Camada de Transporte – Parte I

Prof. Jaime Cohen

2018

Sumário

Camada de Transporte (Parte I)

Protocolos de transporte

- Provê um serviço de comunicação entre processos sobre um serviço de comunicação entre hospedeiros.
- Implementado nos hospedeiros
- Envia segmentos entre processos
- Segmentos encapsulam dados da aplicação (ex. HTTP, SMTP, FTP)
- Protocolos de Transporte: TCP e UDP

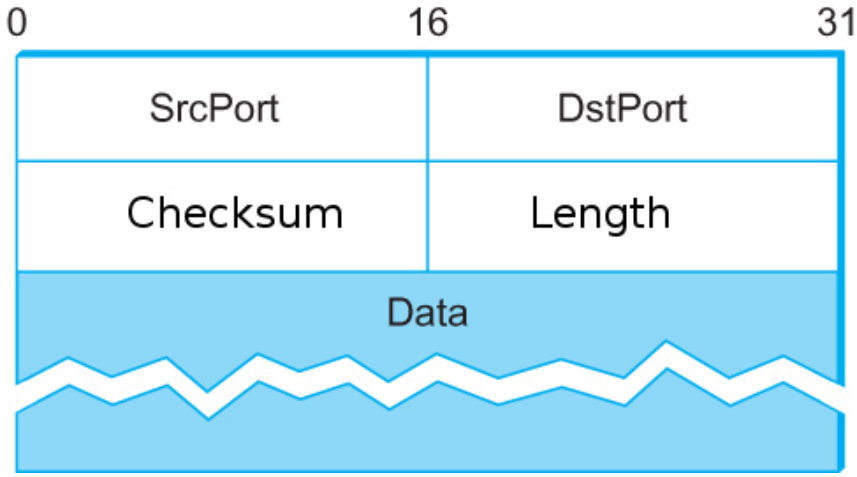
UDP

- Demultiplexação
- Checagem simples de erros (*checksum*)

