

Aula 15 - Roteadores: Arquitetura, *Buffers*, Políticas de Enfileiramento

Diego Passos

Universidade Federal Fluminense

Redes de Computadores I

Material adaptado a partir dos slides
originais de J.F Kurose and K.W. Ross.

Revisão da Última Aula...

- **Camada de rede:**

- Comunicação fim a fim **entre hosts**.
- Encapsula segmentos em **datagramas**.
- Roda em todos os nós.

- **Funções chave:**

- Encaminhamento: mover datagrama da entrada para saída.
- Roteamento: encontrar **rota** fim-a-fim.

- **Tabela de roteamento:**

- Montada pelo roteamento.
- Usada pelo encaminhamento.

- **Modelos de Serviço:**

- O que a rede promete.
- e.g., banda mínima, atraso máximo, entrega ordenada.
- Modelo da Internet: **melhor esforço**.

- **Redes de Circuitos Virtuais:**

- Serviço orientado a conexão.
- Pacotes carregam identificador do VC.
 - E não endereço de destino.
- Recursos muitas vezes reservados.
- Tabela de roteamento associa:
 - Interface, # do VC de entrada.
 - Interface, # do VC de saída.

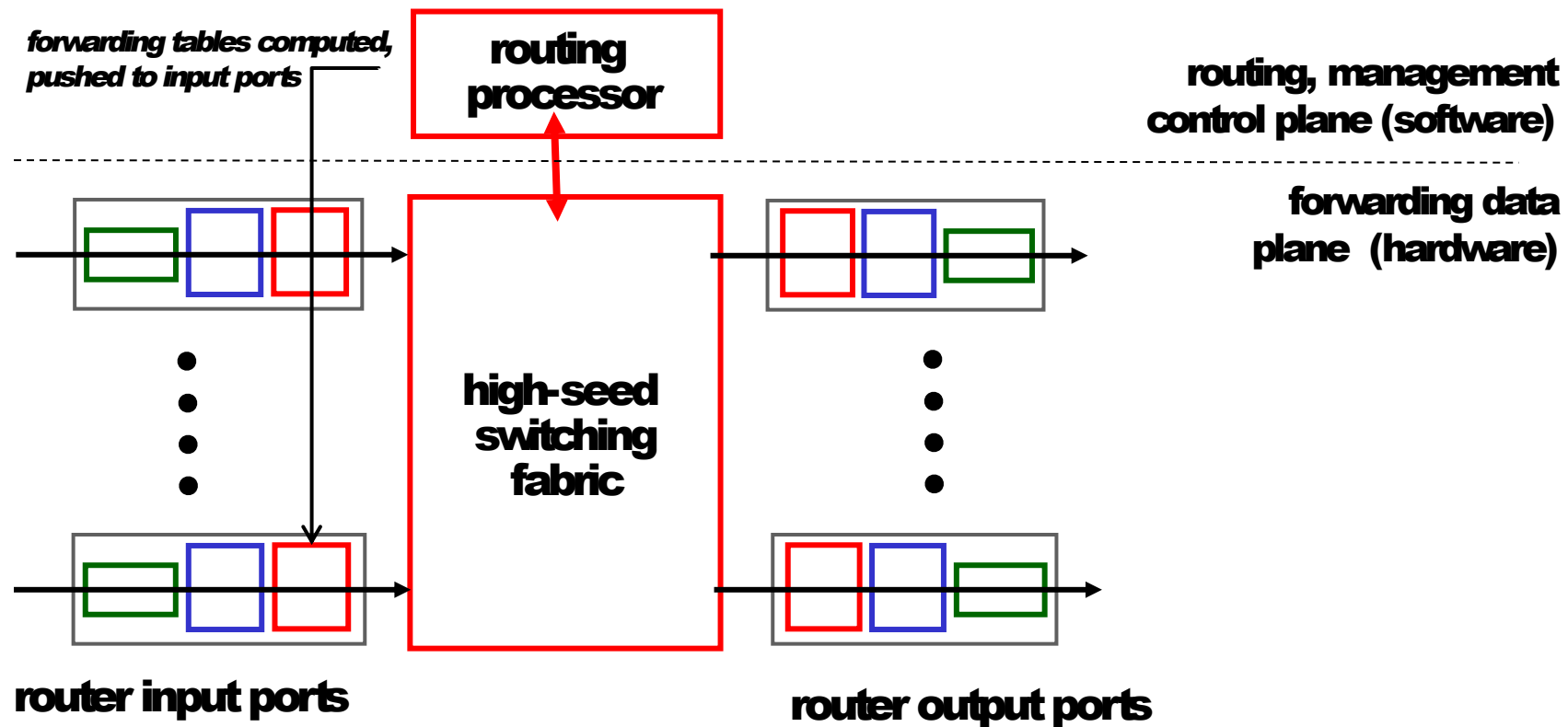
- **Redes de datagramas:**

- Paradigma usado na Internet.
- Encaminhamento baseado no **endereço de destino**.
- **Agregação de endereço**.
- **Casamento por prefixo mais longo**.

