

# Redes de Computadores II

## Continuando o Protocolo TCP



**Prof. Elias P. Duarte Jr.**

Universidade Federal do Paraná (UFPR)

Departamento de Informática

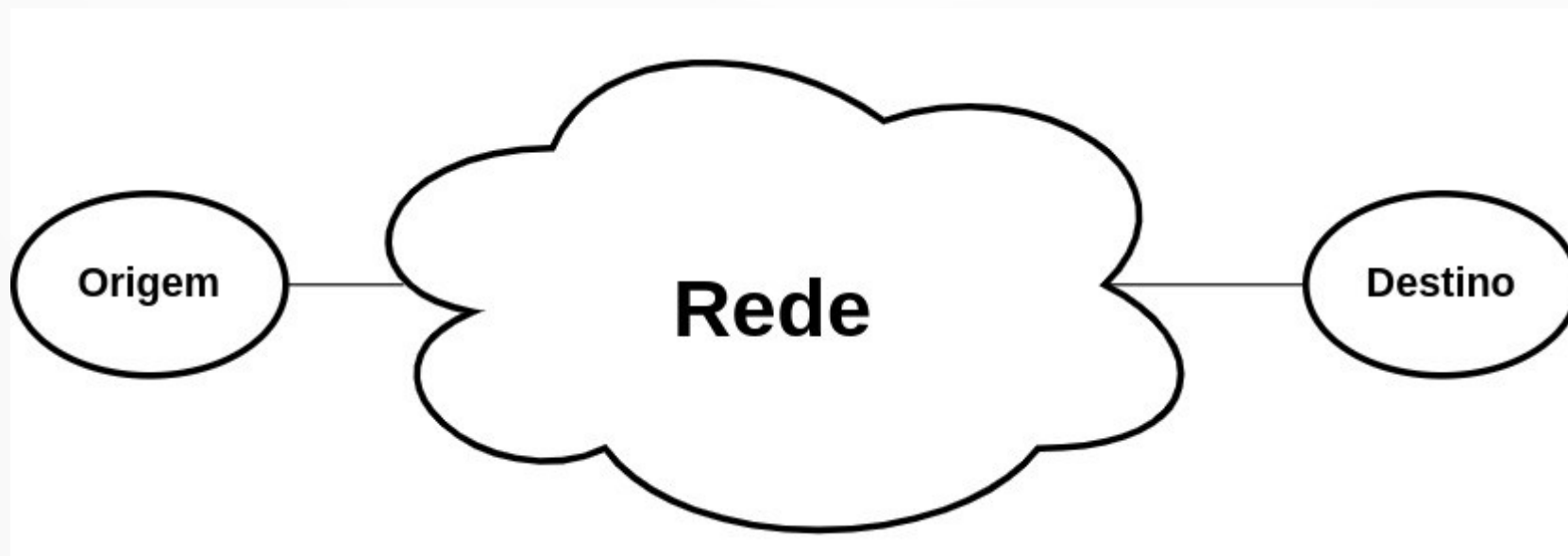
[www.inf.ufpr.br/elias/redes](http://www.inf.ufpr.br/elias/redes)

# Sumário da Aula de Hoje

- Hoje continuamos nosso estudo do protocolo TCP
- Foco no Controle de Congestionamento

