

TCP

*FEUP/MRSC/AMSR
MPR*

Bibliografia

- » Aula preparada com base nos seguintes documentos
 - L. Peterson, B. Davie, “Computer Networks – A Systems Approach”, Morgan Kaufmann, 2000 (Sec. 5.1 e 5,2)
 - Acetatos do autor do livro, L. Peterson, “Reliable Byte Stream (TCP)”

Introdução

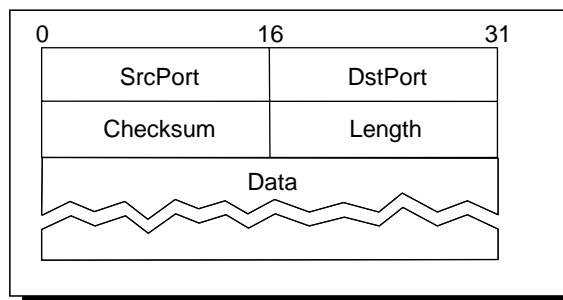
- ◆ Estabelecimento e terminação de ligação
- ◆ Mecanismo de janela deslizante
- ◆ Controlo adaptativo
- ◆ Timeout adaptativo

Protocolos Extremo a Extremo

- » *Rede best-effort*
 - Perde mensagens
 - Não preserva ordem de mensagens
 - Pode entregar de duplicados da mesma mensagem
 - Limita tamanho de mensagens
 - Entrega mensagens com atraso desconhecido
- » *Serviços extremo a extremo comuns*
 - Entrega garantida das mensagens
 - Entrega das mensagens pela ordem de envio
 - Entrega de uma cópia de cada mensagem, no máximo
 - Suporte de mensagens arbitrariamente grandes
 - Receptor controla o fluxo do emissor
 - Suporte de múltiplos processo de aplicação em cada computador

UDP – Desmultiplexador Simples

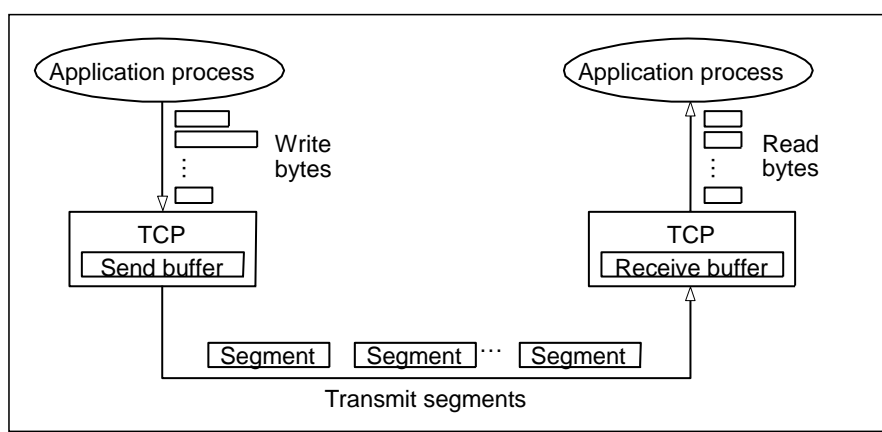
- ◆ Serviço de datagramas não ordenado, não fiável
- ◆ Suporta multiplexagem de processos de aplicação
- ◆ Não permite controlo de fluxo
- ◆ Pontos de acesso a serviço designados por *Porta*
- ◆ Formato do pacote



