

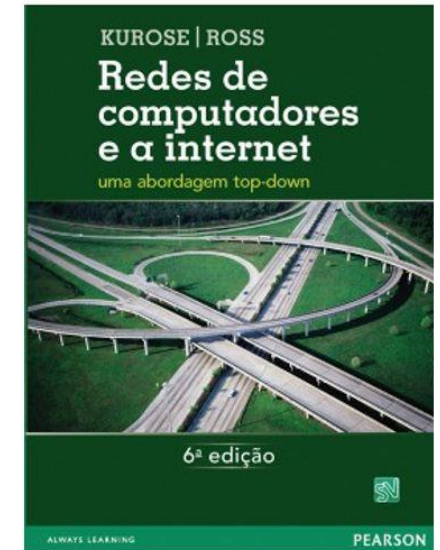
## Redes de Computadores

### 05 – Camada de Transporte I

Prof. Edson J. C. Gimenez  
*[soned@inatel.br](mailto:soned@inatel.br)*

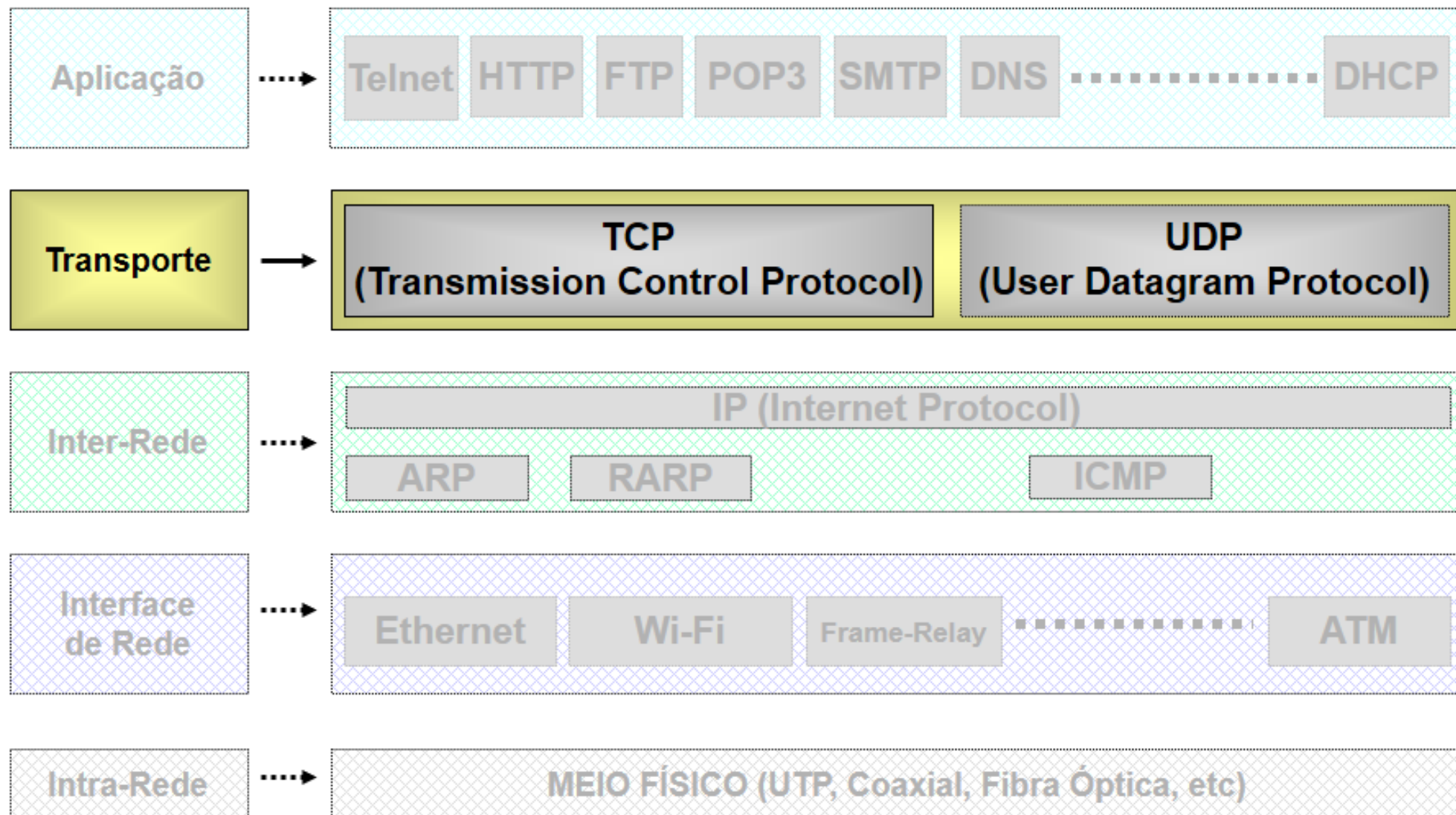
## Referências:

- ✓ Kurose & Ross. Redes de Computadores e a Internet: uma abordagem top-down. Capítulo 3.



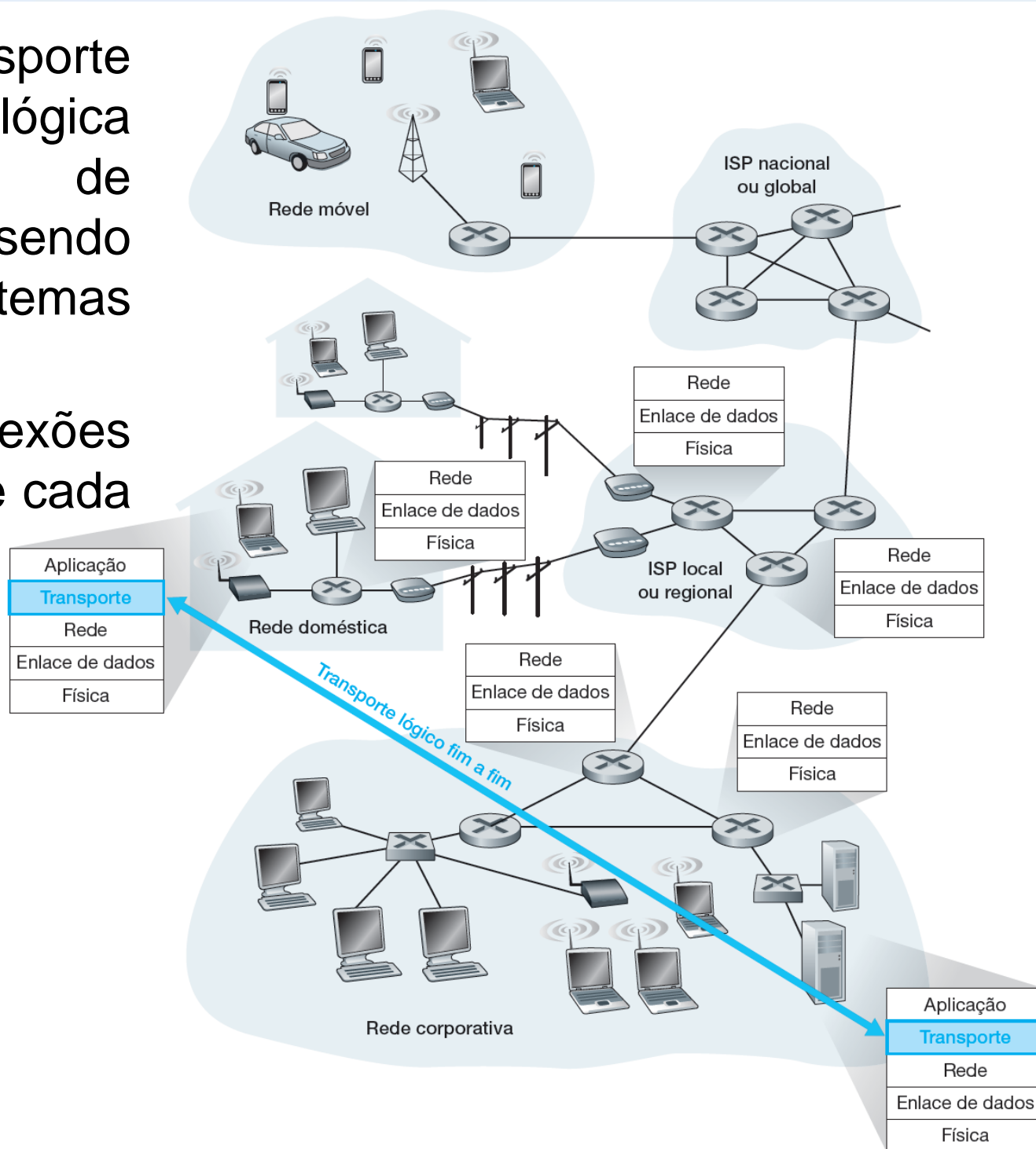
## Outras referências:

- ✓ Tanenbaum & Wetherall. Redes de Computadores. Capítulo 6.
- ✓ Farrel. A Internet e seu Protocolos: uma Análise Comparativa. Capítulo 7.
- ✓ Comer. Interligação de Redes com TCP/IP, Volume 1. Cap. 10, 11, 12 e 28.



- ✓ A camada de transporte fornece comunicação lógica entre processos de aplicações, sendo implementados nos sistemas finais.
- ✓ São estabelecidas conexões lógicas, fim-a-fim, entre cada par de aplicações que se comunicam.

\* Protocolos fim-a-fim.



### Serviços providos pela camada de transporte

- ✓ A função básica da camada de transporte é promover uma transferência de dados fim-a-fim confiável\*, eficiente e econômica entre aplicações origem e destino, independente das camadas abaixo.
- ✓ São oferecidos dois tipos de serviços:
  - O serviço orientado a conexão e confiável – protocolo **TCP\***.
  - O serviço não orientado a conexão e não confiável – protocolo **UDP**.

