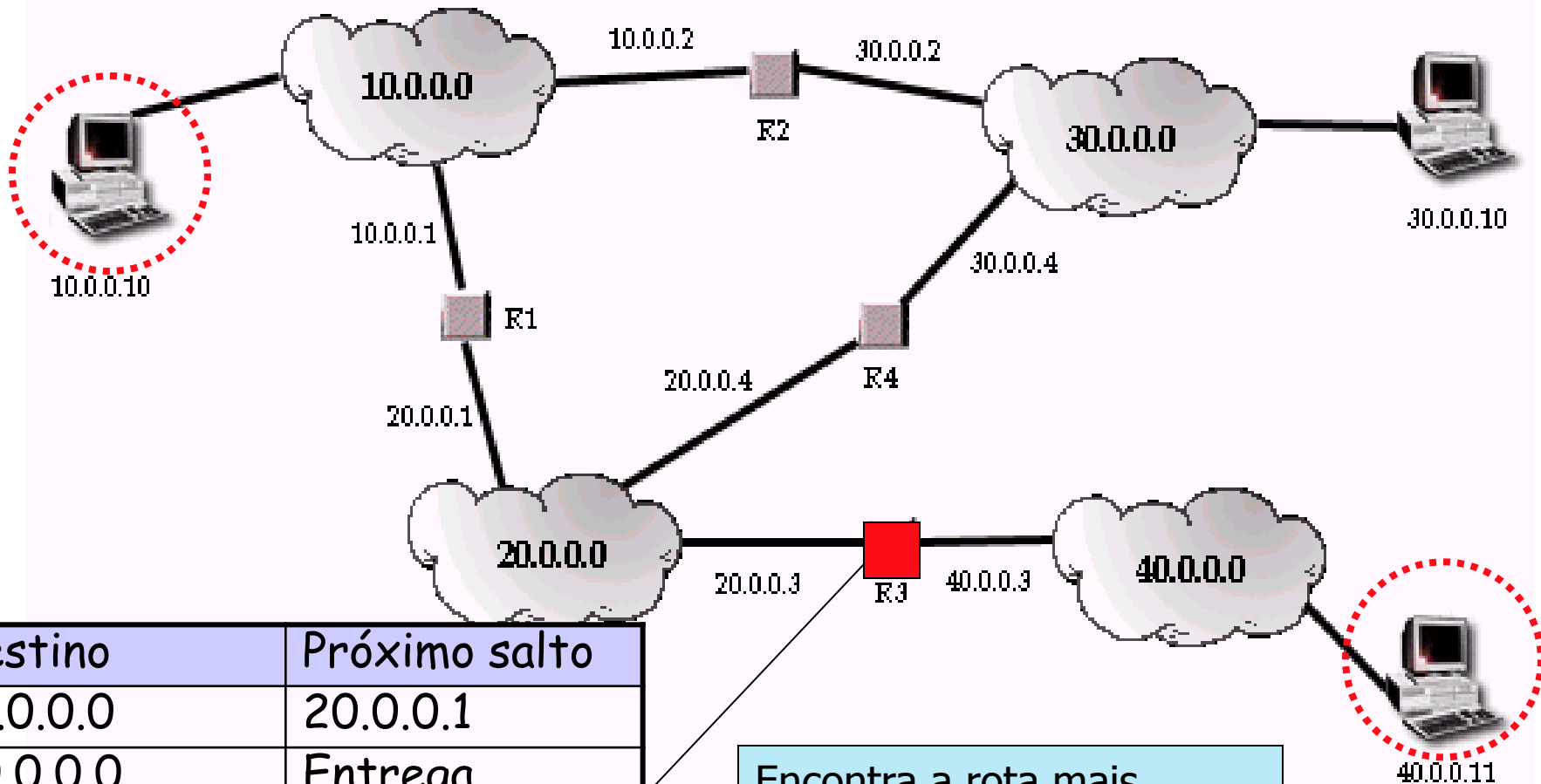
Encaminhamento IP



Destino	Próximo salto
10.0.0.0	20.0.0.1
20.0.0.0	Entrega Direta
30.0.0.0	20.0.0.4
40.0.0.0	Entrega Direta

Consulta a tabela de roteamento

Encaminhamento IP



Destino	Próximo salto
10.0.0.0	20.0.0.1
20.0.0.0	Entrega Direta
30.0.0.0	20.0.0.4
40.0.0.0	Entrega Direta

Encontra a rota mais apropriada e encaminha para a interface

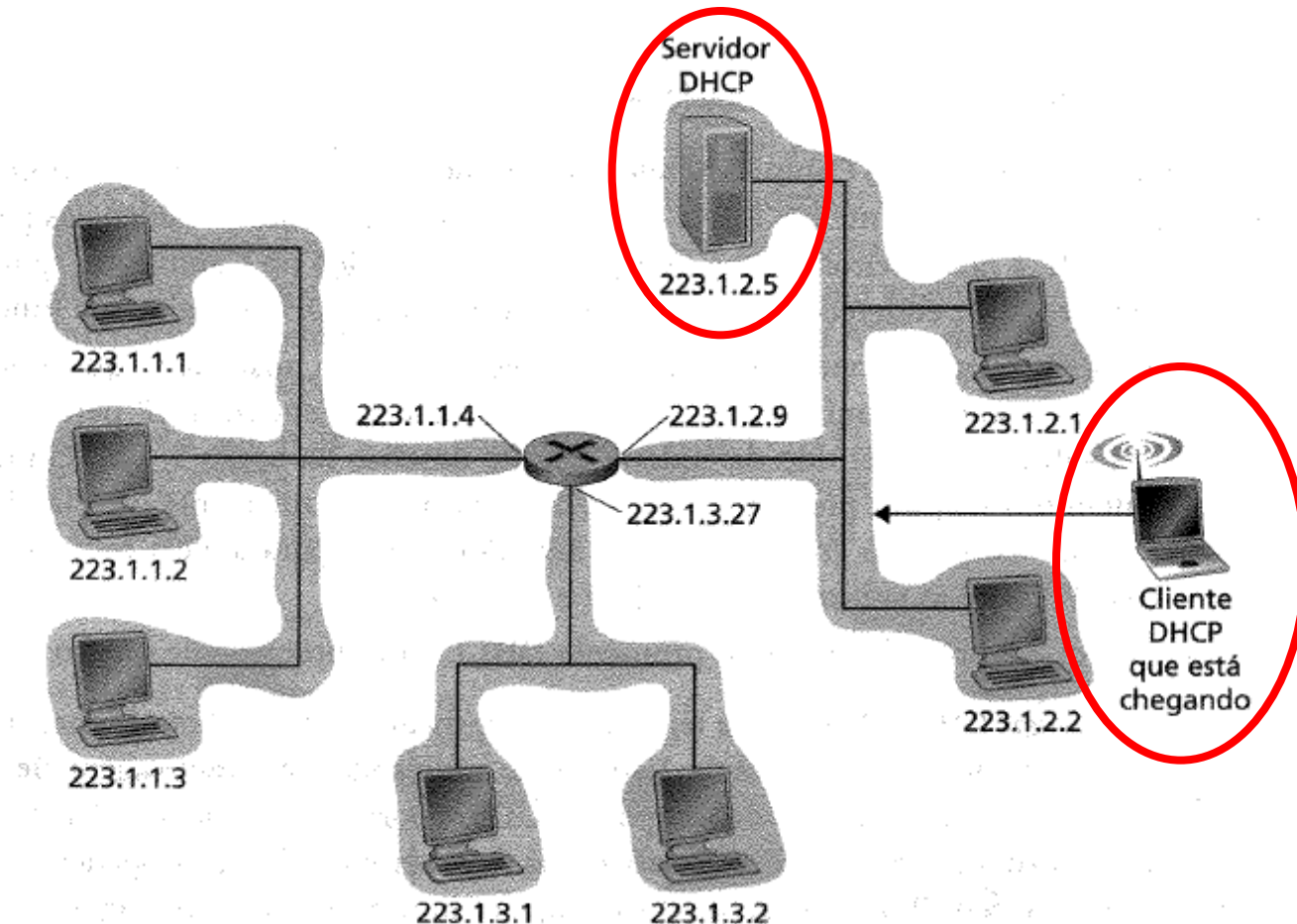
Como obter um endereço IP

P.: Como um hospedeiro obtém endereço IP ?

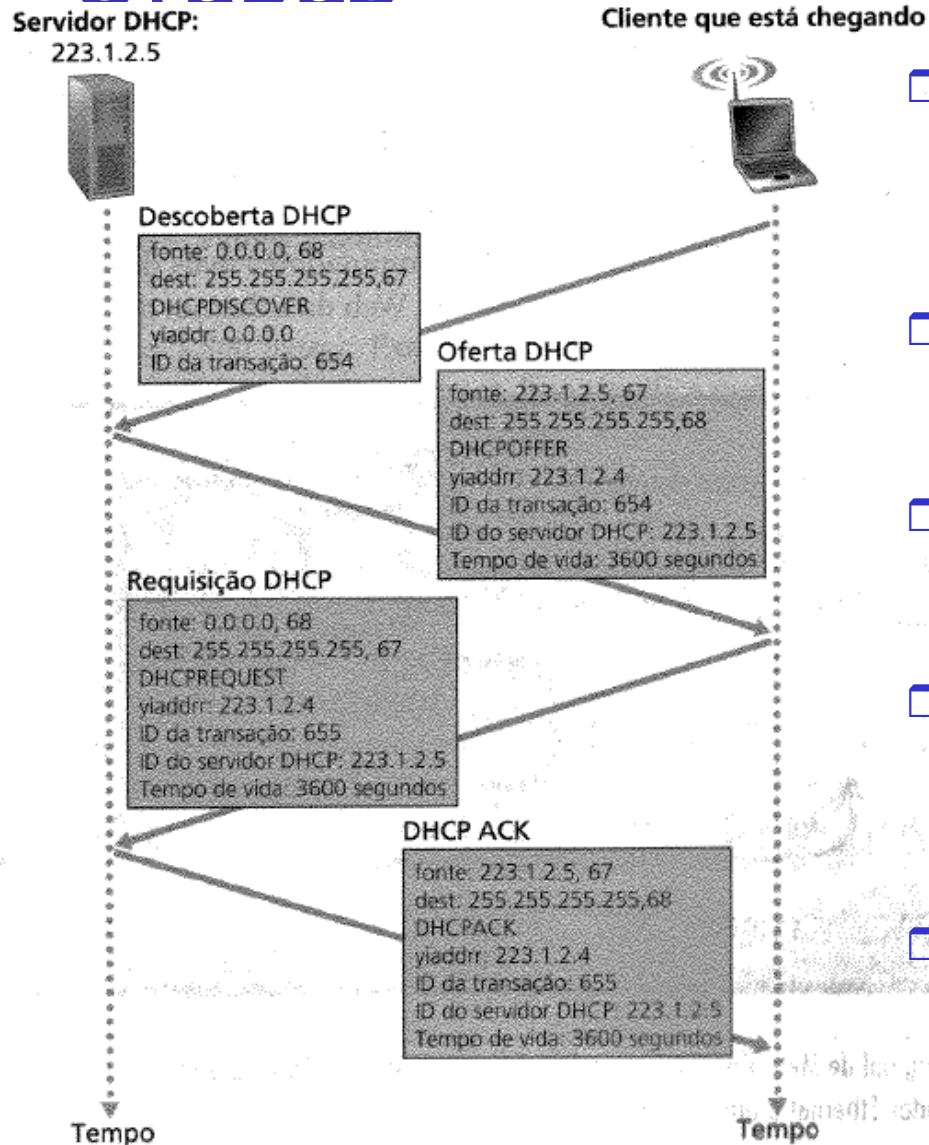
- Definido pelo administrador do sistema
 - Wintel: control-panel->network->configuration->tcp/ip->properties
 - UNIX: /etc/rc.config
- **DHCP**: dynamic host configuration protocol: obtém dinamicamente endereços IP de um servidor
 - “plug-and-play”
 - (mais no próximo capítulo)

Protocolo de Configuração Dinâmica de Hospedeiro

Veremos como o DHCP realmente funciona!



