

# PTC 2550 - Aula 18

4.2 O que há dentro de um roteador?

4.3 O Protocolo da Internet (IP): Repasse e Endereçamento na Internet

(Kurose, p. 241 - 254)

(Peterson, p. 124-144)

02/06/2017

# Capítulo 4: conteúdo

## 4.1 Introdução à camada de rede

- Plano de dados
- Plano de controle

## 4.2 o que tem dentro de um roteador?

## 4.3 IP: *Internet Protocol*

- formato do datagrama
- fragmentação
- endereçamento IPv4
- NAT
- IPv6

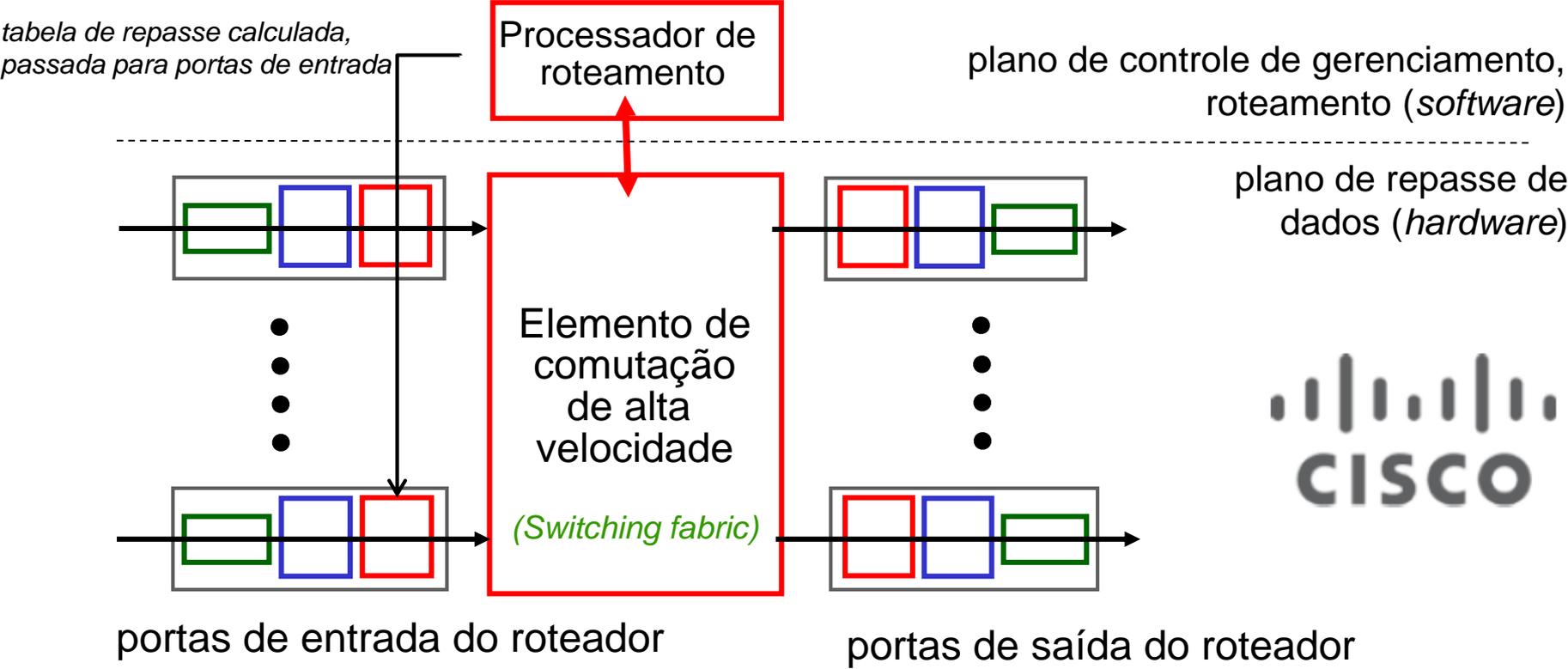
## 4.4 Repasse generalizado e SDN

- Casamento
- Ação
- Exemplos OpenFlow de casamento-mais-ação em andamento

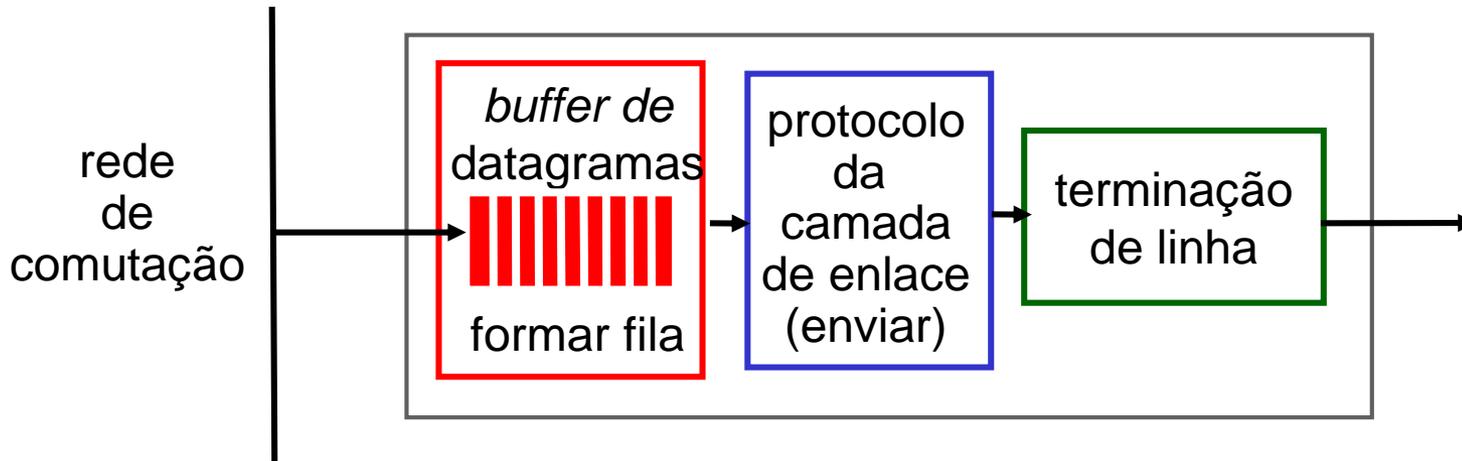
# Arquitetura de roteadores (CISCO/Huawei/Alcatel-Lucent/Juniper)