### Serviços providos pela camada de transporte

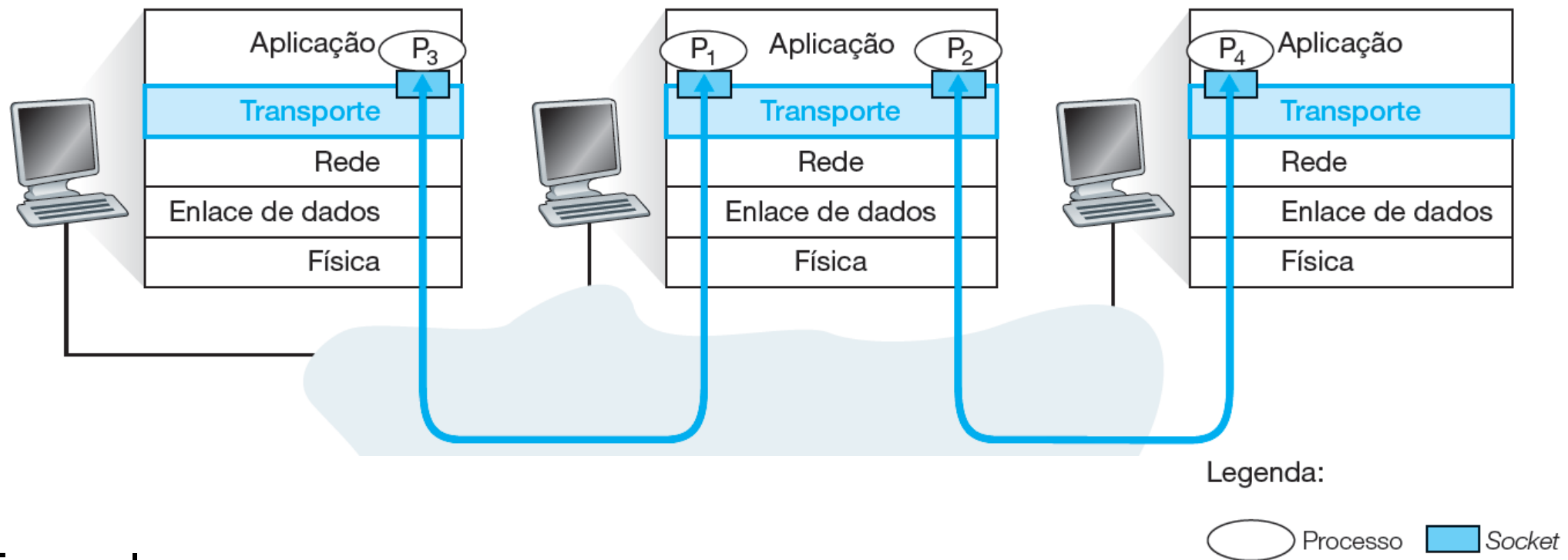
#### ✓ Multiplexação/Demultiplexação – TCP e UDP

- Conceito associado à capacidade de ampliação do serviço de entrega hospedeiro a hospedeiro provido pela camada de rede para um serviço de entrega processo a processo para aplicações que rodam nesses hospedeiros.
- O trabalho de reunir, no hospedeiro de origem, partes de dados provenientes de diferentes sockets, encapsular cada parte de dados com informações de cabeçalho (que mais tarde serão usadas na demultiplexação) para criar segmentos, e passar esses segmentos para a camada de rede e denominada multiplexação
- A tarefa de entregar os dados contidos em um segmento da camada de transporte ao socket correto é denominada demultiplexação.



Serviços providos pela camada de transporte

✓ Multiplexação/Demultiplexação – TCP e UDP



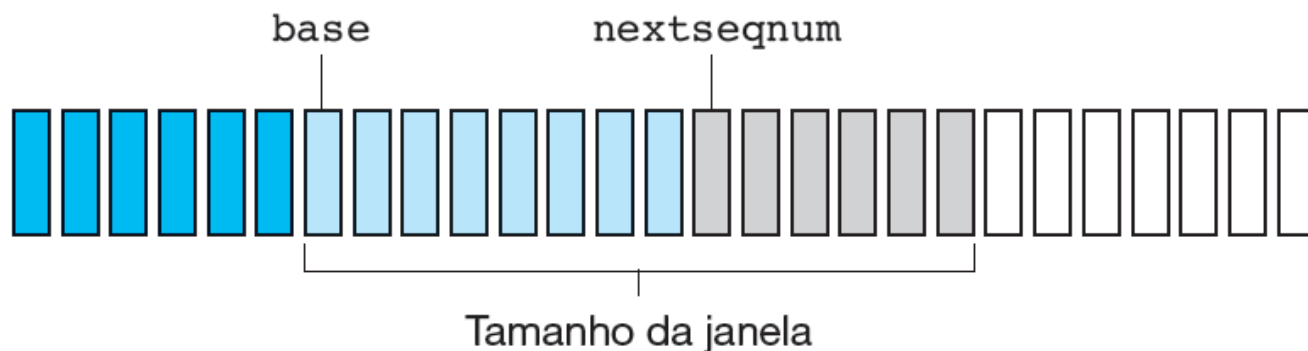
Exemplo:

- Aplicação P<sub>1</sub> e P<sub>3</sub> (uma conexão lógica) – um par de identificadores de portas.
- Aplicação P<sub>2</sub> e P<sub>4</sub> (outra conexão lógica) – um par de identificadores de portas diferentes.

### Serviços providos pela camada de transporte

#### ✓ Controle de fluxo/controle de congestionamento

- O controle de fluxo tem por objetivo evitar que um sistema origem, enviando mensagens, sobrecarregue um sistema destino.
- Ao iniciar um fluxo de dados entre dois sistemas, os mesmos fazem uso dos seguintes parâmetros: janela de transmissão, bytes transmitidos esperando confirmação, bytes transmitidos confirmados.
- O “tamanho de janela” indica a quantidade de bytes que cada sistema está apto a receber, podendo variar conforme o sistema esteja mais ou menos congestionado.
- Assim, o sistema que envia o fluxo de bytes, ao atingir o limite da janela de transmissão, deve interromper sua transmissão até que receba uma confirmação dos bytes transmitidos, liberando então sua janela de transmissão.



#### Legenda:

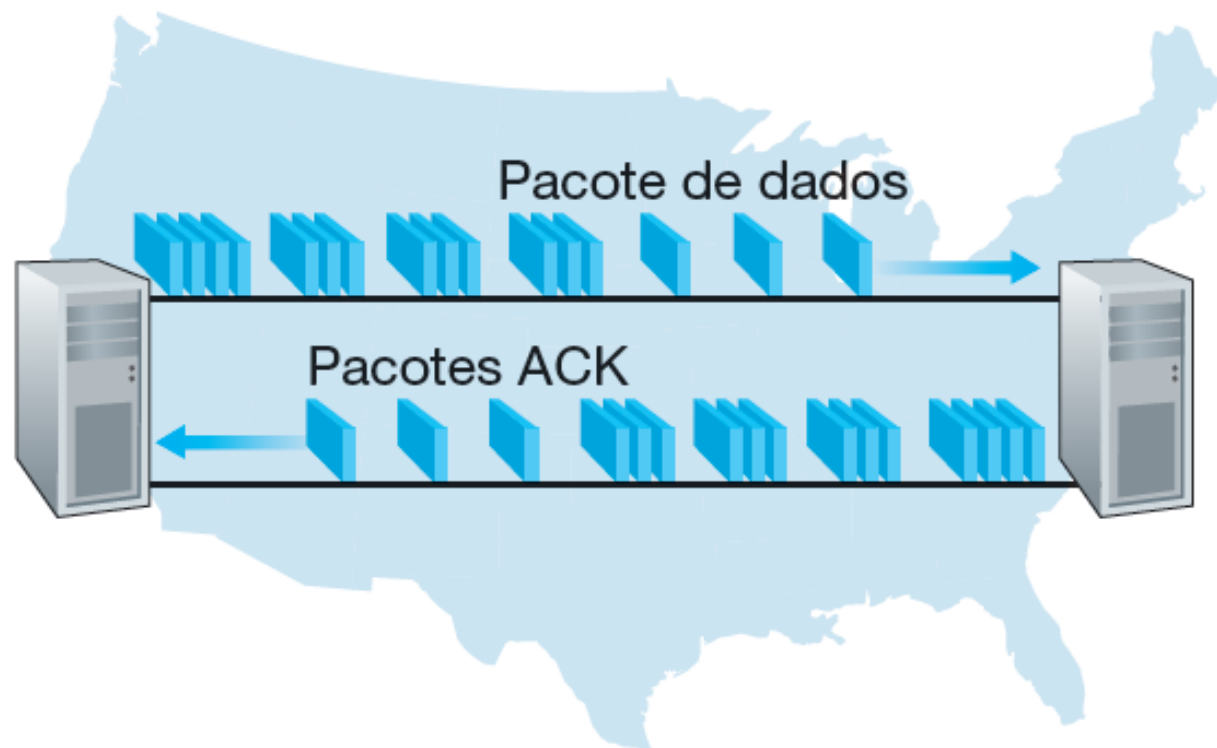
	Já reconhecido		Autorizado, mas ainda não enviado
	Enviado, mas ainda não reconhecido		Não autorizado



### Serviços providos pela camada de transporte

#### ✓ Controle de erro e retransmissão

- Tem por objetivo fazer com que dois sistemas troquem dados de maneira confiável
- Cada segmento de dados transmitido é identificado por um “número de segmento”; e conforme esses segmentos vão chegando ao sistema destino, este pode confirmar seu recebimento através de mensagem de reconhecimento (ACK).