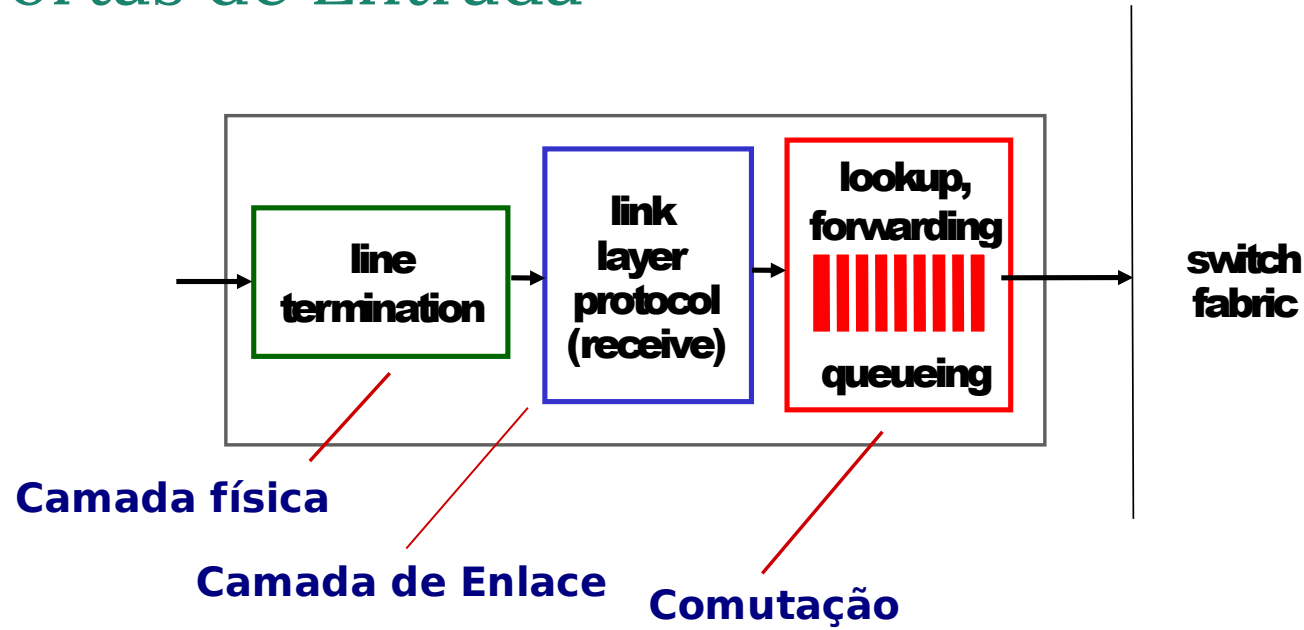
Roteadores: Arquiteturas

Aquiteturas de Roteadores: Visão Geral

- Duas funções chave em um roteador:
 - Execução de algoritmos/protocolos de roteamento (RIP, OSPF, BGP).
 - **Encaminhamento de datagramas de enlaces de entrada para enlaces de saída.**



Funções das Portas de Entrada



- **Camada física:**

- Recepção no nível dos bits.

- **Camada de enlace:**

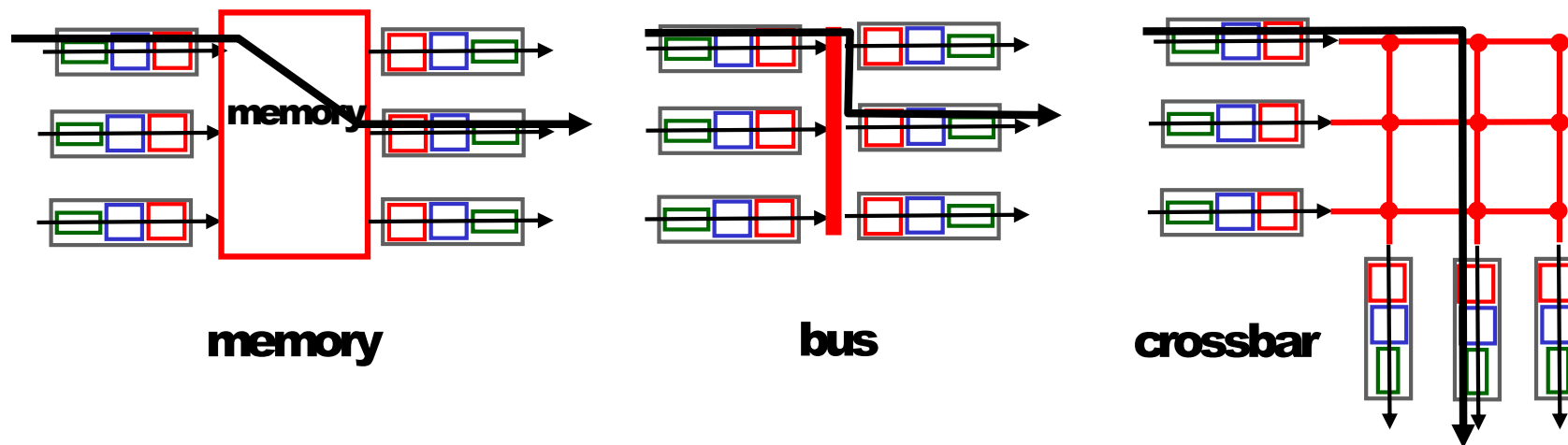
- e.g., Ethernet.
- Vide capítulo 5 (Redes II).

- **Comutação:**

- Dado destino do datagrama, procurar porta de saída usando tabela de roteamento em memória.
- Objetivo: completar processamento da porta de entrada na “velocidade de linha”.
- Enfileiramento: se datagramas chegam mais rápido que taxa de encaminhamento para dentro da malha de comutação.

Malhas de Comutação

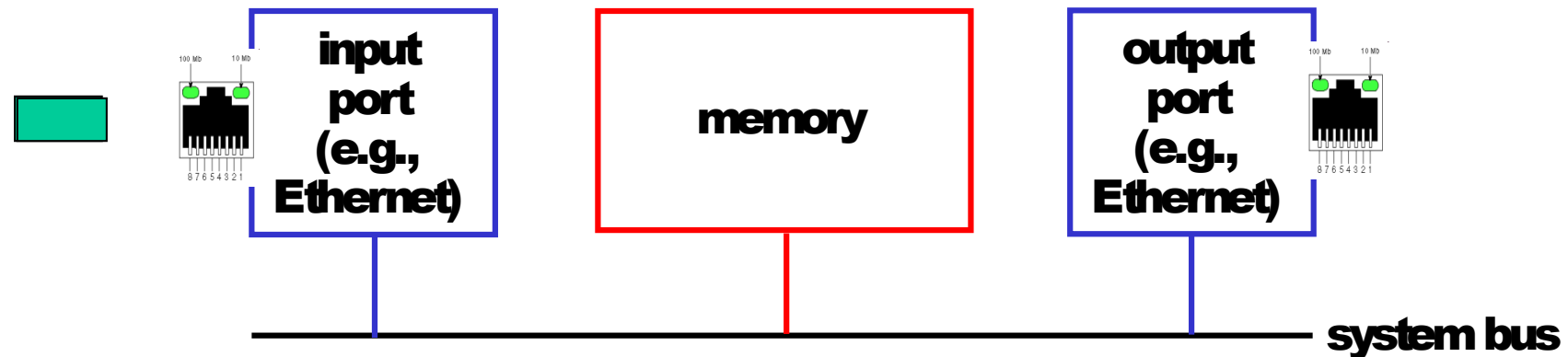
- Transferem pacotes do *buffer* da porta de entrada para o *buffer* da porta de saída apropriada.
- **Taxa de comutação:** taxa na qual pacotes podem ser transferidos das entradas para as saídas.
 - Comumente medida como um múltiplo da velocidade da linha das portas de entrada/saída.
 - N entradas: taxa de comutação desejada de N vezes a velocidade da linha.
- Três tipos básicos de malha de comutação:



Comutação Através da Memória

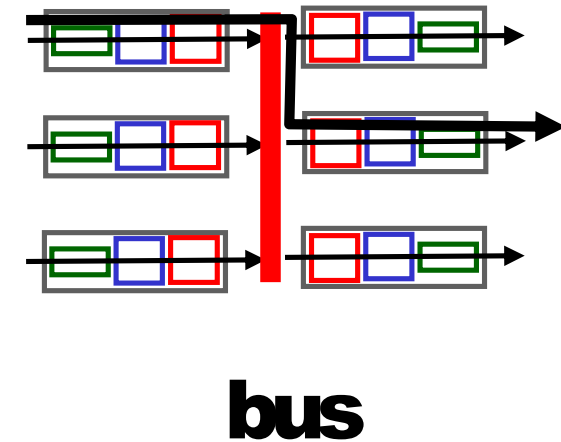
- **Primeira geração de roteadores:**

- Computadores comuns com comutação feita diretamente pelo processador.
- Pacotes copiados para a memória principal do sistema.
- Taxa limitada pela vazão da memória.
 - Dois usos do barramento por datagrama.



