

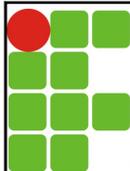
INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
RIO GRANDE DO NORTE



# Redes de Computadores

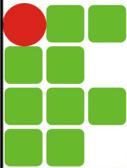
## Camada de Transporte - Parte I

Prof. Thiago Dutra <[thiago.dutra@ifm.edu.br](mailto:thiago.dutra@ifm.edu.br)>



## Agenda

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

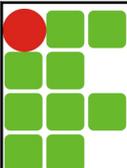


# Introdução

- Pilha TCP/IP

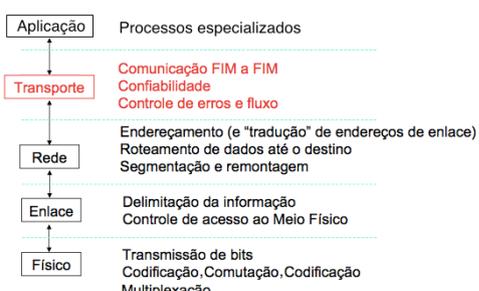
5. Aplicação
4. Transporte
3. Rede
2. Enlace
1. Física

3



# Introdução

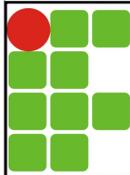
- Camada de Transporte
  - Situada entre a camada de aplicação e a camada de rede
  - Tem o papel fundamental de fornecer serviços de comunicação diretamente aos processos de aplicação



```

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

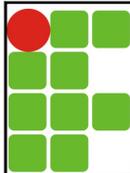
4



# Introdução

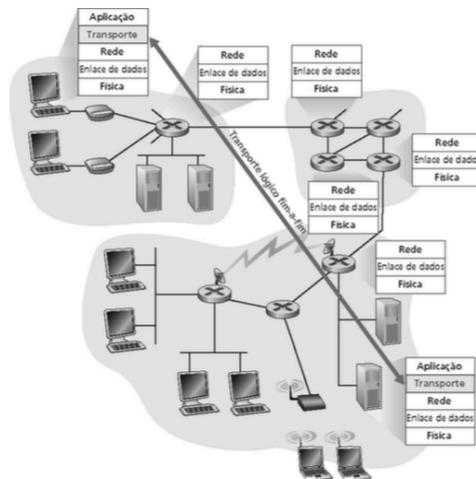
- Um protocolo de camada de transporte fornece **comunicação lógica** entre processos de aplicação que rodam em hospedeiros diferentes
  - Nesse contexto, comunicação lógica significa que: do ponto de vista da aplicação **é como se os hospedeiros que rodam os processos estivessem diretamente conectados**
    - 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**

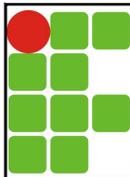
5



# Introdução

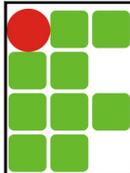
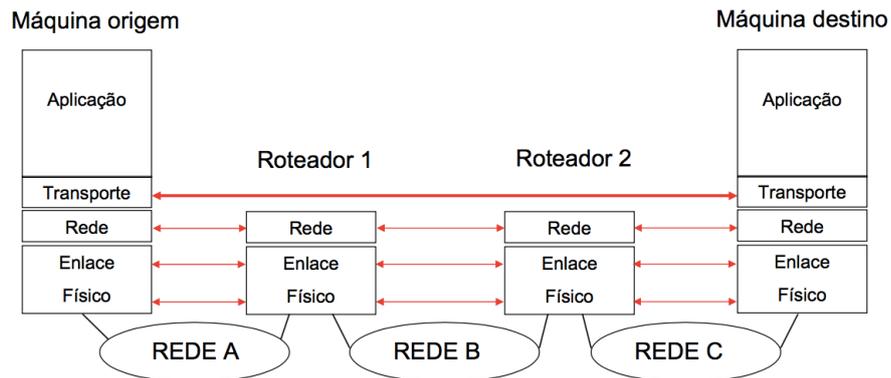
- **Comunicação Lógica**
  - Entidades de transporte da máquina de origem e de destino comunicam-se **diretamente**, de forma **independente a todos os sistemas intermediários** existentes entre elas
  - Nos níveis físico, enlace e rede isto não é possível





