

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

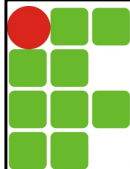


Infraestrutura de Redes de Computadores

Turma : TMS – 20171.3.01112.1M

Camada de Transporte

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



Agenda

- Introdução
- Protocolos de Transporte Internet
- Multiplexação e Demultiplexação
- UDP
- TCP
- Controle de Congestionamento

Introdução

- Pilha TCP/IP

5. Aplicação	
4. Transporte	Pacote = segmento
3. Rede	
2. Enlace	
1. Física	

3

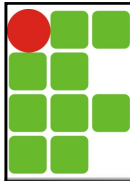
Introdução

- Camada de Transporte
 - Situada entre a camada de aplicação e a camada de rede
 - Tem o papel fundamental de fornecer serviços de comunicação diretamente aos processos de aplicação

```

graph TD
    A[Aplicação] <--> T[Transporte]
    T <--> R[Rede]
    R <--> E[Enlace]
    E <--> F[Físico]
  
```

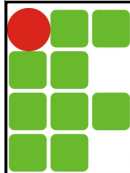
4



Introdução

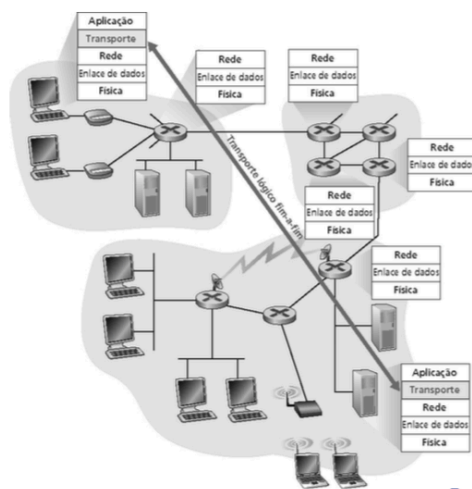
- Um protocolo de camada de transporte fornece **comunicação lógica** entre processos de aplicação que rodam em hospedeiros diferentes
 - Nesse contexto, comunicação lógica significa que: do ponto de vista da aplicação **é como se os hospedeiros que rodam os processos estivessem diretamente conectados**
 - Na verdade **podem existir diversos roteadores e vários tipos de enlace** entre estes hospedeiros
 - Usando a comunicação lógica oferecida na camada de transporte, as aplicações trocam mensagens **livres da preocupação com a real infraestrutura física** utilizada para entrega dessas mensagens

5



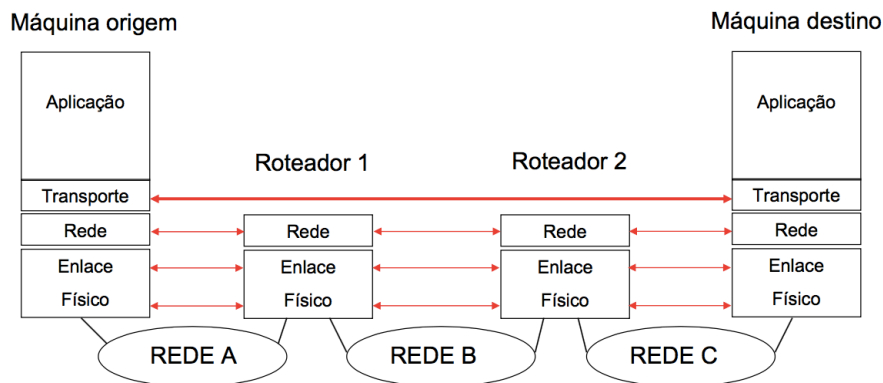
Introdução

- **Comunicação Lógica**
 - Entidades de transporte da máquina de origem e de destino comunicam-se **diretamente**, de forma **independente a todos os sistemas intermediários** existentes entre elas
 - Nos níveis físico, enlace e rede isto não é possível



