

Camada de transporte

Pág. 85 da Apostila



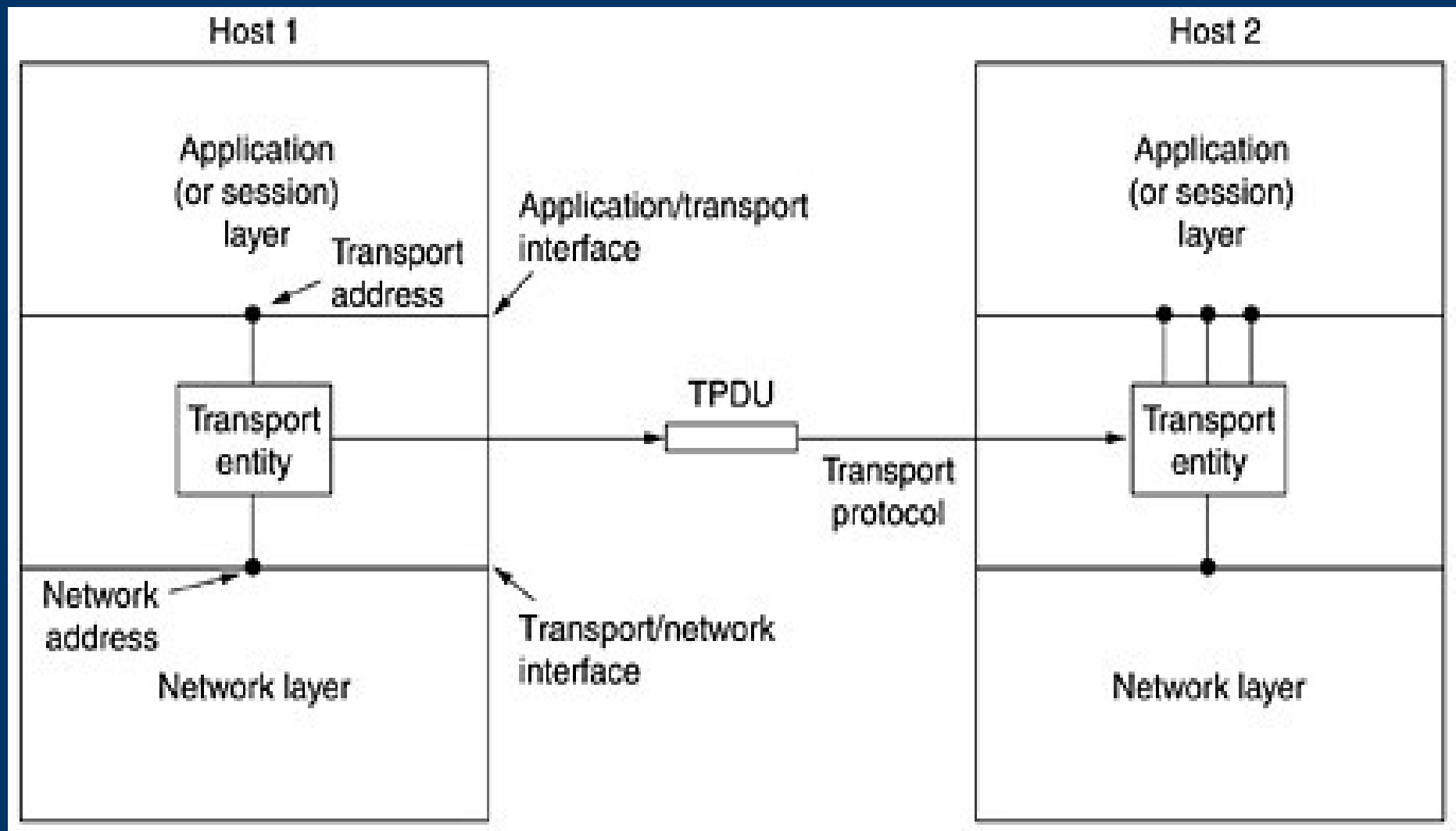
Funções da camada de transporte

- Responsável pela movimentação de dados, de forma eficiente e confiável, entre processos em execução nos equipamentos conectados a uma rede de computadores, independentemente das redes físicas
 - Deve poder regular o fluxo de dados e garantir confiabilidade, assegurando que os dados cheguem em seu destino sem erros e na sequência correta
-
-

