

Camada Transporte

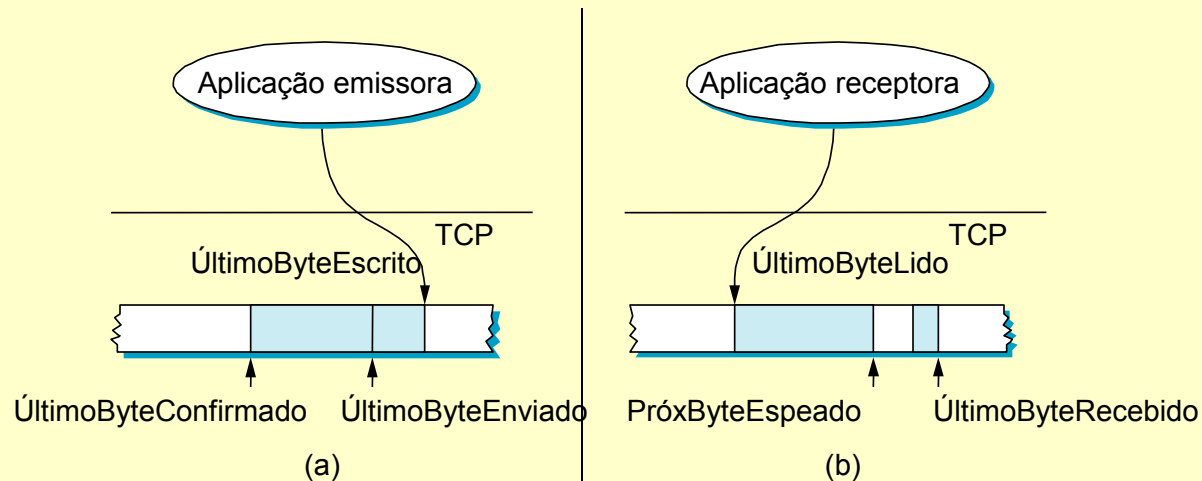
Parte 2

Prof. Dr. S. Motoyama

Algoritmo de Janela Deslizante em TCP

- O TCP clássico emprega um protocolo de janela deslizante com confirmação positiva e sem repetição seletiva.
- O TCP usa o algoritmo de janela deslizante para
 - Garantir a entrega confiável de dados,
 - Assegurar uma entrega de dados em seqüência,
 - Implementar um controle de fluxo entre emissor e o receptor, e
 - Desempenha controle de congestionamento

Revisão de janela deslizante



Lado emissor

- $\text{ÚltimoByteConfirmado} \leq \text{ÚltimoByteEnviado}$
- $\text{ÚltimoByteEnviado} \leq \text{ÚltimoByteEscrito}$
- Buffer de bytes entre $\text{ÚltimoByteConfirmado}$ e ÚltimoByteEscrito

Lado receptor

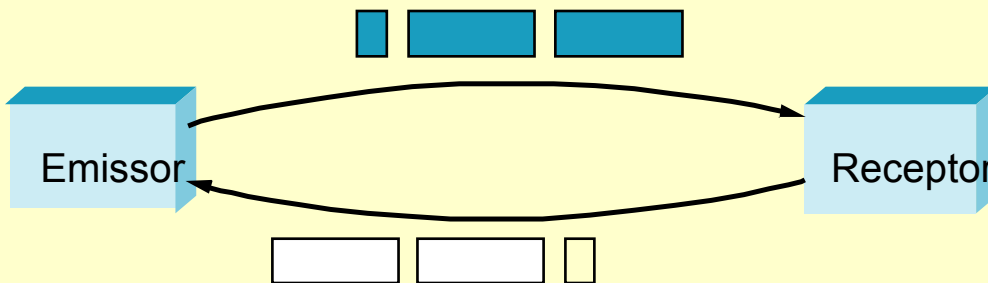
- $\text{ÚltimoByteLido} < \text{PróxByteEsperado}$
- $\text{PróxByteEsperado} \leq \text{ÚltimoByteRecebido} + 1$
- Buffer de bytes entre ÚltimoByteLido e $\text{ÚltimoByteRecebido}$