Introdução

■ Comunicação Lógica



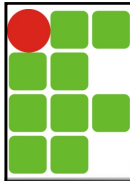
Introdução

■ Os protocolos de transporte são executados apenas nos sistemas finais

■ Facilita a implementação

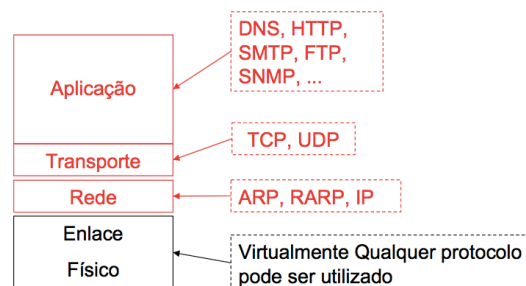
- Emissor : quebra as mensagens da aplicação em segmentos e envia para a camada de rede
- Receptor : remonta os segmentos em mensagens e passa para a camada de aplicação

■ Roteadores (ou qualquer outro sistema intermediário) não necessitam implementar protocolos de transporte

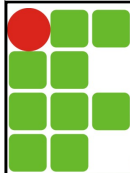


Protocolos de Transporte Internet

- O modelo Internet TCP/IP padroniza dois protocolos de transporte :
 - **TCP** (Transmission Control Protocol)
 - **UDP** (User Datagram Protocol)



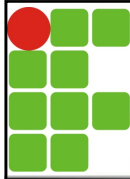
9



Protocolos de Transporte Internet

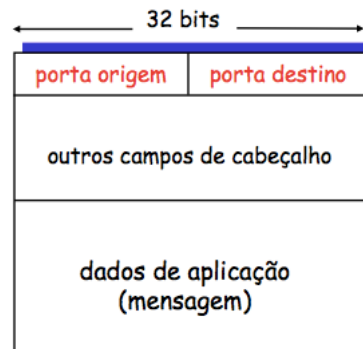
- **TCP x UDP**
 - Completamente diferentes em relação à:
 - Complexidade
 - Conjunto de funcionalidades
 - Aplicações usuárias
 - São semelhantes em relação à :
 - Fornecimento de verificação de integridade
 - Multiplexação de requisições e Demultiplexação das respostas
 - Operações ligadas ao conceito de "Porta"

10

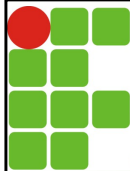


Protocolos de Transporte Internet

- Formato geral de um segmento TCP ou UDP



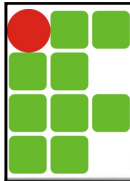
11



Multiplexação e Demultiplexação

- Cenário : usuário em um computador vendo páginas Web, transferindo arquivos via FTP e com dois terminais Telnet abertos.
 - 4 processos : HTTP, FTP e 2x Telnet
 - Quando a camada de transporte recebe dados da camada de rede abaixo dela, como direcionar esses dados para o processo correto ?

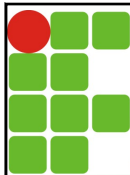
12



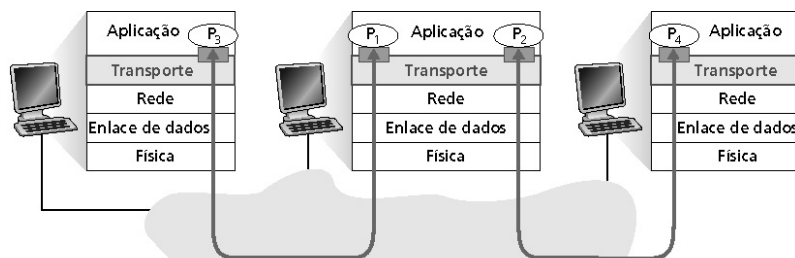
Multiplexação e Demultiplexação

- Multiplexação
 - Ocorre no hospedeiro emissor
 - Coleta dados de múltiplos sockets, encapsula os dados com cabeçalho (usado na demultiplexação) e envia os segmentos para a camada de rede
- Demultiplexação
 - Ocorre no hospedeiro receptor
 - Recebe o segmento da camada de rede, identifica a porta receptora e direciona ao socket associado

13



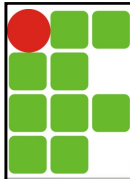
Multiplexação e Demultiplexação



Legenda:

○ Processo ■ Socket

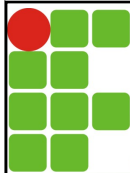
14



Multiplexação e Demultiplexação

- Para que as aplicações que estejam rodando em uma mesma máquina possam transmitir e receber dados simultaneamente elas utilizam “**portas**”
 - Portas são associadas aos **sockets** (canais por onde os dados passam da rede para o processo e vice-versa)
 - O hospedeiro usa **endereços IP e números de porta para direcionar o segmento ao socket apropriado**
 - Socket UDP
 - (IP de destino, PORTA de destino)
 - Socket TCP
 - (IP de origem, PORTA de origem, IP de destino, PORTA de destino)

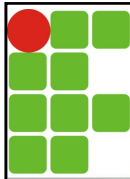
15



Multiplexação e Demultiplexação

- Cada identificador de **porta** possui **16 bits** de comprimento, podendo variar de **0 a 65535**
- Portas de origem e destino são selecionadas aleatoriamente para uso pelo TCP e UDP
 - Na prática, **aplicações “comuns” possuem identificadores (número) de porta fixos**
 - **22** : SSH
 - **25** : SMTP
 - **53** : DNS
 - **80** : HTTP
 - **443** : HTTPS

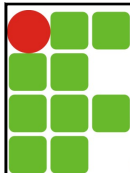
16



Multiplexação e Demultiplexação

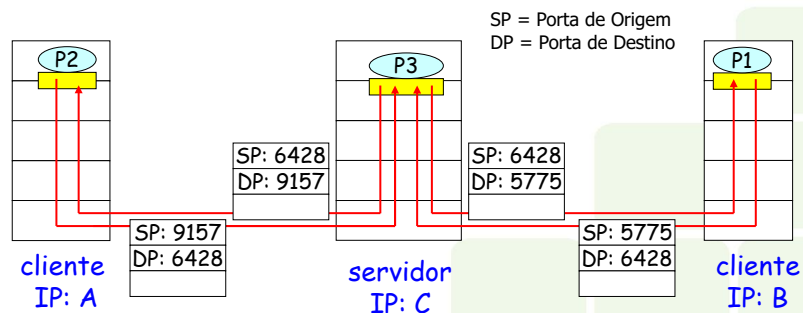
- Essas **portas bem conhecidas** também são chamadas de “portas baixas”
 - Normalmente estão **abaixo de 1024** (0 a 1023)
 - [RFC 1700] – 1994 (<https://tools.ietf.org/html/rfc1700>)
 - O controle de atribuição atual de portas é feita por uma instituição chamada **IANA** (Internet Assigned Numbers Authority)
 - [RFC 3232] – 2002 (<https://tools.ietf.org/html/rfc3232>)
 - <http://www.iana.org/assignments/port-numbers>

17



Multiplexação e Demultiplexação

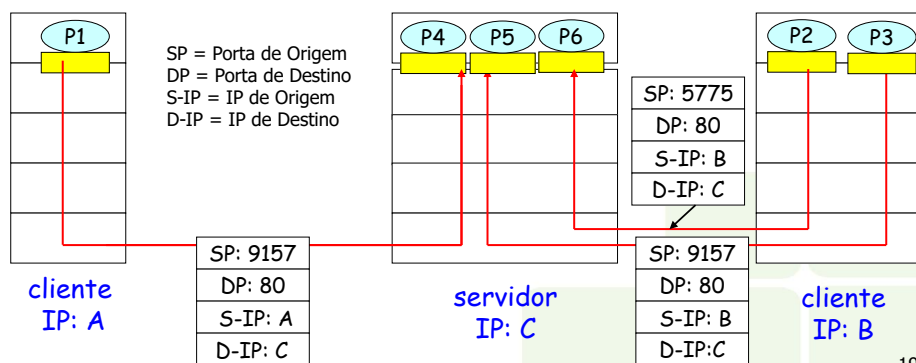
- Demux UDP
 - Datagramas com IP de origem diferentes e/ou portas de origem diferentes são direcionados para o mesmo socket



