

# PTC 3360

## 3. A Camada de Rede – Parte II

(Kurose, Seções 1.4 e 4.2)

Agosto 2023

# Capítulo 2 - Conteúdo

---

## 2.1 A camada de aplicação

## 2.2 Princípios da transferência confiável de dados (recorte da camada de transporte)

## 2.3 Camada de rede

A Introdução à camada de rede

**B O que tem dentro de um roteador?**

C Atrasos e vazão

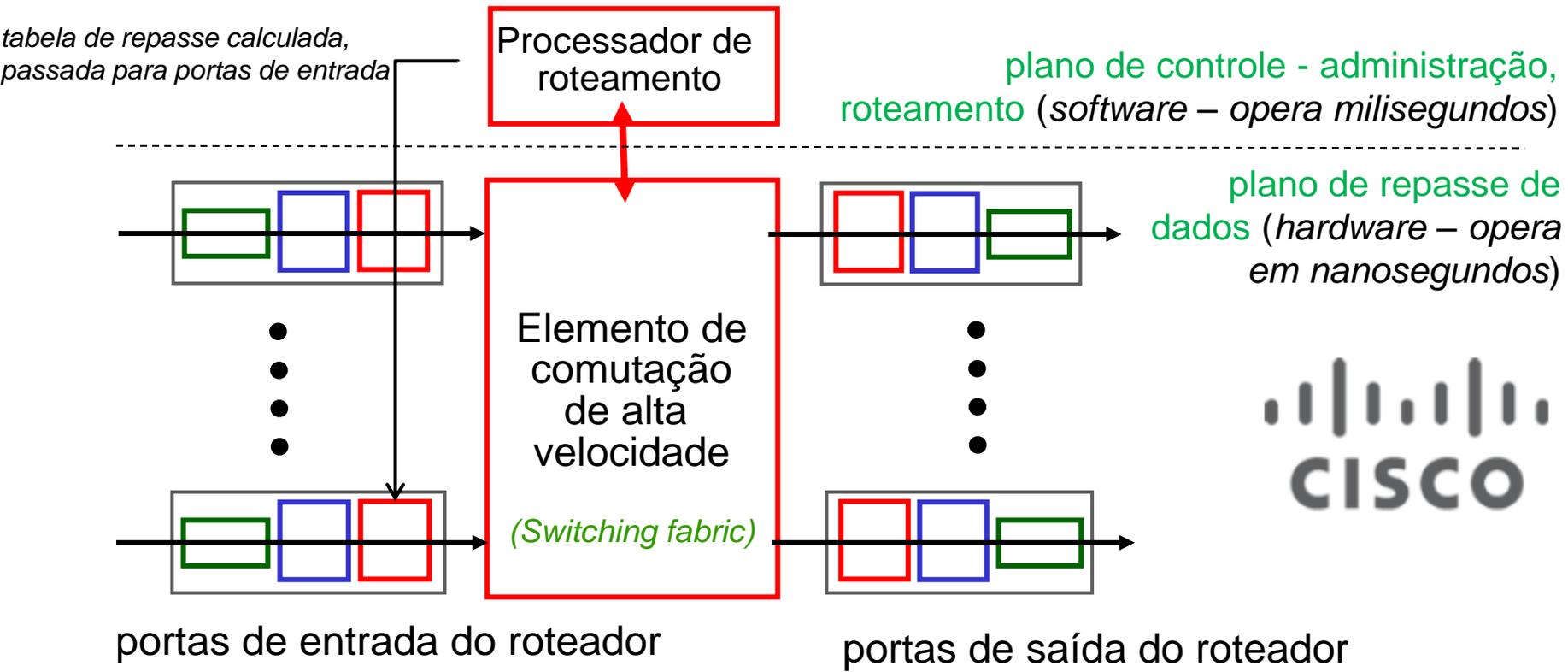
D Endereçamento IP

E Repasse generalizado e SDN

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

## 2 funções chaves do roteador:

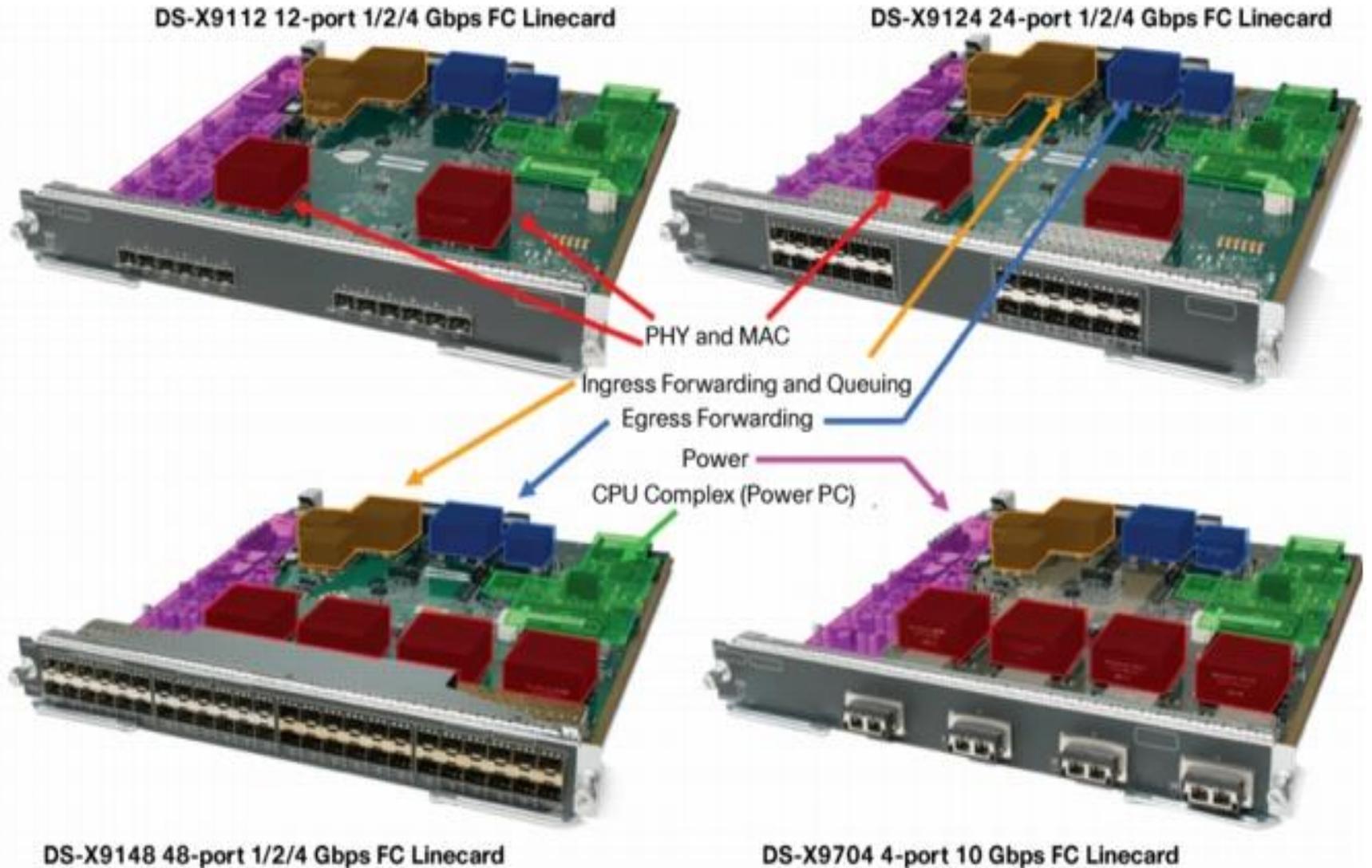
- ❖ Rodar protocolos e algoritmos de roteamento
- ❖ *Repassar ou comutar* datagramas de enlace de entrada para enlace de saída



# Exemplo: Arquitetura da Família Cisco MDS 9000



# Exemplo: Arquitetura da Família Cisco MDS 9000



# Tabela de repasse: ideia inicial

Intervalo do endereço de destino	Interface do Enlace
11001000 00010111 00010000 00000000 : 11001000 00010111 00010111 11111111	0
11011000 00000000 00000000 00000000 : 11011000 11111111 11111111 11111111	1
11111111 00000000 00000000 00000000 : 11111111 11111111 11111111 11111111	2
caso contrário	3

# Casamento com prefixo mais longo

## *Casamento com prefixo mais longo*

Quando busca-se entrada de tabela de repasse para dado endereço de destino, usa-se 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:

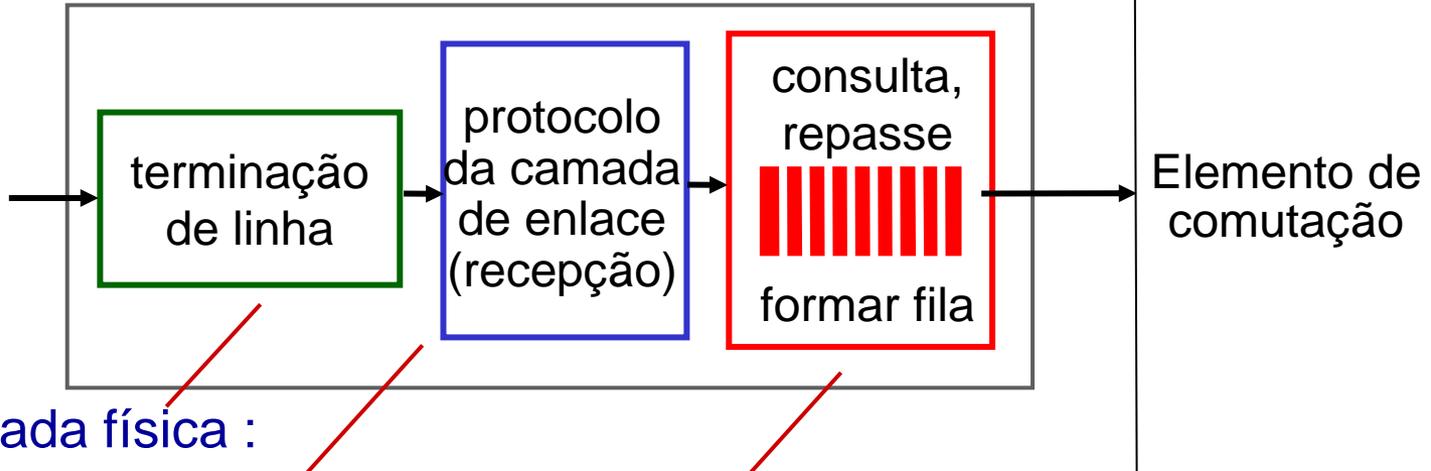
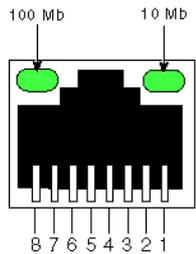
DA: 11001000 00010111 00010**110** 10100001

qual interface?

DA: 11001000 00010111 00011000 **10101010**

qual interface?

# Funções da porta de entrada



Camada física :  
recepção em nível de bit

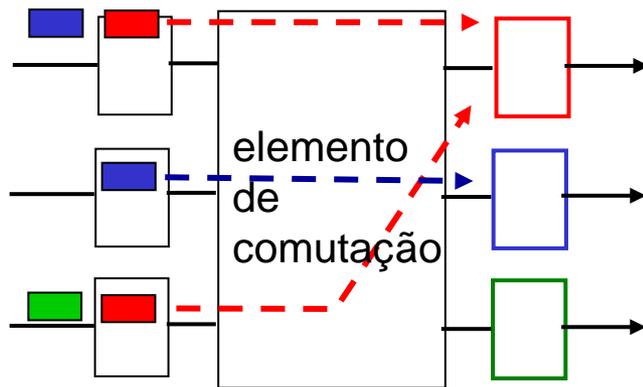
Camada de enlace:  
e.g., Ethernet

## Camada de rede:

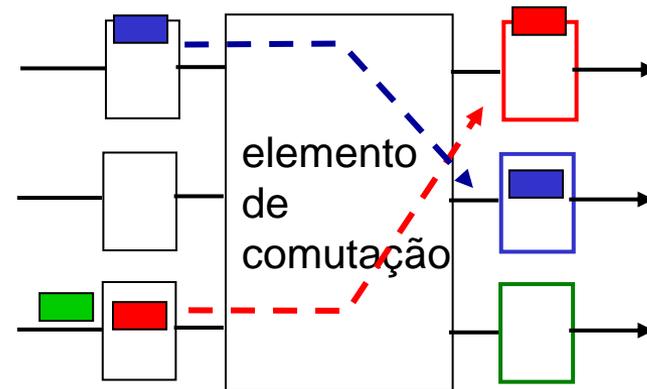
- ❖ Define porta de saída usando tabela de repasse na memória da porta de entrada (“*match plus action*” – *casamento mais ação*)
- ❖ Objetivo: completar processamento da porta de entrada em “velocidade de linha” (ns)
- ❖ Fila: se datagramas chegam mais rápido do que taxa de repasse para elemento de comutação

# Filas na porta de entrada

- ❖ Se a comutação for mais lenta do que as portas de entrada combinadas, filas podem ocorrer nas portas de entrada
  - *Atraso de fila e perdas (lembre do mecanismos de rdt) 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



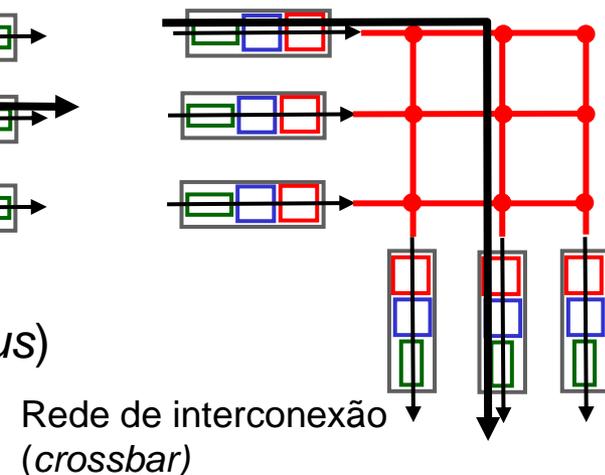
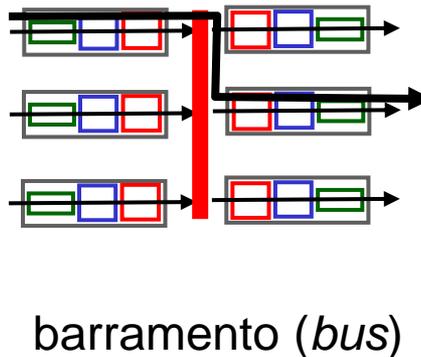
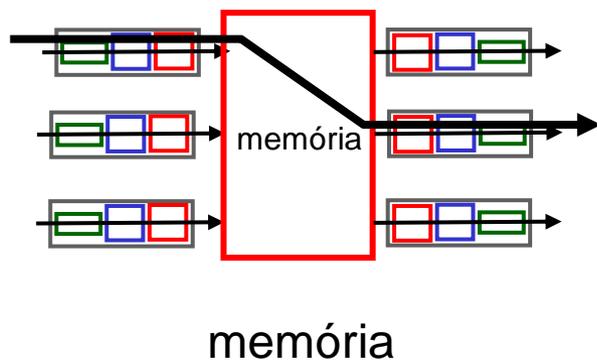
Contenção na porta de saída:  
apenas 1 datagrama vermelho pode  
ser transferido.  
*pacote vermelho inferior é bloqueado*



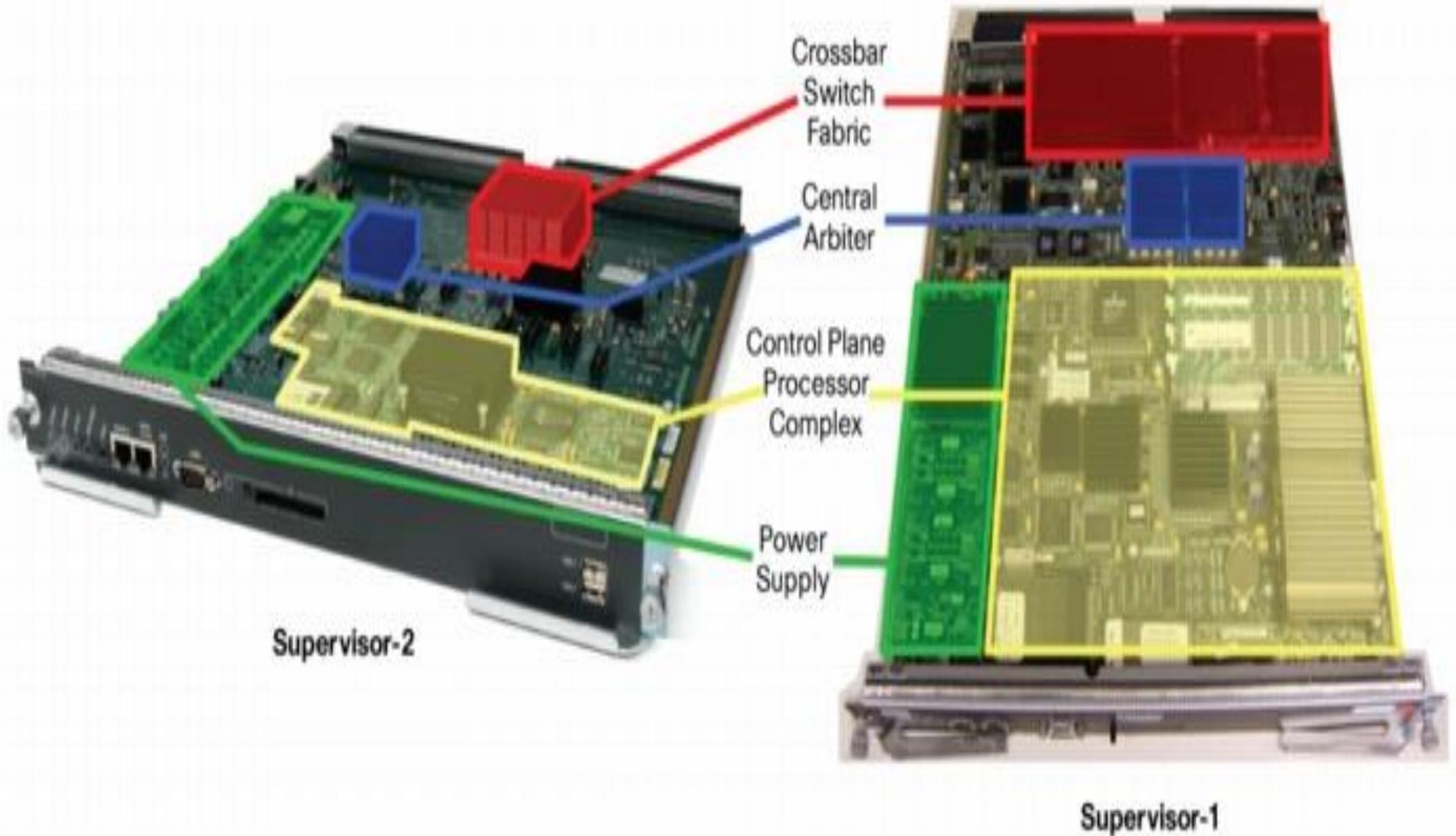
1 tempo de pacote  
depois: pacote verde  
sofre bloqueio HOL

