

Redes de Computadores

Camada de Transporte





Agenda

Protocolos

UDP

TCP

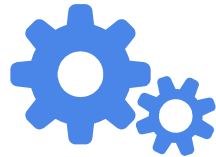
Serviços

Controle de Fluxo

Controle de Congestionamento

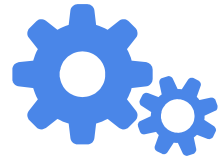
Protocolos

Sem Orientação a Conexão (UDP), Com Orientação a Conexão (TCP)



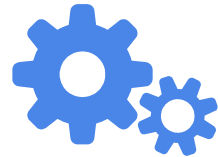
Protocolo UDP

- User Datagram Protocol
- Protocolo de transporte da Internet “sem gorduras”, “sem frescuras”
 - Nenhuma garantia para as camadas superiores
 - A aplicação estará interagindo quase diretamente com o IP
- Serviço “best effort”, segmentos UDP podem ser:
 - Perdidos
 - Entregues fora de ordem para a aplicação



Protocolo UDP

- Não orientado a conexão (sem verificações)
 - Não há apresentação entre o UDP transmissor e o receptor
 - Cada segmento UDP é tratado de forma independente dos outros
 - Utilizado nos ataques por enviar diversos pacotes sem apresentação
- Por que existe UDP?

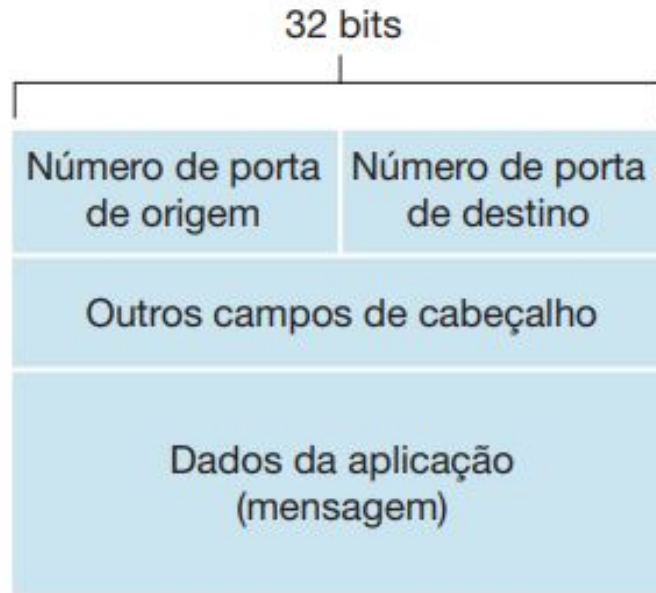


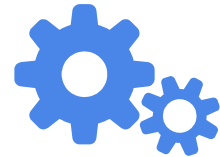
Protocolo UDP

- Muito usado por aplicações de multimídia contínua (streaming)
 - Tolerantes à perda
 - Sensíveis à taxa
 - Rapidez (menor pacote)
- Outros usos do UDP: DNS, SNMP



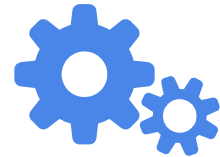
Protocolo UDP





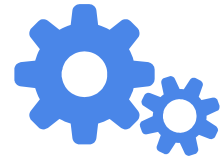
Protocolo UDP

- Para haver segurança usando UDP
 - Implementação de transferência confiável sobre UDP na camada de aplicação
 - Recuperação de erro específica de cada aplicação



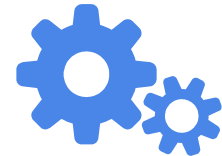
Protocolo UDP

- UDP Checksum: detecta “erros” (ex.: bits trocados) no segmento transmitido
 - Transmissor:
 - Trata o conteúdo do segmento como sequência de inteiros de 16 bits
 - Checksum: soma (complemento de 1 da soma) do conteúdo do segmento
 - Transmissor coloca o valor do checksum no campo de checksum do UDP



Protocolo UDP

- UDP Checksum
 - Receptor:
 - Computa o checksum do segmento recebido
 - Verifica se o checksum calculado é igual ao valor do campo checksum:
 - NÃO - erro detectado
 - SIM - não há erros. Mas talvez haja erros apesar disso?



Protocolo UDP

- Exemplo: Enviar os bits 1100 e 0111

Transmissor

1100

+ 0111

1 + 0011

0100 -> 1011

Receptor

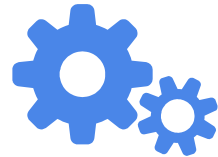
1100

+ 0111

1011

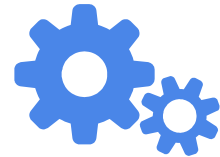
1 + 1110

1111 (envio correto)



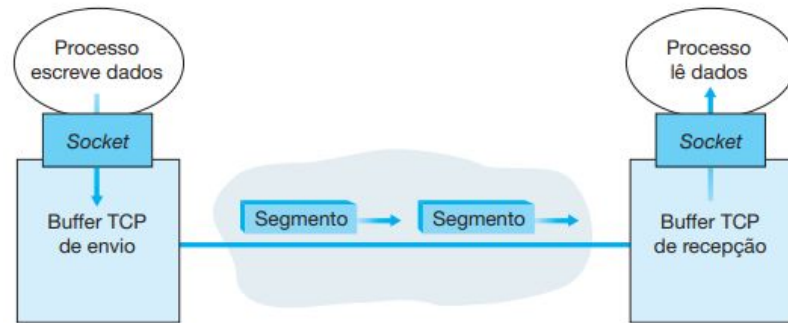
Protocolo TCP

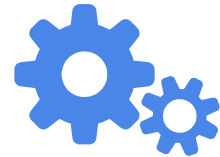
- Ponto-a-ponto: Um transmissor, um receptor
- Confiável, seqüencial byte stream: Não há contornos de mensagens
- Pipelined: (transmissão de vários pacotes sem confirmação)
- Controle de congestionamento e de fluxo definem tamanho da janela