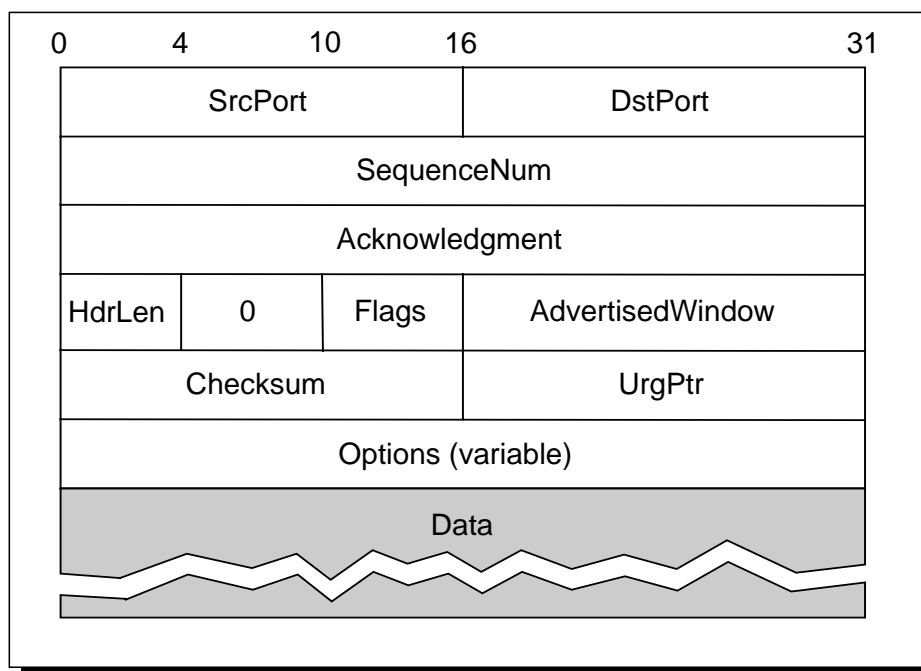
- ◆ *Checksum* opcional
 - » pseudo cabeçalho + cabeçalho UDP + dados

TCP

- ◆ Orientado às ligações
- ◆ Fluxo de bytes
 - » aplicação escreve bytes
 - » TCP envia segmentos
 - » aplicação lê bytes
- ◆ Full duplex
- ◆ Controlo de fluxo ←
- » Evita que emissor congestionue receptor
- ◆ Controlo de congestão ←
- » Evita que emissor congestionue a rede

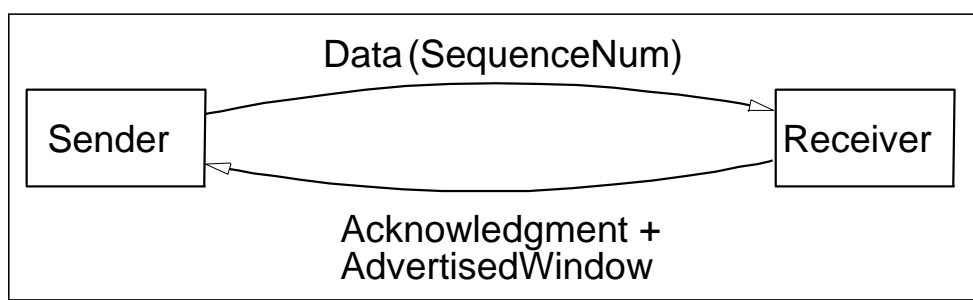


Formato do Segmento



Formato do Segmento

- ◆ Cada ligação identificada pelo vector
 - (SrcPort, SrcIPAddr, DstPort, DstIPAddr)
- ◆ Janela deslizante + controlo de fluxo
 - acknowledgment, SequenceNum, AdvertisedWindow



- ◆ Flags
 - SYN, FIN, RESET, PUSH, URG, ACK
- ◆ Checksum
 - pseudo cabeçalho + cabeçalhoTCP + dados

