

Disciplina: Redes de Computadores

Nível de Transporte
Portas, Protocolos UDP e TCP

Profa. Débora Christina Muchaluat Saade

debora@midia.com.uff.br

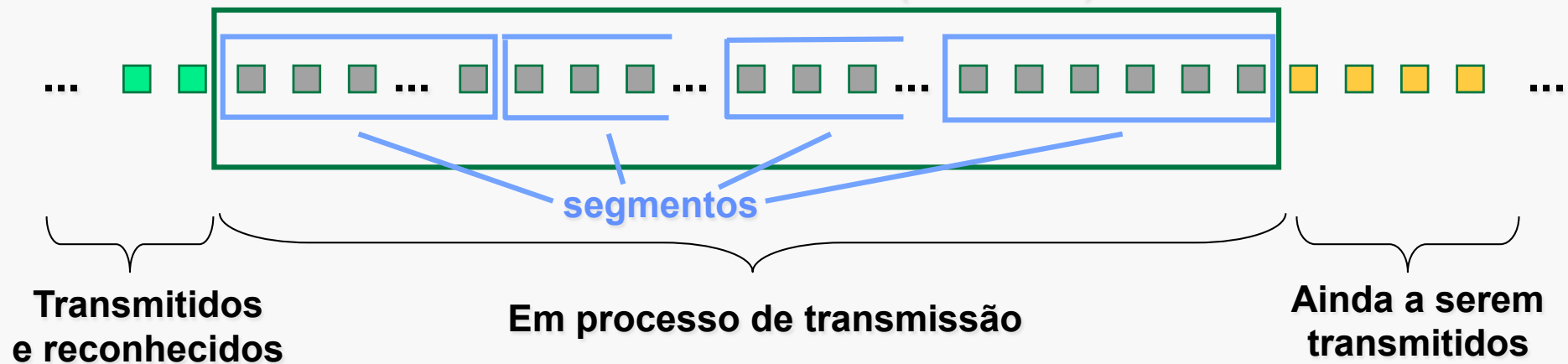
Protocolo TCP

Janelas deslizantes no TCP

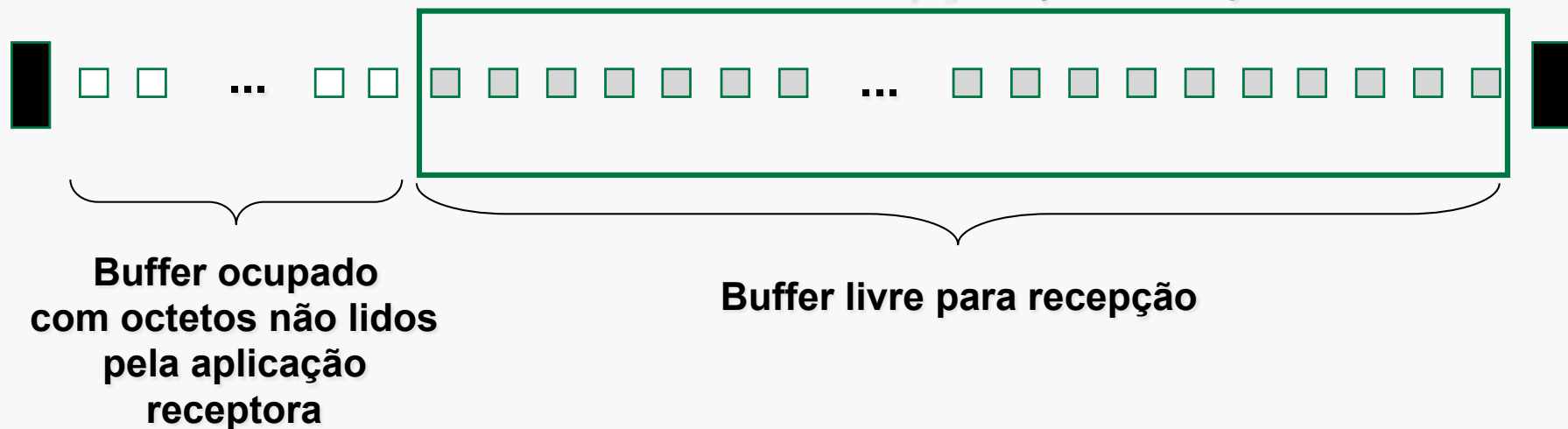
- **Atuam no controle de fluxo E no controle de erro**
 - *Controle de fluxo*
 - Restringem a velocidade do transmissor caso o receptor não tenha buffer suficiente para receber mais dados
 - *Controle de erro*
 - Permitem identificar os segmentos já reconhecidos
- **Quatro janelas por conexão**
 - *Tamanho das janelas é variável*
- **Janelas operam sobre octetos, não sobre segmentos**

Janelas deslizantes no TCP

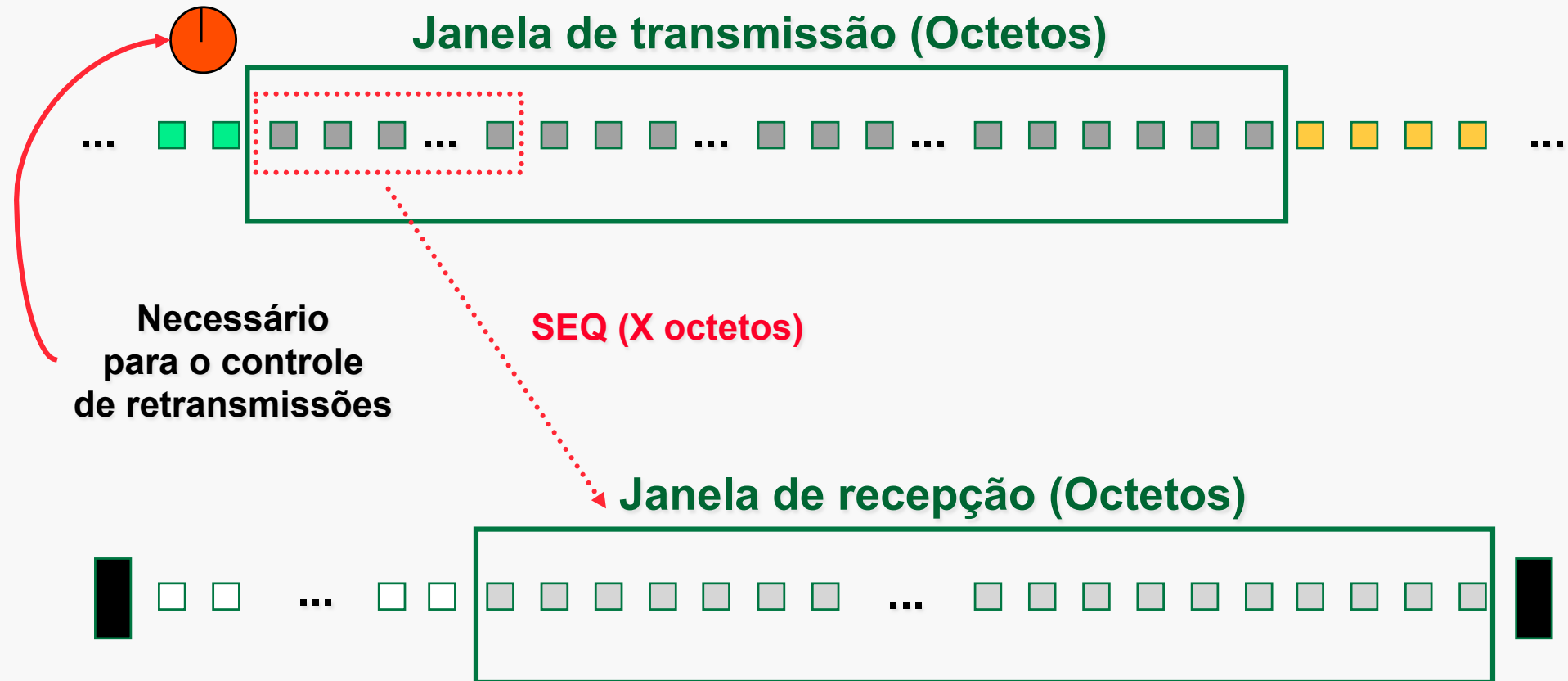
Janela de transmissão (Octetos)



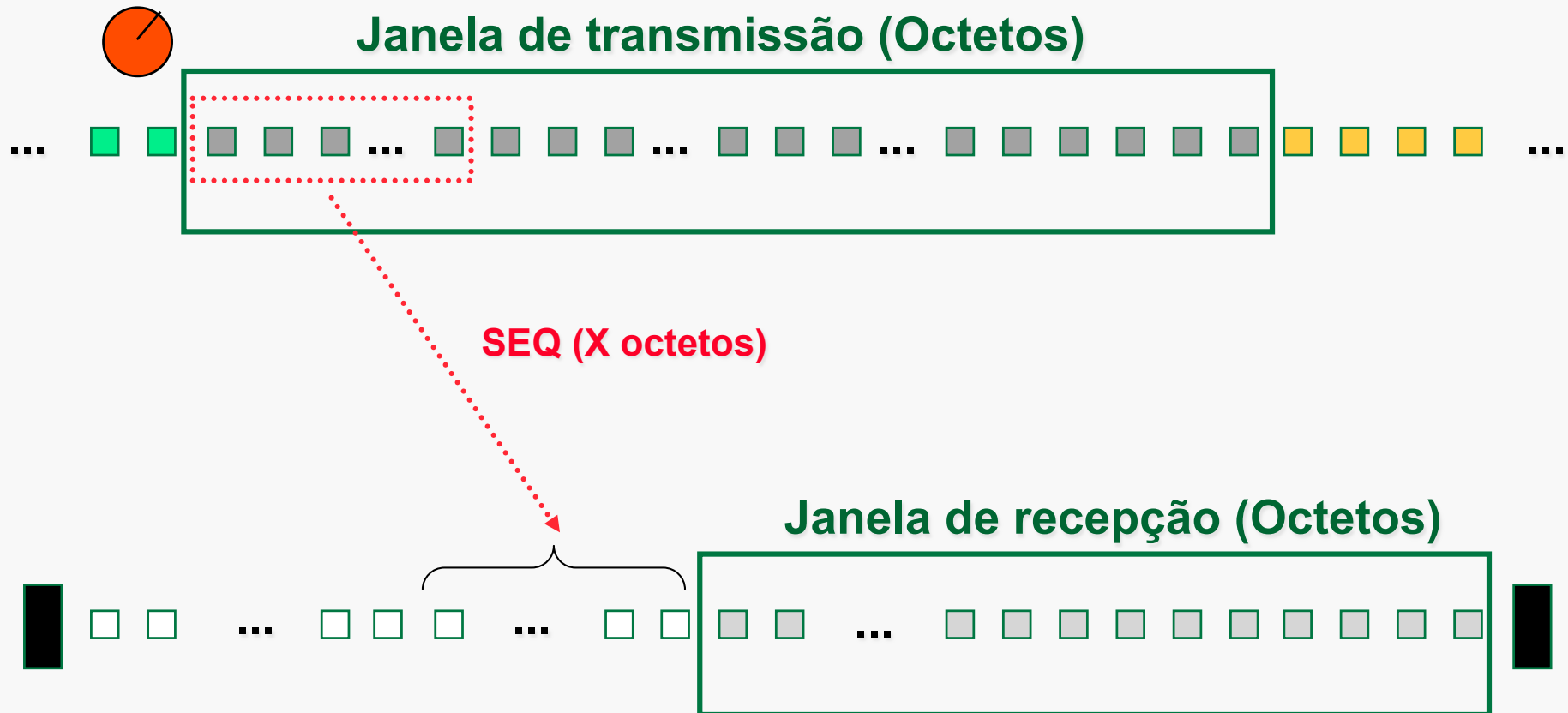
Janela de recepção (Octetos)



