

# Redes de Computadores

**Prof. Miguel Elias Mitre Campista**

`http://www.gta.ufrj.br/~miguel`

# Roteiro Resumido

- Princípios básicos da Internet
- Princípios básicos de comunicação em redes
- Descrição das diferentes camadas de protocolos
  - Camada de aplicação e os seus protocolos
  - Camada de transporte e os seus protocolos
  - Camada de rede
  - Camada de enlace

# Parte III

## Camada de Rede e seus Protocolos

# Camada de Rede

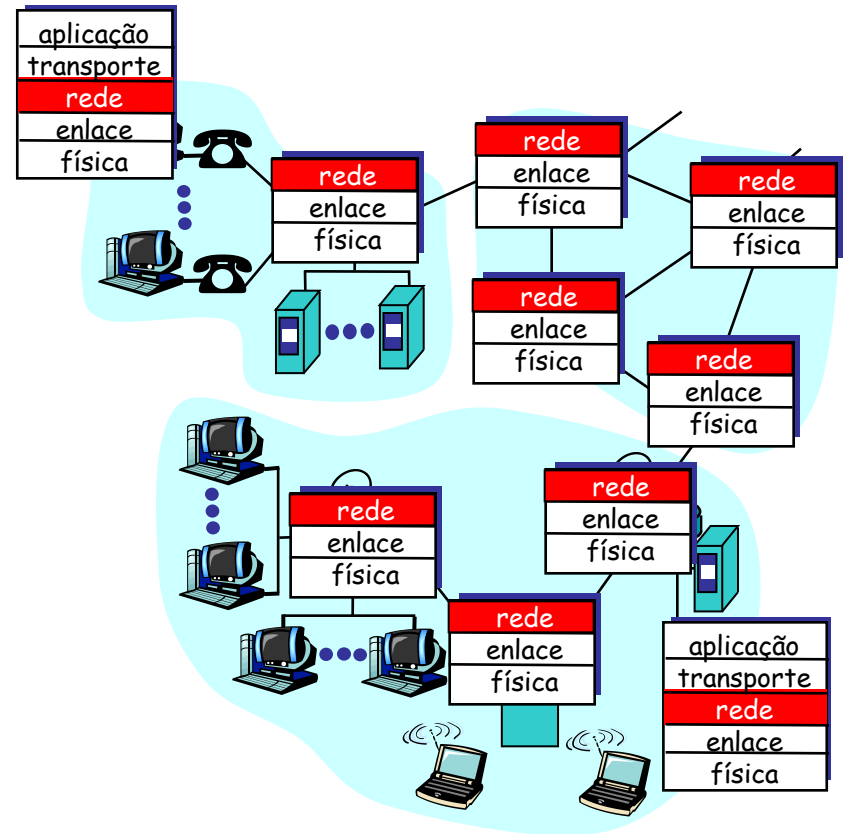


# Camada de Rede

- Responsável por:
  - Determinar o melhor caminho para o envio dos pacotes
    - É função dos **protocolos de roteamento**
  - Encaminhar os pacotes até o destino
    - É função do **protocolo IP**
  - Interconectar redes de diferentes tecnologias
    - É função do **protocolo IP**

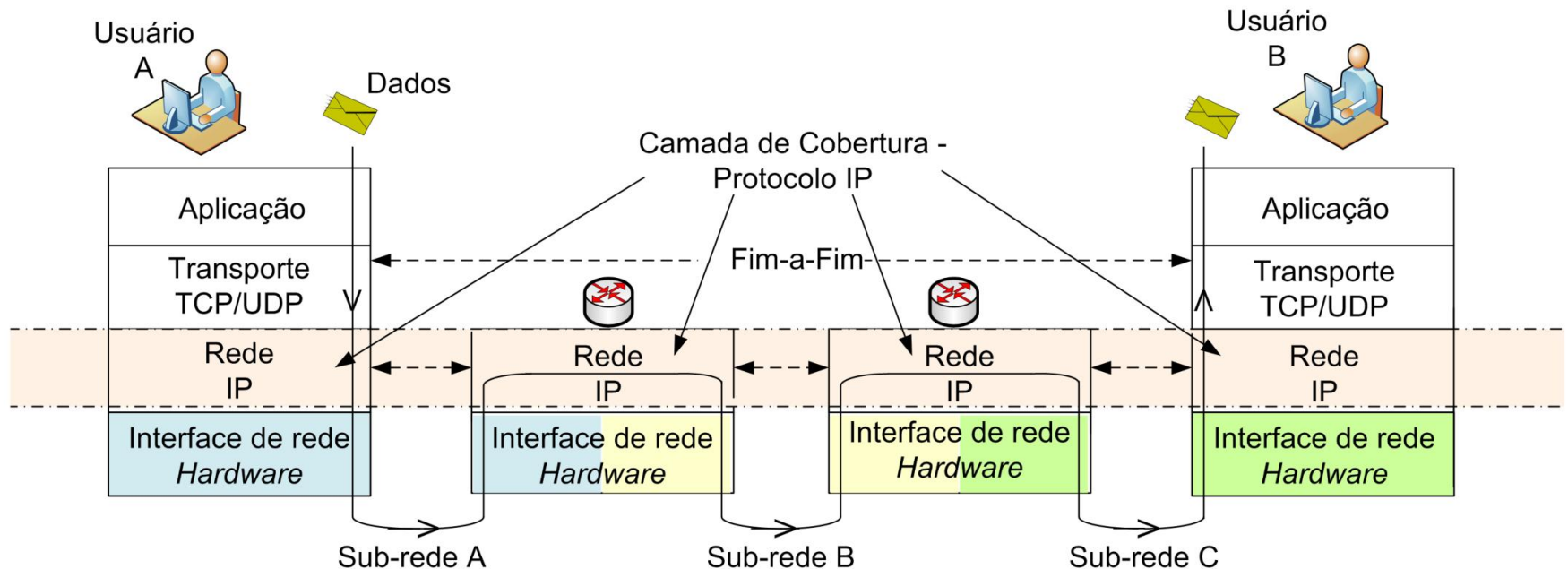
# Camada de Rede

- Protocolos da camada de rede
  - Executados nos sistemas finais e nos roteadores



# Camada de Rede

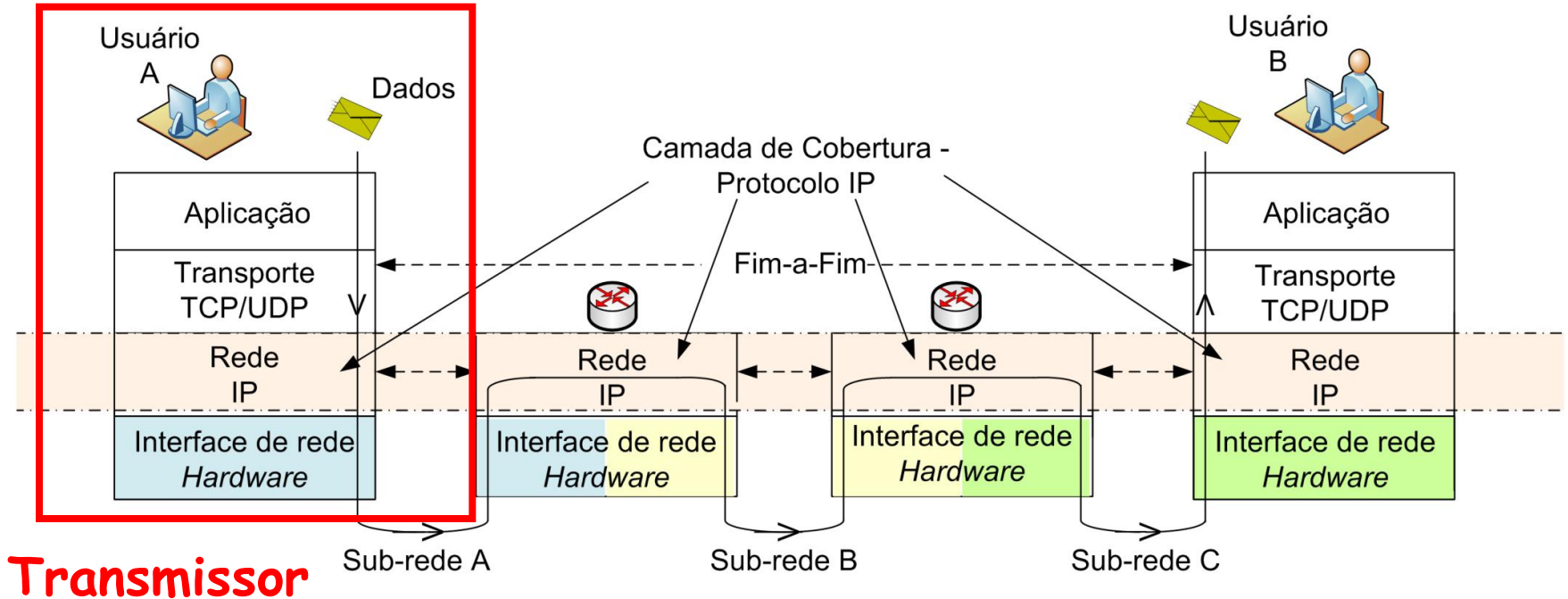
- Protocolos da camada de rede
  - Executados nos sistemas finais e nos roteadores



**Transporta segmentos da estação remetente à receptora**

# Camada de Rede

- Protocolos da camada de rede
  - Executados nos sistemas finais e nos roteadores

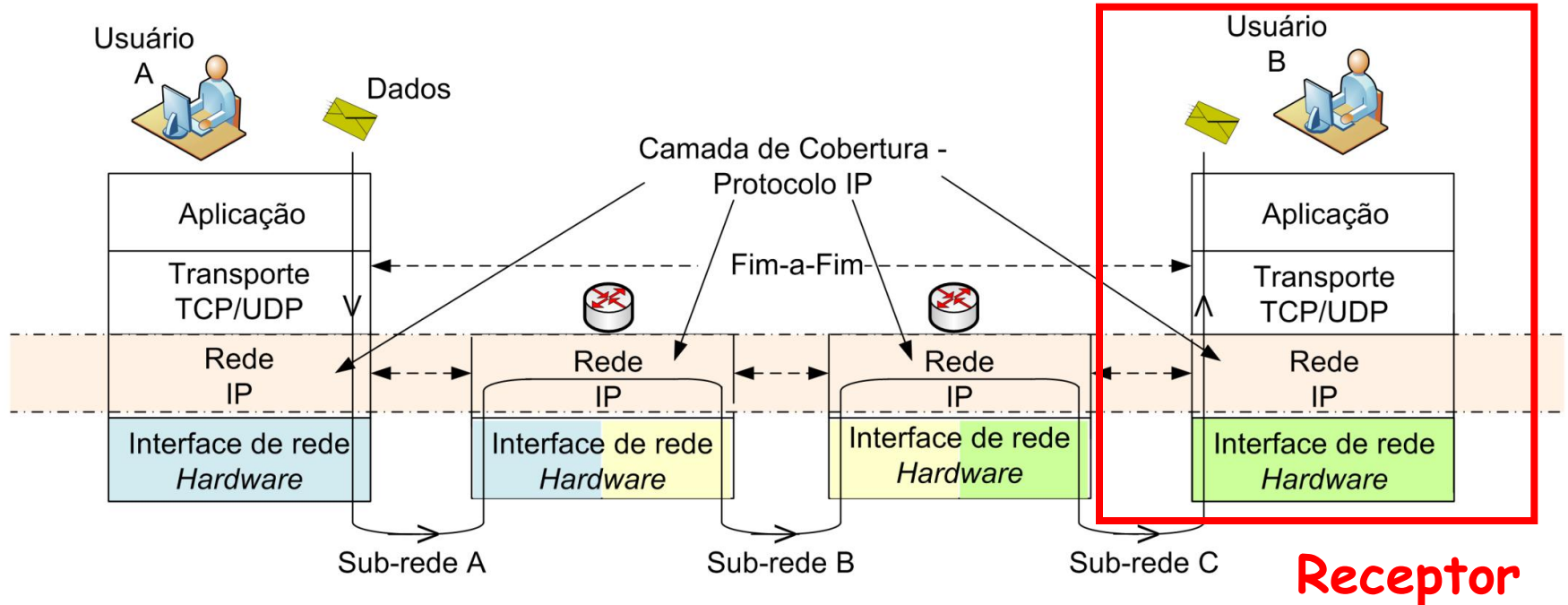


**Transmissor encapsula segmentos dentro de datagramas**



# Camada de Rede

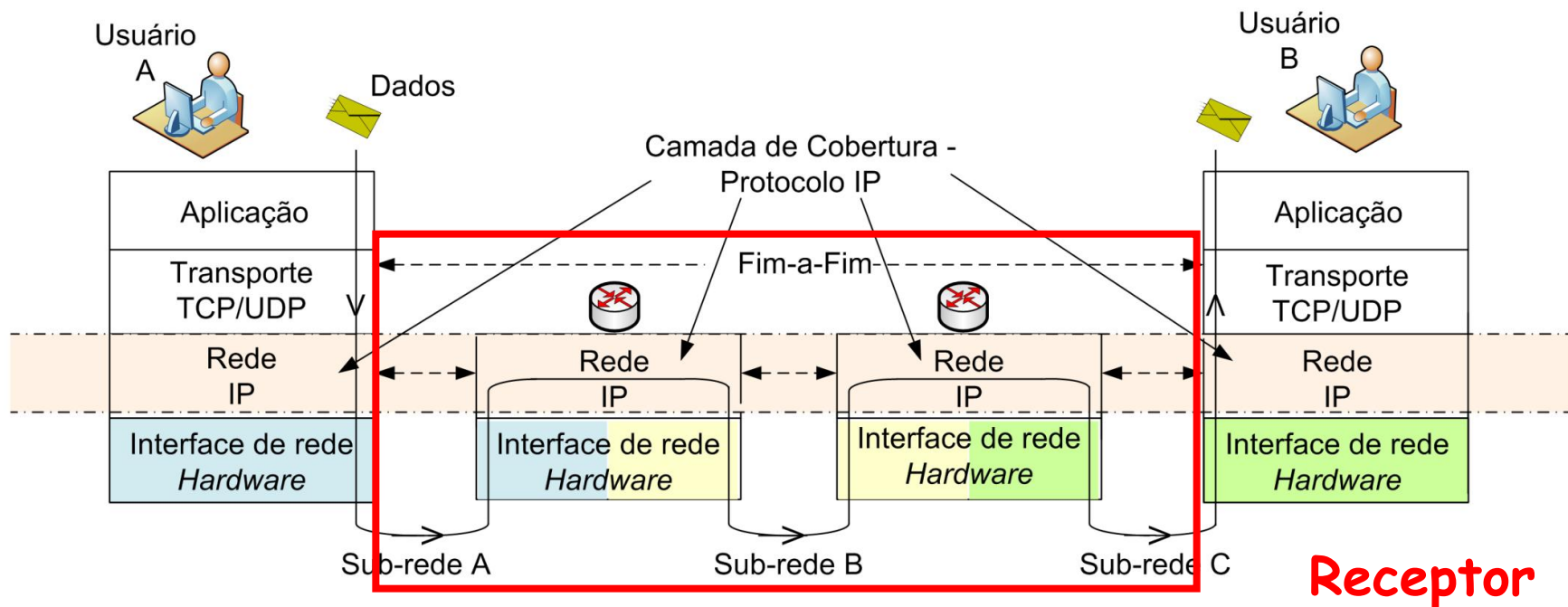
- Protocolos da camada de rede
  - Executados nos sistemas finais e nos roteadores



**Receptor entrega os segmentos para a camada de transporte**

# Camada de Rede

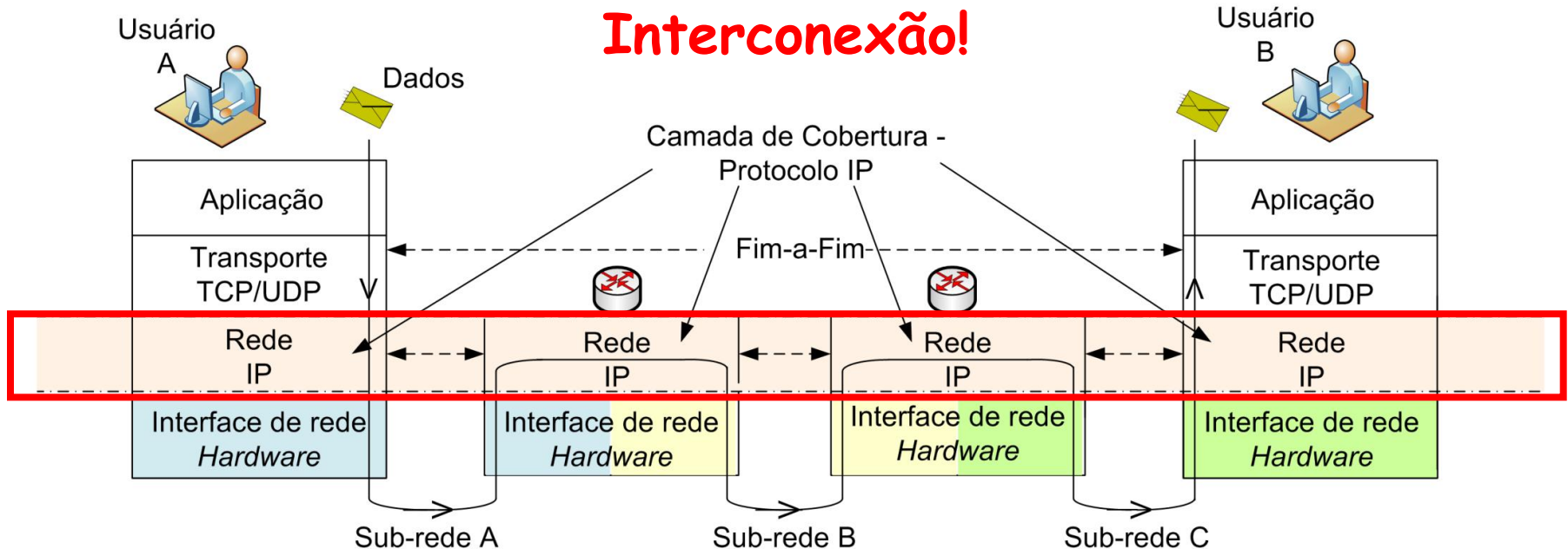
- Protocolos da camada de rede
  - Executados nos sistemas finais e nos roteadores



**Roteadores examinam campos de cabeçalho de todos os datagramas IP que passam por eles**

# Camada de Rede

- Protocolos da camada de rede
  - Executados nos sistemas finais e nos roteadores

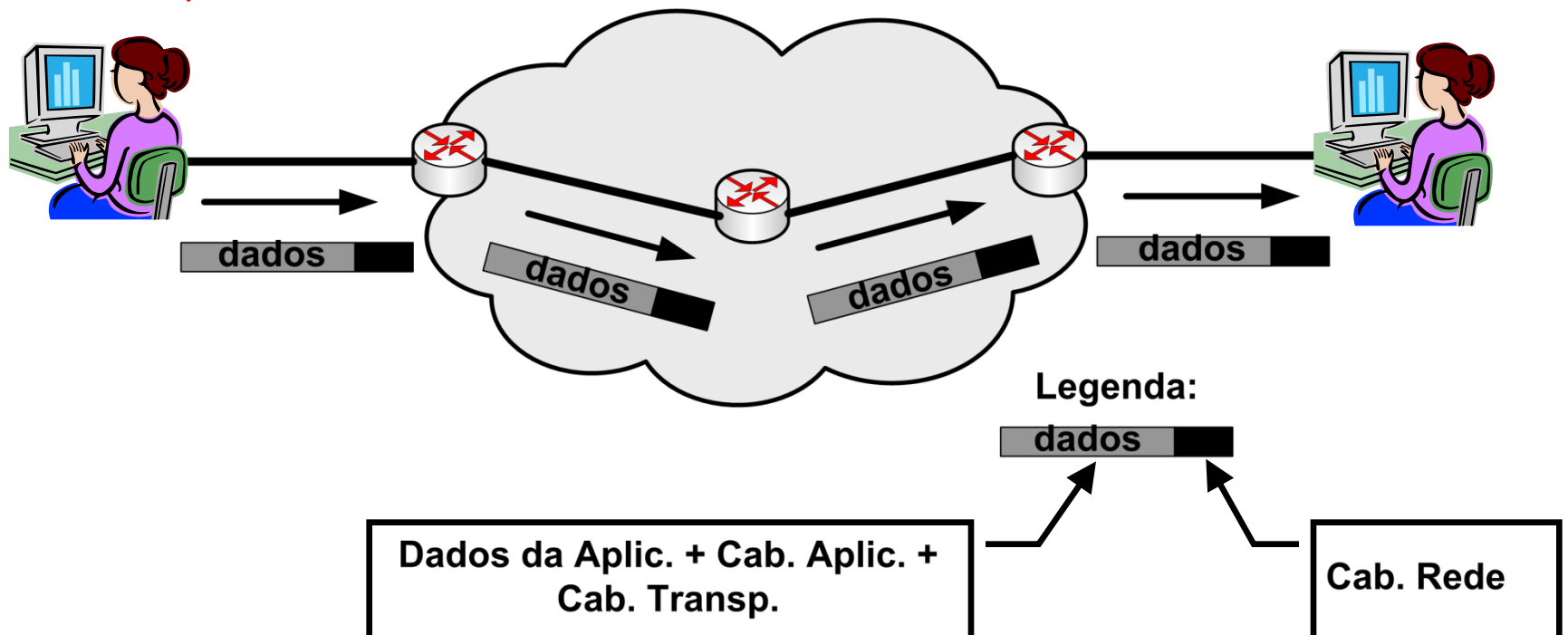


# Transparência

- Transparência sintática

- Pacotes são transferido da origem ao destino sem que a rede modifique os dados

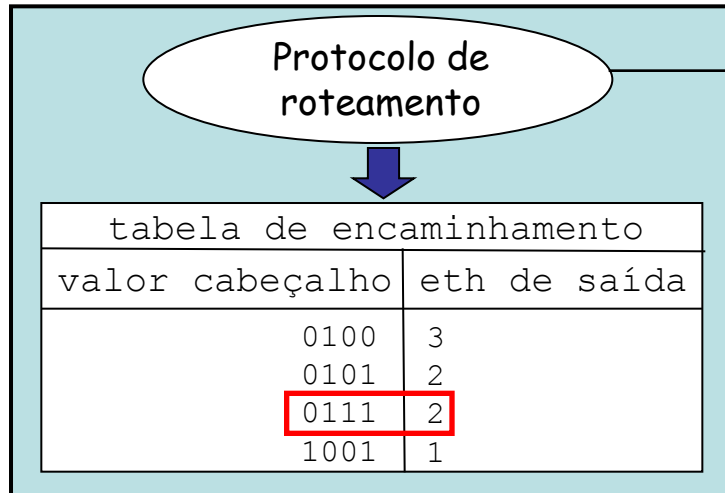
- Apenas erros de transmissão modificam os dados



# Encaminhamento X Roteamento

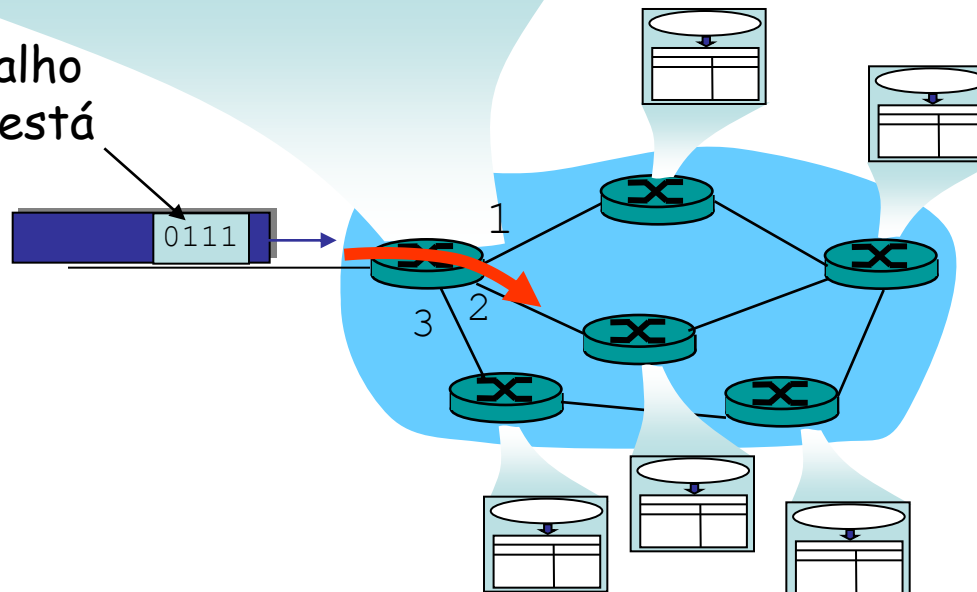
- Encaminhamento (repasse)
  - "Mover" pacotes de uma entrada do roteador para a saída apropriada
    - É função do protocolo IP
- Roteamento
  - Determinar a rota a ser seguida pelos pacotes da fonte até o destino
    - É função dos protocolos de roteamento

# Encaminhamento X Roteamento



Responsável por construir a tabela de encaminhamento

valor no cabeçalho do pacote que está chegando



# Modelos de Serviço

- Tipos de serviços que **poderiam** ser oferecidos pela camada de rede
  - Definem as características do transporte de pacotes fim-a-fim entre transmissor e receptor
- Para pacotes individuais
  - Entrega garantida
    - Pacote irá chegar ao destino "mais cedo ou mais tarde"
  - Entrega garantida com atraso limitado
    - Pacote irá chegar com atraso menor que 100 ms

# Modelos de Serviço

- Para fluxos de pacotes
  - Entrega ordenada de pacotes
  - Largura de banda mínima garantida
  - Jitter máximo garantido
  - Serviços de segurança
    - Usando uma chave secreta de sessão o transmissor poderia cifrar o conteúdo de todos os pacotes enviados para o destinatário
- Na Internet
  - Apenas um protocolo: o IP
    - Apenas um serviço oferecido → Melhor esforço



# Melhor Esforço

- Roteadores se esforçam ao máximo para entregar os pacotes
  - Da melhor maneira possível e sem distinção
- Nós simples e de baixo custo - sem estados na rede
  - Encaminhamento de pacote independente um dos outros
  - Sem reserva de recursos, recuperação de erros, garantia de acesso
  - Atraso dependente do tamanho da fila
  - Sem garantia de entrega do pacote ao destino
    - Pacote é descartado no roteador se a fila estiver cheia

# Serviços da Camada de Rede

- **Orientado à conexão**
  - Redes de circuitos virtuais
- **Não-orientado à conexão**
  - Redes de datagramas
- **Análogos aos serviços da camada de transporte, porém...**
  - É um serviço estação-a-estação
    - E não processo-a-processo...
  - É orientado à conexão ou não orientado à conexão
    - E não com escolha (p.ex. a camada de transporte que oferece escolha: TCP ou UDP)
  - É implementado no núcleo da rede
    - E não somente nas bordas

# Circuitos Virtuais

- Emular uma rede de comutação de circuitos utilizando comutação de pacotes
  - Caminho da origem ao destino "se comporta" como um circuito telefônico
    - Em termos de desempenho
    - Em ações da rede ao longo do caminho

