Redes de Computadores

O protocolo IP

Departamento de Informática da
FCT/UNL

Objetivos do Capítulo

- O encaminhamento de pacotes na Internet é realizado pela colaboração entre os sistemas terminais (os computadores) e os comutadores de pacotes

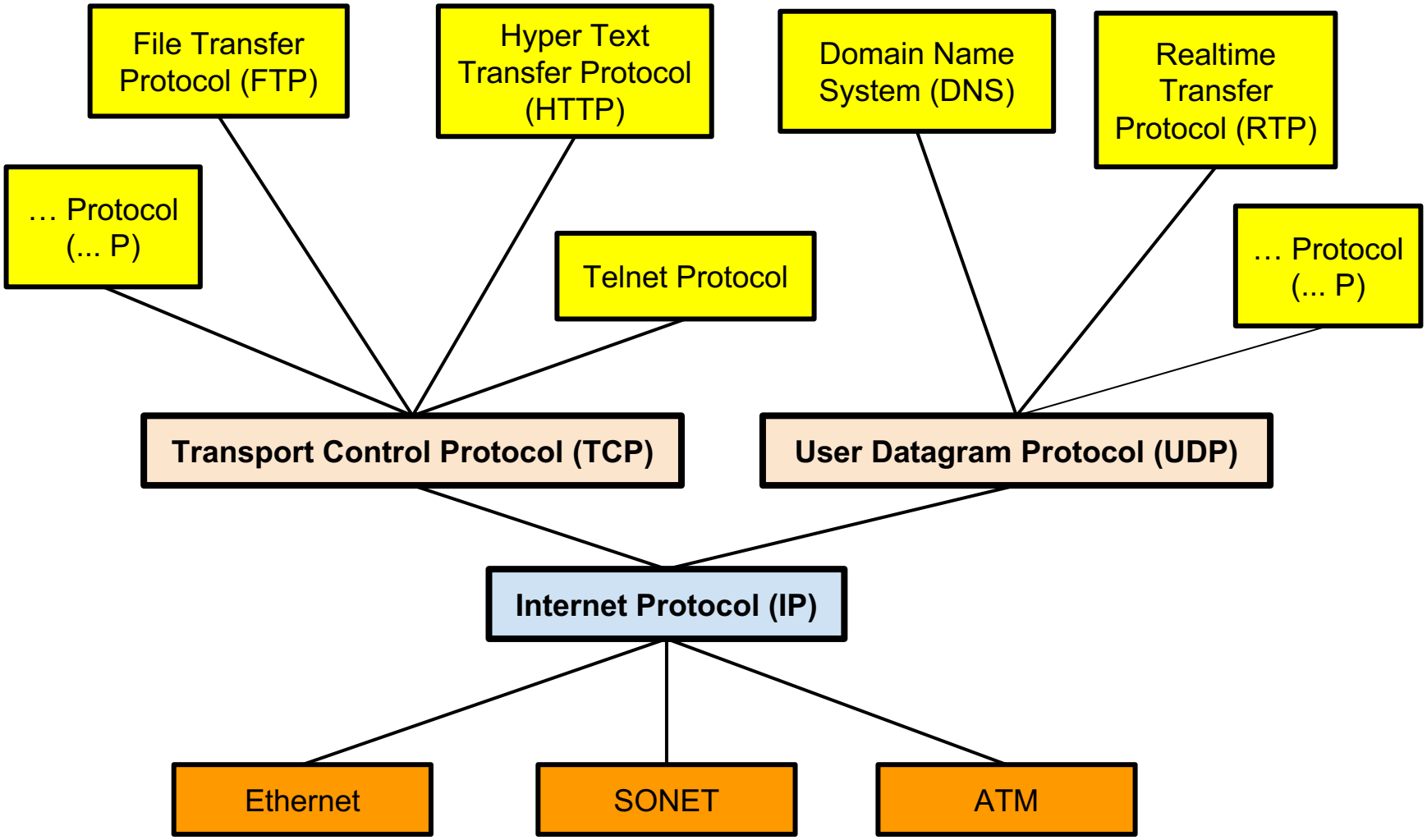
- Para que seja possível esta colaboração são necessárias um conjunto de convenções, formatos e procedimentos comuns, reunidos no protocolo IP

- Este protocolo é o principal protocolo usado no *core* da Internet e a sua descrição é o objeto desta lição.

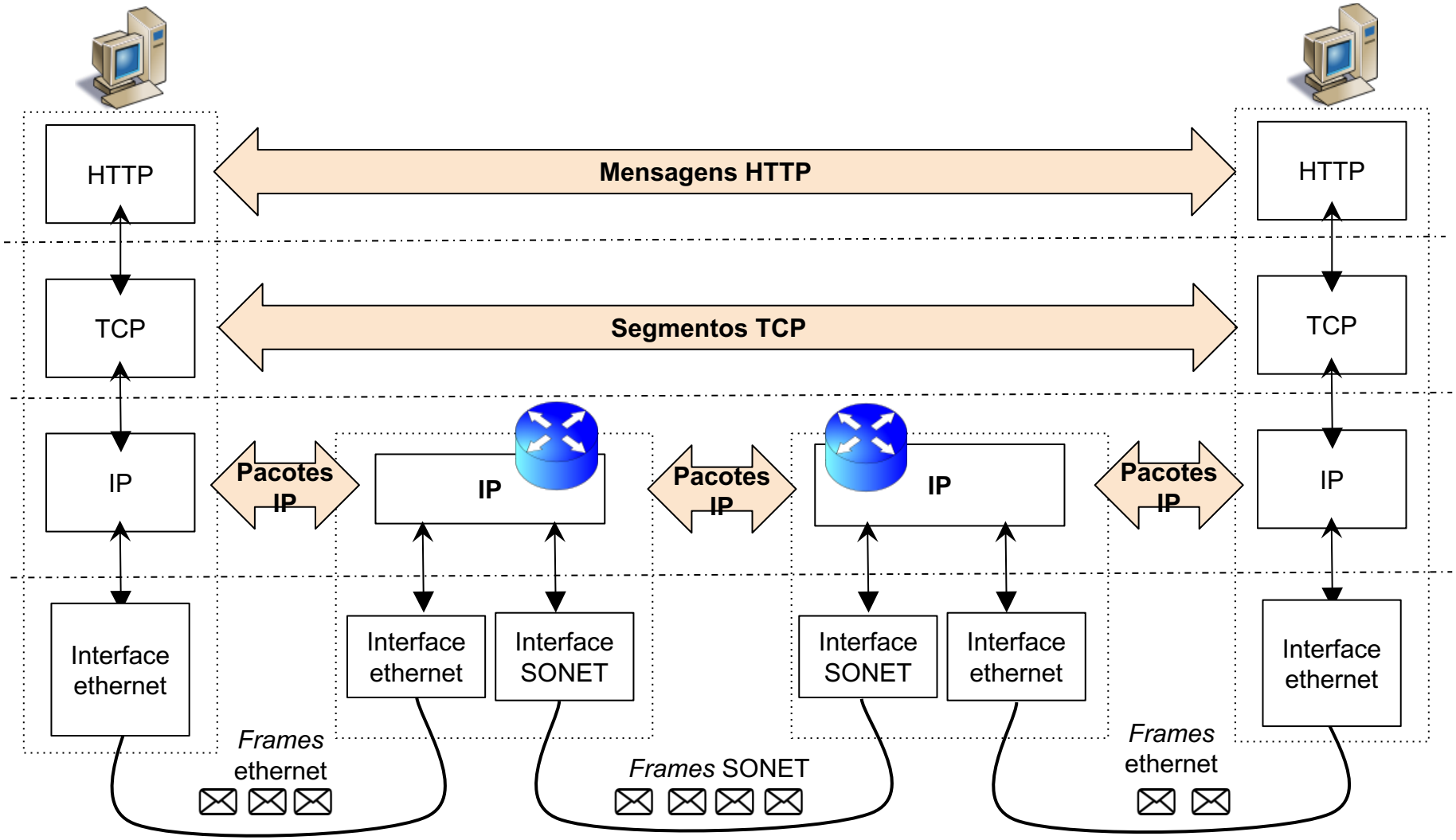
An expert is a man who made all the mistakes, which can be made, in a very narrow field.