## 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

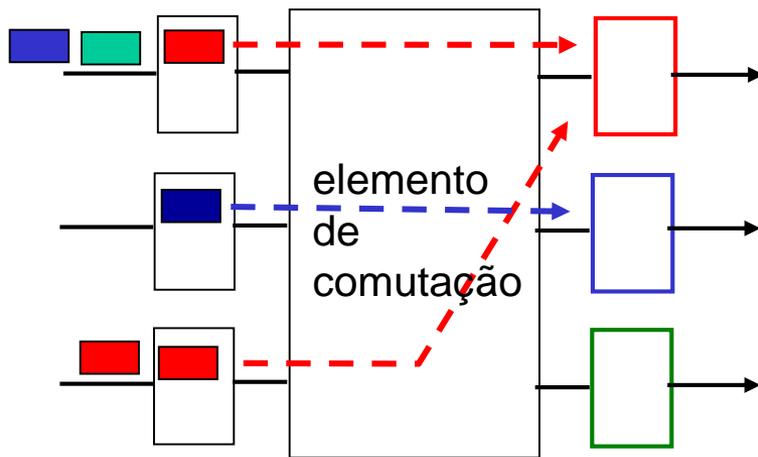


# Portas de saída

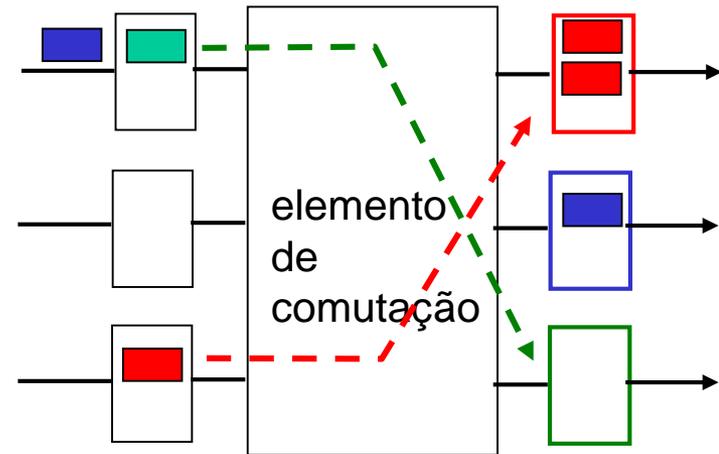


- ❖ **buffer** necessário quando datagramas chegam da rede de comutação mais rapidamente do que a taxa de transmissão – *isso é comum!*
- ❖ **disciplina de agendamento:** escolhe entre os datagramas enfileirados para transmissão
  - ❖ FCFS (*First-come, first-served*)
  - ❖ WFQ ([\*Weighted fair queuing\*](#))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

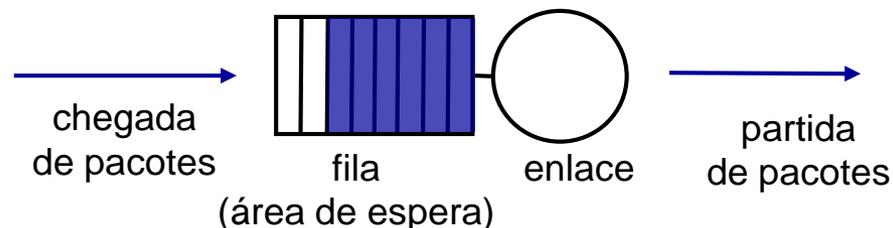
- ❖ colocar em *buffer* quando taxa de chegada excede velocidade de linha de saída
- ❖ *filas (atraso) e perdas devido a transbordamento do buffer da porta de saída!*
- ❖ *Qual pacote descartar? Active Queue Management*

# 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

# Mecanismos de agendamento

- **agendamento**: escolher próximo pacote a ser enviado pelo enlace
- **agendamento FIFO (first in first out)** : enviar na ordem de chegada à fila
  - exemplo do mundo real?
  - **politica de descarte**: se pacotes chegam a uma fila cheia: qual descartar?
    - **tail drop**: elimina pacote chegando
    - **prioridade**: elimina/remove pacote baseado em prioridade
    - **aleatório**: elimina/remove aleatoriamente

