

Redes de Computadores II

Camada de Transporte Visão Geral de Sockets

Prof. Ricardo Couto A. da Rocha

rcarochoa@ufg.br

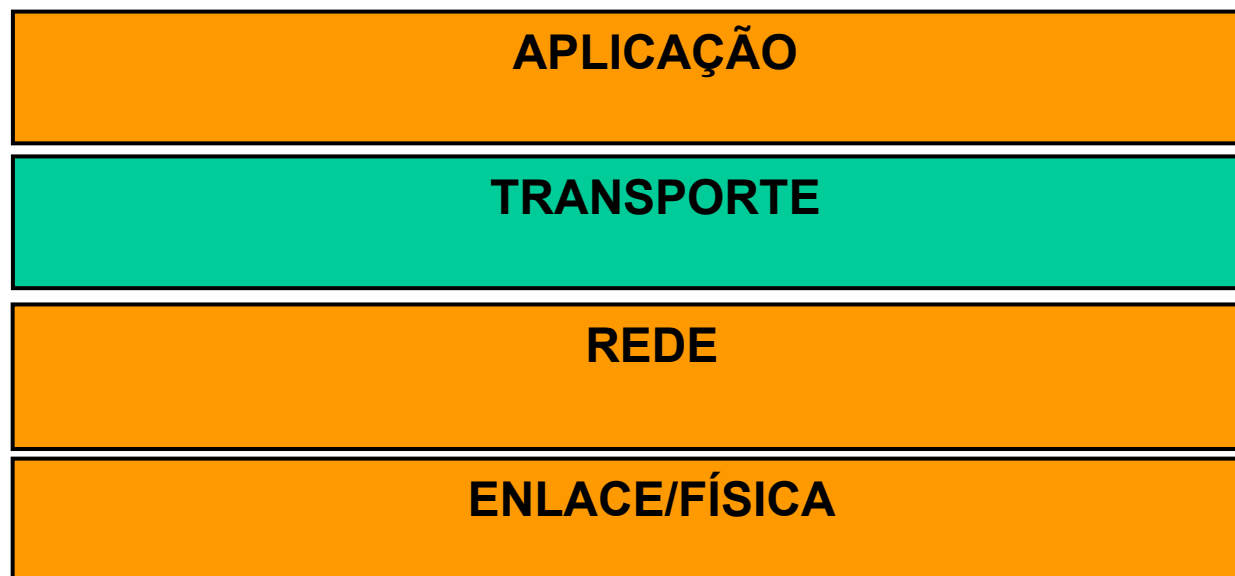
UFG – Regional de Catalão

Roteiro

- Visão geral e objetivos
- Multiplexação
- Gerenciamento de Conexão

Camada de Transporte

Arquitetura TCP/IP



Características

- Responsável pelo transporte dos dados do serviço de transporte
- Comunicação fim-a-fim entre aplicações
- Protocolos orientado a conexão:
 - Controle do fluxo de dados fim-a-fim
 - Controle de erro → detecção e correção de erro fim-a-fim
 - Controle de seqüência (sequenciação)
 - Divisão de mensagens em segmentos

Características

- Mecanismos de identificação de processos
- O nível de transporte oferece:
 - serviços sem conexão (datagramas)
 - serviços orientados à conexão (circuitos virtuais)