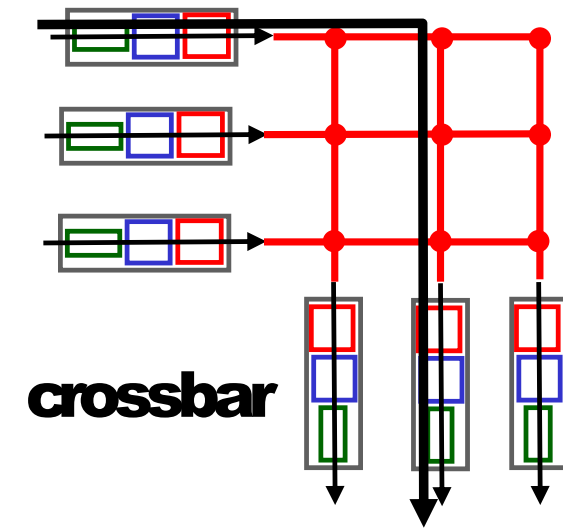
Comutação Através de um Barramento

- Datagrama passa da porta de entrada para a porta de saída através de um barramento compartilhado.
- **Contenção no barramento:** taxa de comutação limitada pela banda do barramento.
- Barramento de 32 Gb/s, Cisto 5600: taxa suficiente para roteadores de acesso e empresariais.

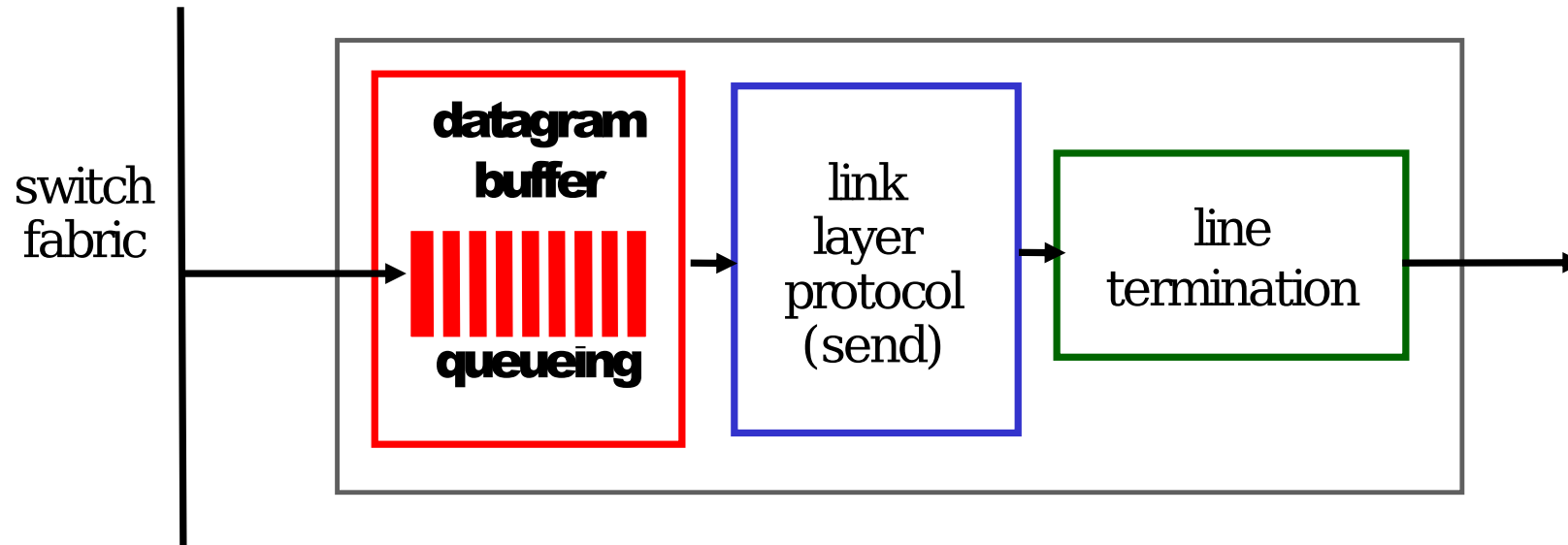


Comutação Através de uma Rede de Interconexão

- Supera limitações de banda do barramento.
- Rede banyan, *crossbar*, outras redes de interconexão inicialmente desenvolvidas para conectar processadores em computadores multiprocessados.
- Projeto avançado: fragmentam datagramas em células de comprimento fixo comutadas pela malha.
- Cisco 12000: comuta 60 Gb/s através da rede de interconexão.

