

Controle de Congestionamento em TCP Parte 2

Prof. Dr. S. Motoyama

Controle de Congestionamento em TCP

- Princípios de controle de congestionamento
 - Saber que congestionamento está ocorrendo
 - Adaptar para aliviar o congestionamento
- Controle de congestionamento do TCP
 - Aumento aditivo, decréscimo multiplicativo
 - Início lento e reinício também lento
- Prevenção de congestionamento
 - Detecção aleatória antecipada (Random Early Detection) - RED
 - Notificação explícita de congestionamento (Explicit Congestion Notification) – ECN
 - TCP Vegas

Controle de fluxo vs. Controle de congestionamento

- Controle de fluxo
 - Evitar que um emissor rápido não transborde um receptor lento
- Controle de congestionamento
 - Evitar que um conjunto de emissores não sobrecarregue a rede

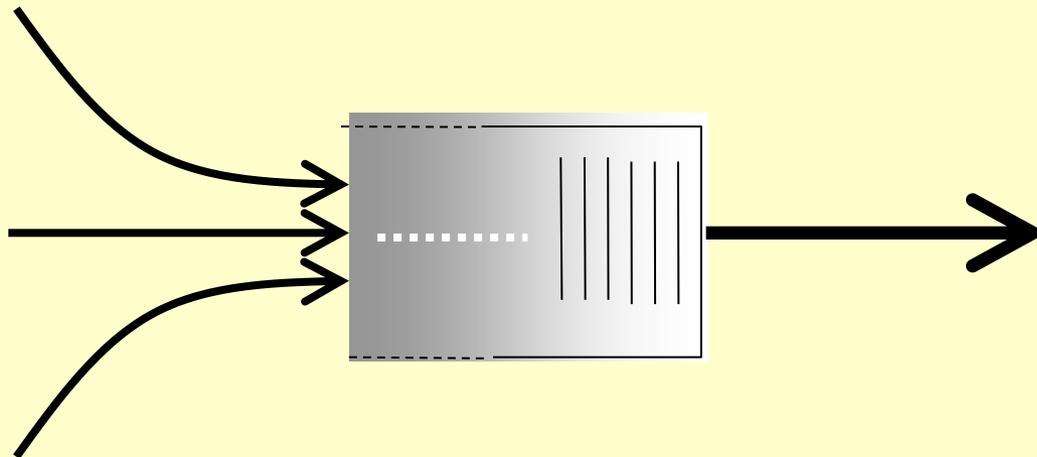
- Conceitos diferentes, mecanismos similares
 - Controle de fluxo do TCP: JanelaAnunciada
 - Controle de congestionamento do TCP: JanelaCongestionamento
 - Janela do TCP: $\min\{JanelaCongestionamento, JanelaAnunciada\}$

Três principais características da Internet

- Comutação de pacote
 - Uma dada origem pode ter uma capacidade suficiente para enviar dados
 - Mas os pacotes podem encontrar o enlace sobrecarregado
- Fluxos de conexões
 - Não há noções de conexões dentro da rede
 - Não há reserva antecipada dos recursos da rede
 - Mesmo assim, os pacotes podem estar relacionados em um grupo (fluxo)
 - Ex., os pacotes em uma mesma transferência TCP
- Serviço de melhor esforço
 - Nenhuma garantia de entrega de pacotes ou atraso
 - Nenhum tratamento preferencial para os pacotes

Congestionamento é inevitável

- Dois pacotes chegam ao mesmo tempo
 - O nó pode transmitir apenas um
 - Ou armazena ou descarta o outro
- Se muitos pacotes chegam em um pequeno intervalo de tempo
 - O nó não consegue tratar todo o tráfego que está chegando
 - E o buffer pode eventualmente transbordar



Colapso de Congestionamento

- Definição: O aumento na carga da rede resulta em um decréscimo do trabalho normal feito.
- Muitas possíveis causas
 - Retransmissões espúrias de pacotes que estão ainda a caminho
 - Colapso clássico de congestionamento
 - Solução: melhores temporizadores e controle de congestionamento do TCP
 - Pacotes não entregues
 - Pacotes consomem recursos e são descartados em alguma parte da rede
 - Solução: controle de congestionamento para todos os tráfegos

Detecção Simples de congestionamento

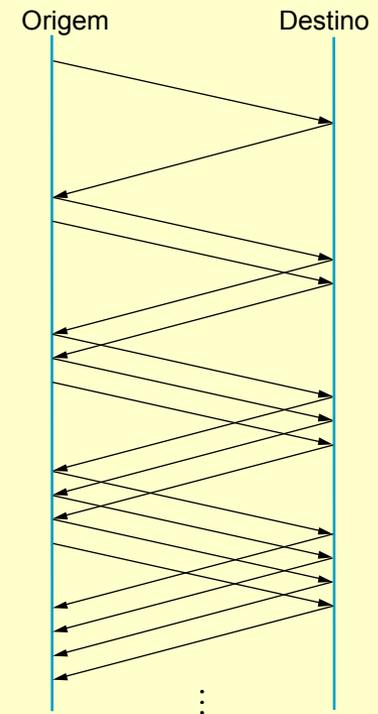
- Perda de pacote
 - Pacote é descartado ao longo do seu caminho ao destino
- Atraso de pacote
 - Pacote é sujeito a um atraso grande
- Como o emissor de TCP sabe disso?
 - Perda
 - Timeout
 - Confirmações duplicadas
 - Atraso
 - Estimação de RTT

