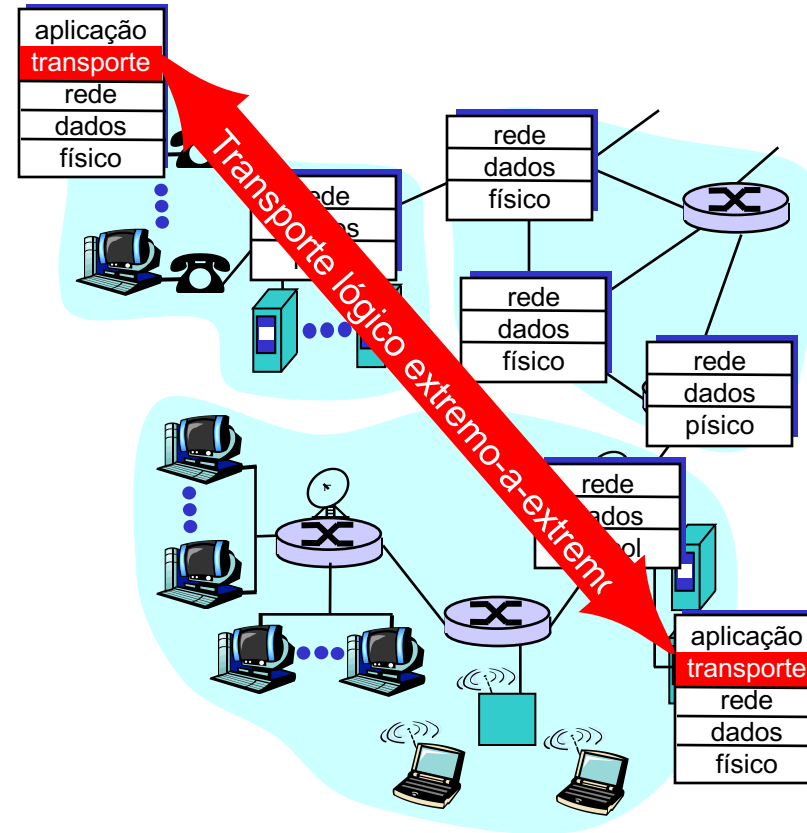


Camada de transporte

- ❑ Serviços fornecidos pela camada de transporte
- ❑ Camada de transporte não-orientada à sessão
 - O caso UDP
- ❑ Controlo de erros: canal fiável e sequencial
 - O caso TCP
- ❑ Controlo de fluxo
 - O caso TCP
- ❑ Congestionamento e seu controlo
 - O caso TCP
- ❑ Máquinas de estado

Serviços

- ❑ Proporciona comunicação lógica entre processos de aplicação
- ❑ Suportado na camada de rede
- ❑ Melhora os serviços da camada de rede
- ❑ Camada extremo-a-extremo



Serviços da camada de transporte na Internet

- ❑ Entrega fiável e ordenada para unicast: TCP
 - Controlo de erros
 - Controlo de fluxo
 - Controlo de congestão
- ❑ Entrega não-fiável para unicast ou multicast: UDP
 - Detecção de erros
- ❑ Segurança: Transport Layer Security (TLS) sobre TCP
 - Autenticação
 - Confidencialidade
 - Integridade
- ❑ Serviços indisponíveis
 - Limites nos atrasos
 - Garantias de débito
 - Multicast fiável

Portos

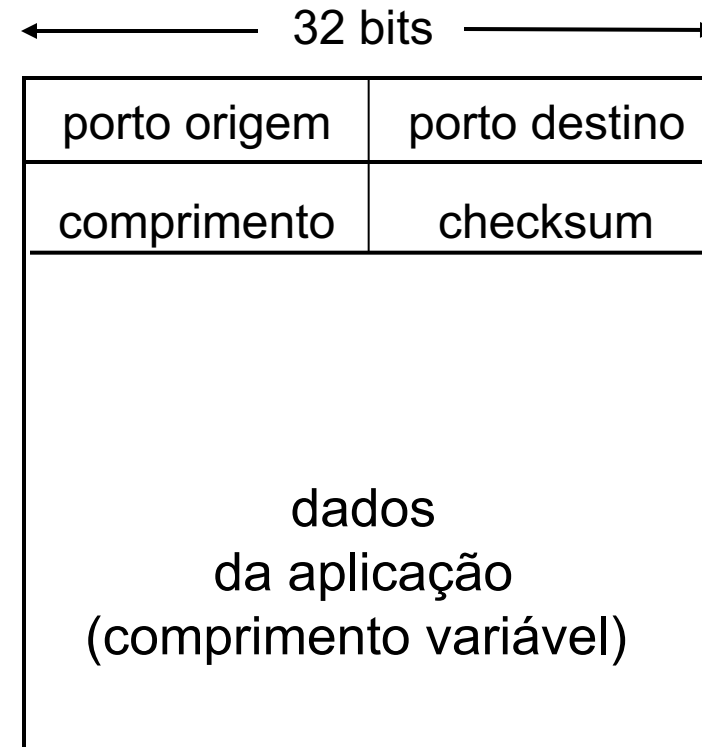
- ❑ Porto origem e porto destino enviados em cada segmento
- ❑ Portos bem-conhecidos e portos efêmeros



Formato dos segmentos
TCP/UDP

Anatomia de um segmento UDP

- ❑ Portos
- ❑ Comprimento
 - Dados + cabeçalho
- ❑ Soma de verificação
 - $cksum := x0000$
 - $segsum :=$ soma, em complemento para 1, de todas as palavras de 16 bits do segmento*
 - $cksum :=$ complemento para 1 de $segsum$
 - $cksum + segsum = xffff$



*também inclui pseudo-cabeçalho com endereços IP

Checksum: algebra

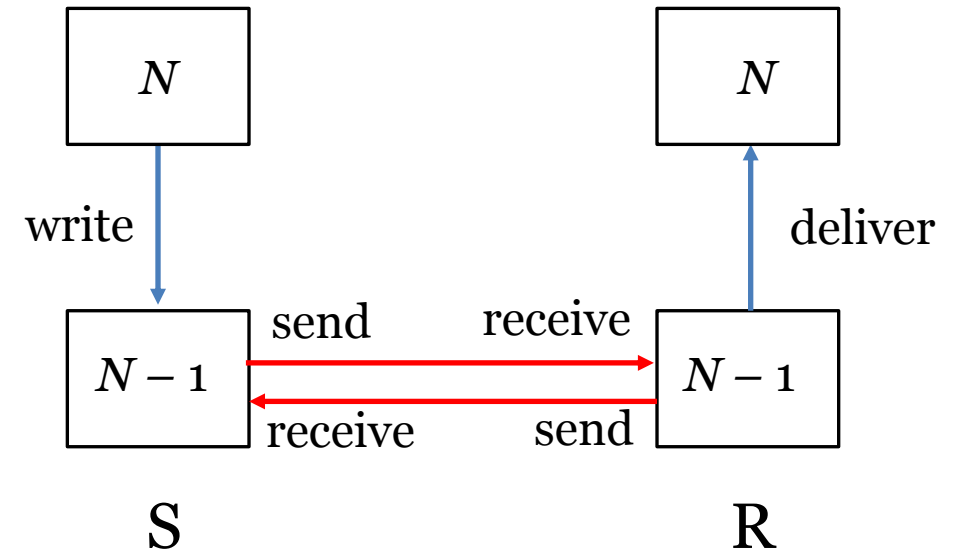
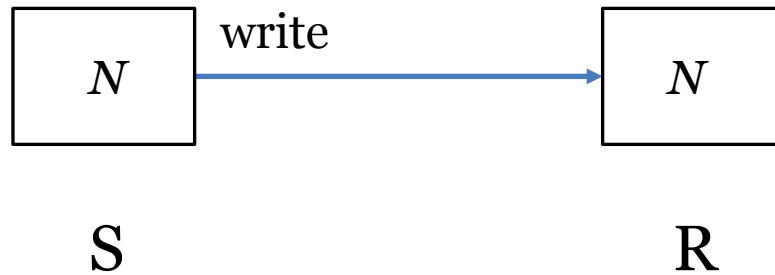
$$x \oplus y = \begin{cases} x + y, & x + y < 2^n \\ x + y - 2^n + 1, & x + y \geq 2^n \end{cases}$$

Operação \oplus é comutativa e associativa

Soma de verificação $\overline{x_1 \oplus \cdots \oplus x_k} = 2^n - 1 - x_1 \oplus \cdots \oplus x_k$

Complementaridade $\bar{x} \oplus x = 2^n - 1$

Controlo de erros: canal fiável e sequencial



—→ fiável e sequencial

—→ não-fiável, sequencial (camada de ligação-de-dados)
ou não-sequencial (camada de transporte)

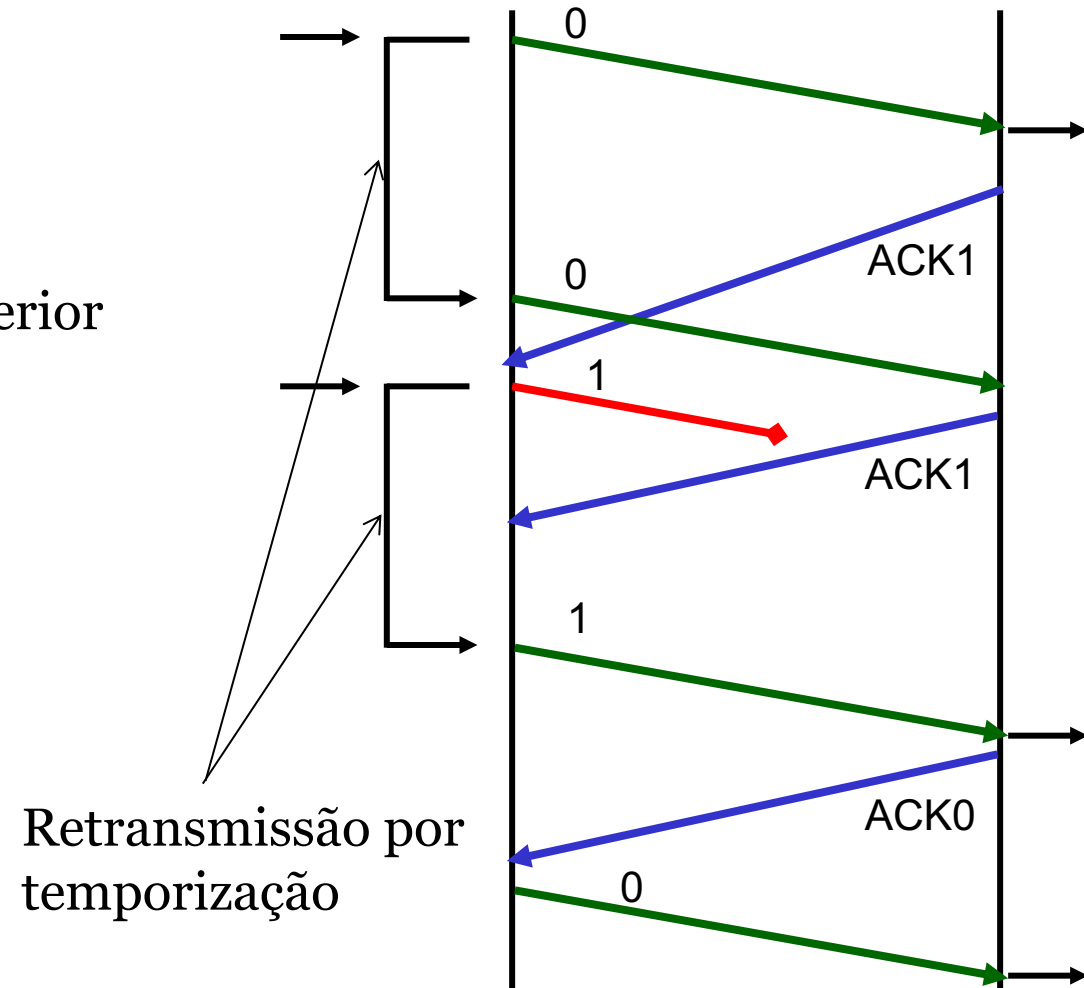
Pára-arranca: não-fiável; sequencial

❑ Pára-arranca (*stop-and-wait*)