Características

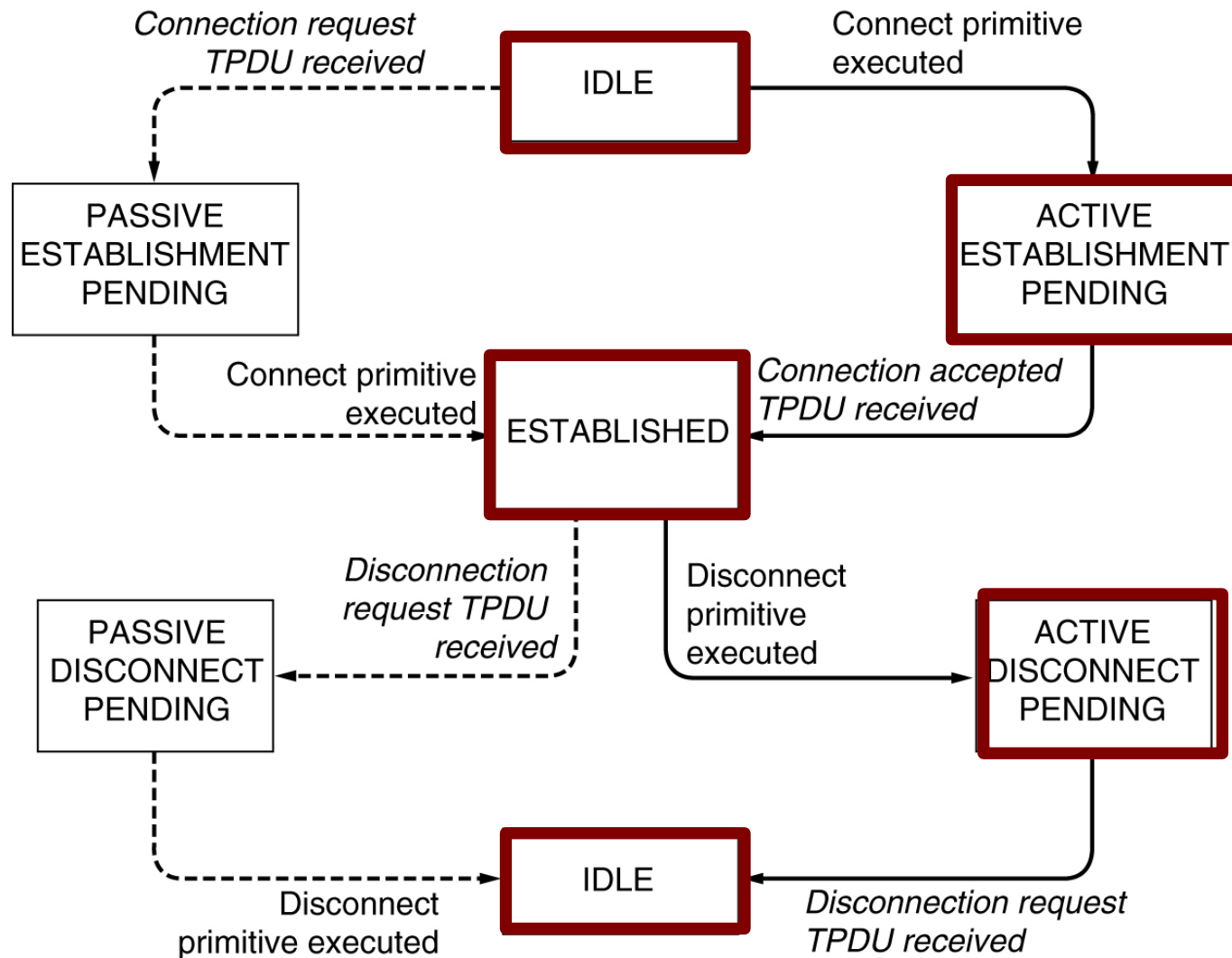
- serviços sem conexão (datagramas)
 - Exemplo: UDP
 - mapeia os pedidos de transmissão para serviços oferecidos pela camada de rede
 - Apresenta baixo overhead
 - Não implementa nenhum mecanismo de confiabilidade
 - ~~Deteção de erro~~
 - ~~Controle de fluxo~~
 - ~~Sequenciação~~

Características

- Serviços orientados à conexão
 - Exemplo: TCP
 - Etapas:
 - Estabelecimento da conexão
 - Transferência de dados
 - Encerramento da conexão