Idéia do controle de congestionamento TCP

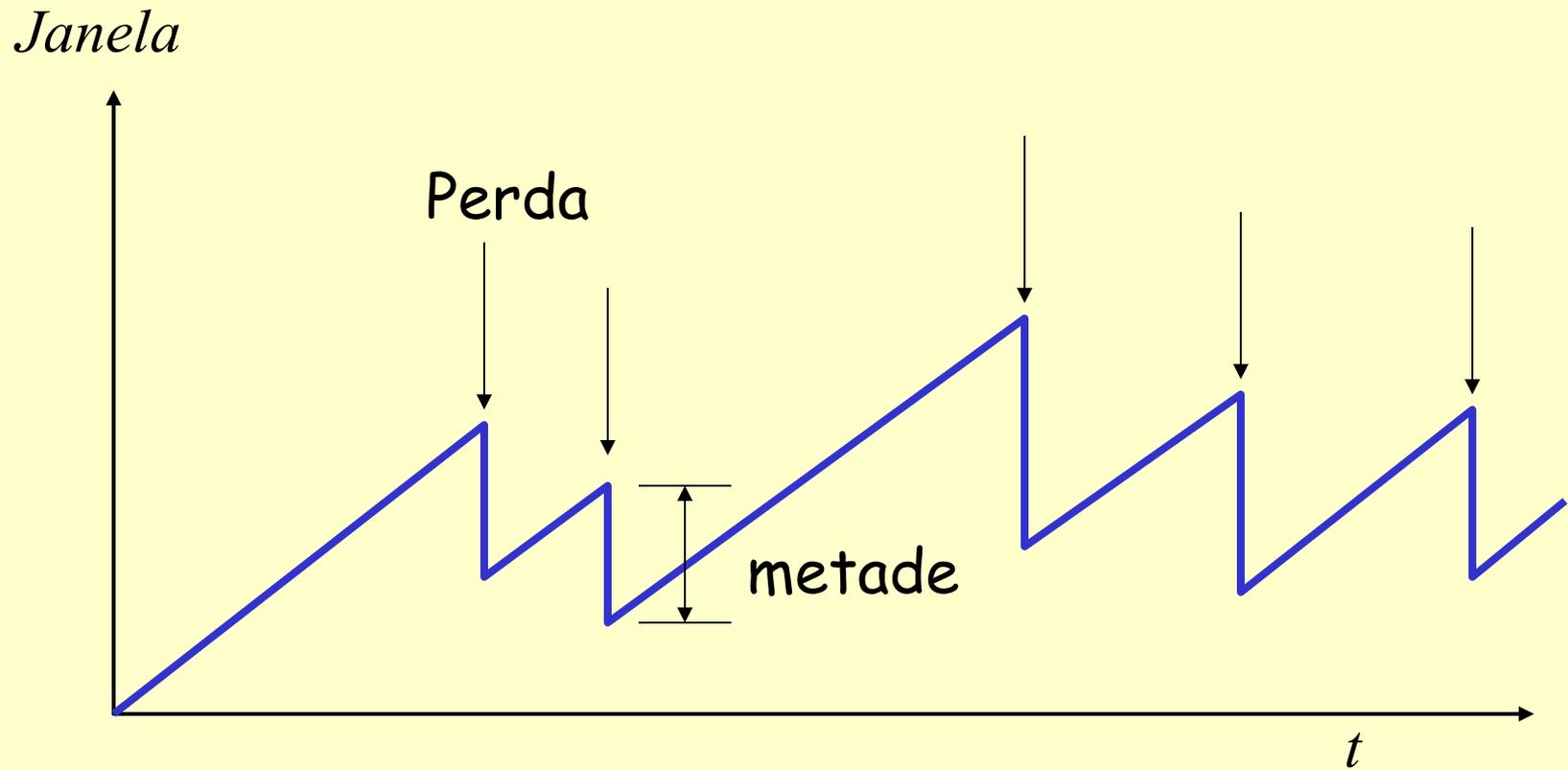
- Cada origem determina a capacidade disponível
 - Assim sabe quantos pacotes tem em trânsito
- Janela de congestionamento – outra variável de estado na origem
 - Máximo número de bytes não confirmados que tem em trânsito
 - $\text{MaxJanela} = \min\{\text{JanelaCongestionamento}, \text{JanelaAnunciada}\}$
 - Enviar a uma taxa de componente mais baixo
- Adaptação da janela de congestionamento
 - Decrescer sob a perda de pacote
 - Aumentar sob sucesso

Aumento aditivo, decréscimo multiplicativo

- Quanto aumentar e quanto diminuir?
 - Aumentar linearmente, diminuir multiplicativamente
 - Conseqüências da janela de tamanho grande são bem piores do que ter uma janela pequena
 - Janelas maiores: muitos pacotes descartados e retransmitidos
 - Janelas menores: throughput menor.
- Decréscimo multiplicativo
 - Na perda de pacote, divide a janela de congestionamento por dois
- Aumento aditivo
 - No sucesso da ultima janela de dados, aumente linearmente



