

# Aula 3 - Comutação de pacotes, Multiplexação, 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.

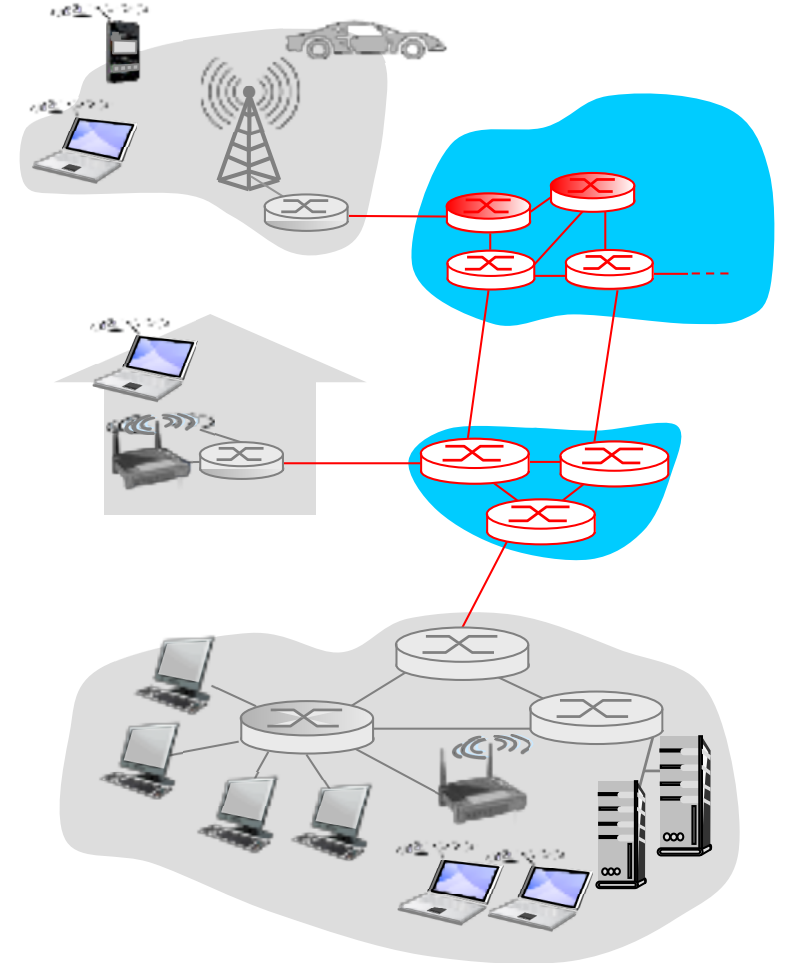
# Revisão da Última Aula...

- Arquitetura da Internet pode ser dividida:
  - **Borda**: *hosts*, redes de acesso.
  - **Núcleo**: roteadores, ISPs interconectados.
- Redes de acesso:
  - Múltiplas tecnologias: DSL, HFC, Ethernet, Wi-Fi, ...
  - Algumas vezes combinadas.
  - Se conectam ao núcleo através de um **roteador de borda**.
- Roteamento e encaminhamento:
  - Responsabilidades do núcleo.
  - Tarefas ligadas ao repasse dos pacotes entre roteadores, *hosts*.
- Princípios e filosofias:
  - Rede heterogênea.
    - Necessita **conjunto de protocolos comuns**.
    - **Não se deve assumir muito** sobre a rede.
  - Argumento fim-a-fim.
    - Funcionalidades fim-a-fim dependem das pontas.
    - Logo, tentar mantê-las nas pontas.
    - Corolário: **inteligência nas bordas, núcleo simples**.
  - Princípio KISS.
    - Evitar complexidades desnecessárias.
    - Sistemas mais simples são melhores que equivalentes complexos.

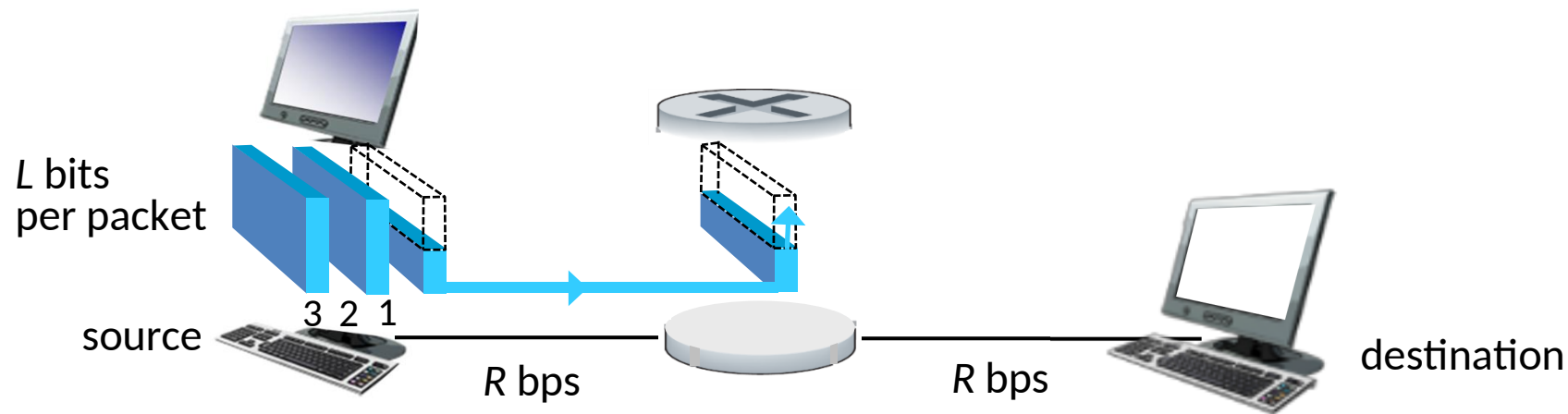
# Comutação de Pacotes

# Comutação de Pacotes: Conceitos Básicos

- Hosts quebram dados em **pacotes**.
  - Pacotes são a unidade fundamental de transferência.
- Pacotes são **encaminhados** por comutadores.
  - Passam por sequência de enlaces, formando um caminho ou **rota**.
- Sempre que pacote é transmitido por enlace, **transmissão utiliza toda a capacidade do enlace**.