Protocolo TCP

- Buffers de transmissão e de recepção
- Dados full-duplex: Transmissão bidirecional na mesma conexão
- MSS: maximum segment size



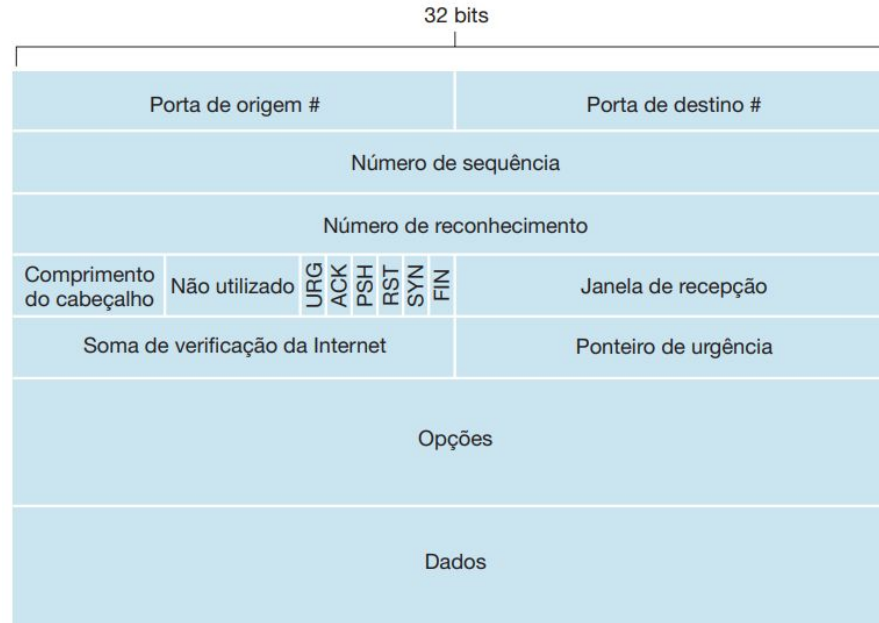


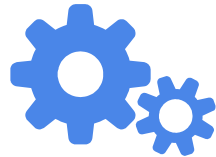
Protocolo TCP

- Orientado à conexão:
- Apresentação (troca de mensagens de controle) inicia o estado do transmissor e do receptor antes da troca de dados
- Controle de fluxo: Transmissor não esgota a capacidade do receptor



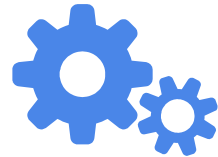
Protocolo TCP





Protocolo TCP

- URG: dados urgentes (pouco usados)
- ACK: campo de ACK é válido
- PSH: produz envio de dados (pouco usado)
- RST, SYN, FIN: estabelecimento de conexão (comandos de criação e término)
- Internet checksum (como no UDP)
- O número de bytes receptor está pronto para aceitar contagem por bytes de dados (não segmentos!)

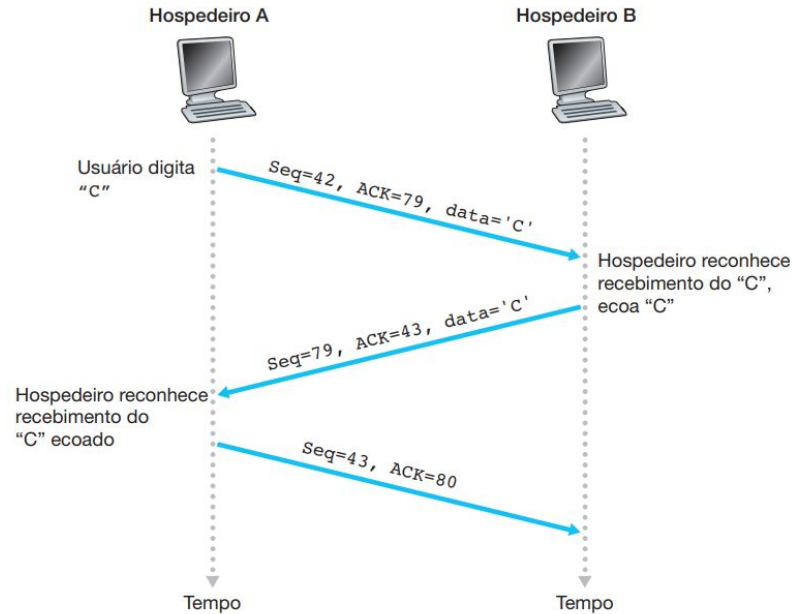


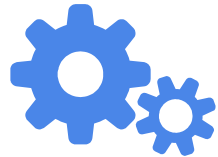
Protocolo TCP

- Números de seqüência:
 - Número do primeiro byte nos segmentos de dados
- ACKs:
 - Número do próximo byte esperado do outro lado ACK cumulativo
- Como o receptor trata segmentos fora de ordem?
 - A especificação do TCP não define, fica a critério do implementador



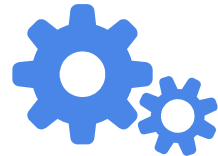
Protocolo TCP





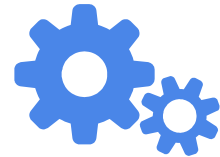
Protocolo TCP

- Estabelecimento de Conexão (Protocolo)
 - Passo 1: o cliente envia um segmento SYN especificando a porta do servidor ao qual deseja se conectar e seu número de sequência inicial
 - Passo 2: o servidor responde enviando outro segmento SYN com o ACK do segmento recebido e o seu próprio número de sequência
 - Passo 3: o cliente retorna um ACK e a conexão se estabelece



