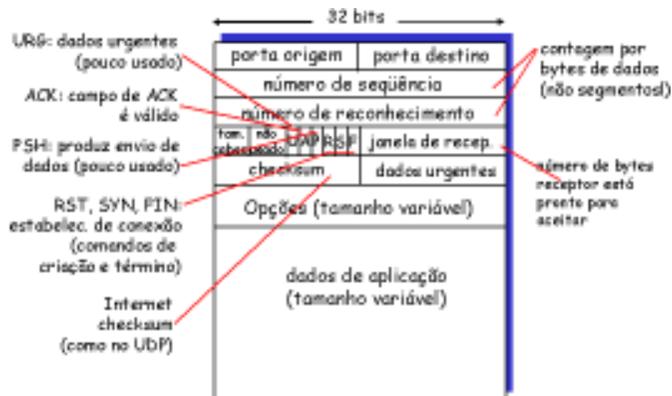


- Conexão TCP
 - Fornece transferência de dados **full-duplex**.
 - Ponto-a-ponto, um único remetente e um único destinatário.
 - Multicast não é possível com TCP.
- Estabelecimento de conexão:
 - Processo cliente informa ao TCP cliente que quer estabelecer conexão com o processo servidor.
 - Primeiramente é enviado pelo cliente um segmento TCP especial ao servidor;
 - O servidor responde com um segundo segmento TCP especial ao cliente;
 - Por fim, o cliente responde com um terceiro segmento TCP especial ao servidor;
 - Como três segmentos são enviados entre dois hosts, esse modo de estabelecimento de conexão é chamado 3-way handshake (apresenta de três vias)



- Estrutura do Segmento TCP



- Assim como o UDP, TCP fornece:
 - Multiplexação e demultiplexação
 - Detecção de erros
- TCP difere do UDP em vários aspectos:
 - Maior diferença:
 - TCP é orientado à conexão enquanto o UDP não.
 - 3-way handshake (apresentação em 3 vias)

• Números de reconhecimento

- TCP é full-duplex. A pode estar recebendo dados de B, enquanto A envia dados para B.
- Cada segmento que chega do host B tem um número de sequência para os dados que estão fluindo de B para A.
- O número de reconhecimento que o host A atribui a seu segmento é o número de sequência do próximo byte que ele estiver aguardando do host B.

• Estrutura do segmento

- Campos:
 - Número de porta (origem e destino) (16 bits cada): multiplexação e demultiplexação de dados.
 - Número da sequência e número do reconhecimento (32 bits cada): serviço confiável e transferência de dados.
 - Tamanho da janela do receptor (16 bits): controle de fluxo Comprimento do cabeçalho (4 bits): comprimento do cab. TCP em palavras de 32 bits. Tamanho típico = 20 bytes.
 - Opções (comprimento variável): negociação do MSS (tamanho máximo do segmento) ou fator de escala da janela em redes de alta velocidade.
 - Flag (6 bits): ACK (se o reconhecimento é válido), RST, SYN e FIN para estabelecer e derrubar uma conexão, PSH e URG para entrega urgente à aplicação. Em geral PSH e URG não são usados.

• Números de sequência e ACK: Exemplo 1

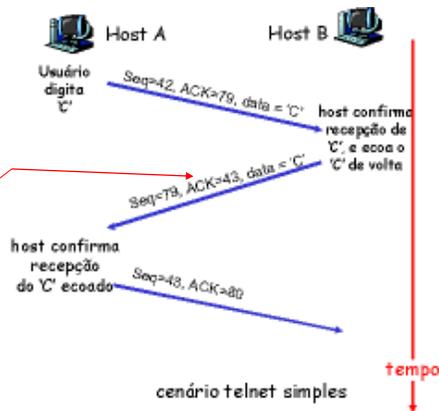
Números de seqüência:

- número do primeiro byte no segmentos de dados

ACKs:

- número do próximo byte esperado do outro lado.