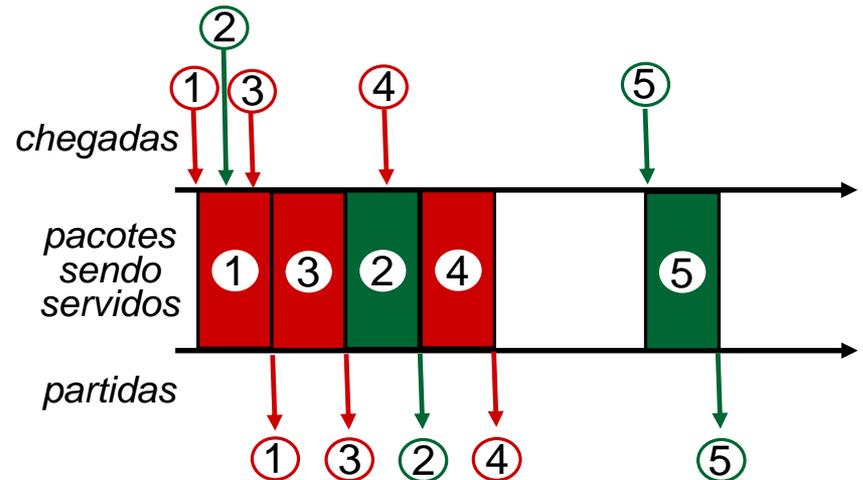
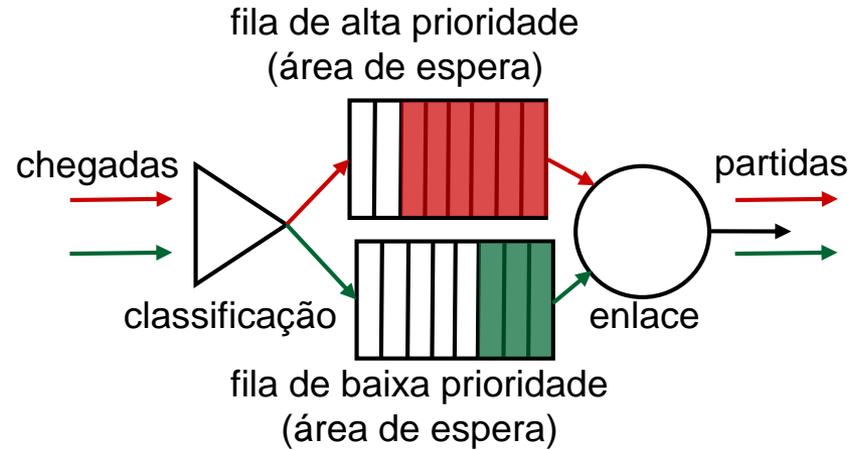


# Políticas de agendamento: prioridade

## agendamento por prioridade:

enviar pacote em fila com maior prioridade com maior prioridade

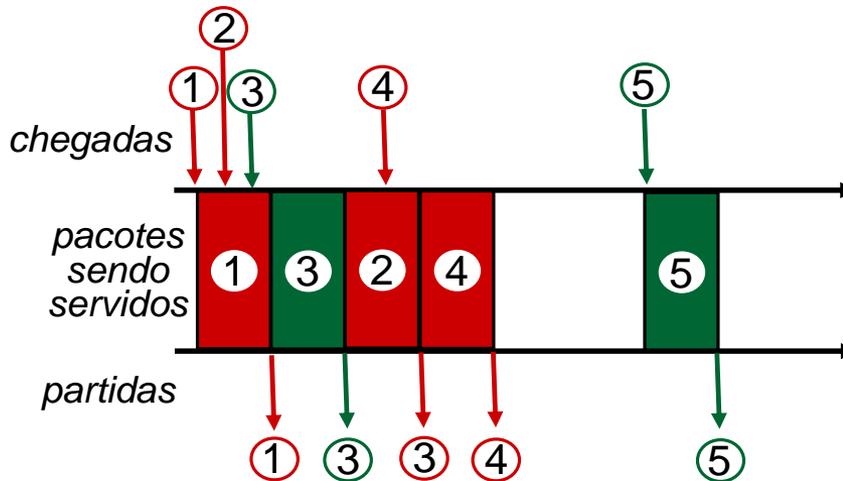
- múltiplas *classes*, com diferentes prioridades
  - classe pode depender de informações no cabeçalho como endereço IP de fonte/destino, número de porta, etc.
  - exemplo do mundo real?



# Políticas de agendamento: mais

## *agendamento Round Robin (RR) (rodízio):*

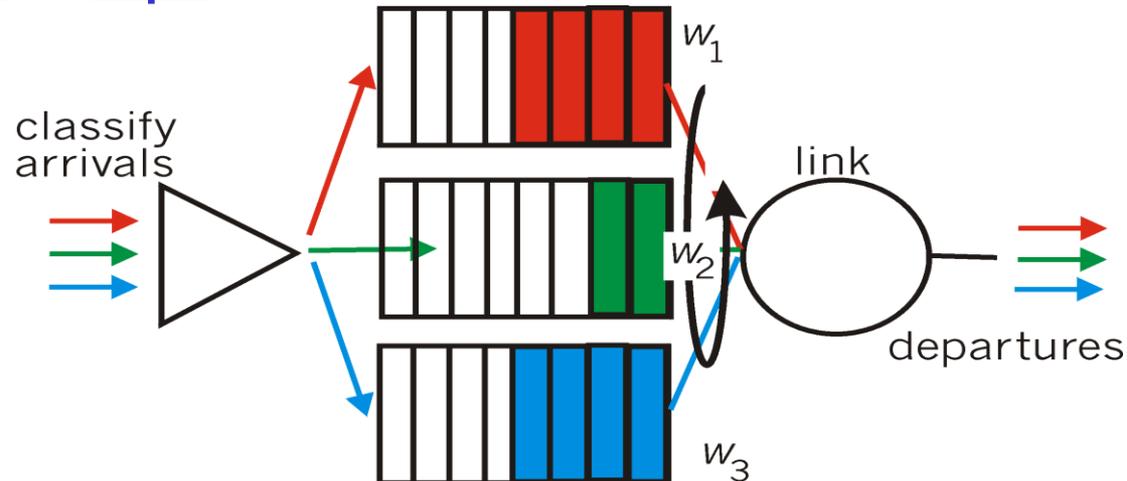
- múltiplas classes
- varre ciclicamente as filas, enviando um pacote completo de cada classe (se disponível)
- exemplo de tempo real?