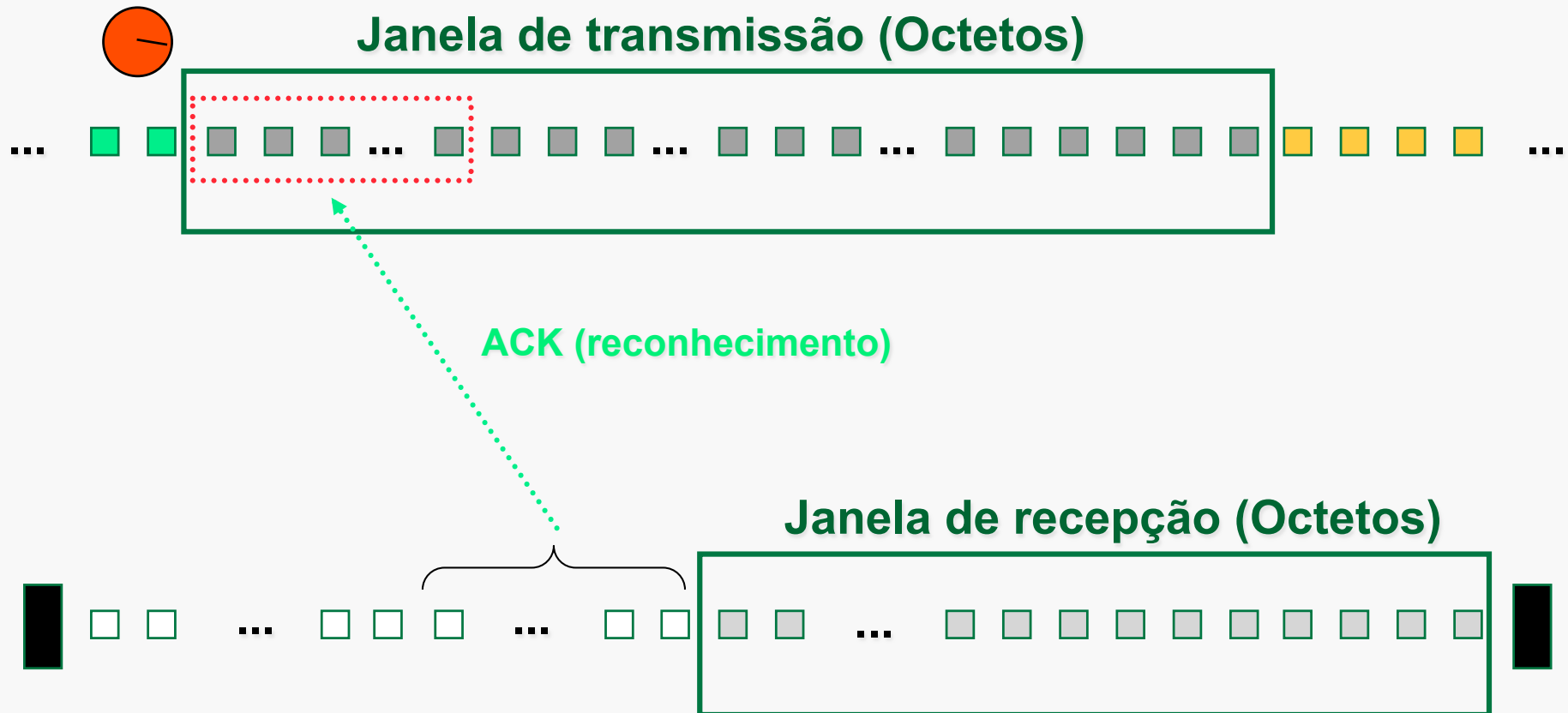
Janelas deslizantes no TCP



Janelas deslizantes no TCP



Janelas deslizantes no TCP



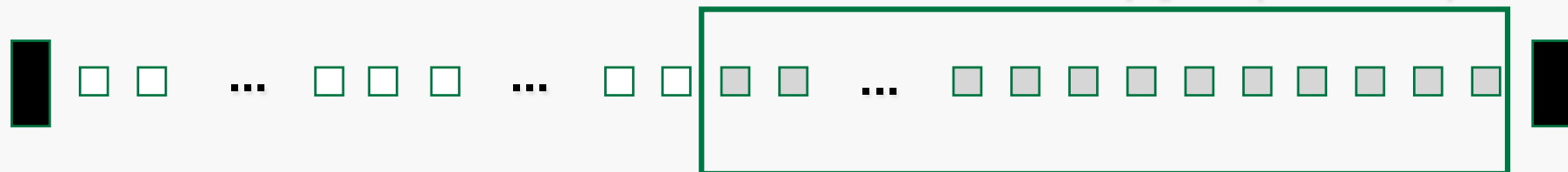
Reconhecimentos são sempre cumulativos, isto é, especificam o número de sequência do próximo OCTETO que o receptor espera receber.

Janelas deslizantes no TCP

Janela de transmissão (Octetos)



Janela de recepção (Octetos)

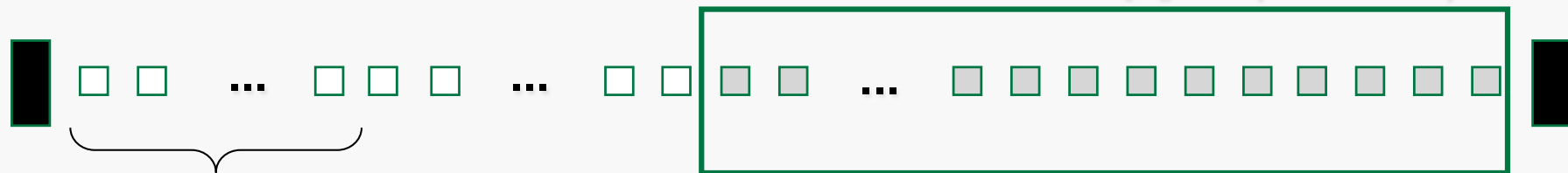


Janelas deslizantes no TCP

Janela de transmissão (Octetos)

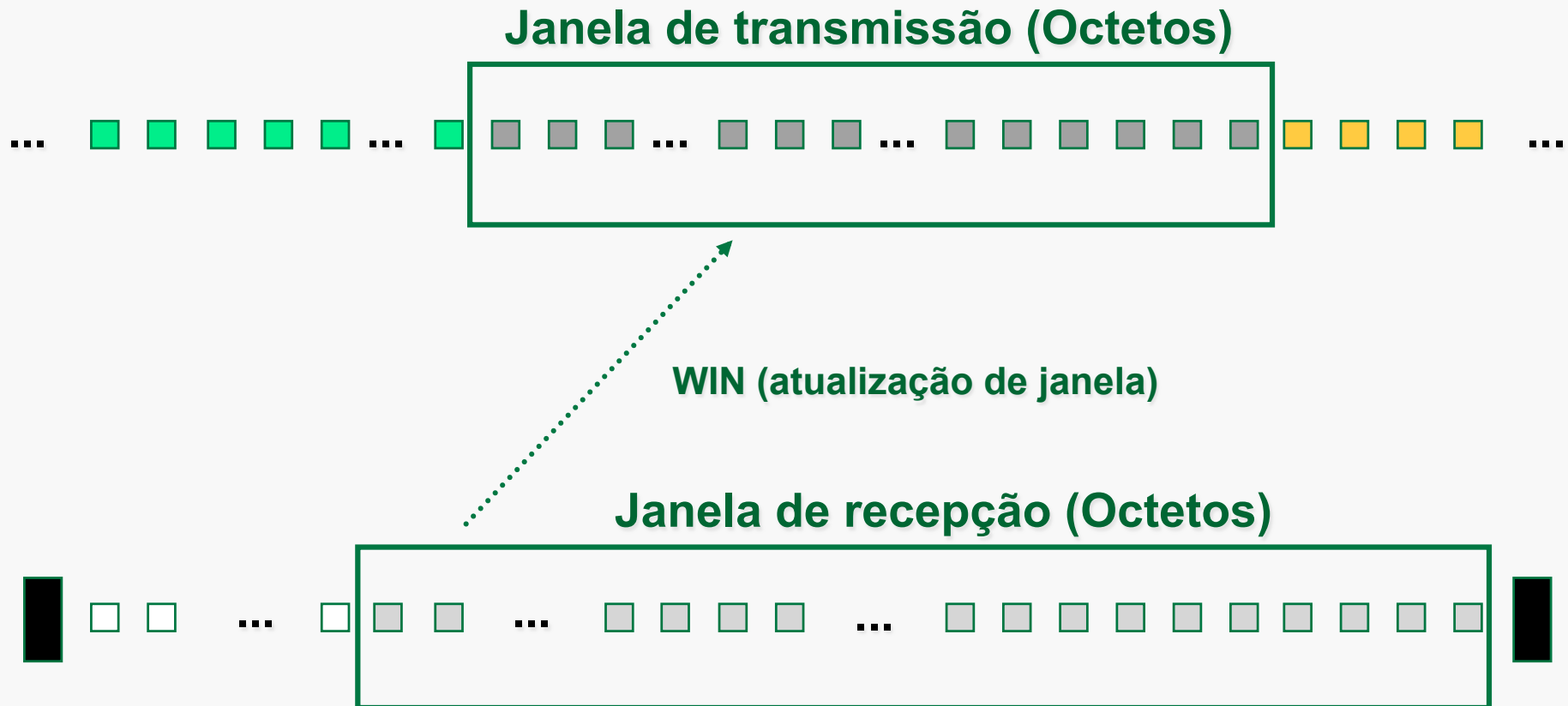


Janela de recepção (Octetos)