– **Autor:** *Niels Bohr*

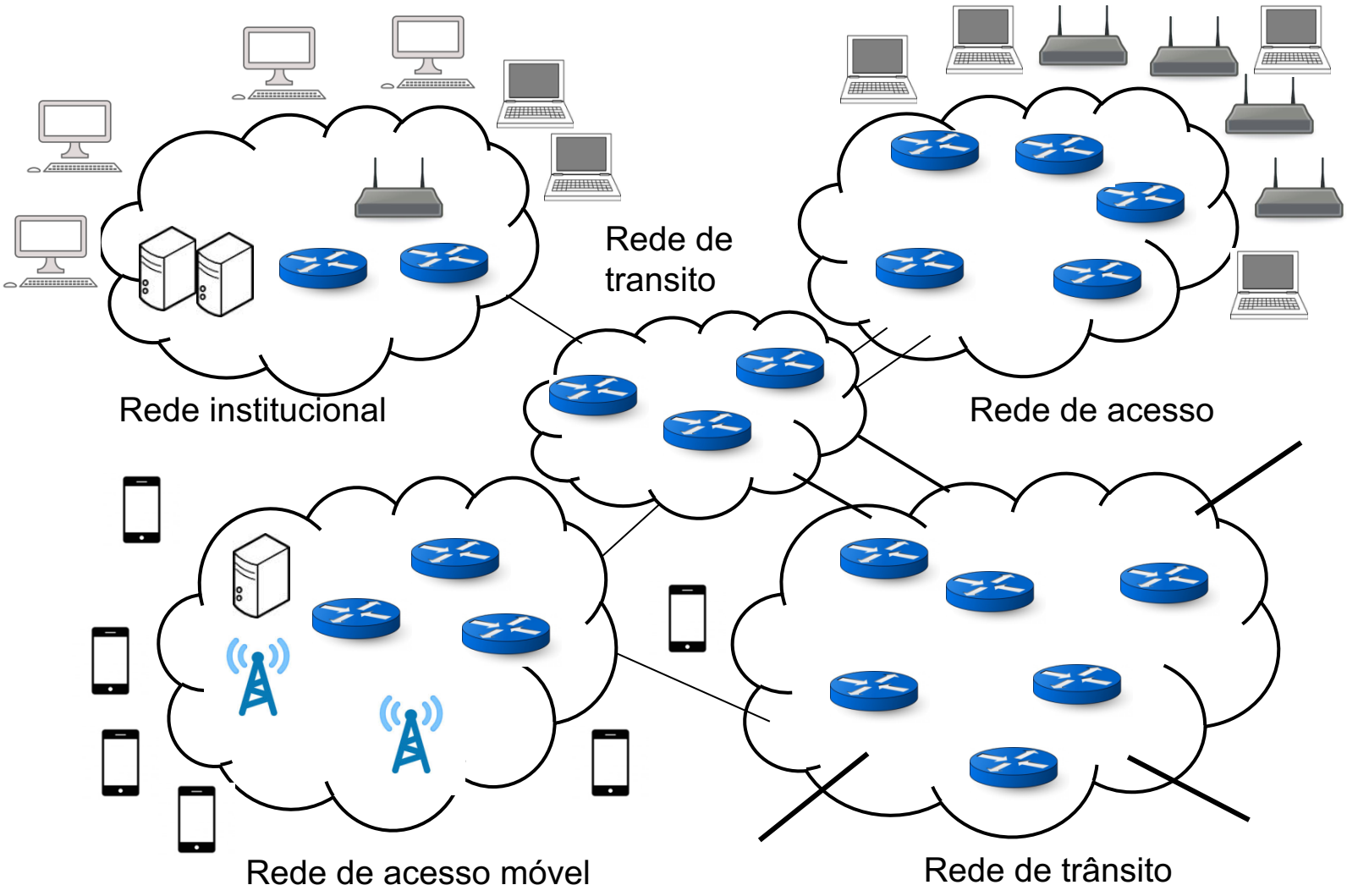
As Camadas em TCP/IP



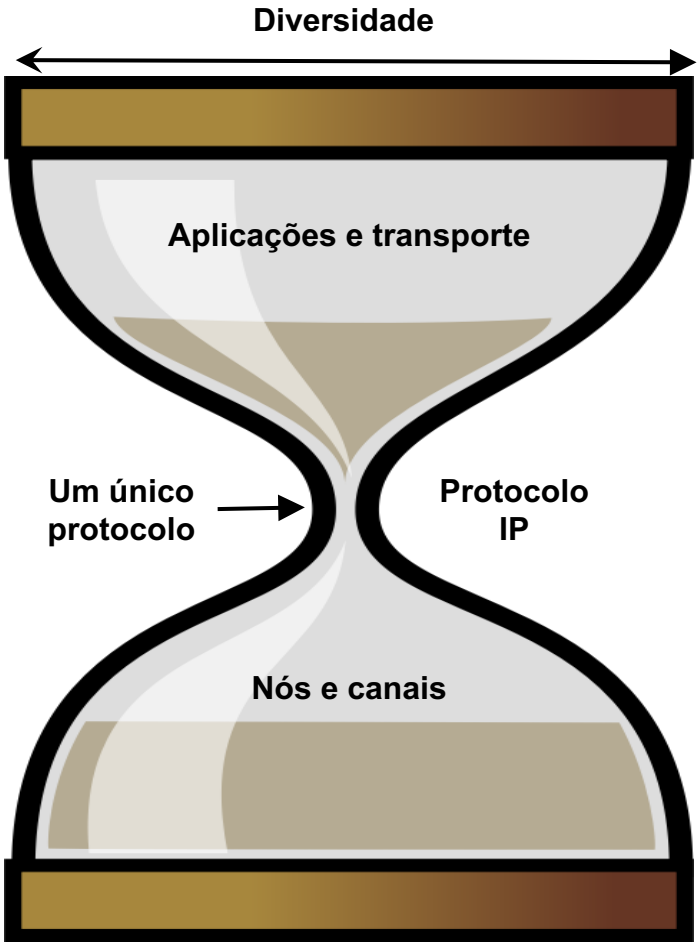
A Posição do Protocolo IP



O IP como Interface entre Redes

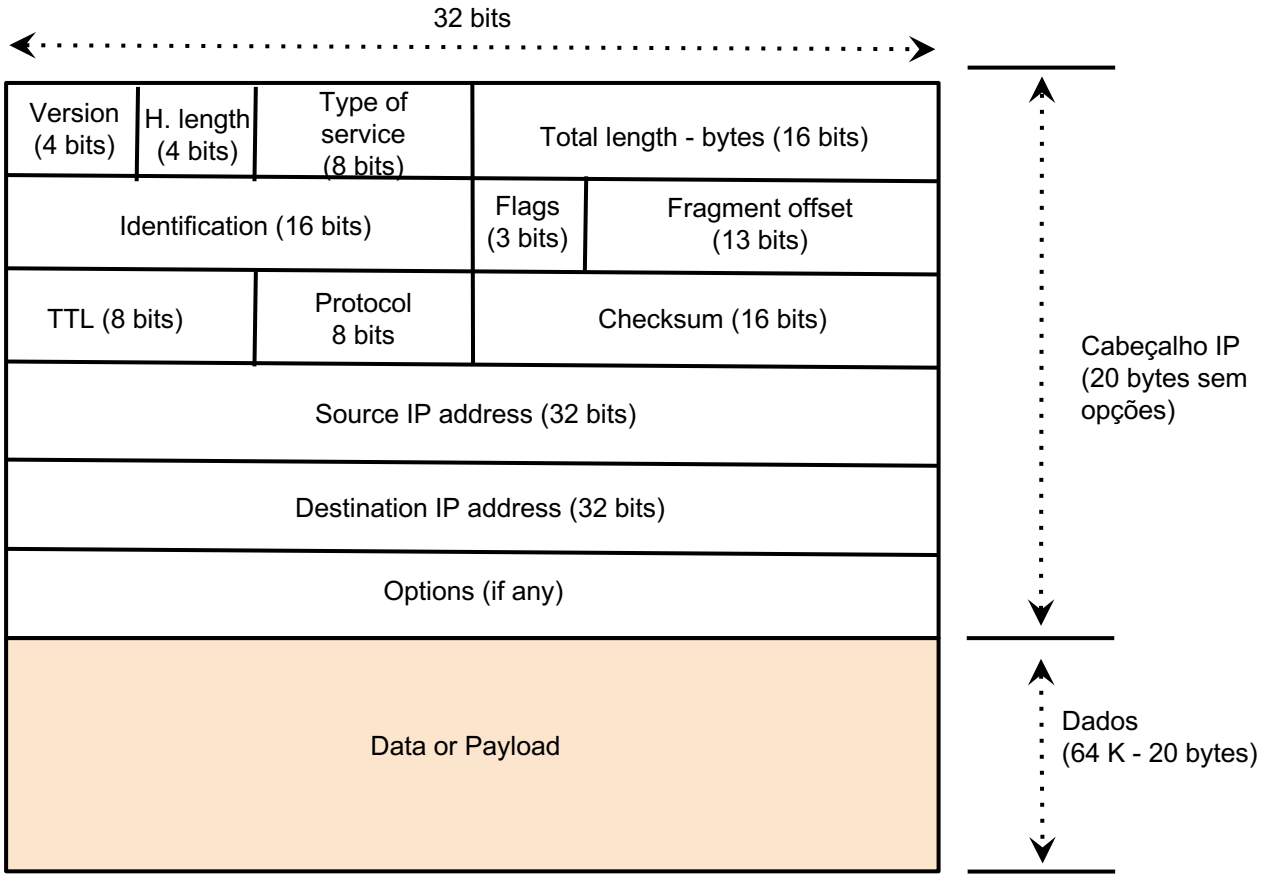


A Posição do Protocolo IP



O gargalo facilita a interoperação

Formato de um Pacote IP (v4)



Campos de Cabeçalho (1)

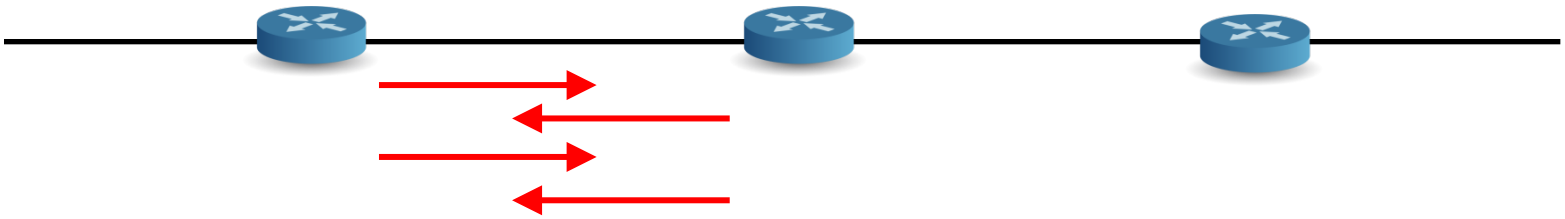
- Versão (4 bits)
  - Geralmente é a versão 4, a mudar para 6
  - É necessário para saber como interpretar o resto
- *Header length* (4 bits)
  - Número de palavras de 32 bits que formam o cabeçalho
  - geralmente vale 5 (20 bytes)
- Tipo de serviço (8 bits)
  - Prioridade a dar ao pacote (geralmente é por classes)
  - O *router* pode ignorar este campo
- *Total length* (16 bits)
  - O maior pacote pode ter 64 K bytes mas em geral usa-se um muito menor pois quase todos os canais impõem um valor mais baixo ao maior *frame* que aceitam

Campos de Cabeçalho (2)

- Fragmentação (32 bits)
  - Na versão 4 permite subdividir no interior da rede um pacote em fragmentos quando o mesmo não cabe no canal
  - O destinatário final deve recompor o pacote original
- TTL – Time to Live (8 bits)
  - Número de *routers* máximo a atravessar
  - Funciona como um mecanismo de proteção contra caminhos demasiado longos ou infinitos, introduzidos por erros
- Protocolo (8 bits)
  - Tipo do *payload* (suporta a *demultiplexagem* a nível superior)
- *Header Checksum* (16 bits)
- *Options* (variável)
  - *Source route, record route, ...*

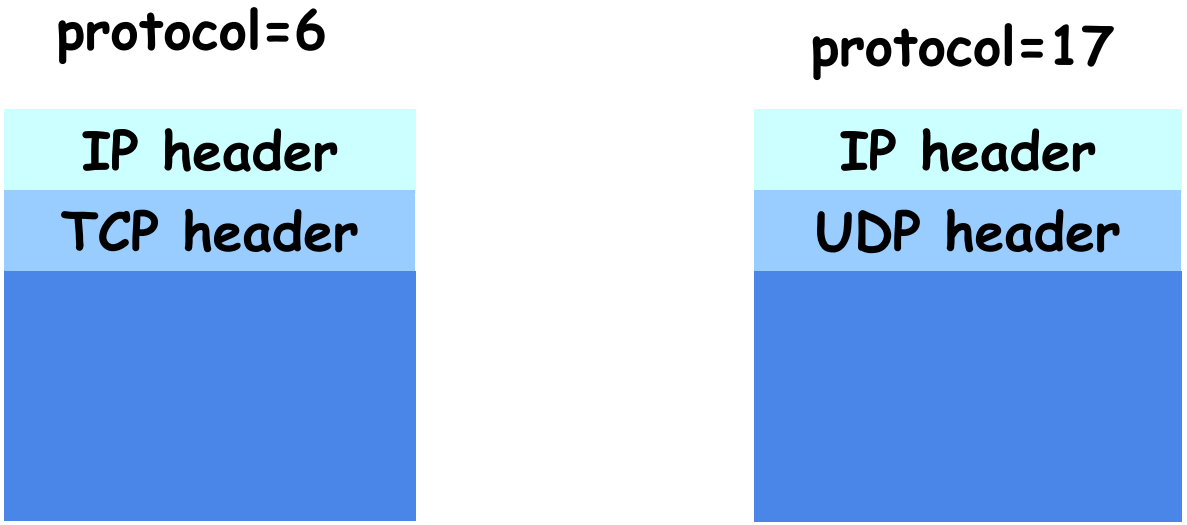
Time-to-Live (TTL)

- Mecanismo de segurança se existirem problemas
  - Ciclos de encaminhamento por erros ou instabilidade
  - Saturam completamente os canais em jogo

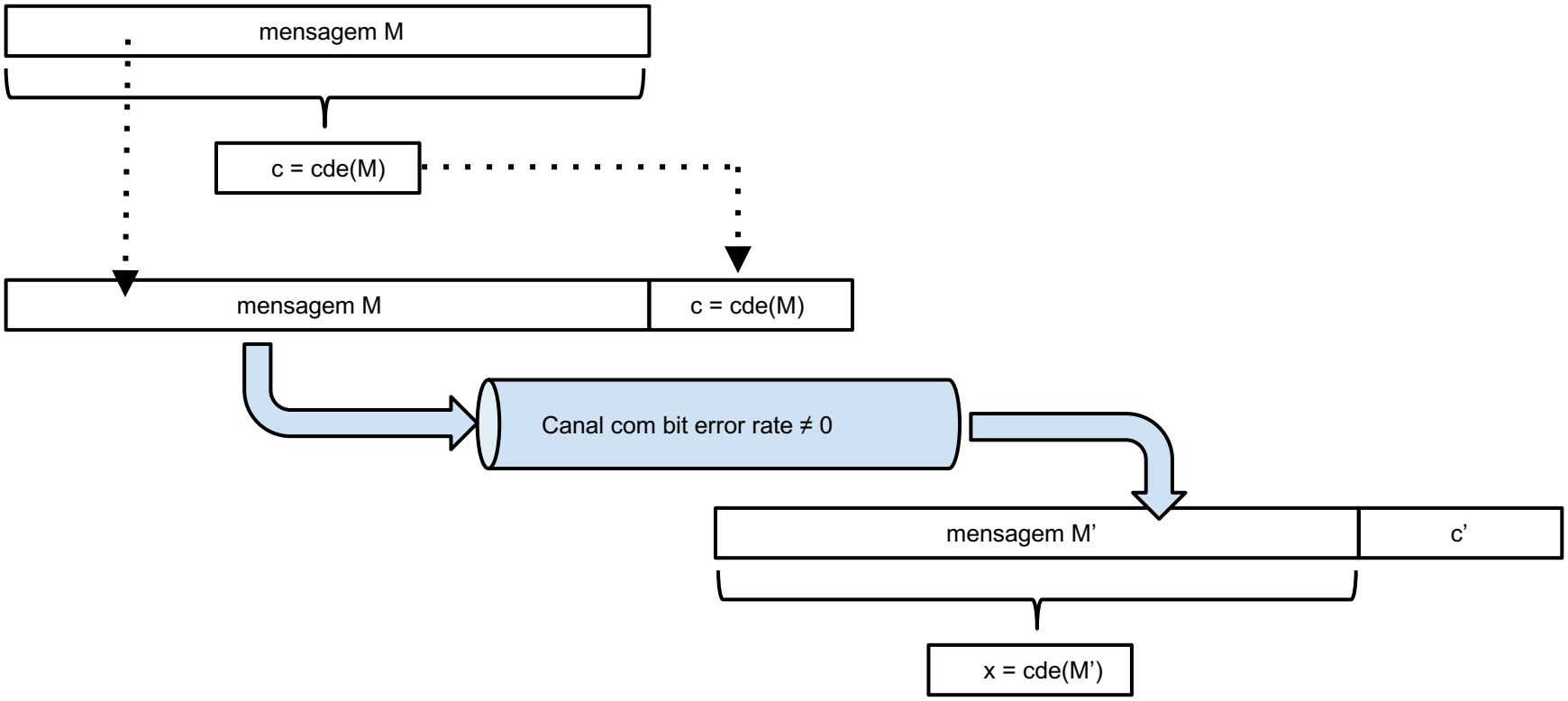


- O campo *time-to-live* do cabeçalho IP
  - O campo é decrementado sempre que o pacote chega a um nó de comutação
  - Se chega a 0 é suprimido …
  - …e uma mensagem "*time excedeed*" é enviada à origem

IP Header: Payload Protocol



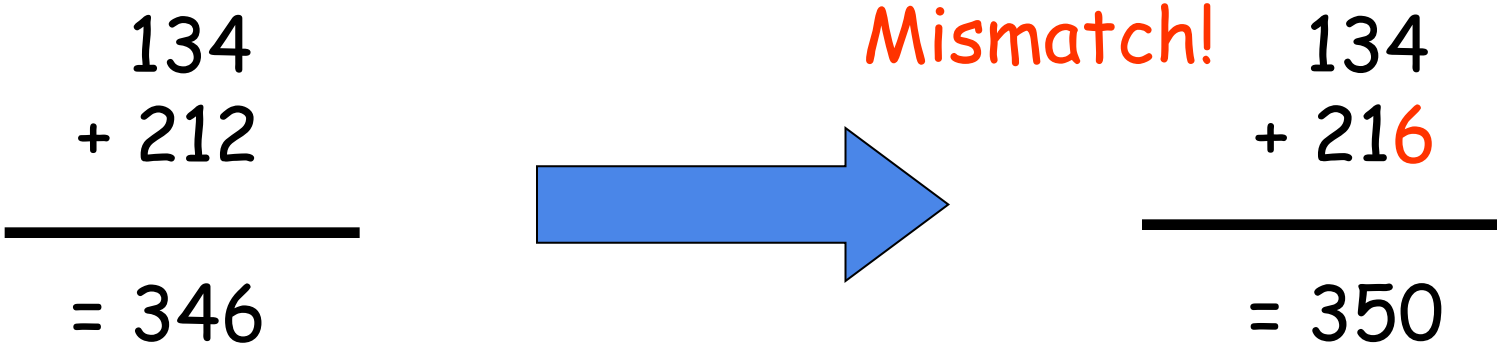
Checksum – Código de Deteção de Erros (CDE)



x = c' ?
- Sim, aceitar a mensagem (M' = M)
- Não, rejeitar a mensagem (M' ≠ M)

Header Checksum

- Checksum (16 bits)
  - Soma de todas as palavras de 16-bit do cabeçalho
  - Se o cabeçalho se corromper num canal, as somas diferem na emissão e na receção
  - O que permite não tratar pacotes com cabeçalho corrompido
  - Este mecanismo usa um algoritmo simples para poder ser implementado por software



Encaminhamento IP (*Routing* IP)