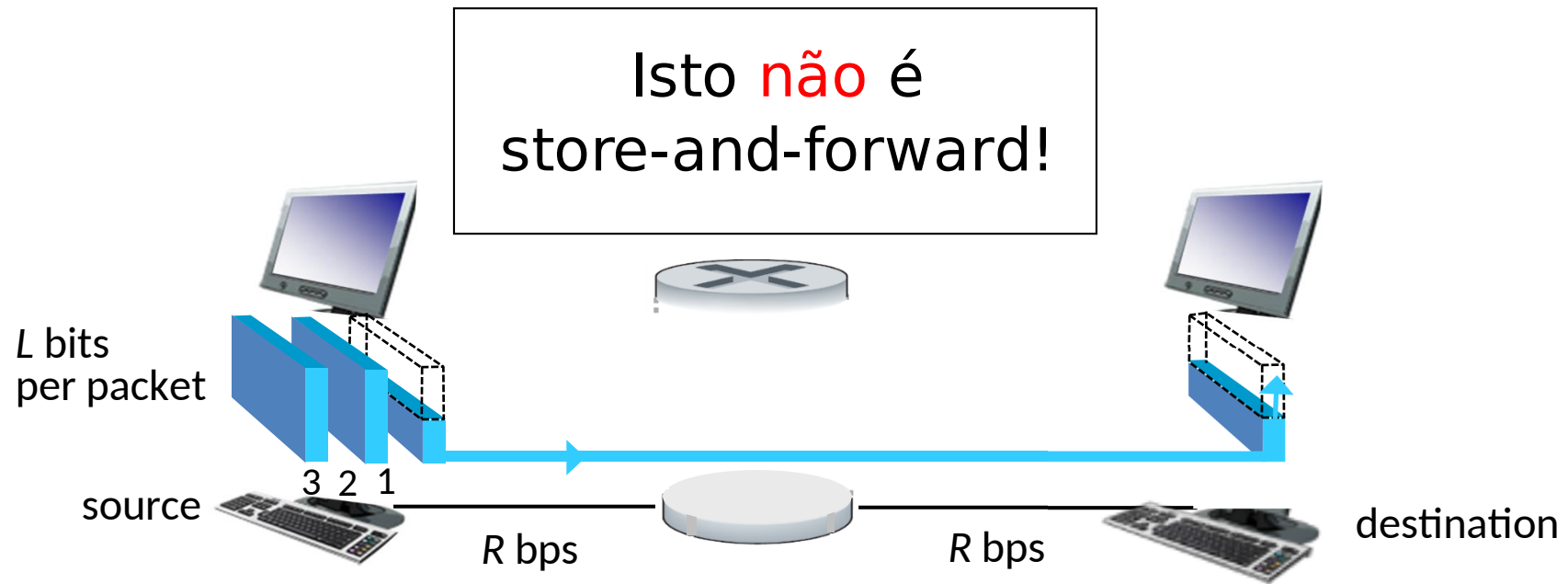
# Comutação de Pacotes: Store-and-Forward (I)



- São necessários  $\frac{L}{R}$  segundos para transmitir (“colocar no enlace”) pacote de  $L$  bits em enlace de capacidade  $R$ .
- **Store-and-Forward:** pacote precisa ser recebido por **completo** antes transmissão começar no próximo enlace.
  - Comutador recebe pacote **inteiro**, armazena em memória, processa, ...
- **Consequência:** tempo total até o destinatário é  $\frac{2L}{R}$ .
  - Ignorando tempo de propagação.
- Exemplo numérico:
  - $L = 4 \text{ MB}$ .
  - $R = 2 \text{ Mb/s}$ .
  - Tempo de transmissão por um enlace: 16 s.
  - Para transmissão **fim-a-fim**: 32 s.

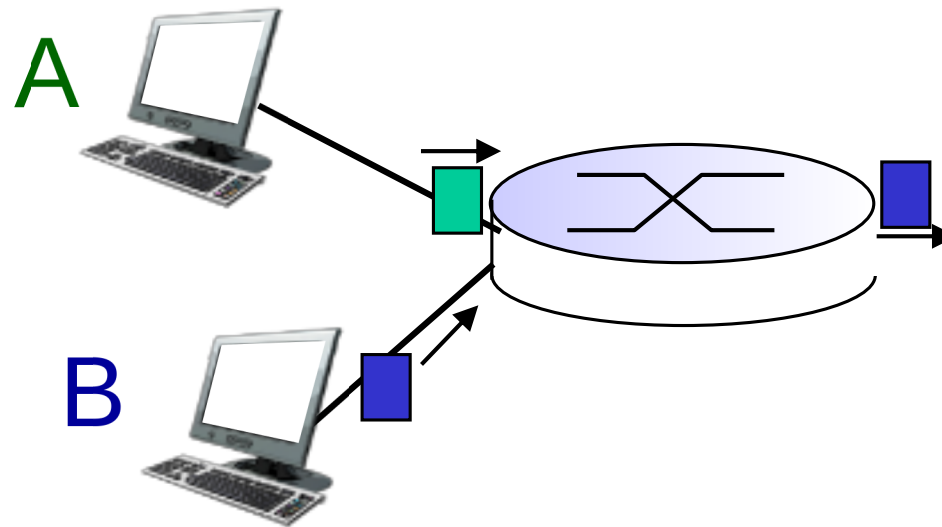
# Comutação de Pacotes: *Store-and-Forward* (II)

