

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE



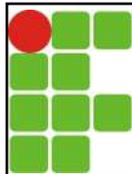
Introdução às Redes de Computadores

Turma : 20192.1.01405.1N

Camada de Transporte - Parte II

Prof. Thiago Dutra <thiago.dutra@ifrn.edu.br>

1



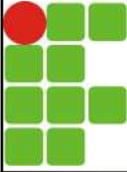
Agenda

- Parte I
 - Introdução
 - Protocolos de Transporte Internet
 - Multiplexação e Demultiplexação
 - UDP

- Parte II
 - TCP
 - Controle de Congestionamento

2

2

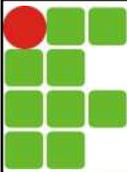


Agenda Parte II

- TCP - Introdução
- TCP - Estabelecendo uma Conexão
- TCP - Encerrando uma Conexão
- TCP - Estados da Conexão
- Transferência de Dados Confiável
- TCP - Transferência de Dados
- Controle de Congestionamento
- TCP - Controle de Congestionamento

3

3



TCP - Introdução

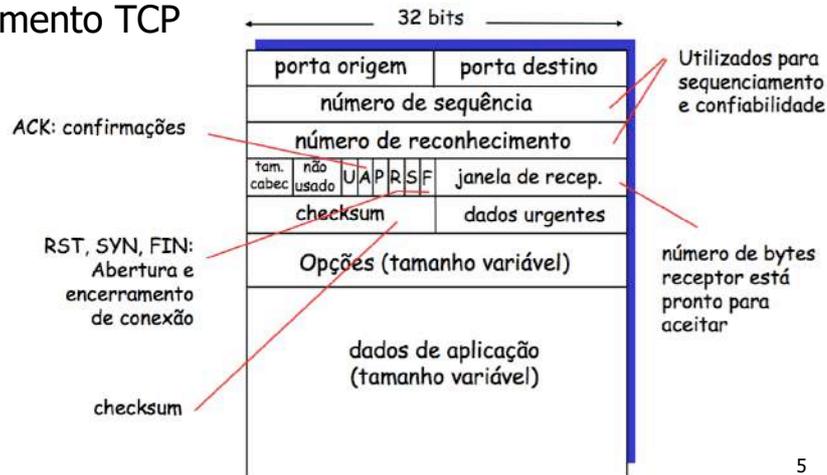
- **T**ransmission **C**ontrol **P**rotocol
 - Protocolo de Controle de Transmissão
 - RFC's: 793, 1122, 1323, 2018, 2581, ...
 - Oferece um serviço :
 - Orientado à conexão
 - Confiável
 - Com controle de erros
 - Com controle de fluxo
 - Com controle de congestionamento

4

4

TCP - Introdução

■ Segmento TCP



5

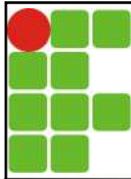
5

TCP – Estabelecendo uma Conexão

- Para que cliente e servidor possam se comunicar usando TCP, eles **precisam primeiramente estabelecer uma conexão**
- Durante o estabelecimento dessa conexão, diversas "preparações" são realizadas, ex.:
 - Cliente escolhe uma porta de origem
 - Servidor aloca buffers e variáveis à conexão
- O procedimento de estabelecimento da conexão é denominado **apresentação de três vias**
 - 3-Way Handshake