DHCP: Processo de quatro etapas



- ❑ Pacote de Descoberta DHCP é enviado em um quadro com endereço de *broadcast* MAC.
- ❑ O cliente poderá receber diversas ofertas DHCP e deverá escolher um deles.
- ❑ Uma requisição DHCP será enviada ao servidor escolhido.
- ❑ Há também um mecanismo para que o cliente possa renovar o aluguel de um endereço IP.
- ❑ **Alternativa: IP Móvel (Capítulo 6)**

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

Como obter um endereço IP

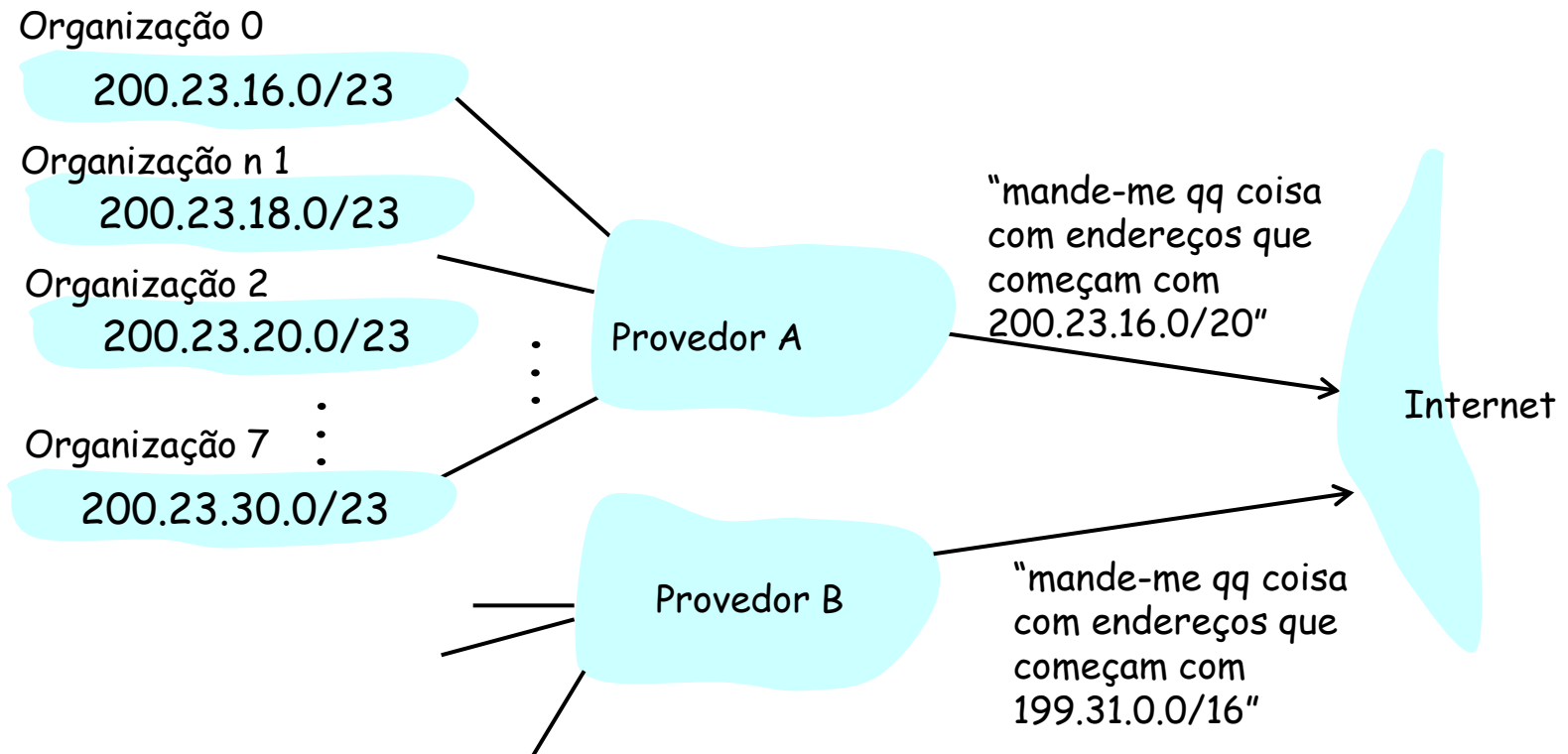
P.: Como uma **rede** obtém a parte de sub-rede do endereço IP ?

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

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

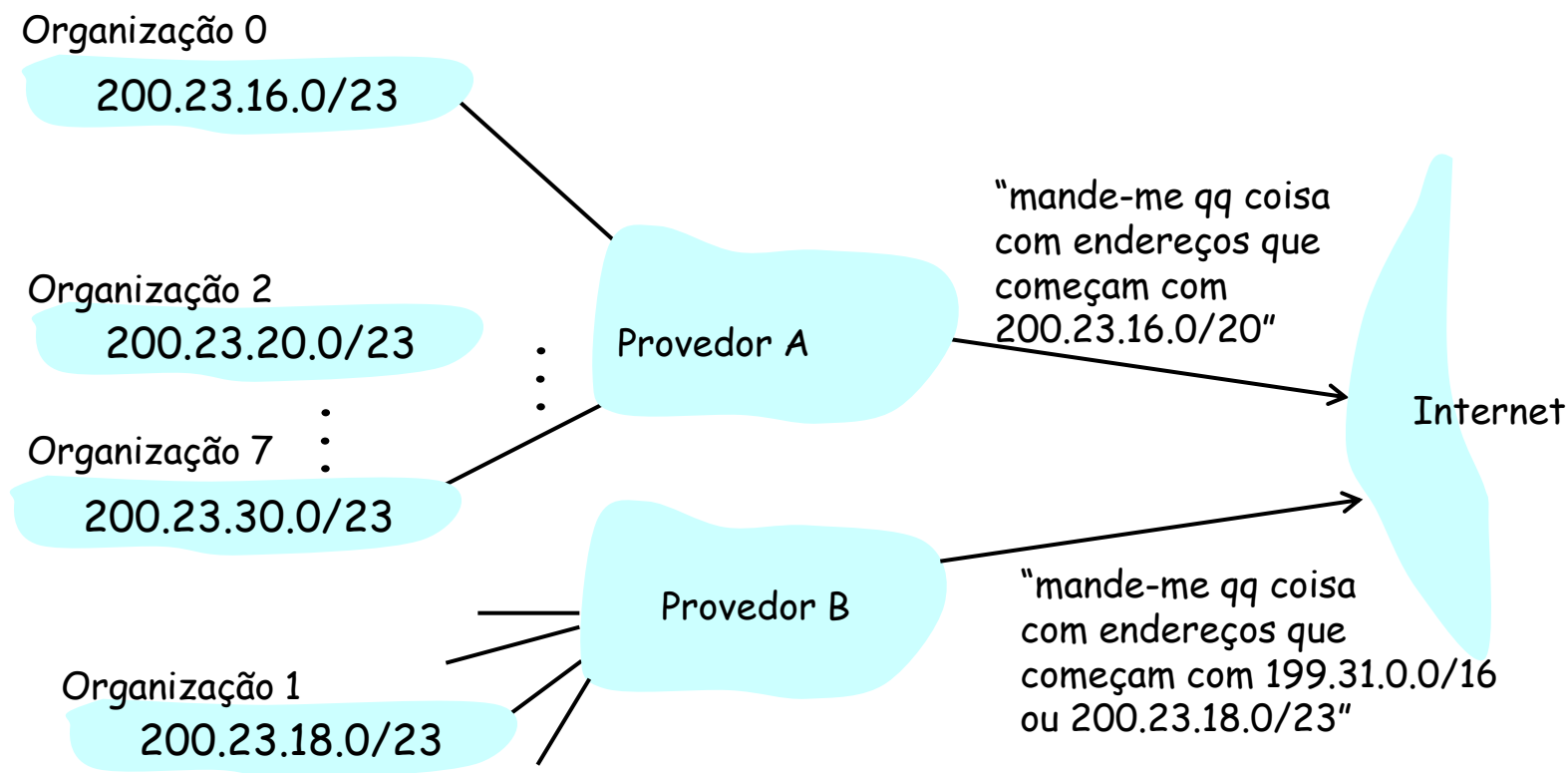
Endereçamento hierárquico: agregação de rotas

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 **C**orporation for **A**ssigned **N**ames and **N**umbers

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

<http://www.internic.net/>

DNS: Domain Name System

Pessoas: muitos identificadores:

- CPF, nome, no. de Passaporte

hospedeiros, roteadores Internet :

- endereço IP (32 bit) - usado p/ endereçar datagramas.
- "nome", e.g., marajo.ufpa.br - usado por gente.

P: como mapear entre nome e endereço IP?

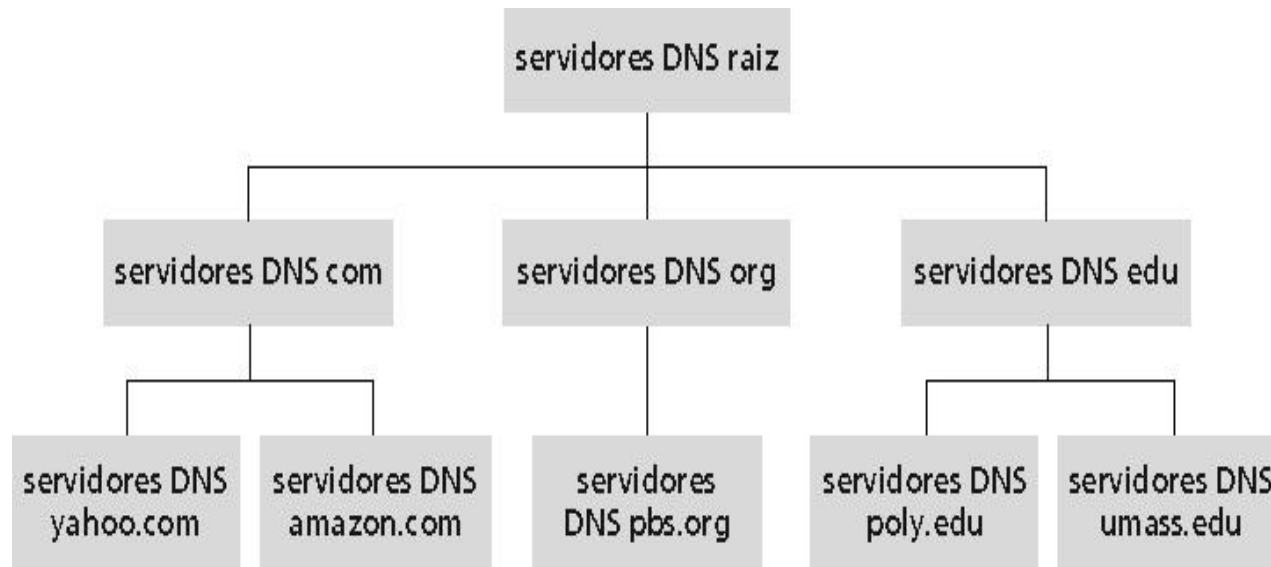
Domain Name System:

- *base de dados distribuída* implementada através de uma hierarquia *servidores de nomes*.
- *protocolo de camada de aplicação* permite que hospedeiros, roteadores e servidores de nomes se comuniquem para *resolver* nomes (tradução endereço/nome)
 - note: função imprescindível da Internet implementada como protocolo de camada de aplicação
 - complexidade na borda da rede

Nomes DNS

- ❑ Um nome de domínio é uma concatenação de nomes:
 - nome-n.nome-2.nome-1
- ❑ Conceitualmente, o nível mais alto (nome-1) permite duas formas diferentes de nomeação:
 - Organizacional
 - com, edu, gov, int, mil, net e org
 - Geográfica
 - Código dos países: xx (código ISO de 2 letras - exceto uk)
- ❑ Exemplos:
 - inf.puc-rio.br jb.com.br microsoft.com purdue.edu

Base de dados distribuída, hierárquica



Cliente quer o IP para www.amazon.com; 1ª aprox.:

- Cliente consulta um servidor de raiz para encontrar o servidor DNS com
- Cliente consulta o servidor DNS com para obter o servidor DNS amazon.com
- Cliente consulta o servidor DNS amazon.com para obter o endereço IP para www.amazon.com

Servidores de nomes DNS

- Nenhum servidor mantém todos os mapeamento nome-para-endereço IP

servidor de nomes local:

- cada provedor, empresa tem *servidor de nomes local (default)*
- pedido DNS de hospedeiro vai primeiro ao servidor de nomes local

servidor de nomes com autoridade:

- p/ hospedeiro: guarda nome, endereço IP dele
- pode realizar tradução nome/endereço para este nome

