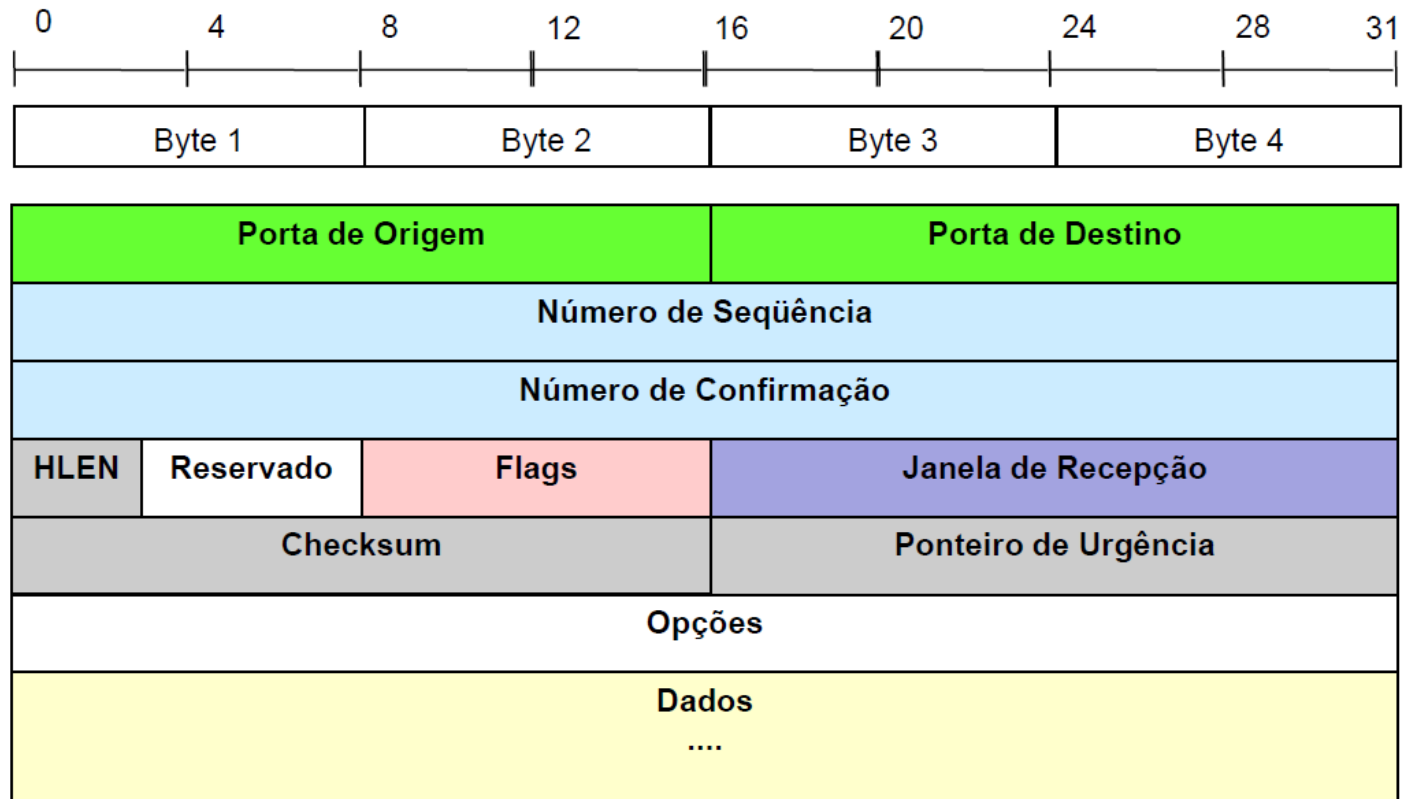


Protocolos de Comunicação

TCP - Detalhamento

Protocolo TCP

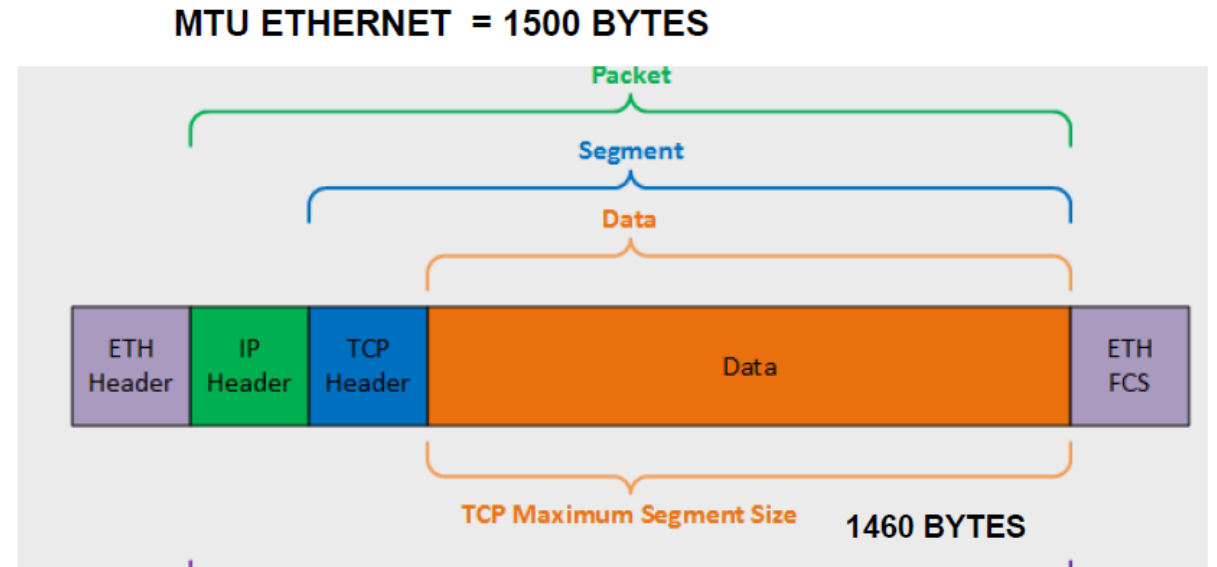
- DETALHAMENTO



- FLAGS TCP: ACK , SYN , FIN , RST , URG, CWR, ECN, PUSH

Protocolo TCP

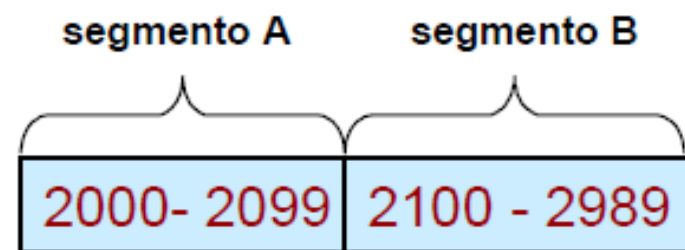
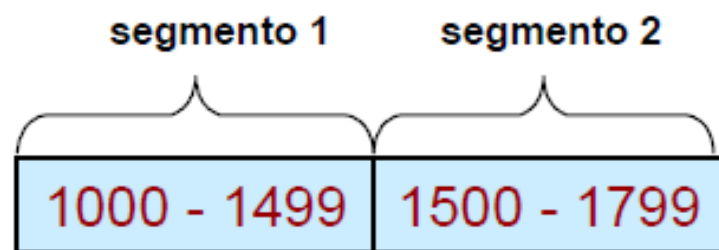