# 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 para evitar filas na entrada.
- ❖ 3 tipos de elementos de comutação



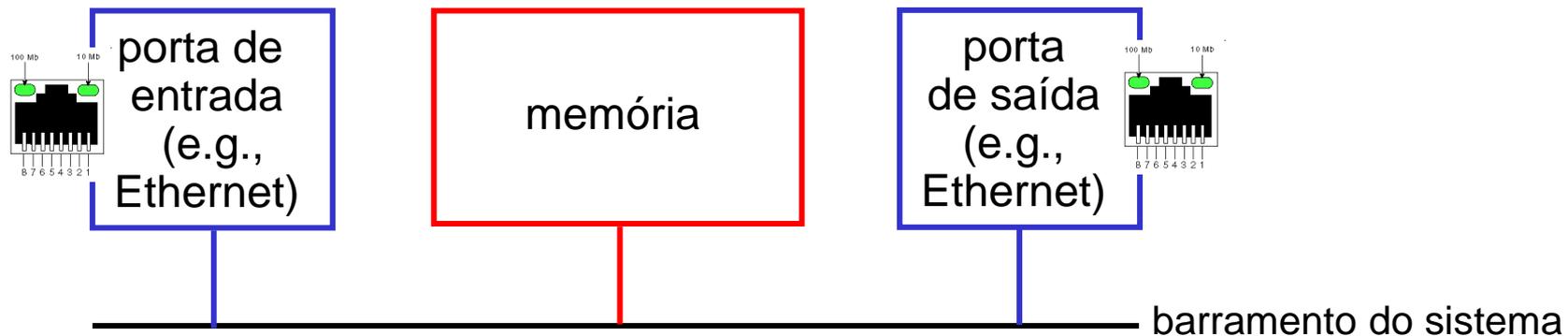
# Exemplo: Arquitetura da Família Cisco MDS 9000



# Comutação via memória

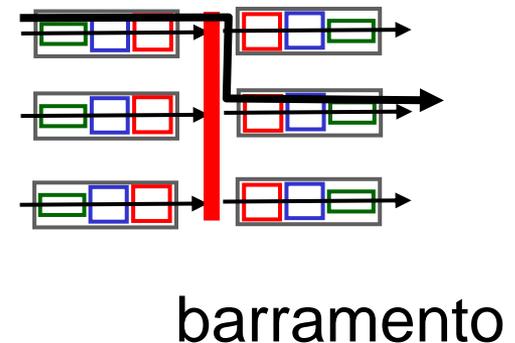
## *1ª geração de roteadores:*

- ❖ 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](#)



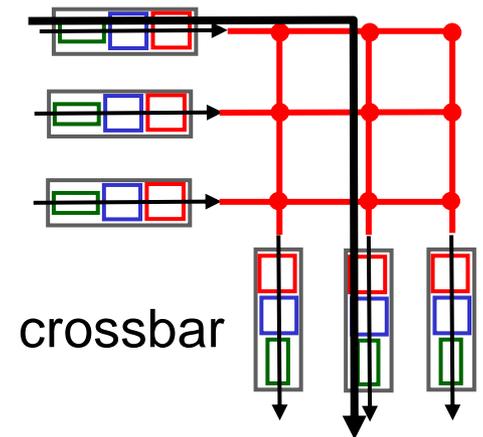
# Comutação via barramento

- ❖ Transmissão do datagrama da porta de entrada para porta de saída é feita via barramento único.
- ❖ Porta de saída é definida em cabeçalho interno ao roteador.
- ❖ Apenas 1 pacote pode cruzar o barramento de cada vez
- ❖ *Contenção pelo barramento*: velocidade de comutação limitada pela velocidade do barramento

