

T02 – A/B/C Redes de Dados I

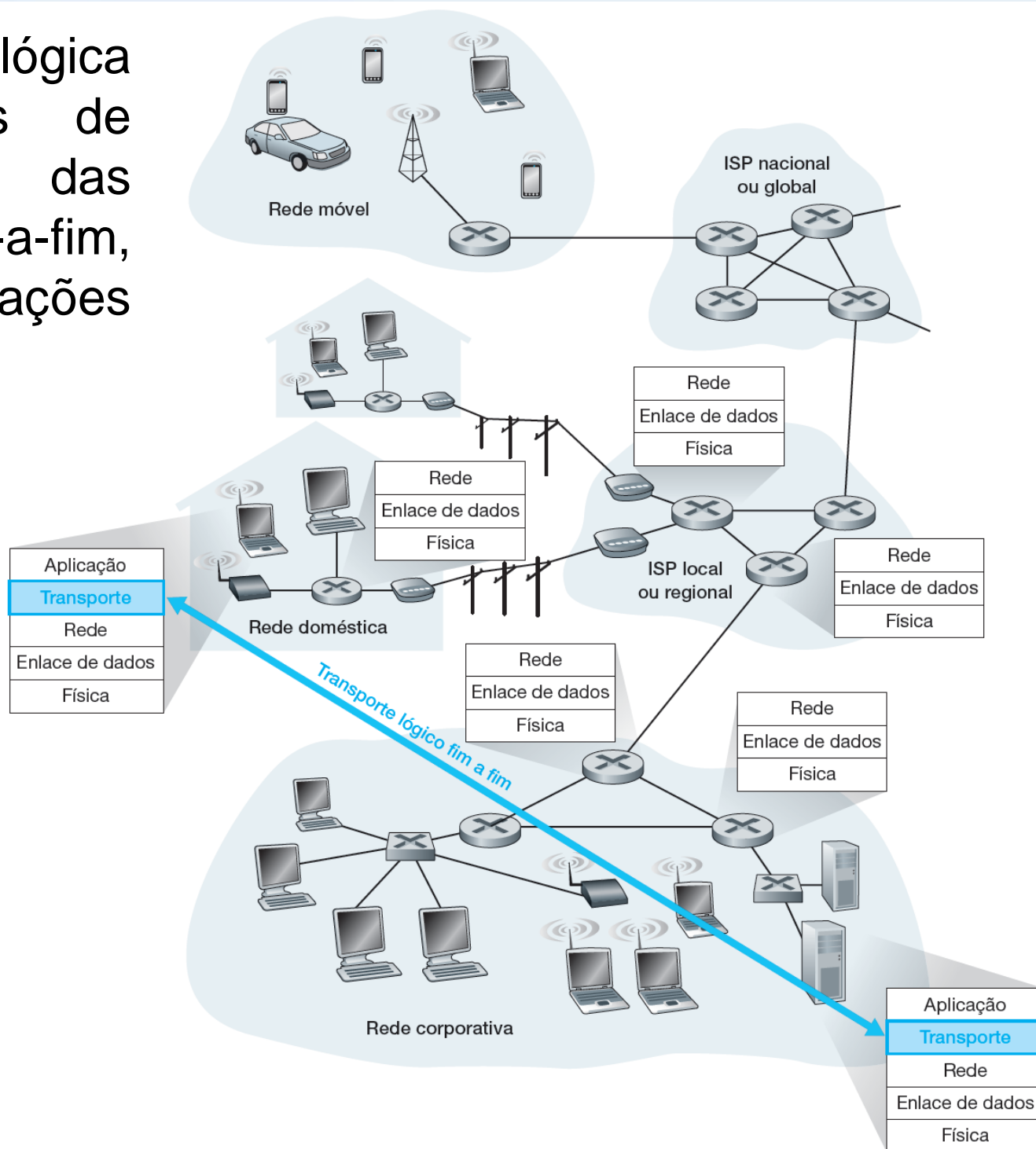
04 – Camada de Transporte (UDP - TCP)

Prof. Edson J. C. Gimenez
soned@inatel.br

Referências:

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

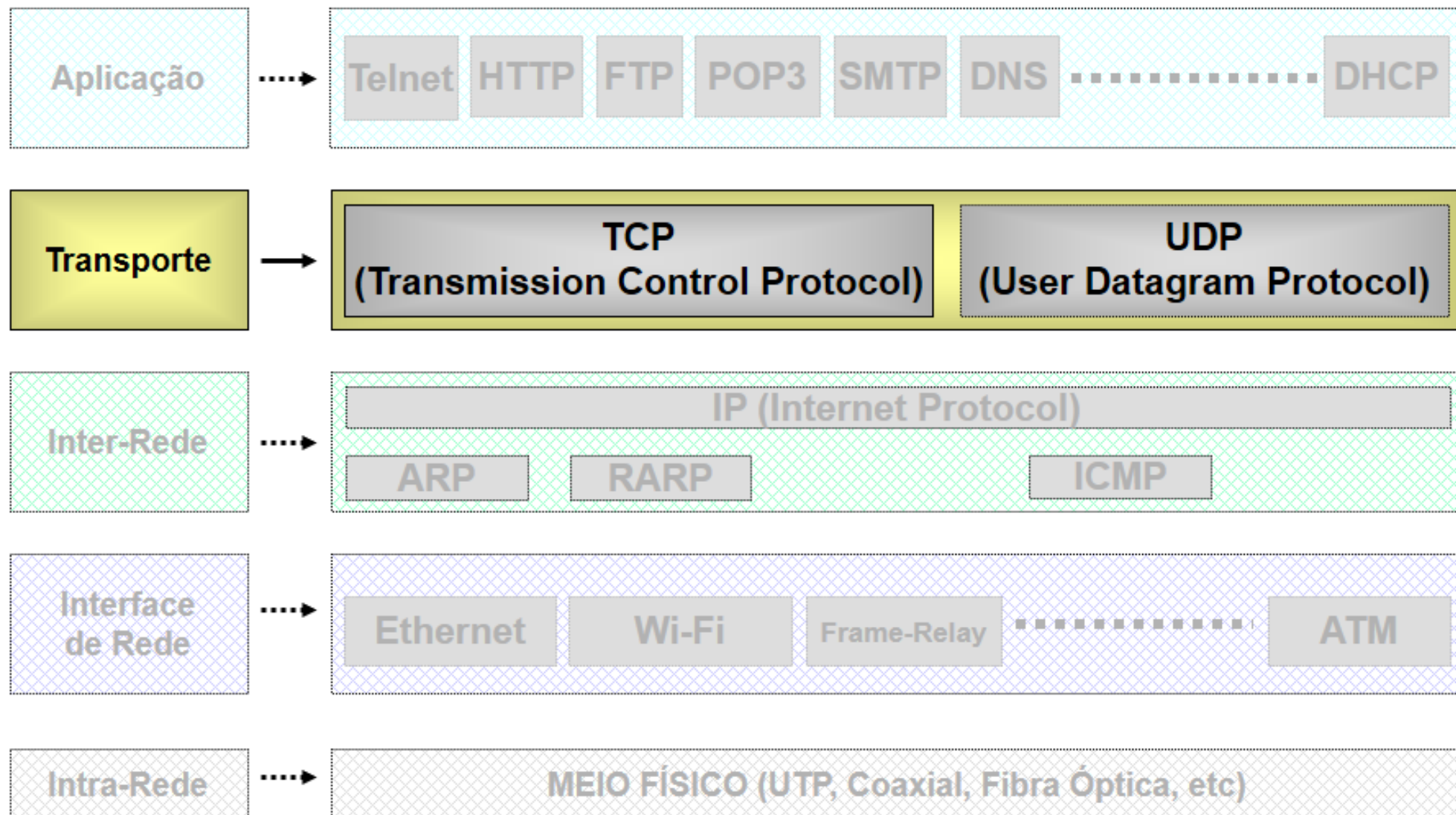
- ✓ Fornece comunicação lógica entre os processos de aplicações, através das conexões lógicas fim-a-fim, entre cada par de aplicações que se comunicam.
- ✓ Protocolos TCP e UDP



* Protocolos fim-a-fim.