- Piggyback: Quando o reconhecimento pega carona em um pacote de dados.



cenário telnet simples

• Números de sequência

- O número da sequência para um segmento TCP é o número do primeiro byte do segmento.
- Suponha que um processo no host A quer enviar uma cadeia de dados para um processo no host B por meio de uma conexão TCP.
- Suponha que a cadeia de bytes consista em um arquivo de 500.000 bytes, e que o MSS seja de 1000 bytes e a numeração atribuída ao primeiro byte seja 0 (esse número é escolhido aleatoriamente).
- Assim o TCP constrói 500 segmentos a partir da cadeia de dados.
- O primeiro segmento ganha o número de sequência 0, o segundo segmento ganha o número de sequência 1000; o terceiro segmento o número de sequência 2000, e assim por diante.



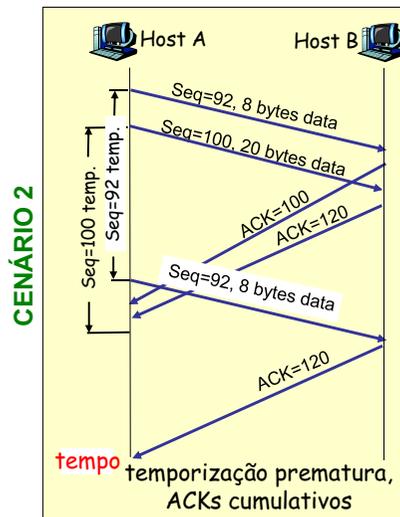
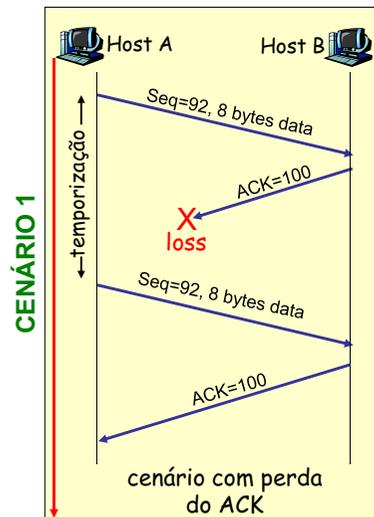
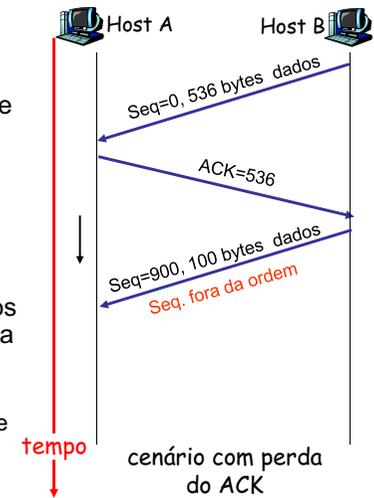
[RFC 1122, 2581]

- Recomendações para geração de ACK pelo TCP

Event0	Ação do TCP Receptor
segmento chega em ordem, não há lacunas, segmentos anteriores já aceitos	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: lacuna detectada	envia ACK duplicado, indicando número de seqüência do próximo byte esperado
chegada de segmento que parcial ou completamente preenche a lacuna	reconhece imediatamente se o segmento começa na extremidade mais baixa da lacuna

Exemplo 2:

- (1) o destinatário descarta de imediato os bytes fora da ordem.
 - Mais simples na implementação do código TCP.
- (2) o destinatário conserva os bytes fora da ordem e espera pelos bytes faltantes para preencher a lacuna.
 - Mais eficiente em termos de largura de banda da rede.



Transferência confiável de dados

- TCP cria um serviço de transferência confiável de dados sobre o serviço de melhor esforço do IP.

Garantindo:

- Que a cadeia de dados que um processo lê a partir de seu buffer de recebimento TCP
 - não está corrompida;
 - não tem lacunas;
 - não possui duplicações e está em sequência.
- Isto é, a cadeia de bytes é exatamente a mesma cadeia de dados enviada pelo sistema final que está do outro lado da conexão.