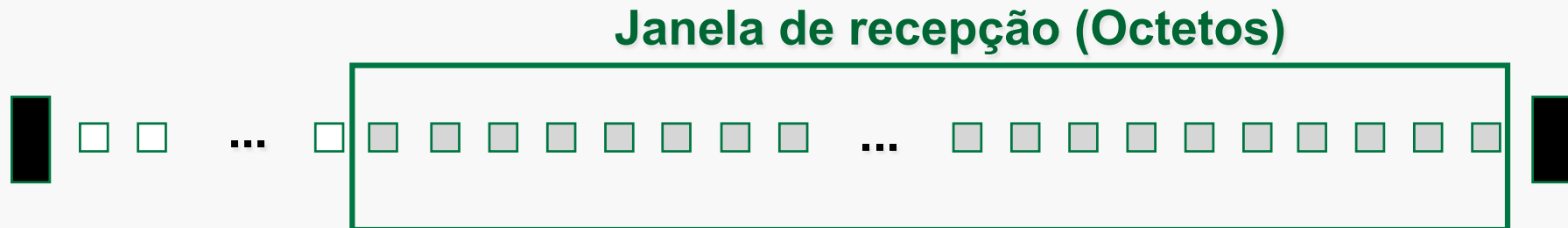
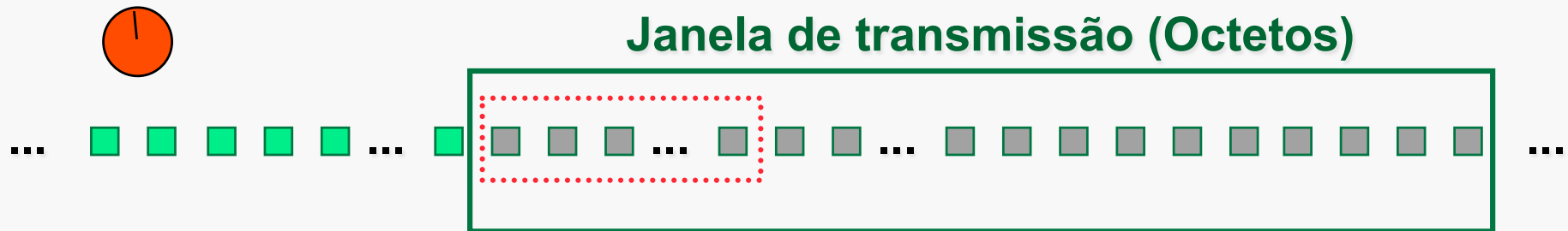
Aplicação
receptora lê
Y octetos

Janelas deslizantes no TCP



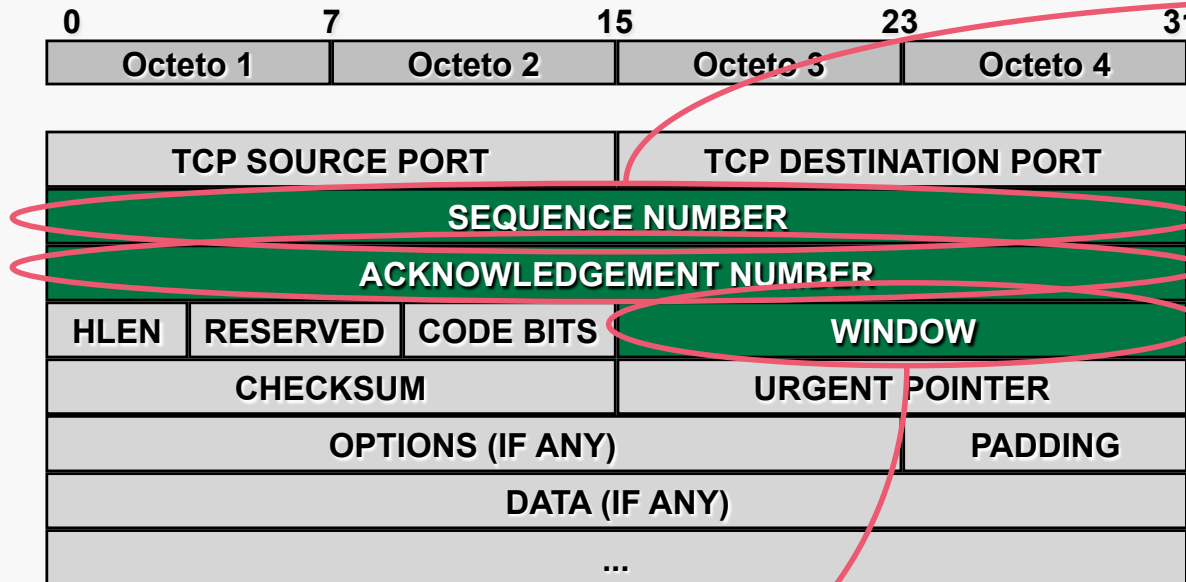
Janelas deslizantes no TCP

Redes de Computadores



Bufferização e temporização são fundamentais para controle de erro e fluxo em TCP

Segmento TCP



Identifica a posição, relativa ao fluxo de bits do transmissor deste segmento, dos dados carregados pelo mesmo. OBS: posição informada em octetos

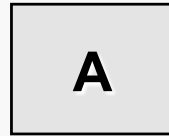
Informa ao receptor deste segmento o próximo número de sequência que o transmissor deste segmento espera receber no próximo segmento

Informa ao receptor deste segmento a quantidade máxima de octetos que o transmissor deste segmento pode aceitar no momento

ACK NUMBER e WINDOW são carregados no segmento do sentido oposto (*piggybacking*)

Suposições no exemplo:

- buffer de Rx de A vazio neste instante
- buffers de Tx de A e B infinitamente cheios



Buffer de
Tx de A

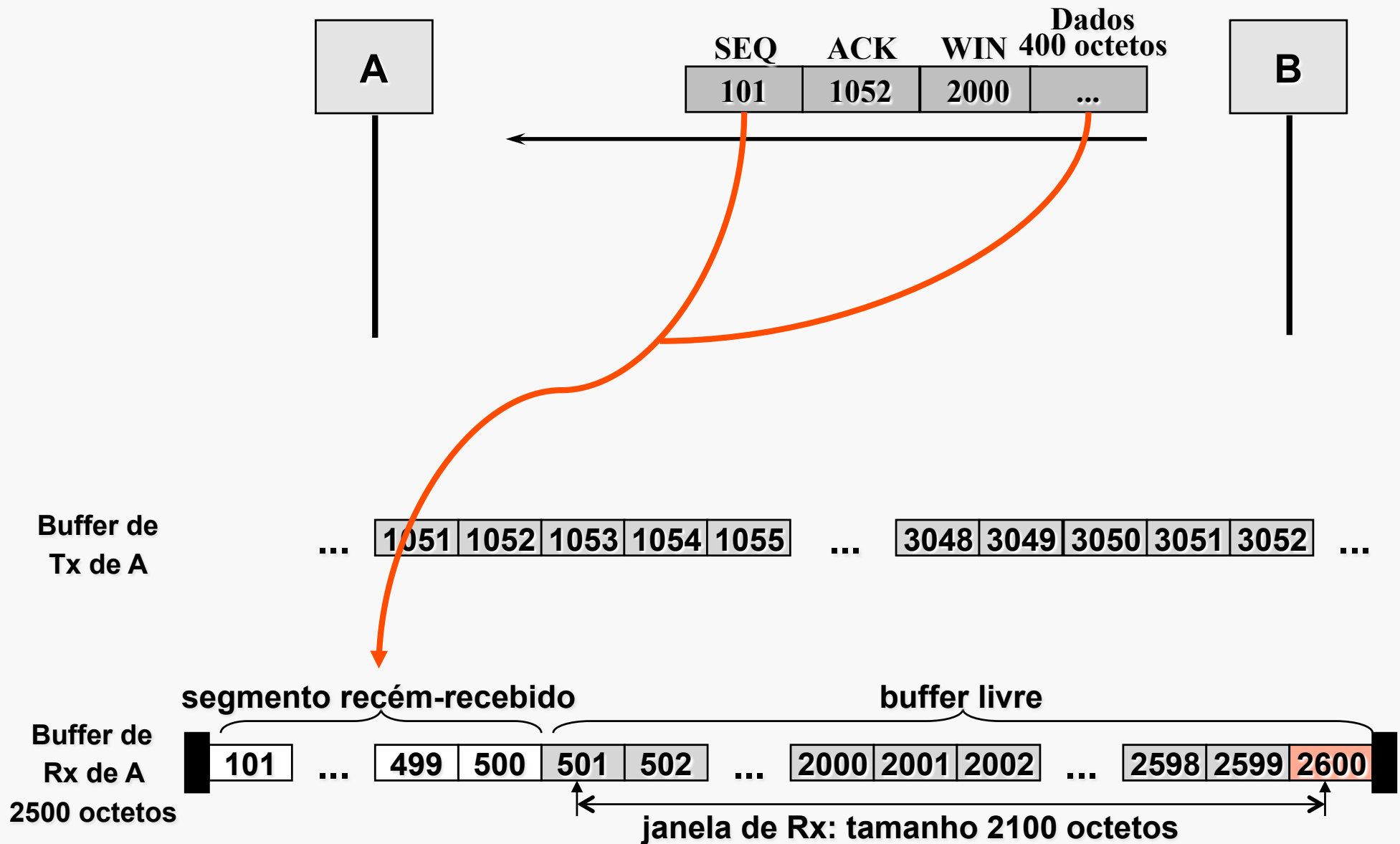


Buffer de
Rx de A
2500 octetos

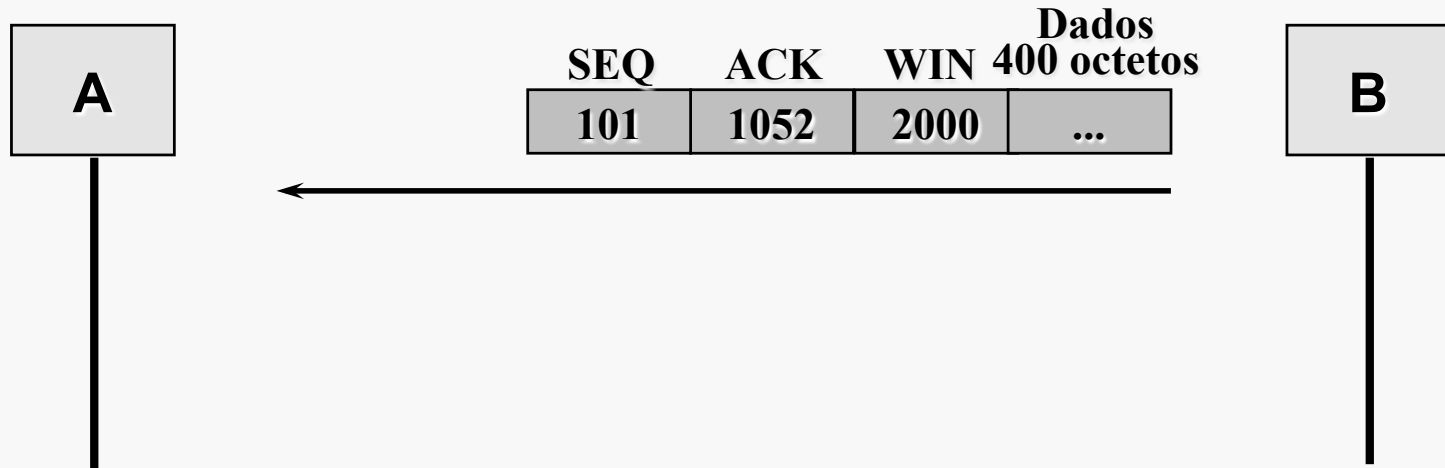


TCP em B “resolve” transmitir 400 octetos.