- Contra-exemplo:

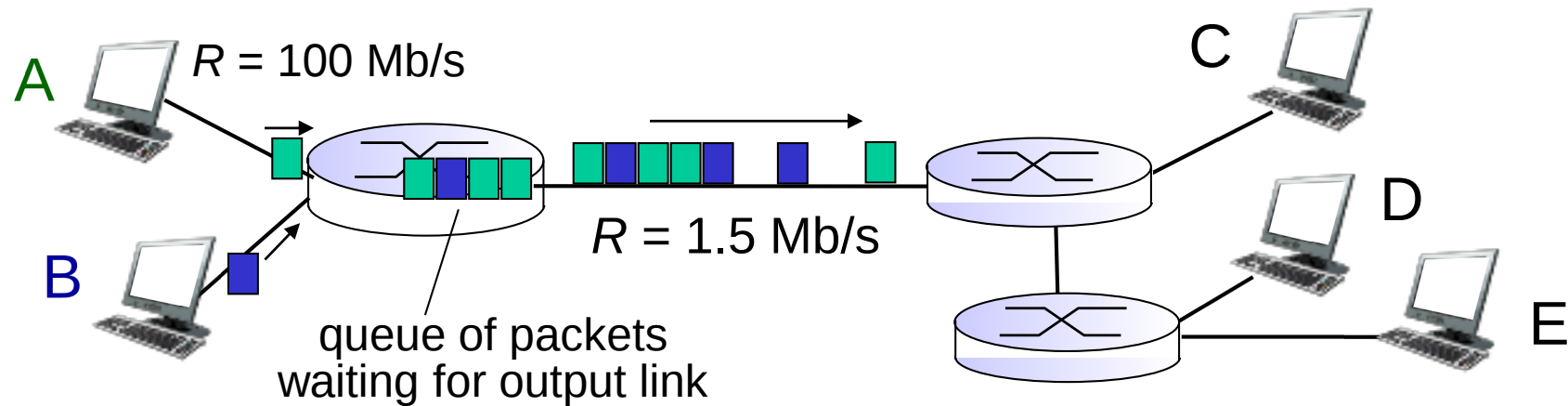


# Comutação de Pacotes: *Store-and-Forward* (III)

- Por que comutador não pode transmitir logo que recebe primeiros bits?
- Por que esperar que o pacote chegue completamente?
- Alguns motivos:
  - Processamentos são necessários.
    - Encaminhamento, verificação de integridade, ...
  - Enlace de saída pode estar ocupado com outro pacote.
    - Pacote que chega pode ter que **esperar**.



# Comutação de Pacotes: Enfileiramento e Descartes

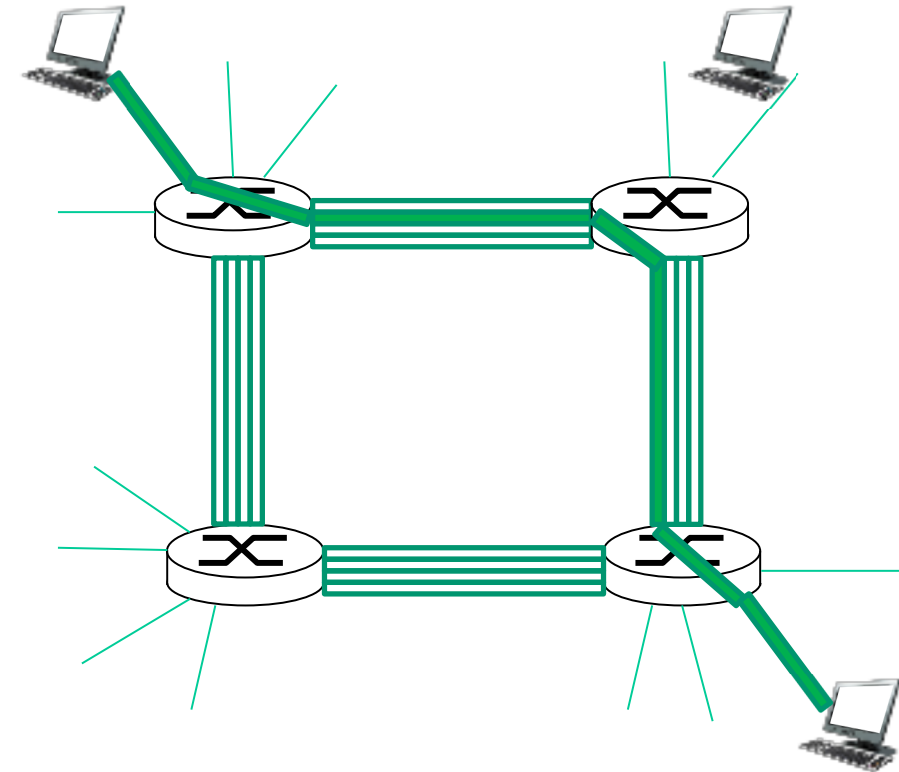


- **Enfileiramento:** pacotes são armazenados em uma **fila**.
- Aguardam oportunidade de transmissão.
- Se a **taxa de chegada** é maior que a capacidade de escoamento, fila tende a crescer.
- Mas o tamanho da fila é limitado (**Por que?**).
  - Quando a capacidade é excedida, pacotes são **descartados**.

