

PTC 2550 - Aula 04

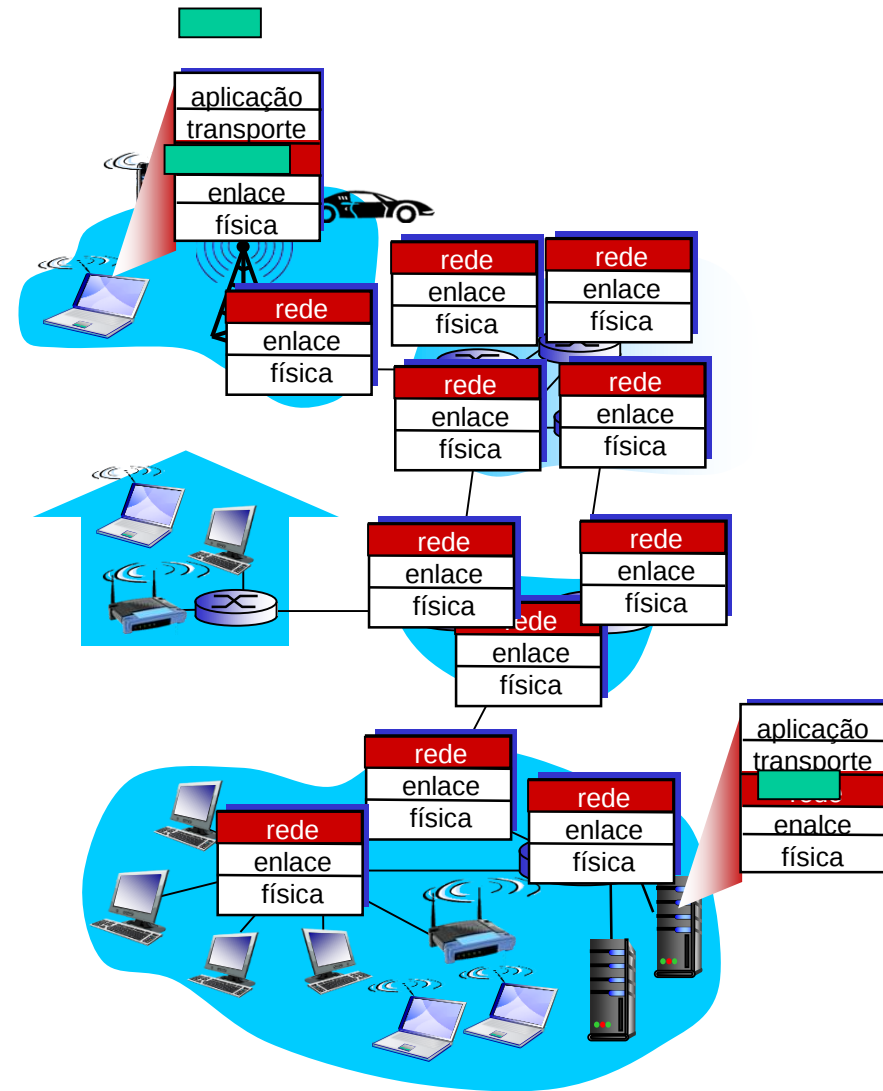
I.4 A camada de rede (I/2)

(Kurose, p. 224 - 306)

17/03/2017

Camada de rede

- ❖ transporta segmentos do *host* fonte para o *host* destino
- ❖ no lado transmissor encapsula segmentos em datagramas
- ❖ no lado receptor, entrega segmentos para camada de transporte
- ❖ protocolos da camada de rede em *todo* *host*, roteador
- ❖ roteador examina campos de cabeçalho em todos os datagramas IP que passam por ele



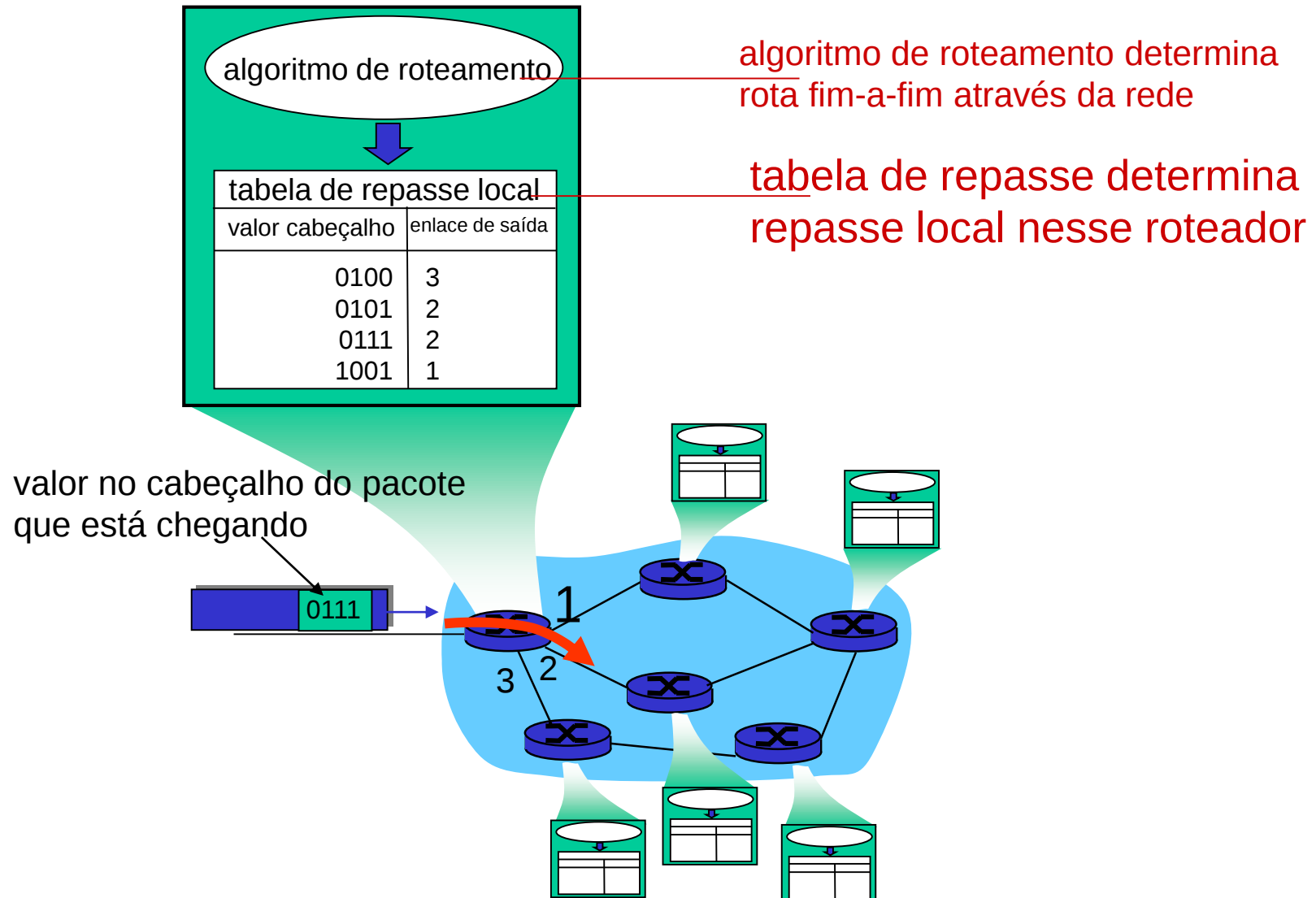
Duas funções chaves da camada de rede

- ❖ *repasse (forwarding)*: mover pacotes da entrada do roteador para saída apropriada (local)
- ❖ *roteamento*: determinar rota tomada por pacotes da fonte ao dest. (global)
 - *algoritmos de roteamento*

analogia:

- ❖ *roteamento*: processo de planejar viagem da origem ao destino
- ❖ *repasse*: processo de passar por uma única encruzilhada

Interação entre roteamento e repasse



Estabelecimento de Conexão (*connection setup*)

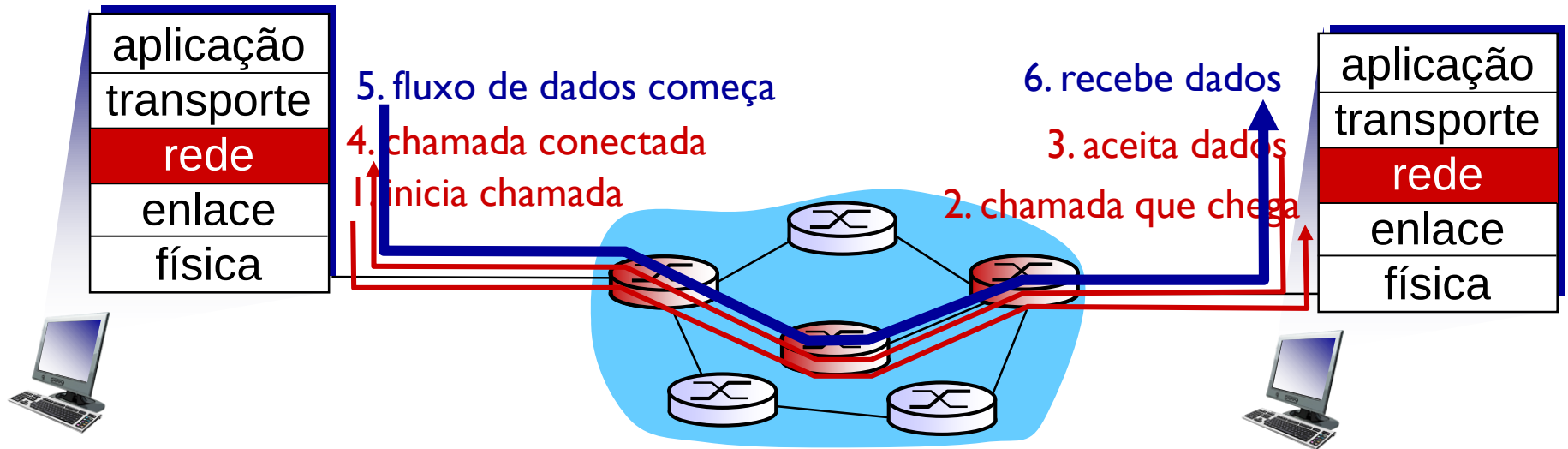
- ❖ 3ª importante função em *algumas* arquiteturas de rede:
 - *ATM, frame relay, MPLS, X.25*
- ❖ antes dos datagramas fluírem, 2 *hosts* finais e roteadores ao longo do caminho estabelecem conexão virtual
 - *roteadores são envolvidos!*
- ❖ serviço de conexão na camada de rede vs na camada de transporte:
 - *rede*: entre 2 *hosts* (pode também envolver roteadores intervenientes no caso de circuitos virtuais (VC))
 - *transporte*: entre dois processos

Modelos de serviços de rede

Arquitetura de rede	Modelo de serviço	Garantias?				Indicação de congestionamento
		Taxa	Perdas	Ordem	Timing	
Internet	Melhor esforço	nenhuma	não	não	não	não (inferido via perdas)
ATM	<u>CBR</u>	taxa constante	sim	sim	sim	não há congestionamento
ATM	<u>VBR</u>	taxa garantida	sim	sim	sim	não há congestionamento
ATM	<u>ABR</u>	mínima garantida	não	sim	não	sim
ATM	<u>UBR</u>	nenhuma	não	sim	não	não

Circuitos virtuais: protocolos de sinalização

- ❖ usados para estabelecer, manter e encerrar VC
- ❖ usados no ATM ([Q.2931](#)), frame-relay, X.25
- ❖ não usados na Internet atual



Redes de datagramas

- ❖ não há estabelecimento de chamada na camada de rede
- ❖ roteadores: sem memória sobre conexões fim-a-fim
 - sem conceito de “conexão” em nível de rede
- ❖ pacotes repassados usando endereço do *host* destino

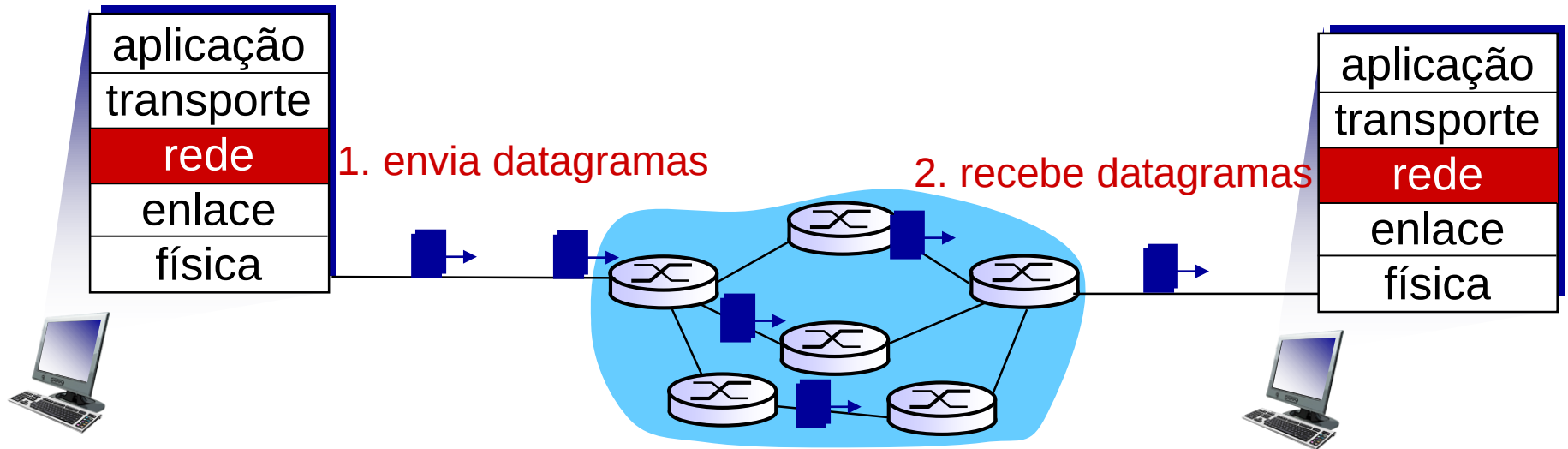


Tabela de repasse de datagramas



Tabela de repasse de datagramas

Intervalo do endereço de destino	Interface do Enlace
11001000 00010111 00010000 00000000 até 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 até 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 até 11001000 00010111 00011111 11111111	2
caso contrário	3

Q: mas o que acontece se intervalos não dividem-se assim tão agradavelmente?

Casamento com prefixo mais longo

casamento com prefixo mais longo

quando busca-se entrada de tabela de repasse para dado endereço de destino (DA), usar prefixo de endereço *mais longo* que casa com endereço de destino.

