

Redes de Computadores

Camada de Transporte

TCP

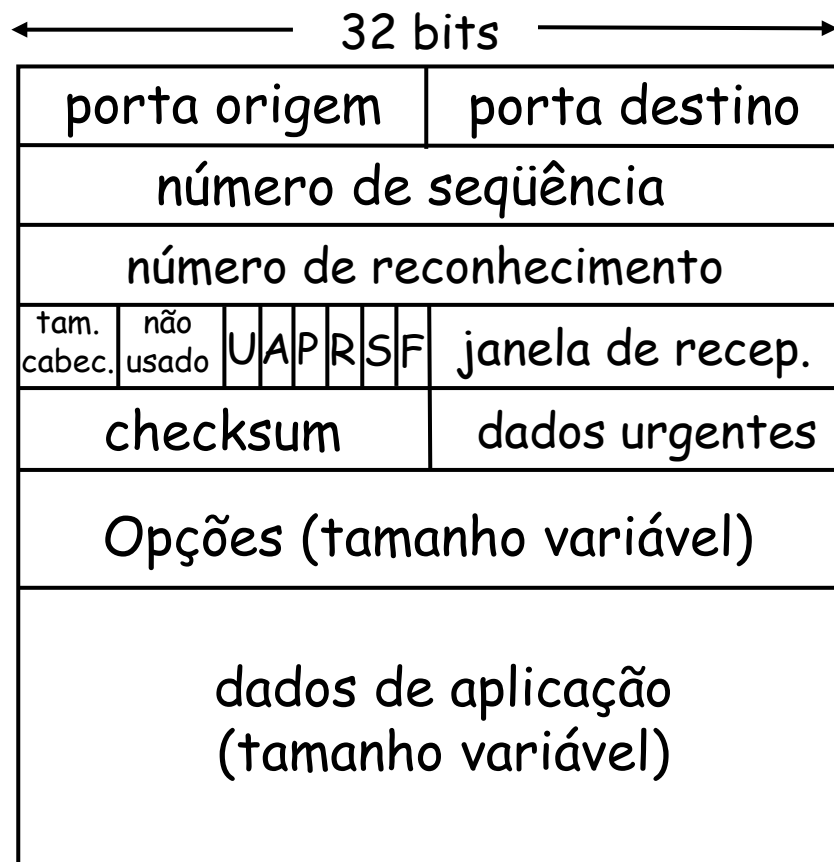
TCP

- RFC's 793, 1122, 1323, 2018 e 2581;
- Orientado a conexão;
- Serviço *full-duplex*;
- Transmissão confiável de dados;
- Conexão ponto a ponto (apresentação de 3 vias);
- Controle de fluxo;
- *Pipelined*;
- *Buffers* de transmissão e recepção.

Rede de Computadores

Flags:

- URG, urgente
- ACK, indica q o valor no campo de reconhecimento é válido;
- PSH, o destinatário deve passar os dados para a camada superior imediatamente;
- RST, SYN e FIN, são utilizados para iniciar, sincronizar e encerrar uma conexão.



Números de Sequência

- **MSS** (*Maximum Segment Size* – tamanho máximo do segmento), quantidade máxima de dados que pode ser colocada em um segmento;
- **MTU** (*Maximum Transmission Unit* – unidade máxima de transmissão), tamanho máximo de um quadro da camada de enlace;
- Valores comuns de MTU são 1.460 bytes, 536 bytes e 512 bytes;
- Número de sequência para um segmento é o número do primeiro byte do segmento;