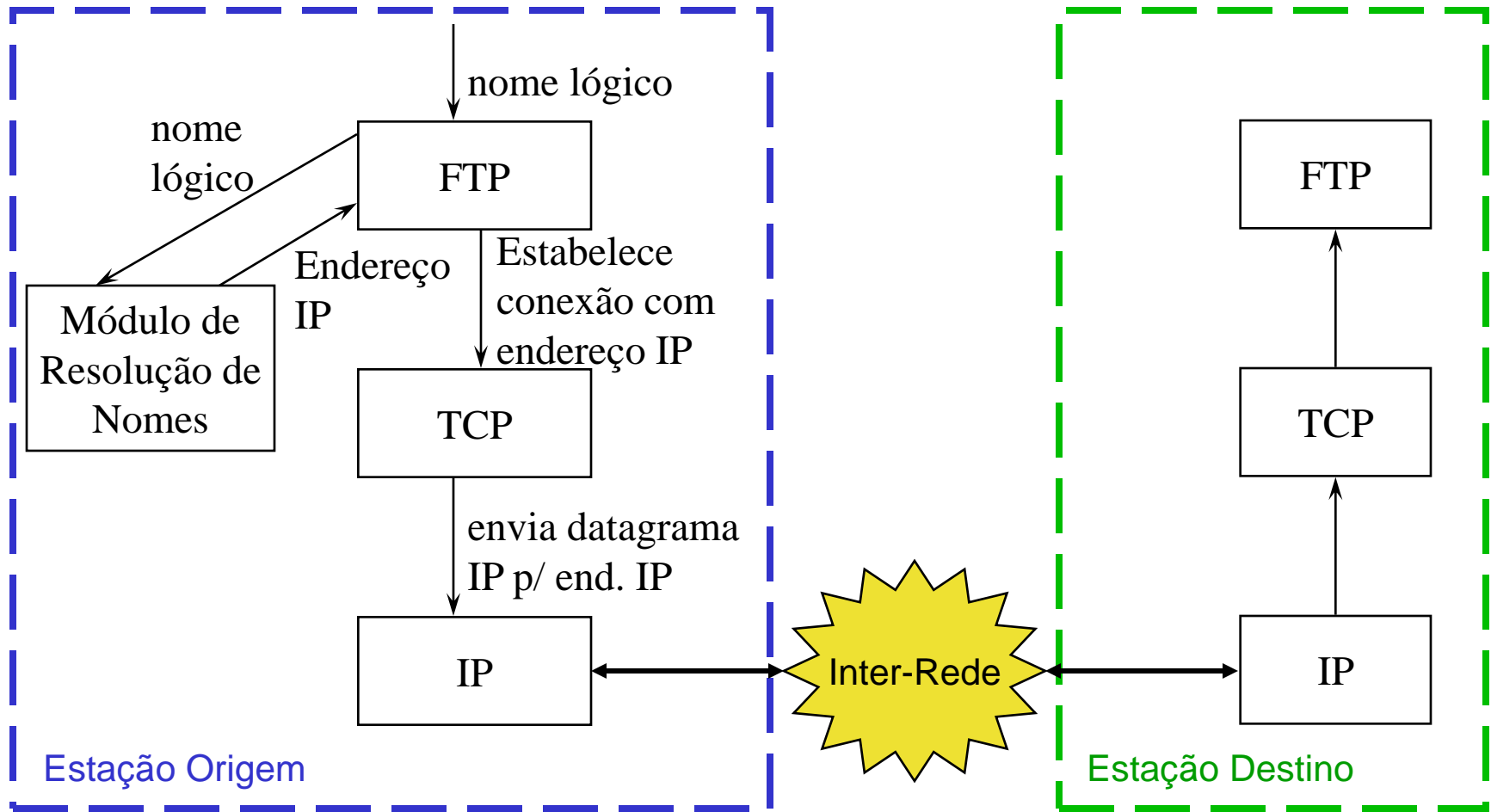
DNS: servidores de nomes raiz

- São contatados pelos servidores de nomes locais que não podem resolver um nome
- Servidores de nomes raiz:
 - Buscam servidores de nomes autorizados se o mapeamento do nome não for conhecido
 - Conseguem o mapeamento
 - Retornam o mapeamento para o servidor de nomes local



Existem 13 servidores de nomes raiz no mundo (Fev. 2004)

Resolução de Nomes

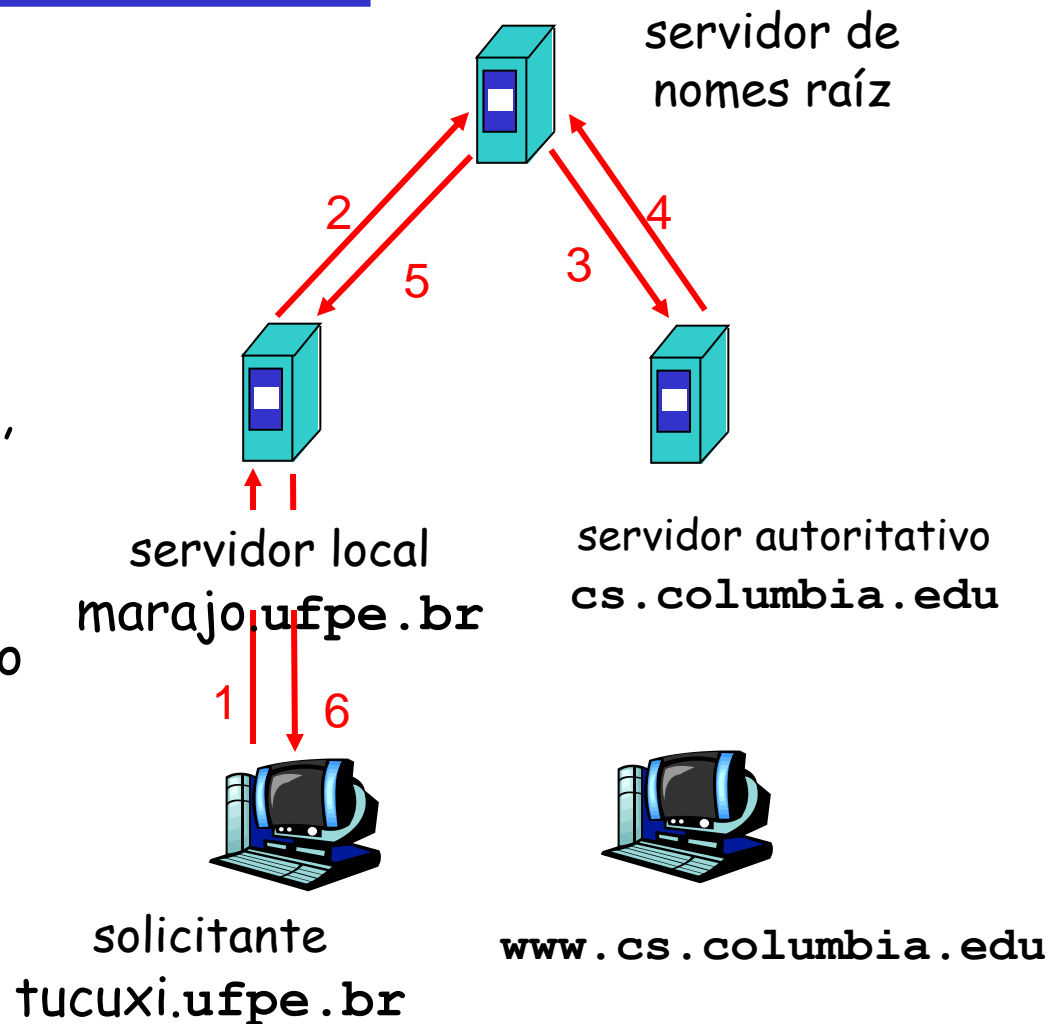


- ❑ **Módulo Resolução de Nomes:** consulta arquivos locais ou um serviço de resolução de nomes

Exemplo simples do DNS

hospedeiro `tucuxi.ufpe.br`
requer endereço IP de
`www.cs.columbia.edu`

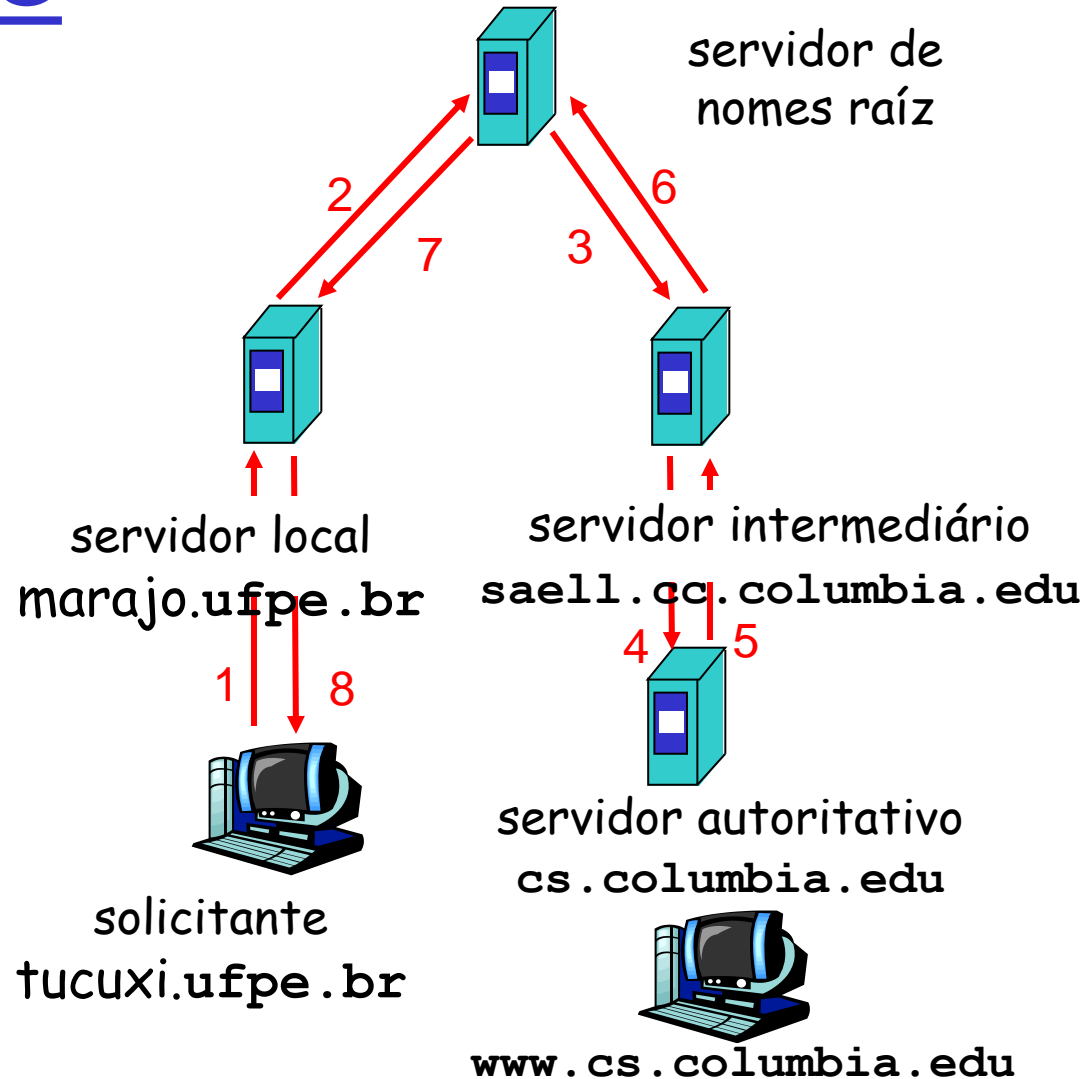
1. Contata servidor DNS local,
`marajo.ufpe.br`
2. `marajo.ufpe.br` contata
servidor raíz, se necessário
3. Servidor raíz contata
servidor autoritativo
`cs.columbia.edu`, se
necessário



Exemplo de DNS

Servidor raiz:

- pode não conhecer o servidor de nomes autoritativo
- pode conhecer *servidor de nomes intermediário*: a quem contacta para descobrir o servidor de nomes autoritativo



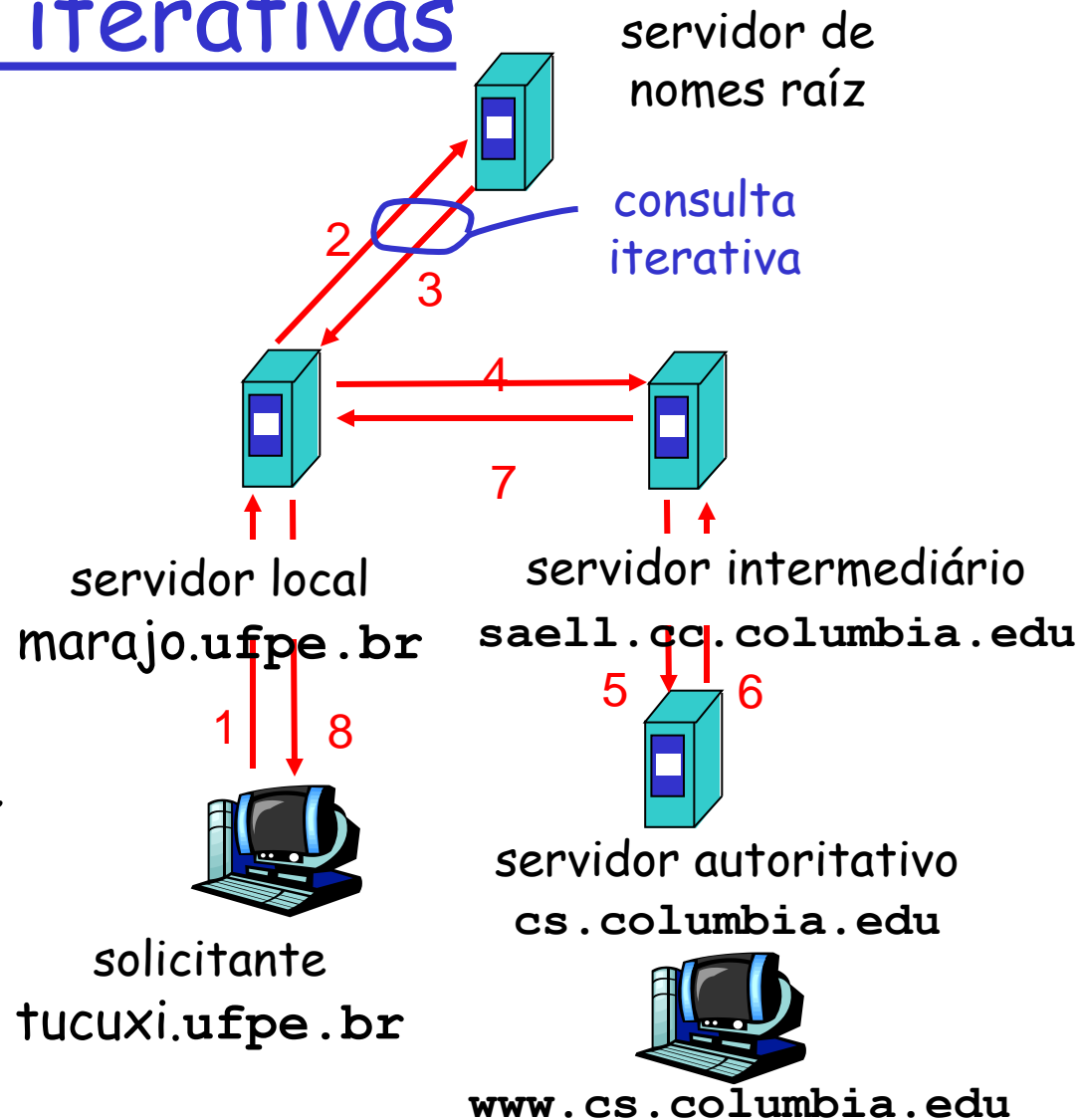
DNS: consultas iterativas

consulta recursiva:

- ❑ transfere a responsabilidade de resolução do nome para o servidor de nomes contatado

consulta iterativa:

- ❑ servidor consultado responde com o nome de um servidor de contato
- ❑ "Não conheço este nome, mas pergunte para esse servidor"



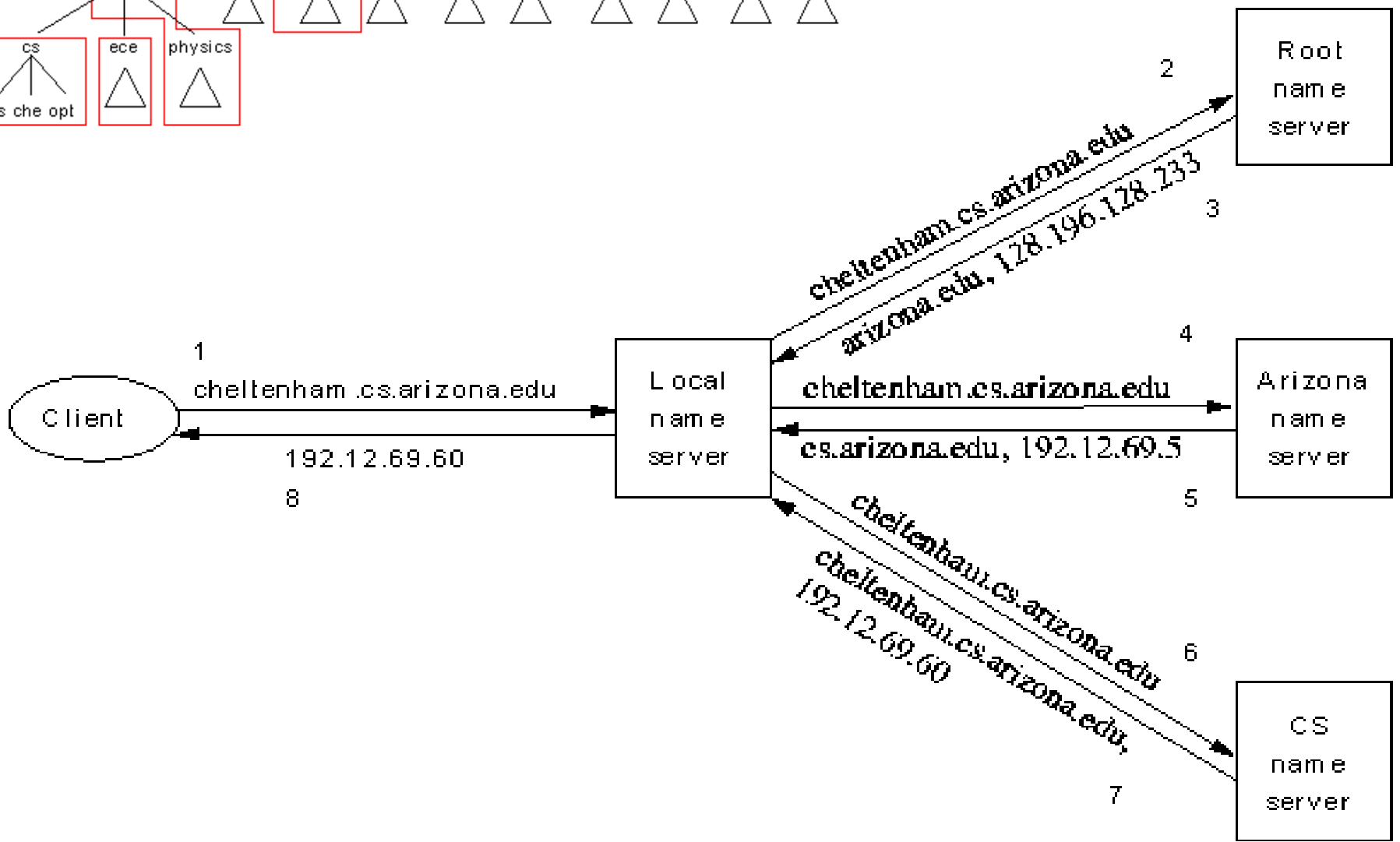
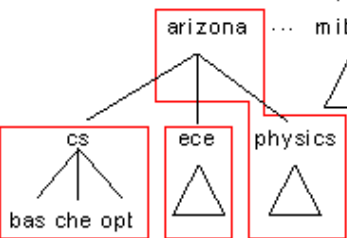
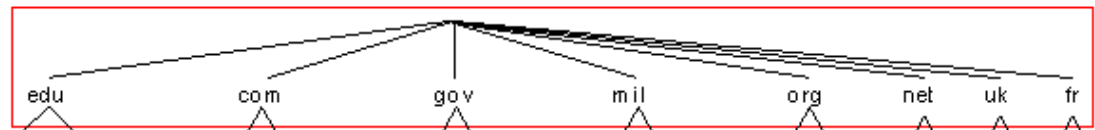
Registros do DNS

DNS: base de dados distribuída que armazena registros de recursos (RR)

formato dos RR: (name, value, type,ttl)

- Type = A
 - **name** é o nome do computador
 - **value** é o endereço IP
- Type = NS
 - **name** é um domínio (ex.: foo.com)
 - **value** é o endereço IP do servidor de nomes autorizados para este domínio
- Type = CNAME
 - **name** é um "apelido" para algum nome "canônico" (o nome real)
www.ibm.com é realmente
servereast.backup2.ibm.com
 - **value** é o nome canônico
- Type = MX
 - **value** é o nome do servidor de correio associado com **name**

Resolução de nome



Atualização de Dados

- Exemplo: empresa recém-criada “Network Utopia”
- Registrar o nome networkutopia.com numa **entidade registradora**” (ex.: Network Solutions)
 - É necessário fornecer a **entidade registradora** os nomes e endereços IP do seu servidor nomes autorizados (primário e secundário)
 - **entidade registradora** insere dois RRs (Registros de Recursos) no servidor TLD (Top Level Domain) do domínio **com**:

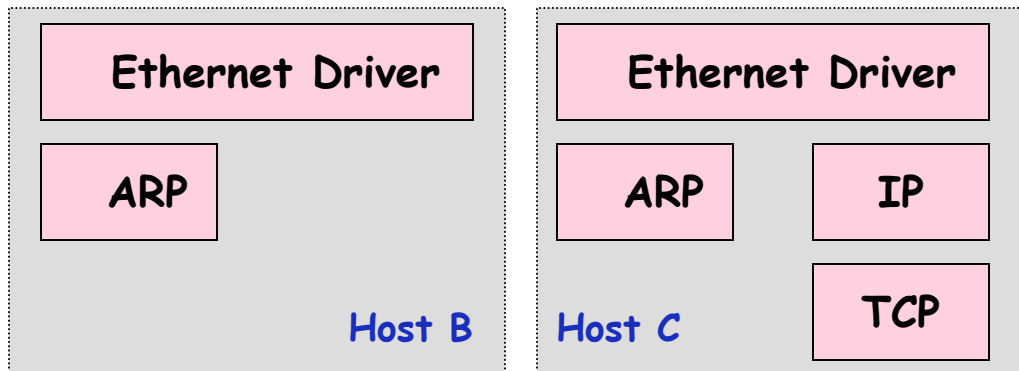
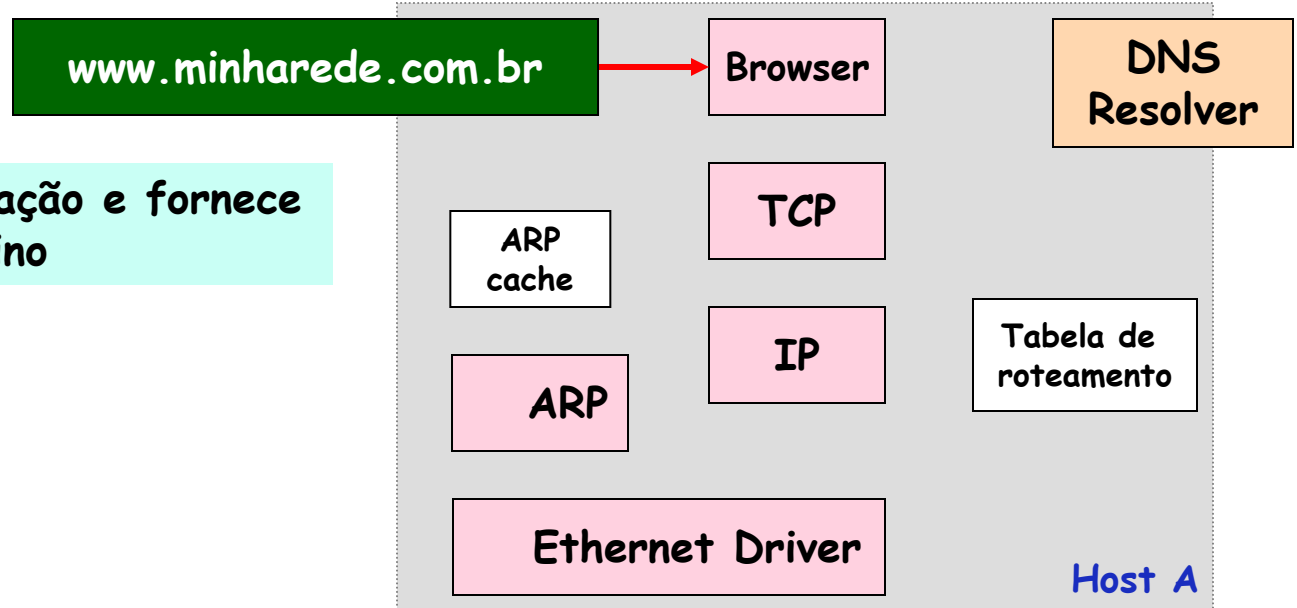
```
(networkutopia.com, dns1.networkutopia.com, NS)
(dns1.networkutopia.com, 212.212.212.1, A)
```

- No servidor autorizado, inserir um registro Tipo A para www.networkutopia.com e um registro Tipo MX para networkutopia.com
- **Como as pessoas obtêm o endereço IP do seu Web site?**