Resulta em “dente de serra” para o TCP



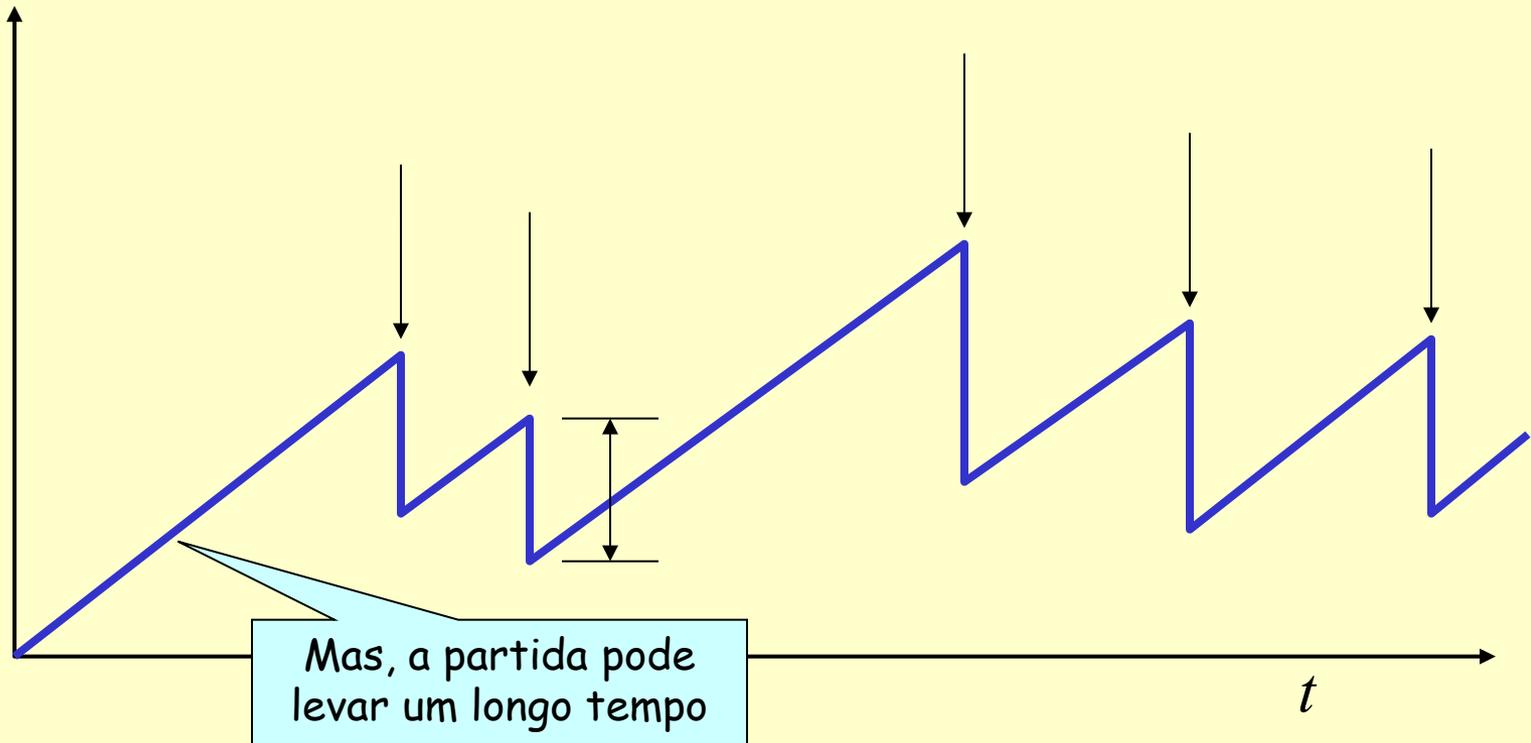
Detalhes Práticos

- Janela de congestionamento
 - Representada em bytes
 - Pacotes tem MSS (Maximum Segment Size) bytes
- Aumento da janela de congestionamento
 - Aumente por MSS no sucesso da ultima janela de dados
 - Na prática, aumente uma fração de MSS por ACK recebido
 - Número de pacotes por janela: $\text{JanelaCongestionamento} / \text{MSS}$
 - Incremento por ACK: $\text{MSS} * (\text{MSS} / \text{JanelaCongestionamento})$
- Decréscimo da janela de congestionamento
 - Nunca deixe a janela de congestionamento abaixo de 1 MSS

Na Partida

Começar com uma pequena Janela
Congestionamento para evitar a
sobrecarga da rede.

Janela

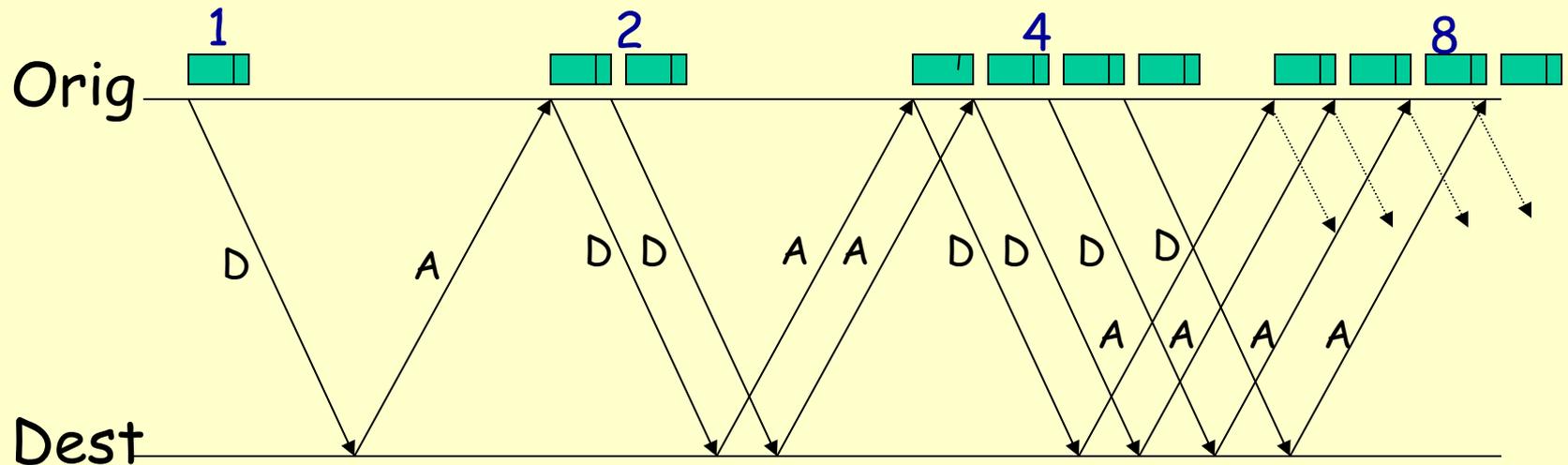


Fase “Partida lenta”

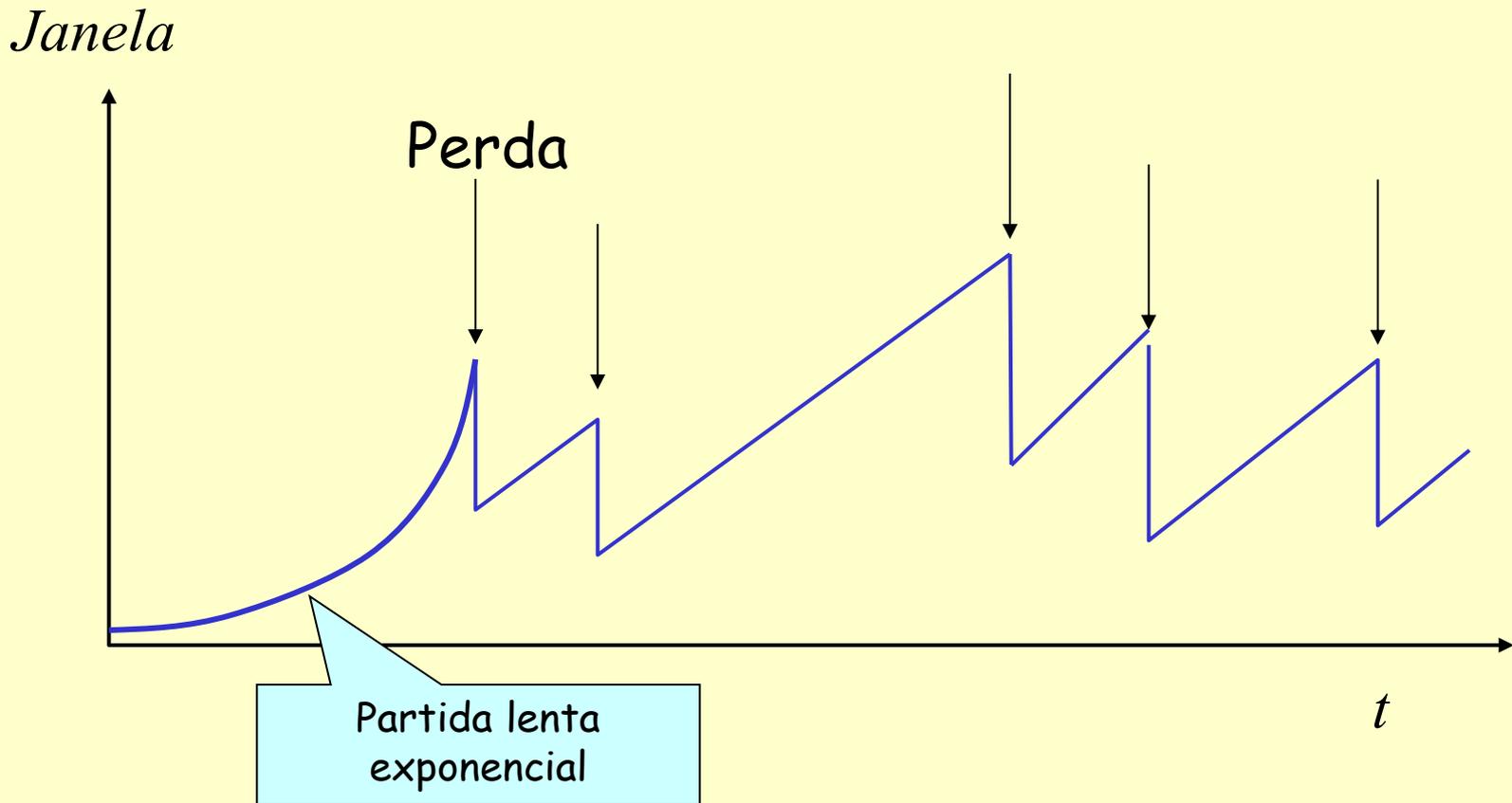
- Partida com uma pequena janela de congestionamento
 - Inicialmente, JanelaCongestionamento é 1 MSS
 - Desse modo, a taxa inicial de envio é MSS/RTT
- Isso pode ter bastante desperdício
 - Pode ser muito menor do que a largura de banda real
 - Aumento linear toma um longo tempo para acelerar
- Fase partida lenta (na realidade “partida rápida”)
 - O emissor inicia em baixa taxa
 - Mas, aumenta a taxa exponencialmente
 - Até que ocorra a primeira perda

