

**INSTITUTO  
FEDERAL**  
Pará

# Redes de Computadores



Ricardo José Cabeça de Souza

[ricardo.souza@ifpa.com.br](mailto:ricardo.souza@ifpa.com.br)

# Camada de Transporte

- **Camada de Transporte**
  - Responsável pela comunicação entre processos
    - Comunicação nó-a-nó (fim-a-fim)
    - Processo
      - É um programa aplicativo em execução em um host
  - Paradigma cliente/servidor
    - Um processo no host local, denominado cliente, solicita serviços de outro processo, normalmente localizado em um host remoto, denominado servidor
  - Comunicação entre processos envolve:
    - Host local
    - Processo local
    - Host remoto
    - Processo remoto

- **Camada de Transporte**





# Camada de Transporte

- Camada de Transporte

## REQUISITOS DE APLICAÇÕES DE REDE SELECIONADAS

Aplicação	Perda de dados	Vazão	Sensibilidade ao tempo
Transferência / download de arquivo	Sem perda	Elástica	Não
E-mail	Sem perda	Elástica	Não
Documentos Web	Sem perda	Elástica (alguns kbits/s)	Não
Telefonia via Internet/ videoconferência	Tolerante à perda	Áudio: alguns kbits/s – 1Mbit/s Vídeo: 10 kbits/s – 5 Mbits/s	Sim: décimos de segundo
Áudio/vídeo armazenado	Tolerante à perda	Igual acima	Sim: alguns segundos
Jogos interativos	Tolerante à perda	Poucos kbits/s – 10 kbits/s	Sim: décimos de segundo
Mensagem instantânea	Sem perda	Elástico	Sim e não

# Camada de Transporte

- **Endereçamento**

- Uso do endereço na camada de transporte
- Denominado **número de porta**
- Define os processos em execução no host
- O número da porta de destino é necessário para entrega
- O número da porta de origem é necessário para resposta

# Camada de Transporte

- **Endereçamento**

- Modelo Internet os números de porta são inteiros de 16 bits
- Variam de 0 a 65.535
- Cliente define para si um número de porta de forma aleatória
  - Denominado **número de porta efêmero**

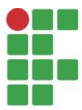
EFÊMERO: **1** Que dura um só dia. **2** Passageiro, transitório.

Fonte: Michaelis On-Line.

# Camada de Transporte

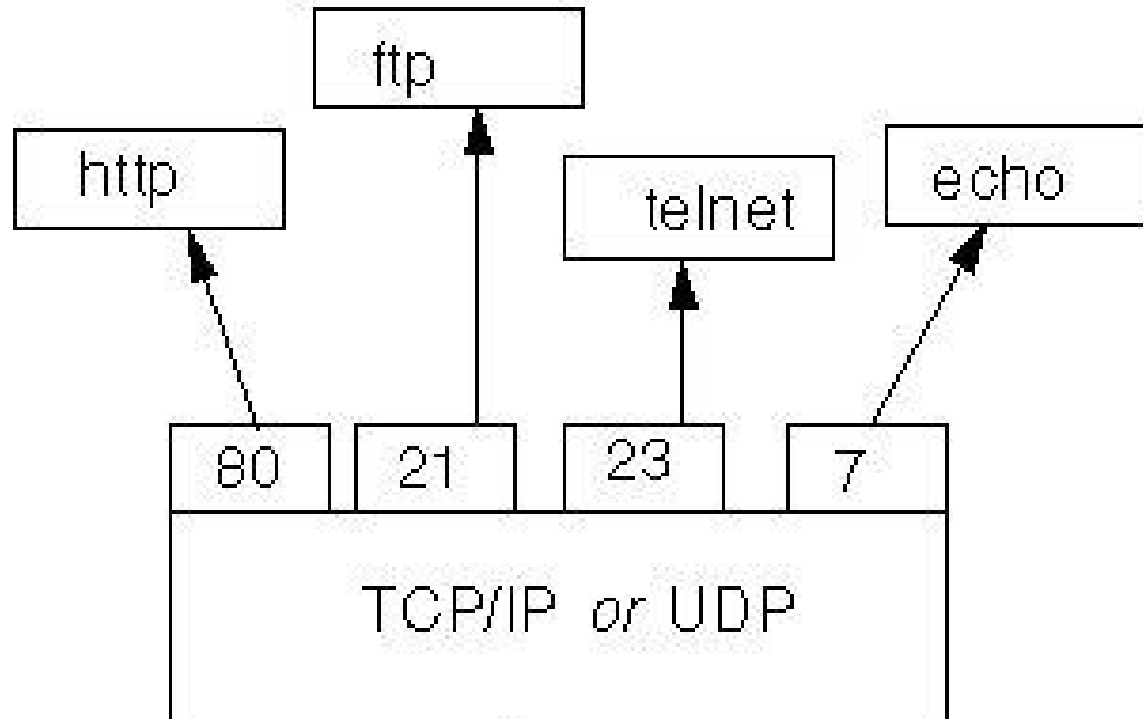
- **Endereçamento**

- Processo no servidor deve definir um número de porta
- No servidor não é escolhido aleatoriamente
- Implementação de números de portas universais para servidores
- Denominados **número de porta conhecidos (well-known port numbers)**



# Camada de Transporte

- Endereçamento







# Camada de Transporte

- Endereçamento

*Popular Applications and Their Well-Known Port Numbers*

Port Number	Protocol	Application
20	TCP	FTP data
21	TCP	FTP control
23	TCP	Telnet
25	TCP	SMTP
53	UDP, TCP	DNS
67, 68	UDP	DHCP
69	UDP	TFTP
80	TCP	HTTP (WWW)
110	TCP	POP3
161	UDP	SNMP

# Camada de Transporte

- **Endereçamento**
  - Faixa de endereços IANA (Internet Assigned Number Authority)
  - Portas conhecidas
    - Atribuídas e controladas pelo IANA
    - De 0 a 1023
  - Portas registradas
    - Podem ser registradas na IANA
    - De 1024 a 49151
  - Portas dinâmicas
    - Podem ser usadas por qualquer processo
    - De 49152 a 65535
- Lista de todas as portas

<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>

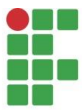
# Camada de Transporte

- **Endereçamento IP x Porta**
  - **IP** seleciona o host
  - **Porta** seleciona o processo