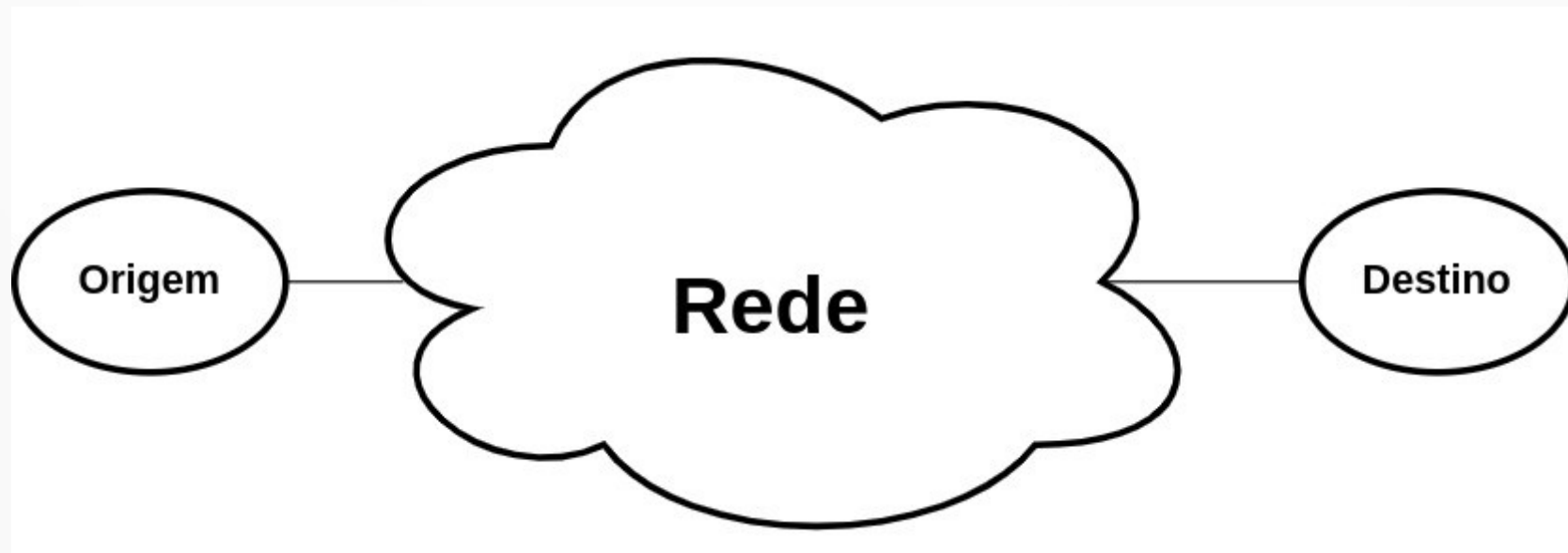
# Controle de Congestionamento

- O controle de fluxo permite à origem estimar a capacidade do destino:



# Controle de Congestionamento

- O controle de fluxo permite à origem estimar a capacidade do destino:
- Entre a origem e destino: a rede!
- O controle de congestionamento permite à origem estimar a capacidade da rede



# Controle de Congestionamento

- Uma janela de congestionamento é constantemente atualizada para refletir esta estimativa
- Janela = número de bytes
- Vamos chamar a janela do controle de congestionamento de JCONG

# Controles de Fluxo & Congestionamento

- Antes de **toda** transmissão as duas janelas são comparadas
  - Janela do Controle de Congestionamento (JCONG)
  - Janela do Controle de Fluxo (WIN)
- Só pode transmitir o menor valor para o número de bytes



# Controle de Congestionamento do TCP

- Não definido no padrão original
- Cada implementação do TCP calcula localmente
- Diversas estratégias foram propostas ao longo das décadas
- Uma versão do TCP que implementa uma estratégia leva o nome da cidade onde ocorreu a reunião do IETF que a padronizou
  - TCP Reno, TCP Tahoe, TCP Vegas, TCP Westwood, TCP Illinois, etc. etc. etc.

# Controle de Congestionamento do TCP

- Vamos antes relembrar a definição de congestionamento em redes de computadores
- Uma rede congestionada: pacotes estão chegando ao(s) roteador(es) em taxa alta
- Maior do que consegue(m) processar
- Resultado: pacotes são **descartados**

