

Redes de Computadores

Nível de Transporte (continuação)

Aula 23

Introdução

- Entidades da camada de transporte disponibilizam serviços para a camada de nível superior
 - No modelo de referência MR-OSI é para a camada de sessão, na arquitetura TCP/IP é para a camada de aplicação
- Modelos de serviços
 - Não orientados a conexão
 - Orientados a conexão
 - Garantia de entrega, de ordem, não duplicação e sem erros
- Conceito fundamental: conexão

Serviço orientado a conexão

- Conexão é composta por três fases
 - Estabelecimento
 - Transferência
 - Encerramento
- Provê um serviço confiável
 - Garante a entrega de um fluxo de dados (*stream*)
 - Provê controle de fluxo e controle de erro
- Conexão é bidirecional
- Modelado por uma máquina de estado finito onde as T-PDUs de controle são os eventos que fazem passar de um estado a outro

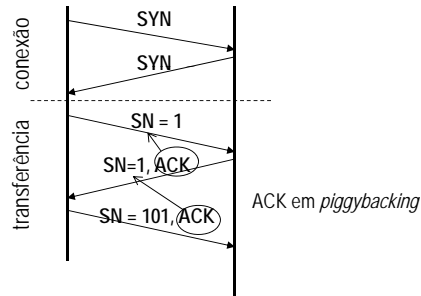
Conexão: estabelecimento

- Estabelecimento da conexão
 - Garante que entidade de transporte correspondente existe
 - Negocia parâmetros (ex.: Tamanho T-PDUs, capacidade de recepção, número de sequência, QoS etc)
 - Reserva recursos nas entidades de transportes envolvidas (ex.: buffers, identificadores de conexão)
- Pode ser de dois tipos
 - Assimétrica (modelo cliente-servidor)
 - Uma entidade (cliente) solicita o estabelecimento da conexão (ativo)
 - Uma entidade (servidor) espera por pedidos de estabelecimento de conexão (passivo)
 - Simétrica
 - As entidades podem iniciar ou receber pedidos de conexão

Estabelecimento de conexões: cenário

T-PDU SYN funciona como solicitação de abertura de conexão e como confirmação de um pedido de conexão

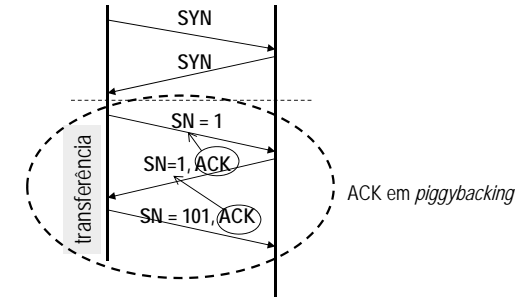
Estados
active open
passive open (Listen)
waiting



ATENÇÃO: não considera erros, nem atrasos! (a completar)

Fase de transferência de dados

- Corresponde ao estado "estabelecido" (*established*)
- Fase que realiza o controle de fluxo e o controle de erros
 - Baseado no envio e recebimentos de T-PDUs com números de sequência
 - Confirmações positivas e negativas



Encerramento de uma conexão

- Objetivo é liberar os recursos alocados pelo estabelecimento da conexão
- Qualquer lado pode pedir o encerramento
- Duas formas:
 - Simétrico: entidade ao encerrar a conexão indica apenas que não tem mais dados a enviar mas ainda está aceitando dados do outro
 - Conexão só é encerrada se ambos os lados concordarem
 - Não há perda de dados
 - Encerramento gracioso (*graceful*)
 - Assimétrico: entidade encerra a conexão independentemente do outro lado
 - Pode haver perdas de dados
- Estados *wait closed*, *closed*

