



## **Capítulo 21**

# **Camada de Rede: Mapeando Endereços, Notificação de Erros e Multicast**

## 21-1 MAPEANDO ENDEREÇOS

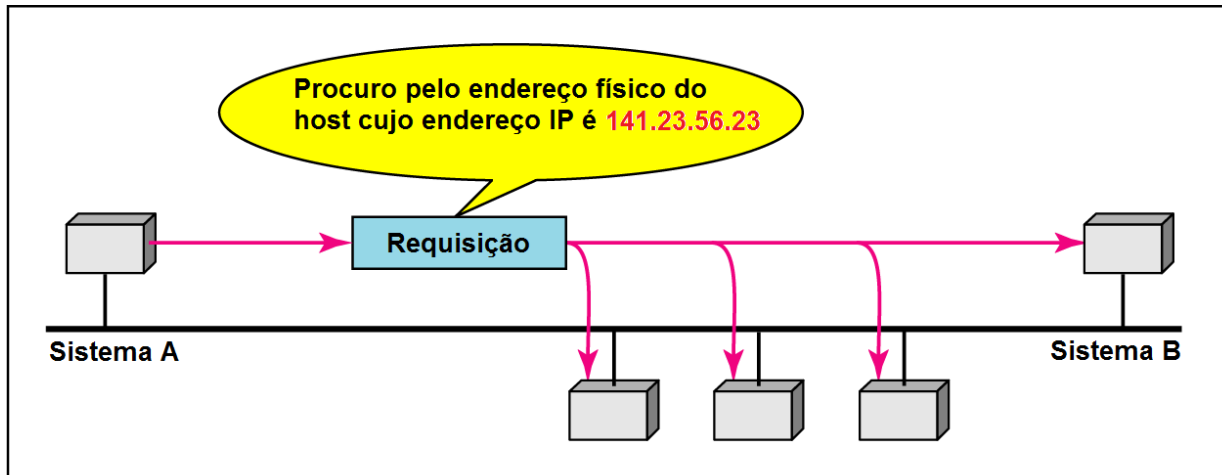
*A entrega de um pacotes para um host ou a um roteador requer 2 níveis de endereçamento: **lógico** e **físico**, sendo necessário que a rede seja capaz de efetuar o mapeamento de um endereço lógico para o seu endereço físico correspondente e viceversa. Isto pode ser feito através de mapeamento estático ou dinâmico.*

### **Tópicos discutidos nessa seção:**

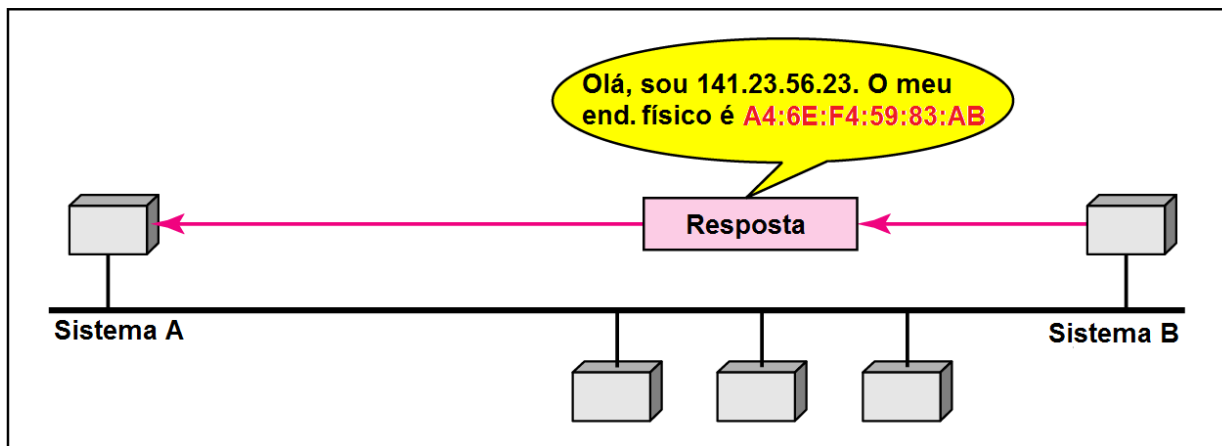
**Mapeando endereço lógico para físico**