- Só transmite um pacote depois de confirmada a receção do pacote anterior

❑ Elementos necessários

- Retransmissão por temporizador
- Canal de retorno: ACKs
- Numeração dos pacotes módulo 2
- Numeração dos ACKs módulo 2



Pára-arranca: não-fiável; não-sequencial

❑ Pára-arranca (*stop-and-wait*)

- Só transmite um pacote depois de confirmada a receção do pacote anterior

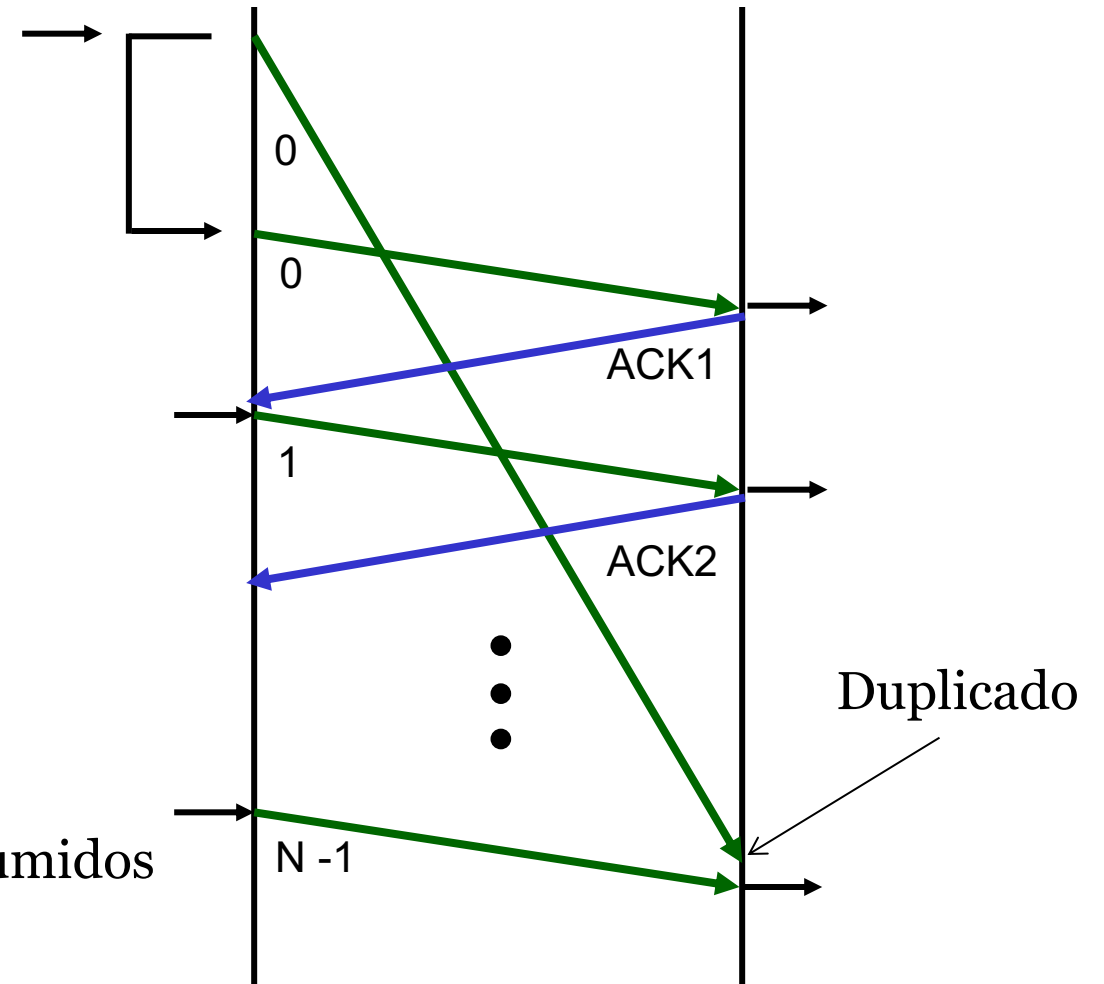
❑ Elementos necessários

- Retransmissão por temporizador
- Canal de retorno: ACKs
- **Numeração dos pacotes módulo $r \times T$**
- **Numeração dos ACKs módulo $r \times T$**

T – tempo máximo de vida de um pacote no canal

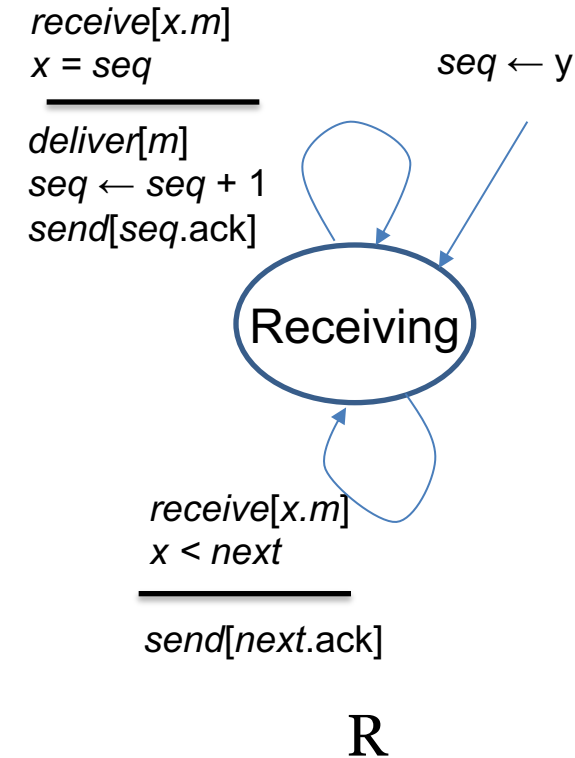
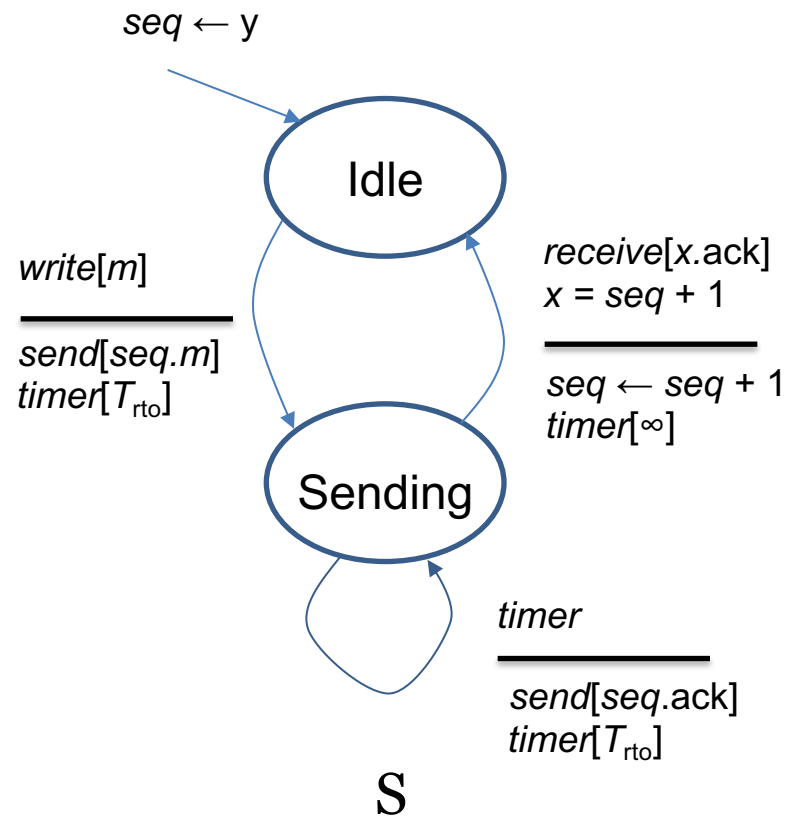
r – débito a que os números de sequência são consumidos

Camada de transporte



Para-arranca: máquinas de estados

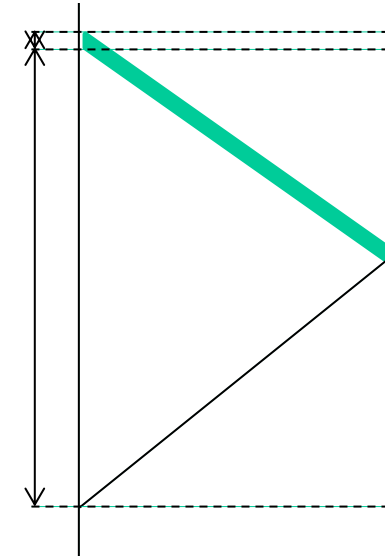
$next$ entre 0 e $r \times T$, no qual r é a taxa máxima de consumo de números de sequência e T é o tempo máximo de vida de um pacote na rede



Pára-arranca: produto largura-de-banda-atraso

Pacotes com 1000 bytes

- ❑ Camada de transporte (Internet)
 - Capacidade 1 Gbit/s e RTT 30 ms
 - Produto largura-de-banda-atraso, $c \times \text{RTT}$, 30 Mbits
 - Eficiência 0,03%!!!
- ❑ Camada de ligação-de-dados Wi-Fi
 - Capacidade 100 Mbit/s e RTT 10 μs
 - Produto largura-de-banda-atraso, $c \times \text{RTT}$, 1 Kbits
 - Eficiência 88,89%