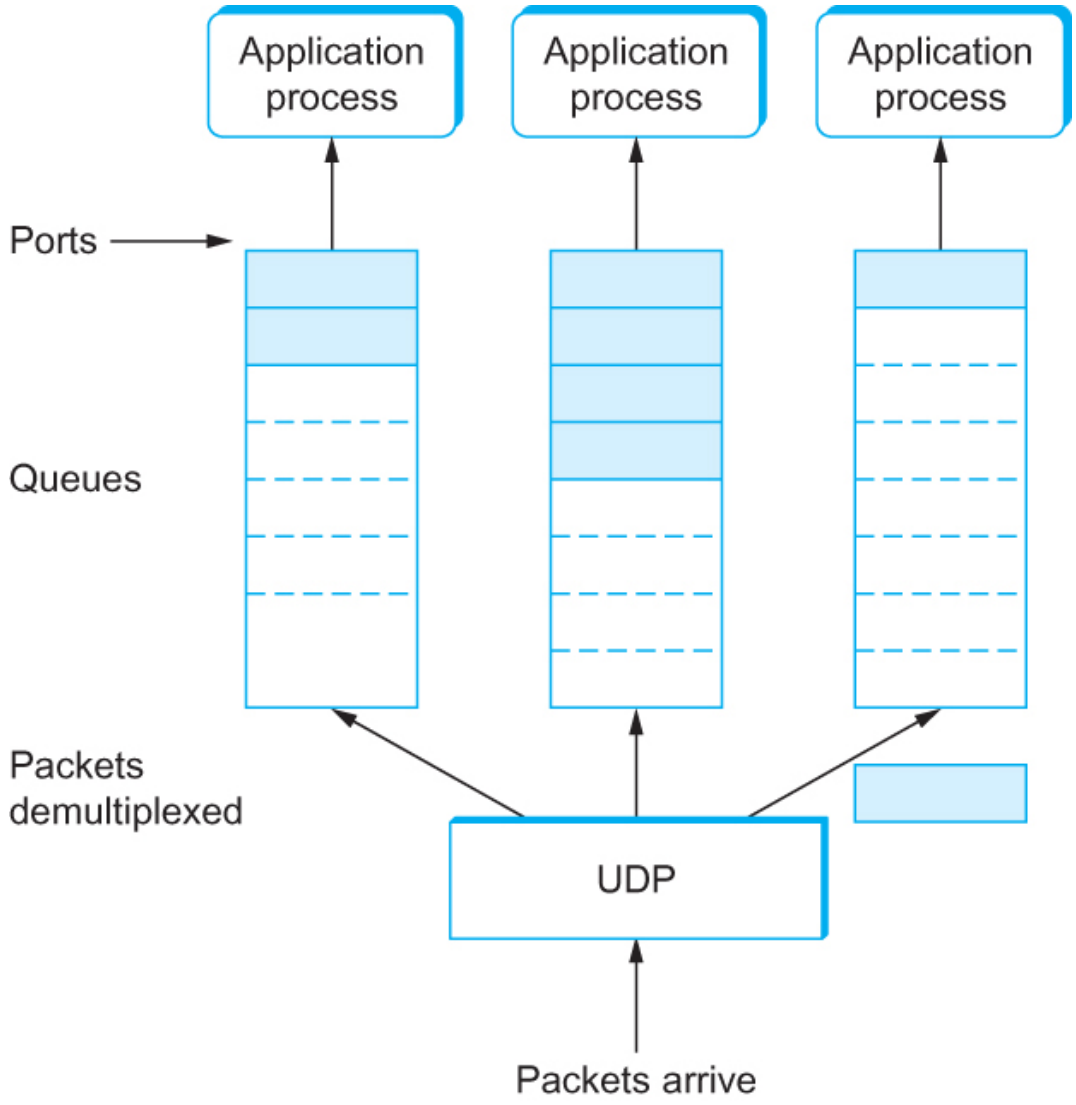
UDP - User Datagram Protocol

- serviço de comunicação entre processos
- permite a múltiplos processos em um mesmo hospedeiro compartilhar a rede
- mensagens são entregues de uma vez para a aplicação
- o tamanho das mensagens é limitado

UDP: Cabeçalho



UDP: Demultiplexação e Filas



UDP: Demultiplexação e Filas

- A demultiplexação do UDP é baseada em dois valores:
  - Endereço IP de Destino
  - Porta de Destino

TCP

Propriedades do TCP

- Transmissão confiável
- Controle de fluxo
- Controle de congestionamento

UDP versus TCP

- UDP
  - mensagens individuais
  - não confiável (perdas, fora de ordem, duplicações)
  - sem conexão
- TCP
  - fluxo de bytes
  - confiável
  - orientado à conexão
  - controle de fluxo
  - controle de congestionamento

Propriedades de um protocolo confiável

- Garantia de entrega das mensagens
- Mensagens entregues em ordem
- Sem mensagens duplicadas
- Suporta mensagens longas
- Sincronização entre transmissor e receptor
- Permitir ao receptor exercer controle de fluxo no transmissor
- Suportar várias aplicações em cada hospedeiro

O problema:

A rede subjacente:

- Perde pacotes (que carregam mensagens ou partes de mensagens)
- Entrega pacotes fora de ordem
- Duplica pacotes
- Envia pacotes de tamanho limitado
- Entrega pacotes com atrasos arbitrários

Problema $\Rightarrow$ prover para a aplicação um serviço de transporte confiável sobre redes com as características listadas acima.

TCP: fluxo de bytes confiável

- Confiável
- Orientado à conexão
- Orientado a fluxo de bytes

Controle de Fluxo e Congestionamento

- Controle de Fluxo $\Rightarrow$ impedir que o transmissor sobrecarregue o receptor
- Controle de Congestionamento $\Rightarrow$ prevenir excesso de dados na rede

Protocolo Fim a Fim

Problemas a serem resolvidos

- conexões lógicas entre processos executados em diferentes hospedeiros
- RTTs (latências) têm grande variação
- pacotes podem ser entregues fora de ordem e perdidos
- cada lado da conexão dever ser capaz de descobrir os recursos que o outro lado utiliza $\Rightarrow$ tamanho do buffer
- o TCP deve descobrir a capacidade da redes