18

Multiplexação e Demultiplexação

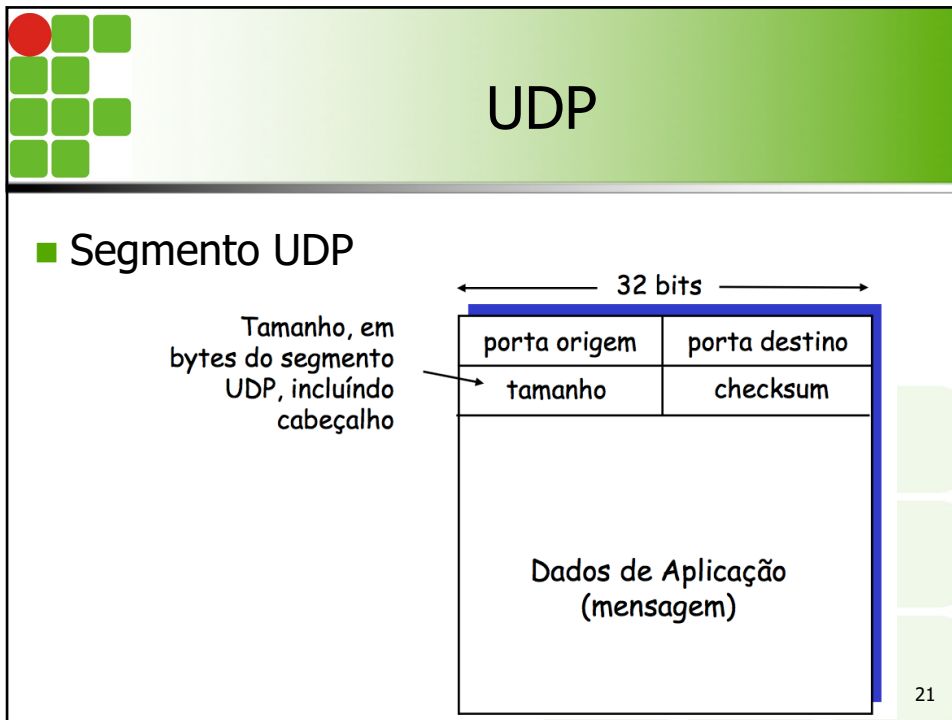
■ Demux TCP



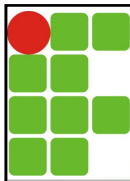
UDP

■ User Datagram Protocol

- Protocolo de Datagrama de Usuário
- [RFC 768] – 1980
 - <https://tools.ietf.org/html/rfc768>
- Oferece um serviço “**best effort**”
 - Melhor esforço
 - Faz o melhor possível, mas não dá garantias
 - Serviço **não confiável**



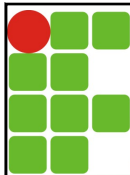
- # UDP
- Não é orientado para conexão
 - Não existe apresentação entre o UDP transmissor e o receptor
 - Cada segmento UDP é tratado de forma totalmente independente
 - Uma aplicação que usa UDP pode ter parte do seu fluxo de dados entre origem e destino :
 - Perdido, Chegando fora de ordem, Chegando com erros
 - O tratamento dessas situações deve ser realizado, se desejado, na própria aplicação
- 22



UDP

- Como justificar a existência do UDP ?
 - Não há estabelecimento de conexão (que possa redundar em atrasos)
 - Simples: não há estado de conexão nem no transmissor, nem no receptor
 - Cabeçalho de segmento reduzido (8 bytes)
 - Não há controle de congestionamento: UDP pode enviar segmentos tão rápido quanto desejado (e possível)

23



UDP

- Utilizam UDP :
 - Aplicações onde o volume de dados é pequeno
 - DNS
 - Aplicações que não exigem alta confiabilidade
 - Transmissão de vídeo e áudio

Servidor remoto de arquivo	NFS	tipicamente UDP
Recepção de multimídia	tipicamente proprietária	UDP ou TCP
Telefonia por Internet	tipicamente proprietária	UDP ou TCP
Gerenciamento de rede	SNMP	tipicamente UDP
Protocolo de roteamento	RIP	tipicamente UDP
Tradução de nome	DNS	tipicamente UDP