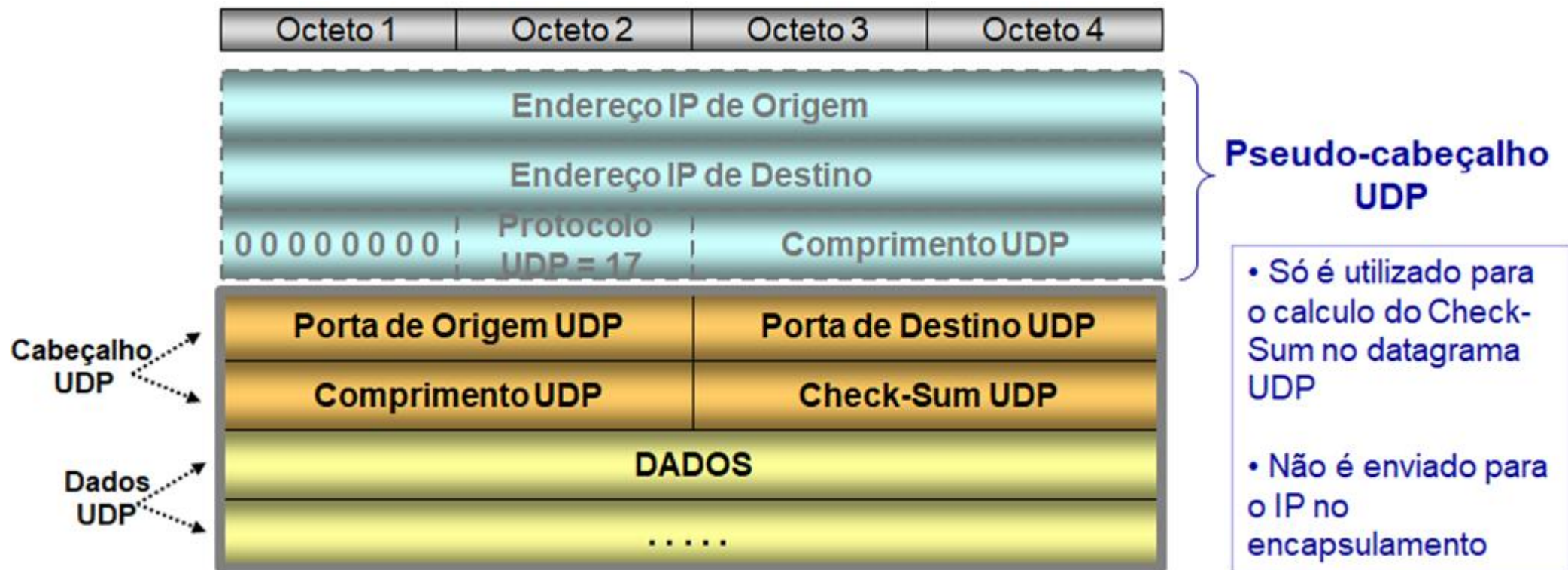
Serviços providos pela camada de transporte

- ✓ 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**.



✓ 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.
- Comprimento: inclui cabeçalho (8 bytes) e dados, com tamanho máximo de 65515 bytes.
- Check-Sum: soma de verificação do segmento UDP (cabeçalho + pseudo-cabeçalho + dados).

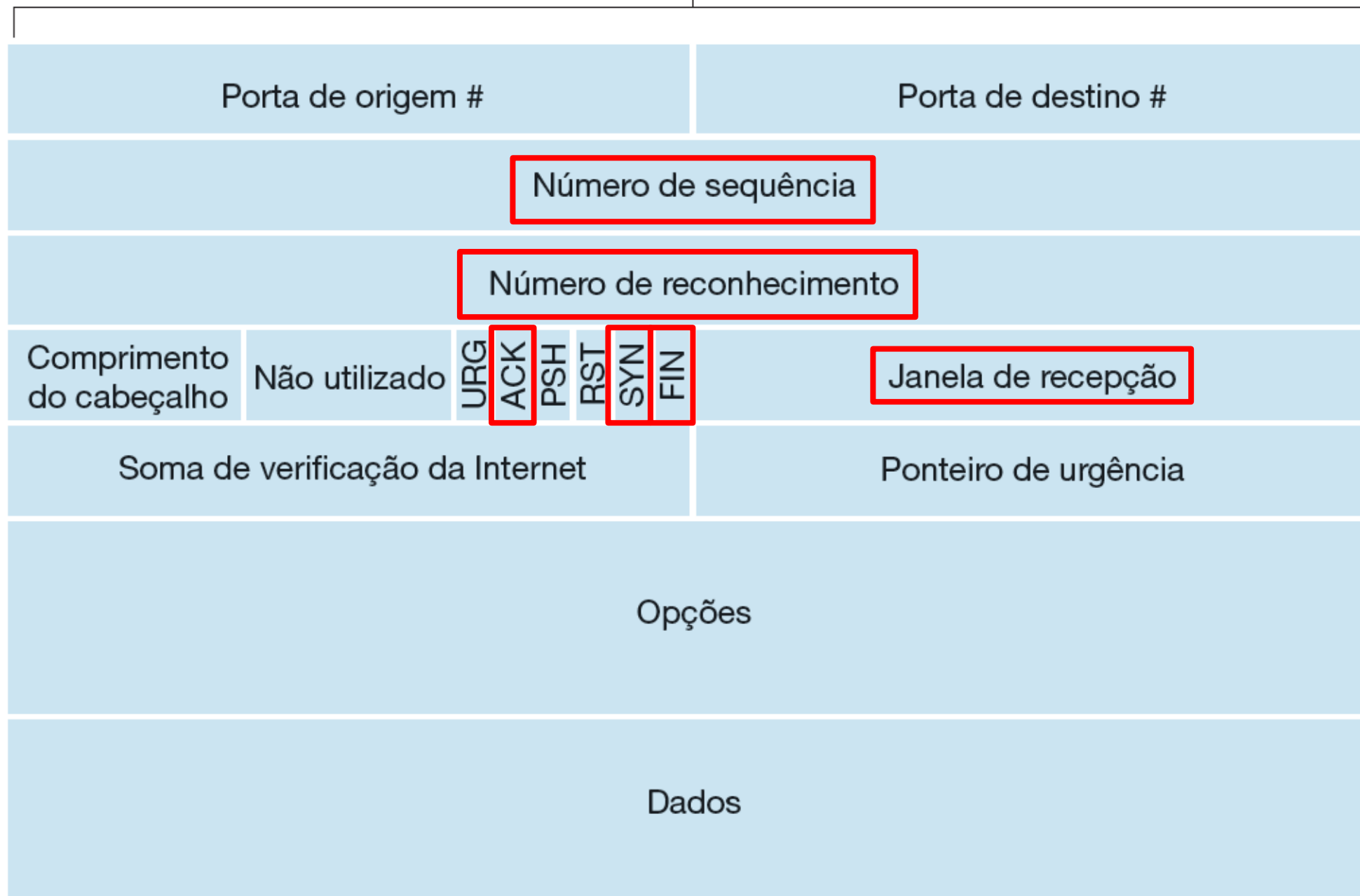


✓ Principais características:

- Protocolo de transporte simplificado.
 - Cabeçalho mais simples – apenas 8 bytes.
- Não orientado à conexão.
 - Não há estabelecimento de conexão antes que os dois processos comecem a se comunicar.
- Não confiável.
 - Quando um processo envia uma mensagem para um socket UDP, o protocolo não oferece garantias de que a mensagem chegará ao processo receptor.
 - Ainda, mensagens que chegam de fato ao processo receptor podem chegar fora de ordem.
- Implementa a multiplexação/demultiplexação de processos, assim como o TCP.
- Não inclui mecanismos de controle de congestionamento.
- Não inclui mecanismos de controle de fluxo.
- Não inclui mecanismos de controle erro e de retransmissão.