Estratégias para encerramento de conexão

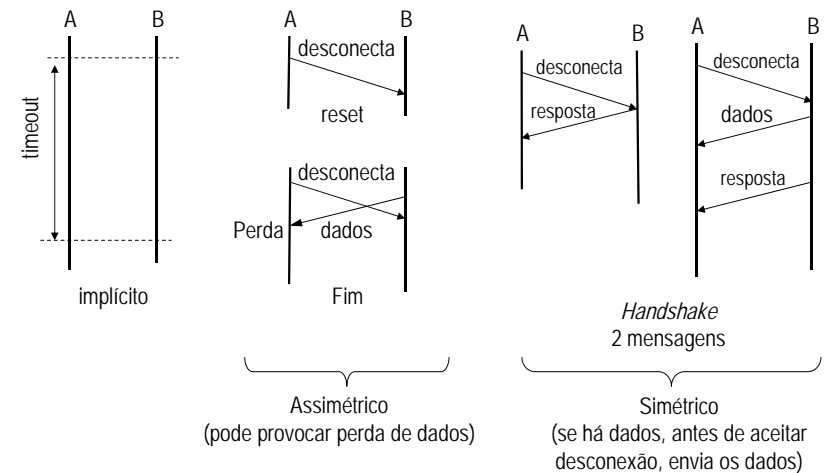
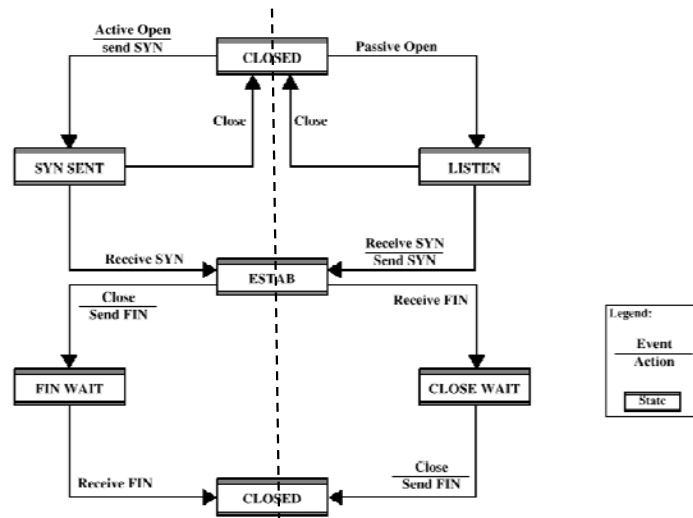


Diagrama de estados de conexão



Redes de Computadores

9

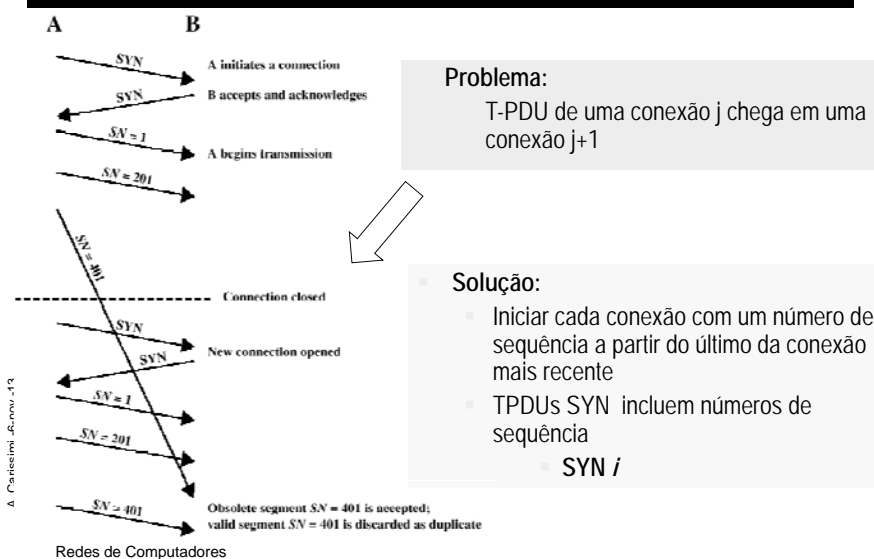
Conexão na presença de erros: problemas

- QUESTÃO: como se define o número de sequência inicial ?
 - A partir de um valor fixo predefinido (e.g. SN=1)
 - Negociando um valor para o SN inicial da conexão
- Problemas:
 - Estabelecimento da conexão: perda ou atraso de T-PDUs SYN
 - Transferência de dados:
 - T-PDUs de dados de conexões anteriores chegando atrasadas
 - Encerramento da conexão
 - Encerramento prematuro (perda de dados) por atraso na T-PDU de dados