Podemos inferir o novo estado do buffer de Rx de A ?



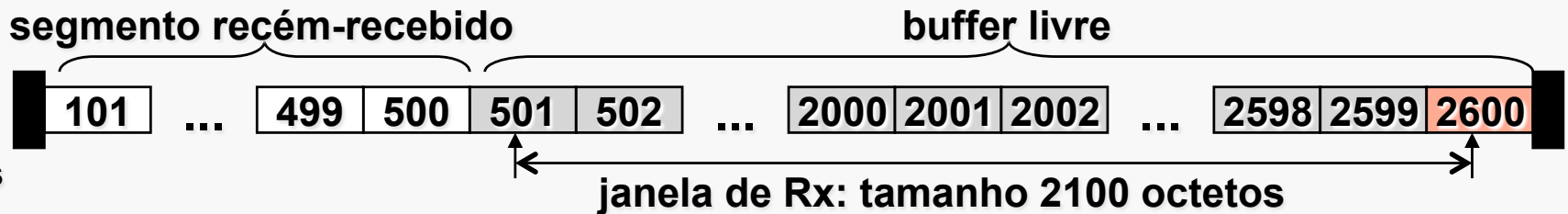
E o estado do buffer de Tx de A ?

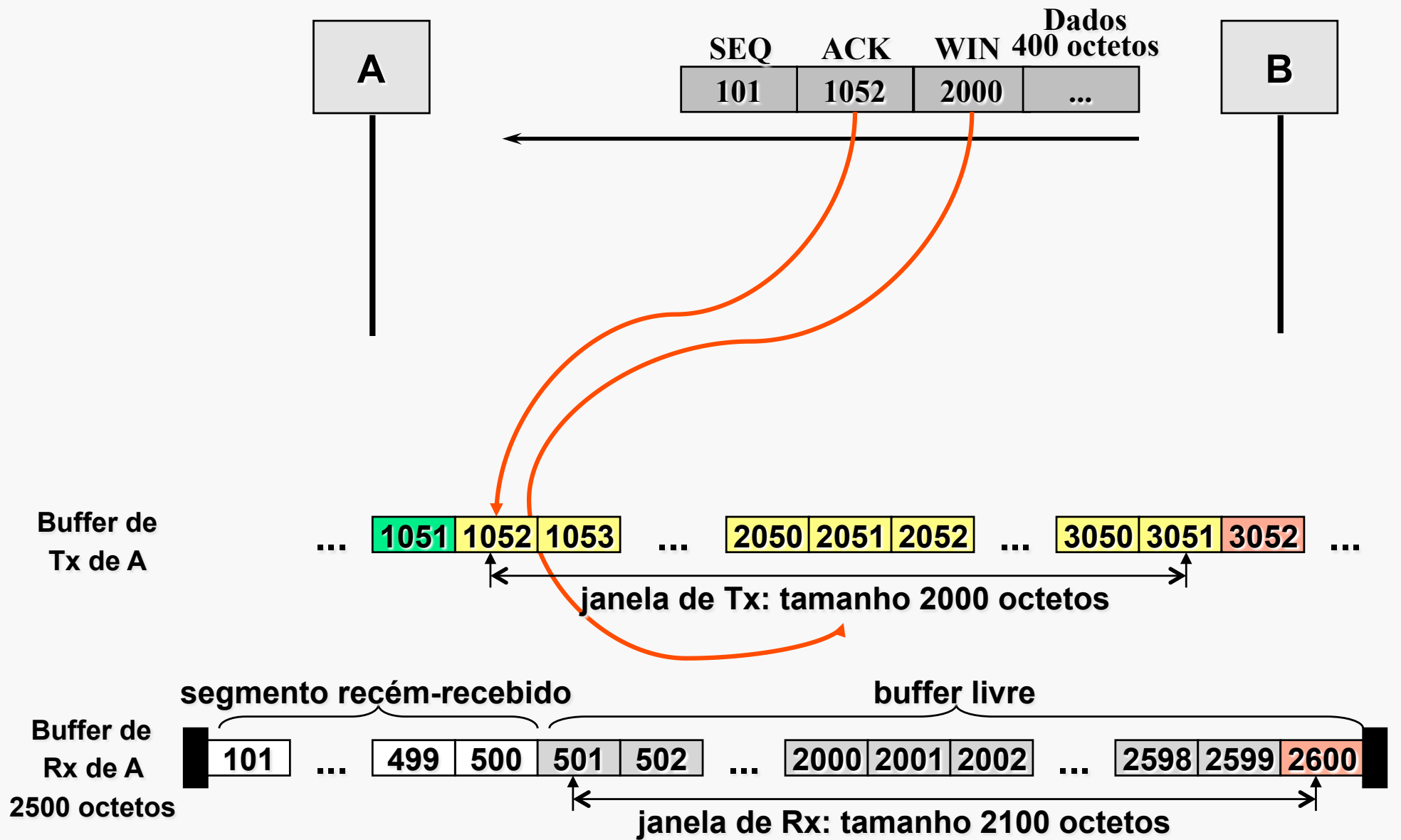


Buffer de Tx de A

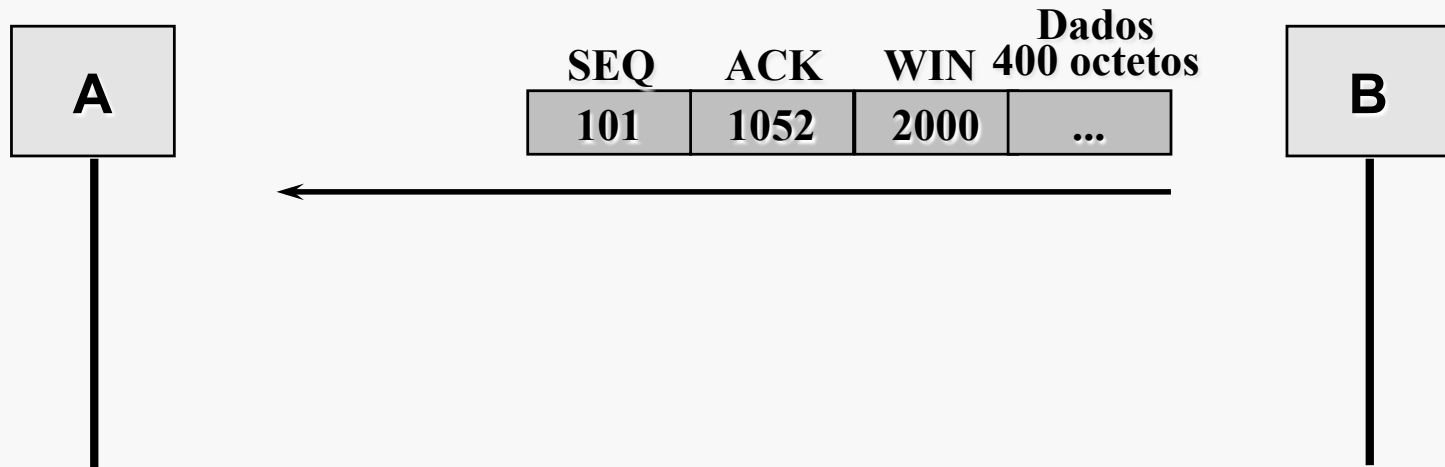


Buffer de Rx de A
2500 octetos

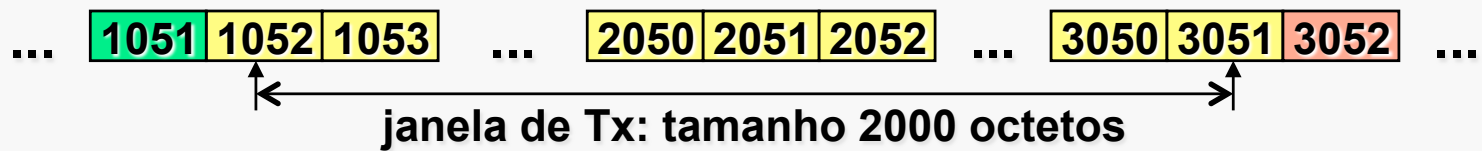