Características

- Ciclo de vida simplificado do TCP



Características - Etapas

- Etapas dos Serviços orientados à conexão
 - Estabelecimento da conexão
 - **Three-Way Handshake**
 - Define o tamanho da janela de transmissão
 - Define número de sequência
 - Transferência de dados
 - Detecção e correção de erros
 - Sequenciação e ordenação dos pacotes
 - Controle de fluxo → sincronização da velocidade
 - Encerramento de conexão
 - Protocolo que provê garantias que os dois lados estão prontos para terminar a conversa (ninguém ainda tem dados a transmitir)
 - Deve considerar encerramento por falha na conexão

Roteiro

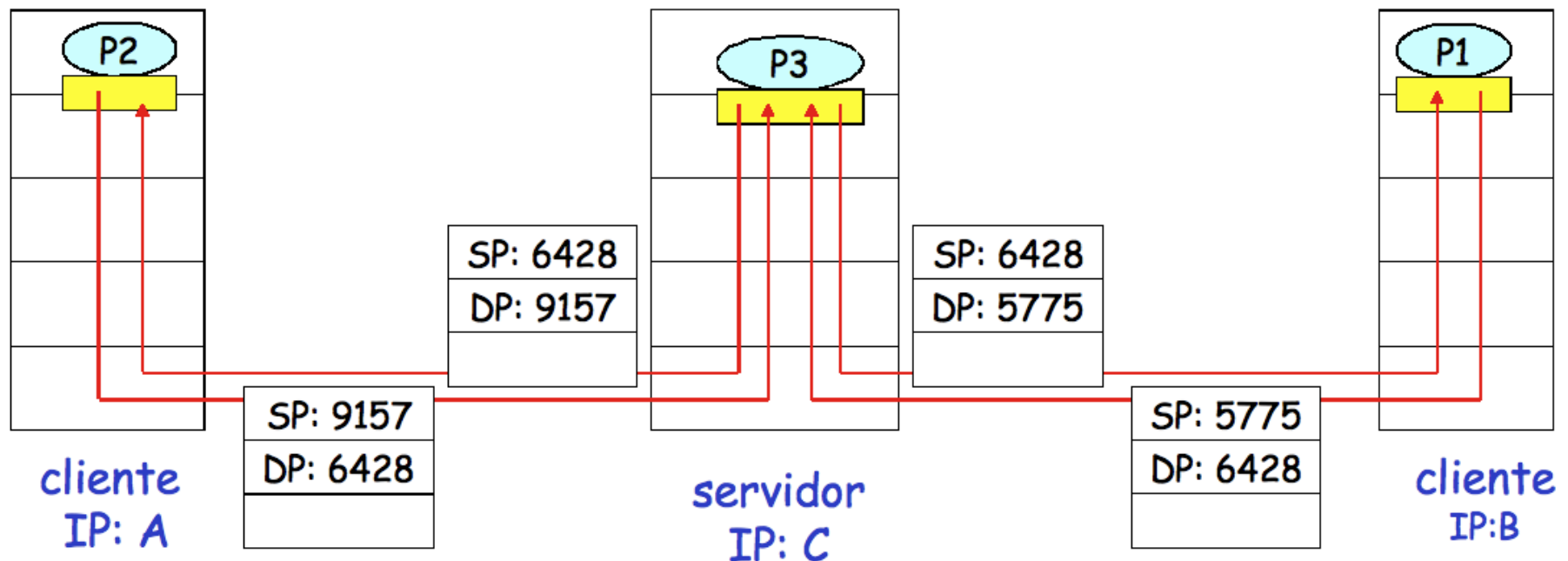
- Visão geral e objetivos
- Multiplexação
- Gerenciamento de Conexão

Multiplexação

- **Multiplexação** → Mecanismo com o qual um meio de comunicação é compartilhado entre várias conversações concorrentes/paralelas
- Na camada de transporte
 - Define como é possível uma ou mais aplicações se comunicarem simultaneamente com vários processos em outras estações
 - Sessão de comunicação na camada de transporte: conexão, em protocolos confiável
 - Especifica o identificador único de uma conexão