Protocolo pode limitar a utilização do canal

Controlo de erros: eficiência

❑ Transações pára-arranca em paralelo

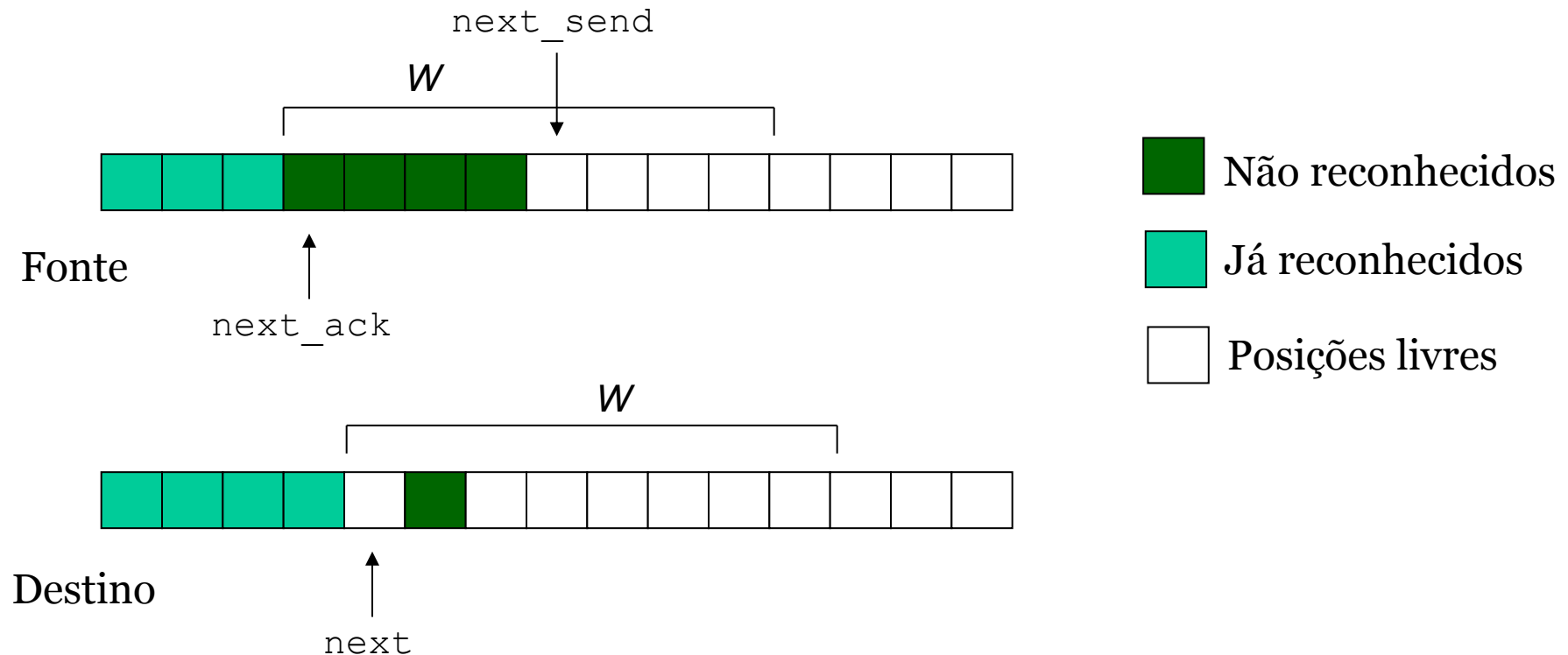
- Campo no cabeçalho para numerar as transações
- Ausência de bloqueio topo-da-fila
- Posterior ordenação dos pacotes

❑ Janela deslizante (canalização; *pipelining*)

- É preciso também uma janela do lado da receção?
- Os ACKs são cumulativos ou seletivos?
- Temporizador por janela ou por pacote?
- Quando é que os pacotes são retransmitidos?

Controlo de erros: janelas deslizantes

- ❑ Fonte e destino mantêm uma janela deslizante de dimensão W



Ações: fonte

- ❑ $\text{write}[m]$ e $\text{next_send} - \text{next_ack} < W$
 - guarda $\text{next_send}.m$ na janela
 - $\text{send}[\text{next_send}.m]$
 - se $\text{next_send} = \text{next_ack}$, então ativa timer
 - $\text{next_send} := \text{next_send} + 1$
- ❑ timer expira
 - $\text{send}[\text{next_ack}.m]$
 - re-ativa timer
- ❑ $\text{recv}[x.\text{ack}]$ e $x > \text{next_ack}$
 - $\text{next_ack} := x$
 - se $\text{next_send} > \text{next_ack}$, então re-ativa timer
- ❑ $\text{Recv}[x.\text{ack}]$ e $x = \text{next_ack}$
 - Desativa timer

Um temporizador por janela

Ações: destino

❑ `recv[x.m]` e $x = \text{next}$

- guarda `m` na janela na posição `next`
- encontrar o número máximo `y` tal que todos os pacotes entre `next` e `y - 1` estão na janela
- `deliver[m]` a todos os pacotes entre `next` e `y - 1`
- `next := y`
- `send[next.ack]`

❑ `recv[x.m]` e $x > \text{next}$

- guarda `m` na janela na posição `x`
- `send[next.ack]`

❑ `recv[x.m]` e $x < \text{next}$

- `send[next.ack]`

ACKs cumulativos

Condição de transmissão contínua

c – capacidade do canal [bit/s]

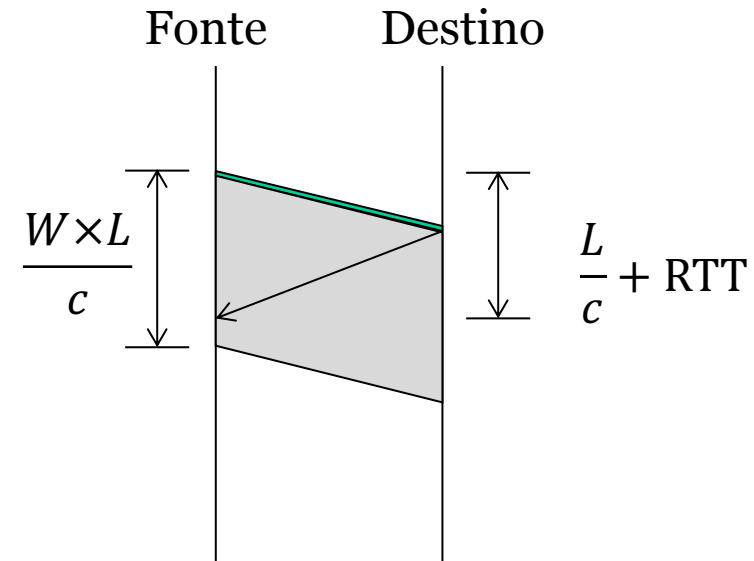
RTT – tempo de ida-e-volta [s]

W – tamanho da janela [nº pacotes]

L – tamanho de um pacote [bits]

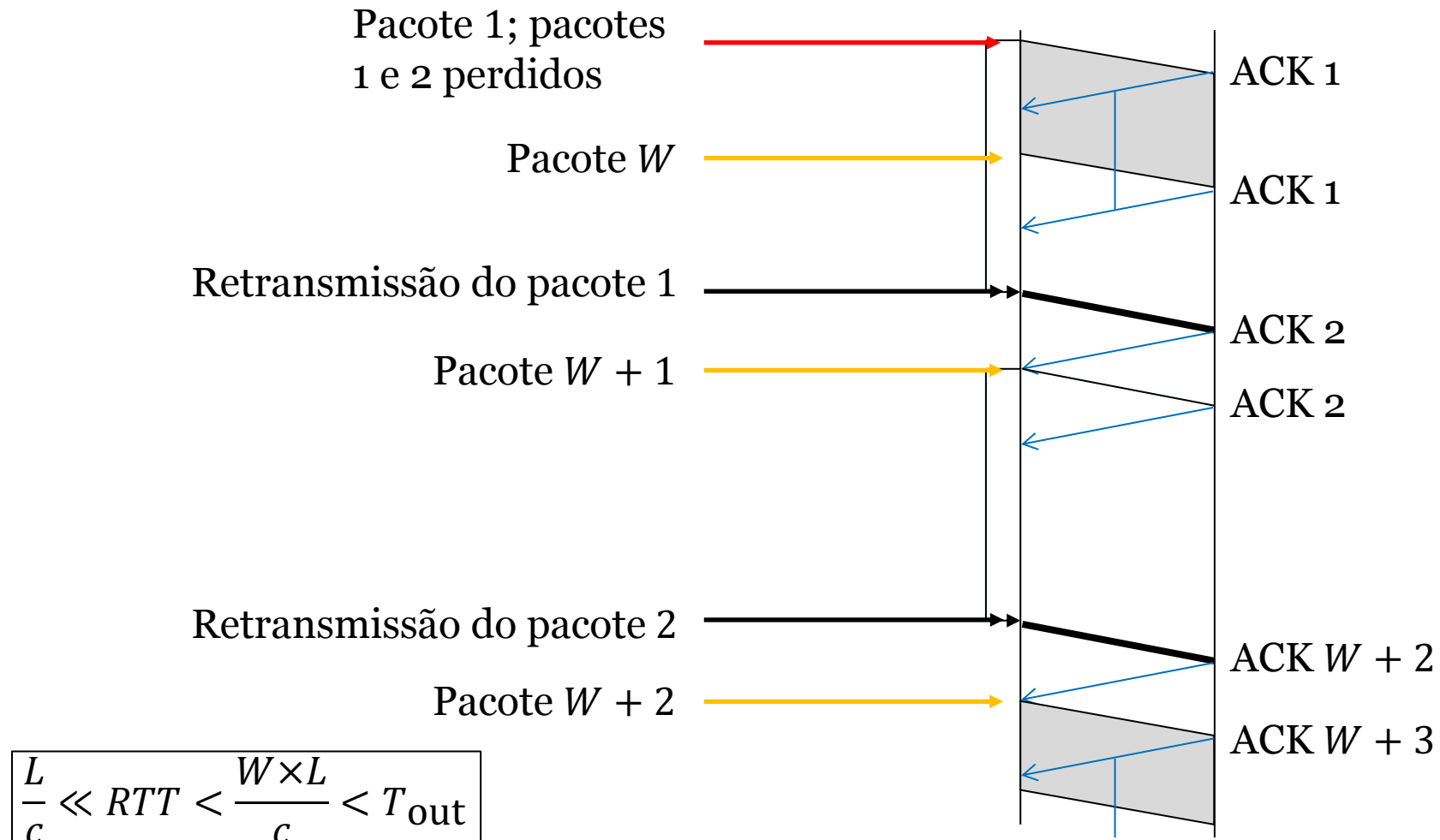
Fonte transmite continuamente se, e só se,