Controle de fluxo em TCP

- Tamanho máximo do buffer de envio: **BufferEnvioMáx**
- Tamanho máximo do buffer de recepção: **BufferRecMáx**
- Lado receptor
 - $\text{ÚltimoByteRecebido} - \text{ÚltimoByteLido} \leq \text{BufferRecMáx}$
 - $\text{JanelaAnunciada} = \text{BufferRecMáx} - (\text{ÚltimoByteRecebido} - \text{ÚltimoByteLido})$
- Lado emissor
 - $\text{ÚltimoByteEnviado} - \text{ÚltimoByteConfirmado} \leq \text{JanelaAnunciada}$
 - $\text{JanelaEfetiva} = \text{JanelaAnunciada} - (\text{ÚltimoByteEnviado} - \text{ÚltimoByteConfirmado})$
 - $\text{ÚltimoByteEscrito} - \text{ÚltimoByteConfirmado} \leq \text{BufferEnvioMáx}$
 - bloqueia emissor se $(\text{ÚltimoByteEscrito} - \text{ÚltimoByteConfirmado}) + y > \text{BufferEnvioMáx}$
- Sempre envia ACK em resposta à chegada do segmento de dados

Síndrome da janela tola

- Situação: quando **JanelaAnunciada = 0**
 - Não chega ACK e o transmissor não sabe quando a janela anunciada fica maior que 0. O transmissor envia segmentos de 1 byte para, em algum momento, obter o ACK com o valor da janela anunciada.
- Situação (síndrome da janela tola): com qualquer tamanho de janela aberta, até que ponto o emissor explora a janela aberta?



Esteira com segmentos saindo do emissor e entrando no receptor e no sentido inverso, os ACKs saindo do receptor e entrando no emissor. Os segmentos pequenos tendem a circular indefinidamente.

- Soluções no lado receptor
 - Após anunciar janela zero, espere por um espaço igual ao tamanho máximo do segmento (MSS)
 - Atrasar ACKs.

Síndrome : Algoritmo de Nagle

- Do lado do transmissor: Por quanto tempo o emissor espera para enviar dados?
 - muito tempo: atrapalha aplicações interativas
 - pouco tempo: fraca utilização da rede
 - estratégias:
 - Timer fixo (por ex., a cada 100 ms)
 - Auto-clock, disparo na ocorrência de evento (Nagle)
- Algoritmo de Nagle:
 - Quando a aplicação gera dados
 - se preenche um segmento máximo e janela aberta: envia
 - senão
 - se houver dados não confirmados em trânsito: mantém em buffer até que o ACK chegue
 - senão: envia para qualquer tamanho

Proteção contra quebra

Problema: Em uma conexão estabelecida com $\text{NumSeq} = x$, e se o espaço de seqüências de 2^{32} for totalmente consumido num instante, no instante seguinte o NumSeq será novamente x . Esse esgotamento de espaço de seqüências dependerá da velocidade com que o byte será enviado, isto é, da largura de banda. O tempo para uma seqüência de 2^{32} seja consumida em um enlace de 1,5 Mbps será $8 \times 2^{32} / 1,5 \times 10^6 = 22906,492$ seg., ou aprox. 6,4 horas.

Largura de banda	Tempo até a quebra
T1 (1,5 Mbps)	6,4 horas
Ethernet (10 Mbps)	57 minutos
T3 (45 Mbps)	13 minutos
FDDI (100 Mbps)	6 minutos
STS-3 (155 Mbps)	4 minutos
STS-12 (622 Mbps)	55 segundos
STS-24 (1,2 Gbps)	28 segundos

Mantendo a tubulação cheia

Tamanho da janela exigido para um RTT de 100 ms

Largura de banda	Retardo x Largura de banda
T1 (1.5 Mbps)	18KB
Ethernet (10 Mbps)	122KB
T3 (45 Mbps)	549KB
FDDI (100 Mbps)	1,2MB
STS-3 (155 Mbps)	1,8MB
STS-12 (622 Mbps)	7,4MB
STS-24 (1.2 Gbps)	14,8MB