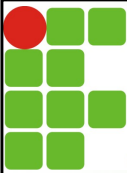
24



Agenda TCP

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

25



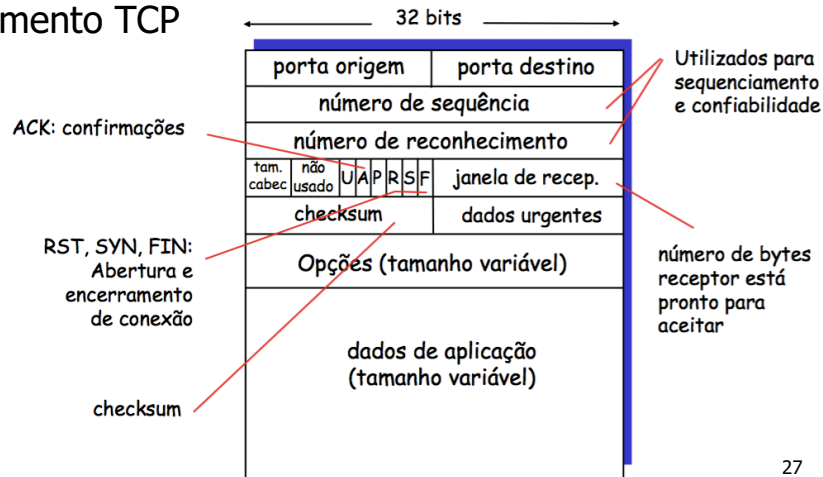
TCP - Introdução

- **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

26

TCP - Introdução

■ Segmento TCP

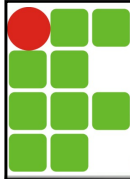


27

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

28

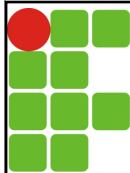


TCP – Estabelecendo uma Conexão

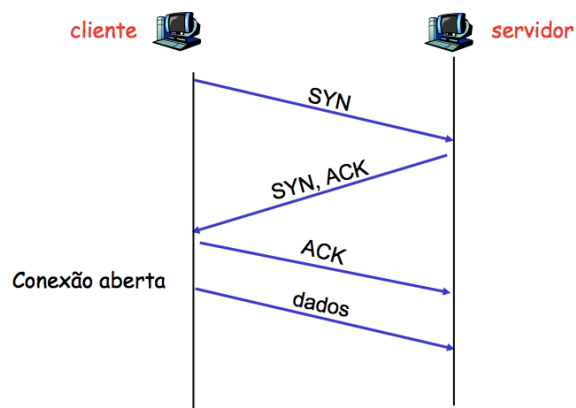
■ 3-Way Handshake

- (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

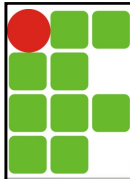
29



TCP – Estabelecendo uma Conexão



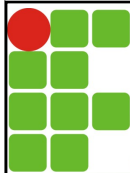
30



TCP – Encerrando uma Conexão

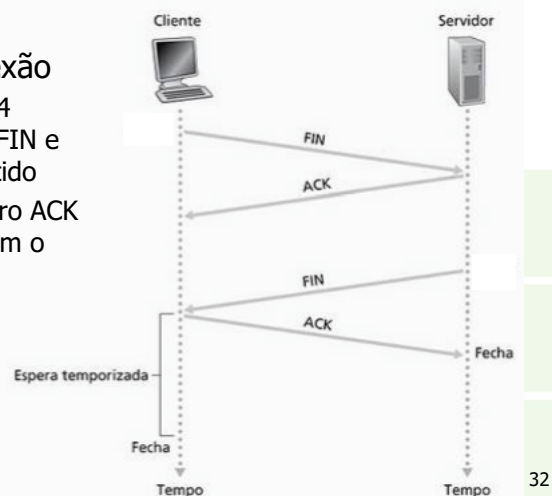
- Tanto o cliente quanto o servidor podem encerrar a conexão
 - Quando uma conexão é encerrada os recursos associados são liberados
- Quem deseja fechar a conexão inicia o processo :
 - O primeiro envia segmento TCP com o bit **FIN** habilitado
 - O outro responde com um **ACK** e envia um **FIN**
 -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