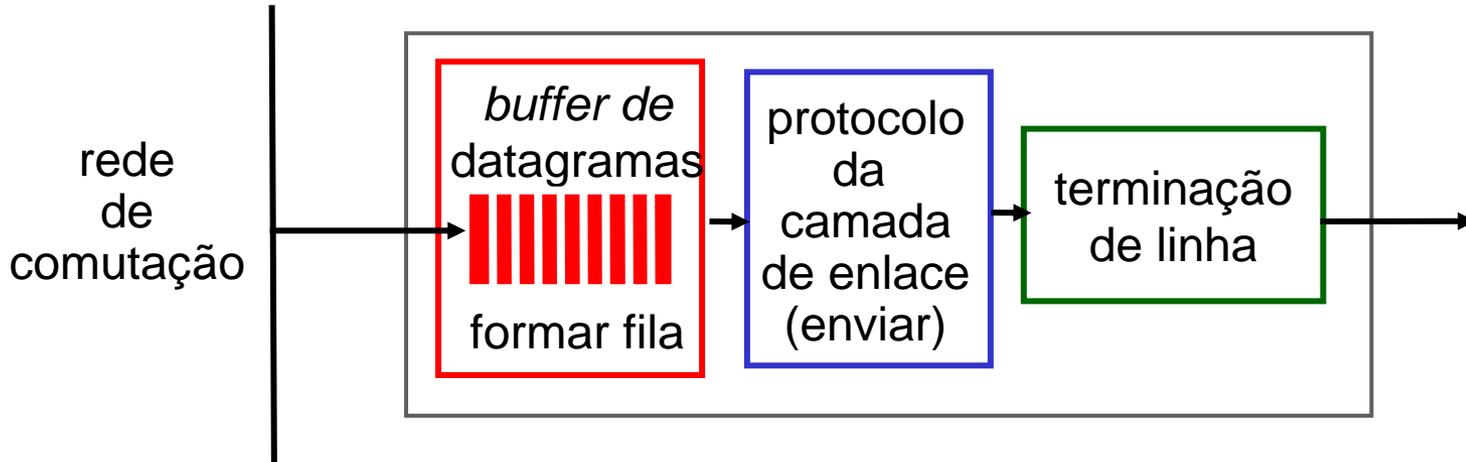


# Comutação via *crossbar*

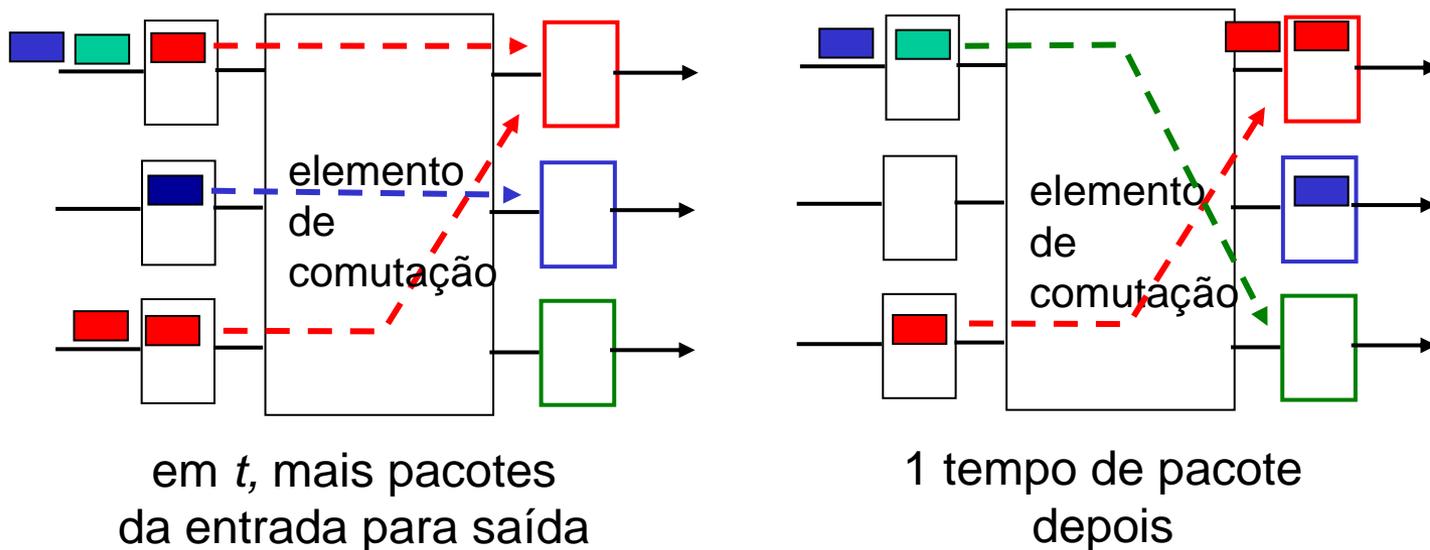
- ❖ Agora são  $2N$  barramentos ligando  $N$  entradas a  $N$  saídas – cruzamento pode ser aberto ou fechado pelo controlador do *switch fabric*.
- ❖ Supera limitações de velocidade do barramento único, sendo capaz de repassar vários pacotes em paralelo.



# Portas de saída

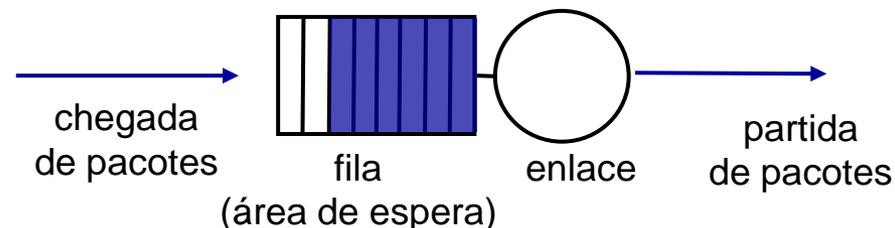


- ❖ Um *buffer* é necessário quando datagramas chegam da rede de comutação mais rapidamente do que a taxa de transmissão.



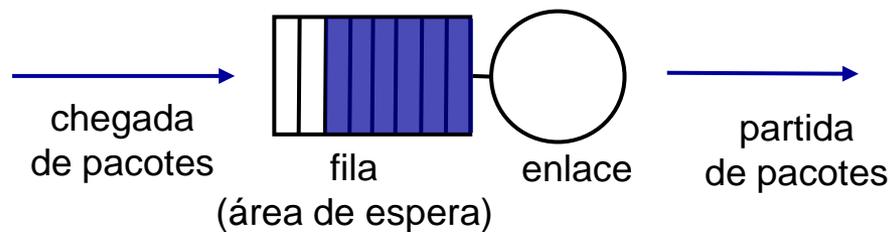
# Fila na porta de saída

- ❖ **Mecanismo de agendamento:** escolhe entre os datagramas enfileirados para transmissão
  - ❖ FIFO (*First-come, first-served*)
  - ❖ Agendamento por prioridade
  - ❖ Agendamento *round robin* (rodízio)
  - ❖ *Weighted fair queuing*
  - ❖ **Pode-se priorizar pacotes de acordo com fonte?**  
Questão da neutralidade da rede.



# Mecanismos de agendamento

- **Agendamento FIFO (First In First Out):** enviar na ordem de chegada à fila
  - Exemplo do mundo real?
    - Fila do bandejão.

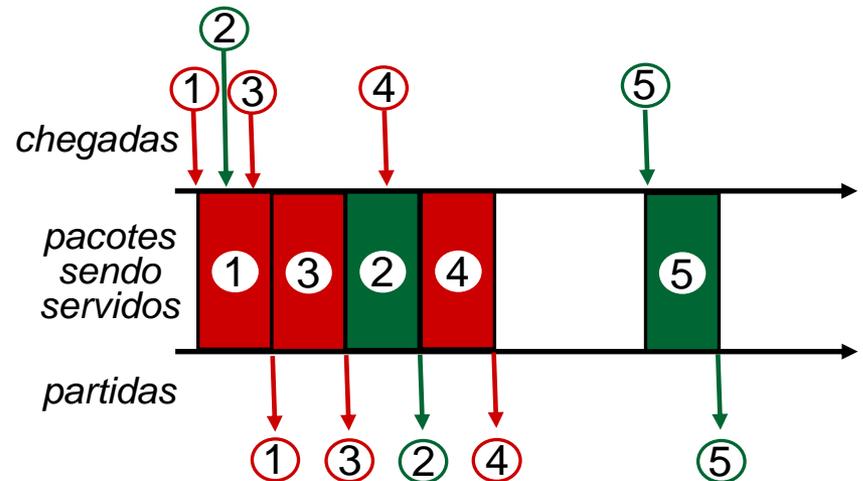
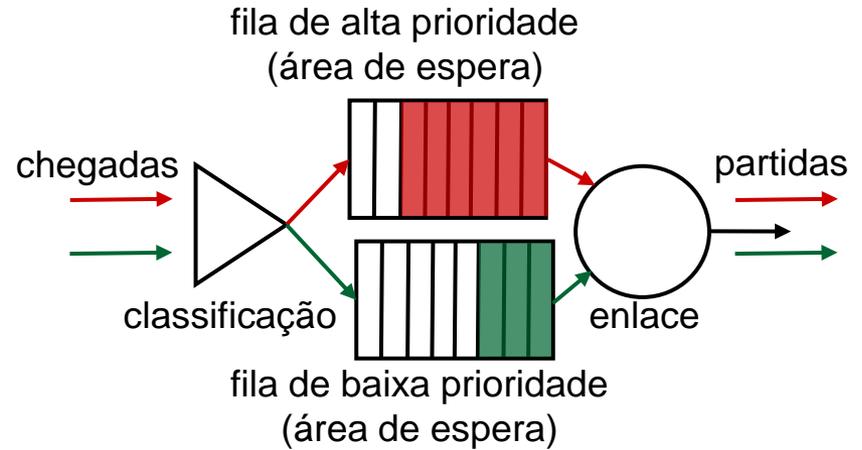


# Mecanismos de agendamento

## Agendamento por prioridade:

diferentes filas para prioridades diferentes; fila com menor prioridade só é atendida se a com maior prioridade está vazia.

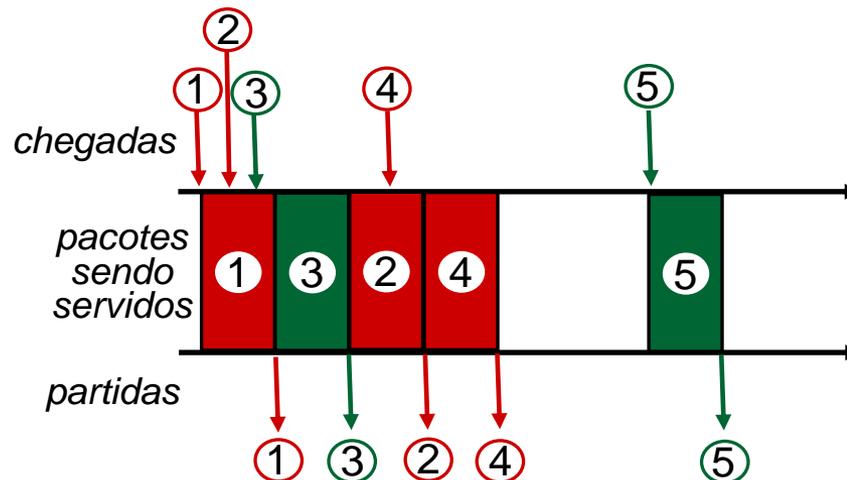
- Permite múltiplas *classes*, com diferentes prioridades:
  - Classe pode depender de informações no cabeçalho como endereço IP de fonte/destino, número de porta, etc.
  - Exemplo do mundo real:
    - Filas separadas num banco; único caixa chama sempre alguém da fila de prioritários se ela não está vazia.