Inserindo registros no DNS

Mecanismo de resolução ARP

1 Usuário ativa a aplicação e fornece o nome do host destino

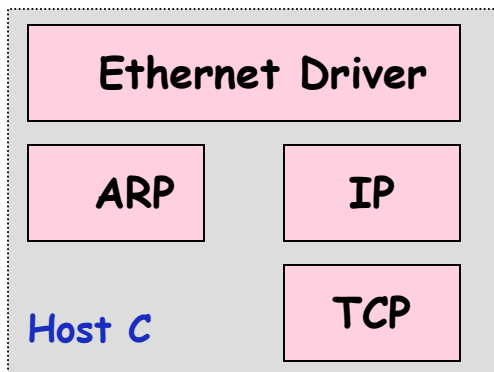
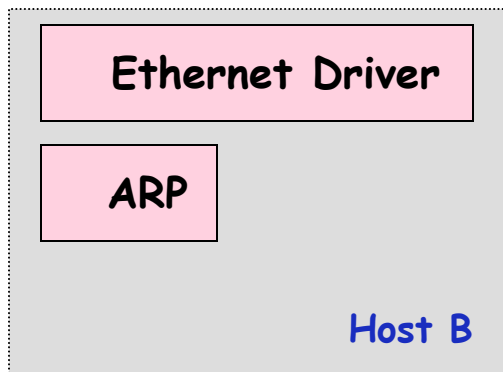
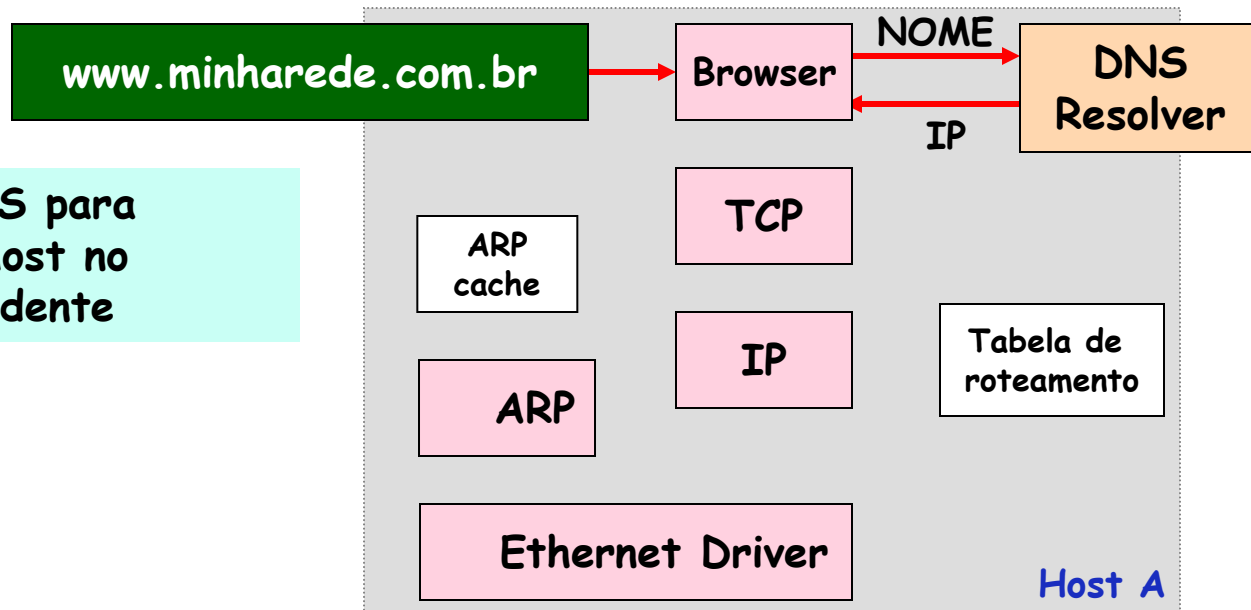


Rede local Ethernet

Mecanismo de resolução ARP

2

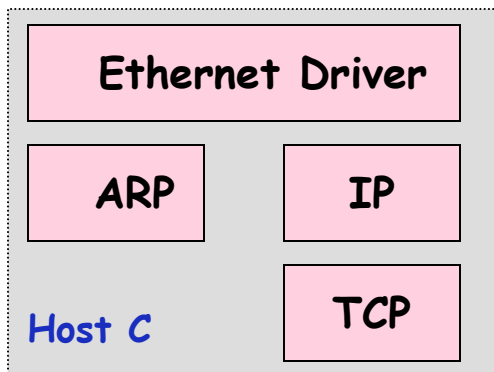
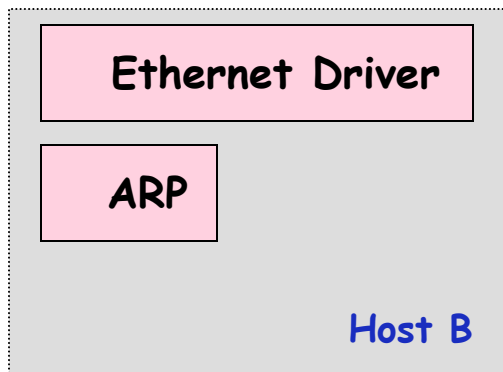
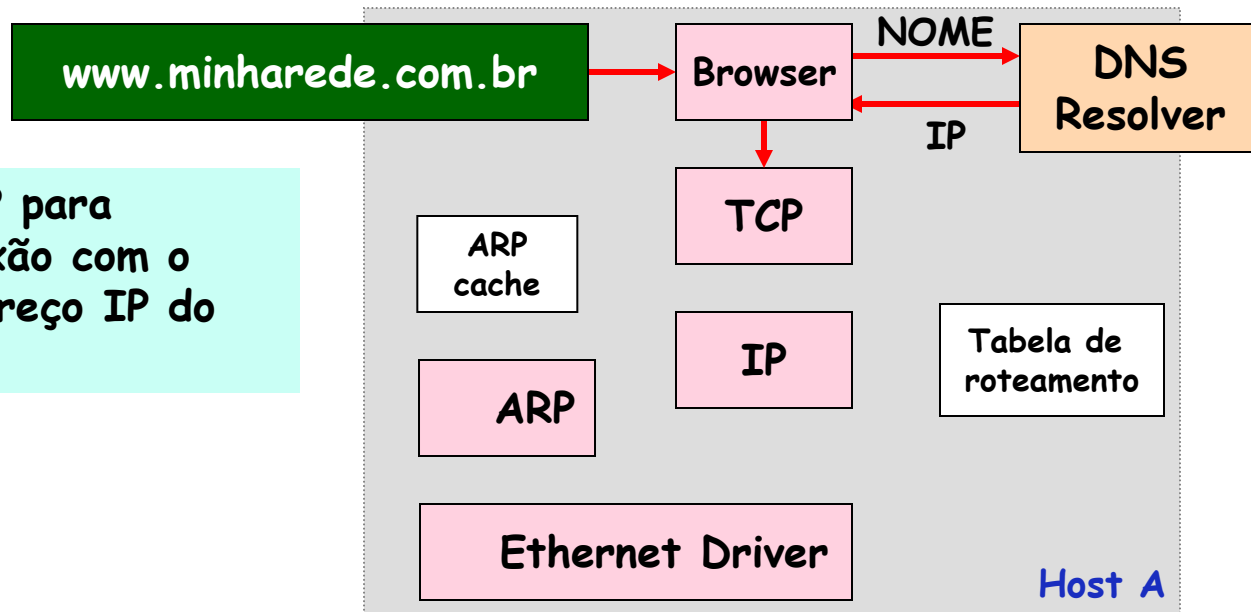
Aplicação chama o DNS para converter o nome do host no endereço IP correspondente



Rede local Ethernet

Mecanismo de resolução ARP

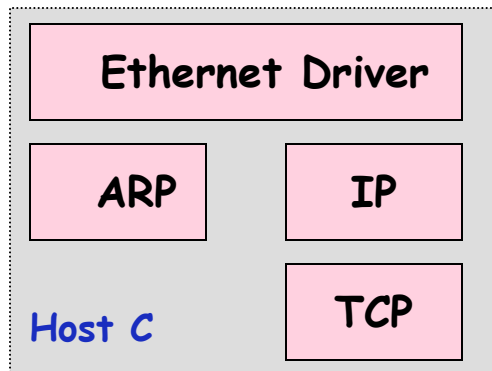
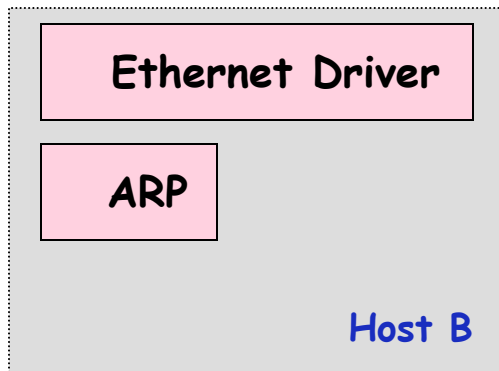
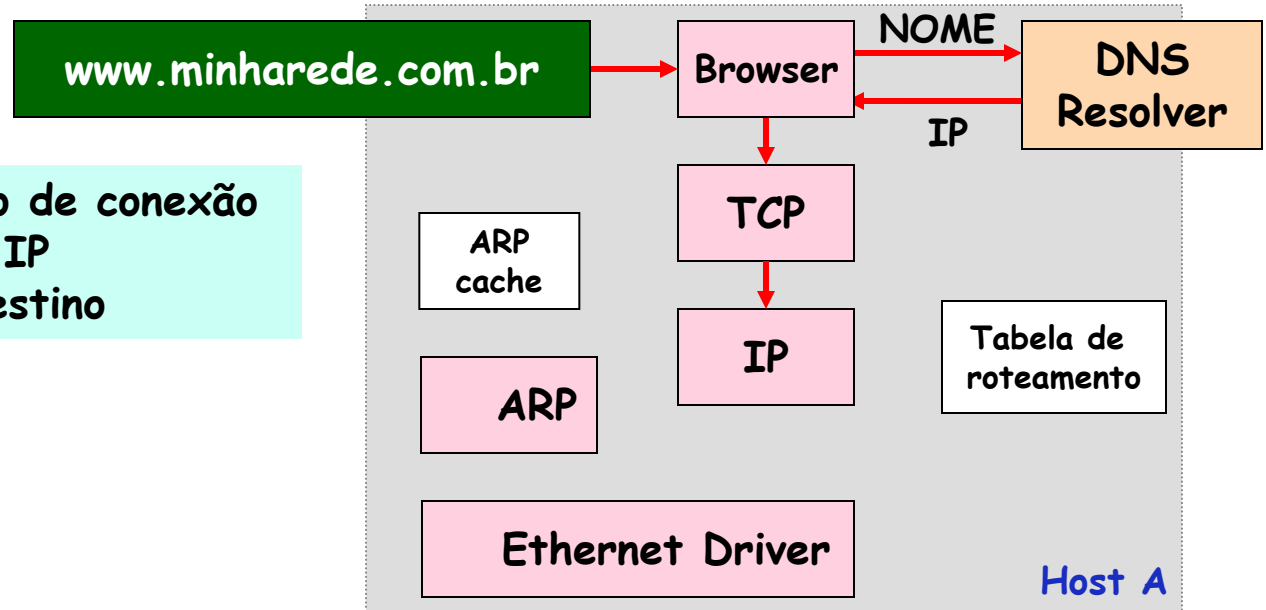
- 3 Aplicação pede ao TCP para estabelecer uma conexão com o host e fornece o endereço IP do destino



Rede local Ethernet

Mecanismo de resolução ARP

- 4 TCP envia a solicitação de conexão através de um pacote IP endereçado ao host destino

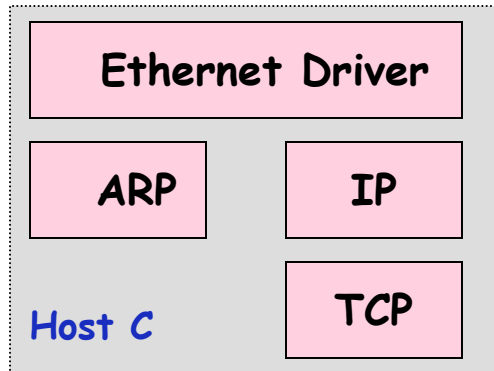
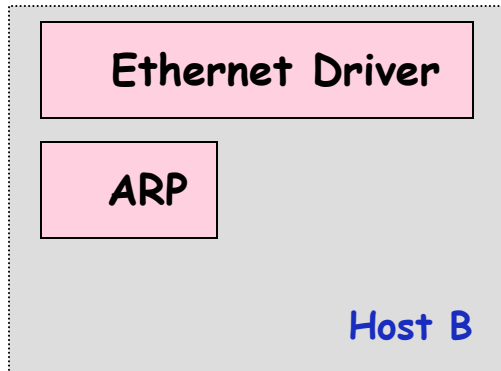
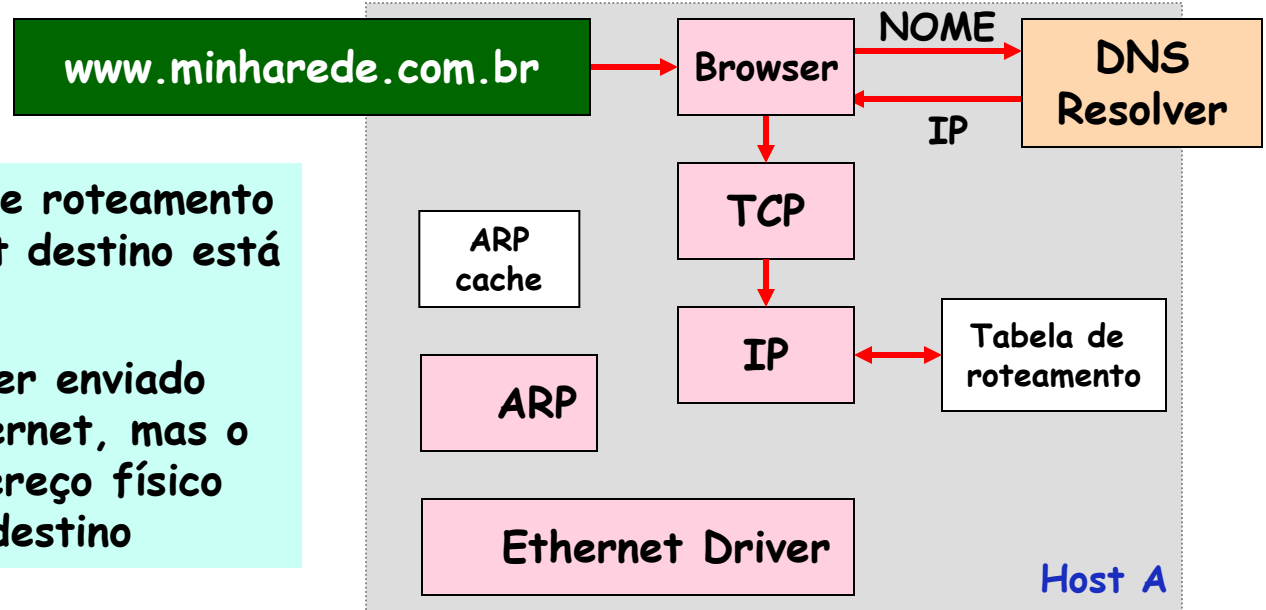