✓ Formato do segmento:

32 bits

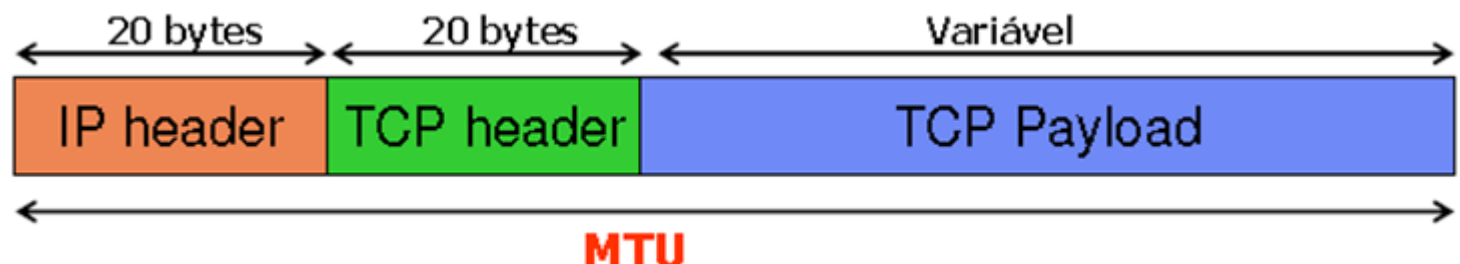


✓ Formato do segmento:

- 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

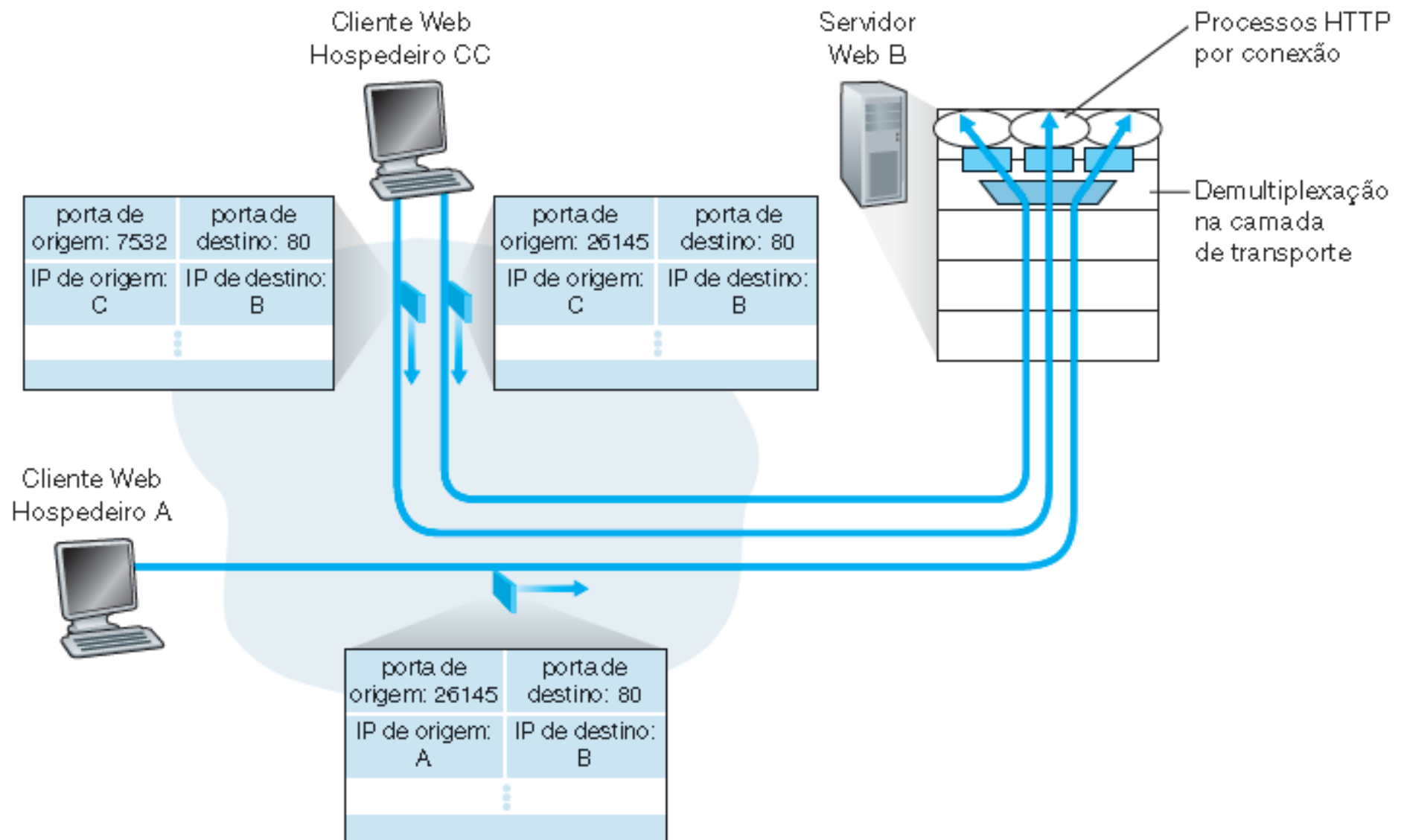
✓ Formato do segmento:

- 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).