Partida lenta

Dobre a Janela Congestionamento a cada RTT



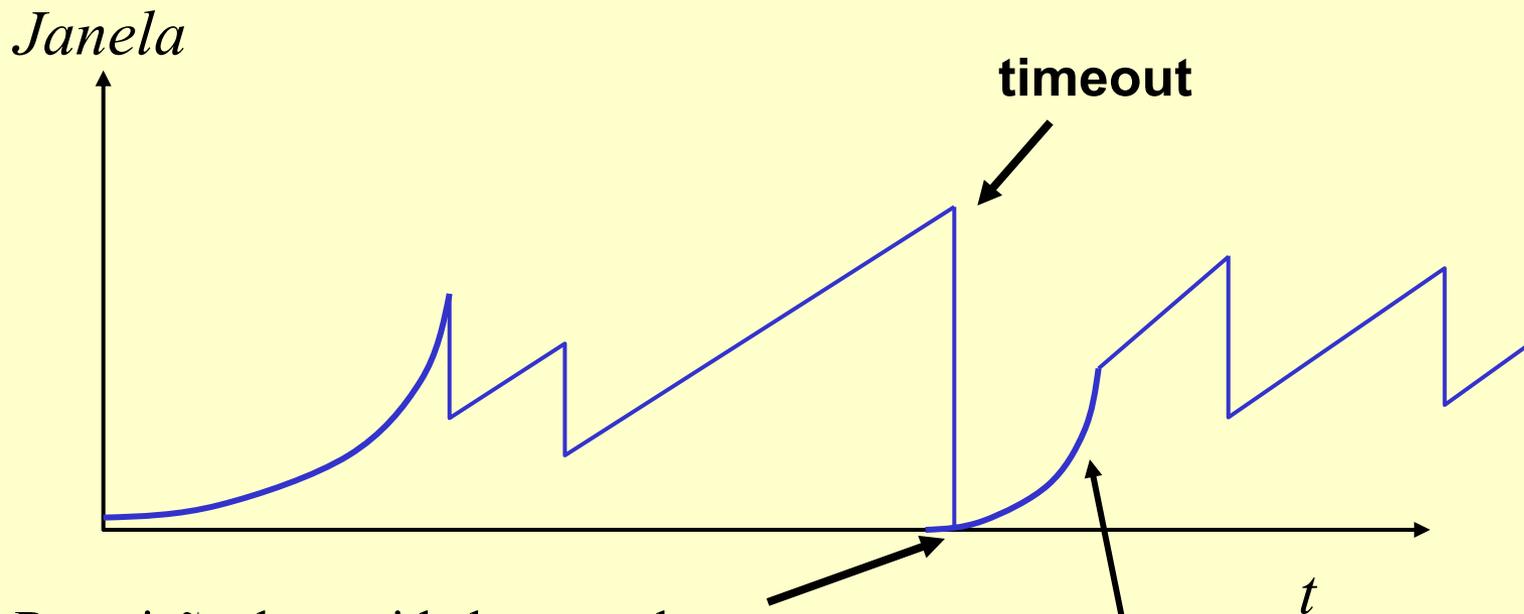
Partida lenta e dente de serra do TCP



Porque é chamada partida lenta? Porque originalmente TCP não tinha mecanismo de controle de congestionamento. A origem poderia iniciar enviando tudo que a janela permitisse. É lenta em relação ao original