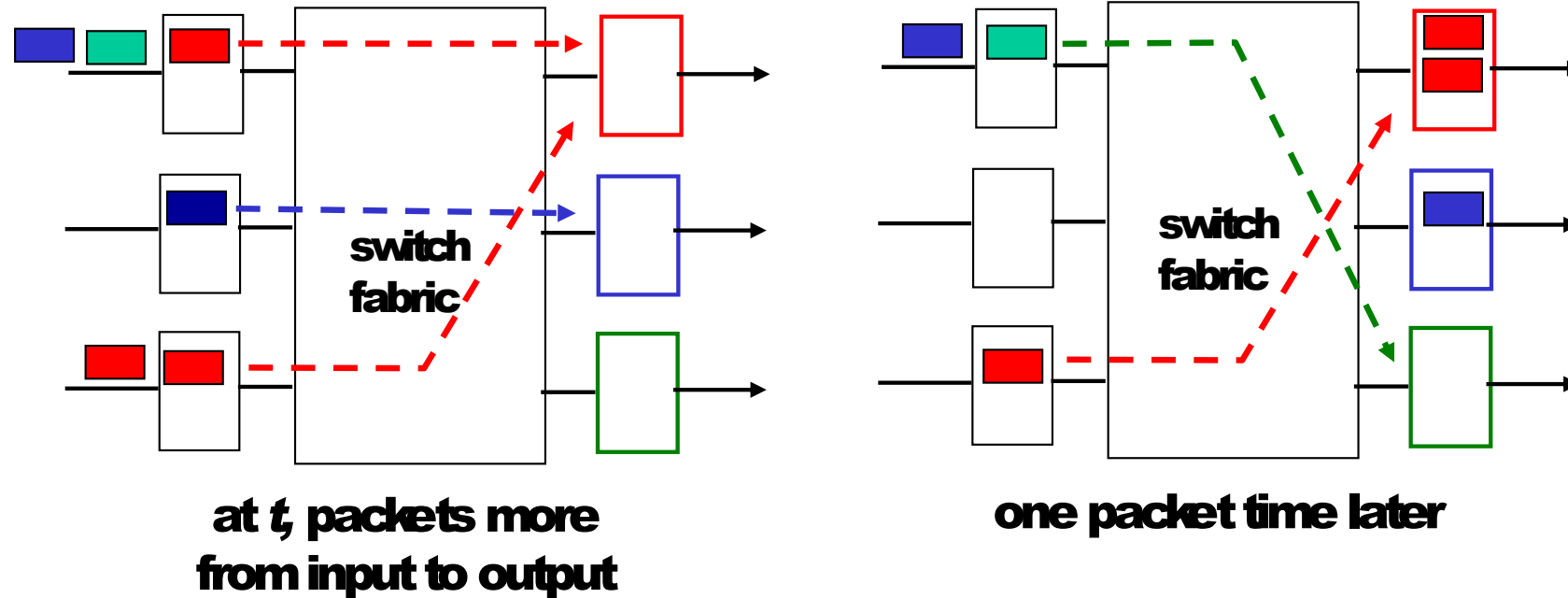


Portas de Saída



- **Enfileiramento** necessário devido à taxa mais alta da malha de comutação.
 - Datagramas podem ser perdidos devido a congestionamento, falta de *buffer*.
- **Escalonamento** de datagramas.
 - Prioridades: quem recebe melhor desempenho? Neutralidade da rede?

Enfileiramento na Porta de Saída



- Ocorre quando a taxa de chegada da malha de comutação excede velocidade da linha da porta de saída.
- **Enfileiramento (atraso) e perdas devido a overflow do buffer da porta de saída!**

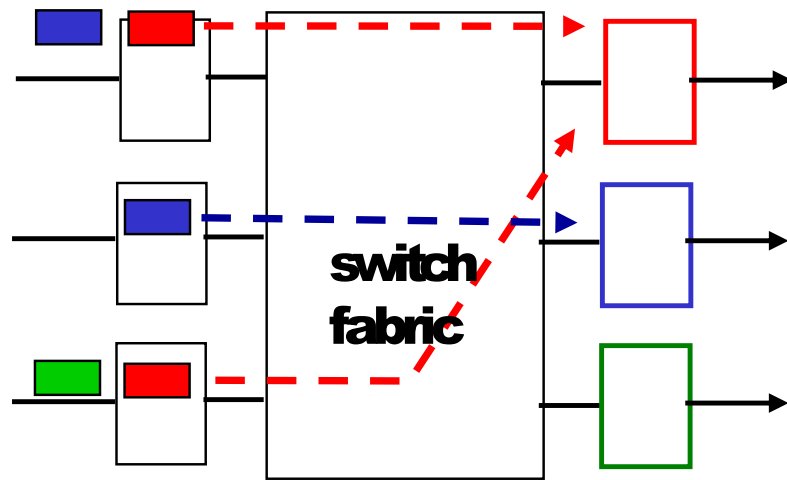
Quanto *Buffer* Alocar?

- Recomendação da RFC 3439: *buffer* deve ser capaz de armazenar aproximadamente 250 ms de dados.
 - 250 ms: RTT “típico” da Internet.
 - Tamanho do *buffer*: $RTT \times C$.
 - Onde C é a capacidade do enlace.
 - e.g., $C = 10 \text{ Gb/s}$, $RTT = 250 \text{ ms}$, 2,5 Gb de *buffer*.
 - Recomendação recente: com N fluxos, tamanho do *buffer* igual a:

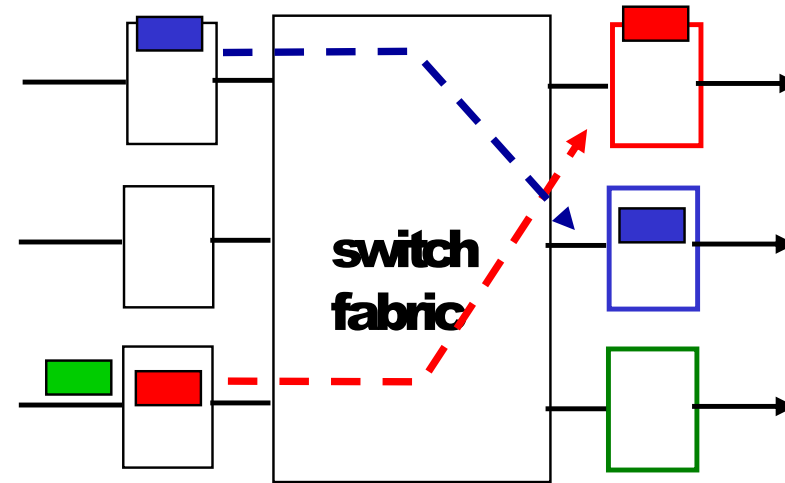
$$\frac{RTT \times C}{\sqrt{N}}$$

Enfileiramento na Porta de Entrada

- Malha de comutação mais lenta que portas de entrada combinadas \Rightarrow enfileiramento pode ocorrer nas filas das portas de entrada.
 - **Enfileiramento (atraso) e perdas devido a overflow do buffer da porta de saída!**
- **Bloqueio de cabeça de linha (Head-of-line Blocking, ou HOL):** datagrama da frente da fila não permite que outros enfileirados sejam comutados.



output port contention:
only one red datagram can be
transferred.
lower red packet is blocked



one packet time later:
green packet
experiences HOL
blocking

Roteadores: Políticas de Enfileiramento

Políticas de Enfileiramento: Escalonamento e Descarte

- Também chamadas de **disciplinas**.
- Duas decisões importantes:
 - **Escalonamento**: em que ordem transmitir os pacotes.
 - *e.g.*, há pacotes mais importantes que outros?
 - *e.g.*, um pacote deve poder “passar a frente” dos demais?
 - Prioridades?
 - **Descarte**: quem (e quando) descartar.
 - Descartar o último pacote?
 - Descartar o primeiro?
 - Descartar aleatoriamente?
 - Descartar apenas quando fila está **completamente cheia**?