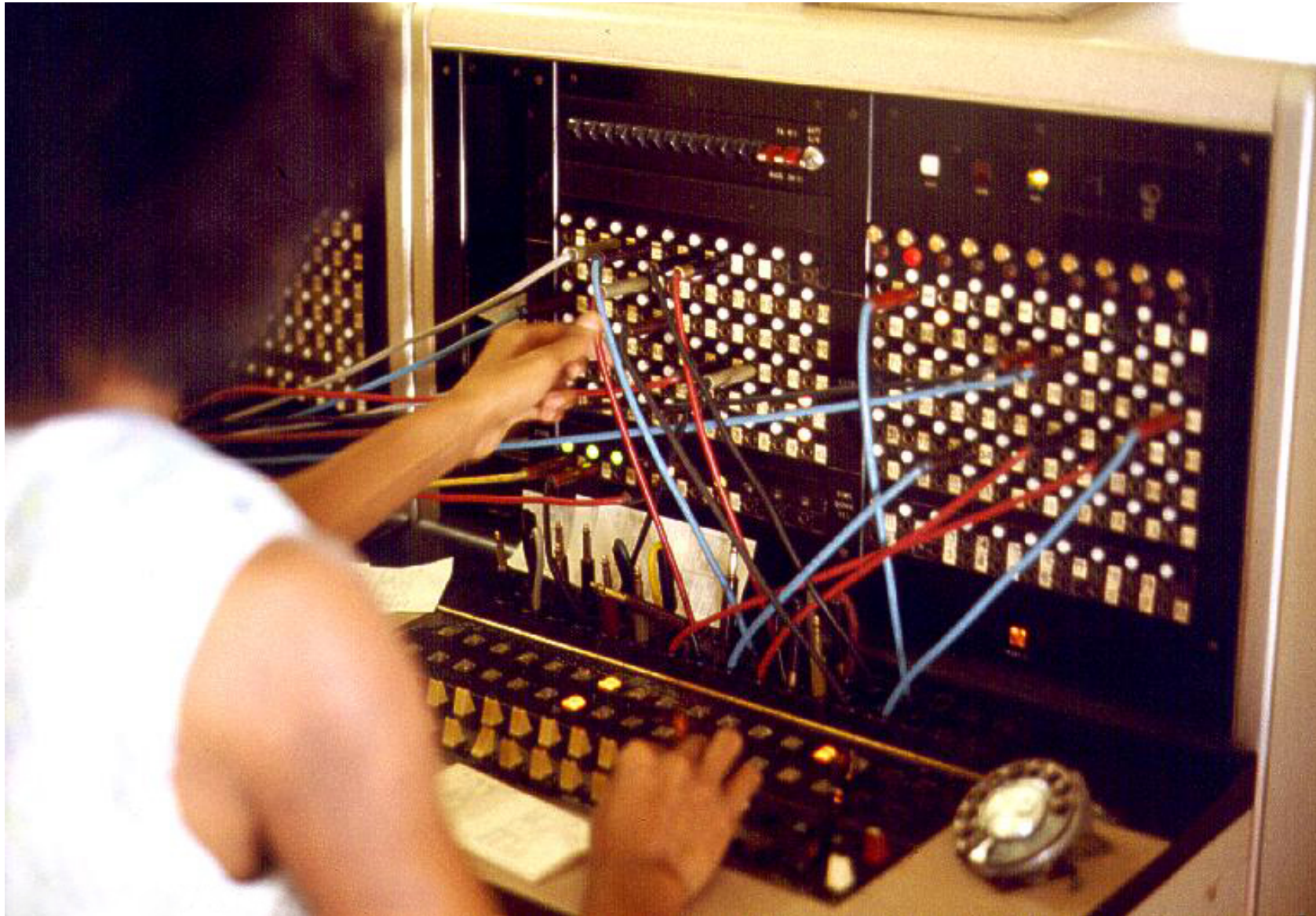


# Alternativa: Comutação de Circuitos (I)

- Recursos fim-a-fim são **alocados, reservados** para uma “chamada” entre origem e destino.
  - No exemplo, cada “enlace” contém 4 circuitos independentes.
    - Chamada segundo circuito do enlace de cima, primeiro circuito do enlace da direita.
- Recursos são dedicados: uma vez alocados, ninguém mais os utiliza.
  - Desempenho de circuito, garantido.
- Recursos ficam **ociosos** se não usados pela chamada.
- Comumente utilizado em redes de telefonia.