Rede local Ethernet

Mecanismo de resolução ARP

5 IP consulta a tabela de roteamento e identifica que o host destino está na rede local.

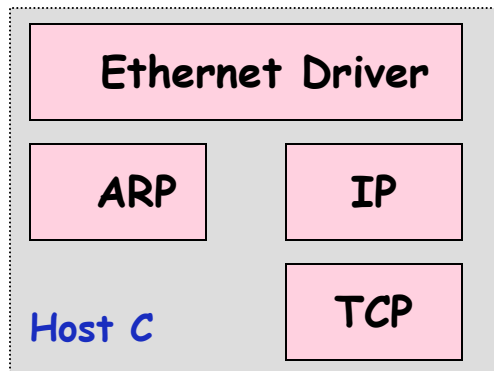
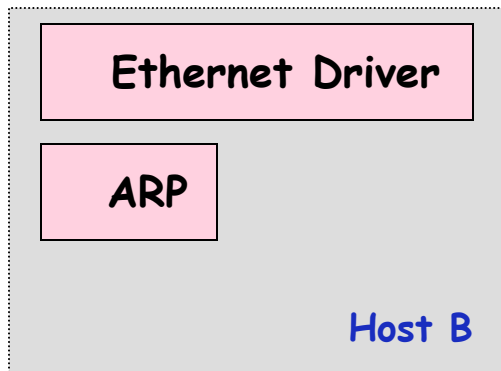
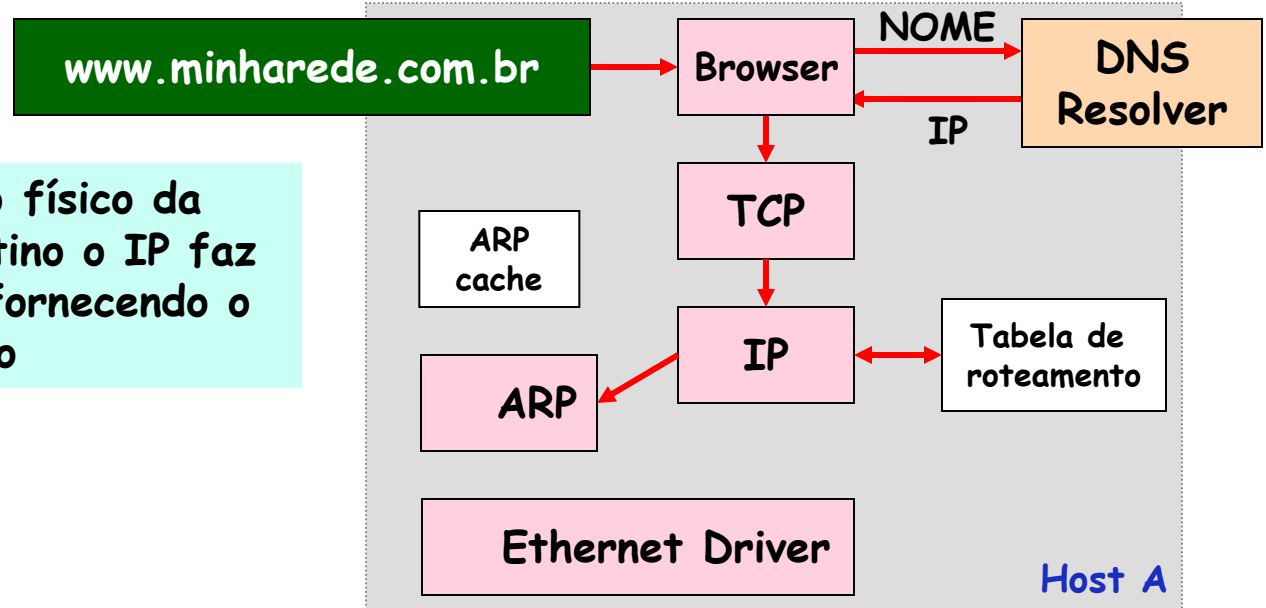
Logo, o pacote deve ser enviado para a rede local Ethernet, mas o IP não conhece o endereço físico da interface do host destino



Rede local Ethernet

Mecanismo de resolução ARP

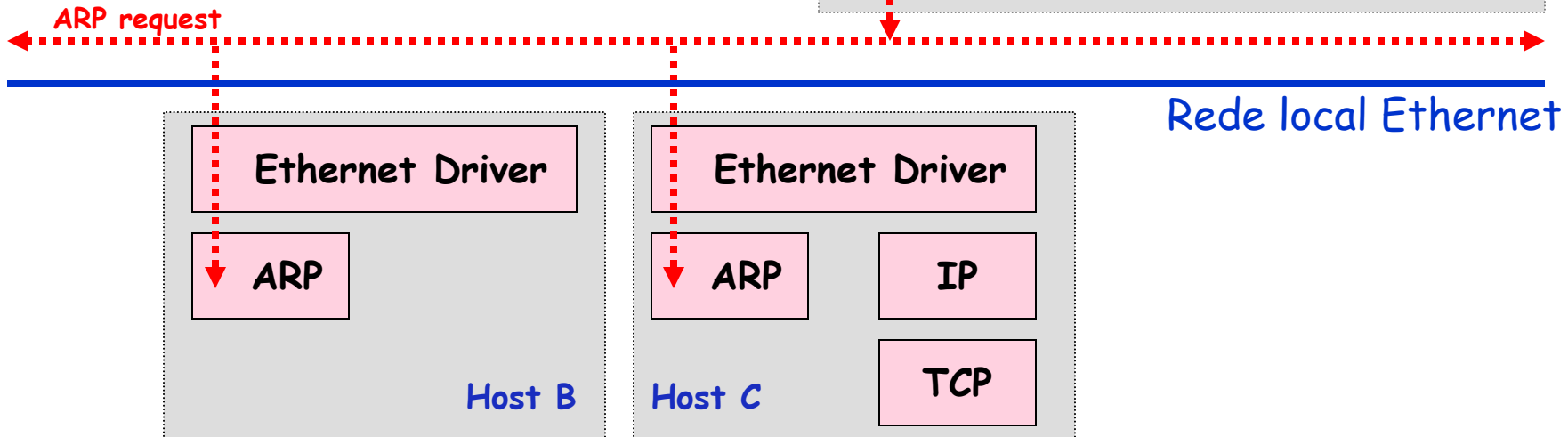
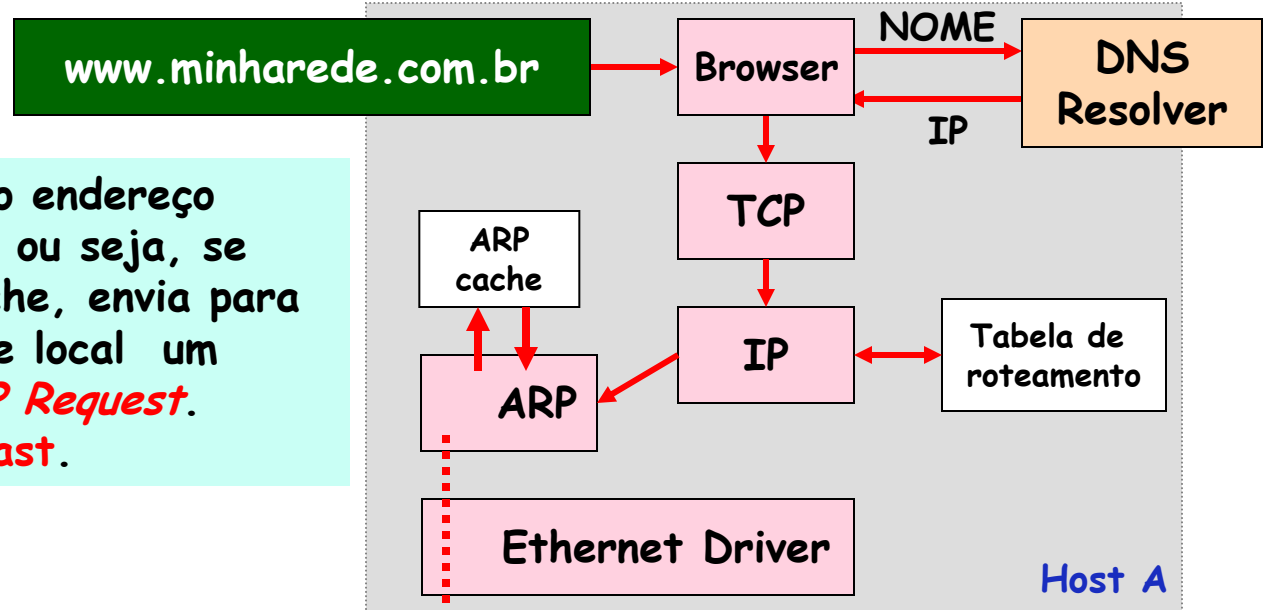
- 6 Para saber o endereço físico da interface do host destino o IP faz uma consulta ao ARP fornecendo o endereço IP do destino



Rede local Ethernet

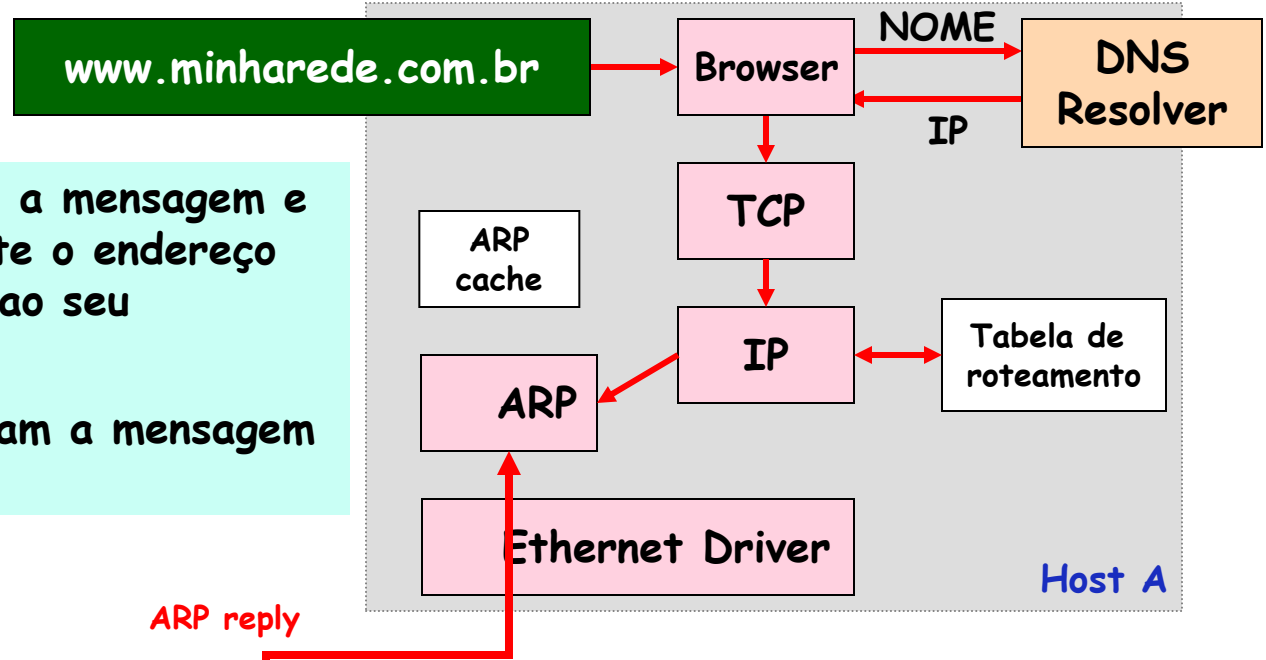
Mecanismo de resolução ARP

7 Se o ARP não souber o endereço físico correspondente, ou seja, se não estiver na sua cache, envia para todos os hosts da rede local um "quadro" chamado *ARP Request*. Isso se chama **broadcast**.