# Mecanismos de agendamento

## Agendamento Round Robin (rodízio):

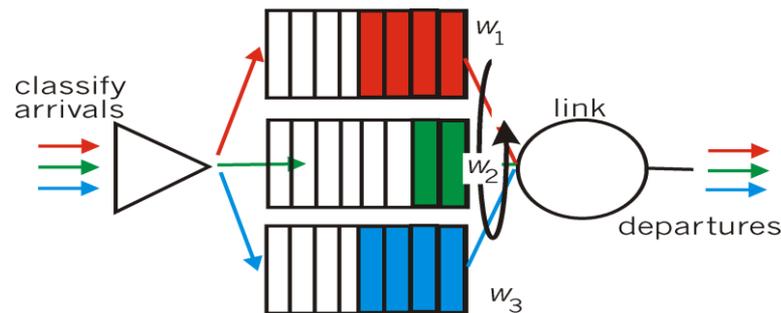
- Múltiplas classes
- Varre ciclicamente as filas, enviando um pacote completo de cada classe (se disponível)
- Exemplo do mundo real?
  - Fila com chamada por senha com atendimento prioritário; é chamado um prioritário e um comum por vez.



# Mecanismos de agendamento

## Weighted Fair Queuing (WFQ):

- *Round Robin* generalizado
- cada classe recebe quantidade de serviço ponderado a cada ciclo
- Exemplo do mundo real?
  - Igual ao anterior, mas com número de chamados de cada grupo escolhido de forma “inteligente”.



- Exercício interativo do livro.

# Transbordamento

---

- Pode haver também transbordamento do buffer da porta de saída.
- Política de qual pacote descartar é chamada de *administração ativa de fila*.
- *Exemplos:*
  - *Tail drop*: elimina pacote chegando
  - *Prioridade*: elimina/remove pacote baseado em prioridade
  - *Aleatório*: elimina/remove aleatoriamente

# A questão da neutralidade da rede

O que é a neutralidade da rede?

- *Técnica*: como uma rede de acesso deve alocar e compartilhar seus recursos
  - Mecanismos: agendamento de pacotes, administração ativa de filas
- Princípios *sociais, econômicos e políticos*.
  - Proteção à liberdade de expressão
  - Encorajamento da inovação e competição
- Implementada via políticas públicas, normas técnicas e leis.

*Países diferentes usam abordagens diferentes para a neutralidade da rede.*

# A questão da neutralidade no Brasil

- A neutralidade da rede regulada no Brasil com o Marco Civil da Internet, [Lei n.º 12.965/2014](#), sancionada em 2014 e regulamentada em 2016.

## Seção I Da Neutralidade de Rede

Art. 9º O responsável pela transmissão, comutação ou roteamento tem o dever de tratar de forma isonômica quaisquer pacotes de dados, sem distinção por conteúdo, origem e destino, serviço, terminal ou aplicação.

# Capítulo 2 - Conteúdo

---

## 2.1 A camada de aplicação

## 2.2 Princípios da transferência confiável de dados (recorte da camada de transporte)

## 2.3 Camada de rede

A Introdução à camada de rede

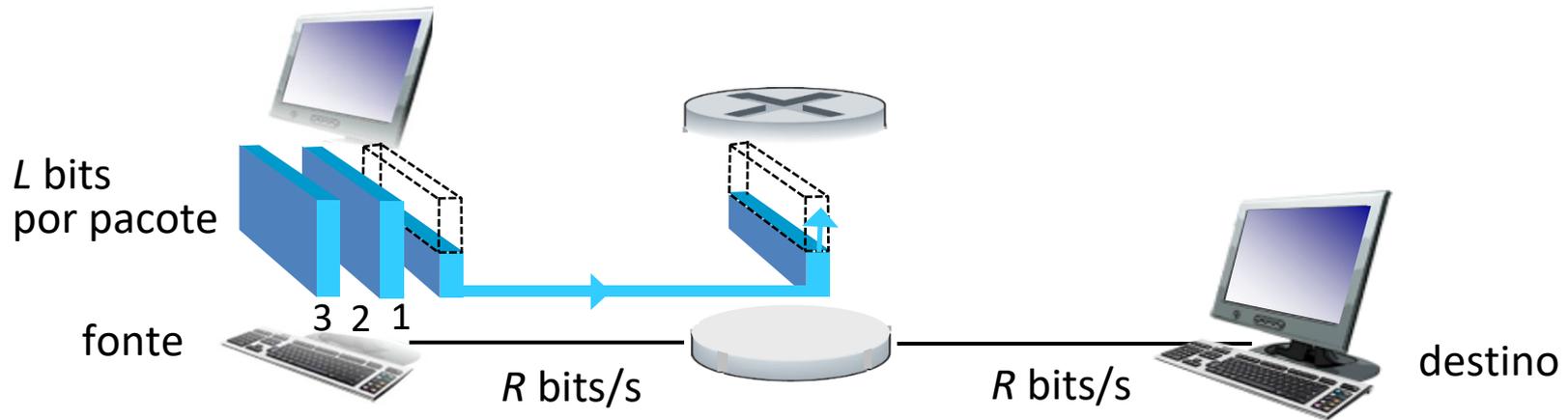
B O que tem dentro de um roteador?

C Atrasos e vazão

D Endereçamento IP

E Repasse generalizado e SDN

# Comutação de pacotes: *store-and-forward*



- ❖ leva  $L/R$  segundos para transmitir (inserir) pacote de  $L$  bits em um enlace a  $R$  bits/s
- ❖ *Store and forward* (adotado em geral em roteadores): pacote inteiro precisa chegar no roteador antes que possa ser transmitido ao próximo enlace
- ❖ No exemplo acima, atraso fim-fim para 1 pacote =  $2L/R$  (assumindo zero atraso de propagação)

Q. Qual o atraso para transmitir os 3 pacotes?

$4L/R$

Q. Qual o atraso para transmitir 1 pacote em uma rota com  $N$  enlaces todos com taxa  $R$ ?

$NL/R$

Q. Qual o atraso para transmitir  $P$  pacotes?

$(P+N-1)L/R$

# Exercício da Lista I

(Kurose2013, p. 57)

Considere o envio de um arquivo grande de  $F$  bits do hospedeiro  $A$  para o hospedeiro  $B$ .

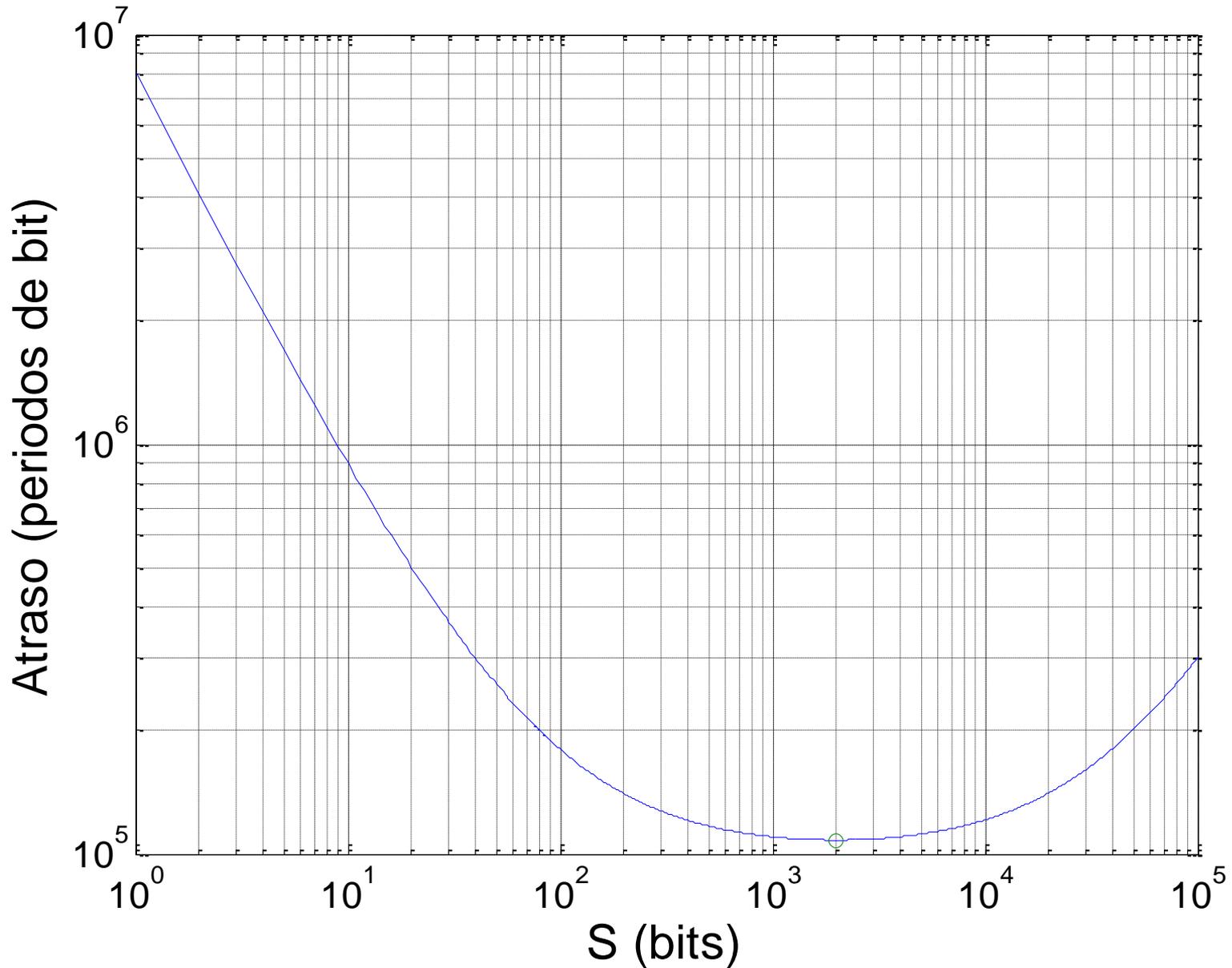
Há três enlaces (e dois comutadores) entre  $A$  e  $B$ , e os enlaces não estão congestionados (isto é, não há atrasos de fila).

