

Redes de Computadores

A Camada de Rede: Plano de Dados

Nota: a maioria dos slides dessa apresentação são traduzidos ou adaptados dos slides disponibilizados gratuitamente pelos autores do livro [KUROSE, James F. e ROSS, Keith W. *Computer Networking: A Top-Down Approach*. 8th Edition](#). Pearson, 2020. Todo o material pertencente aos seus respectivos autores está protegido por direito autoral.

Camada de Rede: nossos objetivos

- entender princípios por trás dos serviços de camada de rede, focando no plano de dados:
- modelos de serviço da camada de rede
- encaminhamento versus roteamento
- como funciona um roteador
- endereçamento
- encaminhamento generalizado
 - arquitetura da Internet
- instanciação e implementação na Internet
- protocolo IP
 - NAT, *middleboxes*

Camada de rede: roteiro do “plano de dados”

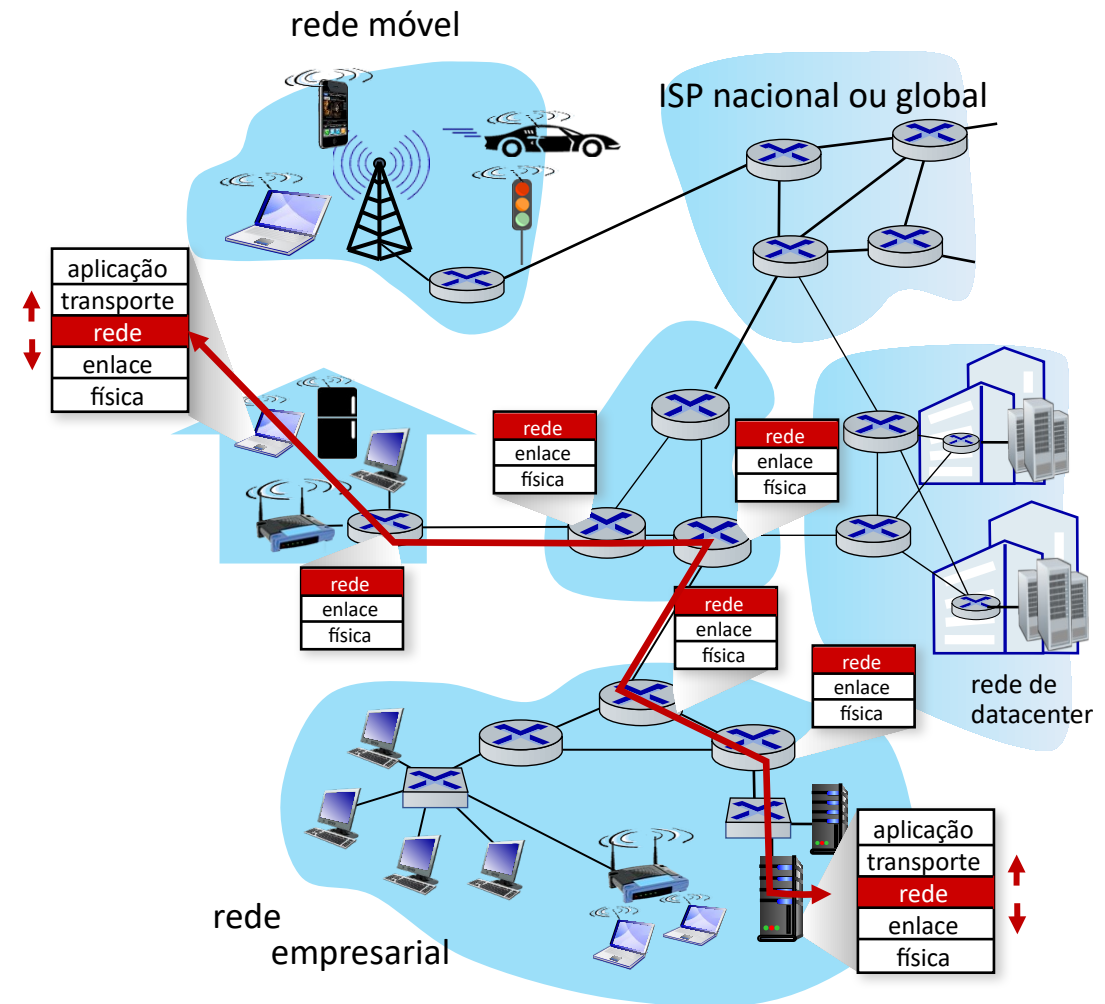
- Camada de rede: visão geral
- plano de dados
 - plano de controle
- O que há dentro de um roteador
 - portas de entrada, comutação, portas de saída
 - gerenciamento de buffer, agendamento
- IP: o Protocolo da Internet
 - formato de datagrama
 - endereçamento
 - tradução de endereço de rede (NAT)
 - IPv6



- Encaminhamento Generalizado e SDN
 - Correspondência+ação
 - OpenFlow: correspondência+ação em ação
- Middleboxes

Serviços e protocolos da camada de rede

- transporta segmento do hospedeiro emissor ao receptor
 - **emissor:** encapsula segmentos em datagramas, passa para a camada de enlace
 - **receptor:** entrega segmentos para protocolo da camada de transporte
- protocolos de camada de rede em *todos os dispositivos de Internet*: hospedeiros e roteadores
- **roteadores:**
 - examinam campos de cabeçalho em todos os datagramas IP que passam por eles
 - move datagramas de portas de entrada para portas de saída para transferi-los ao longo do caminho final



Duas funções-chave da camada de rede

funções da camada de rede:

- *encaminhamento*: move pacotes do enlace de entrada de um roteador para o enlace de saída apropriado
- *roteamento*: determina rota tomada por pacotes da origem para o destino
 - *algoritmos de roteamento*

analogia: fazer uma viagem

- *encaminhamento*: processo de passar por um único cruzamento
- *roteamento*: processo de planejar a viagem da origem ao destino



encaminhamento



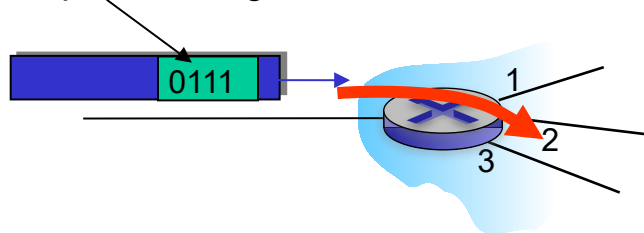
roteamento

Camada de Rede: plano de dados e plano de controle

Plano de dados:

- função *local*, por roteador
- determina como o datagrama que chega em uma porta de entrada do roteador é encaminhado para uma porta de saída

valores em cabeçalhos
de pacote chegando

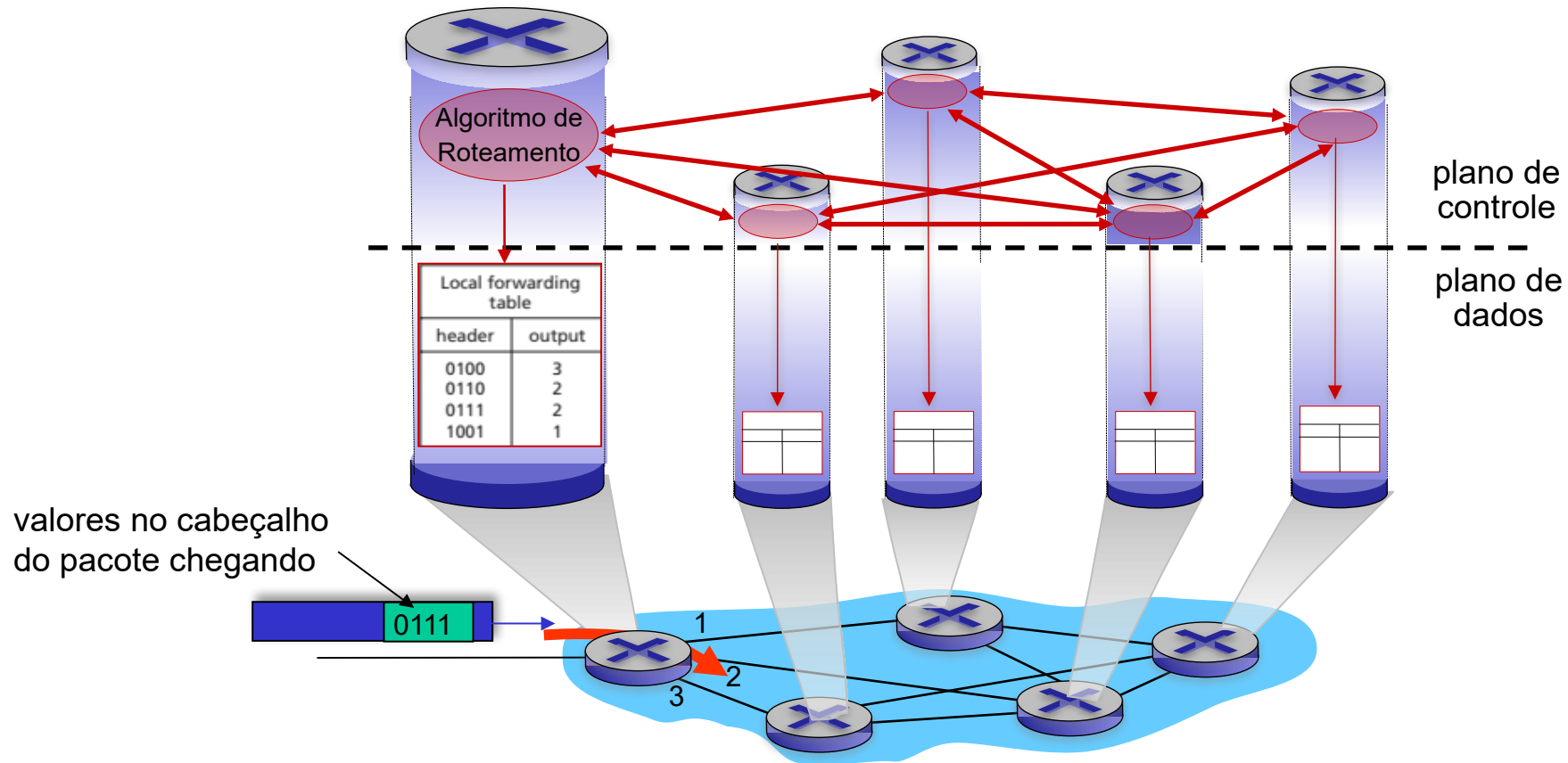


Plano de controle:

- lógica *em toda a rede*
- determina como o datagrama é roteado entre roteadores ao longo do caminho fim-a-fim do hospedeiro de origem ao de destino
- duas abordagens de plano de controle:
 - *algoritmos de roteamento tradicionais*: implementados em roteadores
 - *software-defined networking (SDN – rede definida por software)*: implementada em servidores (remotos)

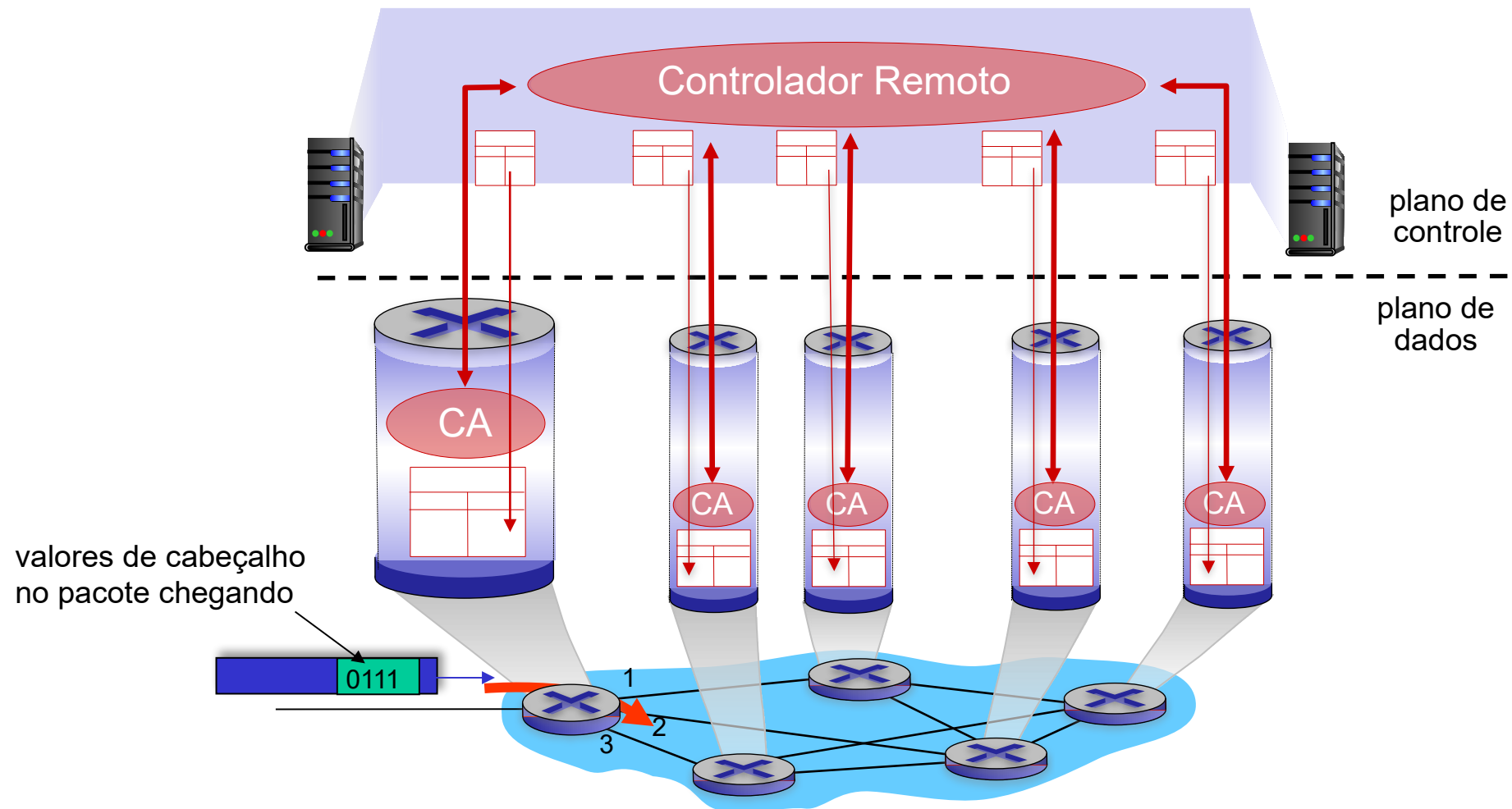
Plano de controle por roteador

Componentes individuais do algoritmo de roteamento *em todo e cada roteador* interagem no plano de controle



Plano de controle de Rede Definida por Software (SDN)

Controlador remoto computa e instala tabelas de encaminhamento nos roteadores



Modelo de serviço de rede

Q: Qual *modelo de serviço* para “canal” transportando datagramas do emissor para o receptor?

exemplo de serviços para datagramas *individuais*:

entrega garantida

- entrega garantida com menos de 40 ms de atraso

exemplos de serviços para um *fluxo* de datagramas:

- entrega de datagramas em ordem
- largura de banda mínima garantida para fluxo
- restrições às alterações no espaçamento entre pacotes

Modelo de serviço da camada de rede

Arquitetura de rede	Modelo de serviço	Garantias de Qualidade de Serviço (QoS) ?			
		Banda	Perda	Ordem	Tempo
Internet	melhor esforço	nenhuma	não	não	não

Modelo de serviço de “melhor esforço” da Internet

Sem garantias sobre:

entrega de datagrama bem-sucedida para o destino
tempo ou ordem de entrega

- i. largura de banda disponível para o fluxo fim-a-fim

Modelo de serviço da camada de rede

Arquitetura de rede	Modelo de serviço	Garantias de Qualidade de Serviço (QoS) ?			
		Banda	Perda	Ordem	Tempo
Internet	melhor esforço	nenhuma	não	não	não
ATM	taxa constante	taxa constante	sim	sim	sim
ATM	taxa disponível	tx. mín. garantida	não	sim	não
Internet	Intserv Guaranteed (RFC 1633)	sim	sim	sim	sim
Internet	Diffserv (RFC 2475)	possível	possivel-mente	possivel-mente	não

Reflexões sobre o serviço de melhor esforço:

- **simplicidade do mecanismo** permitiu que a Internet fosse amplamente implantada e adotada
- **provisionamento de largura de banda** suficiente permite que o desempenho de aplicações de tempo real (por exemplo, voz interativa, vídeo) seja “bom o suficiente” na “maior parte do tempo”
- **serviços da camada de aplicação replicados e distribuídos** (datacenters, redes de distribuição de conteúdo) conectando perto das redes dos clientes permitindo que os serviços sejam fornecidos a partir de múltiplos locais
- controle de congestionamento de serviços “elásticos” ajuda

É difícil argumentar com o sucesso do modelo de serviço de melhor esforço

Camada de rede: roteiro do “plano de dados”

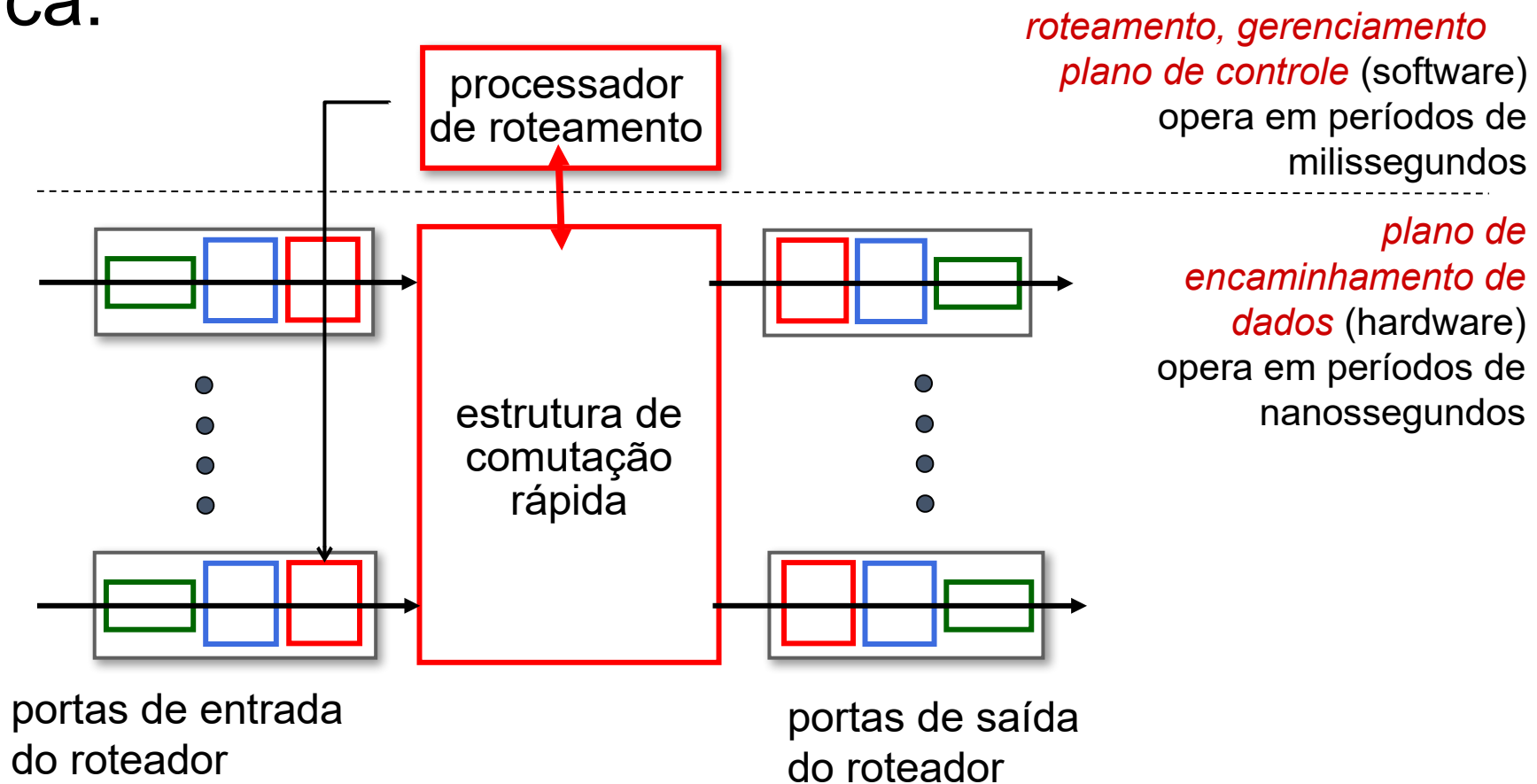
- Camada de rede: visão geral
- plano de dados
 - plano de controle
- O que há dentro de um roteador
 - portas de entrada, comutação, portas de saída
 - gerenciamento de buffer, agendamento
- IP: o Protocolo da Internet
 - formato de datagrama
 - endereçamento
 - tradução de endereço de rede (NAT)
 - IPv6