# Camada de Transporte

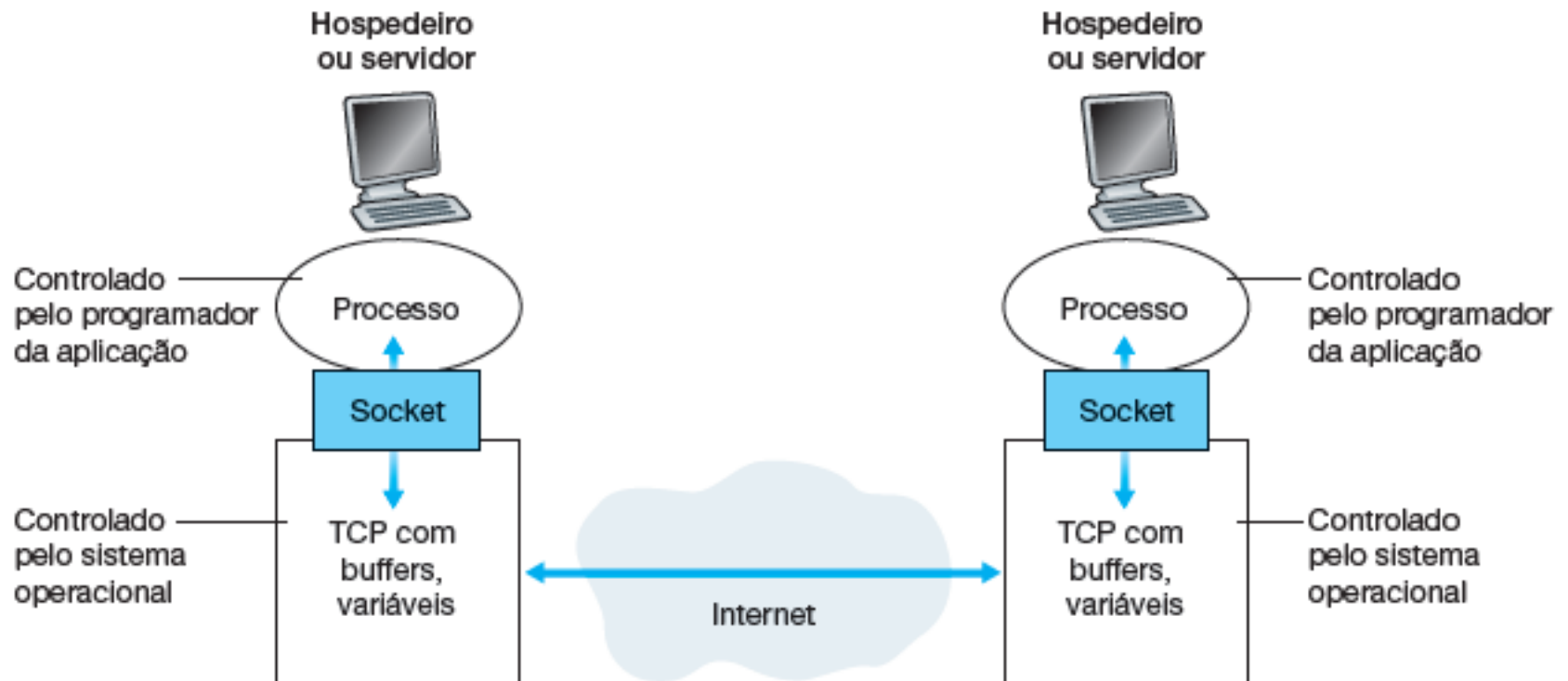
- **Endereço Socket**

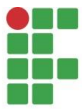
- Para estabelecer uma conexão e permitir a comunicação entre processos necessitamos de dois identificadores:
  - Endereço IP (Rede)
  - Número de Porta (Transporte)
- Essa combinação é chamada **endereço socket**
- Protocolo da camada de transporte precisa de um par de endereços socket, cliente e servidor
- Cabeçalho IP contém o endereço IP
- Cabeçalho UDP ou TCP contém o número das portas



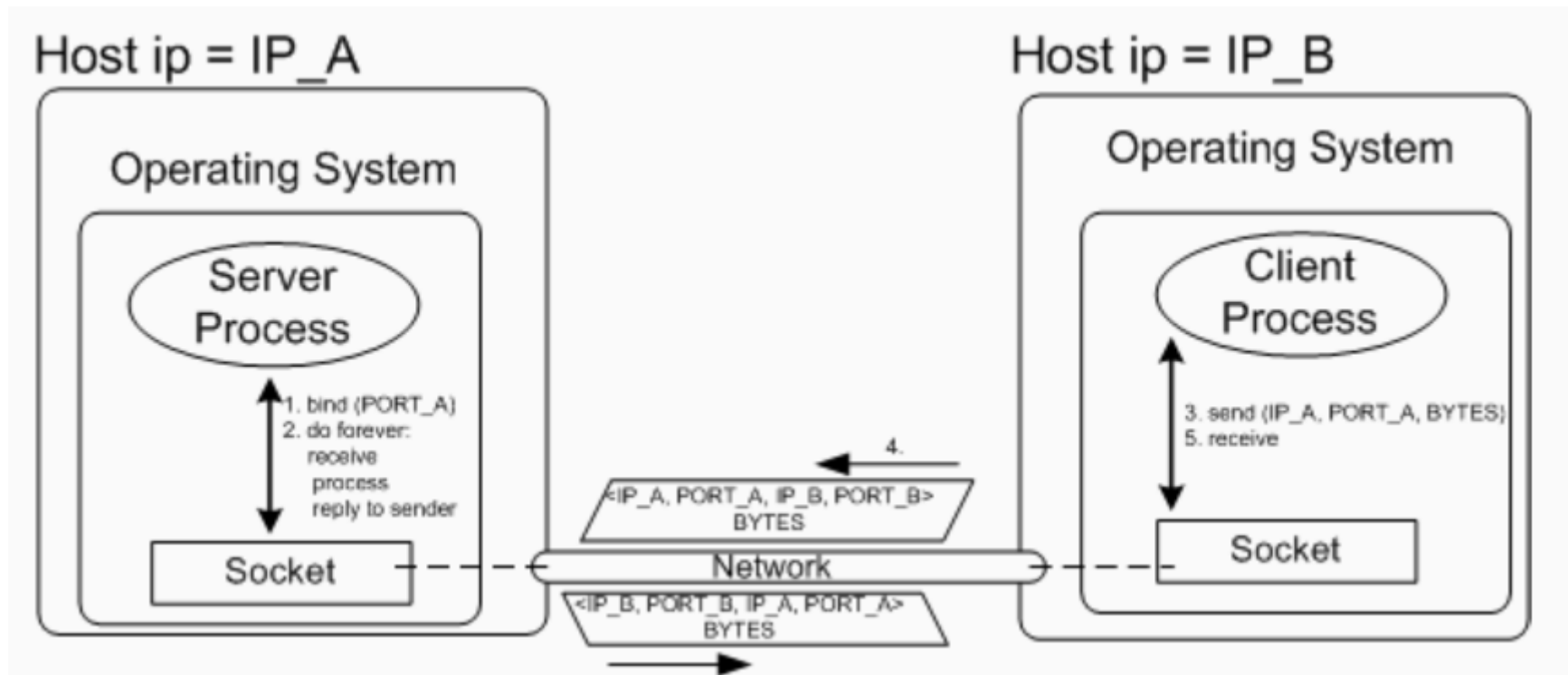
# Camada de Transporte

## PROCESSOS DE APLICAÇÃO, SOCKETS E PROTOCOLO DE TRANSPORTE SUBJACENTE





# Camada de Transporte



# Camada de Transporte

- **Multiplexação**

- Emissor

- Podem existir vários processos que precisam transmitir pacotes e há somente um protocolo de transporte em execução em dado instante
    - Protocolo de transporte aceita várias mensagens provenientes de diferentes processos, diferenciando-os pelos números de porta
    - Relação vários para um

# Camada de Transporte

- **Multiplexação**

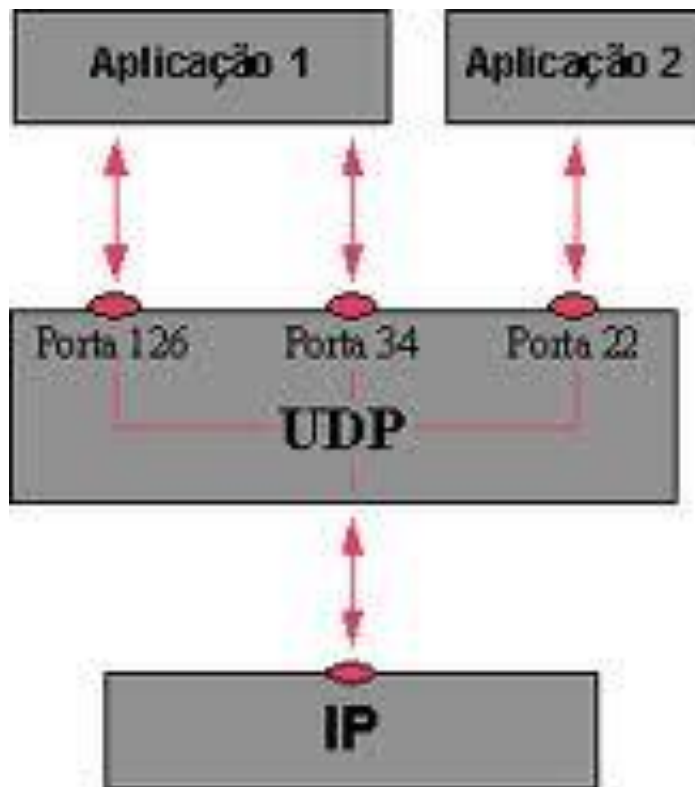
- Receptor

- Camada de transporte recebe os datagramas da camada de rede
    - Após verificar erros e eliminar o cabeçalho, a camada de transporte entrega cada mensagem para o processo usuário apropriado
    - Relação um para vários