$$W \times L \sim (W - 1) \times L \geq c \times \text{RTT}$$

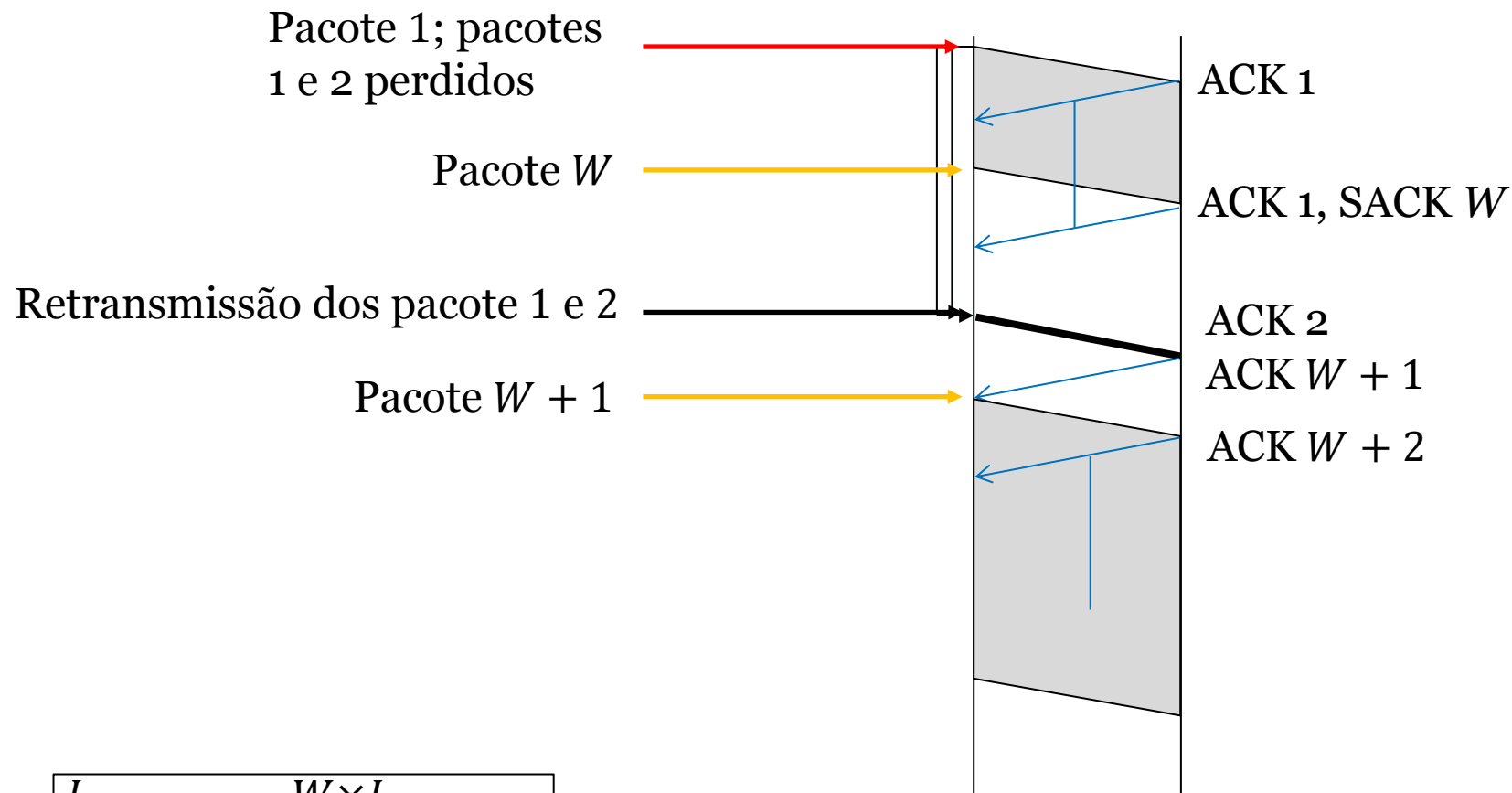


Dimensão da janela não deve ser inferior ao produto largura-de-banda-atraso

Temporizador/janela e ACKs cumulativos

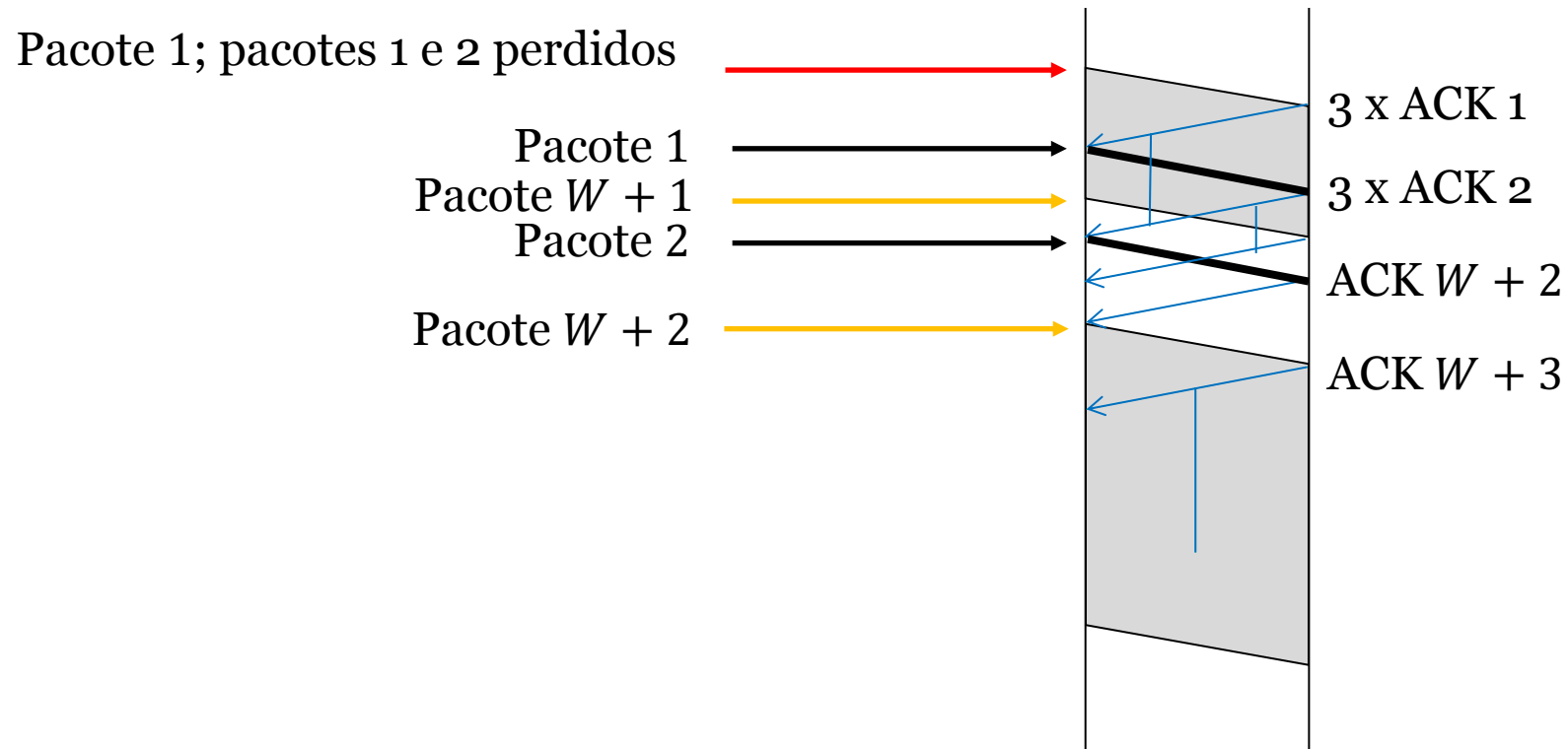


Temporizador/pacote e ACKs seletivos



$$\frac{L}{c} \ll RTT < \frac{W \times L}{c} < T_{out}$$

Retransmissão rápida: 3 ACKs duplicados



$$\frac{L}{c} \ll RTT < \frac{W \times L}{c} < T_{out}$$

RTT e tempo de retransmissão

- ❑ Tempo de retransmissão
 - Pequeno comparado com RTT implica retransmissões desnecessárias
 - Grande comparado com RTT implica reação lenta a perdas
- ❑ RTT varia durante uma sessão
- ❑ Estimação do RTT
 - Extração de amostras de RTT
 - *Ignorar retransmissões na extração de amostras*
 - Calcular estimativa do RTT
- ❑ Determinar o tempo de retransmissão em função da estimativa de RTT

Estimativa de RTT e tempo de retransmissão

- ❑ Média exponencial variante no tempo
 - Impacto de amostras antigas decresce exponencialmente

$$\text{EstimatedRTT} = (1-x) * \text{EstimatedRTT} + x * \text{SampleRTT}$$

$$\text{Deviation} = (1-x) * \text{Deviation} + x * |\text{SampleRTT} - \text{EstimatedRTT}|$$

$$\text{Timeout} = \text{EstimatedRTT} + 4 * \text{Deviation}$$

Tipicamente $x = 0.125$ (1/8)

Estabelecimento e terminação de uma sessão

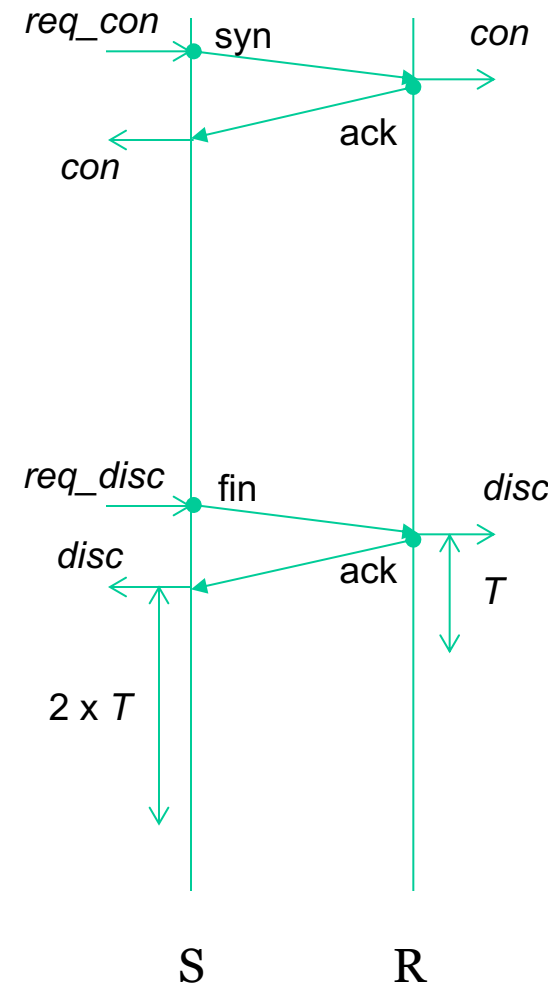
❑ Estabelecimento da sessão

- syn e ack

❑ Terminação da sessão

- fin e ack
- Espera $2 \times T$ por pacotes de terminação

T – tempo máximo de vida de um pacote no canal

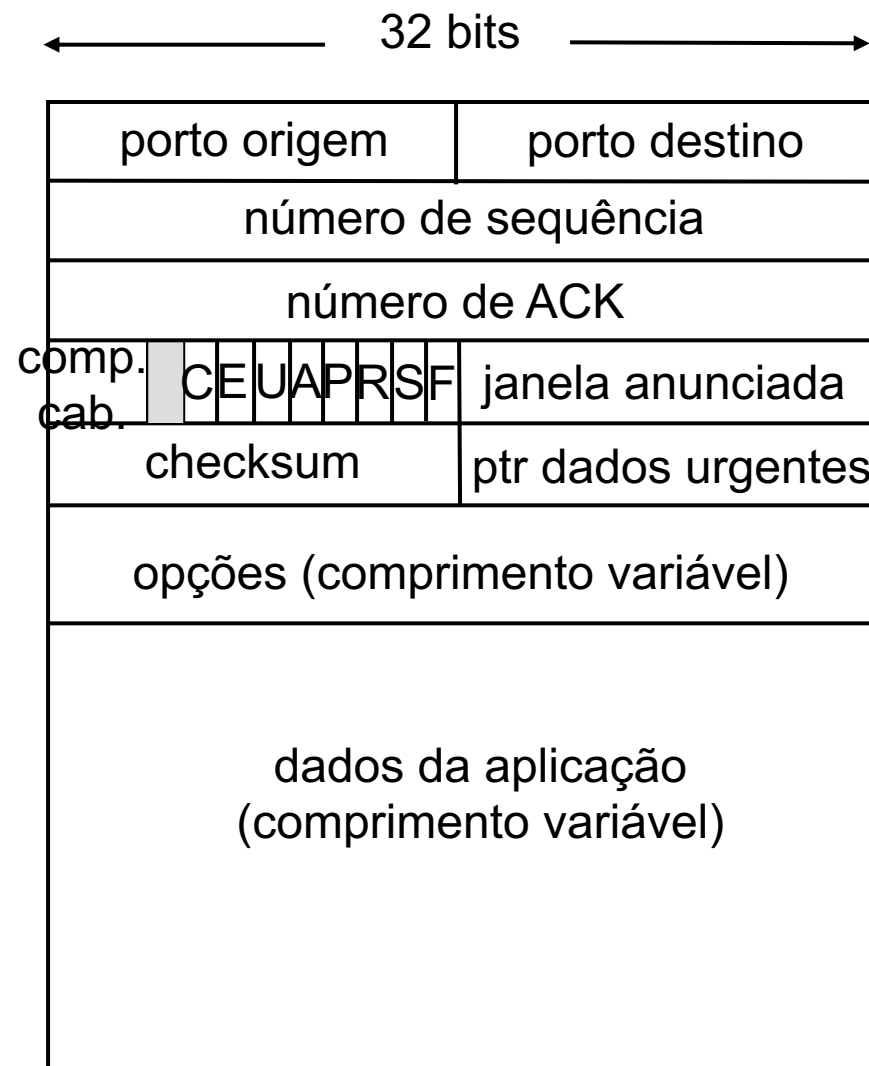


Transmission Control Protocol (TCP)