Protocolo TCP

- Estabelecimento de Conexão (Protocolo)
 - O tamanho máximo de segmento (MSS) que cada lado se propõe a aceitar também é definido no momento do estabelecimento da conexão
 - Pode acontecer um “half open”

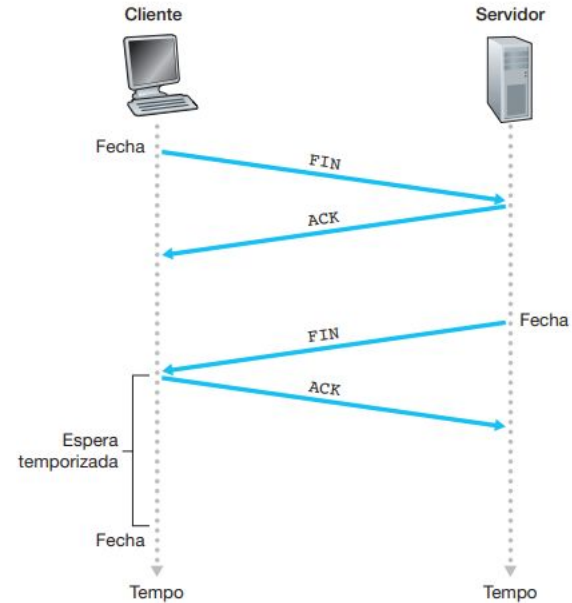
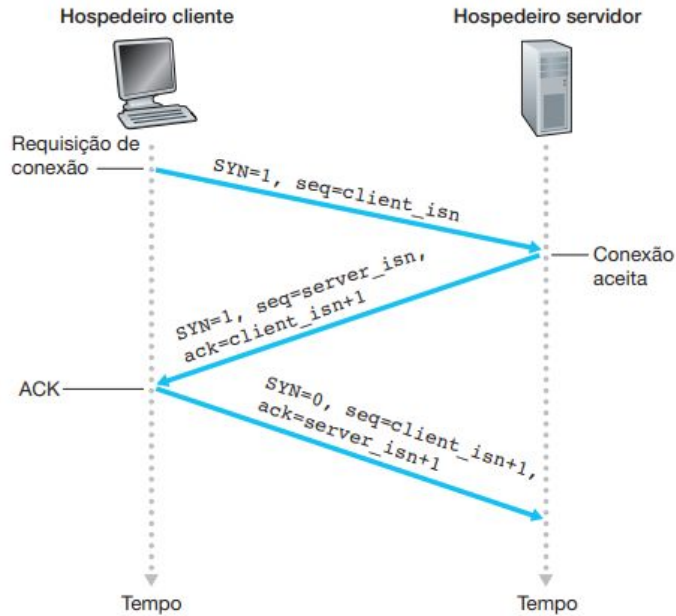


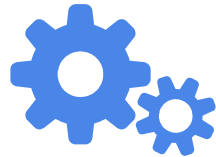
Protocolo TCP

- Término de Conexão (Protocolo)
 - Cada direção da conexão é encerrada independentemente
 - Passo 1: o cliente envia um segmento FIN
 - Passo 2: o servidor retorna um FIN e um ACK para o cliente
 - Passo 3: o cliente envia um ACK e a conexão se encerra
 - É possível efetuar um “half close”, mantendo-se apenas uma conexão simplex



Protocolo TCP





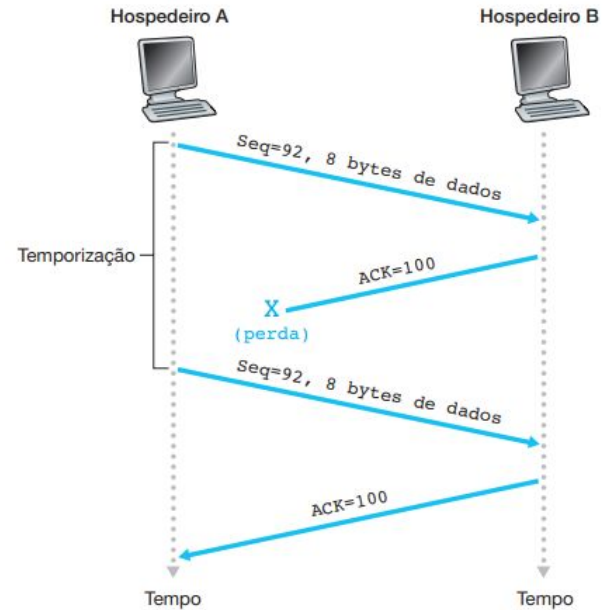
Protocolo TCP

- TCP cria serviços de transferência confiável de dados em cima do serviço não-confiável do IP
- Transmissão de vários segmentos em paralelo (Pipelined segments)
 - ACKs cumulativos
- TCP usa tempo de retransmissão simples
- Retransmissões são disparadas por:
 - Eventos de tempo de confirmação
 - ACKs duplicados