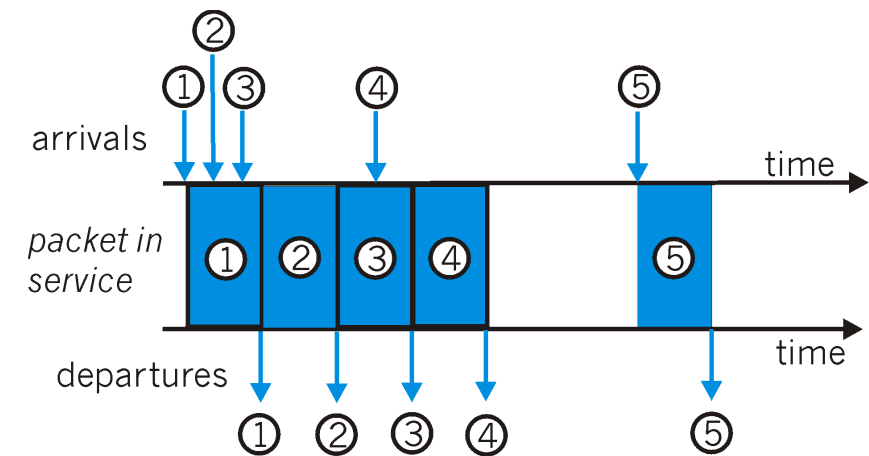
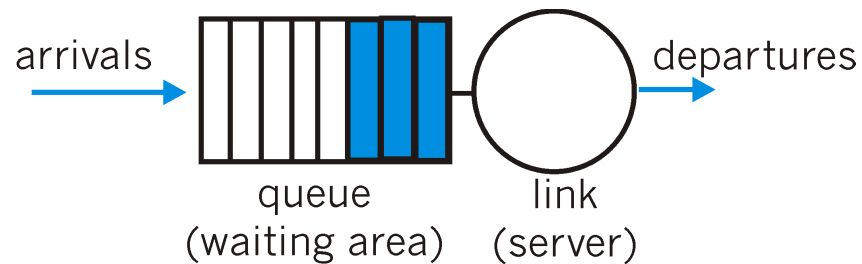
Políticas de Enfileiramento: Por Quê?

- **Solução óbvia:** respeitar ordem de chegada.
 - Transmitir pacotes na ordem em que são recebidos.
 - Descartar novos pacotes quando não há espaço em *buffer*.
- **Nem sempre é o ideal:**
 - Podemos querer **dar importância maior** a certos pacotes/fluxos.
 - e.g., datagrama de um *download* pode esperar, de uma chamada VoIP não.
 - **Controle de congestionamento do TCP pode ser afetado.**
 - Infere congestionamento por perdas.
 - É possível **avisar mais rápido e a todos os fluxos**.

Políticas de Escalonamento: FIFO

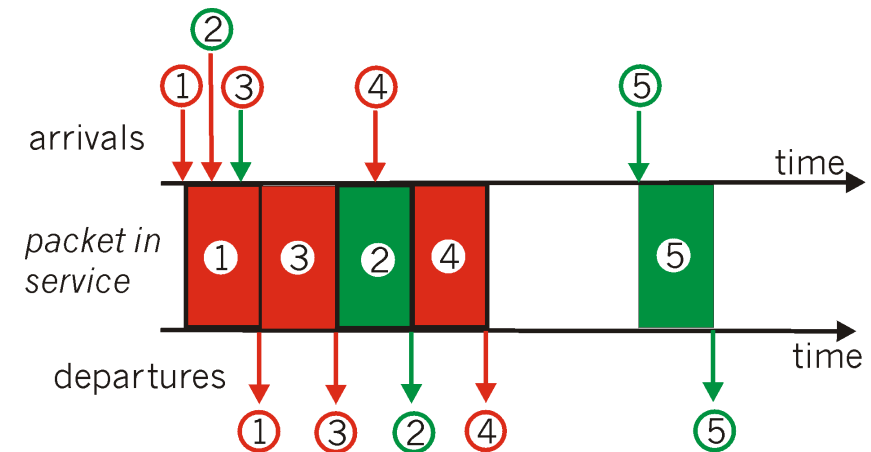
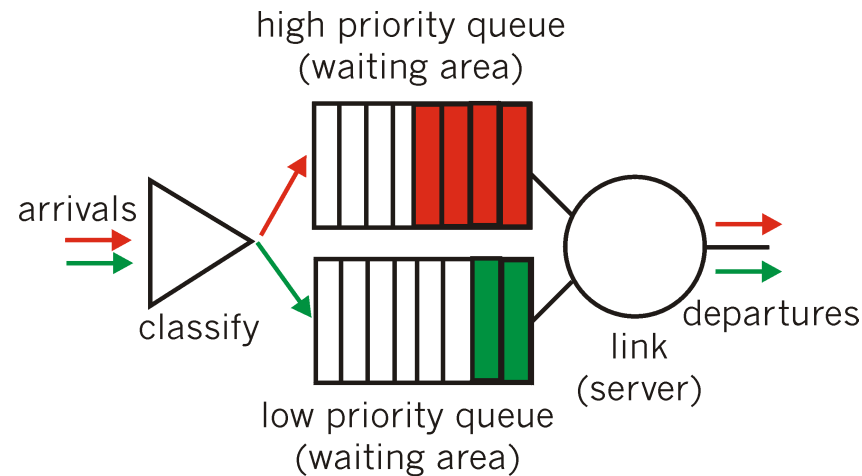
- **First-In, First-Out.**

- Também chamada de FCFS (*First-Come, First-Served*).
- “Solução óbvia”: pacotes servidos na ordem em que chegam.
- Não há prioridades.
 - Nenhum pacote “fura fila”.
- Simples, popular.



Políticas de Escalonamento: *Priority Queuing* (I)

- Divide pacotes em **classes**.
 - Uma fila separada para cada classe.
- Cada classe possui uma **prioridade diferente**.
- Pacotes de classes prioritárias **sempre** são transmitidos antes.
 - i.e., fila de uma classe só é atendida se filas de **todas as classes mais prioritárias estão vazias**.



