

Camada de Transporte

Redes Industriais

Rone Ilídio – UFSJ – CAP

Itens do Livro

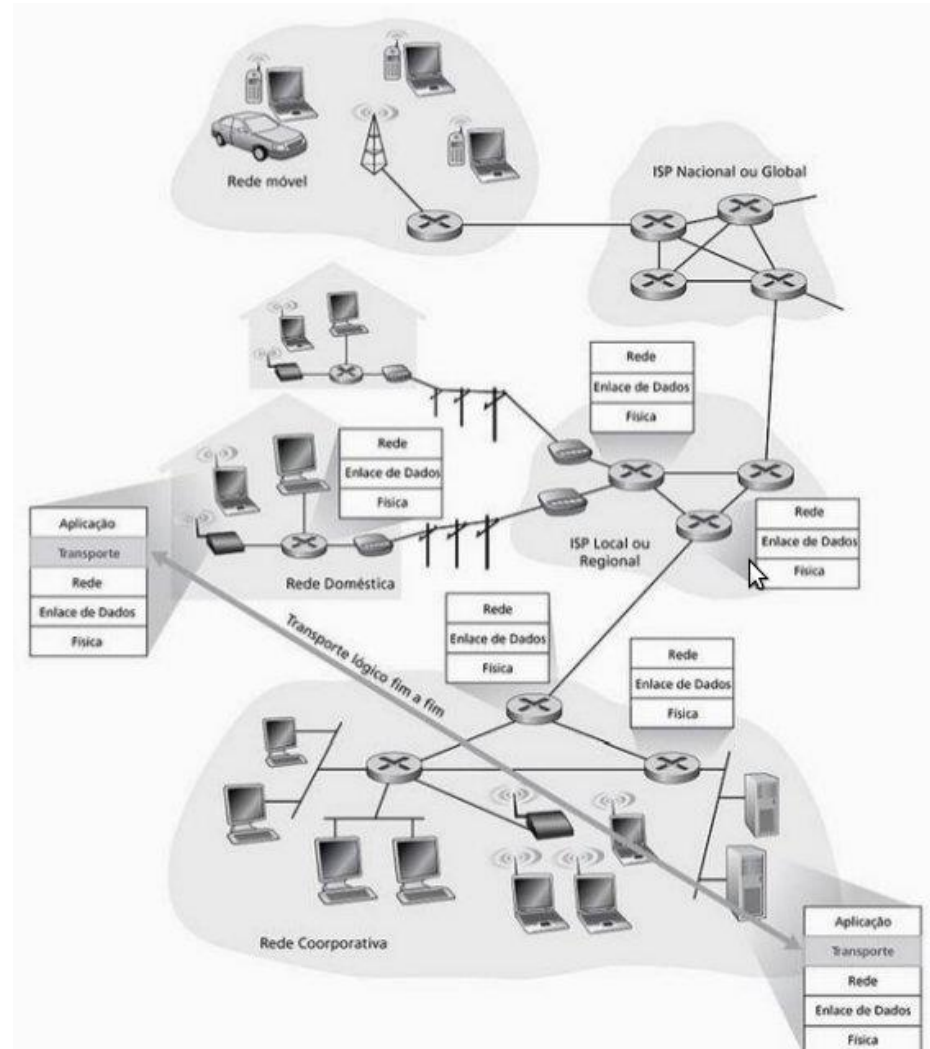
- Livro Rede de Computadores e a Internet
 - Kurose
 - 5ª edição
- 3 Camada de Transporte
- 3.1 Introdução
- 3.3 Transporte não orientado a conexão: UDP
- 3.3.1 Estrutura do segmento UDP
- 3.5 Transporte orientado para conexões: TCP
- 3.5.1 A conexão TCP
- 3.5.2 Estrutura do segmento TCP
- 3.7 Controle de congestionamento TCP

Introdução

A camada de transporte tem a função de fornecer serviços de comunicação diretamente aos processos de aplicação que rodam em hospedeiros diferentes