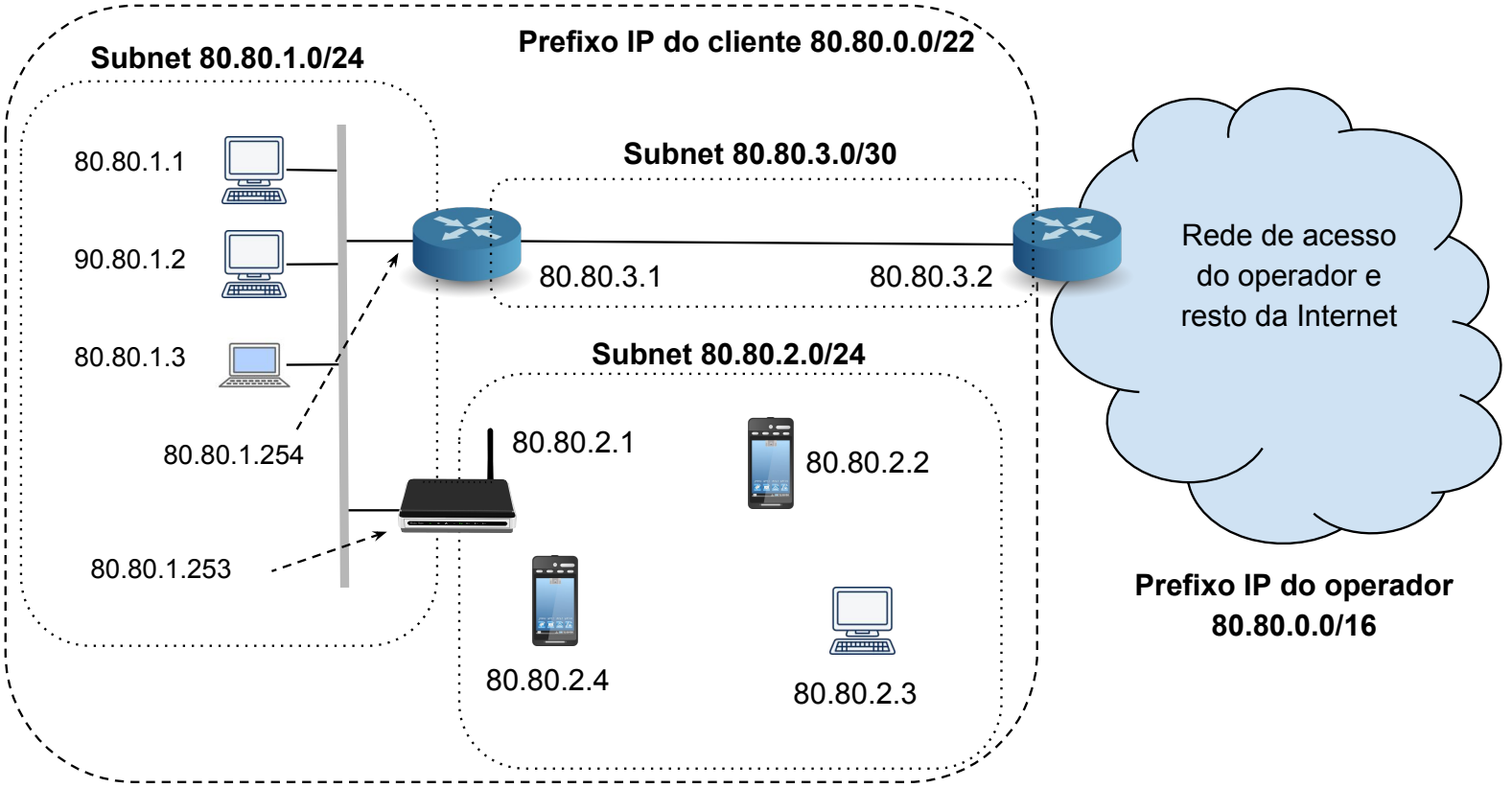
- No mundo IP os comutadores de pacotes chamam-se ***routers* IP**
- Cada computador e cada *router* tem que possuir tabelas de encaminhamento / *forwarding*
- Estas tabelas mapeiam prefixos de rede IP destino com interfaces ou endereços IP de *routers*
- O router começa por determinar a que entrada da tabela corresponde o endereço de destino do pacote (*Longest-Prefix Matching*) e depois segue as instruções da tabela

Prefixos IP e Interfaces dos *Routers*

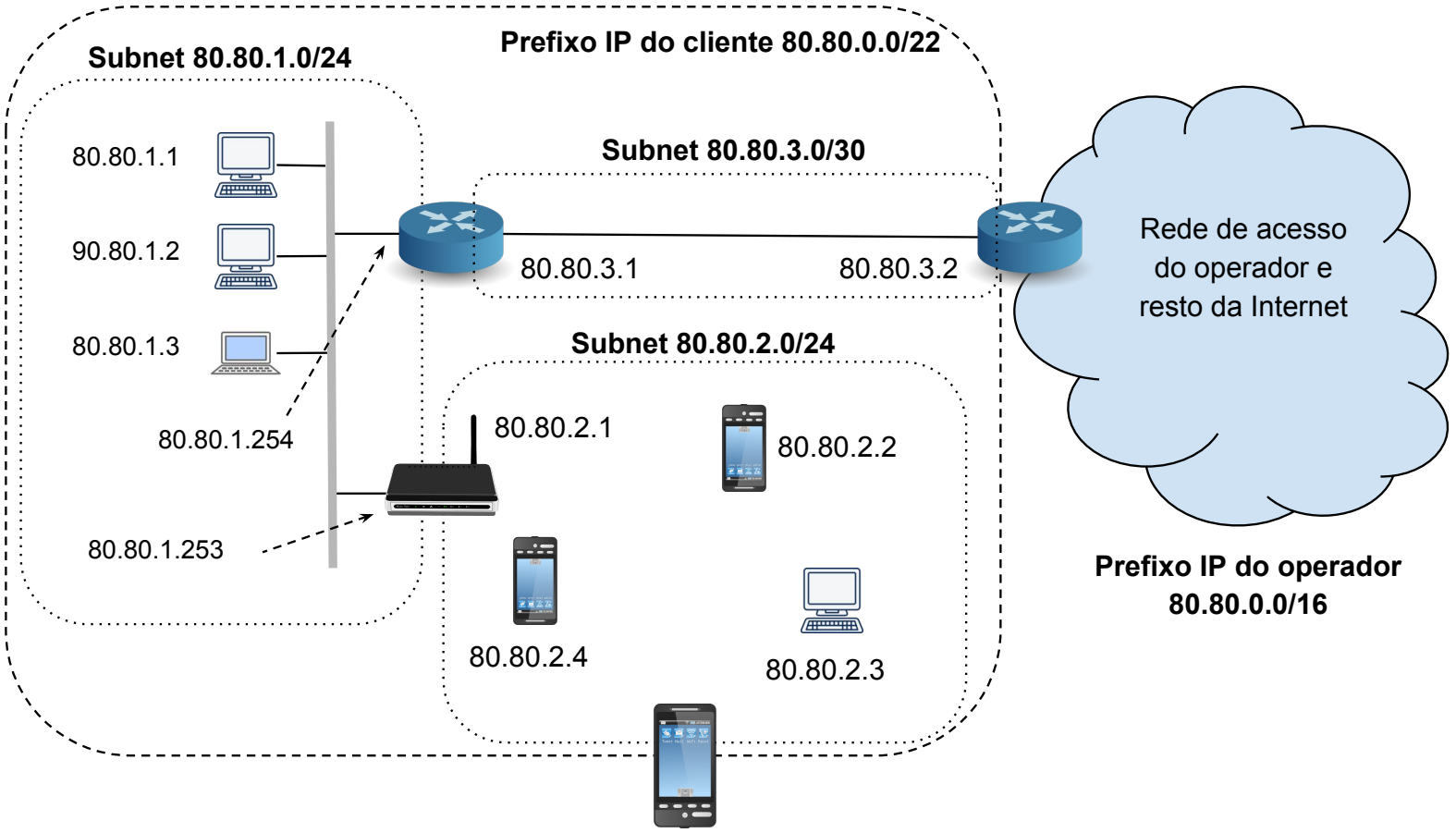
- Cada *router* tem várias interfaces e cada uma delas está ligada a um (ou mais) vizinhos — outros *routers* ou computadores

- Todos os computadores / *routers* ligados ao mesmo canal têm de ter um prefixo IP comum e cada um deles tem um endereço nesse prefixo
- Ao nível mais baixo da hierarquia, cada prefixo IP está associado a uma canal e cada canal tem um prefixo IP distinto associado (muitas vezes chamado uma *subnet*)
  - Um canal ponto a ponto tem um prefixo partilhado por duas interfaces (2 endereços distintos)
  - Um canal em difusão tem um prefixo IP partilhado pelas N interfaces a ele ligados (N endereços IP)

Canais e Prefixos IP



Como obter um Endereço IP?



Qual o meu endereço IP? Que servidor DNS usar ? Como chego à Internet? Resposta: use DHCP RFC 2131 (IPv4) ou 3315 (IPv6)

DHCP - Dynamic Host Configuration Protocol

Cliente novo sem endereço IP

DHCP discover (broadcast)

DHCP offer

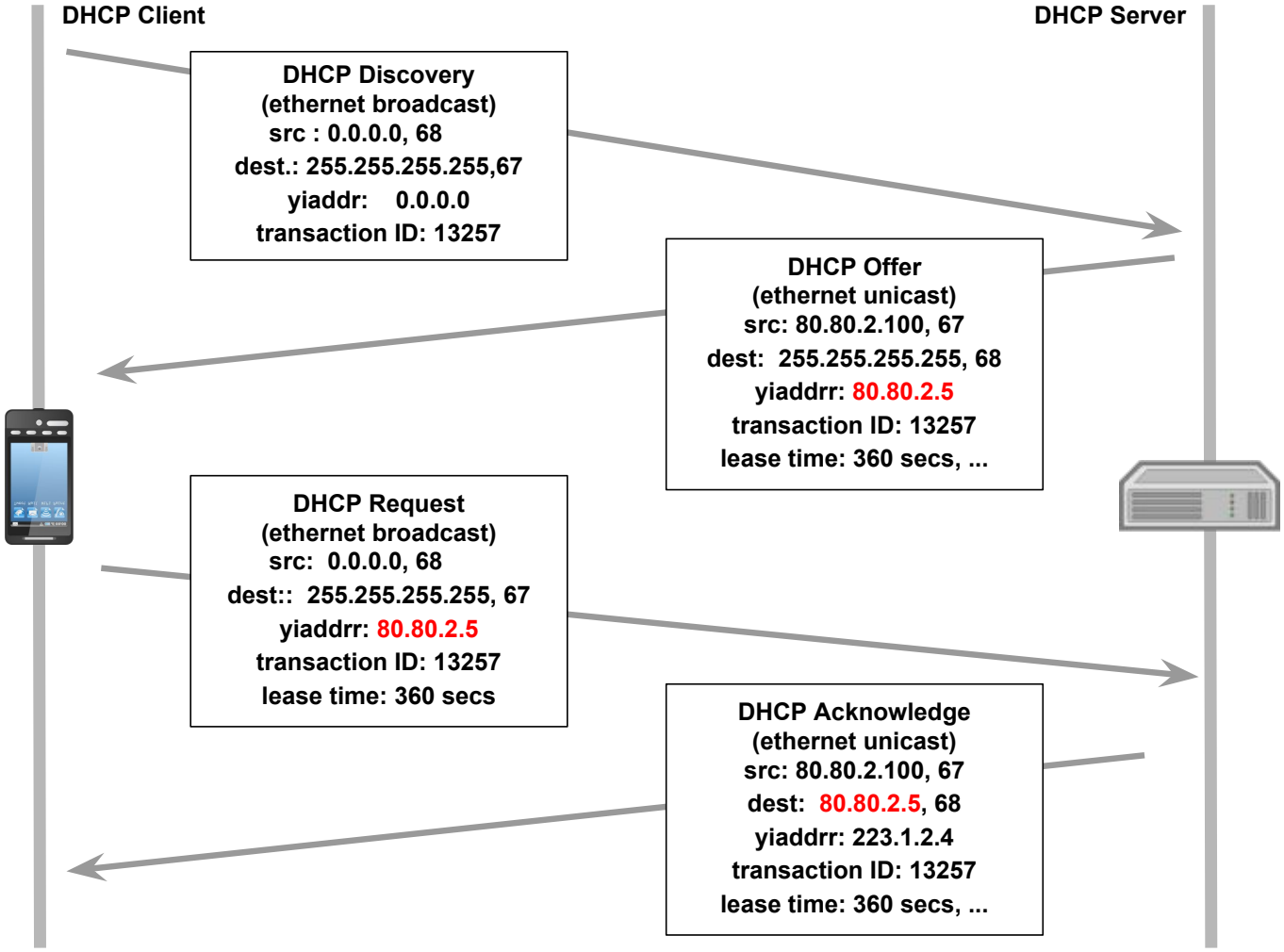
DHCP request (broadcast)

DHCP ACK

UDP port 67 in server, 68 in client

DHCP server at 80.80.2.100

O DHCP usa datagramas UDP e *broadcast*



Resposta de um servidor DHCP

- DHCP "*offer message*"
  - Parâmetros de configuração (*proposed IP address, mask, default router address, DNS server, ...*)
  - *Lease time* (o tempo durante o qual esta informação é válida)
- A resposta pode vir de mais do que um servidor
  - Protege contra um *crash* de um servidor único
  - Os vários servidores respondem com uma oferta
  - O cliente decide qual deve aceitar
- Aceitação de uma das ofertas
  - O cliente envia uma mensagem DHCP com os parâmetros aceites
  - O servidor confirma com um ACK
  - ... e os outros servidores verificam que não foram escolhidos