# Camada de Transporte

- Multiplexação



# Camada de Transporte

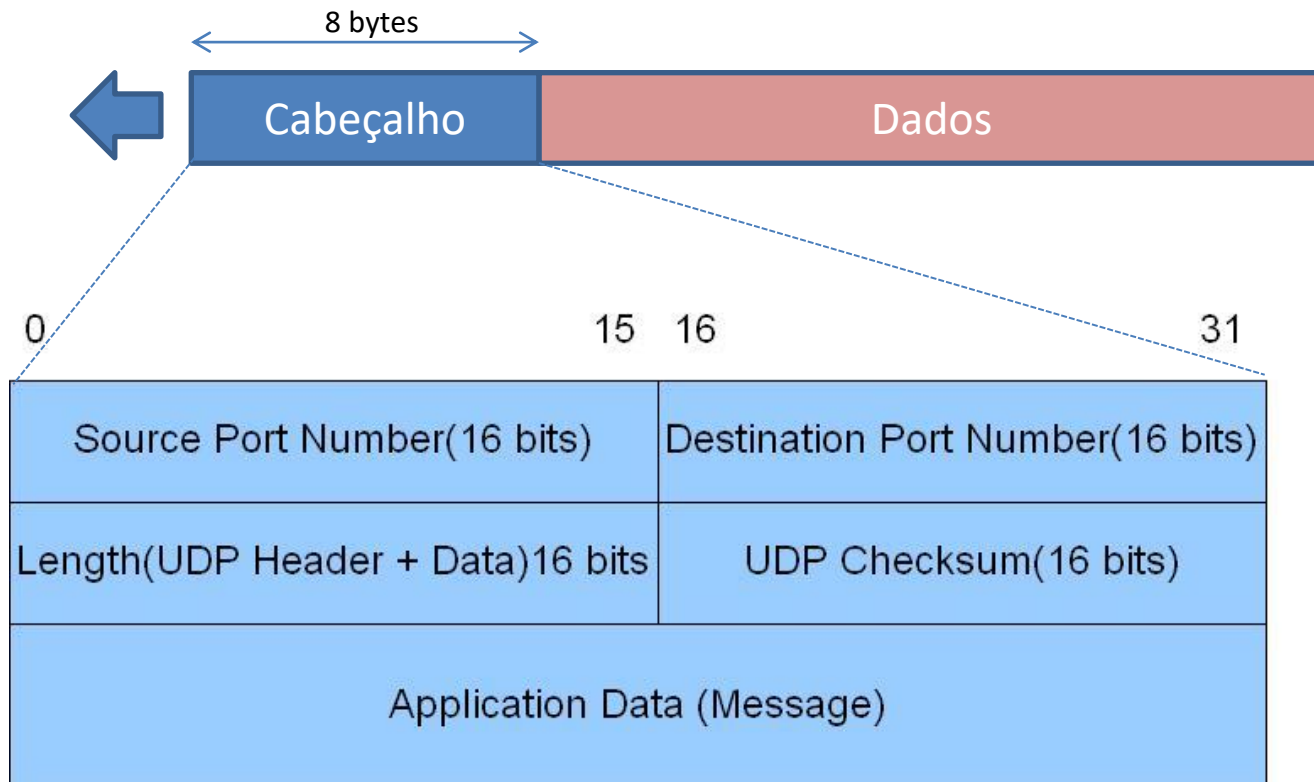
- **Protocolos Camada de Transporte**
  - UDP (User Datagram Protocol)
  - TCP (Transmission Control Protocol)
  - SCTP (Stream Control Transmission Protocol)

# Camada de Transporte

- **UDP (User Datagram Protocol)**
  - Protocolo de transporte sem conexão e não confiável
  - Implementa a comunicação entre processos
  - Simples e mínimo de overhead
  - Verificação de erro limitada
  - Pacotes denominados **Datagrama de Usuário**

# Camada de Transporte

- UDP (User Datagram Protocol)



# Camada de Transporte

- **UDP (User Datagram Protocol)**
  - Porta Origem
    - Número porta processo origem
    - 16 bits de comprimento
    - Varia de 0 a 65.535
    - Número efêmero e escolhido randomicamente(cliente)
    - Número conhecido (servidor)
  - Porta Destino
    - Número porta processo destino
    - 16 bits de comprimento
    - Número efêmero e escolhido randomicamente(cliente)
    - Número conhecido (servidor)

# Camada de Transporte

- **UDP (User Datagram Protocol)**

- Comprimento

- 16 bits
    - Define o comprimento total de um datagrama UDP
    - Cabeçalho + Dados
    - Comprimento entre 0 a 65.535 bytes

COMPRIMENTO DATAGRAMA UDP = COMPRIMENTO TOTA IP – COMPRIMENTO CABEÇALHO IP

- Checksum

- 16 bits
    - Usado para detectar erros na transmissão de datagramas UDP (cabeçalho + dados)
    - Uso opcional. Se não for usado será preenchido com bits = 1

# Camada de Transporte

- **UDP (User Datagram Protocol)**
  - Operação
    - Serviço sem conexão
    - Datagramas independentes
    - Não são numerados
    - Não pode entregar fluxo de dados contínuo
    - Informação deve ser pequena
    - Não confiável
    - Não implementa controle de fluxo nem controle de erros, exceto o checksum
    - Emissor não sabe se uma mensagem foi perdida ou duplicada

# Camada de Transporte

- **UDP (User Datagram Protocol)**

- Usos

- Processo que requeiram comunicação solicitação-resposta simples com pouca preocupação com controle de erros ou fluxo
    - Processos que implementem mecanismos internos de controle de erro ou fluxo
    - Indicado para multicast
    - Utilizado no gerenciamento de redes (SNMP)
    - Usado por alguns protocolos de roteamento como o RIP(Routing Information Protocol)



# Camada de Transporte

- **TCP (Transmission Control Protocol)**
  - Protocolo de comunicação entre processos finais
  - Usa número de portas
  - Orientado a conexão e confiável
    - Cria conexão virtual
  - Implementa mecanismos de controle de fluxo e de erros

# Camada de Transporte

- **TCP (Transmission Control Protocol)**
  - Comunicação entre processos utilizando portas
  - Serviço de entrega de fluxo de bytes
  - Buffer de transmissão
    - Área disponível
    - Bytes enviados e não confirmados
    - Bytes a enviar
  - Buffer de Recepção
    - Área disponível
    - Bytes recebidos pronto para leitura

# Camada de Transporte

- **TCP (Transmission Control Protocol)**
  - Segmentação
    - Cada segmento recebe um cabeçalho TCP
    - Segmentos são encapsulados em datagramas IP
  - Comunicação Full-Duplex
  - Serviço orientado a conexão
  - Serviço confiável
    - Implementa mecanismo de confirmação

# Camada de Transporte

- **TCP (Transmission Control Protocol)**
  - Sistema de Numeração – Número de bytes
    - Os bytes são numerados em cada conexão TCP
    - Começa com um número gerado randomicamente entre 0 e  $2^{32}-1$  como número inicial
    - Exemplo: Se o número randômico for 1.057 e o total de dados a serem transmitidos for de 6.000 bytes, os bytes serão numerados de 1.057 a 7.056