- MAXIMUM SEGMENT SIZE (MSS)
- MSS = quantidade máxima de dados transportada por um segmento
- MTU = Maximum Transmission Unit (camada de enlace)



OVERHEAD: Cabeçalho IP + Cabeçalho TCP (tamanho = 40)

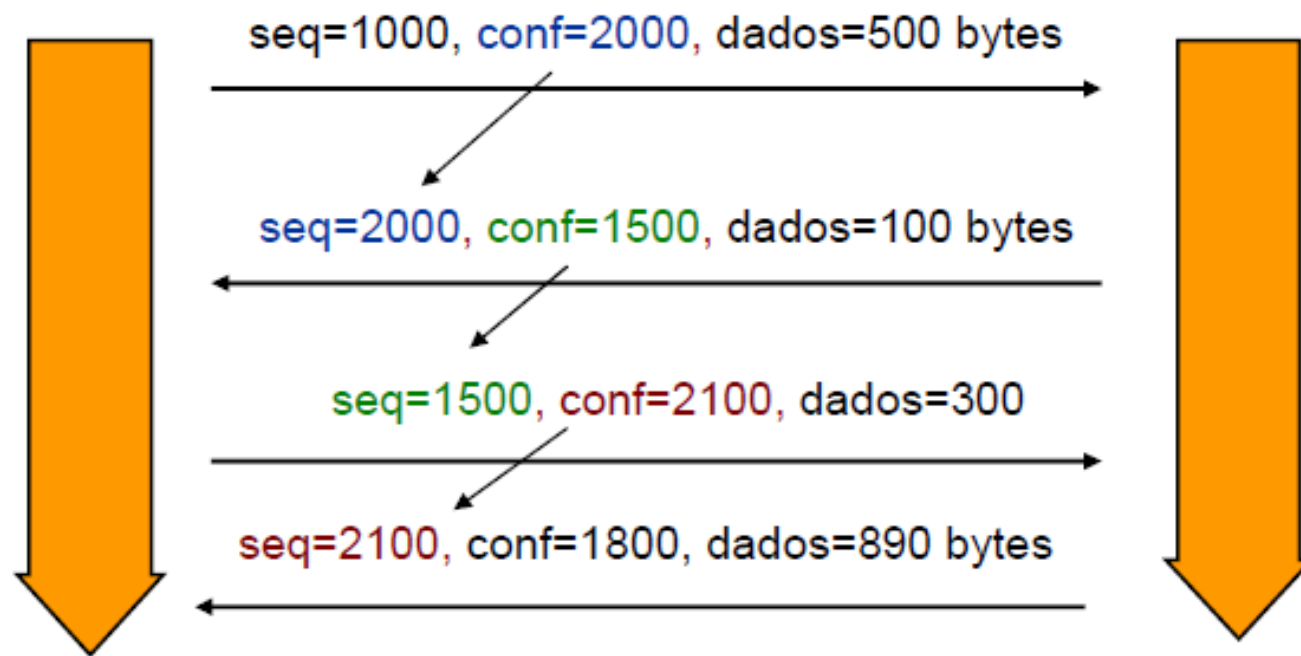
Protocolo TCP (transmissão por fluxo)