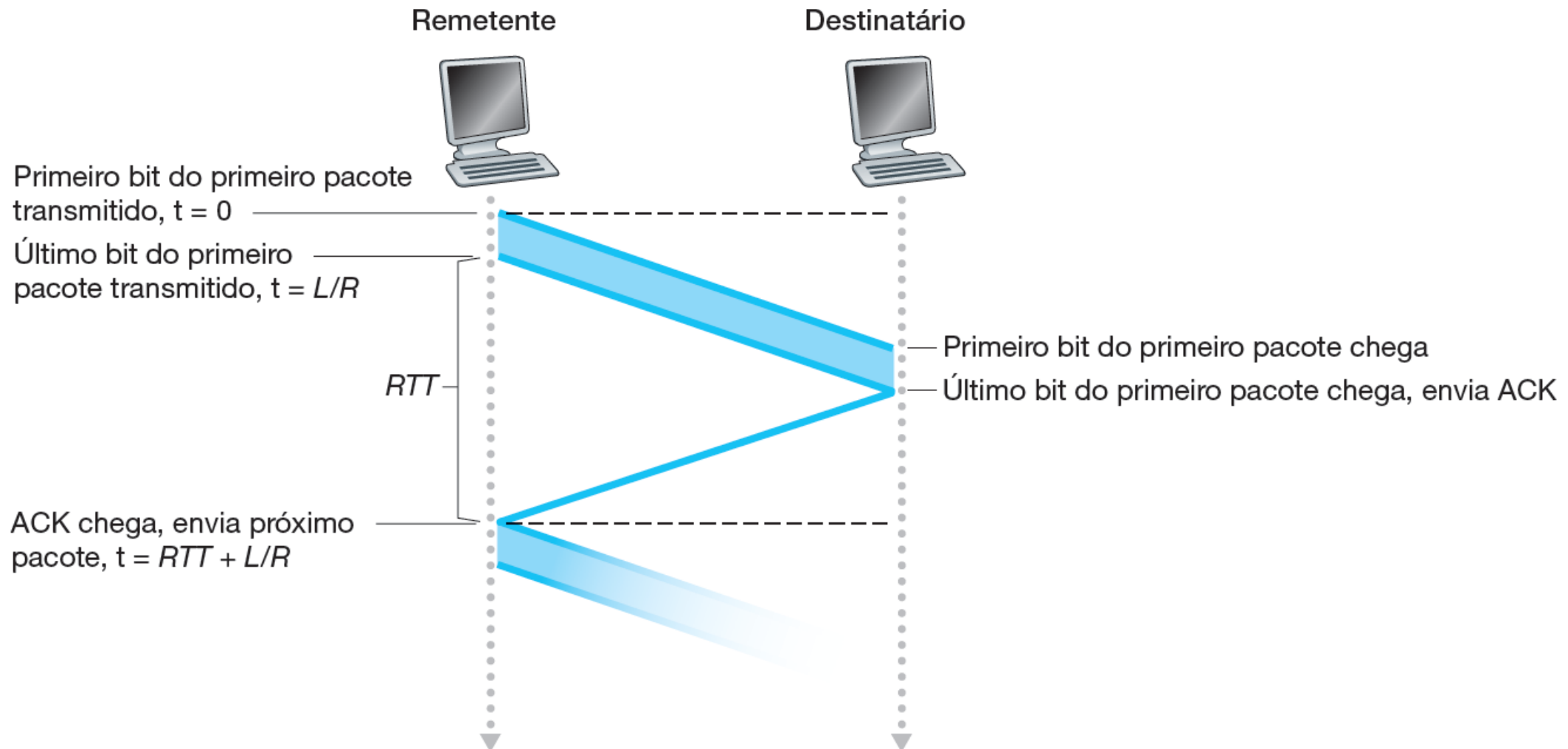


### Serviços providos pela camada de transporte

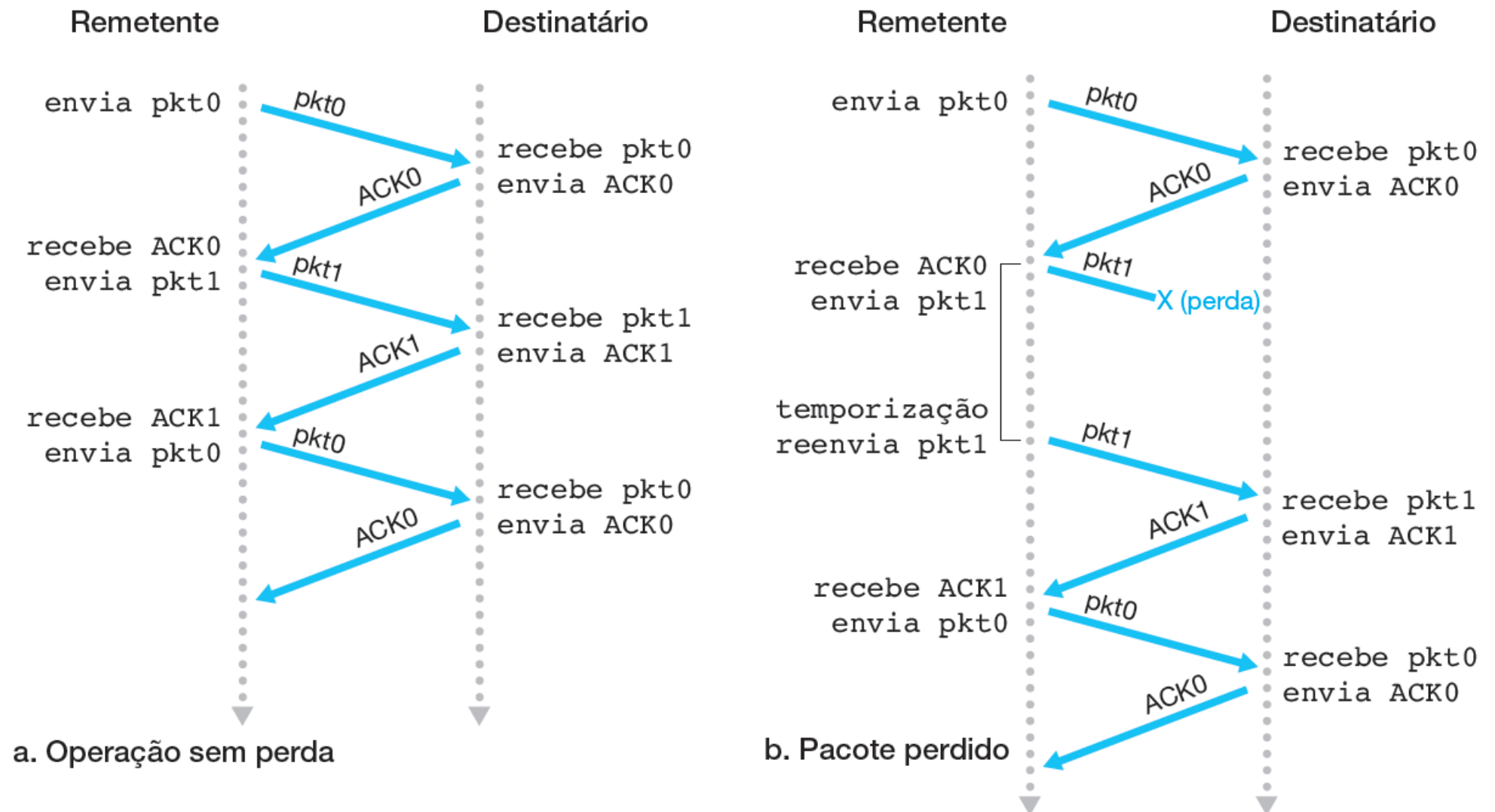
#### ✓ Controle de erro e retransmissão

- O sistema origem mantém em seu buffer os segmentos transmitidos e ainda não confirmados, para possível retransmissão se necessário.
- O controle de retransmissão pode fazer uso de uma dentre três técnicas:
  - **Stop-and-Wait:** a cada segmento transmitido aguarda-se uma confirmação; o próximo segmento só é transmitido após a confirmação do segmento anterior.
  - **Go-back-N:** uma sequência de segmentos podem ser enviados, sem nenhuma confirmação, até um número limite (tamanho da janela); se há um pedido de retransmissão de um segmento, este e os demais transmitidos após este segmento, serão todos retransmitidos.
  - **Selective-repeat:** também permite o envio de uma sequência de segmentos até um número limite (tamanho da janela); se um segmento deve ser retransmitido, apenas este segmento será retransmitido.