Funções da camada de transporte

- Deve tornar transparente para os usuários possíveis variações da confiabilidade do serviço fornecido pela camada de rede
 - Camada fim a fim: entidade da camada de transporte da máquina de origem se comunica diretamente com a camada de transporte da máquina de destino
 - A entidade de transporte pode estar localizada no núcleo do sistema operacional, em um outro processo do usuário ou em um pacote de biblioteca
-
-

Relacionamentos da camada de transporte



Serviços oferecidos pela camada de transporte

- Similares aos da camada de enlace de dados: visa garantir troca de dados entre origem e destino, tornando transparente o fato de existir redes/sub-redes abaixo
- Oferece serviços
 - Não orientados à conexão
 - Orientados à conexão

Serviço não orientado à conexões

- Serviço de transporte não confiável. A camada de transporte somente mapeia o pedido de transmissão de dados em pacotes para a transmissão pela camada de rede

Serviço orientado à conexões

- Serviço de transporte confiável. O transporte orientado a conexões consiste em ocultar as imperfeições do serviço de rede, de modo que os processos do usuário possam simplesmente supor a existência de um fluxo de *bits* livre de erros

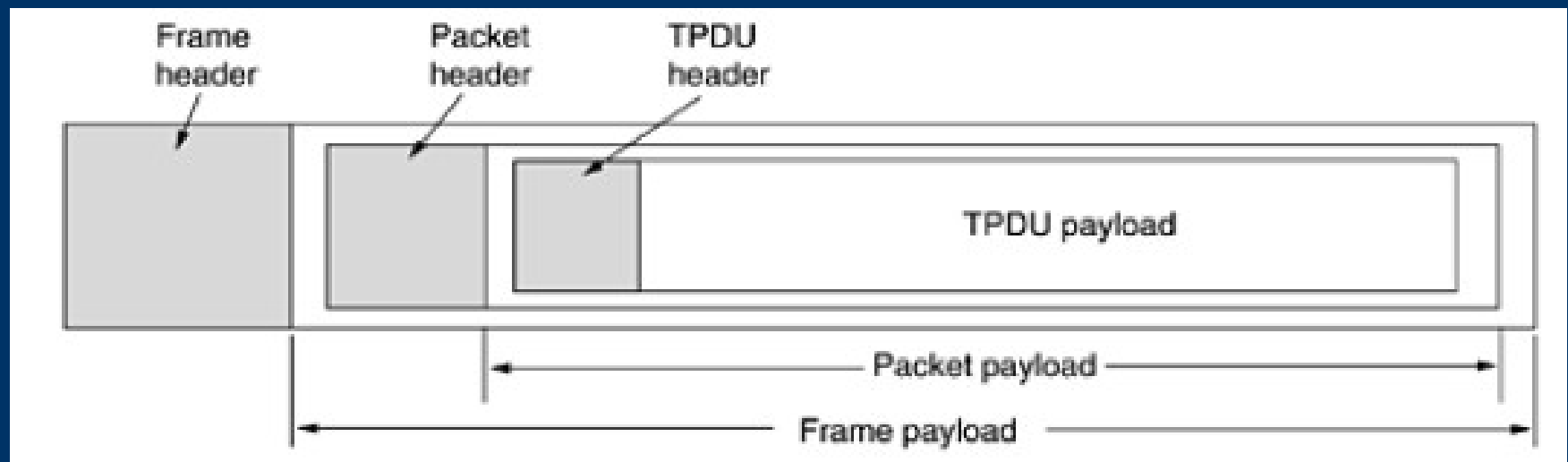
Primitivas de serviço orientado a conexões