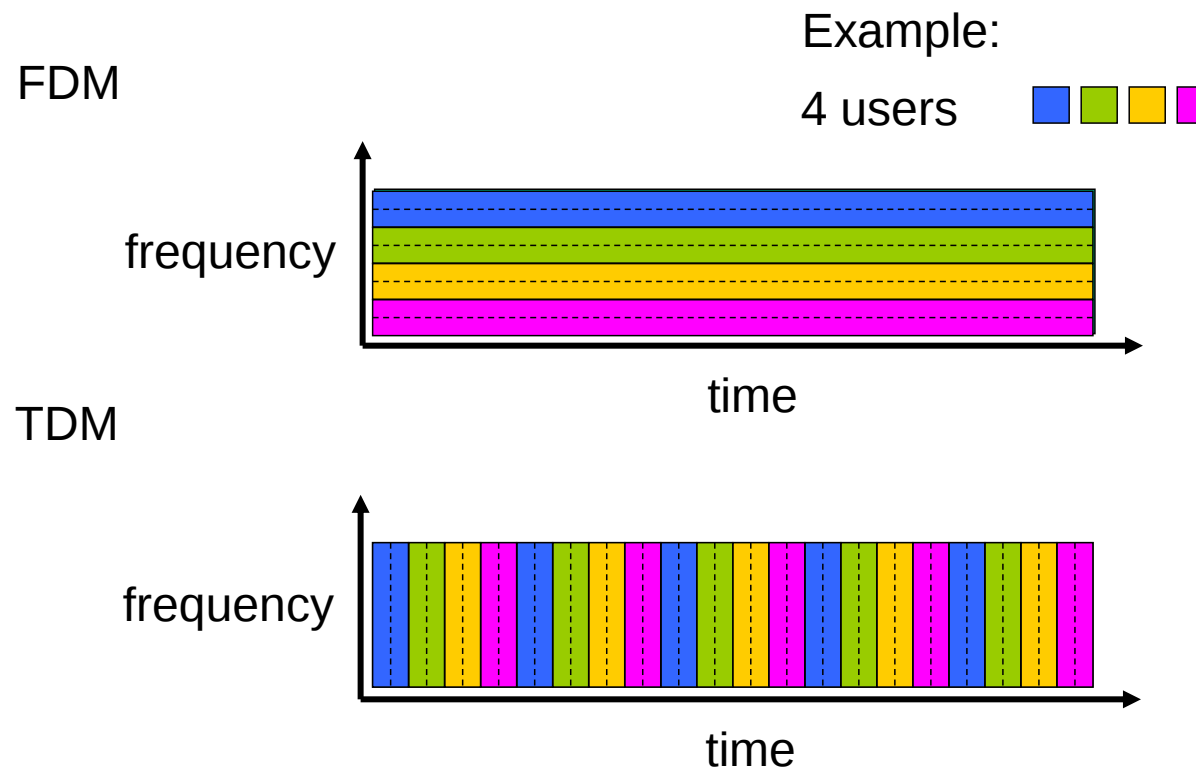


# Alternativa: Comutação de Circuitos (II)



# Comutação de Circuitos: Multiplexação

- Não necessariamente precisamos de circuitos fisicamente separados entre cada comutador.
- Chamadas podem ser **multiplexadas** em um mesmo meio físico.
- Dois exemplos comuns:
  - Multiplexação por Divisão no Tempo (TDM).
  - Multiplexação por Divisão na Frequência (FDM).

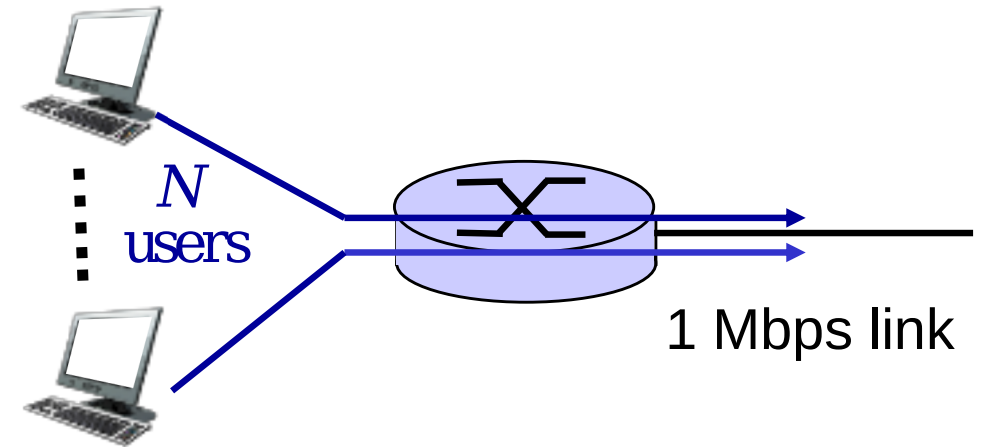


# Comutação de Pacotes: Multiplexação Estatística (I)

- Na comutação de pacotes, também há multiplexação.
  - Multiplexação ocorre “naturalmente”.
  - Pacotes de usuários distintos são transmitidos de forma intercalada.
  - Ordem depende da chegada (geralmente aleatória) dos pacotes no comutador.

- **Pergunta:** quantos usuários podem compartilhar um único enlace de saída?
- Exemplo: enlace de saída de 1 Mb/s.
- “Prometemos” 100 kb/s a cada usuário **quando ativos**.
  - Hipótese: cada usuário ativo apenas 10% do tempo.

- **Comutação de circuitos:**
  - Alocações fixas, recursos reservados.
  - No máximo, 10 usuários.



- **Comutação de pacotes:**
  - Recursos ociosos podem ser utilizados por outros usuários.
  - Logo: espera-se poder colocar **mais de 10** usuários!
- Mas qual é o valor exato de  $N$ ?

# Comutação de Pacotes: Multiplexação Estatística (II)

- Enlace de 1 Mb/s, 100 kb/s para cada usuário = no máximo 10 usuários **simultâneos**.
- Digamos que haja  $N = 35$  usuários.
  - Qual a **probabilidade** de que mais de 10 estejam ativos simultaneamente?
  - Considerando certas simplificações implícitas.
- Um usuário **qualquer** fica ativo com probabilidade  $p = 10\%$ .