### Stop-and-Wait



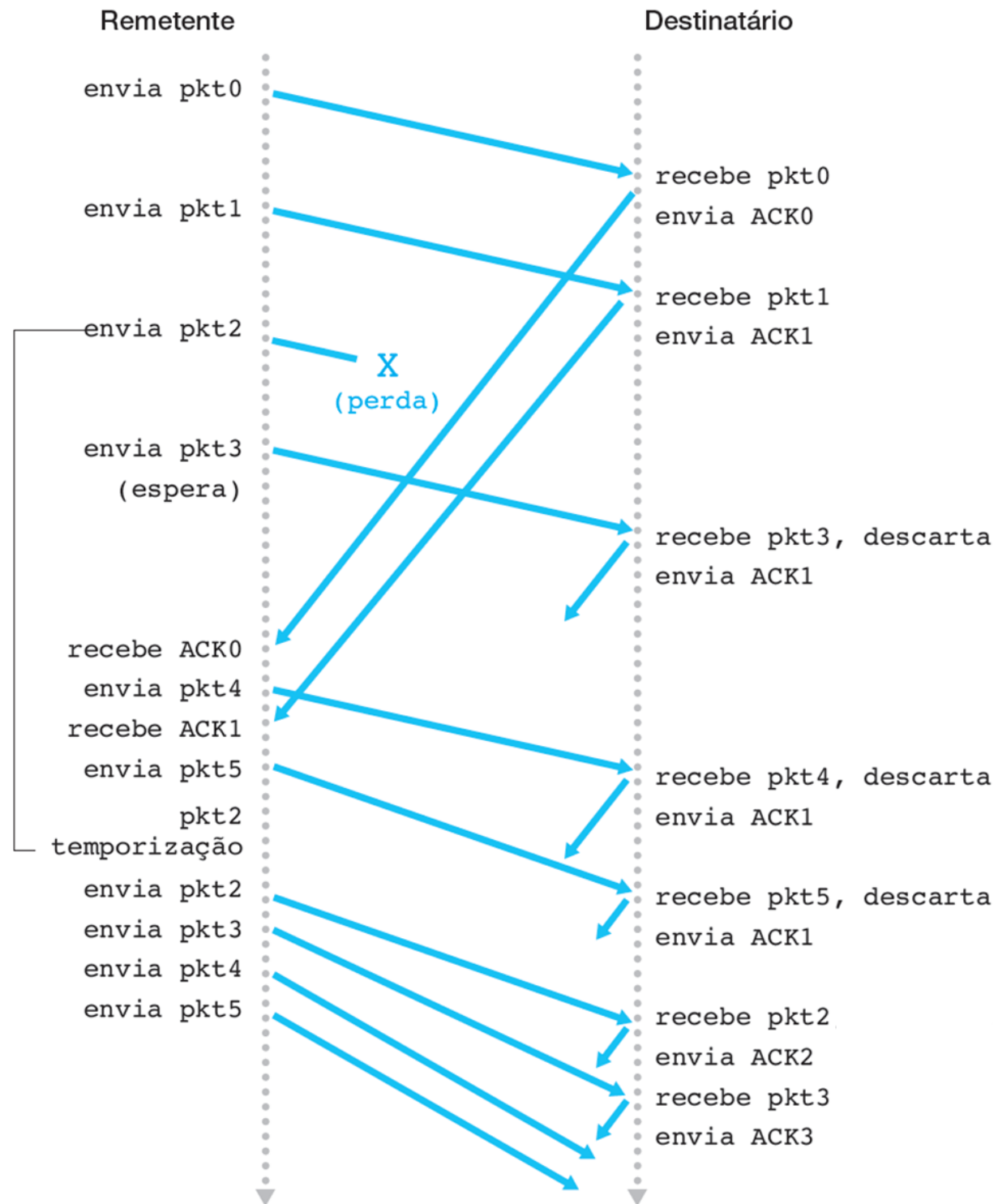
### Stop-and-Wait



- No exemplo, uso da técnica bit alternado

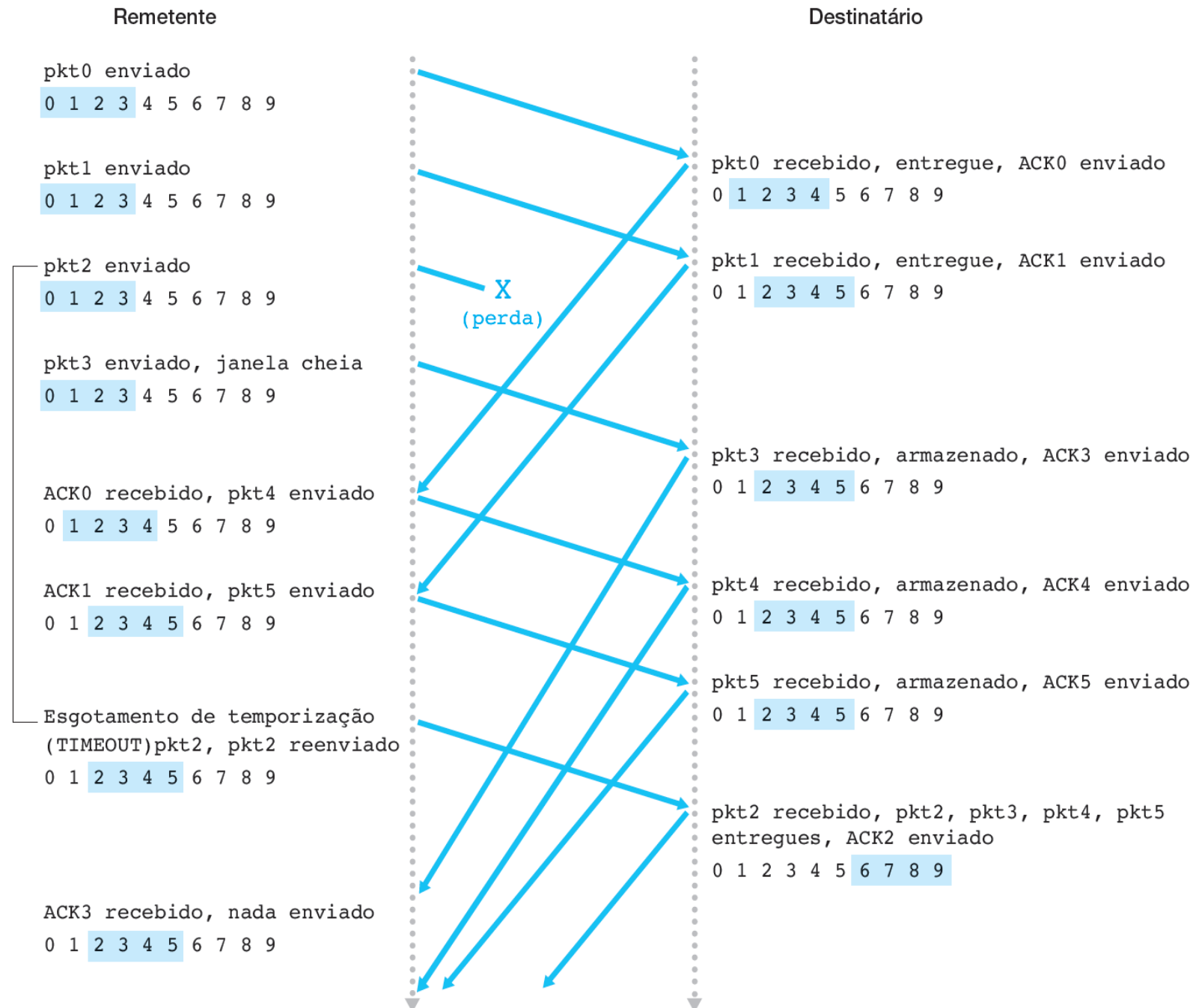
### Go-back-N

✓ Exemplo: janela = 4



### Selective-repeat

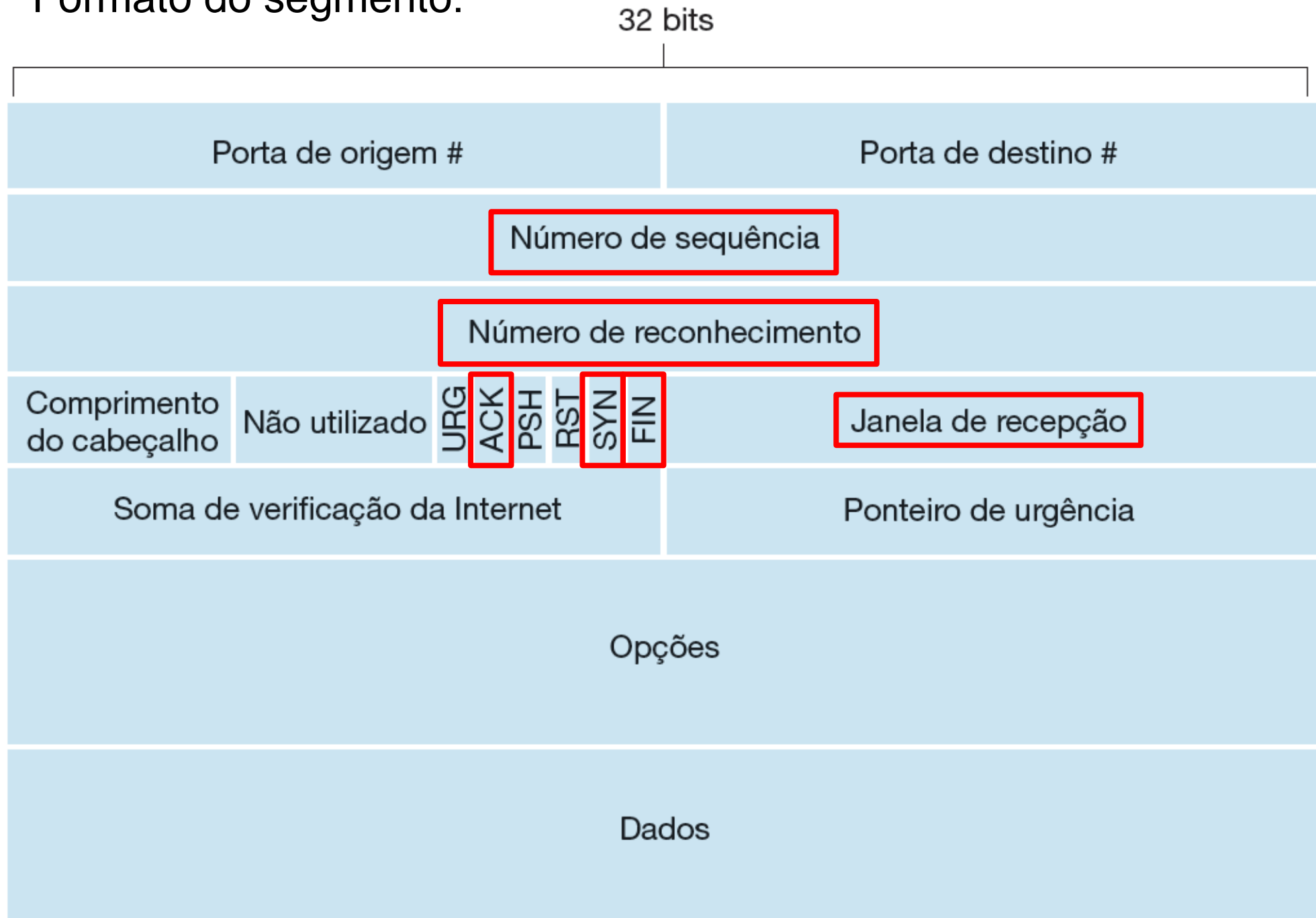
✓ Exemplo:  
janela = 4





- ✓ Principais características:
  - Orientado à conexão
  - Confiável
  
- ✓ Funções oferecidas às aplicações:
  - Conexões lógicas entre processos.
  - Transferência em fluxo (*stream*) de dados.
  - Confiabilidade na transferência dos dados entre processos.
  - Multiplexação de processos.
  - Controle de fluxo.
  - Controle de congestionamento.