# Introdução

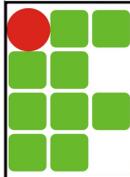
## ■ Comunicação Lógica



# Introdução

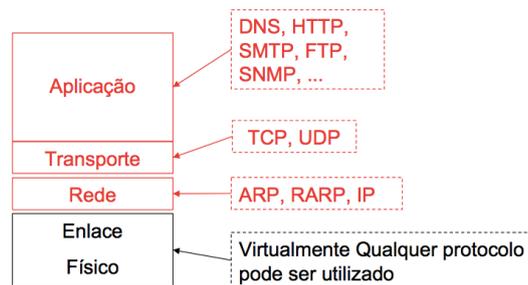
## ■ Os protocolos de transporte são executados apenas nos sistemas finais

- 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**

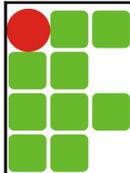


## Protocolos de Transporte Internet

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



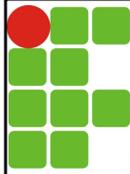
9



## Protocolos de Transporte Internet

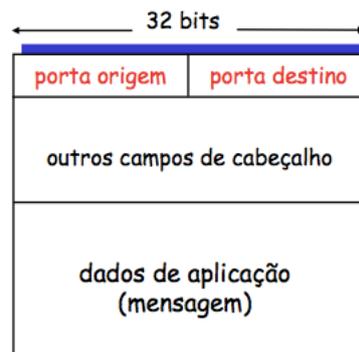
- TCP x UDP
  - Completamente diferentes em relação à:
    - Complexidade
    - Conjunto de funcionalidades
    - Aplicações usuárias
  - São semelhantes em relação à :
    - Fornecimento de verificação de integridade
    - Multiplexação de requisições e Demultiplexação das respostas
      - Operações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

10

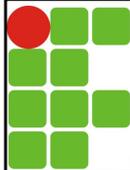


## Protocolos de Transporte Internet

- Formato geral de um segmento TCP ou UDP



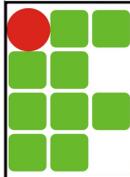
11



## 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 ?

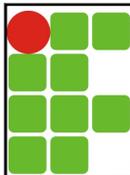
12



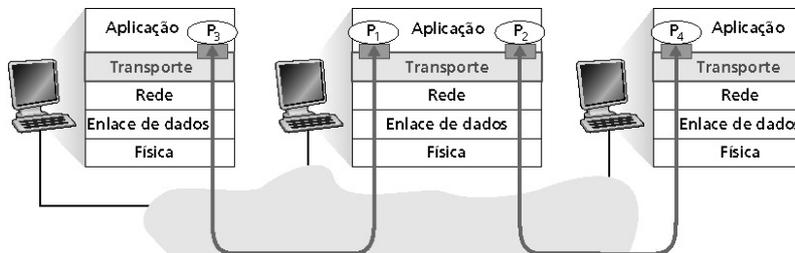
## Multiplexação e Demultiplexação

- 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



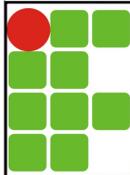
## Multiplexação e Demultiplexação



Legenda:

○ Processo ■ Socket

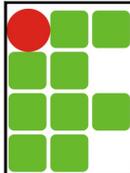
14



## 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)

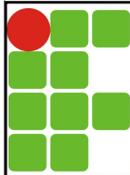
15



## 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

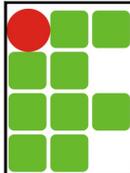