# Camada de Transporte

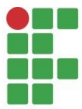
- **TCP (Transmission Control Protocol)**
  - Sistema de Numeração – Número de bytes
    - Número de sequência
      - Para cada segmento é o número do primeiro byte transportado
      - Exemplo: Transmitir 5.000 bytes e o primeiro byte recebe o número 10.001, enviados em 5 segmentos de 1.000 bytes:
        - » Segmento 1: Número sequência: 10.001
        - » Segmento 2: Número sequência: 11.001
        - » Segmento 3: Número sequência: 12.001
        - » Segmento 4: Número sequência: 13.001
        - » Segmento 5: Número sequência: 14.001

# Camada de Transporte

- **TCP (Transmission Control Protocol)**
  - Sistema de Numeração – Número de bytes
    - Número de confirmação
      - Usado para confirmar os bytes que recebeu
      - Identifica o número do próximo byte que a parte espera receber – pega o número do último byte, incrementa 1 e anuncia a soma

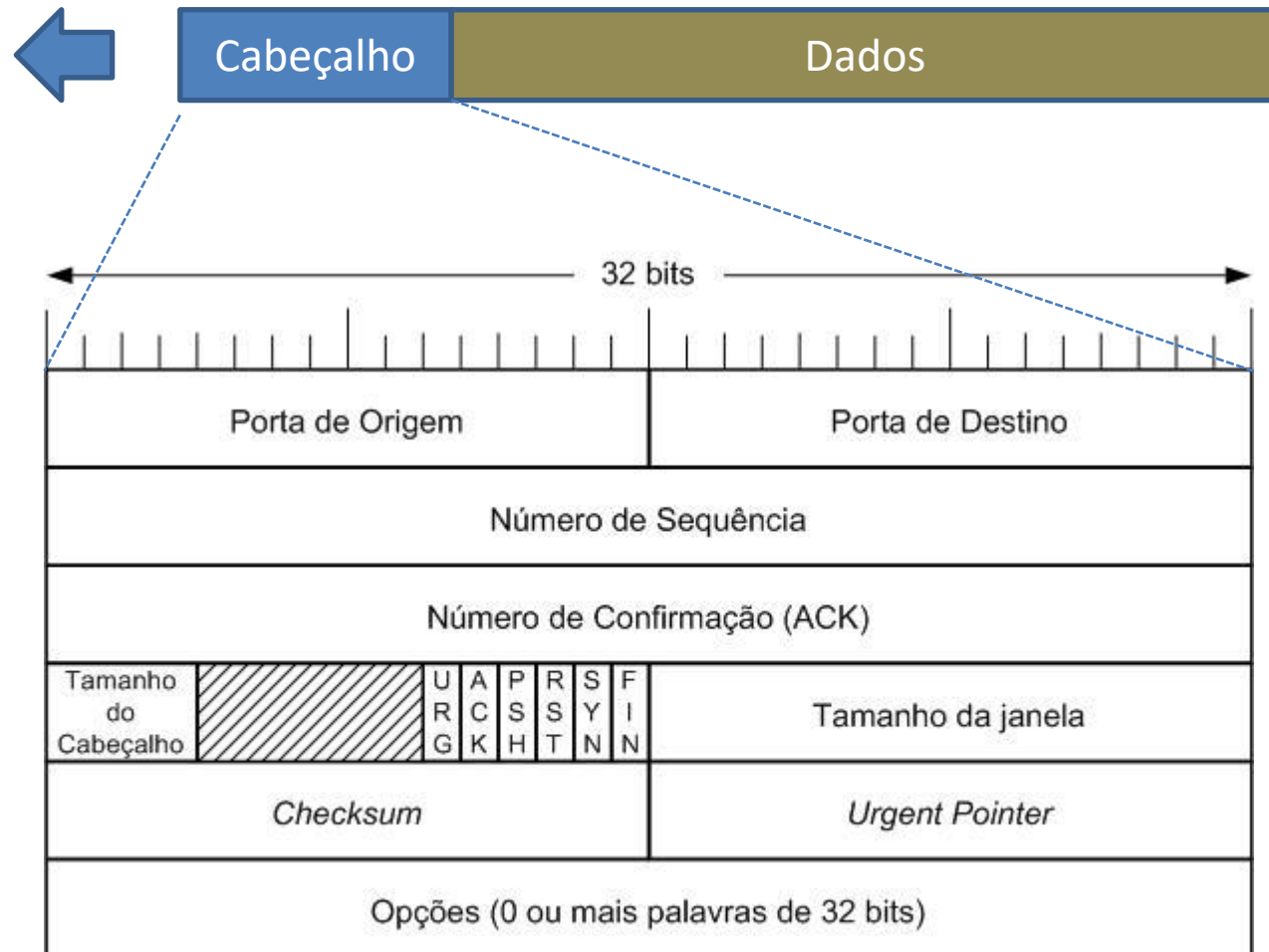
# Camada de Transporte

- **TCP (Transmission Control Protocol)**
  - Controle de fluxo
    - Janela deslizante
  - Controle de erros
    - Mecanismo de detecção de segmentos corrompidos, perdidos ou fora de ordem e segmentos duplicados
  - Controle de congestionamento
    - Alteração da quantidade de bytes transmitidos depende do congestionamento da rede



# Camada de Transporte

- TCP (Transmission Control Protocol)