- Primitivas para estabelecer, manter e liberar conexões de transporte através de pontos de acesso ao serviço de transporte

Primitiva	Pacote enviado	Significado
LISTEN	(nenhum)	Bloquear até que algum processo tente se conectar
CONNECT	CONNECTION REQ.	Tentar ativamente estabelecer uma conexão
SEND	DATA	Enviar informações
RECEIVE	(nenhum)	Bloquear até chegar um pacote DATA
DISCONNECT	DISCONNECTION REQ.	Este lado quer encerrar a conexão

TPDU – Unidade de Dados do Protocolo de Transporte

- São mensagens enviadas entre duas entidades de transporte
- O cabeçalho do TPDU é que carrega as mensagens relacionadas com as primitivas da camada de transporte



Estabelecimento de conexões

- Parece um procedimento simples, mas não é, pois a camada de rede pode perder, armazenar ou duplicar pacotes com mensagens correspondentes as primitivas para criar conexões
 - Uma solução possível é estabelecer um tempo máximo de vida para cada pacote na rede
 - Pode ser usado um contador do número de “saltos” na rede
 - Pode ser associado a cada pacote a estampa de tempo (*timestamp*) do seu nascimento
-
-

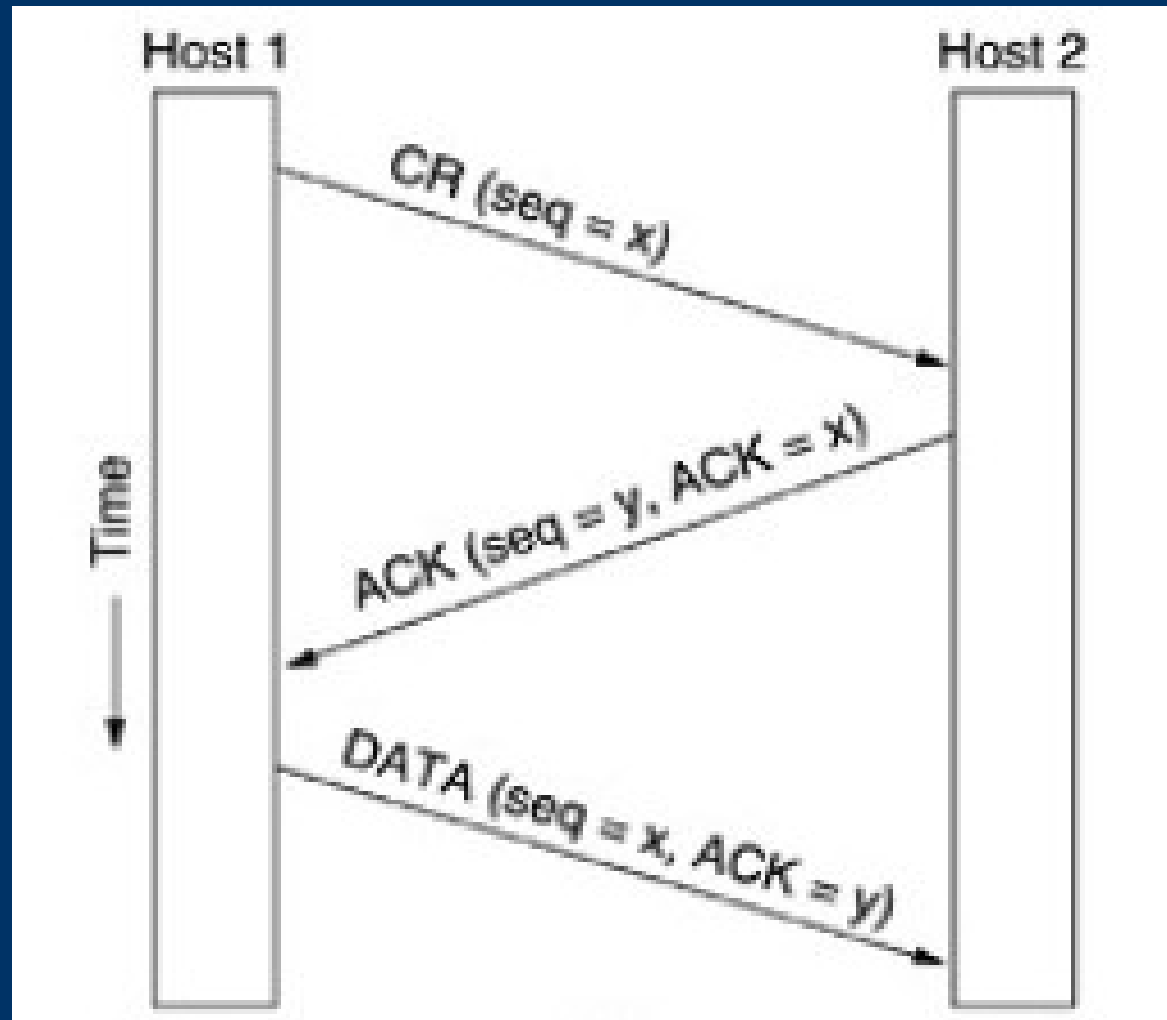
Estabelecimento de conexões

- Método de Tomlinson
 - Cada TPDU é numerada
 - As numerações das TPDUs não se repetem dentro de um período de tempo T
 - Usa-se então o *Tree-Way Handshake* para estabelecer a conexão

Tree-Way Handshake

- Procedimento de conexão normal
 - *Host 1* escolhe um número de seqüência inicial x e o envia em uma *TPDU CONNECTION REQUEST* (CR) para o *host 2*.
 - *Host 2* responde com uma *TPDU ACK* que confirma x e anuncia seu próprio número de seqüência inicial, y
 - *Host 1* confirma o número de seqüência inicial escolhido pelo *host 2* na primeira *TPDU* de dados que enviar
-
-