Estabelecimento e Terminação da Ligação

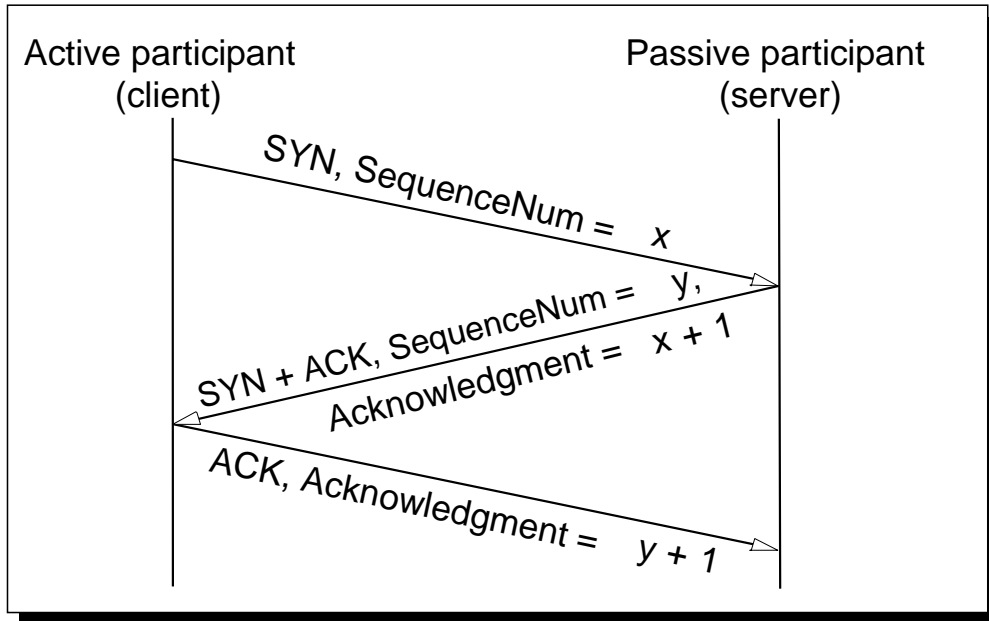
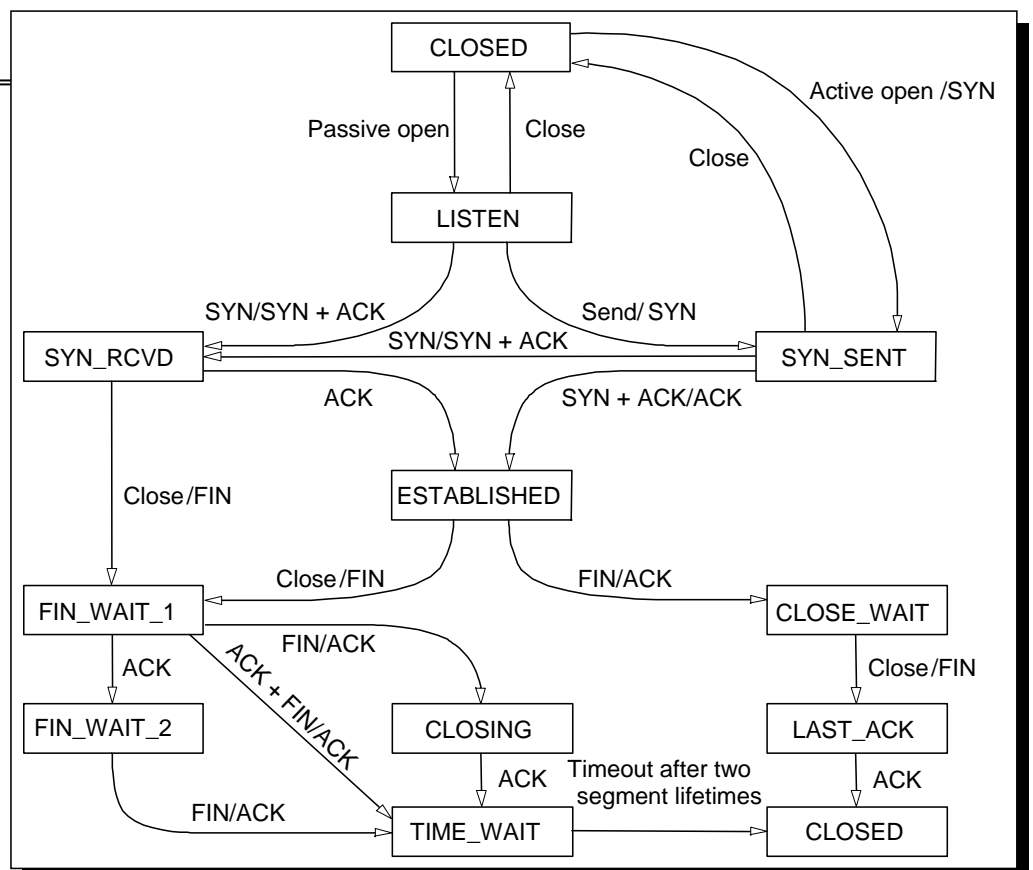
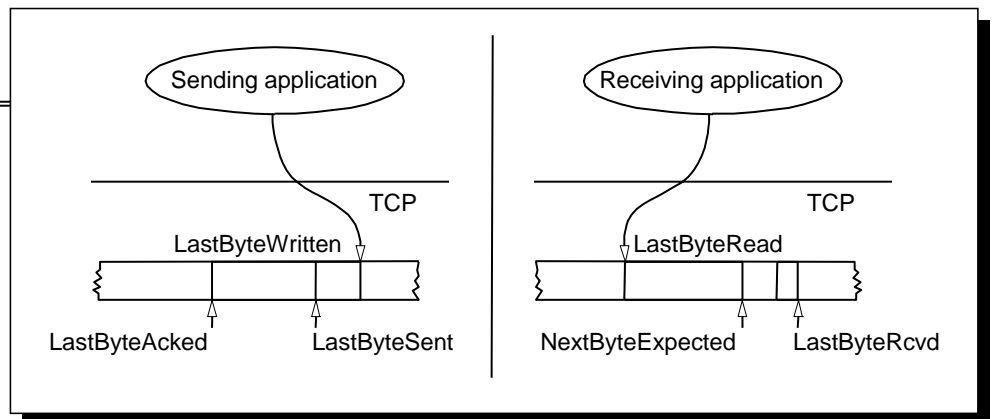