**1 dado usuário ativo  
(demais desativos)**

$$p \cdot (1 - p)^{34}$$

**2 dados usuários ativos  
(demais desativos)**

$$p^2 \cdot (1 - p)^{33}$$

**k dados usuários ativos  
(demais desativos)**

$$p^k \cdot (1 - p)^{(N-k)}$$



# Comutação de Pacotes: Multiplexação Estatística (III)

- Note que com 35 usuários, há múltiplas combinações de  $k$  usuários que podem estar ativos simultaneamente.
  - $P/k = 1$ , 35 combinações.
  - $P/k = 2$ ,  $\frac{35 \cdot 34}{2} = 595$  combinações.
  - ...
  - $P/k$  em geral,  $\binom{35}{k}$  combinações.

**1 usuário ativo  
(demais desativos)**

$$35 \cdot p \cdot (1 - p)^{34}$$

**2 usuários ativos  
(demais desativos)**

$$595 \cdot p^2 \cdot (1 - p)^{33}$$

**$k$  usuários ativos  
(demais desativos)**

$$\binom{35}{k} \cdot p^k \cdot (1 - p)^{(35 - k)}$$

# Comutação de Pacotes: Multiplexação Estatística (IV)

- Voltando ao problema original: probabilidade de mais de 10 ativos.

$$Pr(k > 10) = \sum_{i=11}^{35} \binom{35}{i} \cdot p^i \cdot (1-p)^{(35-i)} \approx 0,00042$$

- Resumindo:
  - Mesmo com 35 usuários, entregamos a banda prometida 99,958% do tempo.
  - **Número muito maior de usuários** que na comutação de circuitos.
  - Percentual de falha “aceitável”.
- Mas o que acontece se mudarmos os parâmetros?
  - e.g., aumentarmos N, aumentarmos p, ou reduzirmos a capacidade do enlace?

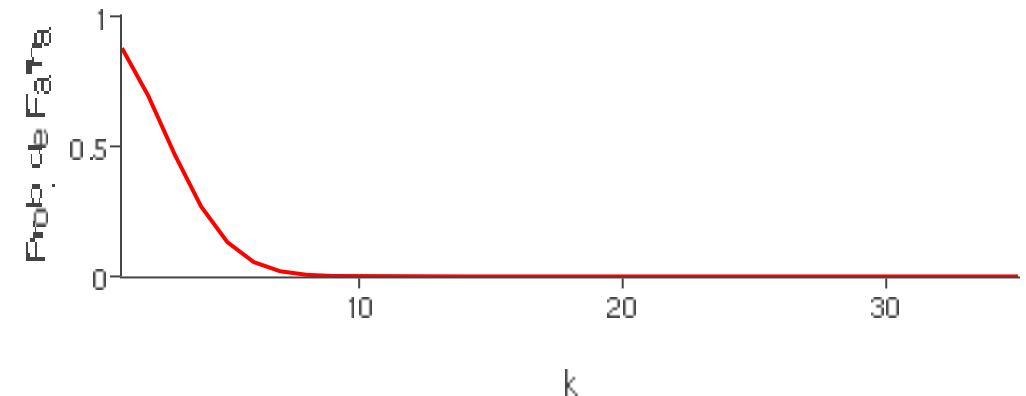
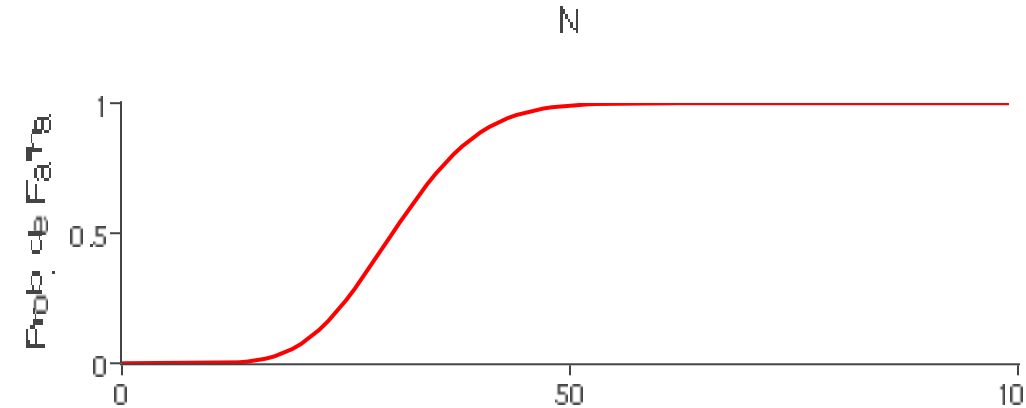
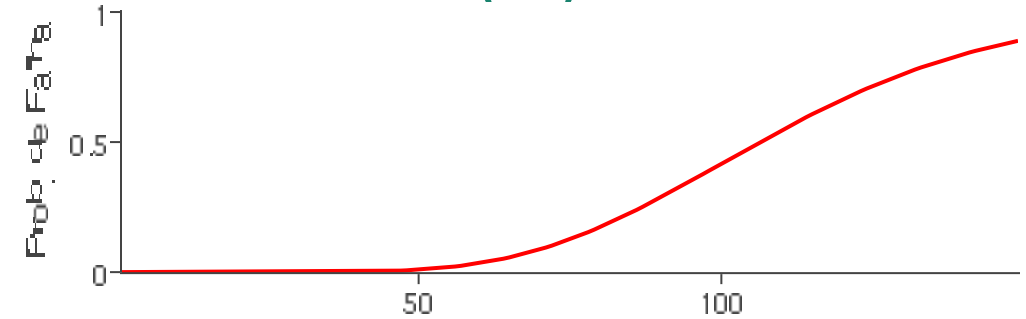
# Comutação de Pacotes: Multiplexação Estatística (V)