# Circuitos Virtuais

- Funcionamento
  - Estabelecimento de uma **chamada** antes do envio dos dados
  - Cada pacote carrega a identificação do circuito virtual (CV)
    - **Ao invés de endereços de origem e destino**
  - Cada roteador no caminho origem-destino mantém **estado** para cada conexão que o atravessa
    - **Cada conexão está associada a um CV**
  - Recursos de enlace, roteador (banda, buffers) podem ser **alocados** ao CV

# Circuitos Virtuais

- Um *CV* consiste de:
  - Caminho da origem para o destino
  - Números (identificadores) de *CV*
    - Um número para cada enlace ao longo do caminho
  - Entradas nas tabelas de encaminhamento dos roteadores ao longo do caminho
- Pacotes de um dado *CV* carregam o número desse *CV*
  - Número do *CV* deve ser trocado a cada enlace
  - Novo número do *CV* vem da tabela de encaminhamento

# Circuitos Virtuais: Encaminhamento

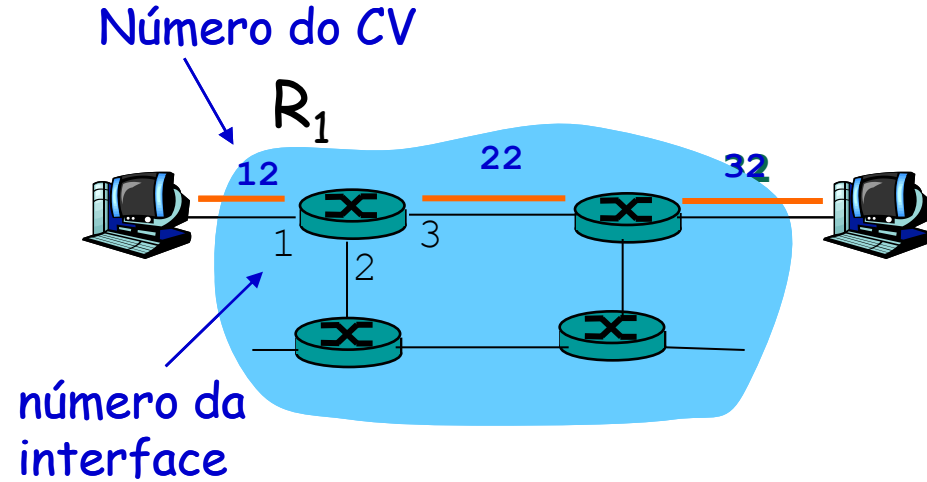
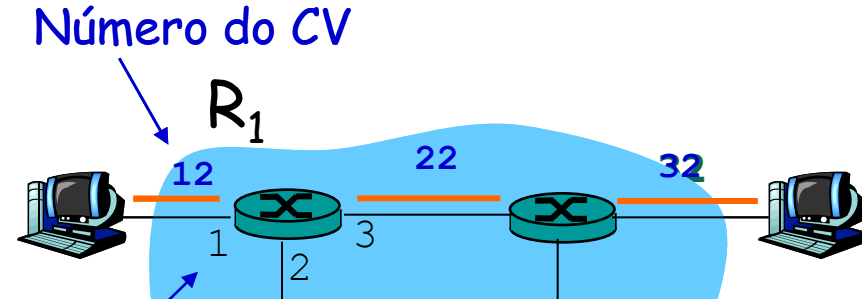


Tabela de encaminhamento  
no roteador  $R_1$

Interf. de entrada	#CV de entrada	Interf. de saída	#CV de saída
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

# Circuitos Virtuais: Encaminhamento



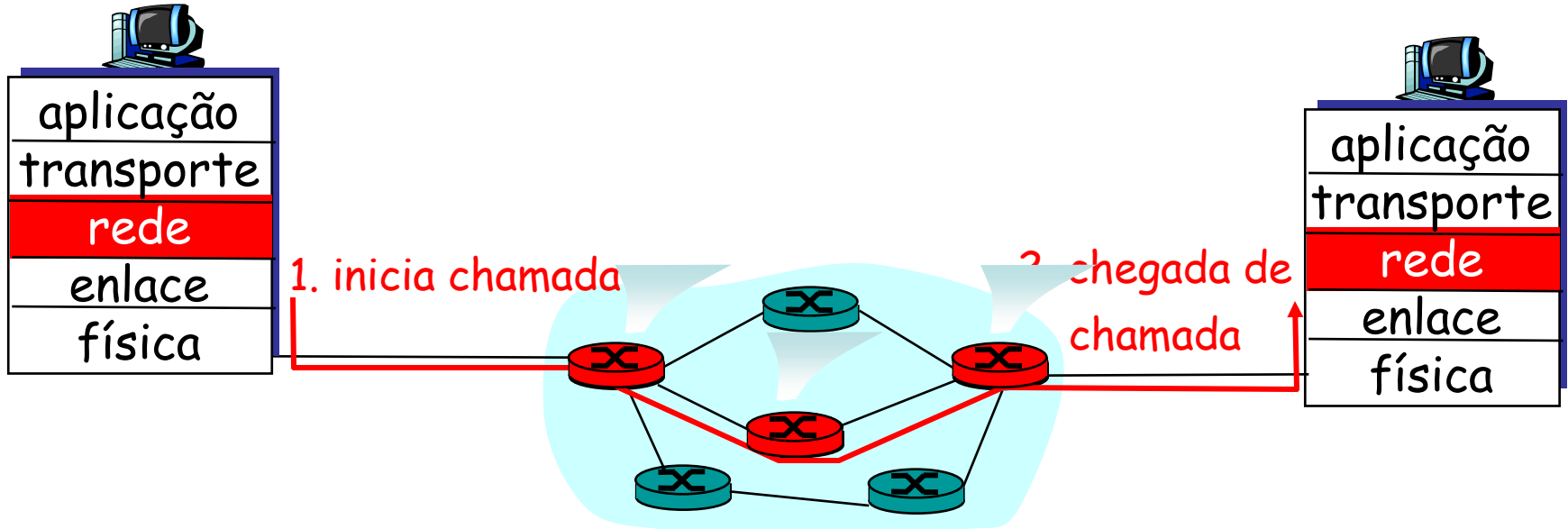
**Roteadores mantêm informação sobre o estado da conexão!**

Tab  
no roteador  $R_1$

Interf. de entrada	#CV de entrada	Interf. de saída	#CV de saída
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

# Circuitos Virtuais: Protocolos de Sinalização

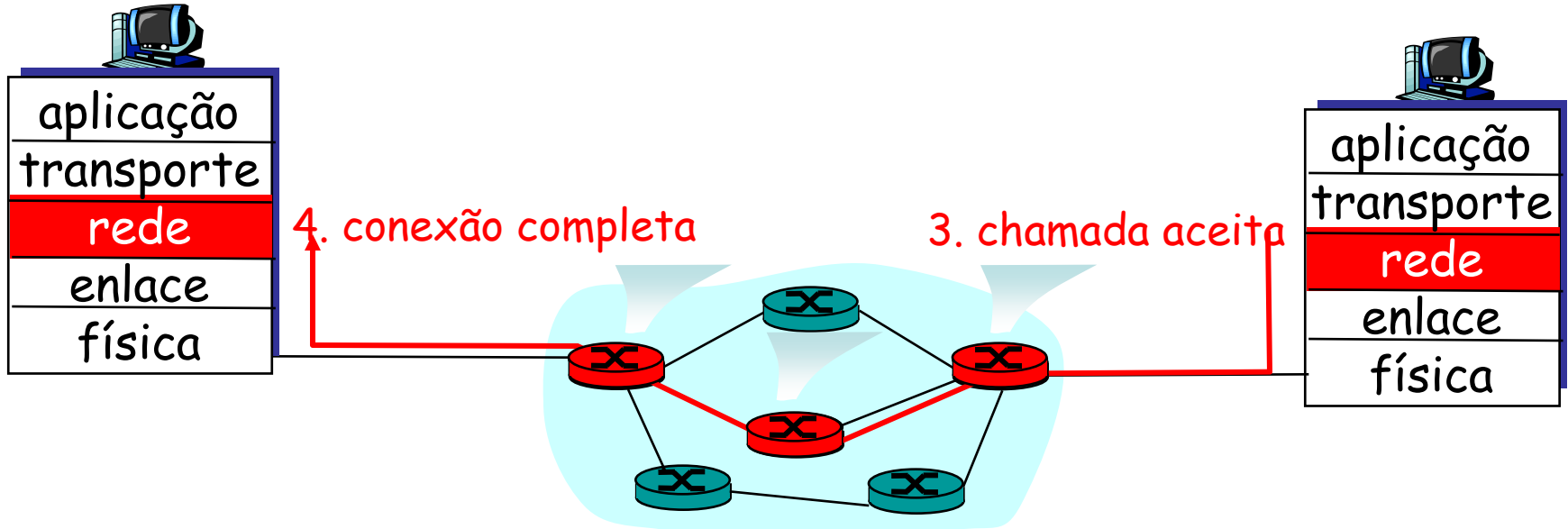
- Responsáveis por estabelecer, manter e destruir um CV
  - Usados em ATM, frame-relay, X.25
  - Não usados na Internet convencional





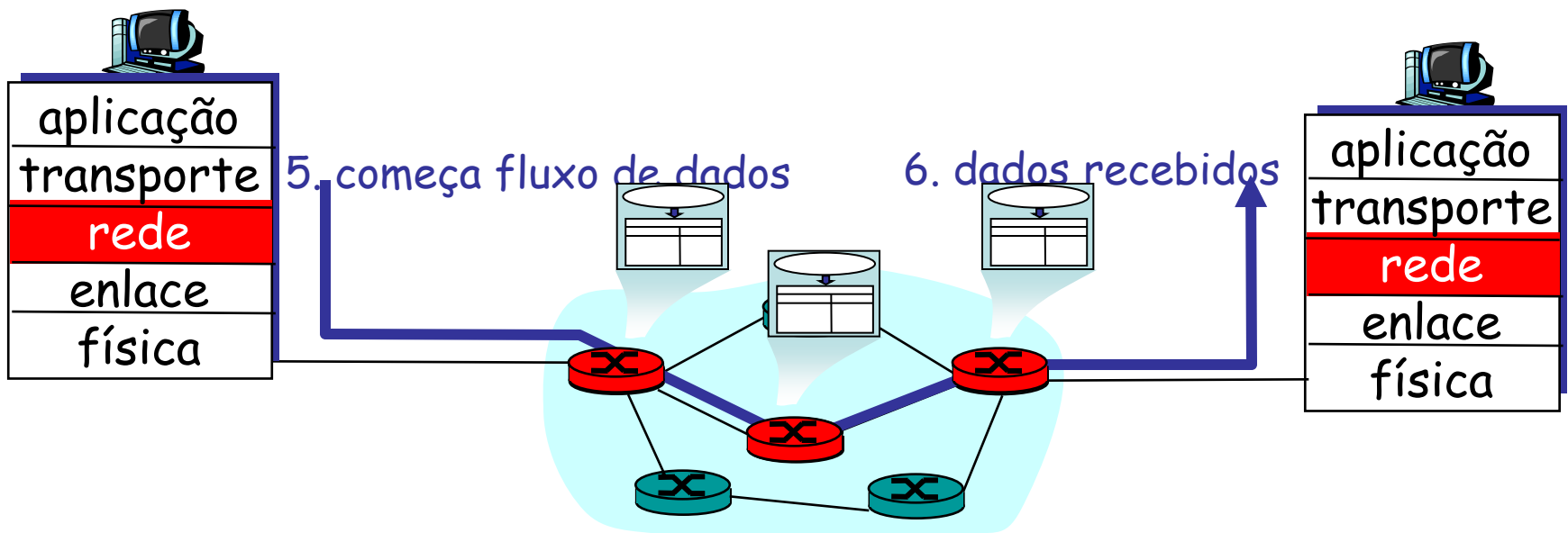
# Circuitos Virtuais: Protocolos de Sinalização

- Responsáveis por estabelecer, manter e destruir um CV
  - Usados em ATM, frame-relay, X.25
  - Não usados na Internet convencional



# Circuitos Virtuais: Protocolos de Sinalização

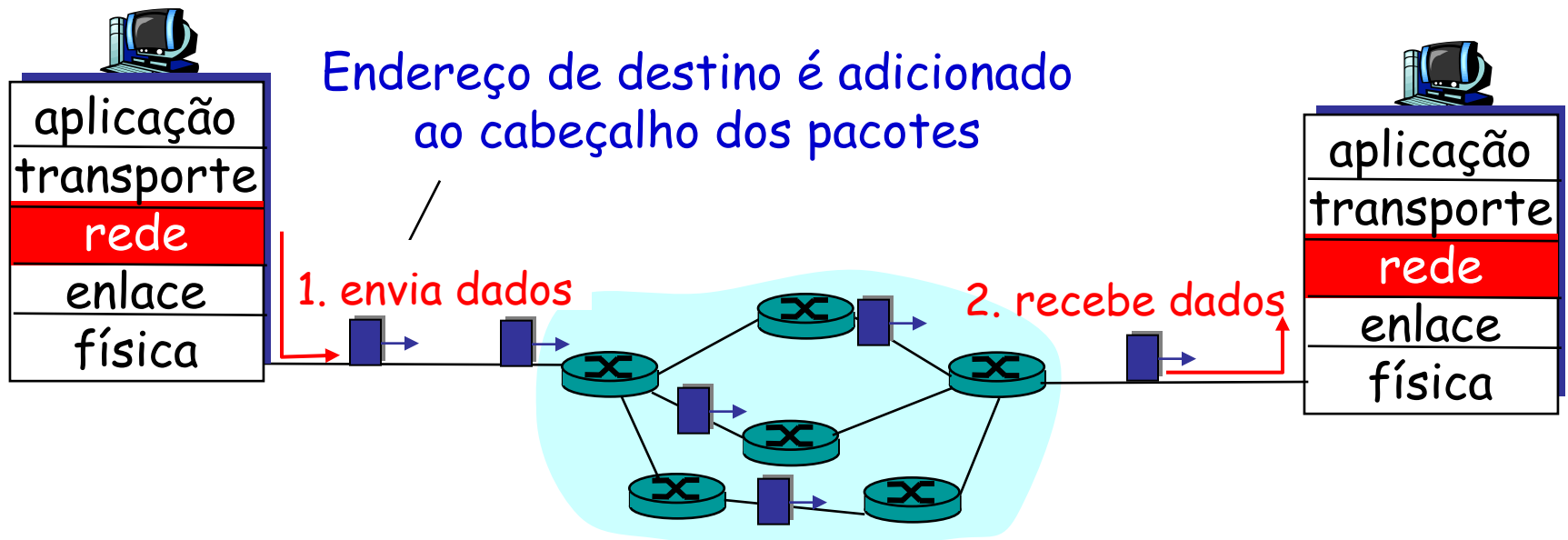
- Responsáveis por estabelecer, manter e destruir um CV
  - Usados em ATM, frame-relay, X.25
  - Não usados na Internet convencional



# Rede de Datagramas

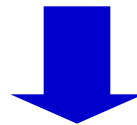
- Serviço não confiável
- Sem estabelecimento prévio de conexão
- Roteadores não guardam estado sobre conexões
- Pacotes são encaminhados
  - Com base no endereço de destino
  - De acordo com o modelo de melhor esforço
- Dois pacotes entre o mesmo par origem-destino podem seguir caminhos diferentes

# Rede de Datagramas



# Datagramas: Encaminhamento

- Endereço IP: 32 bits
  - 4 bilhões de endereços → 4 bilhões de entradas!
  - Agregação de endereços



**Como resumir a tabela?**

# Maior Prefixo

Faixa de endereços de destino	Interface de saída
11001000.00010111.00010000.00000000 a 11001000.00010111.00010111.11111111	0
11001000.00010111.00011000.00000000 a 11001000.00010111.00011000.11111111	1
11001000.00010111.00011001.00000000 a 11001000.00010111.00011111.11111111	2
Caso contrário	3

# Maior Prefixo

Faixa de endereços de destino	Interface de saída
<p>11001000.00010111.00010000.00000000</p> <p>a</p> <p>11001000.00010111.00010111.11111111</p>	0
<p>11001000.00010111.00011000.00000000</p> <p>a</p> <p>11001000.00010111.00011000.11111111</p>	1
<p>11001000.00010111.00011001.00000000</p> <p>a</p> <p>11001000.00010111.00011111.11111111</p>	2
<p>Caso contrário</p>	3

# Maior Prefixo

Faixa de endereços de destino	Interface de saída
11001000.00010111 <b>00010</b> 000.00000000 a 11001000.00010111 <b>00010</b> 111.1111111	0
11001000.00010111 <b>00011000</b> .00000000 a 11001000.00010111 <b>00011000</b> .1111111	1
11001000.00010111 <b>00011</b> 001.00000000 a 11001000.00010111 <b>00011</b> 111.1111111	2
Caso contrário	3



# Maior Prefixo

Faixa de endereços de destino	Interface de saída
11001000.00010111.00010	0
11001000.00010111.00011000	1
11001000.00010111.00011	2
Caso contrário	3

# Maior Prefixo

Faixa de endereços de destino	Interface de saída
11001000.00010111.00010	0
11001000.00010111.00011000	1
11001000.00010111.00011	2
Caso contrário	3

## Exemplos

ED: 11001000 00010111 00010110 10100001

Qual interface?

ED: 11001000 00010111 00011000 10101010

Qual interface?

# Maior Prefixo

Faixa de endereços de destino	Interface de saída
11001000.00010111.00010	0
11001000.00010111.00011000	1
11001000.00010111.00011	2
Caso contrário	3

## Exemplos

ED: 11001000 00010111 00010110 10100001

Interface 0

ED: 11001000 00010111 00011000 10101010

Interface 1

# Circuitos Virtuais X Datagramas

Características	Circuito Virtual	Datagrama
Estabelecimento de conexão	É necessário	Não é necessário
Endereçamento	Identificador do CV	Endereços da fonte e do destino
Estados	Por conexão	Sem estado
Roteamento	Rota escolhida na conexão e seguida posteriormente	Cada pacote é "independente"
Falha de roteadores	Todos os circuitos fechados	Perda de pacotes durante a falha
Qualidade de serviço	Mais fácil	Difícil
Controle de congestionamento	Mais fácil	Difícil

# Arquitetura de Roteadores

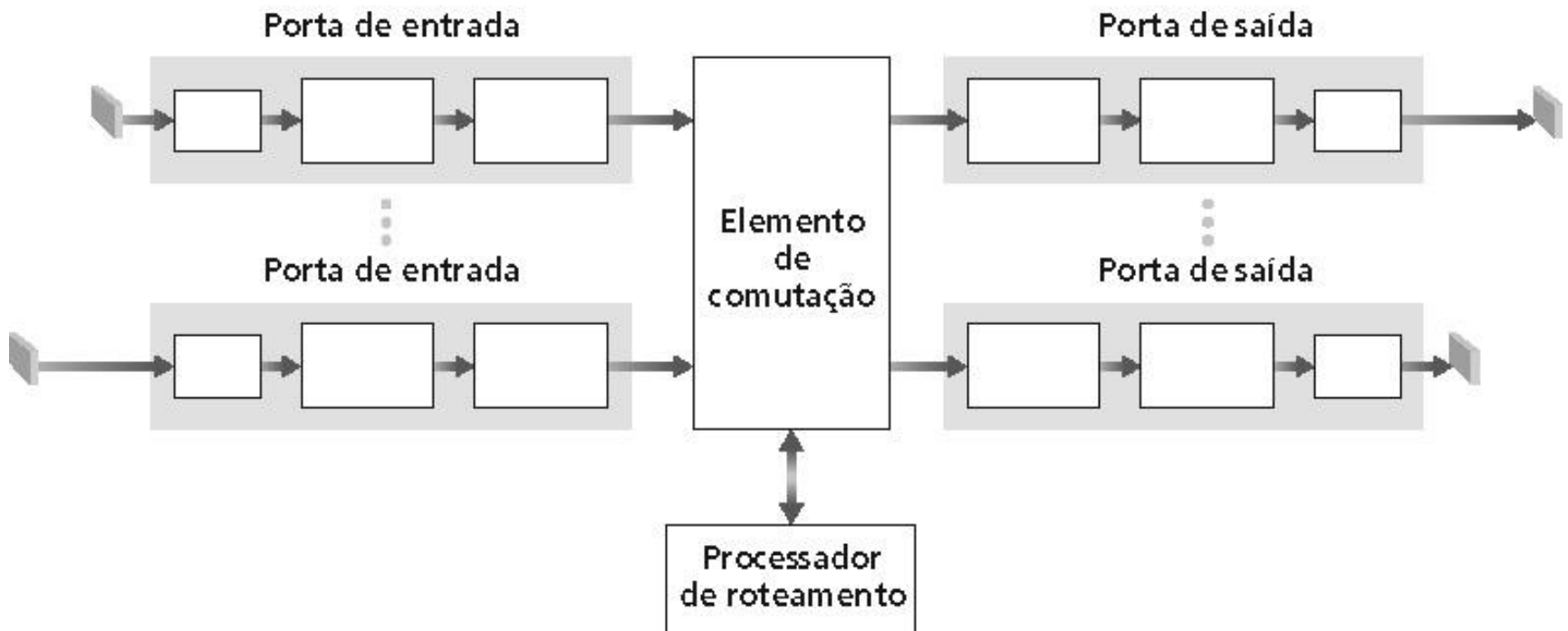
# Roteador

- Elemento responsável por...
  - Determinar o caminho entre um par origem-destino
    - Ação distribuída
  - Encaminhar pacotes
  - Interconectar redes distintas

# Roteador

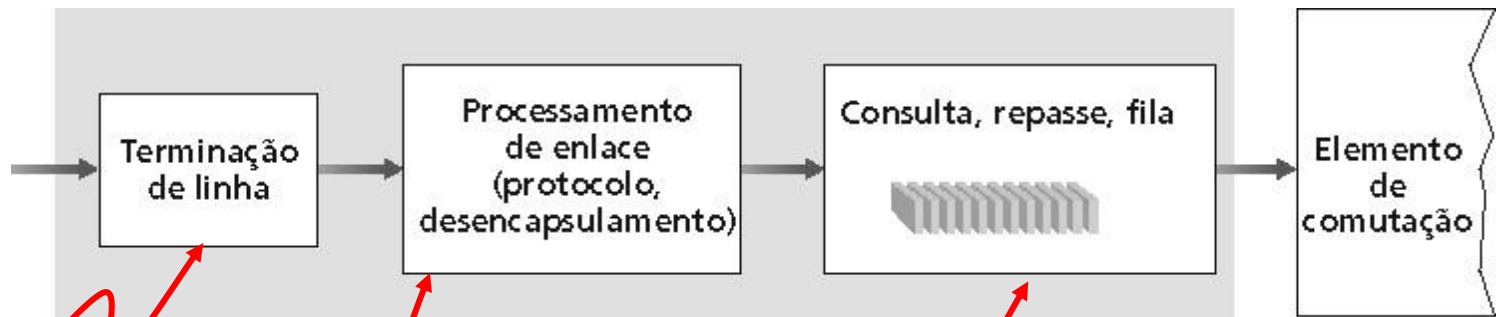
- Cada pacote ao chegar a um roteador...
  - Tem seu endereço de destino analisado (*best-prefix match*)
    - Se o endereço for igual ao de uma das interfaces do roteador
      - Pacote é enviado para camada de transporte
    - Caso contrário
      - Pacote é encaminhado a outro roteador pela interface mais indicada

# Roteador





# Funções das Portas de Entrada



**Camada física:**  
recepção de bits

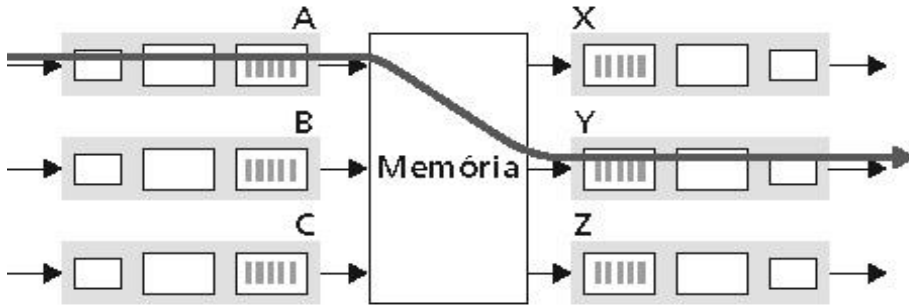
**Camada de enlace:**  
p.ex., Ethernet

**Comutação descentralizada (realizada em cada porta de entrada):**

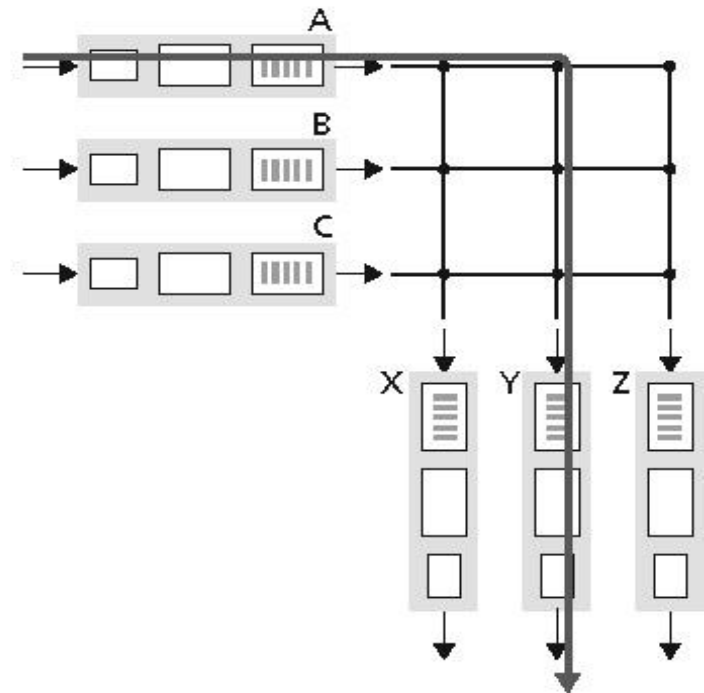
- dado o dest. do datagrama, procura porta de saída usando tab. de rotas na memória da porta de entrada
- meta: completar processamento da porta de entrada na "velocidade da linha"
- filas: se datagramas chegam mais rápido que taxa de reenvio para matriz de comutação

# Três Técnicas de Comutação

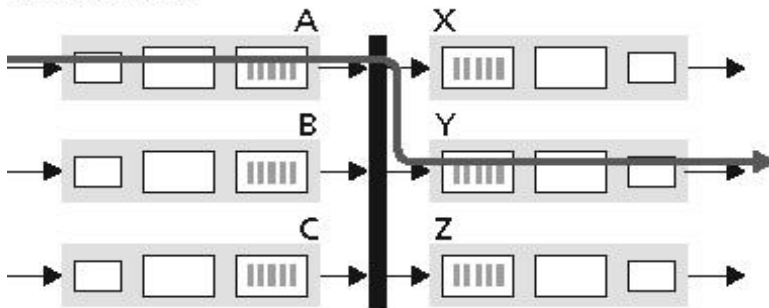
Memória



Crossbar



Barramento

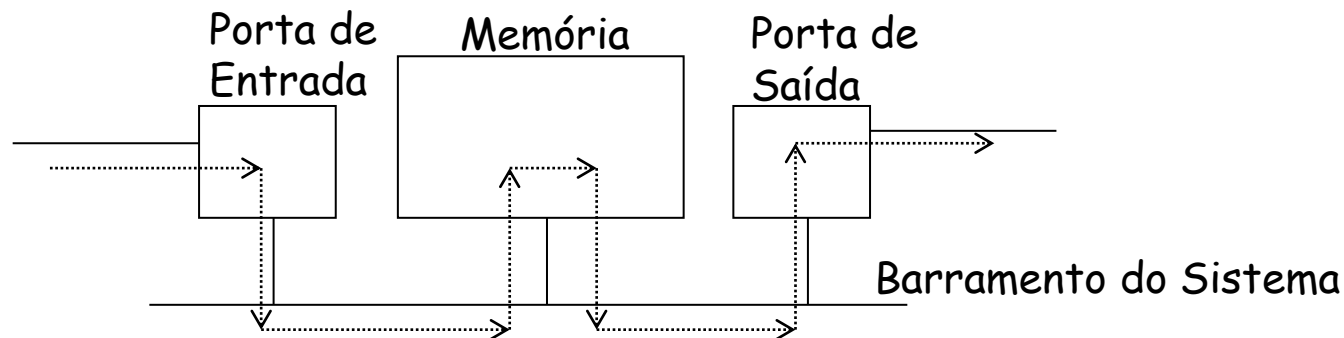


Legenda:



# Comutação por Memória

- Roteadores da primeira geração
- Pacote copiado pelo processador (único) do sistema para a memória compartilhada
  - Velocidade limitada pela largura de banda da memória
    - **Duas travessias do barramento por datagrama**



# Comutação por Barramento

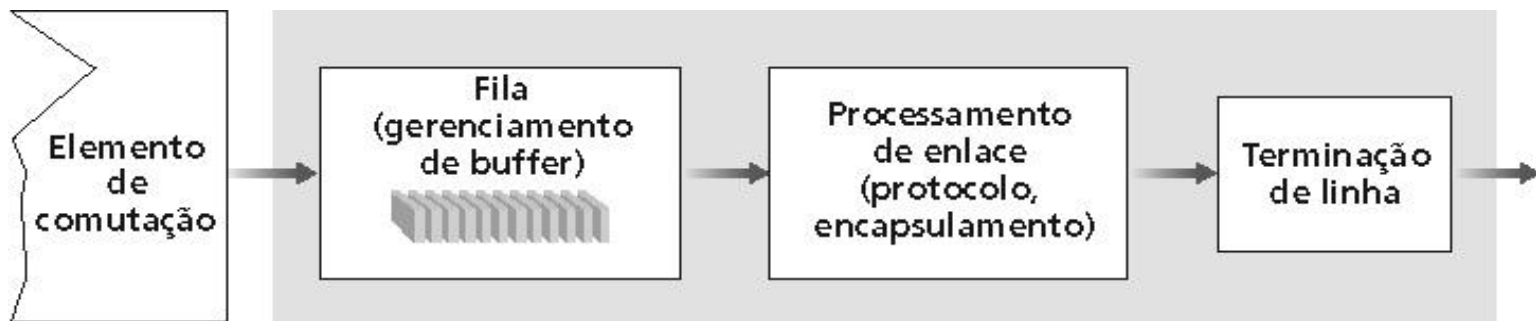
- Datagrama da memória da porta de entrada é transferido para a memória da porta de saída via um barramento compartilhado
  - Não há intervenção do processador de roteamento
- Disputa (contenção) pelo barramento
  - Taxa de comutação limitada pela largura de banda do barramento
- Caso o barramento esteja ocupado
  - Pacotes são enfileirados na porta de entrada

# Comutação por Rede de Interconexão (Crossbar)

- Reduz a disputa pelo acesso ao barramento
  - Disputa passa a ser "por porta de saída"
- Define uma rede de interconexões com  $2N$  barramentos
  - Interconecta  $N$  portas de entrada a  $N$  portas de saída
- Caso um barramento esteja ocupado
  - Pacotes são enfileirados na porta de entrada

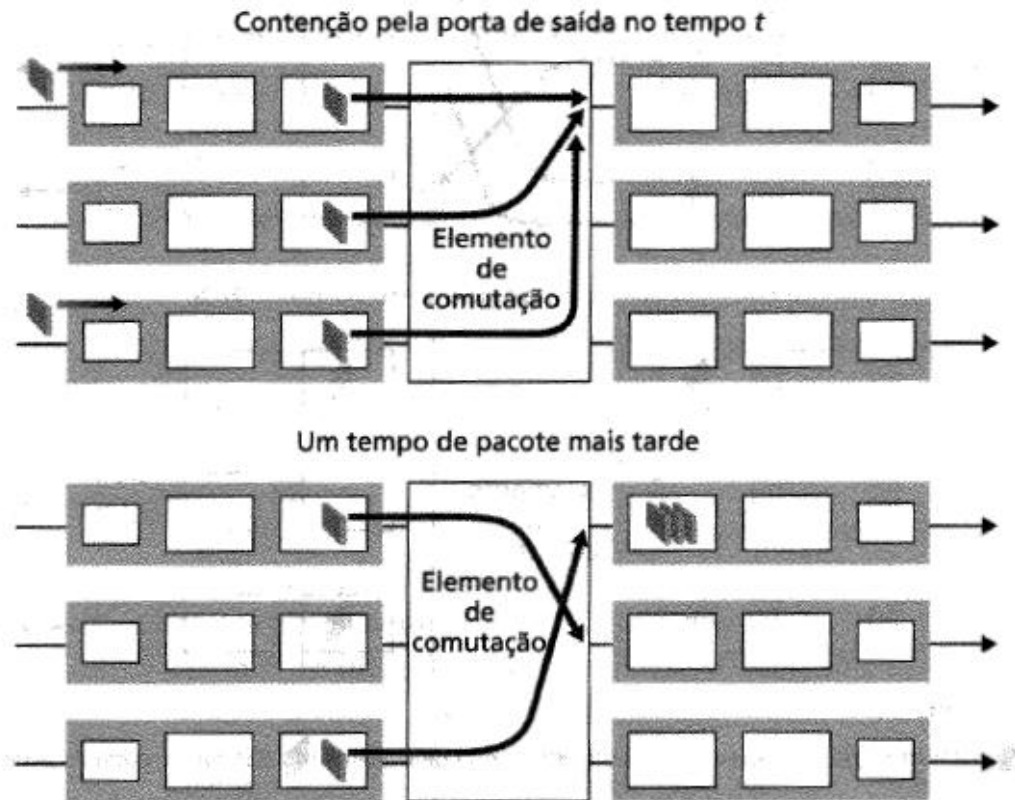
# Funções das Portas de Saída

- Filas
  - Necessárias quando datagramas chegam do elemento de comutação mais rapidamente do que a taxa de transmissão
- Escalonador de pacotes escolhe um dos datagramas enfileirados para transmissão



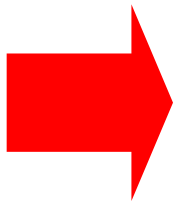
# Como Ocorrem as Filas?

- Portas de saída
  - Usam buffers quando taxa de chegada através do comutador excede taxa de transmissão de saída
  - enfileiramento (retardo) e perdas devido ao transbordo do buffer da porta de saída!



# Como Ocorrem as Filas?

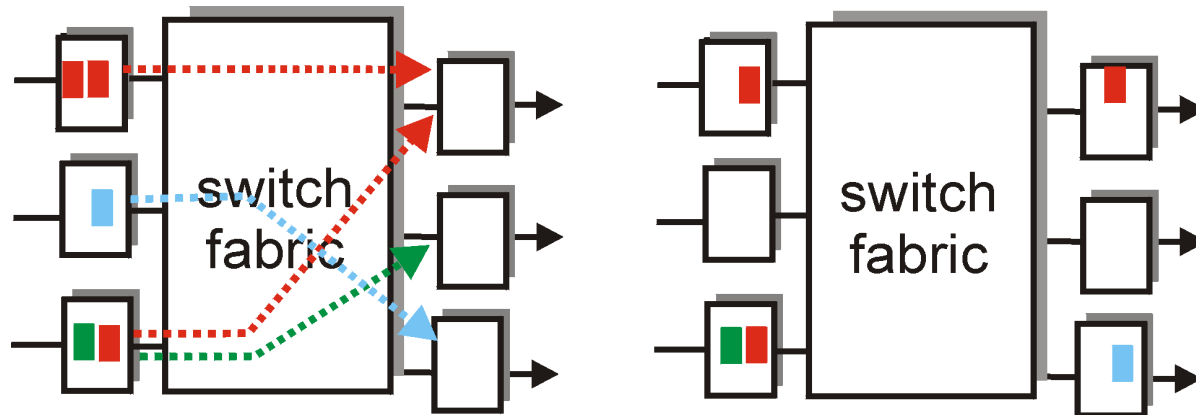
- Portas de entrada
  - Se o elemento de comutação for mais lento do que a soma das portas de entrada juntas
    - Pode haver filas nas portas de entrada
  - Se um datagrama na cabeça da fila impede outros na mesma fila de avançarem
    - Há bloqueio de cabeça de fila



**Como consequência, pode haver retardo de enfileiramento e perdas devido ao transbordo do buffer de entrada!**

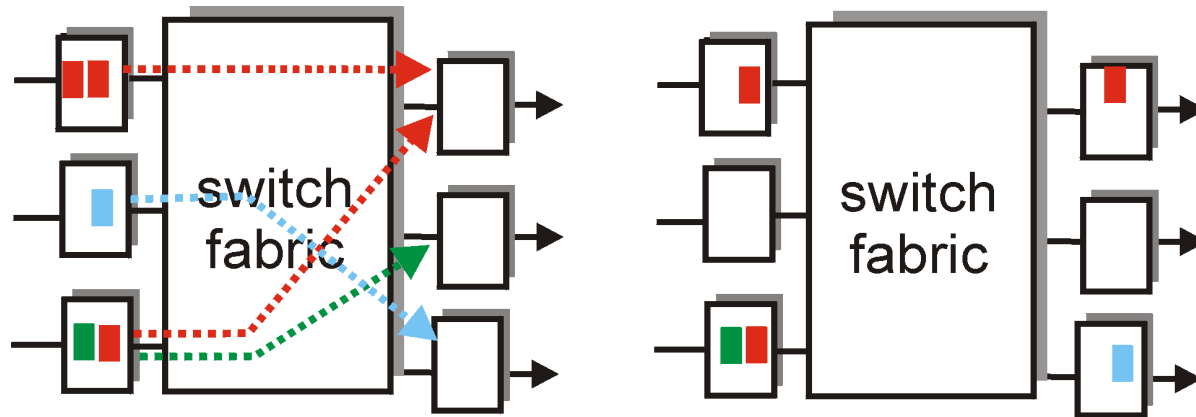


# Como Ocorrem as Filas?



**Contenção na porta de saída no instante  $t$ : somente um pacote vermelho pode ser transferido...**

# Como Ocorrem as Filas?



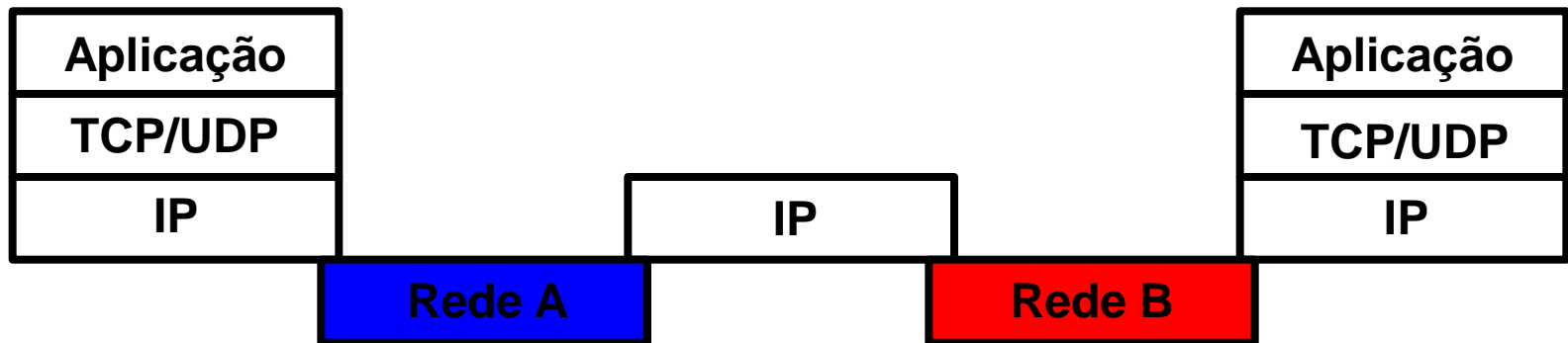
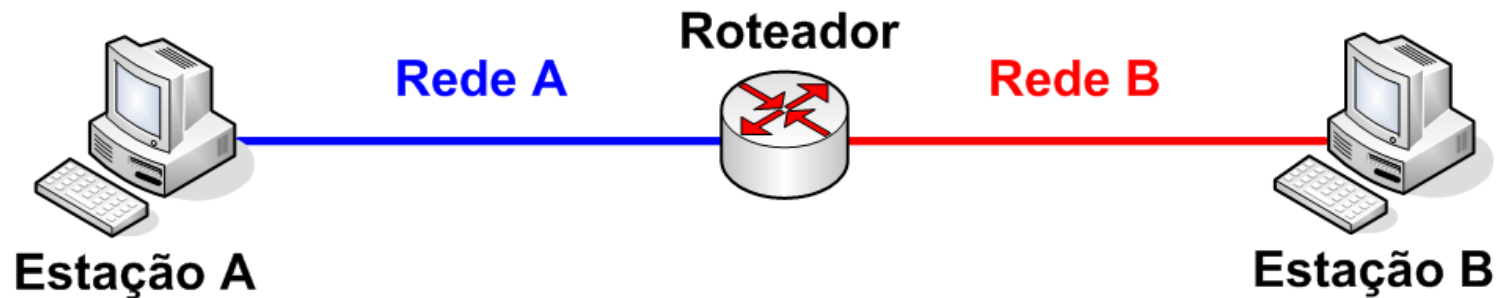
**Pacote verde fica  
bloqueado na cabeceira  
da fila...**

# Internet Protocol (IP)

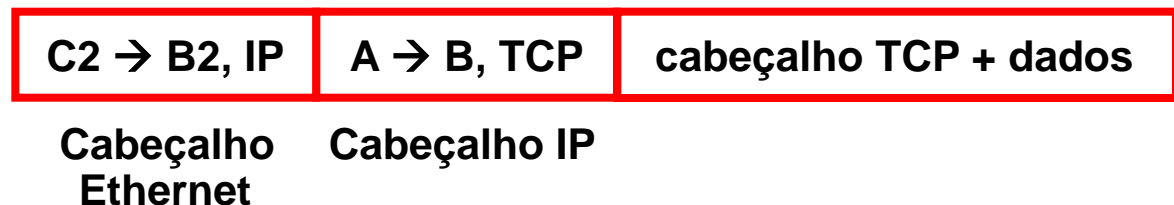
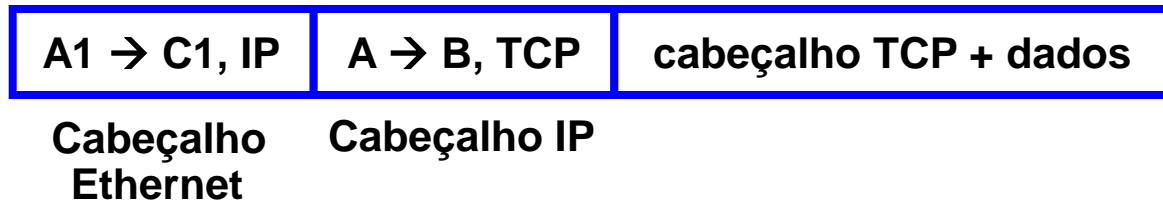
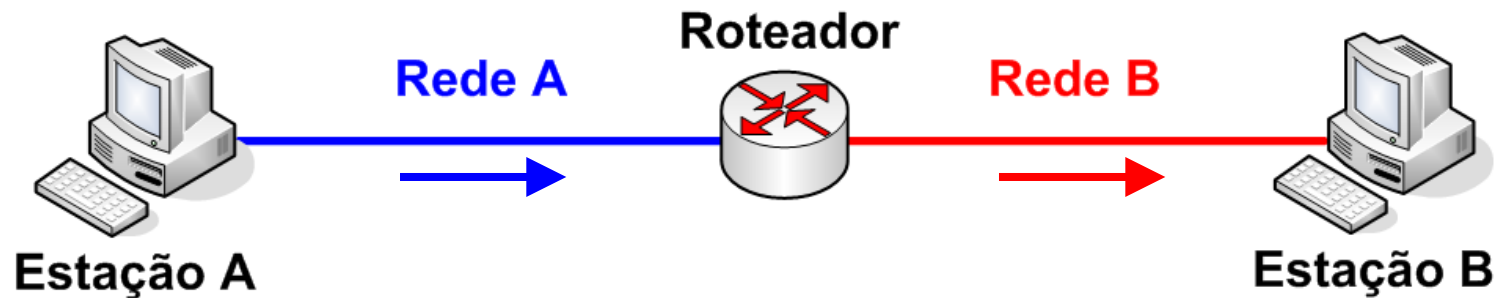
# Internet Protocol

- Definido pela RFC 791
- É o responsável pelo:
  - Encaminhamento de pacotes
    - Não pelo roteamento!
  - Endereçamento e identificação de estações e roteadores
    - Semântica sobrecarregada

# Operação do IP



# Transmissão de um Pacote IP



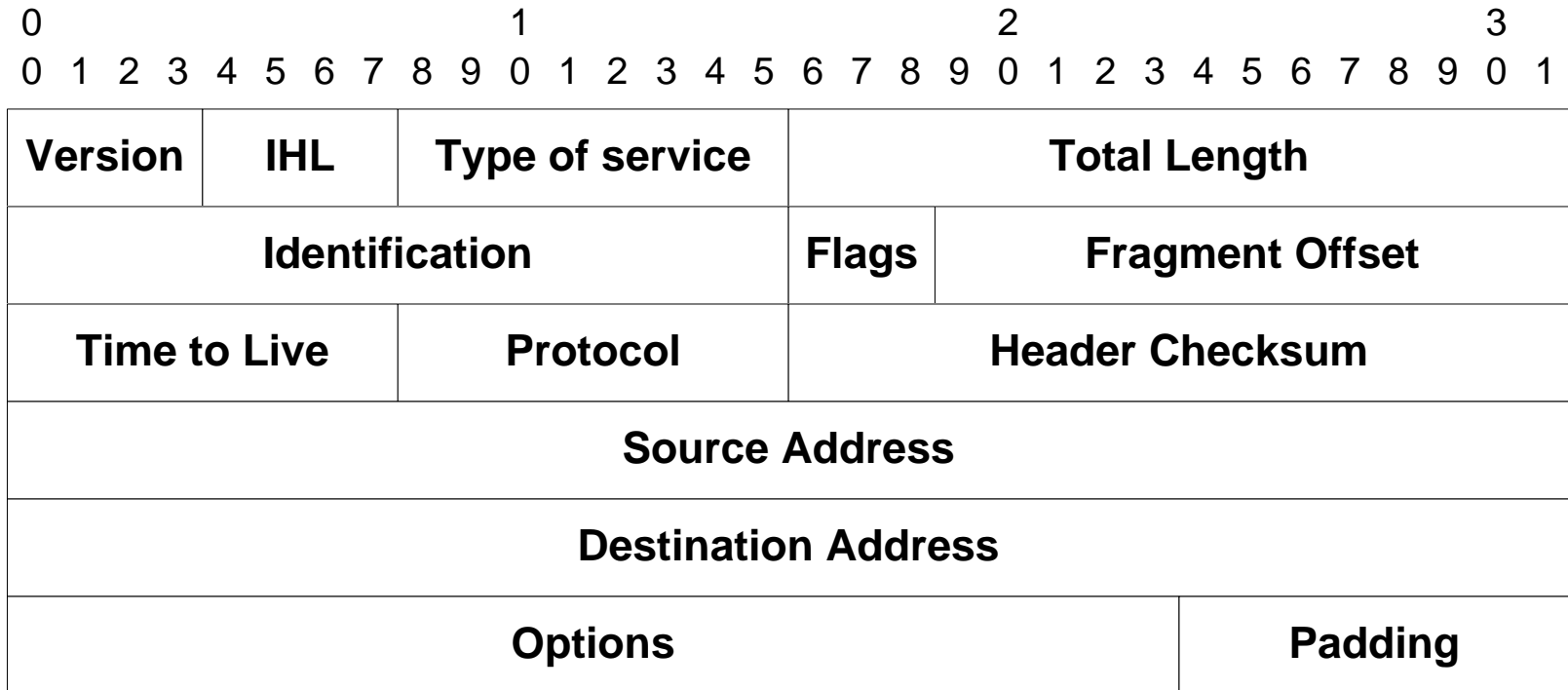
# IPv4

# O Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
<b>Version</b>				<b>IHL</b>				<b>Type of service</b>				<b>Total Length</b>									
<b>Identification</b>								<b>Flags</b>				<b>Fragment Offset</b>									
<b>Time to Live</b>				<b>Protocol</b>				<b>Header Checksum</b>													
<b>Source Address</b>																					
<b>Destination Address</b>																					
<b>Options</b>																<b>Padding</b>					



# O Cabeçalho IP



**Todos os campos possuem tamanho fixo, exceto o campo de opções**

# Campos do Cabeçalho IP

0	1	2	3																												
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service								Total Length															
Identification										Flags				Fragment Offset																	
Time to Live								Protocol								Header Checksum															
Source Address																															
Destination Address																															
Options																								Padding							

- **Versão (4bits)**
  - Versão atual = 4
  - Versão 5 = Protocolo ST-2 (Internet Stream Protocol)
    - **Versão do IP orientado à conexão para tráfego de voz**
  - Versão 6 = "A próxima geração"
  - Versões 7 e 8

# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags		Fragment Offset											
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options																Padding					

- IHL (*Internet header's length*) (4 bits)
  - Comprimento do cabeçalho, em palavras de 32 bits
  - Varia de 5 palavras (quando não há opções) a 15 palavras
    - Ou seja, podem haver 40 bytes de opções, no máximo

# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags				Fragment Offset									
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options														Padding							

- Tipo de serviço (*Type of Service*) (8 bits)
  - Define a precedência e o tipo de roteamento desejado para o pacote
    - Utilizado para qualidade de serviço (QoS)

# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags				Fragment Offset									
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options																Padding					

- Comprimento total (*Total Length*) (16 bits)
  - Comprimento total do pacote, incluindo o cabeçalho
  - Limita o tamanho do pacote a 65.535 bytes
    - Entretanto, os pacotes raramente são maiores que 1.500 bytes

# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
<b>Version</b>				<b>IHL</b>				<b>Type of service</b>				<b>Total Length</b>									
<b>Identification</b>								<b>Flags</b>		<b>Fragment Offset</b>											
<b>Time to Live</b>				<b>Protocol</b>				<b>Header Checksum</b>													
<b>Source Address</b>																					
<b>Destination Address</b>																					
<b>Options</b>																<b>Padding</b>					

- Identification, Flags e Fragment Offset
  - Utilizados no processo de fragmentação e remontagem
    - Identification: identificação do pacote
    - Flag: Indica se o segmento é o último da série
    - Offset: Indica a posição do fragmento no datagrama

# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification										Flags		Fragment Offset									
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options																Padding					

- Tempo de Vida (*Time to Live* -TTL) (8 bits)
  - Tempo de vida máximo do pacote na rede em segundos
    - Um dos objetivos era saber que depois do TTL máximo, nenhum outro pacote daquela comunicação estaria em trânsito
      - Evita-se misturar pacotes de fluxos de dados diferentes

# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags		Fragment Offset											
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options																Padding					

- Tempo de Vida (*Time to Live* -TTL) (8 bits)
  - RFC-791: Um roteador deve sempre decrementar o TTL antes de retransmitir um pacote
    - O TTL deve ser decrementado de 1, se o tempo gasto nas filas e na transmissão ao próximo nó for menor que 1 segundo
    - Ou do número de segundos estimado



# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags				Fragment Offset									
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options																Padding					

- Tempo de Vida (Time to Live -TTL) (8 bits)
  - Na prática, estimar este tempo é difícil e o tempo de transmissão nos enlaces dificilmente ultrapassa 1s
    - **Maioria dos roteadores decrementa o TTL de 1**
  - Se o TTL atinge o valor 0, o pacote deve ser descartado
    - **Sinal que o pacote já trafegou por mais tempo que devia...**

**Valor padrão: TTL = 64**

# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags				Fragment Offset									
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options												Padding									

- Source Address e Destination Address (32 bits cada)
  - Identificam a fonte e o destino do pacote, respectivamente

# Campos do Cabeçalho IP

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags				Fragment Offset									
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options												Padding									

- Protocol (8 bits)
  - Determina o programa para o qual o pacote é passado, no destino

# Campos do Cabeçalho IP

- Diferentes protocolos

Decimal	Sigla	Protocolo
0		Reservado
1	ICMP	Internet Control Message
2	IGMP	Internet Group Management
4	IP	IP em IP (encapsulação)
6	TCP	Transmission Control

Decimal	Sigla	Protocolo
17	UDP	User Datagram
29	ISO-TP4	ISO Transport Prot Class 4
80	ISO-IP	ISO Internet Protocol (CLNP)
89	OSPF	Open Shortest Path First
255		Reservado

# Campos do Cabeçalho IP

0				1				2				3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length																			
Identification								Flags				Fragment Offset																			
Time to Live				Protocol				Header Checksum																							
Source Address																															
Destination Address																															
Options																								Padding							

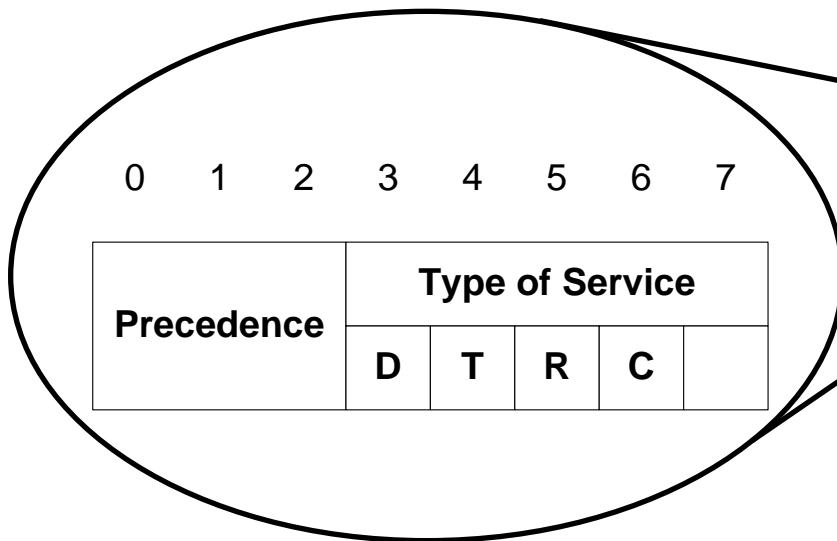
- Header Checksum (16 bits)
  - Proteção do cabeçalho contra erros
  - Muda a cada salto já que o TTL é decrementado e campo de opções pode ser alterado

# Campos do Cabeçalho IP

- *Header Checksum*
  - Calculado como:
    - Complemento a 1 da soma de todas as palavras de 16 bits do cabeçalho
      - Considera os bits do *checksum* em 0
      - Considera o campo de opção
  - Compromisso
    - Não protege contra inserção de palavras em zero (16 bits iguais a zero) ou inversão de palavras...
    - Mas é de simples implementação
  - Calculado a cada salto
    - Caso a verificação falhe, a mensagem é descartada
    - Se não falhar, o *checksum* é recalculado
      - Campo TTL é decrementado a cada salto