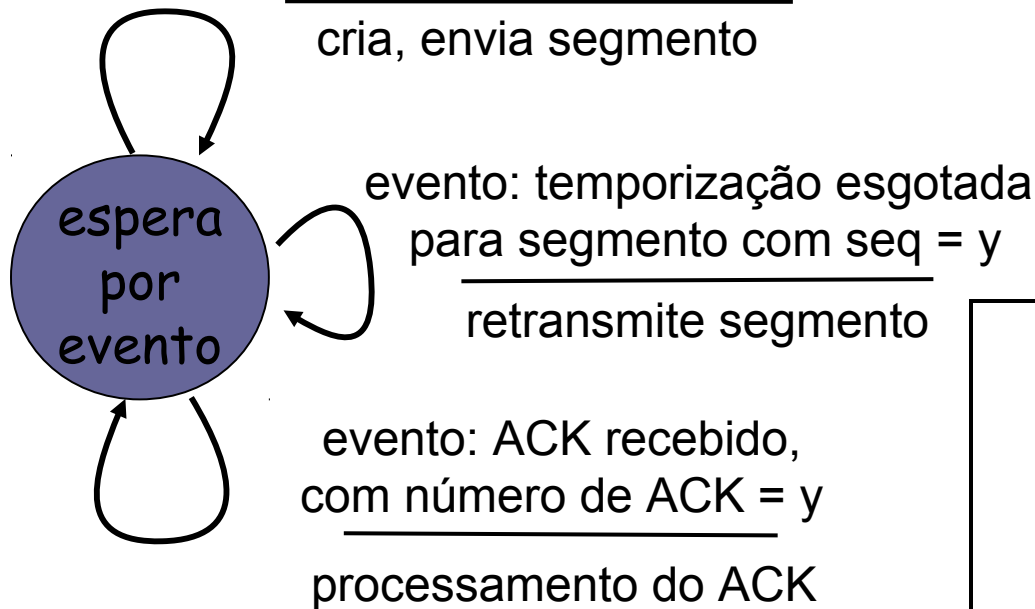
Números de Sequência

- O número de reconhecimento é o número do primeiro byte do segmento que está sendo aguardado;
- O primeiro número de sequência é escolhido de forma aleatória para evitar que atrasos no recebimento antigo afete um recebimento novo;
- Reconhecimentos cumulativos;

Transferência Confiável de Dados

evento: dados recebidos
da aplicação acima

cria, envia segmento



transmissor simplificado,
assumindo que não há
controle de fluxo nem de
congestionamento.

Rede de Computadores

Transferência Confiável de Dados

```
00  sendbase = initial_sequence number
01  nextseqnum = initial_sequence number
02
03  loop (forever) {
04      switch(event)
05          event: dados recebidos da aplicação acima
06              cria segmento TCP com número de seqüência
nextseqnum
07              inicia temporizador para segmento nextseqnum
08              passa segmento ao IP
09              nextseqnum = nextseqnum + length(data)
10  event: esgotamento de temp. para segmento com
seq = y
11      retransmite segmento com seq = y
12      calcule nova temporização para o segmento y
13      reinicia temporizador para número de seqüência
y
```

```
14      event: ACK recebido, com valor y no campo de ACK
15          if (y > sendbase) { /* ACK cumulativo de todos os
dados até y */
16              cancela todas as temporizações para
segmentos com seq < y
17              sendbase = y
18          }
19          else { /* um ACK duplicata para um segmento já
reconhecido */
20              incrementa numero de duplicatas de ACKs
recebidas para y
21              if (numero de duplicatas de ACKS recebidas
para y == 3) {
22                  /* TCP fast retransmit */
23                  reenvia segmento com número de seqüência
y
24                  reinicia temporização para segmento y
25              }
26          } /* end of loop forever */
```

Rede de Computadores

Política de Geração de ACK's

Event0

segmento chega em ordem,
não há lacunas,
segmentos anteriores já aceitos

Ação do TCP Receptor

ACK retardado. Espera até 500ms
pelo próximo segmento. Se não chegar,
envia ACK

segmento chega em ordem,
não há lacunas,
um ACK atrasado pendente

imediatamente envia um ACK
cumulativo

segmento chega fora de ordem
número de seqüência chegou
maior: gap detectado

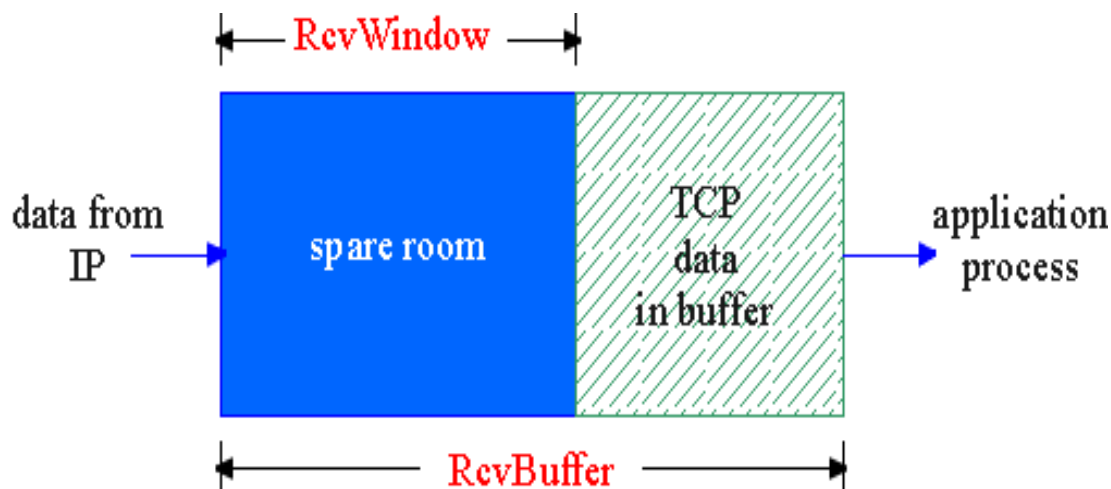
envia ACK duplicado, indicando número
de seqüência do próximo byte esperado

chegada de segmento que
parcial ou completamente
preenche o gap

reconhece imediatamente se o segmento
começa na borda inferior do gap

Controle de Fluxo

- Eliminar a possibilidade de o remetente saturar o *buffer* do destinatário;
- Janela de recepção == espaço de *buffer* livre;



Estabelecimento de uma Conexão

- O cliente estabelece uma conexão antes de iniciar a transferência de dados;
- Inicializa variáveis: controle de fluxo, números de sequência;
- A conexão é estabelecida através de uma **comunicação de 3 vias** (*three-way handshake*).