✓ Formato do segmento:

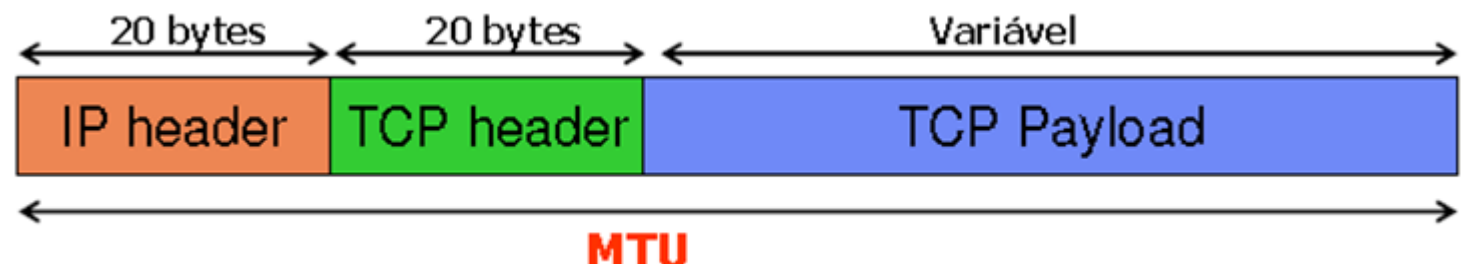


### ✓ Formato do segmento:

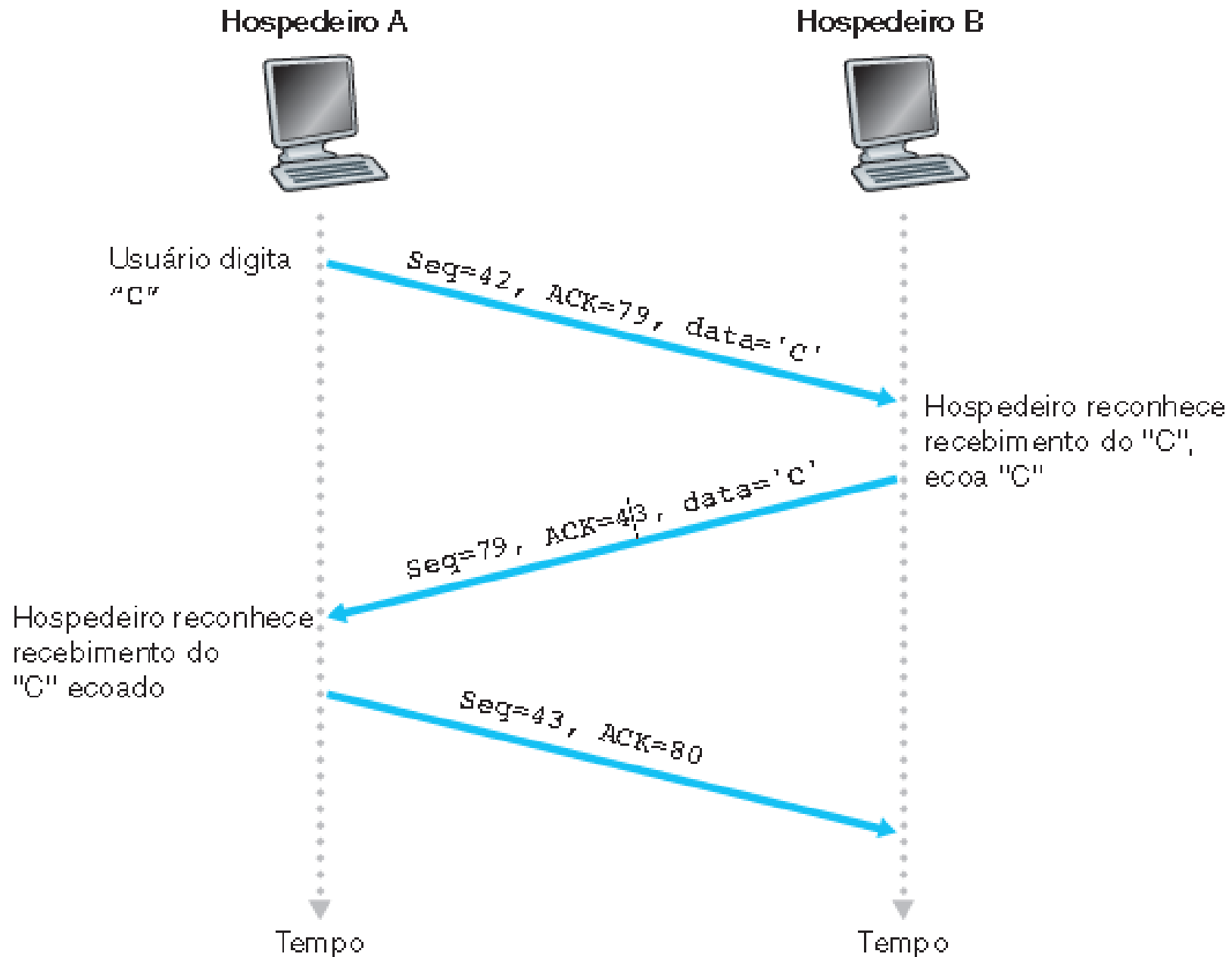
- Porta de Origem: número de 16 bits da porta de origem
- Porta de Destino: número de 16 bits da porta de destino
- Número de Sequência: o número sequencial do primeiro byte de dados neste segmento
- Número de Reconhecimento: se o bit de controle ACK estiver definido, este campo contém o valor do próximo número sequencial que o receptor está esperando receber
- Comprimento do cabeçalho: número de palavras de 32 bits no cabeçalho TCP
- Não utilizado: seis bits reservados para uso futuro; devem ser 0

### ✓ Formato do segmento:

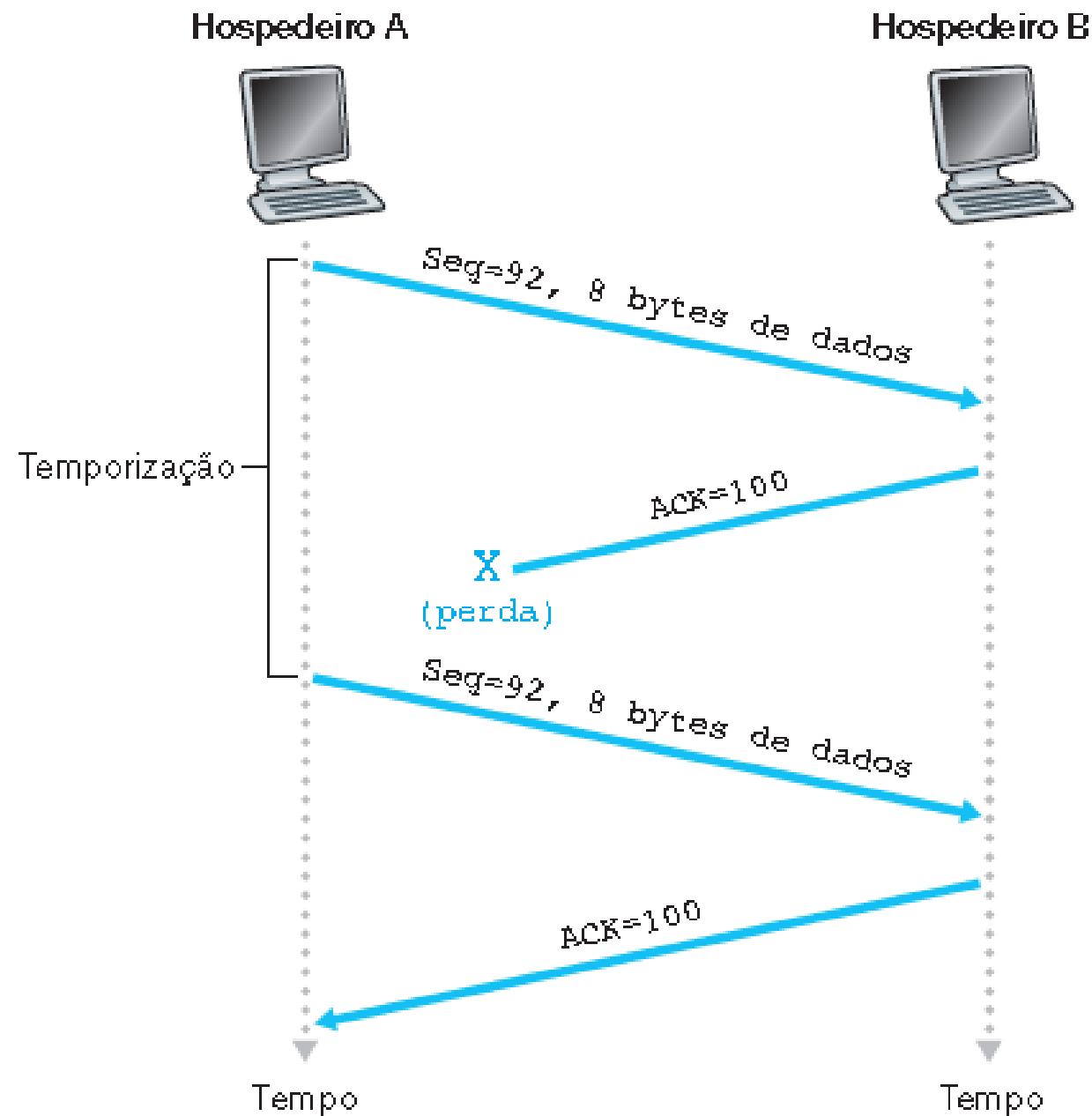
- Bits de código: ACK (campo de reconhecimento significativo), SYN (sincroniza os números de sequência), FIN (não há mais dados no emissor), etc.
- Janela de recepção: indica o número de bytes de dados começando com aquele indicado no campo número de reconhecimento que o receptor quer aceitar
- Soma de verificação: usada para detectar erros de bits em um segmento recebido..
- Preenchimento: bytes zero para preencher o cabeçalho TCP em múltiplos de 32 bits.
- Tamanho máximo do segmento (MSS): depende da MTU da interface (camada 2).