# Scheduling policies: still more

## Weighted Fair Queuing (WFQ):

- Round Robin generalizado
- cada classe rece quantidade de serviço ponderado a cada ciclo
- exemplo do mundo real?
- Mais: [aqui](#) e [aqui](#)



# Capítulo 4: conteúdo

## 4.1 Introdução à camada de rede

- Plano de dados
- Plano de controle

## 4.2 o que tem dentro de um roteador?

## 4.3 IP: *Internet Protocol*

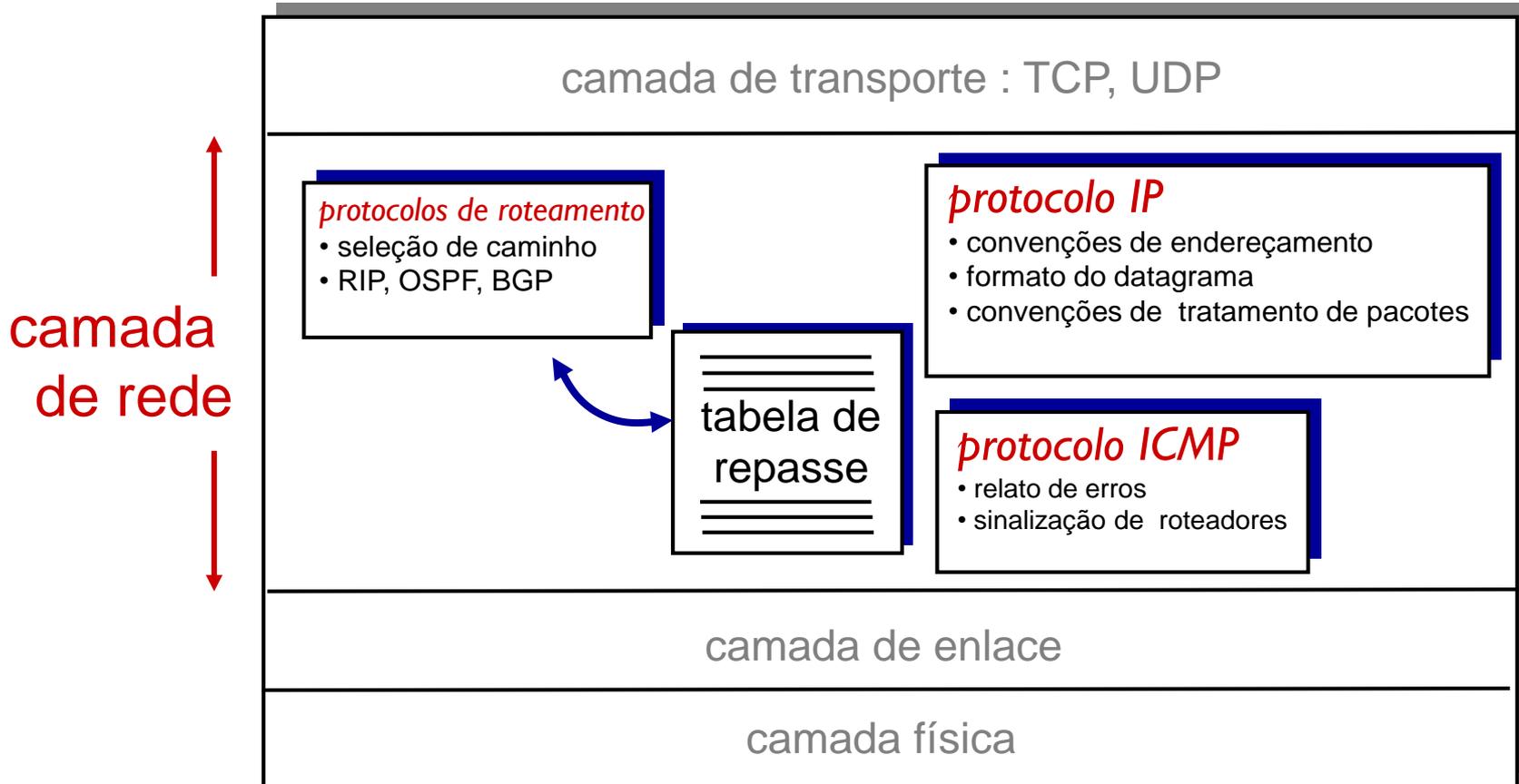
- **formato do datagrama**
- fragmentação
- endereçamento IPv4
- NAT
- IPv6

## 4.4 Repasse generalizado e SDN

- Casamento
- Ação
- Exemplos OpenFlow de casamento-mais-ação em andamento

# A camada de rede Internet

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



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

número da versão do protocolo IP (4 bits)

comp. cabeçalho (4 bits)