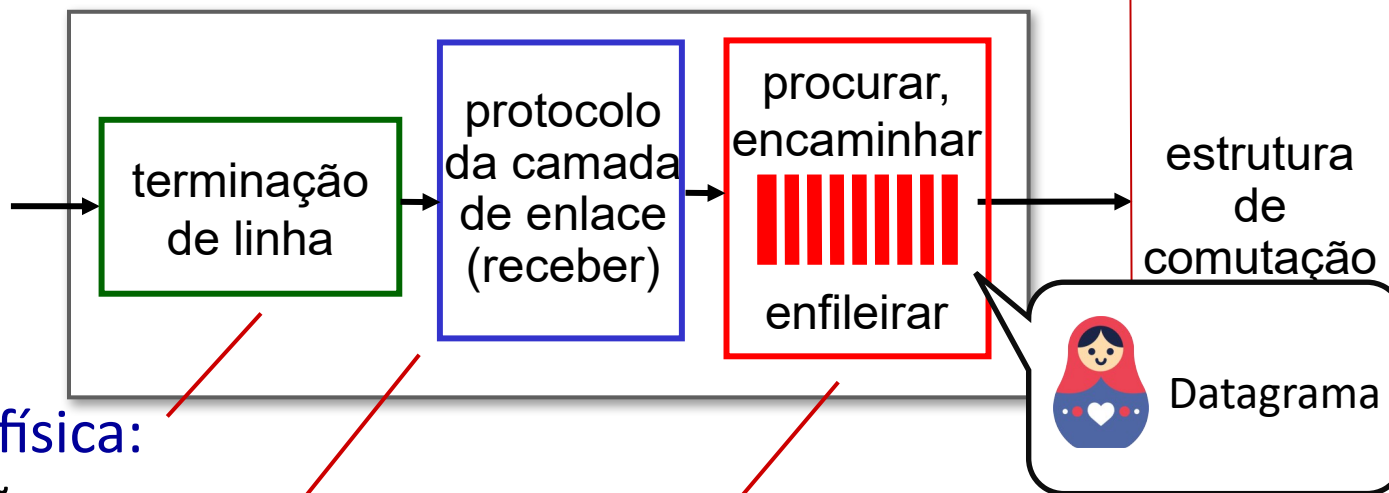
- Encaminhamento Generalizado e SDN
 - Correspondência+ação
 - OpenFlow: correspondência+ação em ação
- Middleboxes

Visão geral da arquitetura do roteador

visão de alto nível de uma arquitetura de roteador genérica:



Funções da porta de entrada



camada física:
recepção em
nível de bits

camada de enlace:
ex.: Ethernet
(capítulo 6

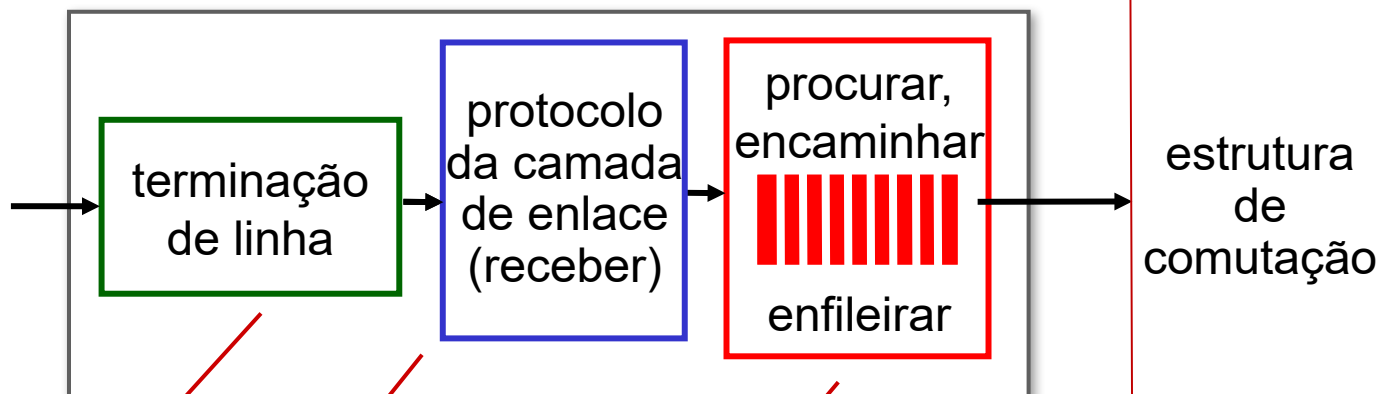


Quadro

comutação descentralizada:

- usando valores de campo de cabeçalho, procurar porta de saída usando tabela de encaminhamento na memória da porta de entrada (*"correspondência mais ação"*)
- objetivo: processamento completo da porta de entrada na 'velocidade da linha'
- **enfileiramento na porta de entrada:** ocorre se os datagramas chegarem mais rápido do que a taxa de encaminhamento para a estrutura de comutação

Funções da porta de entrada



camada física:
recepção em
nível de bits

camada de enlace:
ex.: Ethernet
(capítulo 6)

comutação descentralizada:

- usando valores de campo de cabeçalho, procurar porta de saída usando tabela de encaminhamento na memória da porta de entrada (*"correspondência mais ação"*)
- **encaminhamento baseado em destino:** encaminha com base apenas no endereço IP de destino (tradicional)
- **encaminhamento generalizado:** encaminha com base em qualquer conjunto de valores de campo de cabeçalho

Encaminhamento baseado em destino

<i>forwarding table</i>	
Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: mas o que acontece se os intervalos não se dividem tão bem?

Correspondência com prefixo mais longo

correspondência com prefixo mais longo

ao procurar a entrada da tabela de encaminhamento para determinado endereço de destino, use o prefixo de endereço *mais longo* que corresponde ao endereço de destino.

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

exemplos:

11001000 00010111 00010110 10100001 qual interface?

11001000 00010111 00011000 10101010 qual interface?

Correspondência com prefixo mais longo

correspondência com prefixo mais longo

ao procurar a entrada da tabela de encaminhamento para determinado endereço de destino, use o prefixo de endereço *mais longo* que corresponde ao endereço de destino.

Faixa de Endereços de Destino	Interface do enlace
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001 correspondência! 0011*** *****	2
caso contrário	3

exemplos:

11001000 00010111 00010	110 10100001	qual interface?
11001000 00010111 00011000	10101010	qual interface?

Correspondência com prefixo mais longo

correspondência com prefixo mais longo

ao procurar a entrada da tabela de encaminhamento para determinado endereço de destino, use o prefixo de endereço *mais longo* que corresponde ao endereço de destino.

Faixa de Endereços de Destino	Interface do enlace
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherwise	3

correspondência!

exemplos:

11001000 00010111 00010110 10100001	qual interface?
11001000 00010111 00011000 10101010	qual interface?

Correspondência com prefixo mais longo

correspondência com prefixo mais longo

ao procurar a entrada da tabela de encaminhamento para determinado endereço de destino, use o prefixo de endereço *mais longo* que corresponde ao endereço de destino.

Faixa de Endereços de Destino	Interface do enlace
11001000 00010111 00010*** *****	0
11001000 00010111 00011000 *****	1
11001000 00010111 00011*** *****	2
otherw	3

correspondência!

exemplos:

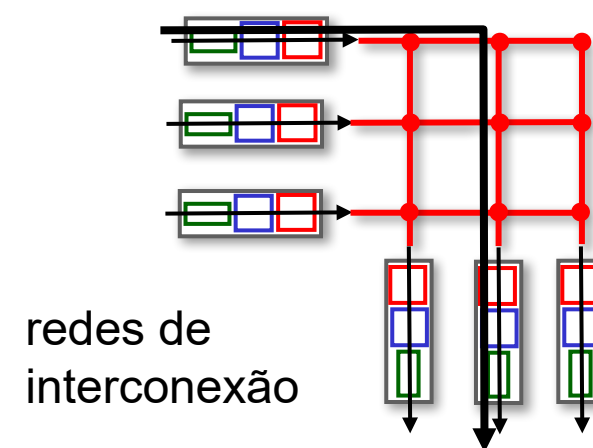
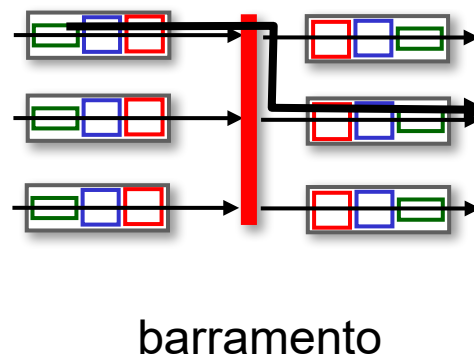
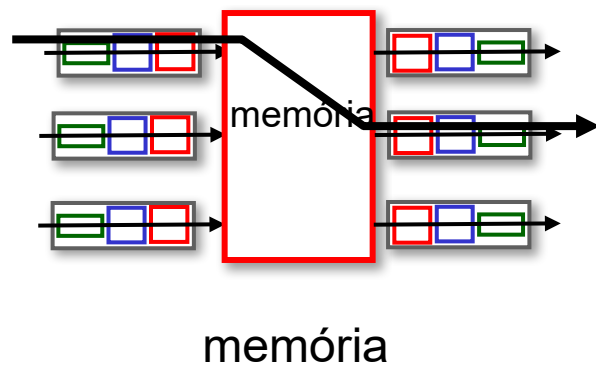
11001000 00010111 00010110 10100001	qual interface?
11001000 00010111 00011000 10101010	qual interface?

Correspondência com prefixo mais longo

- vamos ver *por que* a correspondência de prefixo mais longo é usada em breve, quando estudarmos endereçamento
- correspondência de prefixo mais longo: frequentemente realizada usando memórias endereçáveis de conteúdo ternário (*ternary content addressable memories* - TCAMs)
 - *conteúdo endereçável*: apresenta endereço para TCAM: recupera endereço em um ciclo de clock, independentemente do tamanho da tabela
 - Cisco Catalyst: ~1M entradas de tabela de roteamento em TCAM

Estruturas de comutação

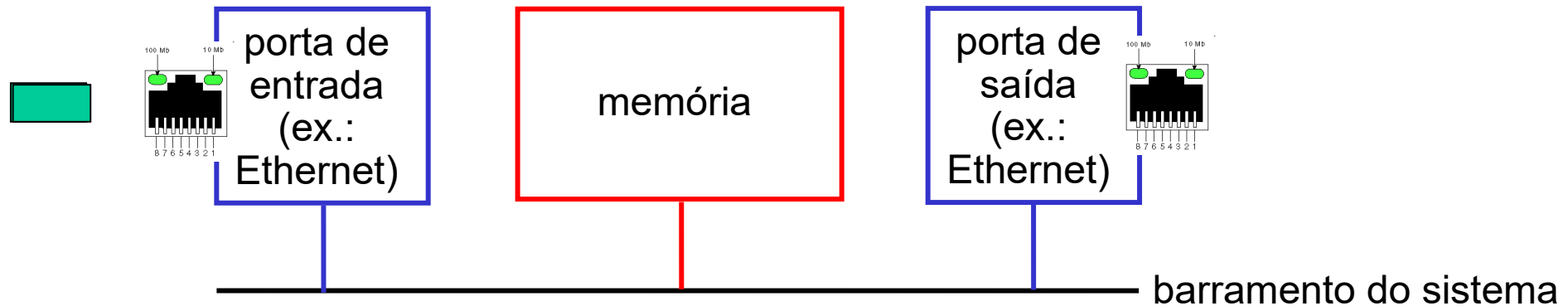
- transferem pacotes do enlace de entrada para o enlace de saída apropriado
- **taxa de comutação:** taxa em que pacotes podem ser transferidos de entradas para saídas
 - frequentemente medida como múltiplos da taxa de linha de entrada/saída
 - N entradas: taxa de comutação de N vezes a taxa de linha é desejável
- três principais tipos de estruturas de comutação:



Comutação via memória

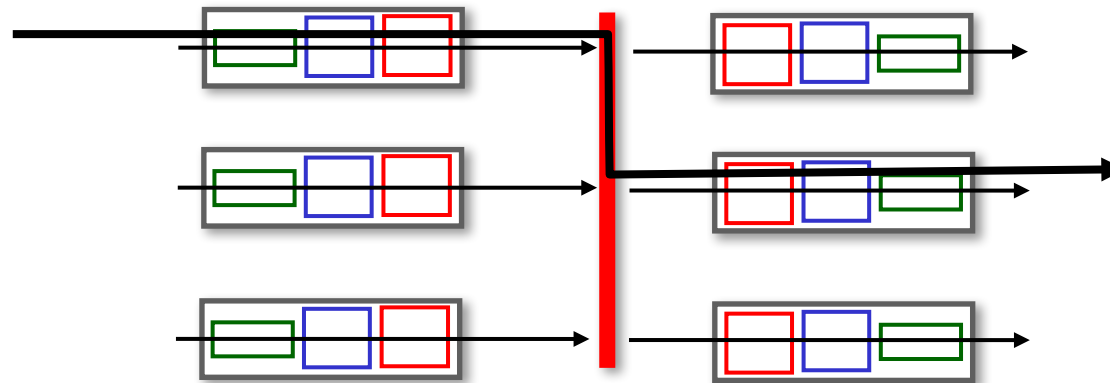
roteadores de primeira geração:

- computadores tradicionais com comutação sob controle direto da CPU
- pacote copiado para a memória do sistema
- velocidade limitada pela largura de banda de memória (2 travessias de barramento por datagrama)



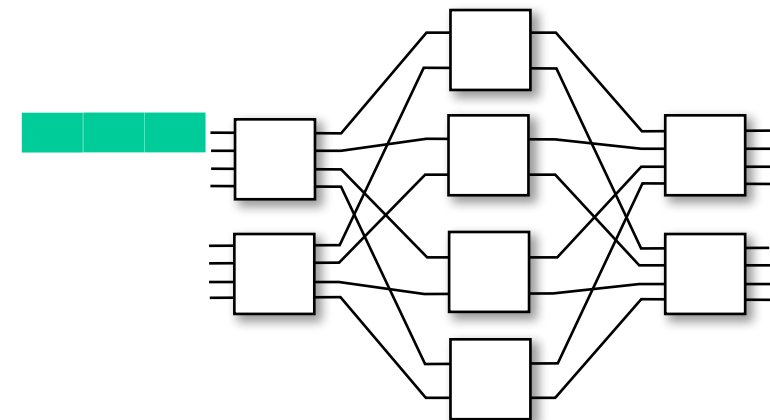
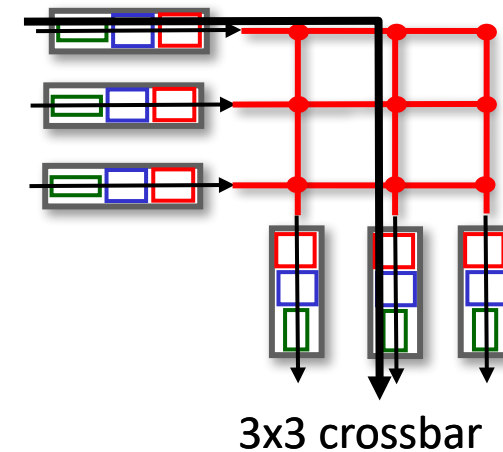
Comutação via barramento

- datagrama da memória da porta de entrada para a memória da porta de saída através de um barramento compartilhado
- *contenção de barramento*: velocidade de comutação limitada pela largura de banda do barramento
- barramento de 32 Gbps, Cisco 5600: velocidade suficiente para roteadores de acesso



Comutação via rede de interconexão

- redes Crossbar, Clos e outras redes de interconexão foram inicialmente desenvolvidas para conectar processadores em multiprocessador
- **comutador de vários estágios:** comutador $n \times n$ composto por múltiplos estágios de comutadores menores
- **explorando o paralelismo:**
 - fragmenta datagrama em células de comprimento fixo na entrada
 - comuta células através da estrutura, remonta datagrama na saída

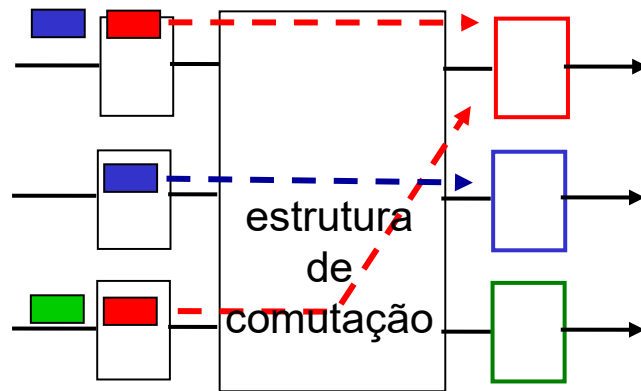


comutador multiestágio 8x8
construído a partir de comutadores
de tamanho menor Network Layer: 4-30

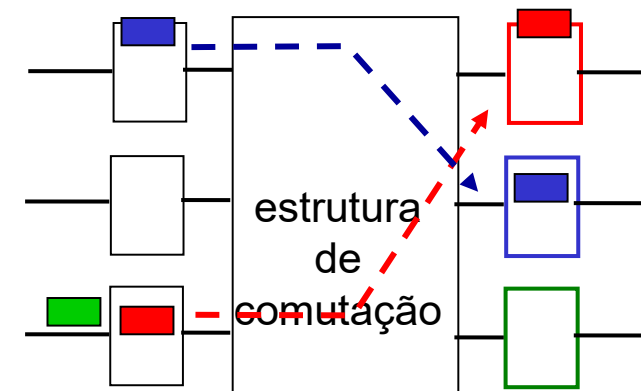
Fila da porta de entrada



- Se a estrutura de comutação for mais lenta do que as portas de entrada combinadas -> enfileiramento pode ocorrer nas filas de entrada
 - atraso na fila e perda devido ao estouro do buffer de entrada!
- **Bloqueio de cabeça de linha (*Head-of-the-Line* - HOL):** datagrama na primeira posição da fila impede que outros na fila avancem