Repetição da partida lenta após o timeout

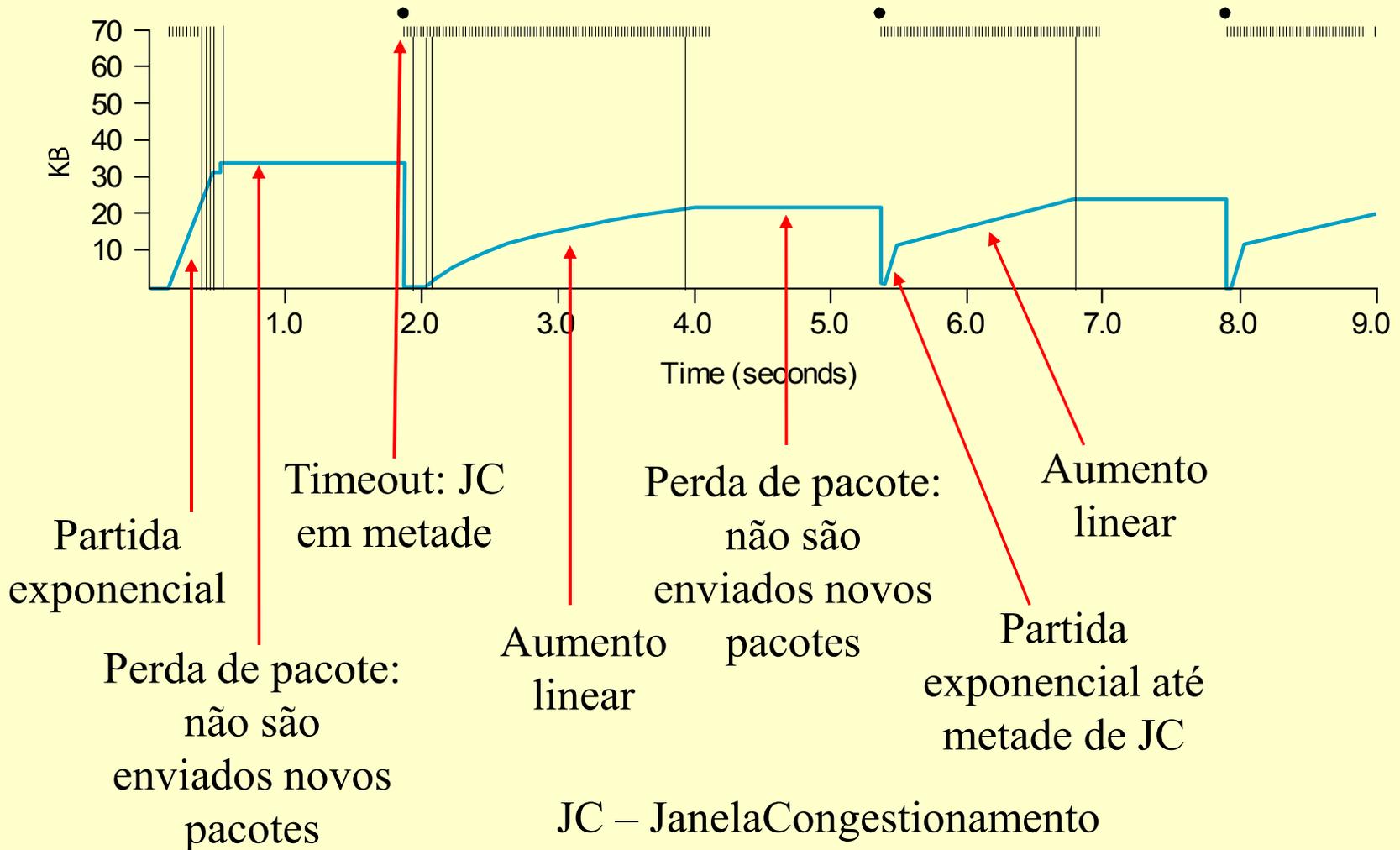
- Depois do timeout, utilizar toda a JanelaCongestionamento, pode disparar um tráfego intenso. Assim, é melhor começar com uma baixa JanelaCongestionamento



Repetição da partida lenta: volta a JanelaCongestionamento a 1, mas, leva em conta o conhecimento do valor anterior.

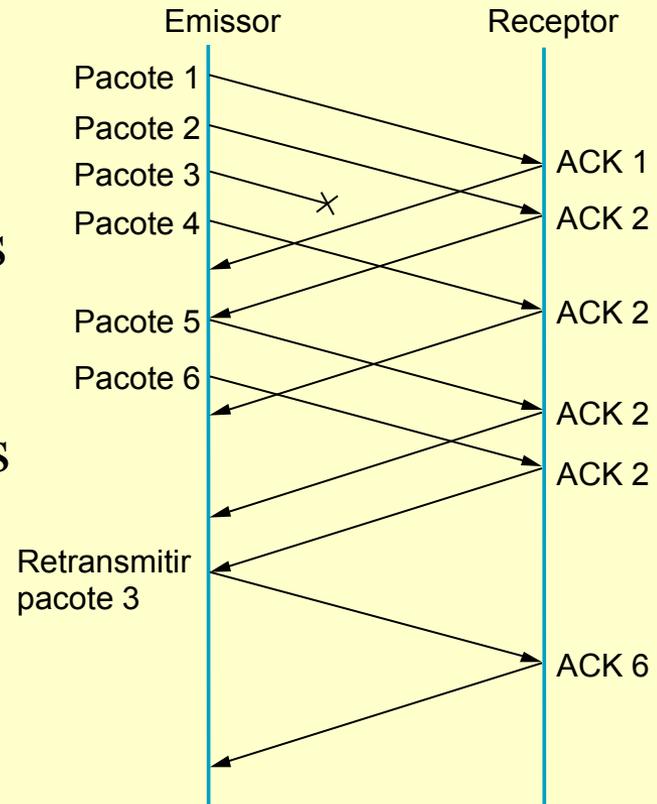
Partida lenta em operação até que alcance a metade da JanelaCongestionamento anterior

Controle de congestionamento TCP: Medições práticas



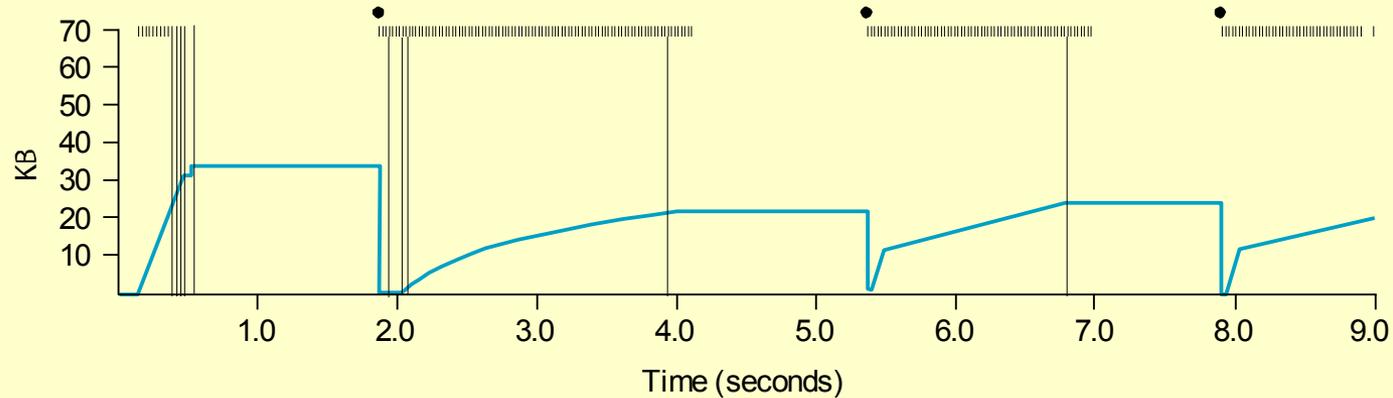
Retransmissão rápida e recuperação rápida