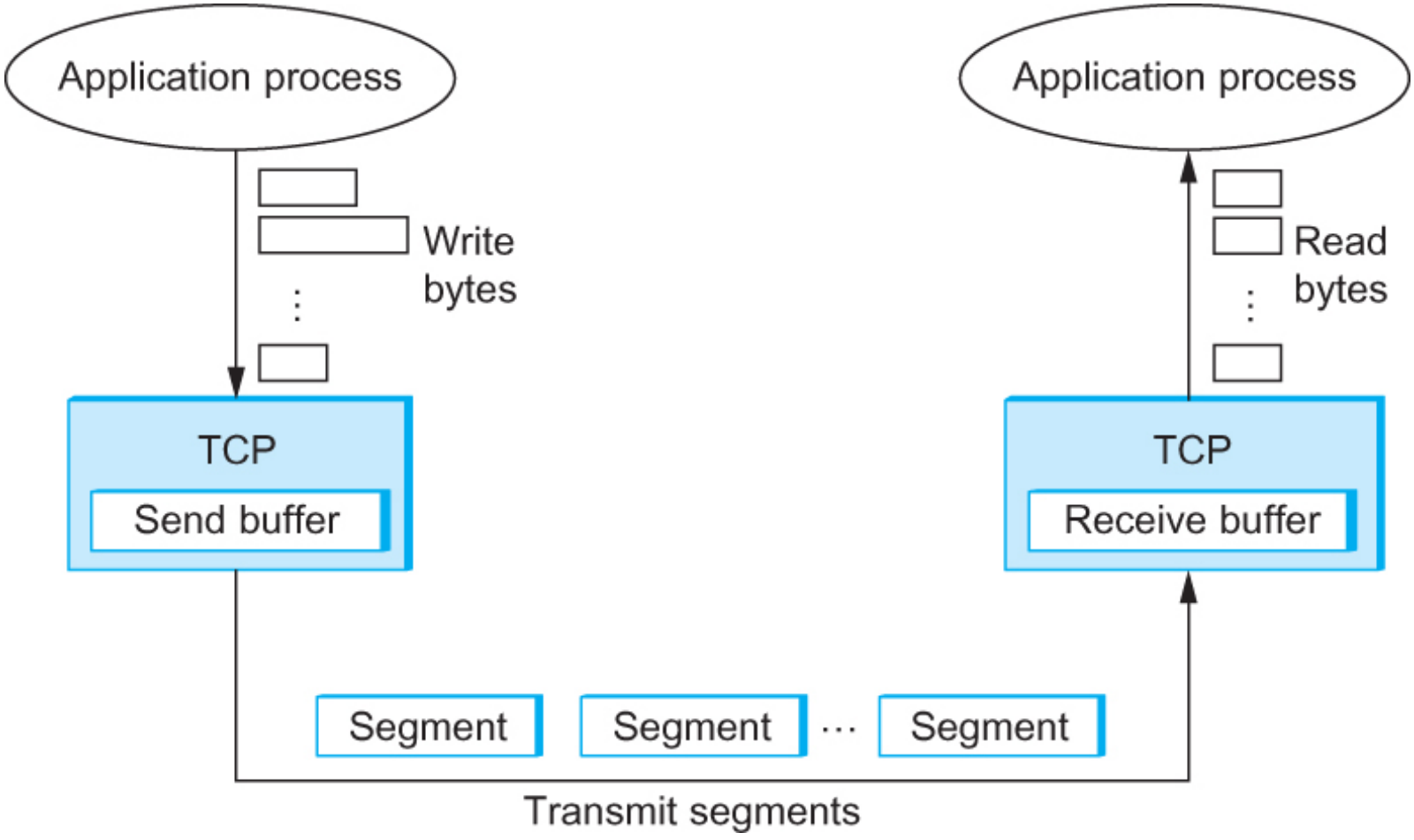
Segmentos TCP

- as aplicações enviam/recebem fluxos de bytes
- mas o TCP transmite blocos de bytes