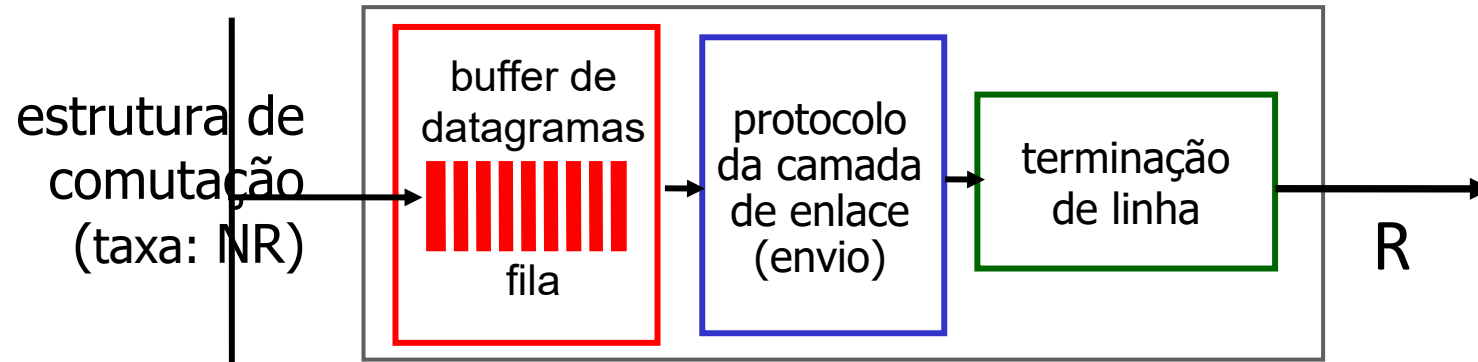


contenção da porta de saída: apenas um datagrama vermelho pode ser transferido. o pacote vermelho inferior está *bloqueado*



um pacote depois: pacote verde experimenta bloqueio de cabeça de linha

Fila da porta de saída



Este é um slide muito importante

- **Buffering** necessário quando os datagramas chegam da estrutura de comutação mais rápido do que a taxa de transmissão de enlace. **Política de perda:** quais datagramas descartar se não houver buffers disponíveis?
- **Disciplina de agendamento** escolhe entre datagramas enfileirados para transmissão

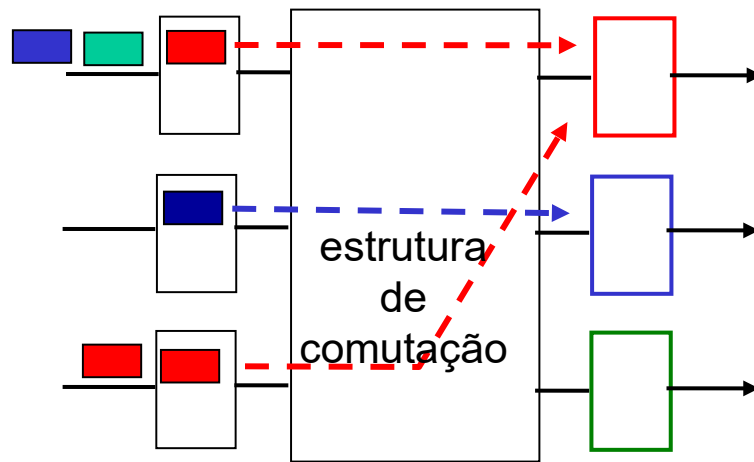


Datagramas podem ser perdidos devido ao congestionamento, falta de buffers

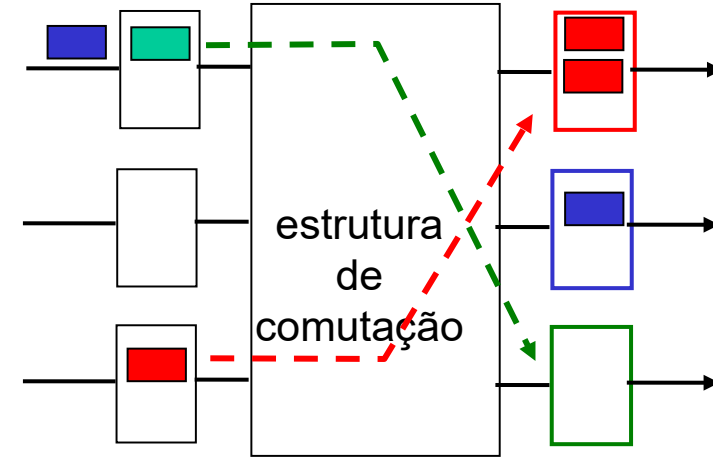


Agendamento prioritário – quem obtém melhor desempenho, neutralidade da rede

Fila da porta de saída



em t , mais que um
pacote de entrada
para a mesma saída



um tempo de pacote
depois

- buffering quando a taxa de chegada via comutador excede a velocidade da linha de saída
- *enfileiramento (atraso) e perda devido ao estouro do buffer da porta de saída!*

Quanto espaço em buffer?

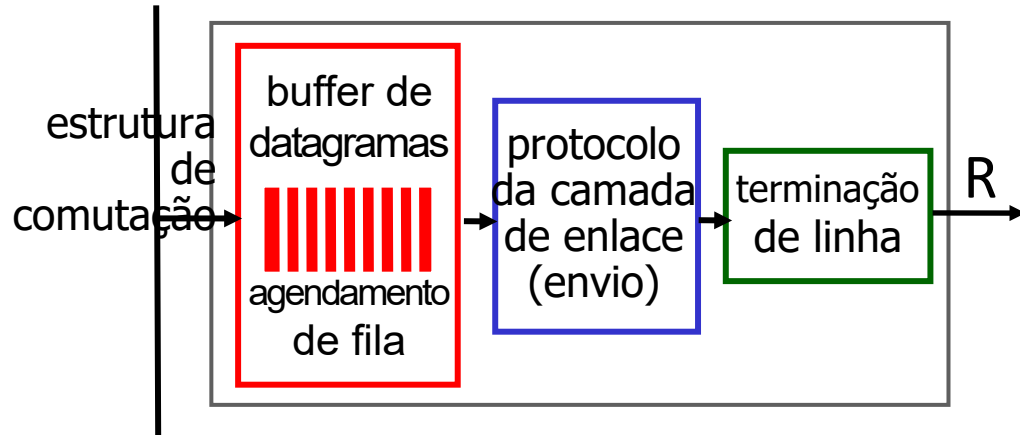
- Regra de ouro da RFC 3439: *buffering* médio igual a RTT “típico” (digamos 250 ms) vezes capacidade do enlace C
 - ex.: C = enlace de 10 Gbps: 2,5 Gbit de buffer

- recomendação mais recente: com N fluxos, *buffering* igual a

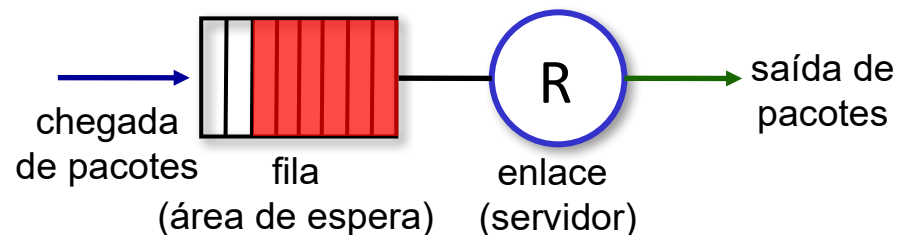
$$\frac{\text{RTT} \cdot C}{\sqrt{N}}$$

- mas buffer *demais* pode aumentar atrasos (particularmente em roteadores domésticos)
- RTTs longos: desempenho ruim para aplicações em tempo real, resposta TCP lenta
 - lembre-se do controle de congestionamento baseado em retardo: “mantenha o enlace de gargalo cheio o suficiente (ocupado), mas não mais cheio que isso”

Gerenciamento de buffer



Abstração: fila



gerenciamento de buffer:

- **perda:** quais pacotes adicionar/descartar quando buffers estão cheios
 - **tail drop:** descarta pacote que está chegando
 - **priority:** descarta/remove com base em prioridades
- **marcação:** quais pacotes marcar para sinalizar congestionamento (ECN, RED)

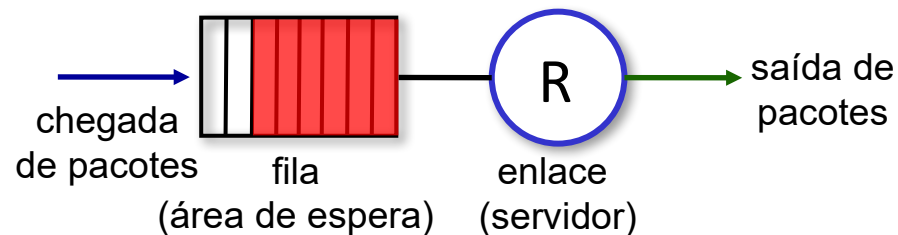
Agendamento de pacotes: FCFS

agendamento de pacotes:

decidir qual pacote enviar a seguir no enlace

- first come, first served (primeiro a chegar, primeiro a ser servido)
- prioridade
- round robin (rodízio)
- weighted fair queueing (fila justa ponderada)

Abstração: fila



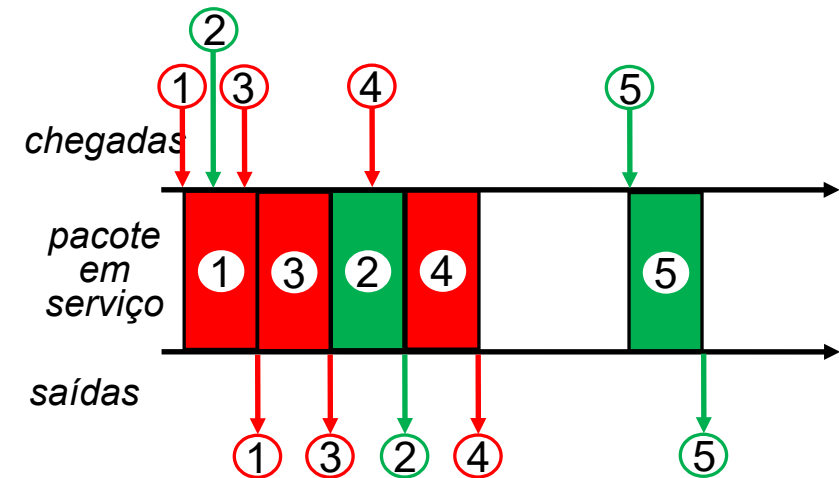
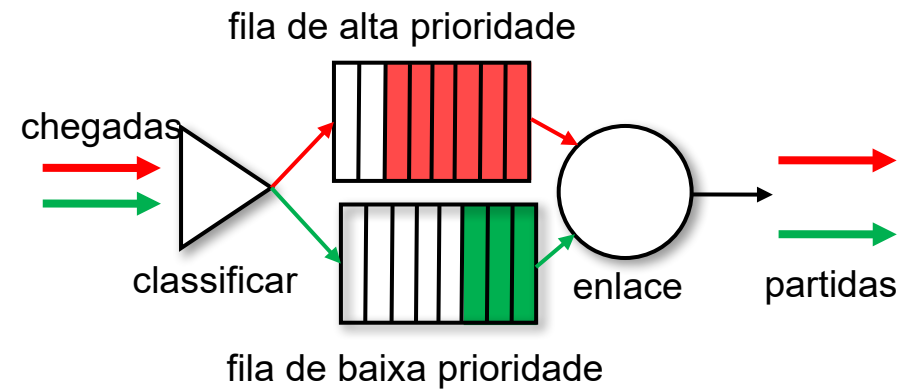
FCFS: pacotes transmitidos em ordem de chegada à porta de saída

- também conhecido como: First-in-first-out (FIFO)
- exemplos do mundo real?

Políticas de agendamento: prioridade

Agendamento prioritário:

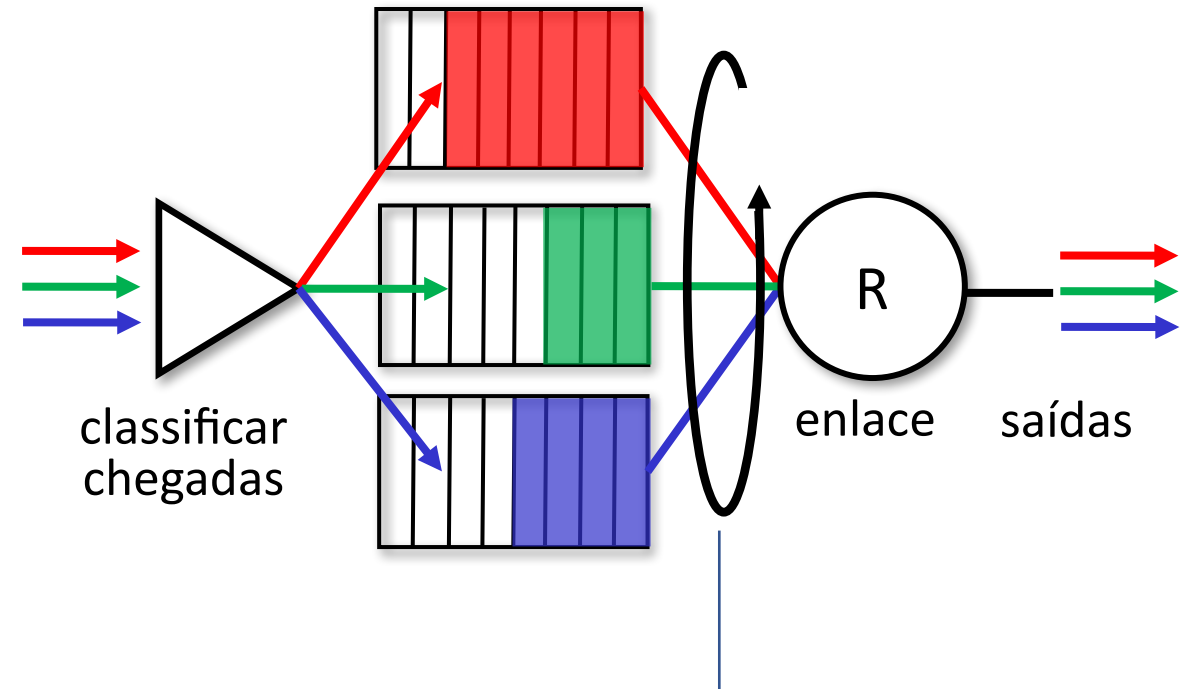
- tráfego chegando é classificado e enfileirado por classe
 - quaisquer campos de cabeçalho podem ser usados para classificação
- envia pacote da fila de prioridade mais alta que tem pacotes em buffer
 - FCFS é usado dentro da classe prioritária



Políticas de agendamento: round robin

agendamento Round Robin (RR - rodízio):

- tráfego chegando é classificado e enfileirado por classe
 - quaisquer campos de cabeçalho podem ser usados para classificação
- o servidor verifica ciclicamente e repetidamente as filas de classes, enviando um pacote completo de cada classe (se disponível) por vez



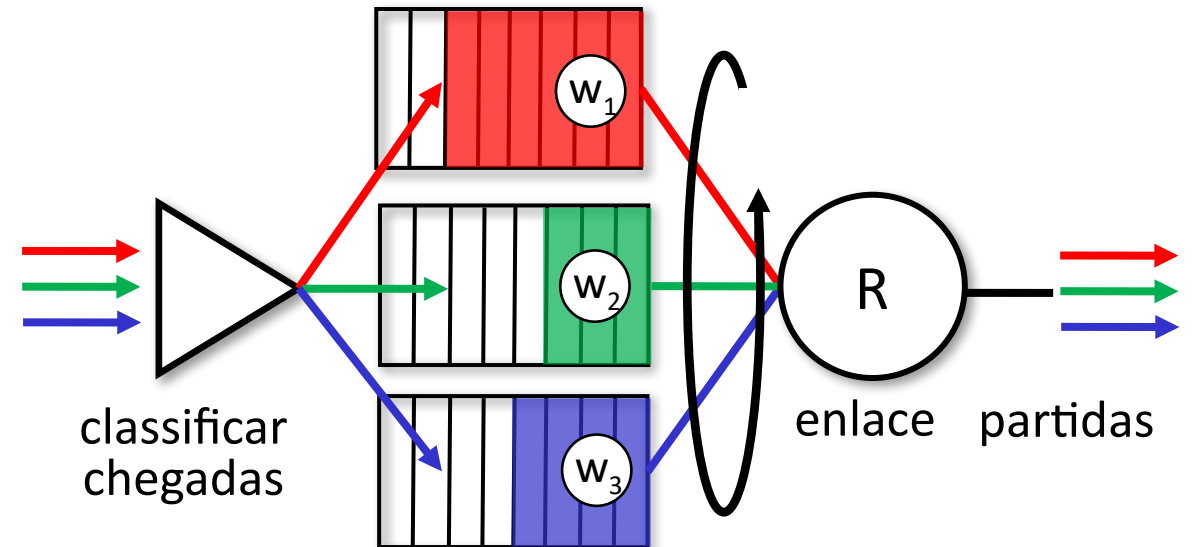
Políticas de agendamento: weighted fair queueing

Weighted Fair Queuing (WFQ – fila justa ponderada):

- Round Robin generalizado
- cada classe, i , tem peso, w_i , e recebe uma quantidade ponderada de serviço em cada ciclo:

$$\frac{w_i}{\sum_j w_j}$$

- garantia mínima de largura de banda (por classe de tráfego)



Camada de rede: roteiro do “plano de dados”

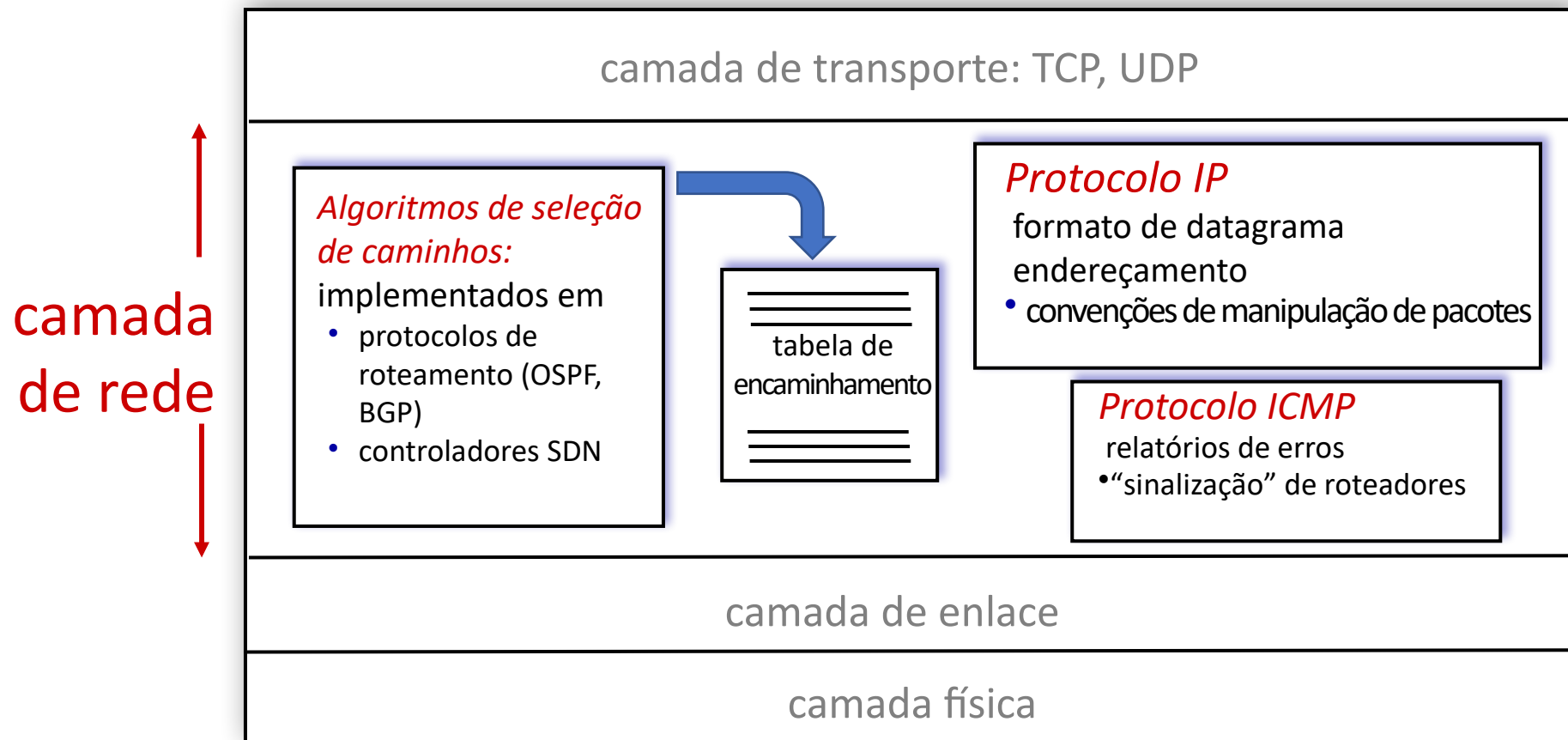
- Camada de rede: visão geral
- plano de dados
 - plano de controle
- O que há dentro de um roteador
 - portas de entrada, comutação, portas de saída
 - gerenciamento de buffer, agendamento
- IP: o Protocolo da Internet
 - formato de datagrama
 - endereçamento
 - tradução de endereço de rede (NAT)
 - IPv6