6

6



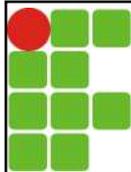
TCP – Estabelecendo uma Conexão

■ 3-Way Handshake

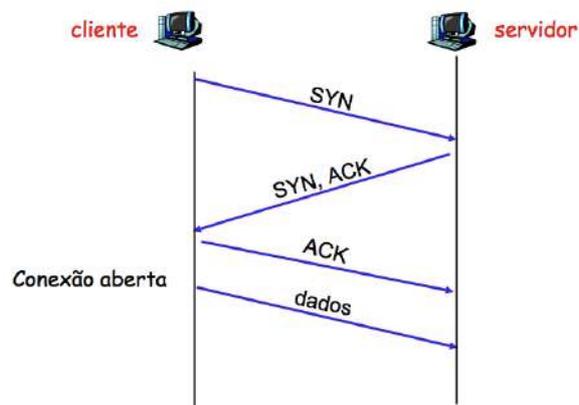
- (1) **Cliente** envia um segmento TCP para o servidor com o bit **SYN** habilitado (igual a "1")
- (2) **Servidor** responde com outro segmento TCP para o cliente com os bits **SYN** e **ACK** habilitados
- (3) **Cliente** responde com outro segmento TCP para o servidor com o bit **ACK** habilitado

7

7

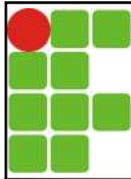


TCP – Estabelecendo uma Conexão



8

8

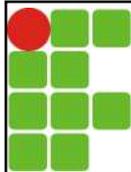


TCP – Encerrando uma Conexão

- Tanto o cliente quanto o servidor podem encerrar a conexão
 - Quando uma conexão é encerrada os recursos associados são liberados
- Quem desejar fechar a conexão inicia o processo :
 - O primeiro envia segmento TCP com o bit **FIN** habilitado
 - O outro responde com um **ACK** e envia um **FIN**
 - O primeiro responde com um **ACK**

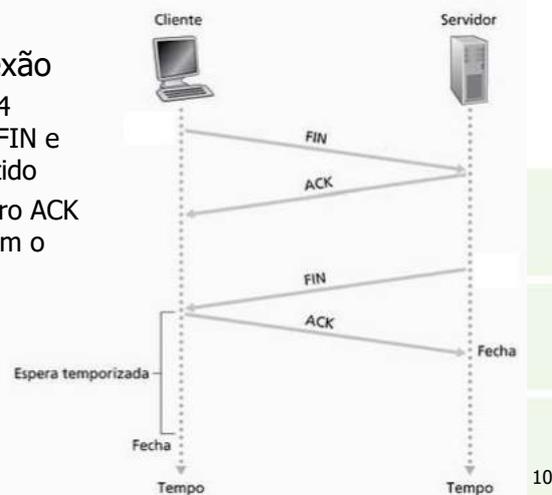
9

9



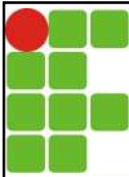
TCP – Encerrando uma Conexão

- Cliente solicitando o encerramento da conexão
 - Em geral são trocados 4 segmentos, isto é, um FIN e um ACK para cada sentido
 - É possível que o primeiro ACK e o segundo FIN ocupem o mesmo segmento



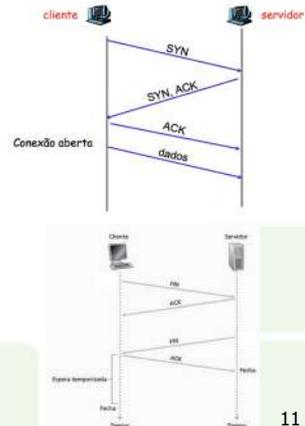
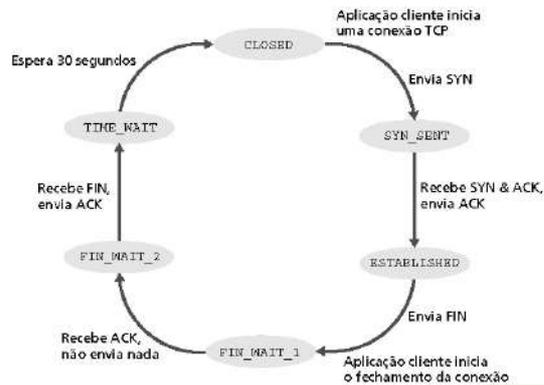
10