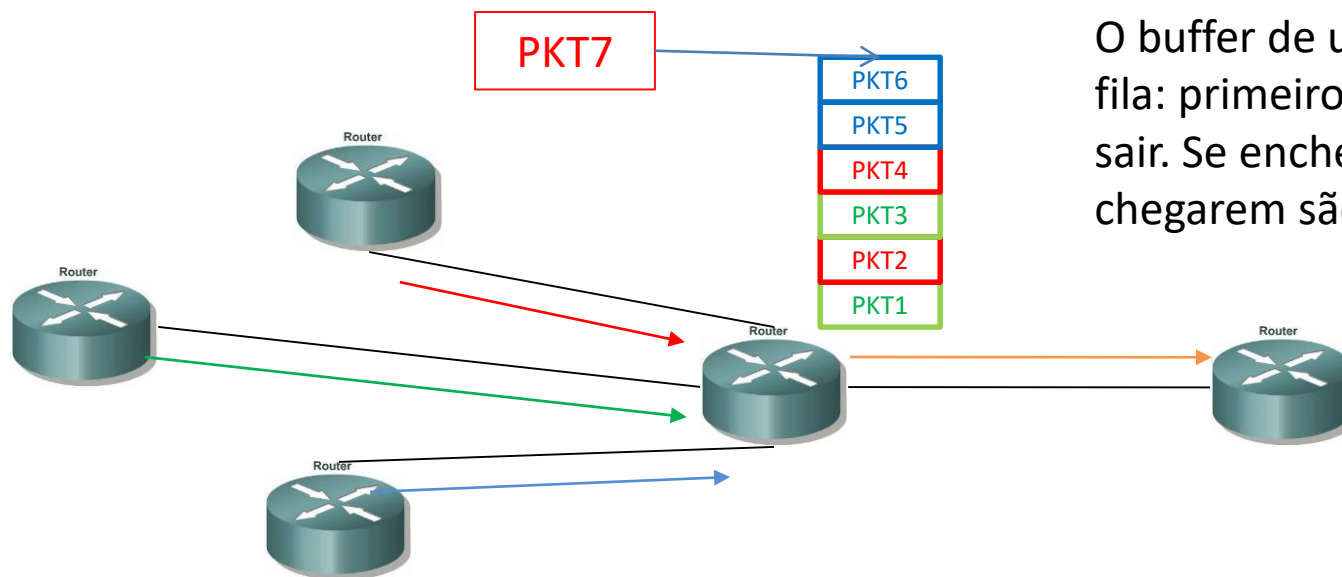
Introdução e serviços da camada de transporte

- Fornece a conexão lógica entre processos
- Divide os dados vindos da aplicação em segmentos
- Dois protocolos: TCP (confiável) e UDP (não confiável)



Transporte não orientado a conexão: UDP

- Fornece 2 serviços
 - Comunicação entre processos
 - Verificação de erros
- Transporte não confiável: Perda de pacotes em buffer cheio da camada de redes



O buffer de um roteador é uma fila: primeiro a chegar, primeiro a sair. Se encher, os pacotes que chegarem são perdidos.

Transporte não orientado a conexão: UDP

- Características do UDP
 - Envio imediato dos dados (o TCP tem controle de congestionamento)
 - Sem conexão: não existe apresentação
 - Não há estado da conexão:
 - Pequena sobrecarga de cabeçalho de transporte: UDP 8 bytes, TCP 20 bytes.
- Muito utilizado para transmissão de video e som

