

*CCT0300 – PROTOCOLOS DE REDES E DE COMPUTADORES*

## **Aula 03**

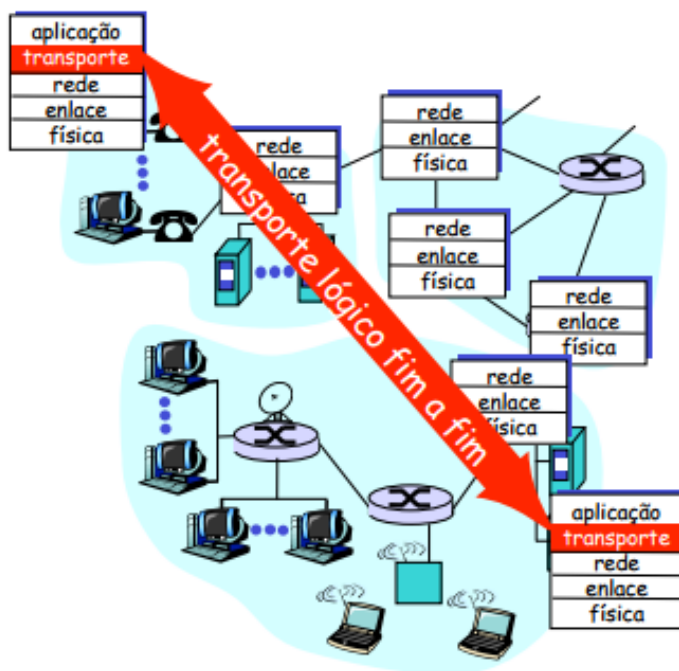


# Camada de Transporte



## Serviços e protocolos de transporte

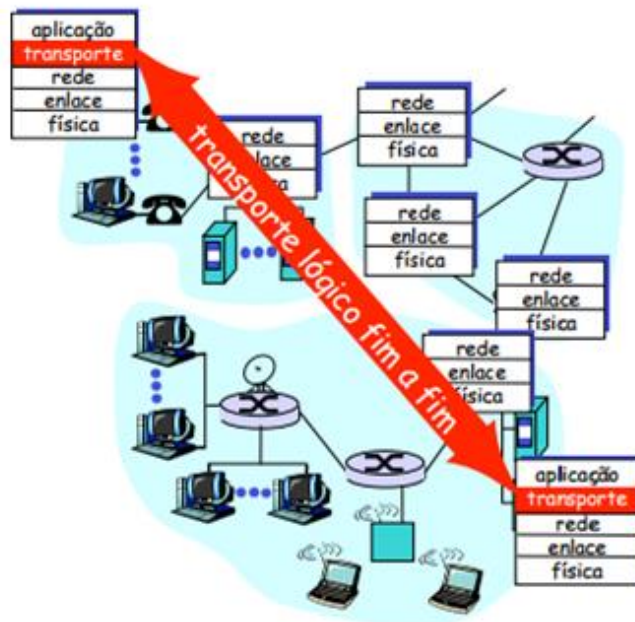
- ❑ provê *comunicação lógica* entre processos de aplicação executando em hospedeiros diferentes
- ❑ protocolos de transporte executam em sistemas terminais
- ❑ *serviços das camadas de transporte X rede:*
- ❑ *camada de rede* : dados transferidos entre sistemas
- ❑ *camada de transporte*: dados transferidos entre processos
  - depende de, estende serviços da camada de rede



## Protocolos da camada de transporte

### Serviços de transporte na Internet:

- ❑ entrega confiável, ordenada, ponto a ponto (TCP)
  - congestionamento
  - controle de fluxo
  - estabelecimento de conexão (setup)
- ❑ entrega não confiável, (“melhor esforço”), não ordenada, ponto a ponto ou multiponto: UDP
- ❑ serviços não disponíveis:
  - tempo-real
  - garantias de banda
  - multiponto confiável



- **Camada de rede:**
  - Comunicação lógica entre hospedeiros
  
- **Camada de transporte:**
  - comunicação lógica entre processos
  - utiliza e aprimora os serviços oferecidos pela camada de rede

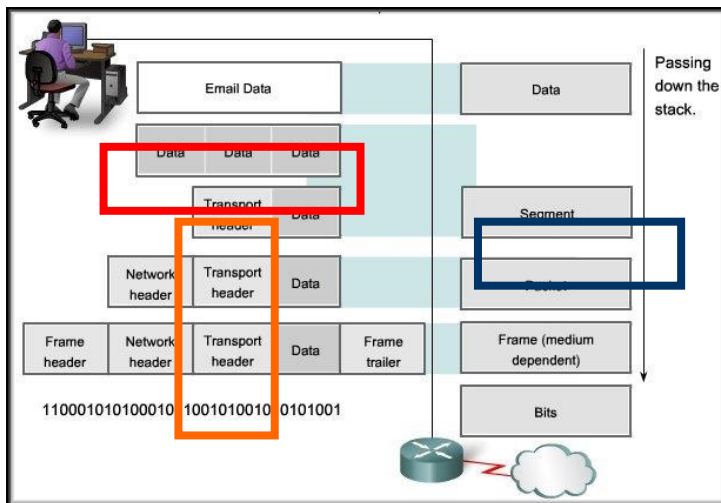
## Protocolos de Redes e de Computadores

### Serviços de Transporte da Internet



- TCP: Entrega unicast, sequencial e confiável
  - Estabelecimento de conexão
  - Controle de fluxo
  - Controle de congestionamento
- UDP: Entrega unicast ou multicast, não confiável (best-effort) e não sequencial
- Não disponíveis:
  - Garantia de retardo
  - Garantia de banda
  - Multicast confiável



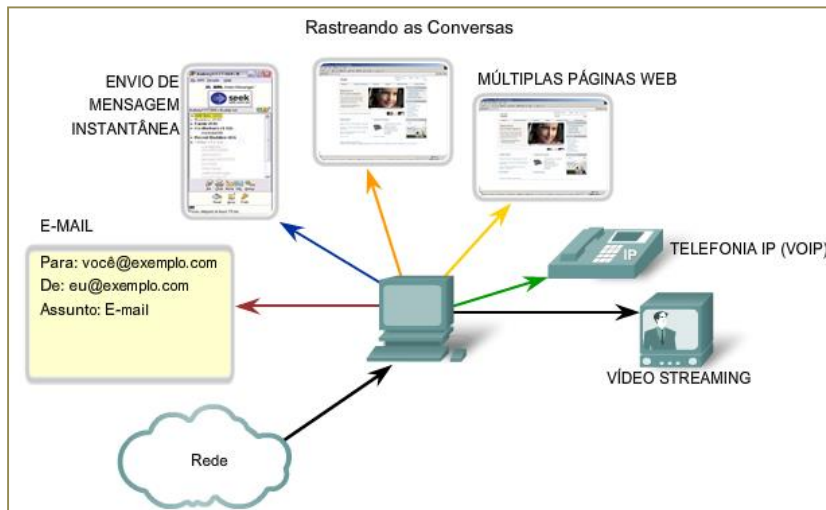


- Responsável pela transferência fim-a-fim geral de dados de aplicação
- Encapsulamento de dados de aplicação para uso pela camada de Rede
- Habilita a comunicação de múltiplas aplicações na rede ao mesmo tempo em um único dispositivo
- Assegura que, se necessário, todos os dados sejam recebidos confiavelmente e em ordem pela aplicação correta.
- Emprega mecanismos de tratamento de erros