- NÚMERO INICIAL DE SEQUÊNCIA (NIS)
 - Valor escolhido de forma aleatória para iniciar a numeração do primeiro byte de uma conexão
 - É alterado a cada conexão
 - É unidirecional (de A-para-B é diferente de B-para-A)
- NÚMERO DE SEQUÊNCIA (NS)
 - Numeração do primeiro byte de cada segmento, sendo o primeiro valor é igual ao NIS
 - Atualizado a cada transmissão de modo a sempre informar número de sequência do primeiro byte contido no segmento



cliente

servidor



Protocolo TCP (controle de conexão)

Flags ACK, SYN e FIN

- ACK: confirmação de recebimento
 - 0 no primeiro segmento transmitido (não há nada a confirmar)
 - 1 nos demais
- SYN: sincronismo
 - 1 nos dois primeiros segmentos trocados entre o cliente e o servidor
 - 0 nos demais
- FIN: finalização
 - 1 indica o desejo de encerrar a conexão

Protocolo TCP

Controle de conexão

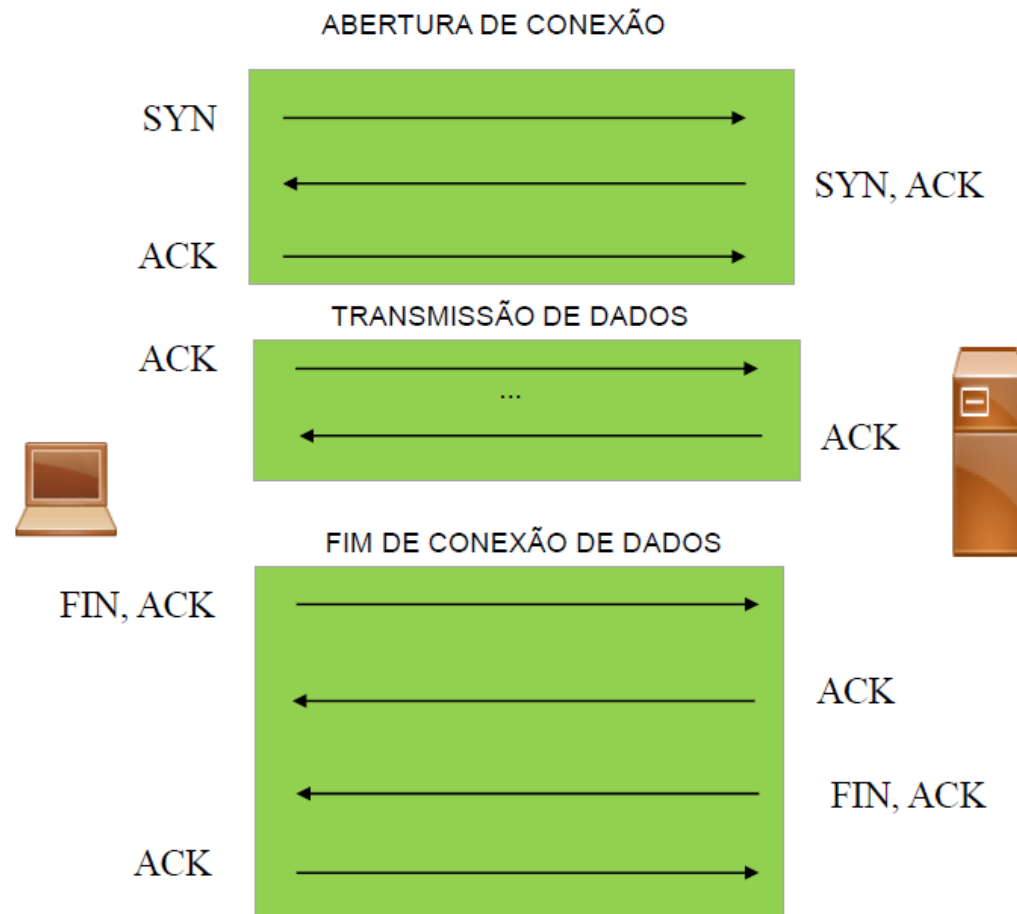
A conexão é controlada por flags:
(campos de 1 bit no cabeçalho do TCP).