Estrutura do segmento UDP

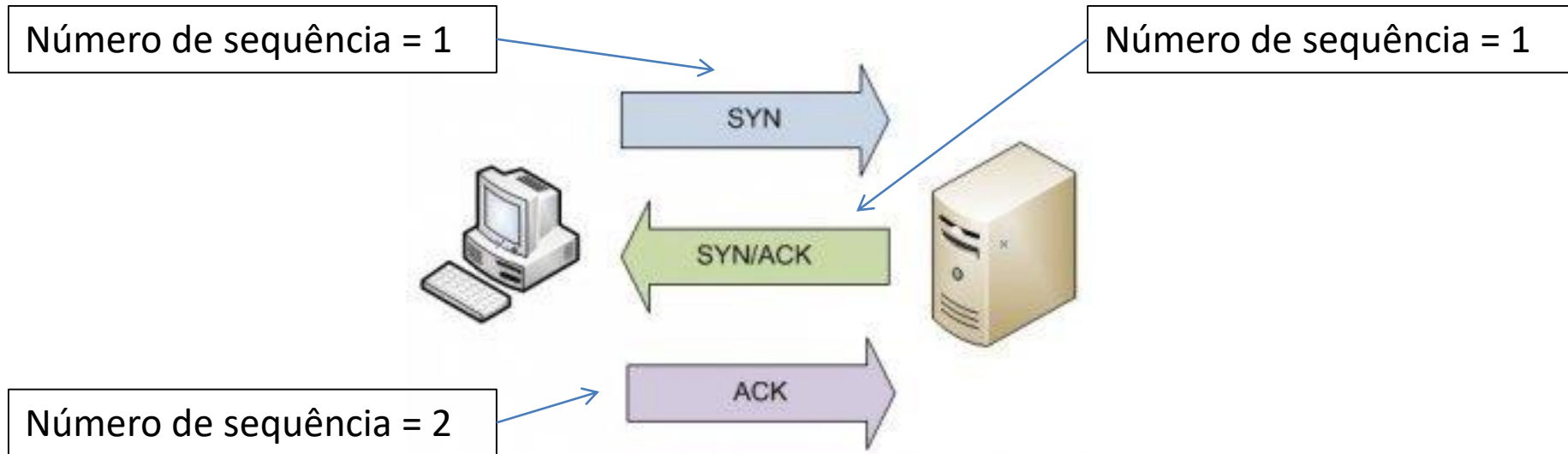
Porta da fonte #	Porta do destino #
Comprimento	Soma de verificação
Dados da aplicação (mensagem)	

Transporte Orientado a Conexão TCP

Introdução - TCP

- Orientado a conexão
 - Conexão entre dois processos (em máquinas distintas)
 - Primeiramente os processos precisam estabelecer uma conexão para depois efetuar a troca de dados
 - Intermediários não armazenam variáveis de estado da conexão
 - Canal full-duplex (ambos)
- Cliente: pede para se conectar
- Servidor: escuta em uma determinada porta para receber conexões

Estabelecimento da Conexão 3-way handshake



- Código no servidor

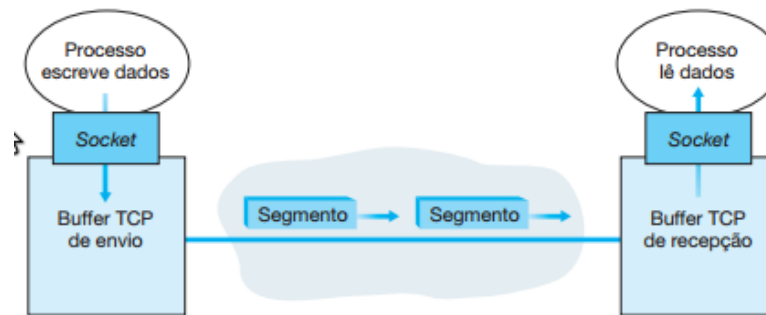
```
ServerSocket socketServidor = new ServerSocket(5000);  
Socket socketCliente = socketServidor.accept();
```

- Código no cliente (quem inicia a conexão)

```
– Socket socket = new Socket("localhost", 5000);
```