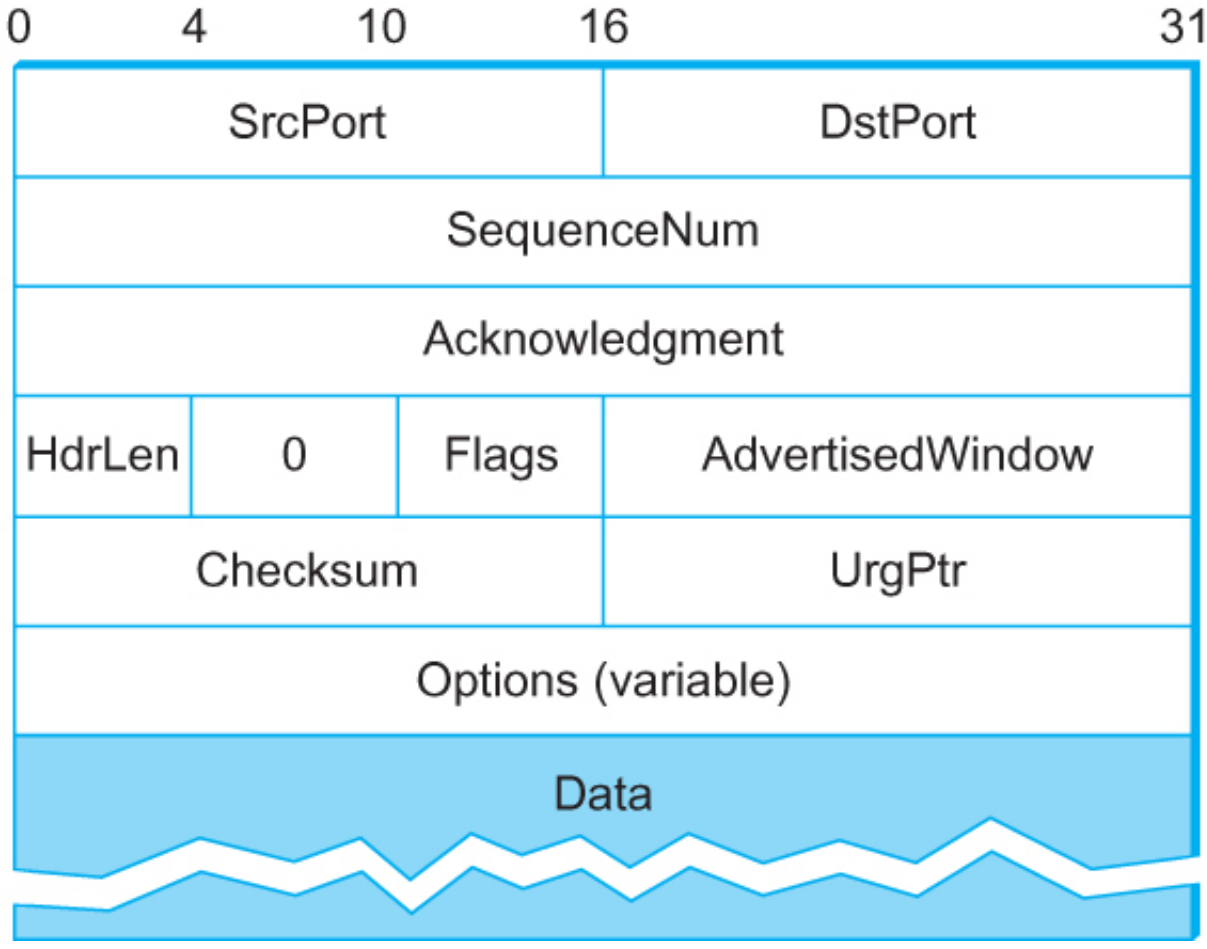
Segmentos TCP

- o transmissor armazena uma quantidade suficiente de bytes antes de enviar:
  - faz isso por questão de eficiência $\Rightarrow$ aproveitar a carga dos pacotes
- o destinatário grava os dados recebidos de uma pacote no buffer
- a aplicação lê os dados do buffer no seu próprio ritmo
- um bloco de dados com cabeçalho enviado pelo TCP é chamado de segmento

Segmento TCP



Cabeçalho TCP



Cabeçalho TCP

- `SrcPort` e `DstPort` identificam as portas
- `Acknowledgment`, `SequenceNum` e `AdvertisedWindow` são utilizados pelo algoritmo de janela deslizante
- cada byte possui um número de sequência $\Rightarrow$ `SequenceNum` contém o número de sequência do primeiro byte de dados contido no segmento
- `Acknowledgment` e `AdvertisedWindow` contém informações sobre o fluxo de dados no sentido oposto.