**Mapeando endereço físico para lógico**

## Figura 21.1 Operação do ARP

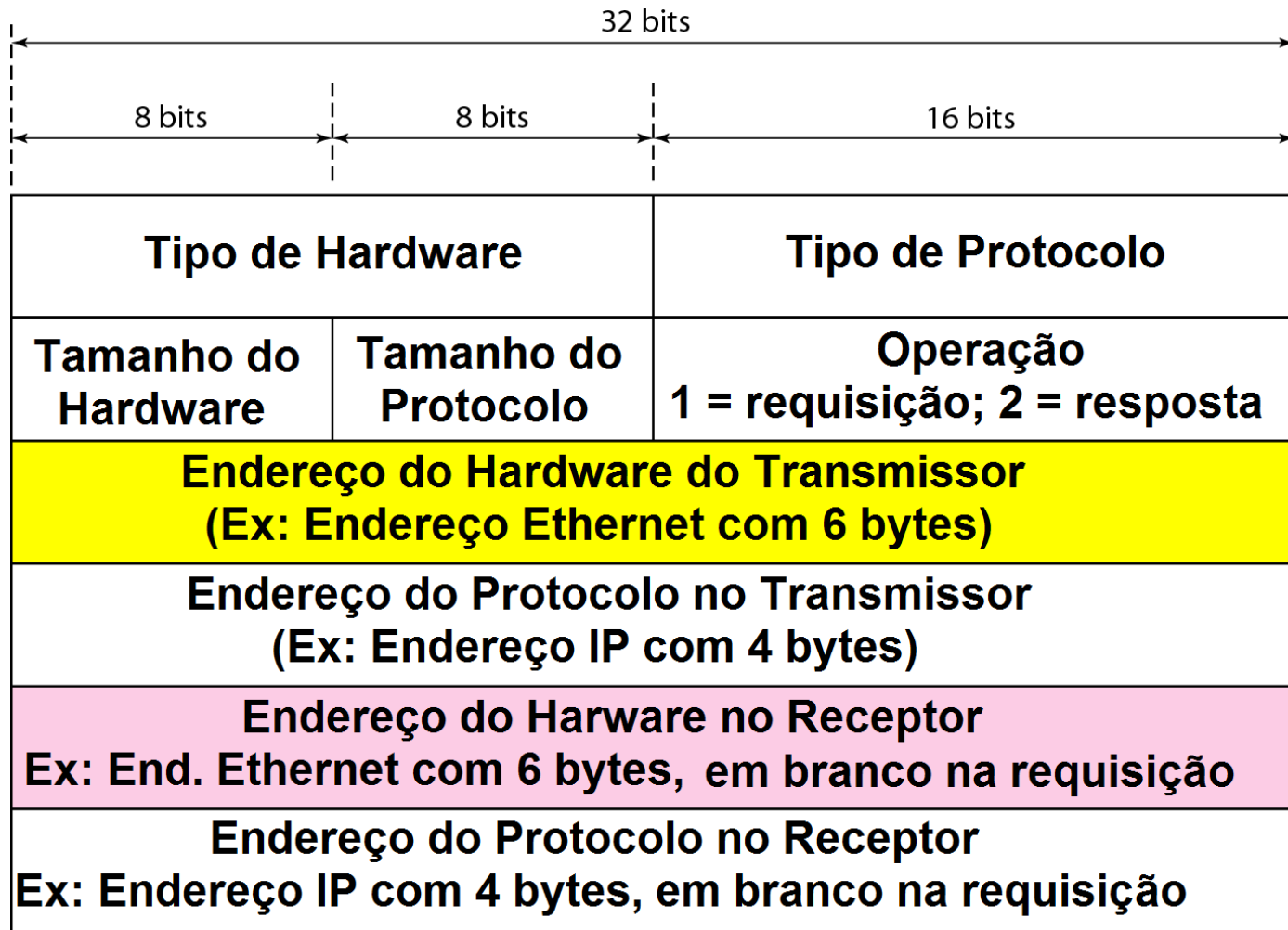


A. A requisição ARP é enviada por broadcast



B. A resposta ARP é enviada por unicast

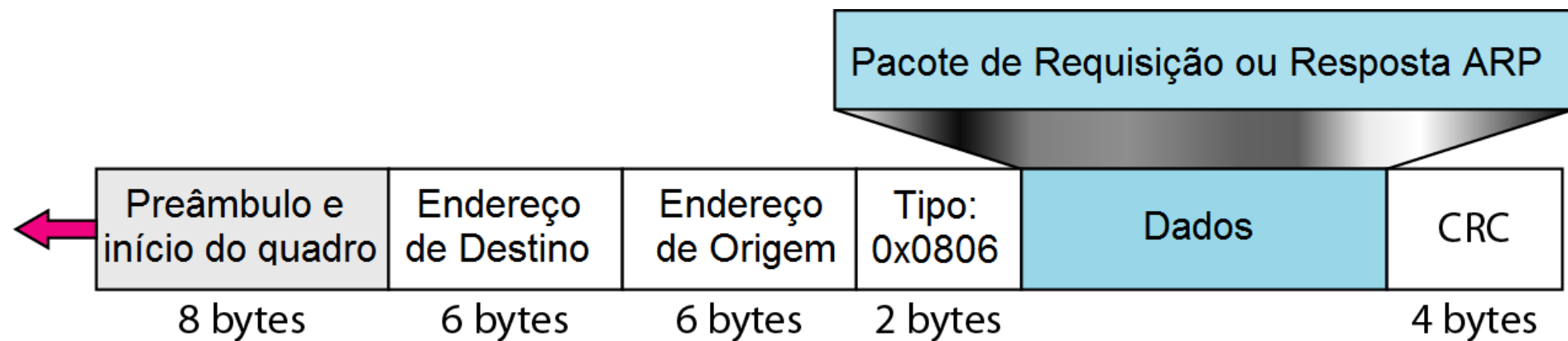
**Figura 21.2** *Pacote ARP*



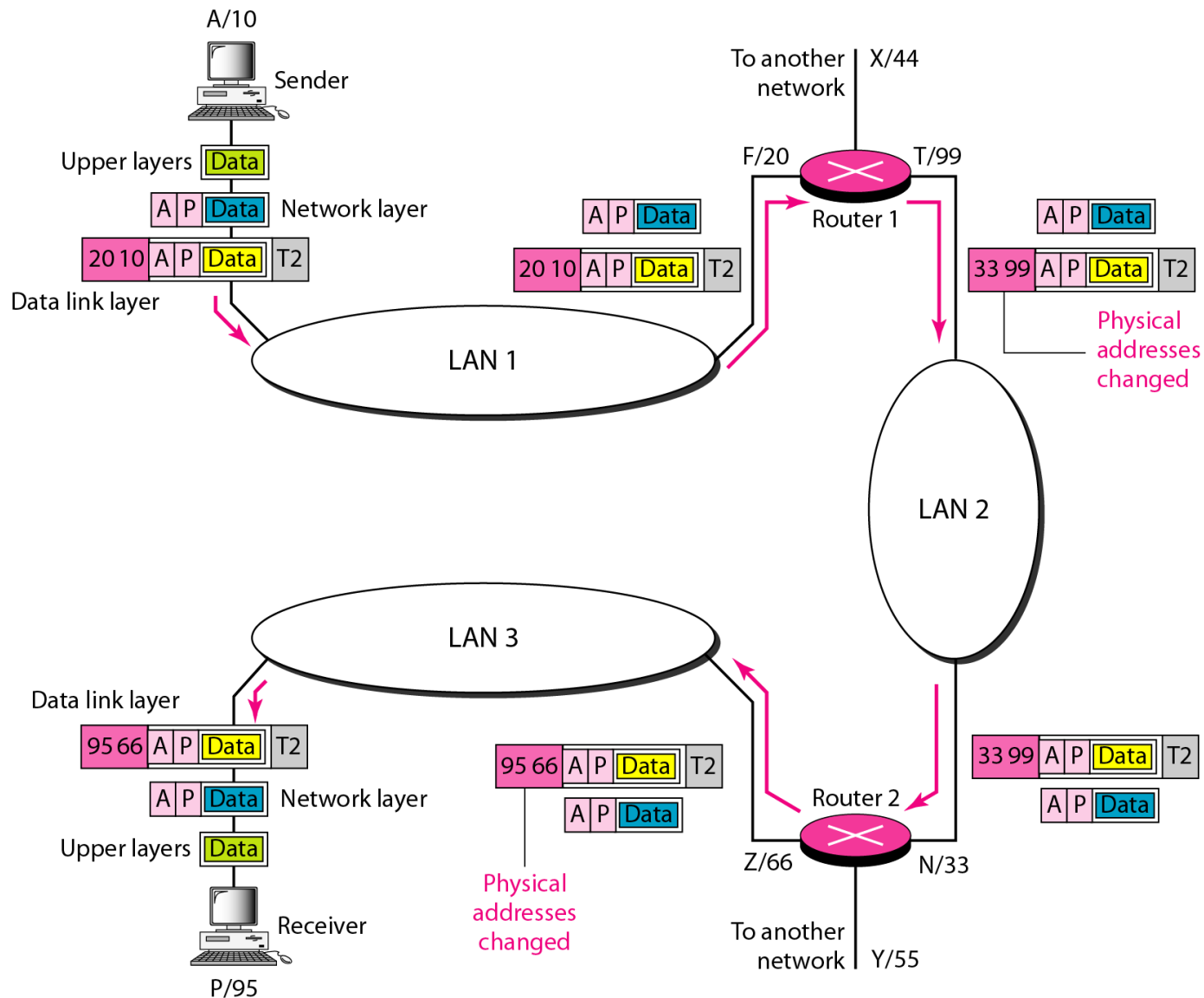
---

**Figura 21.3** *Encapsulamento do pacote ARP no quadro Ethernet*

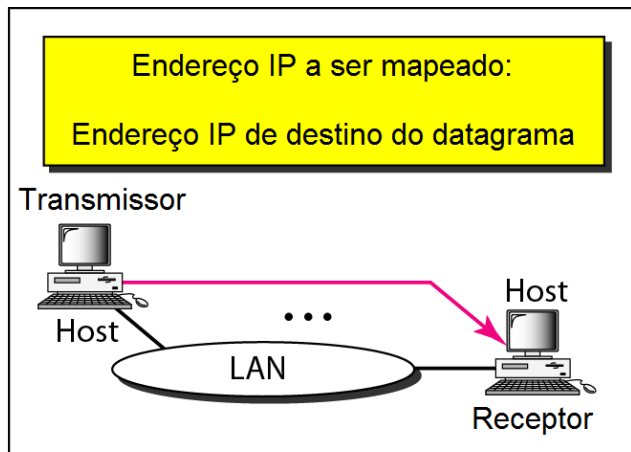
---



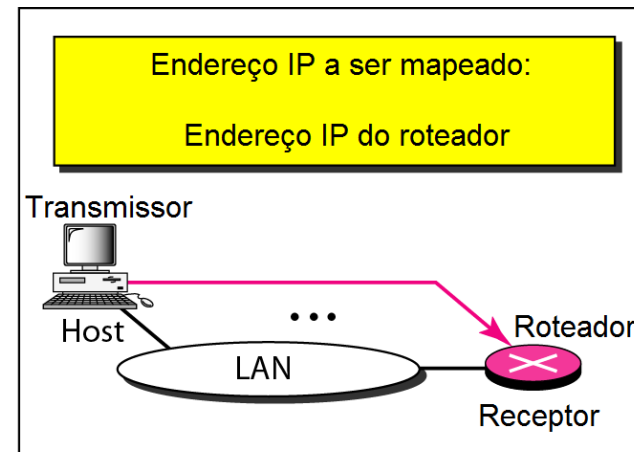
## Figura 21.4 *Quatro casos de uso do ARP*



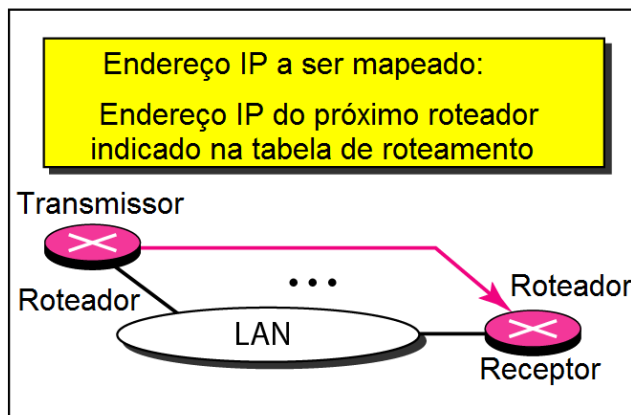
## Figura 21.4 Quatro casos de uso do ARP



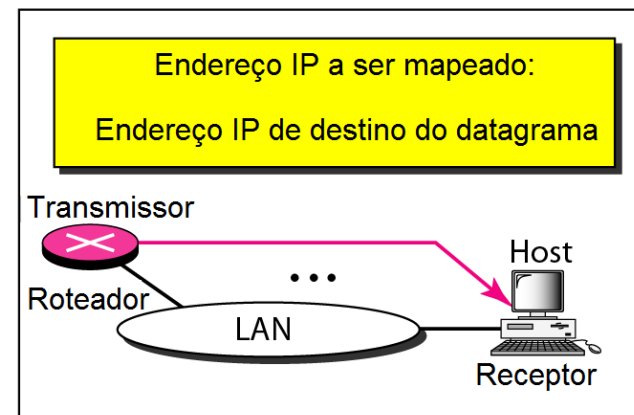
Caso 1. Um host deseja enviar um pacote IP para outro host na mesma rede.