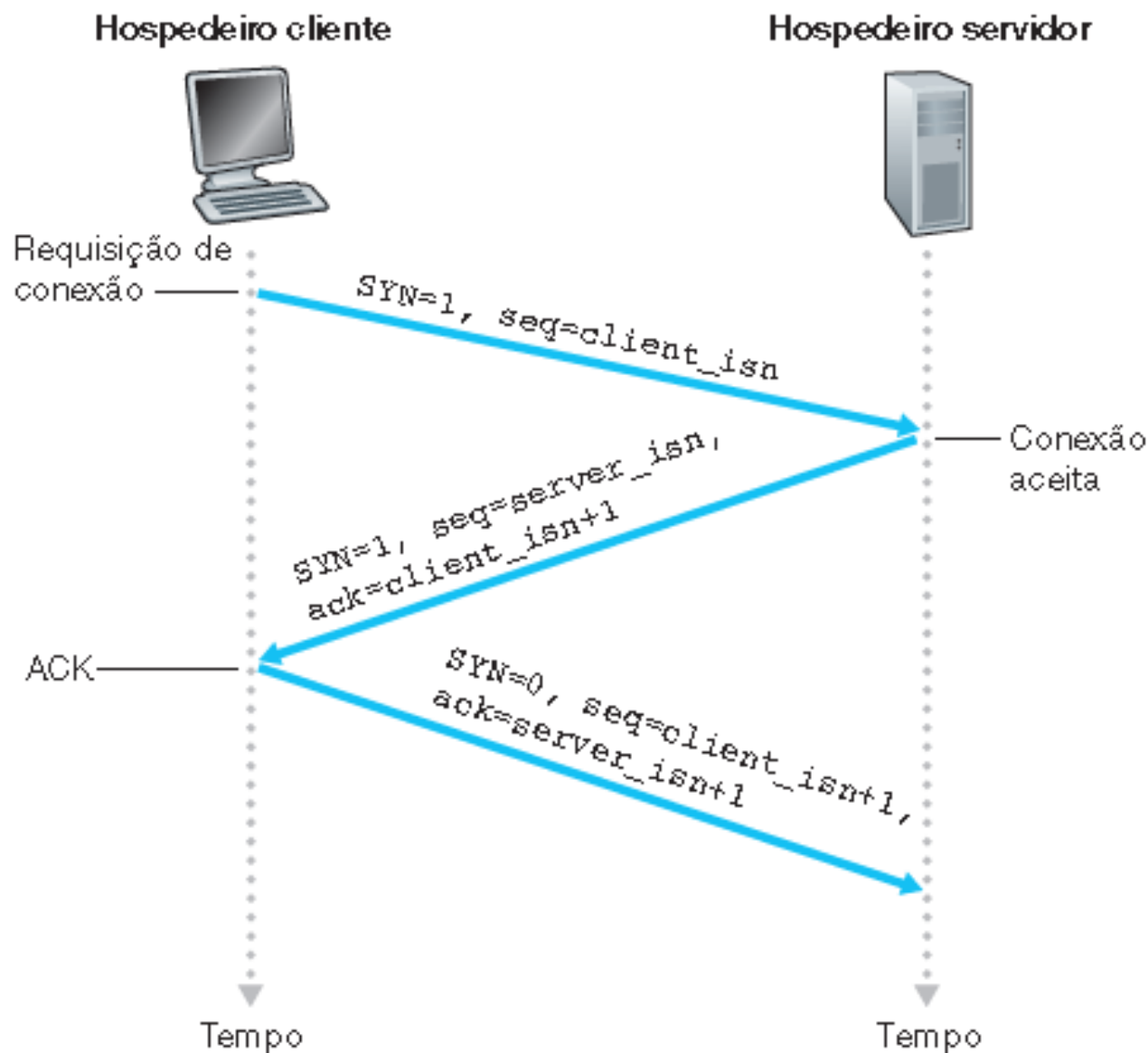


- ✓ Multiplexação/Demultiplexação → TCP e UDP
 - Conceito associado à capacidade de ampliação do serviço de entrega hospedeiro a hospedeiro provido pela camada de rede para um serviço de entrega processo a processo para aplicações que rodam nesses hospedeiros.
 - O trabalho de reunir, no hospedeiro de 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 diferentes segmentos, e passar esses segmentos para a camada de rede é denominada multiplexação
 - A tarefa de entregar os dados contidos em um segmento da camada de transporte ao socket correto é denominada demultiplexação.

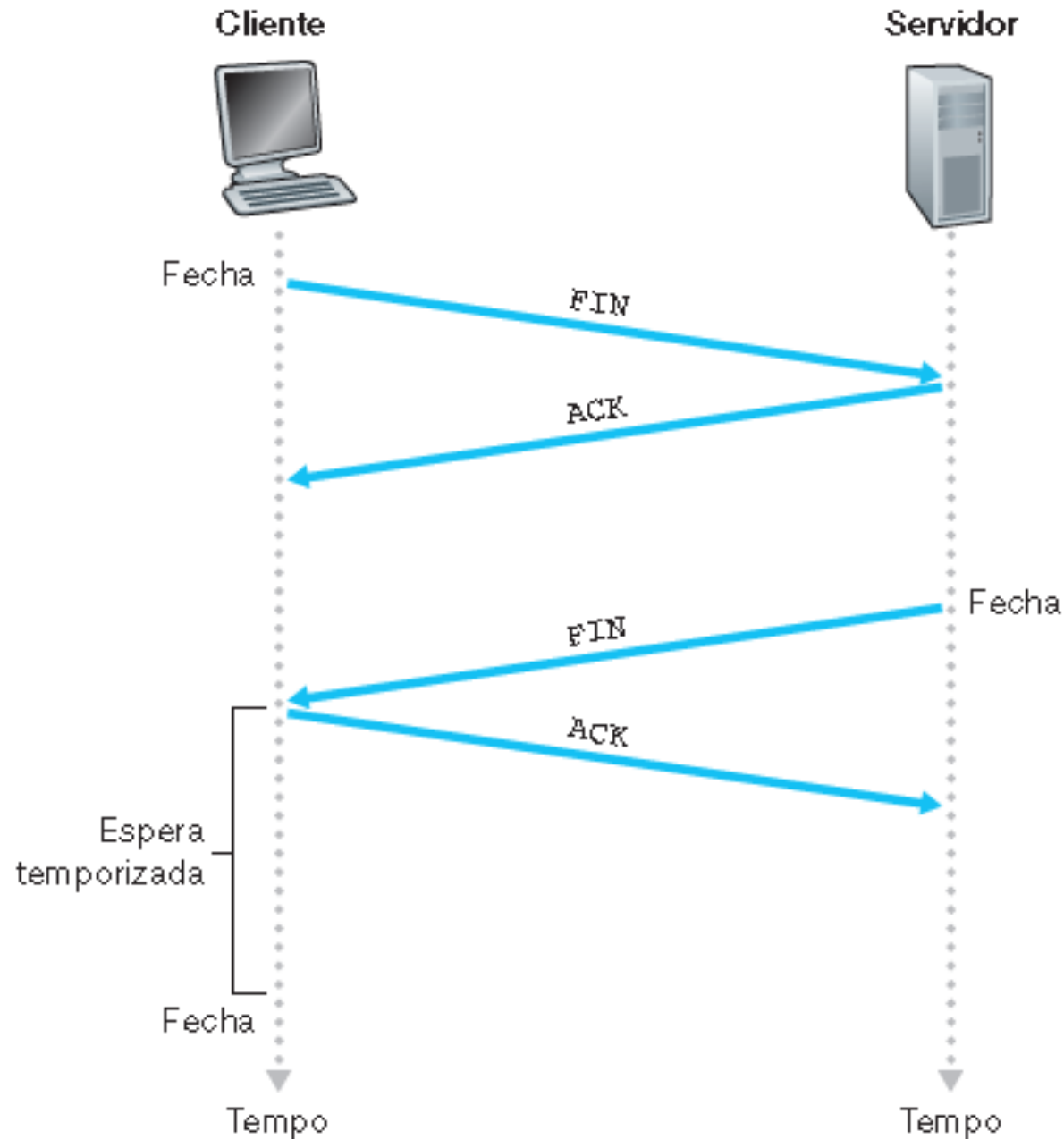
✓ Multiplexação/Demultiplexação – TCP e UDP