$$Pr(k > x) = \sum_{i=x}^N \binom{N}{i} \cdot p^i \cdot (1-p)^{(N-i)}$$

## Parâmetros

k:	10
N:	35
p:	0.1
<input type="button" value="Recalcular"/>	

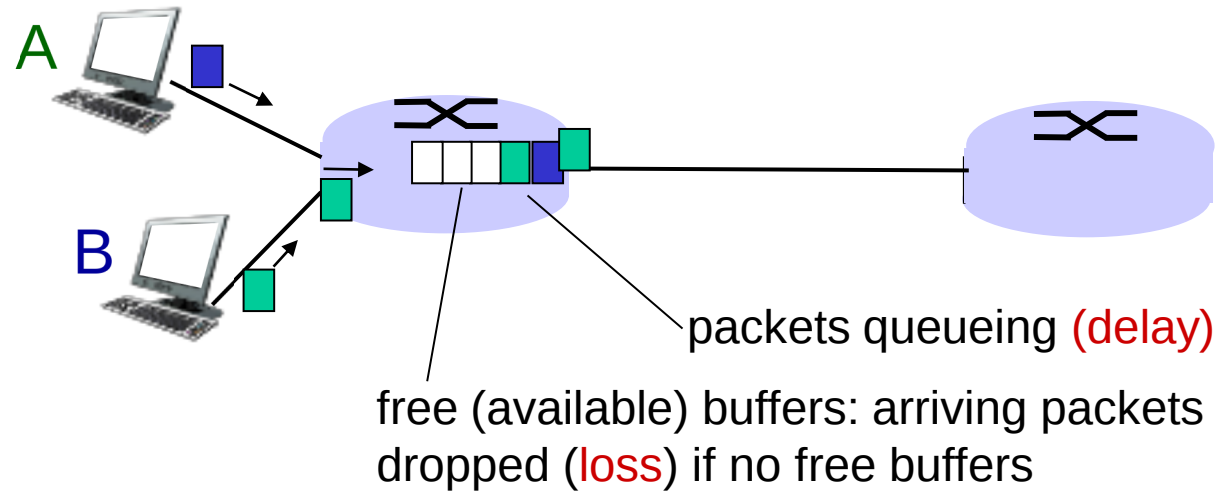
- Onde:
  - k: máximo de usuários simultâneos.
    - Depende da capacidade do enlace/rede e da banda para cada usuário.
  - N: número total de usuários.
  - p: probabilidade de um dado usuário estar ativo.





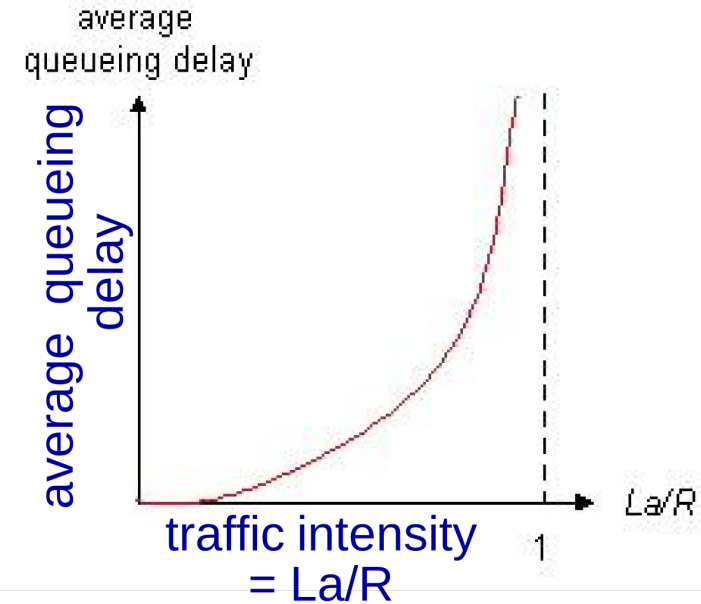
# Comutação de Pacotes: Enfileiramento e Atraso (I)

- Já sabemos que enfileiramento pode levar a descarte de pacotes.
  - Pacote chega ao comutador, não há mais espaço disponível no *buffer*.
- Mas o enfileiramento tem outra consequência: **o aumento no atraso**.
- Pacote enfileirado precisa aguardar que todos os pacotes a sua frente sejam transmitidos.
  - Maiores filas → maior tempo de espera.
- e.g., assumindo que cada pacote leve 10 ms para ser transmitido.
  - Com 5 pacotes na fila, 50 ms.
  - Com 100 pacotes na fila, 1 s.



# Comutação de Pacotes: Enfileiramento e Atraso (II)

- Sejam:
  - $R$ : capacidade do enlace (b/s).
  - $L$ : tamanho dos pacotes (b).
  - $a$ : taxa de chegada de pacotes (p/s).
- Logo,  $L \cdot a$  é a taxa de chegada em b/s.
- Se:
  - $\frac{L \cdot a}{R} \approx 0$ , pacotes esperam pouco na fila.
  - $\frac{L \cdot a}{R} \rightarrow 1$ , pacotes esperam muito na fila.
  - $\frac{L \cdot a}{R} > 1$ , mais trabalho que o comutador consegue suportar  $\rightarrow$  tempo de espera “infinito”.
    - i.e., descarte de pacotes inevitável!