Caso 2: Um host deseja enviar um pacote IP para outro host em outra rede. O pacote deve primeiramente se entregar ao roteador.



Caso 3: Um roteador recebe um pacote para ser enviado para um host em outra rede. O pacote deve primeiro ser entregue para o próximo roteador.



Caso 4: Um roteador recebe um pacote para ser entregue para um host na mesma rede.



*Nota*

**Uma requisição ARP é feita por  
broadcast;  
A resposta ARP é unicast.**





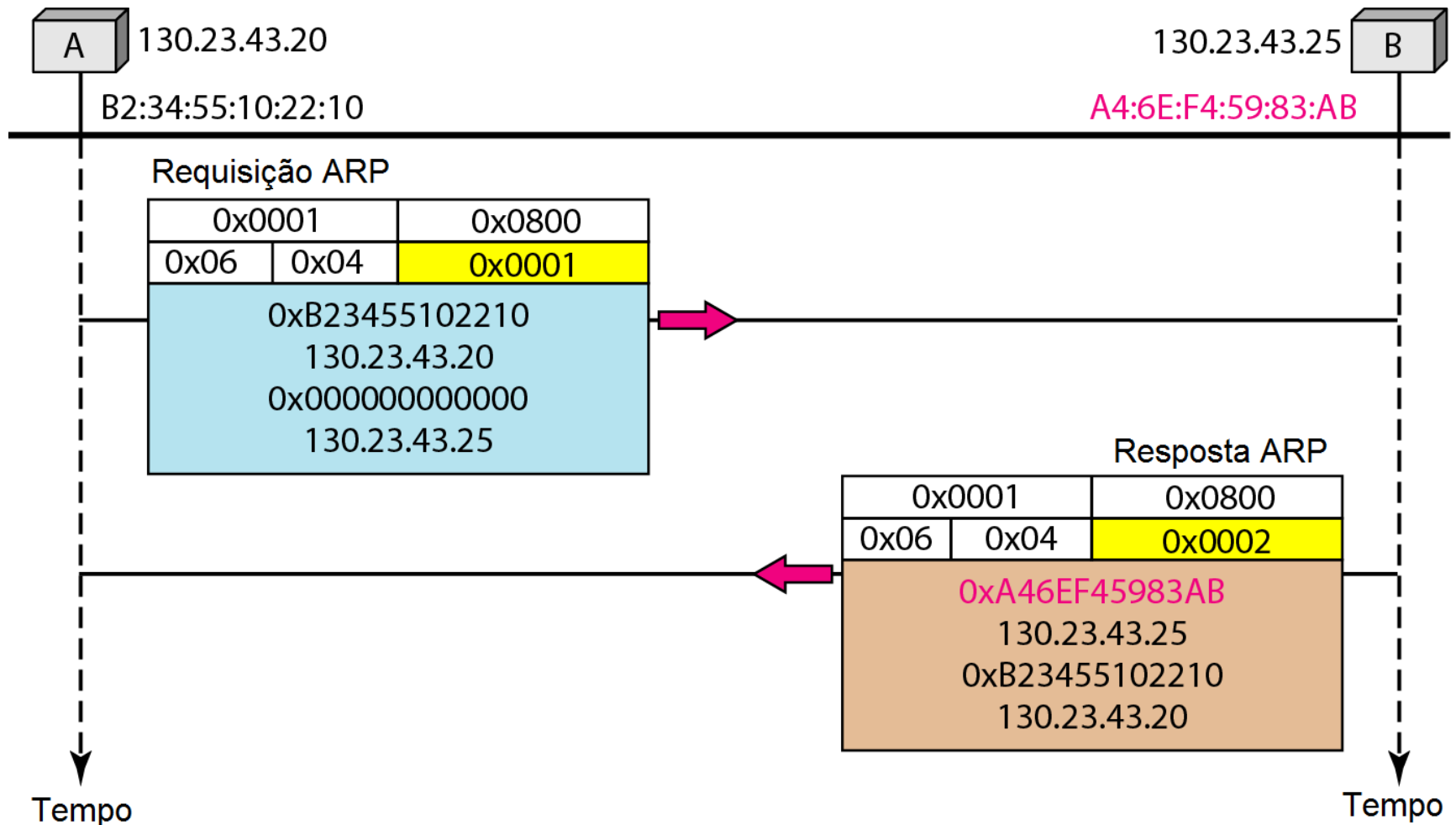
## *Exemplo 21.1*

*Um host com endereço IP 130.23.43.20 e endereço físico B2:34:55:10:22:10 deseja enviar um pacote para outro host com endereço IP 130.23.43.25 e endereço físico A4:6E:F4:59:83:AB. Os dois hosts estão na mesma rede Ethernet. Mostre os pacotes de requisição ARP e a resposta encapsulada no quadro Ethernet.*

### *Solução*

*A Figura 21.5 mostra os pacotes de requisição e resposta ARP. Note que o campo de dados ARP, neste caso, é de 28 bytes, e que os endereços individuais não se encaixam no espaço de 4 bytes.*

**Figura 21.5** *Exemplo 21.1, requisição e resposta ARP*



*É inviável enviar requisições ARP para cada pacote*

## *Solução*

*Manutenção de uma tabela cache de vínculos de IP*

## *Efeito*

*Uso do protocolo ARP apenas 1 vez e coloca os resultados em uma tabela*

*Consulta essa tabela a cada transmissão de um pacote para o mesmo destino*

*Entradas da tabela ARP expiram após algum tempo*

*Evita excesso de vínculos de IPs*

*Timeout: geralmente de 20 minutos*

---

## *Visualizando a Tabela ARP no Linux*


---

```
$ arp -a
is1 (172.16.55.251) at 08:00:20:82:D5:1D [ether] on eth0
dog (172.16.55.178) at 08:00:20:9A:4F:25 [ether] on eth0
warthog (172.16.55.33) at 08:00:20:87:B7:6E [ether] on eth0
fs1 (172.16.55.250) at 08:00:20:8D:19:7B [ether] on eth0
sloth (172.16.55.36) at 08:00:20:71:97:06 [ether] on eth0
crow (172.16.55.183) at 08:00:20:82:DA:43 [ether] on eth0
snipe (172.16.55.1) at 0A:00:20:18:48:31 [ether] on eth0
duck (172.16.55.110) at 08:00:5A:09:C3:46 [ether] on eth0
? (172.16.55.216) at 08:00:4E:34:70:92 [ether] on eth0
rabbit (172.16.55.181) at 00:20:AF:16:95:B0 [ether] on eth0
deer (172.16.55.145) at 08:00:69:0A:47:2E [ether] on eth0
```

---

# *Visualizando a tabela ARP no Windows*

---

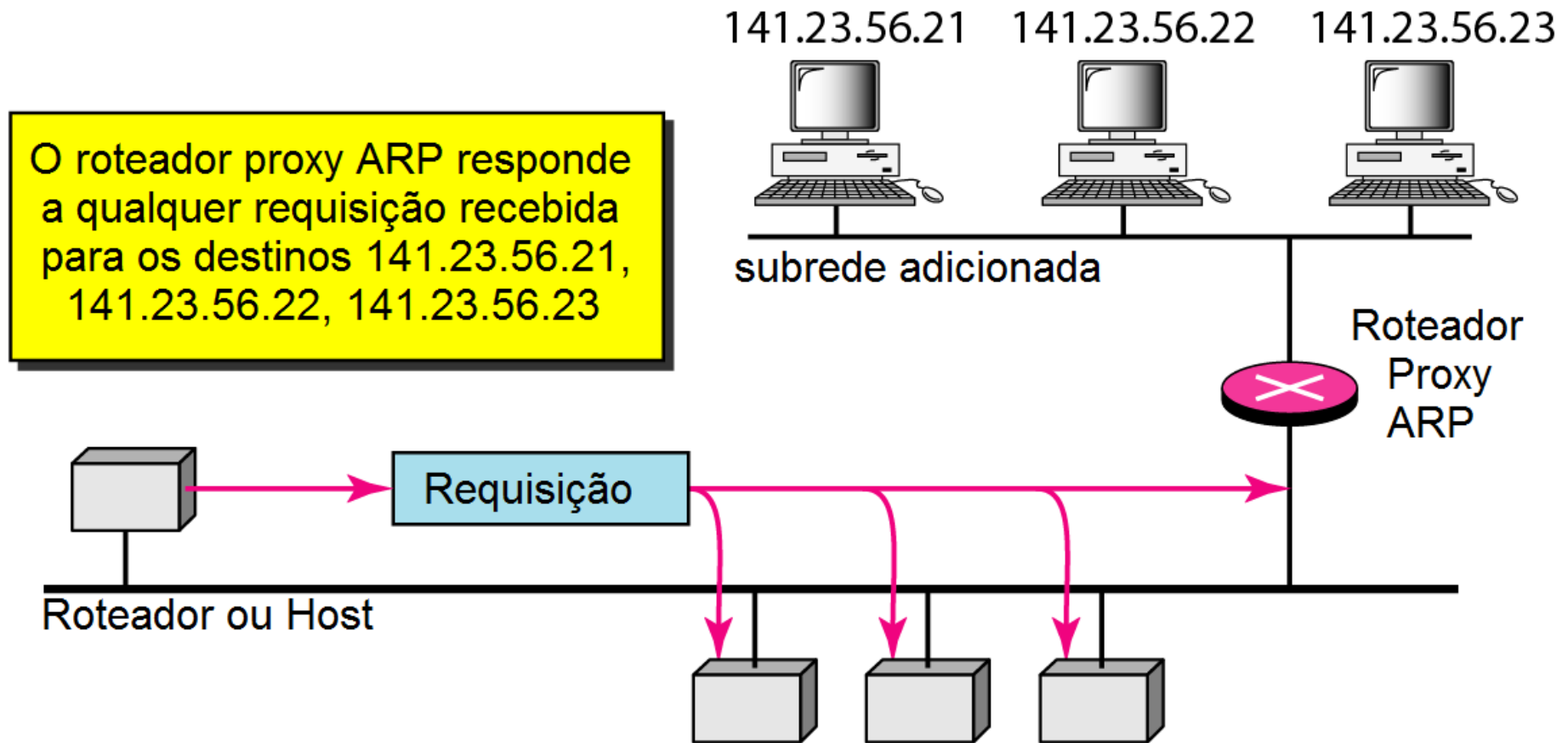
 Administrador: Prompt de Comando

```
C:\Windows\system32>arp -a
```

```
Interface: 192.168.0.126 --- 0x7
```