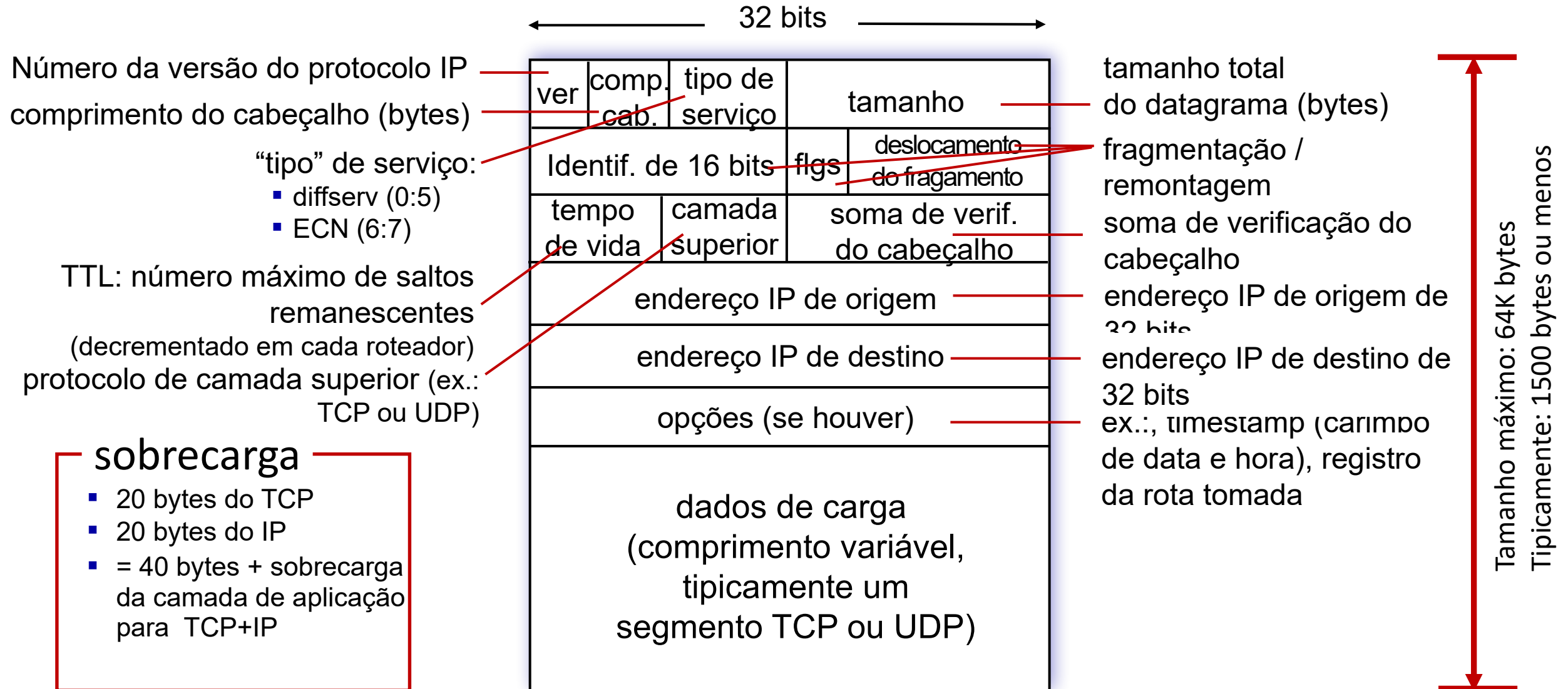
- Encaminhamento Generalizado e SDN
 - Correspondência+ação
 - OpenFlow: correspondência+ação em ação
- Middleboxes

Camada de Rede: Internet

funções de camada de rede de hospedeiros e roteadores:

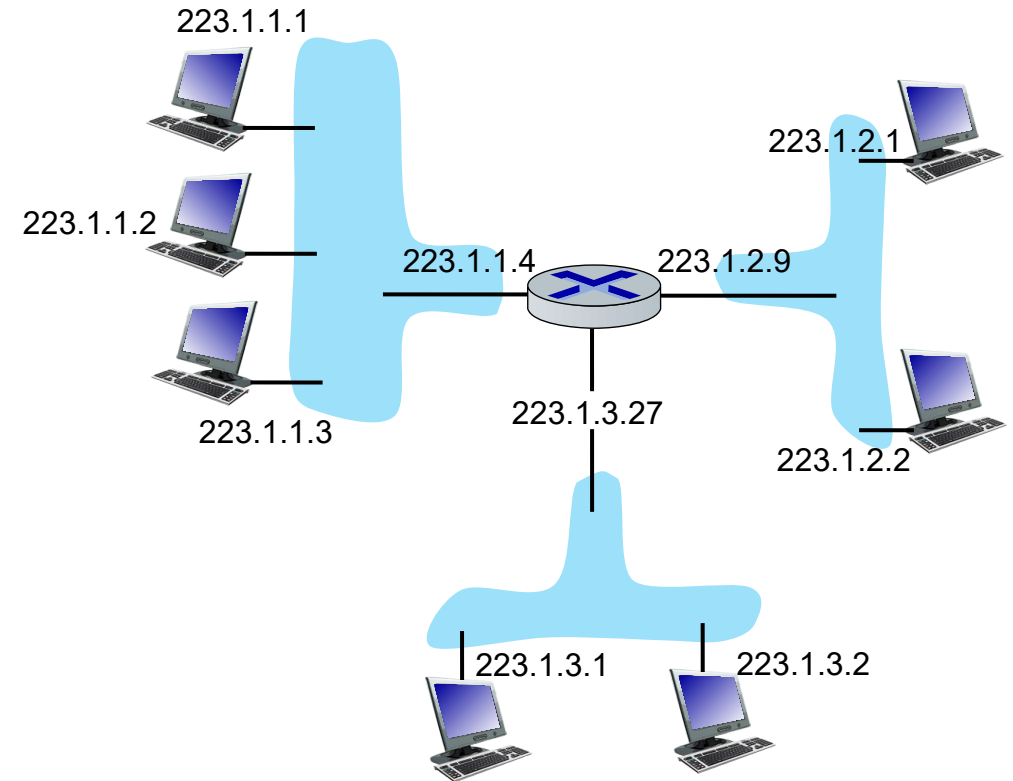


Formato do Datagrama IP



Endereçamento IP: introdução

- **endereço IP:** identificador de 32 bits associado a cada *interface* de hospedeiro ou roteador
- **interface:** conexão entre hospedeiro / roteador e enlace físico
 - roteador normalmente tem várias interfaces
 - hospedeiro tipicamente tem uma ou duas interfaces (ex.: Ethernet cabeada, 802.11 sem fio)



notação decimal com pontos de endereço IP:

223.1.1.1 = 11011111 00000001 00000001 00000001

223

1

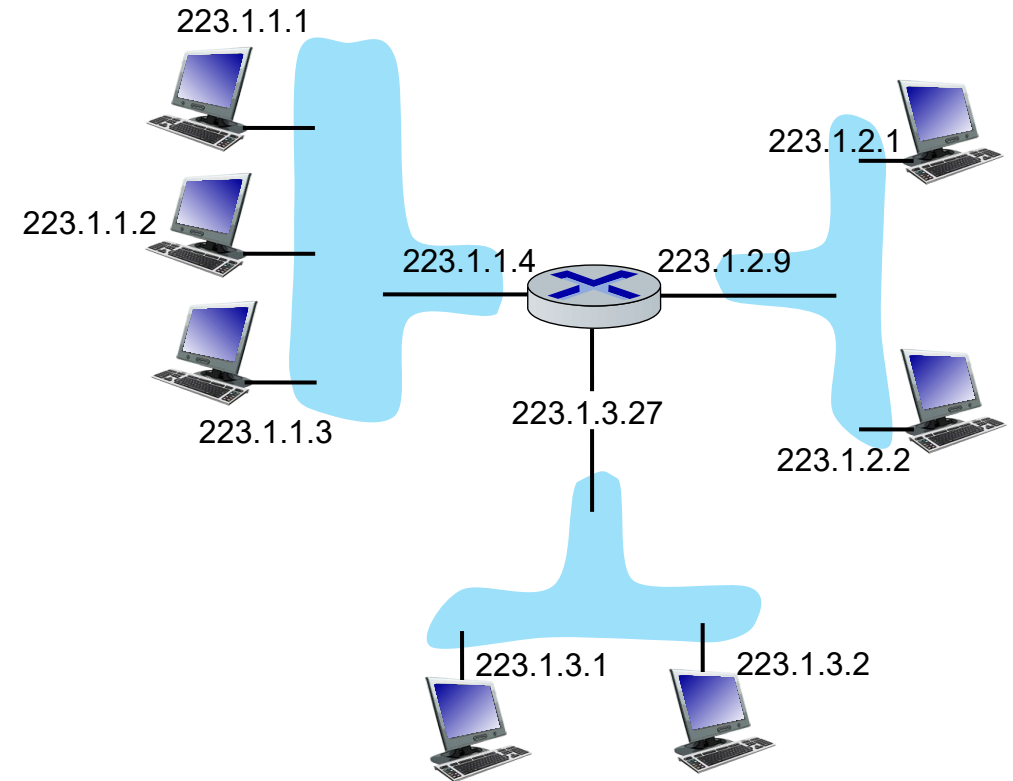
1

1

Network Layer: 4-51

Sub-redes

- *O que é uma sub-rede?*
 - interfaces de dispositivo que podem se alcançar fisicamente **sem passar por um roteador intermediário**
- Endereços IP têm a estrutura:
 - **parte da sub-rede:** dispositivos na mesma sub-rede têm bits comuns de alta ordem
 - **parte do hospedeiro:** bits de baixa ordem **remanescentes**

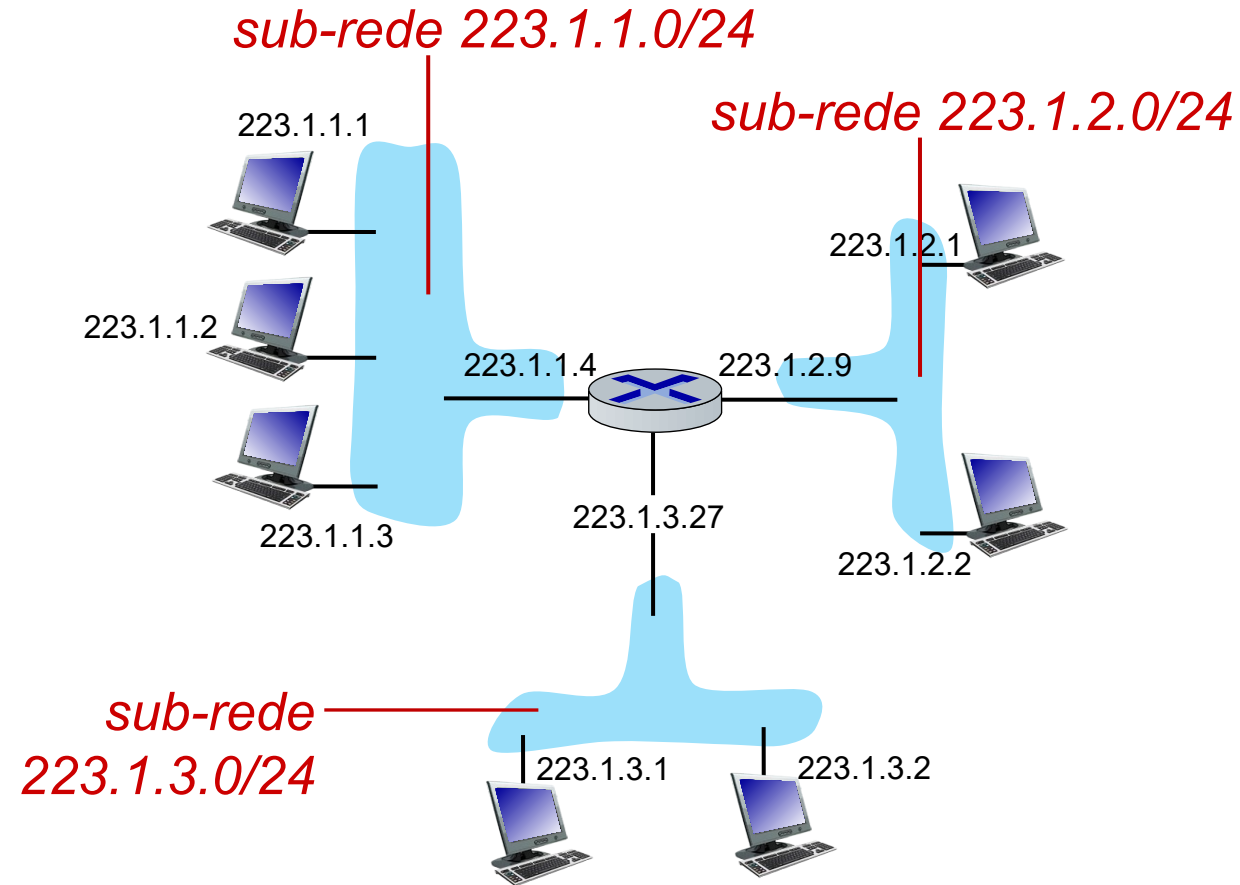


rede composta por 3 sub-redes

Sub-redes

Receita para definição de sub-redes:

- desconecte cada interface de seu hospedeiro ou roteador, criando “ilhas” de redes isoladas
- cada rede isolada é chamada de *sub-rede*

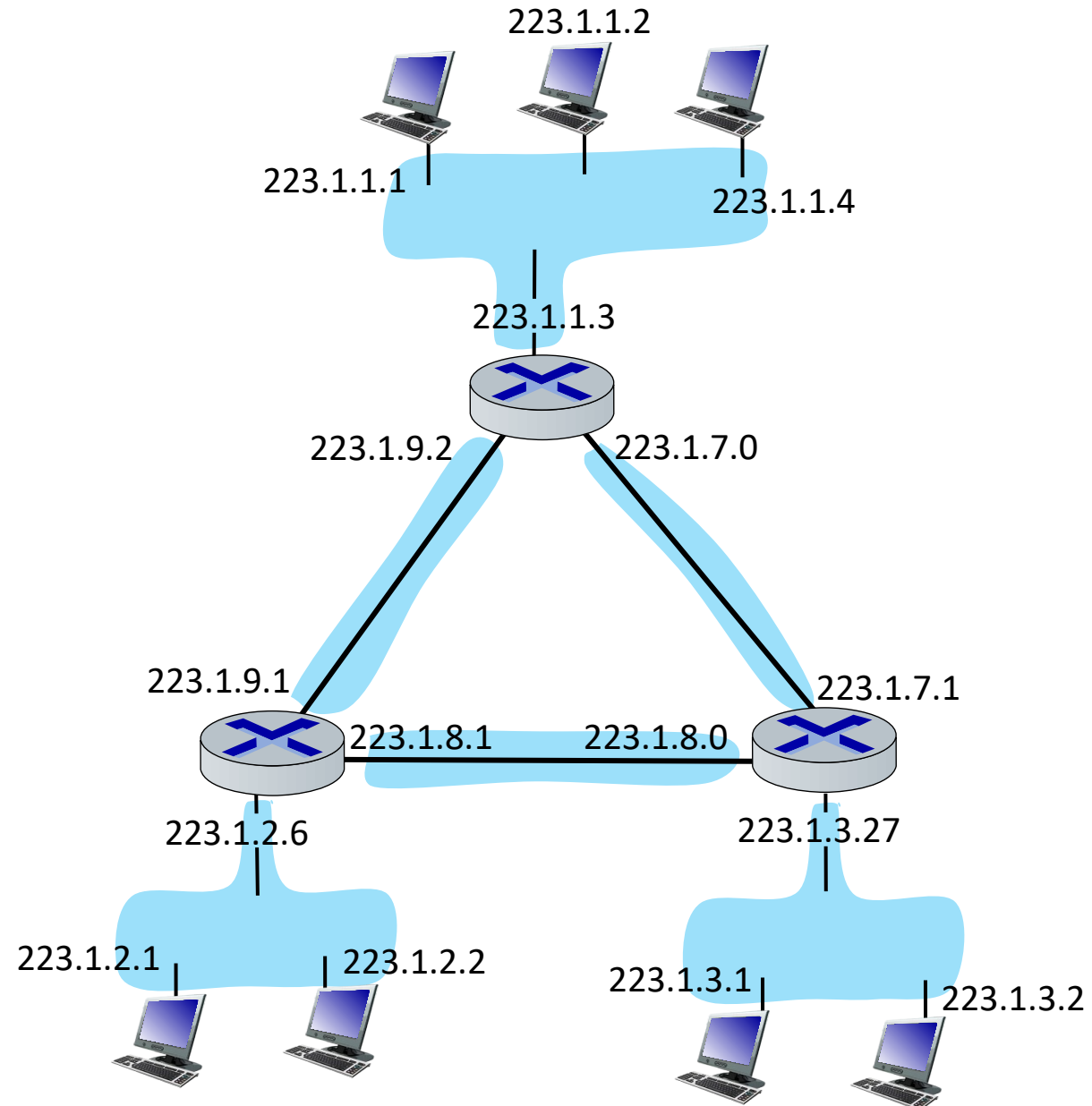


Máscara de sub-rede: /24
(24 bits de alta ordem: parte da sub-rede do endereço IP)

Sub-redes

onde estão as
sub-redes?