# Precedência e Tipo de Serviço

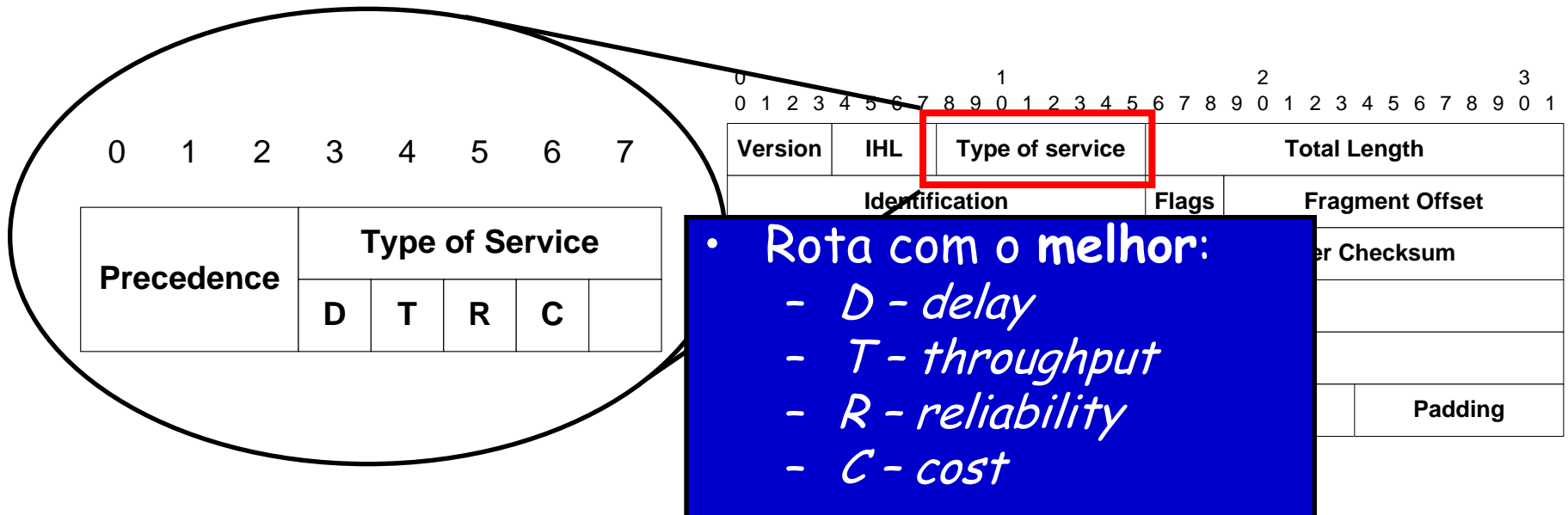
- *Precedence (3 bits)*
  - Indica a prioridade de transmissão do pacote em fila
  - Valores maiores, maior prioridade
  - RFC791 diz que a precedência é válida apenas dentro de uma rede
    - **Evita usuários mal-intencionados**



0	1	2	3
0 1 2 3 4 5 6 7	8 9 0 1 2 3 4 5 6 7 8 9 0 1	2	3
Version	IHL	Type of service	Total Length
Identification		Flags	Fragment Offset
Time to Live	Protocol	Header Checksum	
Source Address			
Destination Address			
Options			Padding

# Precedência e Tipo de Serviço

- *Type of Service (5 bits)*
  - Útil quando existem múltiplas rotas
  - Indicação para o roteamento
    - **Nunca são utilizados mais de um campo**
      - **Combinação ilegal**





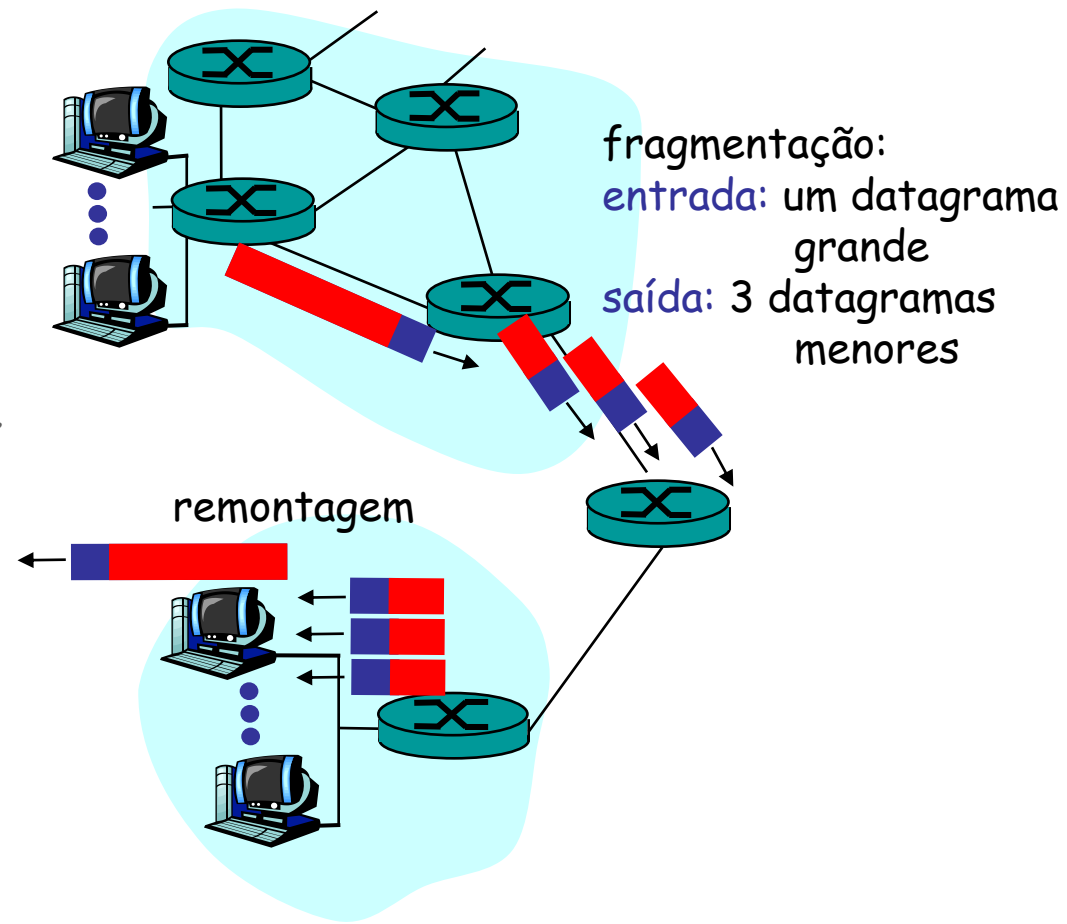
# Fragmentação e Remontagem

- A fragmentação é necessária
  - Roteador conecta duas tecnologias de rede diferentes
    - Cada uma possui um tamanho máximo de pacote
      - Ex.: Rede com alta perda → pacotes devem ser pequenos
      - Rede com baixa perda → pacotes podem ser grandes
  - Menor MTU do caminho não é conhecida
    - Comumente se utiliza a MTU da rede da fonte

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags				Fragment Offset									
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options												Padding									

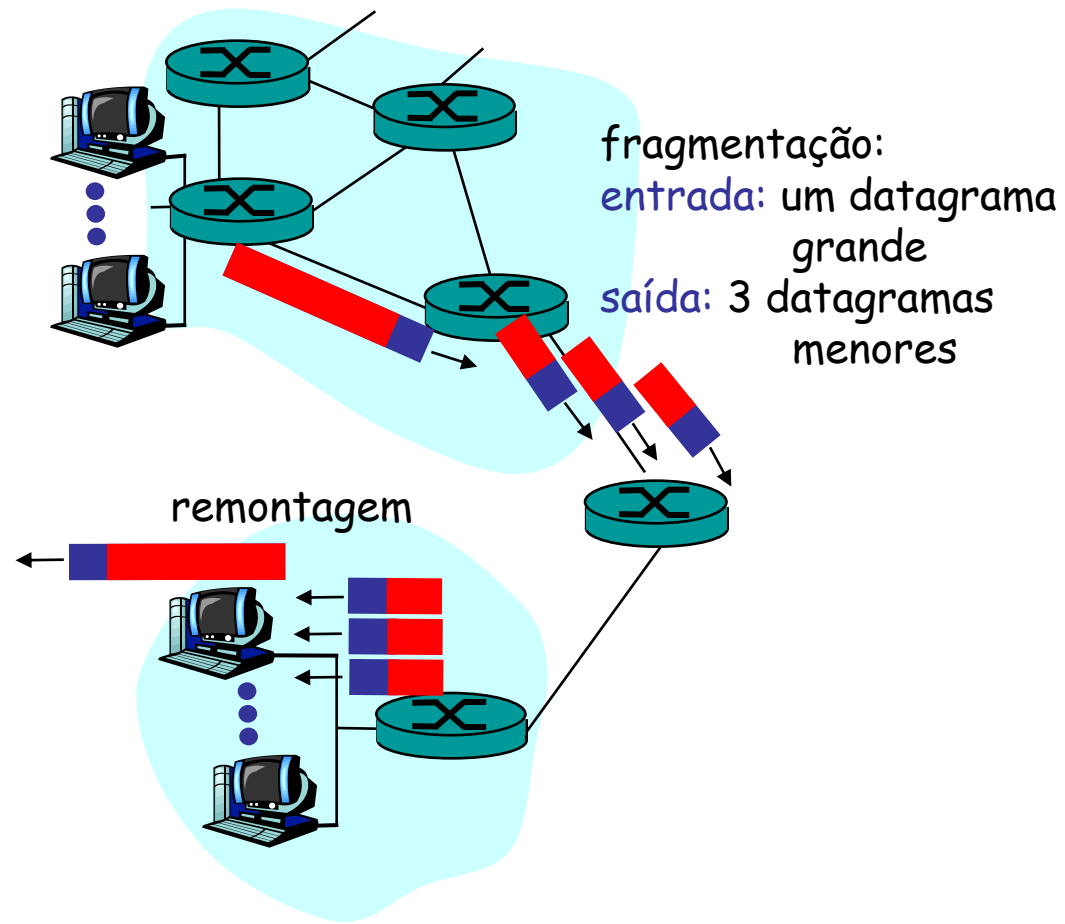
# Fragmentação e Remontagem

- Cada enlace de rede tem MTU (*max. transmission unit*) - maior tamanho possível de quadro neste enlace
  - Tipos diferentes de enlace têm MTUs diferentes



# Fragmentação e Remontagem

- 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



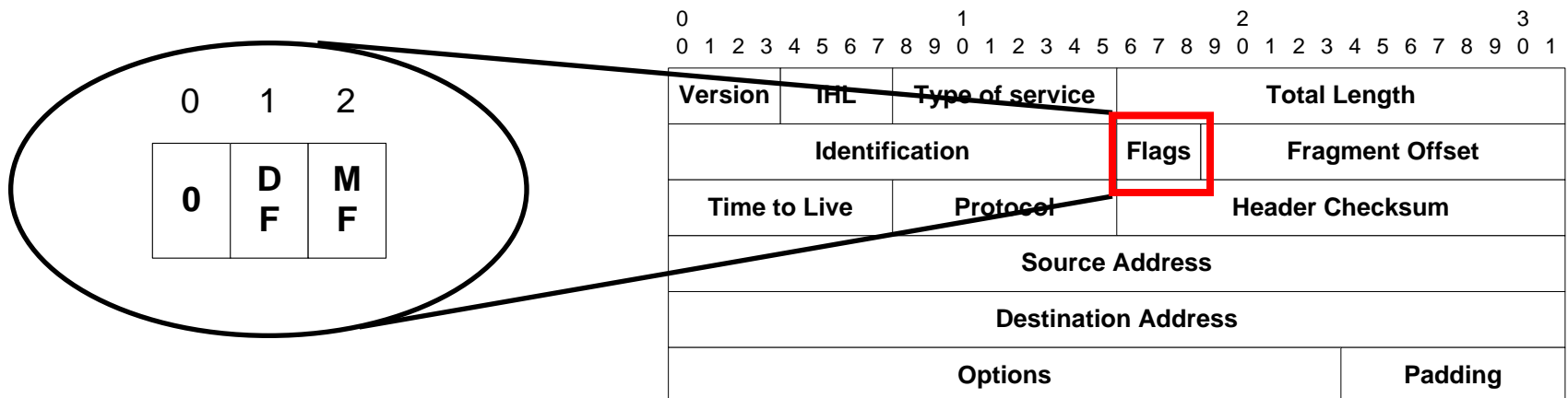
# Fragmentação e Remontagem

- *Identification* (16 bits)
  - Junto ao campo endereço de origem, identifica a qual pacote pertence o fragmento
- *Fragment Offset* (13 bits)
  - Identifica a posição do fragmento no pacote
  - Palavras de 8 bits

0				1				2				3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Version				IHL				Type of service				Total Length									
Identification								Flags				Fragment Offset									
Time to Live				Protocol				Header Checksum													
Source Address																					
Destination Address																					
Options																Padding					





# Fragmentação e Remontagem

- *Flags* (3 bits)
  - Informa se o pacote pode ser fragmentado (DF) e se ainda existem mais fragmentos a serem recebidos (MF)
    - Bit 0 - reservado
    - Bit 1 - *don't fragment* (DF)
    - Bit 2 - *more fragments* (MF)



# Fragmentação e Remontagem

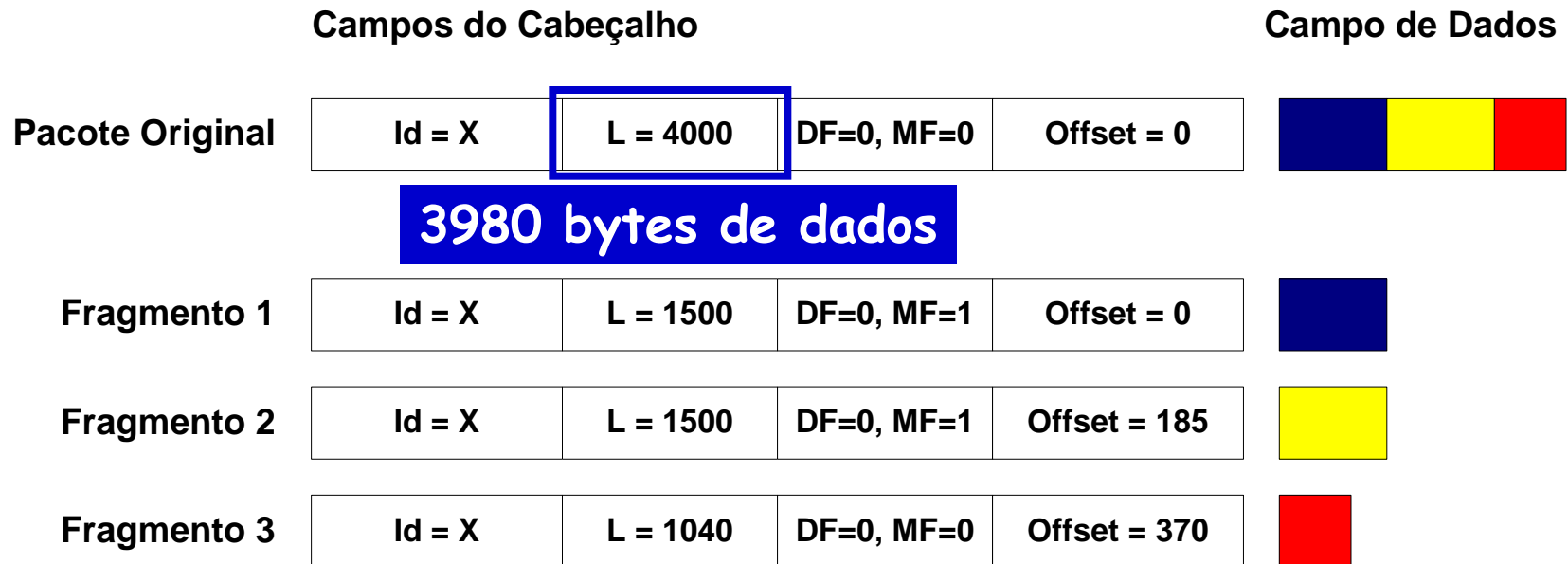
- Cada fragmento possui um cabeçalho completo
  - Igual ao do pacote original, exceto pelos campos de **comprimento**, **offset** e o bit **MF**

	Campos do Cabeçalho				Campo de Dados
Pacote Original	Id = X	L = 4000	DF=0, MF=0	Offset = 0	
Fragmento 1	Id = X	L = 1500	DF=0, MF=1	Offset = 0	
Fragmento 2	Id = X	L = 1500	DF=0, MF=1	Offset = 185	
Fragmento 3	Id = X	L = 1040	DF=0, MF=0	Offset = 370	

**Datagrama de 4000 bytes e MTU = 1500 bytes**

# Fragmentação e Remontagem

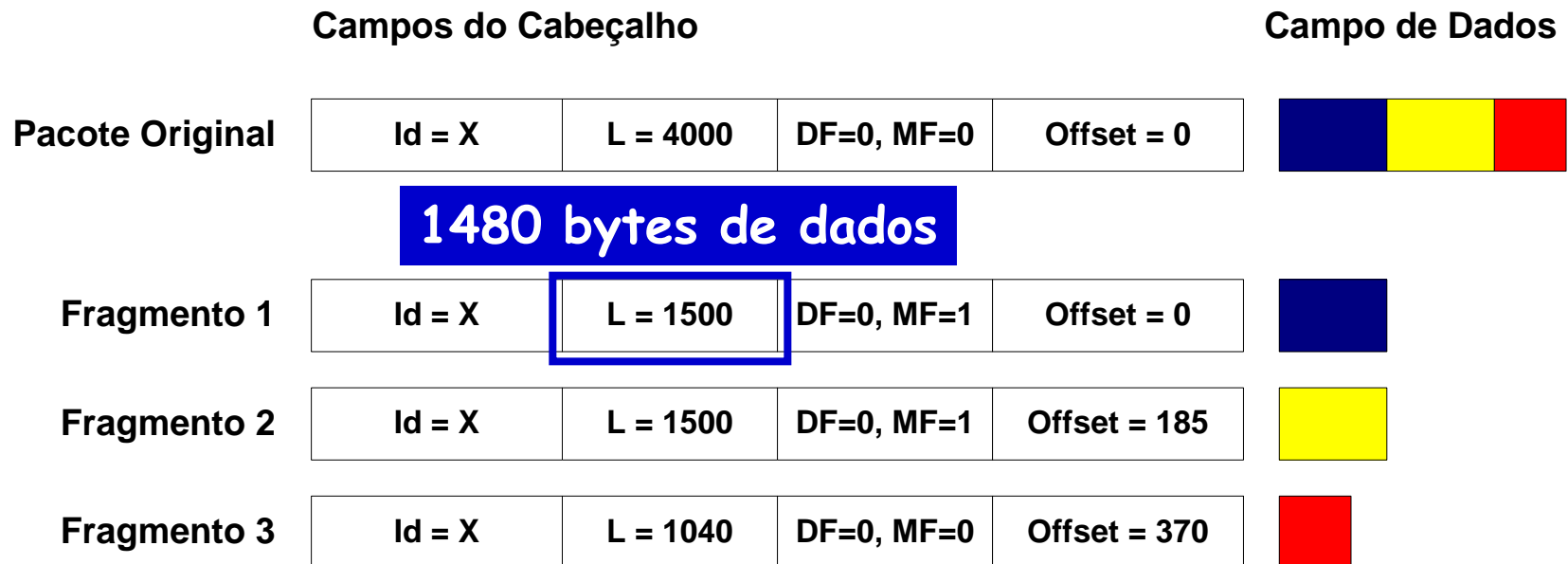
- Cada fragmento possui um cabeçalho completo
  - Igual ao do pacote original, exceto pelos campos de comprimento, *offset* e o bit MF



**Datagrama de 4000 bytes e MTU = 1500 bytes**

# Fragmentação e Remontagem

- Cada fragmento possui um cabeçalho completo
  - Igual ao do pacote original, exceto pelos campos de comprimento, *offset* e o bit MF

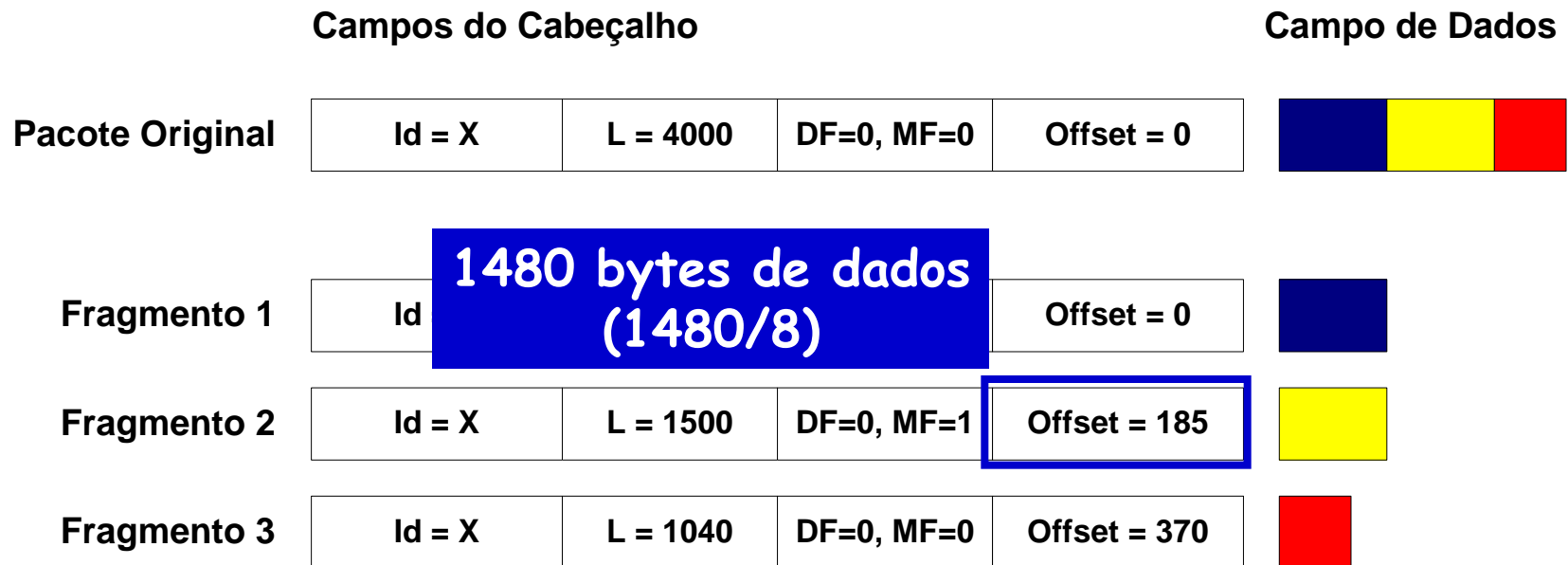


**Datagrama de 4000 bytes e MTU = 1500 bytes**



# Fragmentação e Remontagem





- Cada fragmento possui um cabeçalho completo
  - Igual ao do pacote original, exceto pelos campos de comprimento, *offset* e o bit MF



Datagrama de 4000 bytes e MTU = 1500 bytes

# Fragmentação e Remontagem






- Cada fragmento possui um cabeçalho completo
  - Igual ao do pacote original, exceto pelos campos de comprimento, *offset* e o bit MF

	Campos do Cabeçalho				Campo de Dados
Pacote Original	Id = X	L = 4000	DF=0, MF=0	Offset = 0	
Fragmento 1	Id = X	L = 1500	DF=0, MF=1	Offset = 0	
Fragmento 2	Id = X	L = 1500	DF=0, MF=1	Offset = 185	
Fragmento 3	Id = X	L = 1040	DF=0, MF=0	Offset = 370	

 O bit MF é sempre 1, exceto no último fragmento

# Fragmentação e Remontagem

- Em caso de nova fragmentação
  - *MF* e *offset* são calculados com relação ao pacote original

	Campos do Cabeçalho				Campo de Dados
Fragmento 2	Id = X	L = 1500	DF=0, MF=1	Offset = 185	
Fragmento 2a	Id = X	L = 500	DF=0, MF=1	Offset = 185	
Fragmento 2b	Id = X	L = 500	DF=0, MF=1	Offset = 245	
Fragmento 2c	Id = X	L = 500	DF=0, MF=1	Offset = 305	
Fragmento 2d	Id = X	L = 60	DF=0, MF=1	Offset = 310	

# Fragmentação e Remontagem

- O campo identificação (16 bits) associado ao endereço de origem identifica a qual pacote pertence o fragmento
- Pacotes são remontados no destino
  - O receptor deve "expirar" pacotes **parcialmente** remontados, após um certo período de espera
    - **Ex.: decrementando o campo TTL a cada segundo**
  - O emissor só pode reutilizar um identificador após o período igual ao TTL utilizado

# Fragmentação e Remontagem

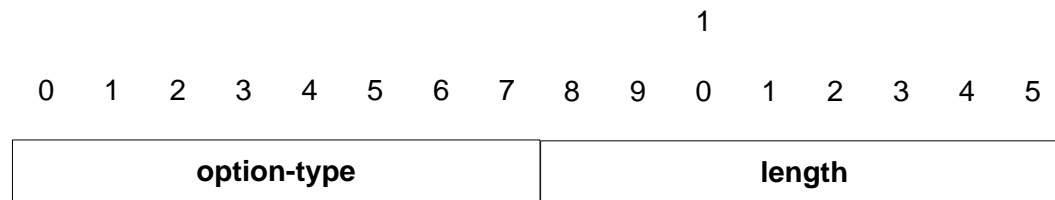
- A fragmentação é ineficiente combinada com o TCP
  - Perda de um fragmento implica retransmissão do pacote inteiro
- A memória dos roteadores pode ser desperdiçada
  - Os fragmentos de um determinado pacote ficam armazenados em buffers antes de serem encaminhados

# Como Evitar a Fragmentação?

- O TCP implementa um mecanismo de descoberta da MTU (*Maximum Transmission Unit*) do caminho
  - Tentativas com diferentes tamanhos de pacote e com o campo DF (Don't Fragment) em 1
    - O TCP utiliza como MTU o maior tamanho entregue

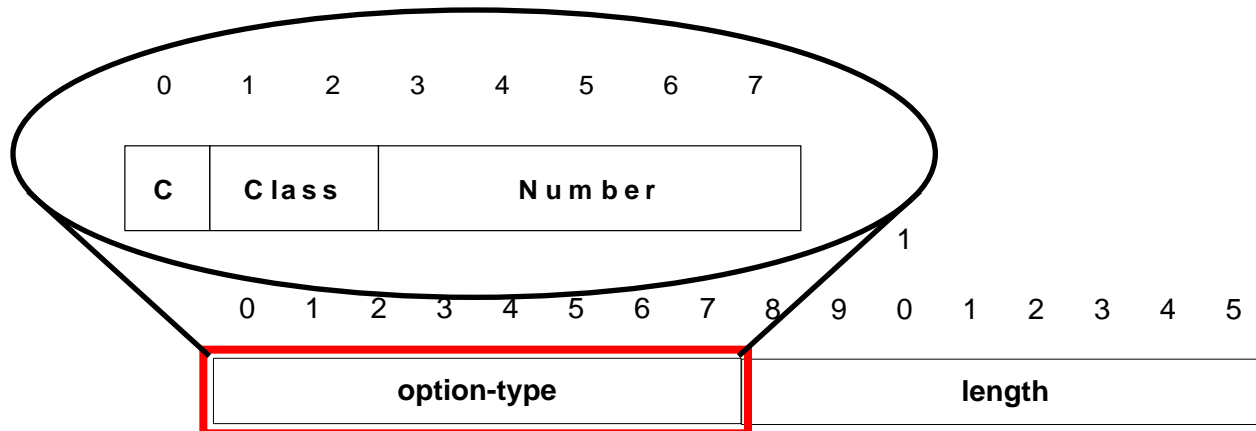
# Opções do IP

- Definido para criação de funcionalidades especiais, através do roteamento específico de alguns pacotes
- *Options*
  - Pode transportar vários parâmetros
  - Cada opção começa por um byte de "tipo de opção"
  - O segundo byte normalmente indica o comprimento da opção