Endereço IP	Endereço físico	Tipo
192.168.0.1	c8-d7-19-d4-3c-61	dinâmico
192.168.0.106	6c-ad-f8-e7-6f-9c	dinâmico
192.168.0.255	ff-ff-ff-ff-ff-ff	estático
224.0.0.2	01-00-5e-00-00-02	estático
224.0.0.22	01-00-5e-00-00-16	estático
224.0.0.251	01-00-5e-00-00-fb	estático
224.0.0.252	01-00-5e-00-00-fc	estático
239.255.255.250	01-00-5e-7f-ff-fa	estático
255.255.255.255	ff-ff-ff-ff-ff-ff	estático

**Figura 21.6** *Proxy ARP*



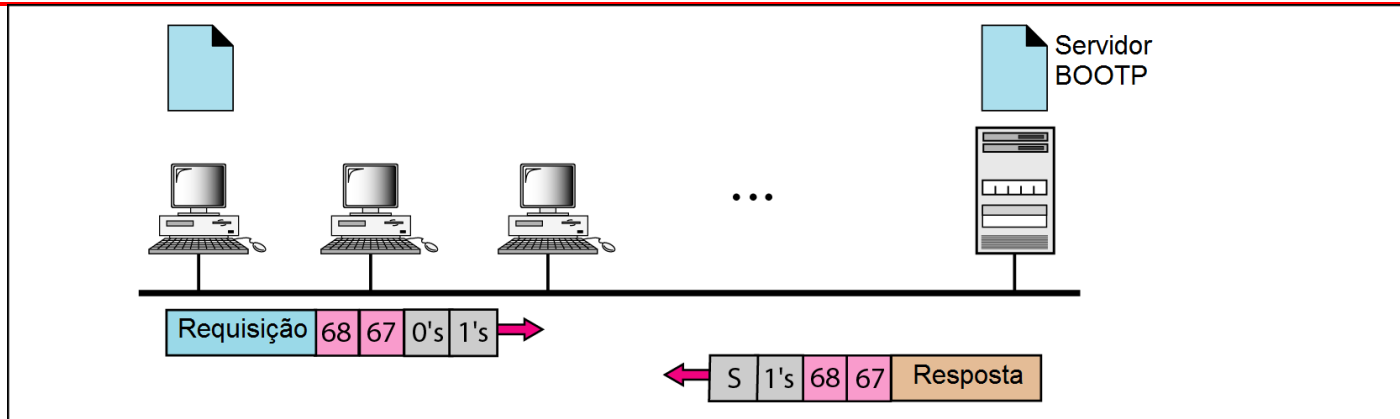
- *RARP (Reverse Address Resolution Protocol)*
  - *protocolo de resolução de endereço reverso*
  - *tem por finalidade mapear o endereço lógico de uma máquina a partir de seu endereço físico*
- *Exemplo de uso*
  - *PC sem disco pode ser iniciada a partir da memória ROM*
  - *Não inclui o endereço IP, pois estes, em uma rede, são atribuídos logicamente e sob demanda por um administrador de redes*
  - *O endereço físico, que é localmente exclusivo pode ser usado para obter um endereço lógico*

- *Mensagens*
  - *RARP Request*
  - *RARP Reply*
- *A máquina solicitante deve estar rodando RARP Client*
- *A máquina que responde tem de estar rodando RARP Server*
- *Problema do RARP*
  - *No broadcast no nível de enlace de dados, o endereço físico de broadcast tem todos os bits em 1s, mas a requisição RARP, não ultrapassa os limites de uma rede local*
  - *Se um administrador tiver várias redes ou várias sub-redes, ele precisa configurar um RARP Server para cada uma delas*

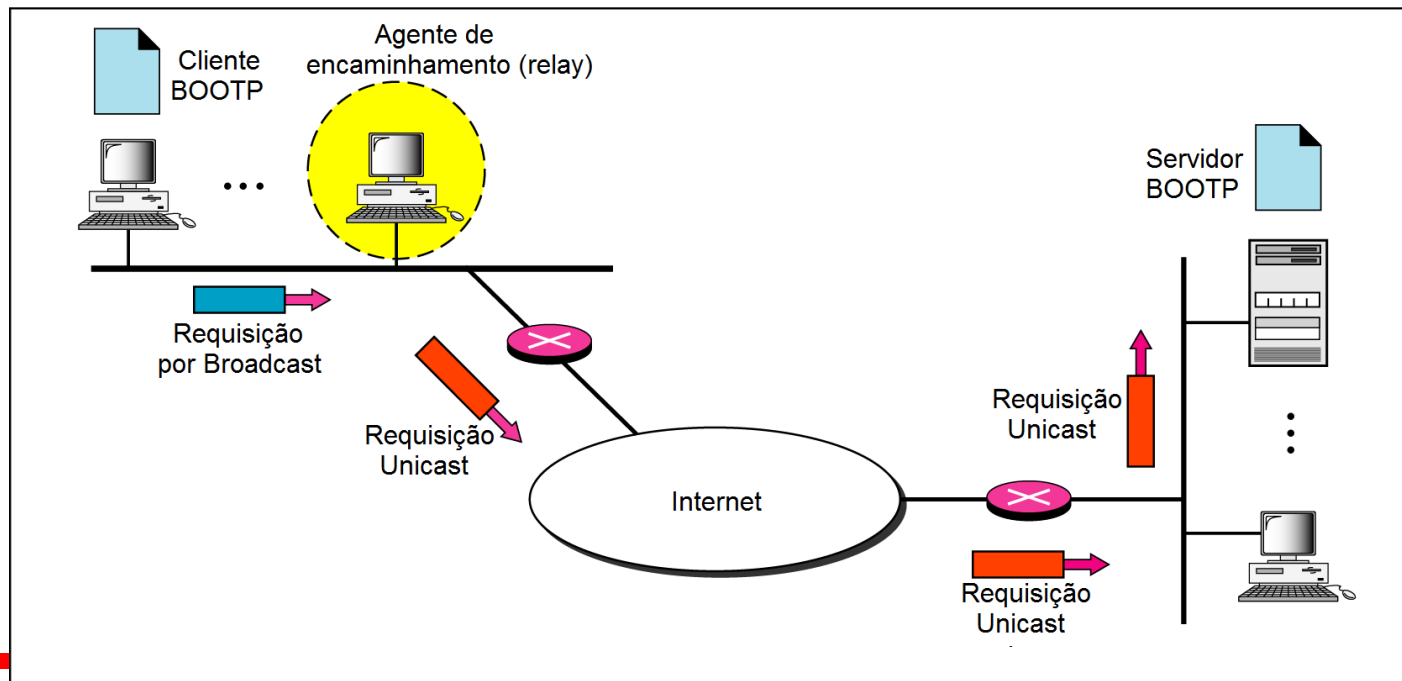


- *Protocolo cliente/servidor desenvolvido para facilitar o mapeamento entre endereços físicos e endereços lógicos*
- *Protocolo de camada de aplicação*
- *Vantagem:*
  - *O administrador pode implementar o cliente e o servidor em uma mesma rede ou então em redes diferentes*
  - *Mensagens BOOTP são encapsuladas em pacotes UDP e o próprio pacote UDP é encapsulado em um pacote IP.*