Diagrama de Transição de Estados



Janela Deslizante



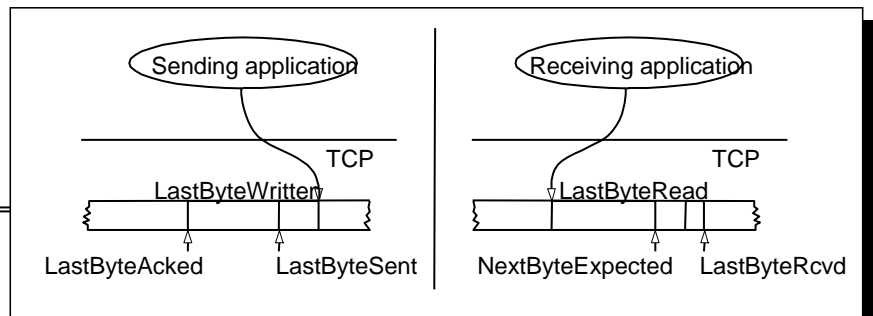
» No emissor

- $LastByteAacked \leq LastByteSent$
- $LastByteSent \leq LastByteWritten$
- Buferiza bytes entre **LastByteAacked** e **LastByteWritten**

» No receptor

- $LastByteRead < NextByteExpected$
- $NextByteExpected \leq LastByteRcvd + 1$
- Buferiza bytes entre **NextByteRead** e **LastByteRcvd**

Controlo de Fluxo



◆ Comprimento do buffer

- » no emissor → **MaxSendBuffer**
- » no receptor → **MaxRcvBuffer**

◆ No receptor

$$LastByteRcvd - LastByteRead \leq MaxRcvBuffer$$
$$AdvertisedWindow = MaxRcvBuffer - (LastByteRcvd - LastByteRead)$$

◆ No Emissor

$$LastByteSent - LastByteAacked \leq AdvertisedWindow$$
$$EffectiveWindow = AdvertisedWindow - (LastByteSent - LastByteAacked)$$
$$LastByteWritten - LastByteAacked \leq MaxSendBuffer$$

Envio de bloco se

$$(LastByteWritten - LastByteAacked) + y > MaxSenderBuffer$$

- ◆ Envia sempre **ACK** como resposta à chegada de um segmento
- ◆ Persiste, quando **AdvertisedWindow = 0**