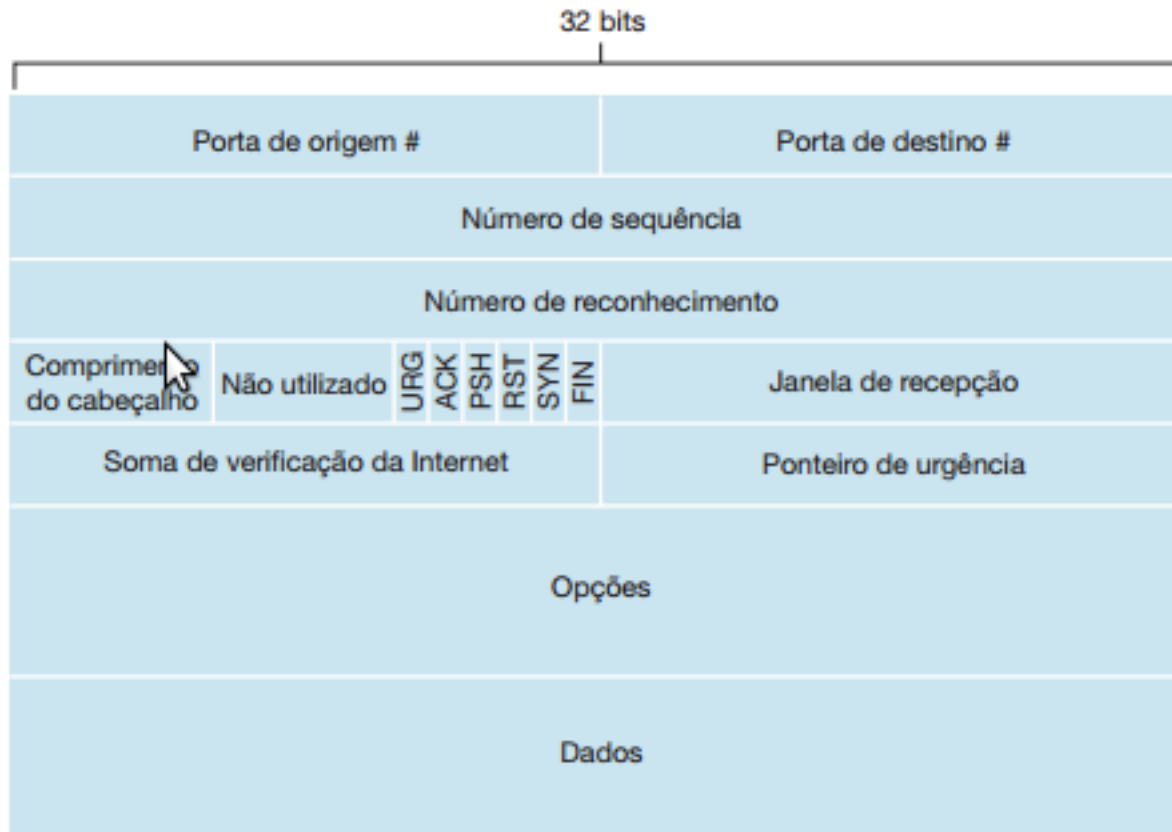
Envio de Dados

- Armazenamento em buffers no cliente e no servidor



- Periodicamente o TCP busca uma certa quantidade de dados e envia
- MSS (maximum segment size): tamanho máximo do seguimento
 - Normalmente é estabelecido para que caiba dentro de um pacote da camada de enlace (MTU – maximum transmission unit)