## Figura 21.7 *Cliente e servidor BOOTP na mesma rede e em redes distintas*



A. Cliente e Servidor na mesma rede



B. Cliente e Servidor em redes distintas



*Nota*

**O DHCP provê alocação de endereços  
estaticos e dinamicos que podem ser  
manual ou automatico.**

# DHCP: Dynamic Host Configuration Protocol

Objetivo: permitir que o host obtenha dinamicamente um endereço IP da rede

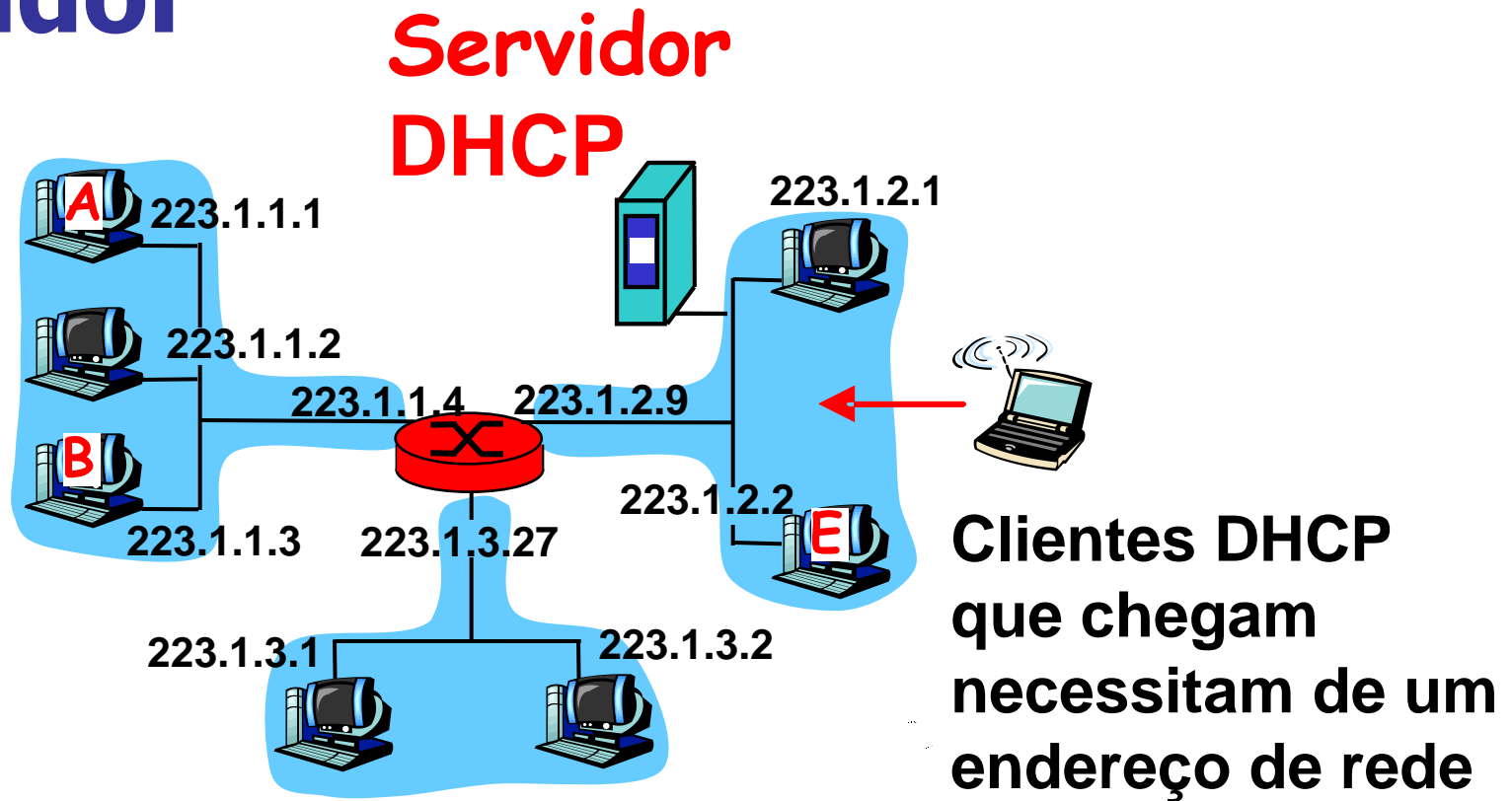
Protocolo Cliente-Servidor

Permite o reuso de endereços

Mensagens DHCP:

- DHCP discover
- DHCP offer
- DHCP request
- DHCP ack

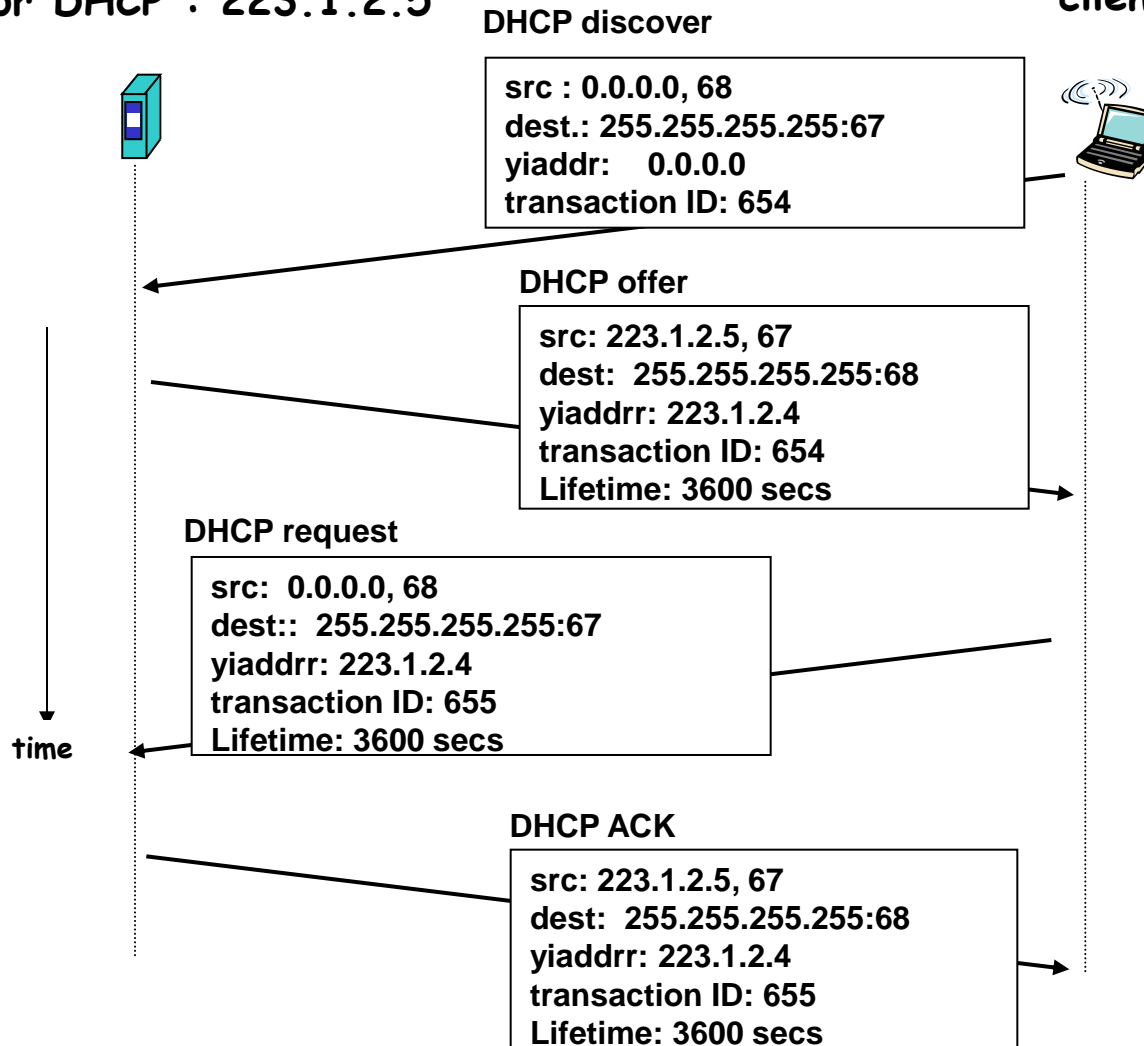
# Cenário DHCP cliente-servidor



# Cenário DHCP cliente-servidor

Servidor DHCP : 223.1.2.5

cliente



## 21-2 ICMP

*O protocolo IP não possui mecanismos de notificação e correção de erros. O protocolo IP também não tem mecanismo para o gerenciamento de consultas dos host. O **Protocolo de Mensagens de Controle na Internet (ICMP)** foi desenvolvido para compensar essas deficiências, complementando o protocolo IP.*