Voltemos à nossa Rede

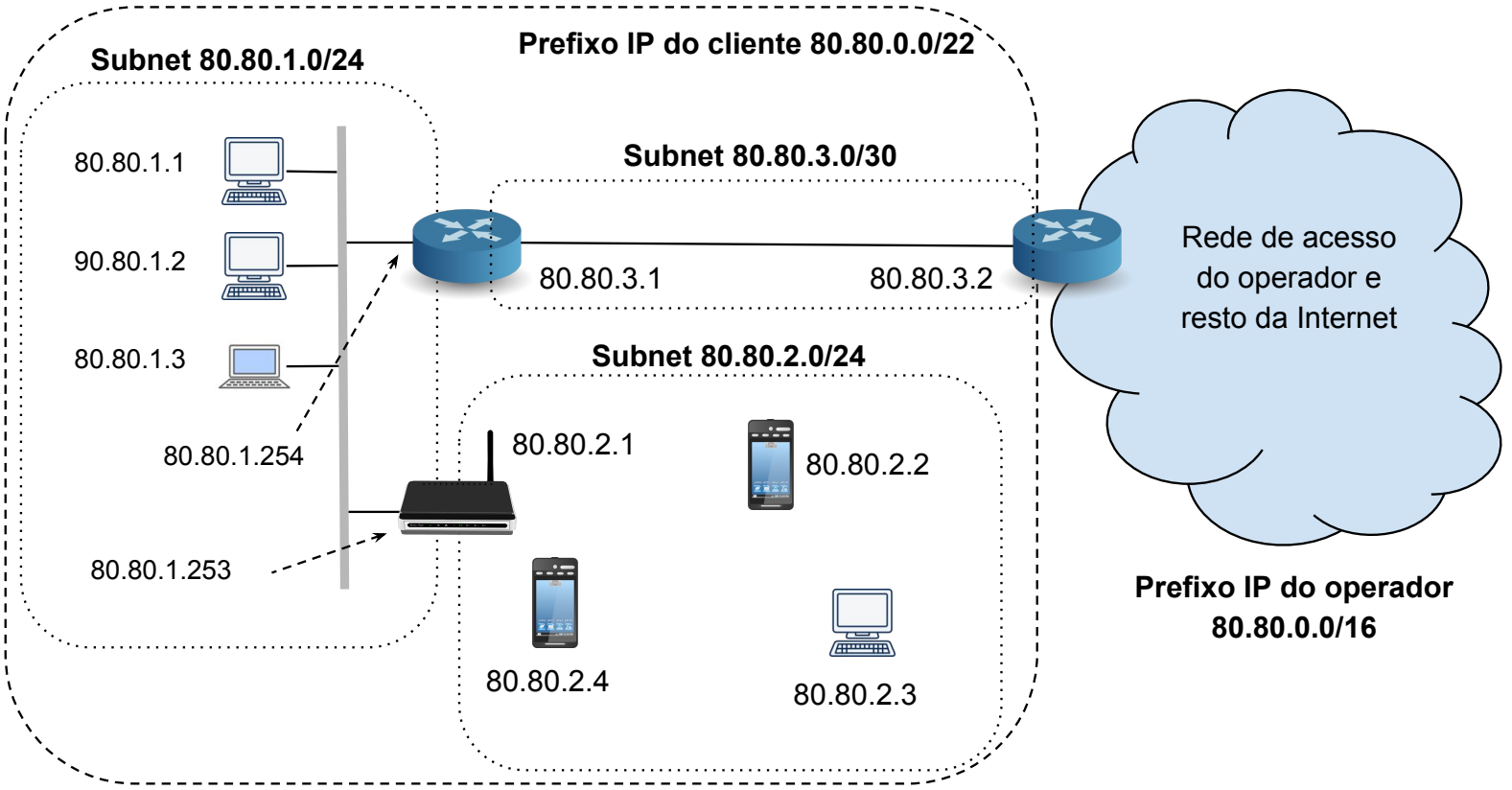


Tabela de Encaminhamento

Tabela de encaminhamento ou *forwarding table* do computador com endereço 80.80.1.3

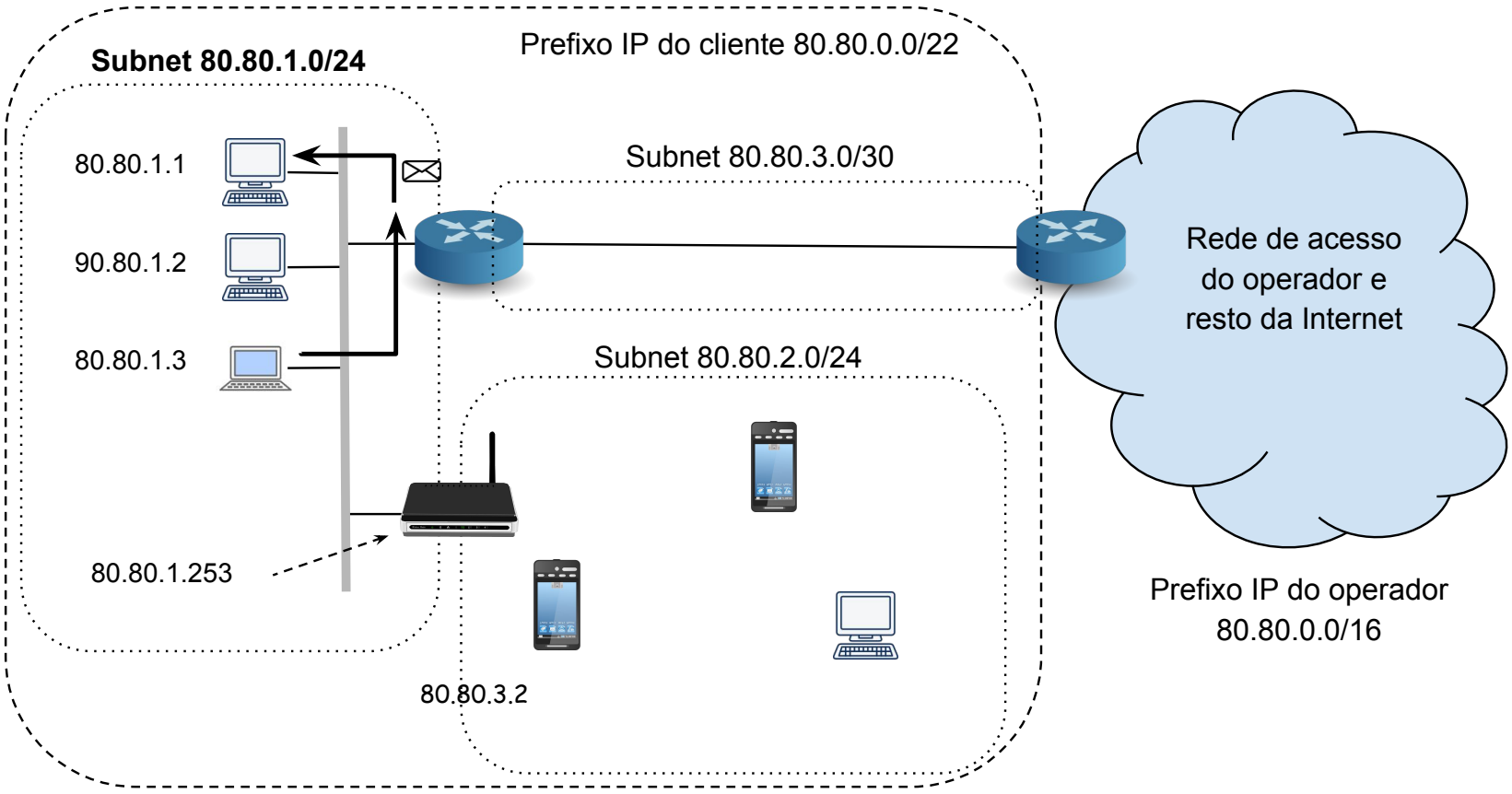
Rede ou Prefixo	Tipo de encaminhamento	Endereço do Gateway	Interface	Métrica
80.80.1.3/32	Endereço local	80.80.1.3/32	eth0	0
80.80.1.0/24	Direto	80.80.1.3/32	eth0	0
80.80.2.0/24	Indireto	80.80.1.253	eth0	1
80.80.3.0/30	Indireto	80.80.1.254	eth0	1
0.0.0.0/0	Indireto	80.80.1.254	eth0	1

Pode consultar a do seu computador através dos comandos "netstat -r" ou "route"

Encaminhamento Direto

- Se a interface com o endereço IP de destino está ligada a um canal a que o computador ou o *router* estão diretamente ligados
  - O destino está numa *subnet* a que o computador ou o *router* estão ligados
  - O encaminhamento diz-se direto
  - Tal reconhece-se pois o prefixo é o prefixo de uma *subnet* diretamente ligada
  - Se o canal for multi-ponto (tipicamente um canal baseado em *broadcast*) usa-se o protocolo ARP (ver adiante) para conhecer o endereço de nível MAC no canal do destino

Encaminhamento Direto



	origem	destino	dados
Pacote IP:	80.80.1.3	80.80.1.1	bla bla bla ….

Tabela de Encaminhamento

O computador com endereço o 80.80.1.3 envia um pacote IP para o computador com o endereço 80.80.1.1

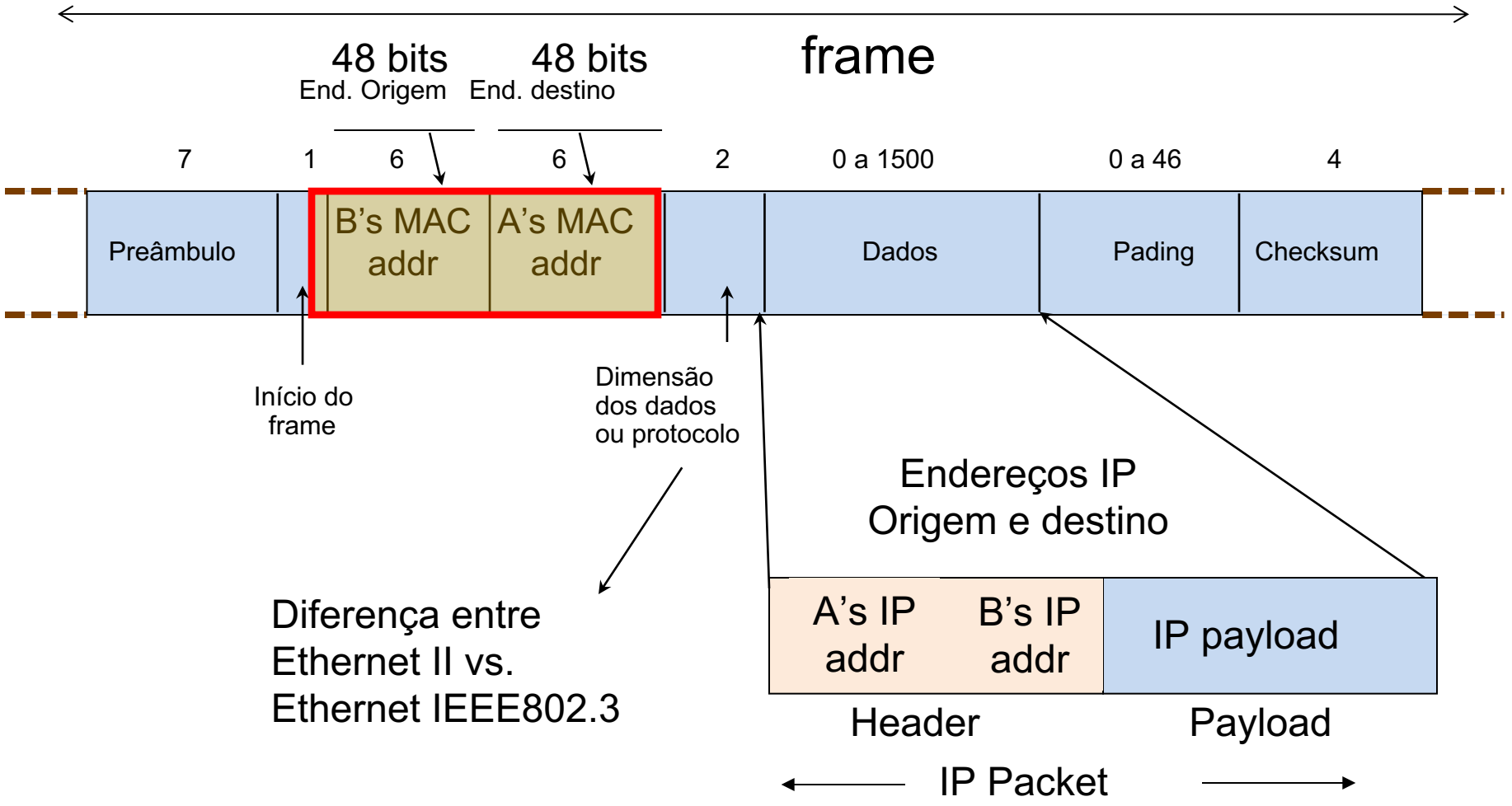
Tabela do computador com endereço 80.80.3.1

Rede ou Prefixo	Tipo de encaminhamento	Endereço do Gateway	Interface	Métrica
80.80.1.3/32	Local	80.80.1.3/32	eth0	0
80.80.1.0/24	Direto	80.80.1.3/32	eth0	0
80.80.2.0/24	Indireto	80.80.1.253	eth0	1
80.80.3.0/30	Indireto	80.80.1.254	eth0	1
0.0.0.0/0	Indireto	80.80.1.254	eth0	1