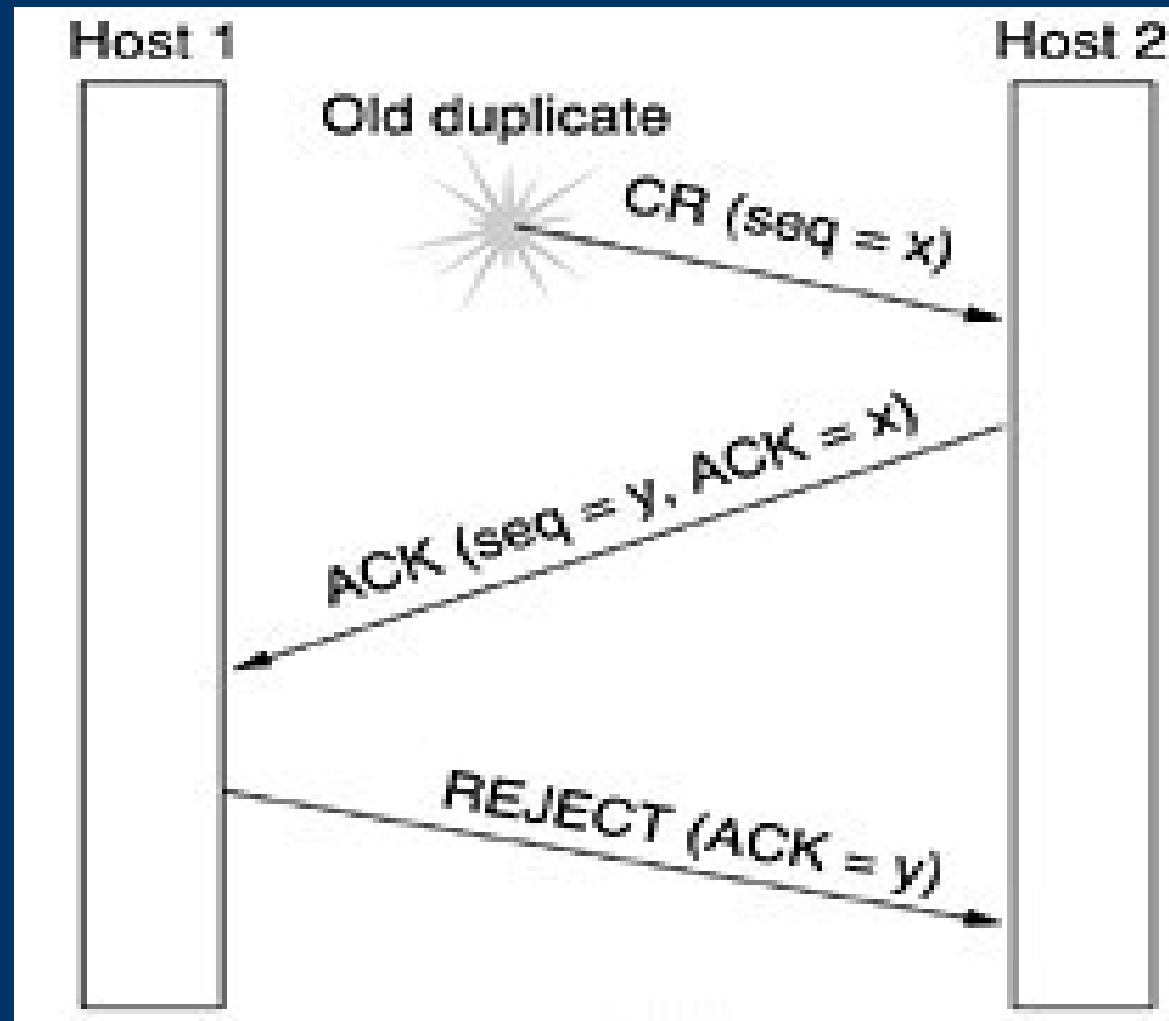
Tree-Way Handshake



Tree-Way Handshake

- Procedimento com duplicata atrasada de *CR*
 - Primeira *TPDU* é uma duplicata atrasada da primitiva *CONNECTION REQUEST* de uma antiga conexão
 - Essa *TPDU* chega ao *host 2* sem o conhecimento do *host 1*
 - O *host 2* reage a essa *TPDU* transmitindo uma *TPDU ACK* ao *host 1*, para verificar se o *host 1* deseja realmente estabelecer uma nova conexão
 - Quando o *host 1* rejeita a tentativa de conexão feita pelo o *host 2*, este percebe que foi enganado por uma duplicata atrasada e abandona a conexão
-
-