Com um campo de 16 bits, pode-se anunciar uma janela de somente 64 KB

Extensões do TCP

- Como o cabeçalho do TCP possui comprimento variável, pode ser usado para transportar informações adicionais.
- Nem todos os hosts precisam reconhecer as extensões, entretanto, quando o emissor e o receptor concordarem em utilizá-las devem fazer durante a fase de estabelecimento de conexão.
- Extensão 1: Melhor estimativa do RTT
 - Escreva um timbre (marca) de hora de 32 bits no segmento que está transmitindo.
 - O receptor simplesmente ecoa esse timbre de hora de volta ao emissor no segmento de ACK.
 - O emissor pode ler o seu relógio quando recebe o ACK e computar uma estimativa mais precisa do RTT.

Extensões do TCP

- Extensão 2: NumSeq “aumentada”
 - Escreva o timbre de hora de 32 bits no segmento transmitido.
 - Usando o campo de NumSeq de 32 bits junto com o timbre de hora, aumenta o intervalo para 64 bits.

Como o timbre de hora é monotonicamente crescente, na realidade não há aumento de espaço, mas uma maneira de detectar se um NumSeq está repetido.

- Extensão 3: JanelaAnunciada maior
 - Redes de alta velocidade tem tubos atraso x largura de banda maiores. Para mante-los cheios, pode necessitar mais do que 16 bits no campo JanelaAnunciada.
 - Um fator de escala pode ser usado junto com JanelaAnunciada.
 - Se o fator de escala for 2, então a JanelaAnunciada estabelece que a unidade não é mais 8 bits, mas 16 bits.