ACK: estou confirmando algo

SYN: quero sincronizar

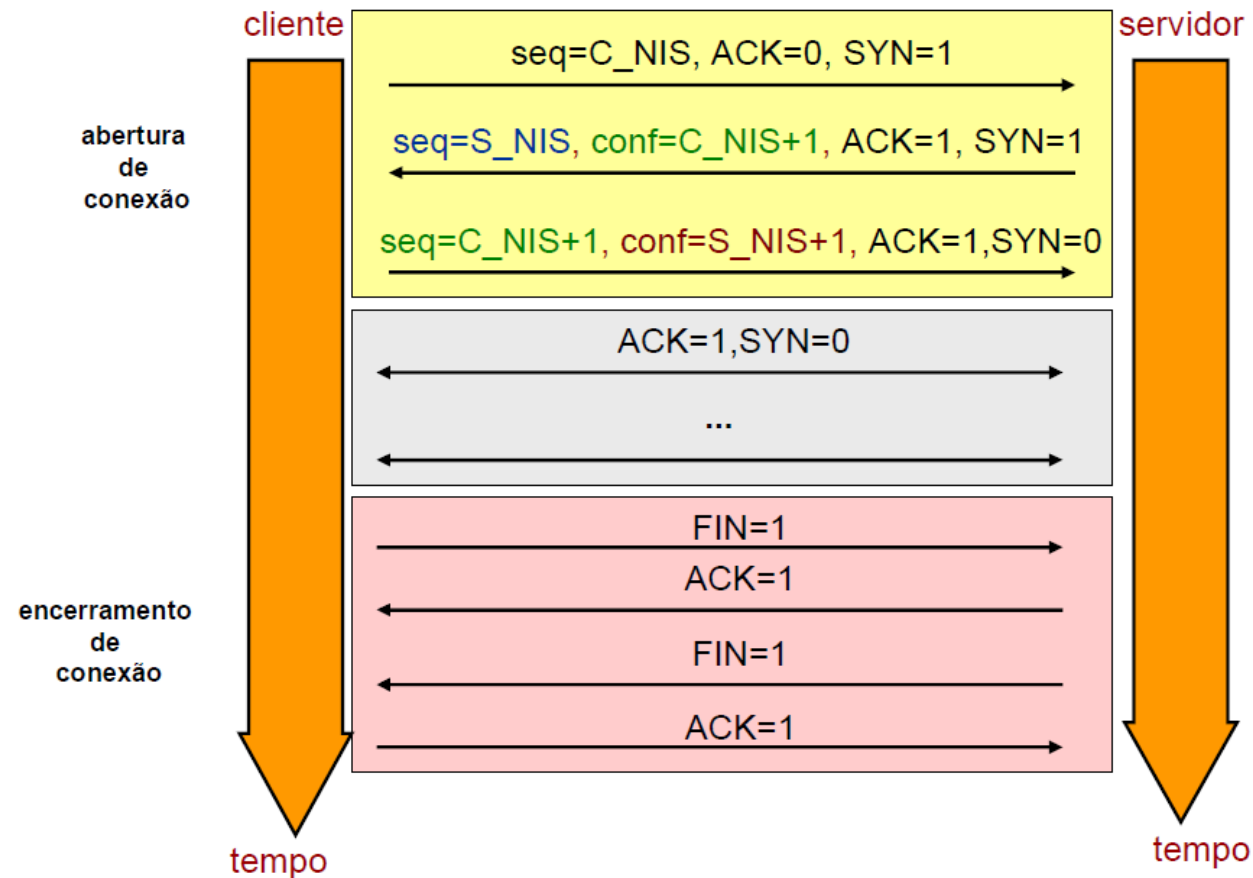
FIN: quero finalizar

RST: pacote rejeitado.
Essa conexão NÃO EXISTE!!!



Protocolo TCP

Controle de conexão



ABERTURA

1. Pedido de abertura de conexão ($\text{SYN}=1$ $\text{ACK}=0$) - define o valor inicial do NIS do cliente (C_NIS)
2. Confirmação da abertura da conexão ($\text{SYN}=1$ $\text{ACK}=1$) - define o valor inicial do NIS do servidor (S_NIS)
3. Confirmação do recebimento da confirmação ($\text{SYN}=1$ $\text{ACK}=1$).

TROCA DE DADOS

- Dados podem ser trocados indefinidamente entre o cliente e o servidor ($\text{ACK}=1$ e $\text{SYN}=0$)

ENCERRAMENTO

- Envolve a troca dos quatro segmentos com $\text{FIN}=1$ e $\text{ACK}=1$
- Qualquer um dos lados pode iniciar a sequência

QUIZ 1. Transmissão por fluxo

1. No TCP, todos os bytes são numerados a partir do início da conexão. O campo NS (Número de Sequência) do cabeçalho TCP traz o número do primeiro byte contido no segmento.

A. Verdadeiro ***

B. Falso

2. Imagine que uma conexão caiu, e foi reestabelecida automaticamente pelas aplicações usando as mesmas portas. O que poderia acontecer se o TCP numerasse os bytes de uma conexão sempre a partir de zero, ao invés de utilizar um número aleatório. Assinale a alternativa correta.