Segmento TCP

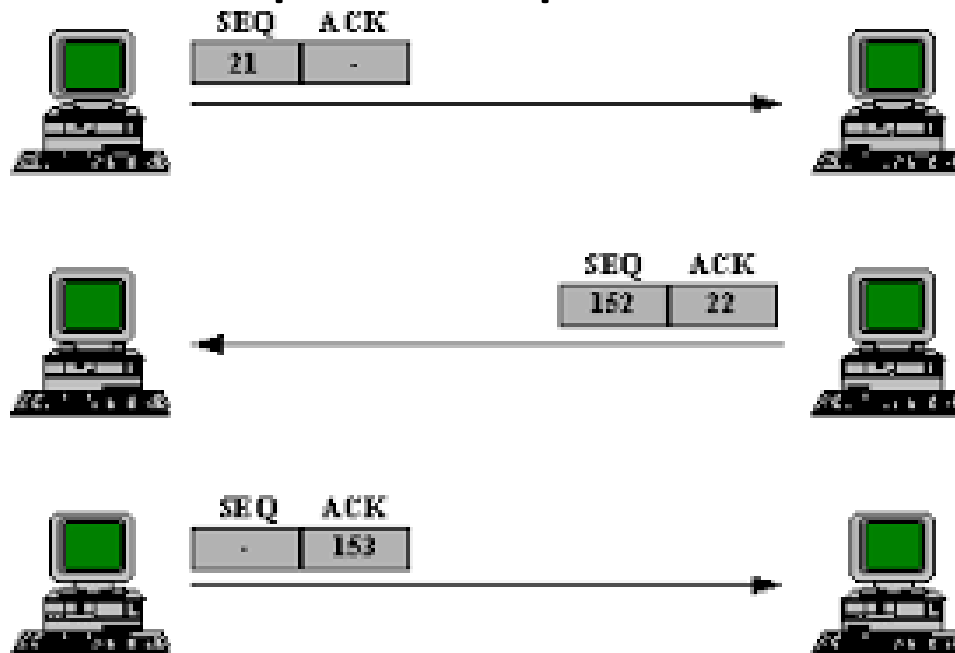


Segmento TCP

- Portas de origem e destino
- Número de sequência
- Número de reconhecimento
- Janela de recepção : número de dados que o receptor está disposto a aceitar
- Comprimento do cabeçalho
- Campo flag: URG, ACK, PSH, RST, SYN E FIN: tipo de forma de tratamento do pacote

Números de sequência e reconhecimento

- Número de sequência ordena os segmentos
- Número de reconhecimento ou ACK: número de sequência esperado pelo destino