10



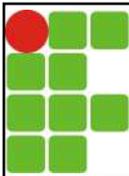
TCP – Estados da Conexão

■ Sequência típica de estados de uma conexão TCP cliente



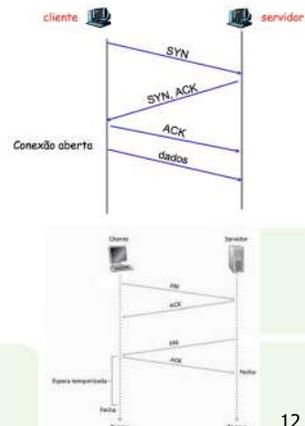
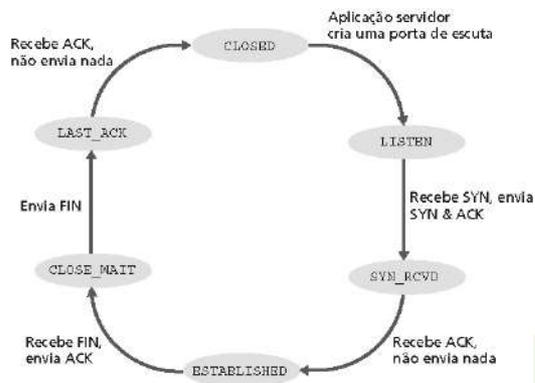
11

11



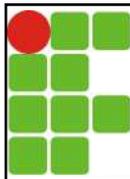
TCP – Estados da Conexão

■ Sequência típica de estados de uma conexão TCP servidor



12

12

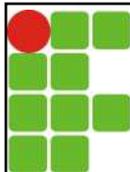


Transferência de Dados Confiável

- 1º dentre os TOP 10 problemas fundamentalmente importantes para o trabalho em rede.
- O serviço fornecido para as camadas superiores por um **protocolo de transferência de dados confiável** deve ser um **canal confiável** de comunicação
- Em um canal confiável os dados transferidos:
 - Não são corrompidos
 - Não são perdidos
 - São entregues na ordem em que foram enviados

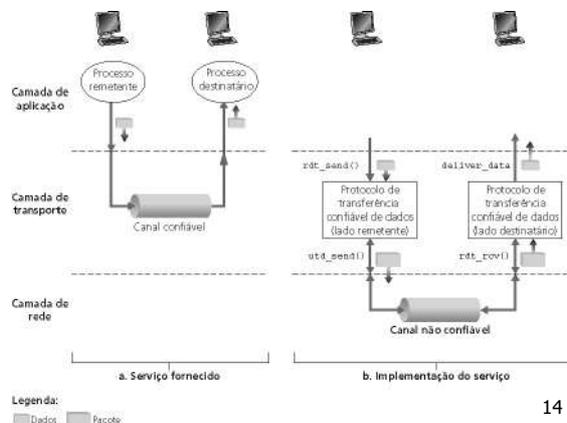
13

13



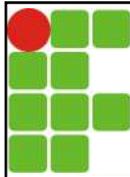
Transferência de Dados Confiável

- Protocolos de camadas mais baixas podem ser não confiáveis
 - Ex.: TCP sobre IP