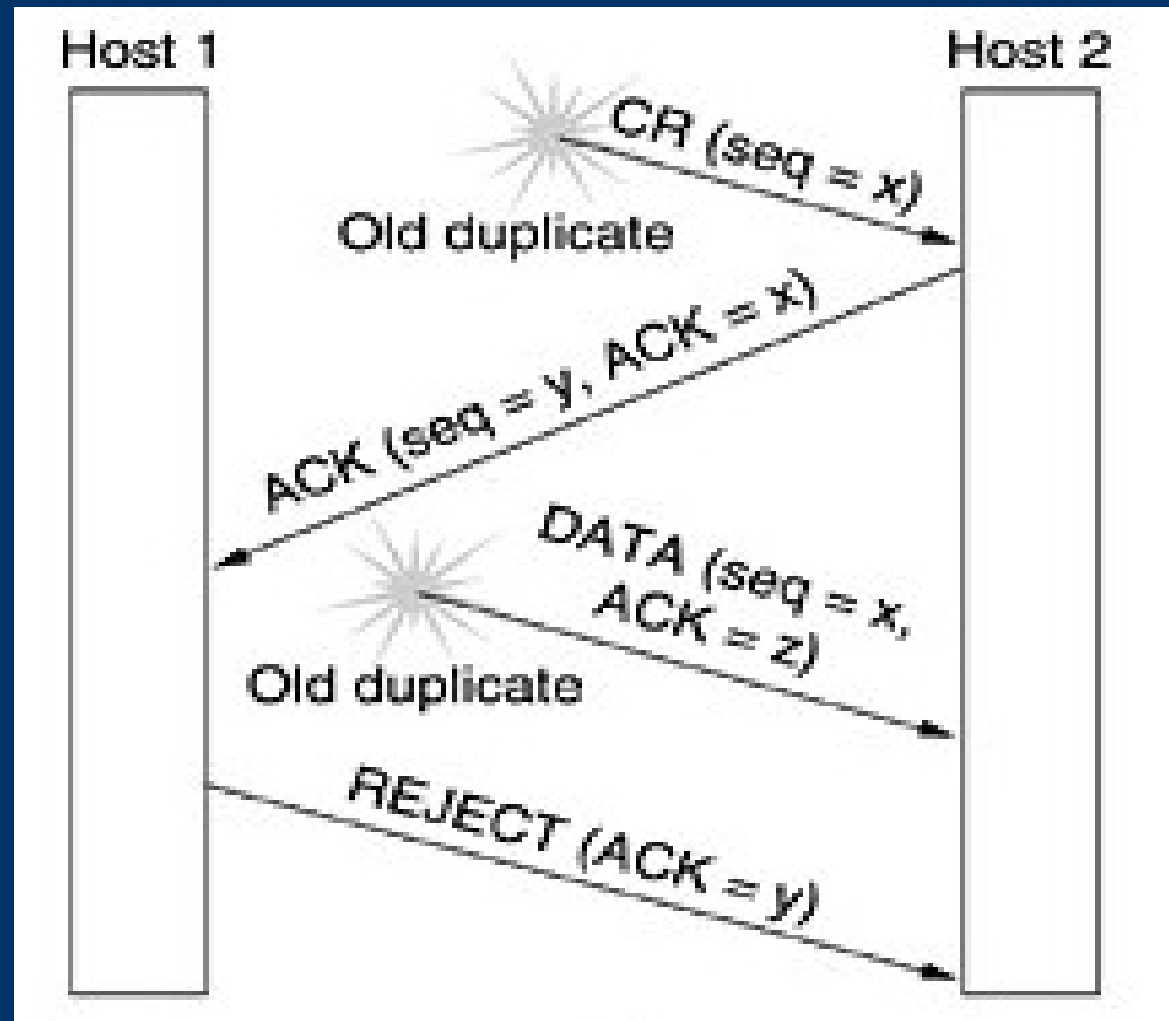
Tree-Way Handshake



Tree-Way Handshake

- Procedimento com duplicatas atrasadas de *CR* e *ACK*
 - *Host 2* recebe uma *CONNECTION REQUEST (CR)* atrasada e responde a ela, usando *y* como número de seqüência inicial para o tráfego do *host 2* ao *host 1*
 - Quando a segunda *TPDU* atrasada chega ao *host 2*, o fato de *z* ter sido confirmado no lugar de *y* faz com que o *host 2* também perceba que se trata de uma duplicata antiga
 - não existe nenhuma combinação de *TPDUs* antigas que possa fazer o protocolo falhar e configurar uma conexão por acidente quando ela não for solicitada
-
-

Tree-Way Handshake



Controles nas conexões

- Controle de Fluxo e de Erros
 - Similar aos métodos usados na camada de Enlace, porém com mais conexões para gerenciar
 - Qualidade de Serviço (*QoS*)
 - Parâmetros definidos pelo usuário
 - Retardos máximos para estabelecer/encerrar conexões
 - Probabilidade de falha no estabelecimento/encerramento da conexão
 - Vazão em cada sentido da conexão
 - Retardo de transferência médio e máximo
 - Taxa de erro
 - Prioridade e probabilidade da queda de conexão
-
-