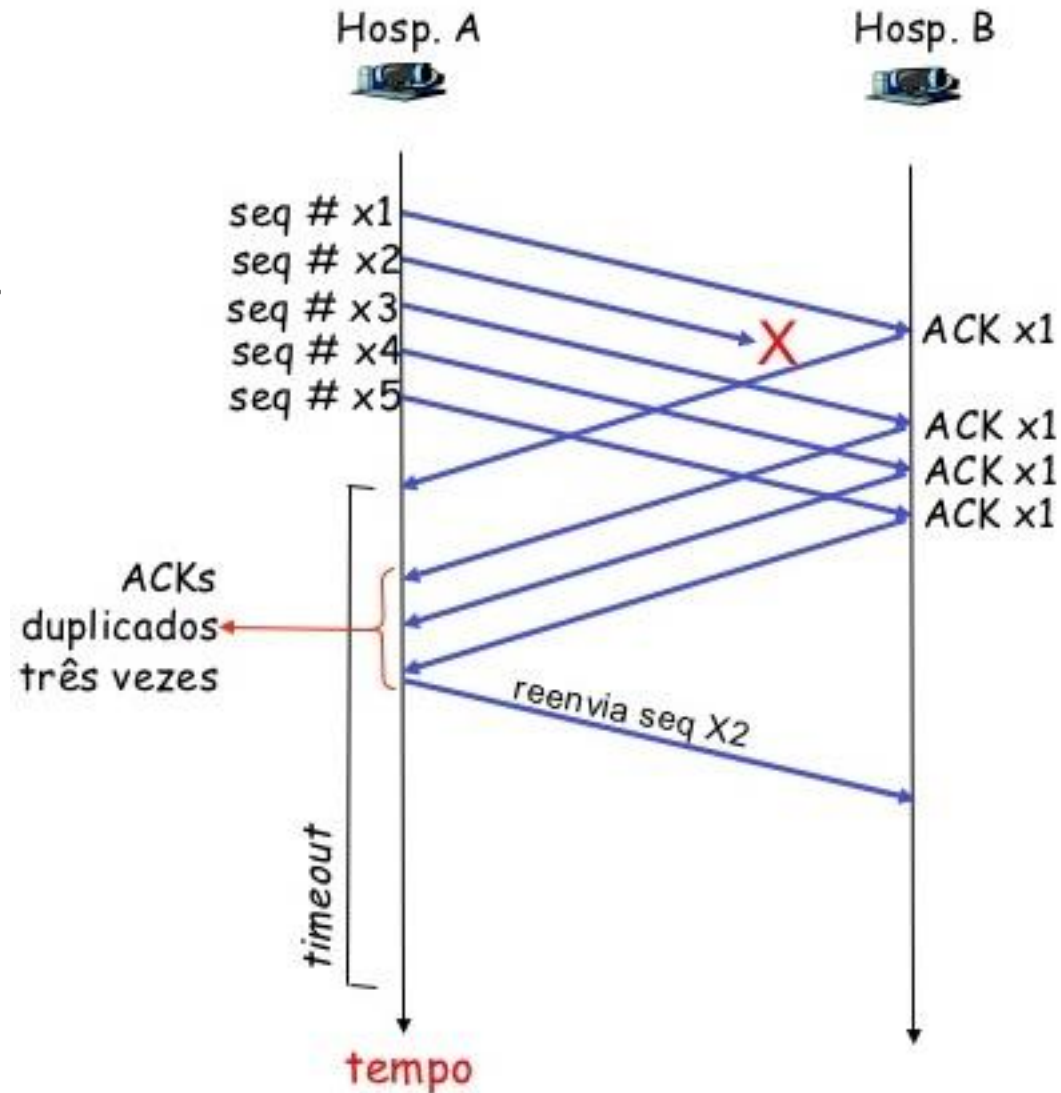


Controle de Congestionamento no TCP

- Princípios:
 - Um segmento perdido implica em congestionamento, portanto a taxa de remetente TCP deve diminuir quando um segmento é perdido.
 - Um segmento reconhecido significa que a rede está enviando os segmentos do remetente ao destinatário e, por isso, a taxa do remetente pode aumentar quando um ACK chegar para um segmento anterior não reconhecido.
 - Busca por lagura de banda.

Controle de Congestionamento no TCP

- Detecção de congestionamento:
 - Final de temporizador
 - Três ACKs duplicados recebidos