Redes de Computadores

10

Problema: T-PDU de dados de conexões anteriores

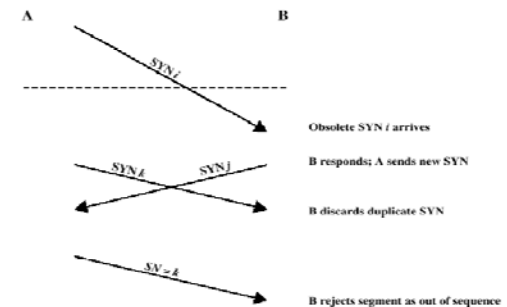


Redes de Computadores

11

Mais problemas: atraso na T-PDU de controle

- T-PDU que chega atrasado é um T-PDU SYN i



B pensa que está abrindo conexão que espera números de sequência a partir de i e responde a partir de j . A considera que a conexão é com k e j . Ao chegar T-PDU com k , B descarta como erro por não pertencer a conexão negociada!!

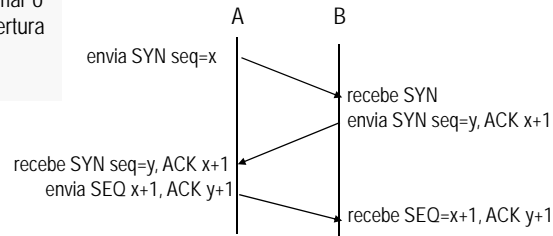
Redes de Computadores

12

Three-way handshake

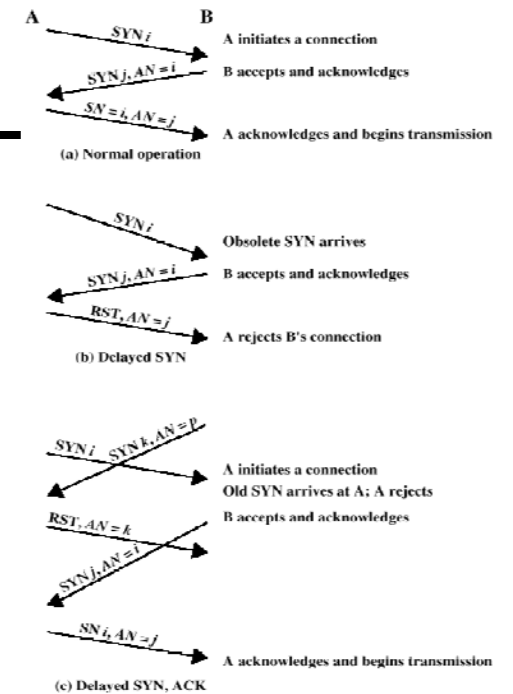
- Solução para problema do *two-way handshake*
 - Formalmente provado que a troca de 3 mensagens é condição necessária e suficiente para assegurar um acordo não ambíguo entre duas partes
 - Mesmo considerando perdas, duplicação e atrasos

Consiste em cada lado confirmar o recebimento da T-PDU de abertura de conexão e seu número de sequência.



13

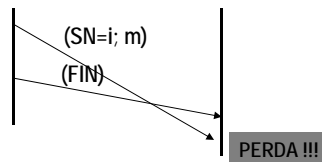
Three Way Handshake: Exemplos



Redes de Computadores

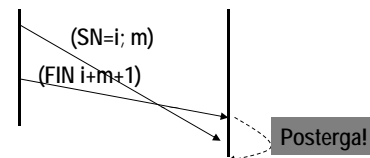
Problema: encerramento de conexão

- Baseado em duas T-PDUs FIN (*two way handshake*)
- Apresenta problemas similares ao estabelecimento da conexão
 - Perdas de T-PDUs e T-PDUs atrasadas



- Solução: numerar T-PDUs FIN

Espera dados chegarem antes de encerrar a conexão.