14

14



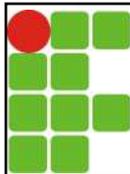
Transferência de Dados Confiável

- Resumo de mecanismos de transferência confiável de dados

Mecanismo	Utilização, Comentários
Soma de verificação	Usado para detectar erros de bits em um pacote transmitido.
Temporizador	Usado para controlar a temporização/retransmitir um pacote, possivelmente porque o pacote (ou seu ACK) foi perdido dentro do canal. Como pode ocorrer esgotamento de temporização quando um pacote está atrasado, mas não perdido (esgotamento de temporização prematuro), ou quando um pacote foi recebido pelo destinatário mas o ACK remetente-destinatário foi perdido, um destinatário pode receber cópias duplicadas de um pacote.
Número de sequência	Usado para numeração sequencial de pacotes de dados que transitam do remetente ao destinatário. Lacunas nos números de sequência de pacotes recebidos permitem que o destinatário detecte um pacote perdido. Pacotes com números de sequência duplicados permitem que o destinatário detecte cópias duplicadas de um pacote.
Reconhecimento	Usado pelo destinatário para avisar o remetente que um pacote ou conjunto de pacotes foi recebido corretamente. Reconhecimentos normalmente portam o número de sequência do pacote, ou pacotes, que estão sendo reconhecidos. Reconhecimentos podem ser individuais ou cumulativos, dependendo do protocolo.
Reconhecimento negativo	Usado pelo destinatário para avisar o remetente que um pacote não foi recebido corretamente. Reconhecimentos negativos normalmente portam o número de sequência do pacote que não foi recebido corretamente.
Janela, paralelismo	O remetente pode ficar restrito a enviar somente pacotes com números de sequência que caem dentro de uma determinado faixa. Permitindo que vários pacotes sejam transmitidos, ainda que não reconhecidos, a utilização do remetente pode ser aumentada em relação ao modo de operação pare e espere. Em breve veremos que o tamanho da janela pode ser estabelecido com base na capacidade do destinatário receber e fazer buffer de mensagens ou no nível de congestionamento na rede, ou em ambos.

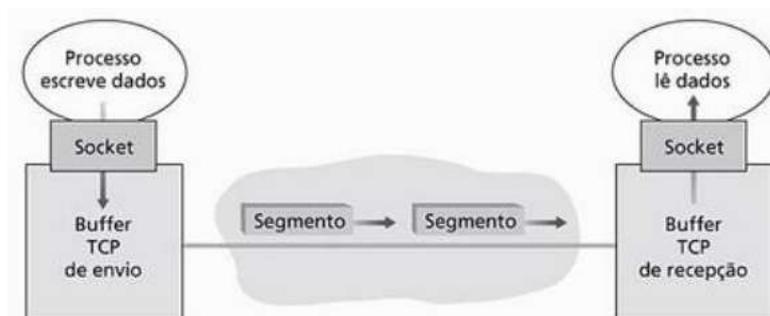
15

15



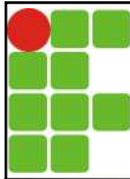
TCP – Transferência de Dados

- Após estabelecida uma conexão TCP os processos de aplicação podem trocar dados



16

16

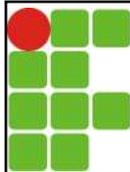


TCP – Transferência de Dados

- Em uma transferência do cliente para o servidor:
 - **Processo cliente envia dados através do socket associado** (após essa passagem, o TCP possui o controle sobre os dados)
 - **Os dados são direcionados para o buffer de envio da conexão** (buffer criado durante o 3-way handshake)
 - **De tempos em tempos o TCP retira grandes porções de dados do buffer de envio** (não existe normatização de quando o TCP deve realmente retirar os dados do buffer)
 - **O TCP encapsula os dados com um cabeçalho criando os segmentos que são enviados para a camada de rede**
 - **O servidor recebe os segmentos e coloca os dados do segmento no buffer de recepção**
 - **A aplicação lê a cadeia de dados desse buffer**

17

17

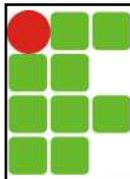


TCP – Transferência de Dados

- **Números de sequência**
 - **Possibilitam a “remontagem” dos dados da mesma forma que foram transmitidos na origem**
 - **Apesar da nomenclatura ser número de sequência seu valor não é sequencial**
 - **São aplicados sobre os dados transmitidos, e não sobre a série de segmentos**
 - **O número de sequência para um segmento é o número do primeiro byte dos dados do segmento**