Cabeçalho TCP

- `Flags` contém informações de controle
  - `SYN`, `FIN`, `RESET`, `PUSH`, `URG`, `ACK`.
- `SYN` e `FIN` são utilizados no estabelecimento e fechamento da conexão
- `ACK` determina se o campo `Acknowledgment` carrega informação

Cabeçalho TCP

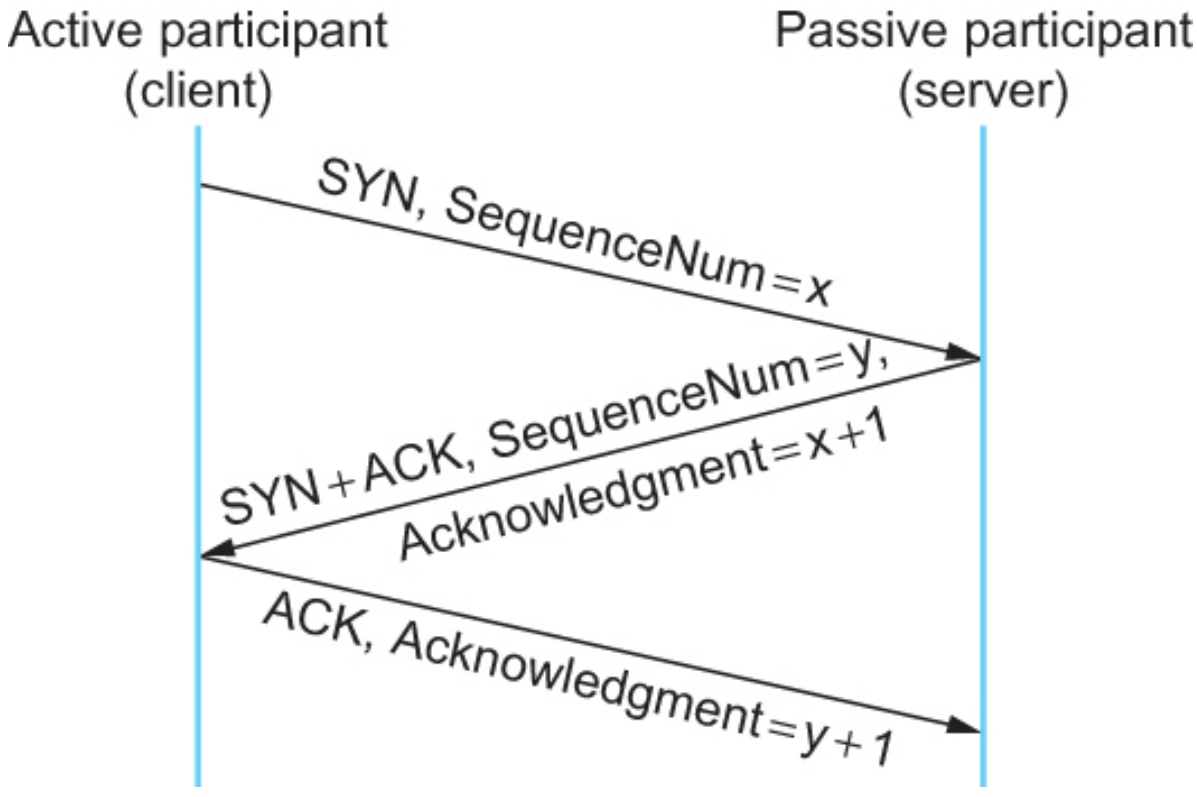
- `URG` marca dados urgentes.
  - permite a transmissão de dados fora da sequência
  - `UrgPtr` indica onde os dados não urgentes começam
- `PUSH` indica que o transmissor forçou o envio com um *push*
  - o transmissor notifica o processo receptor