O hospedeiro  $A$  fragmenta o arquivo em segmentos de  $S$  bits cada e adiciona 80 bits de cabeçalho a cada segmento, formando pacotes de  $L=80+S$  bits.

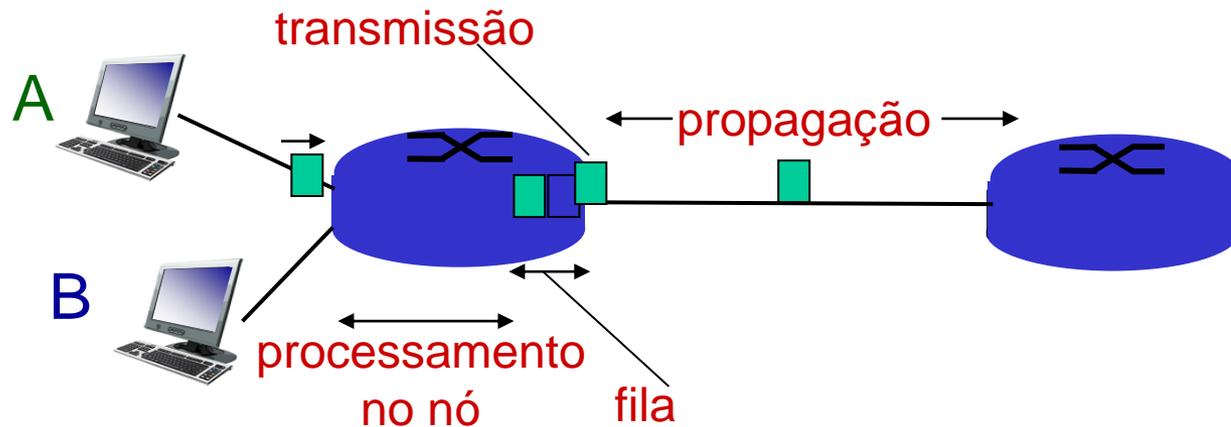
Cada enlace tem uma taxa de transmissão de  $R$  bits/s.

Qual o valor de  $S$  que minimiza o atraso para levar o arquivo de  $A$  para  $B$ ? Desconsidere o atraso de propagação e considere que  $F$  é um múltiplo inteiro de  $S$ .

# Exercício da Lista 1 – Gráfico do Resultado para $F = 10^5$ ([colab](#))



# Quatro fontes de atraso de pacotes



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{fila}} + d_{\text{trans}} + d_{\text{prop}}$$

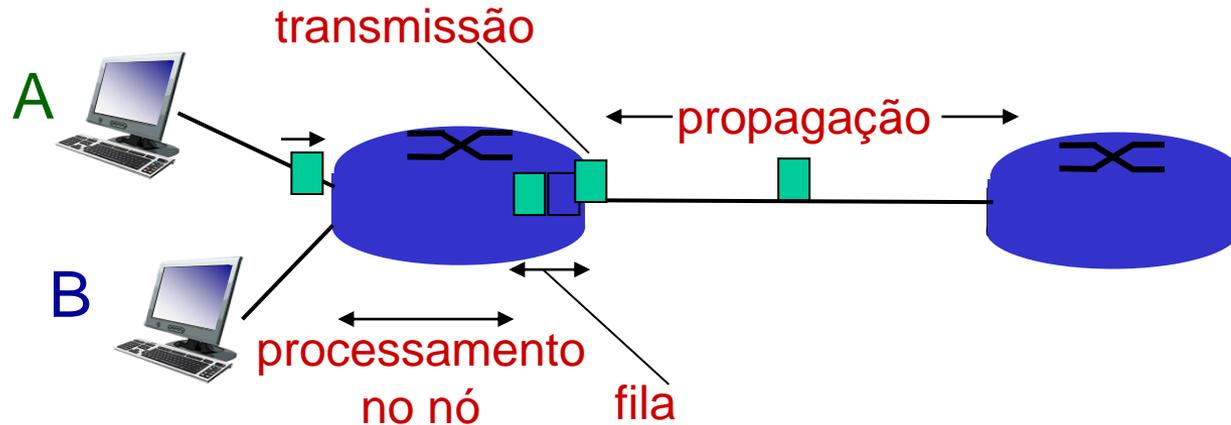
## $d_{\text{proc}}$ : processamento no nó

- Detecção de erros em bits
- Determinação do enlace de saída
- De nano a microssegundos

## $d_{\text{fila}}$ : atraso de fila

- Tempo esperando no enlace de saída para transmissão
- Depende do nível de congestionamento no roteador
- De micro a milissegundos

# Quatro fontes de atraso de pacotes



$$d_{\text{nodal}} = d_{\text{proc}} + d_{\text{fila}} + d_{\text{trans}} + d_{\text{prop}}$$

## $d_{\text{trans}}$ : atraso de transmissão:

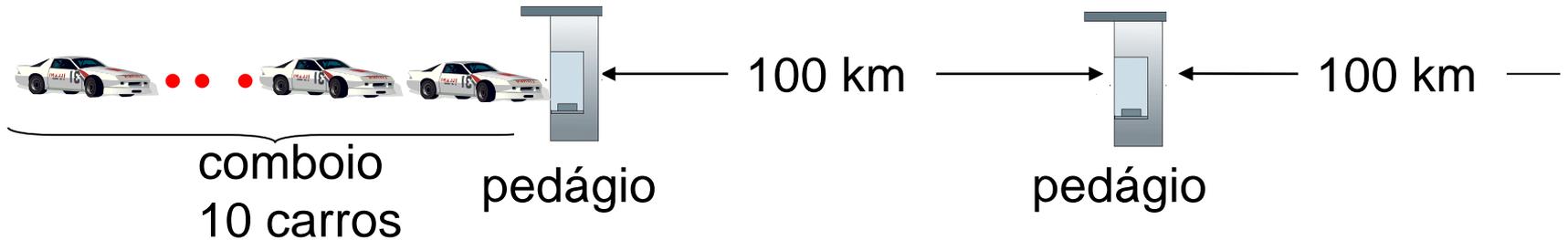
- de nano a milissegundos
- $L$ : comp. do pacote (bits)
- $R$ : capacidade do link (bits/s)
- $d_{\text{trans}} = L/R$

## $d_{\text{prop}}$ : atraso de propagação:

- milissegundos (WAN)
- $s$ : comprimento do enlace físico
- $v$ : vel. de propagação no meio ( $\sim 2 \times 10^8$  m/s)
- $d_{\text{prop}} = s/v$

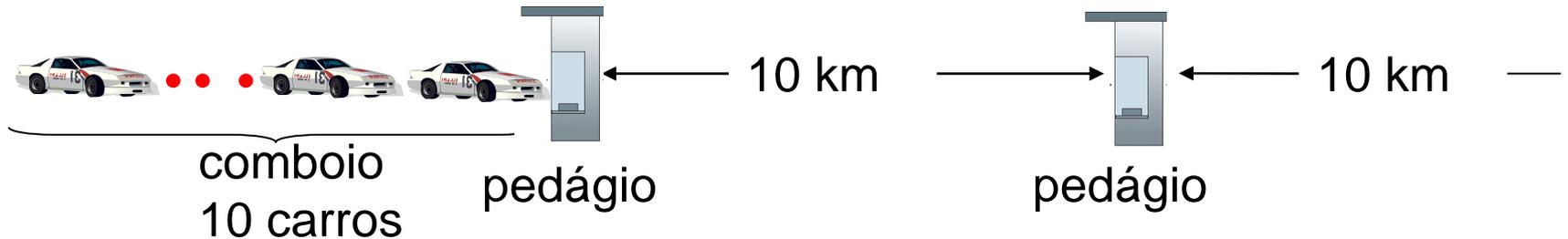
$d_{\text{trans}}$  e  $d_{\text{prop}}$   
muito diferentes!