18

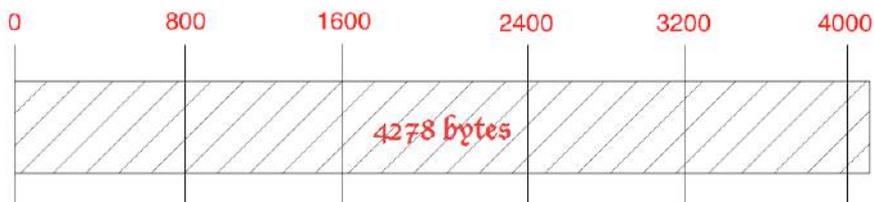
18



TCP – Transferência de Dados

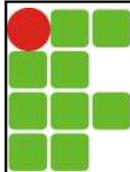
■ Números de sequência

- Suponha que um servidor irá enviar um arquivo de **4287 bytes** para o cliente **em pedaços de 800 bytes**
- Neste caso, os números de sequência dos segmentos TCP serão: **0, 800, 1600, 2400, 3200 e 4000**



19

19



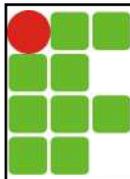
TCP – Transferência de Dados

■ Números de reconhecimento

- Sempre que um servidor ou cliente recebe um segmento TCP ele envia um outro segmento, com o bit ACK habilitado, confirmando sua recepção
 - Esta técnica é chamada **reconhecimento positivo**
- O número de reconhecimento **é igual ao próximo byte que se deseja receber**
- Se após um tempo limite o ACK correspondente não for recebido, o segmento é retransmitido