- Timeouts de TCP espaçados levam a períodos de ociosidade
- Retransmissão rápida: usa ACKs duplicados para disparar retransmissão. No caso do TCP são enviados 3 ACKs duplicados até que o pacote seja retransmitido.
- Envia ACK do último pacote recebido (ACK6, na figura)

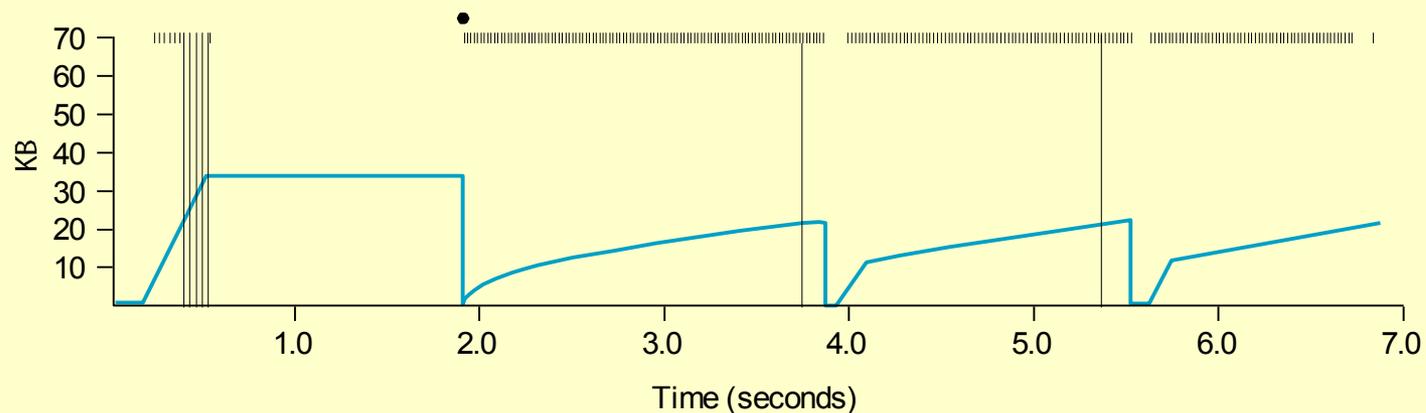


Efeito da retransmissão rápida

Sem retransmissão rápida, com partida lenta



Com retransmissão rápida, com partida lenta



Repetição de partida lenta após um período de inatividade

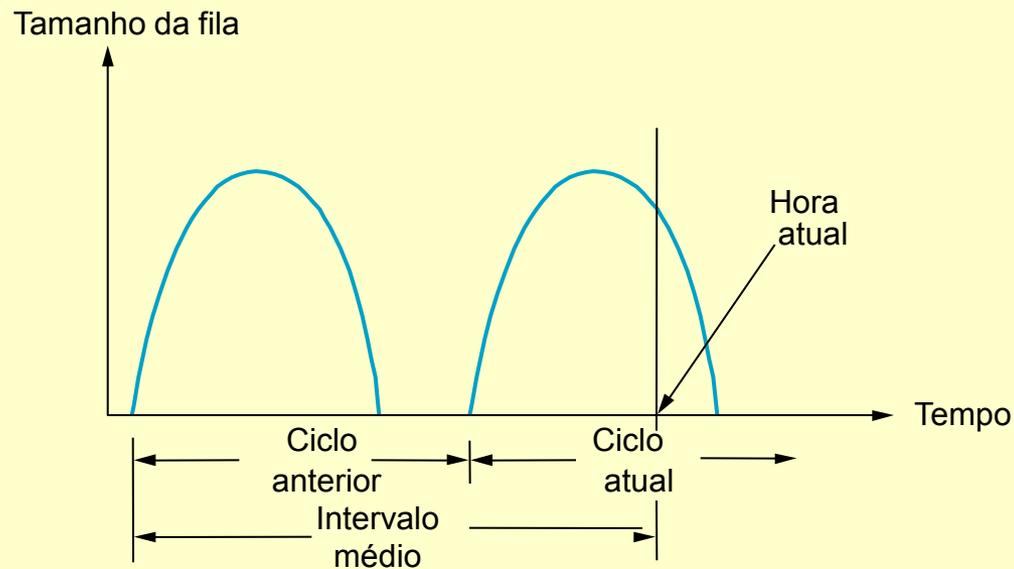
- Suponha que uma conexão TCP fique ocioso por um instante
 - Ex., sessão de Telnet que fica uma hora sem digitação.
- Não é razoável iniciar a transmissão na taxa antiga
 - O emissor que estava sem operação pode inundar a rede.
 - Pode causar congestionamento e perda de pacote.
- Assim, alguns implementações de TCP utilizam a partida lenta
 - Partida lenta no reinício de um período de inatividade.

Prevenção de congestionamento

- Estratégia do TCP
 - Controle o congestionamento quando ele acontecer
 - Aumente a carga repetidamente para encontrar o ponto em que ocorra o congestionamento, e depois recue
- Estratégia alternativa
 - Preveja quando o congestionamento está para acontecer
 - Reduza a velocidade antes que os pacotes comecem a ser descartados
 - Chame isso de *prevenção* de congestionamento, em vez de *controle* de congestionamento
- Duas possibilidades
 - Centrada no roteador: DECbit e RED Gateways
 - Centrada no host: TCP Vegas

DECbit

- Inclua bit de congestionamento a cada cabeçalho de pacote
- Roteador
 - Monitora o tamanho médio da fila durante último ciclo ocupado+ocioso



- Marque o bit de congestionamento se tamanho médio da fila > 1

Hosts finais

- Destino ecoa bit de volta para origem
- Origem registra quantos pacotes resultaram no bit marcado
- Se menos de 50% do tamanho da última janela
 - aumente **JanelaCongestionamento** em 1 pacote
- Se 50% ou mais da última janela com bit marcado
 - diminua **JanelaCongestionamento** em 0,875 vezes

Detecção Aleatória Antecipada - Random Early Detection (RED)

- Notificação é implícita
 - Apenas descarta o pacote
 - Poderia se tornar explícito marcando o pacote
- Descarte aleatório antecipado
 - Em vez de esperar que a fila se encha, descarta cada pacote que chega com alguma *probabilidade de descarte* sempre que o tamanho da fila ultrapassar determinado *nível de descarte*