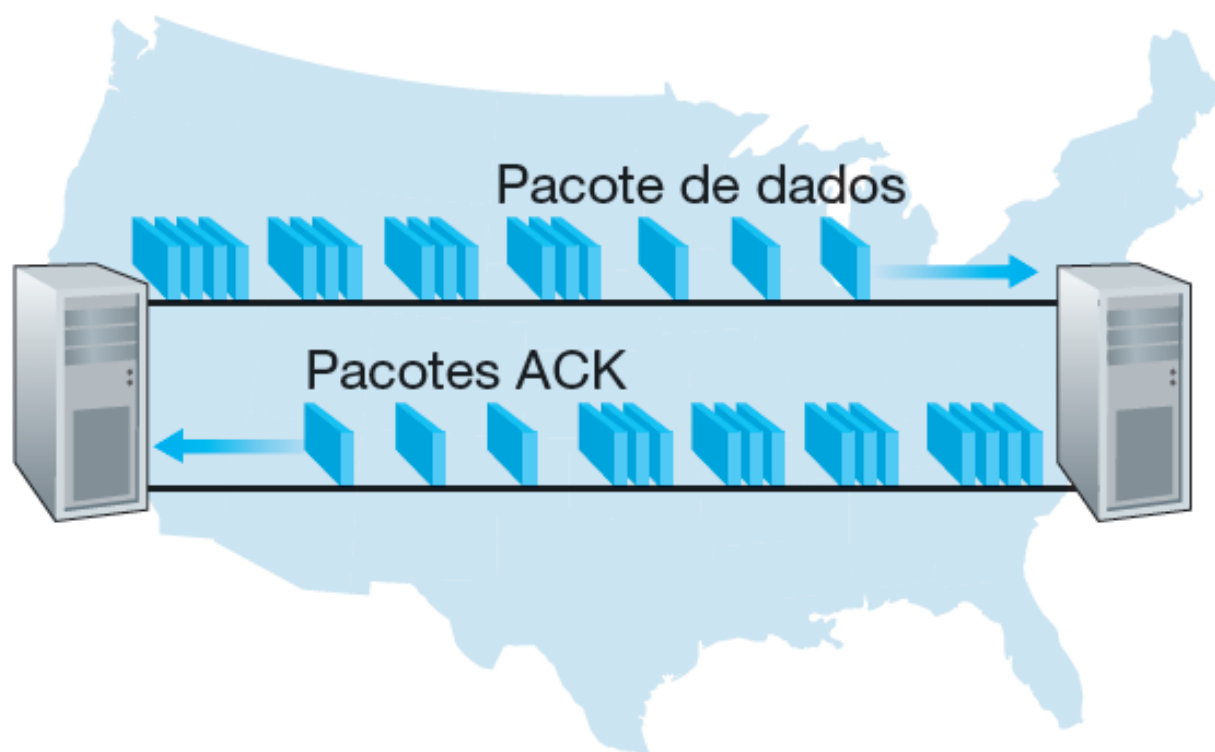
- ✓ Estabelecimento de conexão → TCP
 - 3-way handshake (apresentação de três vias).