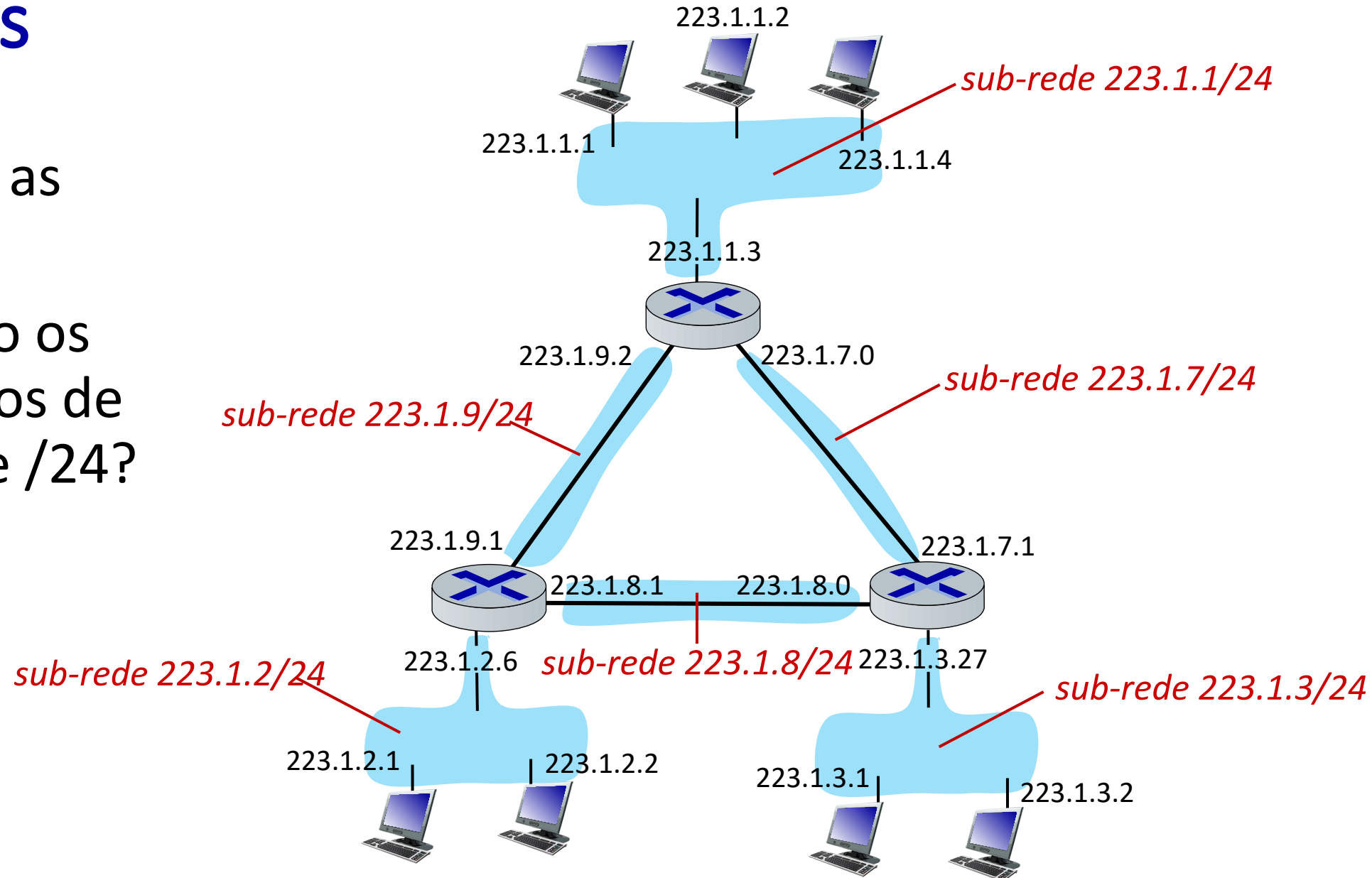
- quais são os
endereços de
sub-rede /24?



Sub-redes

onde estão as sub-redes?

- quais são os endereços de sub-rede /24?

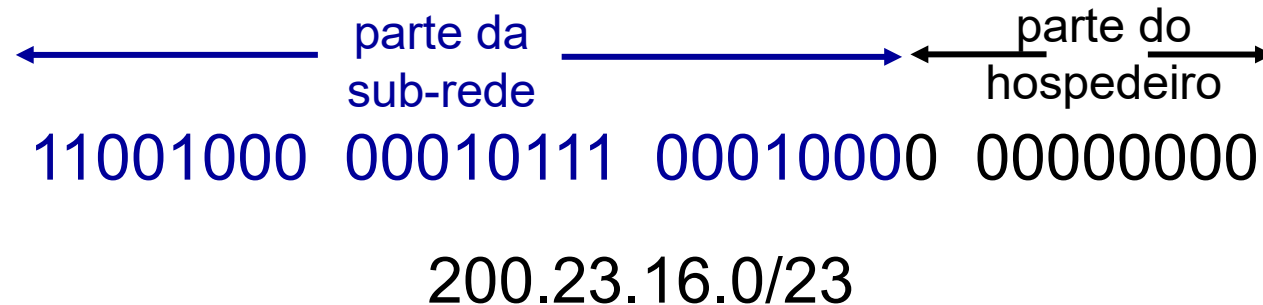


Endereçamento IP: CIDR

CIDR: Classless **I**nter**D**omain **R**outing (pronuncia-se “cider”)

parte da sub-rede de endereço de comprimento arbitrário

- formato de endereço: **a.b.c.d/x**, onde x é o número de bits na parte da sub-rede de endereço



Endereços IP: como conseguir um?

Na verdade, são **duas** questões:

1. Q: Como um *hospedeiro* obtém o endereço IP em sua rede (parte do hospedeiro do endereço)?
2. Q: Como uma *rede* obtém um endereço IP para si mesma (parte da rede do endereço)?

Como um *hospedeiro* obtém o endereço IP?

- codificado pelo administrador do sistema em um arquivo de configuração (ex.: /etc/rc.config no UNIX)
- **DHCP**: **D**ynamic **H**ost **C**onfiguration **P**rotocol: obtém endereços dinamicamente de um servidor
 - “plug-and-play”

DHCP: Dynamic Host Configuration Protocol

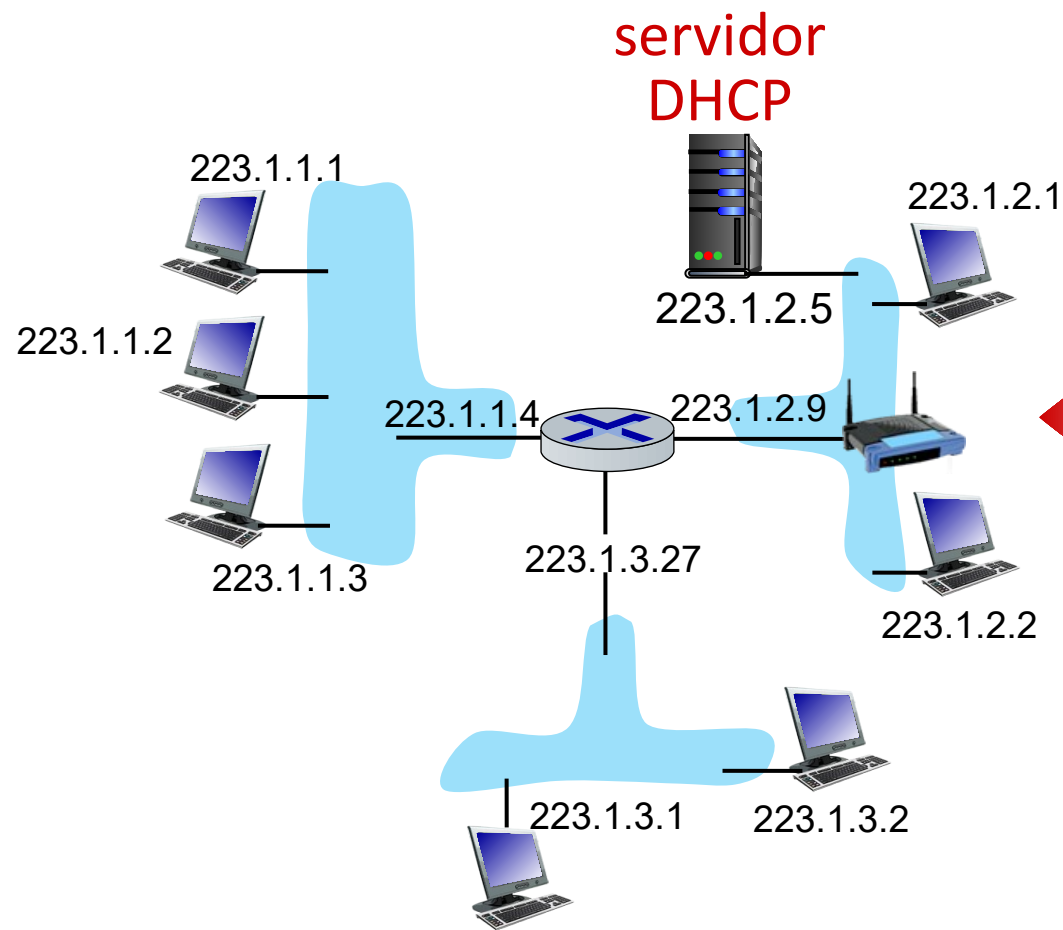
objetivo: o hospedeiro obtém *dinamicamente* o endereço IP do servidor de rede quando “entra” na rede

- pode renovar sua concessão sobre o endereço em uso
- permite a reutilização de endereços (apenas mantem o endereço enquanto conectado / ligado)
- suporte para usuários móveis que ingressam / saem da rede

visão geral do DHCP:

- hospedeiro difunde (*broadcasts*) mensagem **DHCP discover** [opcional]
- servidor DHCP responde com mensagem, **DHCP offer** [opcional]
- hospedeiro requisita endereço IP: mensagem **DHCP request**
- servidor DHCP envia endereço: mensagem **DHCP ack**

Cenário cliente-servidor DHCP



Normalmente, o servidor DHCP será co-localizado no roteador, servindo todas as sub-redes às quais o roteador está conectado



cliente DHCP chegando precisa de endereço nesta rede

Cenário cliente-servidor DHCP

Servidor DHCP: 223.1.2.5



DHCP discover

Difusão: há um servidor DHCP por aí?

Cliente chegando



DHCP offer

Difusão: Eu sou um servidor DHCP! Aqui está um endereço IP que você pode usar

DHCP request

Difusão: OK. Eu gostaria de usar este endereço IP!

DHCP ACK

Transmissão: OK. Aquele endereço IP é seu!

As duas etapas acima podem ser ignoradas "se um cliente se lembrar e quiser reutilizar um endereço de rede previamente alocado" [RFC 2131]

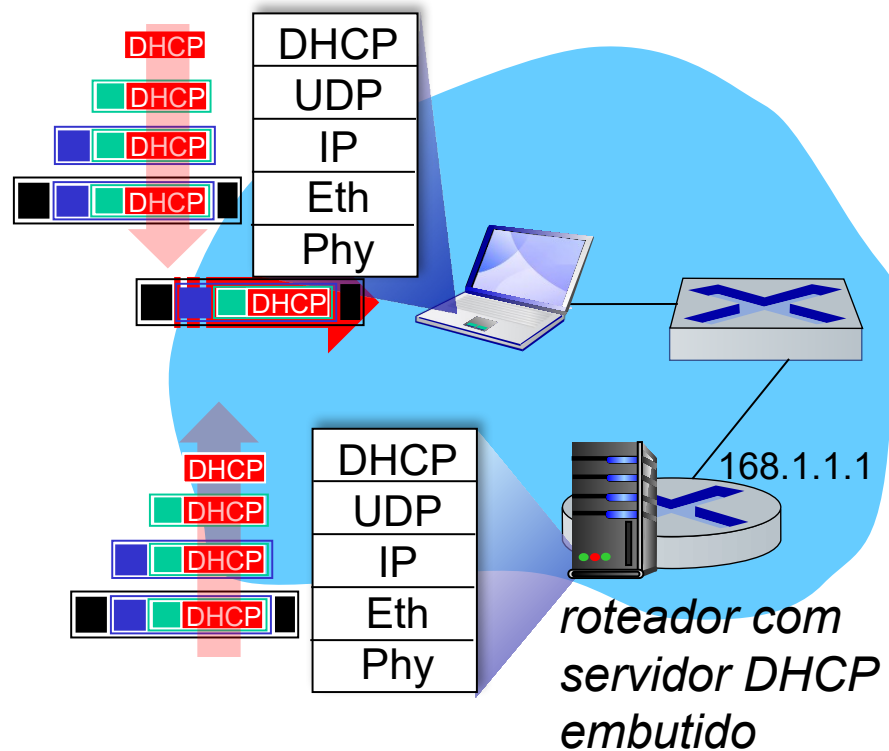
DHCP: mais que endereços IP

O DHCP pode retornar mais do que apenas o endereço IP alocado na sub-rede:

endereço do roteador de primeiro salto para o cliente

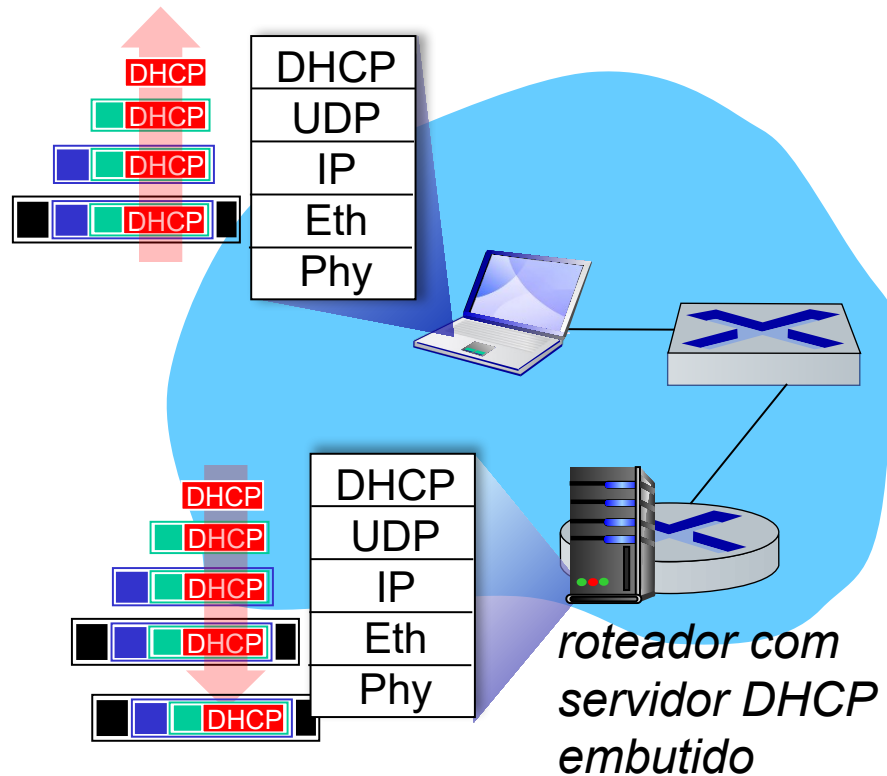
- nome e endereço IP de servidor DNS
- máscara de sub-rede (indicando porção de rede versus porção de hospedeiro do endereço)

DHCP: exemplo



- Um laptop se conectando usará o DHCP para obter o endereço IP, o endereço do roteador de primeiro salto e o endereço do servidor DNS.
- Mensagem DHCP REQUEST encapsulada em UDP, encapsulada em IP, encapsulada em Ethernet
- Difusão do quadro Ethernet (dest: FFFFFFFF) na LAN, recebido no roteador executando o servidor DHCP
- Ethernet demultiplexada para IP, demultiplexada para UDP, demultiplexada para DHCP

DHCP: exemplo



- O servidor DHCP formula o DHCP ACK contendo endereço IP do cliente, endereço IP do roteador de primeiro salto para cliente, nome e endereço IP do servidor DNS
- resposta encapsulada do servidor DHCP encaminhada ao cliente, demultiplexando até o DHCP no cliente
- o cliente agora conhece seu endereço IP, nome e endereço IP do servidor DNS, e endereço IP do seu roteador de primeiro salto

Endereços IP: como conseguir um?

Q: como a *rede* obtém a parte de sub-rede de endereço IP?

R: obtém parte alocada do espaço de endereços de seu provedor

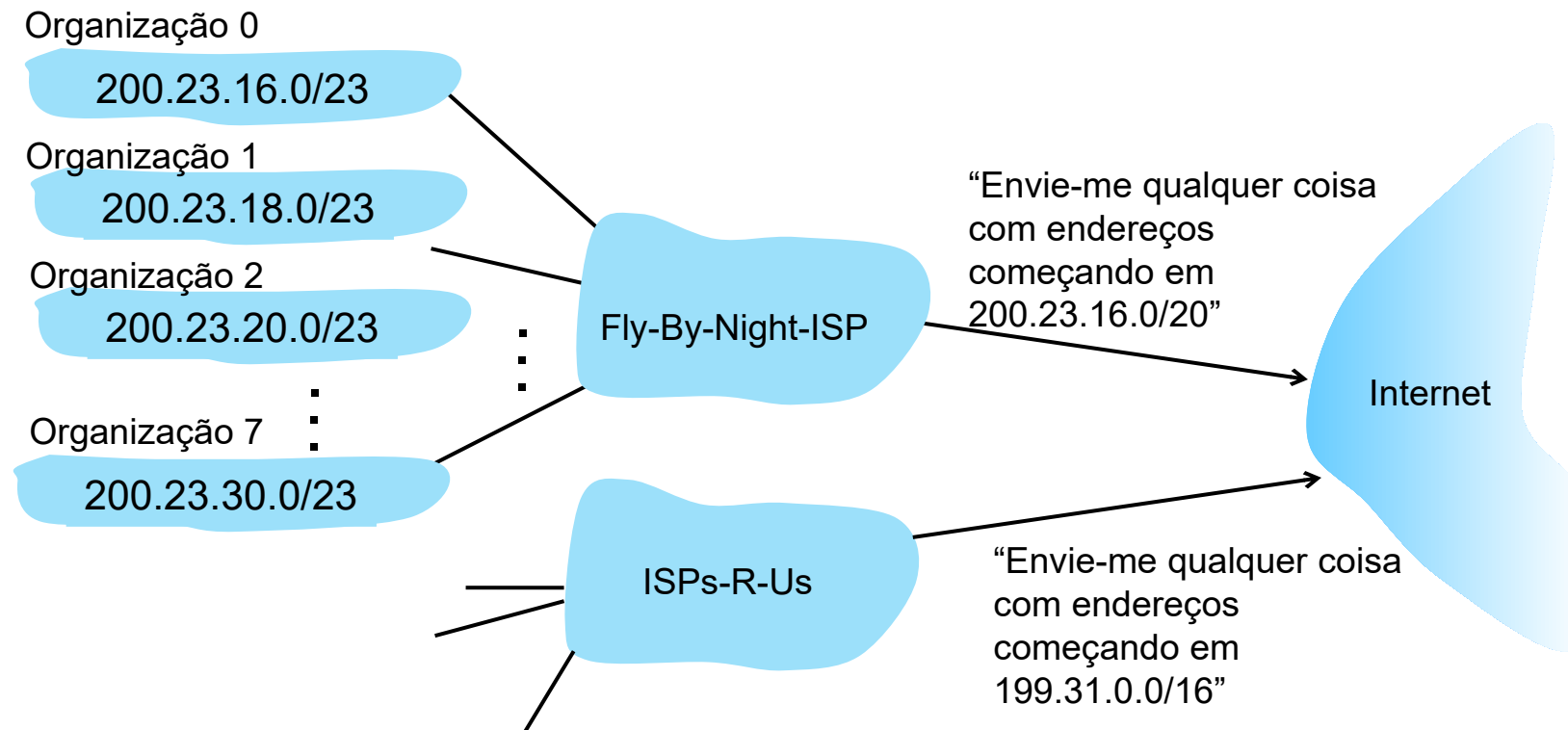
Bloco do ISP 11001000 00010111 00010000 00000000 200.23.16.0/20

O ISP pode então alocar seu espaço de endereço em 8 blocos:

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

Endereçamento hierárquico: agregação de rotas

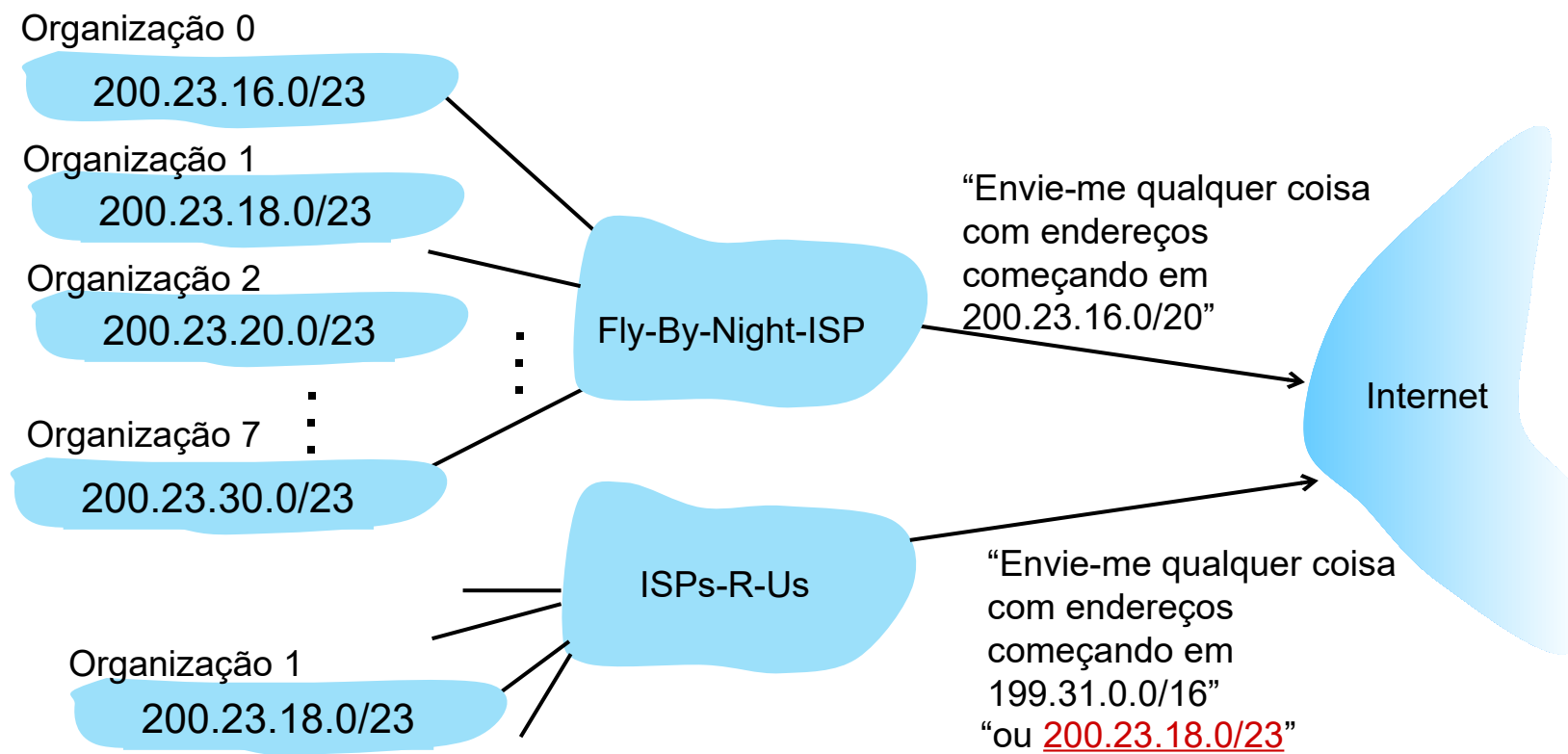
o endereçamento hierárquico permite a divulgação eficiente de informações de roteamento:



Endereçamento hierárquico: rotas mais específicas

Organização 1 passa de Fly-By-Night-ISP para ISPs-R-Us

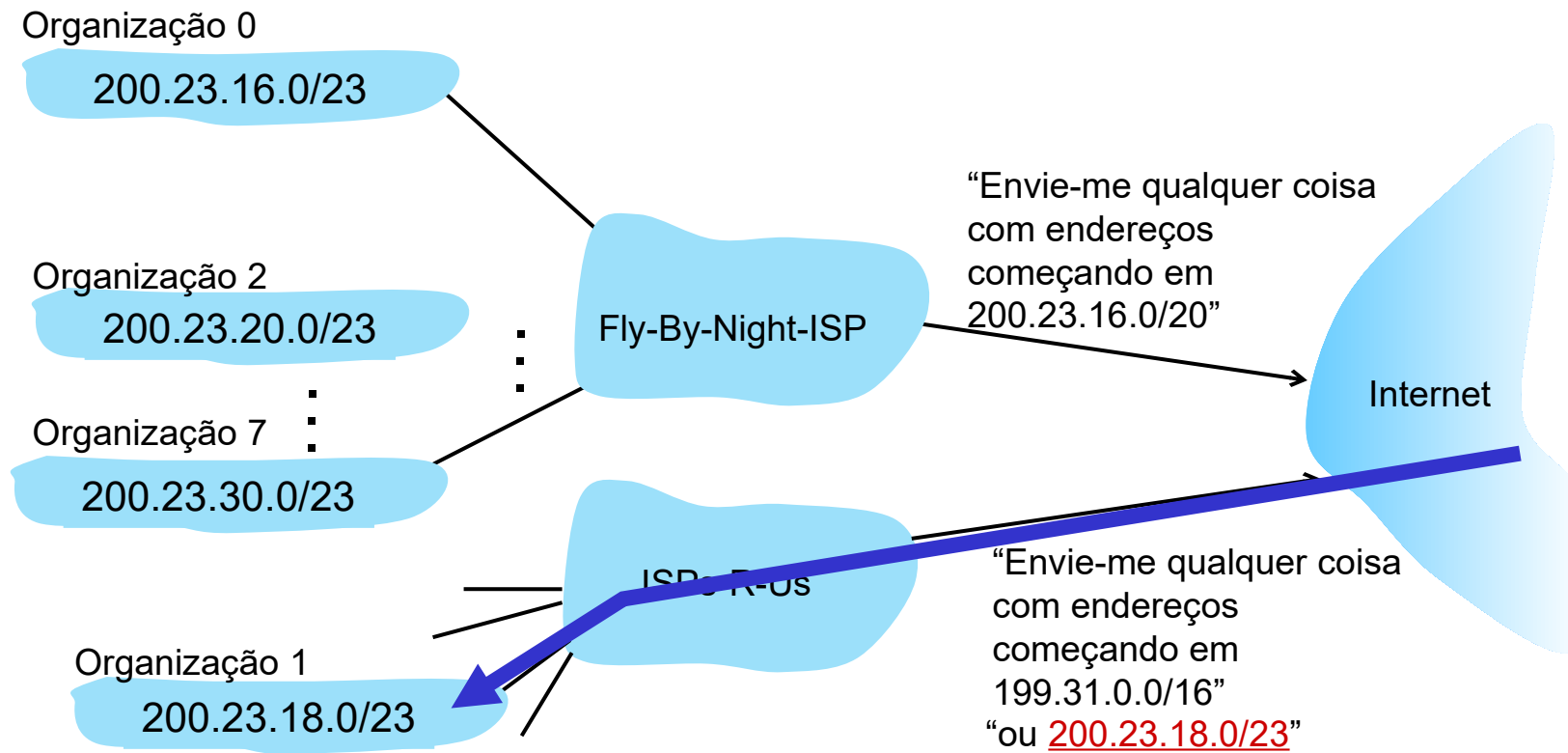
- ISPs-R-Us anuncia agora uma rota mais específica para a Organização 1



Endereçamento hierárquico: rotas mais específicas

Organização 1 passa de Fly-By-Night-ISP para ISPs-R-Us

- ISPs-R-Us anuncia agora uma rota mais específica para a Organização 1



Endereçamento IP: últimas palavras ...

Q: como um ISP recebe bloco de endereços?

R: ICANN: Internet Corporation for Assigned Names and Numbers
<http://www.icann.org/>

- aloca endereços IP, através de 5 regional registries (RRs) (que podem, então, alocar para registros locais)
- gerencia a zona raiz do DNS, incluindo gerenciamento de delegação de TLD individual (.com, .edu , ...)

Q: existem endereços IP suficientes de 32 bits?

- ICANN alocou último pedaço de endereços IPv4 para RRs em 2011
- NAT (a seguir) ajuda o IPv4 a lidar com o esgotamento do espaço
- IPv6 tem espaço de endereço de 128 bits

“Quem diabos sabia quanto espaço de endereço precisávamos?” - Vint Cerf (refletindo sobre a decisão de fazer endereços IPv4 de 32 bits de comprimento)

Camada de rede: roteiro do “plano de dados”

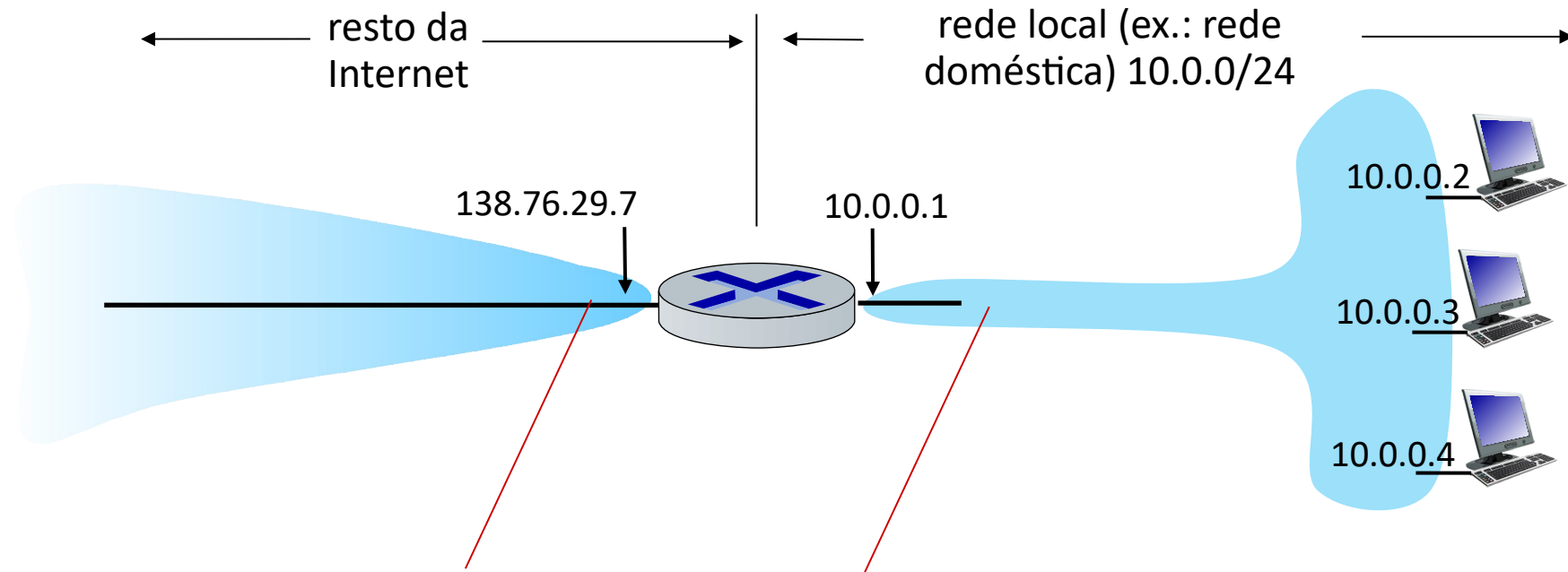
- Camada de rede: visão geral
- plano de dados
 - plano de controle
- O que há dentro de um roteador
 - portas de entrada, comutação, portas de saída
 - gerenciamento de buffer, agendamento
- **IP: o Protocolo da Internet**
 - formato de datagrama
 - endereçamento
 - tradução de endereço de rede (NAT)
 - IPv6



- Encaminhamento Generalizado e SDN
 - Correspondência+ação
 - OpenFlow: correspondência+ação em ação
- Middleboxes

NAT: network address translation

NAT: todos os dispositivos em uma rede local compartilham apenas **um** endereço IPv4 no que diz respeito ao mundo exterior



todos os datagramas *deixando* a rede local tem o *mesmo* endereço IP NAT fonte: 138.76.29.7, mas *diferentes* números de porta de origem

datagramas com origem ou destino nesta rede tem endereços 10.0.0/24 para origem e destino (como de costume)

NAT: network address translation

- todos os dispositivos na rede local têm endereços de 32 bits em um espaço de endereço IP “privado” (prefixos 10/8, 172.16/12, 192.168/16) que só podem ser usados em redes locais
- vantagens:
 - apenas **um** endereço IP do provedor é necessário para *todos* os dispositivos
 - é possível mudar endereços de hospedeiro em redes locais sem notificar o mundo exterior
 - pode alterar o provedor sem alterar endereços de dispositivos na rede local
 - segurança: dispositivos dentro da rede local não são diretamente endereçáveis ou visíveis pelo mundo exterior

NAT: network address translation

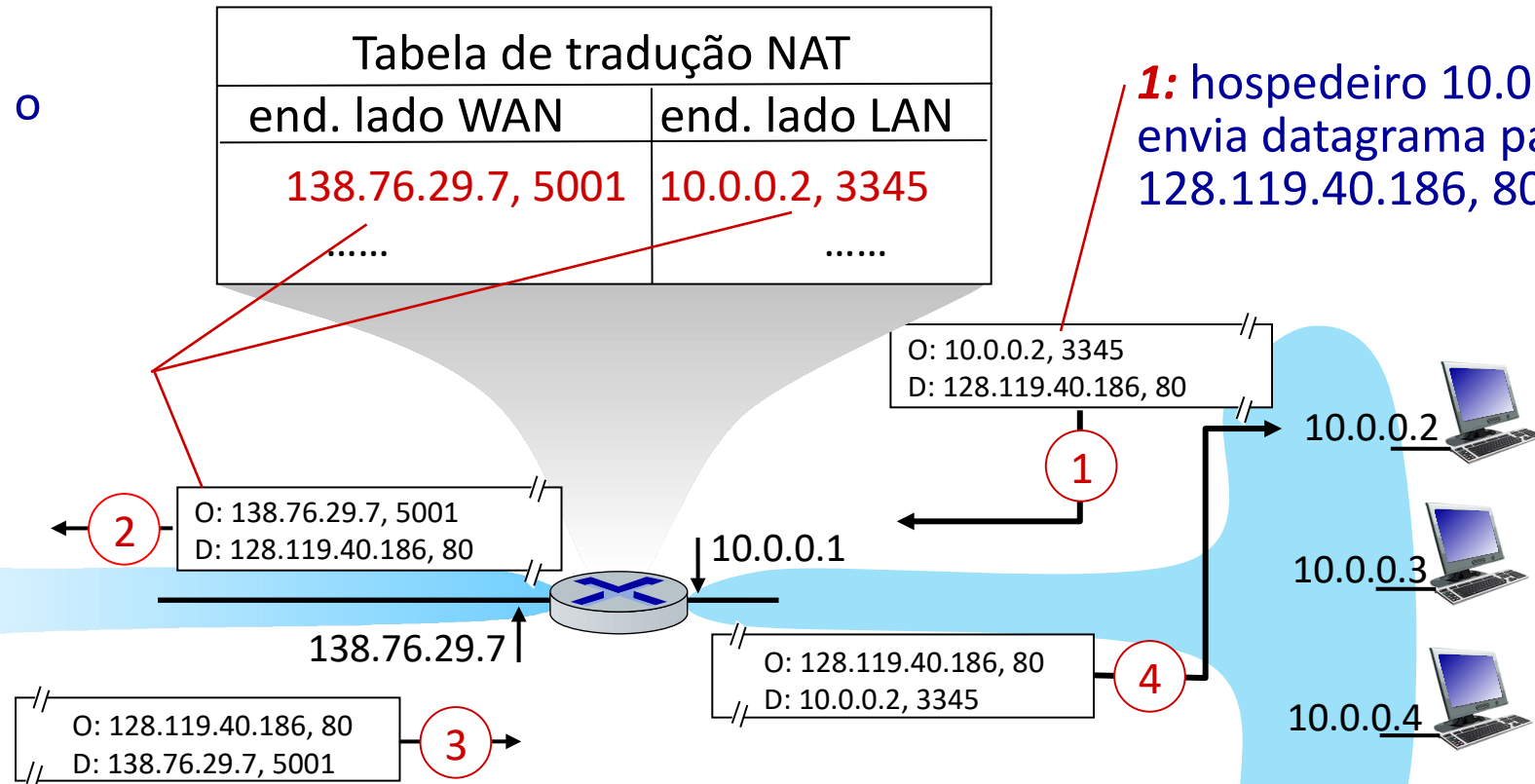
implementação: roteador NAT precisa (transparentemente):

- **datagramas de saída: substituir** (endereço IP de origem, número de porta) de cada datagrama de saída para (endereço IP NAT, novo número de porta)
 - clientes/servidores remotos responderão usando (endereço IP NAT, novo número de porta) como endereço de destino
- **lembrar (na tabela de tradução NAT)** todo par de tradução (endereço IP de origem, número de porta) para (endereço IP NAT, novo número de porta)
- **datagramas de entrada: substituir** (endereço IP NAT, novo número de porta) em campos de destino de cada datagrama de entrada com os correspondentes (endereço IP de origem, número de porta) armazenados na tabela NAT

NAT: network address translation

2: O roteador NAT altera o endereço de origem do datagrama de 10.0.0.2, 3345 para 138.76.29.7, 5001, e atualiza a tabela

1: hospedeiro 10.0.0.2 envia datagrama para 128.119.40.186, 80



3: resposta chega, endereço de destino: 138.76.29.7, 5001

NAT: network address translation

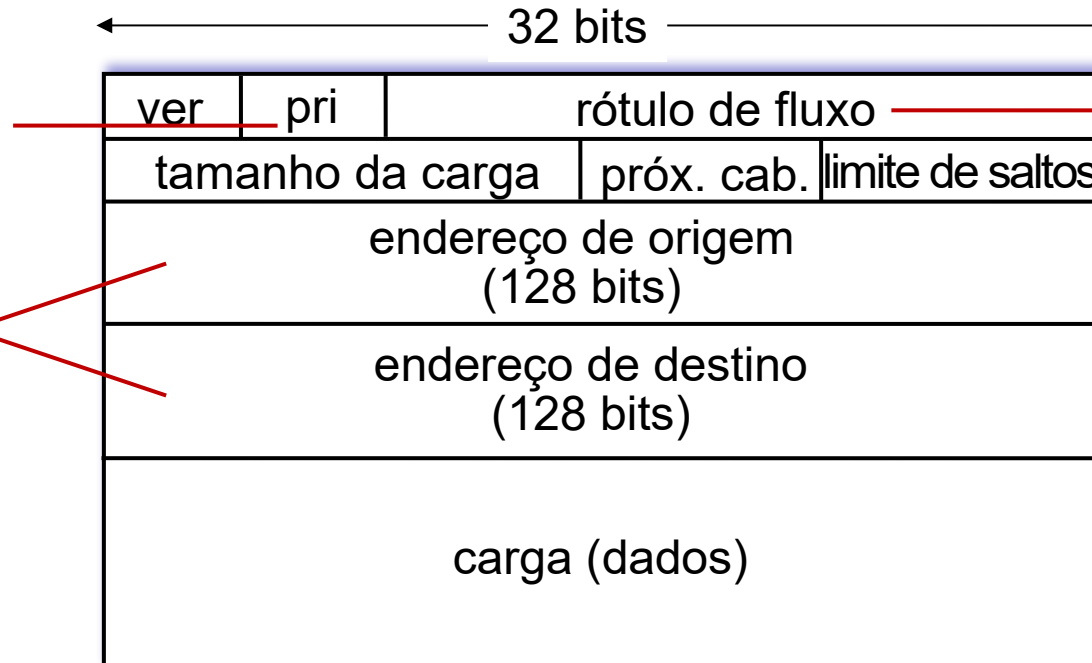
- NAT tem sido controverso:
 - roteadores “deveriam” processar apenas até a camada 3
 - “escassez” de endereços deve ser resolvida pelo IPv6
 - viola argumento fim-a-fim (manipulação de número de porta por dispositivo da camada de rede)
 - Travessia de NAT: e se o cliente quiser se conectar a um servidor atrás de um NAT?
- mas o NAT está aqui para ficar:
 - amplamente utilizado em redes domésticas e institucionais e redes celulares 4G/5G

IPv6: motivação

- **motivação inicial:** espaço de endereço IPv4 de 32 bits seria completamente alocado
- motivação adicional:
- processamento/encaminhamento de alta velocidade: cabeçalho de comprimento fixo de 40 bytes
 - permitir diferentes tratamentos de “fluxos” na camada de rede

Formato de datagrama IPv6

prioridade: identifica
prioridade entre
datagramas no fluxo
endereços IPv6 de
128 bits



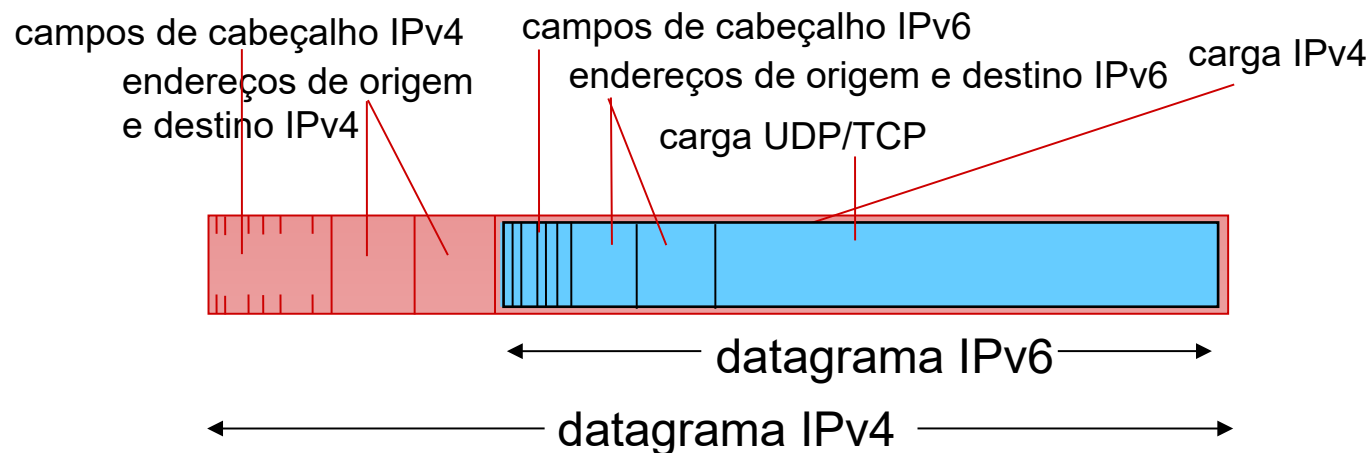
rótulo de fluxo:
identifica datagramas no
mesmo “fluxo”.
(conceito de “fluxo” não
é bem definido).