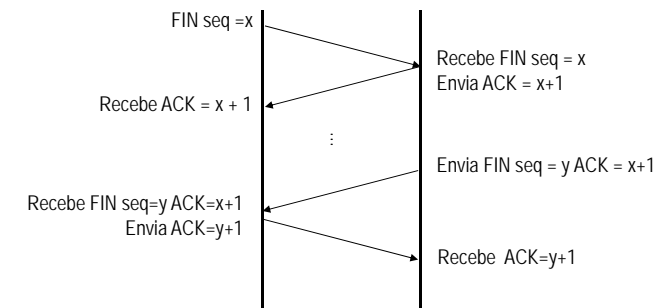


15

Four way handshake: encerramento de conexão simétrico

- Antes de receber a confirmação do encerramento (T-PDU FIN da outra entidade) ainda recebe T-PDUs de dados

Four-way handshake



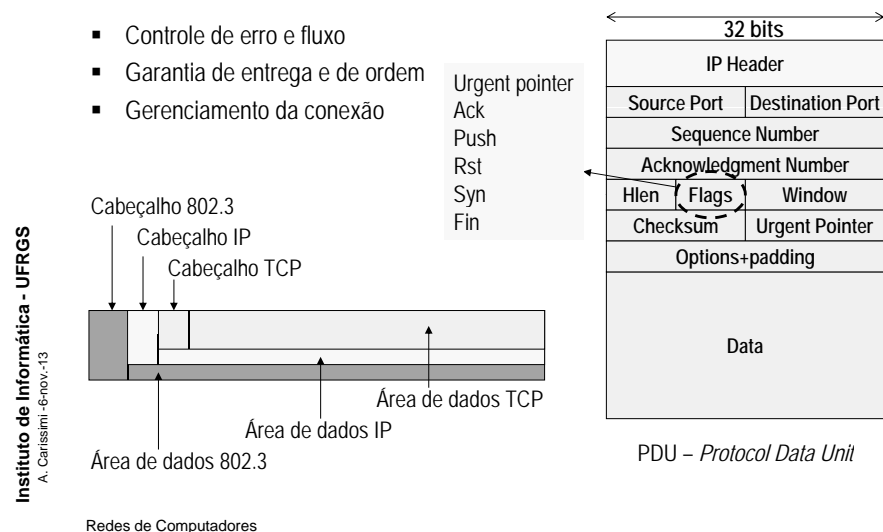
Redes de Computadores

16

Estudo de caso: *Transmission Control Protocol* (TCP)

- Protocolo orientado a conexão
- Descrito nas RFCs 793, 1122, 1323, 2018 e 2581
- PDU do TCP é denominada de segmento
 - Mensagens geradas por processos são fragmentadas em segmentos TCP
- Abordagem baseada em fluxo de dados (*data stream*)
 - Trata os dados como uma cadeia contínua de bytes
 - Decide como agrupar os bytes em segmentos
- Fornece comunicação confiável ponto a ponto entre dois processos
 - Cada conexão TCP envolve exatamente duas extremidades
 - Envio pode ocorrer entre uma ou outra direção de uma mesma conexão TCP

Segmento TCP (relembrando...)



Os campos mais simples: *hlen*, *flags*...