Intervalo de Endereços de Destino	Interface do enlace
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
caso contrário	3

exemplos:

DA: 11001000 00010111 00010**110** 10100001

qual interface?

DA: 11001000 00010111 00011000 **10101010**

qual interface?

Rede de datagramas ou VC: por que?

Internet (datagramas)

- ❖ dados trocados entre computadores
 - serviço “elástico”, sem req. *timing* estritos
- ❖ muitos tipos de enlaces
 - características diferentes
 - difícil serviço uniforme
- ❖ sistemas finais “inteligentes” (computadores)
 - podem adaptar-se, realizar controle de desempenho e recuperação de erro
 - ***simples dentro da rede, complexidade na “borda”***

ATM (VC)

- ❖ evoluiu da telefonia
- ❖ conversação humana:
 - requerimentos de confiabilidade e *timing* estrito
 - necessidade de garantia de serviço
- ❖ sistemas finais “burros”
 - telefones
 - ***complexidade dentro da rede***

Capítulo 4: conteúdo

4.1 introdução

4.2 redes a circuitos virtuais e de datagramas

4.3 o que tem dentro de um roteador?

4.4 IP: *Internet Protocol*

- formato do datagrama
- endereçamento IPv4
- ICMP
- IPv6

4.5 algoritmos de roteamento

- estado de enlace
- vetor de distâncias
- roteamento hierárquico

4.6 roteamento na Internet

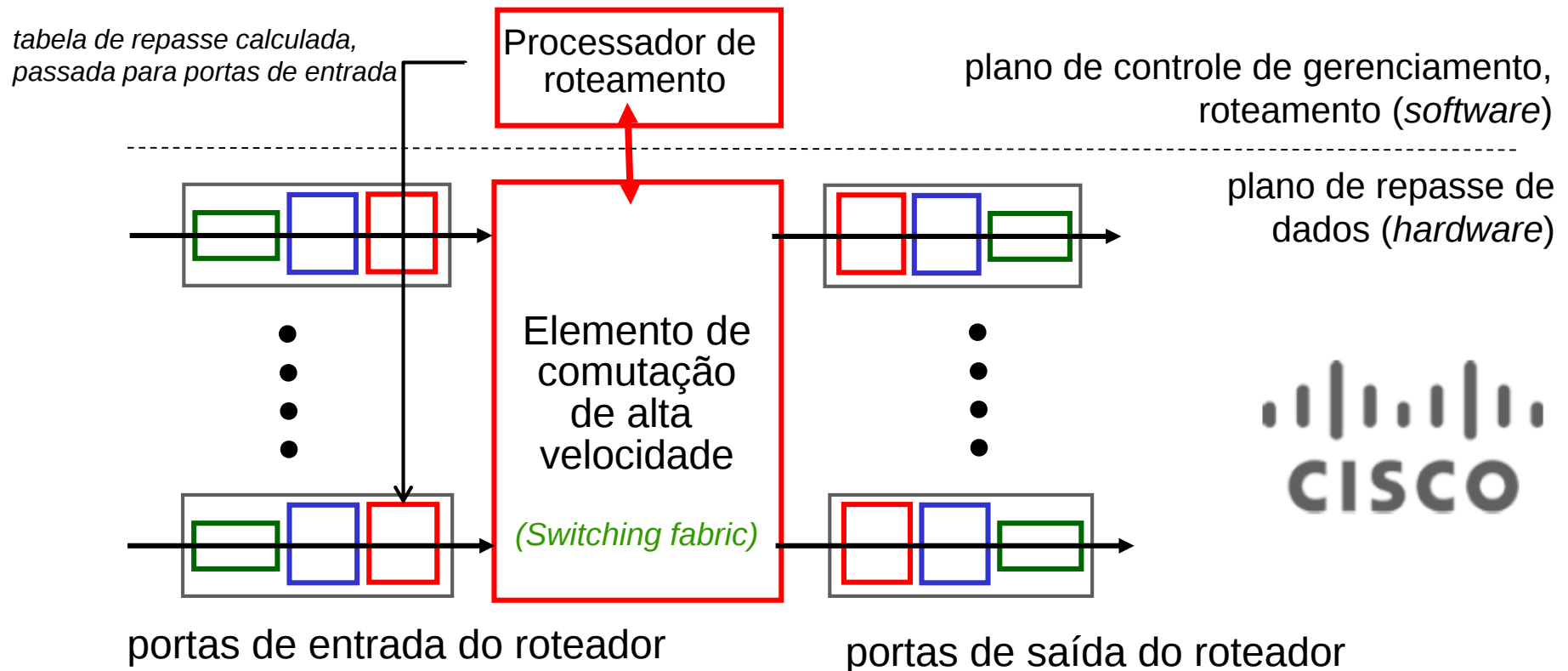
- RIP
- OSPF
- BGP

4.7 roteamento com *broadcast* e *multicast*

Arquitetura de roteadores ([CISCO](#)/[Huawei](#)/[Alcatel-Lucent](#)/[Juniper](#))

2 funções chaves do roteador:

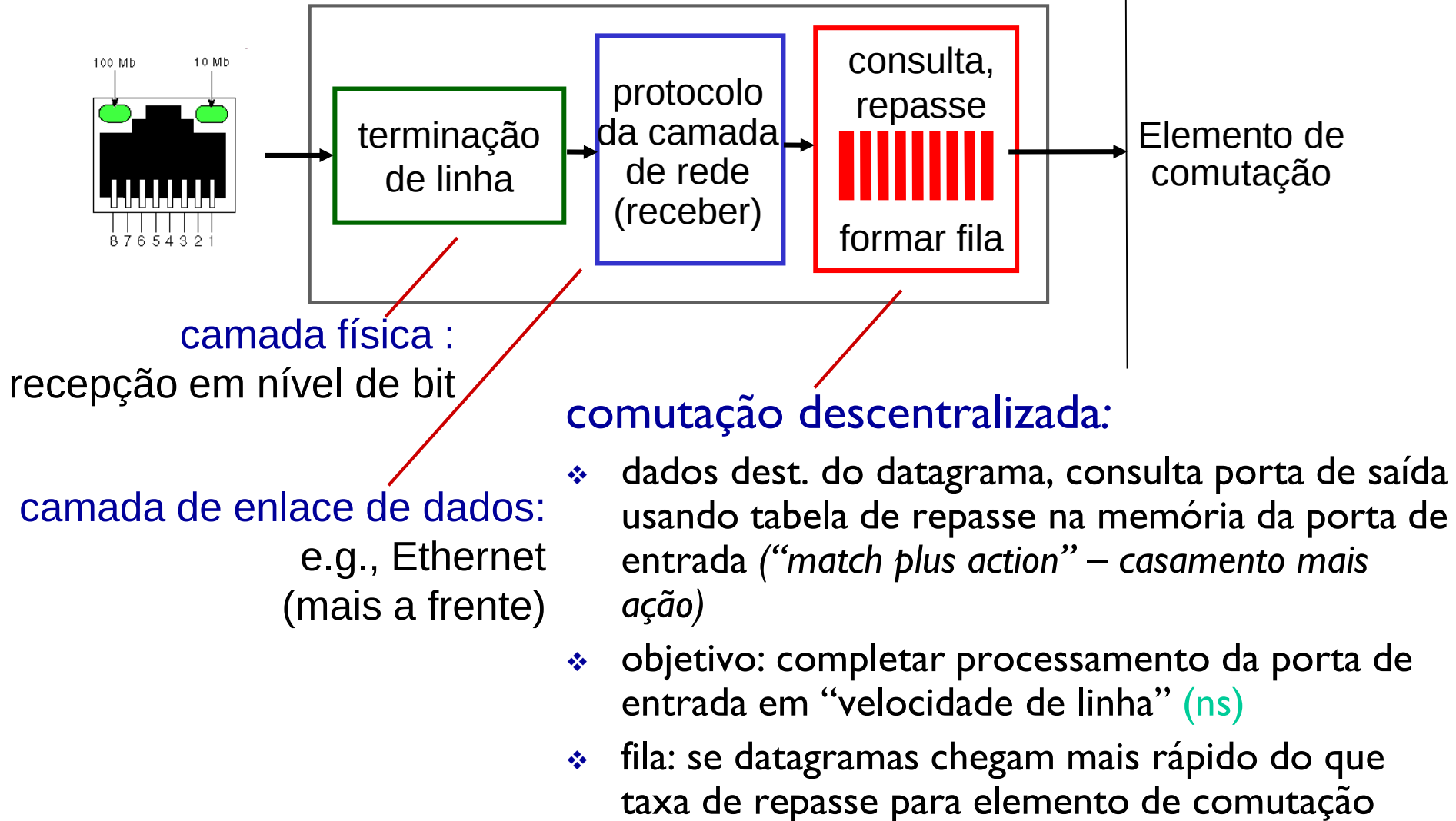
- ❖ rodar protocolos e algoritmos de roteamento (RIP, OSPF, BGP)
- ❖ *repassar* ou *comutar* datagramas de enlace de entrada para enlace de saída