# Opções do IP

- Flag *C (Copied)*
  - Indica que a opção deve ser copiada em todos os fragmentos ou apenas no primeiro
- Class
  - 0: opções de controle e 2: opções de debug e medidas
- Number
  - Identifica uma opção dentro de cada classe





# Opções do IP

Classe	Número	Compr.	Significado
0	0	-	<b>End of Option list.</b> Indica o fim da lista de opções, possui apenas 1 byte. Não há byte de comprimento.
0	1	-	<b>No Operation.</b> Possui apenas 1 byte. Não há byte de comprimento.
0	2	11	<b>Security.</b> Utilizada para carregar parâmetros de segurança definidos pelo dep. de defesa americano.
0	3	var.	<b>Loose Source Routing.</b> Utilizada para rotear o pacote IP de acordo com a informação fornecida pela fonte.
0	7	var.	<b>Record Route.</b> Utilizada para registrar a rota atravessada pelo pacote IP.
0	8	4	<b>Stream ID.</b> Utilizada para carregar o identificador do stream.
0	9	var.	<b>Strict Source Routing.</b> Utilizada para rotear o pacote IP de acordo com a informação fornecida pela fonte.
2	4	var.	<b>Internet Timestamp.</b>

# Opções do IP

Classe	Número	Compr.	Significado
0	0	-	<b>End of Option list.</b> Indica o fim da lista de opções, possui apenas 1 byte. Não há byte de comprimento.
0	1	-	<b>No Operation.</b> Possui apenas 1 byte. Não há byte de comprimento.
0	2	11	<b>Security.</b> Utilizada para carregar parâmetros de segurança definidos pelo dep. de defesa americano.
0	3	var.	<b>Loose Source Routing.</b> Utilizada para rotear o pacote IP de acordo com a informação fornecida pela fonte.
0	7	var.	<b>Record Route.</b> Utilizada para registrar a rota atravessada pelo pacote IP.
0	8	4	<b>Stream ID.</b> Utilizada para carregar o identificador do stream.
0	9	var.	<b>Strict Source Routing.</b> Utilizada para rotear o pacote IP de acordo com a informação fornecida pela fonte.
2	4	var.	<b>Internet Timestamp.</b>

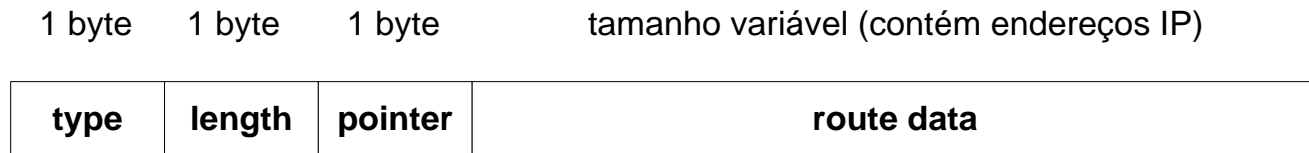
# Opções do IP

- *No operation*
  - Utilizada para enchimento entre opções, de forma que o início da opção esteja alinhado em 32 bits
- *End of option*
  - Indica o ponto onde a opção termina, mesmo se o campo IHL indicar mais espaço alocado para opções
- **A maioria das opções não é usada**
  - Stream ID foi usada apenas no experimento Satnet
  - Security codifica necessidades militares dos anos 70
  - *Timestamp e route record* visavam serviços que o programa traceroute implementa

# Roteamento pela Fonte

- Caminho do pacote é definido no nó de origem
- Duas possibilidades
  - *Strict Routing*
    - Define o caminho completo
  - *Loose Routing*
    - Define alguns nós do caminho

# Roteamento pela Fonte



## Campo de opções

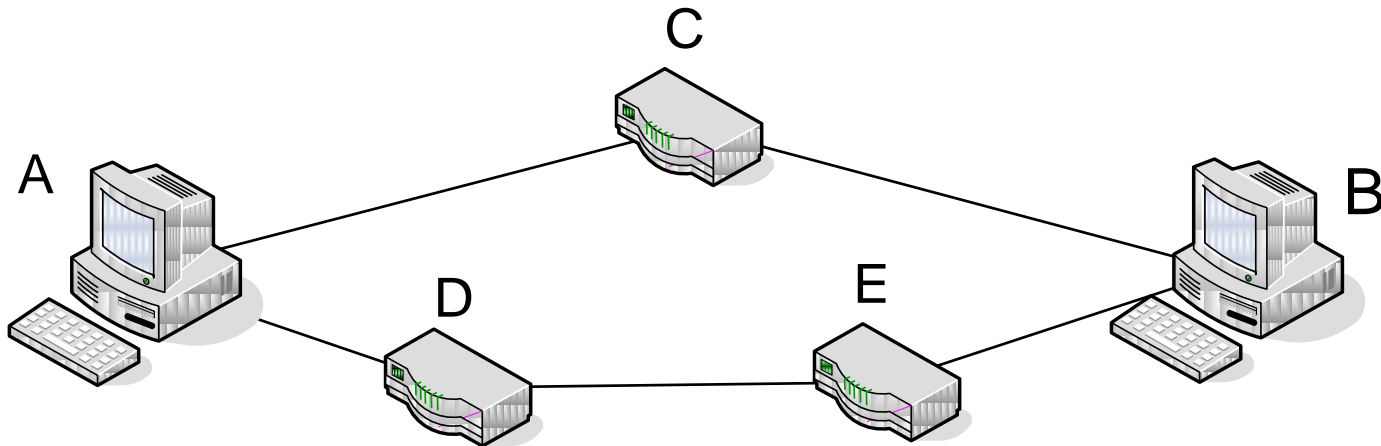
- *Route data*
  - Contém a lista de endereços pelos quais o pacote deve passar
- *Pointer*
  - Aponta para o próximo endereço da lista a ser utilizado

# Roteamento pela Fonte

- Funcionamento
  - O campo *Destination Address* do cabeçalho possui o endereço IP do próximo nó pelo qual o pacote deve passar
  - Quando este destino é atingido, a opção é examinada
  - O campo *pointer* indica um número de octetos a partir do início da opção, de onde deve ser lido o próximo endereço
  - Se *pointer* maior que o comprimento da opção
    - O destino final foi atingido

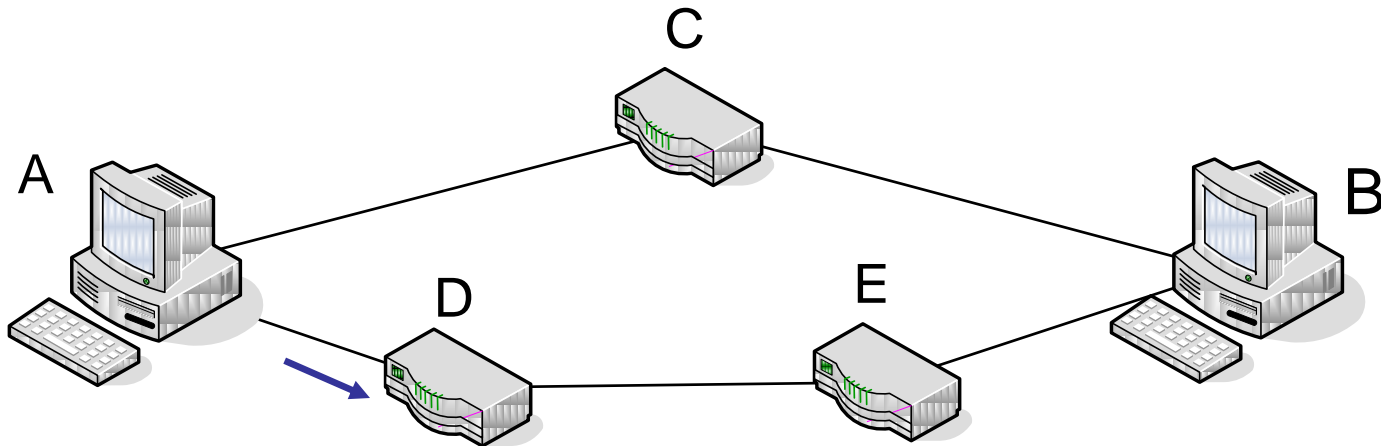
# Como Evitar o *Source Routing*?

- Como enviar um pacote de A para B, passando pelos roteadores D e E
  - Encapsulamento IP sobre IP → tunelamento



# Como Evitar o *Source Routing*?

- Como enviar um pacote de A para B, passando pelos roteadores D e E
  - Encapsulamento IP sobre IP → tunelamento



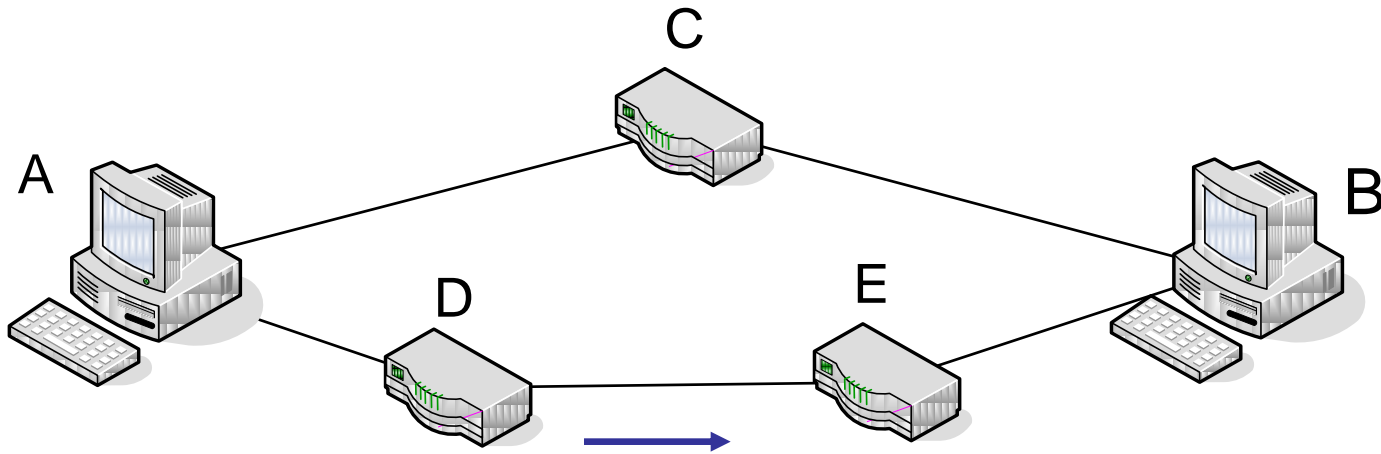
A → D, IP	A → E, IP	A → B, TCP	cabeçalho TCP + dados
-----------	-----------	------------	-----------------------

Cabeçalho IP(1) Cabeçalho IP(2) Cabeçalho IP(3)



# Como Evitar o *Source Routing*?

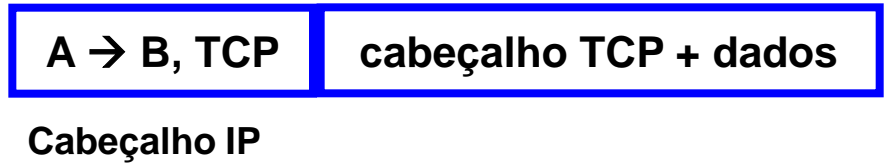
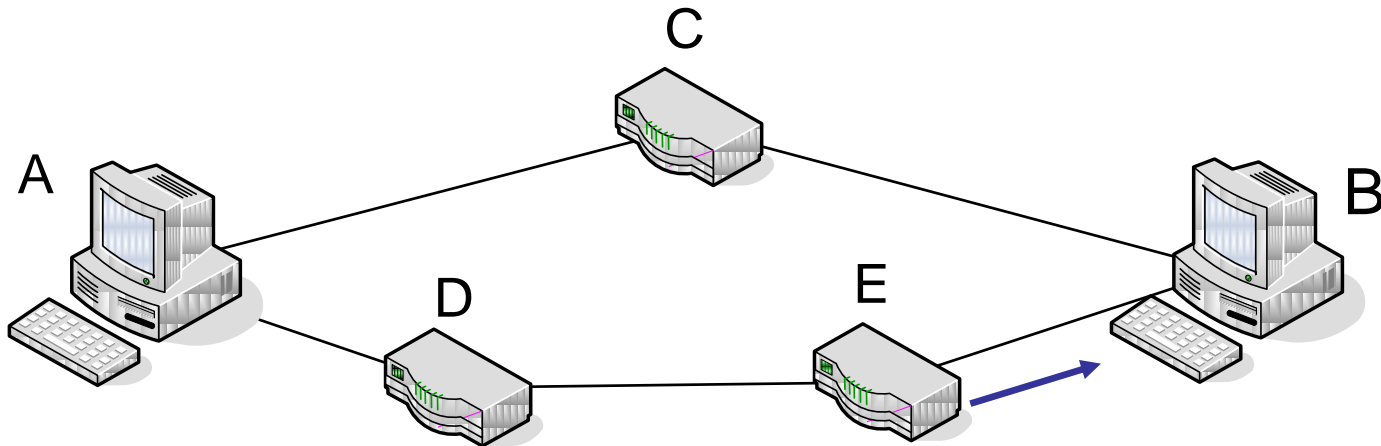
- Como enviar um pacote de A para B, passando pelos roteadores D e E
  - Encapsulamento IP sobre IP → tunelamento



Cabeçalho IP(1) Cabeçalho IP(2)

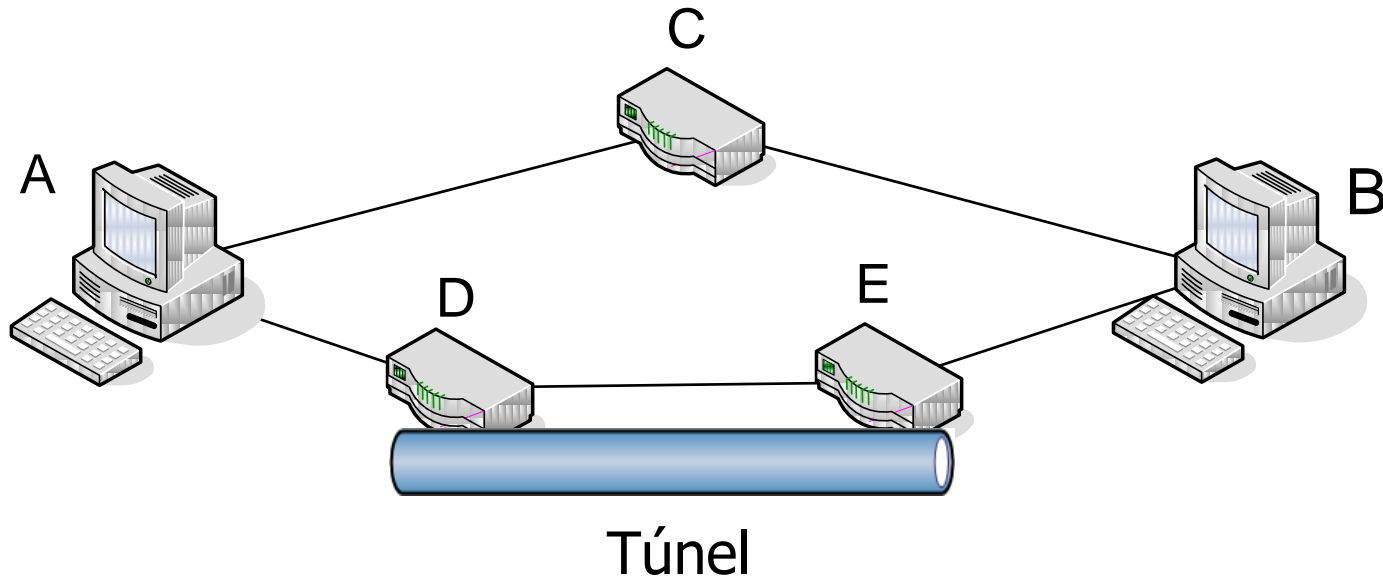
# Como Evitar o *Source Routing*?

- Como enviar um pacote de A para B, passando pelos roteadores D e E
  - Encapsulamento IP sobre IP → tunelamento



# Como Evitar o *Source Routing*?

- Como enviar um pacote de A para B, passando pelos roteadores D e E
  - Encapsulamento IP sobre IP → tunelamento



**Tunelamento é usado para realizar encapsulamento de dados de "mesma camada"**

# Processamento do Cabeçalho IP

- Operações para encaminhar um pacote
  1. Verificação da versão, do *checksum*, tamanho do pacote, e leitura das opções (se houver)
  2. Consultar a tabela de roteamento para o destino e tipo de serviço do pacote
  3. Obter a interface e endereço no meio físico

# Processamento do Cabeçalho IP

- Operações para encaminhar um pacote
  1. Verificação da versão, do *checksum*, tamanho do pacote, e leitura das opções (se houver)
  2. Consultar a tabela de roteamento para o destino e tipo de serviço do pacote
  3. Obter a interface e endereço no meio físico



**Número grande de operações!**

**Como encaminhar pacotes a taxas da ordem de Gb/s?**

# Processamento do Cabeçalho IP

- Roteadores otimizam as operações mais comuns (*fast-path*)
  - Ex.: *caches* com rotas mais utilizadas, processamento em paralelo de múltiplos campos
- Pacotes **sem opções**
  - Possuem cabeçalho de tamanho fixo
    - *Passam pelo fast-path*
- Pacotes **com opções**
  - Seguem o caminho "normal"
    - *Além disso, em alguns roteadores, pacotes com opções possuem menos prioridade para aumentar o desempenho global*

# Endereçamento IP

- Cada **interface** de rede é identificada por um **endereço IP** de 32 bits

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

# Endereçamento IP

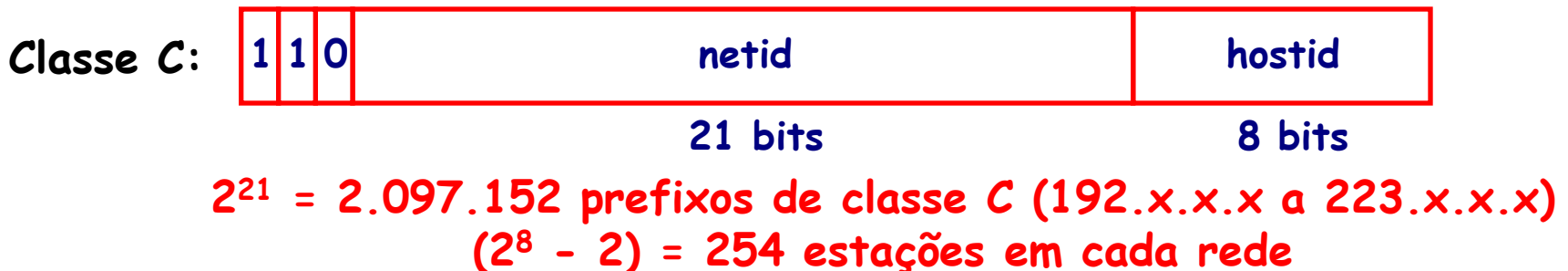
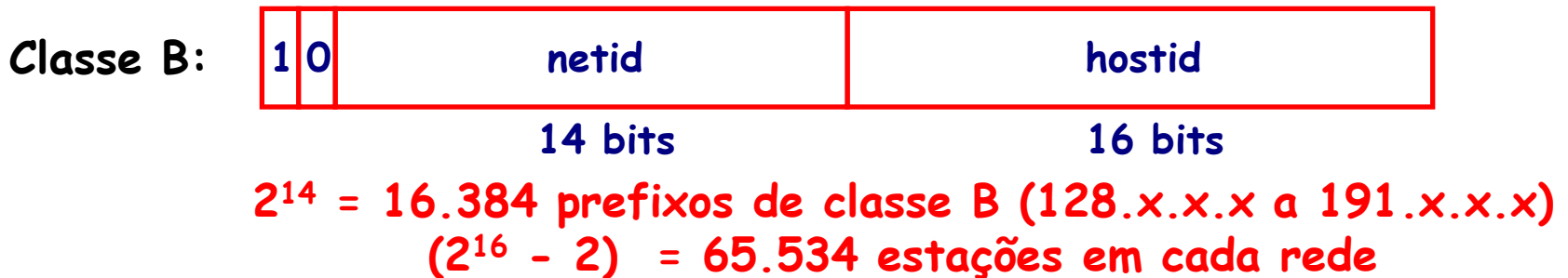
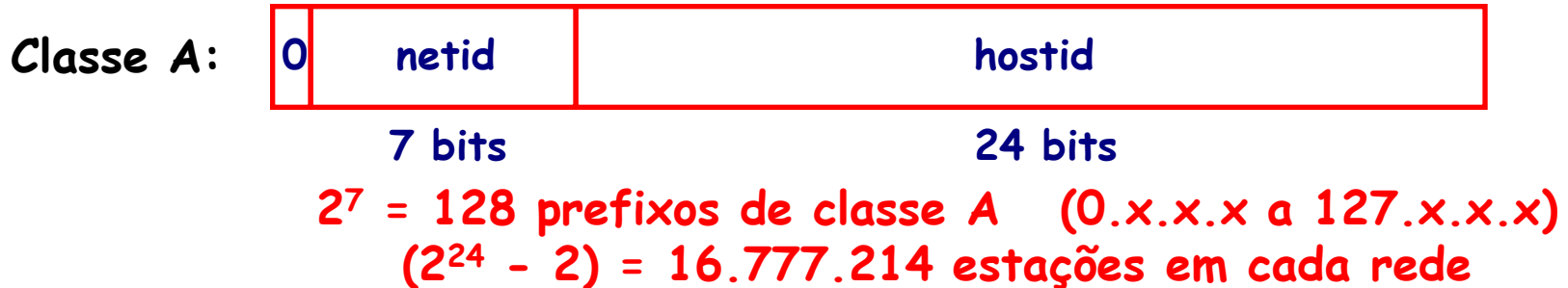
- Formato do endereço IP
  - Dividido em duas partes:
    - "identificador de rede" e "identificador de estação"
- 3 classes de "números de rede": A, B e C
- Mais tarde, classe D definida para endereços *multicast*
- A classe E possui endereços reservados para utilização experimental



# Classes de Endereços IP

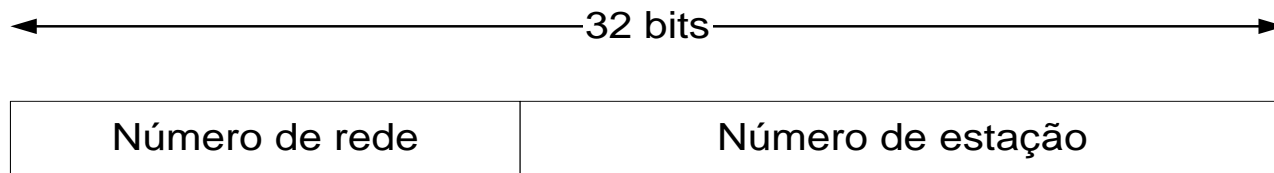
<b>Classe</b>	<b>Bits mais significativos</b>	<b>Formato</b>	
<b>A</b>	<b>0</b>	<b>7 bits de redes</b>	<b>24 bits de estações</b>
<b>B</b>	<b>10</b>	<b>14 bits de redes</b>	<b>16 bits de estações</b>
<b>C</b>	<b>110</b>	<b>21 bits de redes</b>	<b>8 bits de estações</b>
<b>D</b>	<b>1110</b>	<b>28 bits de endereços de grupo multicast</b>	
<b>E</b>	<b>1111</b>	<b>reservados para testes</b>	

# Classes A, B e C



# Estrutura de Endereçamento

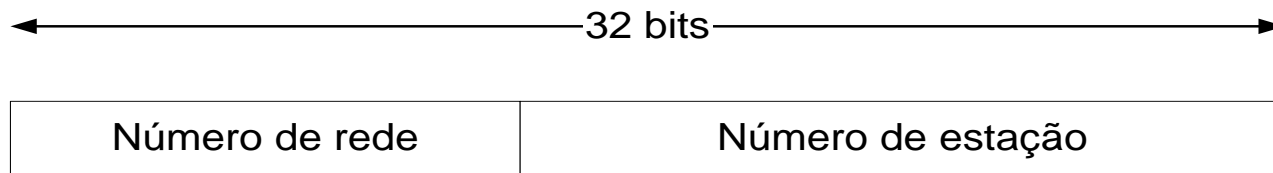
- Quando o IP foi padronizado, em 1981



- Números de rede (*netid*)
  - Alocados pela autoridade de numeração da Internet
- Números de estação (*hostid*)
  - Alocados pelo gerente de rede

# Estrutura de Endereçamento

- Quando o IP foi padronizado, em 1981



- Números de rede (*netid*)
  - Alocados pela autoridade de numeração da Internet
- Números de estação (*hostid*)
  - Alocados pelo gerente de rede

**Unicidade do número de rede + unicidade do número da estação  
→ Garantem a UNICIDADE GLOBAL do endereço IP**

# Problema das Classes de Endereço

- Número fixo de redes e estações por rede
  - Classe A
    - Número pequeno de redes
    - Número excessivo de estações por rede
  - Classe C
    - Número pequeno de estações por rede
    - Número excessivo de redes
- Resultado



**Esgotamento da classe B!**

# *Classless Inter-Domain Routing architecture (CIDR)*

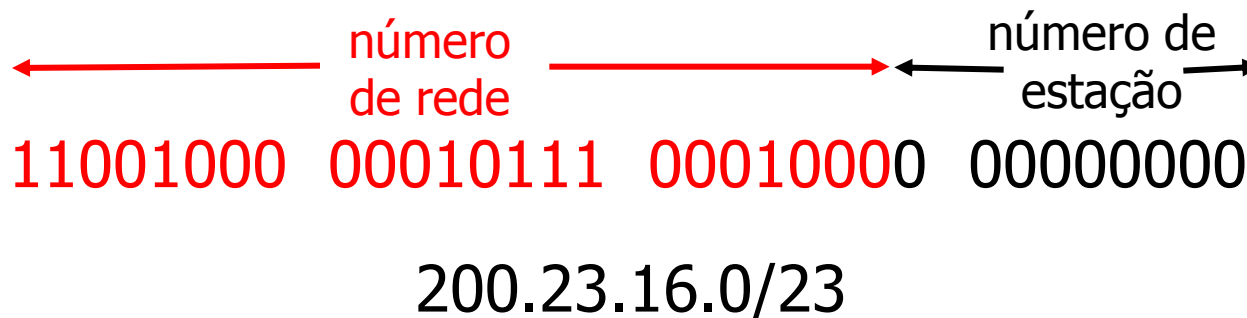
- Acaba com as classes
  - Introduz o conceito de máscara de rede
- Permite
  - Agregação de rotas
    - Aumenta a escalabilidade
      - Reduz o tamanho das tabelas de roteamento
  - Distribuição mais adequada dos endereços IP
    - Resolve o esgotamento dos endereços da classe B
    - Permite melhor planejamento de endereços
      - Número de máquinas vs. número de endereços IP

# Estrutura de Endereçamento CIDR

- Número de rede de comprimento variável

**a . b . c . d / x**

- Os **x** bits mais significativos do endereço são o número de rede → **prefixo**
- Os 32-**x** bits são o número de estação



# Máscaras de Sub-rede

- Uma máscara de sub-rede pode ser representada através da notação:
  - Endereço da rede+sub-rede/<número de bits em 1 da máscara>
    - Ex1.: **192.168.0.0/16**
      - Notação equivalente a dizer que a máscara é **255.255.0.0**
    - Ex2.: **192.168.3.0/26**
      - Notação equivalente a dizer que a máscara é **255.255.255.192**



# Estrutura de Endereçamento CIDR

- Como obter o número de rede/prefixo a partir do endereço IP?