- *Hlen*
 - Fornece o tamanho do cabeçalho TCP em múltiplos de 4 bytes (palavra)
 - Tipicamente assume o valor 5 (20 bytes), exceto quando há *options*
- *Flags*
 - Indica o propósito e o tipo do segmento
 - URG: segmento transporta dados urgentes
 - ACK: segmento transporta uma confirmação positiva
 - PSH: mecanismo *push* foi adotado no envio do segmento
 - RST: a conexão deve ser imediatamente abortada (*reset*)
 - SYN: segmento transporta requisição de abertura de conexão
 - FIN: segmento transporta requisição de encerramento da conexão

...*checksum*, *options* e *padding*

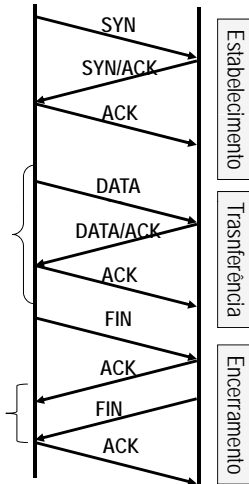
- *Checksum*
 - Verificação da integridade dos dados (complemento de 1 em 16 bits)
 - Calculado sobre um pseudo-cabeçalho (IP destino, IP fonte, campo protocolo, tamanho da T-PDU e a constante zero), o cabeçalho TCP e os dados
- *Options*
 - Lista variável de informações e parâmetros para o TCP
 - Usado principalmente para definir o tamanho máximo de um segmento (MSS)
- *Padding*
 - Usado para "arredondar" o número de bytes do cabeçalho para este ter sempre um número inteiro de palavras de 32 bits.

Conexão TCP

OBS: no desenho, foram omitidos os número de sequência

Campos *sequence number*, *acknowledgement number*, *window* e *urgent pointer* são usados nos mecanismos de controle de erro e de fluxo (ordenamento).

Possível fazer
FIN+ACK (*piggybacking*)



21

Campo *window*

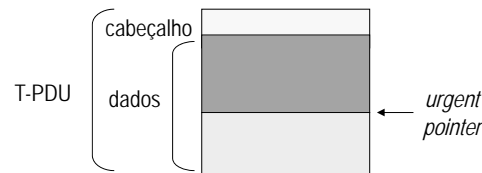
- Define, no estabelecimento da conexão, a capacidade do buffer de recepção (tamanho da janela, i.e., quantidade de créditos)
- Controle de fluxo faz o TCP ajustar as janelas de transmissão e recepção
 - Janela de recepção
 - "fecha" a medida que bytes são recebidos e "abre" quando os dados são lidos pela aplicação (modelo PULL)
 - Janela de transmissão
 - "abre" de acordo com as confirmações do receptor
- O tamanho da janela pode ser renegociado durante a fase de transferência de dados

Redes de Computadores

22

Transferência de dados: *urgent data*

- Flag URG** indica a presença de dados "urgentes" no segmento
 - Usuário (processo) destino decide o que fazer
 - O campo *urgent pointer* sinaliza a posição em que termina os dados urgentes



- Possibilidade raramente usada no TCP
 - Necessário que o processo destino priorize os dados (não seria responsabilidade da aplicação?)

23

Transferência de dados: mecanismo *data stream push* (bit *PSH*)

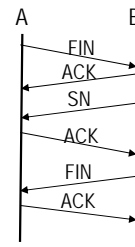
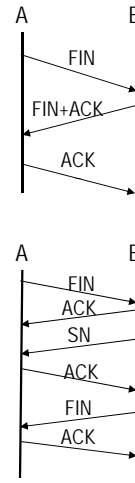
- Buffer de transmissão do TCP tenta otimizar uso da rede
 - Objetivo é diminuir a quantidade de segmentos na rede, bufferizando os dados e enviando quando eles atinjam uma determinada quantidade
 - Problema: prejudicar tempo de resposta da aplicação (ex. telnet)
- Bit PSH
 - Processo origem pode solicitar a transmissão de dados até esse momento
 - Força a geração de um segmento com os dados disponíveis
 - Indica que a entidade de transporte deve repassar imediatamente o segmento para o processo destino
 - Encaminha todos os bytes que estão no buffer e não apenas os do segmento identificado com o PSH