Protocolo TCP

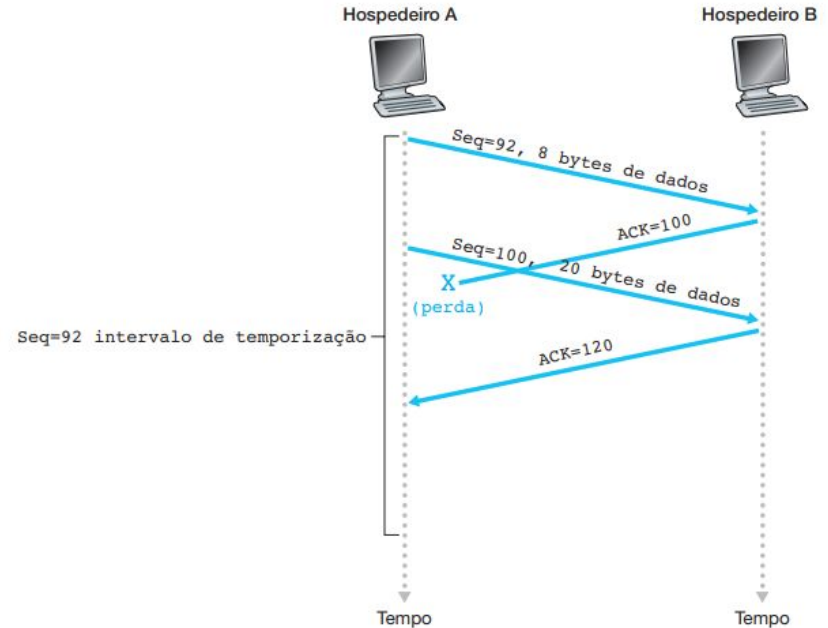
- Cenário com perda do ACK





Protocolo TCP

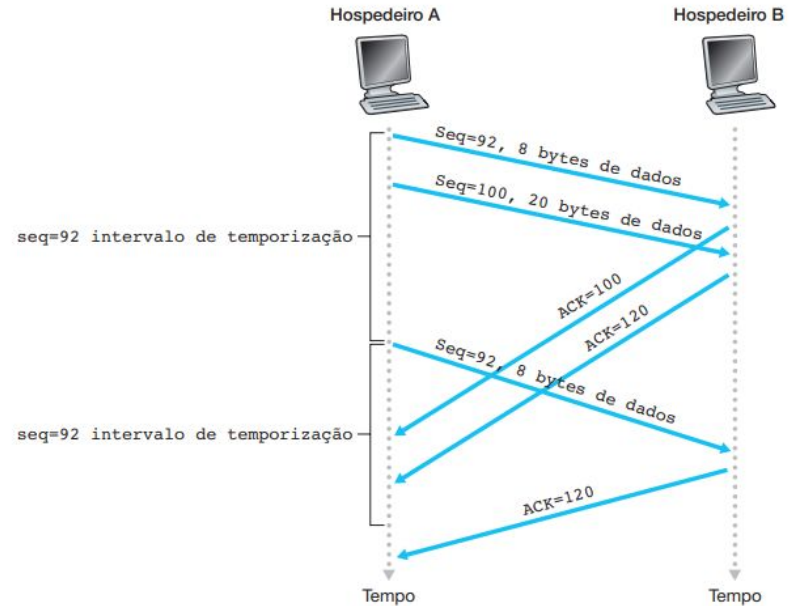
- Cenário com ACK cumulativo





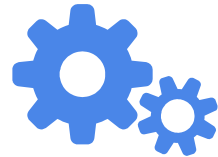
Protocolo TCP

- Cenário de temporização prematura, ACKs cumulativos



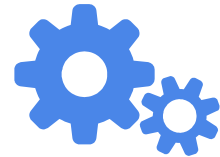
Serviços

Controle de Fluxo, Controle de Congestionamento



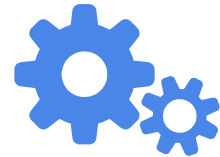
Serviços da Camada de Transporte

- Janela e Paralelismo:
- Controle de fluxo e de congestionamento
- Restrições de envio
 - De pacotes dentro de uma janela
 - Dentro de uma faixa do número de sequência



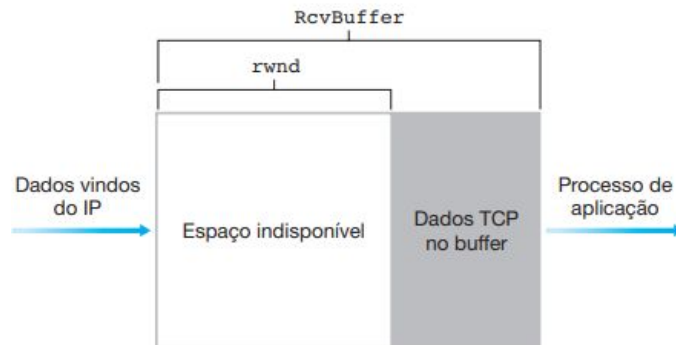
Controle de Fluxo

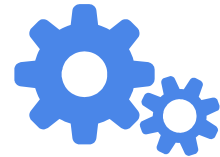
- Controle fim-a-fim dos dados entregues ao destinatário
 - Uso de buffer
- O remetente não deve esgotar os buffers de recepção enviando dados rápido demais
 - Speed-matching: busca pela taxa de envio adequada à taxa da aplicação receptora
 - Manutenção de uma variável (janela)
 - Alterada de acordo com a velocidade de consumo



Controle de Fluxo

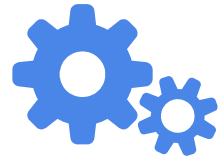
- O destinatário informa a área disponível incluindo valor RcvWindow nos segmentos
- O remetente limita os dados não confinados ao RcvWindow





Controle de Fluxo

- Pare e espere (envia um segmento por vez)
- Janela deslizante (Paralelismo)
 - Envia vários segmentos sem esperar confirmação do receptor