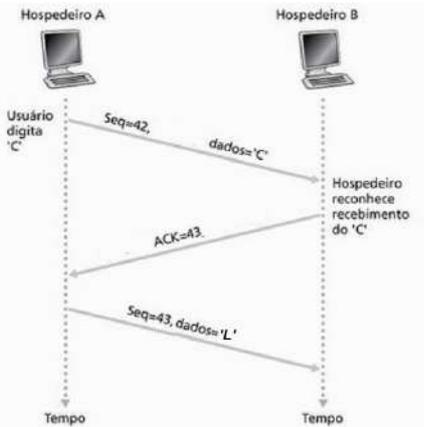
20

20

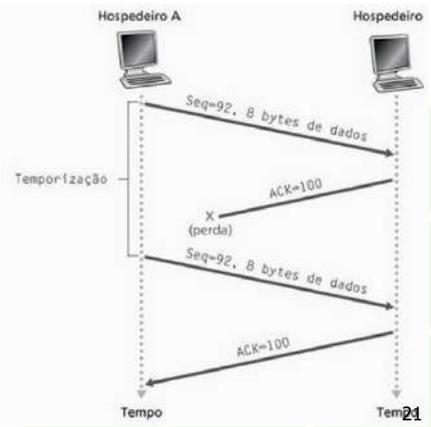


TCP – Transferência de Dados

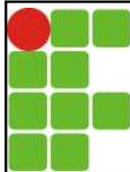
■ Reconhecimento



■ Retransmissão

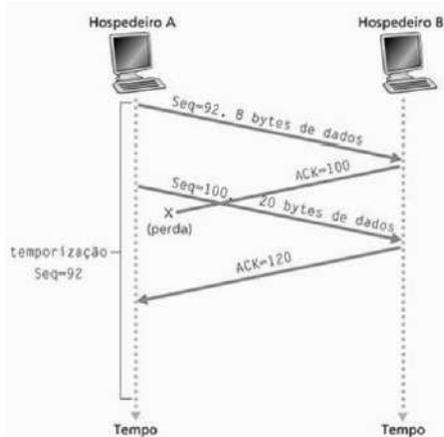


21

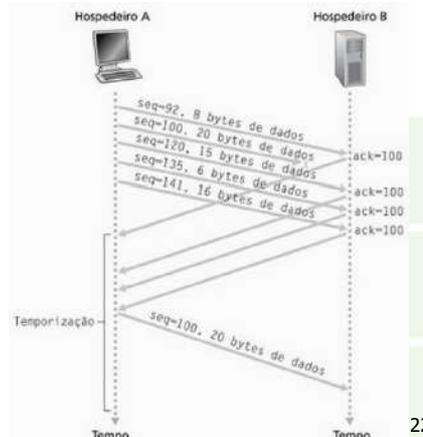


TCP – Transferência de Dados

■ Reconhecimento acumulativo

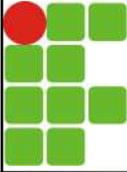


■ Retransmissão rápida



22

22

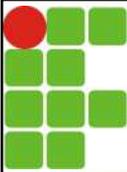


Controle de Congestionamento

- Controle de fluxo x Controle de congestionamento
- Controle de fluxo
 - Eliminar a possibilidade do remetente saturar o buffer do destinatário
 - Serviço de compatibilização de velocidades: taxa de envio com taxa de recepção
- Controle de congestionamento
 - Evitar saturação do canal de comunicação
- Os dois controles utilizam a mesma abordagem
 - Controle do remetente

23

23

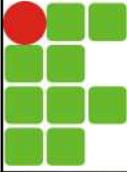


Controle de Congestionamento

- O volume de segmentos transmitidos é ajustado **dinamicamente** através da análise de vários fatores:
 - Tempo médio para o recebimento de ACK's
 - Quantidade de retransmissões
 - Valor do campo "janela de recepção" do segmento
 - ...
- Os ajustes são feitos para que a origem e o destino da conexão não fiquem:
 - **Nem ociosos** (esperando por segmentos)
 - **Nem saturados** (recebimento excessivo de segmentos)

24

24

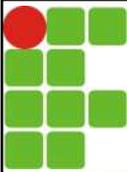


Controle de Congestionamento

- Congestionamento
 - Um dos TOP 10 problemas na Internet
 - Definição informal : "muitas fontes enviando dados acima da capacidade da rede de tratá-los"
 - Sintomas
 - Perda de pacotes (saturação de buffer dos roteadores)
 - Atrasos grandes (fila nos buffers dos roteadores)

25

25



Controle de Congestionamento

- Abordagens gerais para o problema de controle de congestionamento:
 - Controle de congestionamento **assistido pela rede**
 - Controle de congestionamento **fim a fim**

26

26

Controle de Congestionamento

- Controle de congestionamento assistido pela rede
 - Camada de rede (roteadores) realimenta o remetente sobre o estado de congestionamento da rede
 - A realimentação pode ser simples (envio de um bit indicando congestionamento) ou sofisticada (envio da taxa de transmissão que um transmissor suporta)
 - Modos de realimentação :
 - Pelo destinatário
 - Pela rede



27

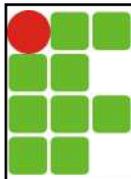
27

Controle de Congestionamento

- Controle de congestionamento fim a fim
 - Não usa realimentação explícita da rede
 - Congestionamento é inferido a partir de observações (ex.: perdas e atrasos) realizadas pelos sistemas finais

28

28

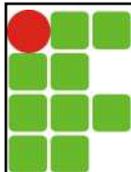


TCP – Controle de Congestionamento

- TCP utiliza **controle de congestionamento fim a fim**
- Obriga o remetente a **limitar sua taxa de envio como uma função do congestionamento de rede percebido**
 - Remetente percebe pouco congestionamento -> aumentar taxa
 - Remetente percebe congestionamento -> reduzir taxa
- Questionamentos:
 - (1) Como o remetente limita sua taxa?
 - (2) Como o remetente percebe um congestionamento?
 - (3) Qual algoritmo utilizado pelo remetente para modificar sua taxa de envio como uma função do congestionamento fim a fim?