# Analogia com comboio



- ❖ Carros “propagam-se” a **100 km/h**
- ❖ Pedágio leva **12 s** para atender carro (tempo de transmissão de bit)
- ❖ carro ~ bit; comboio ~ pacote
- ❖ **Q: Quanto tempo até que o comboio alinhe-se antes do 2o pedágio?**
- Tempo para “empurrar” o comboio inteiro pelo pedágio
$$d_{\text{trans}} = 12 \times 10 = 120 \text{ s}$$
- Tempo para último carro se propagar do 1o para 2o pedágio:
$$d_{\text{prop}} = 100 \text{ km} / (100 \text{ km/h}) = 1 \text{ h}$$
- **R: 62 min** ( $d_{\text{prop}}$  predomina)

# Analogia com comboio (mais)



- ❖ Suponha agora que:
  - distância seja de 10 km
  - pedágio leve 1 minuto para atender um carro
- ❖ **Q:** Carros chegarão ao 2o pedágio antes de todos os carros terem sido atendidos no 1o pedágio?
  - **R: Sim!** depois de 7 min, 1o carro chega ao 2o pedágio; 3 carros ainda estarão no 1o pedágio.
- ❖ **Q:** Quanto tempo até que o comboio alinhe-se antes do 2o pedágio?
  - **R: 16 min** ( $d_{\text{trans}}$  predomina)

# $d_{\text{trans}}$ e $d_{\text{prop}}$

- ❖ Enlace longos e taxas de transmissão ( $R$ ) altas –  $d_{\text{prop}}$  predomina
- ❖ Enlace curtos e taxas de transmissão ( $R$ ) baixas –  $d_{\text{trans}}$  predomina
- ❖ Em muitos casos, uma delas é bem mais importante que a outra

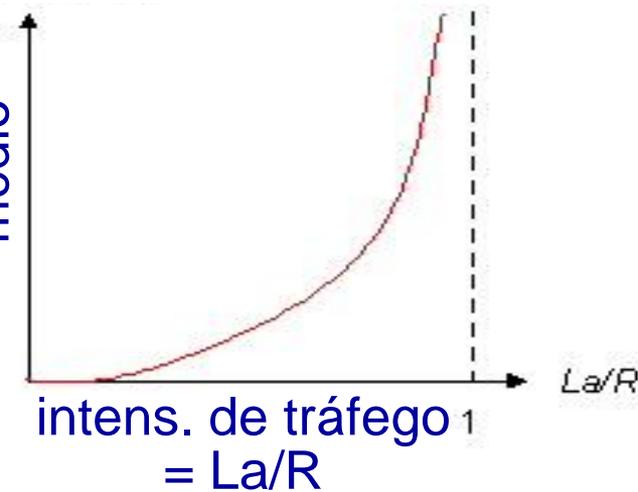
[Exercícios interativos do livro](#)

[Animação do livro](#)

# Atraso de fila $d_{\text{fila}}$

- ❖ *Componente mais complicado e interessante da latência*
- ❖ *Varia de pacote a pacote – trabalhar com médias!*
- ❖  $R$ : capacidade do enlace de saída (bits/s)
- ❖  $L$ : comprimento do pacote (bits/pacote)
- ❖  $\alpha$ : taxa média de chegada de pacotes (pacotes/s)
  
- ❖  $La/R = \textit{intensidade de tráfego}$
- ❖  $La/R \sim 0$ : atraso de fila médio pequeno
- ❖  $La/R \rightarrow 1$ : atraso de fila médio grande
- ❖  $La/R > 1$ : chegando mais pacotes do que pode ser atendido, atraso médio infinito!

atraso de fila  
médio



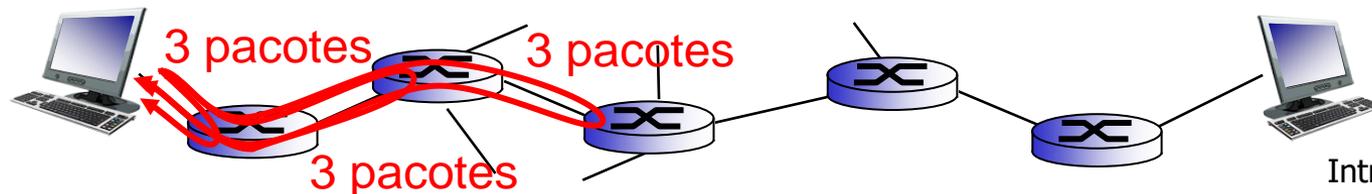
$La/R \sim 0$



$La/R \rightarrow 1$

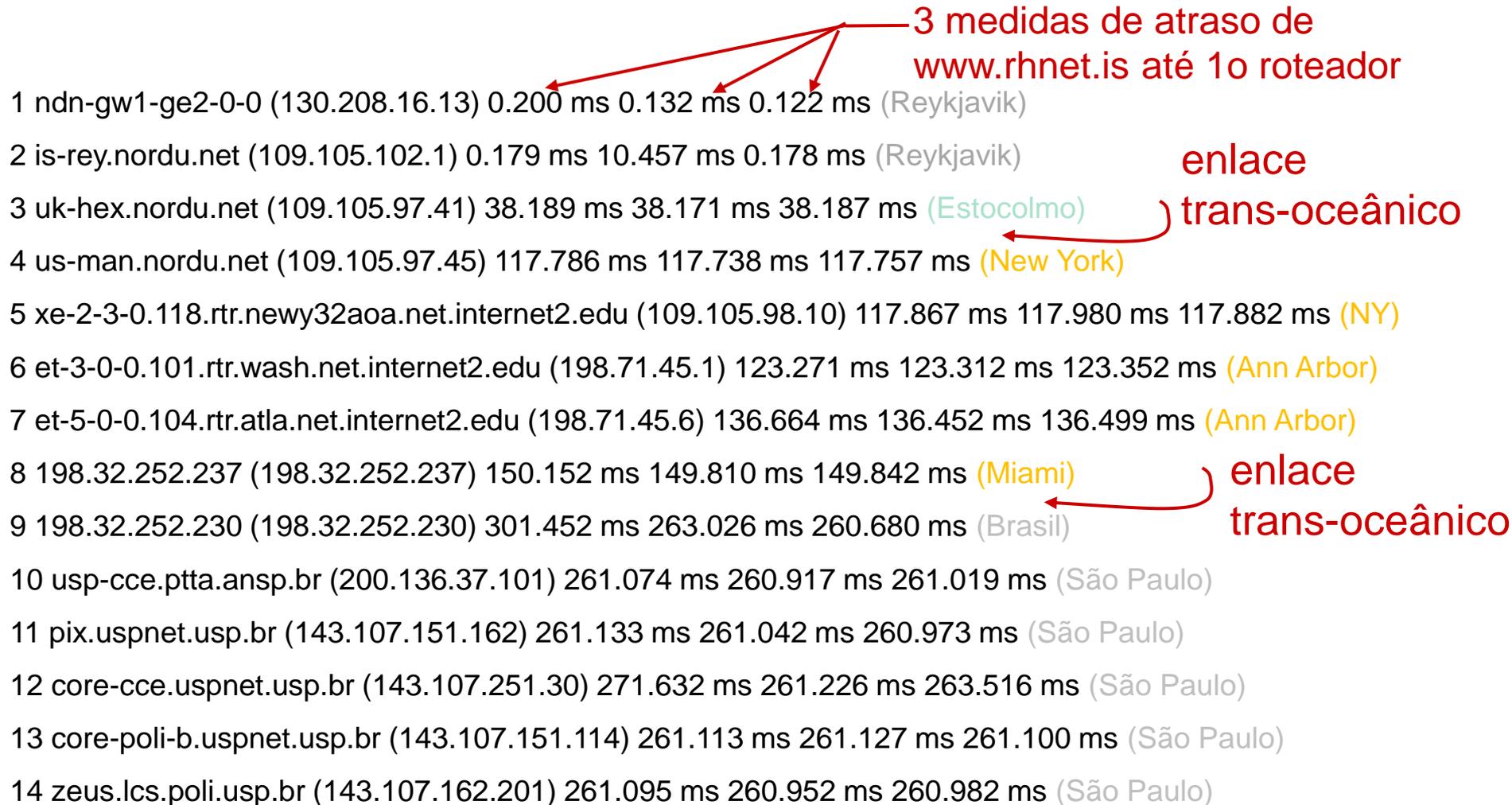
# Atrasos e rotas da Internet “real”