### **Tópicos discutidos nessa seção:**

**Tipos de Mensagens**

**Formato das Mensagens**

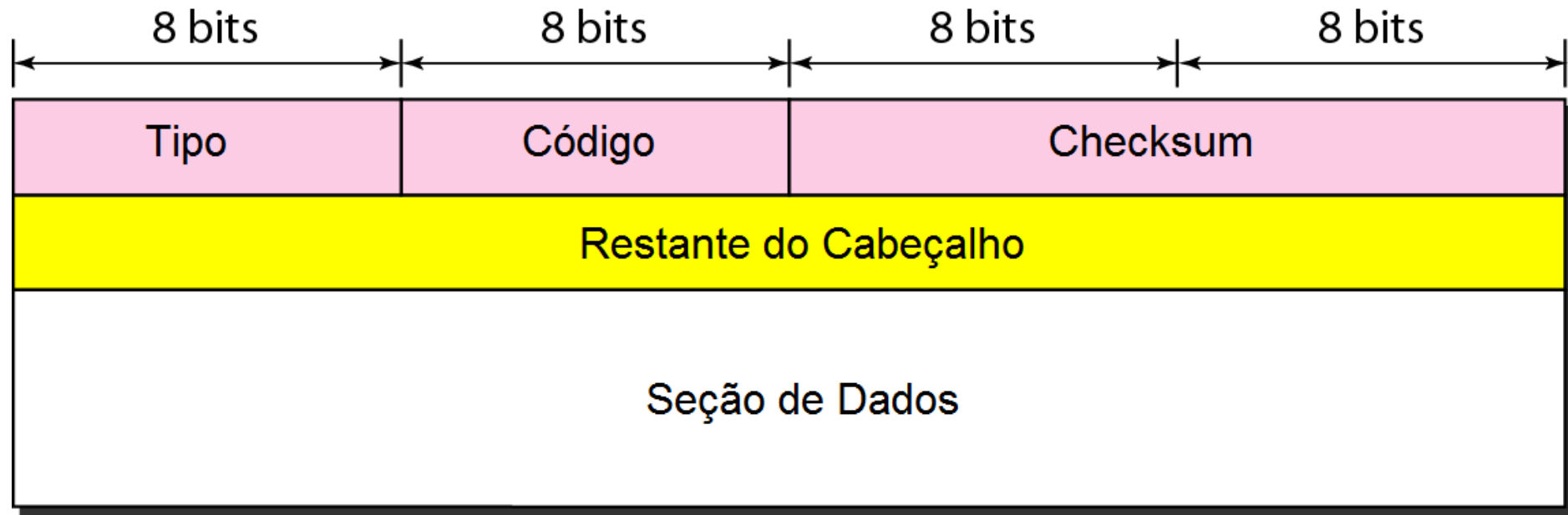
**Notificação de Erros e Consultas**

**Ferramentas de Debugging**

---

**Figura 21.8** *Formato geral das mensagens ICMP*

---







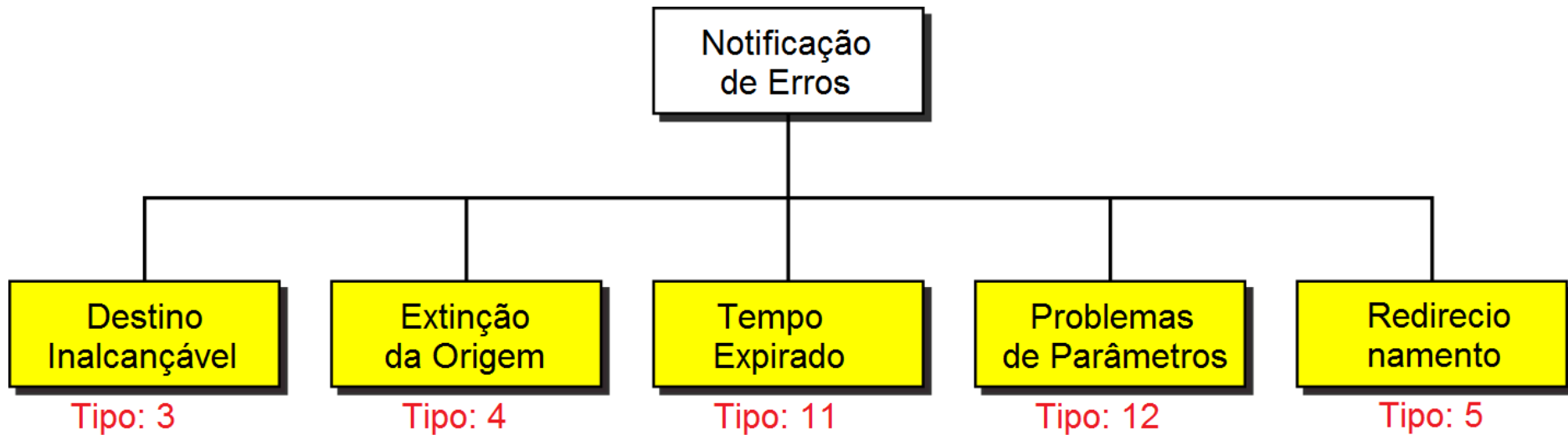
*Nota*

**As mensagens ICMP de notificação de erros sempre são enviadas para a origem dos dados.**

---

**Figura 21.9** *Mensagens de notificação de erros*

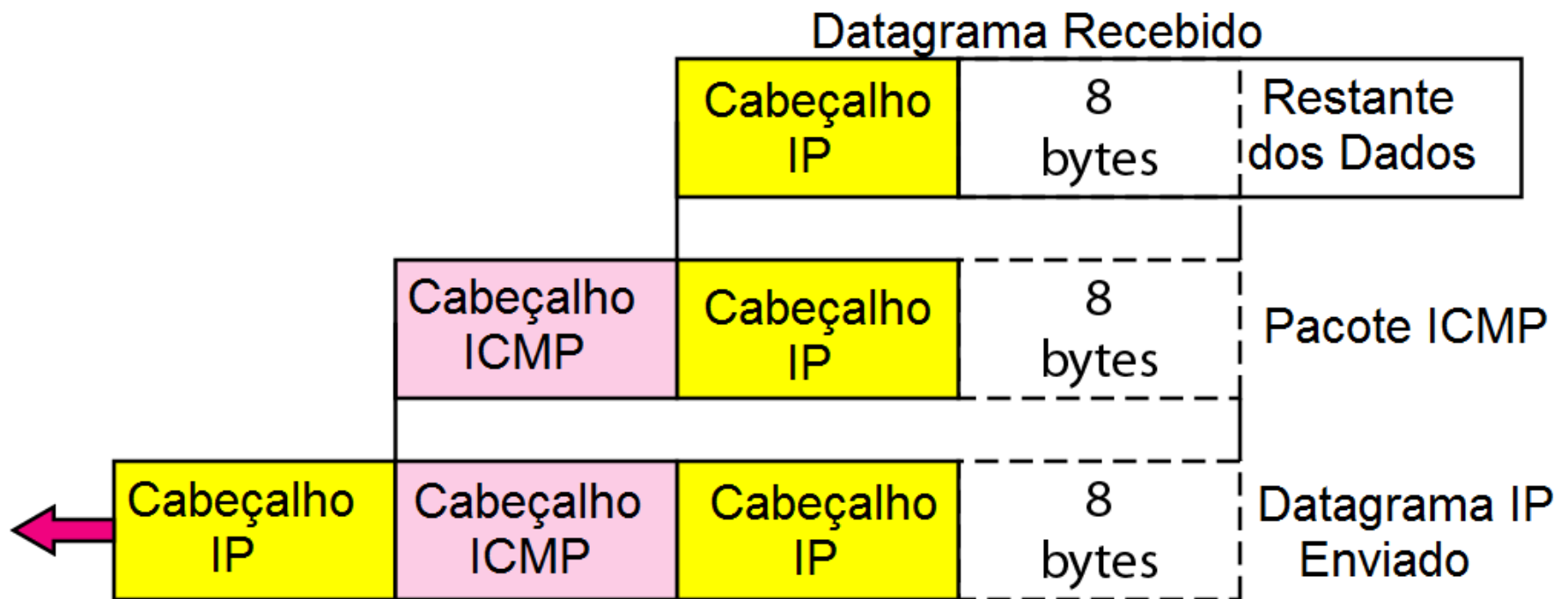
---



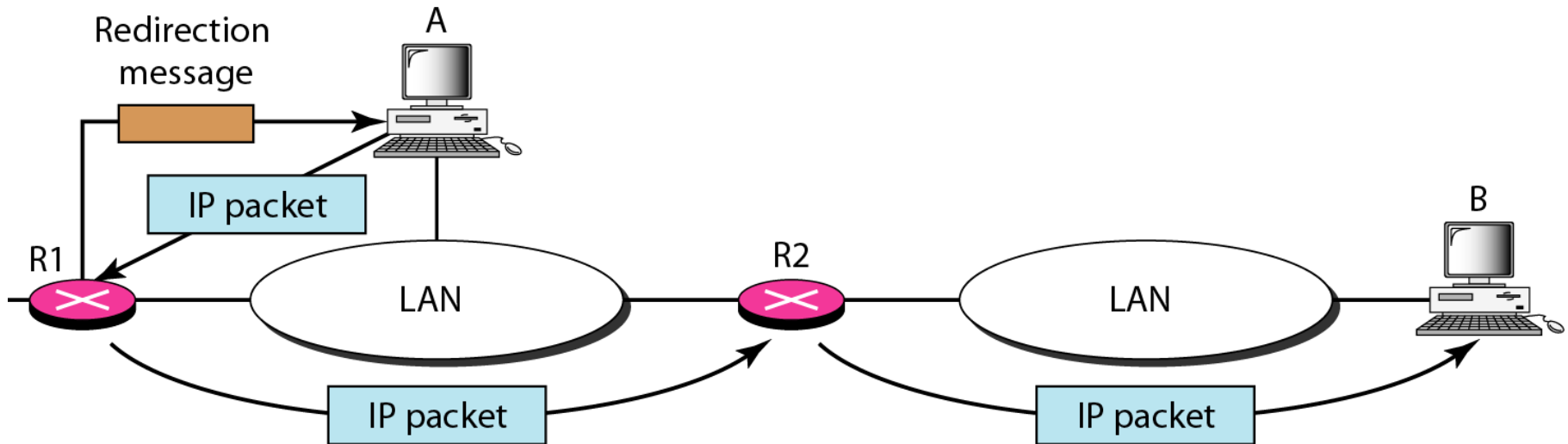
## Pontos importantes sobre as mensagens ICMP:

- ❑ Nenhuma mensagem de erro ICMP será gerada em resposta a um datagrama carregando uma mensagem de erro ICMP.
- ❑ Nenhuma mensagem de erro ICMP será gerada para um datagrama fragmentado que não é o 1º fragmento.
- ❑ Nenhuma mensagem de erro ICMP será gerada para um datagrama contendo um endereço multicast.