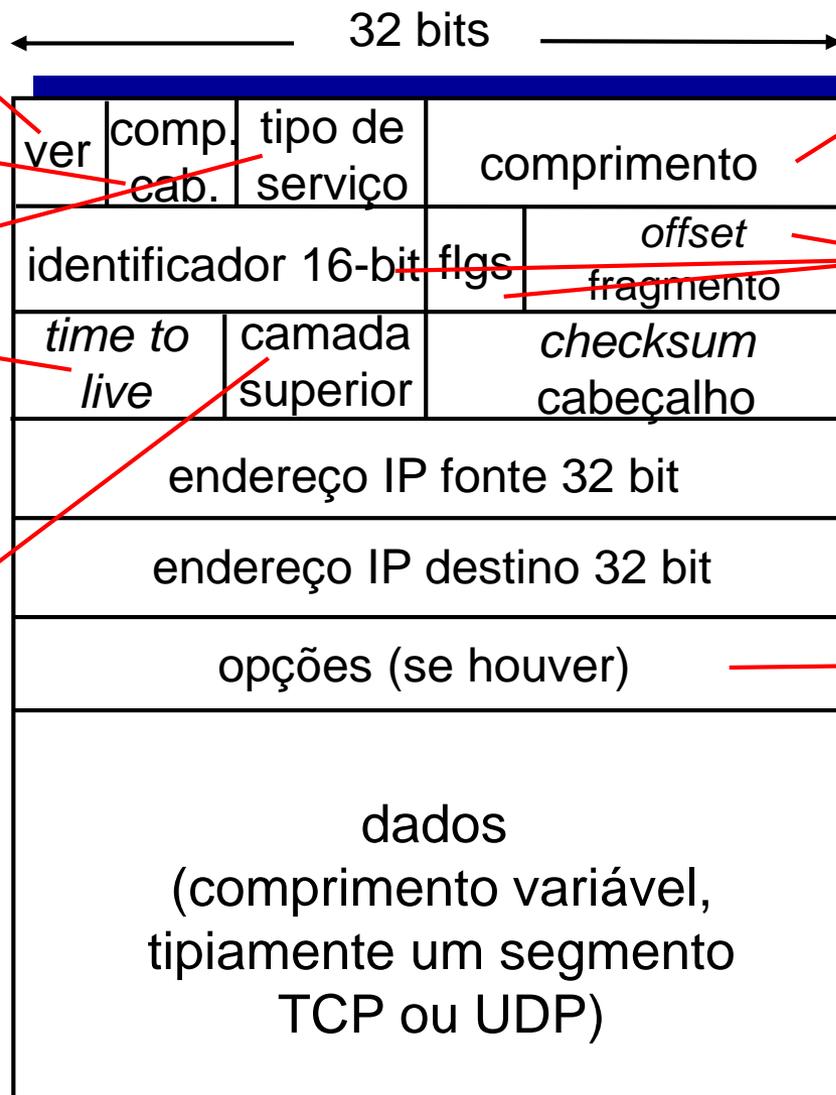
“tipo” de dados

max número de saltos restantes (decrecido a cada roteador)

protocolo da camada superior para entrega dos dados  
(<http://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml>)

## Quanto overhead?

- ❖ 20 bytes do TCP
- ❖ 20 bytes do IP
- ❖ = 40 bytes + overhead da camada de aplicação



comp. total do datagrama (bytes) (16 bits) para fragmentação/remontagem (Eliminado no IPv6)

e.g. *timestamp*, grava rota tomada, lista específica de roteadores a visitar (Eliminado no IPv6)

# Capítulo 4: conteúdo

## 4.1 Introdução à camada de rede

- Plano de dados
- Plano de controle

## 4.2 o que tem dentro de um roteador?

## 4.3 IP: *Internet Protocol*

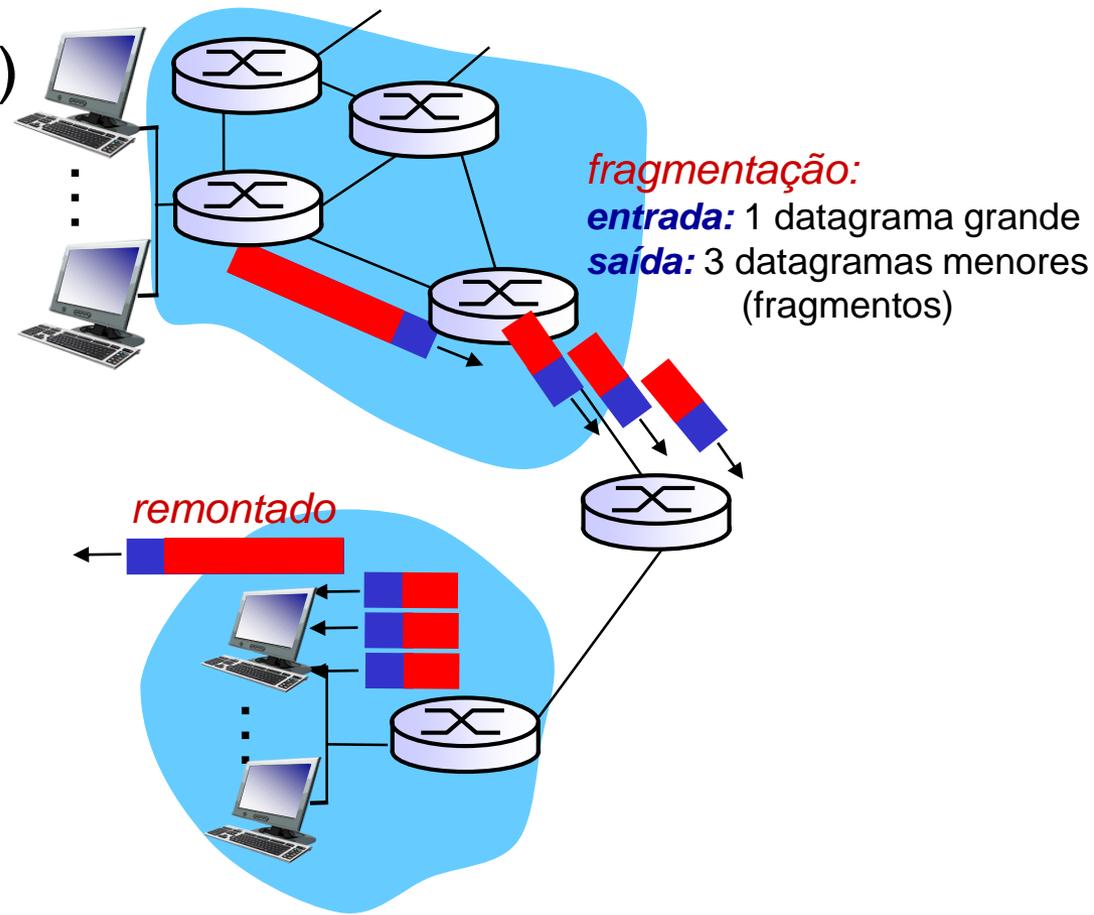
- formato do datagrama
- fragmentação
- endereçamento IPv4
- NAT
- IPv6

## 4.4 Repasse generalizado e SDN

- Casamento
- Ação
- Exemplos OpenFlow de casamento-mais-ação em andamento

# Fragmentação e remontagem IP

- ❖ enlaces de rede 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
  - I 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](http://media.pearsoncmg.com/aw/aw_kurose_network_2/applets/ip/ipfragmentation.html)