## Protocolos de Redes e de Computadores

### Separação de Múltiplas Comunicações

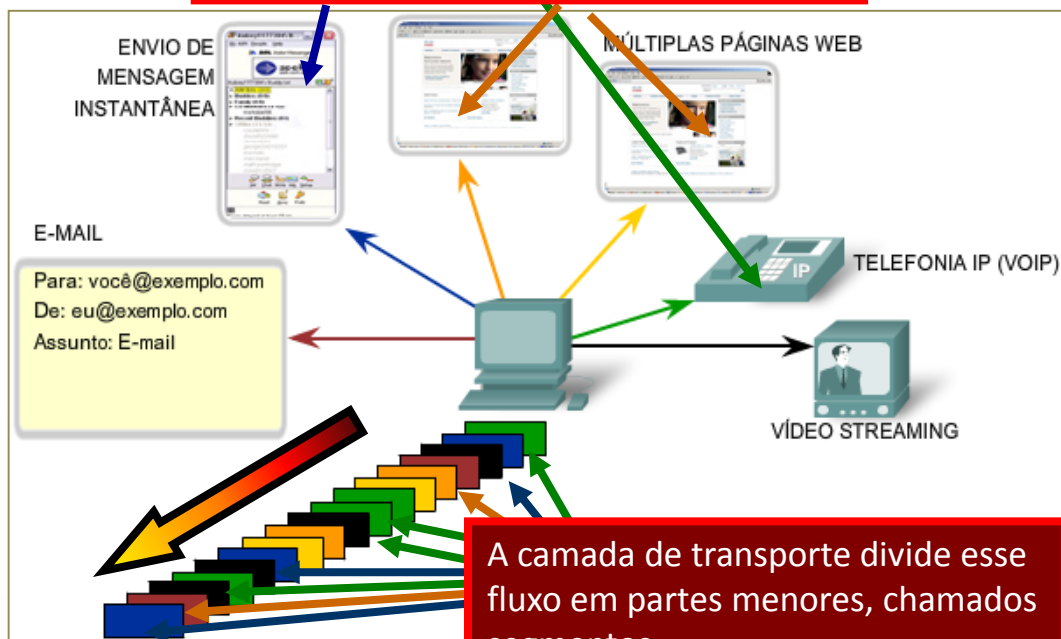


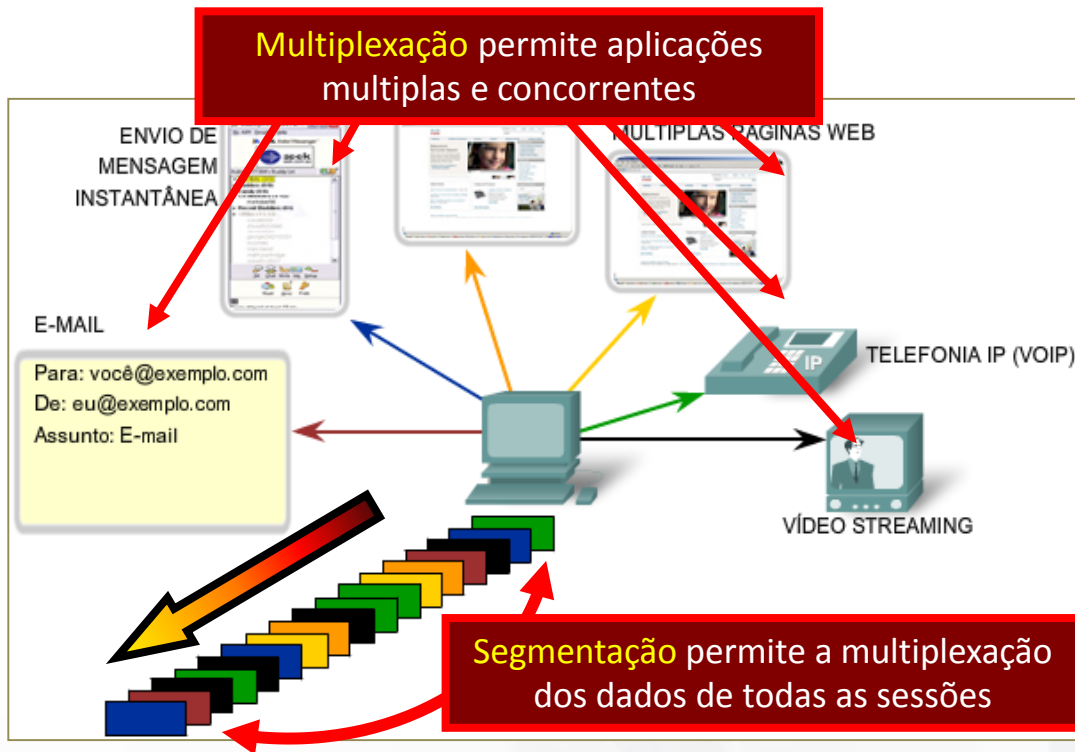
- Qualquer host pode ter múltiplas aplicações que se comunicam através da rede
- A camada de Transporte mantém esses fluxos separados
  - Por exemplo, o texto de uma mensagem instantânea não aparece em um e-mail

## Protocolos de Redes e de Computadores

### Segmentação dos Dados

A camada de aplicação envia um fluxo completo (tamanho grande)





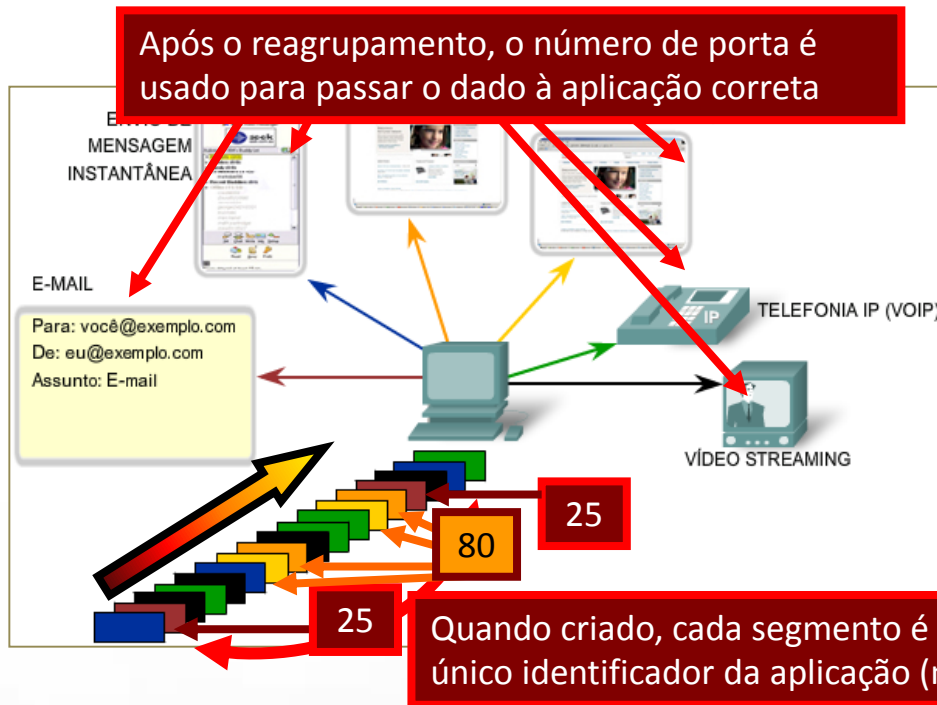
## Protocolos de Redes e de Computadores

### Reagrupamento dos Segmentos



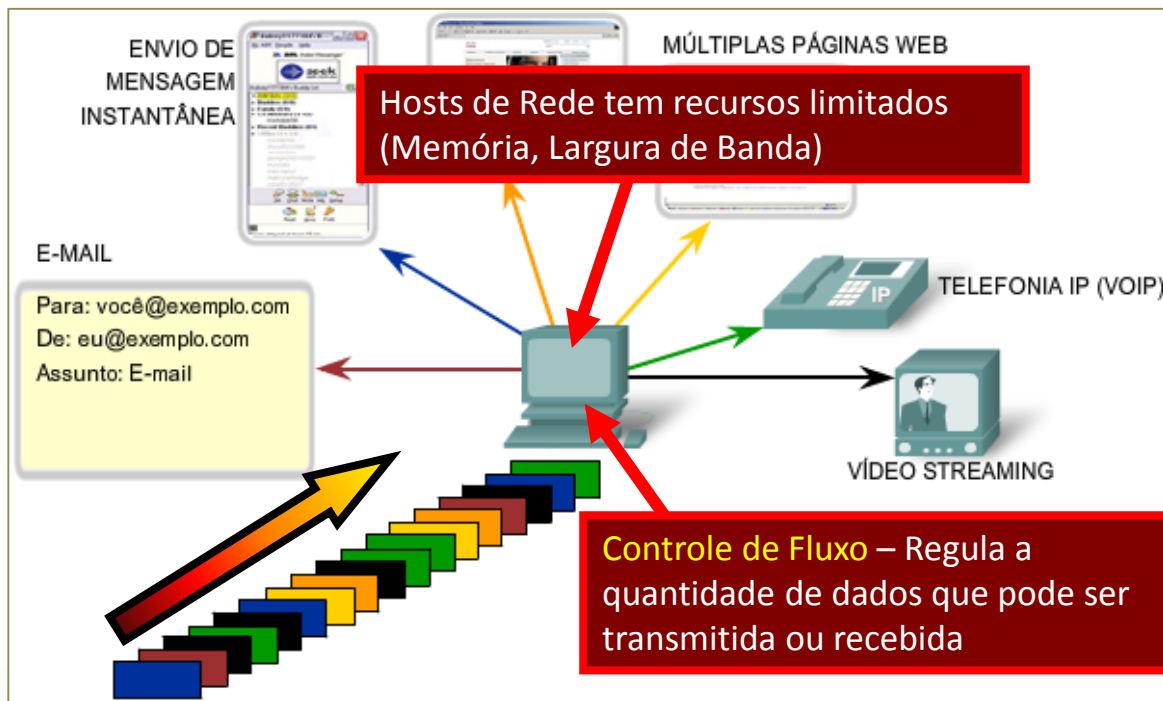
## Protocolos de Redes e de Computadores