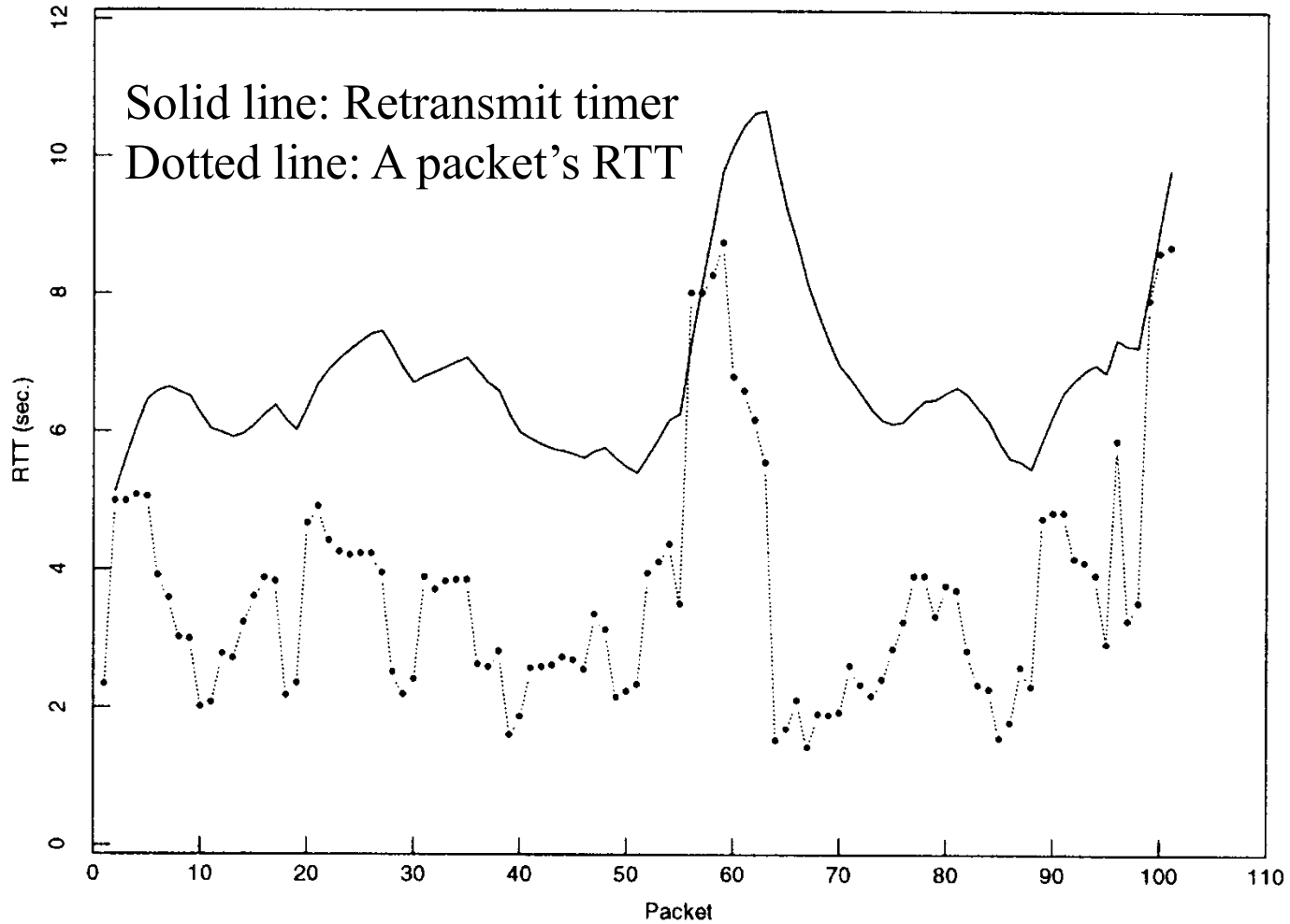
Retransmissão adaptável

- TCP sempre retransmite o segmento se o correspondente ACK não chegar. Define o timeout em função do RTT.
- Problema: Como um emissor TCP determina o seu valor de timeout?
 - Se o valor de timeout é super estimado, atrasa a retransmissão.
 - Se o valor timeout é sub estimado, envia pacotes duplicados na rede.
- TCP usa algoritmo de transmissão adaptativa para acomodar atrasos variáveis na Internet:
 - Um emissor TCP monitora o tempo de ida e volta, RTT (round trip time), por medição grosseira ou com maior precisão.
 - Recuo exponencial

Medições de RTT e timeout

- Dada uma nova medição de RTT, M , TCP atualiza uma estimativa média de RTT por $R = \alpha R + (1 - \alpha)M$.
 - α é uma constante ($0 < \alpha < 1$), que define quanto a nova medição contribui para a estimativa.
 - α é geralmente ajustado em 0.9.
- O valor de timeout RTO é ajustado para $\beta \times R$.
 - β leva em conta a variação em RTT.
 - β é geralmente ajustado em 2.