# Capítulo 4: conteúdo

## 4.1 Introdução à camada de rede

- Plano de dados
- Plano de controle

## 4.2 o que tem dentro de um roteador?

## 4.3 IP: *Internet Protocol*

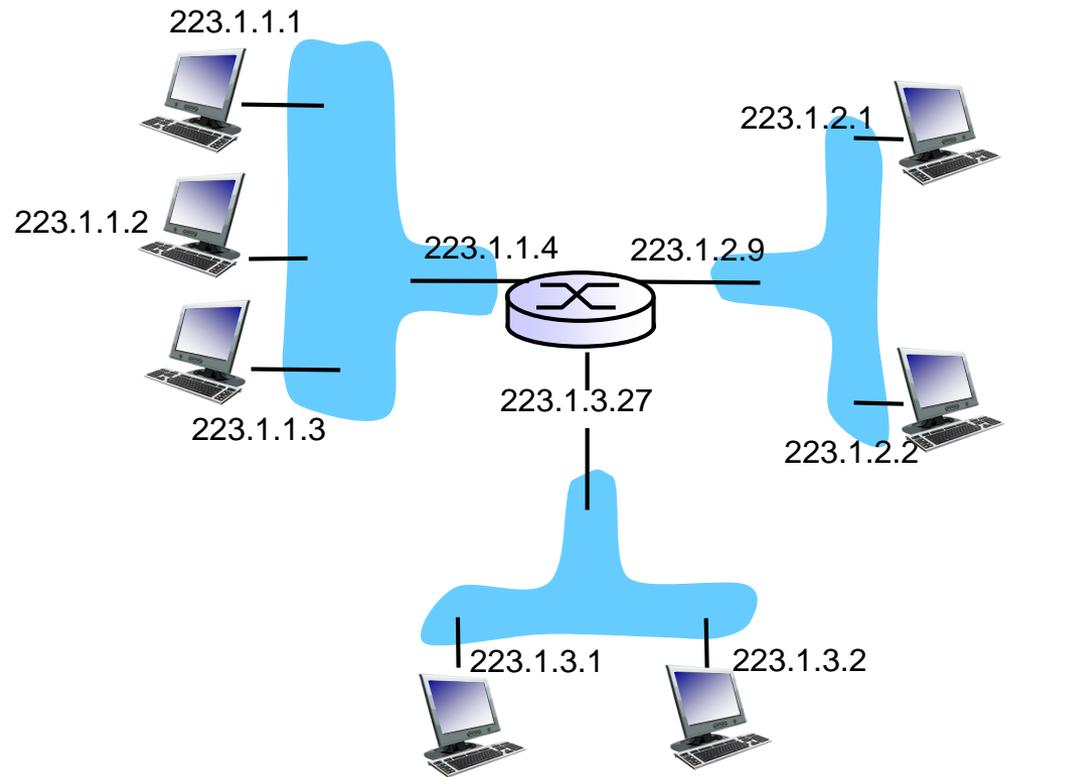
- formato do datagrama
- fragmentação
- **endereçamento IPv4**
- NAT
- IPv6

## 4.4 Repasse generalizado e SDN

- Casamento
- Ação
- Exemplos OpenFlow de casamento-mais-ação em andamento

# Endereçamento IP : introdução

- ❖ **endereço IP :** identificador de 32 bits para *interfaces* de *hosts* e roteadores
- ❖ **interface:** conexão entre *host* ou roteador e enlace físico
  - roteador tipicamente têm múltiplas interfaces
  - *host* tipicamente tem 1 ou duas interfaces (e.g., Ethernet cabeada, 802.11 sem fio)
- ❖ **endereços IP associados com cada interface – aprox. 4 bilhões de interfaces**



$$223.1.1.1 = \underbrace{11011111}_{223} \underbrace{00000001}_{1} \underbrace{00000001}_{1} \underbrace{00000001}_{1}$$