### Reagrupamento dos Segmentos



## Protocolos de Redes e de Computadores

### Controle de Fluxo



## Protocolos de Redes e de Computadores

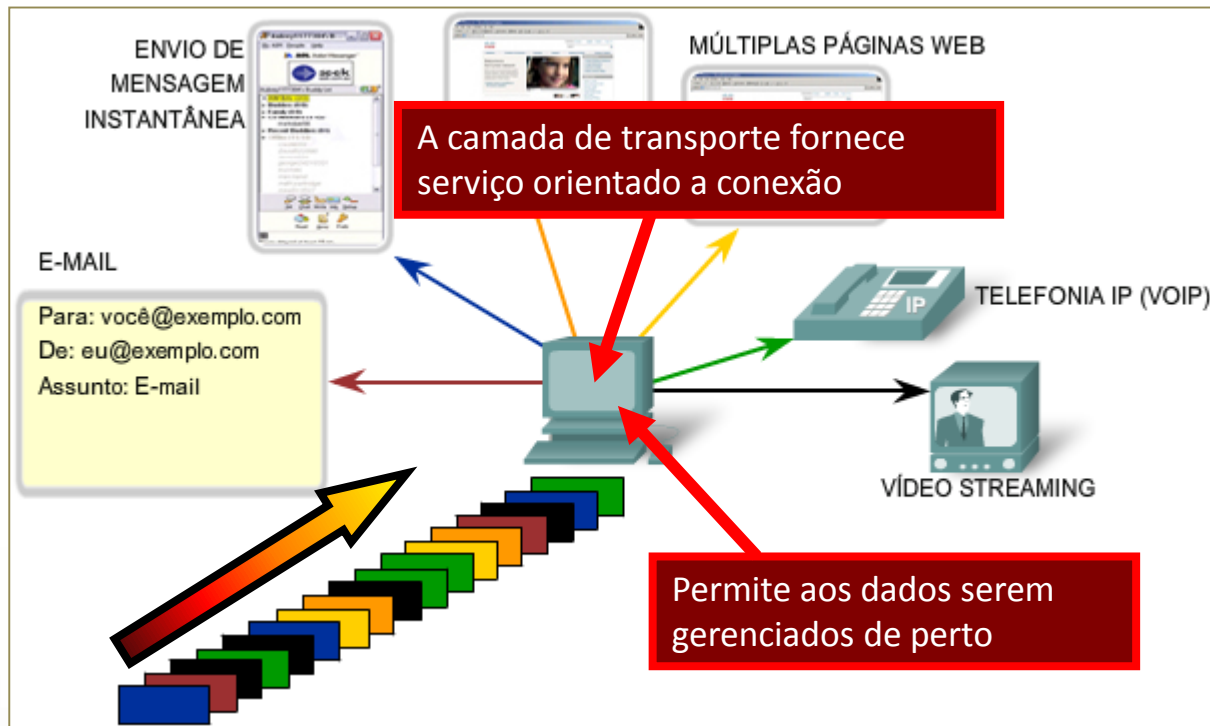
### Entrega Confiável





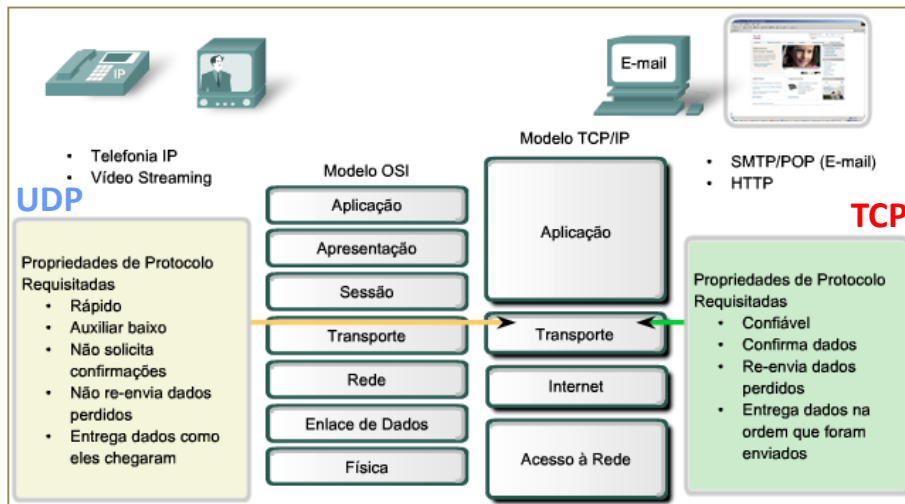
## Protocolos de Redes e de Computadores

### Orientado a conexão



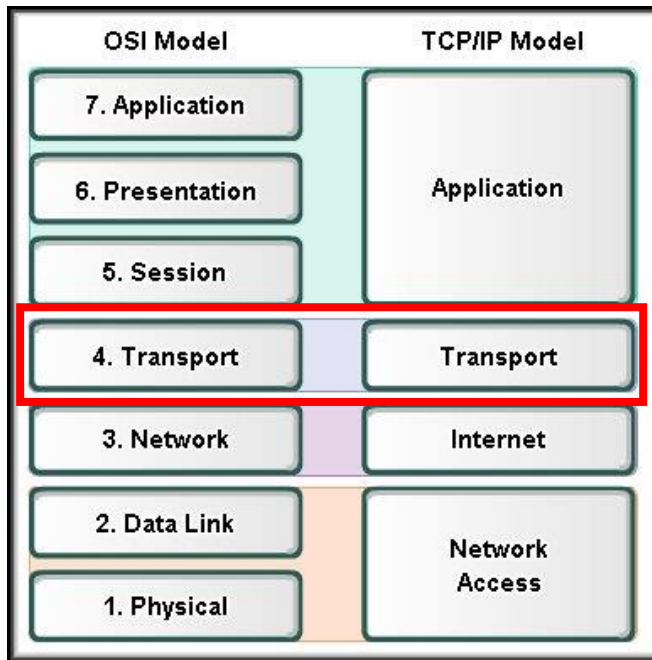
# Protocolos de Redes e de Computadores

## Suporte à Comunicação Confiável



- Relembre – Principal função da camada de Transporte é gerenciar os dados da aplicação para as conversações entre os hosts
- Diferentes aplicações têm diferentes necessidades para seus dados
- Diferentes protocolos de Transporte têm sido desenvolvidos para satisfazer estas necessidades

## TCP e UDP



# TCP e UDP

TCP



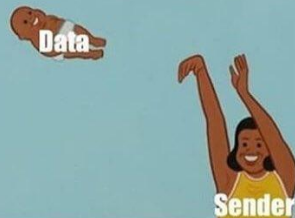
UDP



TCP



UDP



TCP + SSH

TCP

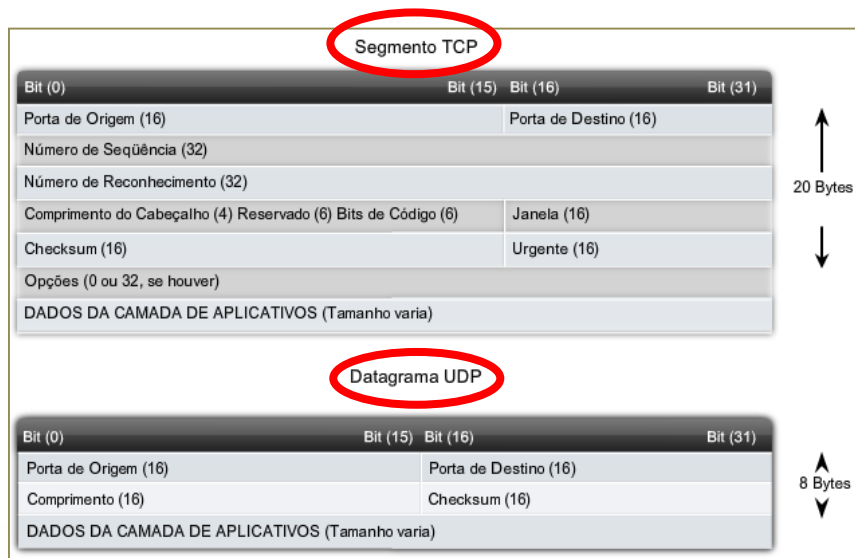


UDP

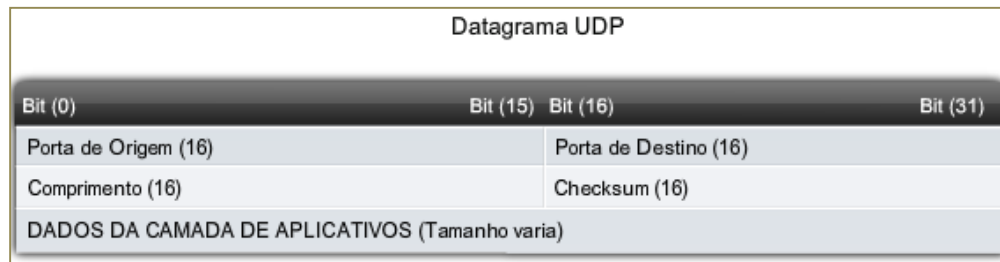


UDP and guess  
what lost packets  
are





- Protocolos da camada de transporte mais comuns



- Sem-conexão
- Entrega de melhor esforço (Best Effort)
- Baixo overhead

#### Exemplos de Aplicação

- Domain Name System (DNS)
- Jogos online
- Voice over IP (VoIP)
- Dynamic Host Configuration Protocol (DHCP)
- Trivial File Transfer Protocol (TFTP)

## Protocolos de Redes e de Computadores

### Transmission Control Protocol (TCP)



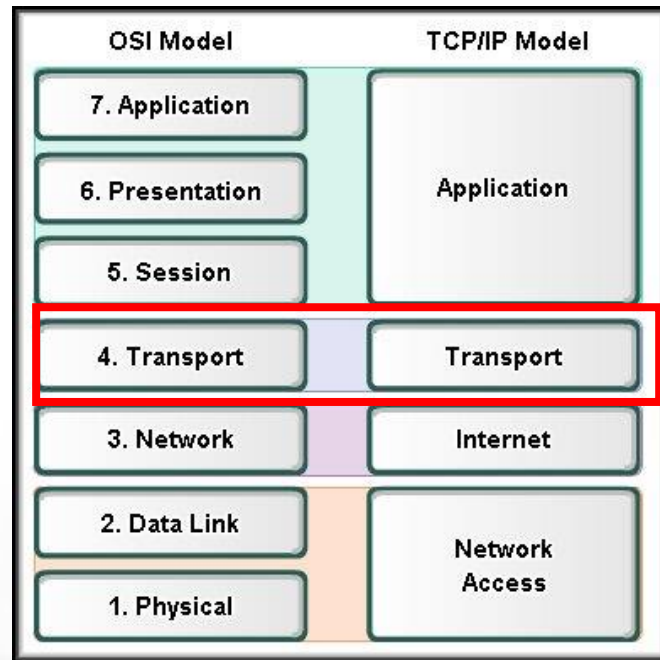
- Orientado a conexão
- Entrega confiável
- Checagem de erro
- Controle de Erro

#### Exemplos de Aplicações

- Hypertext Transfer Protocol (HTTP)
- File Transfer Protocol (FTP)
- Telnet
- Simple Message Transfer Protocol (SMTP)