Redes de Computadores

24

Encerramento de conexões TCP

- Encerramento completo
 - Estratégia three-way handshake
 - Sequência FIN; FIN+ACK; FIN
- Semifechamento
 - Estratégia four-way handshake
 - Sequência FIN; ACK; FIN; ACK
 - A entidade que iniciou o encerramento, fecha a conexão para o envio de novos dados, mas aceita dados que porventura ainda estejam em trânsito

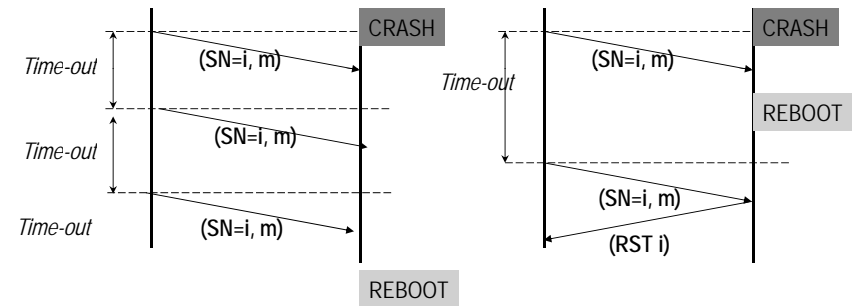


Redes de Computadores

25

Recuperação de falhas (*crash recovery*)

- Em caso de falhas as informações de estado de uma conexão são perdidas por um lado, o outro mantém meia-conexão



Reseta a meia conexão por excesso de retransmissões

Reseta a meia conexão por receber comando de reset (RST)

Redes de Computadores

26

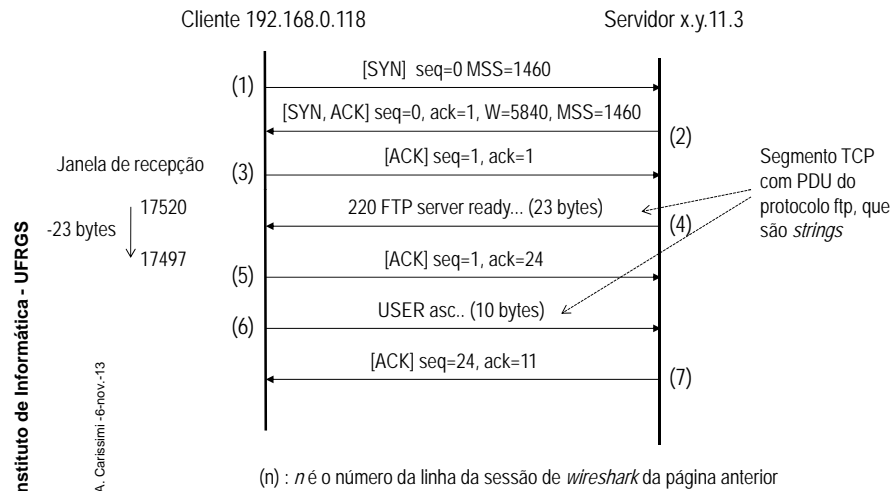
Wireshark packet capture showing an HTTP connection. The packet list shows a GET request and a 200 OK response. The packet details pane shows the HTTP request and response structure. The packet bytes pane shows the raw data.

Wireshark packet capture showing an FTP connection. The packet list shows a login attempt and a 530 Login incorrect response. The packet details pane shows the FTP request and response structure. The packet bytes pane shows the raw data.

Redes de Computadores

28

Análise da sessão ftp



Leituras complementares

- Stallings, W. *Data and Computer Communications* (6th edition), Prentice Hall 1999.
 - Capítulo 17, seção 17.2
- Tanenbaum, A. *Redes de Computadores* (4^a edição), Campus, 2000.
 - Capítulo 6, seções 6.1., 6.2, 6.4 e 6.5 (até 6.5.8)
- Carissimi, A.; Rochol, J; Granville, L.Z; *Redes de Computadores*. Série Livros Didáticos. Bookman 2009.
 - Capítulo 6, seções 6.2, 6.3, 6.4, 6.5 e 6.6