- ❑ Nenhuma mensagem de erro ICMP será gerada para um datagrama contendo um endereço especial com 127.0.0.0 or 0.0.0.0.

**Figura 21.10** *Conteúdo do campo de dados para as mensagem de erros*



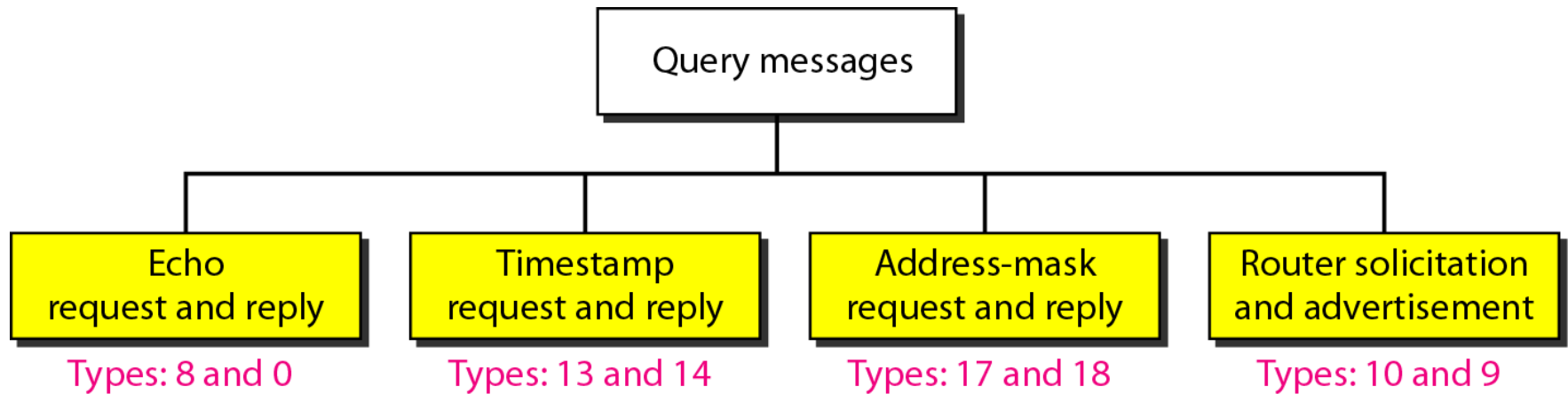
**Figura 21.11** *Conceito de redirecionamento*



---

## Figura 21.12 *Mensagens de Consulta*

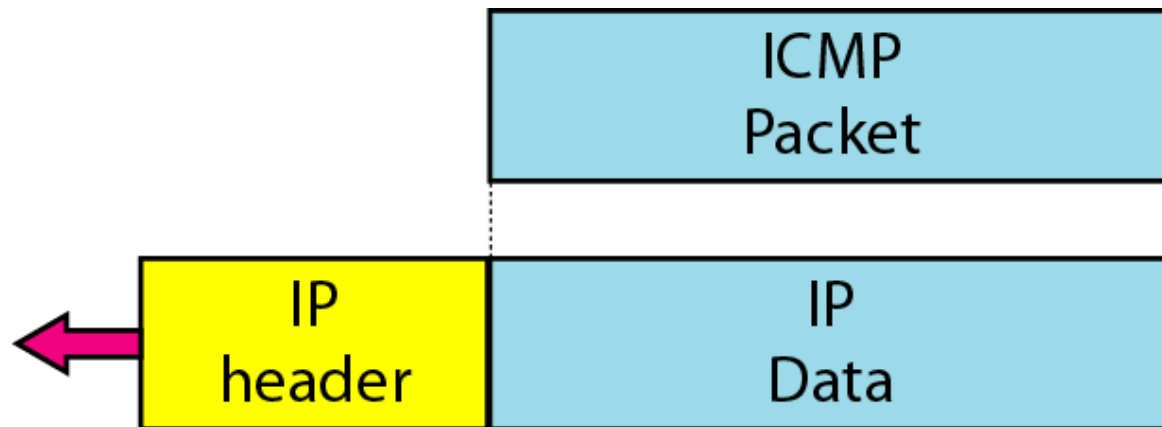
---



---

**Figura 21.13** *Encapsulamento mensagens ICMP de consulta*

---



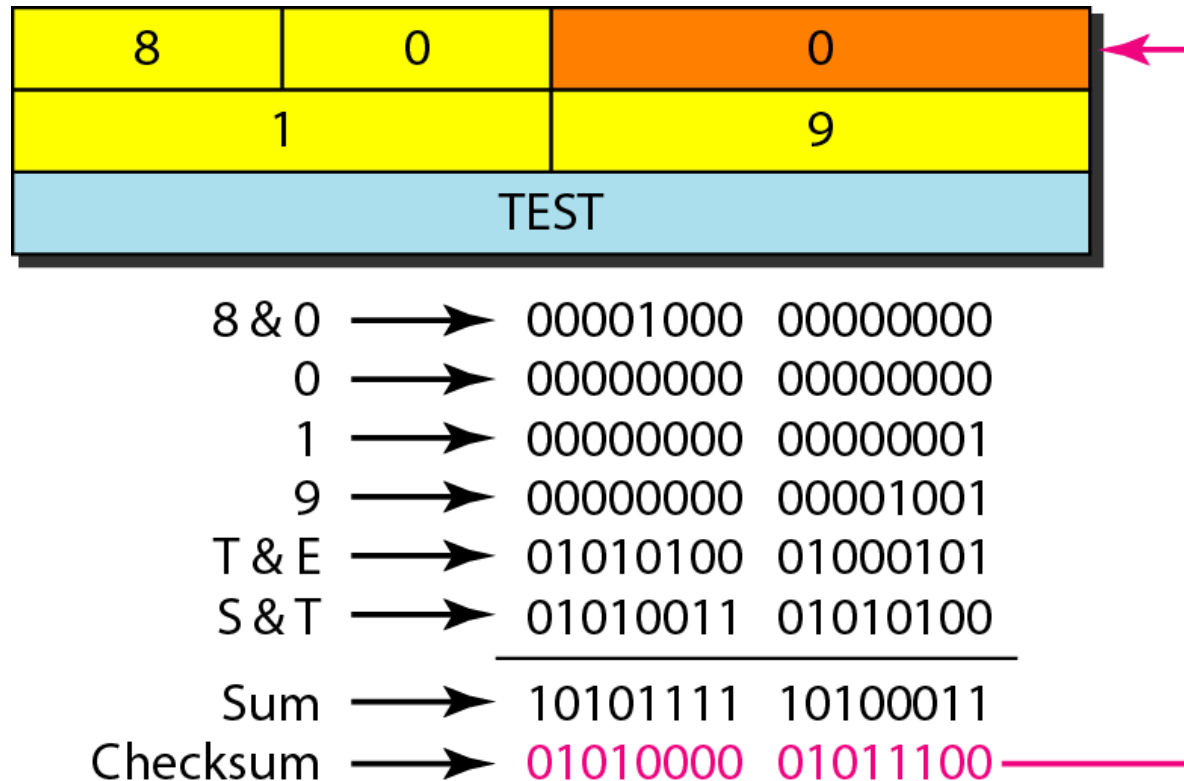


## Exemplo 21.2

*A Figura 21.14 mostra um exemplo de cálculo de soma de verificação para uma mensagem de requisição de echo. Inicialmente define-se o identificador com o valor 1 e o n° de sequência igual a 9. A mensagem é dividida em 16 bits (2 bytes). São somados os bits de cada byte e a determinado o complemento da soma. O resultado é colocado no campo checksum.*