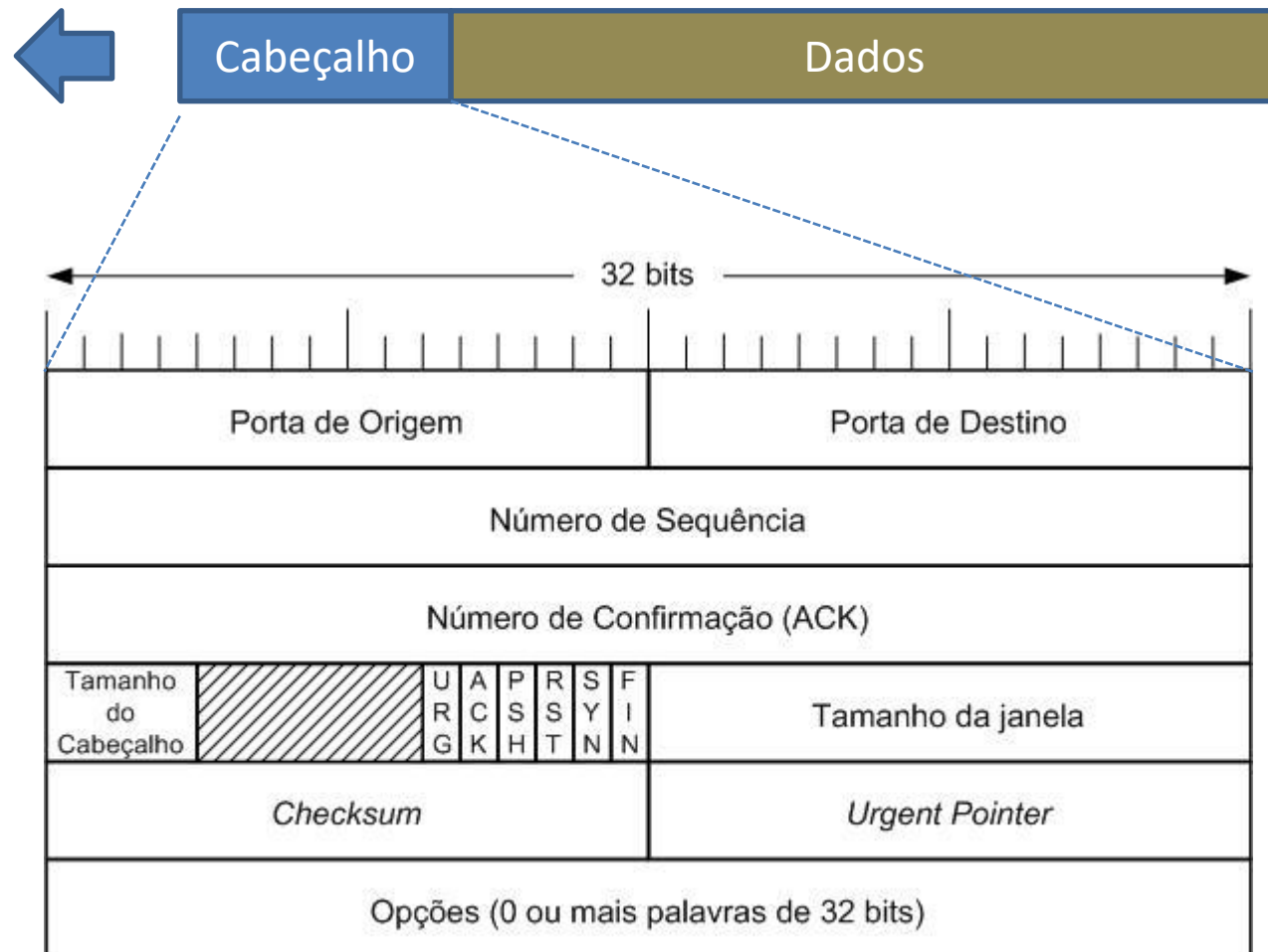


# Camada de Transporte

- Cabeçalho TCP
  - Endereço porta origem
  - Endereço porta destino
  - Número de sequência
  - Número de confirmação
  - Tamanho Cabeçalho (palavras de 4 bytes)
    - 20 a 60 bytes (entre 5 e 15)
  - Reservado

# Camada de Transporte

- TCP (Transmission Control Protocol)



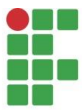
# Camada de Transporte

- Cabeçalho TCP
  - Controle

FLAG	DESCRIÇÃO
URG	Valor campo Urgent Point é válido
ACK	Valor campo de confirmação é válido
PSH	Empurra os dados
RST	Reinicia conexão
SYN	Sincroniza número de sequência durante conexão
FIN	Encerra conexão

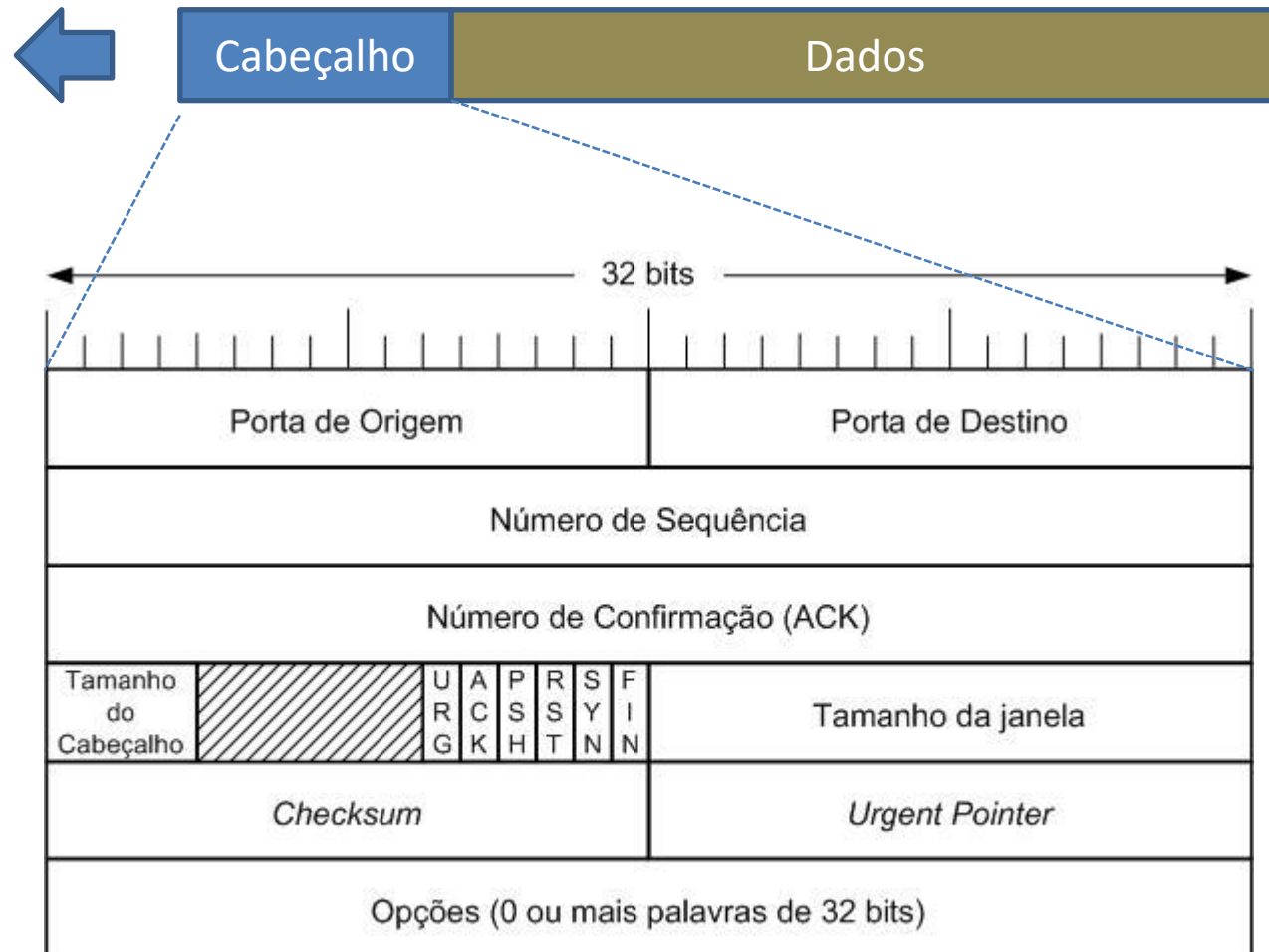
# Camada de Transporte

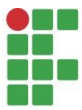
- Cabeçalho TCP
  - Tamanho Janela
    - Máximo 65.535 bytes
  - Cheksum
  - Urgent Point
    - Flag URG está ativo
    - Segmento contém dados urgentes
  - Opções
    - 40 bytes de informações adicionais



# Camada de Transporte

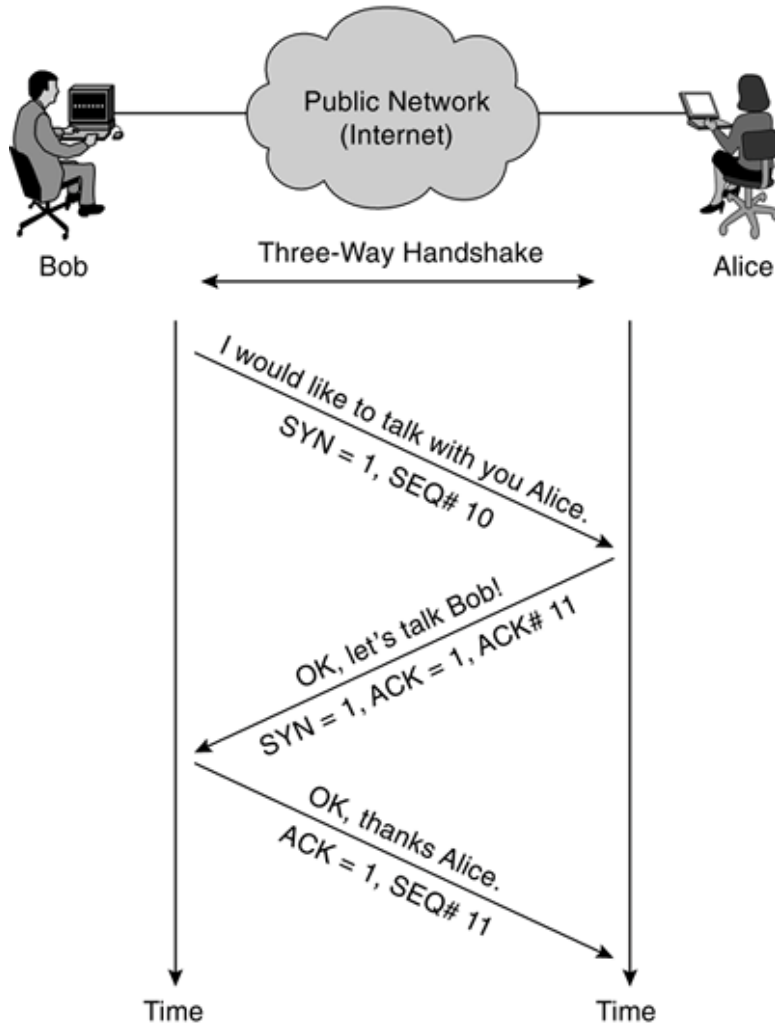
- TCP (Transmission Control Protocol)