A. Pacotes em trânsito da conexão antiga poderiam ser inseridos fora de sequência na nova conexão. ****

B. Nada de ruim. Não é possível recriar uma conexão com as mesmas portas, pois a porta do cliente é aleatória.

QUIZ 2. Controle de conexão TCP

1. No TCP, o número inicial de sequência é o mesmo nos dois sentidos da conexão.

A. Verdadeiro

B. Falso ****

2. Sobre o encerramento da conexão, indique a afirmativa correta.

A. A conexão é sempre encerrada por iniciativa do cliente.

B. Quando o cliente encerra a conexão, o servidor não pode mais transmitir pacotes, e deve encerrar a conexão imediatamente.

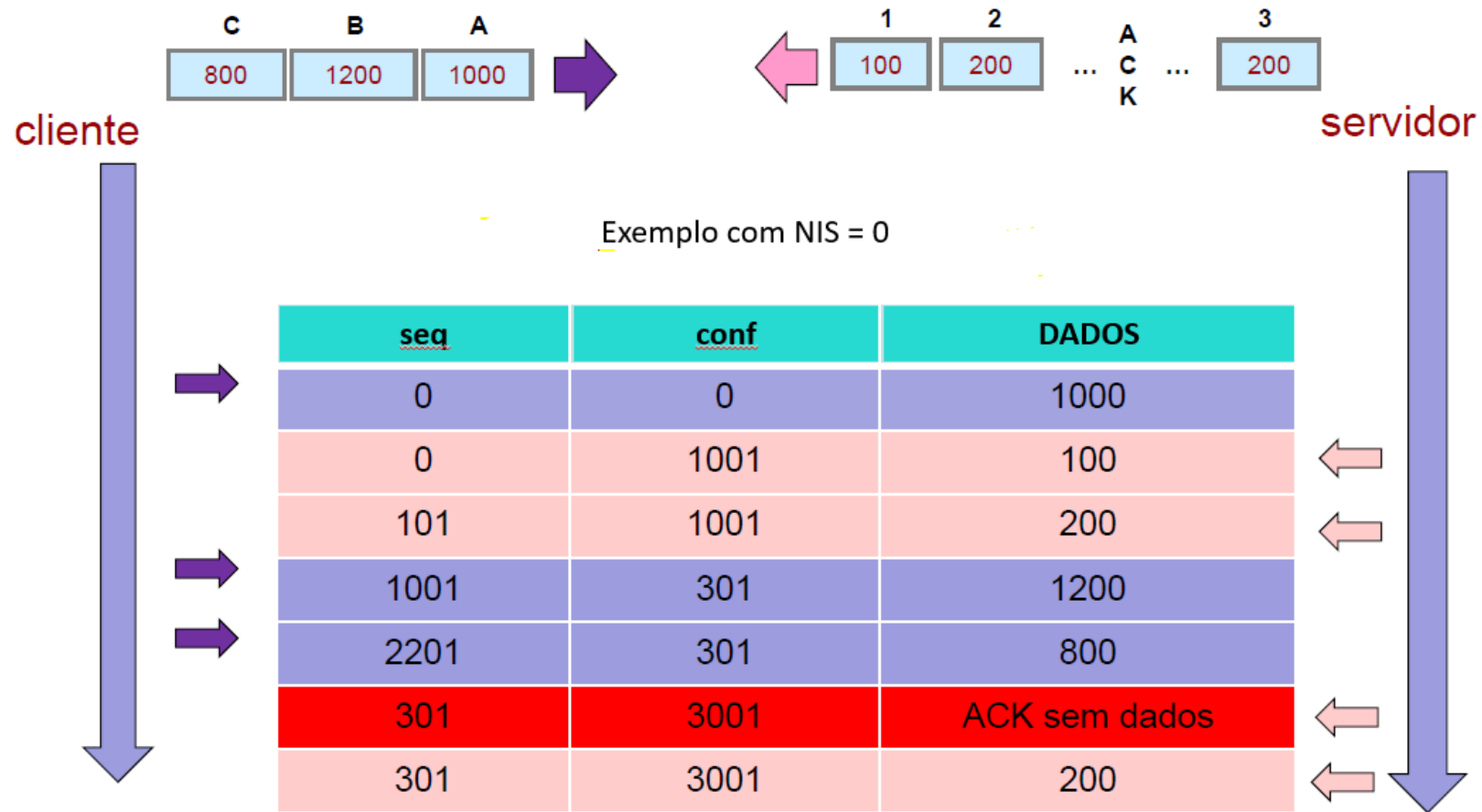
C. Quando o cliente encerra a conexão o S.O. do servidor aguarda que sua aplicação sinalize que não tem mais dados para transmitir. ***

Protocolo TCP (comunicação confiável)