Demux não orientada a conexão

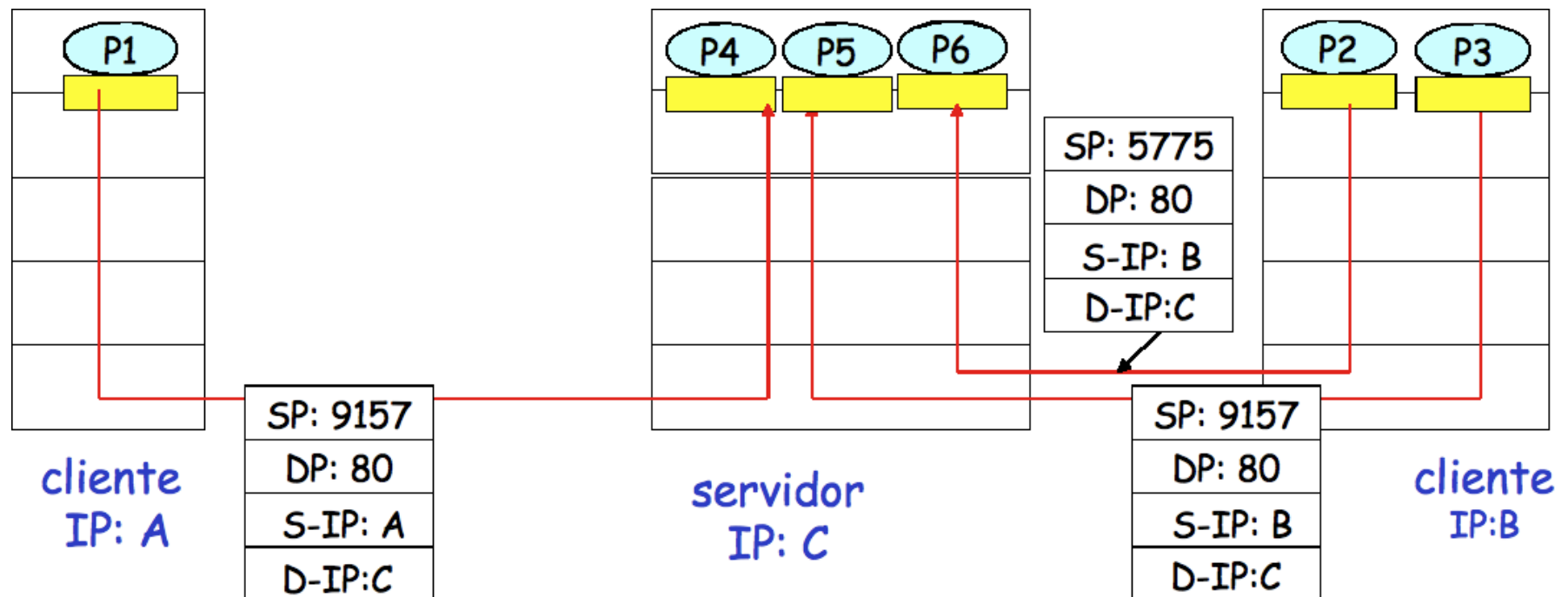
```
DatagramSocket serverSocket = new DatagramSocket(6428);
```