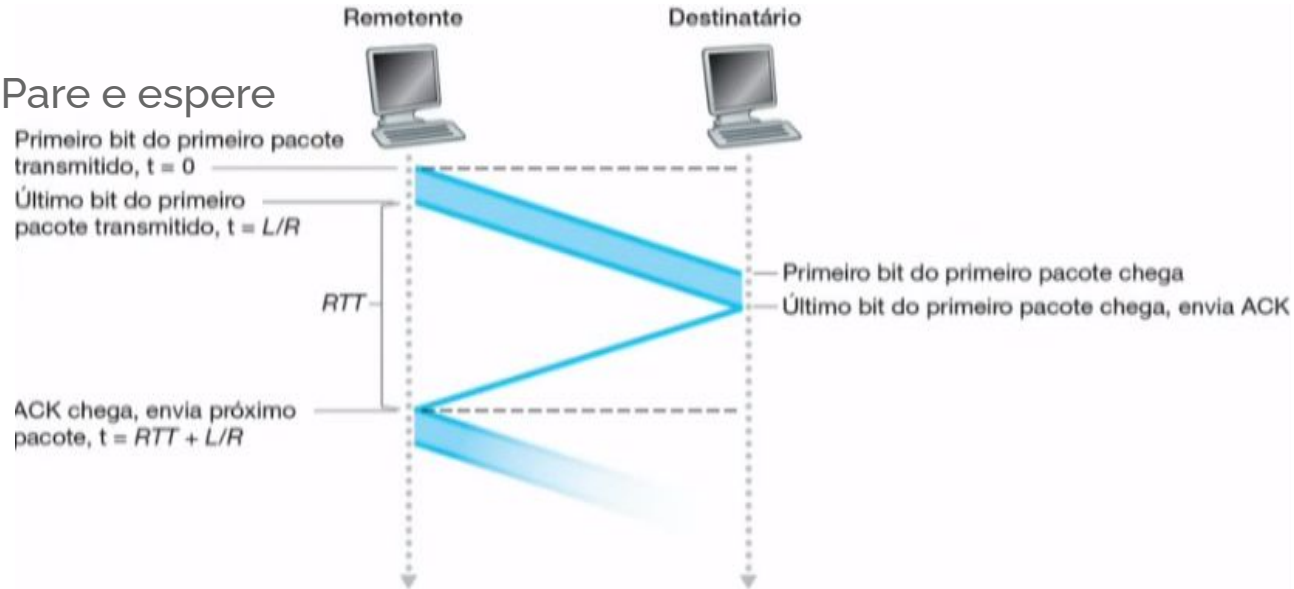
Controle de Fluxo

- Pare e espere
 - Um segmento é enviado de cada vez
 - O próximo só é enviado após seu reconhecimento
 - Com o controle de fluxo, seriam enviados mais segmentos em paralelo