- ❖ Como são os atrasos e perdas reais da Internet?
- ❖ Programa `traceroute` (`tracert` no Windows e `traceroute` no macOS): fornece medidas de atraso da fonte aos roteadores ao longo do caminho até o destino ([RFC 1393](#))
- ❖ Para cada  $i$ :
  - Envia três pacotes que alcançam roteador  $i$  no caminho para o destino ( $TTL = i$ )
  - Roteador  $i$  retorna pacotes para a fonte
  - Mostra RTT para os três pacotes, caso tenha, sucesso



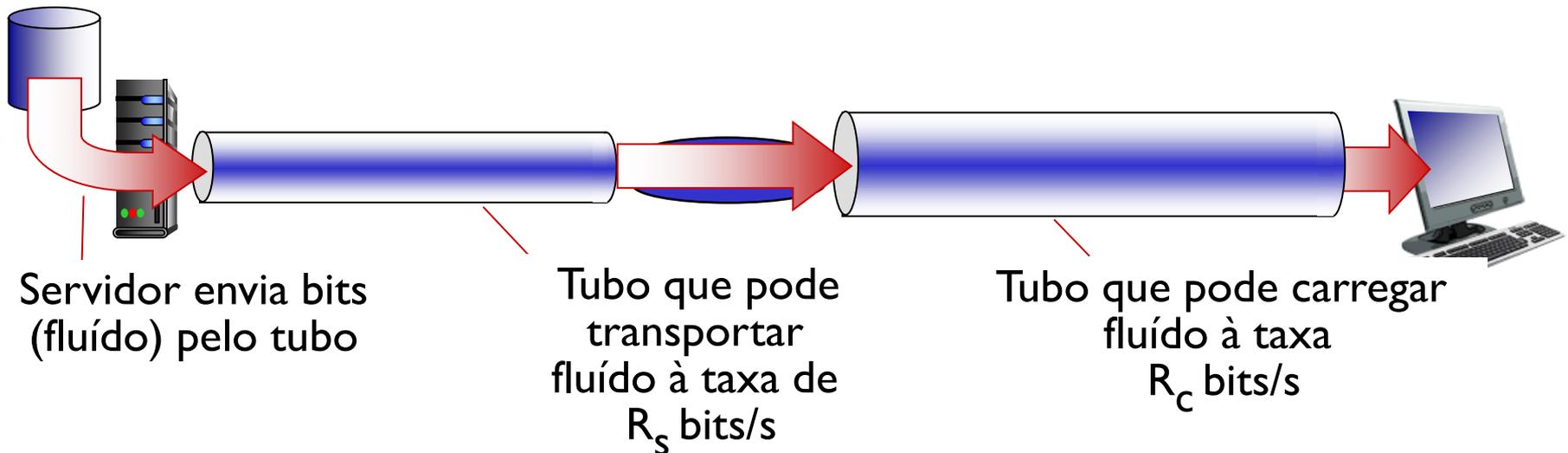
# Atrasos e rotas da Internet “real”

**traceroute:** www.rhnet.is (Islândia) até www.lcs.poli.usp.br



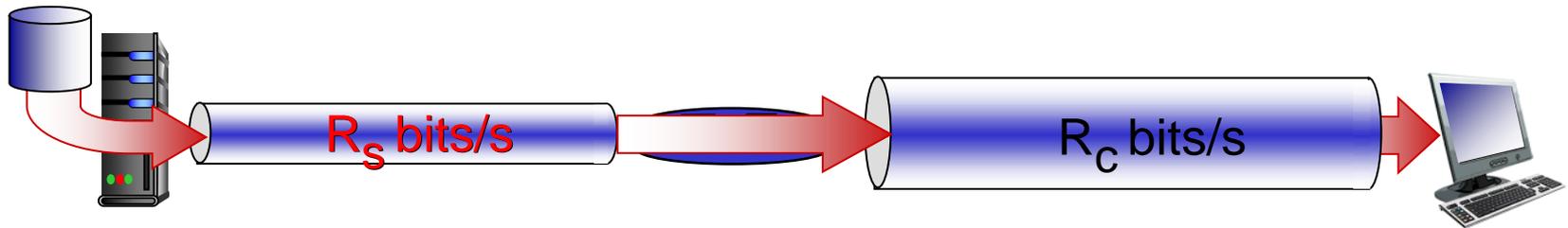
# Vazão (throughput)

- ❖ **Vazão:** taxa (bits/s) em que bits são transferidos entre fonte/destino
  - **Instantânea:** taxa num certo instante de tempo
  - **Média:** taxa média sobre um período de tempo

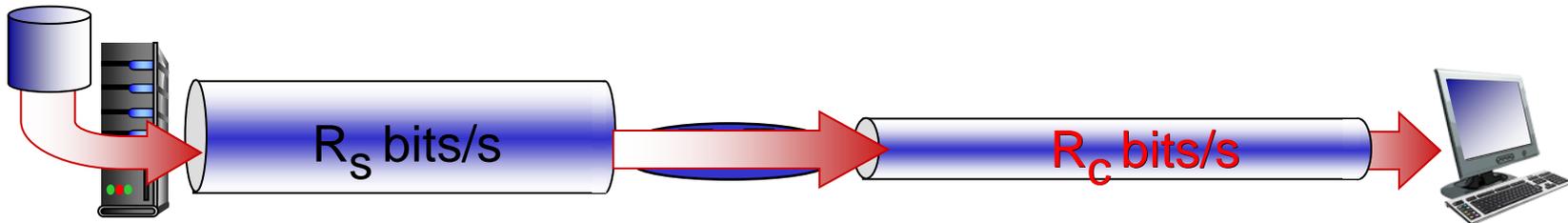


# Vazão (mais)

❖  $R_s < R_c$  Qual a vazão média fim a fim?



❖  $R_s > R_c$  Qual a vazão média fim a fim?

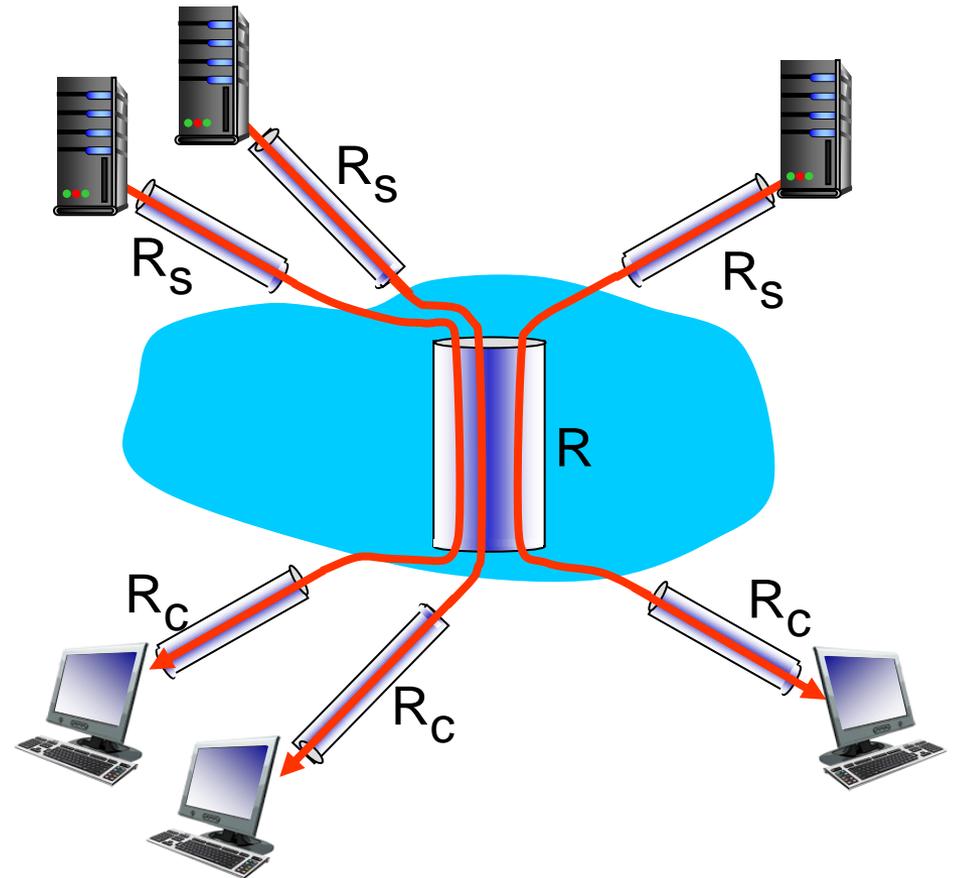


*Enlace de gargalo*

Enlace na rota fim-a-fim que restringe vazão fim-a-fim

# Vazão: cenário Internet

- ❖ Vazão por conexão:  
 $\min(R_c, R_s, R/I_0)$
- ❖ Na prática:  $R_c$  or  $R_s$  é geralmente o gargalo



10 conexões partilham conexões (de forma justa) no enlace gargalo  
*backbone*  $R$  bits/s