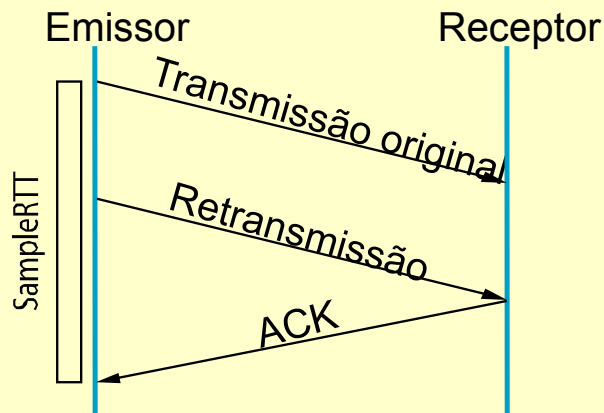
Medições de RTT e timeout



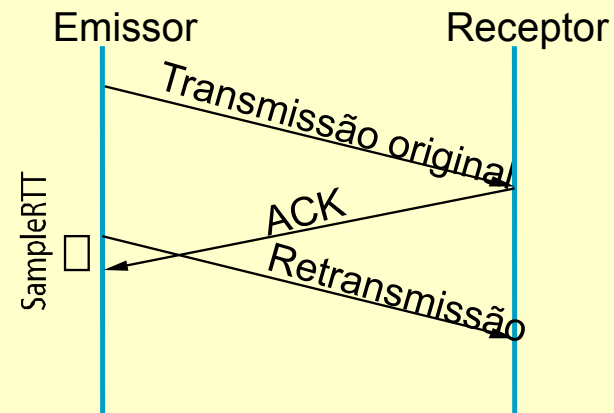
Ref.: "Van Jacobson, "Congestion avoidance and control," *Proc. SIGCOMM*, vol. 18, no. 4, Aug. 1988"

Ambigüidade de Retransmissão

- Problema: Quando ocorre retransmissão, como o emissor sabe se o ACK recebido é do segmento original ou do segmento retransmitido?
- A resposta a esta pergunta afeta a estimacão de timeout.
- Uma soluçãõ simples é excluir da medições de RTT os segmentos retransmitidos.



(a)



(b)

Algoritmo de Karn/Partridge

- Quando ocorre um timeout ou retransmissão,
 - Não atualize o estimador de RTT quando chega o ACK do segmento retransmitido.
 - Além disso, coloque $RTO = \gamma RTO$, $\gamma > 1$ (backoff RTO).
Tipicamente o γ é 2.
 - Calcule um novo RTO quando chega o ACK de um segmento que não foi retransmitido.

Algoritmo de Jacobson/Karels

- Estima a variação de RTT por

$$Diff = M - R$$

$$R = R + (\delta \times Diff)$$

$$D = D + \delta \times (|Diff| - D).$$

– $D = desvio$

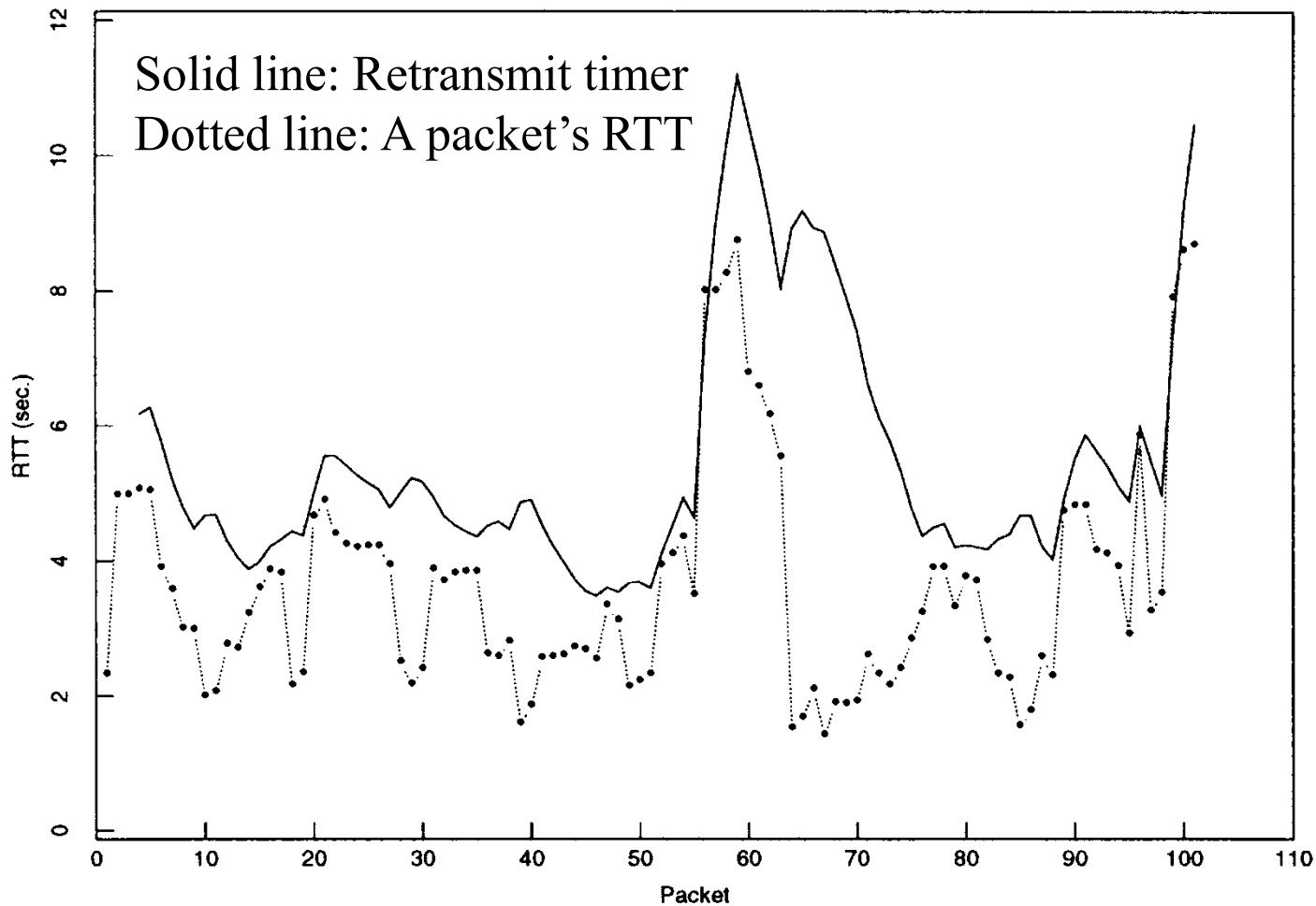
– δ é um valor entre 0 e 1.