Demux orientada a conexão

- Conexão (socket TCP) é identificado por quatro valores

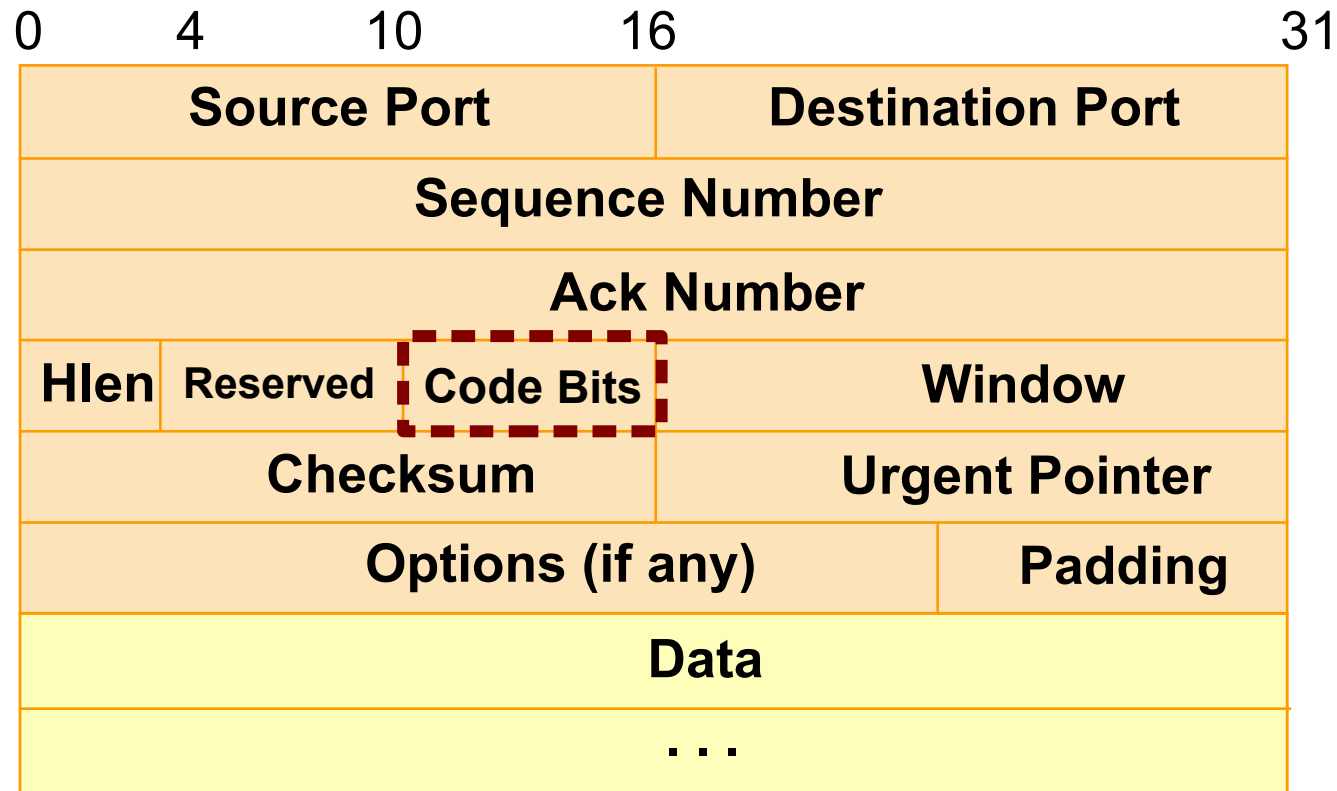
(IP_origem, porta_origem, IP_destino, porta_destino)