Políticas de Escalonamento: *Priority Queuing* (II)

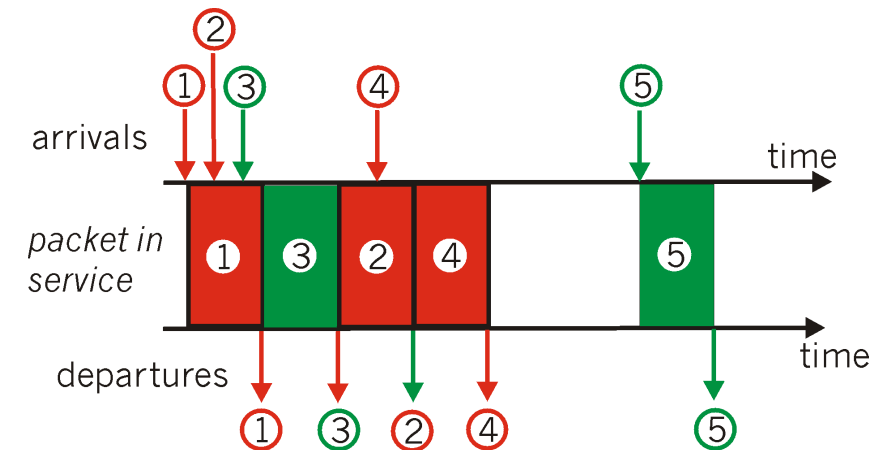
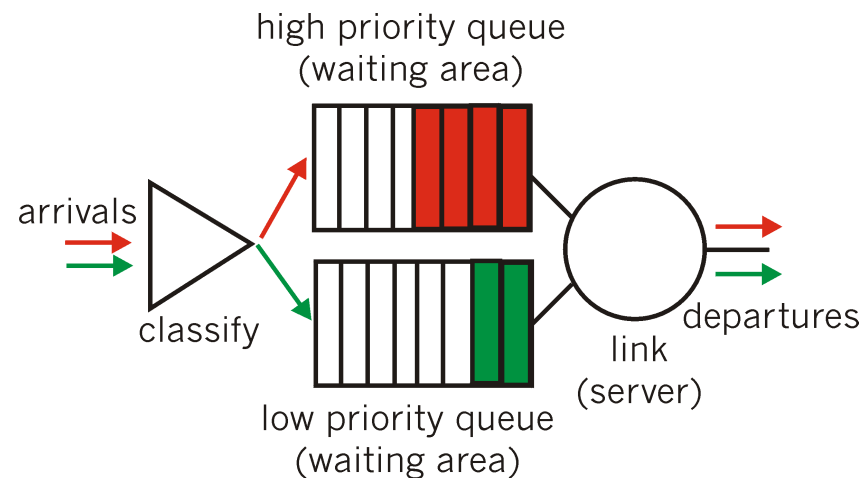
- Classe do pacote é definida de acordo com sua importância.
 - Termo relativo, definido pelo administrador do roteador.
- **Exemplo:**
 - Pacotes VoIP ficam na classe de mais alta prioridade.
 - Demais pacotes ficam na classe de prioridade mais baixa.
- **Problemas?**

Políticas de Escalonamento: *Priority Queuing* (III)

- Classe do pacote é definida de acordo com sua importância.
 - Termo relativo, definido pelo administrador do roteador.
- **Exemplo:**
 - Pacotes VoIP ficam na classe de mais alta prioridade.
 - Demais pacotes ficam na classe de prioridade mais baixa.
- **Problemas?**
 - Pode causar **esfomeação** (starvation).

Políticas de Escalonamento: Round-Robin

- Assim como a *Priority Queuing*, pacotes são divididos em classes.
- **Mas objetivo é diferente:**
 - Garantir uma divisão **justa** do uso dos recursos.
 - Todas as classes recebem **oportunidades iguais**.
- Filas das classes são servidas através de um *round-robim*, um pacote por vez.

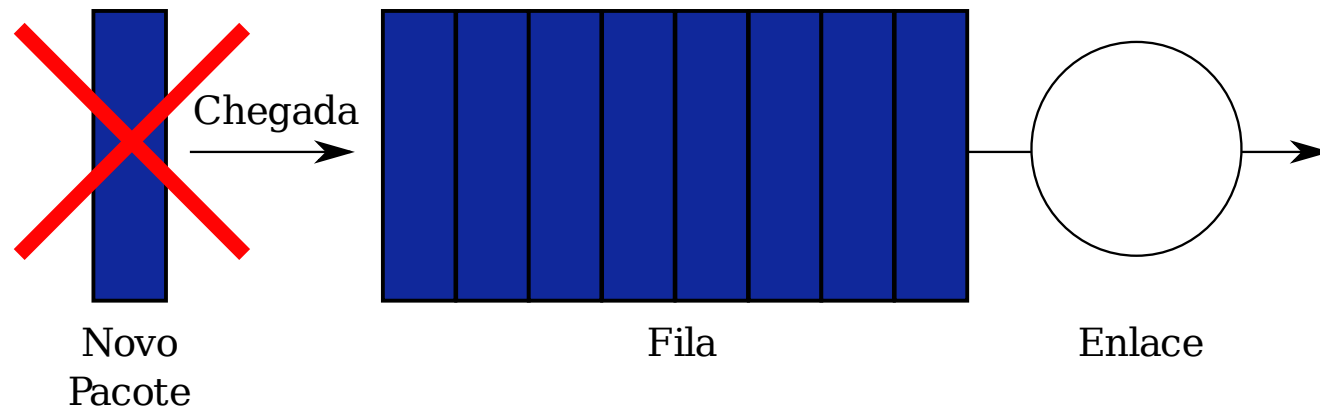


Políticas de Escalonamento: Round-Robin e Prioridades?

- Round-Robin não incorre em esfomeação.
- **Mas também não atribui prioridades diferentes!**
- É possível **combinar** as duas abordagens?
 - i.e., atribuir prioridades maiores a certas classes, **mas garantindo que todas as classes receberão** um certo grau de oportunidade?
- **Sim!**
 - Alcançado pela política *Weighted Fair Queueing*.
 - Mais detalhes em Redes II.