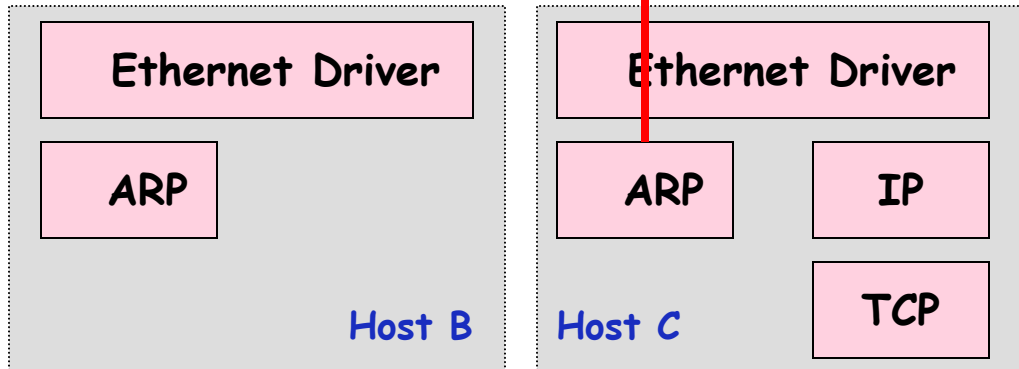


Mecanismo de resolução ARP

- 8 O host destino recebe a mensagem e indica para o solicitante o endereço físico correspondente ao seu endereço IP.
Todos os demais ignoram a mensagem



ARP reply



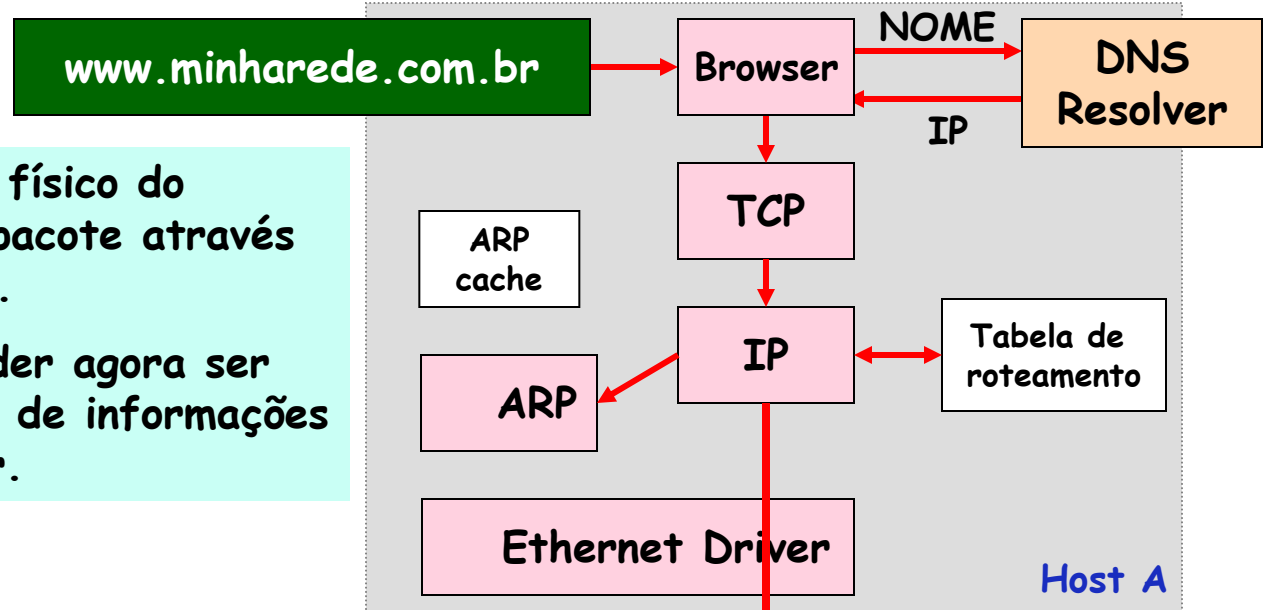
Rede local Ethernet

Mecanismo de resolução ARP

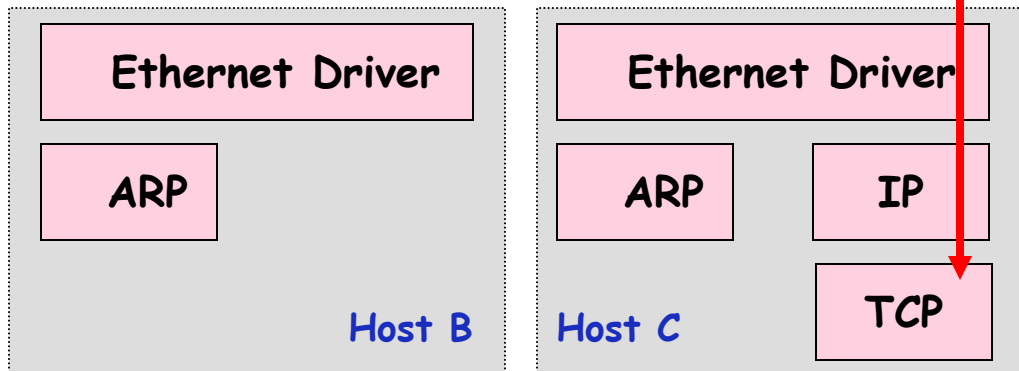
9

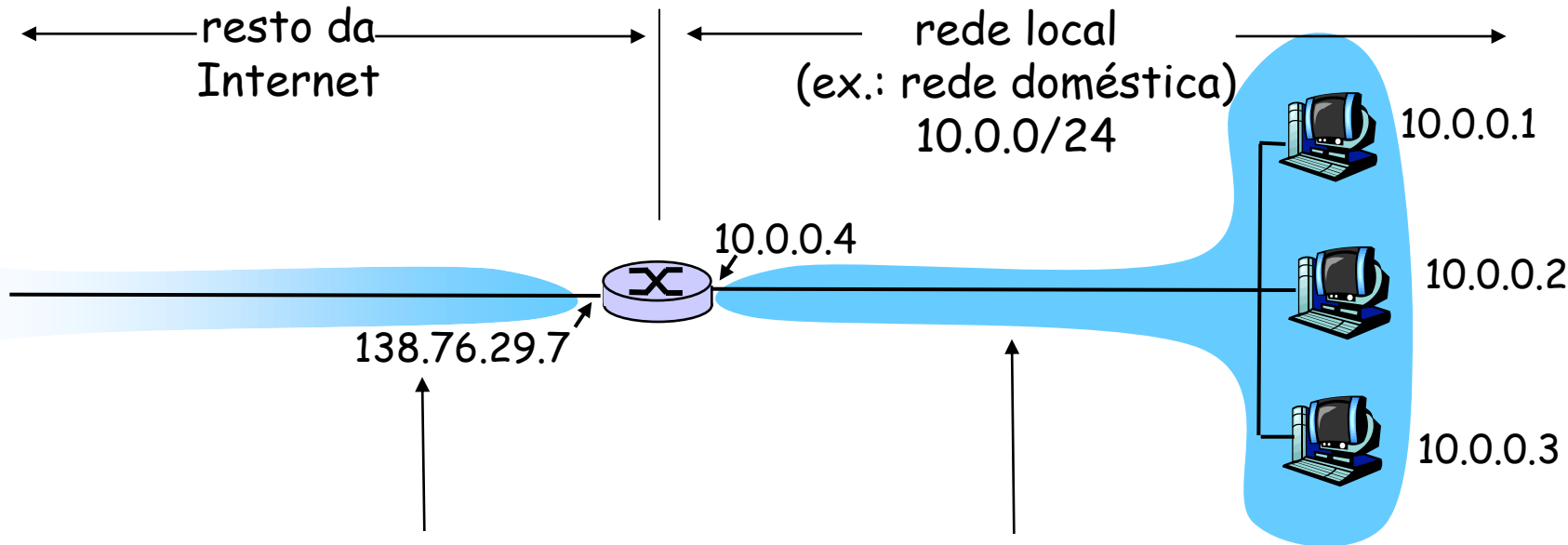
De posse do endereço físico do destino, o IP envia o pacote através da interface Ethernet.

A conexão TCP vai poder agora ser estabelecida e o fluxo de informações da aplicação vai iniciar.



Rede local Ethernet





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

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

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

Implementação: o roteador NAT deve:

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

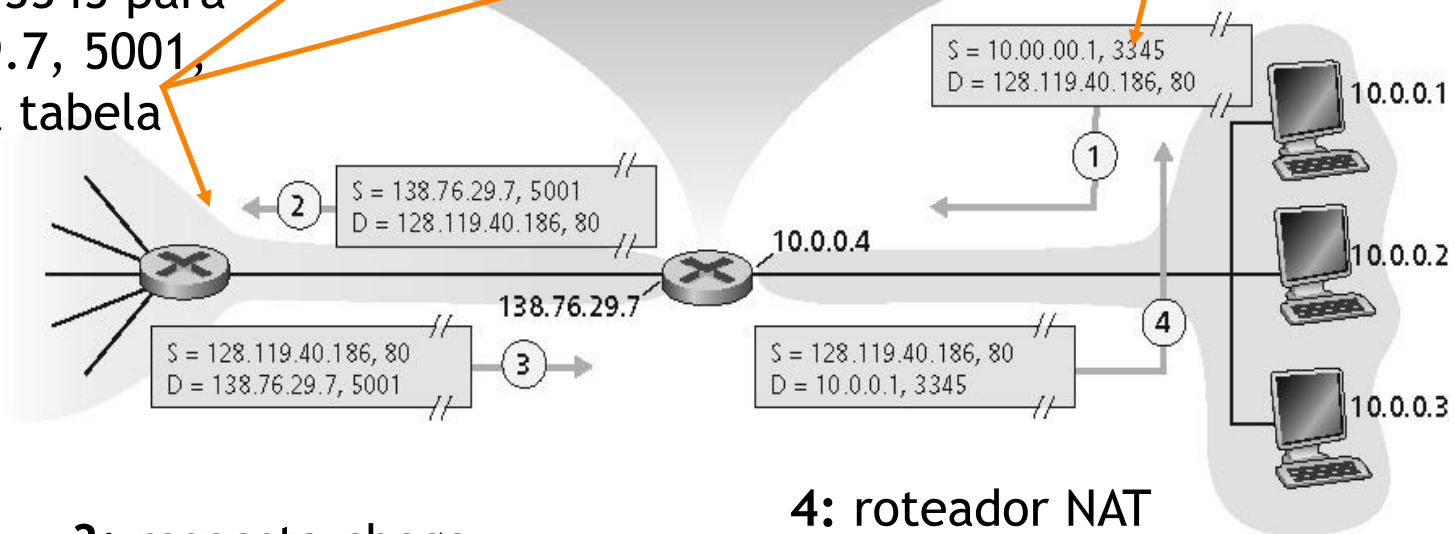
... clientes/servidores remotos responderão usando (endereço IP do NAT, nova porta #) como endereço de destino.

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

1: hospedeiro 10.0.0.1 envia datagrama para 128.119.40, 80

NAT translation table	
WAN side	LAN side
138.76.29.7, 5001	10.00.00.1, 3345
...	...