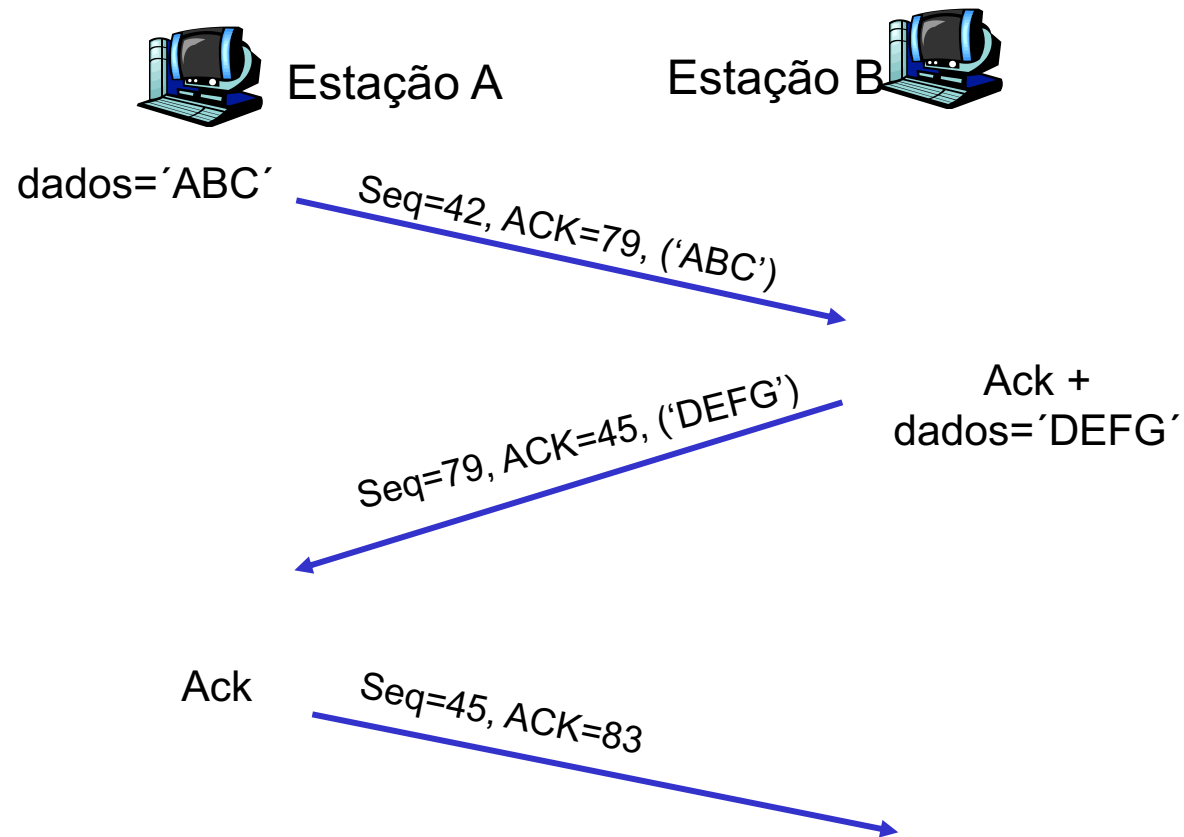
- ❑ Fornece um serviço de sequência de bytes ordenada e fiável
- ❑ Sessões ponto-a-ponto bi-direcionais
- ❑ Protocolo de janela deslizante
 - Dimensão da janela é dinâmica em função do controlo de fluxo e de congestionamento

Anatomia de um segmento TCP

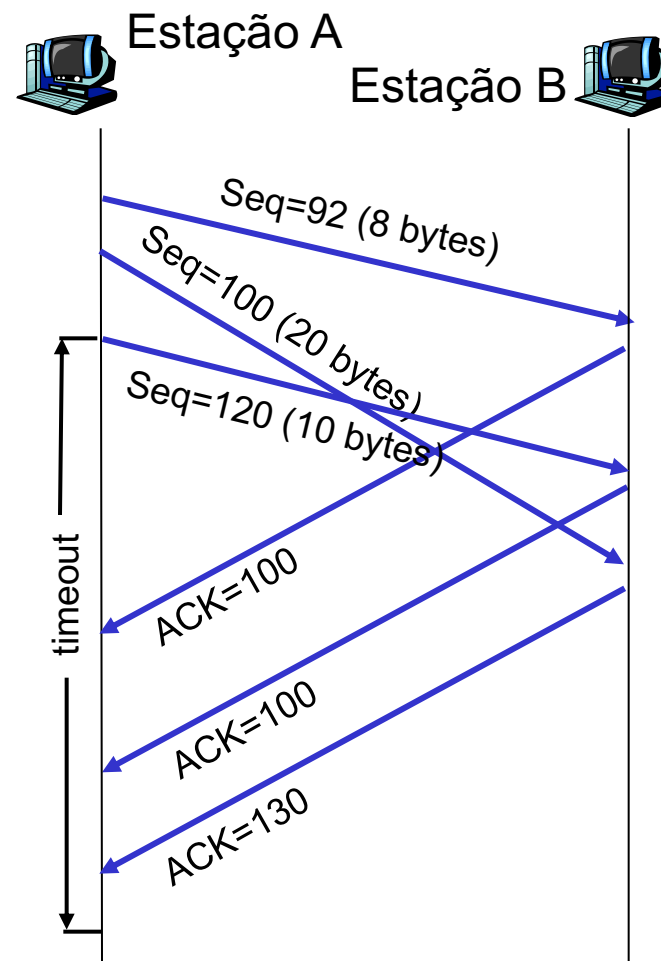
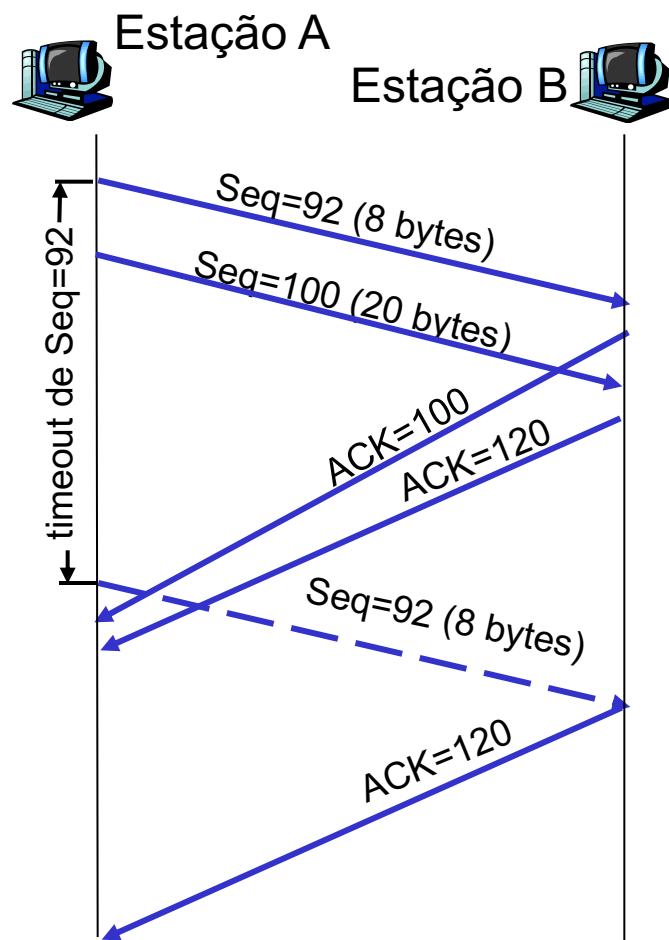
- ❑ Números de sequência e de ACKs
 - Cada byte tem um número de sequência
- ❑ Sinalizadores
 - SYN e FIN: estabelecimento e terminação de sessão
 - RST: rejeição de sessão
 - ACK: número de sequência é válido
 - PSH: dados devem ser passados à aplicação
 - ECE: indicação de congestionamento
 - CWR: confirmação de reação ao congestionamento
 - URG: ptr dados urgentes é válido
- ❑ Ptr dados urgentes
 - Deslocamento para o fim dos dados urgentes