### ✓ Reconhecimento e retransmissões:

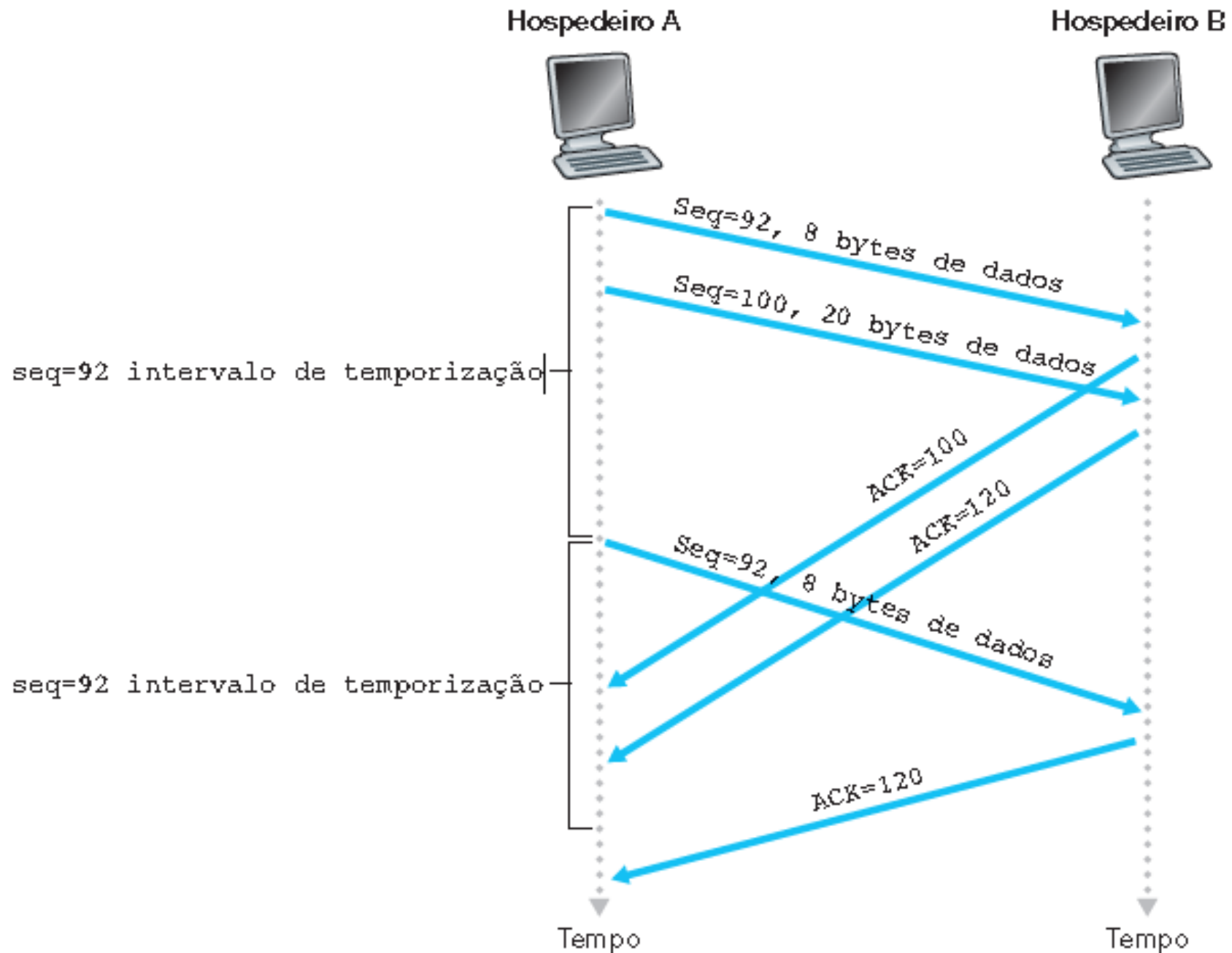


### ✓ Reconhecimento e retransmissões:

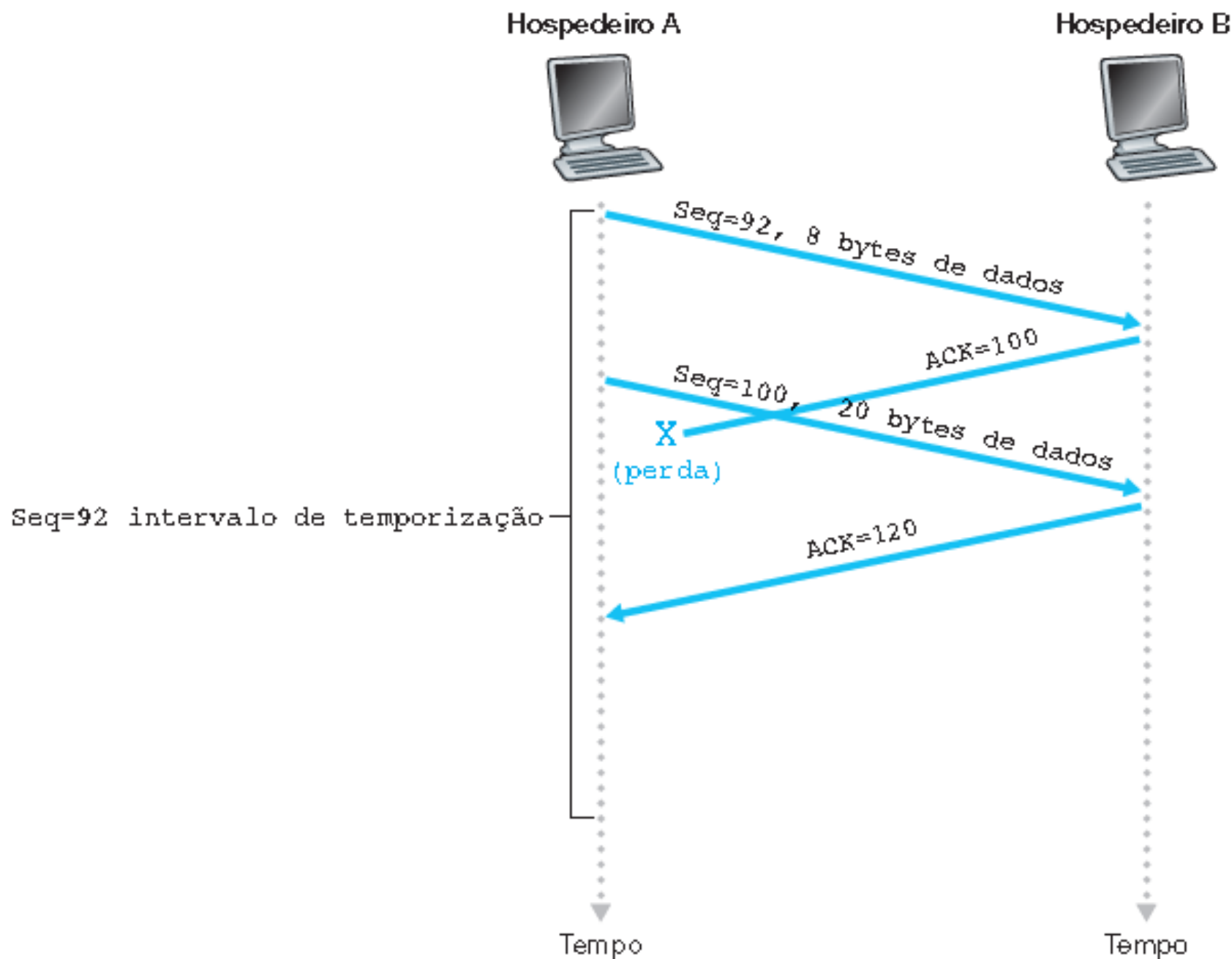




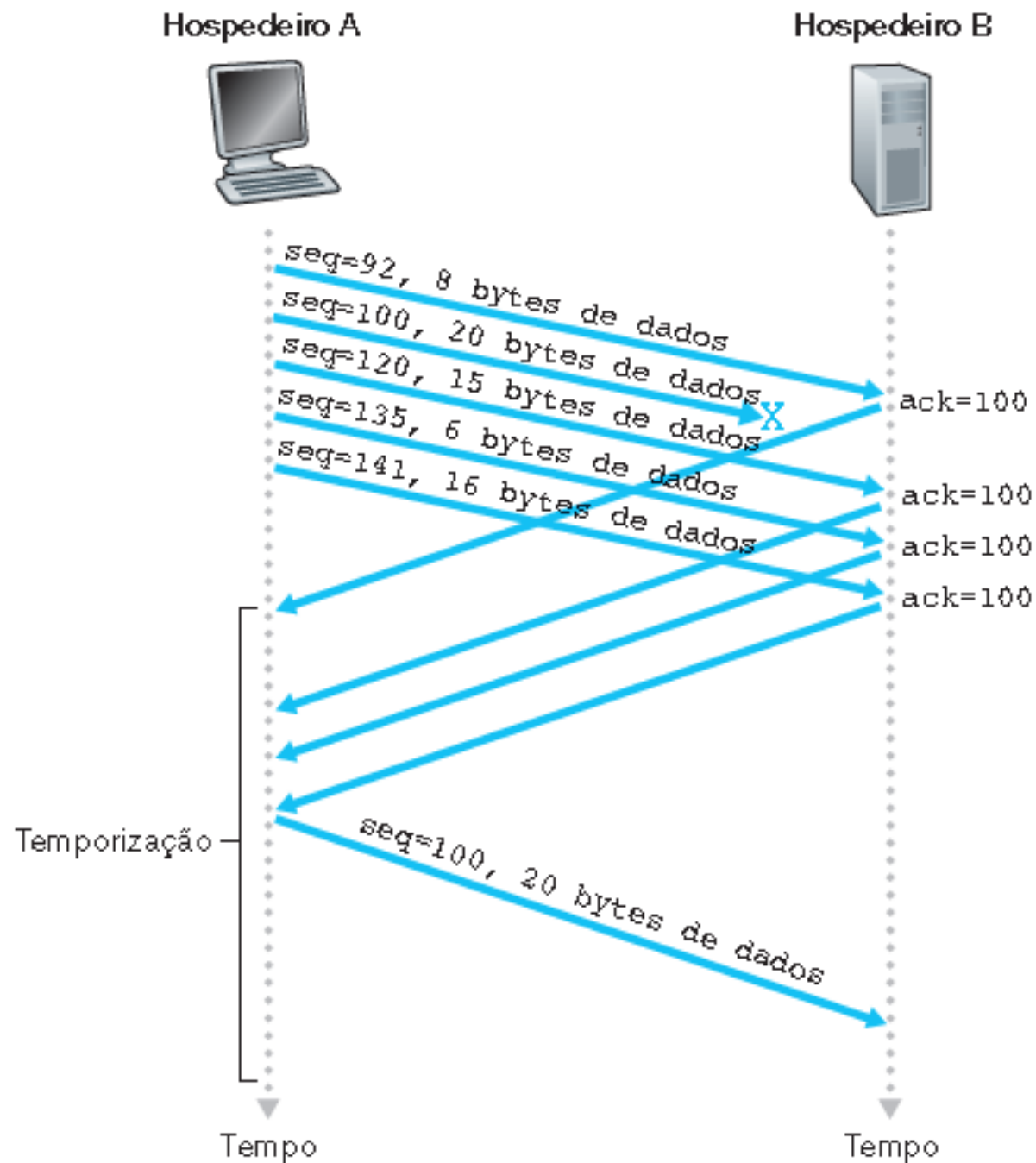
### ✓ Reconhecimento e retransmissões:



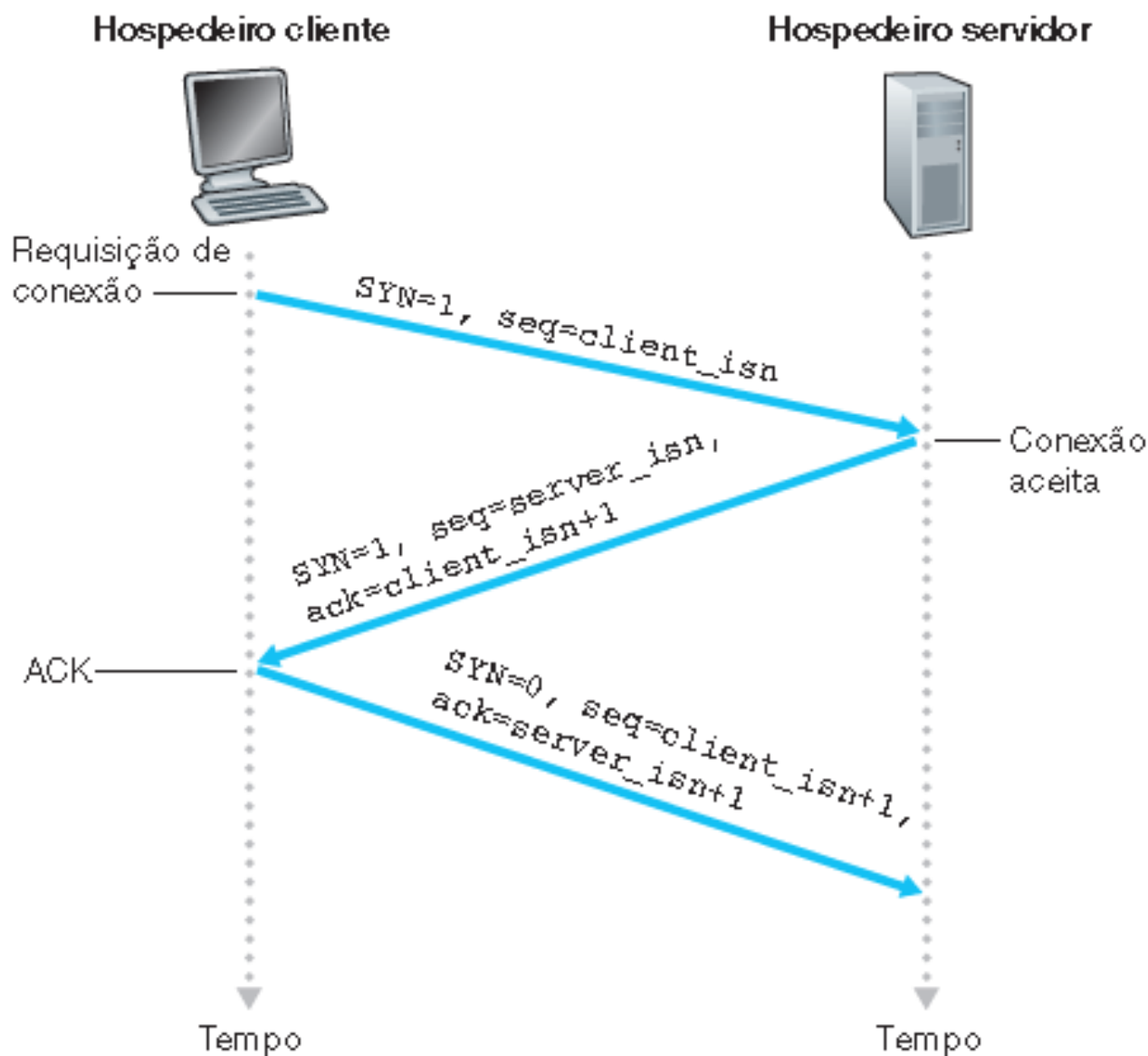
### ✓ Reconhecimento e retransmissões:



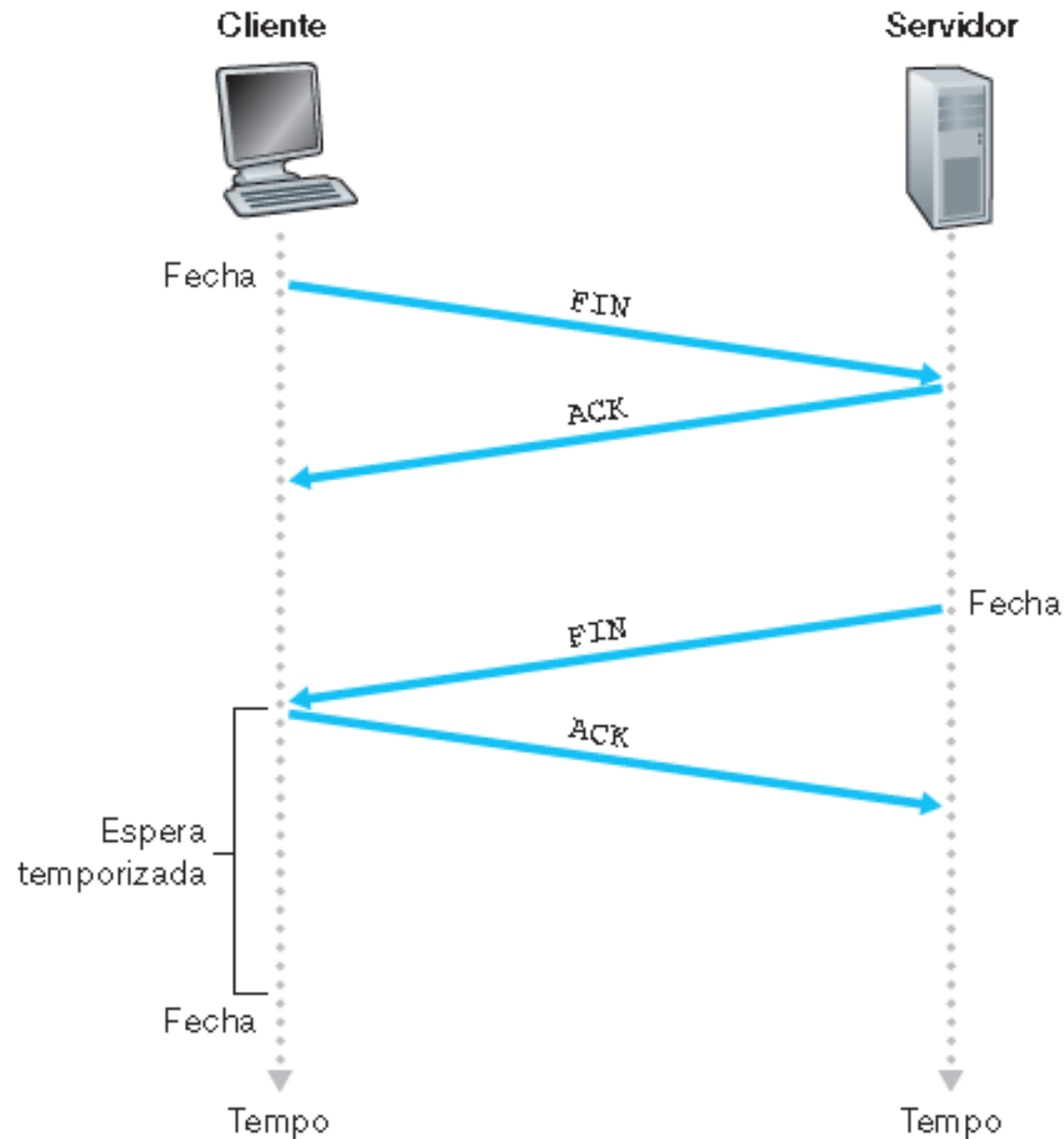
### ✓ Reconhecimento e retransmissões:



- ✓ Estabelecimento de conexão:
  - 3-way handshake (apresentação de três vias).



