

Diálogo Cliente-Servidor com TCP

Fundamentos de Redes de Computadores

Prof.: Fernando William Cruz



Possibilidades

- Servidor **orientado à conexão** / não orientado à conexão
- Servidor Concorrente / **Iterativo**
- Servidor *Stateful* / *Stateless*
 - Eficiência X Confiabilidade
- Servidor **apenas como Servidor** ou também como Cliente
- Diálogos entre Cliente e Servidor em geral, envolvem:
 - Protocolo ou regra de comunicação → **TCP**
 - Endereços do cliente e do servidor → **sockaddr_in**

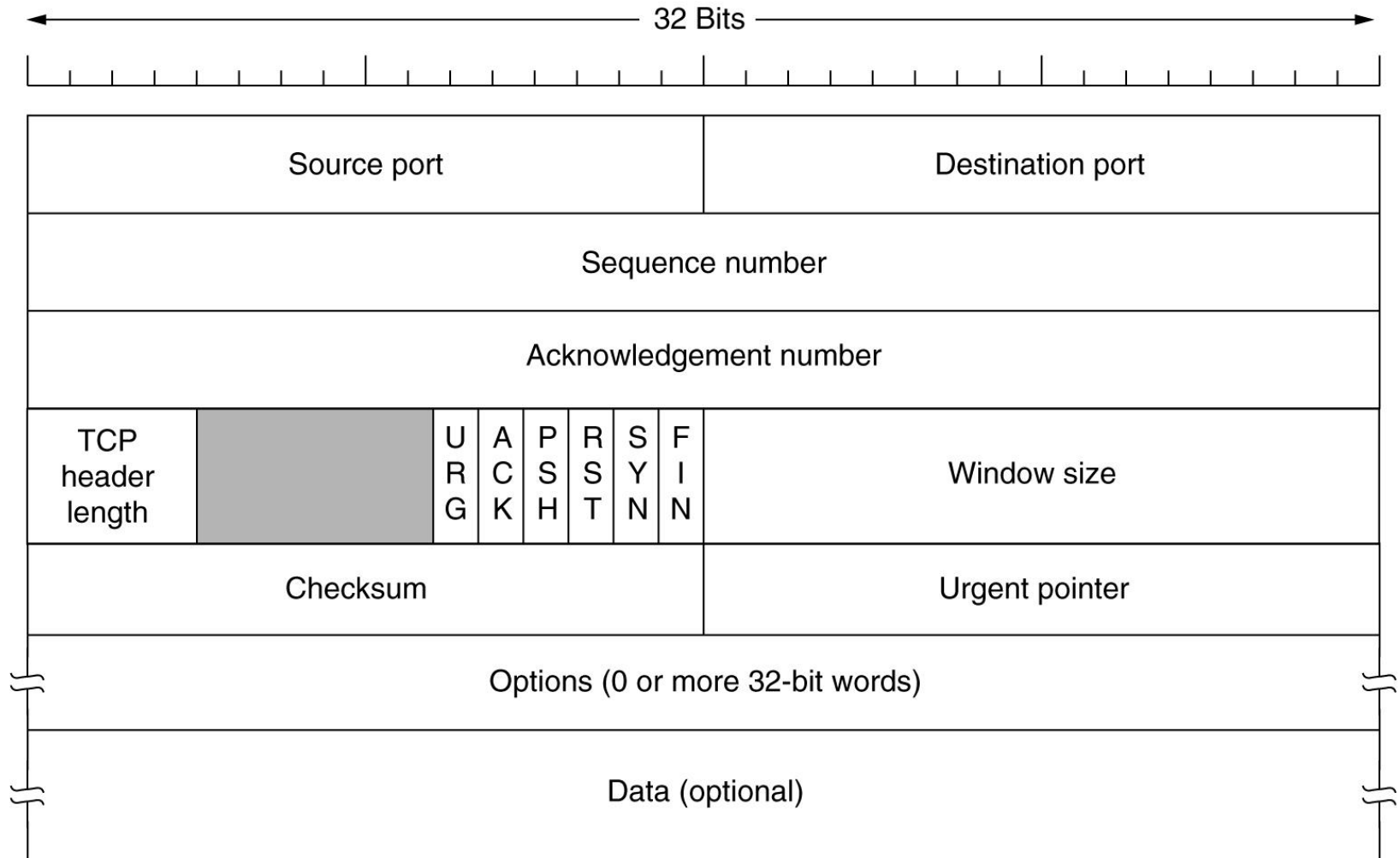


TCP (RFC 793)

- Orientado à conexão
- Provê confiabilidade na transmissão
 - Seqüenciamento
 - Retransmissão
- Provê controle de fluxo



Cabeçalho TCP





Conexão TCP

↖ Ciclo de vida de uma conexão TCP

- **Estabelecimento da conexão**
 - *Three-way handshake*
- **Transferência de dados**
 - Controle de fluxo
 - Sequenciamento
 - Retransmissão
- **Encerramento da conexão**