Políticas de Descarte: *Drop-tail* (I)

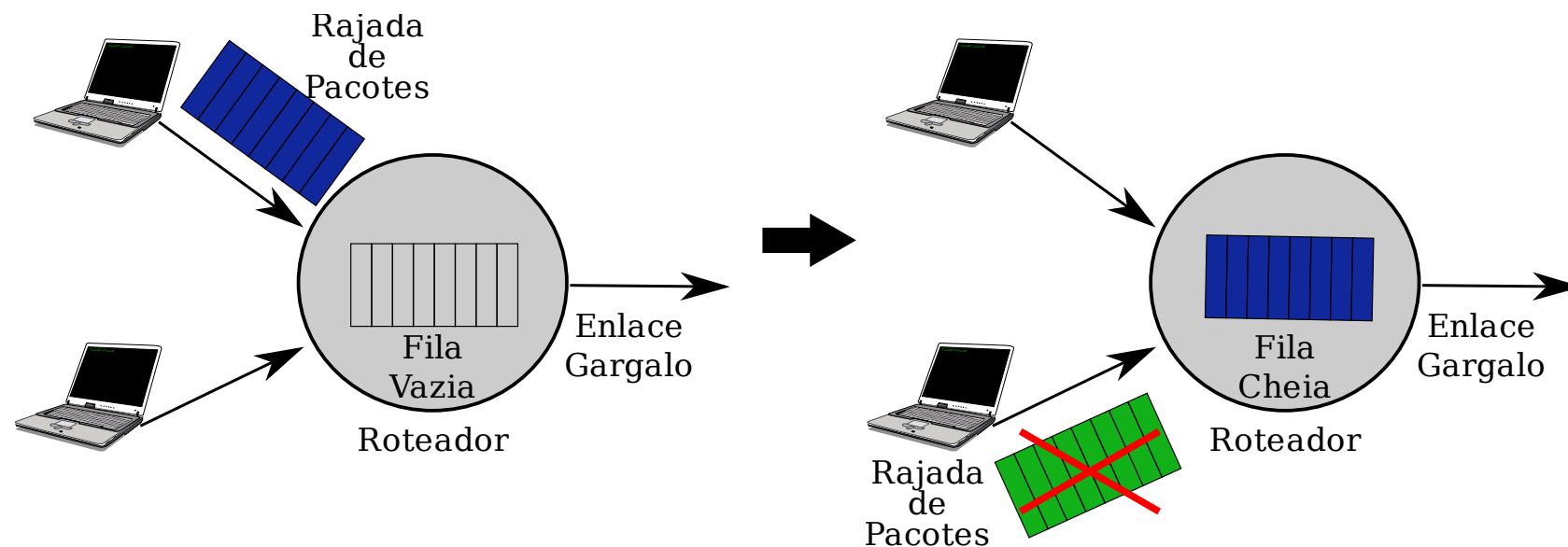
- Está para o descarte como a FIFO está para o escalonamento.
 - Ideia simples, imediata.
- Funcionamento:
 - Quando fila está cheia e novo pacote chega, **novo pacote sempre é descartado.**



- Política amplamente implementada e adotada.
 - Mas não necessariamente a melhor em todos os casos.

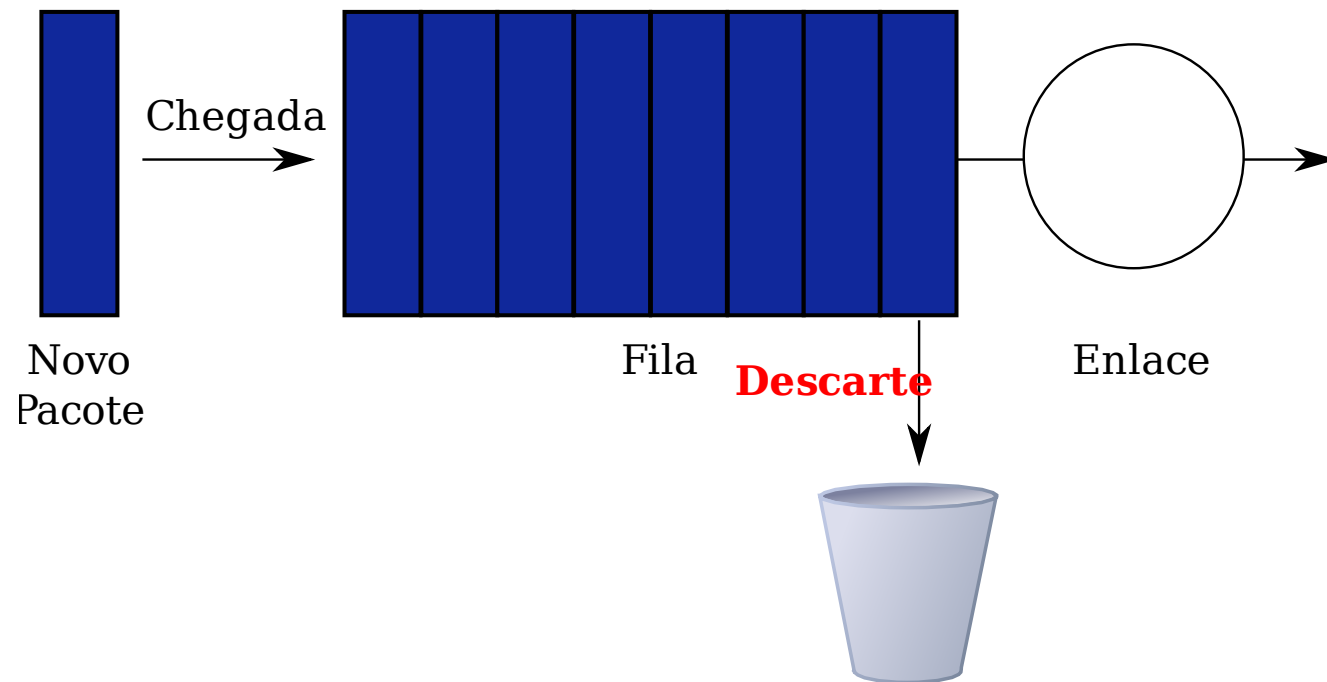
Políticas de Descarte: *Drop-tail* (II)

- Potencial problema: sincronização de fluxos.
 - Suponha dois *hosts* compartilhando um único enlace de saída de um roteador.
 - Assuma que ambos geram **tráfego em rajada**.
 - i.e., quando *host* transmite, envia **vários pacotes em sequência**.
 - Em seguida, passa algum sem geração de tráfego.
 - Dependendo da ordem dos envios, um dos *hosts* pode ser prejudicado.
 - i.e., seus pacotes comumente encontram a fila cheia e são descartados.



Políticas de Descarte: *Drop-head* (I)

- “Contrário” da *drop-tail*.
- Se novo pacote chega e fila está cheia, **primeiro pacote da fila é sempre descartado**.
 - *i.e.*, o pacote a mais tempo na fila.



- **Pergunta:** resolve o problema da injustiça na *Drop-tail*?