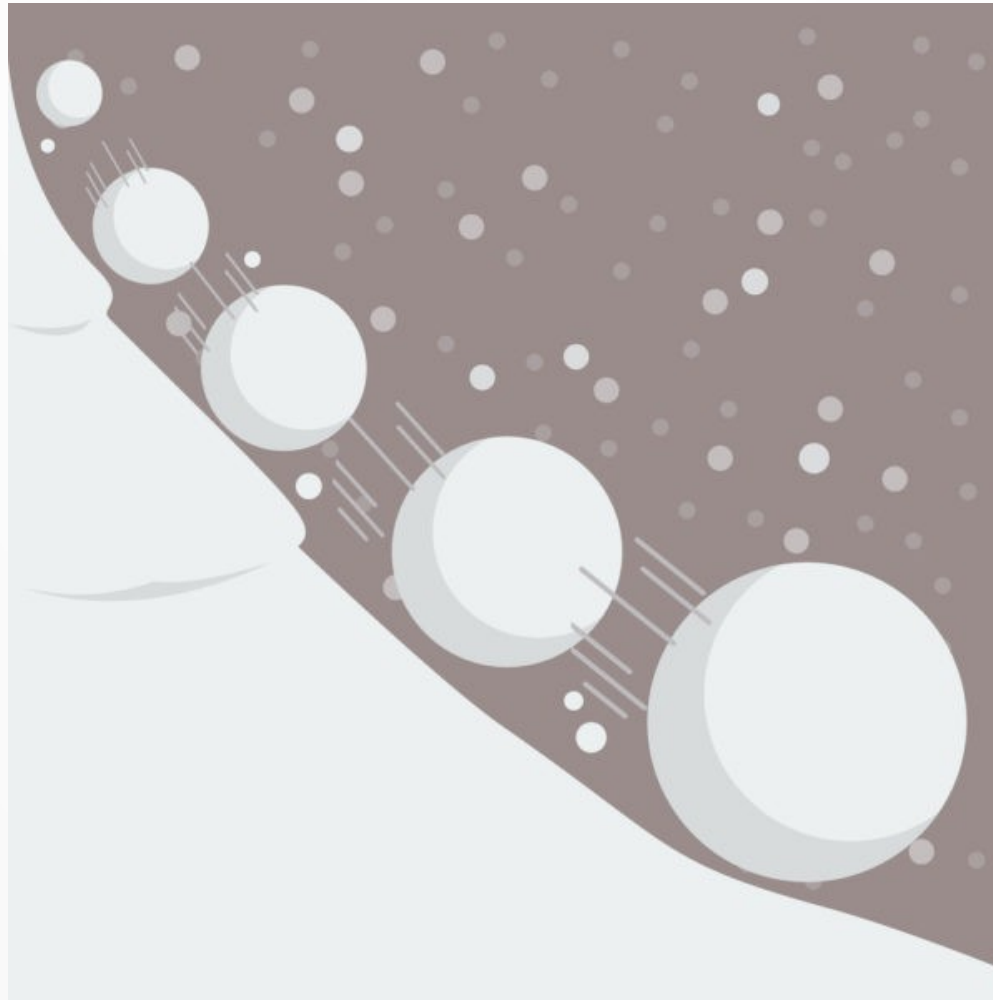


# Congestionamento: Definição

- Vamos aproveitar este momento para definir congestionamento em redes de computadores
- Uma rede congestionada: pacotes estão chegando ao(s) roteador(es) em taxa alta
- Maior do que consegue(m) processar
- Resultado: pacotes são descartados
- Definição alternativa (livro Kurose/Ross): carga alta que interfere no desempenho

# Congestionamento

- O congestionamento é dramático: bola de neve!



# Congestionamento

- Por que acontece o efeito bola de neve?
- Considere que em  $t = 1$  a rede ficou congestionada
- Considere que há uma média de pacotes transmitidos em cada unidade de tempo
- Aqueles pacotes transmitidos em  $t = 1$  são descartados
- Vão ser re-transmitidos em  $t = 2$ , além dos pacotes que seriam inicialmente transmitidos em  $t = 2$

# Congestionamento

- $t = 1 \rightarrow$  transmit.
- $t = 2 \rightarrow$  transmit./retransmit.
- $t = 3 \rightarrow$  transmit./retransmit./re-retransmit
- $t = 4 \rightarrow$  transmit./retransmit./re-retransmit/re-re-retransmit
- $t = 5 \rightarrow$  transmit./retransmit./re-retransmit/re-re-retransmit/re-re-re-retransmit
- $t = 6 \rightarrow$  transmit./retransmit./re-retransmit/re-re-re-retransmit/re-re-re-re-re-re-retransmit
- ...

# Congestionamento: Solução

- Só existe 1 solução para o congestionamento

# Congestionamento: Solução

- Só existe 1 solução para o congestionamento
- Diminuir a taxa em que novos pacotes entram na rede
- Na prática é isso que o controle de congestionamento do TCP faz → de formas diversas



# Slow-Start TCP

- Solução clássica para controle de congestionamento do TCP
- Inicialmente: a rede é desconhecida
- Faz  $J_{\text{Cong}} \leftarrow 1$  segmento
  - Na verdade o número de bytes do MSS
  - MSS: Maximum Segment Size
  - Em geral setado para caber em um quadro Ethernet
  - Quantos bytes?