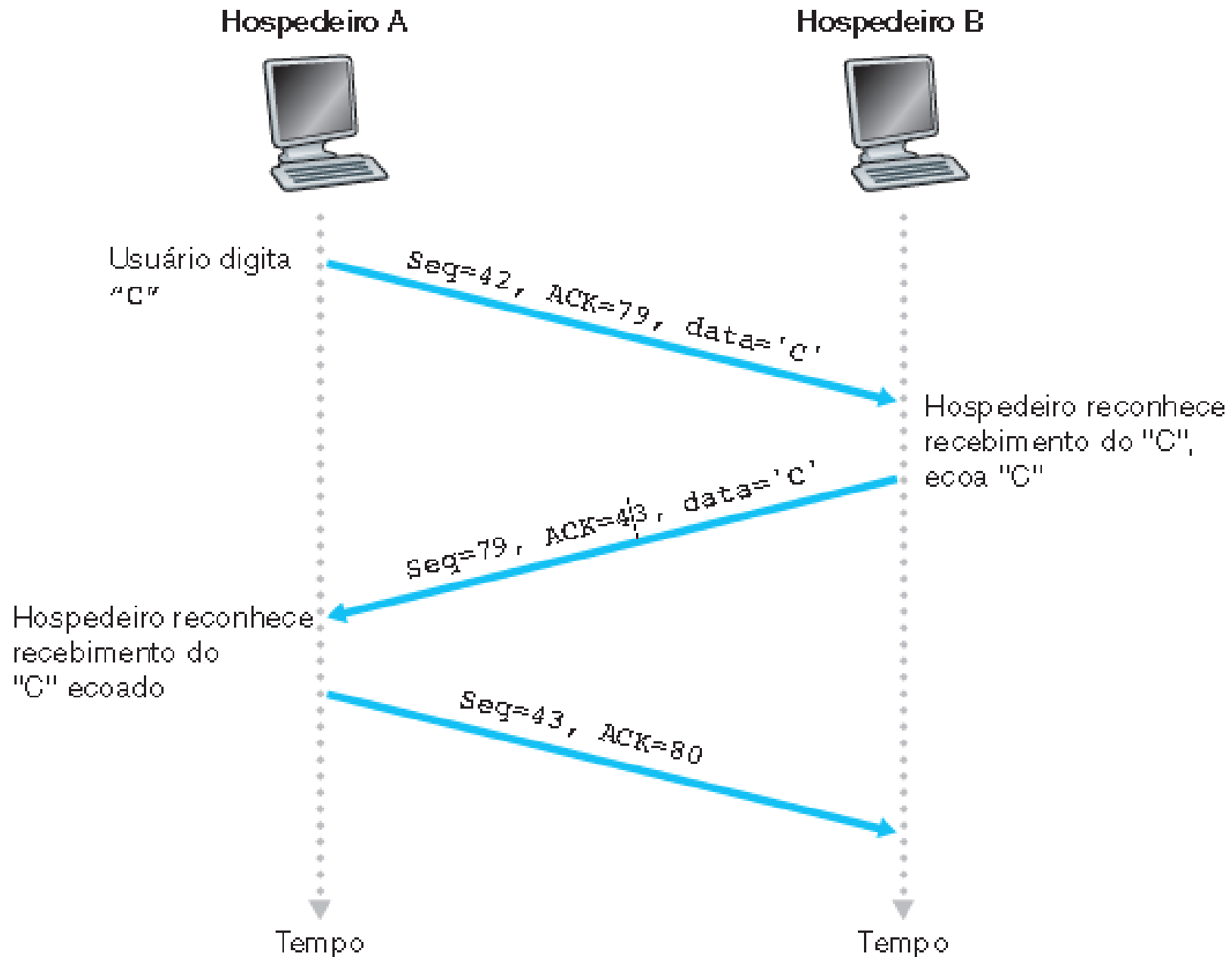
- ✓ Encerramento de conexão → TCP



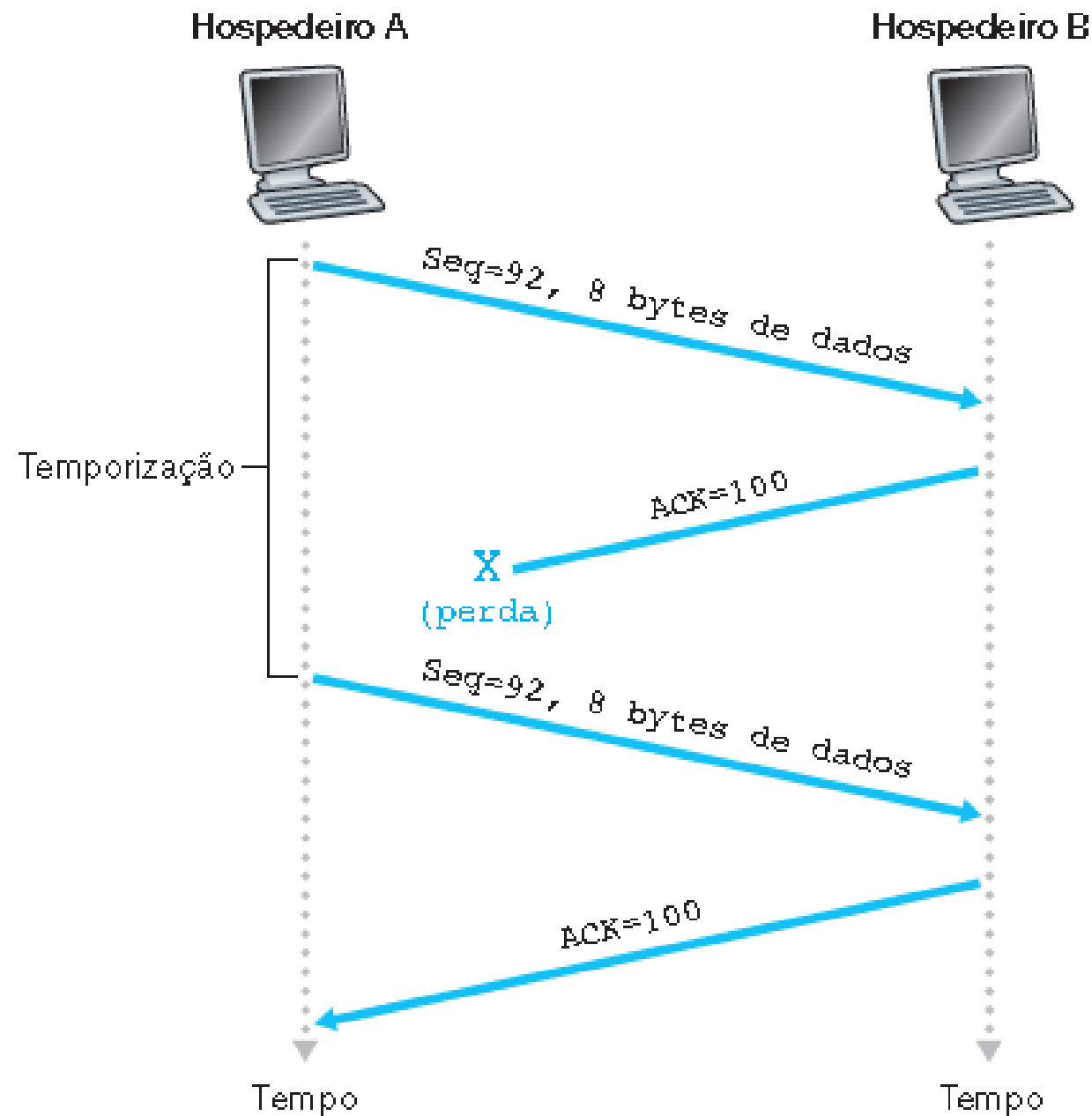
- ✓ Controle de erro e retransmissão → TCP
 - Tem por objetivo fazer com que dois sistemas troquem dados de maneira confiável
 - Cada segmento de dados transmitido é identificado por um “número de segmento”; e conforme esses segmentos vão chegando ao sistema destino, este pode confirmar seu recebimento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