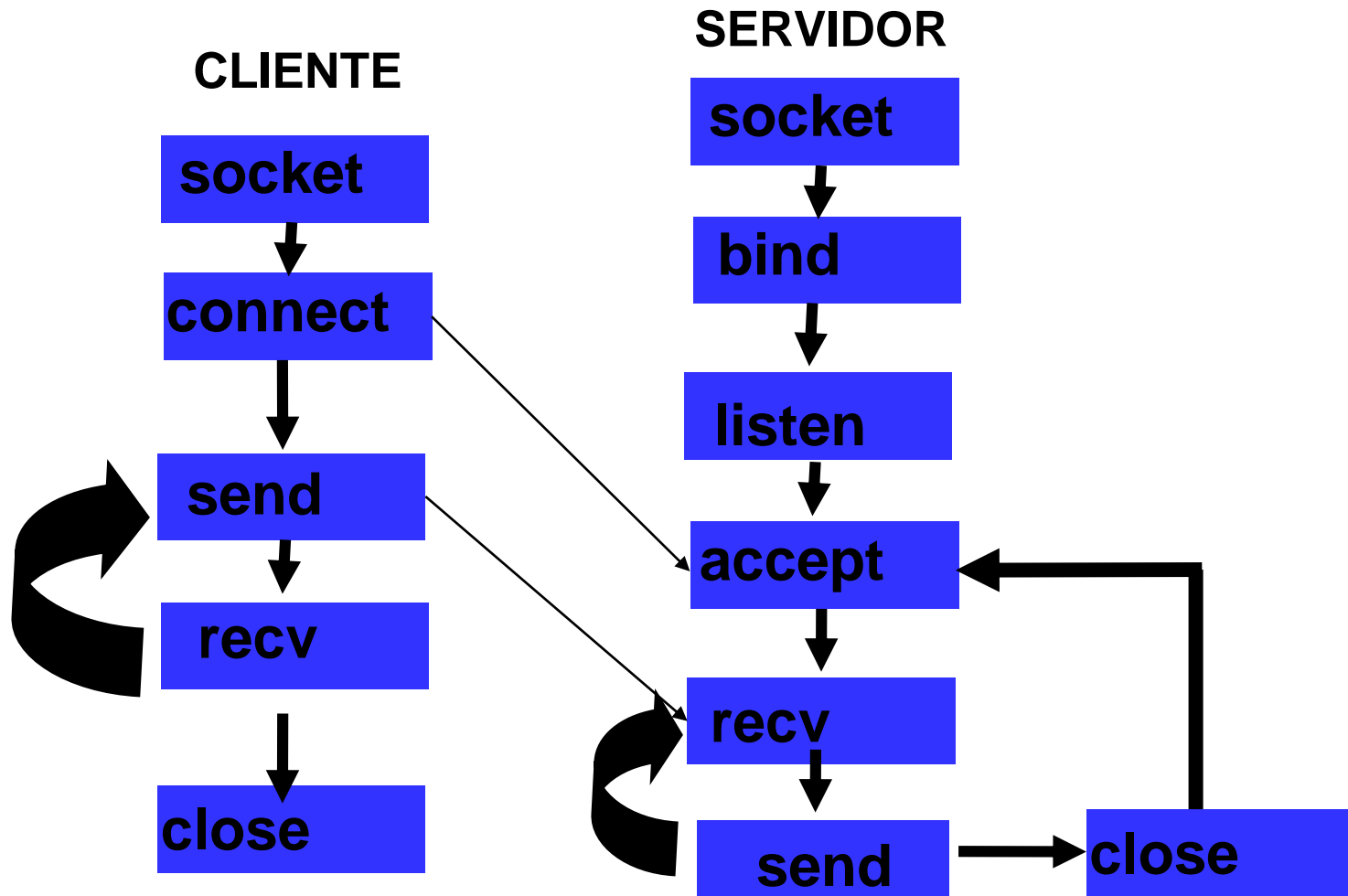
Three-way Handshake

- Servidor deve estar “escutando” em uma porta
- Cliente envia um segmento SYN (SYN=1;ACK=0)
- Servidor envia um segmento SYN-ACK (SYN=1;ACK=1)
- Cliente envia uma ACK de volta (SYN=0;ACK=1)



Interação orientada à conexão

- Usa sockets TCP nas entidades comunicantes
- Funções envolvidas (mais comuns)





API Socket: connect

result = connect (socket, endRemoto, tamEndRemoto)

- Permite um cliente TCP iniciar uma conexão com endRemoto
- Inicia o 3-way handshake
- Resultado indica se a conexão foi estabelecida com sucesso
- Com o connect, criamos um socket ponto-a-ponto ou um socket *conectado*
 - Conexão TCP é **ponto-a-ponto**



API Socket: send

result = send (socket, msg, msglen, flags)

- Envia informação para outra máquina
 - Cadê o endereço de destino (endpoint remoto)?
- Retorna o número de bytes enviados



API Socket: recv

result = recv (socket, buffer, bufferlen, flags)

- Recebe informação de outra máquina a partir de um socket
- Retorna o número de bytes recebidos



API Socket: close

result = close (socket)

- Encerra uma comunicação e elimina o socket



API Socket: listen

result = listen (socket, tamFila)

- Torna o socket passivo, isto é, pronto para aceitar conexões
- tamFila é o tamanho da fila de conexões TCP que chegam
 - Número de requisições de conexão que devem ser armazenadas enquanto o servidor manipula outra requisição



API Socket: accept

novoSocket = accept (socket, endRemoto, tamEndRemoto)

- Extrai a próxima requisição de conexão da fila associada a *socket*
- A requisição vem de endRemoto
- Retorna um novo socket através do qual o servidor conversará com o cliente
 - O socket original é usado apenas para armazenar requisições de conexão adicionais