# Camada de Transporte

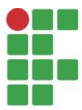
- Three-way handshake



1. Cliente transmite **SYN**, destinado a sincronizar o número de sequência
2. Servidor transmite o segundo segmento, **SYN + ACK**, **SYN** para iniciar a comunicação na outra direção e o **ACK** com a confirmação do **SYN** do cliente
3. Cliente transmite terceiro segmento, **ACK**, confirmando o **SYN** enviado pelo servidor

# Camada de Transporte

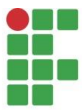
- **ATAQUE DE INUNDAÇÃO DE SYN**
  - Three-way handshaking é suscetível a ataques
  - Chamado de ataque de inundação de SYN (SYN Flooding Attack)
  - Invasor transmite um grande número de segmentos SYN a um servidor, simulando que cada um deles provém de um cliente diferente, forjando endereços IP de origem dos datagramas
  - Servidor aloca recursos necessários para os falsos clientes, que são ignorados ou perdidos
  - Ocupa uma grande quantidade de recursos do servidor
  - Configura ataque de negação de serviço (Denial of Service)



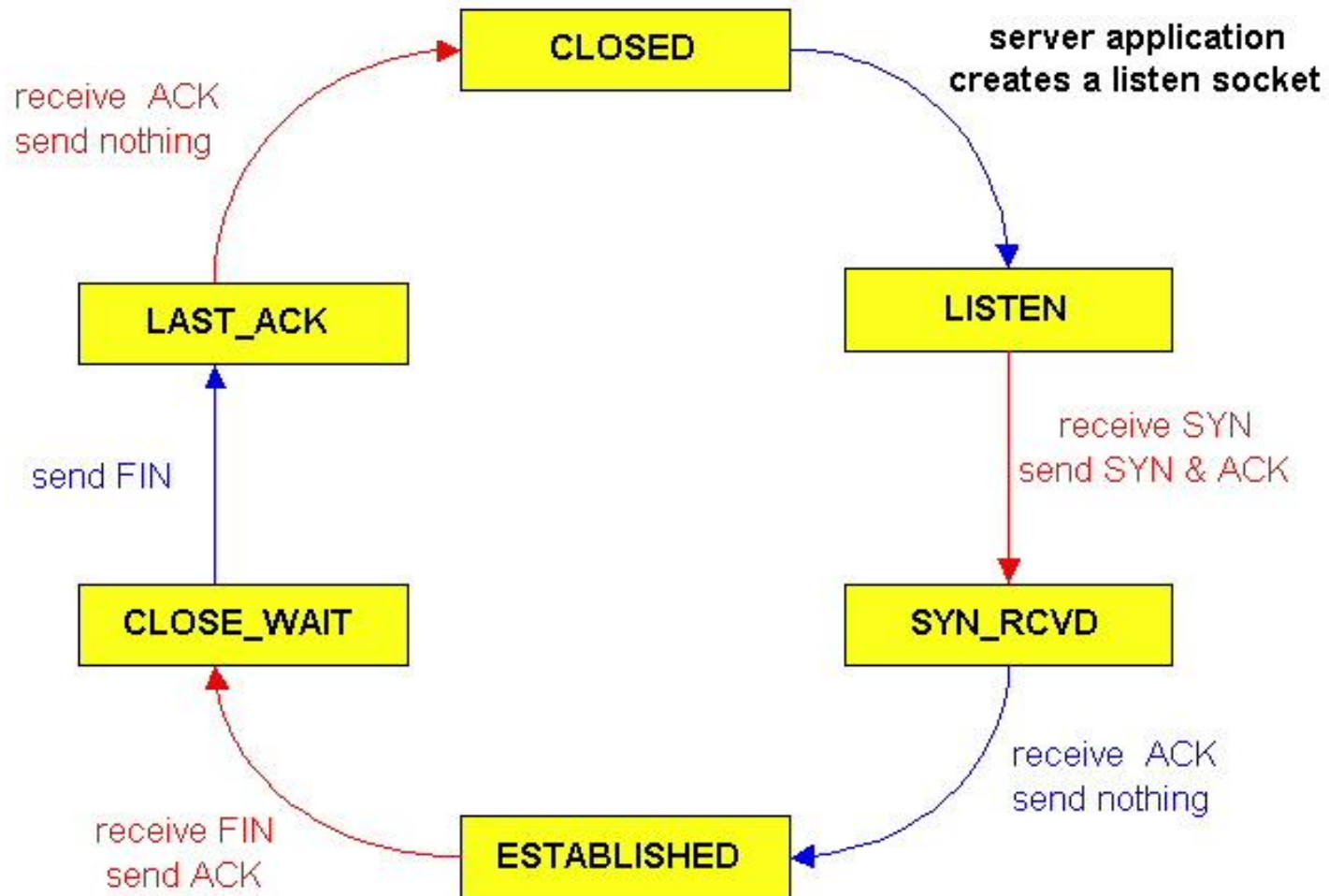
# Camada de Transporte

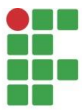
Tempo	Evento	Diagrama
	O computador A envia um pacote de sincronismo (SYN) para o computador B	<pre>sequenceDiagram     participant A as HOST A     participant B as HOST B     Note over A: t     A-&gt;&gt;B: syn     Note over B: t+1     B-&gt;&gt;A: ack syn     Note over A: t+3     A-&gt;&gt;B: ack     Note over B: t+5</pre>
+1	O computador B recebe o pacote (SYN) do computador A	
+2	O computador B envia seu próprio pacote de sincronismo (SYN) e o reconhecimento (ACK)	
+3	O computador A recebe o pacote SYN de B	
+4	O computador A envia o seu pacote de reconhecimento positivo (ACK)	
+5	O computador B recebe o ACK, e finalmente a conexão TCP é estabelecida e a transmissão dos pacotes de dados é iniciada até finalizar a sessão TCP.	