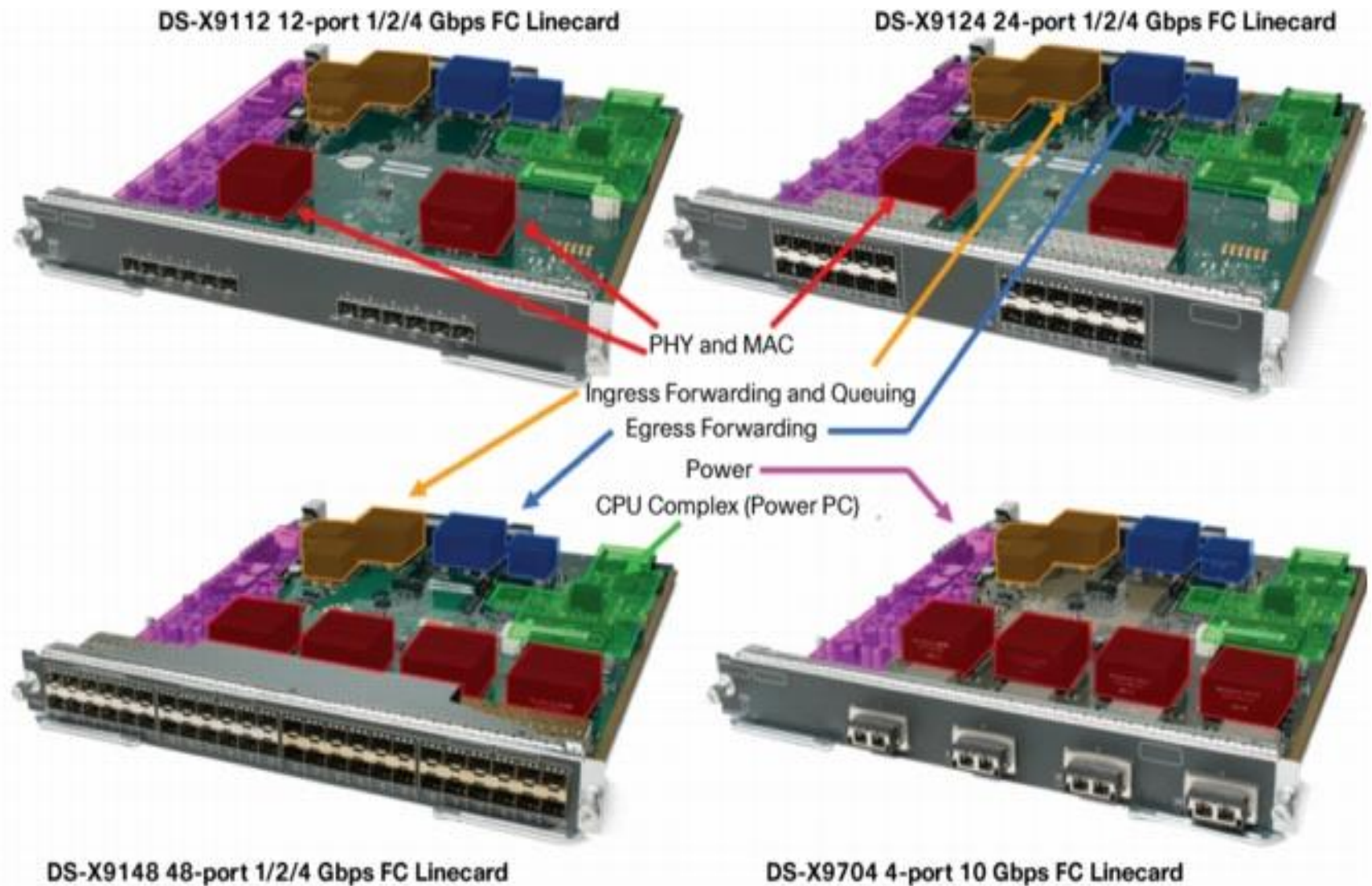
Exemplo: Arquitetura da Família Cisco MDS 9000



Funções da porta de entrada

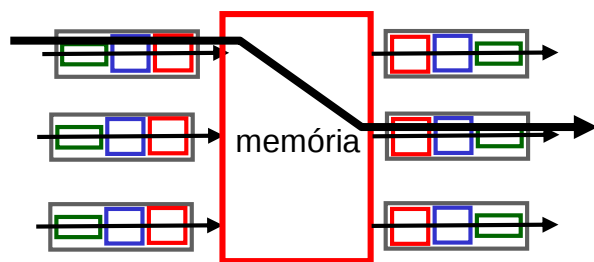


Exemplo: Arquitetura da Família Cisco MDS 9000

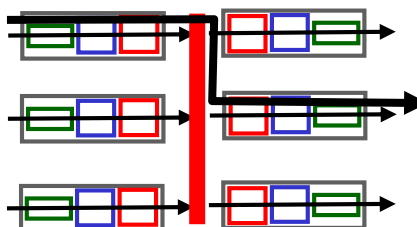


Elementos de comutação (*Switching fabrics*)