TCP em A “resolve” transmitir 1000 octetos



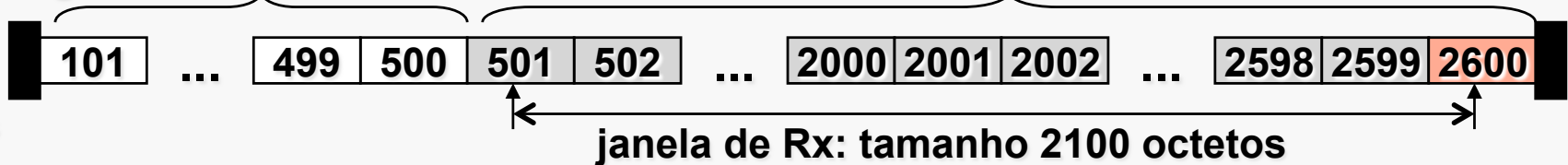
Buffer de Tx de A

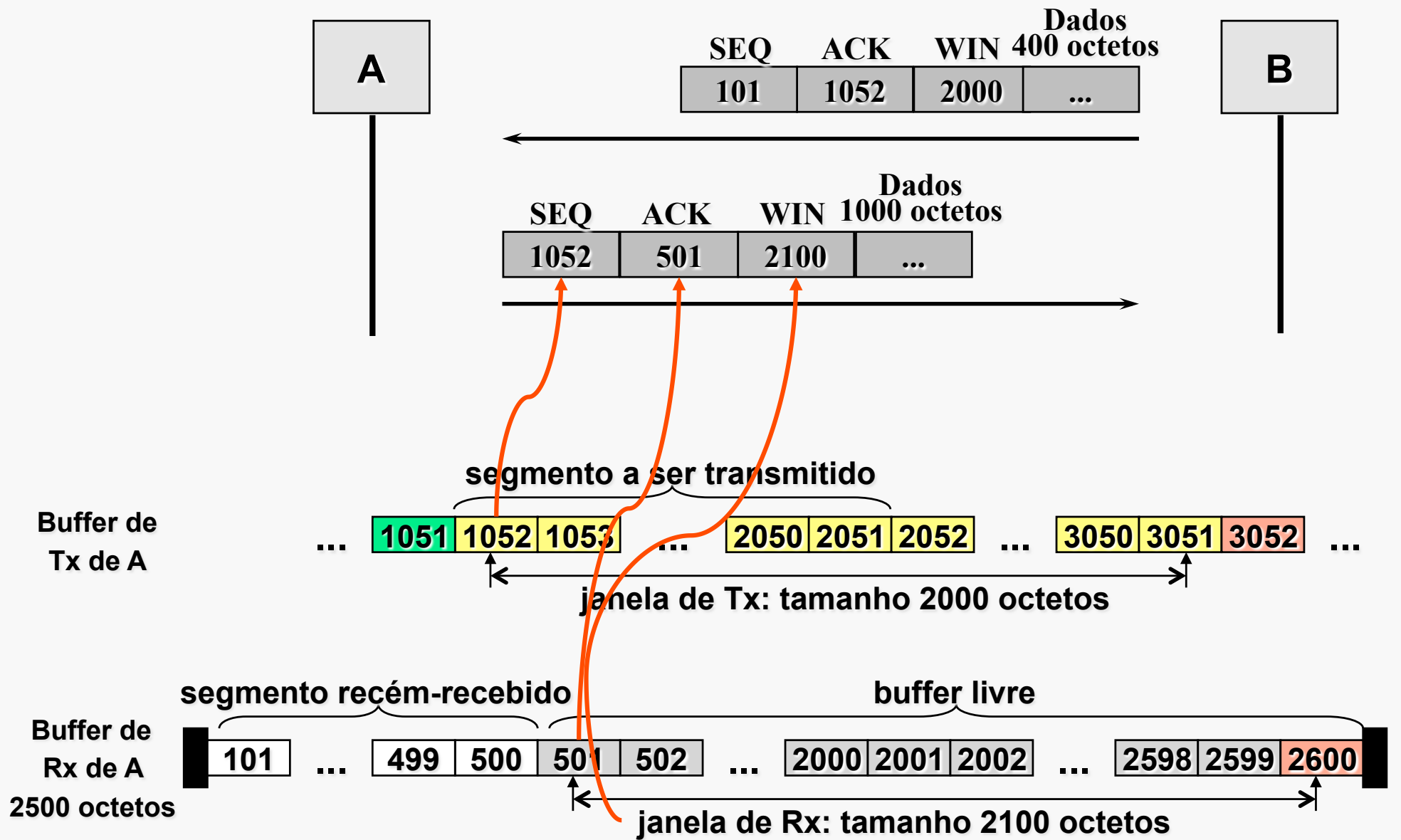


Buffer de Rx de A
2500 octetos

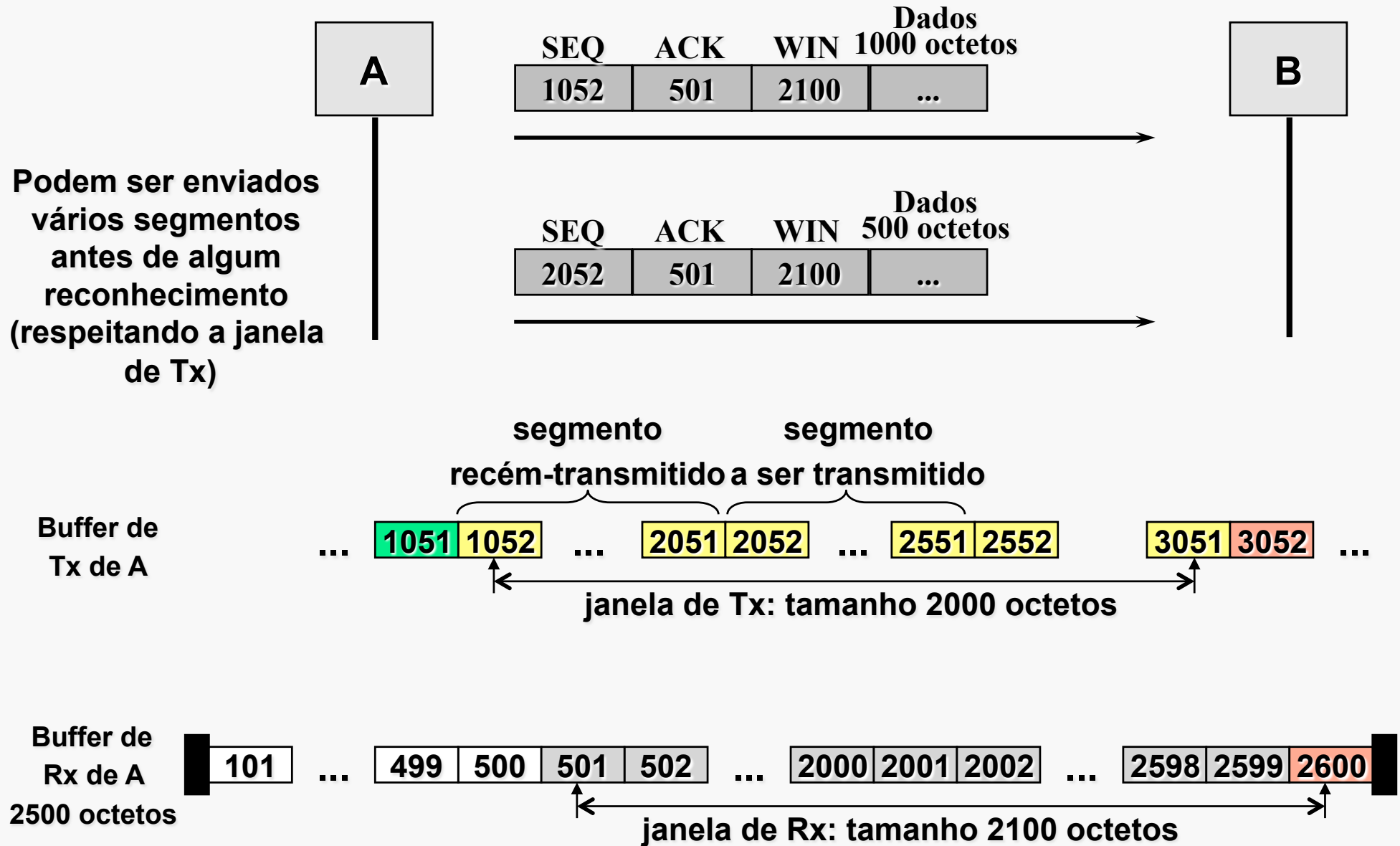
segmento recém-recebido

buffer livre

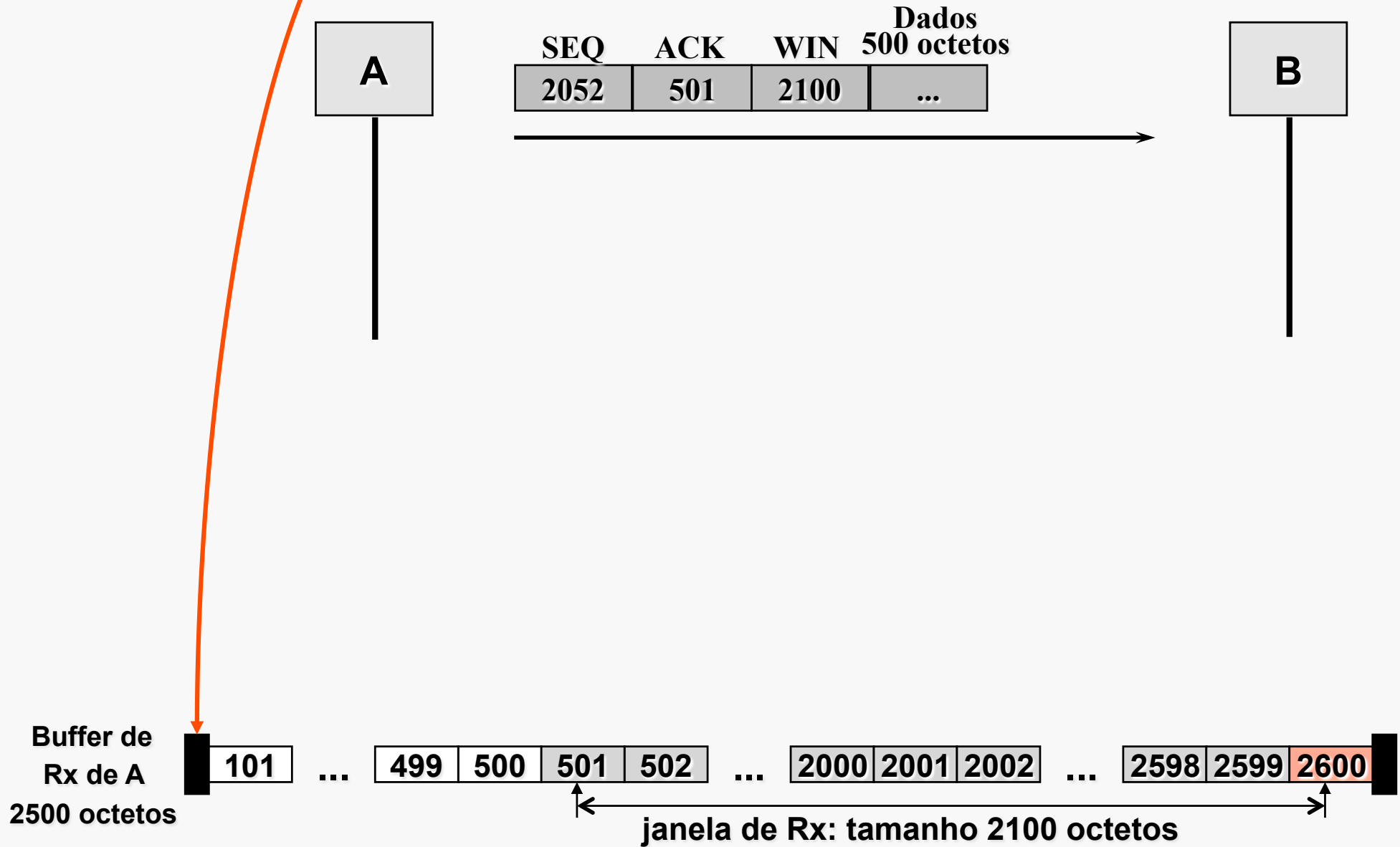