## Endereçamento de Porta

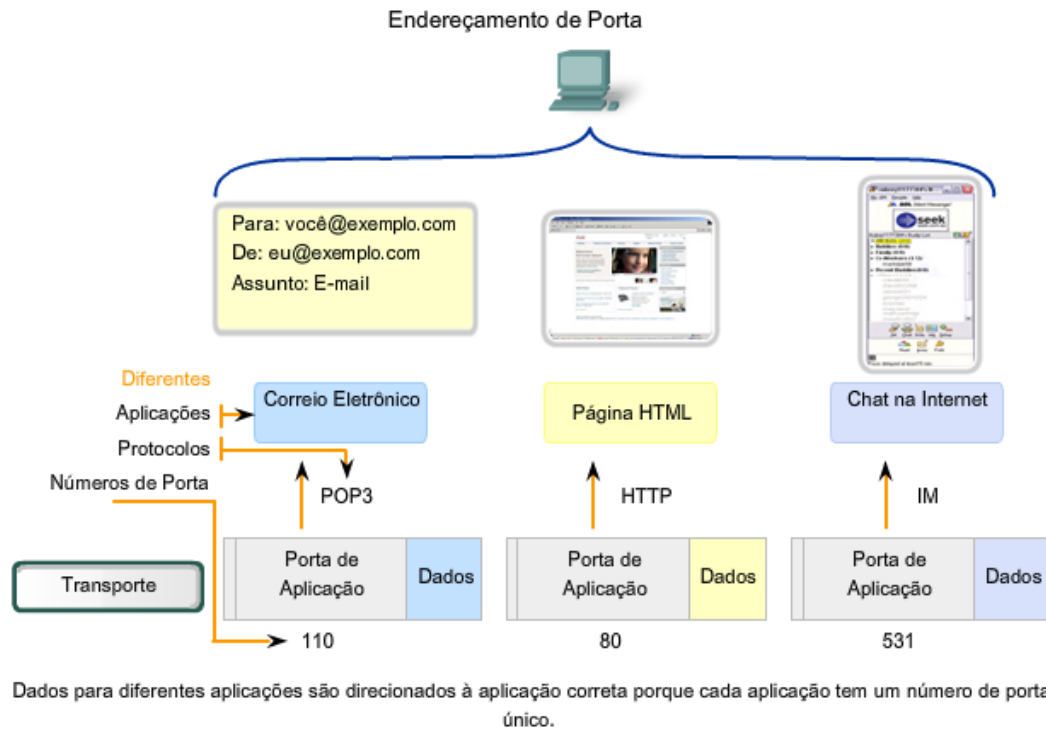


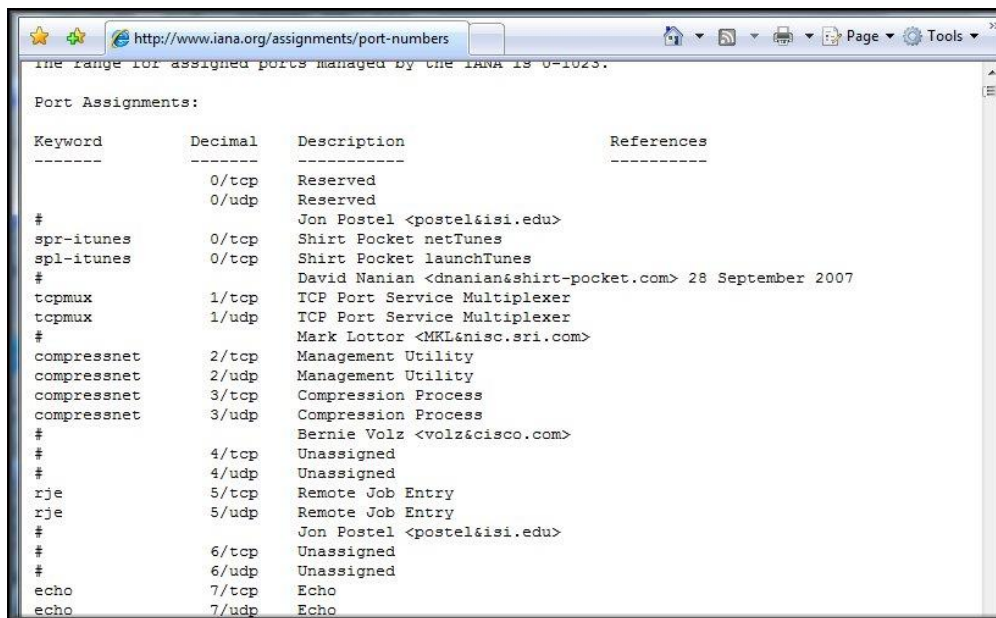


- TCP e UDP usam números de porta para passar a informação as camadas superiores
- Socket - combinação do número de porta e do endereço IP
- Identifica exclusivamente um processo particular sendo executado em um dispositivo de host específico
- e.g. 207.134.65.2:80 referencia um socket HTTP

# Protocolos de Redes e de Computadores

## Identificação de Conversações





The screenshot shows a web browser window with the address bar displaying <http://www.iana.org/assignments/port-numbers>. The page content includes a note: "The range for assigned ports managed by the IANA is 0-1023." Below this, under the heading "Port Assignments:", there is a table with four columns: Keyword, Decimal, Description, and References. The table lists various port assignments, including reserved ports, ports for specific services like iTunes, and ports for protocols like TCPMUX and COMPRESSNET.

Keyword	Decimal	Description	References
	0/tcp	Reserved	
	0/udp	Reserved	
#		Jon Postel <postel@isi.edu>	
spr-itunes	0/tcp	Shirt Pocket netTunes	
spl-itunes	0/tcp	Shirt Pocket launchTunes	
#		David Nanian <dnanian@shirt-pocket.com> 28 September 2007	
tcpmux	1/tcp	TCP Port Service Multiplexer	
tcpmux	1/udp	TCP Port Service Multiplexer	
#		Mark Lottor <MKL@nisc.sri.com>	
compressnet	2/tcp	Management Utility	
compressnet	2/udp	Management Utility	
compressnet	3/tcp	Compression Process	
compressnet	3/udp	Compression Process	
#		Bernie Volz <volz@cisco.com>	
#	4/tcp	Unassigned	
#	4/udp	Unassigned	
rje	5/tcp	Remote Job Entry	
rje	5/udp	Remote Job Entry	
#		Jon Postel <postel@isi.edu>	
#	6/tcp	Unassigned	
#	6/udp	Unassigned	
echo	7/tcp	Echo	
echo	7/udp	Echo	

- IANA - órgão de padrões responsável pela designação de vários padrões de endereçamento
- Designa números de porta

### Tipos de Endereço de Porta

- Portas Conhecidas:
  - Reservados para serviços e aplicações comuns

Faixa de Números de Portas	Grupo de Portas
0 a 1023	Portas conhecidas (Contato)
1024 a 49151	Portas Registradas
49152 a 65535	Portas Privadas e/ou Dinâmicas

20 – FTP Data

25 – SMTP

443 – HTTPS

21 – FTP Control

110 – POP3

69 – TFTP

23 – Telnet

194 – IRC

520 – RIP

### Tipos de Endereço de Porta

- Portas Registradas:
  - Aplicações e processos do usuário (opcional)

Faixa de Números de Portas	Grupo de Portas
0 a 1023	Portas conhecidas (Contato)
1024 a 49151	Portas Registradas
49152 a 65535	Portas Privadas e/ou Dinâmicas

1863 – MSN Messenger

1812 – RADIUS

8008 – Alternate HTTP

2000 – Cisco VoIP

8080 – Alternate HTTP

5004 – RTP

5060 – SIP (VoIP)

### Tipos de Endereço de Porta

- Portas Dinâmicas:
  - Atribuída dinamicamente a aplicações de cliente quando se inicia uma conexão

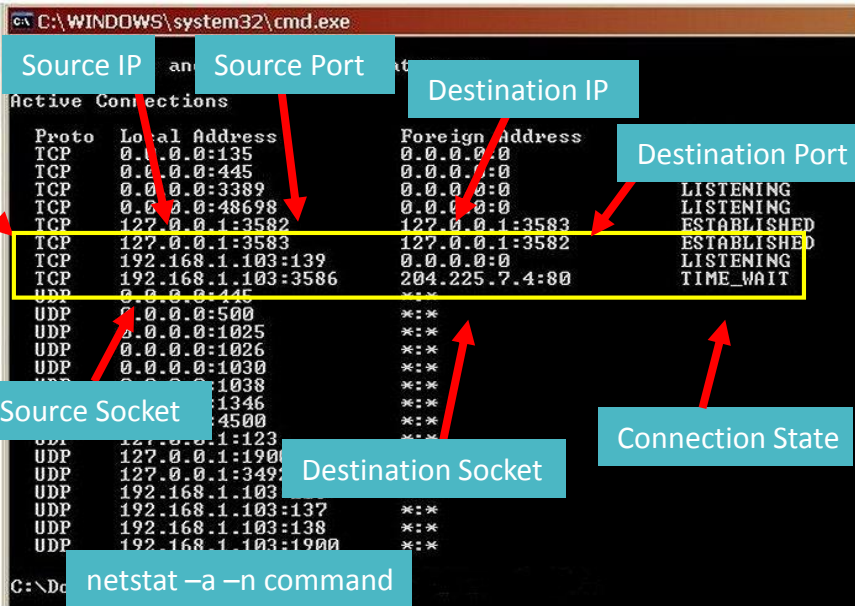
Faixa de Números de Portas	Grupo de Portas
0 a 1023	Portas conhecidas (Contato)
1024 a 49151	Portas Registradas
49152 a 65535	Portas Privadas e/ou Dinâmicas



### Tipos de Endereço de Porta

- Utilização do TCP e do UDP :
  - Algumas aplicações podem usar tanto TCP como UDP
  - Por exemplo, o baixo overhead (sobrecarga) do UDP habilita ao DNS servir a muitas solicitações de clientes muito rapidamente
    - As vezes, no entanto, o envio da informação solicitada pode exigir a confiabilidade do TCP. Neste caso, o número 53 de porta conhecida é usado por ambos os protocolos com este serviço

Faixa de Números de Portas	Grupo de Portas
0 a 1023	Portas conhecidas (Contato)
1024 a 49151	Portas Registradas
49152 a 65535	Portas Privadas e/ou Dinâmicas