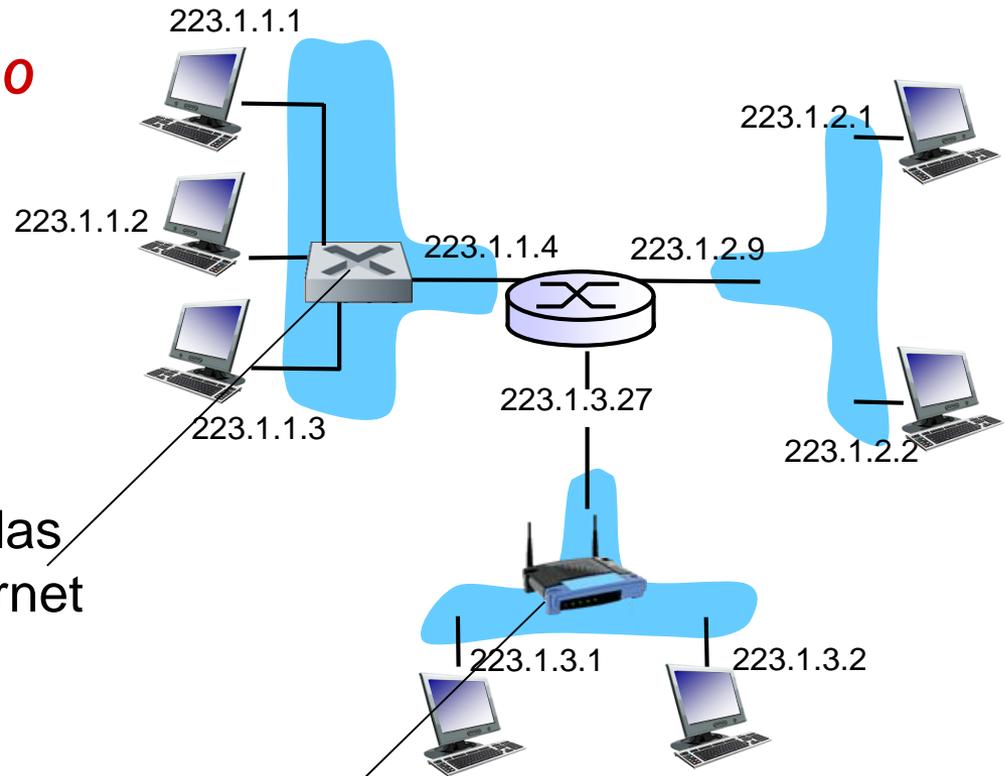
# Endereçamento IP : introdução

**Q:** como as interfaces são de fato conectadas?

**R:** mais à frente.

**R:** interfaces Ethernet cabeadas conectadas a switches Ethernet

**Por hora:** não precisamos nos preocupar em como as interfaces se conectam (sem roteador interveniente)



**R:** interfaces WiFi sem fio conectadas a estação base WiFi

# Sub-redes

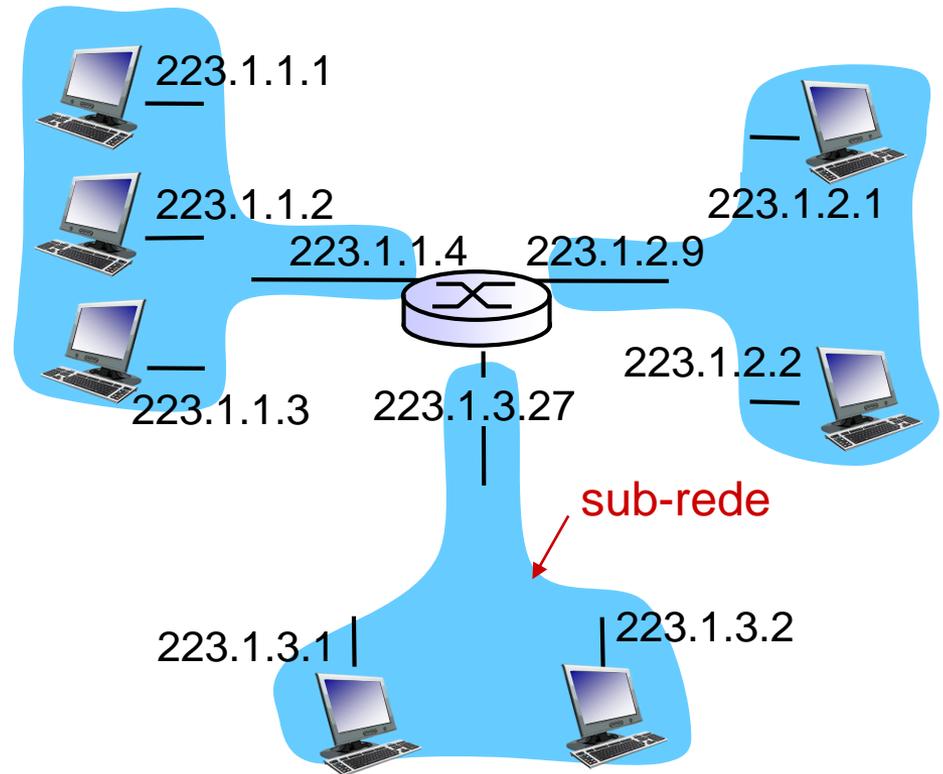
Com os prefixos **sub** e **sob**, usa-se o hífen diante de palavra iniciada por r. Exemplos: sub-região sub-reitor sub-regional sob-roda

## ❖ endereço IP :

- parte sub-rede - bits de ordem mais alta
- parte *host* - bits de ordem mais baixa

## ❖ o que é uma sub-rede ?

- interfaces de dispositivos com mesma parte sub-rede do endereço IP
- podem alcançar-se fisicamente *sem passar por roteador*