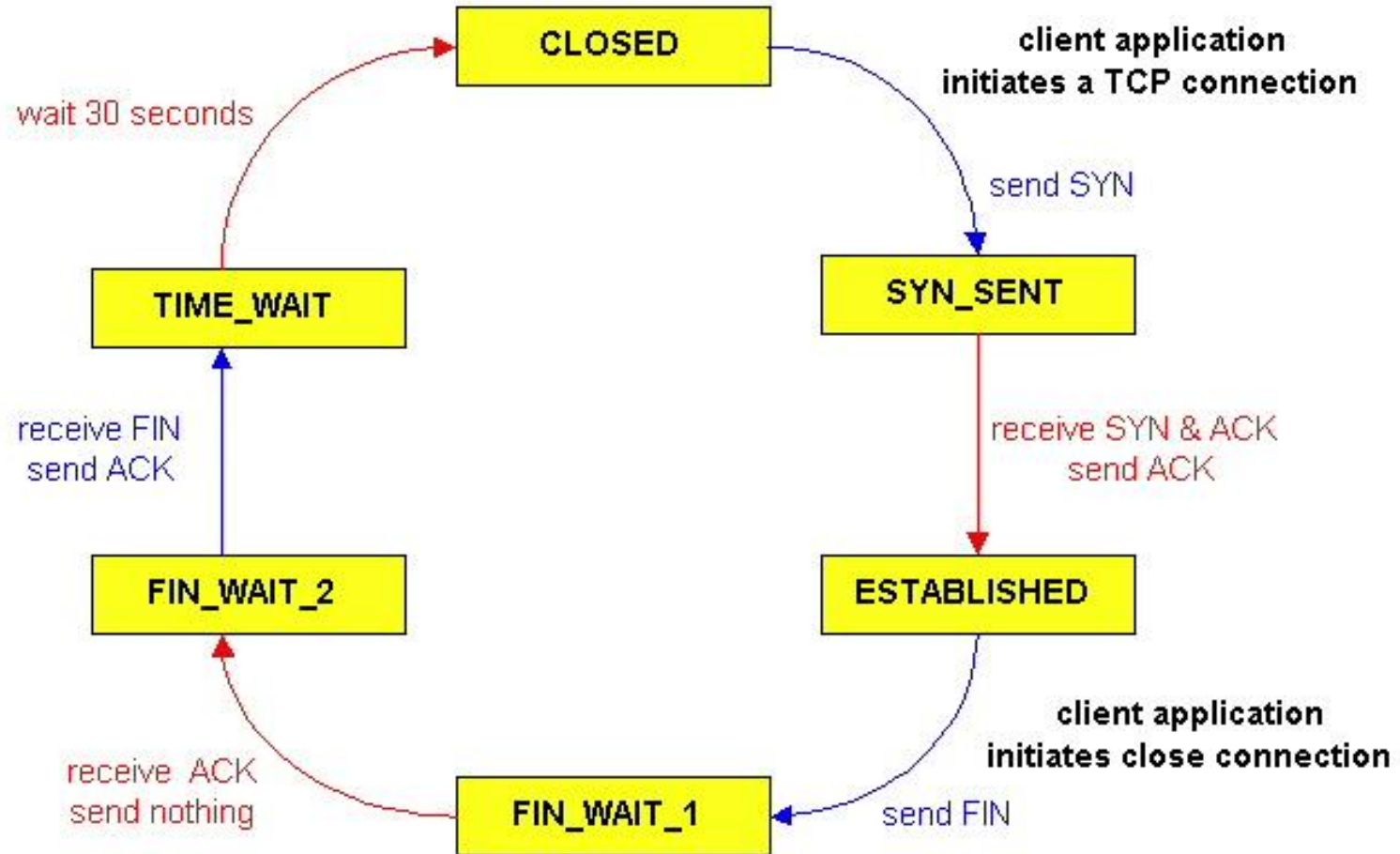


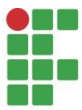
# Camada de Transporte





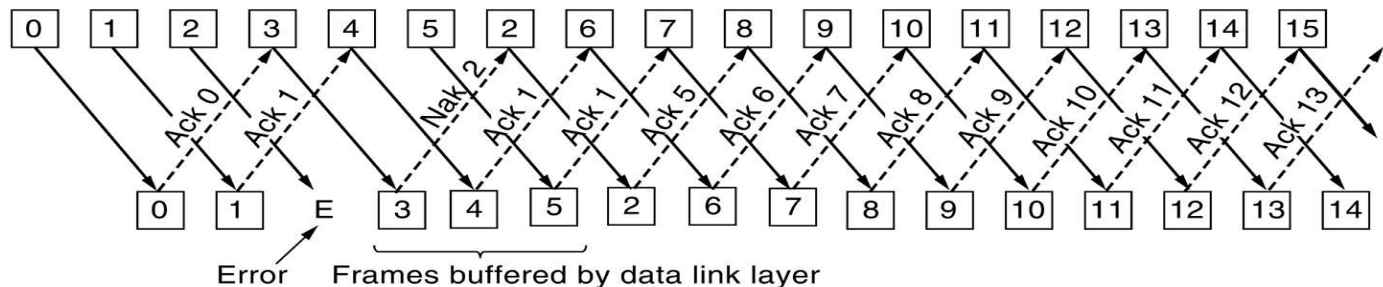
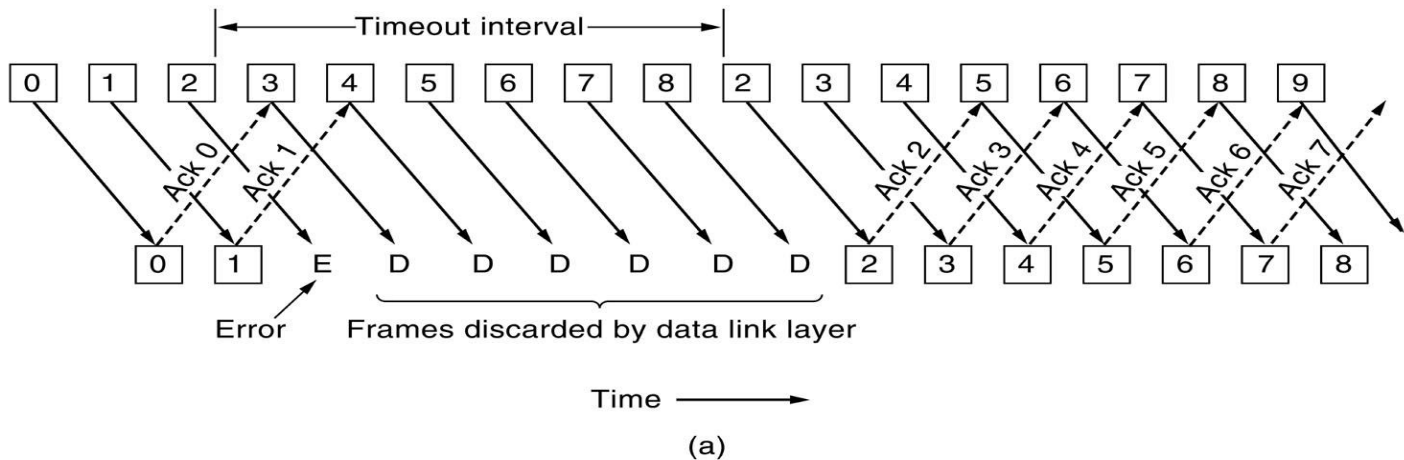
# Camada de Transporte





# Camada de Transporte

- CONTROLE DE FLUXO
  - Utiliza técnica de janela deslizante (sliding window)



(b)

# Camada de Transporte

- **CONTROLE DE ERROS**

- Checksum

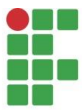
- Validar existência de segmentos corrompidos
    - Se estiver corrompido, é descartado

- Confirmação (ACK)

- Validar o recebimento do segmento de dados

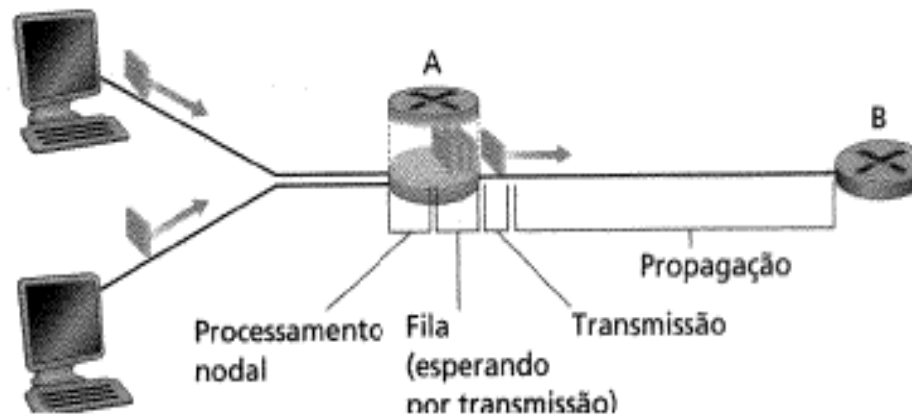
- Retransmissão

- Se o segmento estiver corrompido, perdido ou com atraso, ele é retransmitido
    - Timer de retransmissão se esgota



# Camada de Transporte

- **CONTROLE DE CONGESTIONAMENTO**
  - Ocorre congestionamento quando a carga na rede
    - número de pacotes enviados para a rede – é maior que a capacidade da rede
  - Ocorre porque os roteadores e comutadores possuem filas – buffers que retêm os pacotes