# Slow-Start TCP

- Solução clássica para controle de congestionamento do TCP
- Inicialmente: a rede é desconhecida
- Faz  $J_{\text{Cong}} \leftarrow 1$  segmento
  - Na verdade o número de bytes do MSS
  - MSS: Maximum Segment Size
  - Em geral setado para caber em um quadro Ethernet
  - 1480 bytes de TCP (mais 20 bytes de header IP)

# Slow-Start TCP

- Transmite 1 segmento e aguarda...
- Chega a confirmação (ACK)
- Dobra a JCong  $\leftarrow 2$
- Transmite 2 segmentos e aguarda...
- Chega a confirmação (ACK)
- Dobra a JCong  $\leftarrow 4$
- Transmite 4 segmentos e aguarda...

# Slow-Start TCP

- Transmite 4 segmentos e aguarda...
- Chega ACK → dobra JCong ← 8
- Transmite 8 segmentos e aguarda.....
- Aguarda.....
- Aguarda.....

# Slow-Start TCP

- Transmite 4 segmentos e aguarda...
- Chega ACK → dobra JCong ← 8
- Transmite 8 segmentos e aguarda.....
- Aguarda.....
- Aguarda.....
- Timeout!



# Slow-Start TCP: Timeout

- Quando ocorre o timeout: o TCP considera que a rede está congestionada
  - vamos conversar sobre isso



# Slow-Start TCP: Timeout

- Quando ocorre o timeout: o TCP considera que a rede está congestionada
  - vamos conversar sobre isso
- Rede congestionada: situação dramática
- Reduz a JCong  $\leftarrow$  1 segmento
- Marca um limite (*threshold*) de metade da JCong que levou ao congestionamento da rede

# Slow-Start TCP

- Transmite 1 segmento e aguarda...
- Chega a confirmação (ACK)
- Dobra a JCong  $\leftarrow 2$
- Transmite 2 segmentos e aguarda...
- Chega a confirmação (ACK)
- Dobra a JCong  $\leftarrow 4$
- Transmite 4 segmentos e aguarda...

# Slow-Start TCP

- Transmite 4 segmentos e aguarda...
- Chega ACK → dobra JCong??
- Lembra da história do gato escaldado que tem medo de água fria?



# Slow-Start TCP

- Transmite 4 segmentos e aguarda...
- Chega ACK → dobra JCong??
- Se dobrar a JCong vai para 8, que é justamente o valor que causou o congestionamento da última vez!
- A partir deste ponto limite (JCong == 4) o Slow-Start incrementa a JCong de 1 em 1

# Slow-Start TCP

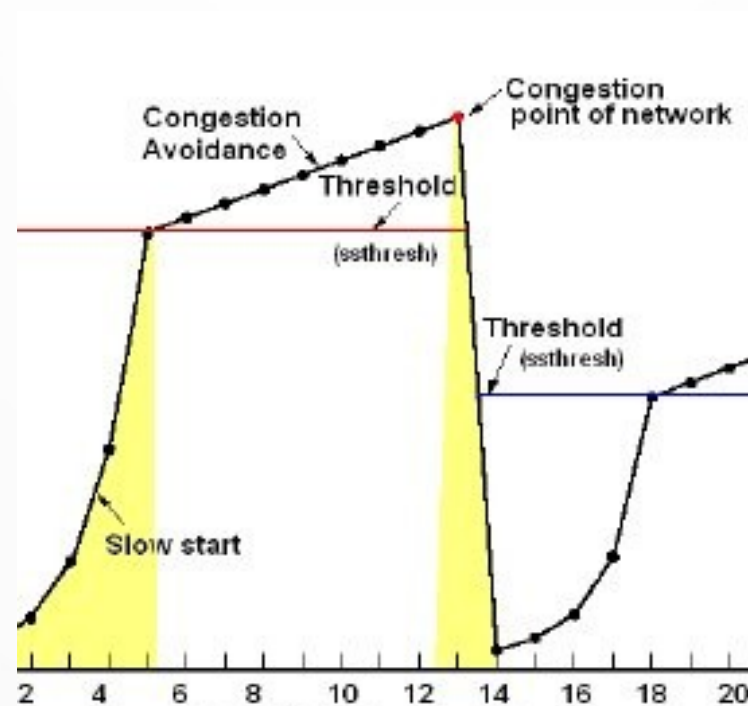
- Transmite 4 segmentos e aguarda...
- Chega ACK → incrementa JCong ← 5
- Transmite 5 segmentos e aguarda...
- Chega ACK → incrementa JCong ← 6
- Transmite 6 segmentos e aguarda....