O que está faltando (comparado com IPv4):

- sem soma de verificação (para acelerar o processamento em roteadores)
- sem fragmentação/remontagem
- sem opções (disponíveis como protocolo de próximo cabeçalho de camada superior no roteador)

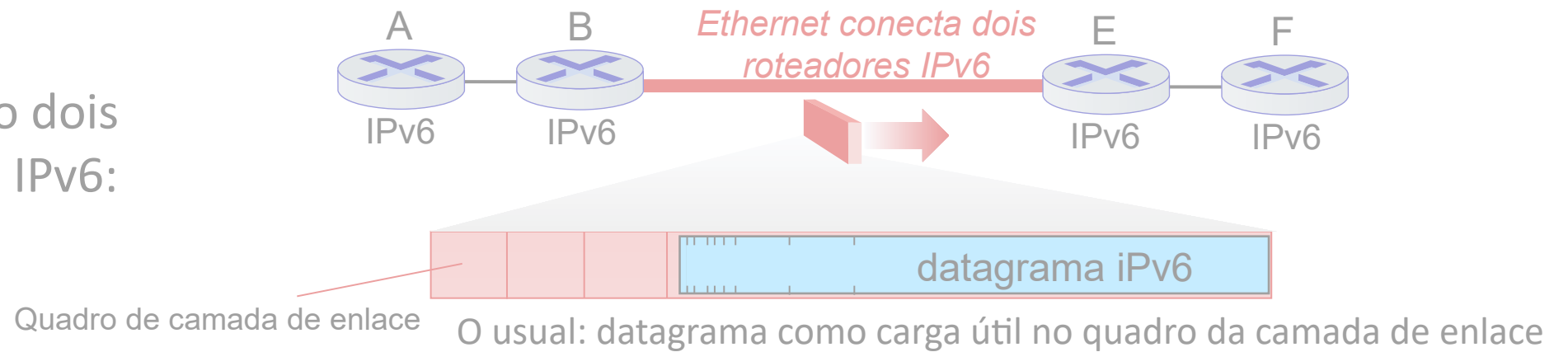
Transição de IPv4 para IPv6

- nem todos os roteadores podem ser atualizados simultaneamente
 - sem data marcada para a mudança
 - como a rede funcionará com roteadores IPv4 e IPv6 misturados?
- **tunelamento**: datagrama IPv6 carregado como *carga* em datagrama IPv4
 - tunelamento é usado extensivamente em outros contextos (4G / 5G)

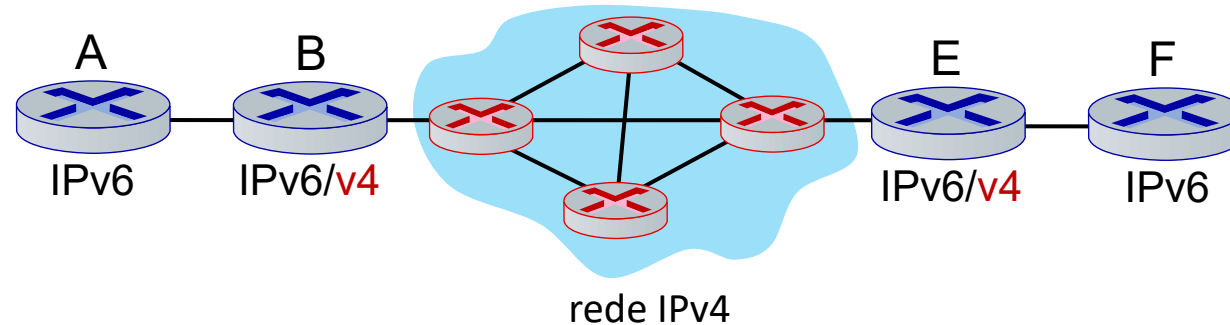


Tunelamento e encapsulamento

Ethernet
conectando dois
roteadores IPv6:

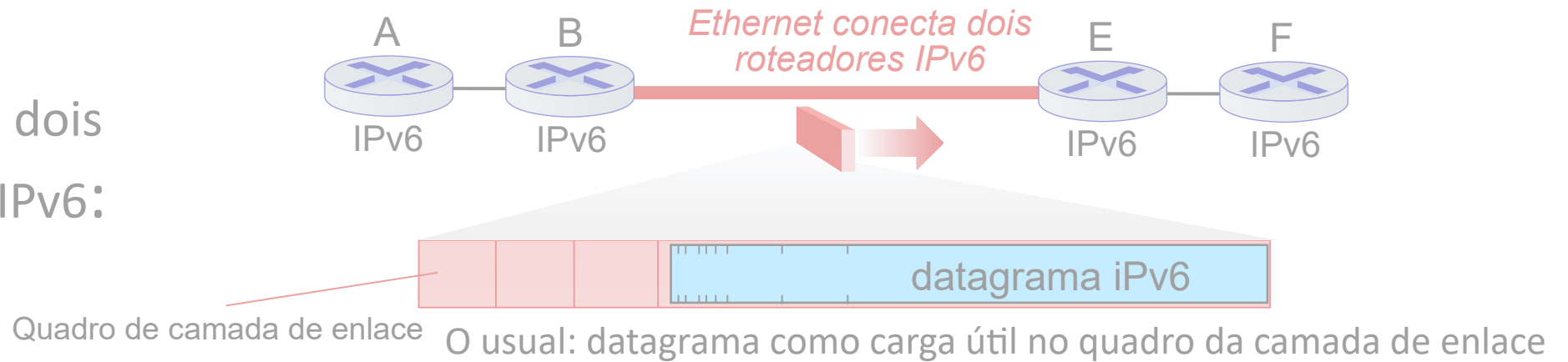


Rede IPv4
conectando dois
roteadores IPv6

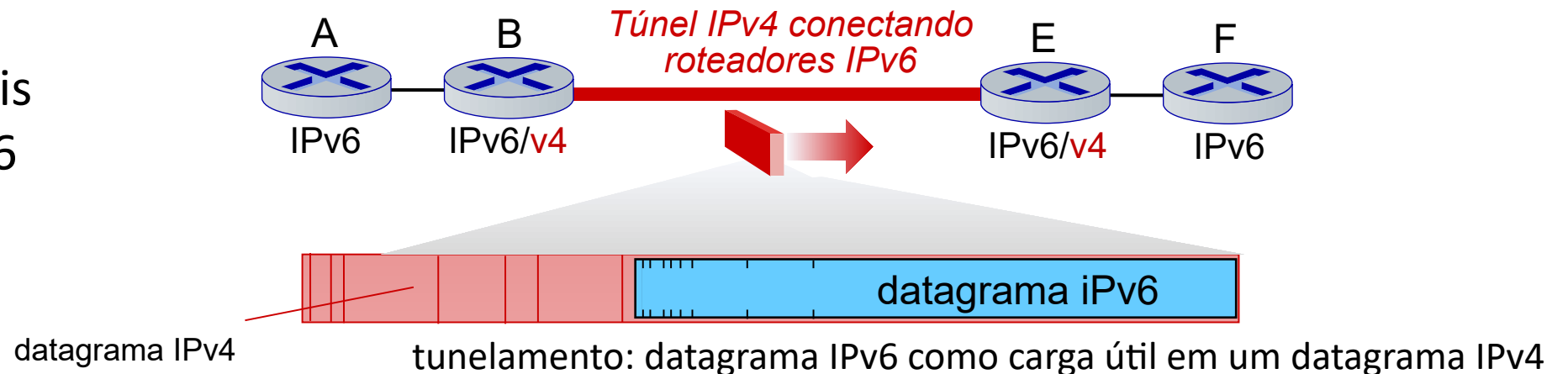


Tunelamento e encapsulamento

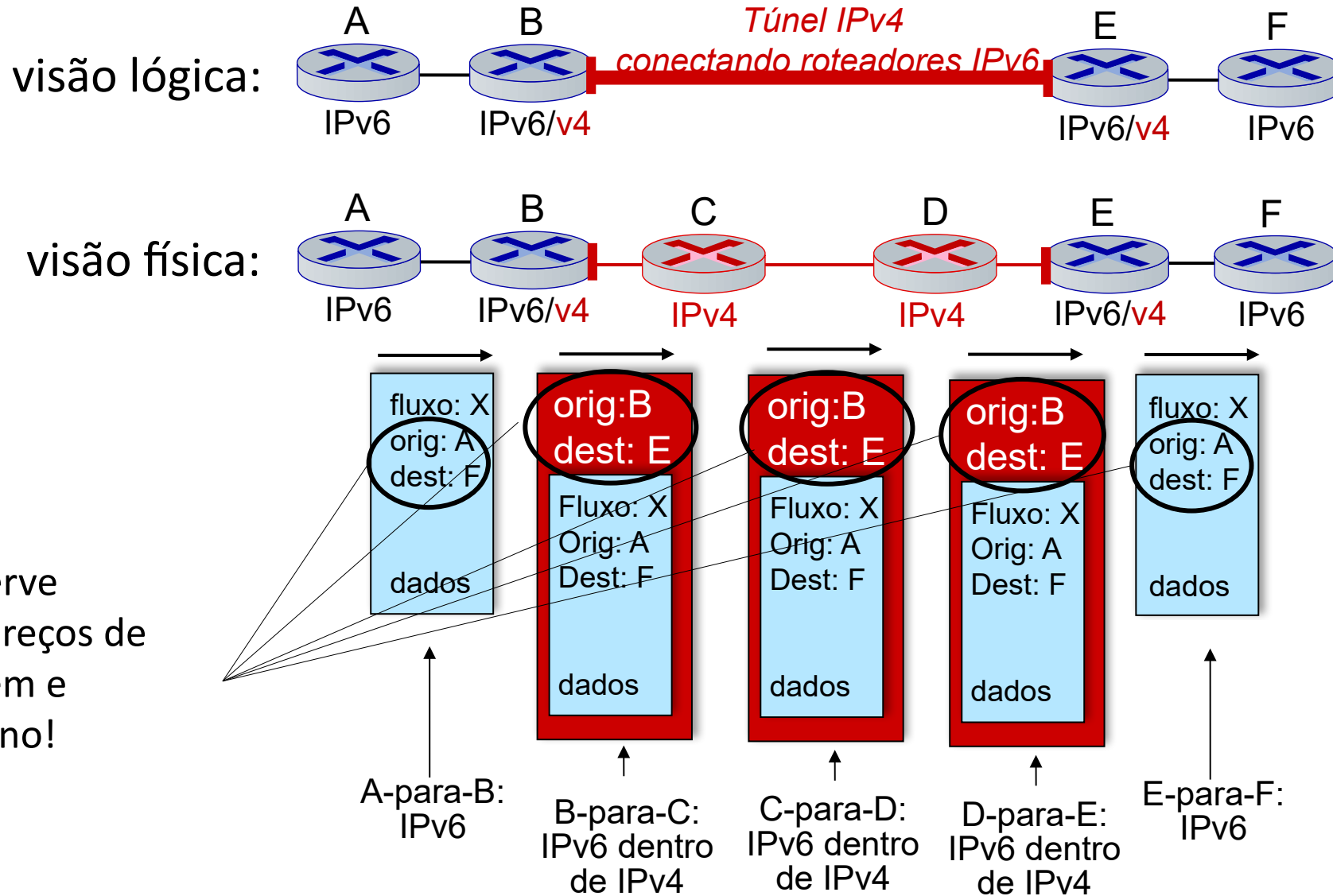
Ethernet
conectando dois
roteadores IPv6:



Túnel IPv4
conectando dois
roteadores IPv6



Tunelamento

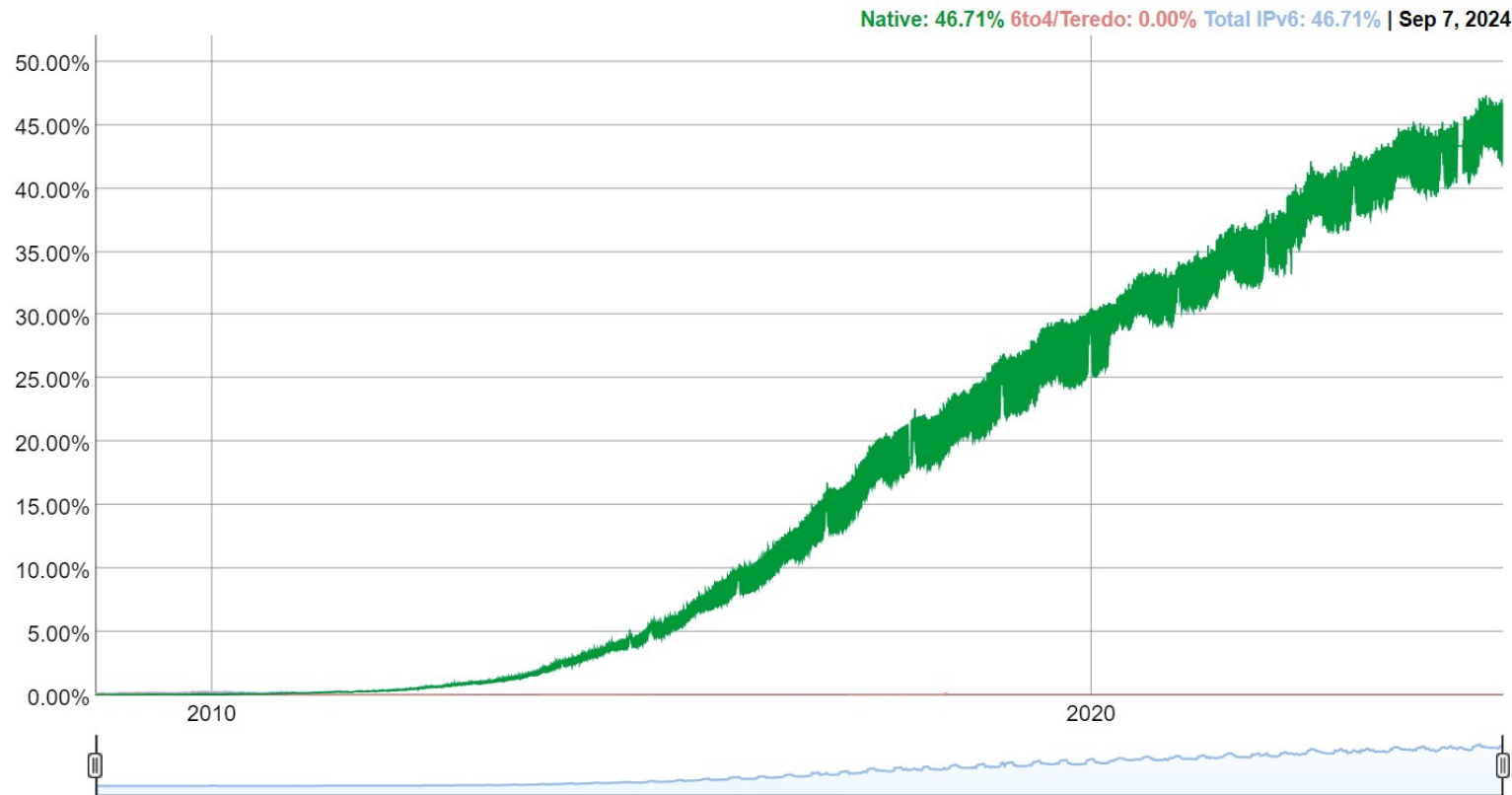


IPv6: adoção

- Google¹: ~ 47% de clientes acessam serviços via IPv6 (2024)
- NIST: 1/3 de todos os domínios do governo dos EUA são compatíveis com IPv6

IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.