# Camada de Transporte

- **CONTROLE DE CONGESTIONAMENTO**
  - Política de retransmissão
    - Temporizadores projetados para otimizar a eficiência e evitar congestionamento
  - Política de janelas
    - Uso de janela de repetição seletiva
  - Política de confirmação
    - Receptor não confirmar todos os pacotes recebidos, desacelerando o transmissor
  - Política de descarte
    - Descartar pacotes menos sensíveis quando é provável a ocorrência de congestionamento (som – por exemplo)

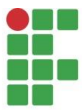
# Camada de Transporte

- **Protocolos Camada de Transporte**
- **SCTP (Stream Control Transmission Protocol)**
  - Novo protocolo da camada de transporte
  - Confiável e orientado a mensagens
  - Atender requisitos das novas aplicações
    - ISDN sobre IP
    - Telefonia IP
    - Outras
  - Oferece maior desempenho e confiabilidade
    - Combina as melhores características do UDP e TCP
      - Orientado a mensagem e confiável

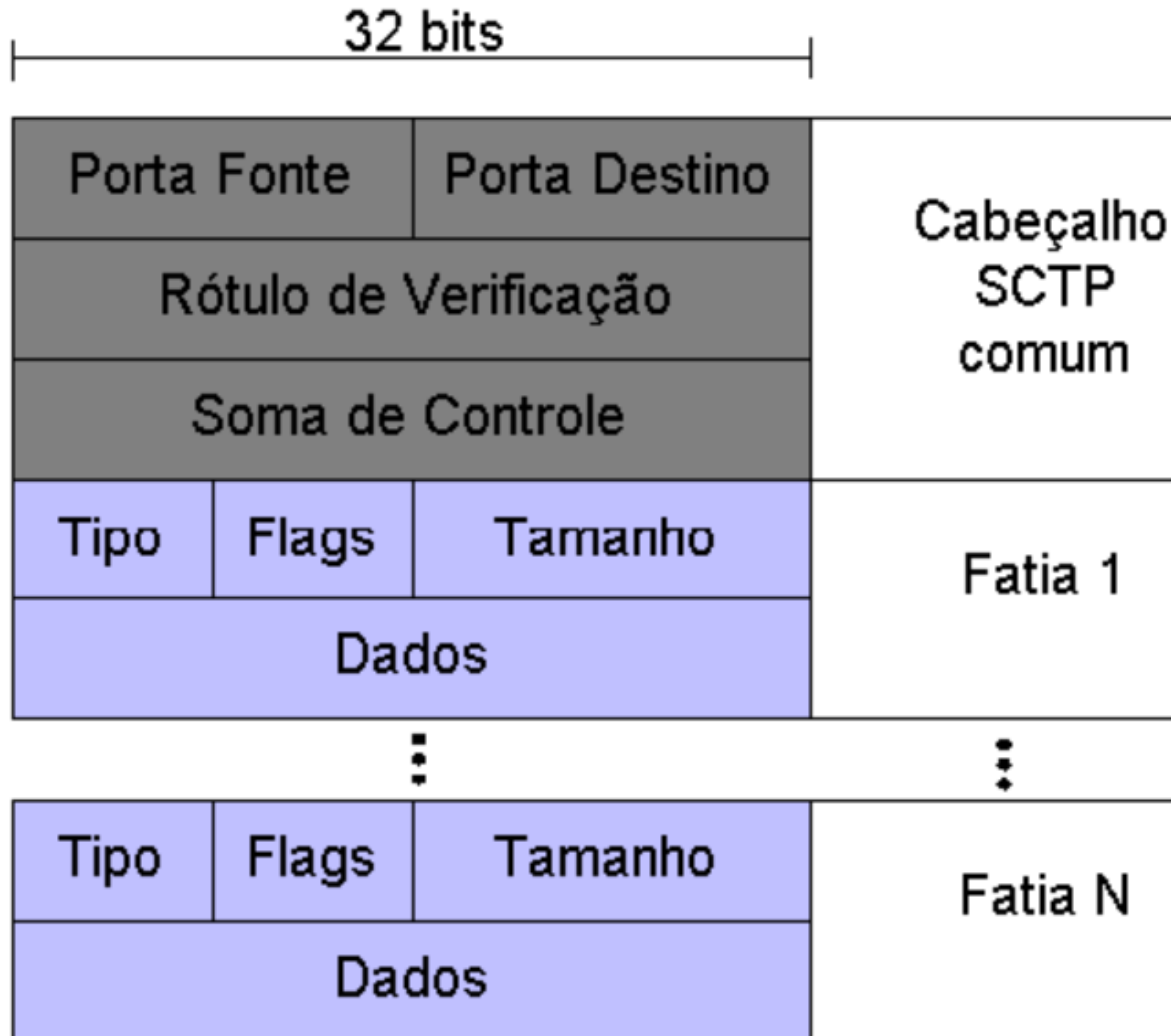
# Camada de Transporte

- **SCTP (Stream Control Transmission Protocol)**
  - Preserva delimitadores de mensagens
  - Detecta mensagens perdidas, duplicadas ou fora de ordem
  - Implementa controle de congestionamento e de fluxo
  - Orientado a conexão
    - Conexão é chamada **associação**
    - Unidade de dados: **conjunto de blocos**





# Camada de Transporte





# Camada de Transporte

## APLICAÇÕES POPULARES DA INTERNET, SEUS PROTOCOLOS DE CAMADA DE APLICAÇÃO E SEUS PROTOCOLOS DE TRANSPORTE SUBJACENTES

Aplicação	Protocolo de camada de aplicação	Protocolo de transporte subjacente
Correio eletrônico	SMTP [RFC 5321]	TCP
Acesso a terminal remoto	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Transferência de arquivos	FTP [RFC 959]	TCP
Multimídia em fluxo contínuo	HTTP (por exemplo, YouTube)	TCP
Telefonia por Internet	SIP [RFC 3261], RTP [RFC 3550] ou proprietária (por exemplo, Skype)	UDP ou TCP

# Referências



- FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores**. 4. ed. São Paulo: McGraw-Hill, 2008.
- KUROSE, Jim F. ROSS, Keith W. **Redes de Computadores e a Internet**. Uma nova abordagem. 3. ed. São Paulo: Addison Wesley, 2006.
- TANENBAUM, Andrew S. **Redes de computadores**. 3. Ed. Rio de Janeiro: Campus, 1997.
- COMER, Douglas E. **Internetworking with TCP/IP. Principal, Protocolos, and Architecture**. 2.ed. New Jersey: Prantice Hall, 1991. v.1.
- OPPENHEIMER, Priscilla. **Projeto de Redes Top-down**. Rio de Janeiro: Campus, 1999.
- GASPARINNI, Anteu Fabiano L., BARELLA, Francisco Rogério. **TCP/IP Solução para conectividade**. São Paulo: Editora Érica Ltda., 1993.

# Referências



- SPURGEON, Charles E. **Ethernet: o guia definitivo**. Rio de Janeiro: Campus, 2000.
- SOARES, Luiz Fernando G. **Redes de Computadores: das LANs, MANs e WANs às redes ATM**. Rio de Janeiro: Campus, 1995.
- CARVALHO, Tereza Cristina Melo de Brito (Org.). **Arquitetura de Redes de Computadores OSI e TCP/IP**. 2. Ed. rev. ampl. São Paulo: Makron Books do Brasil, Brisa; Rio de Janeiro: Embratel; Brasília, DF: SGA, 1997.
- COMER, Douglas E. **Interligação em rede com TCP/IP**. 2. Ed. Rio de Janeiro: Campus, 1998. v.1.
- ARNETT, Matthen Flint. **Desvendando o TCP/IP**. Rio de Janeiro: Campus, 1997. 543 p.
- ALVES, Luiz. **Comunicação de dados**. 2. Ed. rev. ampl. São paulo: Makron Books do Brasil, 1994.