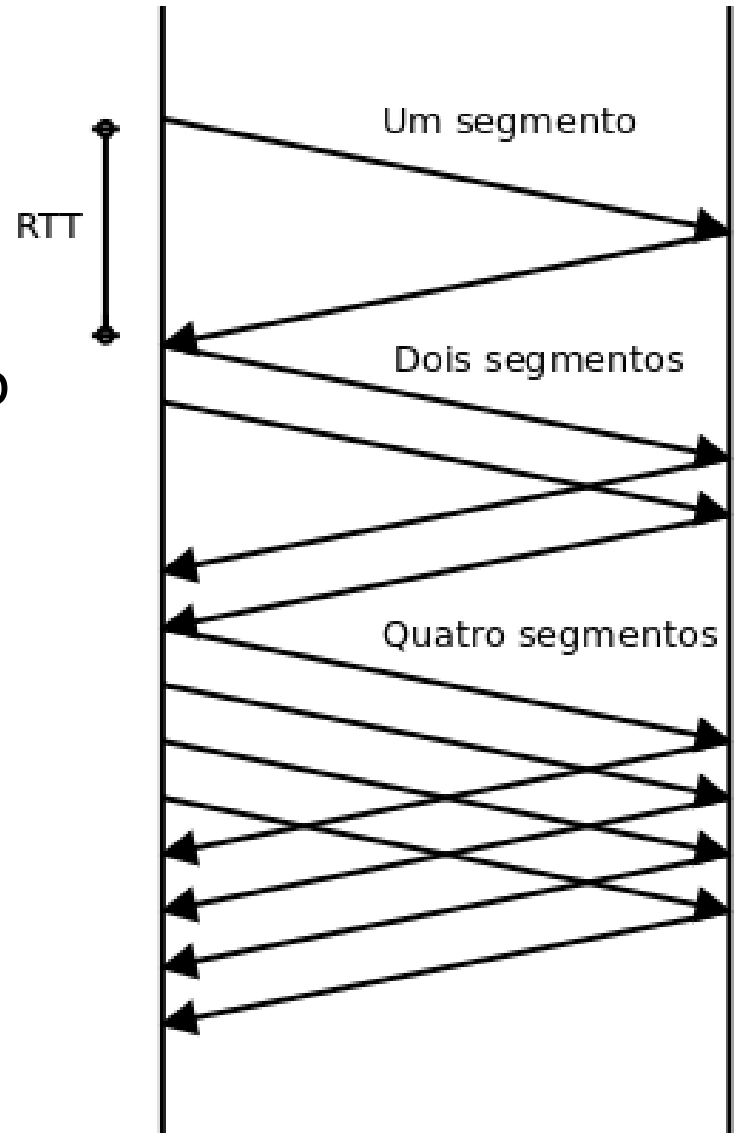


Controle de Congestionamento no TCP

- Variáveis do remetente:
 - **RTT** (Round Trip Time): tempo de viagem de ida e volta de um pacote
 - **MSS** (Maximum Segment Size): maior tamanho dos dados de um pacote TCP, sem considerar o cabeçalhos TCP e IP
 - **cwnd** (congestion window): quantidade de dados que um remetente deve enviar antes de receber um ACK (em segmentos)
 - **rwnd** (receiver window): quantidade de dados que o remetente pode receber

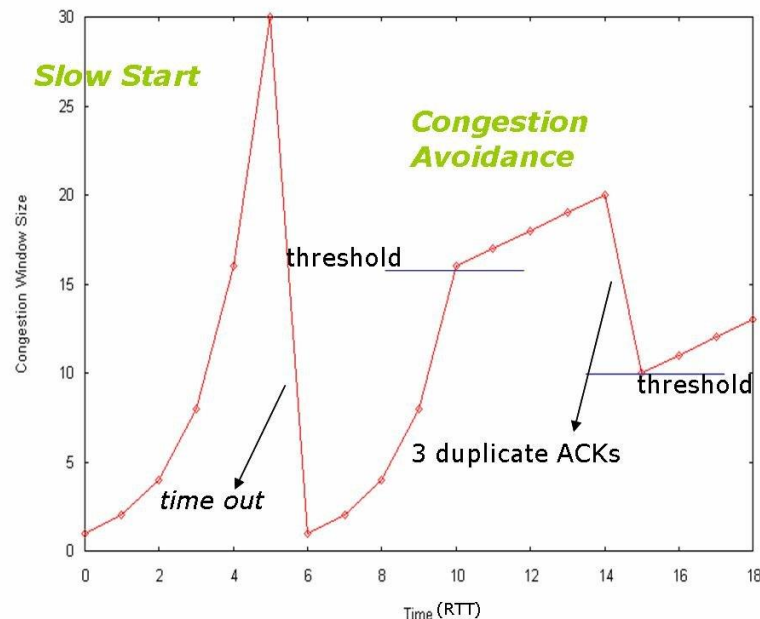
Controle de Congestionamento no TCP

- Partida lenta
 - A cada RTT o cwnd (número de segmentos) dobra
 - Cada segmento possui tamanho MSS
 - Se um ACK não for recebido após RTT (congestionamento), o processo se inicia (cwnd = 1) e o TCP entra no modo prevenção de congestionamento