1

<https://www.google.com/intl/en/ipv6/statistics.html>

IPv6: adoção

- Google¹: ~ 47% de clientes acessam serviços via IPv6 (2024)
- NIST: 1/3 de todos os domínios do governo dos EUA são compatíveis com IPv6
- Longo (longo!) tempo para implantação e uso
 - 29 anos e contando!
 - pense nas mudanças no nível de aplicação nos últimos 29 anos: WWW, mídias sociais, transmissão de mídia, jogos, telepresença, ...
 - *Por que?*

¹ <https://www.google.com/intl/en/ipv6/statistics.html>

Camada de rede: roteiro do “plano de dados”

- Camada de rede: visão geral
- plano de dados
 - plano de controle
- O que há dentro de um roteador
 - portas de entrada, comutação, portas de saída
 - gerenciamento de buffer, agendamento
- IP: o Protocolo da Internet
 - formato de datagrama
 - endereçamento
 - tradução de endereço de rede (NAT)
 - IPv6



- Encaminhamento Generalizado e SDN
 - Correspondência+ação
 - OpenFlow: correspondência+ação em ação
- Middleboxes

Encaminhamento generalizado: correspondência mais ação (*match plus action*)

Revisão: cada roteador contém uma **tabela de encaminhamento** (também conhecida como: **tabela de fluxo**)

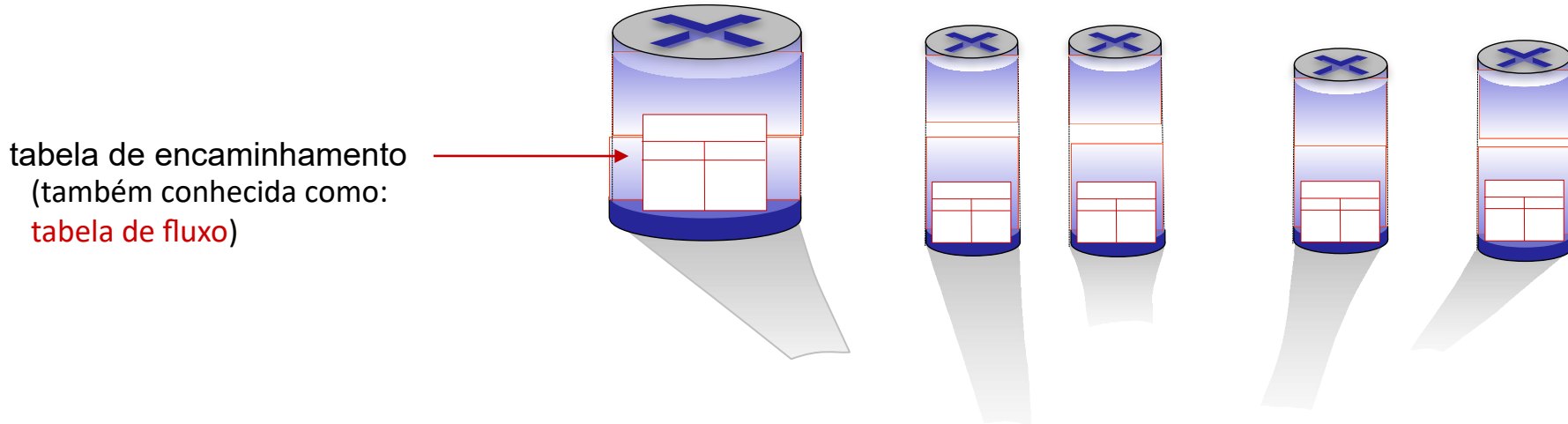
- abstração “**correspondência mais ação**”: corresponde bits no pacote chegando, toma ação

- **encaminhamento baseado em destino:** encaminha com base no endereço IP de destino

- **encaminhamento generalizado:**

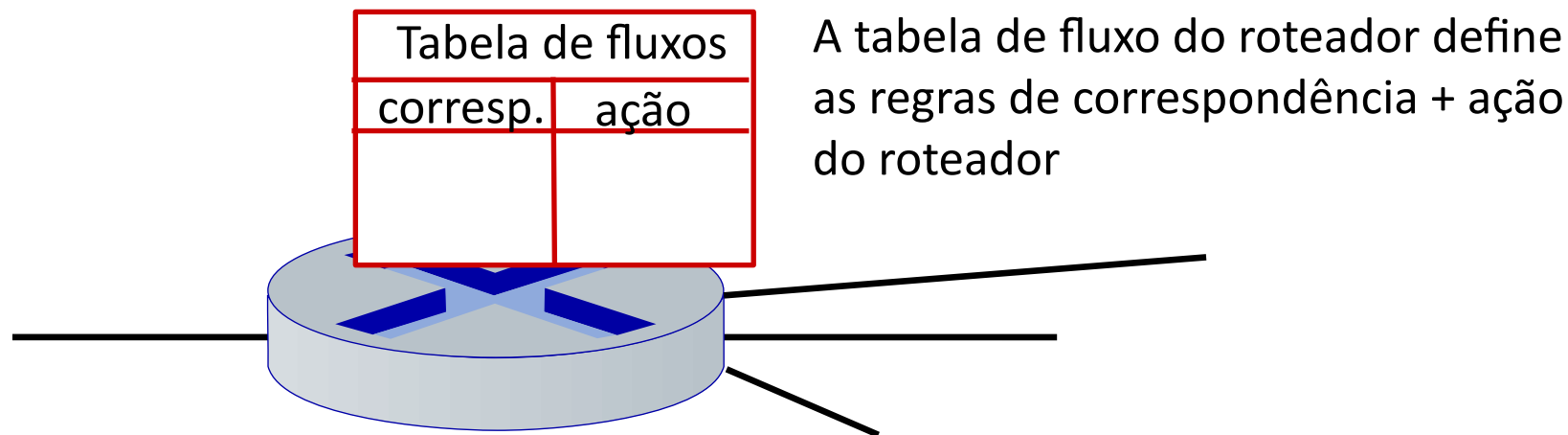
- muitos campos de cabeçalho podem determinar ação

- muitas ações possíveis: descartar/copiar/modificar/logar pacote



Abstração da tabela de fluxo

- **fluxo**: definido por valores de campos de cabeçalho (em campos das camadas de enlace, rede, e transporte)
- **encaminhamento generalizado**: regras simples de manipulação de pacotes
 - **correspondência**: padrões de valores em campos de cabeçalho do pacote
 - **ações**: para pacote correspondente: descartar, encaminhar, modificar ou enviar para o controlador
 - **prioridade**: desambiguação de padrões sobrepostos
 - **contadores**: número de bytes ou número de pacotes

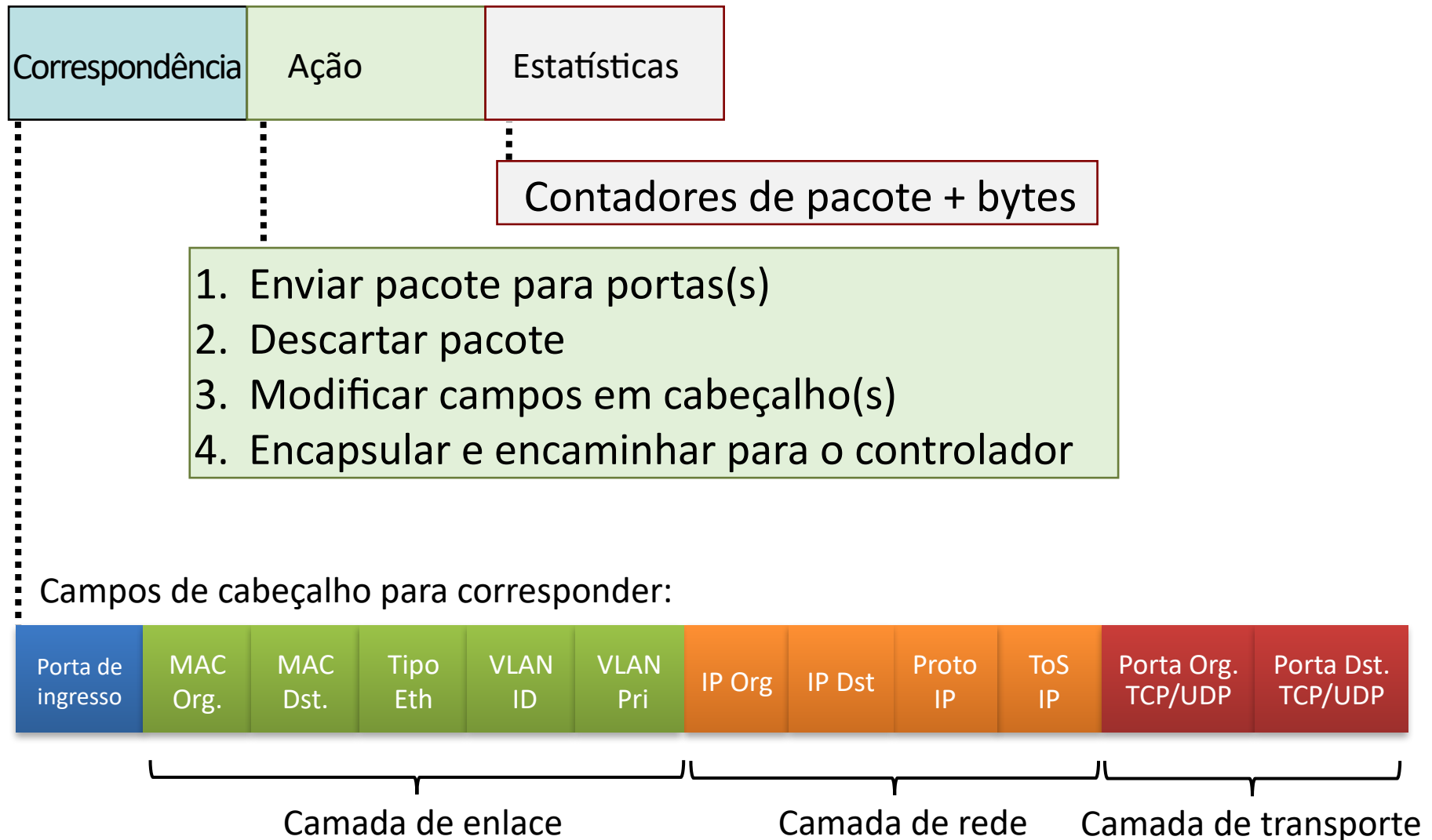


Abstração da tabela de fluxo

- **fluxo**: definido por valores de campos de cabeçalho
- **encaminhamento generalizado**: regras **simples** de manipulação de pacotes
 - **correspondência**: padrões de valores em campos de cabeçalho do pacote
 - **ações**: para pacote correspondente: descartar, encaminhar, modificar ou enviar para o controlador
 - **prioridade**: desambiguação de padrões sobrepostos
 - **contadores**: número de bytes ou número de pacotes



OpenFlow: entradas da tabela de fluxo



OpenFlow: exemplos

Encaminhamento baseado em destino:

Porta Switch	MAC org	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Org	IP Dst	IP Prot	IP ToS	TCP o-port	TCP d-port	Ação
*	*	*	*	*	*	*	51.6.0.8	*	*	*	*	porta6

Os datagramas IP destinados ao endereço IP 51.6.0.8 devem ser encaminhados para a porta de saída 6 do roteador

Firewall:

Porta Switch	MAC org	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Org	IP Dst	IP Prot	IP ToS	TCP o-port	TCP d-port	Ação
*	*	*	*	*	*	*	*	*	*	*	22	descartar

Bloqueie (não encaminhe) todos os datagramas destinados à porta TCP 22 (número de porta do SSH)

Porta Switch	MAC org	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Org	IP Dst	IP Prot	IP ToS	TCP o-port	TCP d-port	Ação
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	descartar

Bloqueie (não encaminhe) todos os datagramas enviados pelo hospedeiro 128.119.1.1

OpenFlow: exemplos

Encaminhamento baseado em destino da camada 2:

Porta Switch	MAC org	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Org	IP Dst	IP Prot	IP ToS	TCP o-port	TCP d-port	Ação
*	*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	*	porta3

quadros da camada 2 com endereço MAC de destino 22:A7:23:11:E1:02 devem ser encaminhados para a porta de saída 3

Abstração do OpenFlow

- **correspondência+ação:** abstração unifica diferentes tipos de dispositivos

Roteador

- *correspondência:* prefixo IP de destino mais longo
- *ação:* encaminhar por um enlace

Comutador

- *correspondência:* endereço MAC de destino
- *ação:* encaminhar ou inundar

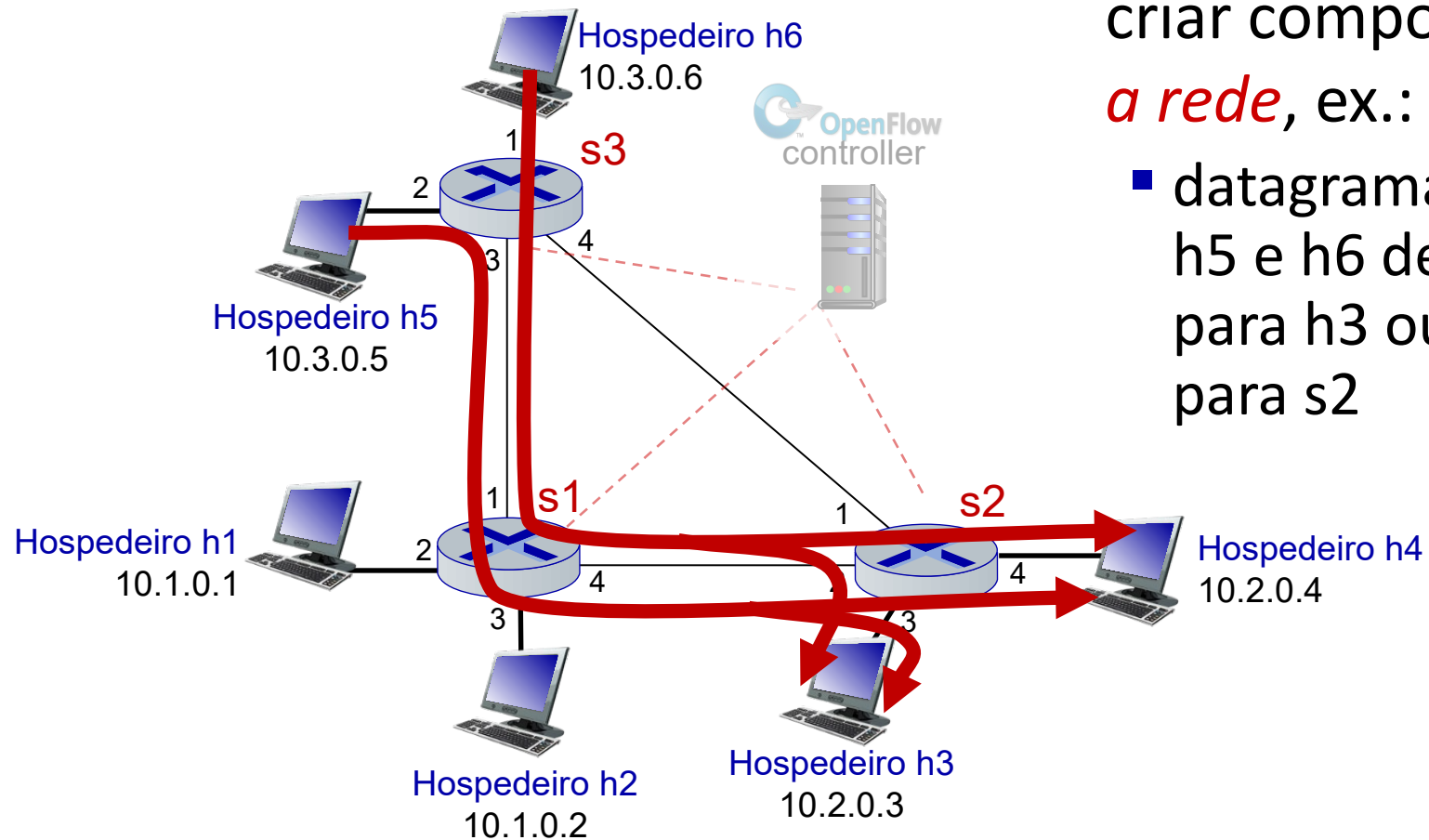
Firewall

- correspondência:* endereços IP e números de porta TCP/UDP
- *ação:* permitir ou negar

NAT

- *correspondência:* endereço IP e porta
- *ação:* reescrever endereço e porta

Exemplo de OpenFlow

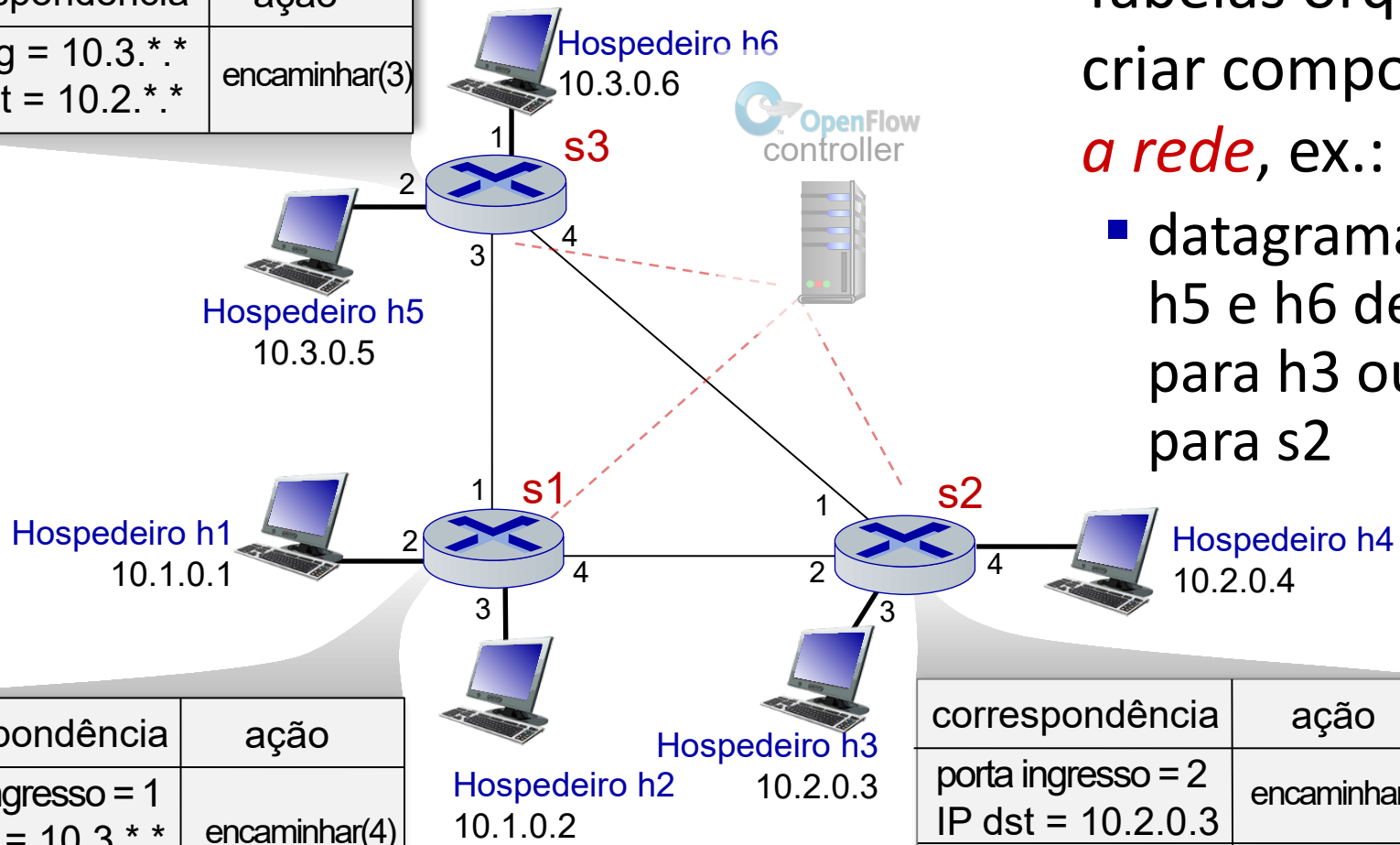


Tabelas orquestradas podem criar comportamento *em toda a rede*, ex.:

- datagramas dos hospedeiros h5 e h6 devem ser enviados para h3 ou h4, via s1 e de lá para s2

Exemplo de OpenFlow

correspondência	ação
IP org = 10.3.*.* IP dst = 10.2.*.*	encaminhar(3)



correspondência	ação
porta ingresso = 1 IP org = 10.3.*.* IP dst = 10.2.*.*	encaminhar(4)

Tabelas orquestradas podem criar comportamento *em toda a rede*, ex.:

- datagramas dos hospedeiros h5 e h6 devem ser enviados para h3 ou h4, via s1 e de lá para s2

correspondência	ação
porta ingresso = 2 IP dst = 10.2.0.3	encaminhar(3)
porta ingresso = 2 IP dst = 10.2.0.4	encaminhar(4)

Encaminhamento generalizado: sumário

- abstração de “**correspondência mais ação**”: corresponder bits nos cabeçalhos de qualquer camada dos pacotes chegando e tomar ação
 - correspondência sobre muitos campos (camadas de enlace, rede e transporte)
 - ações locais: descartar, encaminhar, modificar, ou enviar o pacote correspondente para o controlador
 - “programação” de comportamentos *na rede toda*
- forma simples de “programação da rede”
 - “processamento” por pacote programável
 - *raízes históricas*: rede ativa
 - *hoje*: programação mais generalizada: P4 (veja p4.org).