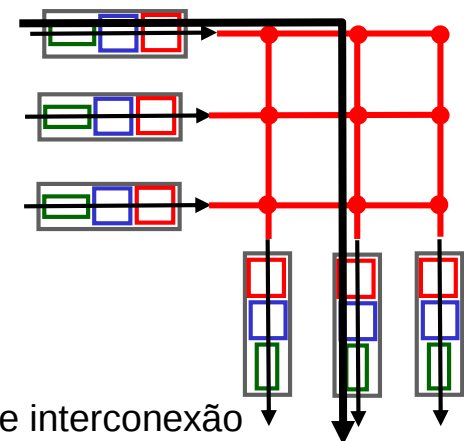
- ❖ transfere pacote do *buffer* de entrada para o *buffer* de saída apropriado
- ❖ taxa de comutação: taxa em que pacotes podem ser transferidos de entradas para saídas
 - muitas vezes medidos em múltiplos da taxa de linha de entrada/saída
 - N entradas: desejável taxa de comutação N vezes taxa de linha
- ❖ 3 tipos de elementos de comutação



memória

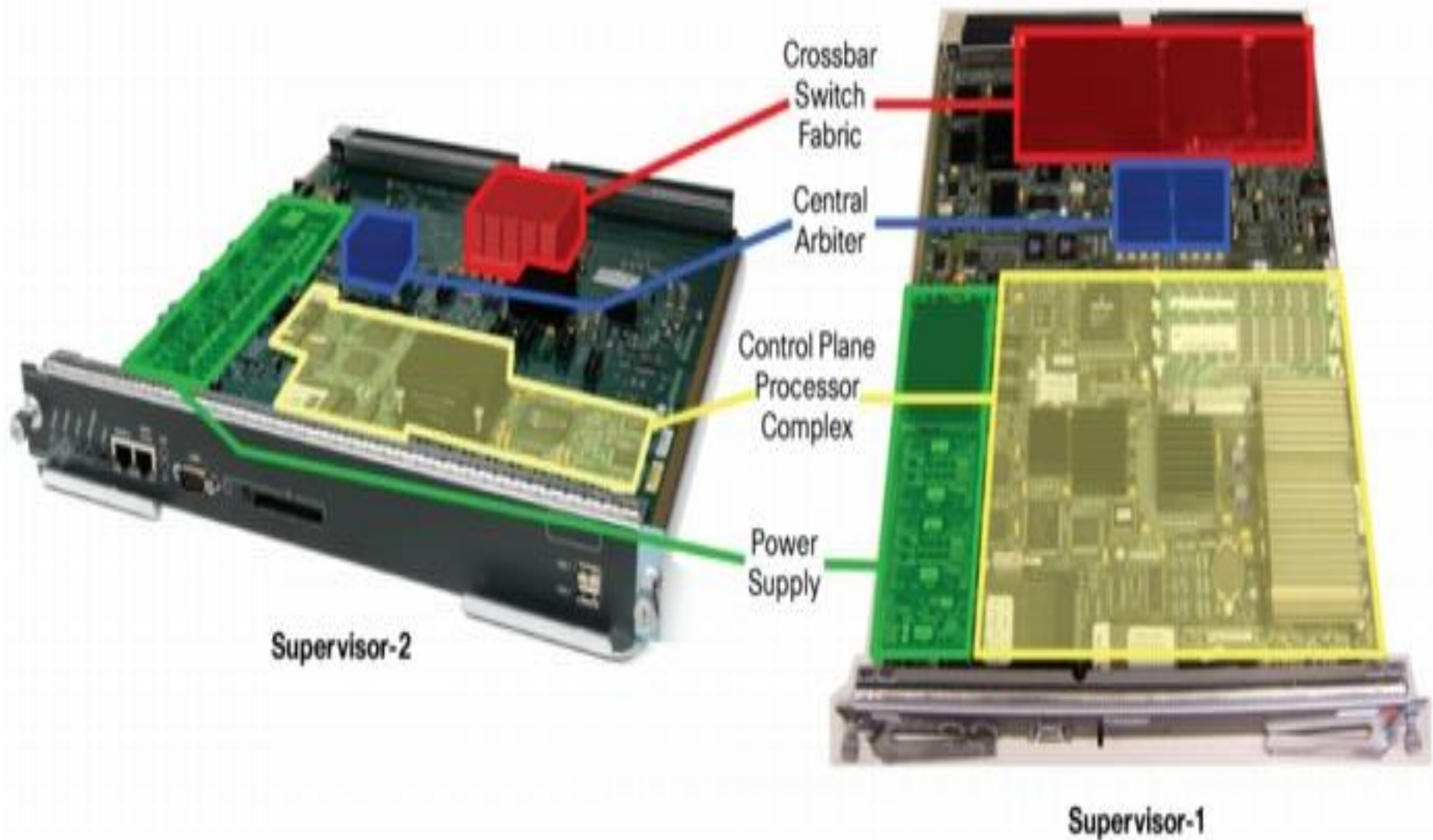


barramento (*bus*)



Rede de interconexão
(*crossbar*)

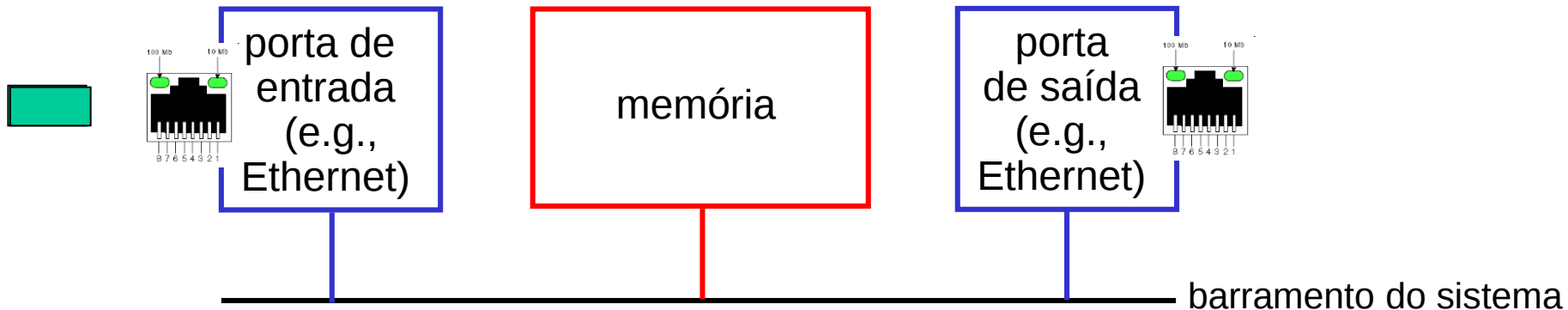
Exemplo: Arquitetura da Família Cisco MDS 9000



Comutação via memória

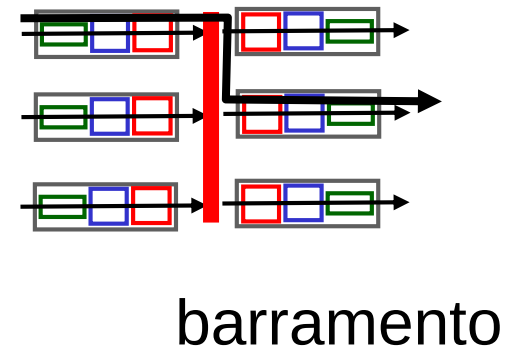
1ª geração de roteadores:

- ❖ computadores tradicionais com comutação sob controle direto da CPU
- ❖ pacote copiado para memória do sistema (1 de cada vez)
- ❖ velocidade limitada pela velocidade da memória ($B/2$ pacotes/s; B = velocidade da memória)
- ❖ atuais também podem usar; mas consulta realizada por processadores nas placas de linha de entrada – [8500 Catalyst Cisco](#)