**Prefixo = (Endereço IP) AND (Máscara)**

200.23.16.1/255.255.254.0

<b>endereço</b>	11001000	00010111	00010000	00000001
<b>máscara</b>	11111111	11111111	11111110	00000000
<b>rede</b>	<b>11001000</b>	<b>00010111</b>	<b>00010000</b>	<b>00000000</b>

**(200.23.16.0)**


# Estrutura de Endereçamento CIDR

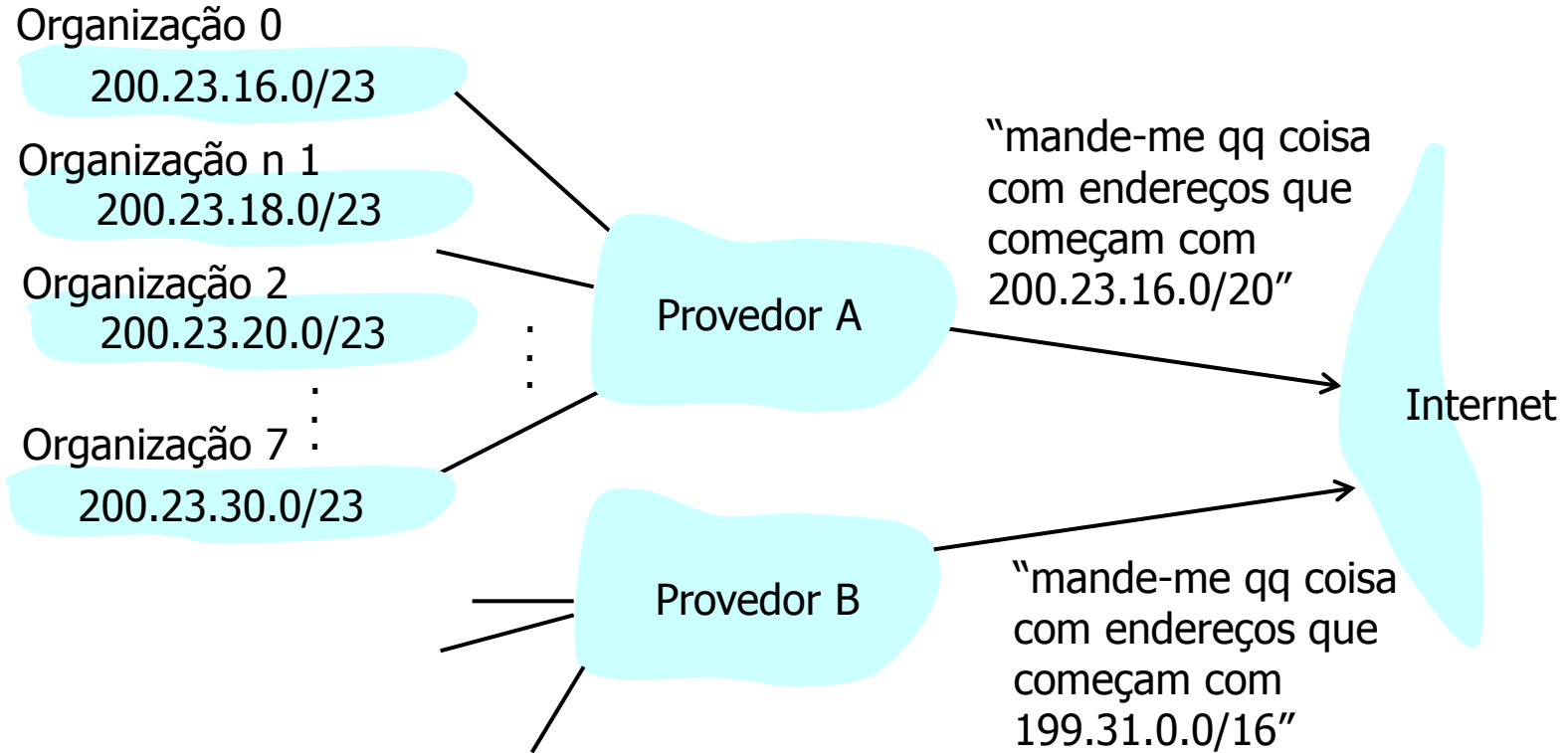
**P:** Como uma sub-rede obtém a parte de rede do endereço IP?

**R:** Recebe uma porção do espaço de endereços do seu ISP (provedor)

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

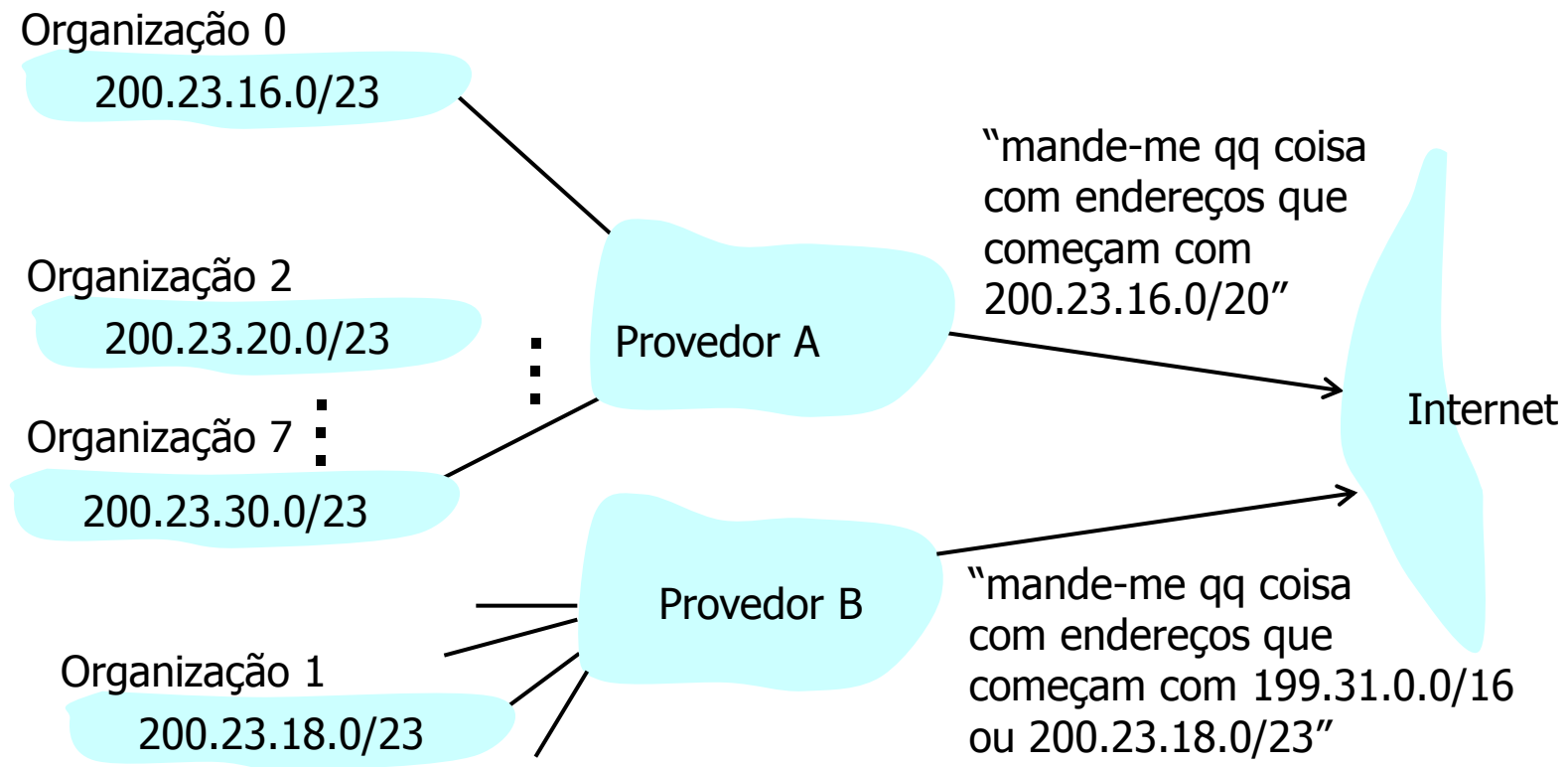
# Endereçamento Hierárquico

Endereçamento hierárquico permite anunciar eficientemente informação sobre rotas  **agregação**



# Endereçamento Hierárquico

Provedor B tem uma rota mais específica para a Organização 1

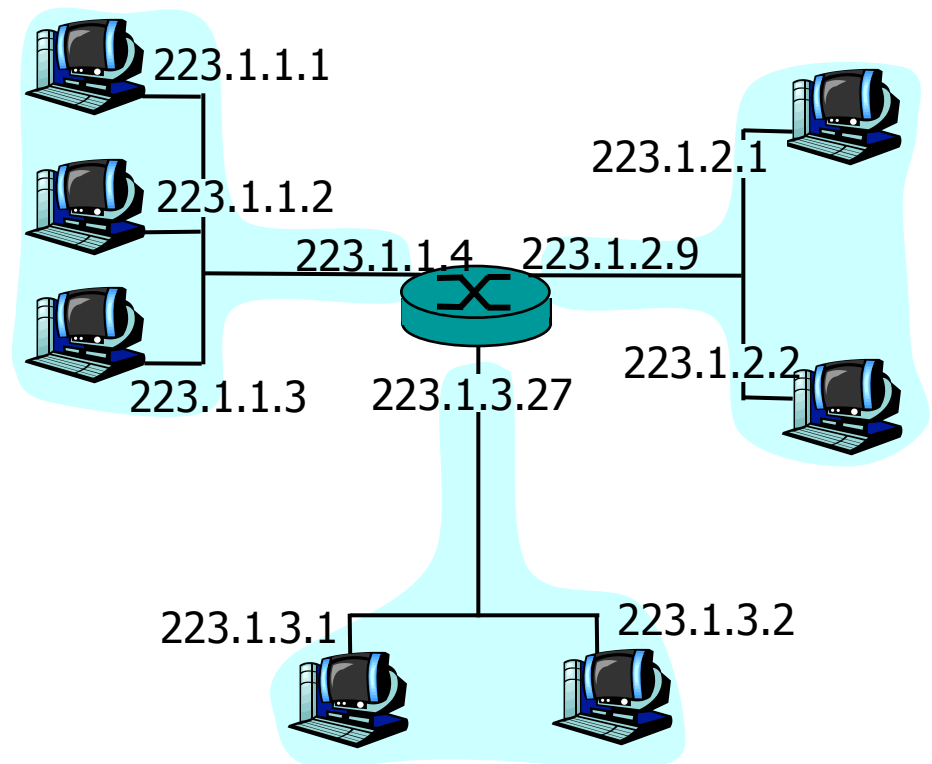


# Endereços e Interfaces

- Endereços IP identificam interfaces de rede
  - NÃO identificam estações
    - Uma única estação pode ter várias interfaces de rede
- Uma estação com várias interfaces de rede possui vários endereços IP
  - Estação *multihomed*
    - Exs. roteadores, estações que balanceiam o tráfego entre diversas redes
- Cada endereço pertence a uma sub-rede, que geralmente corresponde a uma "rede física"

# Sub-redes

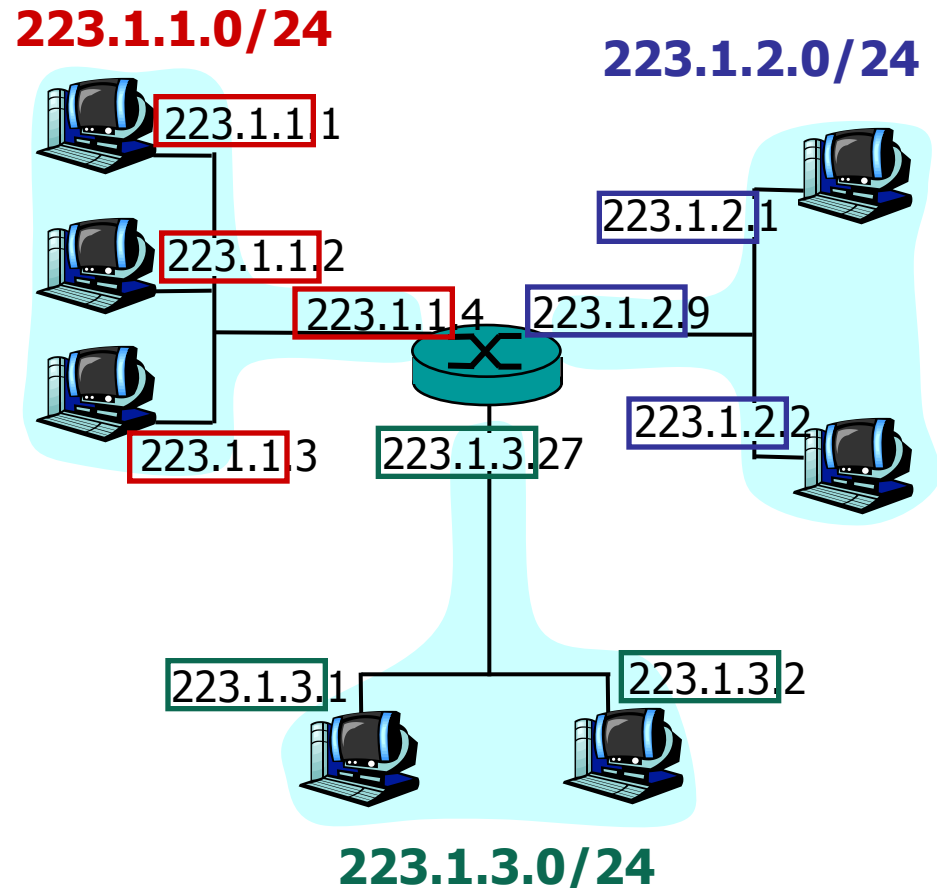
- O que é uma sub-rede IP?
  - Interfaces de dispositivos com a mesma parte de rede nos seus endereços IP
  - Podem alcançar um ao outro sem passar por um roteador



Esta rede consiste de 3 sub-redes IP

# Sub-redes

- O que é uma sub-rede 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 sub-redes IP

# Endereços e Interfaces

- Entradas na tabela de roteamento dos roteadores
  - Normalmente apontam para sub-redes
    - Entretanto, podem eventualmente apontar para endereços de máquinas

```
[user@exemplo ~]$ route -n
```

```
Tabela de Roteamento IP do Kernel
```

Destino	Roteador	MáscaraGen.	Opções	Métrica	Ref	Usa	Iface
200.20.10.64	0.0.0.0	255.255.255.224	U	0	0	0	eth0
169.254.0.0	0.0.0.0	255.255.0.0	U	0	0	0	eth0
0.0.0.0	200.20.10.65	0.0.0.0	UG	0	0	0	eth0



# Endereços e Interfaces

- Por que não um endereço por estação?
  - Um endereço por interface permite *escolher o caminho* utilizado para chegar a uma estação
    - Busca do melhor caminho e balanceamento de carga
  - Endereços por interface permitem a *agregação de endereços* nas tabelas de roteamento
    - Se os endereços não fossem ligados à topologia, seria necessária uma entrada na tabela de roteamento para cada estação
      - Cada interface pertence a uma sub-rede
  - Um endereço por interface permite **manter conectividade** em caso de falha de uma interface
    - Tolerância a falhas

# Endereços e Interfaces

- Desvantagens
  - Todos os endereços de uma estação devem ser incluídos no servidor de nomes
    - Para se comunicar com um determinado nó, deve-se saber todos os possíveis endereços desse nó
  - O "melhor endereço" deve ser escolhido para uma conexão
    - Melhor depende de diversos fatores como caminho, requisitos da aplicação etc.
  - O endereço fonte deve ser cuidadosamente escolhido pela aplicação
    - Determina o caminho seguido pelos pacotes de resposta

# Endereços Especiais

- Endereço de rede
  - Usado para identificar uma rede
  - Geralmente, o primeiro endereço IP da faixa de endereços
    - Ex.: 146.164.0.0
- O "0" pode ser utilizado como **endereço fonte**, quando o número de rede é desconhecido, portanto:
  - 0.0.0.0 significa "esta estação nesta rede"
  - 0.X.Y.Z significa "a estação X.Y.Z nesta rede"
    - Utilizado por ex. quando uma estação está iniciando

# Endereços Especiais

- Difusão limitada (*limited broadcast*)
  - Formado por todos os bits em "1" - **255.255.255.255**
  - Só pode ser utilizado como **endereço destino**
  - Pacote é enviado a todas as estações da sub-rede
  - Não é retransmitido por um roteador

# Endereços Especiais

- Difusão direcionada (*directed broadcast*)
  - Todos os bits da "parte estação" do endereço são colocados em "1"
    - Ex. "A.255.255.255", "C.C.C.255"
  - Com sub-redes a mesma regra é válida
    - todos os bits do complemento da máscara são colocados em "1"

# Endereços Especiais

- Consequências
  - Não existe sub-rede identificada apenas por 0's
  - Assim como não existe sub-rede identificada apenas por 1's
  - O tamanho da sub-rede é maior ou igual a 2 bits
    - Sub-rede com apenas um bit:
      - O "1" seria usado para broadcast
      - O "0" para a própria rede
      - E não sobrariam bits para estações...

# Endereços Especiais

- Endereço de *loopback*
  - Na verdade, existe um número de rede de *loopback*:
    - Rede Classe A: "127.0.0.0\8"
- Qualquer endereço da forma "127.x.y.z" é:
  - Local e não é transmitido para fora da *estação*

# Alocação de Endereços IP

- Atualmente
  - ICANN (*The Internet Corporation for Assigned Names and Numbers*)
    - Organização sem fins lucrativos responsável pela
      - Alocação do espaço de endereçamento IP
      - Atribuição de parâmetros de protocolos
      - Gerenciamento do sistema de nomes de domínios
      - Gerenciamento dos servidores raiz
- Anteriormente
  - IANA (*Internet Assigned Numbers Authority*) e outras entidades através de contratos com o governo americano



# Alocação de Endereços IP

- Os endereços IP são alocados através de delegações de acordo com uma estrutura hierárquica
  1. Usuários recebem endereços IP de um provedor de serviço (ISP - *Internet Service Provider*)
  2. ISPs obtêm faixas de endereços IP de uma autoridade de registro local (LIR - *Local Internet Registry*), nacional (NIR - *National Internet Registry*), ou regional (RIR - *Regional Internet Registry*)
- O papel do ICANN é alocar faixas de endereços aos RIRs, de acordo com suas necessidades e a partir das faixas de endereços livres

# Alocação de Endereços IP

- RIR - *Regional Internet Registry*
  - APNIC (*Asia Pacific Network Information Centre*)
    - Região Ásia/Pacífico
  - ARIN (*American Registry for Internet Numbers*)
    - América do Norte e África ao Sul do Saara
  - LACNIC (*Regional Latin-American and Caribbean IP Address Registry*)
    - América Latina e algumas Ilhas Caribenhas
  - RIPE NCC (*Réseaux IP Européens*)
    - Europa, Oriente Médio, Ásia Central e África do Norte

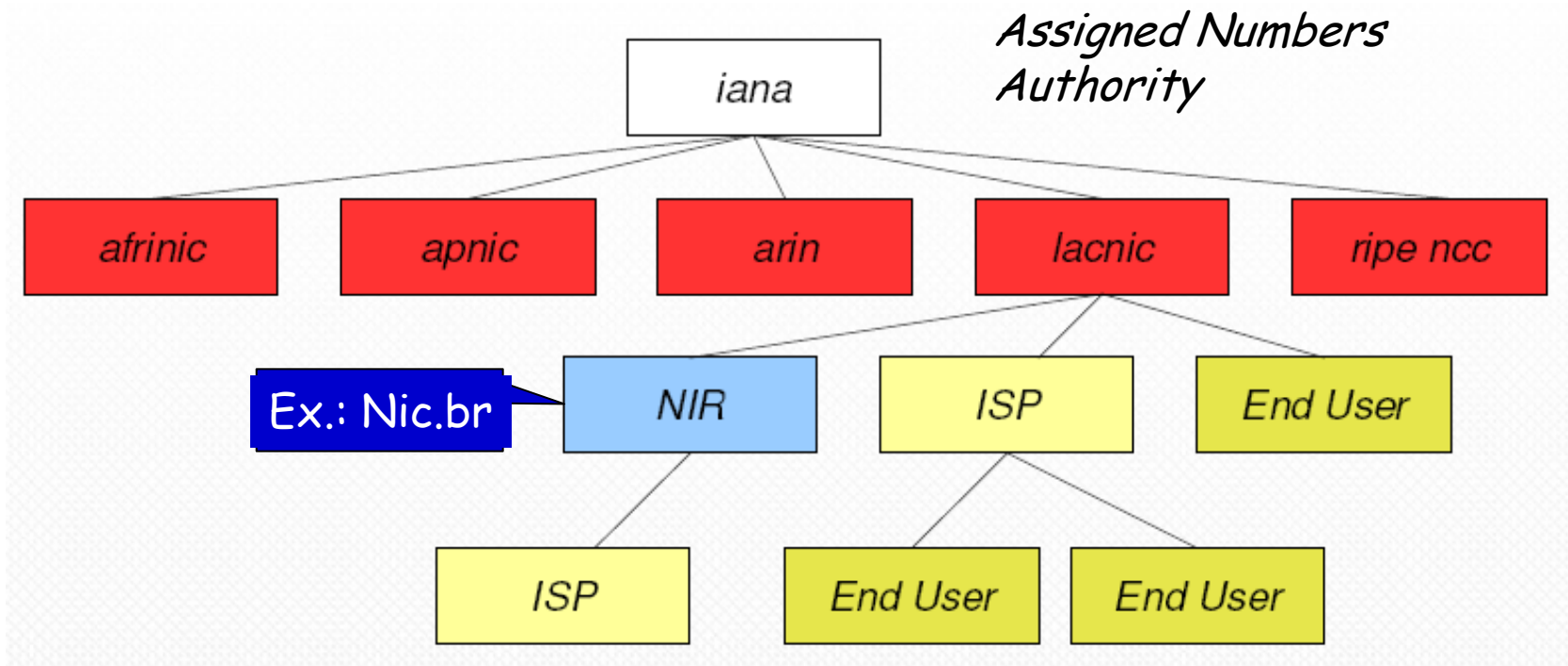
# Alocação de Endereços IP



LACNIC é a instituição responsável para a América Latina e o Caribe

# Alocação de Endereços IP

IANA = *Internet Assigned Numbers Authority*



No Brasil, estas funções foram delegadas ao NIC.br pelo Comitê Gestor da Internet BR (CGI.br)

Regras disponíveis em:

<http://registro.br/provedor/numeracao/regras.html>

# *Internet Control Message Protocol (ICMP)*

- Objetivo
  - Diagnóstico de condições de erro da rede
    - *Simplicidade do IP dificulta diagnóstico de falhas*
- Executado em cima do IP
  - *Protocol type = 1*
- Todo sistema que roda IP deve rodar o ICMP
- Não provê confiabilidade
  - Apenas informação sobre problemas na rede

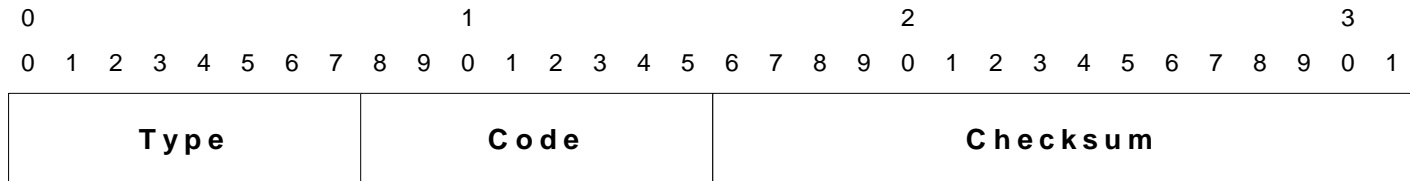
# *Internet Control Message Protocol (ICMP)*

- Erros de transmissão de pacotes IP geram mensagens ICMP
  - Exceto erros nas próprias mensagens ICMP
  - Se as mensagens ICMP também gerassem mensagens de erro
    - Poderia haver recursividade e avalanche de mensagens de controle
      - Ex.: Problemas ligados a congestionamentos na rede

# Mensagens ICMP

- Cabeçalho

- Toda mensagem ICMP possui uma parte do cabeçalho em comum



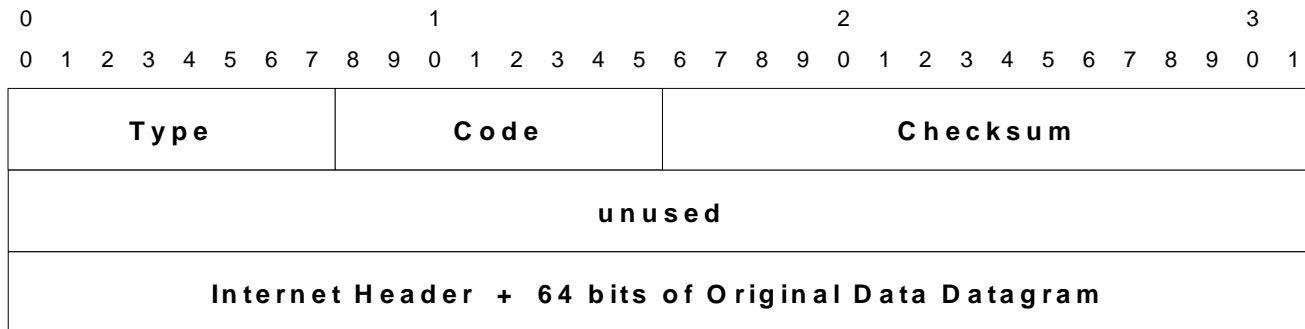
Tipo	Significado
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
9	Router Advertisement

10	Router Solicitation
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

**O checksum do cabeçalho é calculado como no IP**

# Diagnóstico com o ICMP

- Problemas operacionais → Mais comuns
  - *Destination Unreachable*
  - *Time Exceeded*
  - *Source Quench*



- Formato comum
  - Cabeçalho básico do ICMP + 32 bits de enchimento + Primeiros bytes do pacote que causou o envio do ICMP



# Diagnóstico com ICMP

- ICMP envia
  - Cabeçalho IP completo e os 8 primeiros bytes do datagrama
    - Esses dados representam informação suficiente para o nó de origem do pacote IP entender o motivo do erro