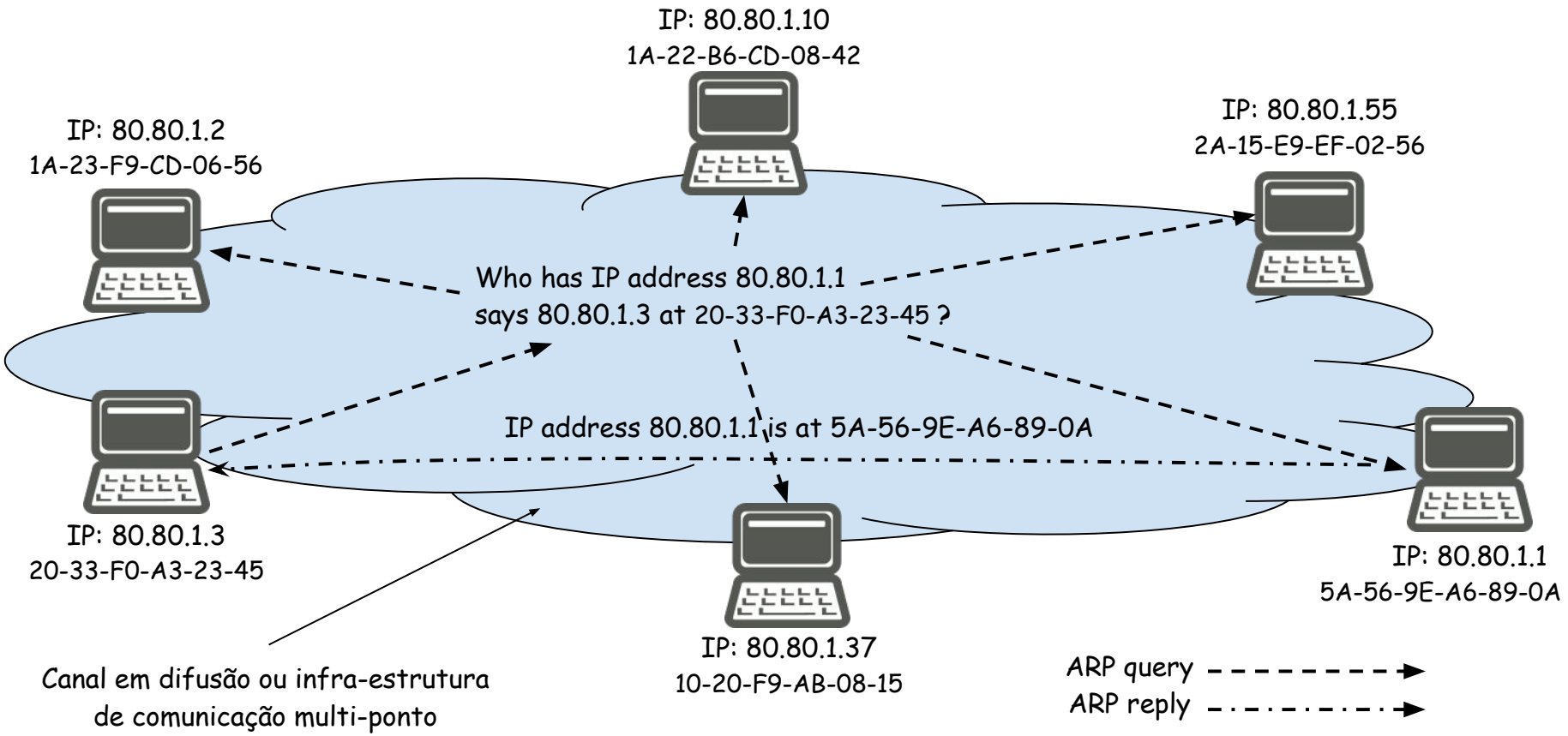
Longest prefix matching entry

other prefix matching entry

Frames Ethernet e pacotes IP



Protocolo ARP (Address Resolution Protocol)



Address Resolution Protocol (ARP) Table

- Cada nó tem uma tabela ARP
  - Com pares (IP address, MAC address)
- E consulta a tabela antes de enviar um pacote
  - Se encontrar o endereço MAC correspondente ao endereço IP de destino
  - Encapsula o pacote IP num *frame* e envia-o
- E se o endereço IP não está na tabela ARP?
  - O pacote IP é suprimido e
  - O emissor envia um broadcast: "Who has IP address 80.80.1.1?"
  - O receptor responde: "IP address is at 5A-56-9E-A6-89-0A"
  - O emissor coloca esses dados na tabela ARP
- Como consultar a tabela ARP
  - Dar o comando "arp –a" na maioria dos sistemas

Encaminhamento Indireto

- Ocorre quando o destino do pacote não se encontra diretamente ligado ao computador ou ao *router*
  - O endereço IP de destino não pertence a nenhum dos prefixos das *subnets* que estão diretamente ligados ao computador ou ao *router*
  - O pacote tem de ser entregue a um *router* (acessível por encaminhamento direto) que o aproxime do destino
  - O endereço IP desse *router* vizinho tem de ser conhecido e o mesmo tem de ser diretamente alcançável por uma das nossas interfaces

Tabela de Encaminhamento

O computador com o endereço o 80.80.1.3 envia um pacote IP para o computador com o endereço 80.80.2.3

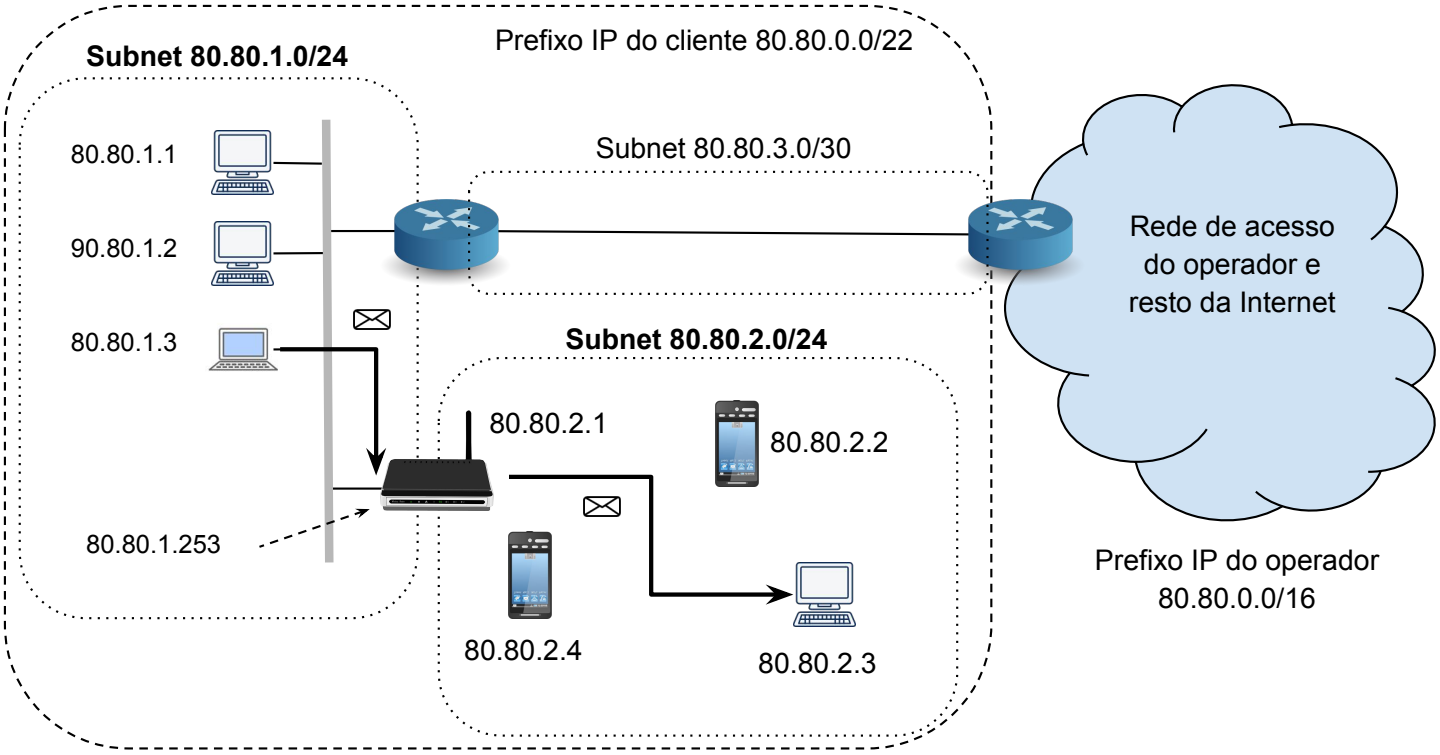
Tabela do computador com endereço 80.80.1.3

Rede ou Prefixo	Tipo de encaminhamento	Endereço do Gateway	Interface	Métrica
80.80.1.3/32	Local	80.80.1.3/32	eth0	0
80.80.1.0/24	Direto ou local	80.80.1.3/32	eth0	0
80.80.2.0/24	Indireto	80.80.1.253	eth0	1
80.80.3.0/30	Indireto	80.80.1.254	eth0	1
0.0.0.0/0	Indireto	80.80.1.254	eth0	1

Longest prefix matching entry

other prefix matching entry

Encaminhamento Indireto



	origem	destino	dados
Pacote IP:	80.80.3.1	80.80.2.3	bla bla bla ….

Encaminhamento por Omissão (Defeito)

- Ocorre quando o destino do pacote só faz *matching* com o endereço por defeito, geralmente denotado pelo prefixo IP de comprimento 0 ou 0.0.0.0/0
- O destino não se encontra diretamente ligado ao computador ou ao *router*
  - Trata-se de uma espécie de "otherwise" ou "else" final
  - O pacote tem de ser entregue a um *router* diretamente ligado que saiba aproximar-se de qualquer destino
  - O endereço IP desse *router* vizinho, dito *default router*, tem de ser conhecido

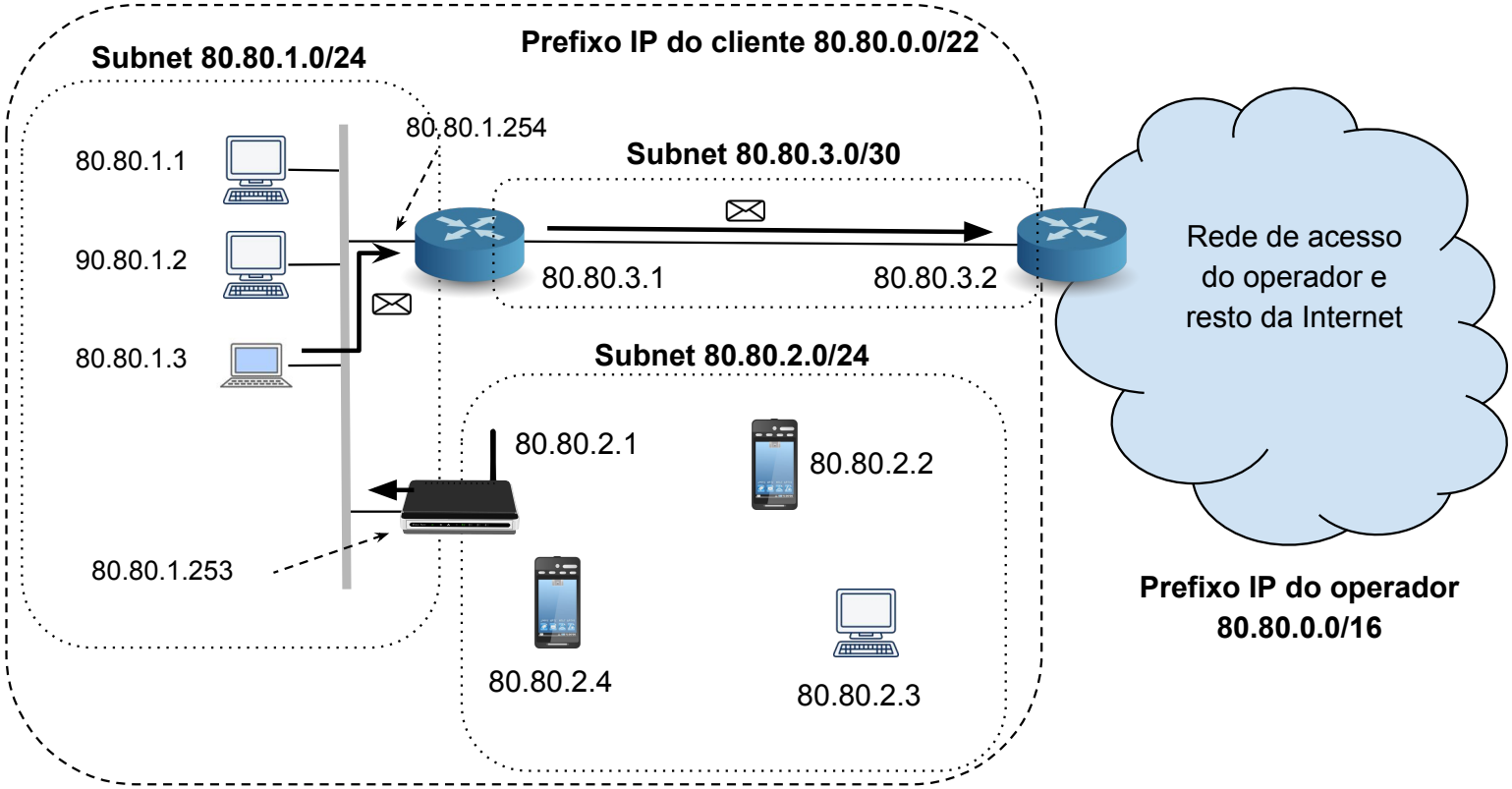
Tabela de Encaminhamento

O computador com endereço o 80.80.1.3 envia um pacote IP para o computador com o endereço 193.136.120.43