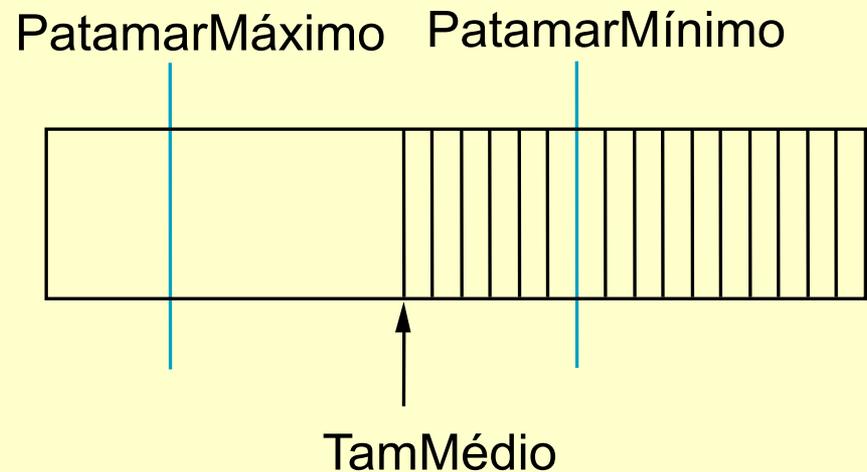
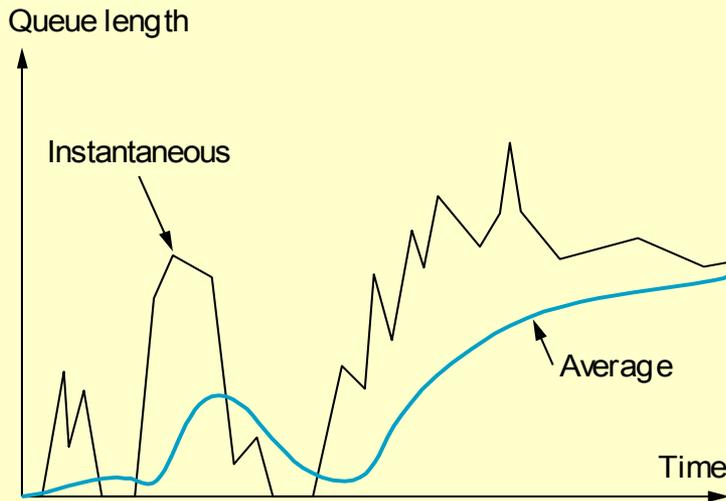
Detalhes da RED

- Calcule o tamanho médio da fila:

$$\text{TamMédio} = (1 - \text{Peso}) * \text{TamMédio} + \text{Peso} * \text{TamAmostra}$$

$0 < \text{Peso} < 1$ (normalmente 0,002)

TamAmostra é o tamanho da fila toda vez que um pacote chega.



Detalhes da RED (cont.)

- Dois limites do tamanho da fila

```
if TamMédio <= PatamarMínimo then
```

```
  coloca o pacote em fila
```

```
if PatamarMínimo < TamMédio < PatamarMáximo then
```

```
  calcula probabilidade P
```

```
  descarta pacote que chega com probabilidade P
```

```
if PatamarMáximo <= TamMédio then
```

```
  descarta pacote que chega
```

Detalhes da RED (cont.)

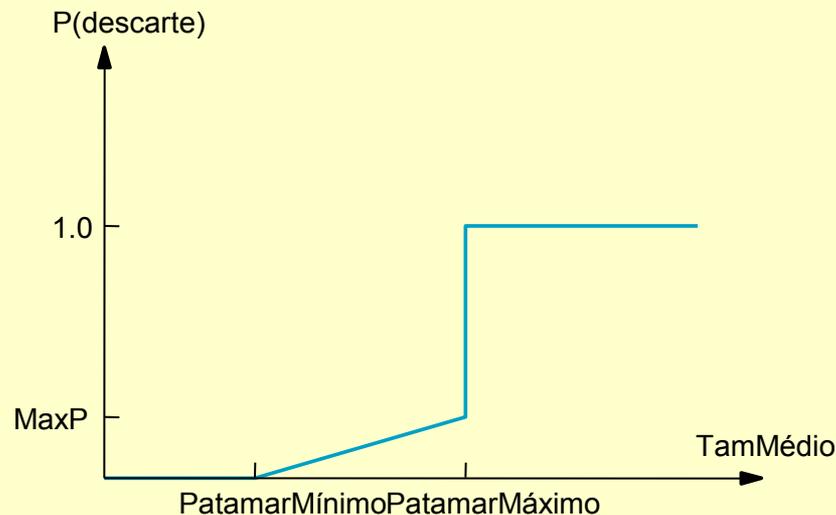
- Cálculo da probabilidade P

$$P_{Temp} = MaxP * (TamMédio - PatamarMínimo) / (PatamarMáximo - PatamarMínimo)$$

$$P = P_{Temp} / (1 - contador * P_{Temp})$$

O contador é o número que indica quantos pacotes foram enfileirados quando o TamMédio ficou entre PatamarMínimo e PatamarMáximo.

- Curva da probabilidade de descarte



Ajustando o RED

- A probabilidade de descarte dos pacotes de um fluxo é proporcional à fatia da largura de banda que o fluxo está obtendo atualmente.
- **MaxP** normalmente é definido como 0,02, significado que, quando o tamanho médio da fila está a meio caminho entre os dois limites, o gateway descarta cerca de um a cada 50 pacotes.
- Se o tráfego for em rajadas, então **PatamarMínimo** deverá ser suficientemente grande para permitir que a utilização do enlace seja mantida em um nível aceitavelmente alto.
- A diferença entre dois patamares deverá ser maior do que o aumento típico no tamanho médio calculado da fila em um RTT; a definição de **PatamarMáximo** com o dobro de **PatamarMínimo** é razoável para o tráfego na Internet de hoje.