Cabeçalho TCP

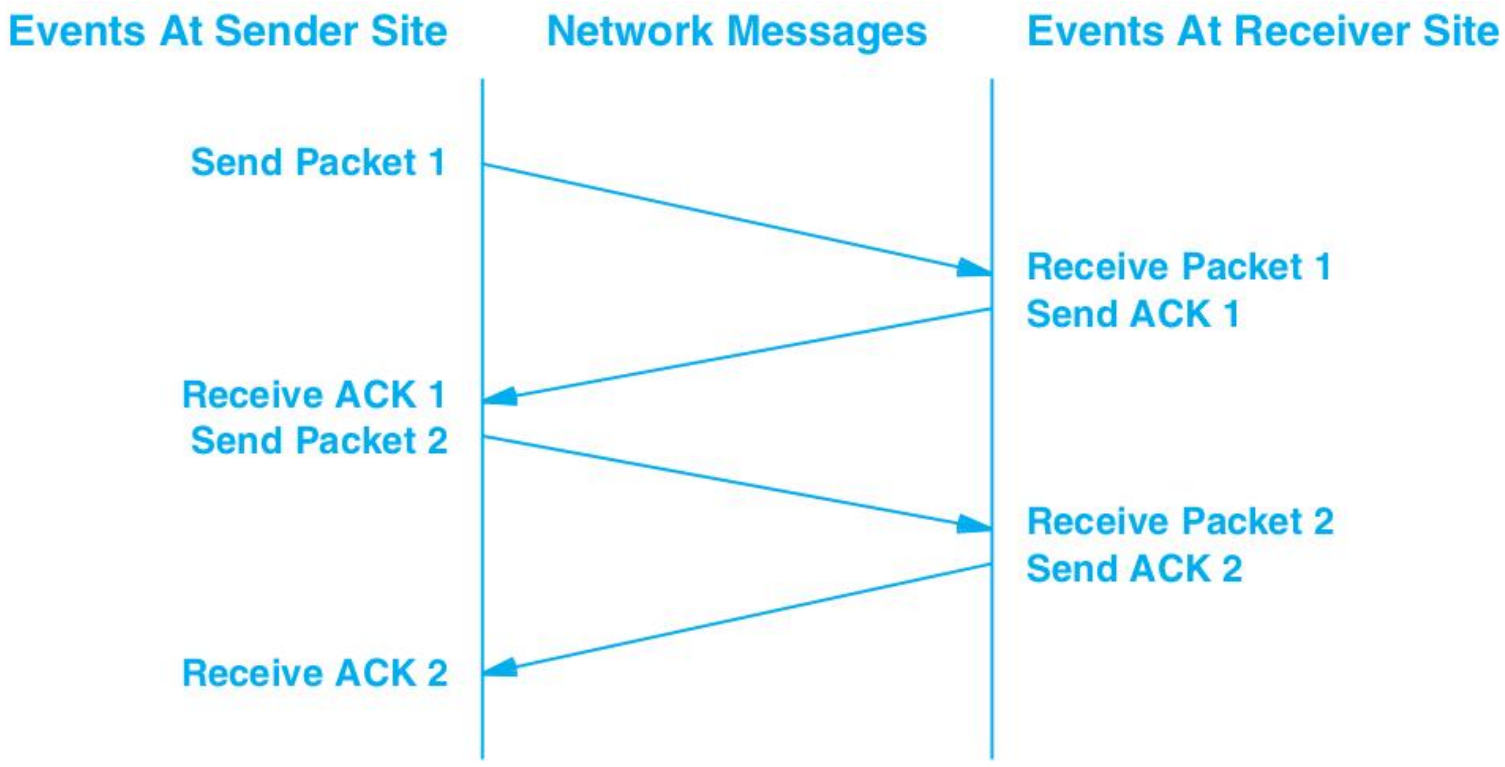
- `RESET` indica uma situação de exceção
  - e solicita o fim da conexão
- `checksum` é usado para verificar a integridade do segmento, e é aplicado ao:
  - cabeçalho
  - dados
  - pseudo-cabeçalho: endereços IP da origem e destino (cabeçalho IP)