Tabela do computador com endereço 80.80.3.1

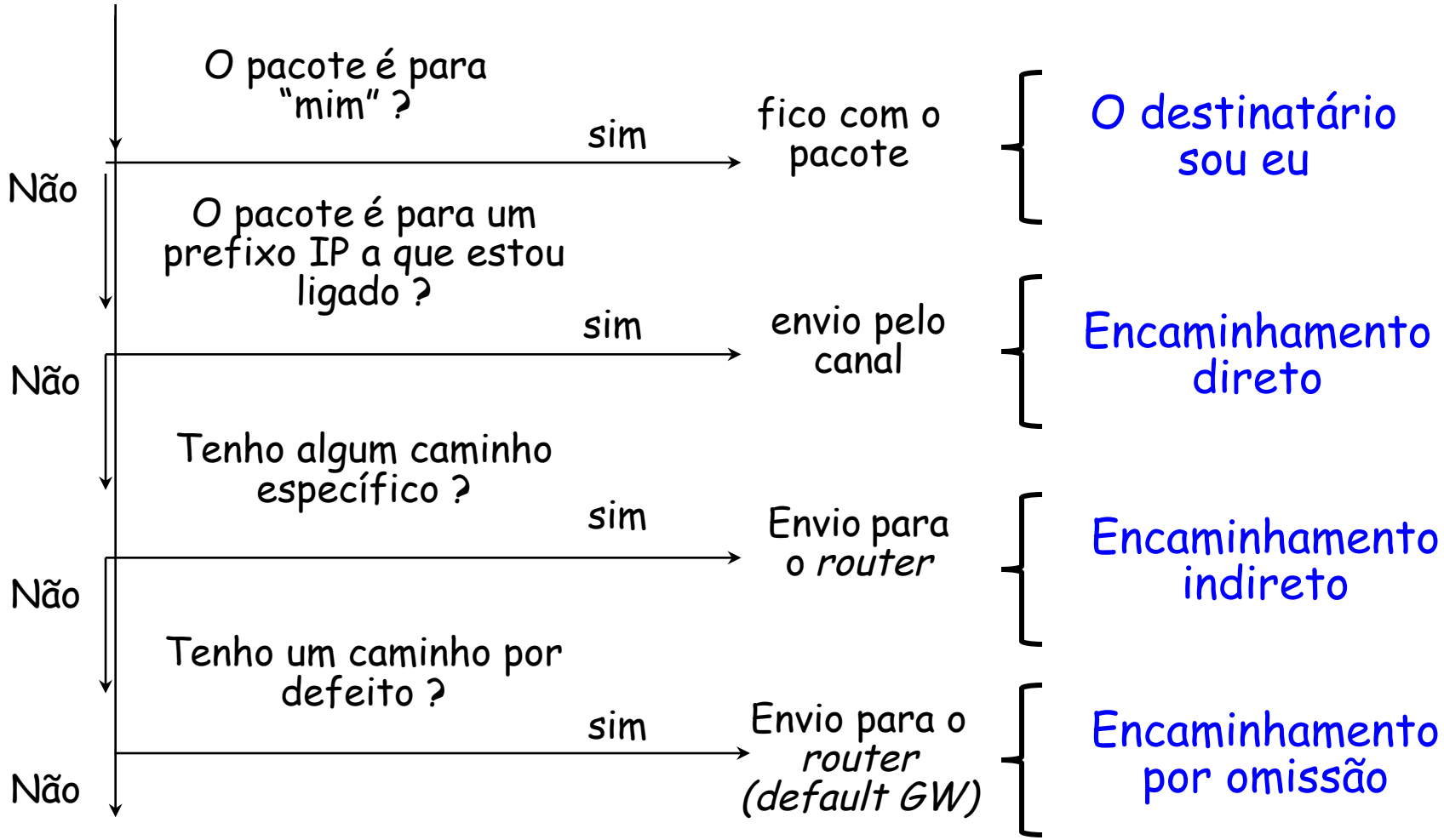
Rede ou Prefixo	Tipo de encaminhamento	Endereço do Gateway	Interface	Métrica
80.80.1.3/32	Local	80.80.1.3/32	eth0	0
80.80.1.0/24	Direto ou local	80.80.1.3/32	eth0	0
80.80.2.0/24	Indireto	80.80.1.253	eth0	1
80.80.3.0/30	Indireto	80.80.1.254	eth0	1
0.0.0.0/0	Indireto	80.80.1.254	eth0	1

Encaminhamento por Omissão



	origem	destino	dados
Pacote IP:	80.80.3.1	193.136.120.1	bla bla bla ….

Tratamento de um Pacote



Não encaminho: descarto o pacote

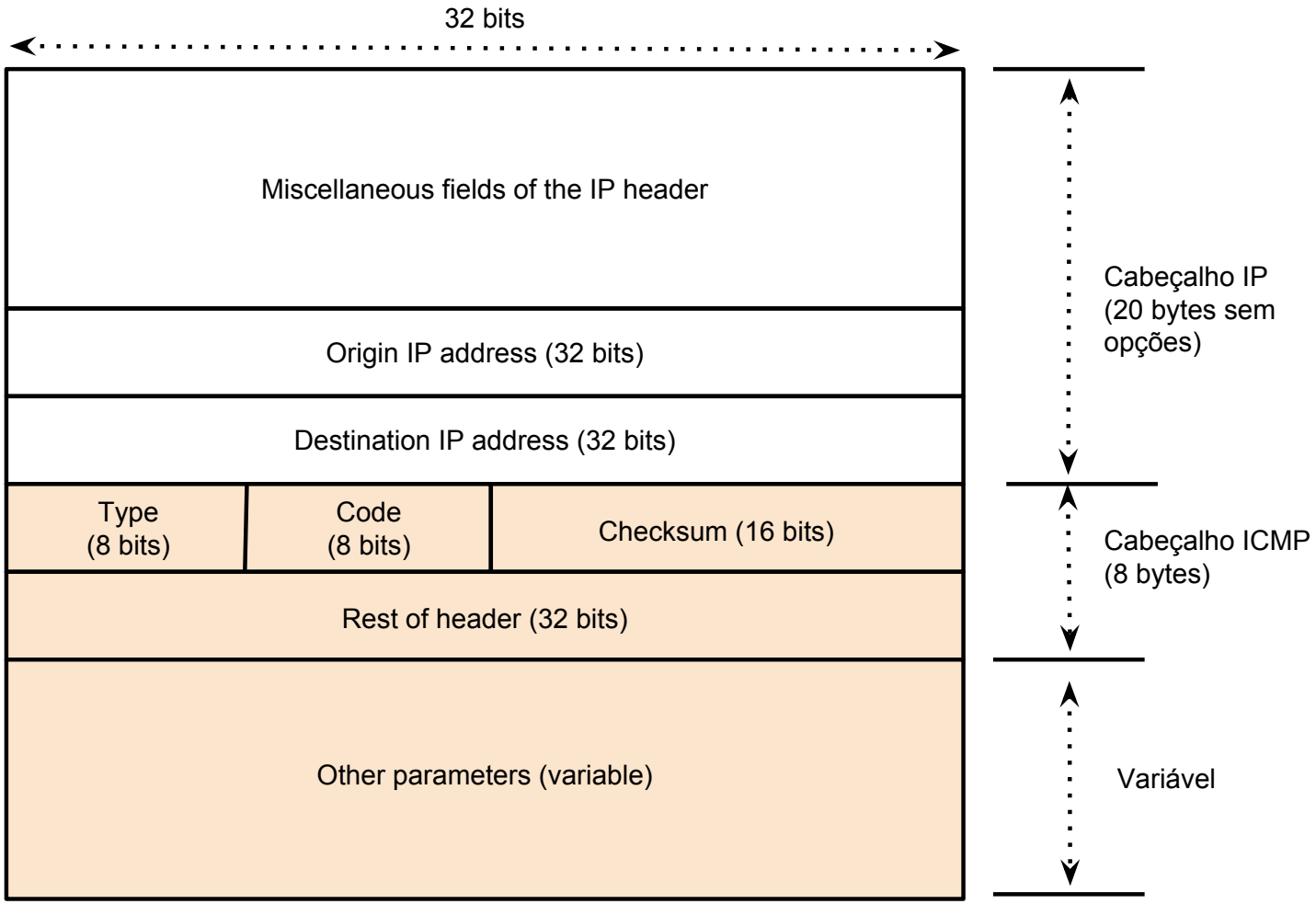
Ideias Base de ARP e DHCP

- *Broadcasting:* "quando tiver dúvidas pergunte a todos" - o broadcasting substituí uma directoria
  - Enviar por *broadcast* para todos os computadores da rede local
  - … mas só quando não se sabe já o que se pretende
- *Caching:* "guarde o que aprendeu por algum tempo"
  - Guardar o que se aprendeu para não repetir o processo
  - Lembrar o endereço e informação sobre os outros computadores (IP address + ARP cache)
- *Soft state:* … "mas mais tarde ou mais cedo esquecer o passado (… e perguntar de novo) pois a informação pode-se ir desactualizando com o tempo"
  - Associar um *time-to-live (TTL)* à informação *cached*
  - … refrescar ou suprimir a informação
  - fundamental "para se adaptar" a modificações inesperadas

Protocolo ICMP

- Quando ocorre um erro no tratamento de um pacote IP, a definição do protocolo IP admite que um computador ou um comutador possam pura e simplesmente descartar o pacote. Exemplos:
  - TTL demasiado curto
  - Incapacidade de encaminhar o pacote por ausência de entrada na tabela de encaminhamento
  - Sistema desconhecido na sub-net de destino (não há resposta ao ARP)
  - Porta desconhecida ou inativa no destino final
  - (e problemas de *checksum*?)
- No entanto, caso consigam, pode ser útil avisar o sistema que está na origem do erro que este ocorreu
- Para este efeito o protocolo IP é complementado pelo protocolo ICMP (Internet Control Message Protocol)

Pacotes ICMP (IP + …)

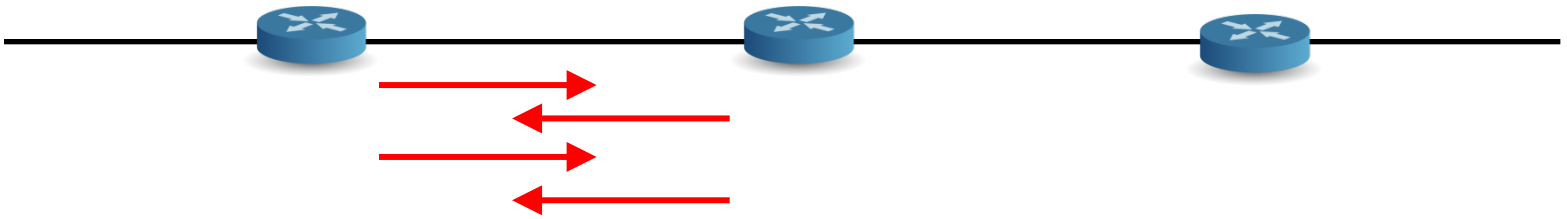


Exemplos de Mensagens ICMP

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Exemplo: *Time-to-Live* (TTL)

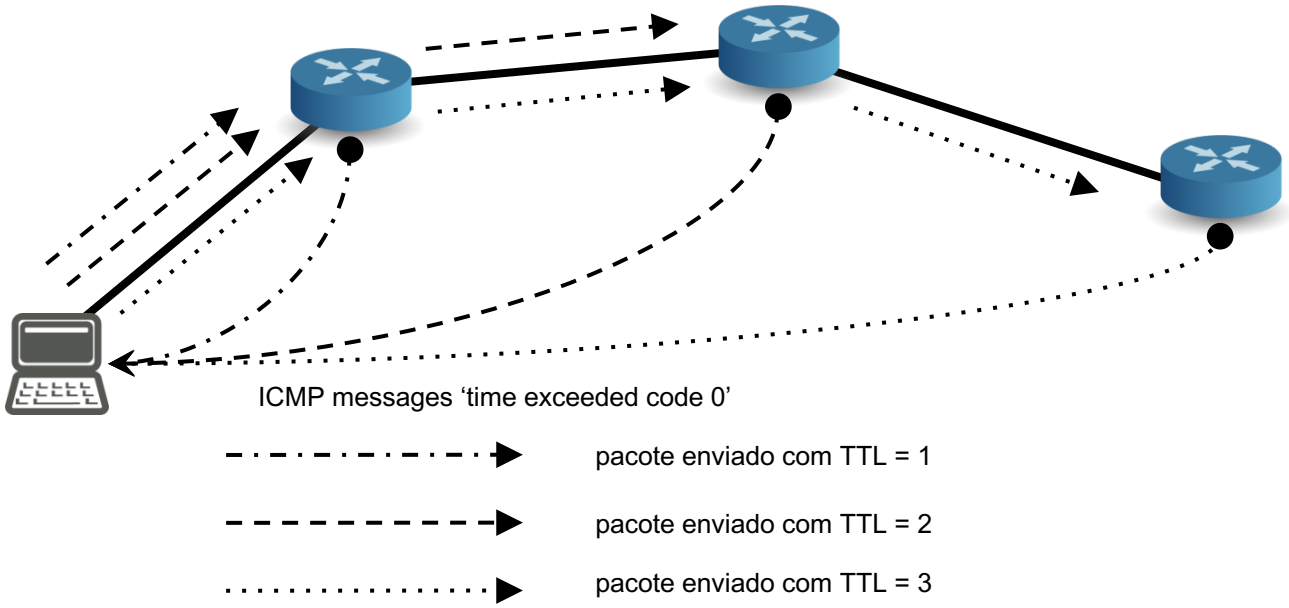
- Mecanismo de segurança se existirem problemas
  - Ciclos de encaminhamento por erros ou instabilidade
  - Saturam completamente os canais em jogo



- O campo *time-to-live* do cabeçalho IP
  - O campo é decrementado sempre que o pacote chega a um nó de comutação
  - Se chega a 0 é suprimido …
  - …e uma mensagem "*time excedeed*" pode ser enviada à origem

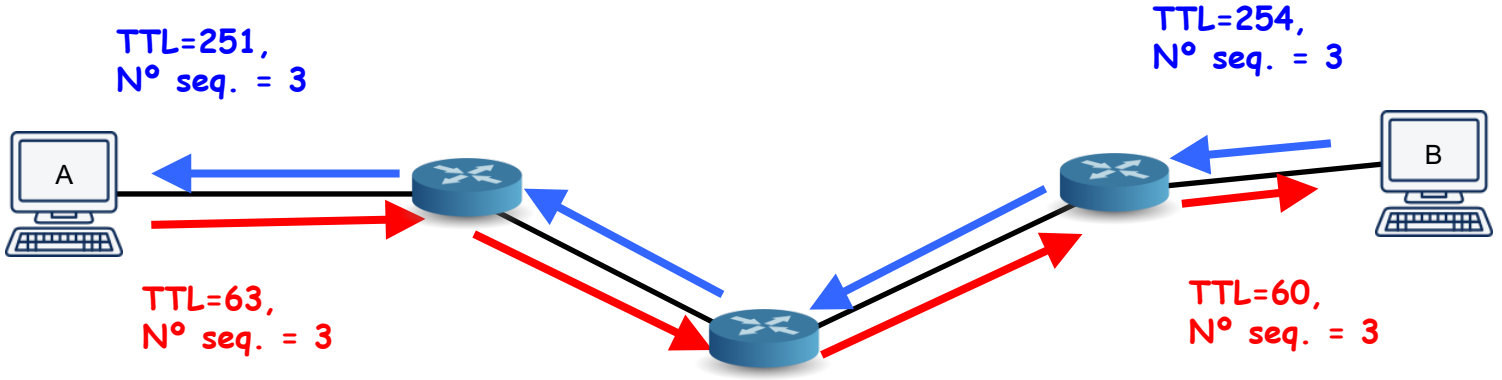
Exemplo: Uso do TTL pelo Traceroute

- Utilização do campo TTL
  - A origem envia um pacote com TTL de *n*
  - Cada nó decrementa o valor do TTL
  - Se chega a 0 envia uma mensagem "TTL *exceeded*"
- O programa *traceroute* explora esta faceta