Encerramento de conexões

- Parece mais simples que o estabelecimento, mas complica quando se deseja evitar que dados sejam perdidos depois que um dos lados encerrou a sua conexão
- Soluções
 - *tree-way handshake* para desconexão
 - um lado encerra apenas depois de um tempo extra de espera por eventuais mensagens a caminho, outro encerra por *time-out*

Protocolos de transporte da Internet

- UDP
 - Protocolo de transporte não orientado à conexão
 - Datagrama não confiável. Recebe os pedidos de transmissão de mensagens pelos processos de aplicação da origem e encaminha ao IP para transmissão
- TCP
 - Protocolo de transporte orientado à conexão

Protocolo TCP

- Projetado para funcionar com base em um serviço de rede sem conexão e sem confirmação
- Capaz de transmitir uma cadeia contínua (*stream*) de octetos (*bytes*) nas duas direções, entre seus usuários
- Normalmente o TCP decide o momento de parar de agrupar os octetos e transmitir o segmento formado

Protocolo TCP

- Cada octeto transmitido é associado a um número de seqüência. O número de seqüência do primeiro octeto contido em um segmento é transmitido junto e é chamado de *número de seqüência do segmento*
 - Os segmentos carregam de “carona” um reconhecimento, que é o número de seqüência do próximo octeto que a entidade transmissora espera receber
 - Segmentos transmitidos são armazenados para eventual retransmissão
-
-

Protocolo TCP

- Receptor pode determinar a janela de transmissão do transmissor, baseada na sua janela de recepção
 - Receptor usa os números de seqüência para ordenar os quadros recebidos
 - Cada segmento tem um *checksum* associado
 - Para permitir vários processos em um único *host*, o TCP utiliza o conceito de porta
 - Cada processo de aplicação sendo atendido possui uma porta associada
-
-

Protocolo TCP

- Para identificação unívoca um endereço TCP em um rede, cada identificador de porta deve ser associado a um endereço IP, e isso define um *socket*
 - Associação de portas a processos pode ser dinâmica, porém processos servidores que são muito usados (FTP, TELNET, HTTP) possuem portas fixas
 - Uma conexão é identificada por um par de *sockets* em cada uma das suas extremidades
-
-