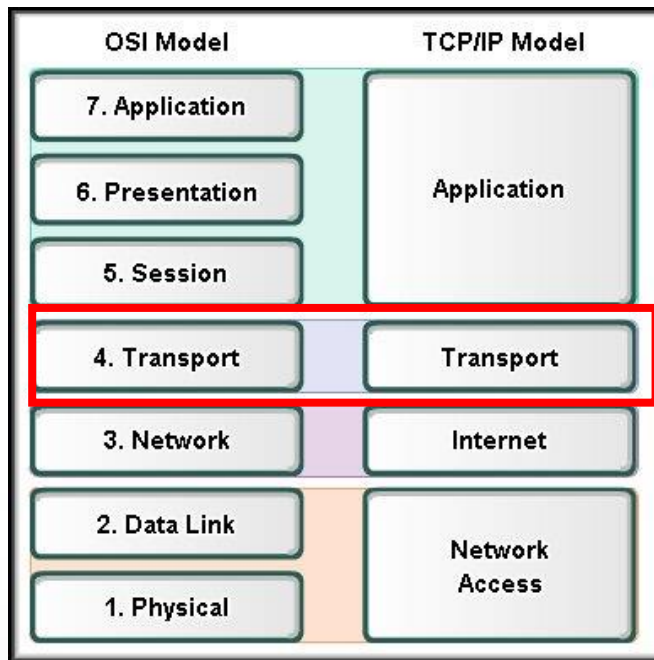
```
C:\WINDOWS\system32\cmd.exe

Active Connections

Proto Local Address Foreign Address Connection State
TCP 0.0.0.0:135 0.0.0.0:0 LISTENING
TCP 0.0.0.0:445 0.0.0.0:0 LISTENING
TCP 0.0.0.0:3389 0.0.0.0:0 LISTENING
TCP 0.0.0.0:48698 0.0.0.0:0 LISTENING
TCP 127.0.0.1:3582 127.0.0.1:3583 ESTABLISHED
TCP 127.0.0.1:3583 127.0.0.1:3582 ESTABLISHED
TCP 192.168.1.103:139 0.0.0.0:0 LISTENING
TCP 192.168.1.103:3586 204.225.7.4:80 TIME_WAIT
UDP 0.0.0.0:445 *:* *:*
UDP 0.0.0.0:500 *:* *:*
UDP 0.0.0.0:1025 *:* *:*
UDP 0.0.0.0:1026 *:* *:*
UDP 0.0.0.0:1030 *:* *:*
UDP 0.0.0.0:1038 *:* *:*
UDP 0.0.0.0:1346 *:* *:*
UDP 0.0.0.0:4500 *:* *:*
UDP 127.0.0.1:123 *:* *:*
UDP 127.0.0.1:1900 *:* *:*
UDP 127.0.0.1:349 *:* *:*
UDP 192.168.1.103 *:* *:*
UDP 192.168.1.103:137 *:* *:*
UDP 192.168.1.103:138 *:* *:*
UDP 192.168.1.103:1900 *:* *:*

C:\>netstat -a -n command
```

- As vezes é necessário conhecer quais conexões TCP ativas estão abertas e sendo executadas em um host de rede



## Protocolos de Redes e de Computadores

### Transmission Control Protocol (TCP)



- Orientado a conexão
- Entrega confiável
- Checagem de erro
- Controle de Erro

#### Exemplos de Aplicações

- Hypertext Transfer Protocol (HTTP)
- File Transfer Protocol (FTP)
- Telnet
- Simple Message Transfer Protocol (SMTP)

### Tipos de Endereço de Porta

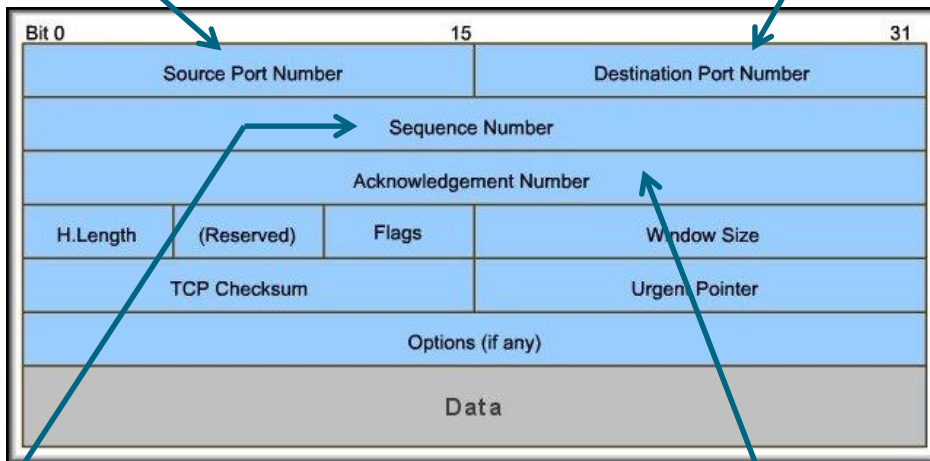
- A distinção principal entre o TCP e o UDP está na confiabilidade.
- TCP usa sessões orientadas à conexão
  - Antes de algum dado ser trocado, a camada de Transporte inicia um processo para criar uma conexão com o destino
  - Esta conexão habilita o rastreamento de uma sessão
    - Números de sequência
    - Reconhecimento
  - Overhead do TCP - Tráfego de rede gerado por confirmações e retransmissões
  - Confiabilidade é alcançada através de campos com funções específicas no segmento TCP

## Protocolos de Redes e Computadores

### Tornando as Conversações Confiáveis

Sessão TCP que abriu uma conexão.  
Normalmente um valor aleatório até 1023

Aplicação da camada superior no destino remoto



Número do último octeto  
(byte) no segmento

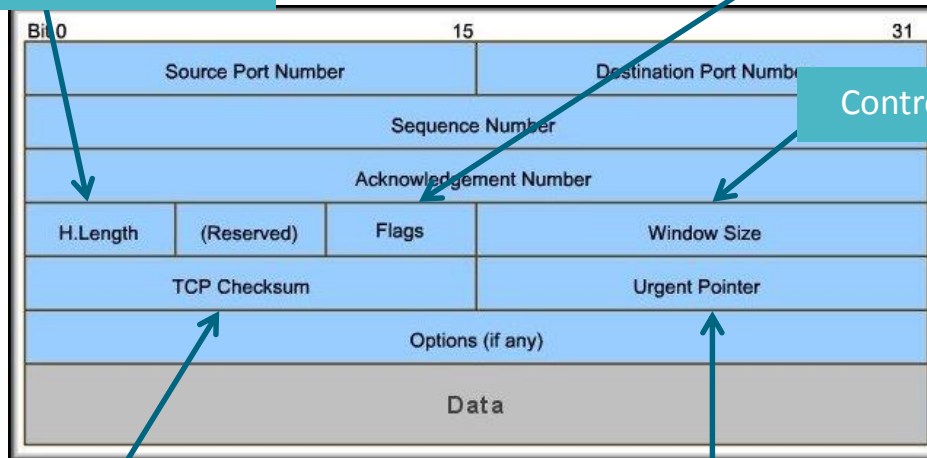
Número do próximo octeto  
(byte) esperado pelo receptor

## Protocolos de Redes e de Computadores

### Tornando as Conversações Confiáveis

Tamanho do cabeçalho do segmento em bytes

Campo de 1 bit, usado na sessão de gerenciamento



Controle de fluxo

Checagem de erro no cabeçalho

Se o Flag urgente é setado, o dado é alocado como urgente



## Processo TCP em Servidores

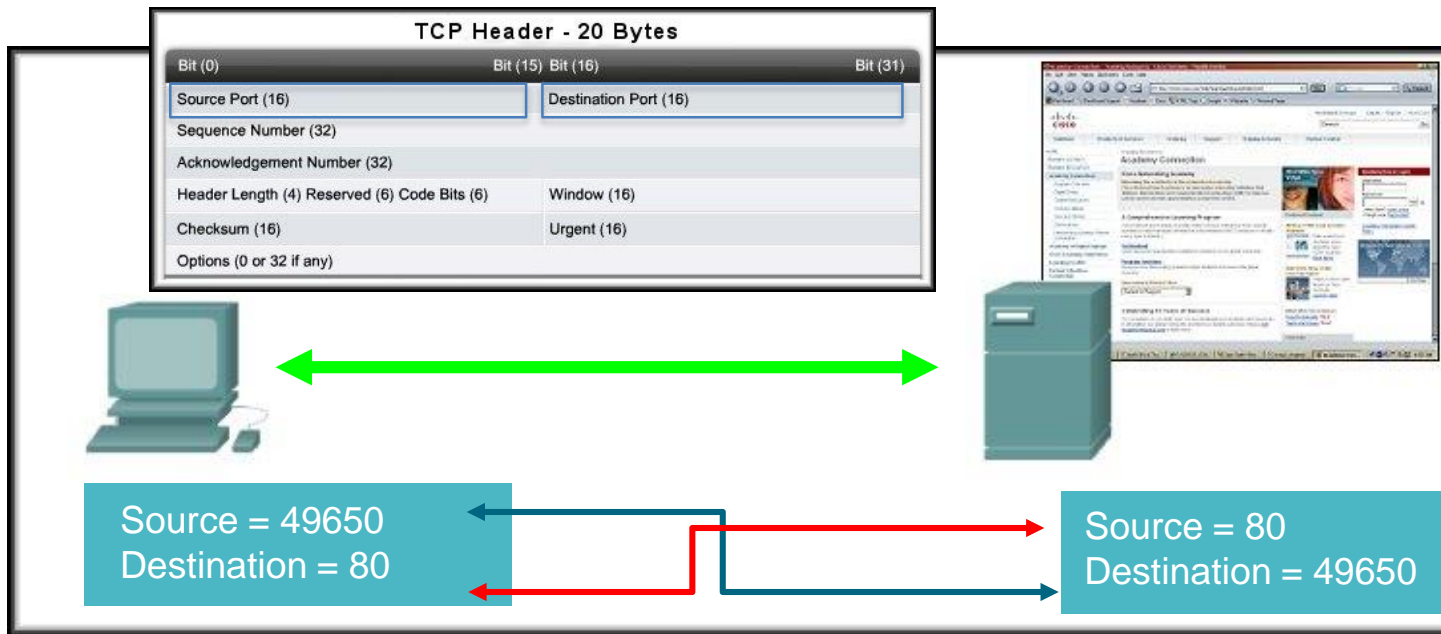


TCP Header - 20 Bytes				
Bit (0)		Bit (15)	Bit (16)	Bit (31)
Source Port (16)		Destination Port (16)		
Sequence Number (32)				
Acknowledgement Number (32)				
Header Length (4)		Reserved (6)		Code Bits (6)
Checksum (16)		Window (16)		
Urgent (16)		Options (0 or 32 if any)		