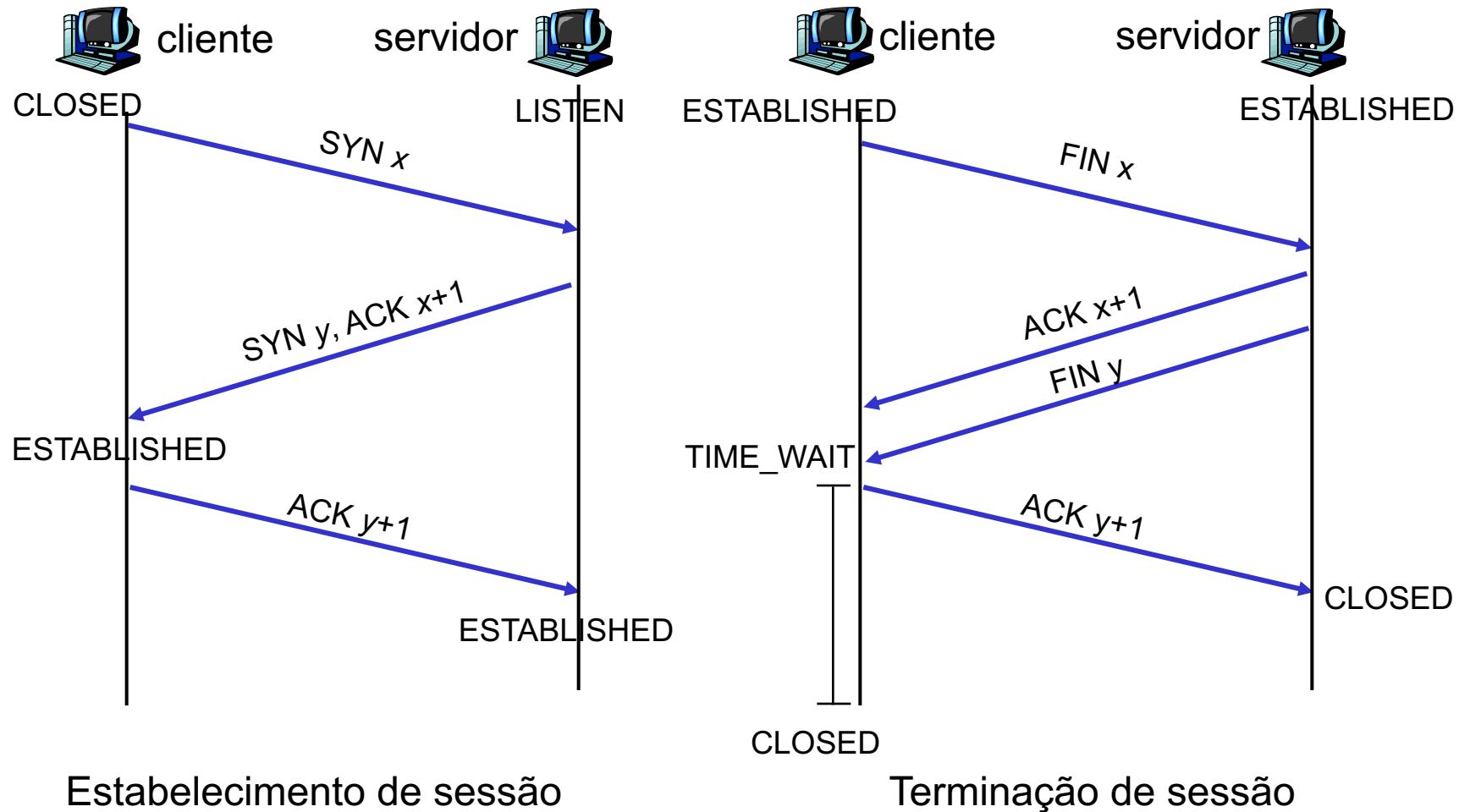
Números de sequência e de ACK



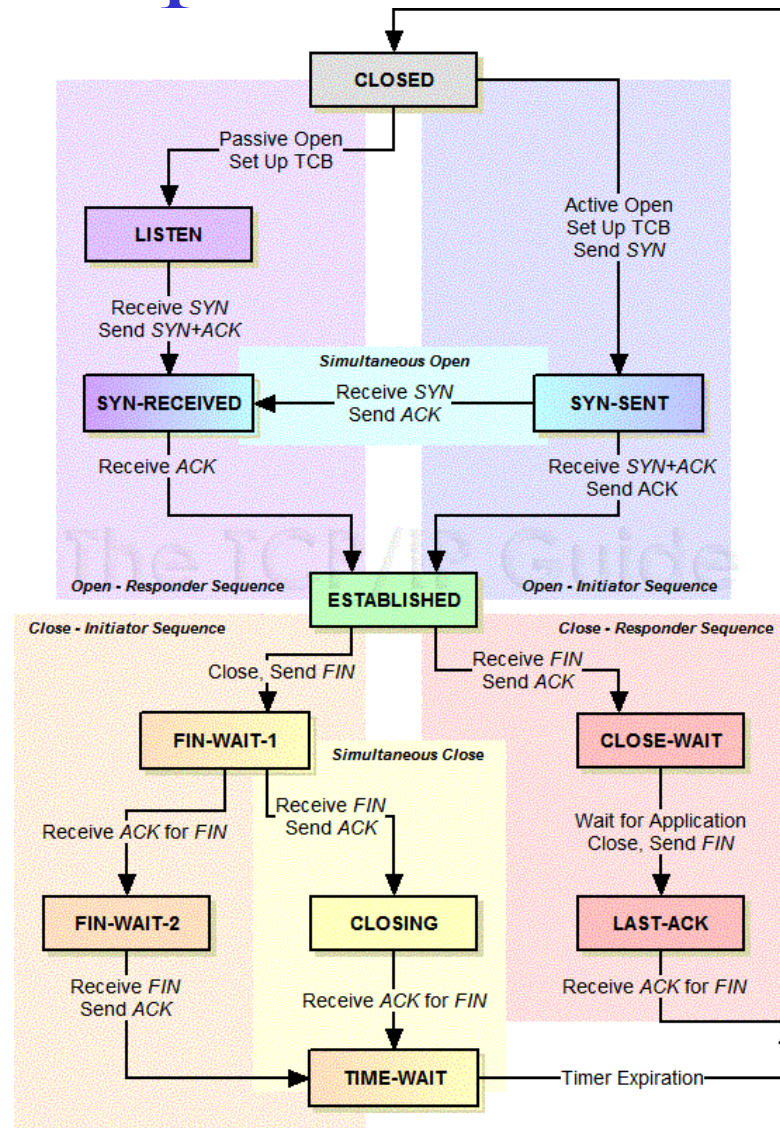
Transferência de dados: exemplos



Estabelecimento e terminação de sessão



Maquina de estados



Envio de segmentos

❑ Segmento de dados novo

- Número de bytes no buffer superior ou igual a *Maximum Segment Size* (MSS)
- Periodicamente, sempre que houver bytes no buffer
- Sempre que bytes são escritos pela aplicação

❑ Retransmissão de um segmento de dados

- Temporizador expirou
- Receção de três ACKs duplicados (retransmissão rápida)

❑ ACK retardado

- Algum tempo depois da chegada de um segmento esperado

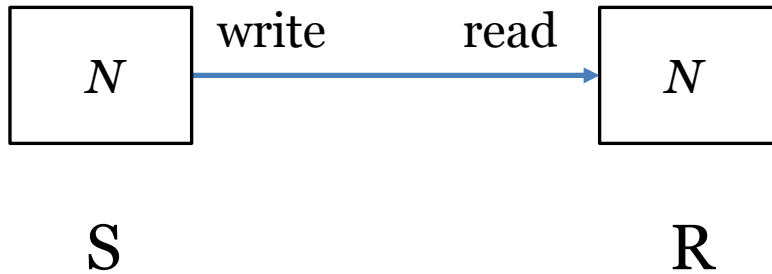
❑ ACK cumulativo

- Chegada de um segmento esperado e o ACK do segmento anterior ainda não tinha sido enviado

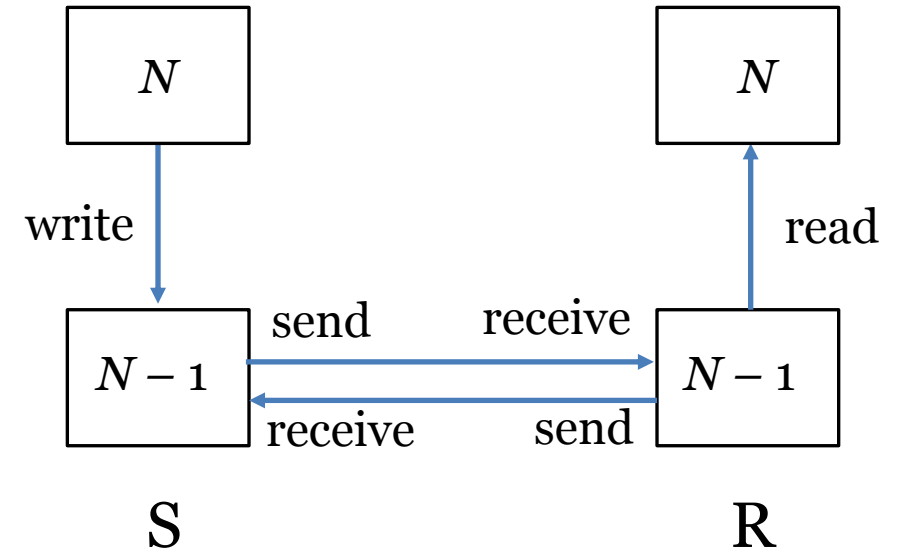
❑ ACK duplicado

- Chegada de um segmento duplicado
- Chegada de um segmento fora de ordem

Controlo de fluxo



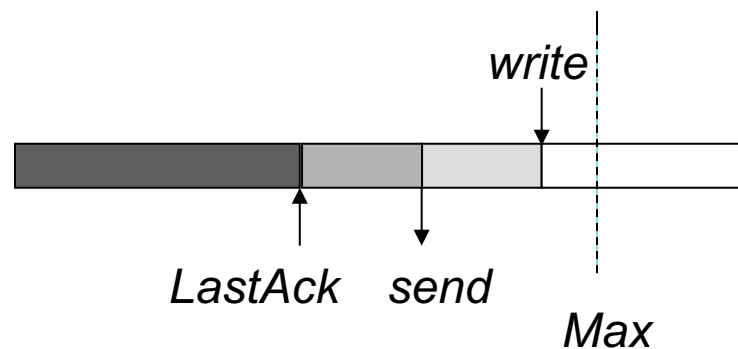
taxa de write pode ser superior à taxa de read



ajuste da taxa de write à taxa de read

Controlo de fluxo: janela anunciada

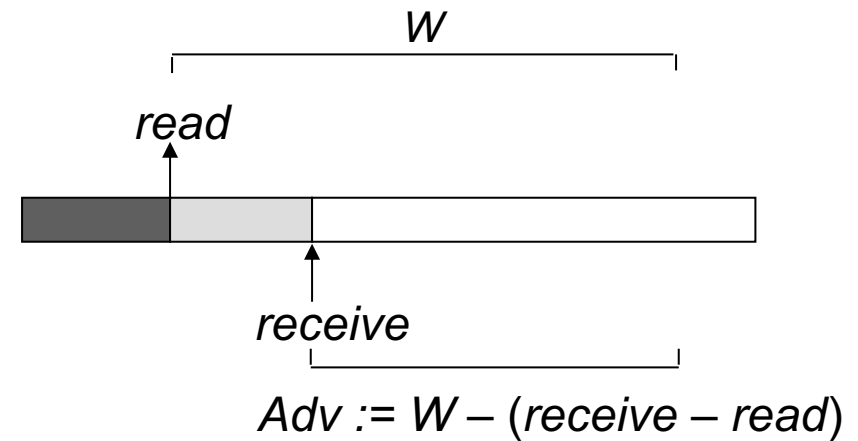
- ❑ Destino anuncia à fonte o espaço disponível na sua janela de dimensão W
- ❑ Anúncios ocorrem quando destino reconhece bytes vindos da fonte



Recebe: $receive'$ e Adv'

Calcula: $LastAck := receive'$
 $Max := Adv' + receive' = read' + W$

S



Anuncia: $receive$ e Adv

R

Dois problemas e suas soluções

- ❑ A janela anunciada é zero
 - A fonte envia um segmento só com um byte para provocar um novo anúncio por parte do destino

- ❑ A fonte só está autorizada a enviar segmentos com muito poucos bytes (síndrome das janelas disparatadas)
 - A fonte espera até ter um número mínimo de bytes para transmitir
 - O destino retém o anúncio do espaço disponível até que tenha um número mínimo de bytes

Controlo de fluxo: débito de leitura constante

- Fonte tem sempre dados para transmitir
$$\begin{aligned}write(t + r2s) &= Max(t + r2s) \\ &= read(t) + W\end{aligned}$$

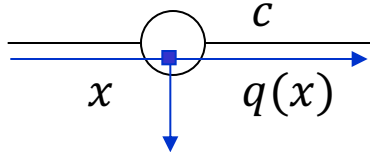
- Destino lê dados a débito constante r
$$\begin{aligned}read(t) + r \times RTT &= read(t + RTT) \\ &\leq receive(t + RTT) \\ &= send(t + r2s)\end{aligned}$$

$r2s$ – Tempo do destino à fonte

- Canal tem capacidade $c, c \geq r$
$$send(t + r2s) = write(t + r2s)$$

Dimensão da janela (W) deverá ser maior ou igual ao produto largura-de-banda-atraso ($r \times RTT$)