Roteiro

- Visão geral e objetivos
- Multiplexação
- Gerenciamento de Conexão

Pacote TCP



Pacote TCP

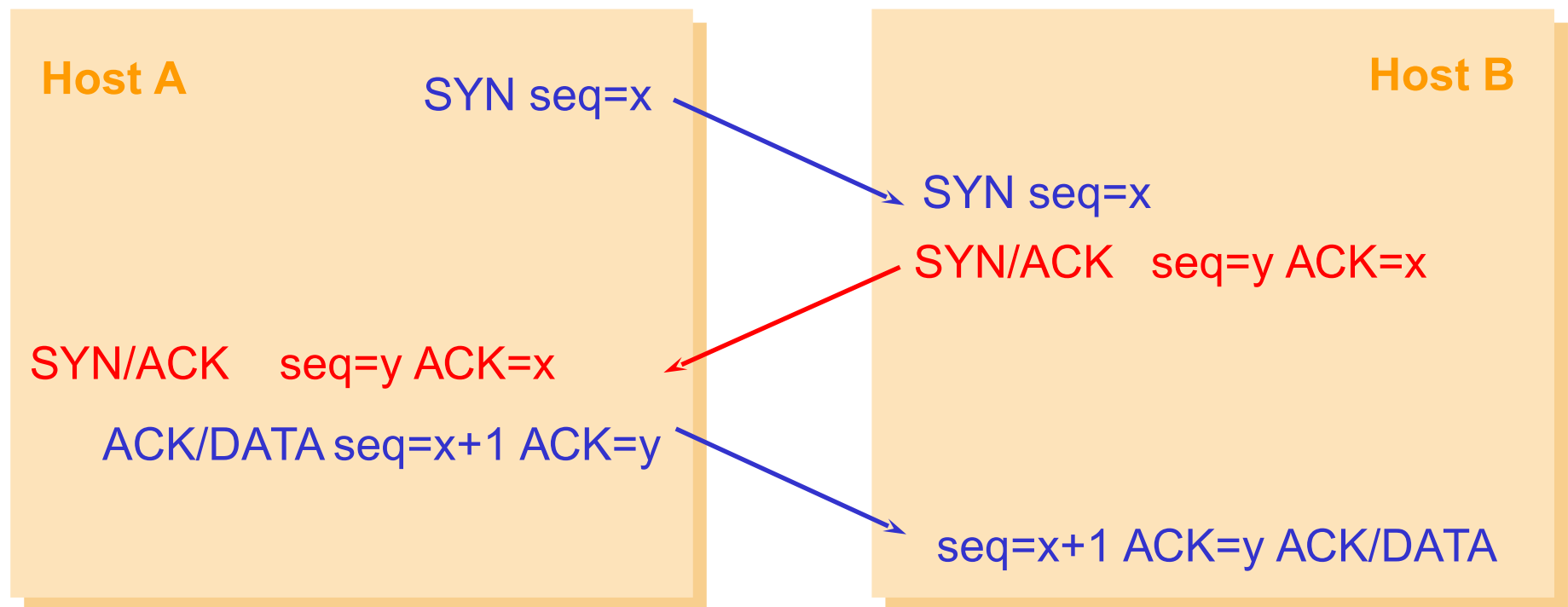
- Campos do Segmento:

- ◆ **Code Bits:**

- URG: Campo Urgent Pointer válido
 - ACK: Confirmação do pedido de conexão
 - PSH: Segmento requer um push
 - RST: Reseta a conexão
 - SYN: Estabelecimento de conexão
 - FIN: Origem finalizou seu stream de bytes.
Encerramento da Conexão.