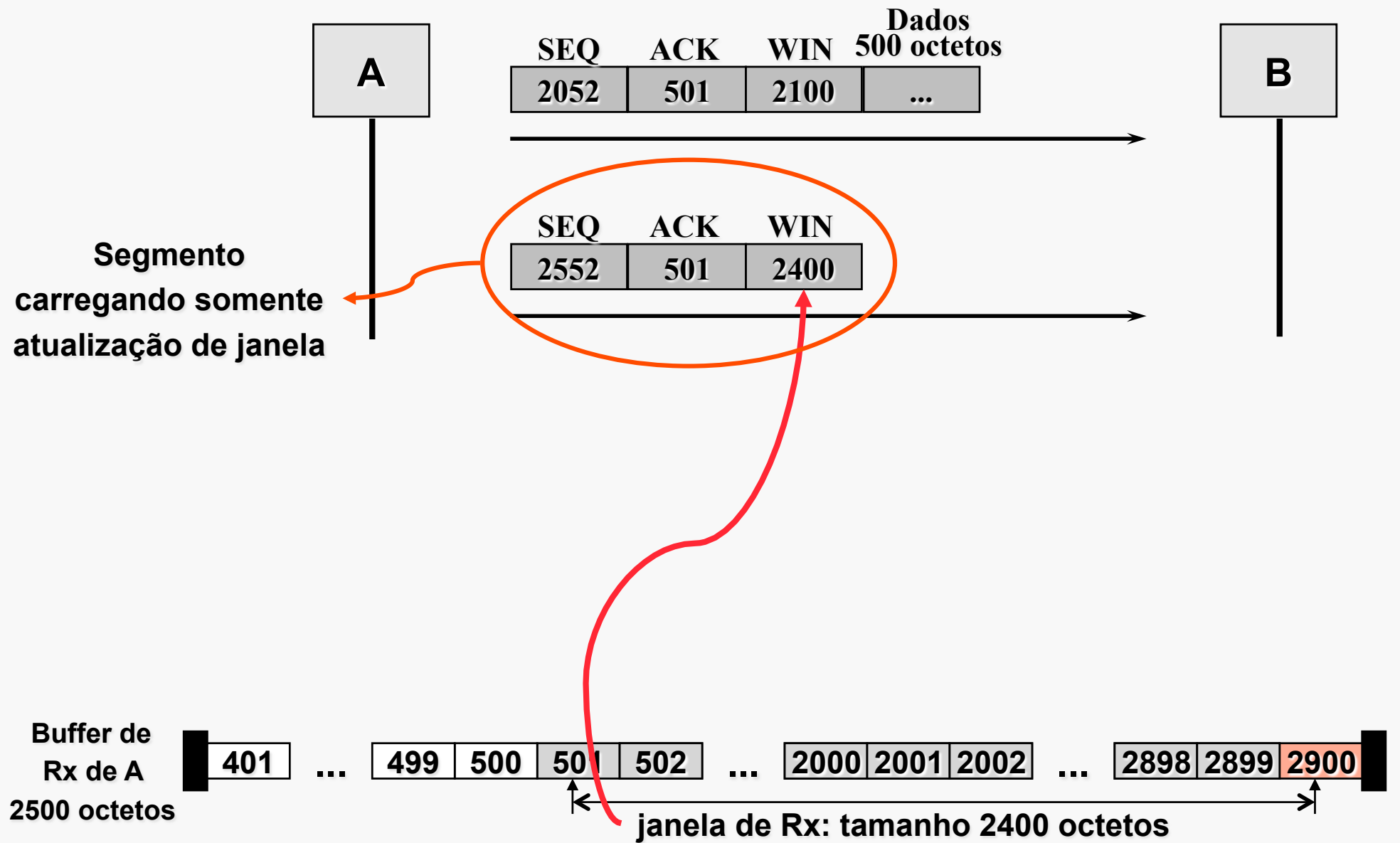


TCP em A “resolve” transmitir mais 500 octetos

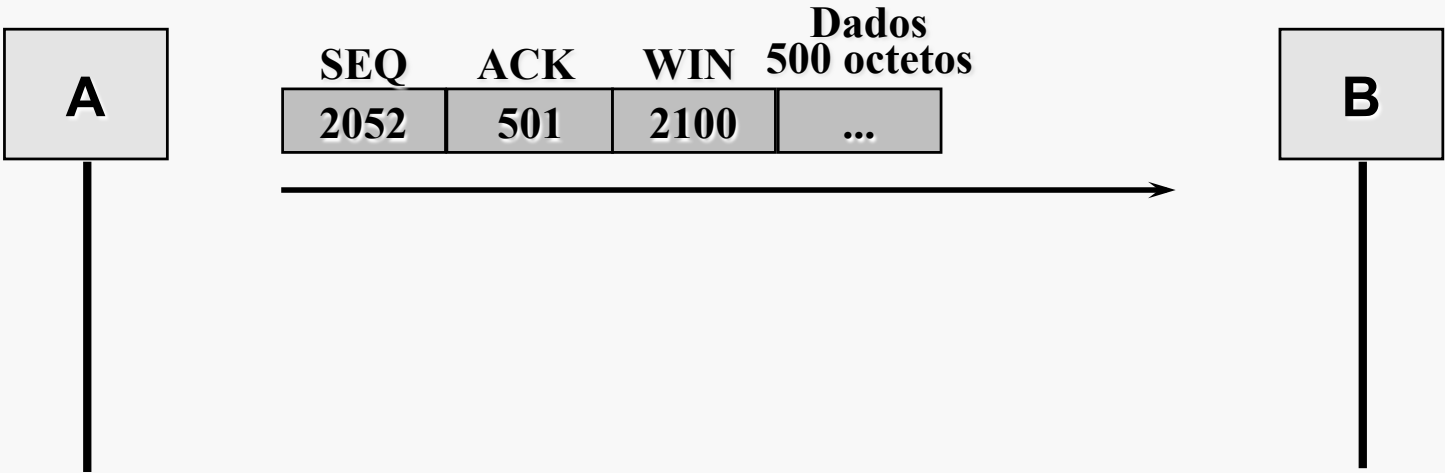


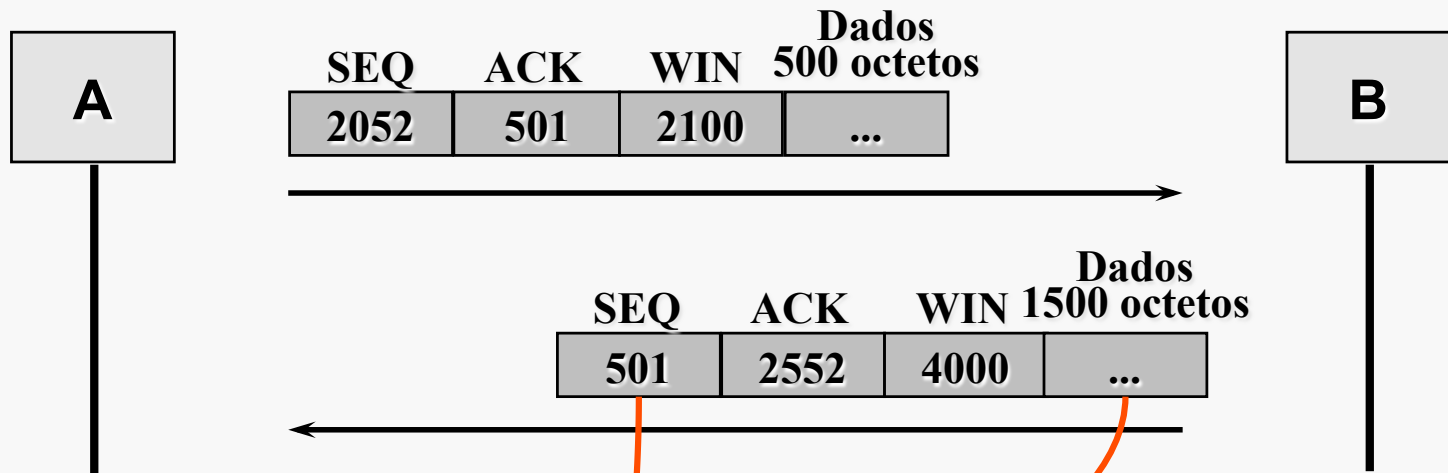
Aplicação em A leu 300 octetos

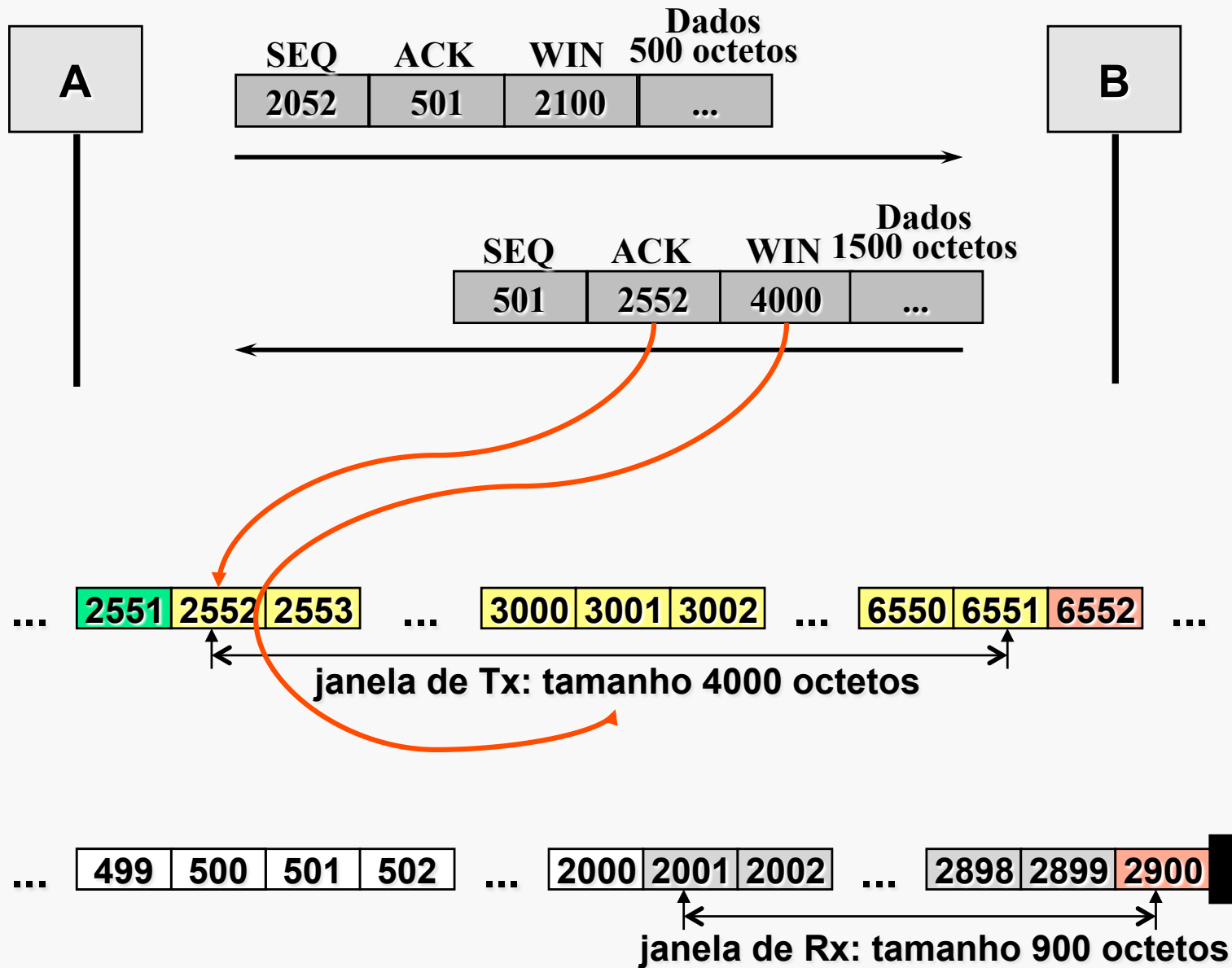




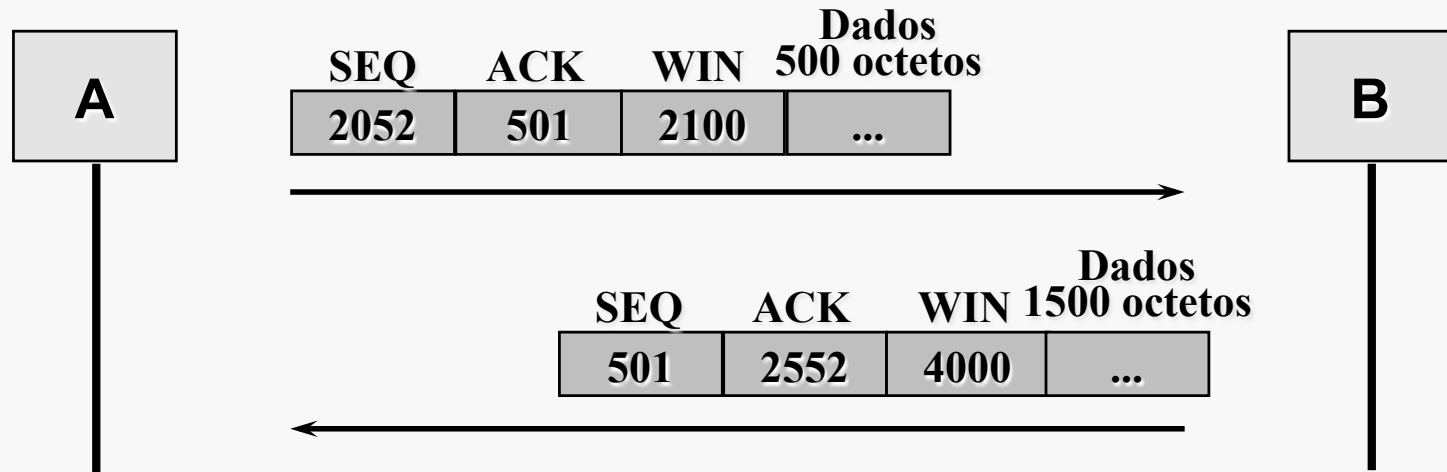
TCP em B “resolve” transmitir 1500 octetos