31

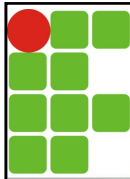


TCP – Encerrando uma Conexão

- 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



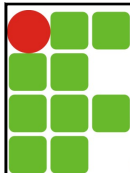
32



Transferência de Dados Confiável

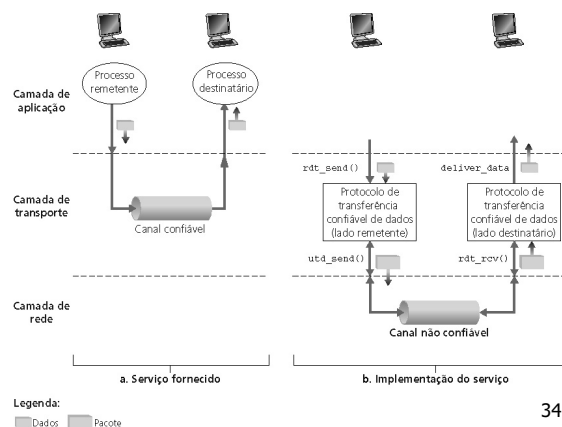
- 1º dentre os TOP 10 problemas fundamentalmente importantes para o trabalho em rede.
- O serviço fornecido para as camadas superiores por um **protocolo de transferência de dados confiável** deve ser um **canal confiável** de comunicação
- Em um canal confiável os dados transferidos:
 - Não são corrompidos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

33

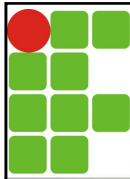


Transferência de Dados Confiável

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



34

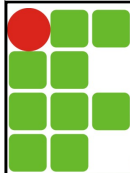


Transferência de Dados Confiável

■ Resumo de mecanismos de transferência confiável de dados

Mecanismo	Utilização, Comentários
Soma de verificação	Usada 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 determinada 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.

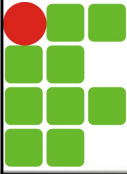
35



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

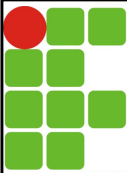
36



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)

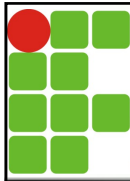
37



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)

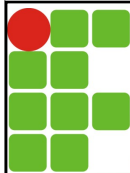
38



Controle de Congestionamento

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

39

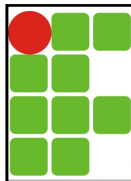


Controle de Congestionamento

- 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



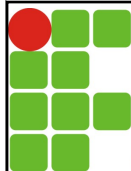
40



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

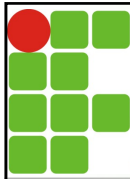
41



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?

42

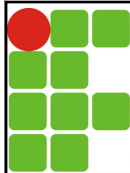


TCP – Controle de Congestionamento

- (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

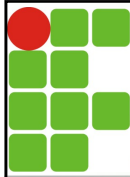
43



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**

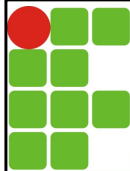
44



TCP – Controle de Congestionamento

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

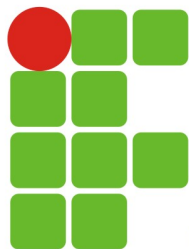
45



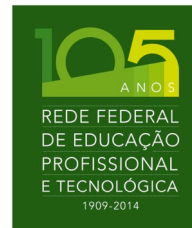
Referências

- KUROSE, J. F. e ROSS, K. - **Redes de Computadores e a Internet** – 6a Ed., Pearson, 2013.
- 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.
- ELIAS, G. e LOBATO, L. C. – **Arquitetura e protocolo de rede TCP-IP** – 2a Ed., RNP/ESR, 2013
- IANA, **Internet Assigned Numbers Authority**, <https://www.iana.org/>

46



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



Infraestrutura de Redes de Computadores

Turma : TMS – 20171.3.01112.1M

Camada de Transporte

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