$La/R \sim 0$

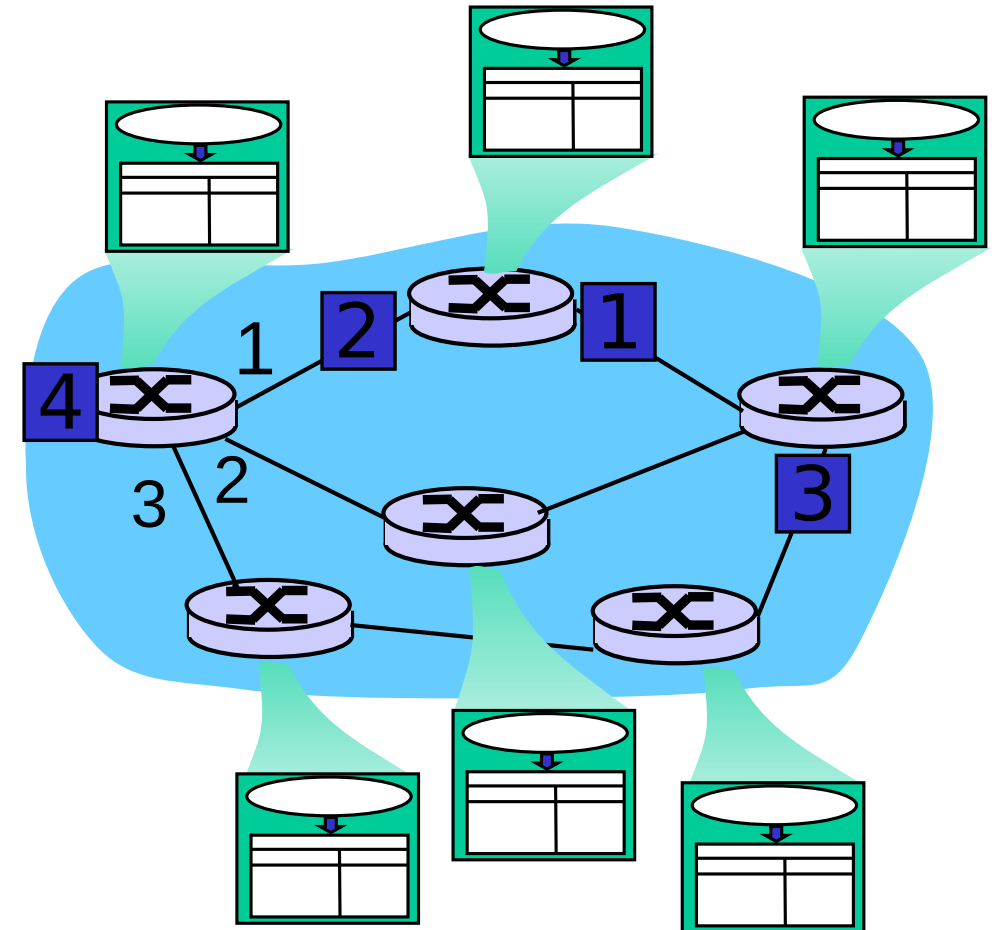


$La/R \rightarrow 1$



# Comutação de Pacotes: Reordenação

- Outra característica da comutação de pacotes.
- Pacotes podem “mudar de ordem”.
  - i.e., pacote transmitido antes, pode chegar depois.
- Razão mais comum: caminhos distintos.
  - Decisão de como encaminhar pacote é (normalmente) do comutador.
  - Guiada por um processo de **roteamento**.
  - Roteamento pode mudar escolhas **dinamicamente**.



# Comutação de Pacotes vs. Comutação de Circuitos

- **Pergunta:** Qual abordagem é melhor?
- **Resposta:** depende do cenário, objetivos.
- Comutação de pacotes funciona bem para tráfego em **rajada**.
  - Permite compartilhamento melhor dos recursos.
  - Não necessita de estabelecimento de chamada.
- Mas pode resultar em **congestionamento excessivo**.
  - Filas longas, descartes de pacotes, atrasos.
  - Não podemos **prometer** muita coisa.
- Certas aplicações precisam de garantias.
  - Como prover garantias em uma rede de comutação de pacotes?
  - Problema ainda em aberto (Redes II).

# Comutação de Circuitos: Quem Lida com os Problemas?

- Usamos a Internet todos os dias para transmitir arquivos.
- Obviamente, queremos que estes arquivos cheguem perfeitos, **íntegros** no destinatário.
- Como isso é possível, se a rede pode descartar, reordenar pacotes?
- **Resposta:** os terminais precisam lidar com isso.
  - *i.e., implantar mecanismos de recuperação.*
  - **Assumindo que isto seja desejável.**
  - Argumento fim-a-fim, inteligência nas bordas.

# Resumo da Aula...

- **Comutação de pacotes.**

- Informação dividida em unidades chamadas pacotes.
- Cada pacote tratado isoladamente.
- Quando transmitido, pacote usa toda a capacidade do enlace.

- **Store-and-Forward.**

- Armazenar e encaminhar.
- Comutador só pode começar transmissão no próximo enlace, quando recepção estiver **completa**.
- Transmissões do pacote são sequenciais pela rota.

- **Comutação de circuitos.**

- Estabelecimento prévio de chamada.
- Recursos pré-alocados, não compartilhados.
- Possibilidade de **ociosidade**.

- **Multiplexação.**

- Dados de vários usuários “combinados” em um mesmo meio.
- FDM, TDM.

- Multiplexação estatística.

- Consequência da comutação de pacotes.
- Ordem intercalada, aleatória de pacotes.
- **Pode melhorar utilização dos recursos.**

- Enfileiramento.

- Consequência do *store-and-forward*.
- Ocorre quando chegada supera a capacidade de saída.
- Descarte de pacotes, atrasos.

# Próxima Aula...

- Conclusão do capítulo introdutório:
  - Métricas de desempenho: perda de pacotes, atraso, vazão.
  - Modelos em camadas: ISO/OSI, TCP/IP.
  - Encapsulamento.
  - Breves considerações sobre segurança.
  - Histórico da Internet.