Retransmissão de pacotes perdidos

- TCP não usa mensagens de erro
- Técnica de retransmissão do TCP: reconhecimento positivo com temporizadores
- Se uma confirmação de recebimento não chegar no transmissor num **tempo pré-determinado**, o segmento é retransmitido
- O receptor pode enviar pacotes sem dados, apenas com confirmação, quando não tem nada para transmitir

Protocolo TCP (comunicação confiável)



Protocolo TCP (comunicação confiável)

Tempo máximo para aguardar uma confirmação (Temporizador)

- Estimado em função do RTT (round trip time)
 - Cálculo do valor do temporizador (quando recebe a primeira mensagem de confirmação)
$$\text{EstimativaRTT} = \text{AmostraRTT}$$
$$\text{Temporizador} = \text{EstimativaRTT}$$
 - Atualização do temporizador (a cada confirmação recebida)
$$\text{EstimativaRTT} = 0.875 \text{ EstimativaRTT} + 0.125 \text{ AmostraRTT}$$
$$\text{Desvio} = 0.875 \text{ Desvio} + 0.125 (\text{AmostraRTT} - \text{EstimativaRTT})$$
$$\text{Temporizador} = \text{EstimativaRTT} + 4 \cdot \text{Desvio}$$
- O estouro do temporizador caracteriza uma falha de transmissão
 - O segmento precisa ser retransmitido

AmostraRTT: última medição de RTT Desvio: medida da flutuação do valor do RTT (iniciado em 0)

QUIZ 3. Comunicação confiável TCP

1. Como é feita a detecção de pacotes perdidos ?

- A. O receptor envia um NACK quando não recebe um pacote.
- B. O TCP usa o método de reconhecimento positivo com temporizadores. ***

2. Indique a resposta correta

- A. O tempo máximo para aguardar uma confirmação depende da taxa de transmissão.
- B. O tempo máximo para aguardar uma confirmação depende da distância entre os computadores (tempo de propagação).
- C. O tempo máximo para aguardar uma confirmação depende da taxa de transmissão, da distância, mas não varia durante a conexão.
- D. O tempo máximo para aguardar uma confirmação depende da taxa de transmissão, da distância, e é afetado pelo congestionamento da rede. *****

Protocolo TCP

Controle de fluxo

Um segmento TCP (recebido pelo SO) é armazenado em um buffer (com capacidade limitada) de forma transparente para a aplicação

O receptor deve ser capaz de remover os dados do buffer numa velocidade compatível com a taxa de recebimento

Se a aplicação receptora não der conta de ler os dados do buffer e este se encher por completo, novos segmentos recebidos são descartados

O **TRANSMISSOR (A)** precisa estimar o espaço livre no buffer do S.O. do **RECEPTOR (B)**.

O espaço livre no buffer de **B** é enviado junto com cada confirmação (**RcvWindow**), mas ela não é absoluta pois podem haver pacotes em trânsito.

Então **A** calcula o espaço livre em **B** pela fórmula:

$$\text{RcvBuffer} = \text{RcvWindow} - [\text{LastByteSent} - \text{LastByteRcvd}]$$

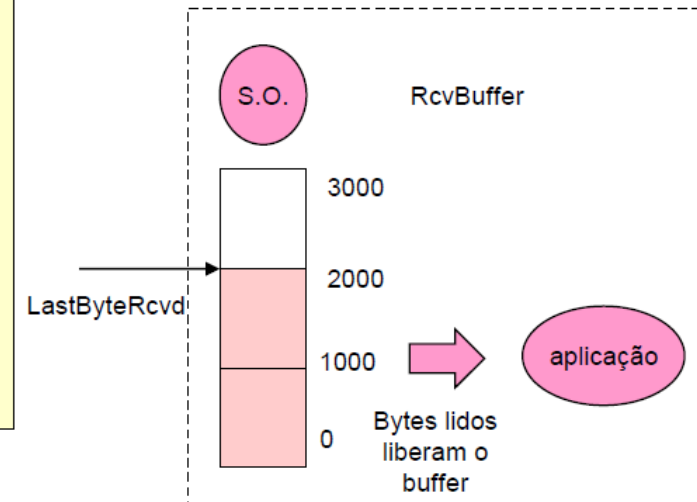
RcvBuffer = estimativa do buffer livre em B

RcvWindow = buffer livre informado por B

LastByteSent = último byte enviado por A

LastByteRcvd = último byte confirmado por B (**NC-1**)

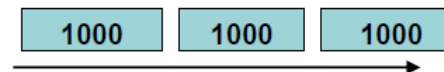
RcvWindow: janela de recepção enviada no pacote TCP



Quantidade de bytes que pode ser transmitida sem confirmação é menor que RcvBuffer

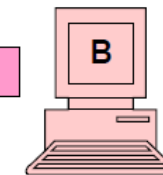


Transmissor



NC=2001, RcvWindow=1000

$$\text{RcvBuffer} = 1000 - (3000 - 2000)$$



Receptor

Protocolo TCP (controle de congestionamento)