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

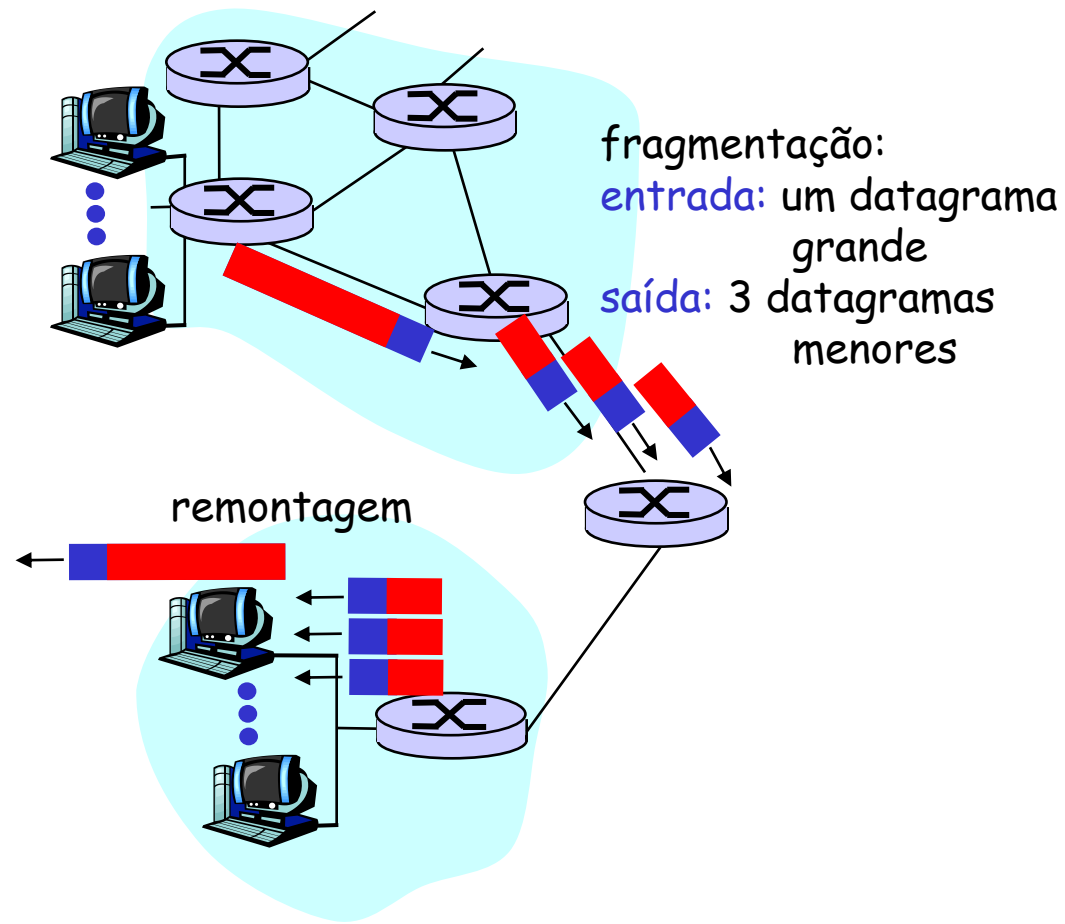


3: resposta chega endereço de destino: 138.76.29.7, 5001

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

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

	compr	ID	bit_frag	início	
	=4000	=x	=0	=0	

um datagrama grande vira
vários datagramas menores

	compr	ID	bit_frag	início	
	=1500	=x	=1	=0	

	compr	ID	bit_frag	início	
	=1500	=x	=1	=1480	

	compr	ID	bit_frag	início	
	=1040	=x	=0	=2960	

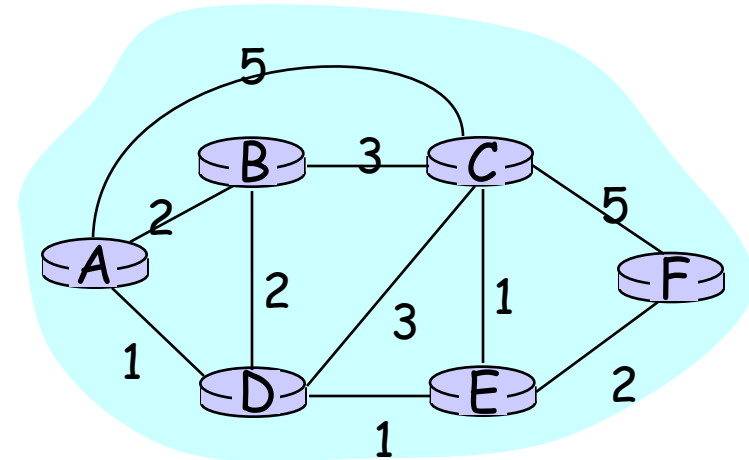
Roteamento

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"

Descentralizada:

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

Estático ou dinâmico?

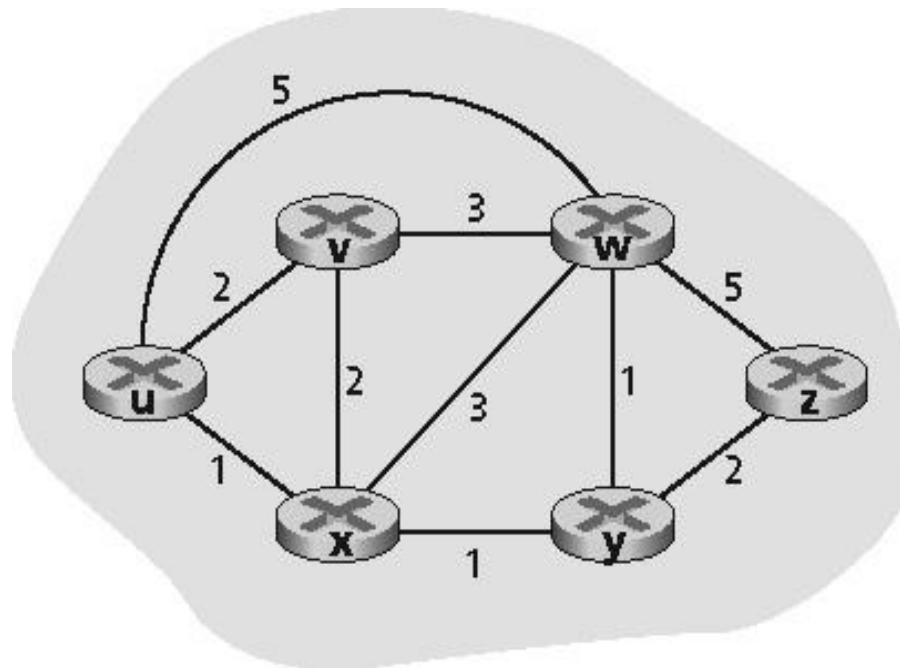
Estático:

- ❑ rotas mudam lentamente com o tempo
- ❑ Em geral, intervenção humana

Dinâmico:

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

Algoritmos de roteamento - Abstração do grafo

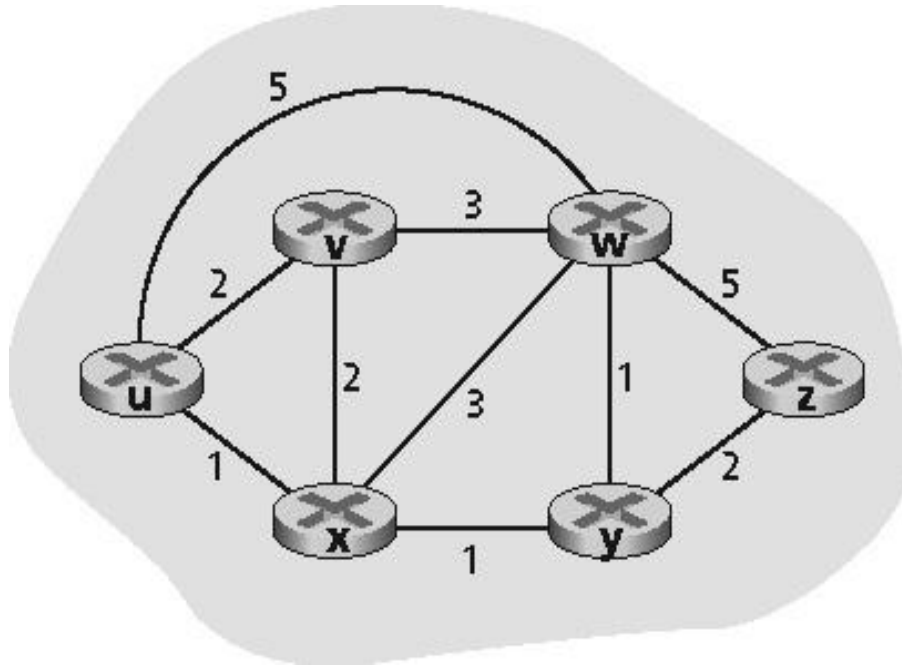


Grafo: $G = (N,E)$

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

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

Abstração do grafo: custo



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

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

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

Existem 17 possibilidades.

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

Uma algoritmo de roteamento de "estado de enlaces"

Algoritmo de Dijkstra

- topologia da rede, custos dos enlaces conhecidos por todos nós
 - realizado através de "difusão do estado de 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 nós V

4 se V adjacente ao nó A

5 então $D(V) = c(A, V)$

6 senão $D(V) = \text{infinito}$

7

8 **Repete**

9 determine W não contido em N tal que $D(W)$ é um mínimo

10 adicione W ao conjunto N

11 atualize $D(V)$ para todo V adjacente ao nó W e ainda não em N :

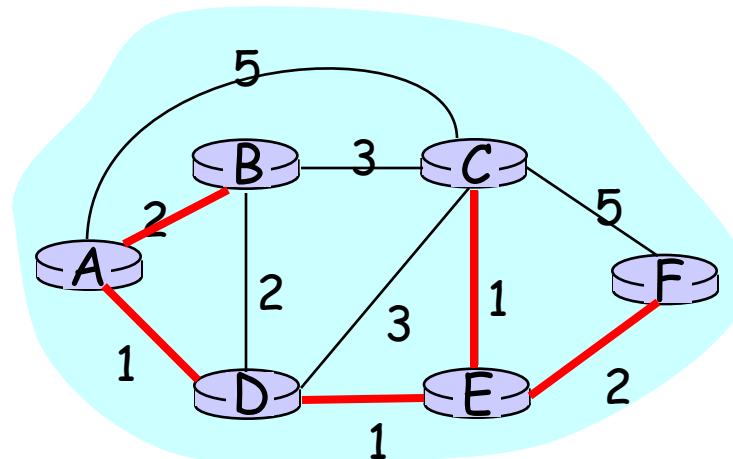
12 $D(V) = \min(D(V), D(W) + c(W, V))$

13 /* novo custo ao nó V ou é o custo anterior a V ou o menor custo de
14 caminho conhecido 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 Estado de Enlaces

- ❑ Cada roteador executa duas tarefas: testa o status de todos os seus roteadores vizinhos e divulga o estado dos enlaces para todos os outros roteadores.
- ❑ Estes pacotes enviados para todos os roteadores, são usados para construir uma base de dados topológica.
- ❑ Cada roteador deve possuir uma base idêntica, e utiliza o algoritmo de Dijkstra (shortest path first - SPF) para calcular as melhores rotas para todos os destinos
- ❑ A informação é retransmitida periodicamente, e depois de mudanças de topologia

Algoritmo Vetor de Distâncias (Bellman-Ford)

- ❑ cada roteador sabe o custo dos seus enlaces à seus vizinhos
- ❑ ele mantém uma *tabela de rotas*, a qual inclui para cada destino conhecido a distância (custo) para alcançá-lo (vetor de distâncias)
- ❑ periodicamente ele envia para seus vizinhos uma cópia da sua tabela de rotas, e recebe cópias das deles.
- ❑ se uma destas tabelas revela uma rota nova, ou uma de menor custo do que uma atual, esta será incorporada na tabela local.

Algoritmo vetor de distância (1)

Equação de Bellman-Ford

Define

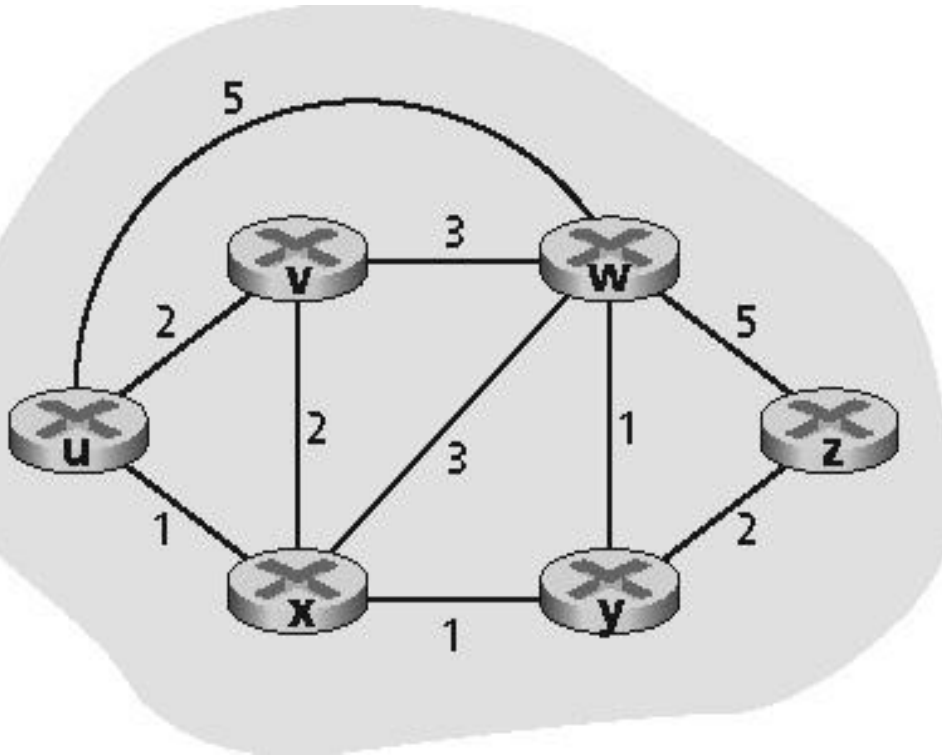
$d_x(y)$:= custo do caminho de menor custo de x
para y

Então

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

Em que min é calculado sobre todos os vizinhos
de x

Exemplo: Bellman-Ford (2)



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

A equação B-F diz que:

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

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

Algoritmo vetor de distância (3)

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

Algoritmo vetor de distância (4)

Idéia básica:

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

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

-

Algoritmo vetor de distância (5)

Iterativo, assíncrono: cada iteração local é causada por:

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

Distribuído:

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

Cada nó:

espera por (mudança no custo do enlace local na mensagem do vizinho)

recalcula estimativas

se o DV para qualquer destino mudou, **notifica** os vizinhos

Vetor de Distâncias

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

Tabela do nó x

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

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

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

Tabela do nó y

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

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

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

Tabela do nó z

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

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

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

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