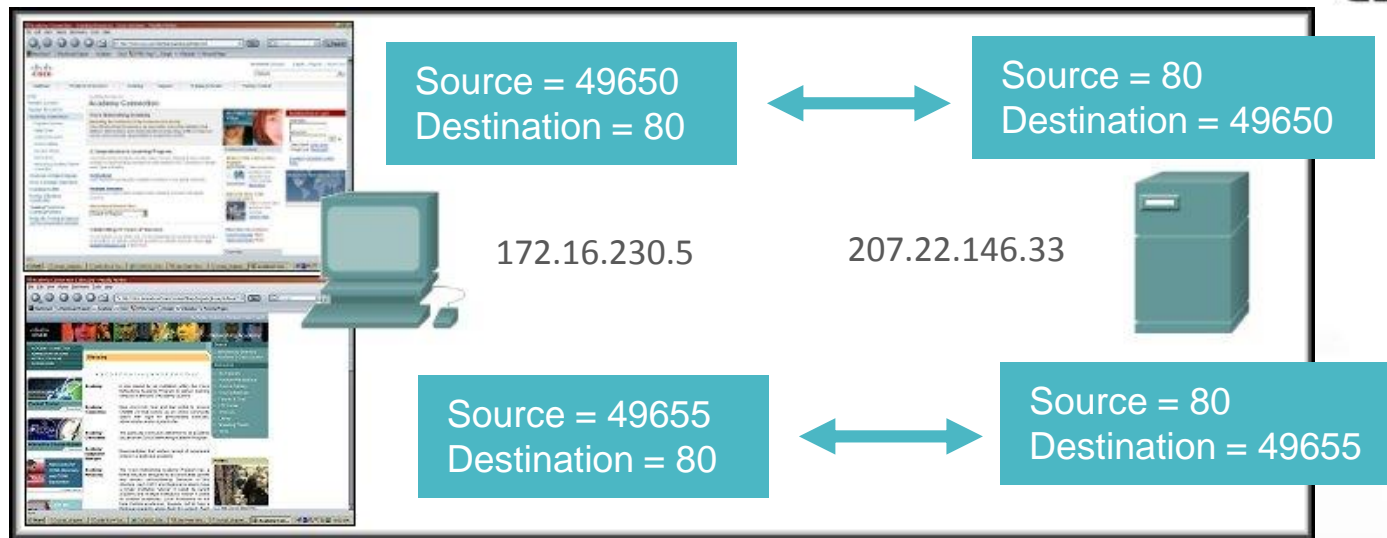


Source = 80  
Destination = 49650

- Servidor responde com a página web
  - Configura a porta fonte para 80 e usa a porta de origem do cliente como a porta de destino



- Note como as portas de origem e destino são usadas



- Como a camada de transporte separa os fluxos?

- Socket (IP Address:Port)

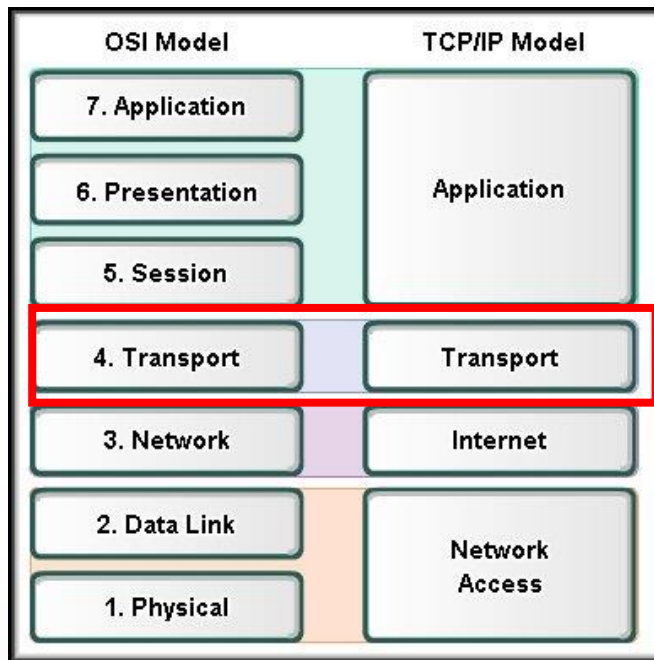
172.16.230.5:49650

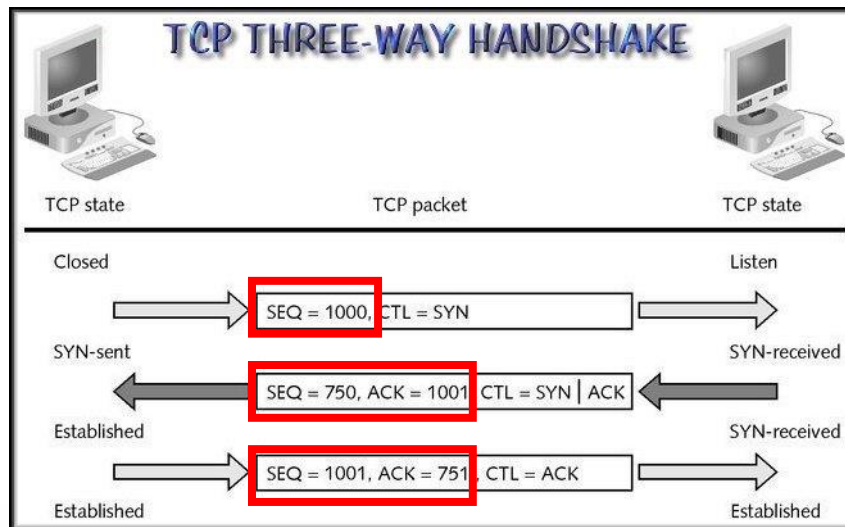
172.16.230.5:49655



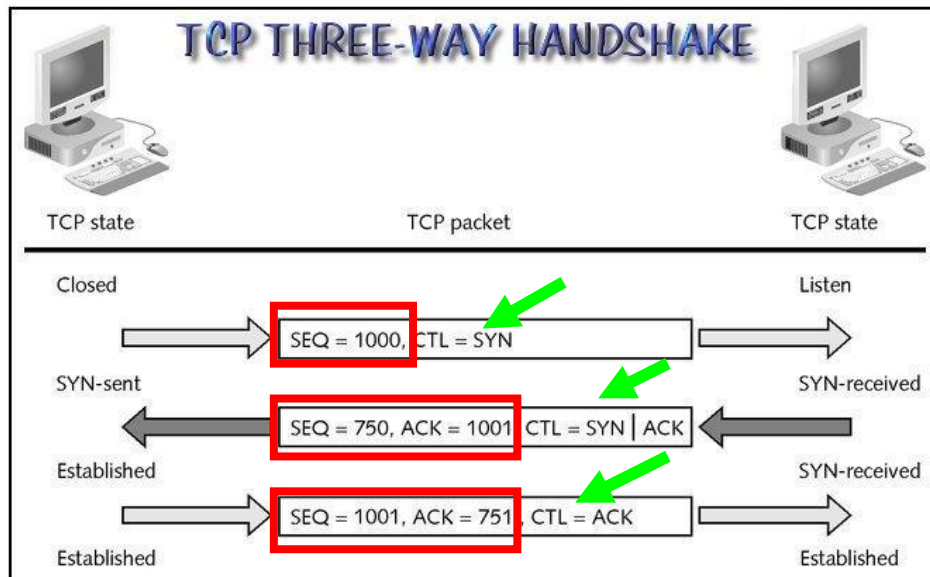
207.22.146.33:49650

207.22.146.33:49655

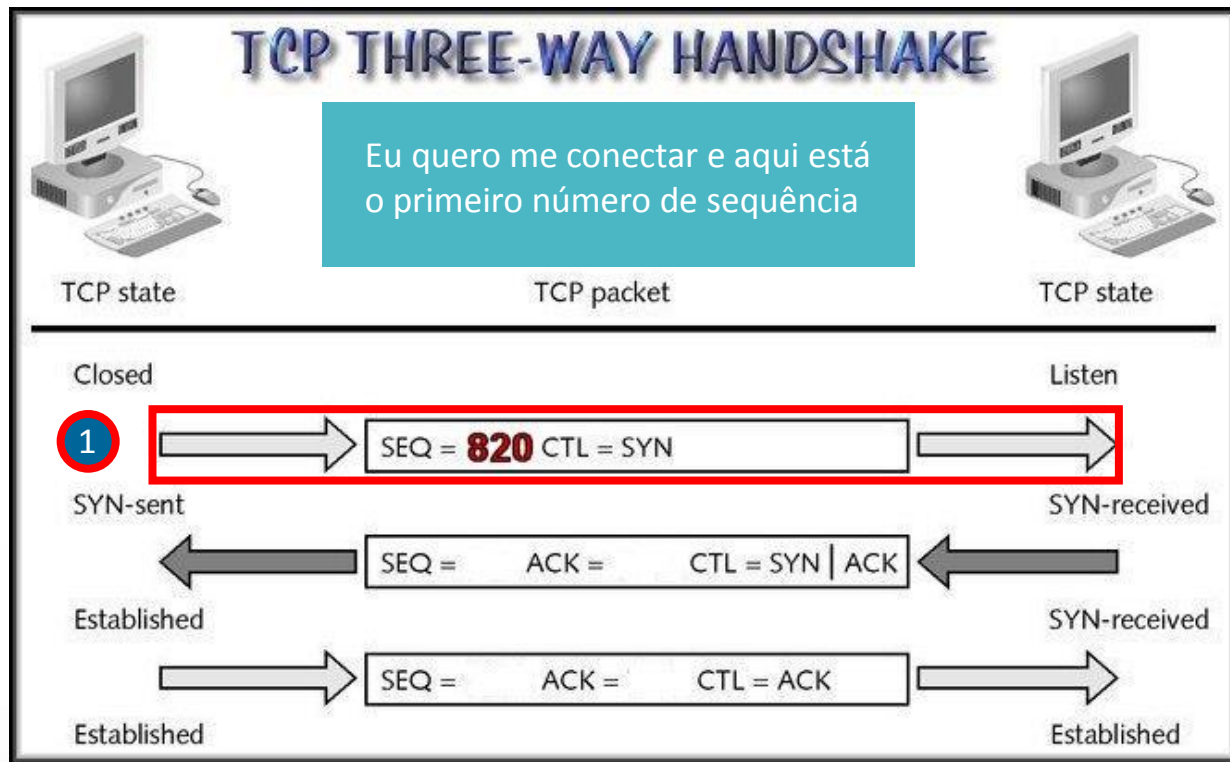




- Para uma conexão ser estabelecida, as duas estações finais devem sincronizar os números de sequencias iniciais (ISNs)
- O ISN é um valor sequencial inicial usado quando uma conexão TCP é estabelecida