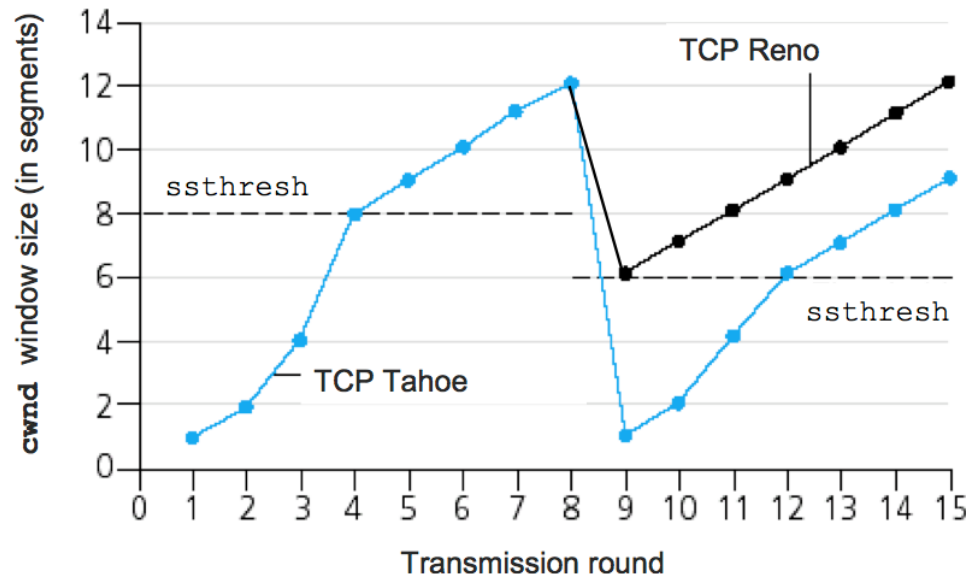
Controle de Congestionamento no TCP

- Prevenção de congestionamento
 - Na detecção de congestionamento cria-se o ssthresh (slow start threshold – limiar de partida lenta) - $\rightarrow$ $\text{ssthresh} = \text{cwnd} / 2$
 - Quando cwnd alcançar o threshold, o incremento de cwnd será de 1 em 1



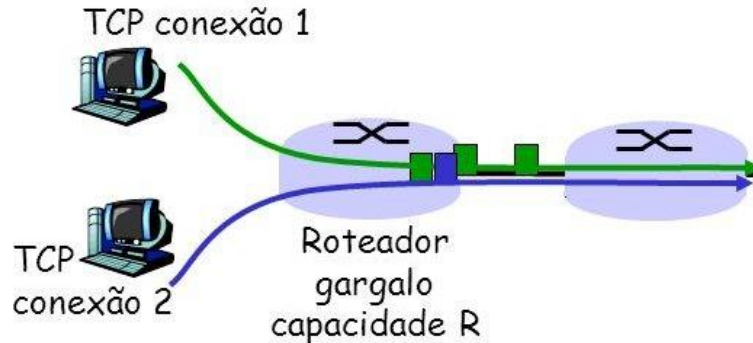
Controle de Congestionamento no TCP

- Recuperação rápida
 - Ao detectar congestionamento, $cwnd = cwnd / 2$
 - Crescimento linear de $cwnd$
- Ocorre na implementação mais nova: TCP Reno
- Na versão antiga, TCP Tahoe, utiliza-se partida lenta

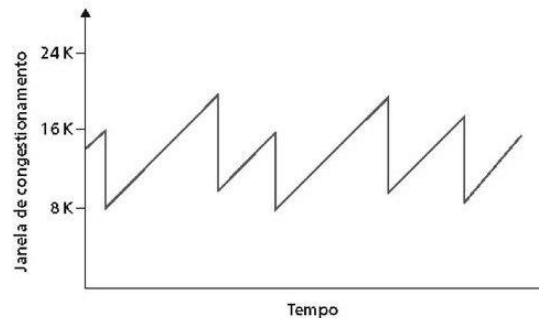


Equidade do TCP

- Meta: se N sessões compartilham o mesmo enlace de gargalo, cada um deve ganhar $1/N$ da capacidade do enlace



- Comportamento dente de serra compartilhado por vários processos



- Um hosp pode abrir mais de uma conexão TCP (exemplo: navegadores de internet) para utilizar melhor a banda disponível