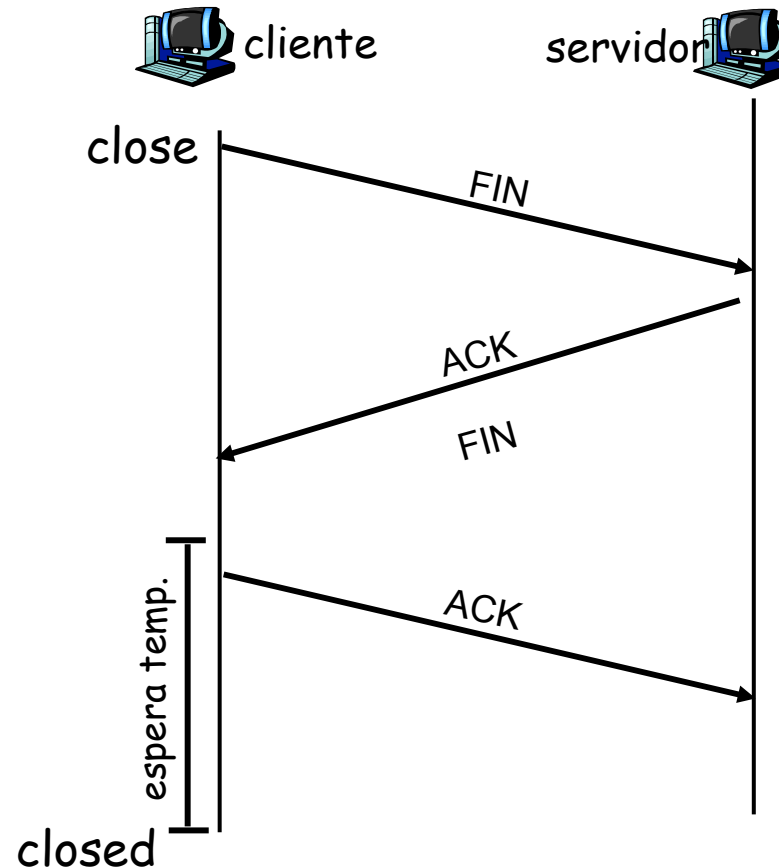
Estabelecimento de uma Conexão

● Comunicação de 3 vias:

1. 1º) o cliente envia um segmento sem nenhum dado de aplicação, o flag SYN é 1, é escolhido um número de sequência inicial para o cliente e enviado;
- 2.
3. 2º) o servidor recebe o segmento SYN e envia um SYNACK (flag SYN é 1 e ACK é 1), é alocado *buffers* e criado um número de sequência inicial para o servidor e enviado;
- 4.
5. 3º) o cliente recebe o segmento SYNACK, é feita a alocação de *buffers* e variáveis para controle do estado da conexão.