✓ Encerramento de conexão:



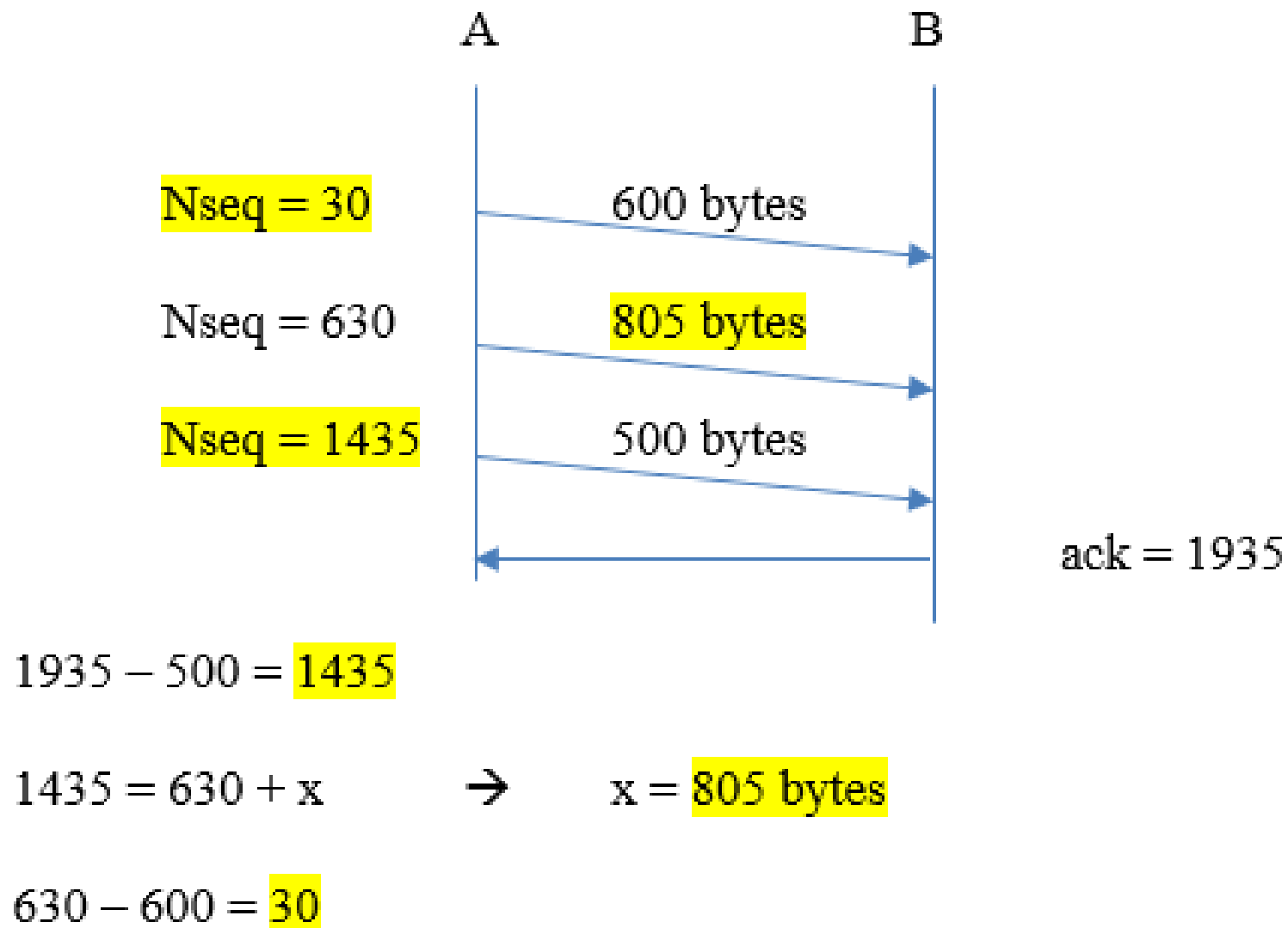
## Exemplo 1

Considere três segmentos TCP contendo dados, enviados de um host A para um host B, um após o outro. No primeiro segmento são enviados 600 bytes, o segundo segmento como “número de sequência” o valor 630, e no terceiro segmento são enviados 500 bytes. Sabendo-se que os três segmentos chegaram sem erros, e que o host B enviou de volta um segmento TCP com o número de reconhecimento (ACK) igual a 1935, responda:

- a) Qual o número de sequência do 1º segmento?
- b) Quantos bytes de dados foram enviados no 2º segmento?



## Exemplo 1

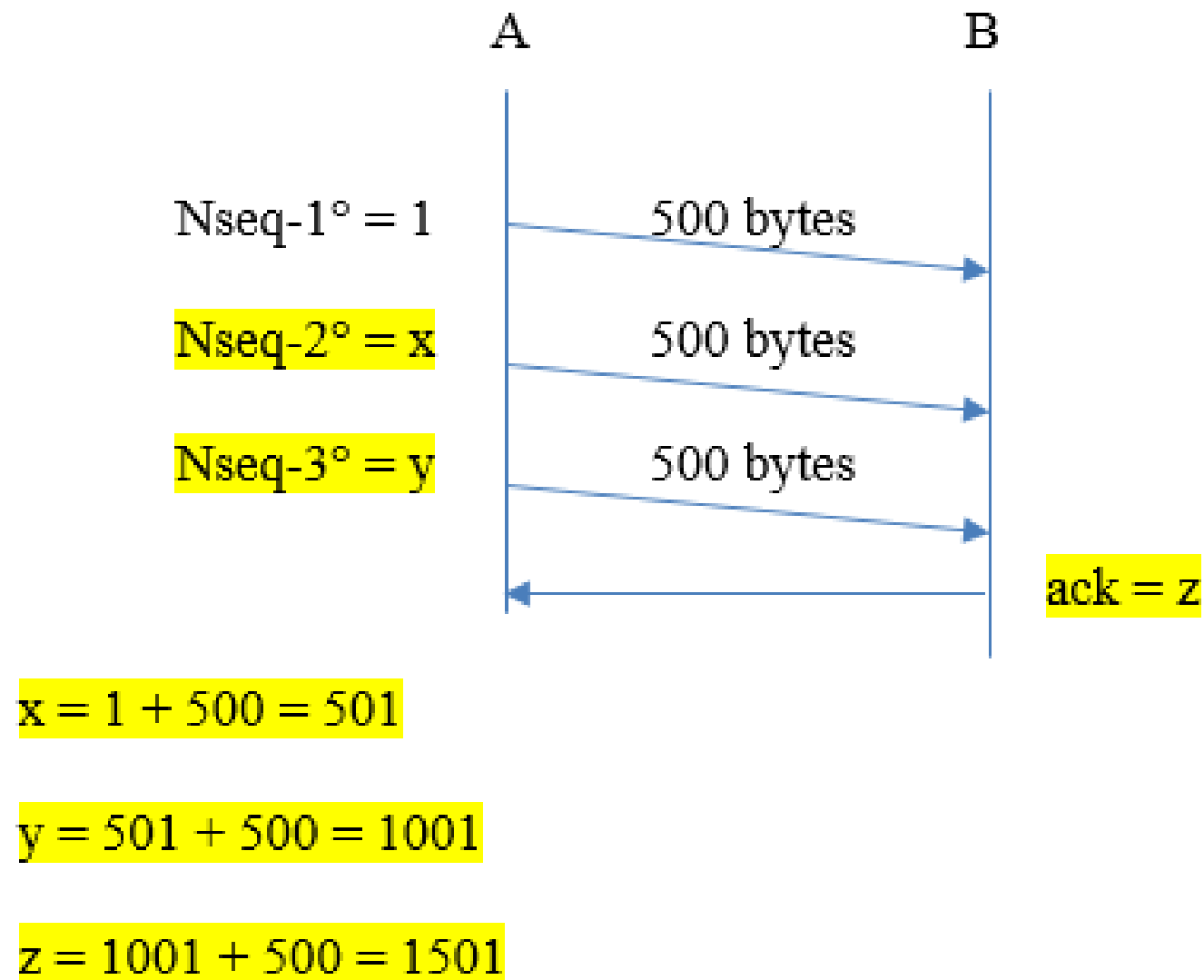


## Exemplo 2

Um arquivo de tamanho de 1500 bytes é totalmente transferido da aplicação origem para a aplicação destino através de segmentos TCP enviados na seguinte ordem e tamanhos: 500 bytes de dados; 500 bytes de dados; 500 bytes de dados. Sabendo que não ocorreu nenhum erro durante essa conexão, e o campo “Número de Sequência - SN” do primeiro segmento TCP enviado era 1, responda:

- a) Qual o número de sequência do segundo segmento TCP?
- b) Qual o número de sequência do último segmento TCP?
- c) Qual o número de reconhecimento enviado pela aplicação destino, após ter recebido o último segmento TCP corretamente?

## Exemplo 2



## Atividade 05 – Transporte I