# Slow-Start TCP

- Transmite 4 segmentos e aguarda...
- Chega ACK → incrementa JCong ← 5
- Transmite 5 segmentos e aguarda...
- Chega ACK → incrementa JCong ← 6
- Transmite 6 segmentos e aguarda....
- Aguarda..... timeout!
- Agora o *threshold* é de 3 segmentos
- Segue!



# Gráfico que Reflete Slow Start



# O Que Faltou Explicar?

- Estudamos o Slow-Start TCP quase todo
- Faltou explicar uma coisa: o que?



# O Timeout

- Como o TCP calcula o *timeout*?
- Veja: é muito importante calcular o *timeout* precisamente
- Se for menor do que deveria: vai causar retransmissões desnecessárias
  - impacto no uso desnecessário da rede
  - também interfere no atraso
- Se for maior do que deveria: demora a detectar perdas, retransmissões lentas
  - diminui o próprio tempo de resposta da rede para o usuário

# O Timeout

- Como o TCP calcula o *timeout*?
- Base: o RTT
- RTT: *Round Trip Time*
- Tempo entre uma mensagem ir da origem ao destino e a resposta chegar do destino à origem
- Fácil de medir!
- O RTT é variável – espera-se que não muito, mas é variável

# O Cálculo do Timeout

- RTT variável: TCP calcula um valor médio
- Média ponderada:
- $RTT\_médio = \alpha * RTT\_médio + (1-\alpha) * Novo\_RTT$
- Em geral  $\alpha = 7/8$  (no Linux = 0.9)
- Peso maior para valor calculado, em comparação com nova medida
- Vamos fazer um exemplo...

# Cálculo do Timeout

- Depois de calcular o  $RTT_{\text{médio}}$ : qual o timeout?



# Cálculo do Timeout

- Depois de calcular o  $RTT_{\text{méd}}io$ : qual o timeout?
- Digamos:  $Timeout \leftarrow \beta * RTT_{\text{méd}}io$
- Qual o valor de  $\beta$ ?

# Cálculo do Timeout

- Depois de calcular o  $RTT_{\text{méd}}io$ : qual o timeout?
- Digamos:  $Timeout \leftarrow \beta * RTT_{\text{méd}}io$
- Qual o valor de  $\beta$ ?