Rede de Computadores

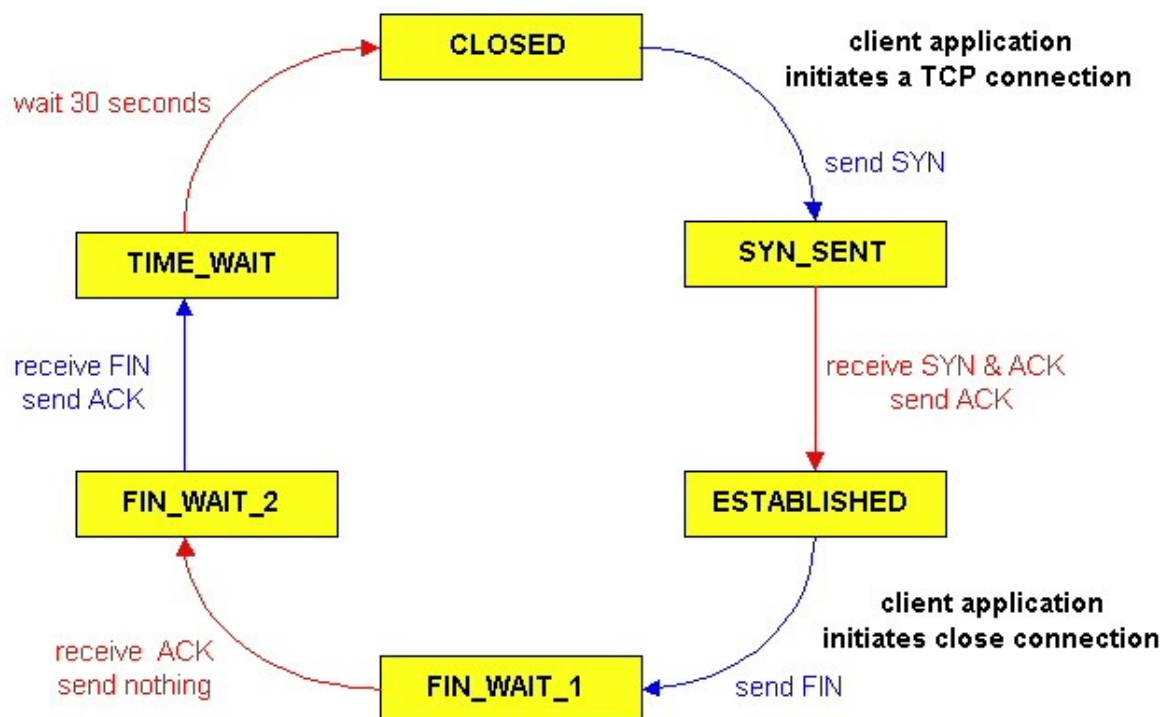
Terminando uma Conexão



Rede de Computadores

Estados do TCP

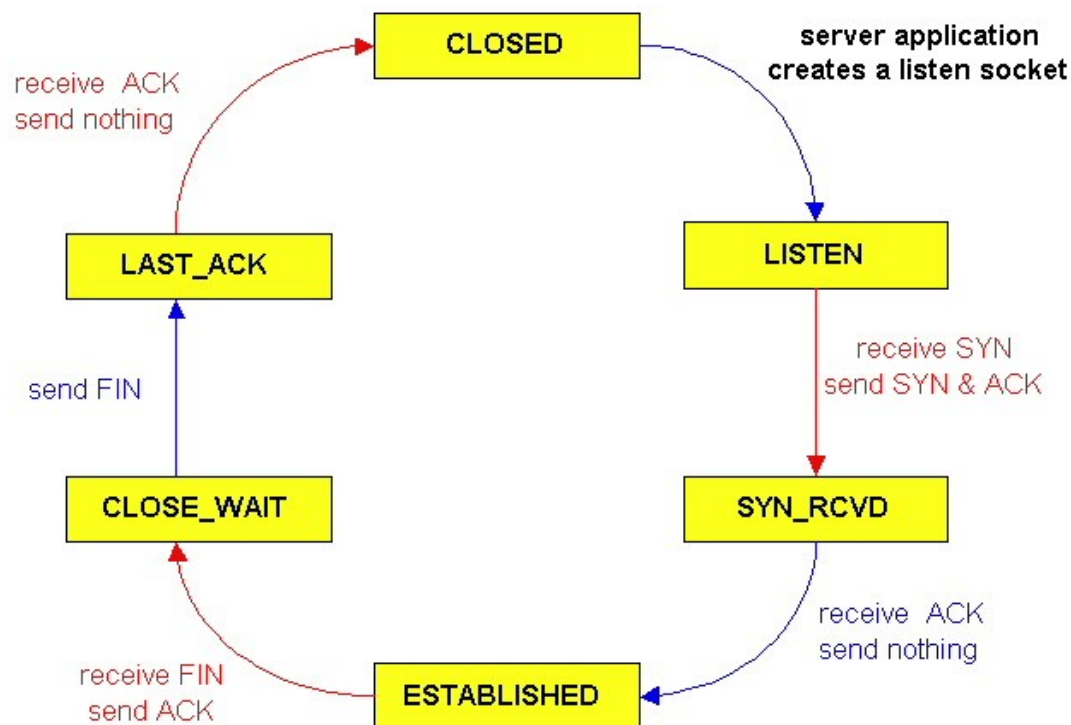
● Sequência de Estados do Cliente



Rede de Computadores

Estados do TCP

● Sequência de Estados do Servidor



Controle de Congestionamento

- Causa, demasiadas fontes tentando enviar dados a uma taxa muito alta.
- O TCP não resolve, mas trata de um dos sintomas, perda de segmentos;
- Um outro sintoma é um grande atraso na entrega;
- Congestionamento != Controle de fluxo.

Controle de Congestionamento

● Controle de congestionamento fim-a-fim:

- A camada de rede não oferece nenhum suporte à camada de transporte;
- A existência de um congestionamento tem de ser detectada com base na observação do comportamento da rede;
- A perda de segmentos (3) é encarada como indicação de congestionamento;
- Reduz o tamanho da janela.

● Controle de congestionamento assistido pela rede:

- A camada de rede fornece realimentação específica de informações ao remetente a respeito do estado de congestionamento;