- Controle de fluxo
 - Janela de recepção (contexto transmissão de arquivo)
 - Host A transmite um grande arquivo para o host B
 - B aloca um buffer de recepção a essa conexão TCP
 - **RcvBuffer** = tamanho do buffer de recepção
 - De tempos em tempos o processo da aplicação em B faz a leitura do buffer. São definidas as seguintes variáveis:
 - **LastByteRead** = o número de último byte na cadeia de dados lido do buffer pelo processo de aplicação em B.
 - **LastByteRcvd** = o número do último byte na cadeia de dados que chegou da rede e foi colocado no buffer de recepção de B.

- Controle de fluxo
 - Hosts de cada lado da conexão reservam um buffer de recepção para a conexão.
 - Bytes corretos recebidos e em sequência são armazenados no buffer de recepção pelo TCP.
 - O processo de aplicação associado vai ler os dados a partir desse buffer, mas não necessariamente no momento em que são recebidos.
 - O buffer pode ser saturado rapidamente por uma leitura demasiadamente lenta por parte do processo de aplicação ou pelo envio rápido de dados por parte do remetente.
 - Para isso, o TCP fornece um **serviço de controle de fluxo** para suas aplicações, a fim de eliminar a possibilidade de o remetente saturar o buffer do Receptor.

- Controle de fluxo
 - Janela de recepção (cont.)
 - Temos a relação:

$$\text{LastByteRcvd} - \text{LastByteRead} \leq \text{RcvBuffer}$$
 - A janela de recepção, denominada **RcvWindow**, é ajustada à quantidade de espaço disponível no buffer.

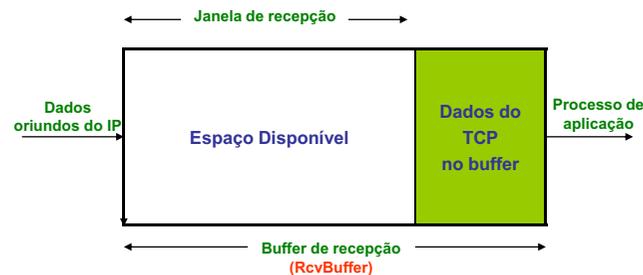
$$\text{RcvWindow} = \text{RcvBuffer} - [\text{LastByteRcvd} - \text{LastByteRead}]$$
 - Como o espaço disponível muda com o tempo, RcvWindow é dinâmica.

- Controle de fluxo
 - O TCP fornece o serviço de controle de fluxo fazendo com que o remetente mantenha uma variável chamada janela de recepção.
 - A janela de recepção é usada para dar ao remetente uma idéia de quanto espaço livre de buffer está disponível no destinatário.
 - Em uma conexão full-duplex, o remetente de cada lado da conexão mantém uma janela de recepção distinta.
 - A janela de recepção é dinâmica (muda durante o tempo de vida de uma conexão).

- Controle de fluxo (problema, cont.)
 - Para resolver este problema, a especificação do TCP requer que o host A continue a enviar segmentos com um byte quando a janela de recepção de B for zero.
 - Estes segmentos serão reconhecidos pelo host B.
 - Inevitavelmente o buffer de B começará a esvaziar e os reconhecimentos vão conter um valor diferente de zero para RcvWindow.

- Controle de fluxo
 - Janela de recepção (cont.)
 - B envia o valor corrente de RcvWindow no campo “janela de recepção” de cada segmento para A.
 - Inicialmente B estabelece $RcvWindow = RcvBuffer$.
 - O host A mantém duas variáveis, LastByteSent e LastByteAced.
 - $LastByteSent - LastByteAced =$ Qtde da dados não reconhecidos que A enviou para B.
 - Se $LastByteSent - LastByteAced \leq RcvWindow$
 - Garante-se que A não vai transbordar o buffer de B