Políticas de Descarte: *Drop-head* (II)

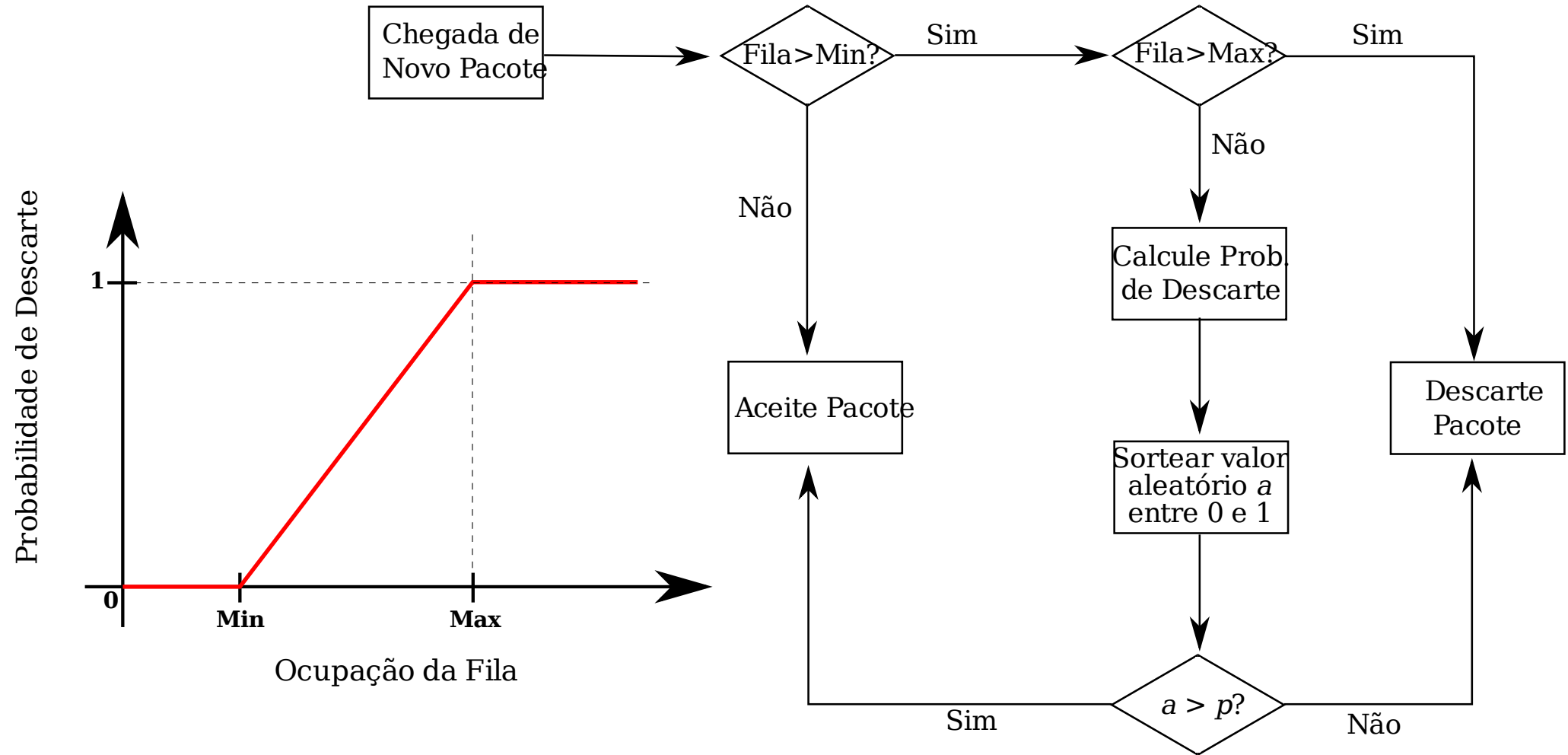
- Qual é a vantagem de se descartar o primeiro pacote da fila?
- Não é injusto descartar o pacote que espera há mais tempo?
- Talvez, mas se este pacote for um segmento TCP, há uma grande vantagem:
 - TCP **precisa ser avisado do congestionamento o mais rápido possível.**
 - Descartar primeiro pacote provavelmente gerará mais rapidamente:
 - Estouro do temporizador do TCP.
 - Ou ACKs duplicados.
 - **Resultado:** TCP reage mais rapidamente reduzindo janela.

Políticas de Descarte: RED (I)

- **Random Early Detection.**
- Começa (possivelmente) a descartar pacotes **antes que a fila esteja completamente cheia.**
- Funcionamento (Simplificado):
 - **Mínimo:** menor ocupação da fila para a qual pacotes podem ser descartados.
 - **Máximo:** tamanho máximo da fila.
 - Quando novo pacote chega:
 - Se fila está menor que Mínimo, nunca descarte.
 - Se fila está igual a Máximo, sempre descarte.
 - Caso contrário, **descarte último pacote com probabilidade p proporcional ao tamanho atual da fila.**

Políticas de Descarte: RED (II)

- Esquema simplificado de funcionamento do RED:



Políticas de Descarte: RED (III)

- Mar **por que descartar** pacotes se o *buffer* ainda não está totalmente cheio?
- Lembre-se:
 - Congestionamento se manifesta como um **aumento no nível de enfileiramento**.
 - *Overflow* é apenas uma **consequência** após um **período estendido de congestionamento**.
 - TCP identifica congestionamento por perdas.
- Ao descartar pacotes quando *buffer* está parcialmente cheio, **sinalizamos congestionamento antecipadamente**.
- **Efeito colateral:** evita problema da sincronização.
 - Descartes tendem a ser mais bem distribuídos entre fluxos.

Resumo da Aula...

- **Roteadores: arquitetura.**

- Plano de controle: protocolos de roteamento.
- Plano de encaminhamento: portas, malha de comutação.

- **Portas de entrada:**

- Terminação do enlace físico.
- Camada de enlace.
- Decisões de comutação, **fila**.
- *Head-of-line Blocking*.

- **Malha de comutação:**

- Transporta pacotes de entradas para saída.
- **Memória, barramento, rede de interconexão.**
- Taxa de comutação.

- **Portas de saída:**

- **Fila, atraso, perdas.**

- **Buffers: tamanho.**

- Objetivo: absorver variações temporárias.
- *Buffer* excessivo → atrasos altos, mascara congestionamento.

- **Políticas de enfileiramento:**

- Descarte e escalonamento.
- Impacto em QoS, no TCP.

- **Políticas de Escalonamento:**

- FIFO, Priority Scheduling, Round-Robin.
- Esfomeação vs. Justiça vs. Simplicidade.

- **Políticas de Descarte:**

- *Drop-tail*, *Drop-head*, RED.
- Sincronização.
- Detecção adiantada de congestionamento.

Próxima Aula...

- Discutiremos o principal protocolo da Internet: o IP.
 - Convenções.
 - Formato do datagrama.
 - Fragmentação.
 - Endereçamento.
 - Sub-redes.
- Também discutiremos outro protocolo importante: o DHCP.
- **Aula muito importante!**