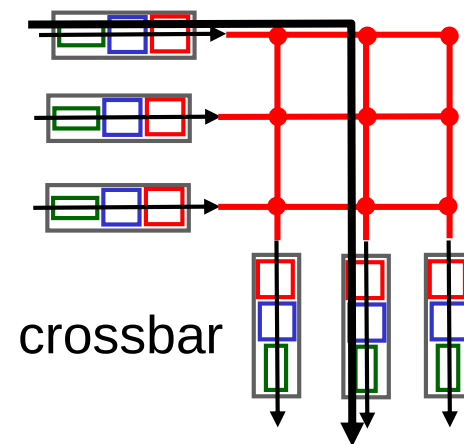
Comutação via barramento

- ❖ datagrama da memória da porta de entrada para porta de saída via barramento
- ❖ cabeçalho interno define porta de saída
- ❖ apenas 1 pacote pode cruzar o barramento de cada vez
- ❖ *contenção pelo barramento*: velocidade de comutação limitada pela velocidade do barramento
- ❖ Cisco 5600 barramento de 32 Gbps: velocidade suficiente para roteadores de acesso e de empresas

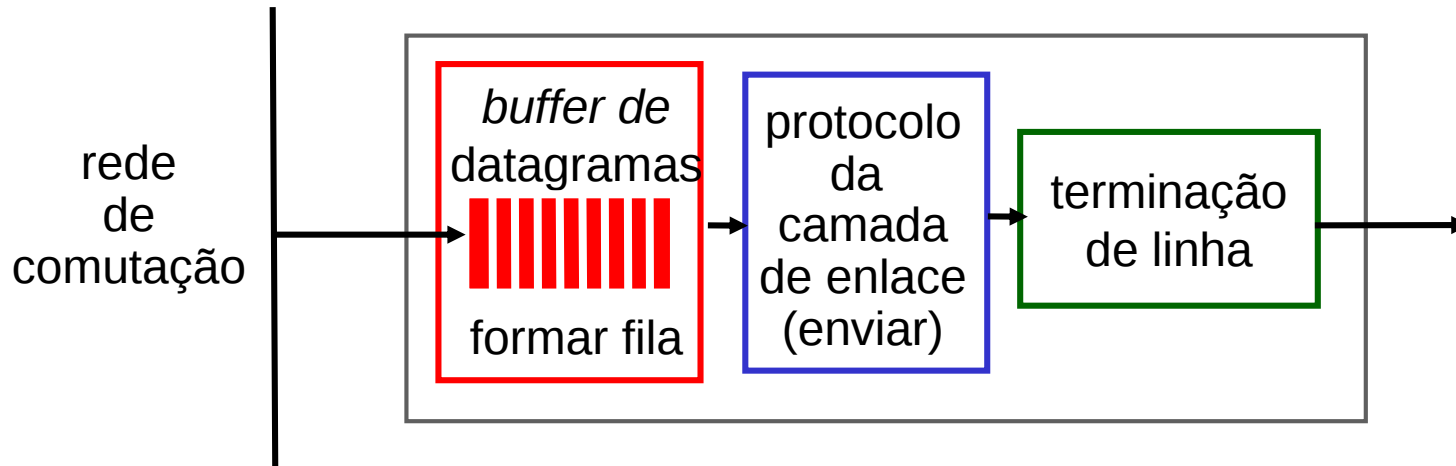


Comutação via crossbar

- ❖ $2N$ barramentos ligando N entradas a N saídas – cruzamento pode ser aberto ou fechado pelo controlador do *switch fabric*
- ❖ supera limitações de velocidade do barramento - capaz de repassar vários pacotes em paralelo
- ❖ *banyan networks*, *crossbar*, e outras redes de interconexão inicialmente desenvolvidas para conectar processadores em arquiteturas multiprocessadores
- ❖ projeto avançado: fragmentando datagramas em células de comprimento fixo, comuta células através da rede
- ❖ Cisco I2000: comuta 60 Gbps através da rede de interconexão
- ❖ Obs: Cuidado com erros na versão em português do livro!!

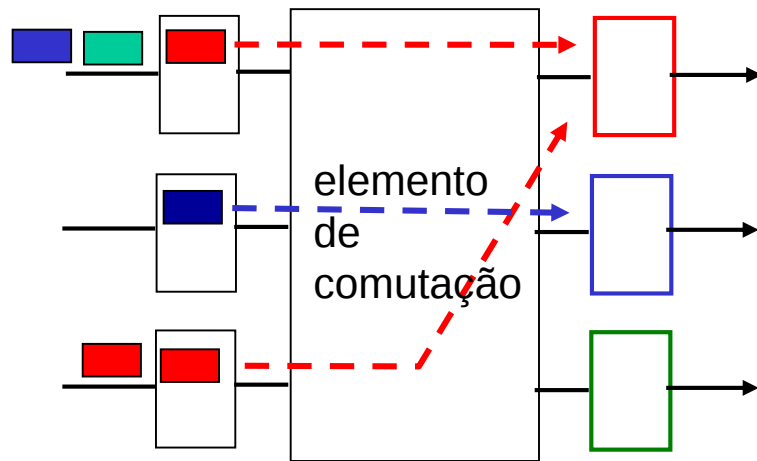


Portas de saída

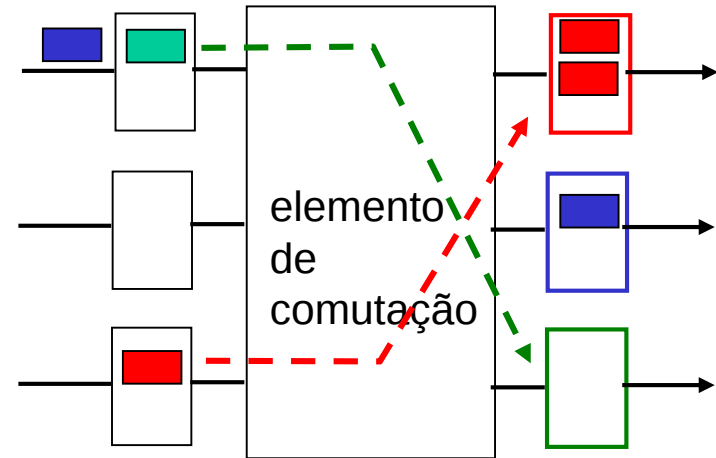


- ❖ **buffer** necessário quando datagramas chegam da rede de comutação mais rapidamente do que a taxa de transmissão – *isso é comum!*
- ❖ **disciplina de agendamento:** escolhe entre os datagramas enfileirados para transmissão
 - ❖ FCFS (*First-come, first-served*)
 - ❖ WFQ ([*Weighted fair queuing*](#))
 - ❖ Priorizar pacotes de acordo com fonte? Neutralidade...

Fila na porta de saída



em t , mais pacotes
da entrada para saída



1 tempo de pacote
depois

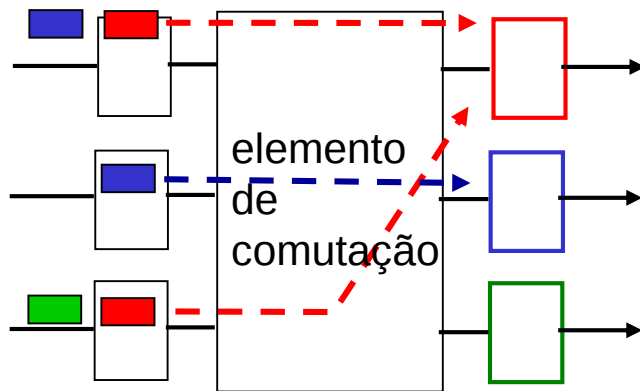
- ❖ colocar em *buffer* quando taxa de chegada excede velocidade de linha de saída
- ❖ *filas (atraso) e perdas devido a transbordamento do buffer da porta de saída!*
- ❖ *Qual pacote descartar? Active Queue Management*

Qual tamanho de *buffer*?