Estabelecimento de Conexão

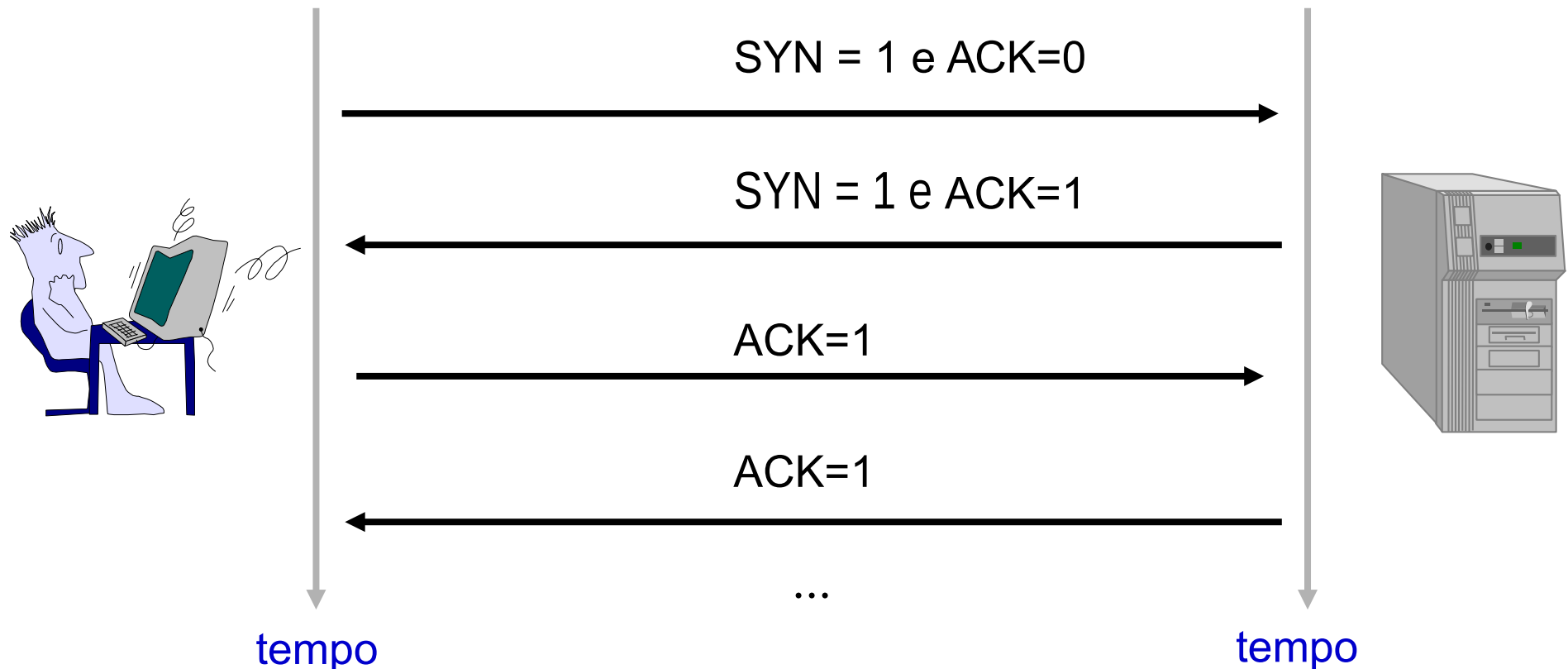
- Exemplo do Three-way Handshake



Flags na Conexão

- Flag ACK

- ◆ Uma conexão TCP sempre se inicia com o cliente enviando um pacote com o flag $SYN = 1$ e $ACK = 0$.