Manter a Linha Cheia

- ♦ $RTT = 100 \text{ ms}$ ($v=5\mu\text{s/km}$, $L=10\,000 \text{ km}$)
- ♦ **AdvertisedWindow** de 16 bits ($2 \cdot 10^{16} = 65 \text{ koctetos}$)

Bandwidth	Delay x Bandwidth
E1 (2 Mbit/s)	25 kocteto
Ethernet (10 Mbit/s)	122 kocteto
E3 (34 Mbit/s)	425 kocteto
FDDI (100 Mbit/s)	1.2 Mocteto
STM-1 (155 Mbit/s)	1.8 Mocteto
STM-4 (622 Mbit/s)	7.4 Mocteto
STM-16 (2.5 Gbit/s)	31.2 Mocteto

Extensões do TCP

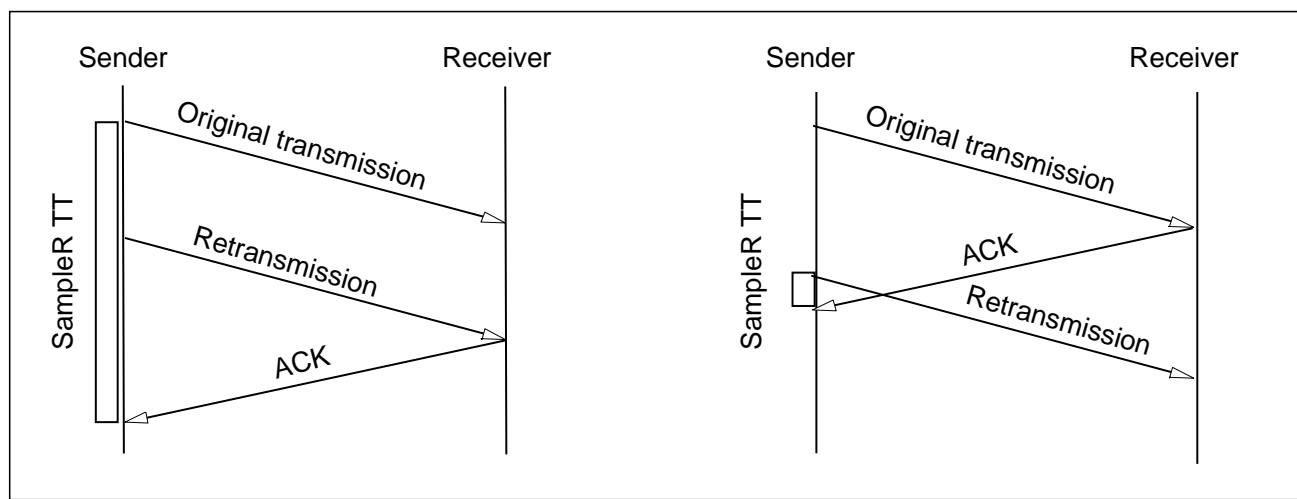
- ♦ Implementadas como opções do TCP
- ♦ Armazenamento de timestamps nos segmentos
- ♦ Shift (escala) da janela recomendada

Retransmissão Adaptativa (Algoritmo Original)

- ◆ RTT → Round Trip Time (tempo de ida e volta)
- ◆ Medida de **sampleRTT** para cada par **segmento/ACK**
- ◆ Cálculo da média pesada do RTT
 - » $RTT = a \times RTT + (1-a) \times SampleRTT$
 - a em [0.8, 0.9]
- ◆ $TimeOut = 2 \times RTT$

Melhoria de Karn/Partridge

- ◆ Não mede **sampleRTT** em caso de retransmissão
- ◆ Duplica valor do timeout em cada retransmissão



Melhoria de Jacobson/ Karels

- ◆ Novo método de cálculo do RTT
 - » $Dif = sampleRTT - RTT$
 - » $RTT = RTT + (p \times Dif)$
 - » $Desv = Desv + p (|Dif| - Desv)$
 - p em [0,1]
- ◆ Considera variância no cálculo do timeout
 - » $Timeout = m \times RTT + f \times Desv$
 - » $m = 1, f = 4$
- ◆ Mecanismo de timeout preciso
 - » importante para controlo de congestão