Início de uma Conexão TCP

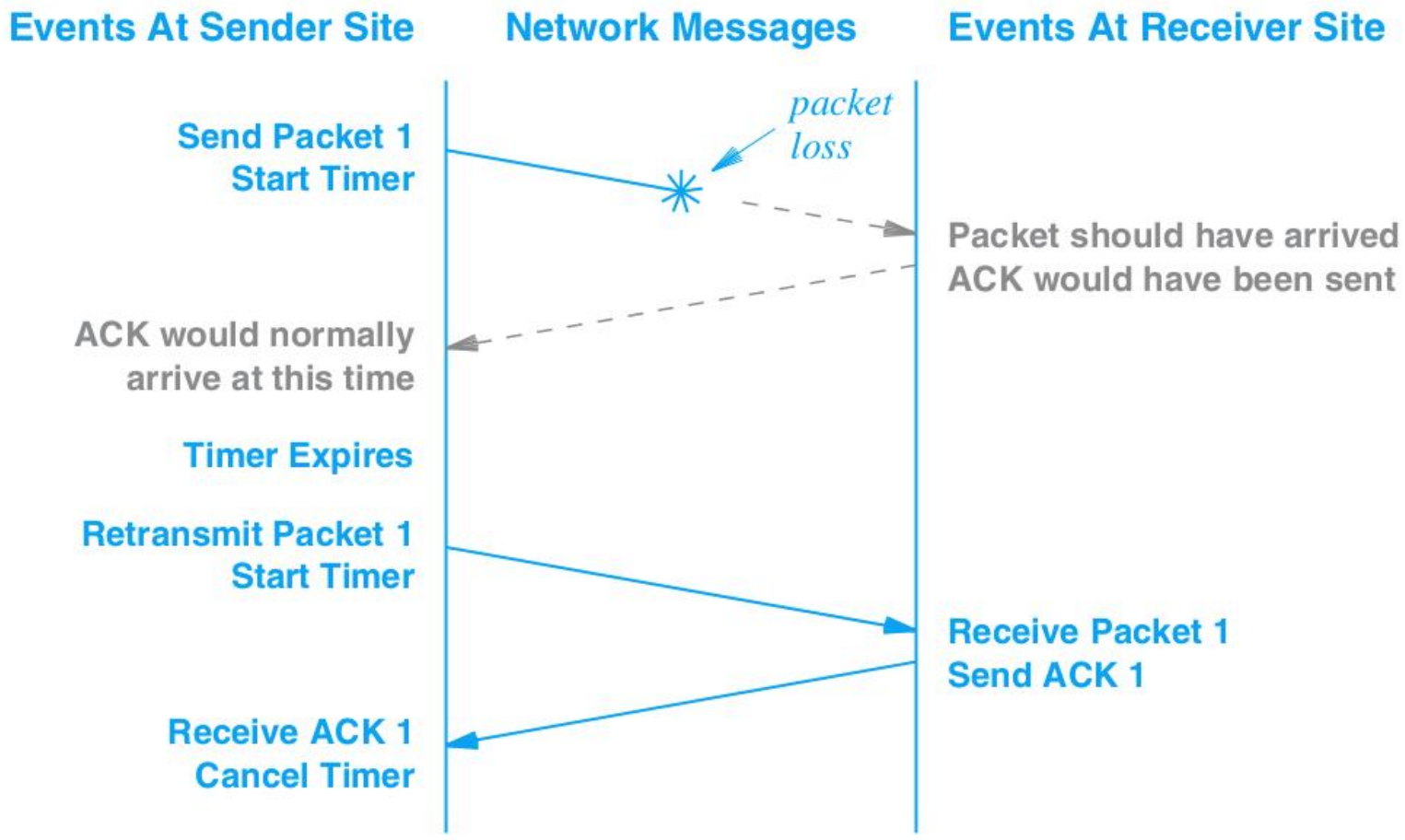
Handshake de 3 vias



Confirmações



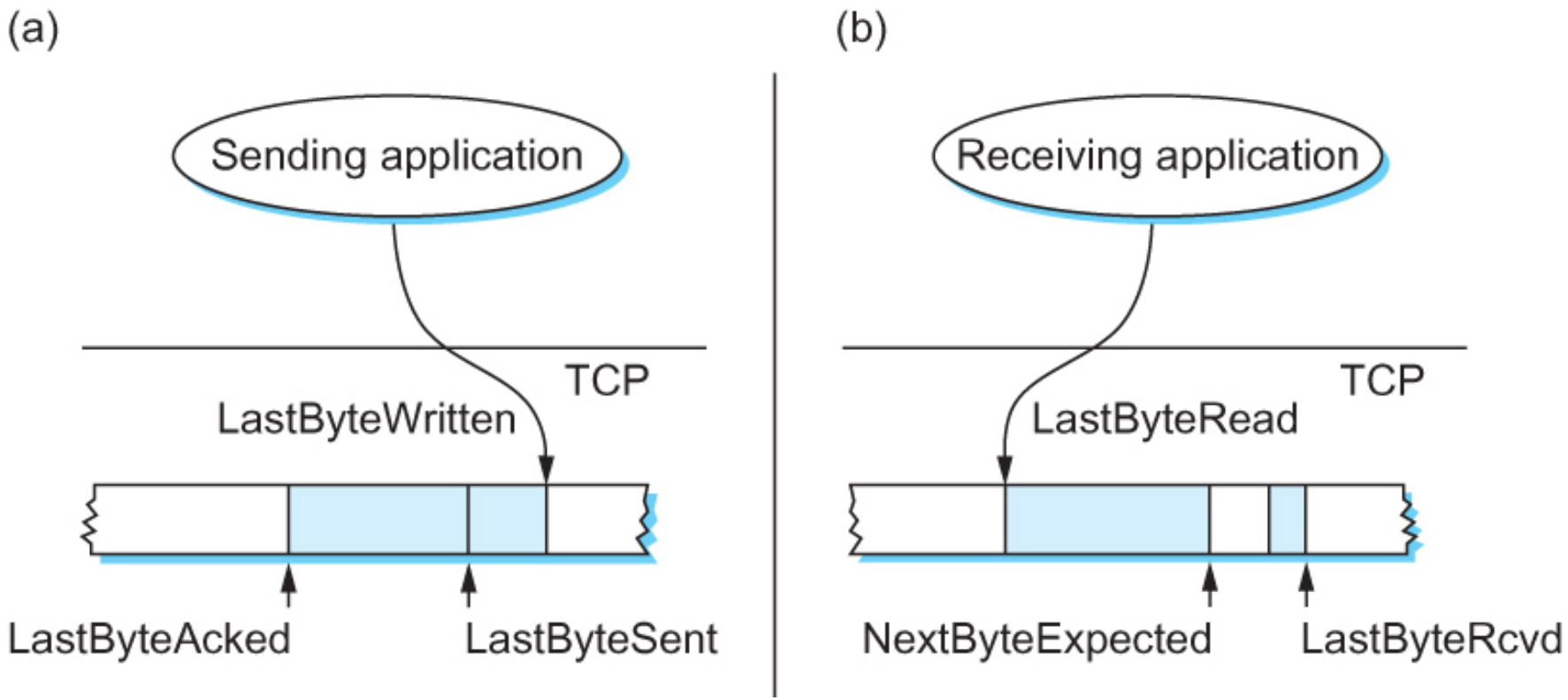
Confirmações e Retransmissões



Janela Deslizante do TCP

- Tem com objetivos
  - garantir entrega confiável
  - garantir entrega dos bytes em ordem
  - forçar o controle de fluxo entre o transmissor e o receptor

Janela Deslizante do TCP



Janela Deslizante do TCP

- Transmissor
  - $LastByteAcked \leq LastByteSent$
  - $LastByteSent \leq LastByteWritten$
- Receptor
  - $LastByteRead < NextByteExpected$
  - $NextByteExpected \leq LastByteRcvd + 1$