- Controle de fluxo

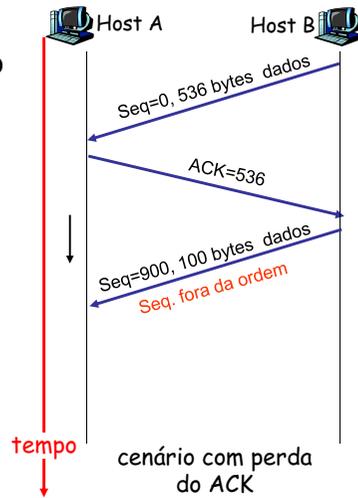


- Controle de fluxo (problema)
 - Suponha que B tenha $RcvWindow = 0$, e anuncia isto para A.
 - Suponha que B não tenha nada para enviar a A.
 - Enquanto o processo de aplicação em B esvazia o buffer, o TCP não envia novos segmentos com novos RcvWindows para A.
 - O TCP só enviará um segmento para A se tiver dados de reconhecimento para enviar.
 - Por conseguinte, o host A nunca é informado de que foi aberto espaço no buffer de recepção do host B.

Exercícios



- 4 - Analisando o cenário ao lado. Quais medidas podem ser tomadas a segmentos recebidos que não tem o nº de sequência esperada?
- Quais as vantagens de cada medida?



23

Exercícios:



- 1 - Quais as características da conexão TCP e como se dá uma conexão TCP entre dois sistemas finais?
- 2 – Quais as funções dos campos “número de sequência” e “número de reconhecimento” em um dado segmento TCP?

21

Exercícios



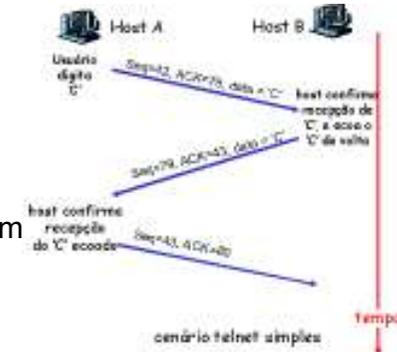
- 5 - Quais as garantidas fornecidas pelo TCP a uma cadeia de dados que um processo lê a partir de seu buffer de recebimento TCP?
- 6 - Cite os eventos e as ações tomadas pelo receptor documentadas pelas **RFCs 1122, 2581**.

24

Exercícios



- 3 - Descreva a utilização dos campos nº sequência e nº reconhecimento no cenário ao lado.
- O que é piggyback? Em que situação ocorre piggyback no cenário ao lado?



22

Exercícios



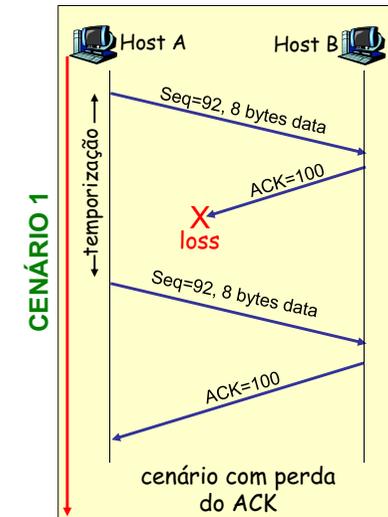
- 9 - Como é dado o controle de fluxo no TCP?
- 10 - Dada uma transmissão entre dois sistemas finais A e B. Como o TCP se comporta quando B informa a A que sua janela de transmissão é igual a 0 e B não possui mais nenhum segmento para transmitir a A?

27

Exercícios



- 7 - Comente o cenário ao lado.

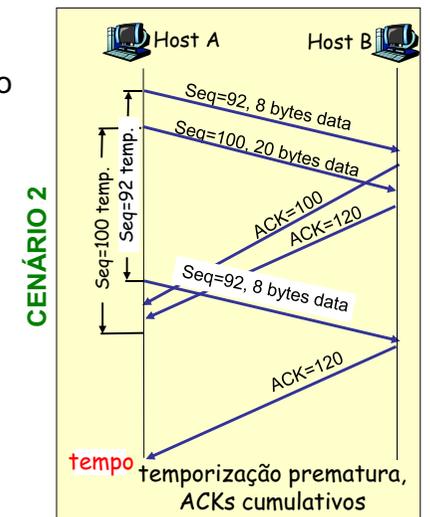


25

Exercícios



- 8 - Comente o cenário ao lado



26