

- **O protocolo TCP**
 - **Características gerais**
 - **comunicação entre processos, full-duplex, MSS (típico 1500 bytes)**
 - **Conexão de 3 msgs, header de 20 bytes, numeração de pacotes e de acks em bytes não em pacotes**
 - **janela do destinatário – para fazer o controle de fluxo**
 - **estrutura do segmento – slide 2**

- **ACKs e sequenciamento**

- **número de sequência = número do primeiro byte no segmento enviado**
- **número do ACK = número do próximo byte esperado**
- **o ACK pega carona nos pacotes de resposta (piggyback) – não há um pacote exclusivo de ACK**
- **ACKs são otimizados pelo lado que recebe. Espera um pouco para confirmar mais segmentos**

- **Controle de fluxo no TCP**

- é uma iniciativa do lado receptor
 - avisa o lado que envia que não tem mais buffer para receber
 - Lado que envia – só envia se:
 - $\text{LastByteSent} - \text{LastByteAcked} \leq \text{RcsWindow}$

- **E o timeout? – pode ou tem que ser grande ou pequeno?**
 - Grande - ???
 - Pequeno - ???
- **Como achar um valor justo??? – baseado no RTT – round trip time**
- **O TCP mantém um histórico e faz uma estimativa sobre esse RTT**
 - **$RTT = 0,9 * RTT + 0,1 * \text{último RTT}$ (Média EWMA)**
 - **slide 14**
 - **$Timeout = RTT + 4 * \text{desvio}$**

- **Gerenciamento da Conexão no TCP**

- **TCP – estabelecimento e fechamento da conexão (connection-oriented)**

- **estabelecimento da conexão**

- **3 mensagens**

- **cliente envia – define início da sequência (seq_cliente_isn)**
 - **servidor envia aceitação - define início da sequência (seq_servier_isn)**
 - **cliente envia reconfirmação que está conectado**

- **slide 16**

- **fechamento da conexão**

- **4 mensagens**

- **cliente envia**
 - **servidor ACK**
 - **servidor envia**
 - **cliente ACK**

- **slide 17**

- **Princípios de controle de congestionamento**
 - **O congestionamento em redes ocasiona**
 - **perda de pacotes, longas filas nos routers, etc ...**
 - **Consequências**
 - **Quando há buffers suficientes – longas filas e atrasos**
 - **Quando não há**
 - **perda de pacotes que precisam ser retransmitidos, diminuindo a capacidade total da rede**
 - **realocação de todos os recursos usados quando da sua retransmissão – idem diminuindo a capacidade**
 - **É melhor tratar bem esse problema**

- **O problema é resolvido de 2 maneiras**
 - **Observando o comportamento da rede**
 - **É o caso do TCP (perda de pacotes – triple ACK e timeout)**
 - **Informação vinda da camada de rede – basta 1 só bit de informação – pode ser feito de 2 maneiras**
 - **o elemento congestionado informa espontaneamente**
 - **envia-se um pedido de informação**
 - **Exemplo – ATM ABR – Available Bit Rate**

- **O caso ATM ABR**

- O ATM também tem CBR (constant), VBR (variable), Ubr (undidentified) – bem sofisticado
- 3 mecanismos de informação efetuado pelos switches da rede
- Uma célula RM é inserida no fluxo a cada 32 células normais
 - 1) todo comutador que se sentir congestionado, pode modificar os bits NI (leve) e CI (pesado). Quando o receptor recebe a célula retorna a mesma para a origem
 - 2) Todo nó pode alterar o bit EFCI de uma célula normal. Quando o destino recebe uma célula assim retorna CI = 1 na próxima RM que aparecer
 - 3) Toda célula RM possui um campo de 2 bytes (ER = explicit rate) que pode ser decrementado por qualquer switch da rede, baixando então a taxa em todos os switches

- **Controle de congestionamento no TCP**

- É uma iniciativa do lado origem
- Vai aumentando o tamanho da janela até que seja observado alguma perda ou timeout – a cada conexão
- O lado origem mantém duas variáveis:
 - **CongWin (congestion window) e Threshold (Limiar)**
- 2 fases distintas – em cada conexão
 - partida lenta – $\text{CongWin} = 1, 2, 4, \dots$ até o Limiar
 - prevenção de congestionamento - $\text{CongWin}++$ até haver alguma perda. Daí, $\text{Limiar} = \text{CongWin}/2$ e $\text{CongWin} = 1$
- slide 34

- **Modelo de latência para o TCP**

- Latência = tempo decorrido entre o pedido de um objeto e sua total entrega
- Limite inferior para a latência: $2RTT + O/R$
 - O = tamanho do objeto; R = taxa de transmissão
 - ocorre quando não há espera por ACKs, isto é, todo o objeto é transmitido sem esperar por confirmação
- Se a janela W fosse fixa e $k = O/(W \cdot MSS)$, isto é, quantos pedaços serão transmitidos sem ACK
 - slide 38
 - Latência = $2RTT + O/R + (k-1)[S/R + RTT - WS/R]$
- Como a janela não é fixa devido à partida lenta o modelo é mais complexo – vide slide 41

- **Casos práticos**

- A partida lenta pode aumentar a latência quando o tamanho do objeto é pequeno e o RTT é grande.
- Não seria esse exatamente o caso da WEB???
- **Ainda há o que ser feito no TCP**