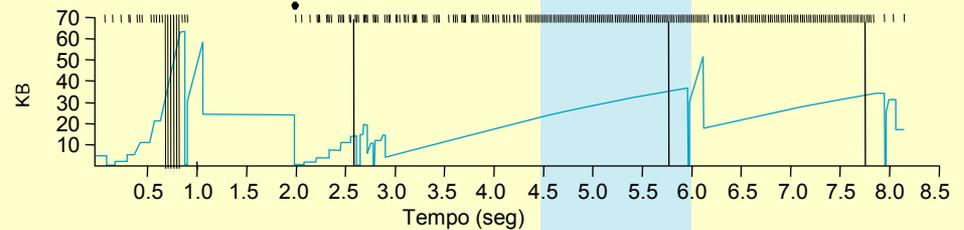
Notificação Explícita de Congestionamento (ECN)

- Descarte antecipado de pacotes
 - Vantagem: dá realimentação antecipada
 - Desvantagem: tem que descartar pacote para realimentar
- Notificação explícita de congestionamento
 - Roteador marca o pacote com um bit ECN
 - Sinaliza para host uma indicação de congestionamento
- Implementação
 - Deve ter suporte dos hosts finais e dos roteadores
 - Requer dois bits no cabeçalho IP (um para a origem indicar que é capaz de usar ECN e um outro definido pelos roteadores para indicar o congestionamento)
 - Solução: utilizar dois bits do campo TOS (Type-Of-Service) do cabeçalho do IPv4

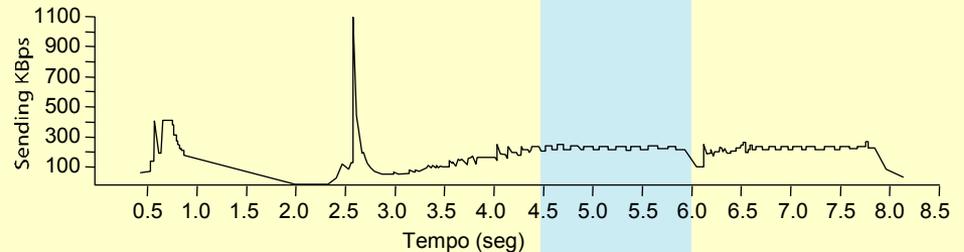
TCP Vegas

- Idéia: origem observa algum sinal de que a fila do roteador está aumentando e que o congestionamento também ocorrerá; por exemplo:
 - RTT cresce
 - taxa de envio nivela

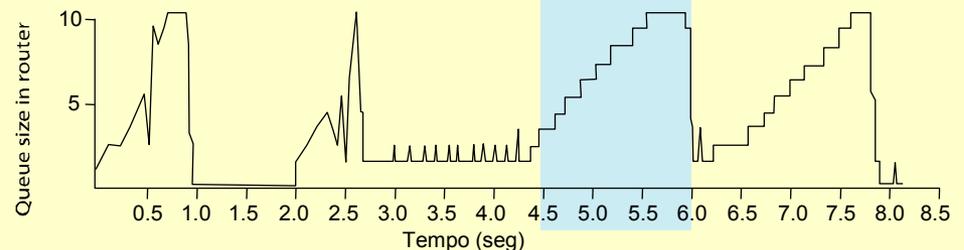
JanelaCongestionamento



Throughput



Tamanho da fila no roteador



Algoritmo

- Seja **RTTBásico** o menor de todos os RTTs medidos (normalmente, o RTT do primeiro pacote)
- Se a conexão não estiver na sobrecarga, então
TaxaEsperada = JanelaCongestionamento/RTTBásico
- Origem calcula taxa de envio (**TaxaReal**) uma vez por RTT
- Origem compara **TaxaReal** com **TaxaEsperada**

Dif = TaxaEsperada - TaxaReal

if Dif < α

aumenta JanelaCongestionamento linearmente

else if Dif > β

diminui JanelaCongestionamento linearmente

else

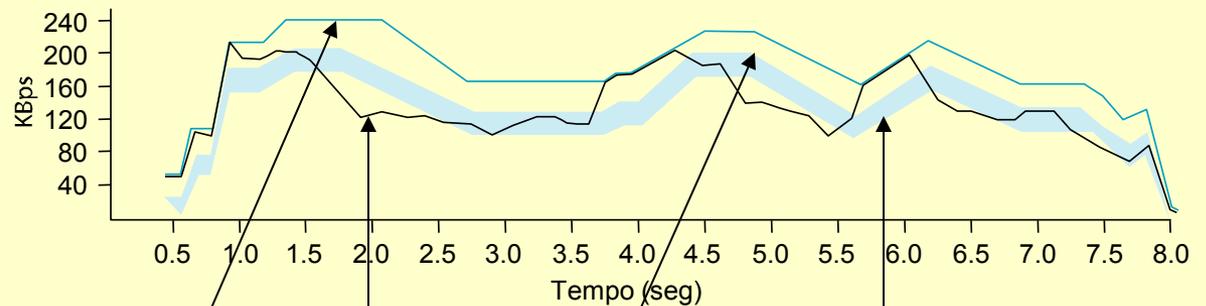
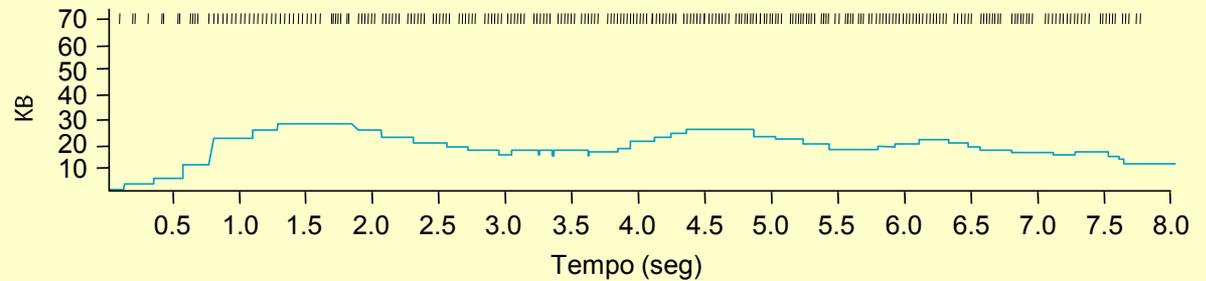
deixa JanelaCongestionamento inalterada

Algoritmo (cont.)

- Parâmetros, ex.
 - $\alpha = 30$ KBps
 - $\beta = 60$ KBps

Sempre que a TaxaReal fica abaixo da faixa, a janela é diminuída (indicação de fila aumentando no roteador). Sempre que a TaxaReal fica acima da faixa, a janela é aumentada (indicação de ociosidade). Se TaxaReal está na faixa, nenhuma providência.

JanelaCongestionamento



TaxaEsperada

TaxaReal

α KBps da TaxaEsperada

β KBps da TaxaEsperada

Exercício

1 Considere o efeito de usar a partida lenta em uma linha com um RTT de 10 ms e sem congestionamento. A JanelaCongestionamento tem 24 KB e o tamanho máximo do segmento é de 2 KB. Quanto tempo é necessário para que a primeira janela completa possa ser enviada?