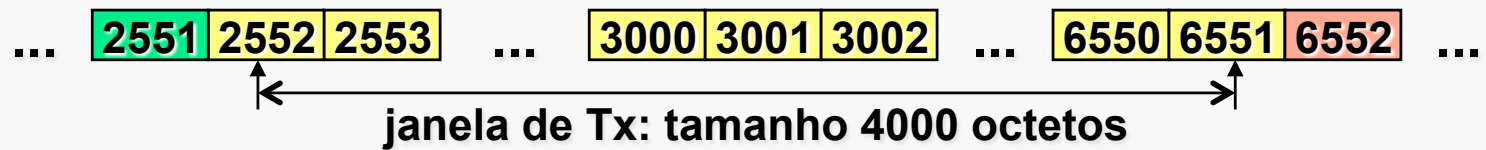




TCP em A “resolve” transmitir 450 octetos

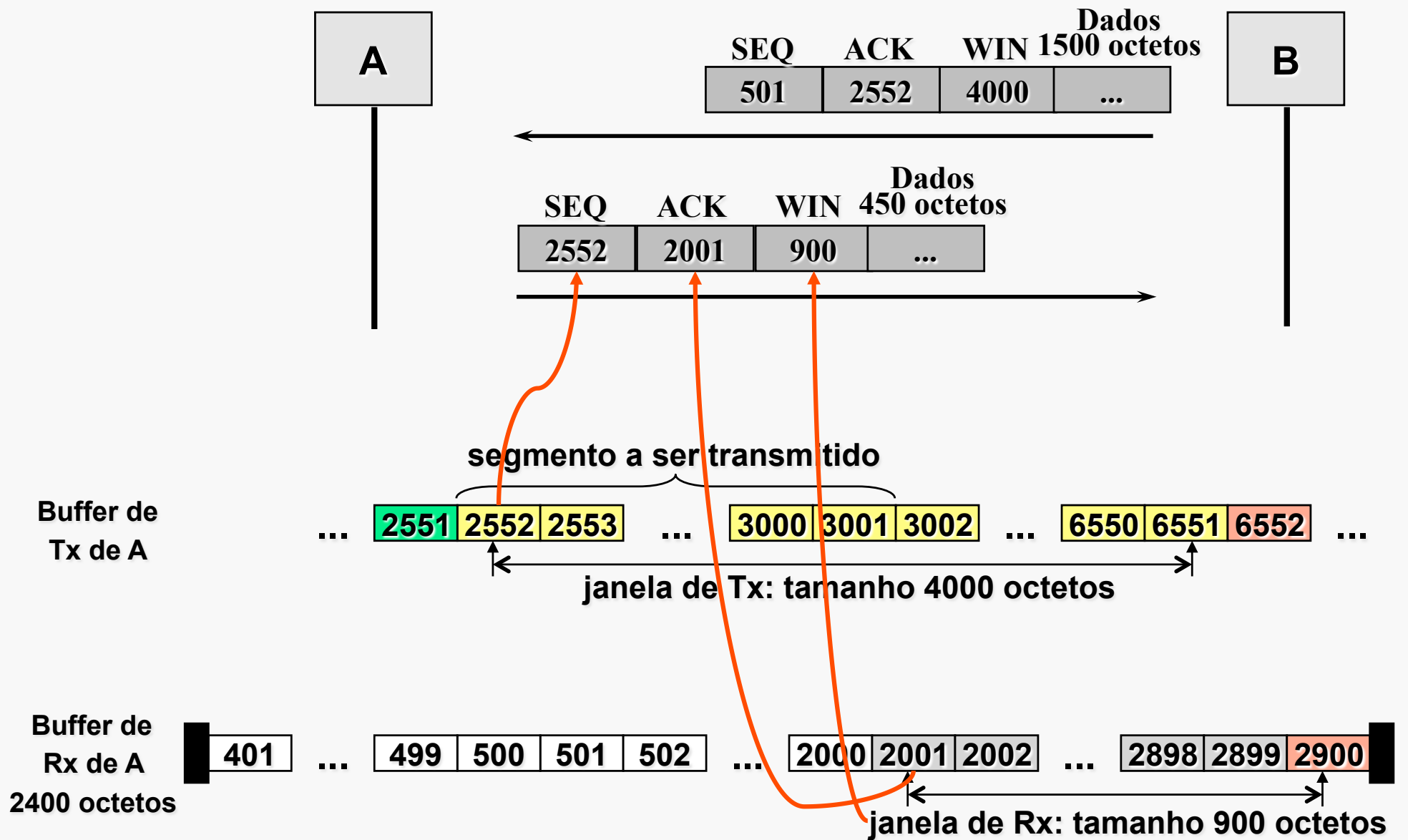


Buffer de Tx de A



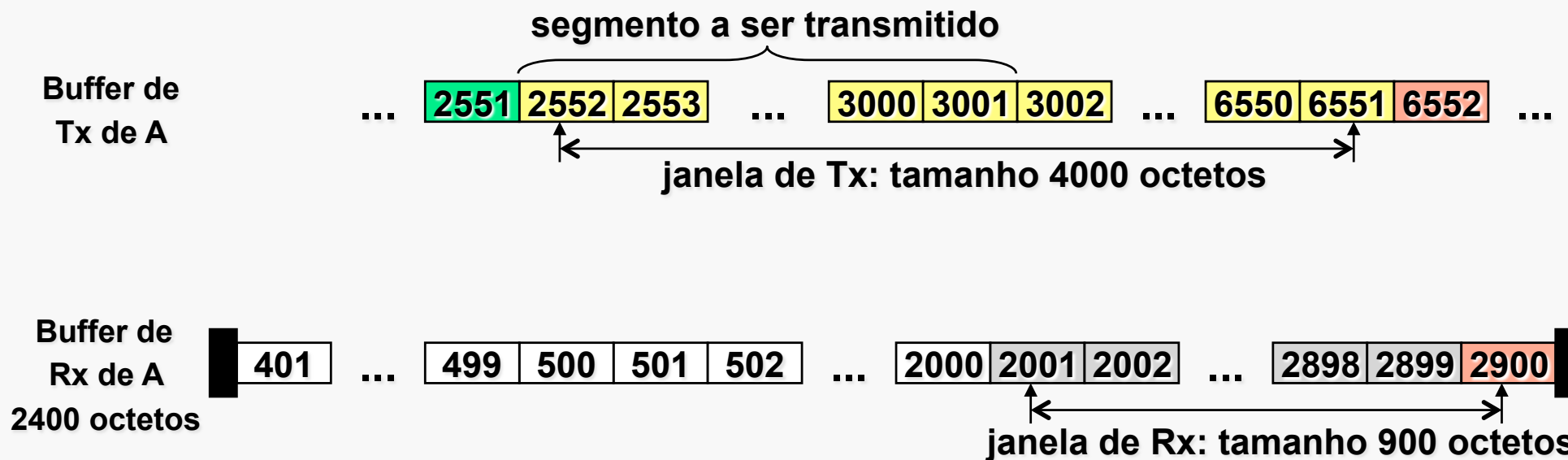
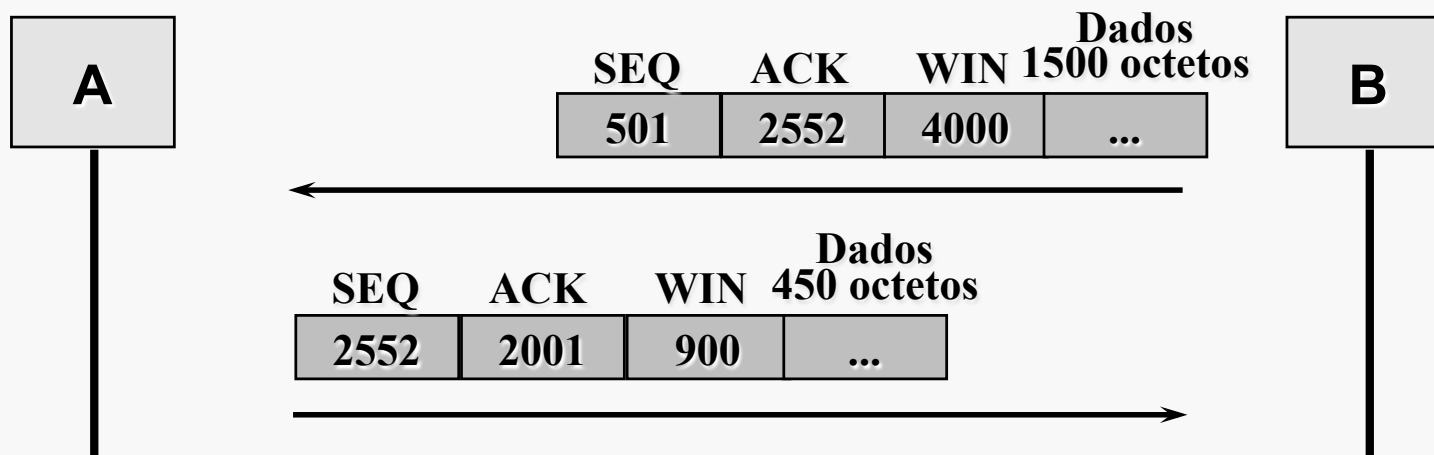
Buffer de Rx de A
2500 octetos

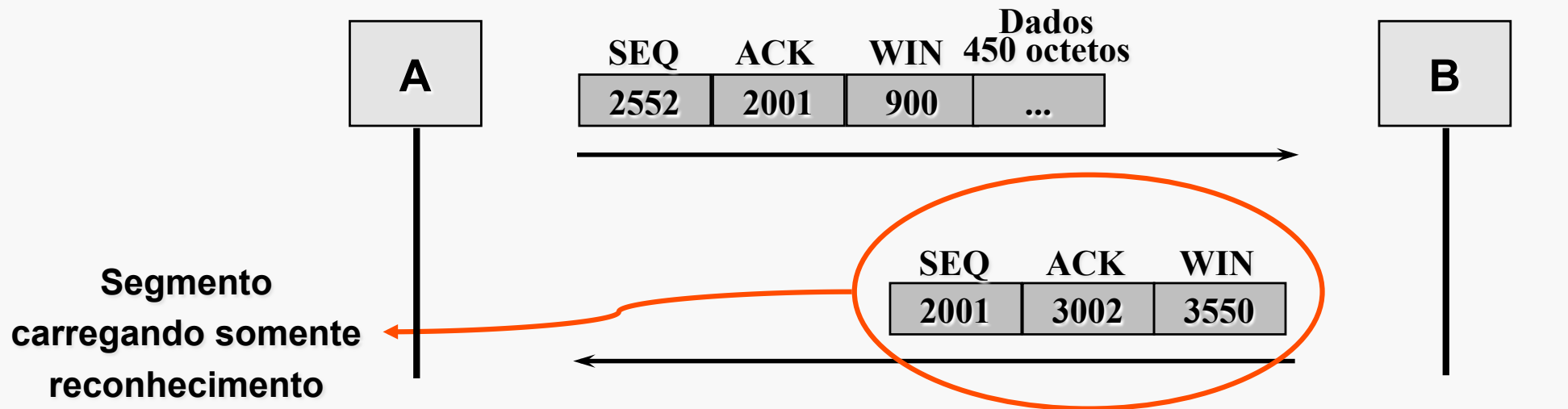




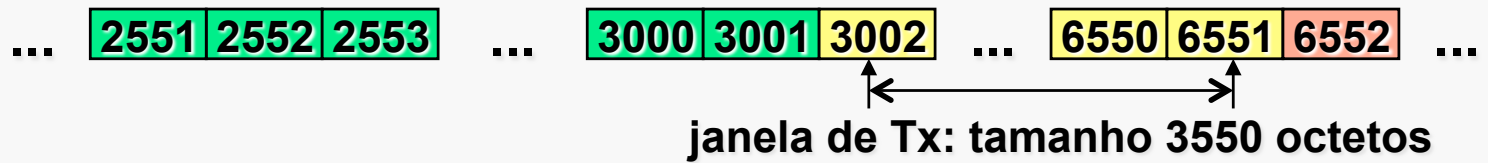
TCP em B “resolve” reconhecer 450 octetos

(por exemplo, já se passaram 0.2 seg desde o último reconhecimento)





Buffer de Tx de A

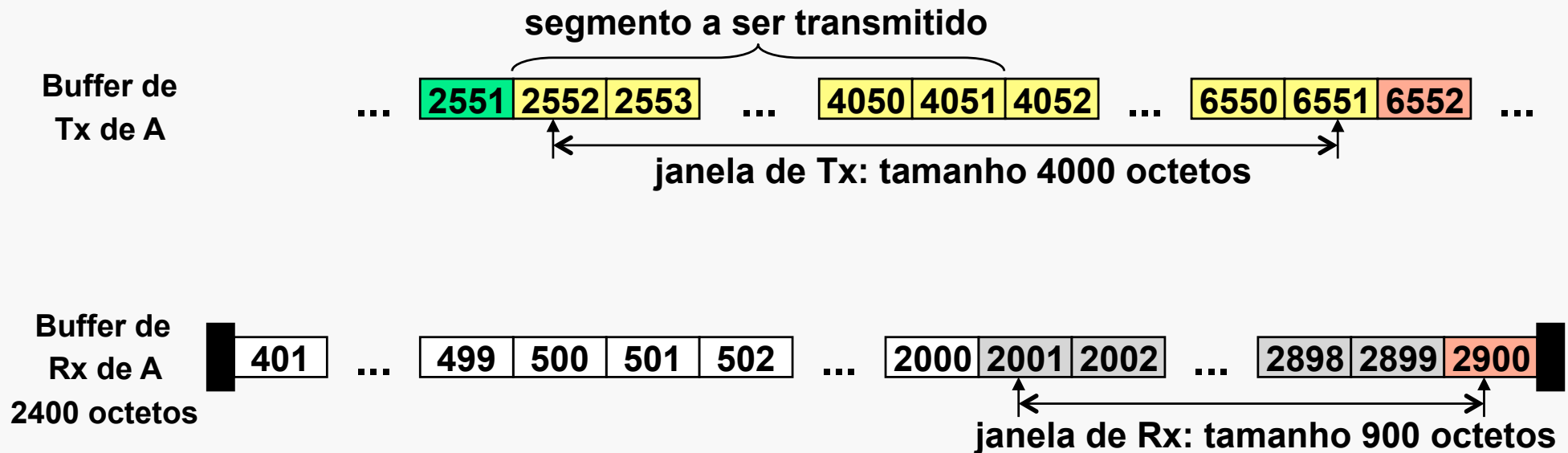
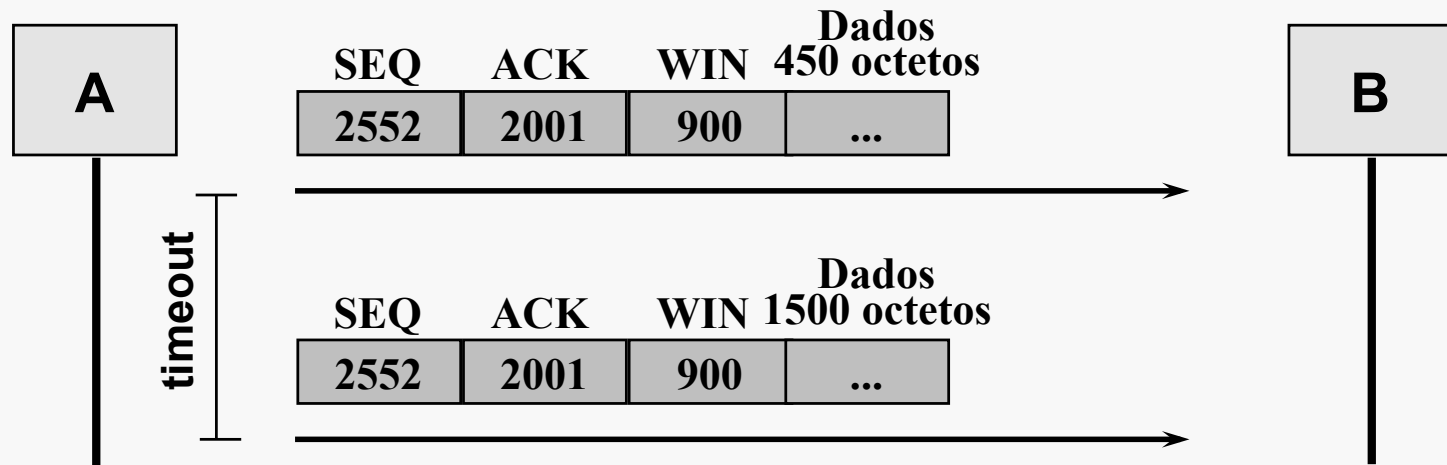


Buffer de Rx de A
2400 octetos



E se o reconhecimento não chegar antes do timeout?

Retransmissão



Considerações a respeito da retransmissão

- **Novamente: reconhecimento especifica próximo OCTETO esperado pelo receptor**
 - *TCP não é obrigado a retransmitir o segmento original. Ele pode transmitir um segmento resultante da união de dois ou mais segmentos consecutivos (repacketization), respeitando o tamanho máximo que um segmento pode ter*
- **Configuração do timeout de retransmissão**
 - *Timeout pequeno: retransmissões desnecessárias*
 - *Timeout grande: baixo throughput*