- O valor de timeout é agora dado por

$$RTO = \mu \times R + \phi \times D.$$

– μ é tipicamente ajustado em 1 e o ϕ em 4.

Algoritmo de Jacobson/Karels



Ref.: "Van Jacobson, "Congestion avoidance and control," *Proc. SIGCOMM*, vol. 18, no. 4, Aug. 1988"

Desempenho

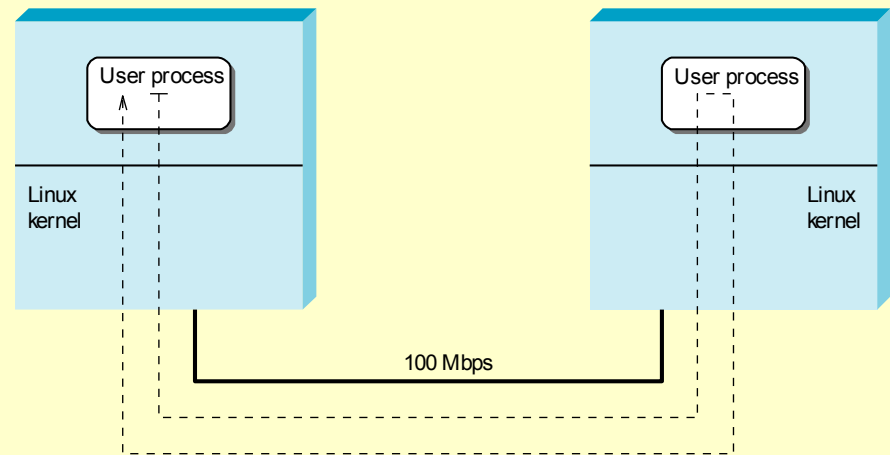
- Como podemos medir o desempenho da rede?
- Quais são as métricas que são necessárias?
- Modelo teórico versus modelo experimental.
 - Modelo teórico:
 - Matemático: Teoria de fila
 - Simulação: Utilizações de pacotes software ou linguagens de programação

Experimental

Objetivo: Medir o desempenho fim-a-fim da rede (hardware e software).