Controle de Fluxo

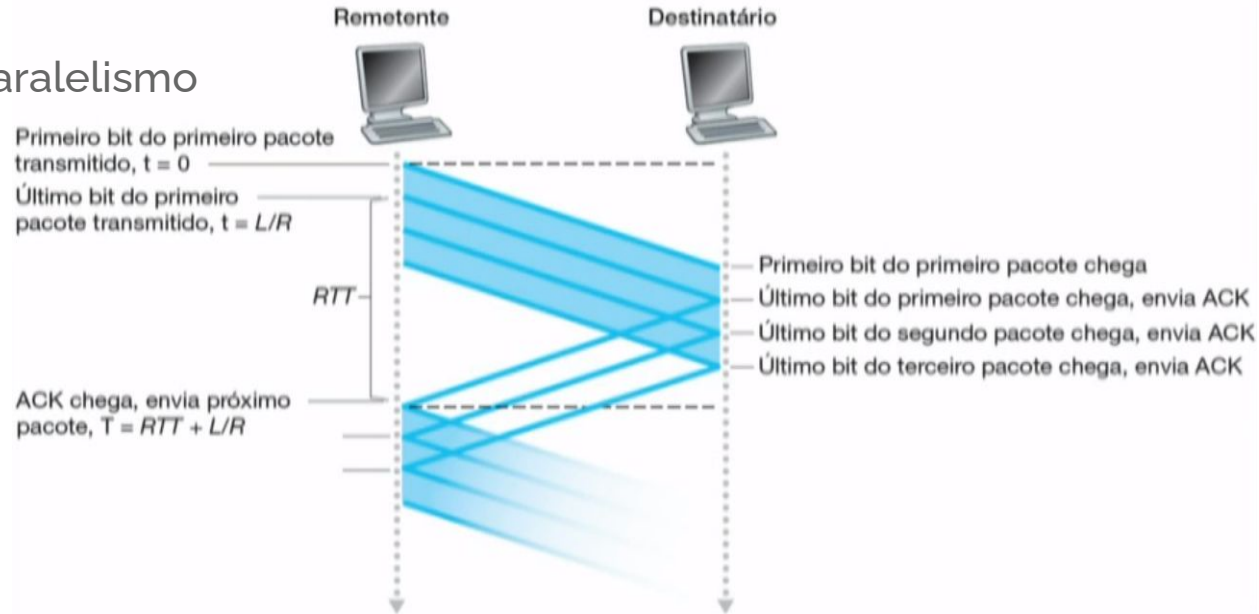
- Pare e espere

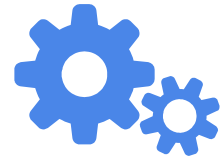




Controle de Fluxo

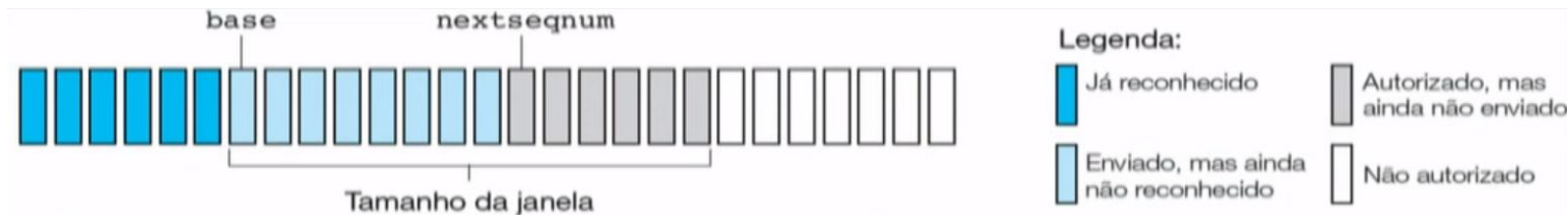
- Paralelismo





Controle de Fluxo

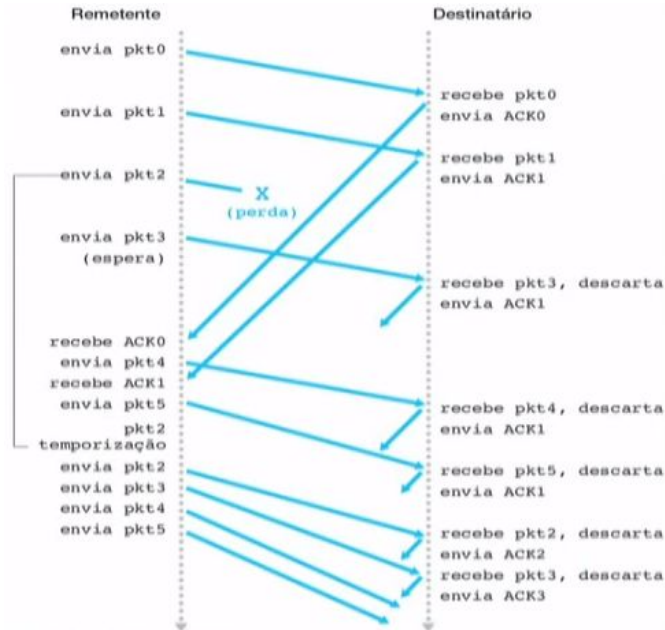
- Go-Back-N
 - O remetente envia N pacotes sem esperar por um reconhecimento
 - Limita-se a N pacotes não reconhecidos

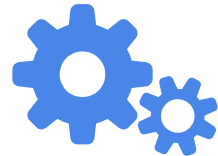




Controle de Fluxo

- Go-Back-N





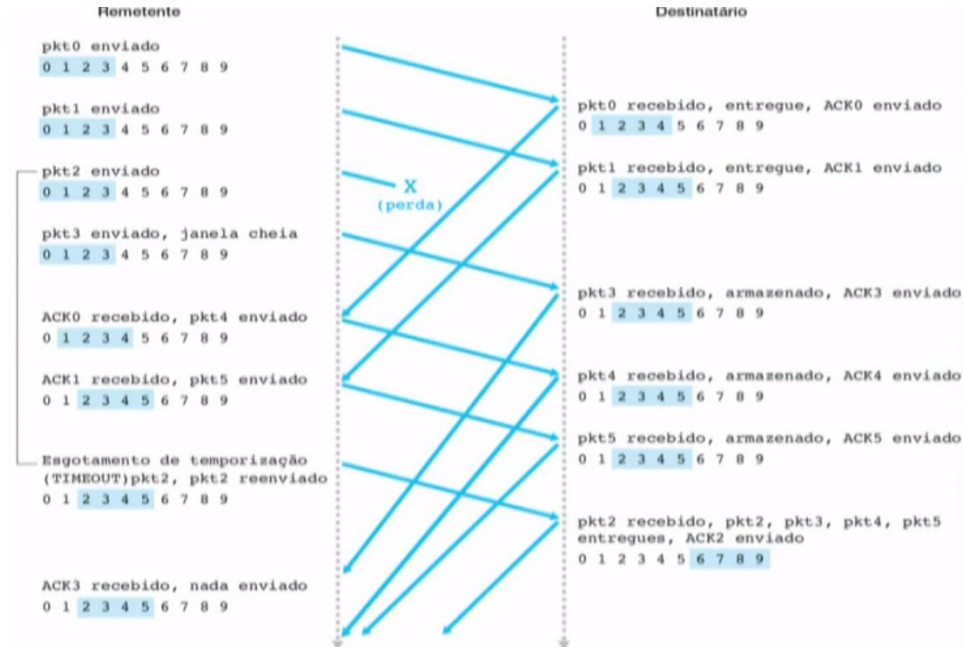
Controle de Fluxo

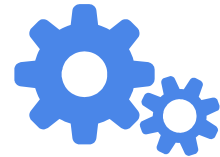
- Repetição Seletiva
 - Evita retransmissões desnecessárias
 - O remetente retransmite apenas os pacotes suspeitos de erro
 - Retransmissão e Reconhecimento dos pacotes corretos de forma individual