Exemplo: Programa Ping

- A envia periodicamente pacotes ICMP a B
  - Cada vez que envia um pacote regista o valor do relógio
  - Cada pacote tem um n.º de sequência crescente
- Cada vez que B recebe um pacote responde a A
  - Com o n.º de sequência recebido
- Cada vez que A recebe um pacote de B
  - Calcula o tempo de trânsito *end-to-end* (RTT – *Round Trip Time*)



IP Version 6 (IPv6)

- Motivação inicial – aumentar o espaço de endereçamento
- Motivações adicionais – simplificar o protocolo IP no que fosse possível
  - Melhorar o tempo de processamento do cabeçalho (e.g. sem checksum)
  - Dar relevo à qualidade de serviço (objectivo falhado)
  - Tornar a implementação das opções de Mobile IP e Segurança obrigatórias (objectivos falhados)
  - Introduzir novos tipos de endereçamento (e.g. IP Anycast)

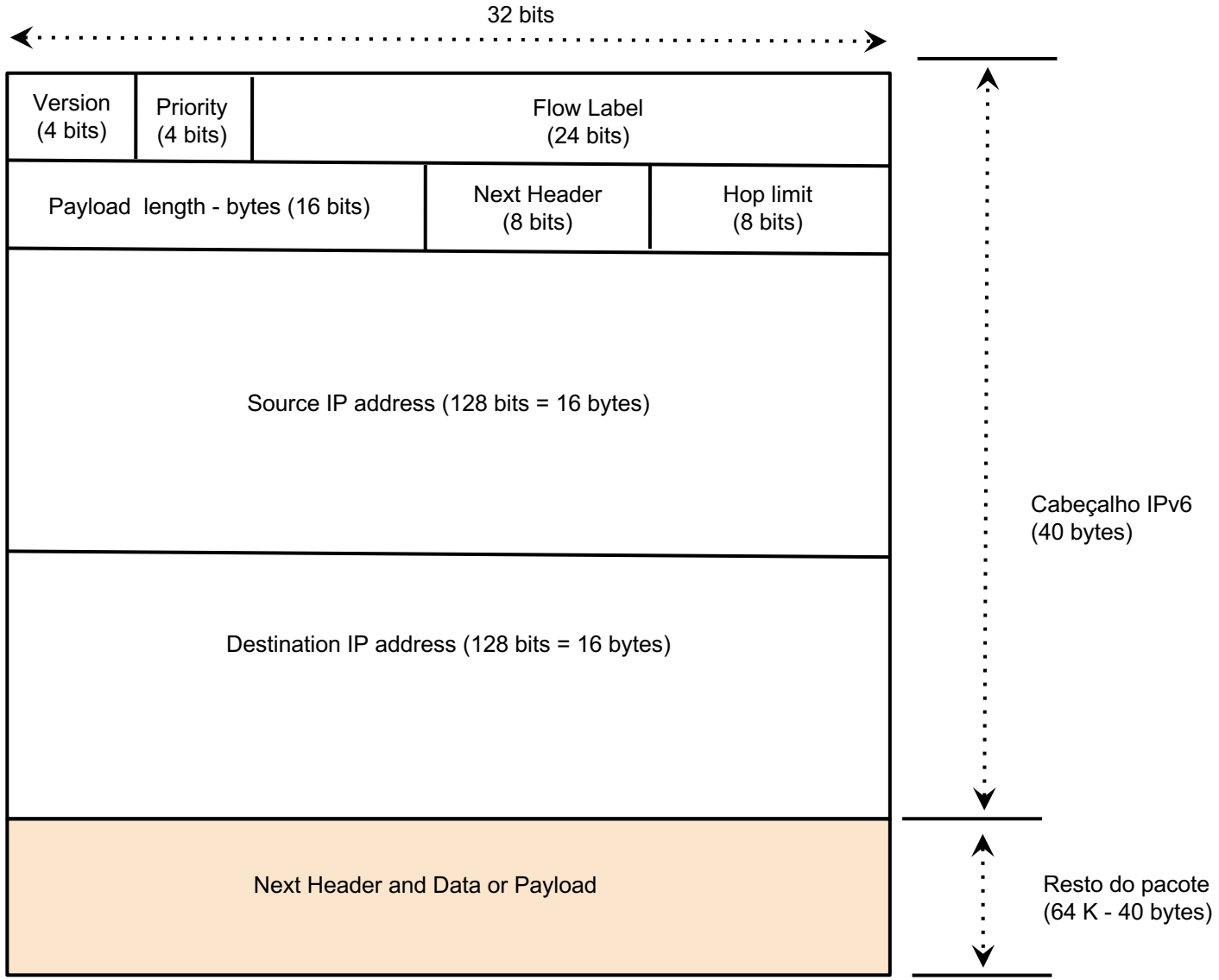
Espaço de Endereçamento IPv6

- 2^{128} endereços = 3.4×10^{38} endereços
- 340,282,366,920,938,463,463,374,607,431,768,211,456
- 5×10^{28} endereços por pessoa ($7 . 10^{9}$ é a população mundial atual)
- No limite os endereços podem ser gerados aleatoriamente
- Os maiores prefixos têm 64 bits – é possível fazer coincidir a parte final do endereço IP com um endereço MAC
- Grande flexibilidade para introduzir convenções de encaminhamento em prefixos
- Pelo que é possível potenciar encaminhamento mais eficiente (e.g. Reservar prefixos para efeitos especiais)
- Multicasting – prefixo 11111111 ou FF:
- Anycasting – vários computadores com o mesmo endereço IP

Pacote IPv6

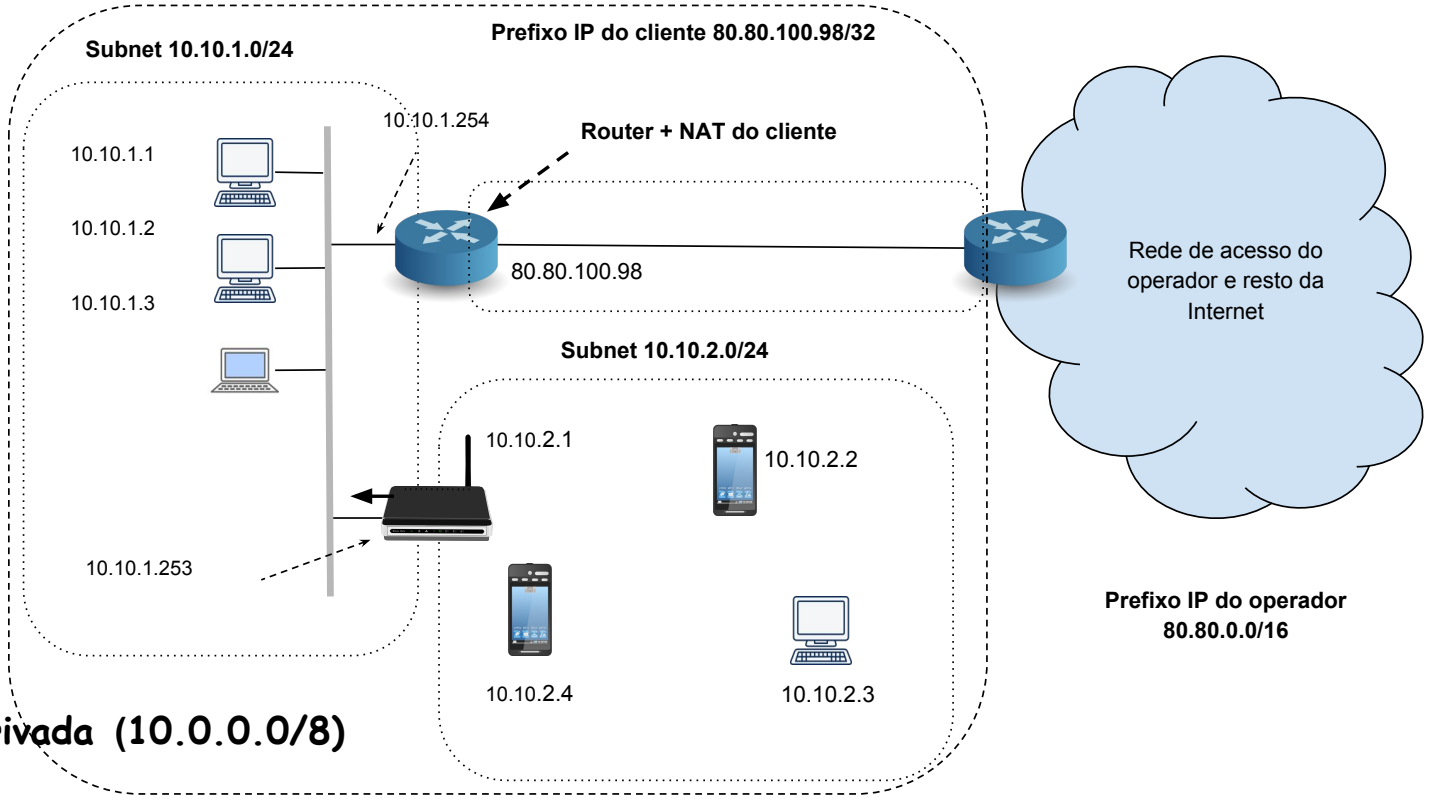
- Com tamanho fixo mas sem *checksum*, fragmentação ou opções
- Prioridade e *flow label* foram pensados para suportar a qualidade de serviço, mas o campo *flow label* não é praticamente usado
- Next hdr ≈ protocol
- Hop limit ≈ TTL
- A sua implementação está generalizada mas é ainda usado apenas quando já não há alternativa
- Atualmente as RIRs já não afetam endereços IPv4 exceto em África ou em situações excecionais
- Existe um mercado de endereços IPv4 livres

Pacote IPv6


32 bits
Version (4 bits)
Priority (4 bits)
Flow Label (24 bits)
Payload length - bytes (16 bits)
Next Header (8 bits)
Hop limit (8 bits)
Source IP address (128 bits = 16 bytes)
Destination IP address (128 bits = 16 bytes)
Next Header and Data or Payload
Cabeçalho IPv6 (40 bytes)
Resto do pacote (64 K - 40 bytes)

NAT – *Network Address Translation*

- Motivação: vários computadores pretendem aceder à Internet mas só se dispõe de 1 endereço IP afetado pelo ISP por DHCP ao *router*



Os pacotes com origem e destino na rede privada têm endereços no prefixo 10.0.0.0/8

Os pacotes que saem da rede local têm o endereço origem 80.80.100.98, e diferentes números de porta

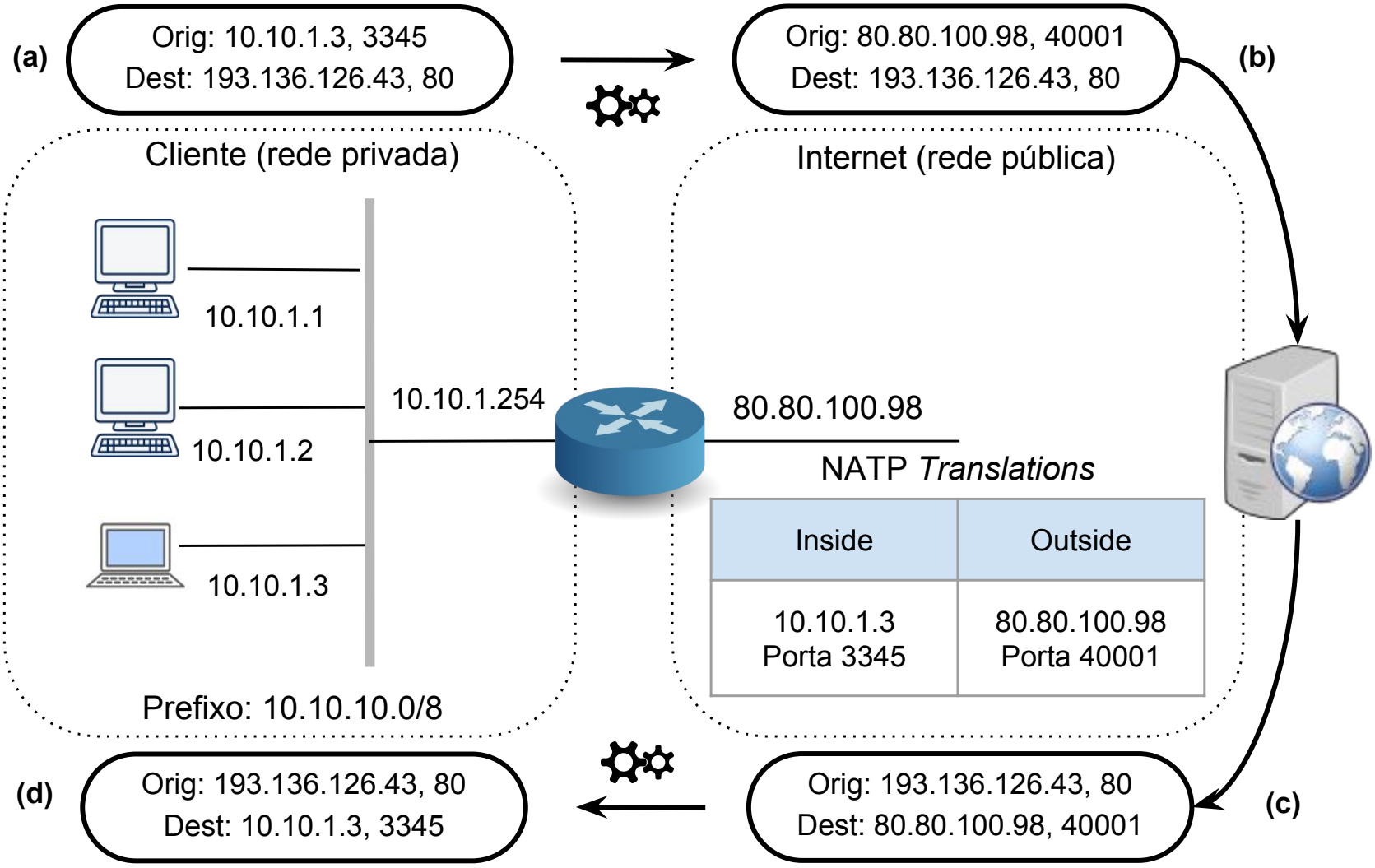
NAT – Network Address Translation

- Os utilizadores internos vão todos aparecer na Internet como tendo sempre um só endereço IP origem, que partilham
- O ISP só necessita de afectar 1 único endereço IP, dito o endereço IP público
- O utilizador pode mudar de ISP e fica com endereços internos independentes do ISP
- Os endereços internos são privados, isto é, desconhecidos no exterior
- Os computadores da rede local não podem ser endereçados do exterior — bom para proteção, mau para certas aplicações