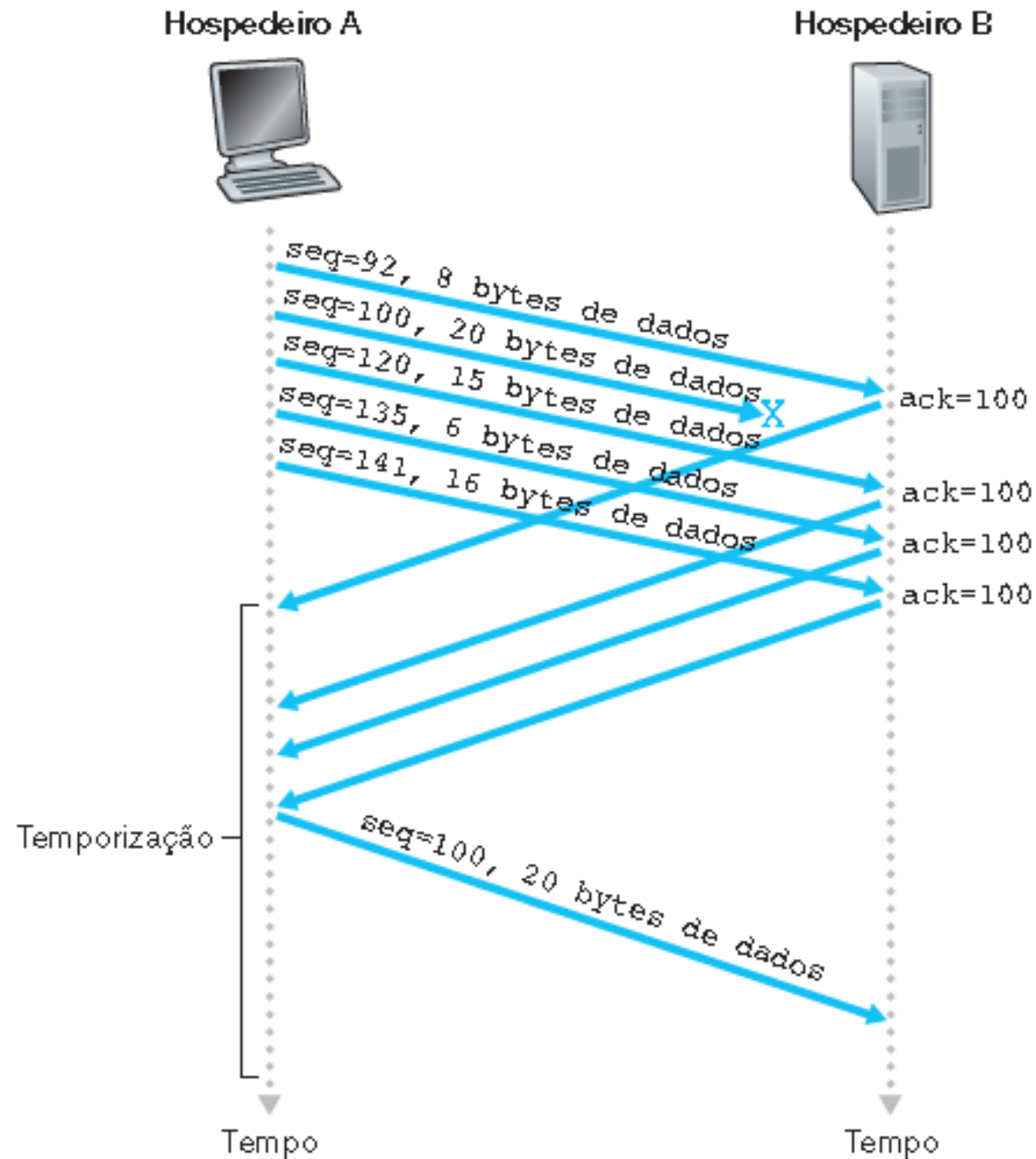
✓ Reconhecimento e retransmissões → TCP



- ✓ Reconhecimento e retransmissões → TCP



- ✓ Reconhecimento e retransmissões → TCP



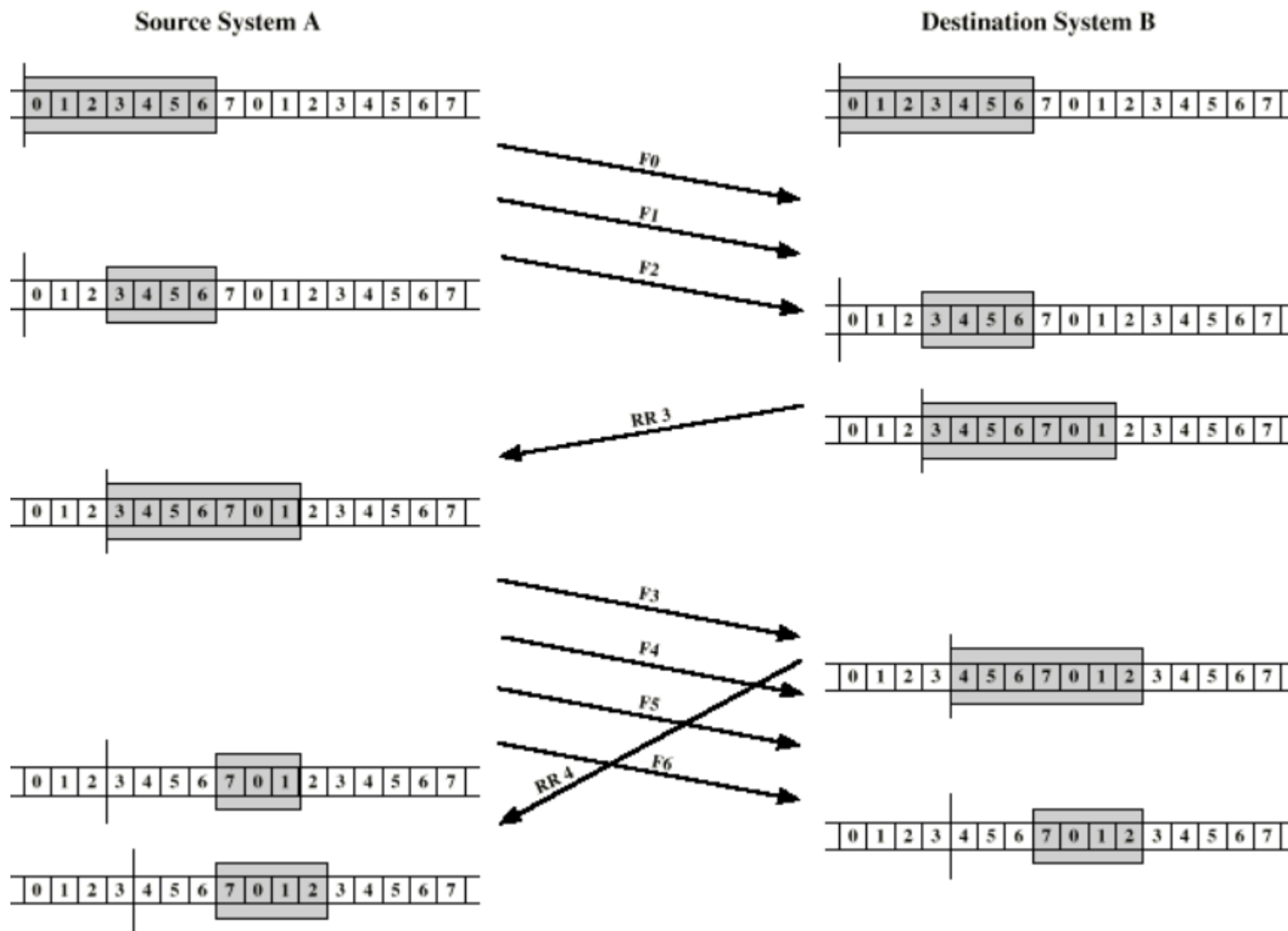
✓ Controle de retransmissão → TCP

- 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.

- ✓ Controle de fluxo → TCP
 - Tem como objetivo compatibilizar as taxas nas quais a aplicação remetente envia os dados e a aplicação receptora faz sua leitura nos buffers de recepção.
 - Visa portanto evitar a possibilidade de o remetente estourar a capacidade do buffer de recepção do destinatário.
 - O TCP *remetente* mantém uma variável denominada **janela de recepção (rwnd)**, usada para dar a ideia do espaço de buffer livre disponível no destinatário.
- ✓ Assim, a ideia básica é identificar a quantidade de dados que podem ser transmitidos sem a recepção de uma confirmação.
- ✓ Esgotou o tamanho da janela, o remetente interrompe o envio de dados e aguarda a chegada de uma confirmação, para poder reiniciar o envio de dados.

✓ Janela deslizante (sliding-window protocol) → TCP

- Exemplo: tamanho de janela = 7



✓ Controle de congestionamento → TCP

- Tem como objetivo evitar que um ponto específico na rede, por exemplo um roteador ou um enlace, possa receber mais pacotes do que sua capacidade de tratá-los.
- Consequências do congestionamento: atrasos na entrega dos pacotes ou possíveis perdas de pacotes causando a necessidade de retransmissão dos mesmos, provocando assim mais congestionamento na rede.
- Controle de congestionamento fim a fim:
 - A camada de rede não fornece nenhuma informação explícita à camada de transporte, com relação ao congestionamento.
 - Os sistemas finais percebem o problema observando o comportamento da rede (atrasos e perdas de pacotes).