Controle de Fluxo

- Repetição Seletiva



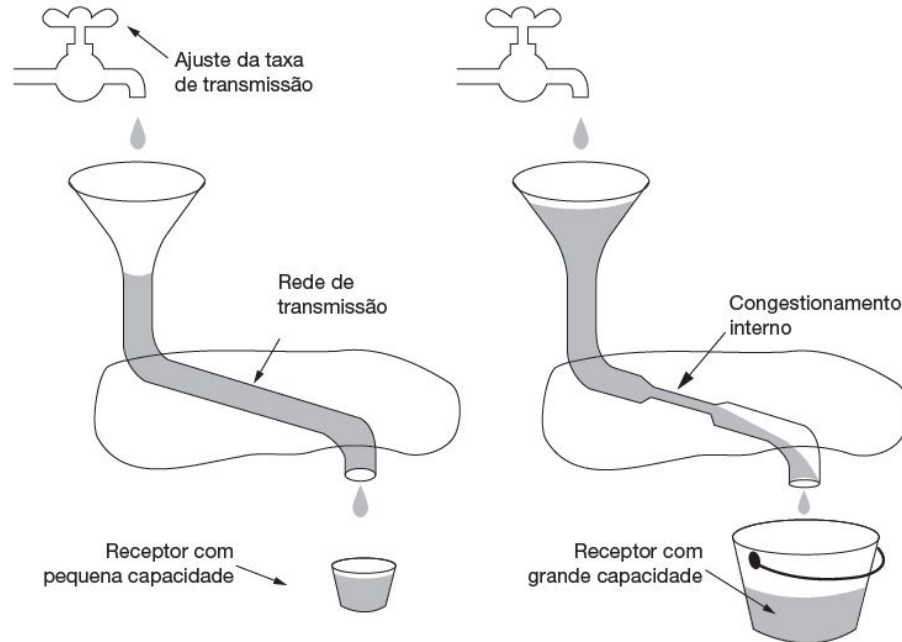


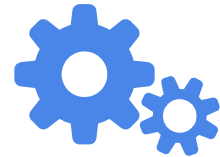
Controle de Congestionamento

- Uma conexão TCP controla sua taxa de transmissão limitando o número de segmentos que podem ser transmitidos sem confirmações
 - Esse número é chamado o tamanho da janela do TCP (w)
- O número máximo de segmentos não confirmados é dado pelo mínimo entre os tamanhos das janelas de congestionamento e do receptor
 - Ou seja, mesmo que haja mais largura de banda, o receptor também pode ser um gargalo



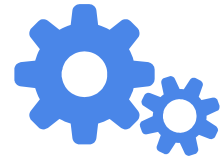
Controle de Congestionamento





Controle de Congestionamento

- O controle é feito através de duas variáveis adicionadas em cada lado da conexão:
 - Janela de Congestionamento (Janela do TCP)
 - Limiar (controla o crescimento da janela de congestionamento)



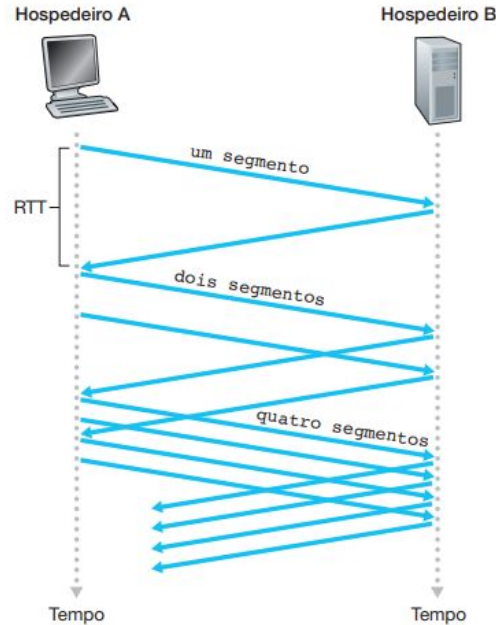
Controle de Congestionamento

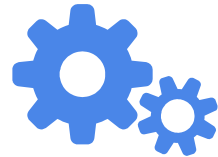
- No início, a janela de congestionamento tem o tamanho de um segmento
 - Tal segmento tem o tamanho do maior segmento suportado
- O primeiro segmento é enviado e então é esperado seu reconhecimento
 - Se o mesmo chegar antes do timeout, o transmissor duplica o tamanho da janela de congestionamento e envia dois segmentos
 - Se esses dois segmentos também forem reconhecidos antes de seus timeouts, o transmissor duplica novamente sua janela, enviando agora quatro segmentos



Controle de Congestionamento

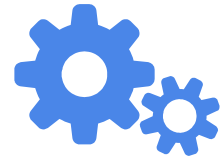
- Partida lenta





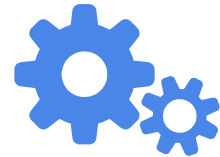
Controle de Congestionamento

- Uma conexão TCP começa com um pequeno valor de w e então o incrementa arriscando que exista mais largura de banda disponível
- Isso continua a ocorrer até que algum segmento seja perdido
- Nesse momento, a conexão TCP reduz w para um valor seguro, e então continua a arriscar o crescimento



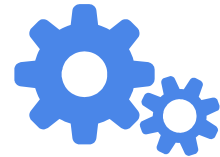
Controle de Congestionamento

- Esse processo continua até que:
 - O tamanho da janela de congestionamento seja maior que o limiar, ou maior que o tamanho da janela do receptor ou
 - Ocorra algum timeout antes da confirmação



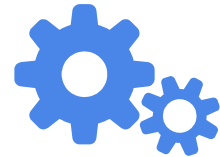
Controle de Congestionamento

- A primeira fase, em que a janela de congestionamento cresce exponencialmente é chamada de inicialização lenta (slow start), pelo fato de começar com um segmento
 - A taxa de transmissão começa pequena porém cresce rapidamente