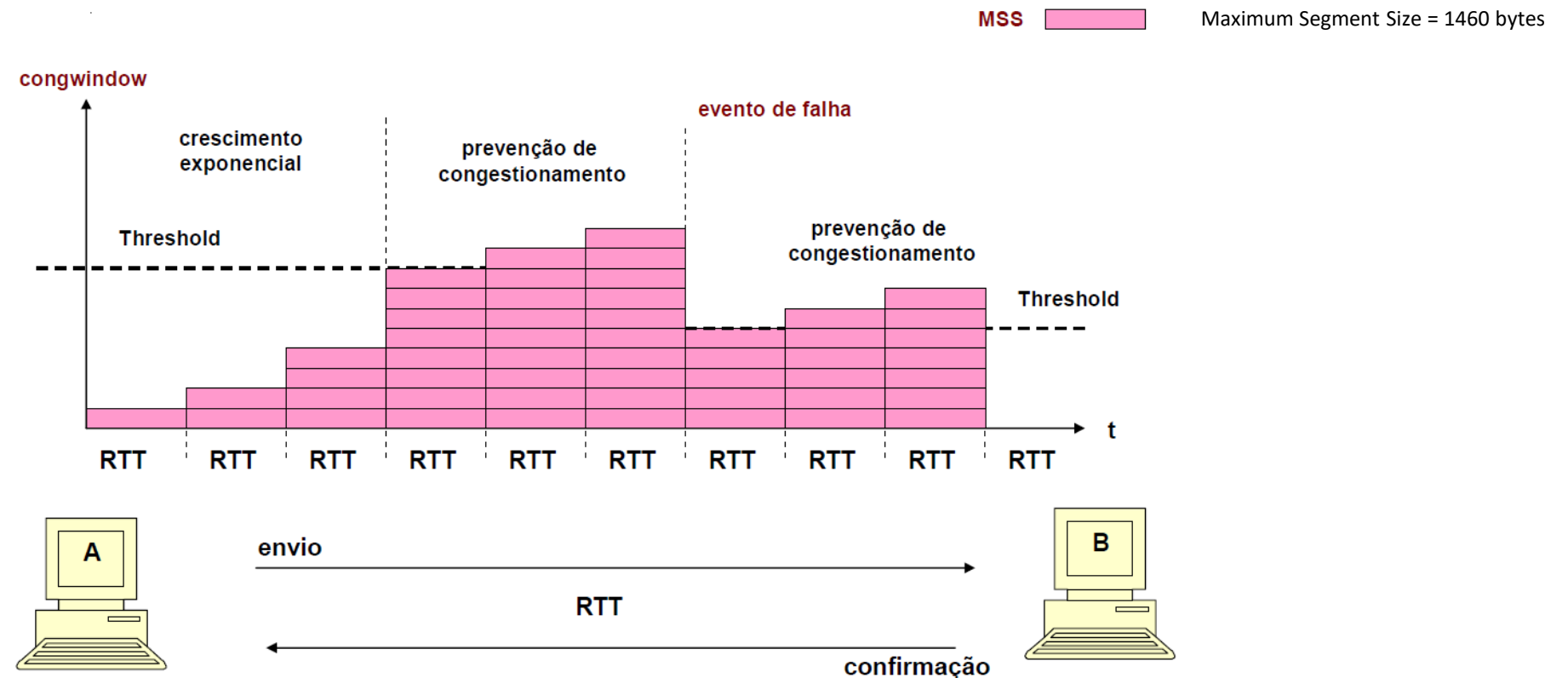
CongWindow

- Define um limite para a quantidade de bytes que pode ser transmitida pelo SO do transmissor
- É calculado com base no sucesso ou fracasso na transmissão de segmentos
- Segmento perdido: assume que a rede está congestionada e reduz o valor de CongWindow
- Segmentos transmitidos com sucesso: aumenta o valor de CongWindow