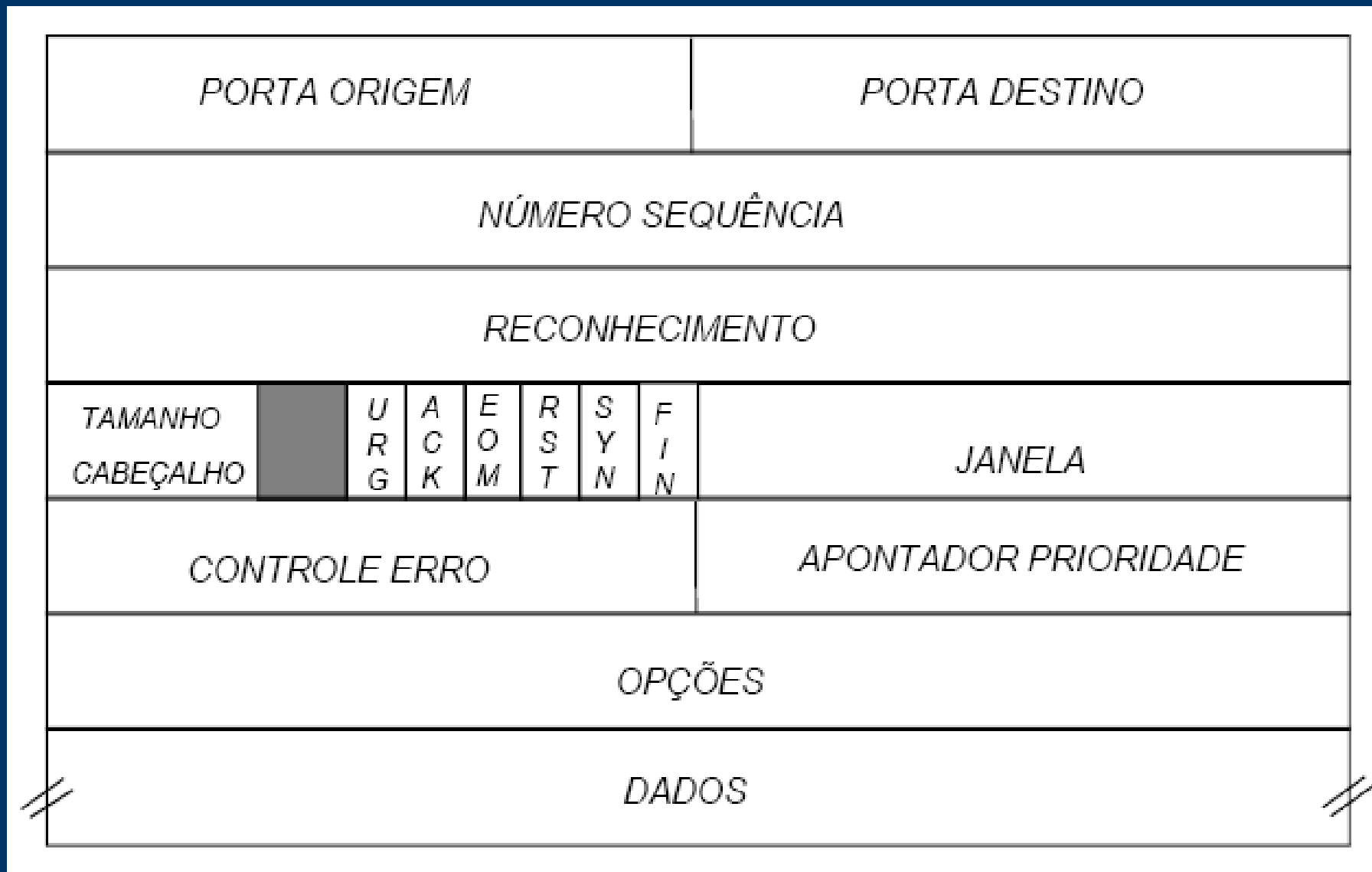
Protocolo TCP

- TCP assume que opera como módulo do sistema operacional e assume que a interface com a rede é controlada por *device drivers*
 - A interface TCP oferecida aos usuários (aplicações) baseia-se em funções (*open, close, send, receive, ...*)
 - A mais conhecida é a *Berkeley Sockets (UNIX)*
-
-

Protocolo TCP

Primitiva	Significado
SOCKET	Criar um novo ponto final de comunicação
BIND	Anexar um endereço local a um soquete
LISTEN	Anunciar a disposição para aceitar conexões; mostra o tamanho da fila
ACCEPT	Bloquear o responsável pela chamada até uma tentativa de conexão ser recebida
CONNECT	Tentar estabelecer uma conexão ativamente
SEND	Enviar alguns dados através da conexão
RECEIVE	Receber alguns dados da conexão
CLOSE	Encerrar a conexão

Quadro TCP



Quadro TCP

- Número de seqüência e reconhecimento: numerar quadros e confirmar recebimento
- Tamanho do cabeçalho: indica o número de palavras de 32 bits contidas no cabeçalho
- URG: Ponteiro de urgência
- SYN: Conexões
 - Pedido: $\text{SYN} = 1$ e $\text{ACK} = 0$
 - Resposta: $\text{SYN} = 1$ e $\text{ACK} = 1$

Quadro TCP

- FIN – liberação de conexão
 - RST – reiniciar conexão
 - EOM – fim de mensagem
 - Controle de Erro = Soma de Verificação
 - Tamanho da Janela: TCP usa janela deslizante
 - Opções: Ex. Informar tamanho dos buffers
-
-