# Diagnóstico com o ICMP

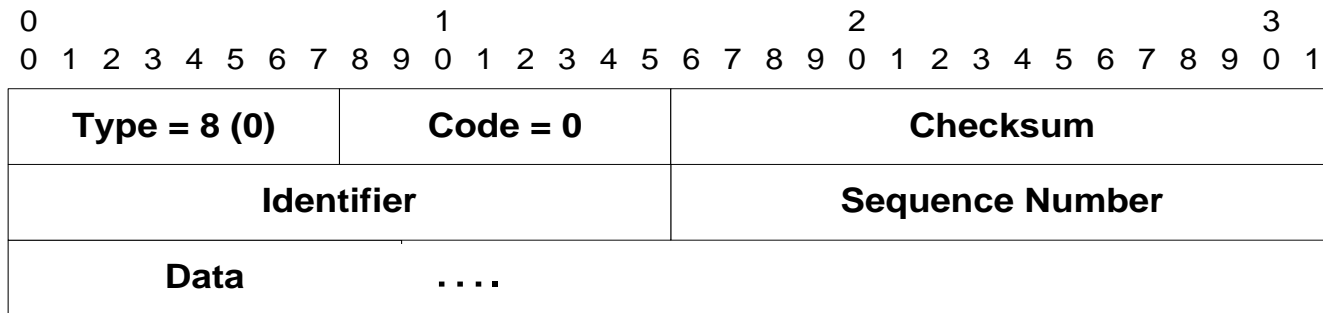
- *Destination Unreachable*
  - Roteador não consegue encaminhar um pacote
  - Código:
    - 0 = *net unreachable*
    - 1 = *host unreachable*
    - 2 = *protocol unreachable*
    - 4 = *fragmentation needed but DF set*
    - 5 = *source route failed*

# Diagnóstico com ICMP

- *Time Exceeded*
  - TTL expirado
  - Código
    - 0 = em trânsito
    - 1 = durante remontagem
- *Source Quench*
  - Enviado pelo roteador para sinalizar congestionamento
  - Não utiliza código (code = 0)

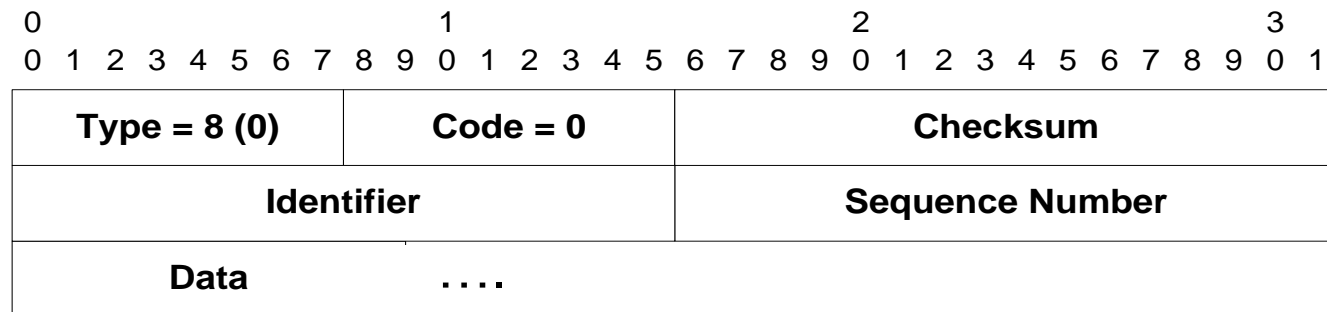
# Ping

- Testa se uma estação está "viva"
  - Ou se a conectividade da rede está funcionando
- Utiliza a função echo do ICMP
  - Tipo:
    - 8 = Echo
    - 0 = Echo Reply



# Ping

- Resposta (*Echo Reply*)
  - Endereços fonte e destino são trocados
  - Troca do valor do tipo da mensagem
  - *Checksums* IP e ICMP recalculados
  - Dados inalterados



# Ping

- Campos identificação e número de sequência possibilitam estatísticas
- Outras mensagens ICMP com funcionalidade semelhante
  - Type = 15 - Information Request
  - Type = 16 - Information Reply

# Exemplo de Ping

```
PING angra (146.164.69.1) from 146.164.69.2 : 56(84) bytes of data.  
recreio::user [ 31 ] ping angra  
64 bytes from angra (146.164.69.1): icmp_seq=1 ttl=64 time=0.471 ms  
64 bytes from angra (146.164.69.1): icmp_seq=2 ttl=64 time=0.404 ms  
64 bytes from angra (146.164.69.1): icmp_seq=3 ttl=64 time=0.544 ms  
64 bytes from angra (146.164.69.1): icmp_seq=4 ttl=64 time=0.388 ms  
64 bytes from angra (146.164.69.1): icmp_seq=5 ttl=64 time=0.398 ms  
64 bytes from angra (146.164.69.1): icmp_seq=6 ttl=64 time=0.398 ms  
64 bytes from angra (146.164.69.1): icmp_seq=7 ttl=64 time=0.495 ms  
64 bytes from angra (146.164.69.1): icmp_seq=8 ttl=64 time=0.436 ms  
64 bytes from angra (146.164.69.1): icmp_seq=9 ttl=64 time=0.413 ms  
64 bytes from angra (146.164.69.1): icmp_seq=10 ttl=64 time=0.407 ms  
64 bytes from angra (146.164.69.1): icmp_seq=11 ttl=64 time=0.393 ms  
64 bytes from angra (146.164.69.1): icmp_seq=12 ttl=64 time=0.391 ms  
  
--- angra ping statistics ---  
12 packets transmitted, 12 received, 0% loss, time 11109ms  
rtt min/avg/max/mdev = 0.388/0.428/0.544/0.049 ms
```

# Traceroute

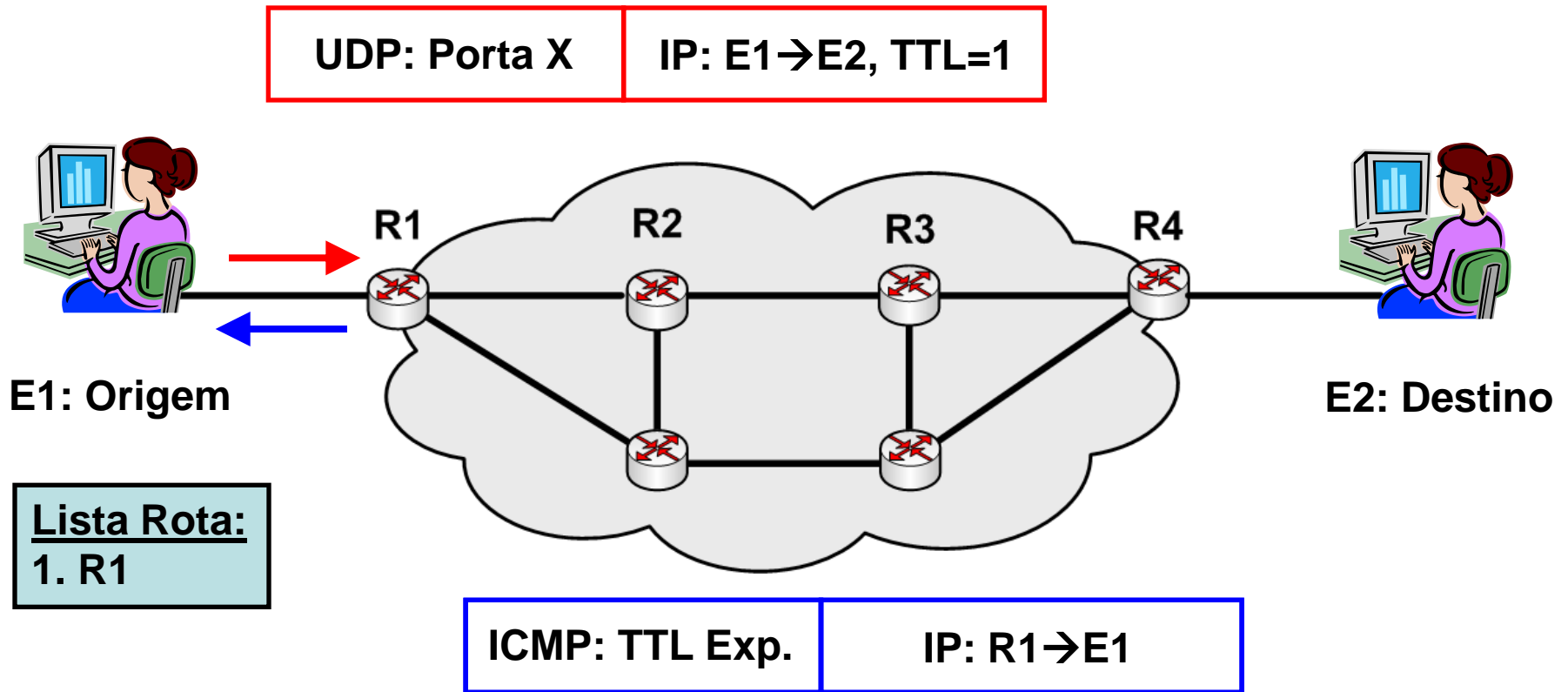
- Identifica os roteadores entre uma fonte e um destino
- Funcionamento:
  - Envio sucessivo de pacotes para o destino, variando o TTL
    - UDP em uma porta não utilizada
    - TTL inicial igual a 1
  - Primeiro roteador decrementa o TTL, descarta o pacote, e envia uma mensagem ICMP TTL Exceeded
    - Roteador identificado através do Source Address da mensagem



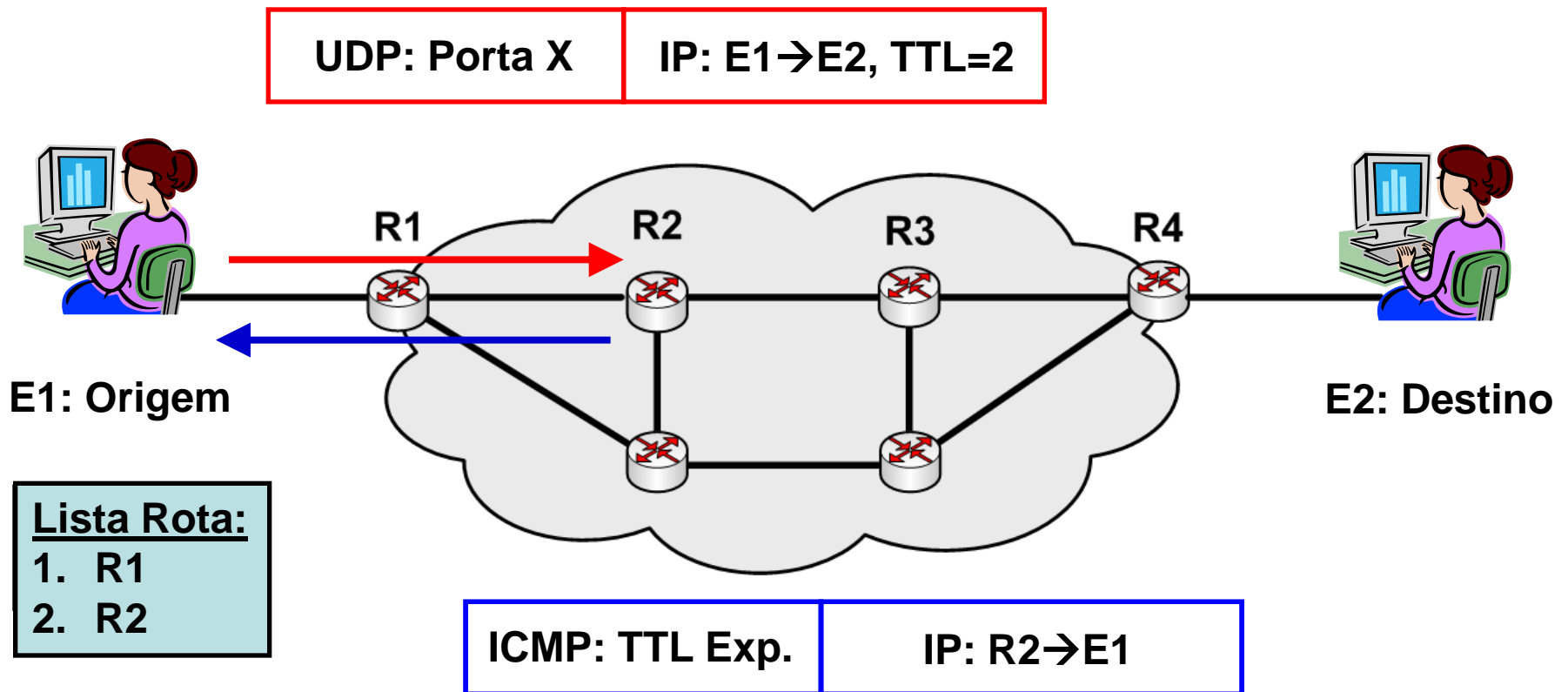
# Traceroute

- Identifica os roteadores entre uma fonte e um destino
- Funcionamento:
  - A fonte continua o processo incrementando o TTL de 1 até chegar ao destino ou alcançar um enlace com problema
  - O destino é identificado, pois ele envia uma mensagem ICMP Port unreachable

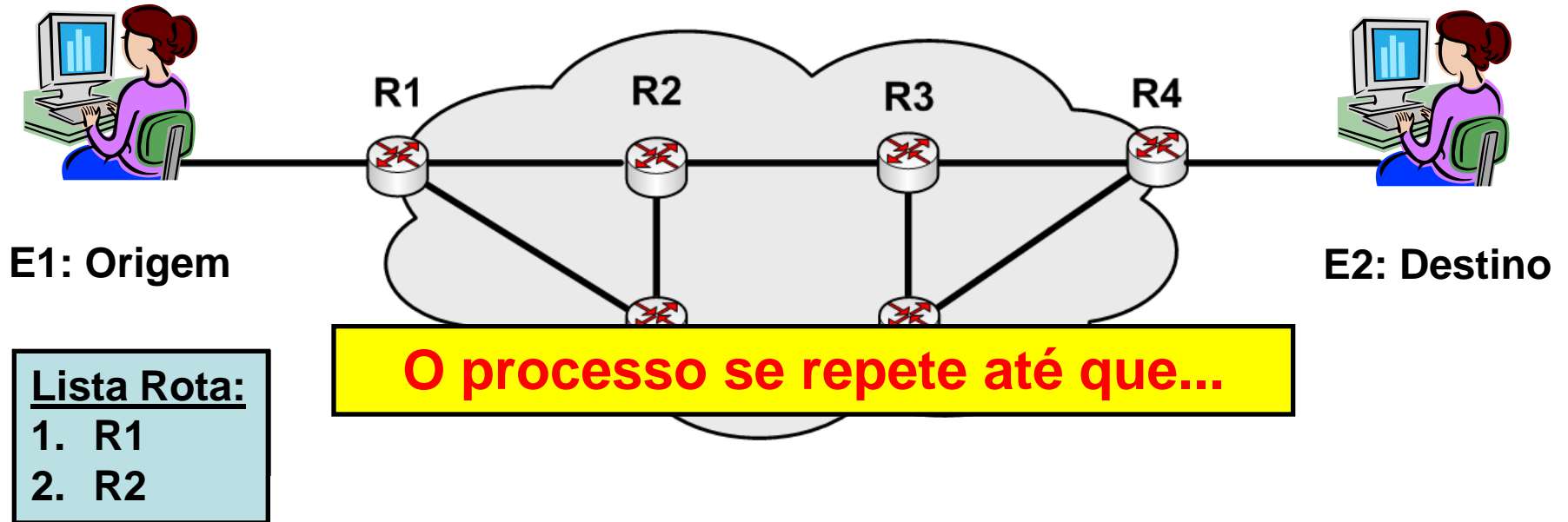
# Traceroute



# Traceroute



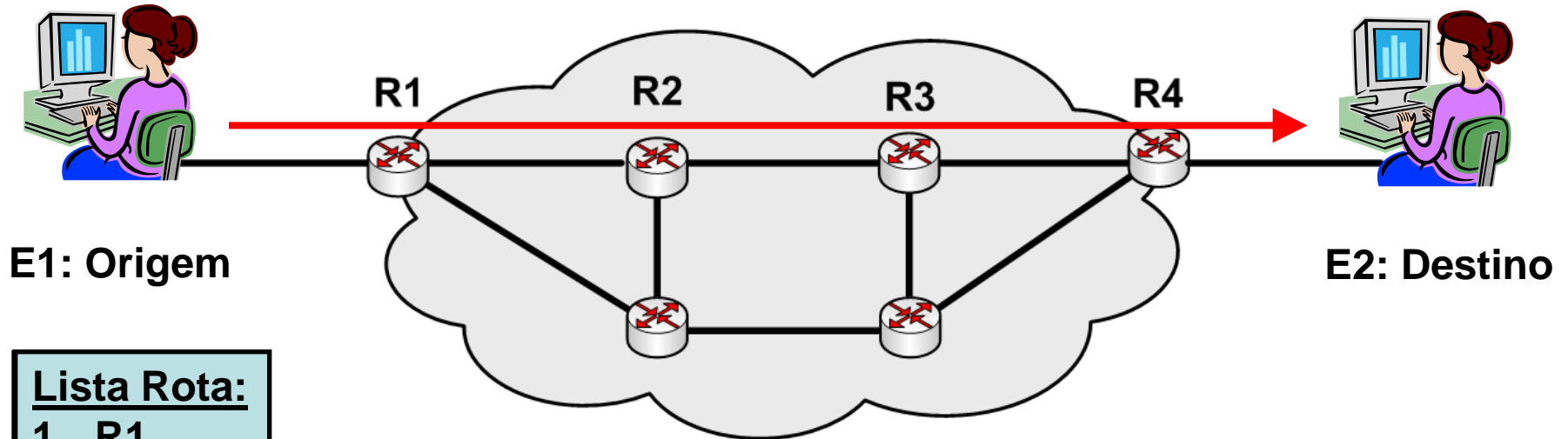
# Traceroute



# Traceroute

UDP: Porta X

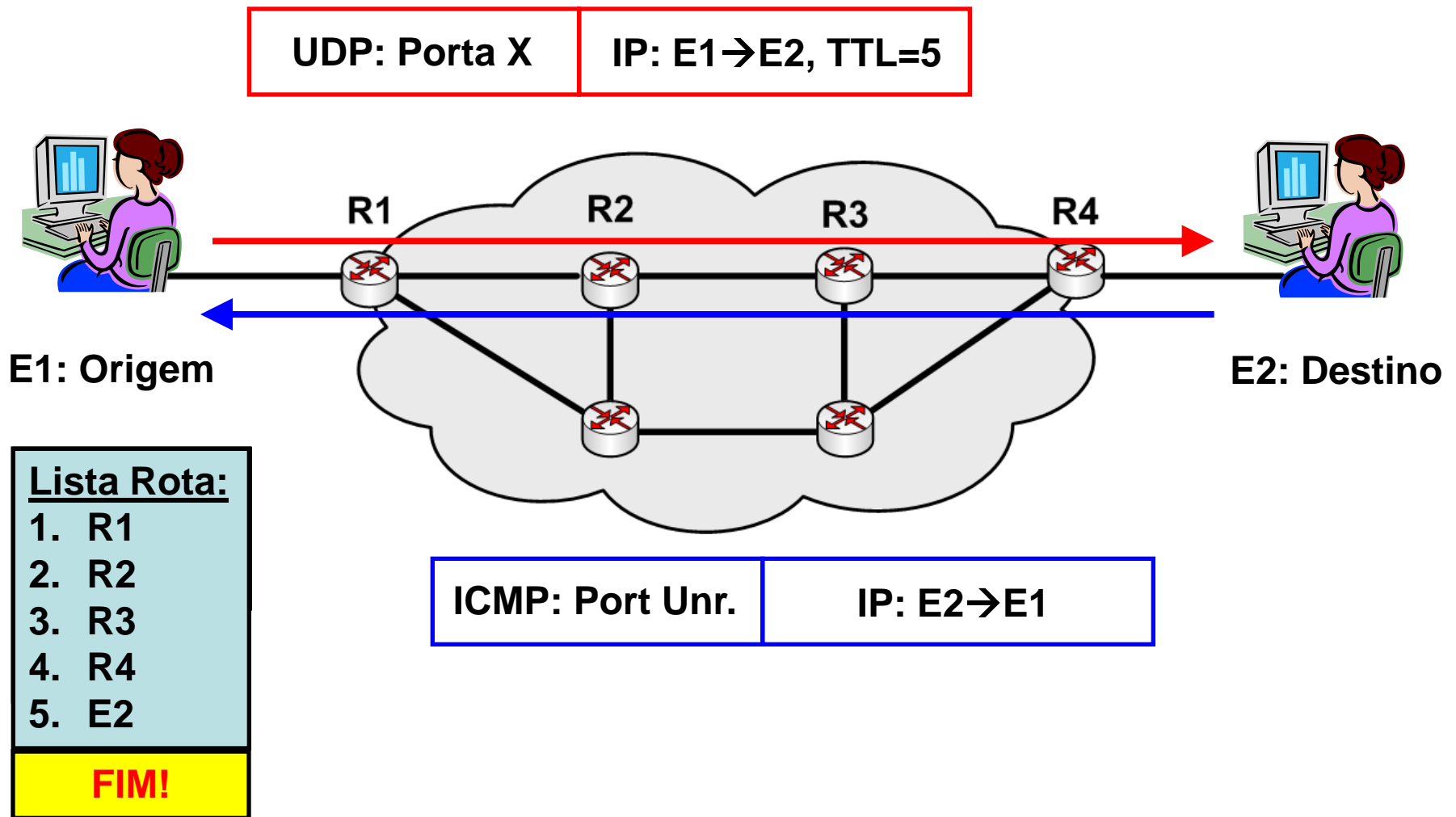
IP: E1→E2, TTL=5



## Lista Rota:

1. R1
2. R2
3. R3
4. R4

# Traceroute



# Exemplo de Traceroute

```
recreio::user [ 38 ] traceroute sphinx.lip6.fr
traceroute to sphinx.lip6.fr (132.227.74.253), 30 hops max, 38 byte packets
 1 angra (146.164.69.1) 0.596 ms 0.349 ms 0.341 ms
 2 rt-ct-bloco-H.ufrj.br (146.164.5.193) 175.723 ms 203.553 ms 30.226 ms
 3 rt-nce2.ufrj.br (146.164.1.5) 51.432 ms 3.994 ms 4.137 ms
 4 rederio2-atm-cbpf.rederio.br (200.20.94.58) 3.495 ms 4.421 ms 4.664 ms
 5 200.143.254.66 (200.143.254.66) 4.184 ms 12.224 ms 200.143.254.78
   (200.143.254.78) 13.372 ms
 6 rj7507-fast6_1.bb3.rnp.br (200.143.254.93) 4.473 ms 4.135 ms 4.550 ms
 7 ds3-rnp.ampath.net (198.32.252.237) 110.658 ms 106.239 ms 107.241 ms
 8 abilene.ampath.net (198.32.252.254) 125.393 ms 135.971 ms 127.111 ms
 9 washng-atla.abilene.ucaid.edu (198.32.8.66) 143.388 ms 154.348 ms 144.619 ms
10 abilene.de2.de.geant.net (62.40.103.253) 234.914 ms 235.300 ms 239.316 ms
11 de2-1.del.de.geant.net (62.40.96.129) 234.644 ms 238.821 ms 236.147 ms
12 de.fr1.fr.geant.net (62.40.96.50) 231.422 ms 232.743 ms 232.437 ms
13 renater-gw.fr1.fr.geant.net (62.40.103.54) 234.984 ms 234.233 ms 231.723 ms
14 jussieu-a1-1-580.cssi.renater.fr (193.51.179.154) 230.906 ms 231.090 ms
   233.714 ms
15 rap-jussieu.cssi.renater.fr (193.51.182.201) 232.602 ms 232.125 ms 238.066 ms
16 cr-jussieu.rap.prd.fr (195.221.126.77) 235.182 ms 239.903 ms 276.221 ms
17 jussieu-rap.rap.prd.fr (195.221.127.182) 234.955 ms 237.264 ms 234.210 ms
18 r-scott.reseau.jussieu.fr (134.157.254.10) 233.992 ms 238.306 ms 239.047 ms
19 olympe-gw.lip6.fr (132.227.109.1) 236.396 ms !N 235.261 ms !N 234.322 ms !N
```

# Exemplo de Ping -R

```
recreio::user [ 35 ] ping -R sphinx.lip6.fr
PING sphinx.lip6.fr (132.227.74.253) from 146.164.69.2 : 56(124) bytes of data.
64 bytes from sphinx.lip6.fr (132.227.74.253): icmp_seq=1 ttl=237 time=252 ms
RR:   recreio (146.164.69.2)
      gtagw (146.164.5.210)
      rt-ct2.ufrj.br (146.164.1.3)
      ufrj-atm.rederio.br (200.20.94.9)
      200.143.254.65
      rj-fast4_1.bb3.rnp.br (200.143.254.94)
      rnp.ampath.net (198.32.252.238)
      abilene-oc3.ampath.net (198.32.252.253)
      atla-washng.abilene.ucaid.edu (198.32.8.65)
64 bytes from sphinx.lip6.fr (132.227.74.253): icmp_seq=2 ttl=237 time=289 ms
RR:   recreio (146.164.69.2)
      ...
64 bytes from sphinx.lip6.fr (132.227.74.253): icmp_seq=3 ttl=237 time=247 ms
RR:   recreio (146.164.69.2)
      ...
--- sphinx.lip6.fr ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2021ms
rtt min/avg/max/mdev = 247.821/263.167/289.150/18.477 ms
```



# Exemplo de Ping -R

```
recreio::user [ 35 ] ping -R sphinx.lip6.fr
PING sphinx.lip6.fr (132.227.74.253) from 146.164.69.2 : 56(124) bytes of data.
64 bytes from sphinx.lip6.fr (132.227.74.253): icmp_seq=1 ttl=237 time=289 ms
RR:   recreio (146.164.69.2)
      gtagw (146.164.5.210)
      rt-ct2.ufrj.br (146.164.1.3)
      ufrj-atm.rederio.br (200.20.94.9)
      200.143.254.65
      rj-fast4_1.bb3.rnp.br
      rnp.ampath.net
      abilene-georgia (198.32.8.65)
      atlanta-georgia (198.32.8.65)
64 bytes from sphinx.lip6.fr (132.227.74.253): icmp_seq=2 ttl=237 time=289 ms
RR:   recreio (146.164.69.2)
      ...
64 bytes from sphinx.lip6.fr (132.227.74.253): icmp_seq=3 ttl=237 time=247 ms
RR:   recreio (146.164.69.2)
      ...
--- sphinx.lip6.fr ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2021ms
rtt min/avg/max/mdev = 247.821/263.167/289.150/18.477 ms
```

**Usa a opção Record Route do IP!  
Limitado ao tamanho do campo "opções"**

# Gerenciamento de Tempo

- Mensagens
  - Type = 13 - Timestamp
  - Type = 14 - Timestamp reply

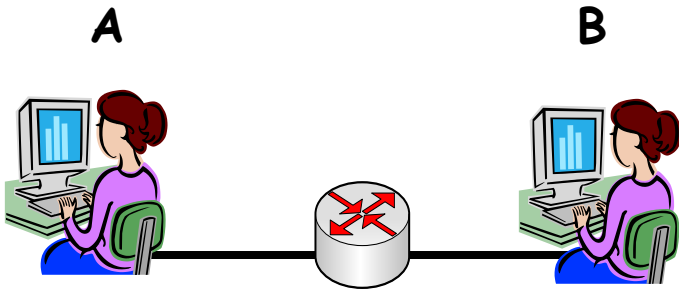
0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1
<b>Type = 8 (0)</b>	<b>Code = 0</b>	<b>Checksum</b>	
<b>Identifier</b>		<b>Sequence Number</b>	
<b>Originate Timestamp</b>			
<b>Receive Timestamp</b>			
<b>Transmit Timestamp</b>			

**Tempos expressos em ms desde 0:00 h GMT**

# Cálculo da Defasagem entre Duas Estações

- Funcionamento
  - Estação A preenche o tempo de origem ( $T_o$ ) pouco antes de enviar a mensagem
  - Na recepção, a estação B preenche o tempo de recepção ( $T_r$ )
    - Assim que a mensagem chega
    - Em seguida, a estação B prepara a resposta
  - Antes do envio da resposta, B preenche o tempo de transmissão ( $T_t$ )
  - Ao receber a resposta, A armazena o tempo de chegada ( $T_c$ )
    - Assim que a mensagem chega

# Cálculo da Defasagem entre Duas Estações



Defasagem = Diferença medida de relógios  
- tempo de transmissão

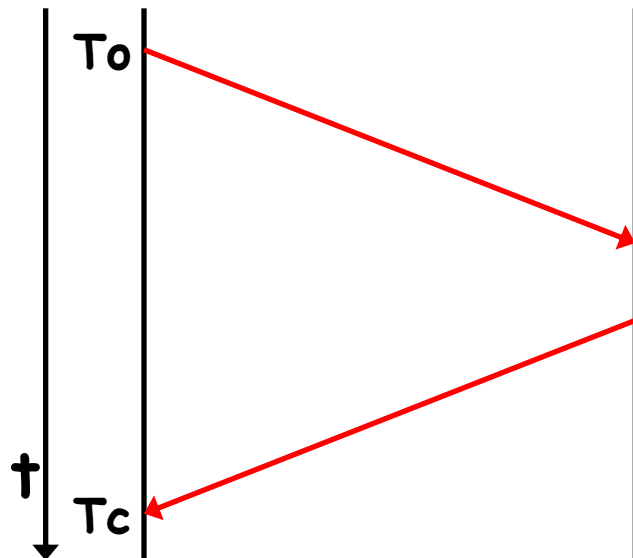
RTT = *Round Trip Time*

Tempo de transmissão =  $RTT/2$

$RTT = T_c - T_o - (T_t - T_r)$

Defasagem =  $T_r - (T_o + RTT/2)$

Se Defasagem  $> 0$ , A está atrasada,  
Caso contrário, A está adiantada



Tempo de processamento  
da mensagem

# Envio de Pacotes IP

- Roteadores
  - Executam um protocolo de roteamento
- Estações
  - Não, necessariamente, executam um protocolo de roteamento
- Porque...
  - Complexidade e variedade dos protocolos de roteamento modernos
  - Poderia-se apenas "ouvir" as mensagens de roteamento
    - Algumas vezes este processo pode não ser fácil
      - Ex. mecanismos de segurança (autenticação, criptografia)

# Envio de Pacotes IP

- Roteadores
  - Executam um protocolo de roteamento
- Estações
  - Não, necessariamente

O que é necessário para uma estação enviar um pacote?