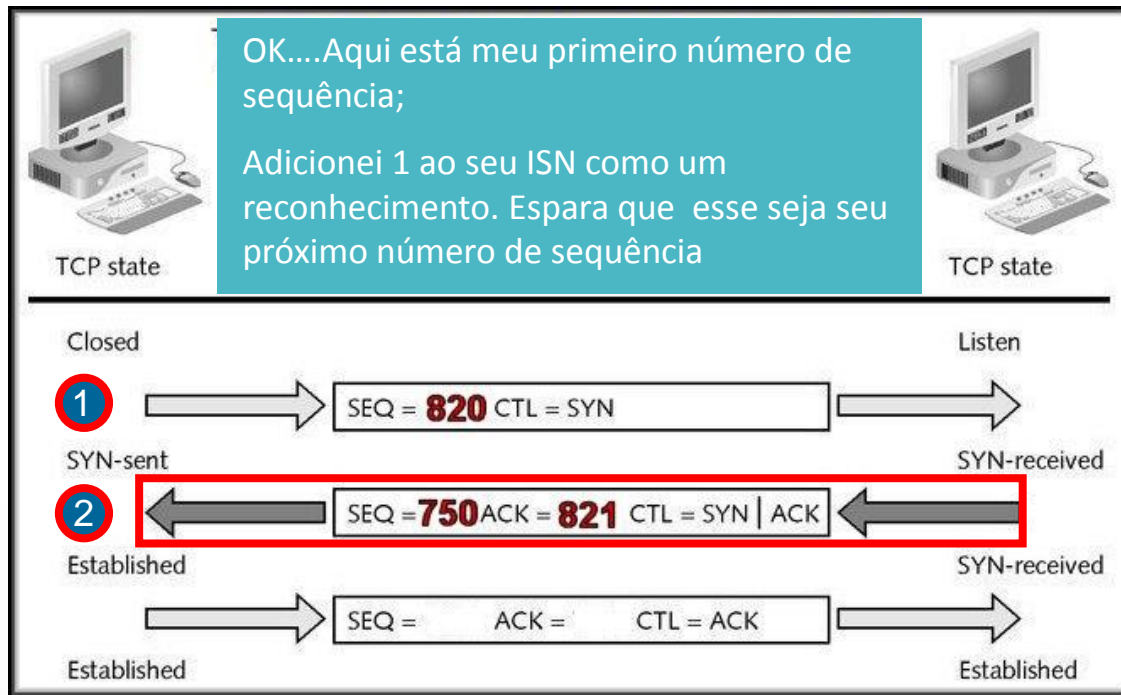


- Números de sequencia são usados para rastrear a ordem dos segmentos e assegurar que não houve perda de segmentos na transmissão
- Os campos Flag são usados para identificar o tipo e segmento