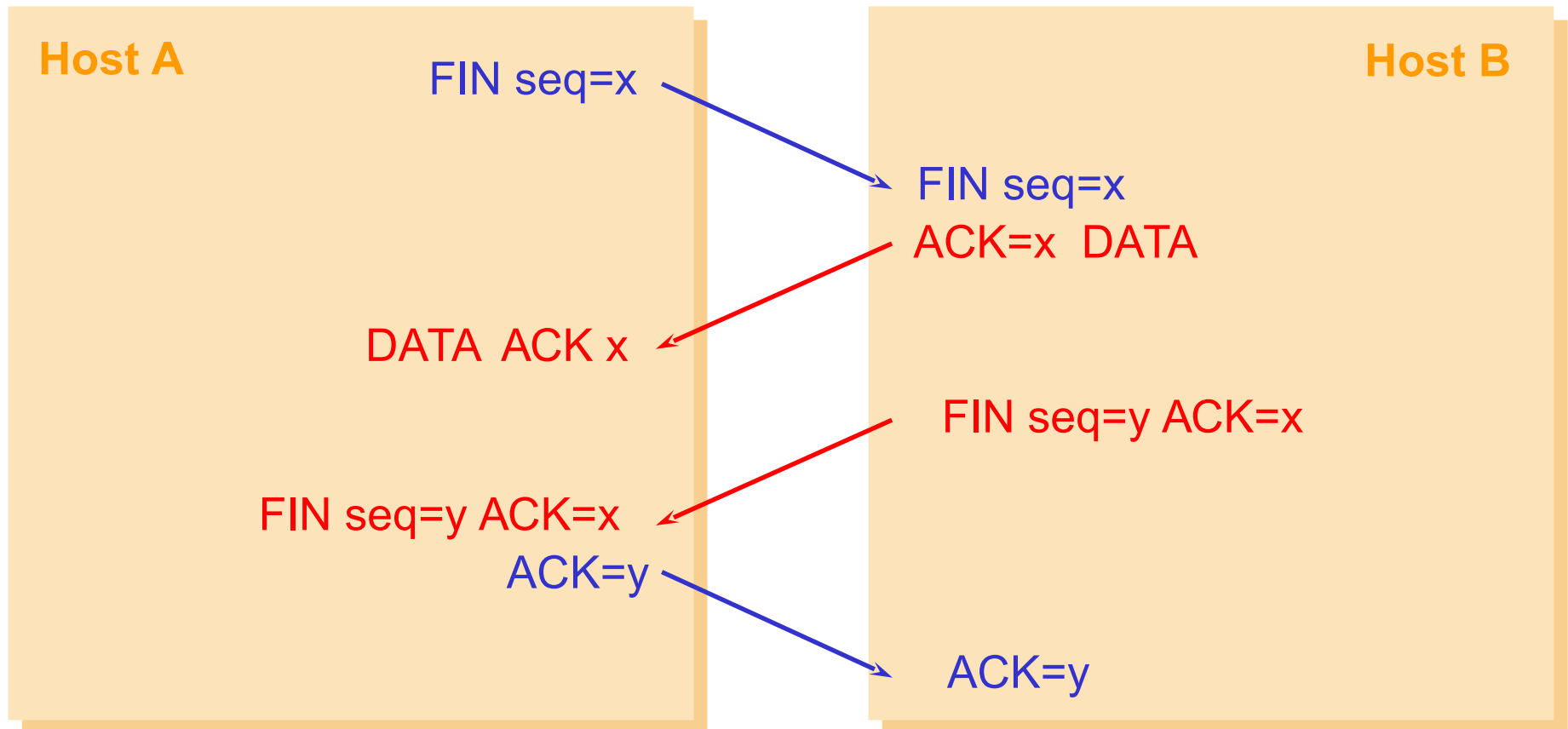
Encerramento de Conexão

- Encerramento de conexão pode ser **intencional** (bem comportado) ou **não-intencional** (por falha e não previsto)
- Encerramento da conexão **por falha**
 - Uma conexão termina depois de decorrido um certo tempo sem que chegue nenhuma **T-PDU** → trivial PDU: pacote sem dados efetivos
 - Se um lado desconectar ou falhar, o outro vai notar a falta de atividade e também se desconectar
 - Para evitar que uma conexão seja desfeita, os participantes devem assegurar o envio de T-PDUs periódicas informando que estão "vivos", quando não têm dados para transmitir;
 - Caso muitas T-PDUs se percam durante uma conexão, um dos lados pode fechar a conexão indevidamente;

Encerramento da Conexão

- Encerramento **bem comportado** da conexão: dois lados concordam e estão a par do encerramento
 - Um problema presente é evitar que dados sejam perdidos depois que um dos lados encerrou a conexão
 - Uma entidade de transporte ao pedir uma desconexão deve aguardar por um tempo antes de fechar a conexão, podendo receber dados durante esse período

Encerramento de Conexão



Estados da Conexão TCP

- **CLOSED**: não há conexão
- **LISTEN**: servidor esperando conexão
- **SYN RECV**: SYN recebido
- **SYN SENT**: aplicação iniciou conexão
- **ESTABLISHED**: transferência de dados
- **FIN WAIT 1**: aplicação finalizou transferência
- **FIN WAIT 2**: outro lado concordou com fechamento
- **TIMED WAIT**: espera por fim dos pacotes
- **CLOSING**: 2 lados tentaram fechar simultaneamente
- **CLOSE WAIT**: outro lado solicitou fim conexão
- **LAST ACK**: espera por fim dos pacotes

Socket Cliente

Socket Servidor