Modelo de perdas de uma fila FIFO

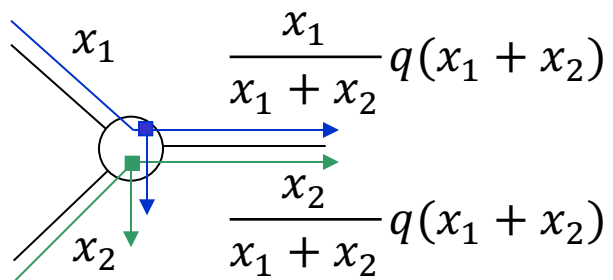
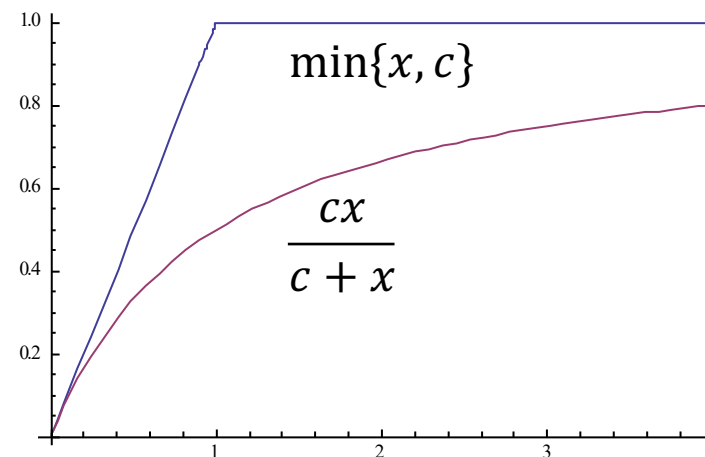


$$q(x) = \min(x, c)$$

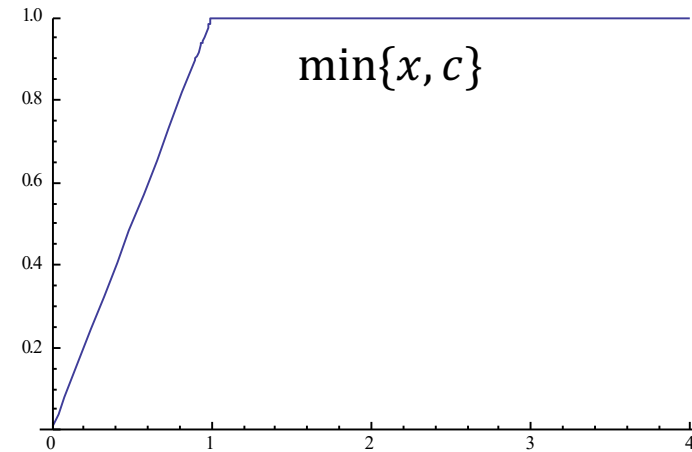
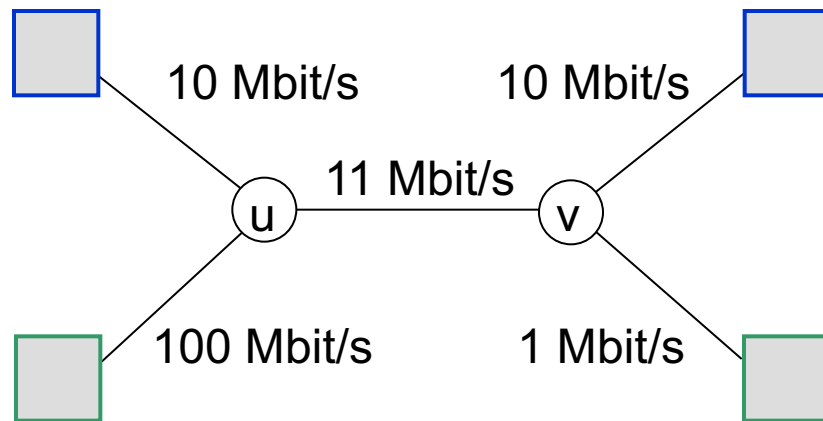
$q(x)$ é crescente

$$\lim_{x \rightarrow 0} \frac{q(x)}{x} = 1$$

$$\lim_{x \rightarrow +\infty} q(x) = c$$



Ausência de retroação: congestionamento

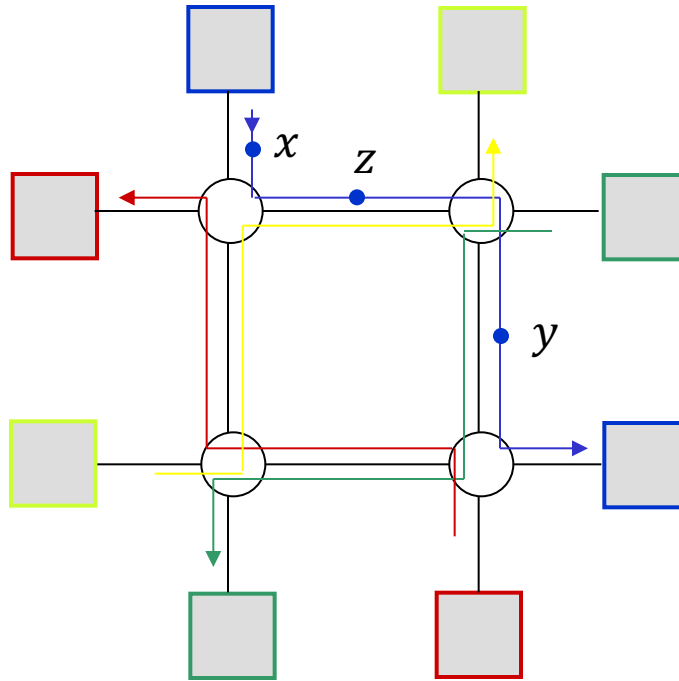


Débito de saída do fluxo verde é 1 Mbit/s

Débito de saída do fluxo azul é 1 Mbit/s!

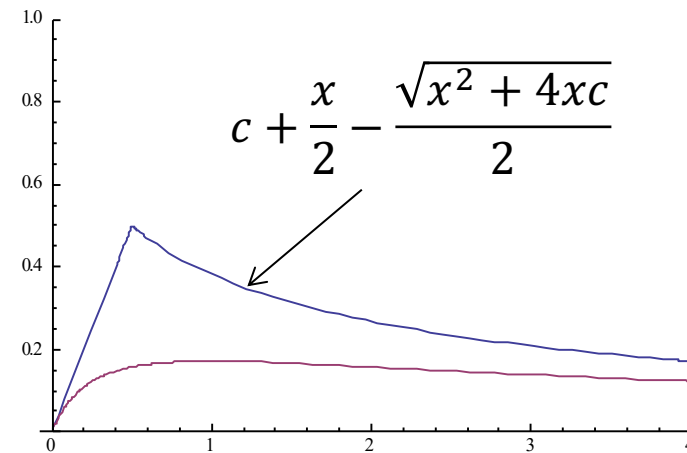
Tráfego verde é transportado na ligação de 11 Mbit/s para ser descartado mais tarde!

Colapso do congestionamento



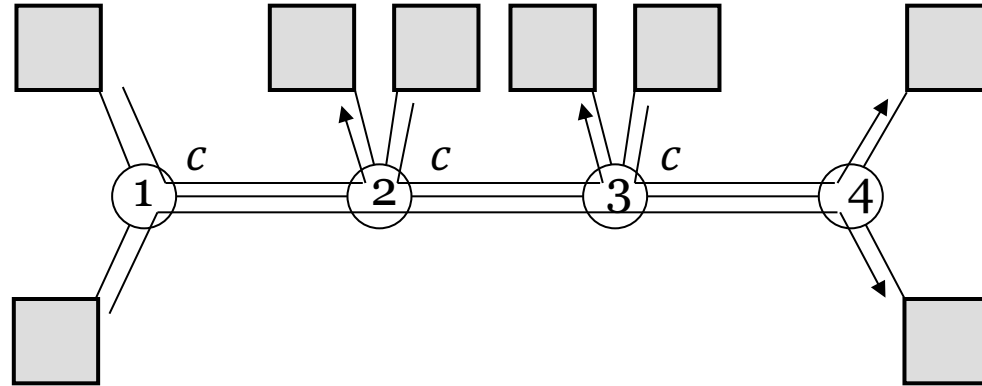
$$z = \frac{x}{x+z} q(x+z)$$

$$y = \frac{z}{x+z} q(x+z)$$



Para cargas elevadas, maioria do tráfego que atravessa 1ª ligação de um caminho é descartado na 2ª ligação

Adaptação de débitos



	Curtos (12, 23, 34)	Longo (1234)	Total
Maxima eficiência	c	0	$3c$
Equidade max-min	$c/2$	$c/2$	$2c$
Equidade proporcional	$3/4 c$	$c/4$	$5/2 c$

Eficiência vs. equidade

Controlo de congestão TCP (Reno)

❑ Extremo-a-extremo

- Fontes inferem e reagem ao congestionamento
- Encaminhadores não participam no controlo do congestionamento

❑ Inferência de congestionamento

- Temporizador expirou
- Receção de três ACKs duplicados

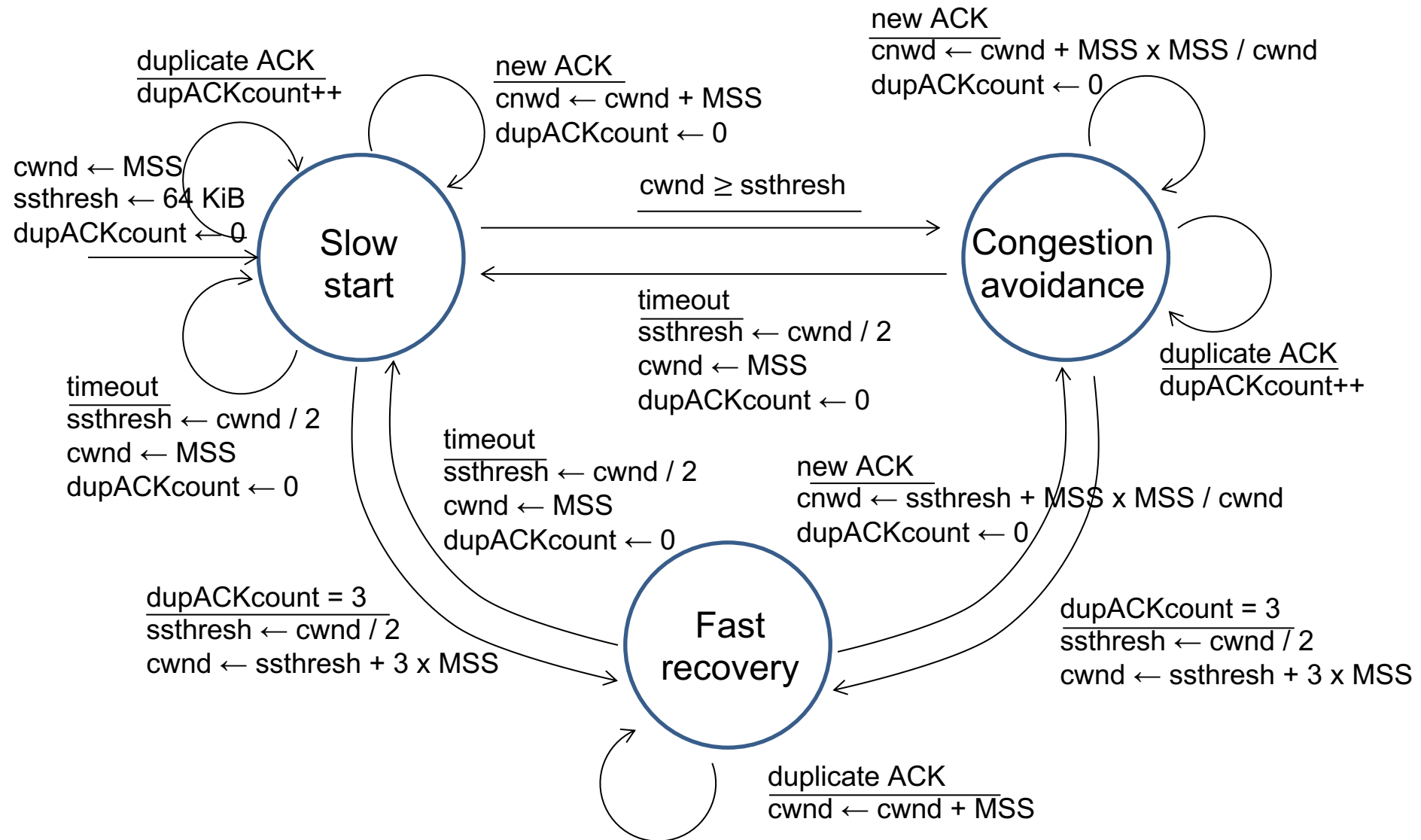
❑ Reação ao congestionamento

- Janela diminui multiplicativamente com o congestionamento (para metade)
- Janela aumenta aditivamente na ausência de congestionamento (um MSS por RTT)

