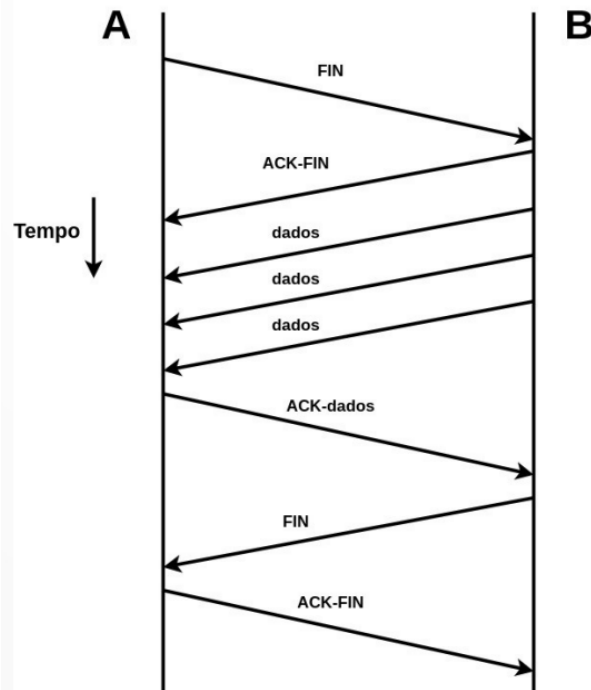


Redes 2

O Protocolo TCP - Continuação

- Encerrando uma conexão TCP
 - A conexão é encerrado de cada lado independentemente



- Se o ACK-FIN for perdido, não há o que fazer. Impossibilidade do acordo entre 2 processos que se comunicam sobre um canal de comunicação que perde mensagens
- Jamais será possível ter um protocolo de encerramento de conexão TCP que termina corretamente 100% das vezes
- Em caso de perda do último ACK-FIN, o processo que transmitiu o último ACK-FIN continua conectado por um período de tempo
- Mas a retransmissão pode acontecer e acontece depois deste período, neste caso TCP envia RST
- Apesar de ser um protocolo “complexo”, com diversos controles, timers, etc. o TCP é um protocolo muito eficiente (a vazão esperada para operação TCP/IP é 90% da taxa de transmissão nominal da rede)

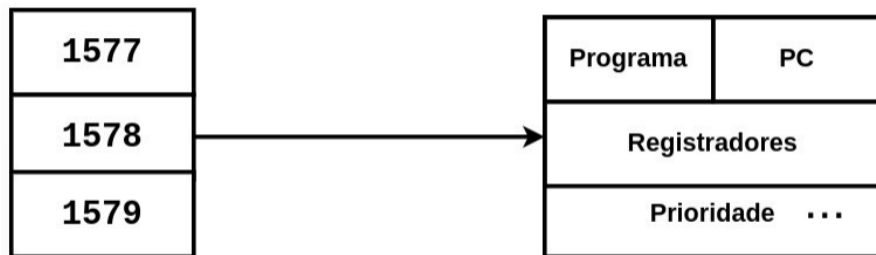
Camada de Aplicação

- Sistemas Cliente-Servidor
 - Servidor: o processo que fica em execução, escutando, aguardando a comunicação
 - Cliente: o processo que inicia a comunicação, dispara a primeira mensagem para o servidor
 - O Cliente oferece uma interface para um usuário (humano ou processo) que é quem demanda a comunicação
 - Importante perceber: tem mais a ver com o modelo de comunicação do que com o processo propriamente dito (mais leve/pesado etc.)
 - Intermediário: peer – ao mesmo tempo cliente e servidor

- O modelo P2P (Peer-to-Peer) pode ser visto como alternativo ao modelo Cliente-Servidor, mas também pode ser visto como parte do modelo como o DNS
- Na Internet, duas classificações são importantes para os sistemas cliente-servidor
 - Protocolo de transporte usado: sobre TCP - orientado à conexão e sobre UDP - não orientado à conexão
 - Concorrência nos servidores: servidores iterativos (que atendem 1 cliente de cada vez) e servidores concorrentes (que atendem múltiplos clientes simultaneamente)

- Concorrência

- Processos concorrentes: processos que executam “ao mesmo tempo”
- Processos paralelos: processos que executam ao mesmo tempo
- Sistemas operacionais: usualmente suportam concorrência
- O servidor deve controlar a concorrência explicitamente: é programada
- Um processo é um programa em execução
 - Cada processo tem um identificador no Sistema Operacional: pid
 - Índice de uma tabela do S.O. com informações sobre o processo
 -



- Programação concorrente

```
int soma; /* variável global */
int main() {
    int i;
    soma = 0;
    fork();
    for (i=1; i<=3; i++) {
        printf("i = %d\n", i);
        soma = soma + i;
    }
    printf("soma = %d\n", soma);
    return 0;
}
```

```
i = 1
i = 2
i = 3
soma = 6
i = 1
i = 2
i = 3
soma = 6
```

- O fork() é uma chamada de sistema que replica o processo em execução, criando um processo filho, e o processo original é chamado “processo pai”
- Retorno do fork
 - Filho: 0 (zero)
 - Pai: pid do filho
- Servidor concorrente
 - Um servidor concorrente cria uma réplica de si próprio a cada cliente, específica para atender aquele cliente
- As duas classificações
 - Sobre TCP: concorrentes ou iterativos
 - Sobre UDP: iterativo