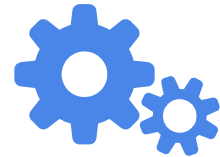
Controle de Congestionamento

- Uma vez ultrapassado o limiar, e a janela do receptor ainda não seja um limitante, o crescimento da janela passa a ser linear
 - A segunda fase é chamada de prevenção de congestionamento (congestion avoidance)
 - Sua duração também depende da não ocorrência timeouts, e da aceitação do fluxo por parte do receptor



Controle de Congestionamento

- Na ocorrência de um timeout o TCP irá configurar o valor do limiar como a metade do tamanho atual da janela de congestionamento
 - O tamanho da janela de congestionamento volta ser do tamanho de um segmento
 - O tamanho da janela de congestionamento volta a crescer exponencialmente

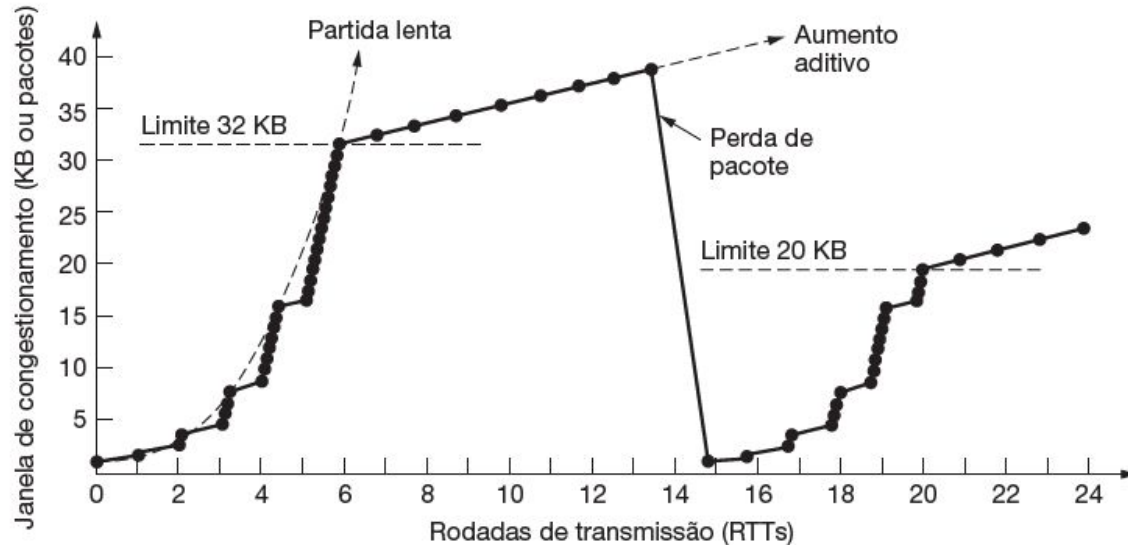


Controle de Congestionamento

- Caso ocorram 3 ACKs duplicados: O valor do limiar é ajustado para metade tamanho atual da janela de congestionamento
 - O tamanho da janela de congestionamento passa igual ao valor do limiar (metade da janela de congestionamento atual)
 - O tamanho da janela de congestionamento cresce linearmente

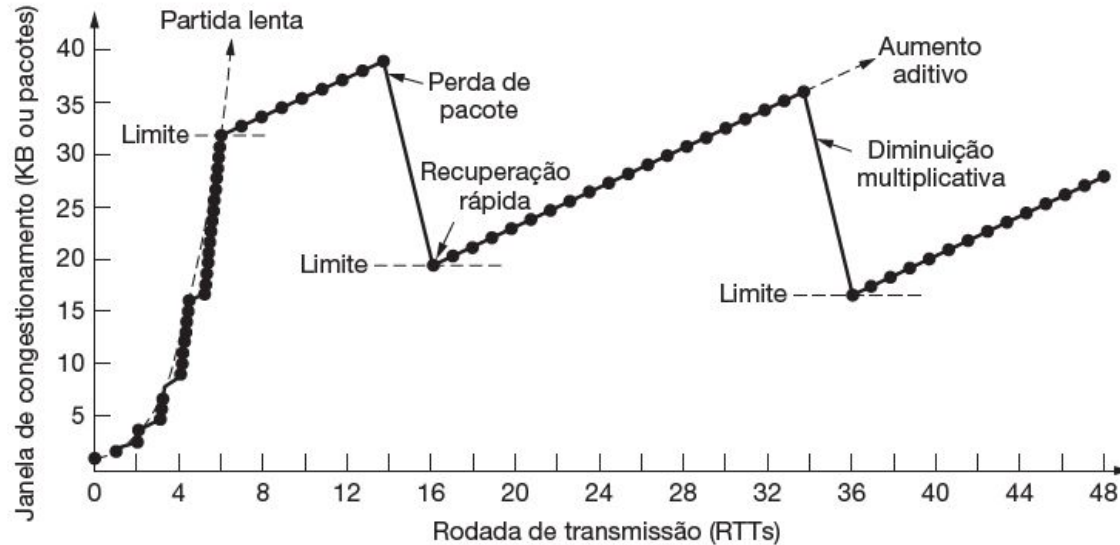


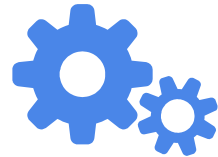
Controle de Congestionamento





Controle de Congestionamento





Controle de Congestionamento

- Em resumo, quando o tamanho da janela de congestionamento está abaixo do limiar, seu crescimento é exponencial; quando este tamanho está acima do limiar, o crescimento é linear
- Todas as vezes que ocorrer um timeout, o limiar é modificado para a metade do tamanho da janela e o tamanho da janela passa a ser 1
 - Congestionamento pesado
- Quando ocorrem ACKs repetidos a janela cai pela metade
 - Congestionamento leve



Obrigado!

Dúvidas?