29

29

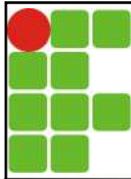


TCP – Controle de Congestionamento

- (1) Como o remetente limita sua taxa?
 - Através do **monitoramento da variável adicional janela de congestionamento**
- (2) Como o remetente percebe um congestionamento ?
 - **Ocorrência de um "evento de perda"**
 - Variável janela de congestionamento é modificada
 - **Eventos de perda:**
 - Esgotamento de temporização
 - Recebimento de 3 ACK's duplicados do destinatário

30

30

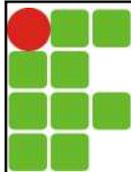


TCP – Controle de Congestionamento

- (3) Qual o algoritmo utilizado pelo remetente para modificar sua taxa como uma função do congestionamento?
 - O algoritmo deve se preocupar para que **não ocorra ociosidade e/ou saturação**
 - Princípios do algoritmo:
 - Um segmento perdido implica em congestionamento, então a taxa deve diminuir
 - Um segmento reconhecido implica que esta ocorrendo entrega, então a taxa deve aumentar
 - Busca por largura de banda: taxa é aumentada até que ocorra um evento de perda, nesse momento a taxa é diminuída

31

31

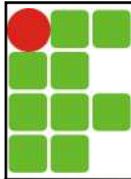


TCP – Controle de Congestionamento

- Componentes principais do algoritmo
 - Partida lenta (**obrigatório**)
 - Prevenção de congestionamento (**obrigatório**)
 - Recuperação rápida

32

32



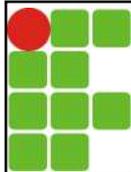
TCP – Controle de Congestionamento

■ Partida lenta

- Inicia a transmissão com um baixo volume de segmentos
- A cada ACK aumenta exponencialmente o volume
- “Evento de perda” = Esgotamento de temporização
 - Reinicia a partida lenta
 - Cria nova variável: $ssthresh = (janela\ de\ congestionamento)/2$
 - Novo início: **ssthresh atingido -> prevenção de congestionamento**
- “Evento de perda” = 3 ACK’s duplicados
 - -> recuperação rápida

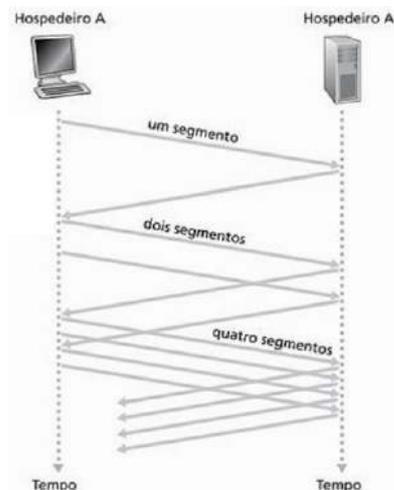
33

33



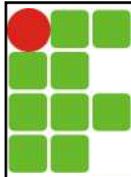
TCP – Controle de Congestionamento

■ Partida lenta



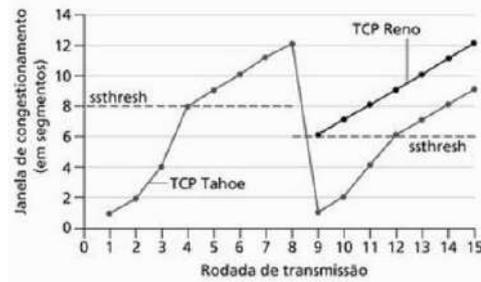
34

34



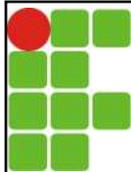
TCP – Controle de Congestionamento

- Prevenção de congestionamento
 - Aumento do volume de forma linear
- Recuperação rápida
 - Modifica o ponto de reinício da transmissão
 - Evita reinício da partida lenta



35

35

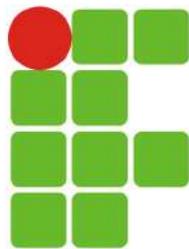


Referências

- KUROSE, J. F. e ROSS, K. - Redes de Computadores e a Internet - 5a Ed., Pearson, 2010.
- TANENBAUM, A. S. - Redes de Computadores - 5a Ed., Pearson, 2011.

36

36



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE



Introdução às Redes de Computadores

Turma : 20192.1.01405.1N

Camada de Transporte - Parte II

Prof. Thiago Dutra <thiago.dutra@ifrn.edu.br>