16



## Multiplexação e Demultiplexação

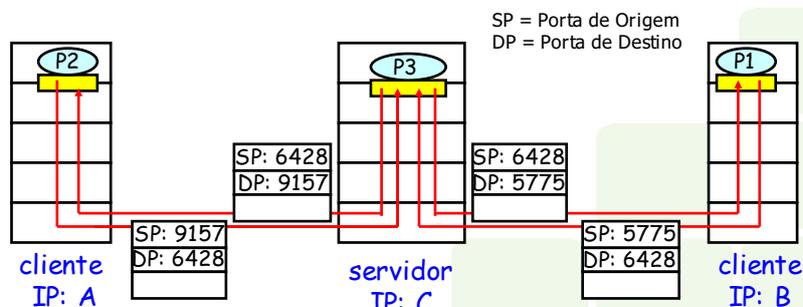
- Essas **portas bem conhecidas** também são chamadas de "portas baixas"
  - Normalmente estão **abaixo de 1024** (0 a 1023)
    - [RFC 1700] – 1994 (<https://tools.ietf.org/html/rfc1700>)
  - O controle de atribuição atual de portas é feita por uma instituição chamada **IANA** (Internet Assigned Numbers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

17

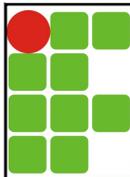


## Multiplexação e Demultiplexação

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

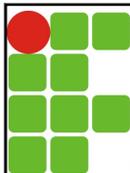
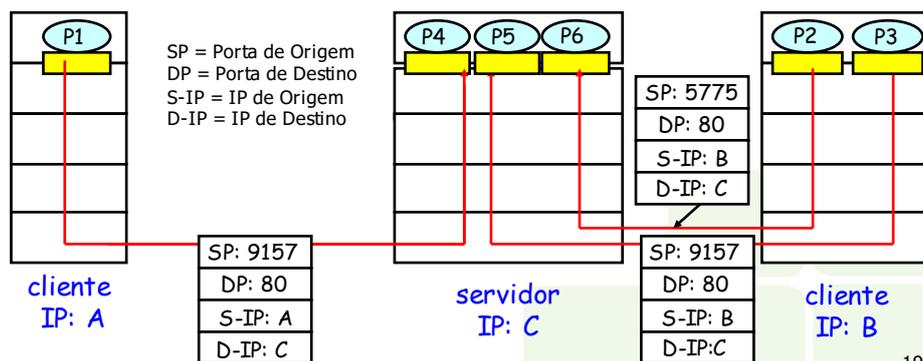


18



# Multiplexação e Demultiplexação

## ■ Demux TCP

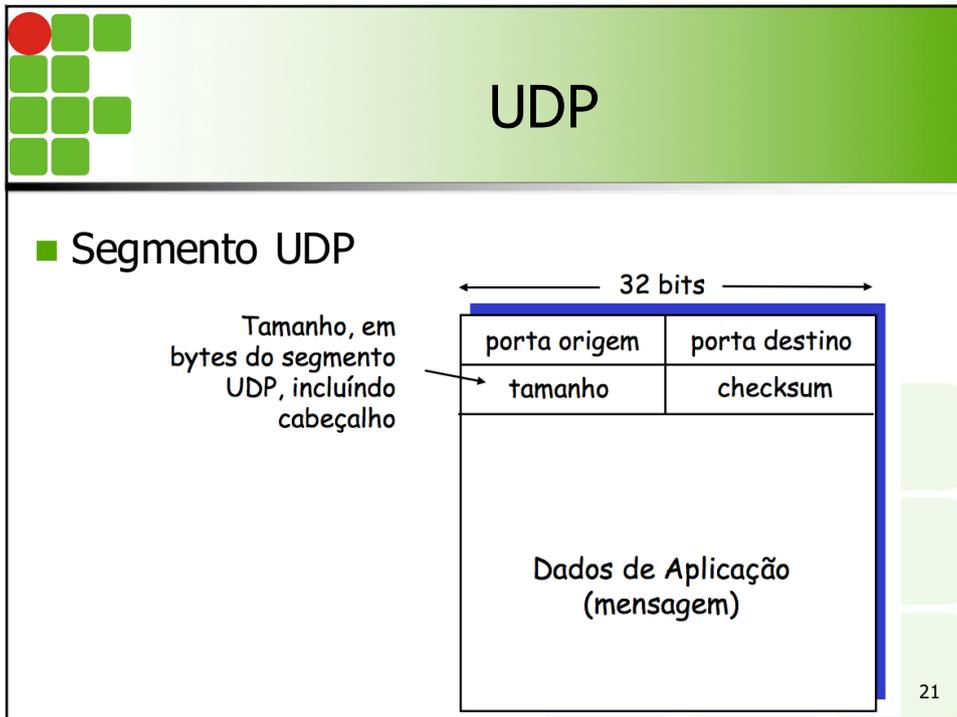


# UDP

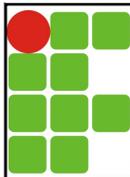
## ■ User Datagram Protocol

- 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**

20



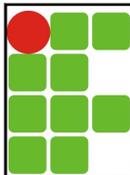
- # UDP
- Não é orientado para conexão
    - Não existe apresentação entre o UDP transmissor e o receptor
    - Cada segmento UDP é tratado de forma totalmente independente
  - Uma aplicação que usa UDP pode ter parte do seu fluxo de dados entre origem e destino :
    - Perdido, Chegando fora de ordem, 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

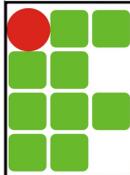


## UDP

- 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

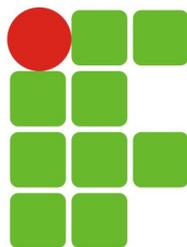
24



## 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/>

25



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



## Redes de Computadores

### Camada de Transporte - Parte I

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