# Cálculo do Timeout

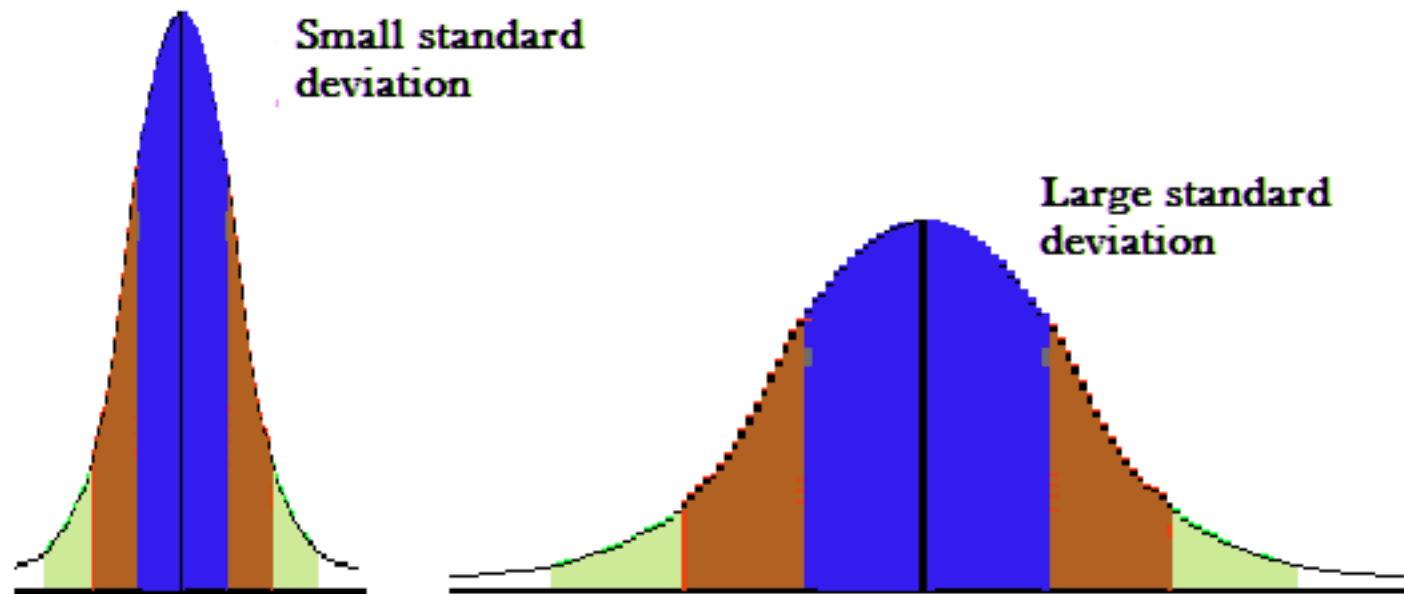
- Depois de calcular o  $RTT_{\text{médio}}$ : qual o timeout?
- Digamos:  $\text{Timeout} \leftarrow \beta * RTT_{\text{médio}}$
- Qual o valor de  $\beta$ ?
- Originalmente  $\beta \leftarrow 2$
- Com o tempo percebeu-se que esta estratégia não era muito eficiente...

# Desvio Médio

- Por que é importante?

# Desvio Médio

- Por que é importante?
- A média não reflete características do conjunto de dados
- O desvio é uma medida de dispersão da média



# Desvio Médio

- Usar o desvio pode ter ajudar a tomar decisões no dia-a-dia
- Reflete o comportamento da conexão
- Desvio padrão caro de calcular: vamos de desvio médio mesmo
- $\text{Novo\_desvio} \leftarrow |\text{RTT\_médio} - \text{Novo\_RTT}|$
- $\text{Desvio\_médio} = \alpha * \text{Desvio\_médio} + (1-\alpha) * \text{Novo\_Desvio}$
- Em geral  $\alpha = 7/8$  (no Linux = 0.9)



# Finalmente o Timeout

- O timeout usado hoje no TCP
- $RTO \leftarrow RTT_{\text{médio}} + \beta * \text{Desvio}_{\text{médio}}$
- O valor sugerido para  $\beta$  é...


# Finalmente o Timeout

- O timeout usado hoje no TCP
- $RTO \leftarrow RTT_{\text{médio}} + \beta * \text{Desvio}_{\text{médio}}$
- O valor sugerido para  $\beta \leftarrow 4$
- Esta forma demonstrou ser muito precisa

# Slow-Start: TCP Tahoe

- O TCP que implementa esta estratégia de controle de congestionamento é o TCP Tahoe
- Outro nome da estratégia: AIMD
  - *Additive Increase Multiplicative Decrease*
- Um dos problemas do TCP Tahoe: quando 1 pacote se perde: efeito terrível na taxa
- Reduz janela drasticamente...

# TCP Reno: Retransmissão Rápida

- O TCP Reno resolve este problema
- Quando recebe 3 ACKs para um mesmo segmento antigo...
- ... alta probabilidade de que o seguinte se perdeu
- Daí retransmite o que foi depois!
- Sem esperar o timeout
- Evitando assim a redução da JCong 

# Muitas Outras Estratégias

- Reforçando que há muitas outras estratégias de controle de congestionamento
- Acredito sinceramente que – tendo esta base – vocês conseguem entender todas

# Conclusão

- Nesta aula vimos a base do controle de congestionamento → várias, várias estratégias
- Slow-Start TCP
- Cálculo do timeout
- Retransmissão rápida



**Obrigado!**

Lembrando: a página da disciplina é:  
<https://www.inf.ufpr.br/elias/redes>