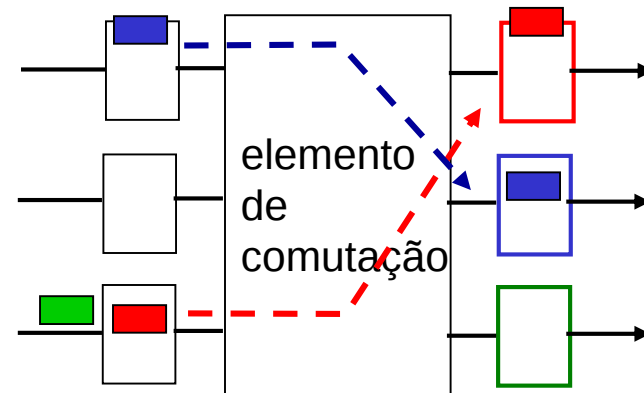
- ❖ Pergunta de projeto importante!
- ❖ [RFC 3439](#) (2002) regra prática: *buffer* médio igual ao RTT “médio” (por exemplo, 250 ms) vezes capacidade do enlace C
 - e.g., C = enlace de 10 Gbps : *buffer* 2.5 Gbit
- ❖ recomendação recente: com N fluxos TCP, *buffer* de
$$\frac{RTT \cdot C}{\sqrt{N}}$$
- ❖ Ver referências na pág. 242 do Kurose

Filas na porta de entrada

- ❖ comutação mais lenta do que portas de entrada combinadas
-> filas podem ocorrer nas portas de entrada
 - *atraso de fila e perdas devido a transbordamento dos buffers de entrada!*
- ❖ *bloqueio Head-of-the-Line (HOL)* : datagrama na frente da fila não deixa que outros na fila movam-se para frente



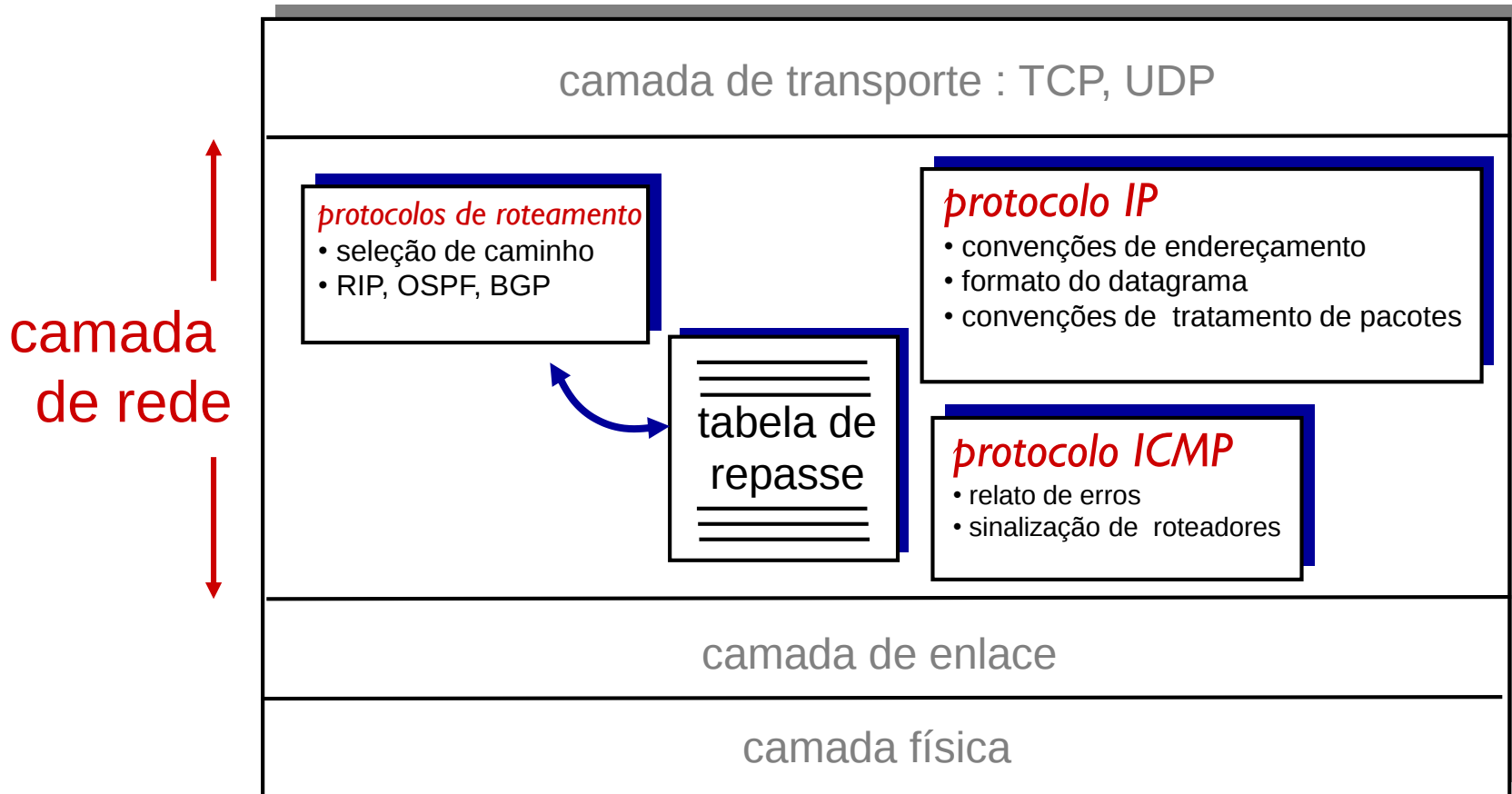
contenção na porta de saída:
apenas 1 datagrama vermelho pode
ser transferido.
pacote vermelho inferior é bloqueado