Protocolo TCP

Controle de congestionamento

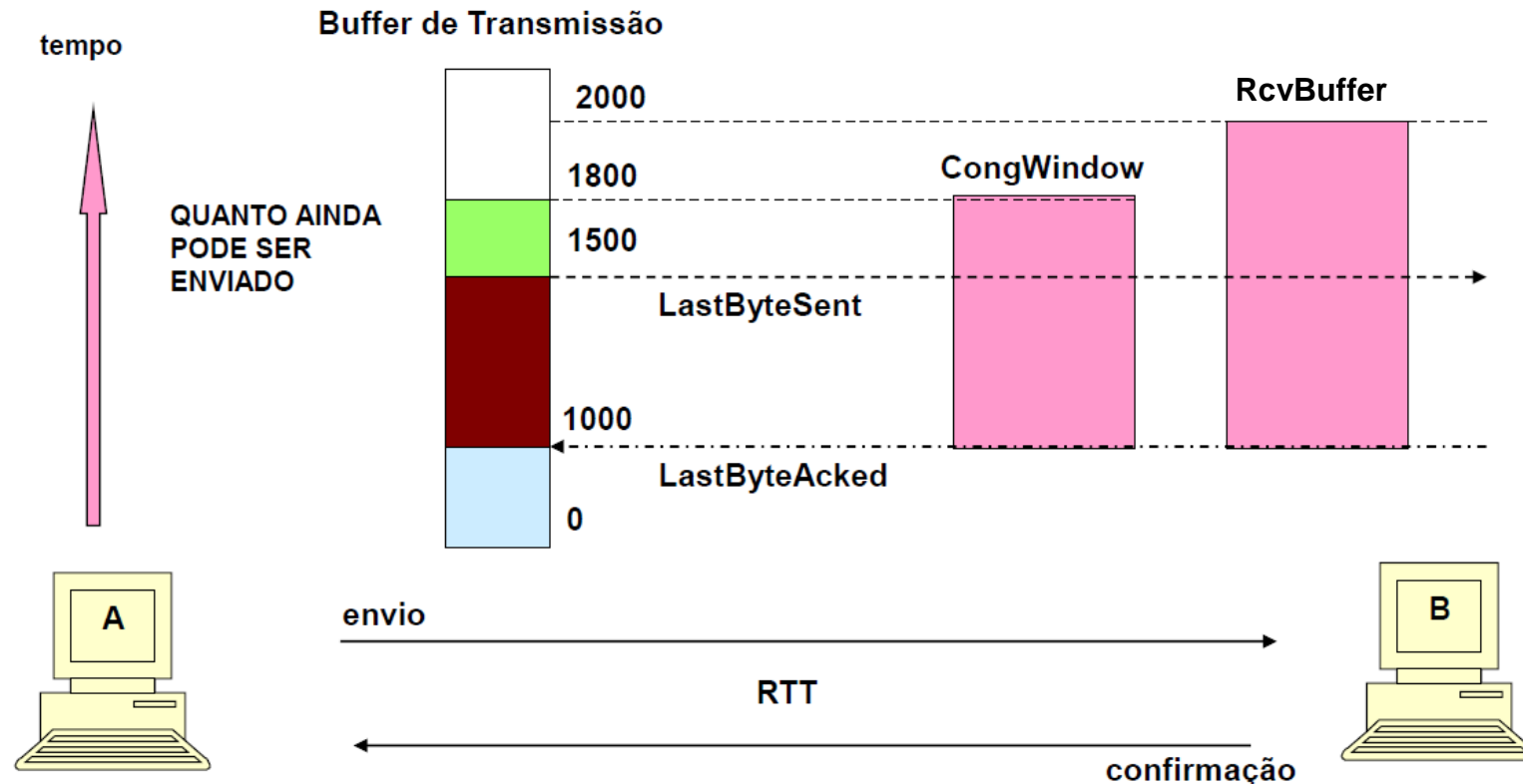
- O TCP impõe uma janela que limita o envio de bytes pelo transmissor (janela de congestionamento ou CongWindow)
- CongWindow é controlada pelo SO do transmissor (não é definida no cabeçalho do TCP)



Protocolo TCP

Controle de congestionamento

- A quantidade de bytes que pode ser transmitida sem confirmação é o menor valor entre CongWindow e RcvBuffer
- RcvWindow e CongWindow são calculados pelo SO do transmissor.



QUIZ 4. Controle de fluxo e congestionamento TCP

1. Indique as afirmativas corretas (existem 4)

- A. A aplicação receptora não precisa se preocupar com o controle de fluxo mas deve receber as mensagens de modo eficiente para não atrapalhar a eficiência da comunicação. ****
- B. A aplicação receptora deve informar ao sistema operacional o espaço livre no buffer de recepção.
- C. A aplicação transmissora não precisa se preocupar com o controle de fluxo. ****
- D. A aplicação transmissora precisa estimar o espaço disponível no buffer de recepção e controlar o envio de segmentos.
- E. O receptor TCP (no S.O.) calcula e transmite o valor da janela de recepção (RcvWindow). ****
- F. O receptor TCP (no S.O.) estima o valor do buffer do receptor (RcvBuffer) e envia para o transmissor TCP.
- G. O transmissor TCP (no S.O.) estima o valor do buffer no receptor (RcvBuffer) usando o valor RcvWindow enviado pelo receptor. ****

QUIZ 4. Controle de fluxo e congestionamento TCP

2. Indique as afirmativas corretas (existem 3)

- A. O controle de congestionamento é feito no transmissor TCP sem nenhuma informação passada pelo receptor.
- B. O controle de congestionamento é feito no transmissor TCP utilizando as variáveis CongWindow e RcvBuffer. ****
- C. A variável CongWindow é calculada no receptor TCP e informada ao transmissor em um campo do pacote TCP.
- D. A variável CongWindow é calculada pelo transmissor TCP de acordo com base no sucesso ou fracasso na transmissão. ****
- E. Após a inicialização e partida lenta, o algoritmo de controle de congestionamento tem duas fases: crescimento exponencial e prevenção de congestionamento. ****

Protocolos de Comunicação

TCP - Detalhamento