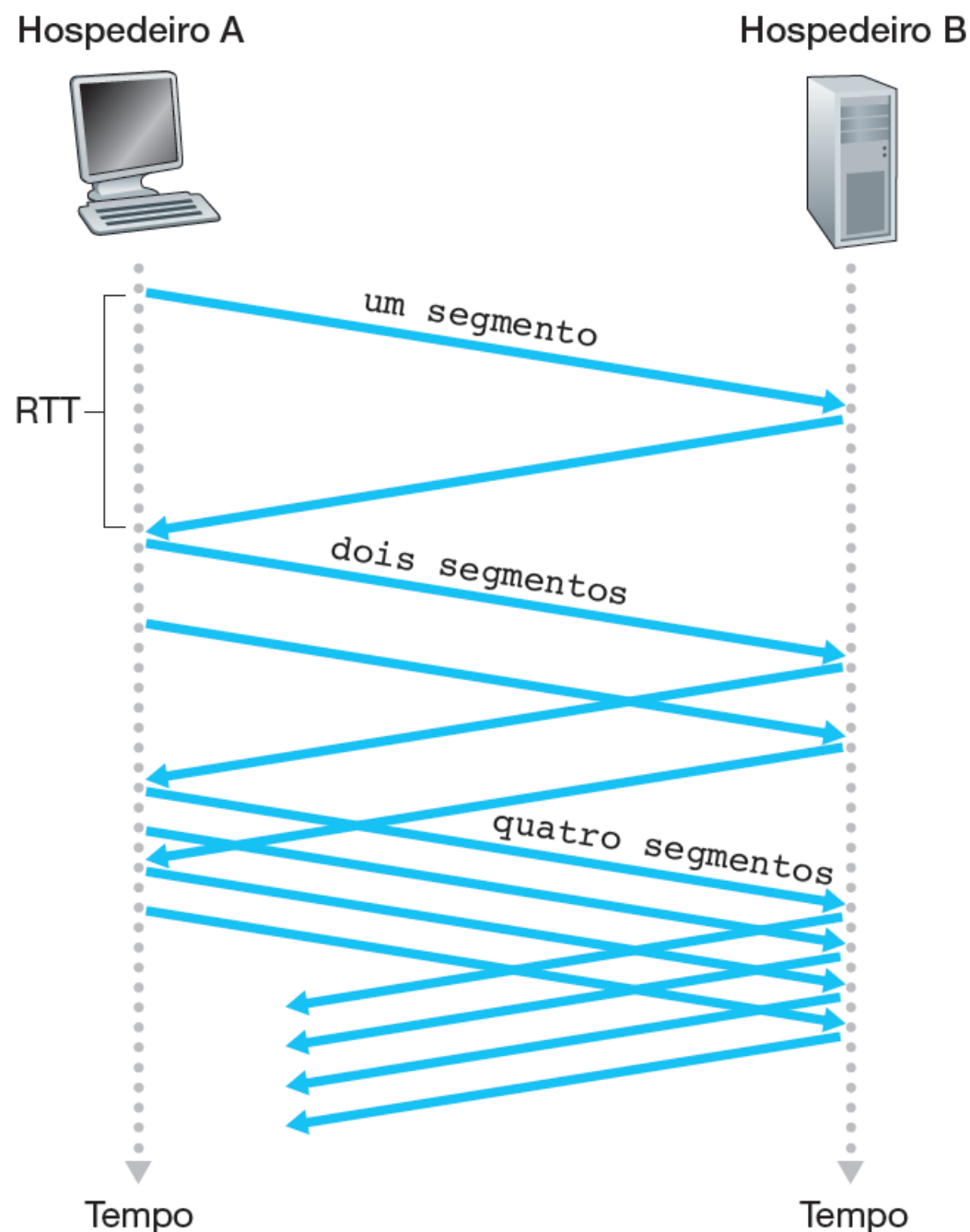
✓ Controle de congestionamento → TCP

- A abordagem adotada pelo TCP é obrigar cada remetente a limitar a taxa à qual enviam os dados (*) para sua conexão como uma função do congestionamento de rede percebido.
 - Quando há congestionamento, existe um aumento no atraso no RTT, podendo ainda acontecer perda de pacotes.
 - Se o remetente TCP perceber que há pouco congestionamento no caminho entre ele e o destinatário, aumenta sua taxa de envio (*).
 - Se perceber que há congestionamento, reduz sua taxa de envio (*).
- Basicamente, pode-se dividir o controle de congestionamento do TCP em três etapas combinadas [RFC 5681(set/2009): TCP Congestion Control]:
 - Partida lenta.
 - Prevenção de congestionamento.
 - Recuperação rápida.

* campo “janela de recepção”, no cabeçalho TCP.

✓ Partida Lenta

- Quando uma conexão TCP começa, o valor do tamanho da janela de transmissão (cwnd – congestion window) é inicializado em 1 MSS, e aumenta 1 MSS toda vez que um segmento transmitido é reconhecido.
- Esse processo resulta em uma multiplicação da taxa de envio a cada RTT.
- Assim, durante a fase de partida lenta, a taxa de envio TCP se inicia lenta, mas cresce exponencialmente.



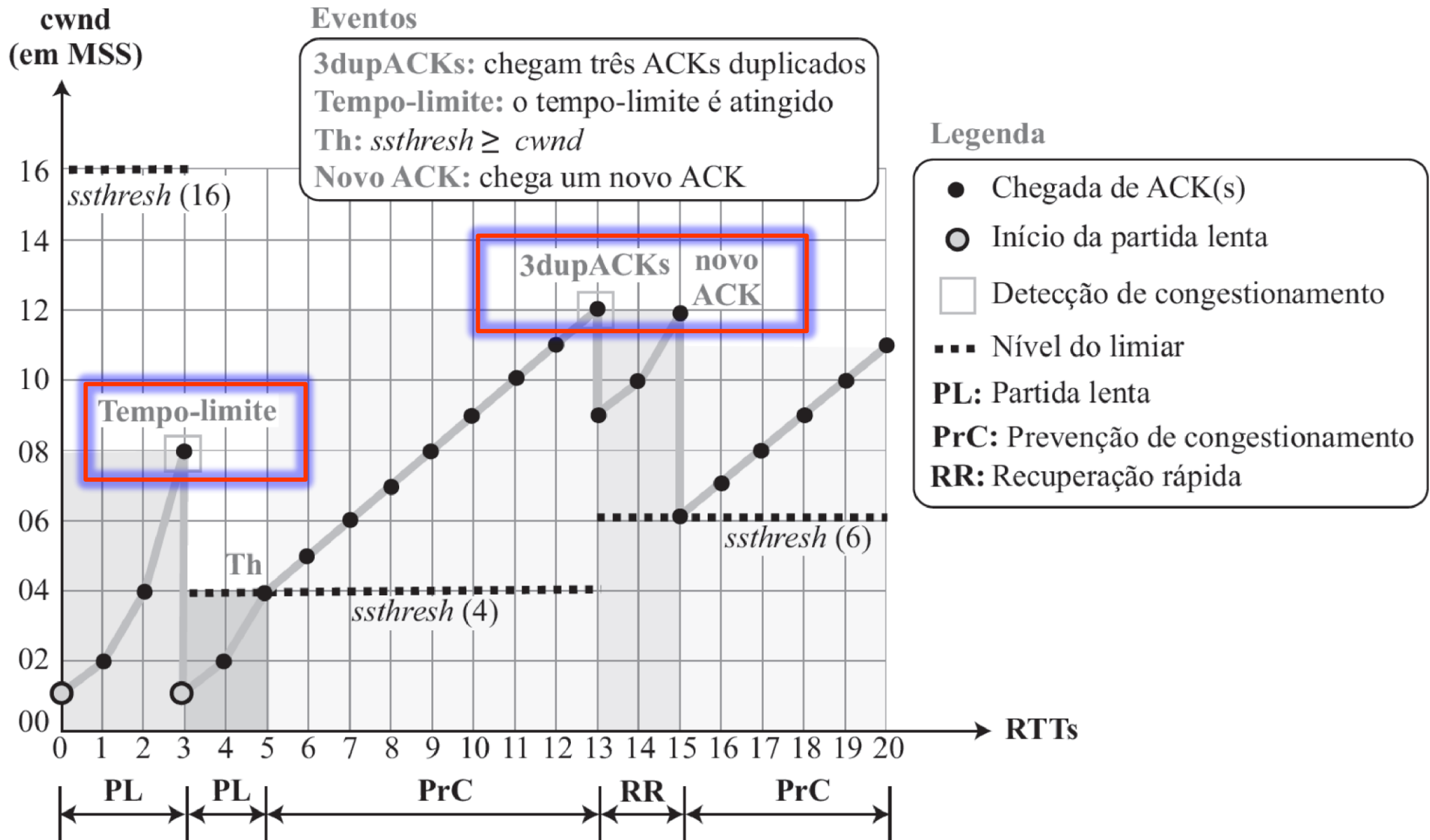
✓ Prevenção de congestionamento

- O valor de `cwnd` (tamanho da janela) dobra até atingir um limiar de partida lenta, chamado `ssthresh` (*slow start threshold*).
 - Este limiar de partida lenta equivale ao valor do `cwnd/2`, ou seja, a metade do valor da taxa de envio quando um congestionamento foi detectado pela última vez
- A partir daí, o valor `cwnd` é incrementado de um MSS a cada RTT.
- Utilizado quando um timeout de um segmento é detectado.

✓ Recuperação rápida

- Ao entrar no modo recuperação rápida, o valor de `cwnd` é aumentado de 1 MSS para cada ACK duplicado recebido no segmento perdido que fez o TCP entrar nesse modo.
- Quando o TCP recebe o ACK correspondente ao segmento perdido, retorna ao modo prevenção de congestionamento.
- Utilizado quando ACKs duplicados de um segmento é detectado.

✓ Controle de congestionamento no TCP:



Atividade 03 – TCP e UDP