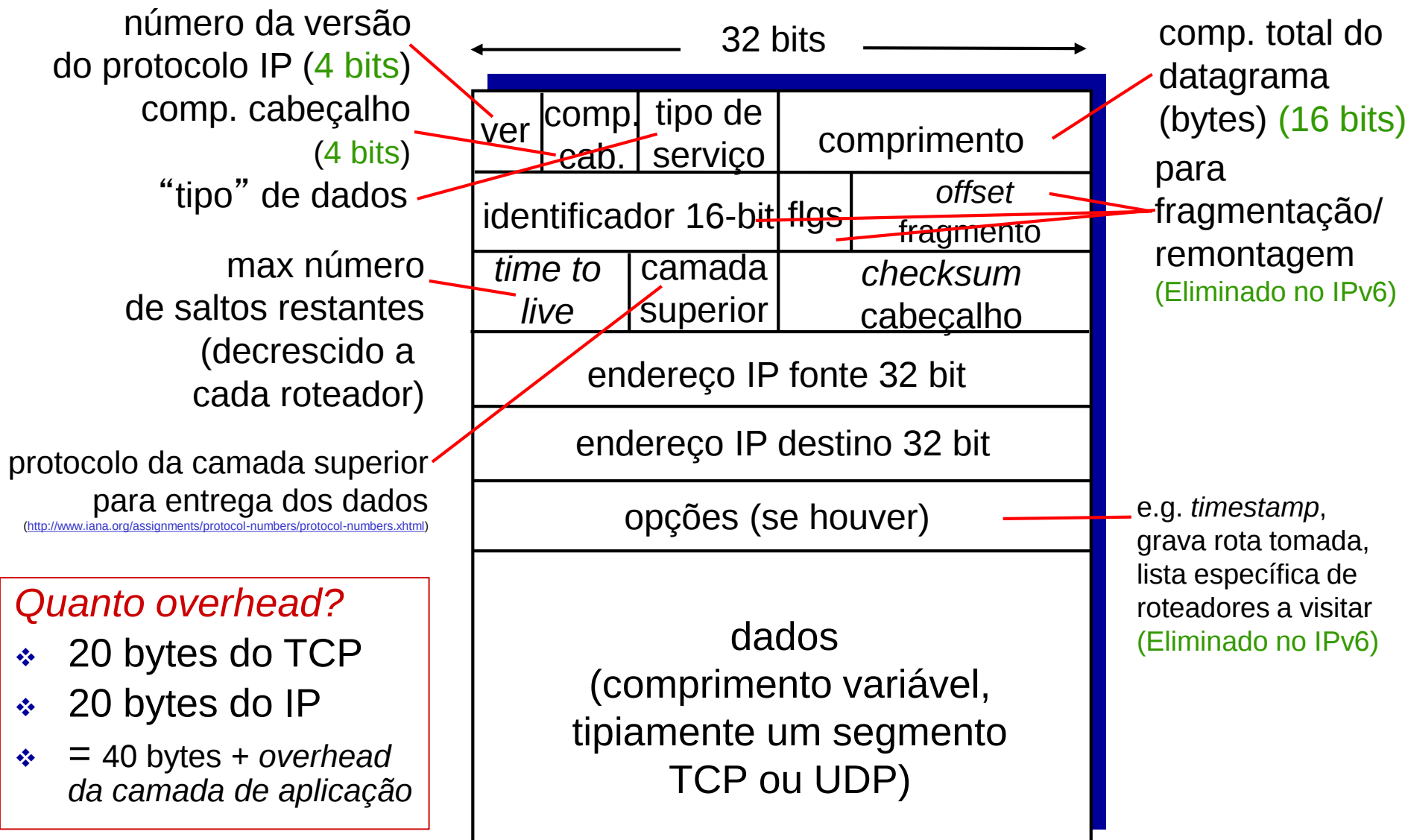
1 tempo de pacote
depois: pacote verde
sofre bloqueio HOL

A camada de rede Internet

funções da camada de rede em *hosts* e roteadores:



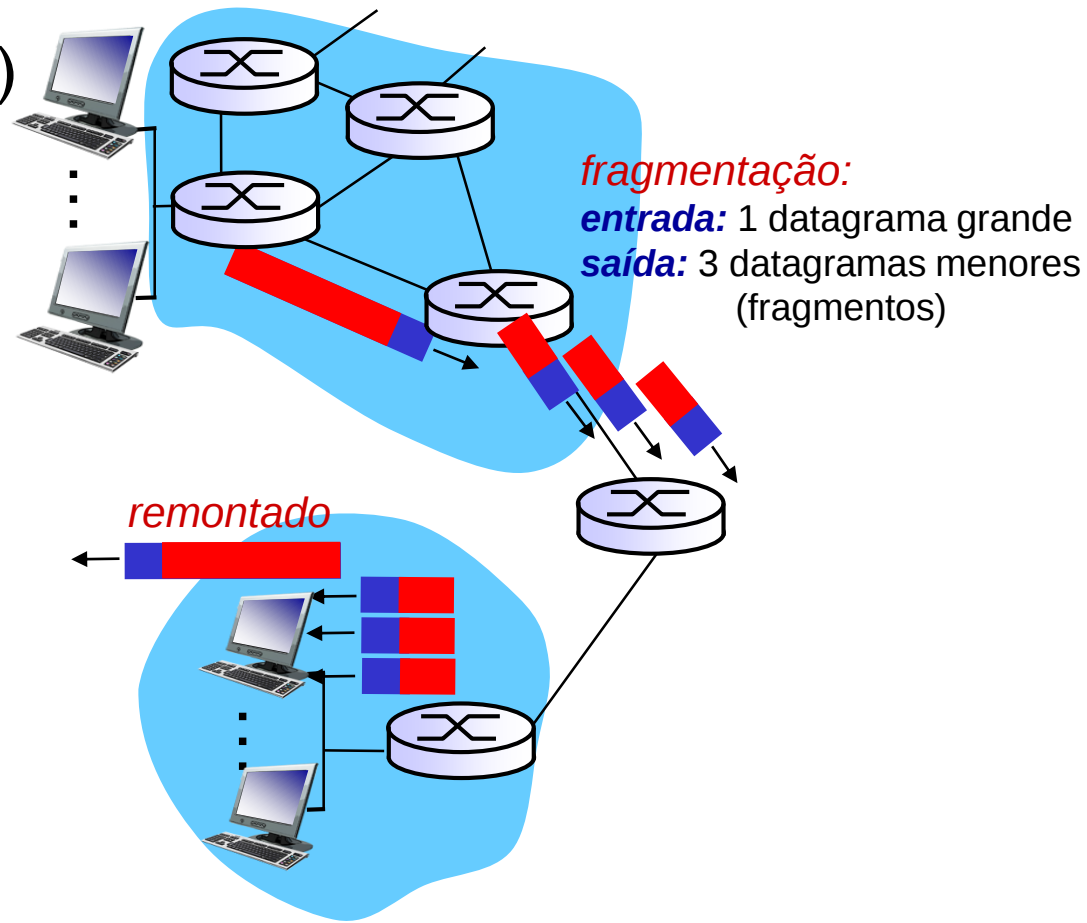
Formato do datagrama IPv4 [[RFC760 \(1980\)](#), ..., [RFC6864 \(2013\)](#)]



Fragmentação e remontagem IP

- ❖ enlaces tem MTU
(*Maximum Transmission Unit*)
 - maior quadro da camada de enlace possível (ex. Ethernet I 500 bytes)
 - diferentes tipos de enlace, diferentes MTUs

- ❖ datagrama IP grande dividido (“fragmentado”) dentro da rede
 - 1 datagrama torna-se diversos datagramas
 - “remontado” apenas no destino final
 - bits no cabeçalho IP usados para identificar e ordenar fragmentos relacionados



Fragmentação e remontagem IP

exemplo:

- ❖ datagrama 4 000 bytes
- ❖ MTU = 1 500 bytes

	compr.	ID	fragflag	offset
	=4000	=x	=0	=0

*1 datagrama grande torna-se
vários datagramas menores*

1480 bytes no
campo de dados

offset =
1480/8
(em múltiplos
de 8 bytes)

	compr.	ID	fragflag	offset
	=1500	=x	=1	=0

	compr.	ID	fragflag	offset
	=1500	=x	=1	=185

	compr.	ID	fragflag	offset
	=1040	=x	=0	=370

- ❖ Problemas:
 - ❖ sobrecarga de roteadores e sistemas finais
 - ❖ Vulnerável a ataques DoS ([Jolt2](#) – pequenos segmentos com offset não nulo enviados)
 - ❖ IPv6 – sem fragmentação!
 - ❖ Ver http://media.pearsoncmg.com/aw/aw_kurose_network_2/applets/ip/ipfragmentation.html