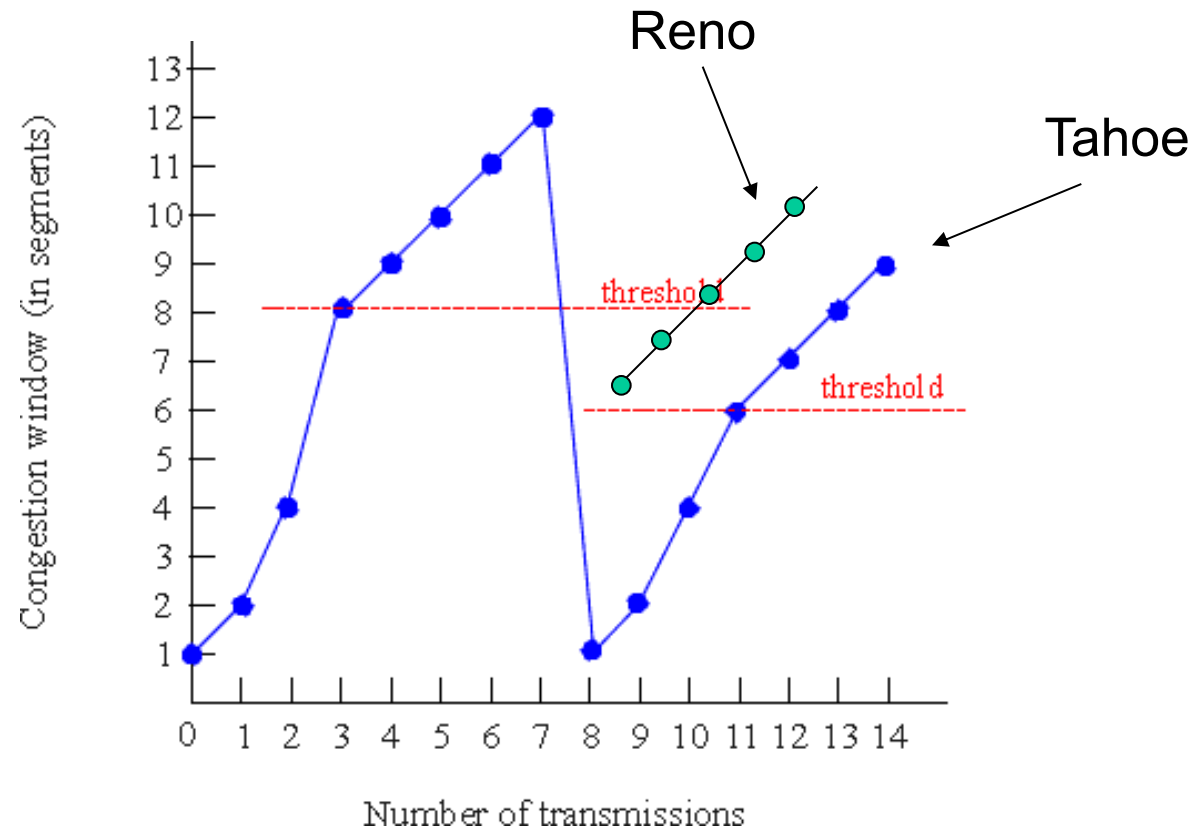
❑ Arranque da sessão

- Janela cresce exponencialmente (um MSS por ACK)

Máquina de estados: fluxograma

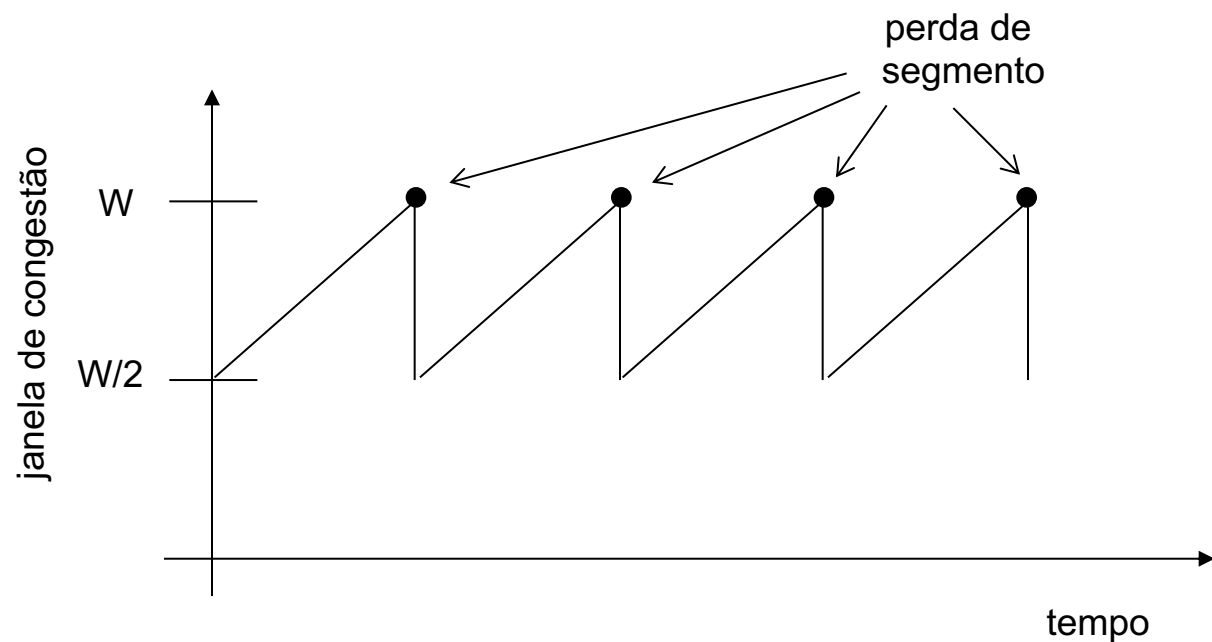


Janela de congestão TCP: exemplo



Crescimento aditivo, decrescimento multiplicativo

Desempenho de uma sessão TCP



$$S = 1.22 \times \frac{8 \times \text{MSS}}{\text{RTT}} \times \sqrt{\frac{1}{p}}$$

S – débito [bit/s]

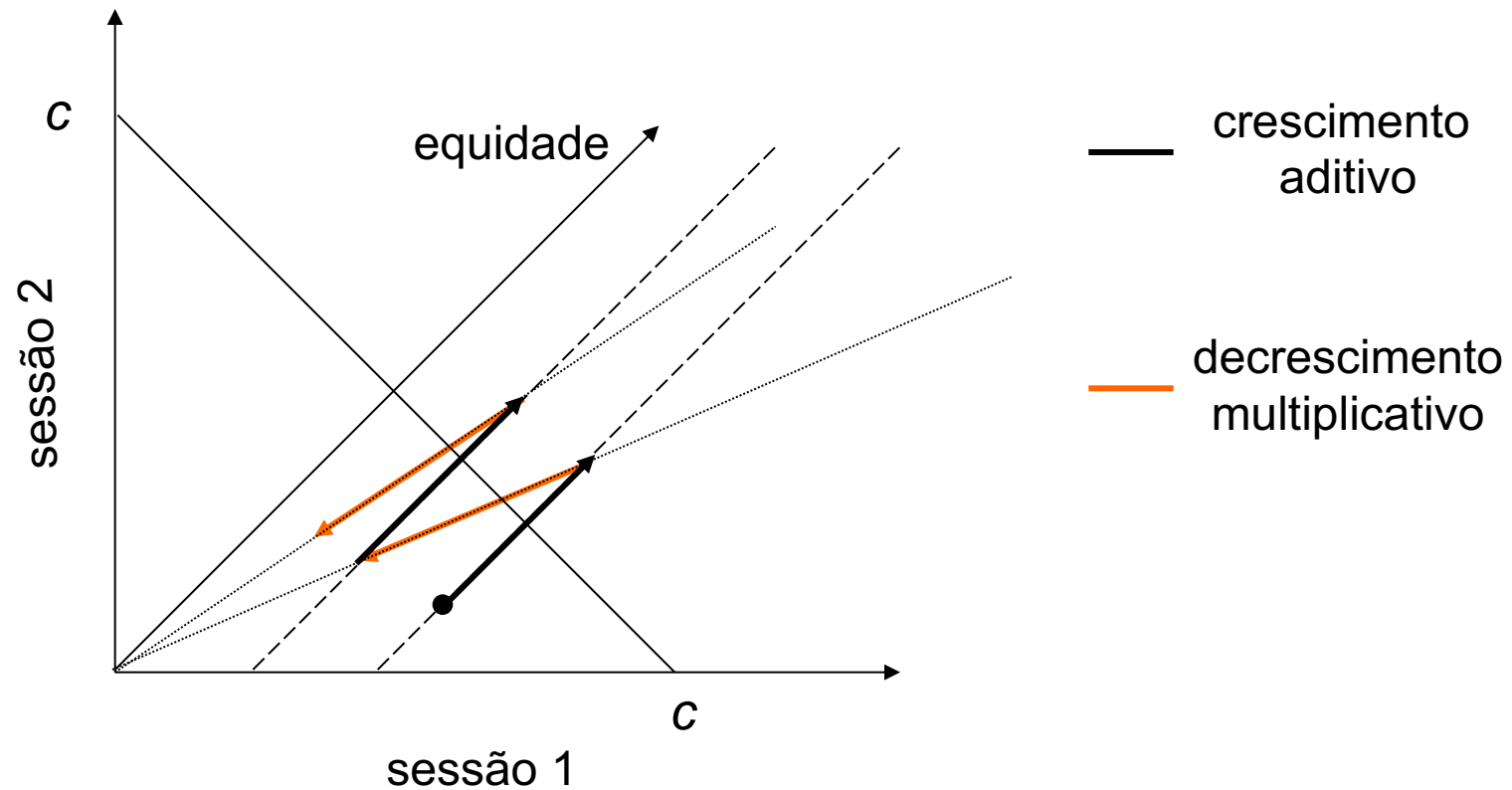
MSS – dimensão de segmento [bytes]

RTT – tempo de ida-e-volta [s]

p – taxa de perda de segmentos

Exemplo: 10 Gbit/s $\rightarrow 2 \times 10^{-10}$

Equidade em sessões TCP com RTT igual



Equidade

- ❑ Equidade entre fluxos com o mesmo RTT
- ❑ Débito de fluxo inversamente proporcional ao RTT

Extensões/alternativas ao TCP

- ❑ Random Early Discard (RED)
 - Encaminhadores descartam pacotes antes do congestionamento propriamente dito
- ❑ Explicit Congestion Notification (ECN)
 - Encaminhadores marcam indicação de congestão, Explicit Congestion Notification
- ❑ TCP Vegas
 - Inferência de congestionamento pelo aumento do RTT
- ❑ CUBIC TCP
 - Crescimento da janela segue uma função cúbica com ponto de inflexão no valor da janela antes do congestionamento