**Figura 21.14** *Exemplo de cálculo do checksum*





## Exemplo 21.3

*O comando ping é usado para testar a conectividade com o servidor fhda.edu. O resultado é mostrado no próximo slide. O comando ping envia mensagens com números de seqüência a partir de 0. Cada teste retorna o tempo de RTT (Round Trip Time – Tempo de Ida e Volta) e o campo TTL (Time to Live – Tempo de Vida) do datagrama IP que encapsula a mensagem ICMP.*

*No início, o comando ping define o número de bytes de dados como 56 e o número total de bytes como 84 (8 bytes do cabeçalho ICMP + 20 bytes do cabeçalho IP + 56bytes de dados).*

*Note que, cada ping define o número de bytes de 64, que é o n° total de bytes no pacote ICMP (56 + 8).*

## *Exemplo 21.3 (continuação)*

```
$ ping fhda.edu
```

```
PING fhda.edu (153.18.8.1) 56 (84) bytes of data.
```

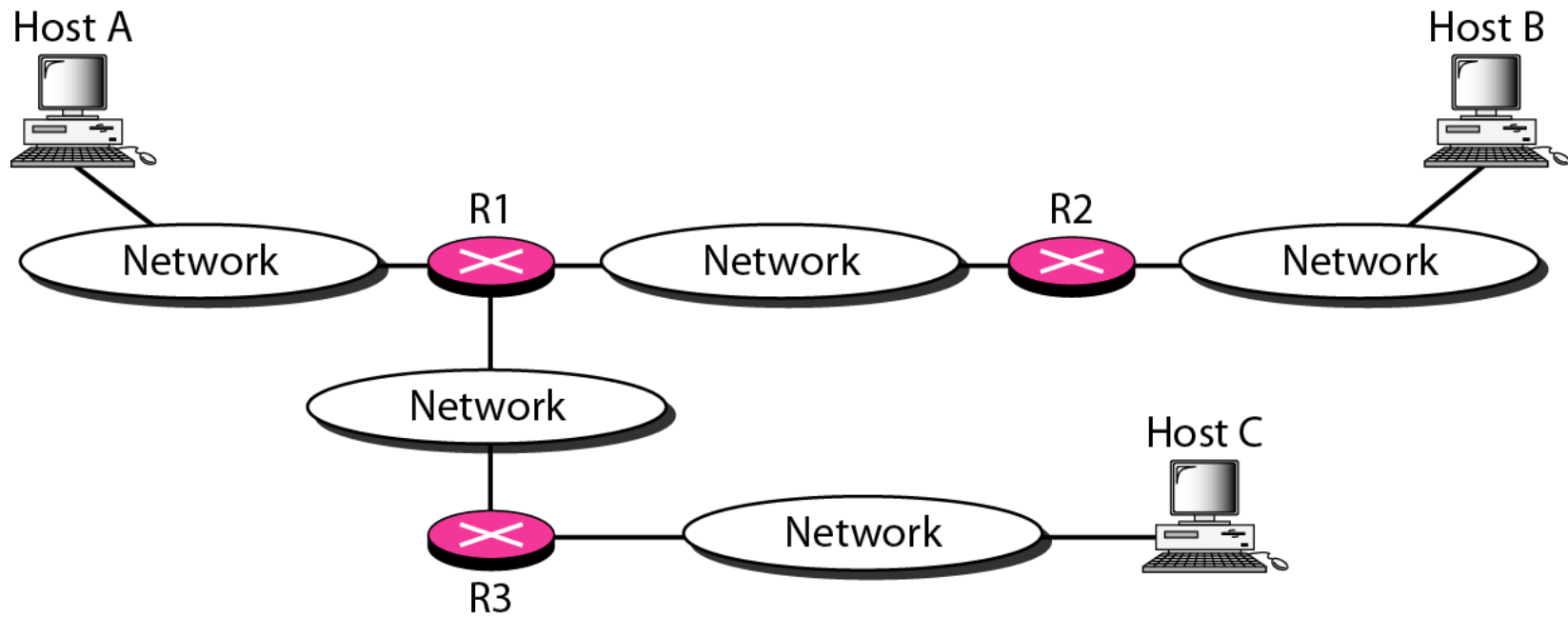
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=0	ttl=62	time=1.91 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=1	ttl=62	time=2.04 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=2	ttl=62	time=1.90 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=3	ttl=62	time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=4	ttl=62	time=1.93 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=5	ttl=62	time=2.00 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=6	ttl=62	time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=7	ttl=62	time=1.94 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=8	ttl=62	time=1.97 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=9	ttl=62	time=1.89 ms
64 bytes from tiptoe.fhda.edu (153.18.8.1): icmp_seq=10	ttl=62	time=1.98 ms

```
--- fhda.edu ping statistics ---
```

```
11 packets transmitted, 11 received, 0% packet loss, time 10103ms
```

```
rtt min/avg/max = 1.899/1.955/2.041 ms
```

**Figura 21.15** *Operação do comando traceroute*



## Exemplo 21.4

*Nós usamos o programa traceroute para encontrar a rota a partir do computador voyager.deanza.edu para o servidor fhda.edu.*

```
$ traceroute fhda.edu
traceroute to fhda.edu (153.18.8.1), 30 hops max, 38 byte packets
 1 Dcore.fhda.edu (153.18.31.254) 0.995 ms 0.899 ms 0.878 ms
 2 Dbackup.fhda.edu (153.18.251.4) 1.039 ms 1.064 ms 1.083 ms
 3 tiptoe.fhda.edu (153.18.8.1) 1.797 ms 1.642 ms 1.757 ms
```

*A linha não numerada após o comando mostra que o IP de destino é 153.18.8.1. O Pacote contém 38 bytes (20 bytes do cabeçalho IP + 8 bytes do cabeçalho ICMP + 10 bytes de dados do aplicativo). Os dados são usados pelo aplicativo traceroute para acompanhar os pacotes.*



## *Exemplo 21.4 (continuação)*

*A primeira linha mostra o primeiro roteador visitado, que tem o nome Dcore.fhda.edu e endereço IP 153.18.31.254. O RTT da 1ª rodada foi de 0,995 ms, o 2º foi de 0,899 ms, e no 3º foi de 0,878 ms.*

*A segunda linha mostra o segundo roteador visitado, que tem o nome Dbackup.fhda.edu e endereço IP 153.18.251.4. Os três tempos de RTT também são mostrados.*

*A terceira linha mostra o host de destino. Sabemos que este é o host de destino, porque não existem mais linhas. O host de destino é o fhda.edu servidor, mas tem o nome de tiptoe.fhda.edu e endereço IP 153.18.8.1. Os RTT também são mostrados.*



## *Exemplo 21.5*

---

*Neste Exemplo, vamos traçar uma rota mais longa, o percurso até xerox.com (ver próximo slide). Aqui há 17 saltos entre destino e origem. Note ocorre em algumas vezes valores de RTT discrepantes. Pode ser que um roteador estava ocupado demais para processar o pacote de imediato.*

## *Exemplo 21.5 (continuação)*

**\$ traceroute xerox.com**

**traceroute to xerox.com (13.1.64.93), 30 hops max, 38 byte packets**

1	Dcore.fhda.edu	(153.18.31.254)	0.622 ms	0.891 ms	0.875 ms
2	Ddmz.fhda.edu	(153.18.251.40)	2.132 ms	2.266 ms	2.094 ms
3	Cinic.fhda.edu	(153.18.253.126)	2.110 ms	2.145 ms	1.763 ms
4	cenic.net	(137.164.32.140)	3.069 ms	2.875 ms	2.930 ms
5	cenic.net	(137.164.22.31)	4.205 ms	4.870 ms	4.197 ms
....		....	...	....	...
14	snfc21.pbi.net	(151.164.191.49)	7.656 ms	7.129 ms	6.866 ms
15	sbcglobal.net	(151.164.243.58)	7.844 ms	7.545 ms	7.353 ms
16	pacbell.net	(209.232.138.114)	9.857 ms	9.535 ms	9.603 ms
17	209.233.48.223	(209.233.48.223)	10.634 ms	10.771 ms	10.592 ms
18	alpha.Xerox.COM	(13.1.64.93)	11.172 ms	11.048 ms	10.922 ms