O *Router* NAT tem de

- Transformar os pacotes em saída
  - Substituir no cabeçalho do pacote (endereço IP origem, porta origem, ...) por (endereço IP público, nova porta, ...). Desta forma os servidores externos vão responder para o *router* (..., endereço IP público, nova porta)
- Memorizar
  - (endereço IP origem, porta origem) e associá-lo a (endereço IP público, nova porta) de forma a poder transformar um pacote (..., endereço IP público, nova porta) em (...., endereço IP origem, porta origem)
- Transformar os pacotes em entrada
  - Transformar os pacotes quer recebe dirigidos a (..., endereço IP público, nova porta) em (...., endereço IP origem, porta origem)

NAT – Exemplo de Funcionamento



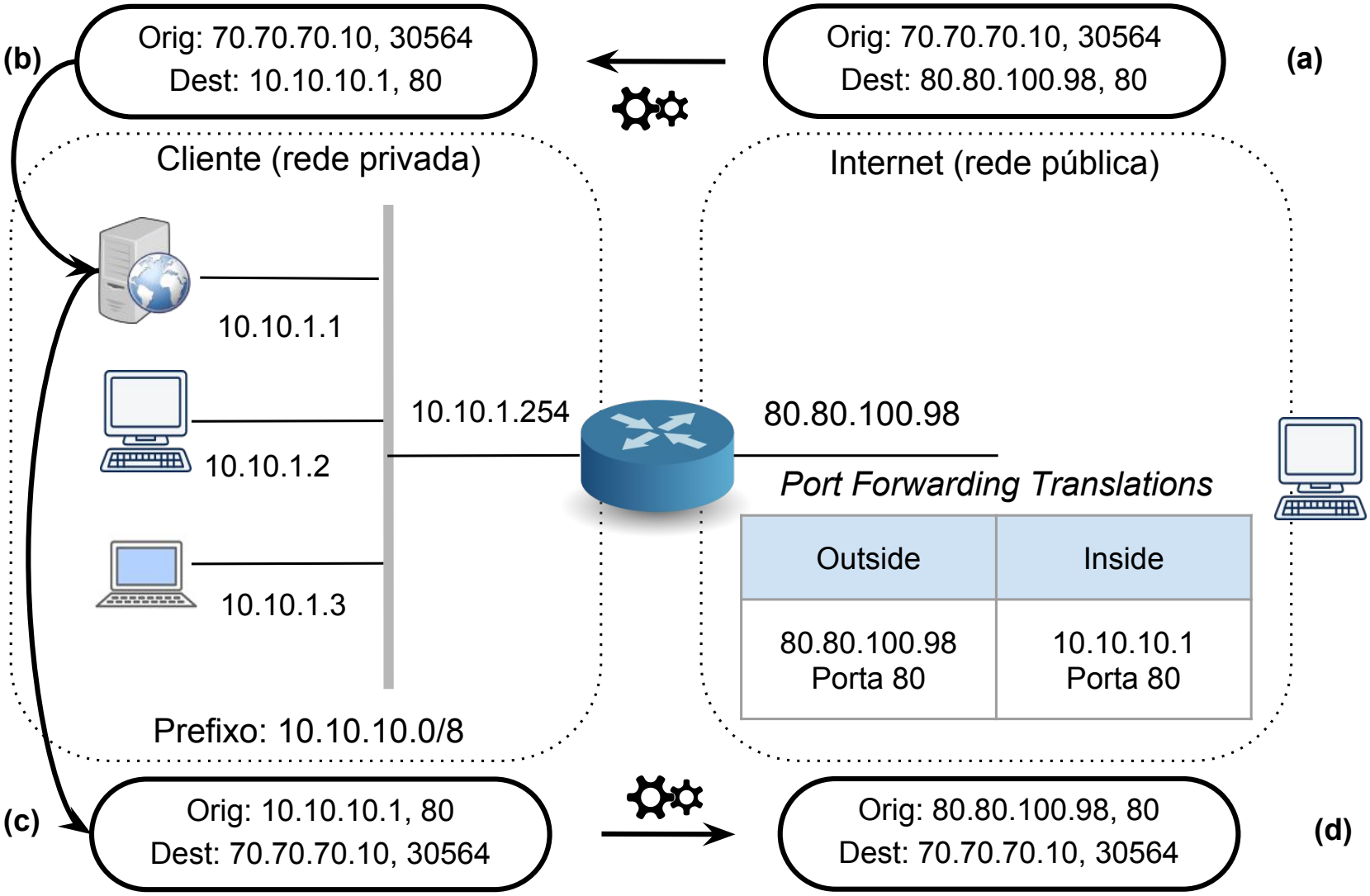
Análise do NAT

- As portas são representadas em 16 bits, logo é possível reservar uma gama de por exemplo 50.000 portas para o NAT
  - 50.000 conexões diferentes com um único endereço IP público
- O NAT é sujeito a controvérsia
  - Certas aplicações necessitam de conhecer endereço e portas das partes em diálogo (e.g. certos jogos) e para atravessarem o *router* os pacotes têm de ser transformados
  - Alguns argumentam que viola a filosofia inicial da Internet
  - Torna difícil ter servidores na rede exceto se estes estiverem dentro do *router*
- Mas o NAT resolve vários problemas reais
  - Permite trabalhar com menos endereços públicos
  - Torna o endereçamento na minha rede independente do ISP e torna essa rede "portável"
  - Tem propriedades suplementares de segurança

Acesso a Servidores com NAT

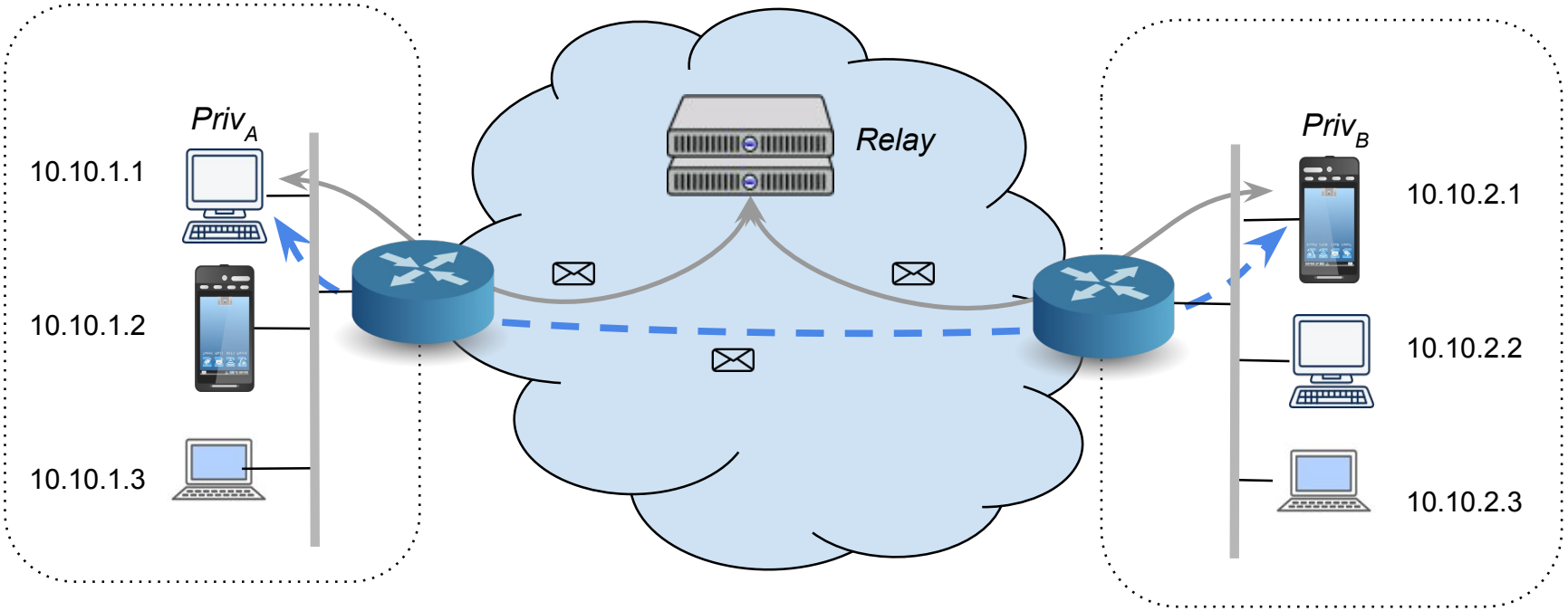
- Gostaríamos de ter um servidor na rede interna acessível do exterior
  - Um servidor com o endereço interno 10.0.0.1 mas apenas visível do exterior como tendo o endereço público do *router*
  - Como abrir conexões de fora para dentro ?
- Solução com *port forwarding*
  - Configurar estaticamente o *router* para redirigir todos os datagramas dirigidos à sua porta 80 para a porta 80 do servidor interno 10.0.0.1

Acesso a Servidores com NAT



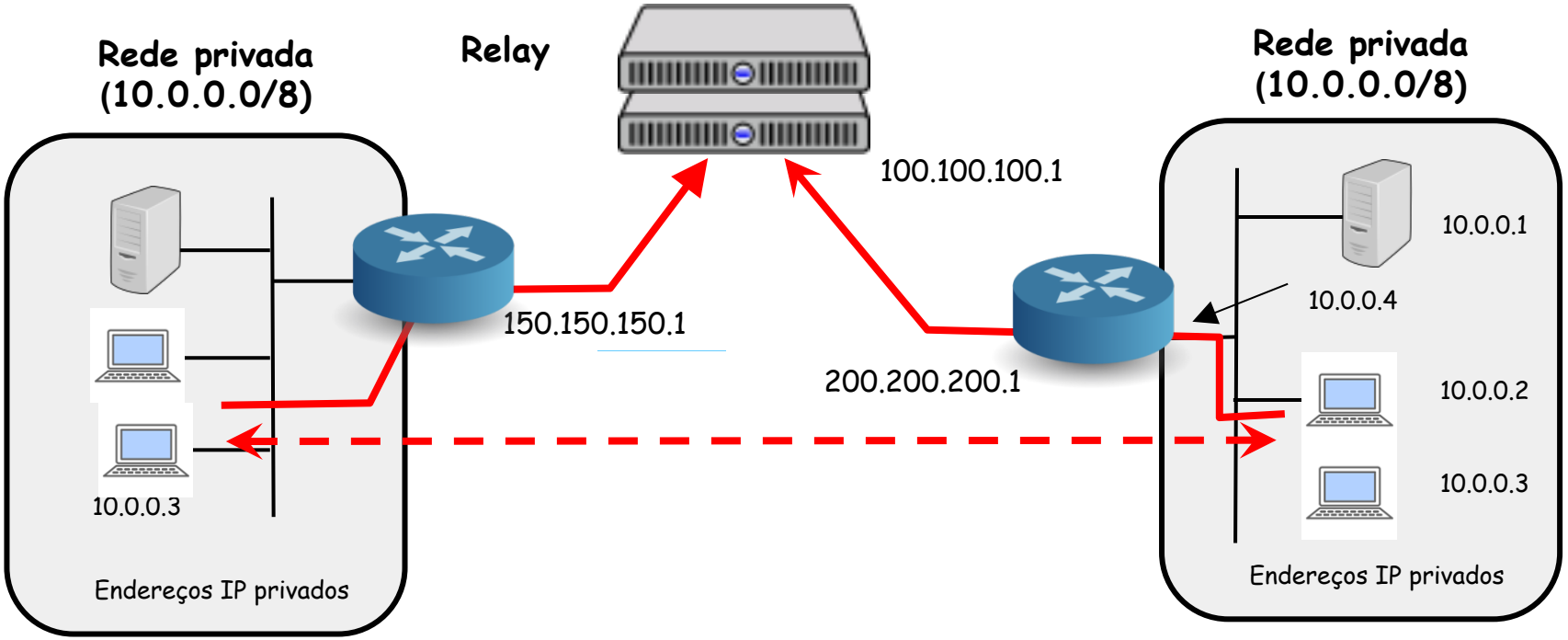
Outra solução - *Relaying*

- O computador interno abre uma conexão para um *relay*
- O cliente abre uma conexão para o *relay*
- O *relay* atua como ponte entre o cliente e o servidor



Relaying - Observações

- Esta solução não é geral pois requer modificações das aplicações ou a utilização de túneis pelos computadores internos como se tivessem um canal (lógico) para um *router* externo (o *relay*)
- Na verdade, nada impede que os dois parceiros estejam ambos por detrás de *routers* NAT e até que tenham os mesmos endereços IP privados



Gamas de Endereços IPv4 Privados

Nome	Gama de endereços IP	Número de endereços	Máscara	Número de bits para a parte *host*
24-bit block	10.0.0.0 – 10.254.254.254	16,777,216	10.0.0.0/8 (255.0.0.0)	24
20-bit block	172.16.0.0 – 172.31.255.255	1,048,576	172.16.0.0/12 (255.240.0.0)	20
16-bit block	192.168.0.0 – 192.168.255.255	65,536	192.168.0.0/16 (255.255.0.0)	16

Conclusões sobre Endereços IP

- Na Internet os computadores têm de ter endereços IP distintos e conhecerem os endereços IP dos *routers* e de diversos servidores
- O protocolo DHCP automatiza a aquisição destas informações
- O protocolo ARP automatiza a aquisição dos endereços nível canal nos canais *broadcasting*
- A técnica de NAT isola uma rede do exterior e permite que os computadores internos partilhem um único endereço IP público

Conclusões

- O protocolo IP está na base do funcionamento da interno da Internet
  - Define o formato dos pacotes IP e o significado dos diferentes campos
  - Define o espaço de endereçamento e como este deve ser interpretado
  - Define como os *routers* devem fazer o encaminhamento dos pacotes
- O protocolo IP é crítico para a eficiência da Internet
- Tem duas versões fundamentais: IPv4 e IPv6
  - Cuja principal diferença é o espaço de endereçamento