- Complexidade e variedade dos protocolos de roteamento modernos
- Poderia-se apenas "ouvir" as mensagens de roteamento
  - Algumas vezes este processo pode não ser fácil
    - Ex. mecanismos de segurança (autenticação, criptografia)

# Envio de Pacotes IP

- Roteadores
  - Executam um protocolo de roteamento
- Estações
  - Não, necessariamente

O que é necessário para uma estação enviar um pacote?  
Descobrir um roteador de saída

- Poderia-se apenas "ouvir" as mensagens de roteamento
  - Algumas vezes este processo pode não ser fácil
    - Ex. mecanismos de segurança (autenticação, criptografia)

# Descoberta do Próximo Salto

- Dado um pacote IP a transmitir, a quem enviar?
  - Estação destino na rede
    - Envio direto
  - Estação destino distante
    - Envio a um roteador que encaminhará o pacote
- Para descobrir se a estação de destino está na sub-rede
  - Testa-se a mascara de rede do endereço IP do destino



# Descoberta do Próximo Salto

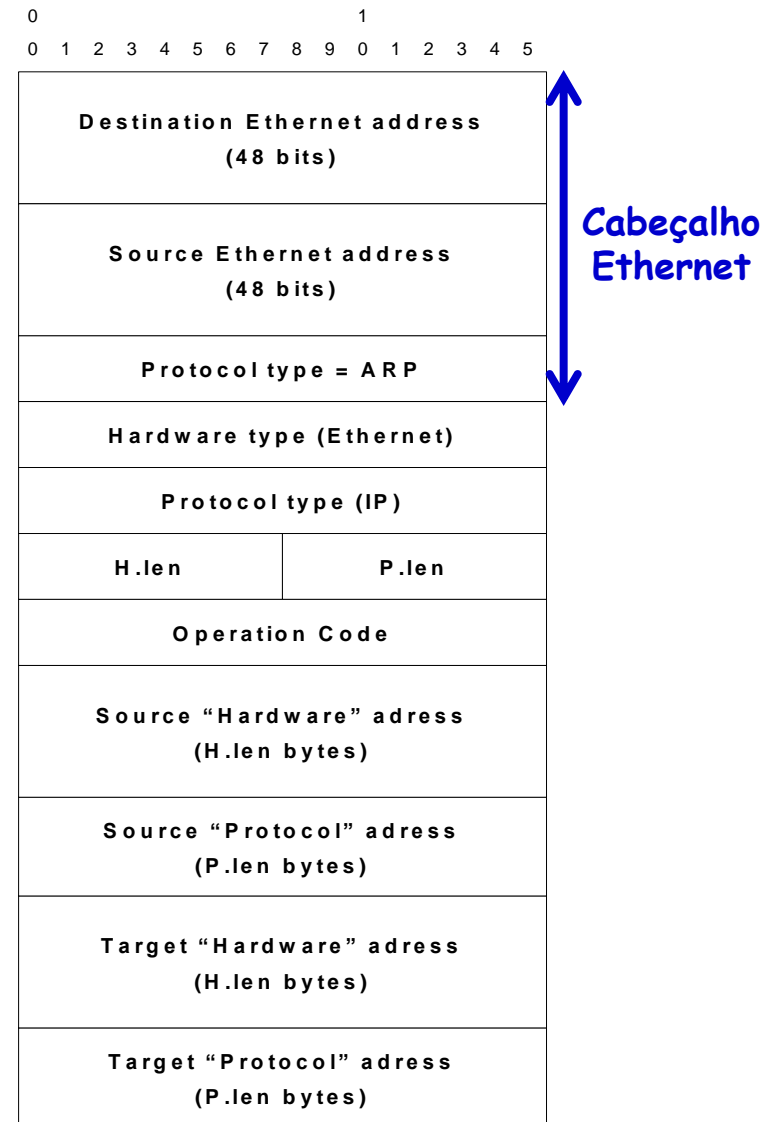
- Dado um pacote IP a transmitir, a quem enviar?
  - Estação destino na rede
    - Envio direto
  - Estação destino distante
    - Envio a um roteador, que encaminhará o pacote
- Para descobrir se a estação de destino está na sub-rede...
  - Testa-se a mascara de rede do endereço IP do destino



**Independente se está na sub-rede, o próximo passo é descobrir o endereço físico (MAC) do próximo salto**

# Address Resolution Protocol (ARP)

- Envio de *ARP request*
  - Realizado em *broadcast*
    - **op. code 1**
- Máquina que reconhece seu IP no *ARP request*
  - Envia um *ARP response*
    - **op. code 2**
- *Uso de cache*
  - Utilizada para evitar o envio frequente de requisições ARPs



# Descoberta do Roteador

- Por configuração  
ou
- Usando o ICMP
  - Roteadores enviam mensagens **ICMP router advertisement** (type = 10) periodicamente
  - Estações podem enviar mensagens **ICMP router solicitation** (type = 9) para requisitar anúncios de rotas
  - O objetivo do procedimento é descobrir **um** roteador de saída, não necessariamente o **melhor** roteador de saída...
    - **Mensagens ICMP *redirect* podem ser utilizadas para informar as estações de rotas melhores**

# Anúncios (*Router Advertisements*)

- Podem conter diversos endereços para o mesmo roteador
  - Várias interfaces conectadas à mesma rede
  - Uma interface de rede com dois endereços IP
    - **Sub-redes IP na mesma rede física (ex. segmento Ethernet)**
  - Preference - prioridade de escolha entre vários roteadores
    - **Configurado pelo administrador da rede**

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

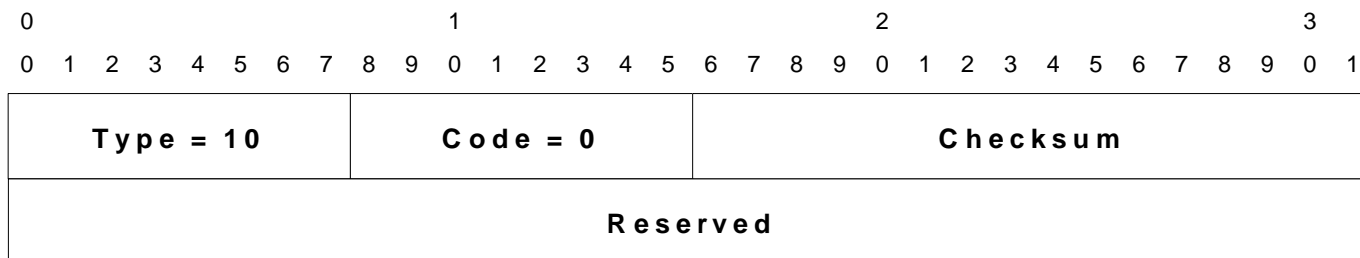
Type = 9	Code = 0	Checksum
Num. Adrs	Addr. Entry Size	Lifetime
Router Address[1]		
Preference Level[1]		
Router Address[2]		
Preference Level[2]		
...		



**Addr. Entry Size = 2 (Router Address + Preference)**

# Anúncios (*Router Advertisements*)

- São enviados ao endereço **224.0.0.1** (todas as máquinas) ou a **255.255.255.255**
- Informação sobre o roteador de saída
  - Deve ser volátil para evitar uso de rotas em desuso
  - Tempo de vida (*Lifetime*)
    - **30 min.**
- Anúncios (*router advertisements*) enviados a cada 7 min.
  - Evitar congestionamento da rede
  - Como o período é longo, estações podem enviar solicitações



# Escolha do Roteador

- *Router solicitation*
  - Enviadas a **224.0.0.2** ("*todos os roteadores*") ou **255.255.255.255**
- O roteador envia a resposta
  - À estação, ou
  - A todas as estações, se o momento do anúncio estiver próximo
- Estações podem receber várias respostas
  - Devem considerar apenas os roteadores na sua sub-rede
  - Devem selecionar o de maior valor de preferência
  - Devem enviar todo o tráfego para este roteador

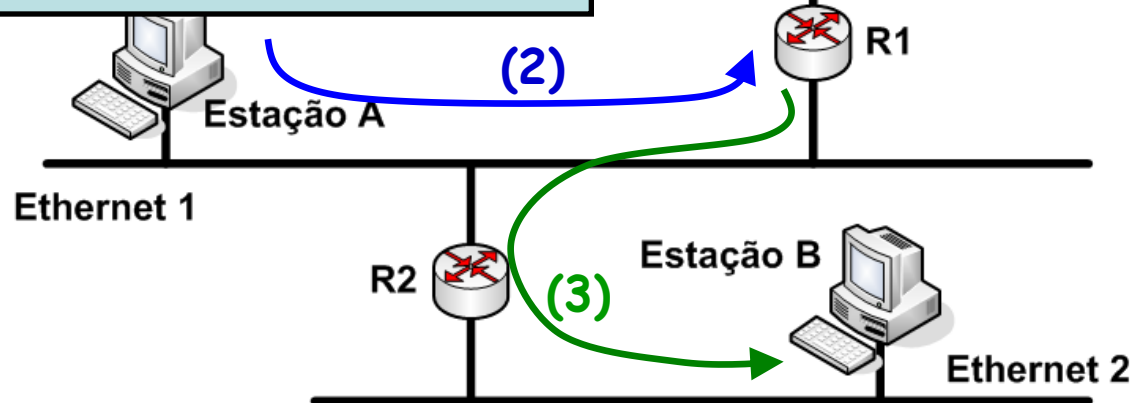
# Redirecionamento ICMP

- Evita rotas ineficientes para outras redes

(1) Estação A conhece apenas rota de saída através de R1 (Rota default)

(2) Estação A envia pacotes para R1 para alcançar a Estação B

(3) R1 conhece o caminho através de R2 para alcançar a Estação B



Como evitar que o tráfego destinado a Estação B passe por R1? (duas vezes no segmento Ethernet 1)

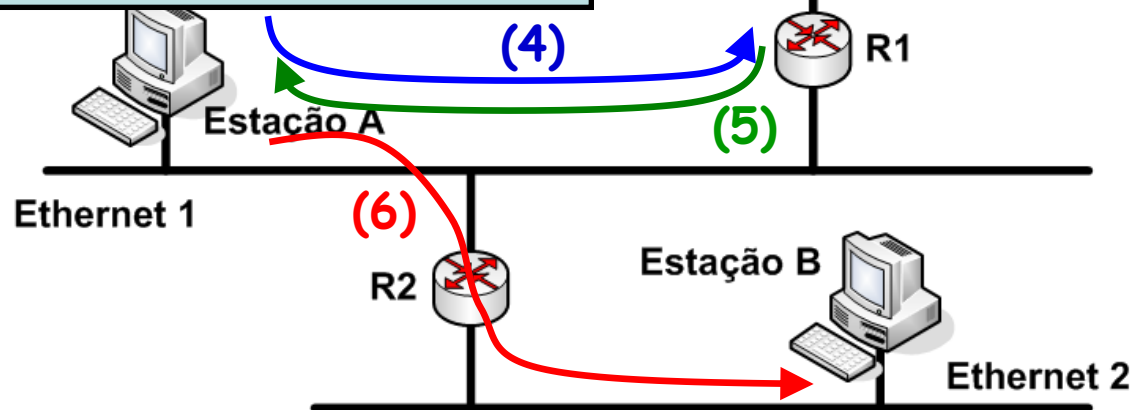
# Redirecionamento ICMP

- São utilizadas mensagens ICMP *redirect*

(4) Estação A envia o primeiro pacote para R1 para chegar na Estação B

(5) R1 envia uma mensagem ICMP *redirect* para a Estação A

(6) Estação A adiciona uma rota para Estação B passando por R2



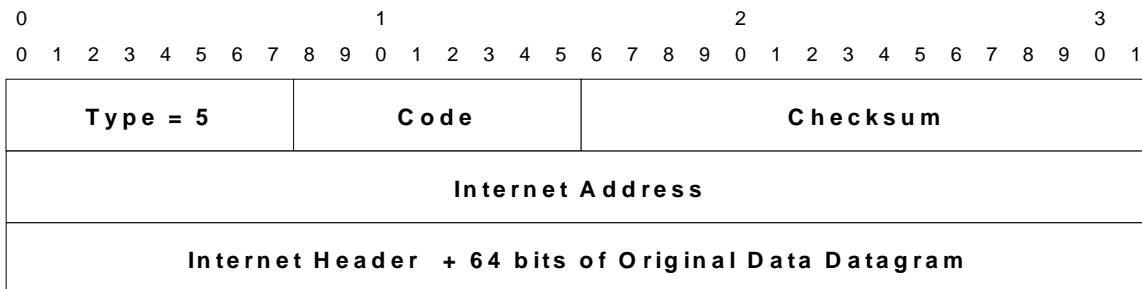
ICMP: Redir., Prox. Salto = R2, Dest. = B

IP: R1 → A



# Redirecionamento ICMP

- Ao receber o ICMP redirect, a estação A deve mudar sua tabela de roteamento
  - Para o endereço contido no campo Internet Header, o próximo salto é dado por Internet Address
- O redirecionamento pode ser para uma rede
  - Indicado no campo código
    - Mas não existe espaço para uma máscara, portanto não é possível redirecionar o tráfego para uma sub-rede



## Code:

- 0: redirecionar pacotes para a Rede**
- 1: redirecionar pacotes para a Estação**
- 2: Rede e ToS**
- 3: Estação e ToS**

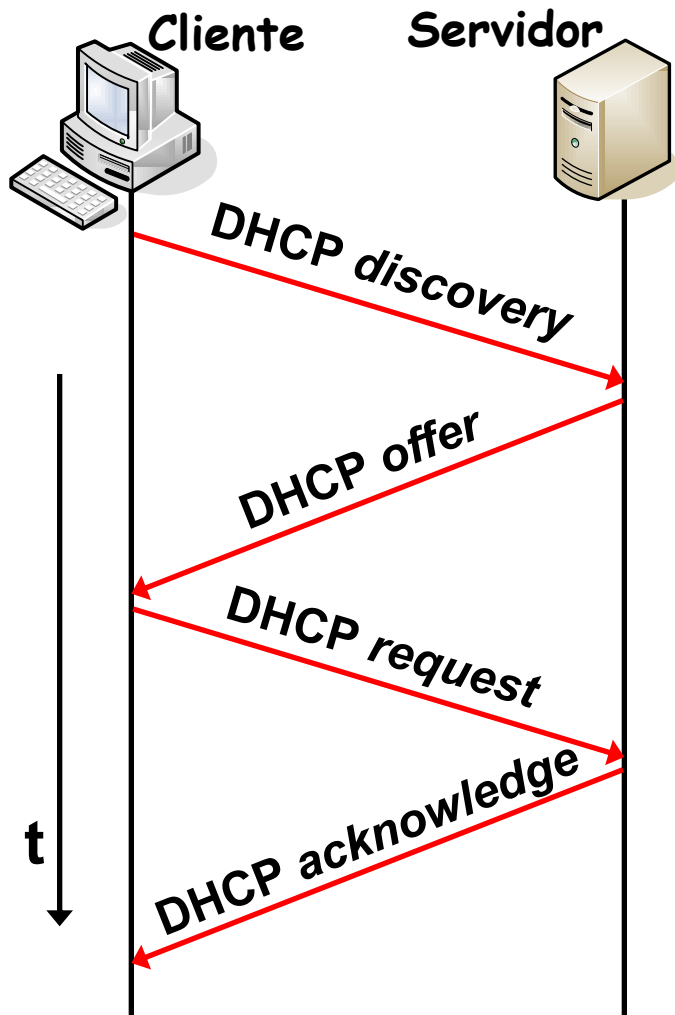
# *Dynamic Host Configuration Protocol (DHCP)*

- A premissa até o momento é que cada estação conhece o seu próprio endereço IP
  - Endereço pré-configurado
- Entretanto, isso pode nem sempre ser verdade...
  - Nesses casos, é necessário obter um endereço IP
- Alguns protocolos com essa finalidade são
  - RARP: *Reverse Address Resolution Protocol*
  - BOOTP: *Bootstrap Protocol*
  - DHCP
    - **Mais utilizado atualmente**

# Dynamic Host Configuration Protocol (DHCP)

- Aloca automaticamente endereços IP para estações em uma sub-rede
  - Os endereços podem ser reusados
- Passa outras informações adicionais
  - Ex. Rota *default*, máscara de sub-rede, servidor DNS
- Utiliza uma arquitetura cliente-servidor
  - Cliente DHCP
    - Estação que solicita parâmetros de configuração de rede
  - Servidor DHCP
    - Estação que responde as solicitações por parâmetros de configuração das estações clientes

# Dynamic Host Configuration Protocol (DHCP)



• Processo realizado em 4 etapas:

- DHCP *discovery*
  - Cliente envia mensagem em *broadcast* para descobrir os servidores disponíveis
- DHCP *offer*:
  - Servidores DHCP disponíveis respondem com um endereço IP disponível e outras configurações de rede
- DHCP *request*
  - Cliente escolhe uma das ofertas recebidas e solicita individualmente a um servidor as suas configurações
- DHCP *acknowledge*
  - Servidor envia endereço IP e as outras configurações de rede

# Network Address Translation (NAT)

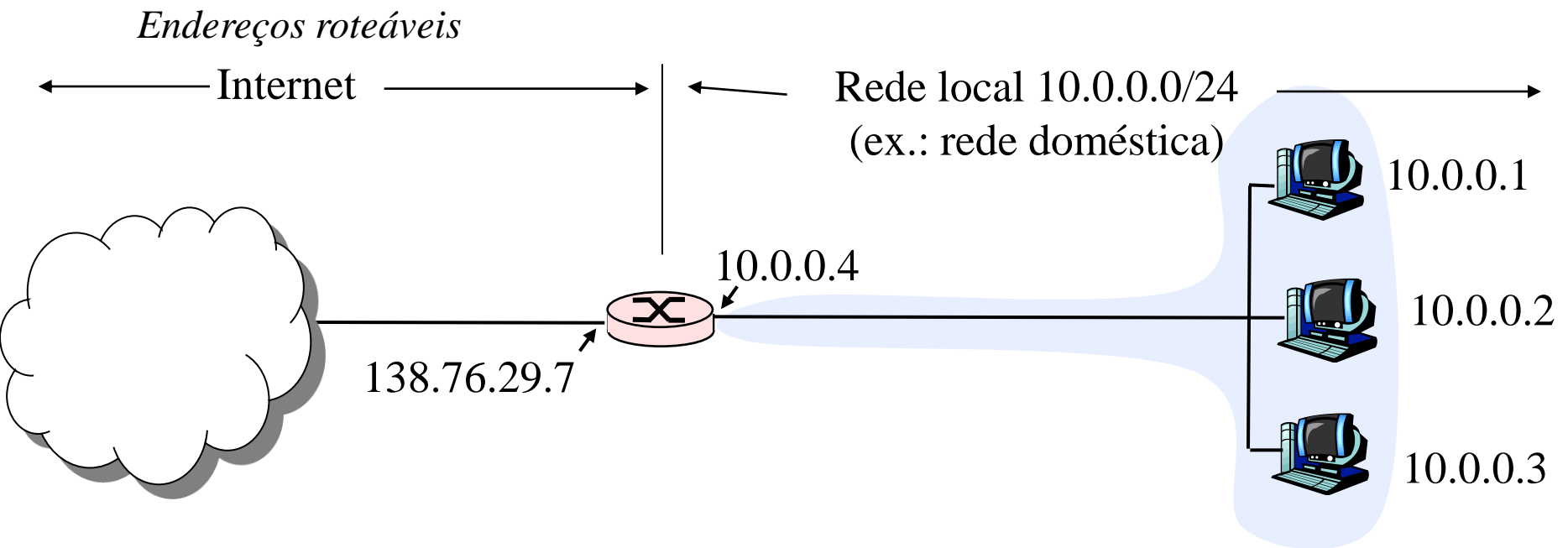
- Recurso utilizado inicialmente para contornar a possível escassez de endereços IP
  - Usado por mais da metade dos usuários domésticos nos EUA
- Endereço IP público X Endereço IP privado
  - Endereço IP público
    - Definido em escopo global → Internet
      - Endereço roteável
  - Endereço IP privado
    - Definido em escopo local → rede local
      - Endereço não roteável
        - » Blocos de endereços definidos pelo IANA: Rede 10.0.0.0/8, 192.168.0.0/16 e 172.16.0.0/12

# Network Address Translation (NAT)

- *IP masquerading*
  - Processo de tradução dos endereços de uma rede local com endereços privados para endereços públicos
    - Consiste em "mascarar" um espaço de endereços privados para Internet
  - Roteador mantém estado dos fluxos que possuem pacotes traduzidos
    - Necessário para encaminhar respostas para a origem
  - Roteador é responsável pela tradução pode converter...
    - Endereço IP da origem para endereço IP próprio
    - Porta de origem para uma porta conhecida

# Network Address Translation (NAT)

- Estrutura



# Network Address Translation (NAT)

- Funcionamento

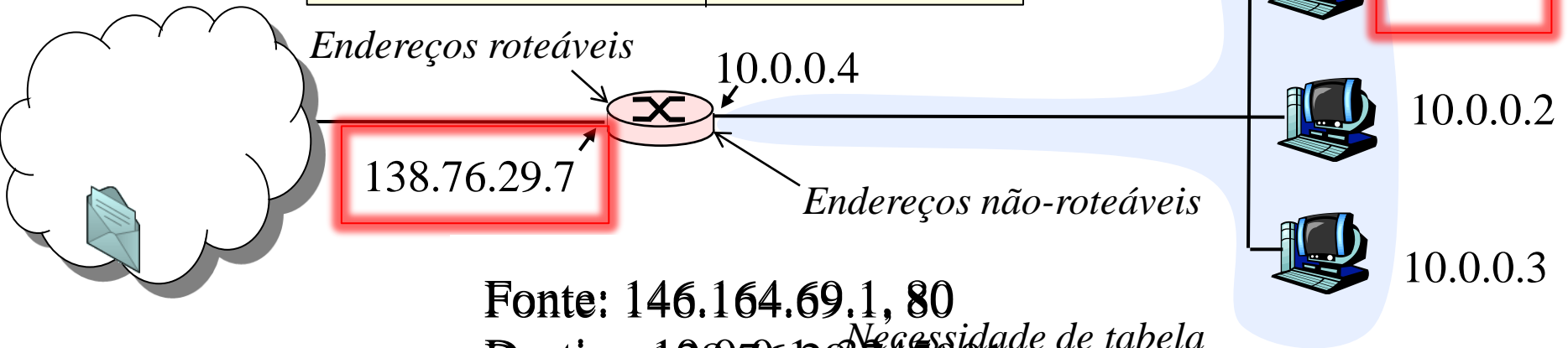
Fonte: 146.164.69.1, 80

Destino: 10.0.0.1, 3345

Fonte: 10.0.0.1, 3345

Destino: 146.164.69.1, 80

Tabela de tradução NAT	
Lado WAN	Lado LAN
138.76.29.7, 5001	10.0.0.1, 3345



Fonte: 146.164.69.1, 80

Destino: 138.76.29.7, 5001

*Necessidade de tabela de tradução NAT*

Fonte: 146.164.69.1, 80  
Destino: 138.76.29.7, 5001



# Network Address Translation (NAT)

- Quebra do requisito fim-a-fim da Internet
  - Nós na Internet não conseguem se comunicar com nós "atrás" de dispositivos NAT
    - Prejudicam as aplicações par-a-par
- Soluções
  - Mapeamento de portas
    - NAT estático
  - UPnP (*Universal Plug-and-Play*)
    - Padrão que utiliza protocolos para realizar mapeamento automático de portas

# Material Utilizado

- Notas de aula do Prof. Igor Monteiro Moraes, disponíveis em <http://www2.ic.uff.br/~igor/cursos/redespg>

# Leitura Recomendada

- Capítulo 4 do Livro "*Computer Networking: A Top Down Approach*", 5a. Ed., Jim Kurose and Keith Ross, Pearson, 2010
- Capítulo 5 do Livro "*Computer Networks*", Andrew S. Tanenbaum e David J. Wetherall, 5a. Ed., Pearson, 2011
- Capítulo 3 do Livro do "*Routing in the Internet*", Christian Huitema, 2ª. Ed., Prentice-Hall, 1999