Cenário

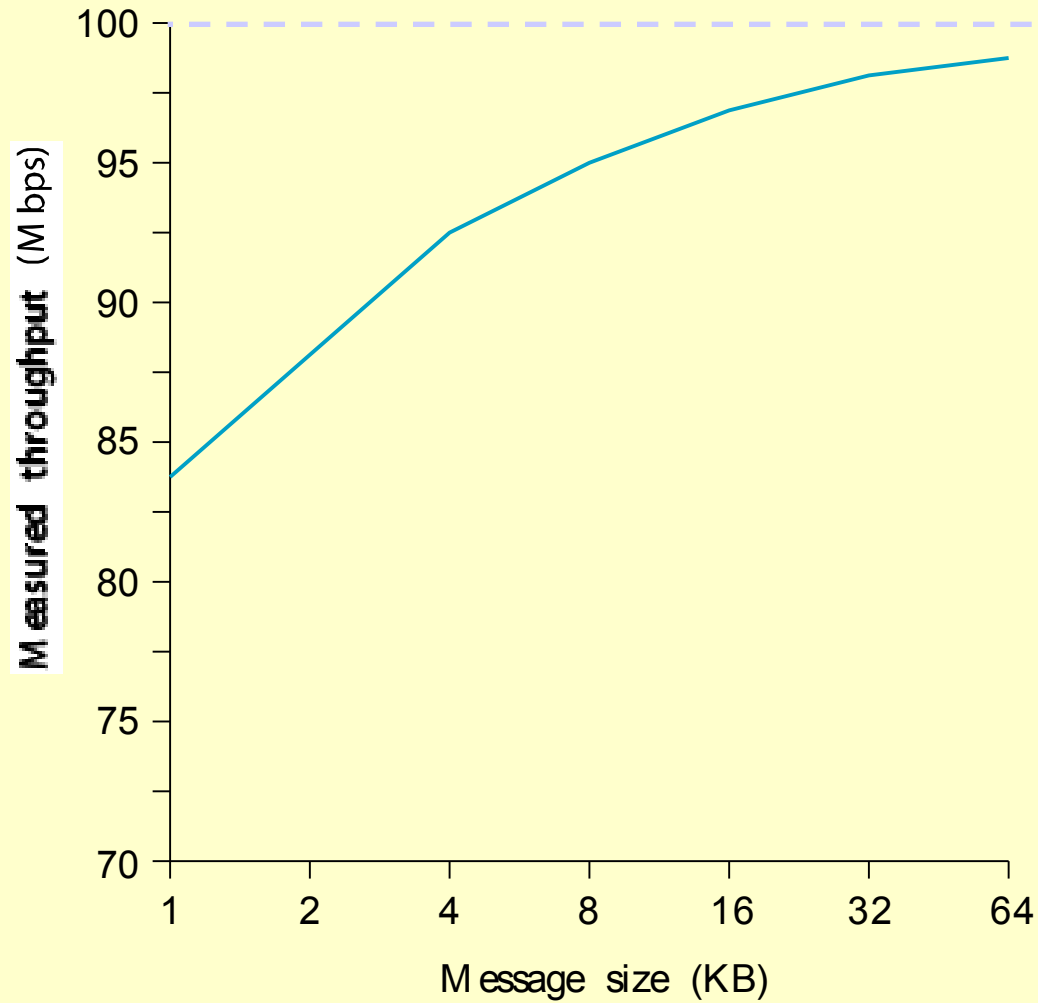
- Dois PCs Pentium 733MHz
- 100-Mbps Ethernet
- Linux 2.4
- Aplicação Ping-pong



Latências de ida e volta

Tamanho das Mensagens (bytes)	UDP	TCP
1	58	66
100	76	84
200	93	104
300	111	124
400	132	136
500	150	159
600	167	176
700	184	194
800	203	210
900	223	228
1000	239	249

Throughput de UDP



Exercício

1. Seja novo um protocolo de fluxo de bytes confiável que use a janela deslizante (como TCP). Esse protocolo será implementado em uma rede de 100 Mbps. O RTT da rede é de 100 ms, e o tempo de vida máximo do segmento (MSL) é de 60 segundos.

a) Quantos bits são necessários nos campos JanelaAnunciada e NúmeroSeq do cabeçalho desse novo protocolo?