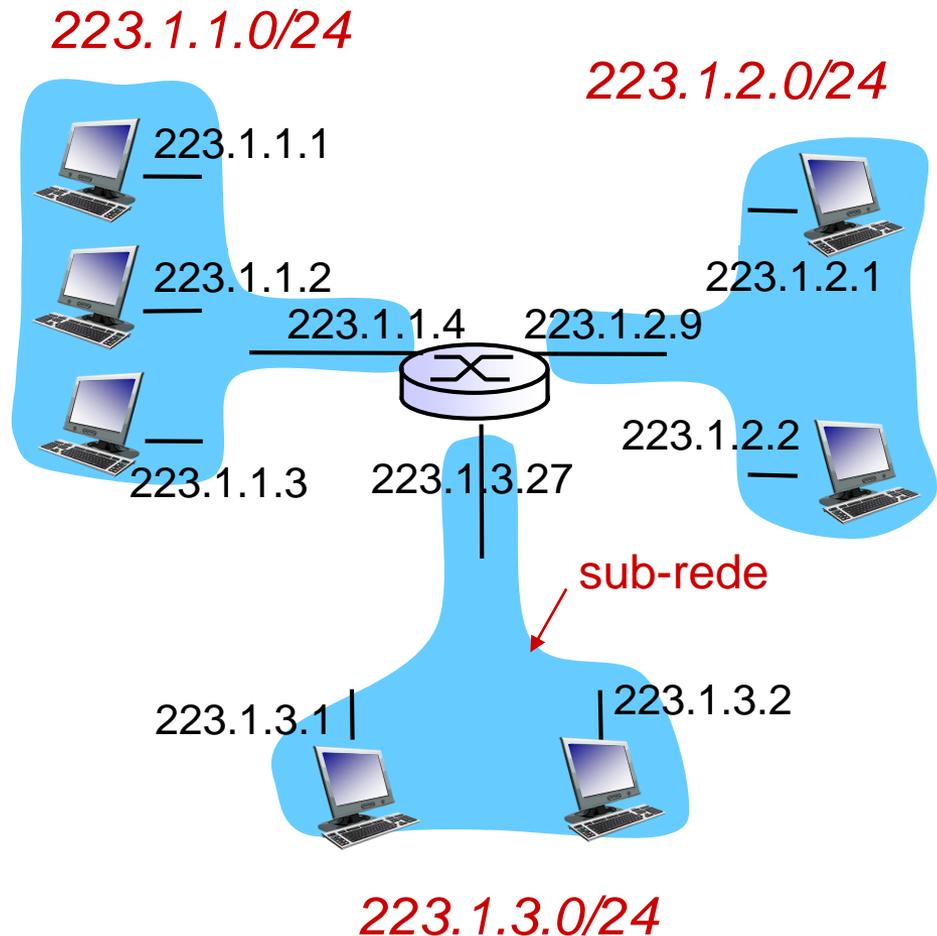


rede consistindo de 3 sub-redes

# Sub-redes

## receita

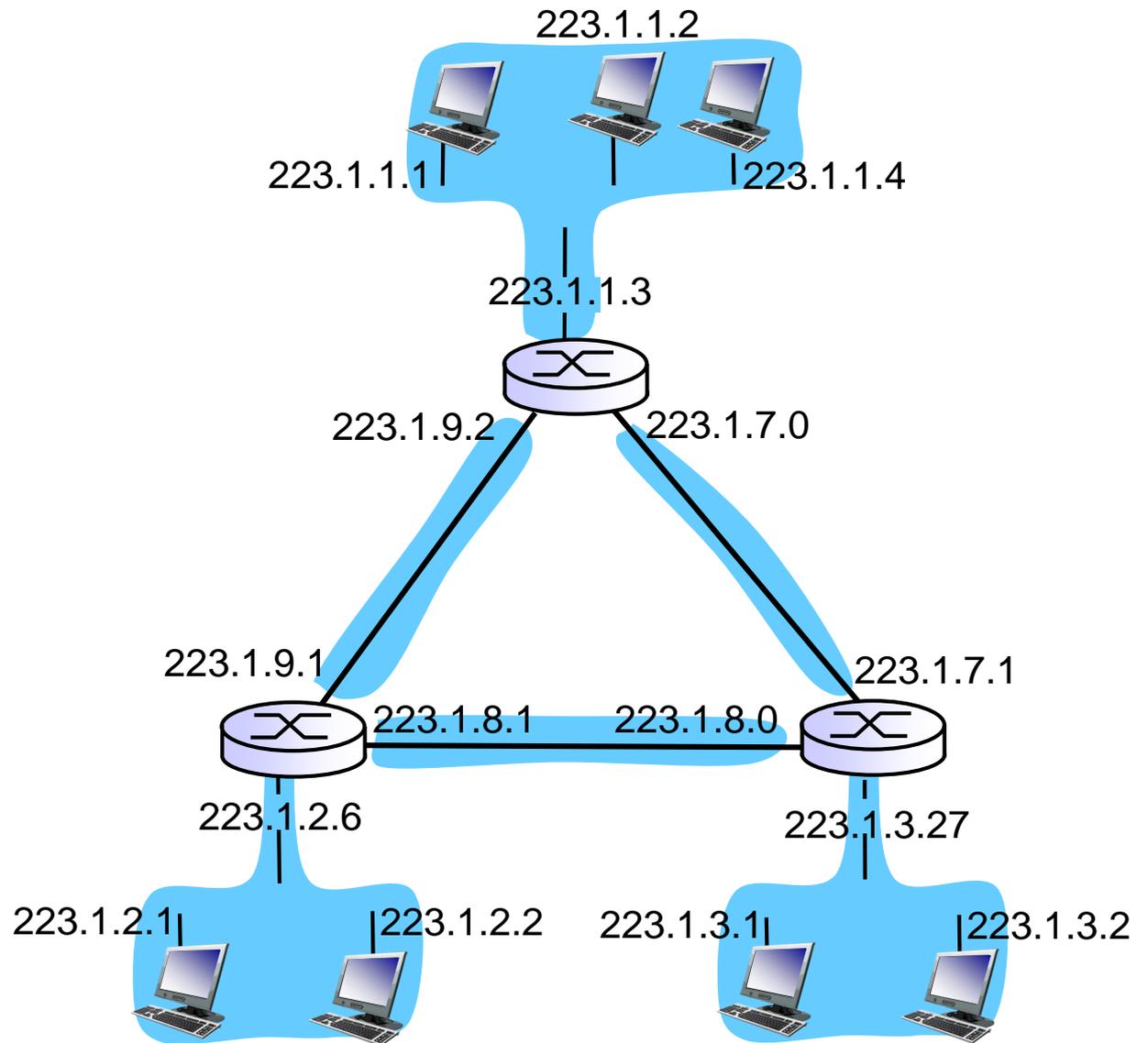
- ❖ para determinar as sub-redes, separe cada interface do seu *host* ou roteador, criando ilhas de redes isoladas
- ❖ cada rede isolada é chamada de *sub-rede*



mascara de sub-rede : /24

# Sub-redes

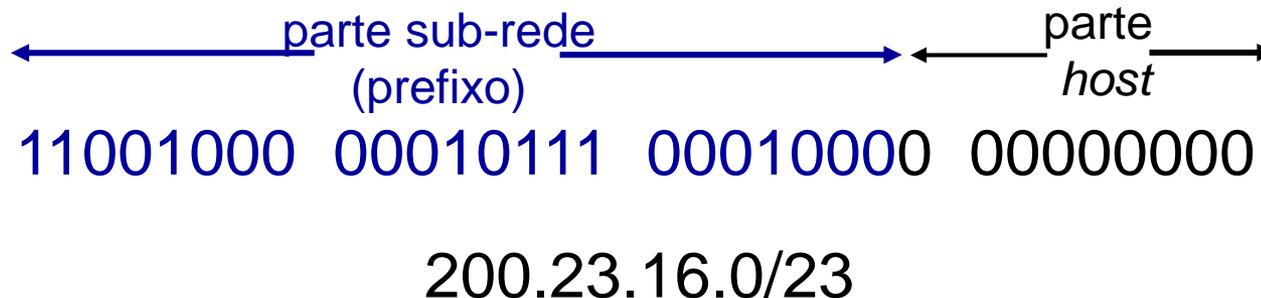
Quantas?



# Endereçamento IP : CIDR [[RFC1338 \(1992\)](#)...[RFC4632 \(2006\)](#)]

## CIDR: *Classless InterDomain Routing* (sáider)

- porção sub-rede do endereço tem comprimento arbitrário (prefixo)
- formato do endereço: **a.b.c.d/x**, sendo x # bits na porção sub-rede do endereço



- *Máscara: 255.255.254.0*