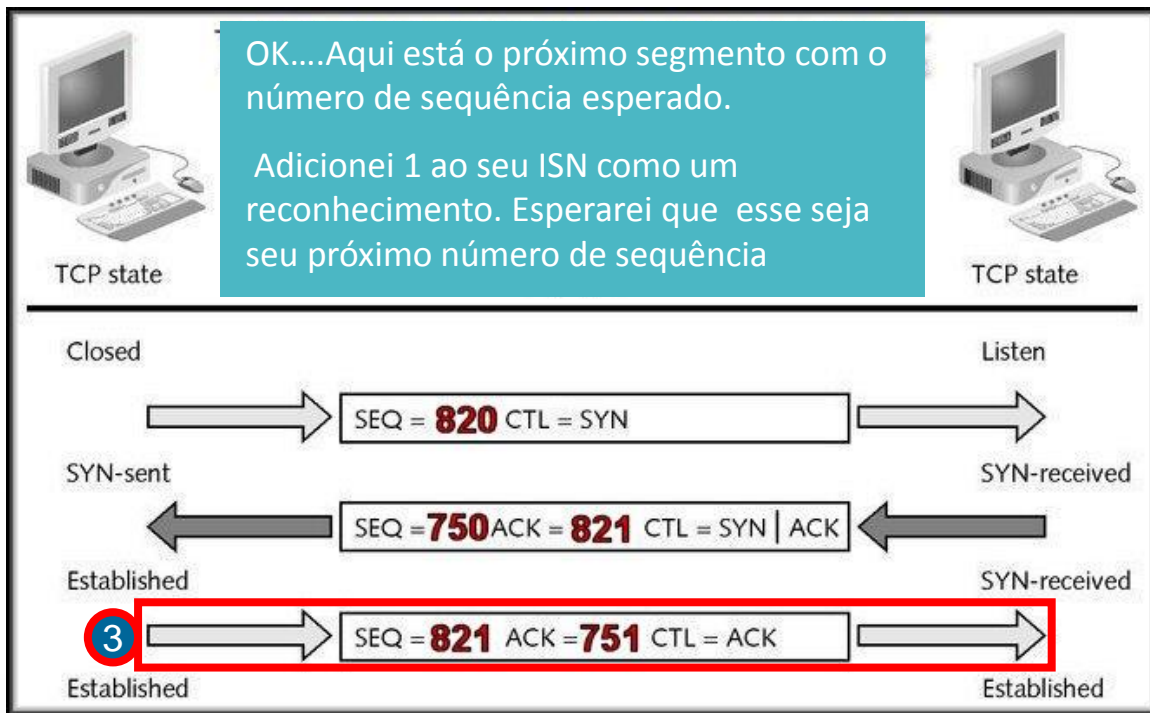


## Protocolos de Redes e de Computadores

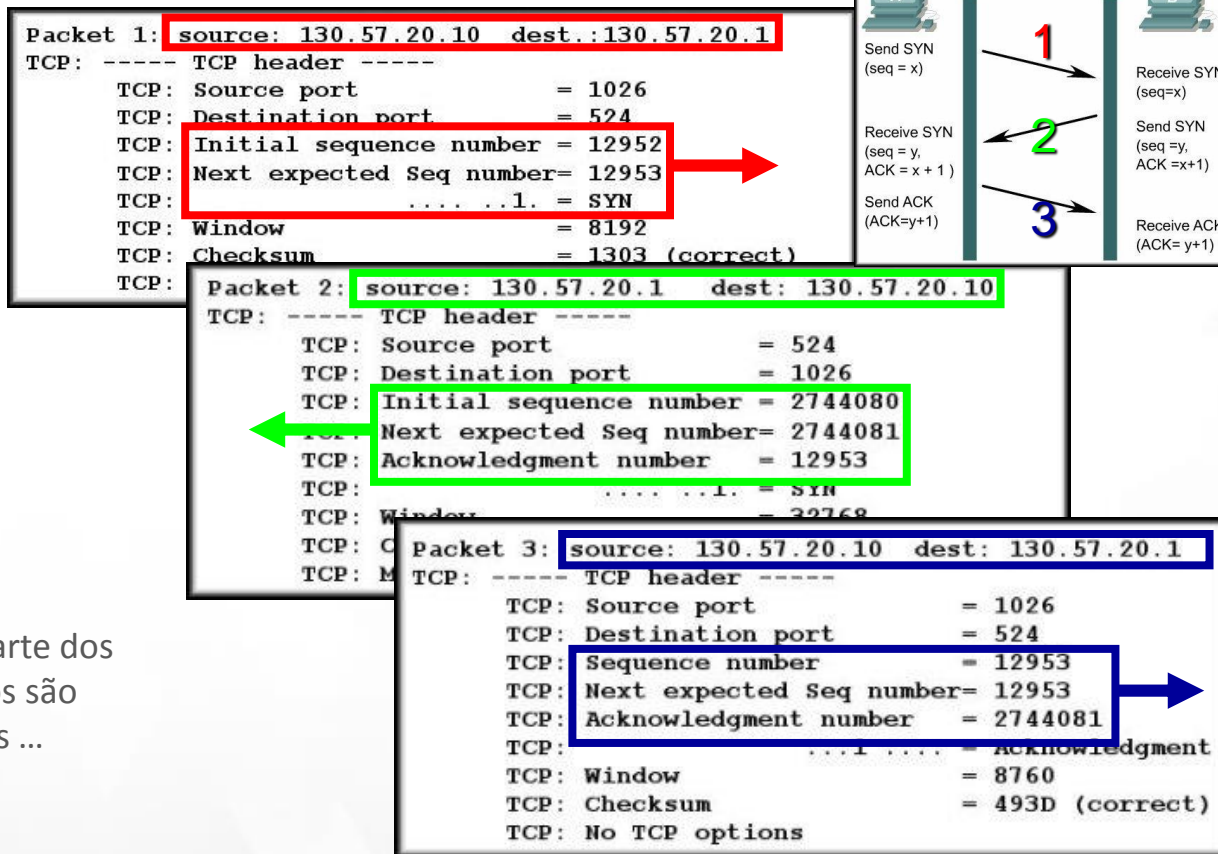
### Handshake Triplo do TCP



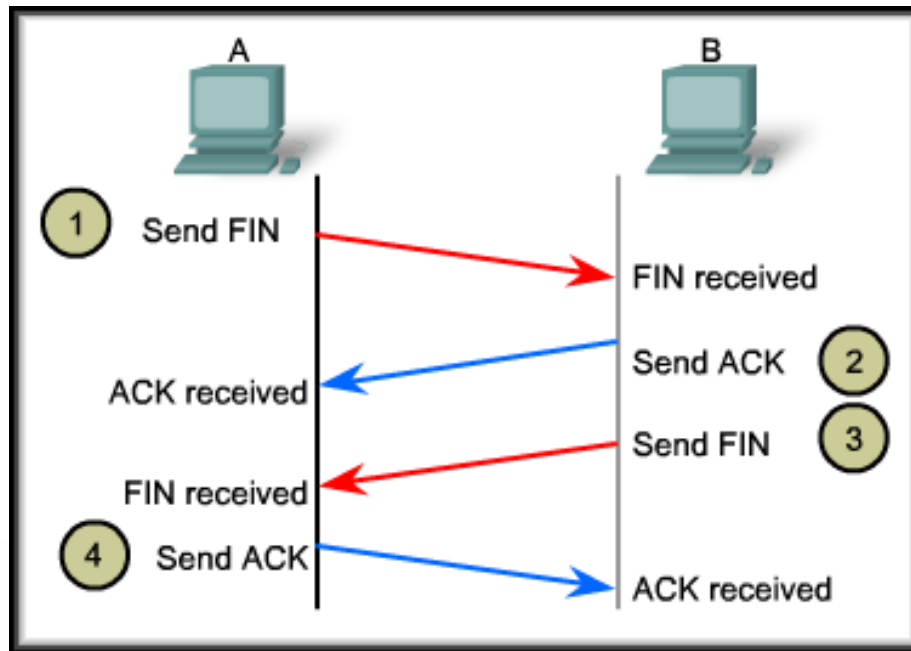




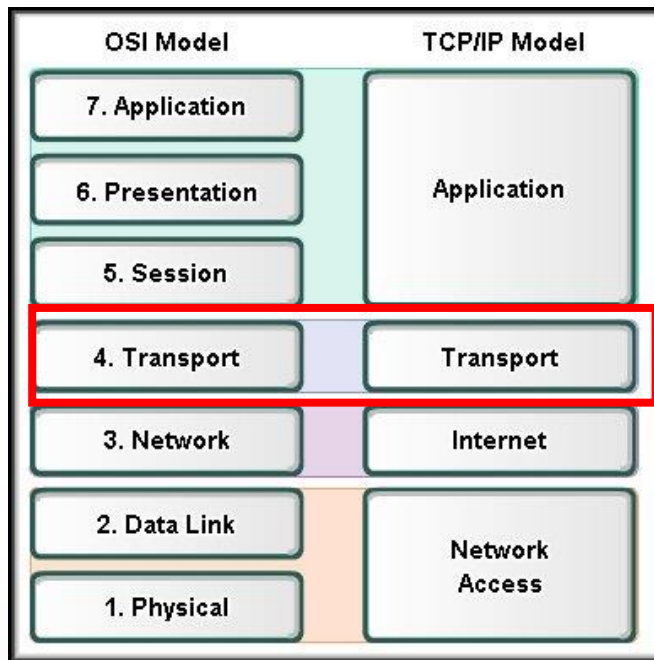
## Handshake Triplo do TCP



- Apenas parte dos cabeçalhos são mostrados ...

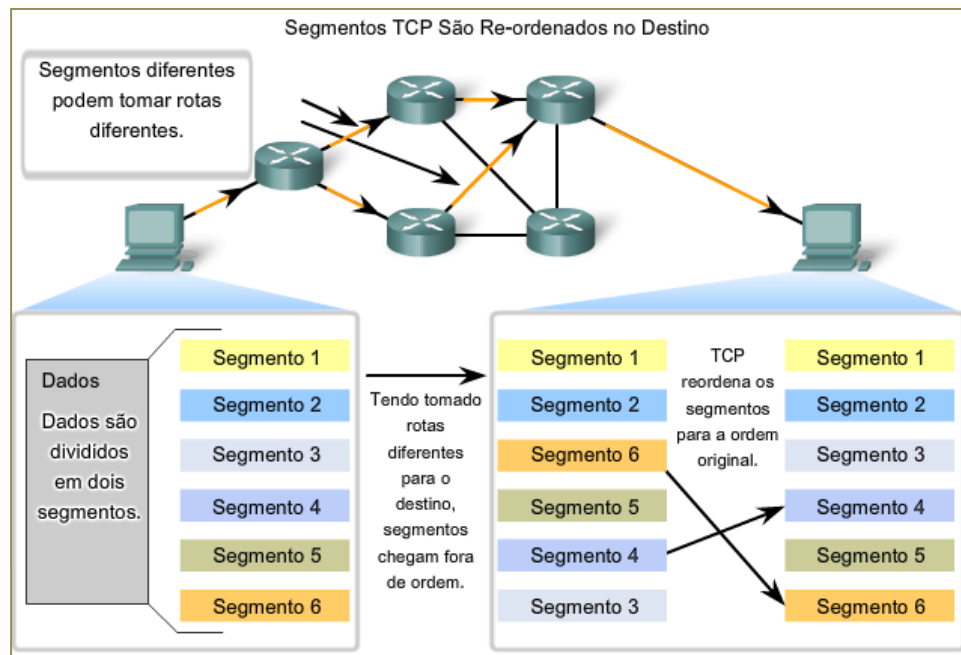


- Processo de quatro passos usando os campos Flag e número de sequência



## Protocolos de Redes e de Computadores

### Reagrupamento de Segmentos TCP



- Lembre-se que a camada de transporte deve reagrupar os segmentos na ordem correta

## Protocolos de Redes e de Computadores

### Confirmação TCP com Janelamento

Usados juntos para confirmar o recebimento da mensagem

Porta de

Porta de Destino

Número de  
Sequência

Números de  
reconhecimento

...

Iniciar com byte no. 1,  
Estou enviando 10  
bytes.



Rede



Recebi 10 bytes  
iniciando com byte no1.  
Espero o byte no. 11 em  
seguida.

Origem	Destino	Seq.	Rec.	...
1028	23	1	1	...

10 bytes

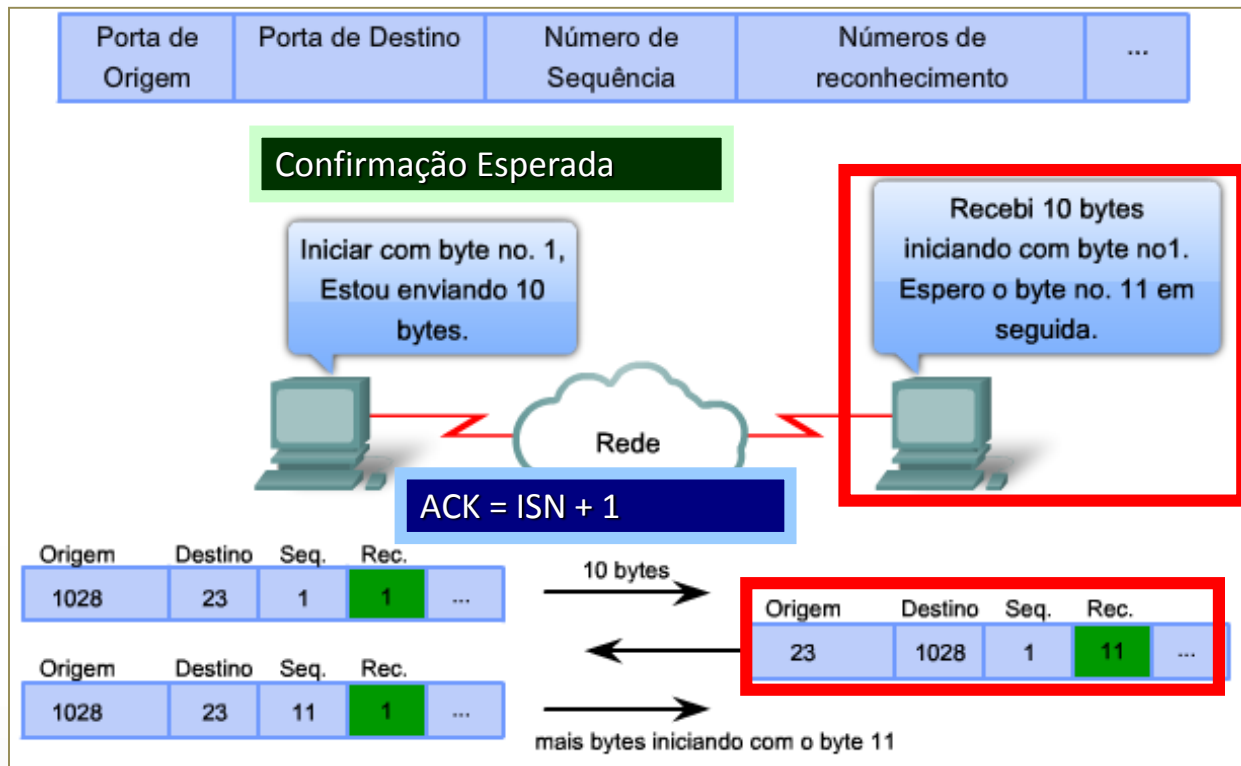
Origem	Destino	Seq.	Rec.	...
1028	23	11	1	...

Origem	Destino	Seq.	Rec.	...
23	1028	1	11	...

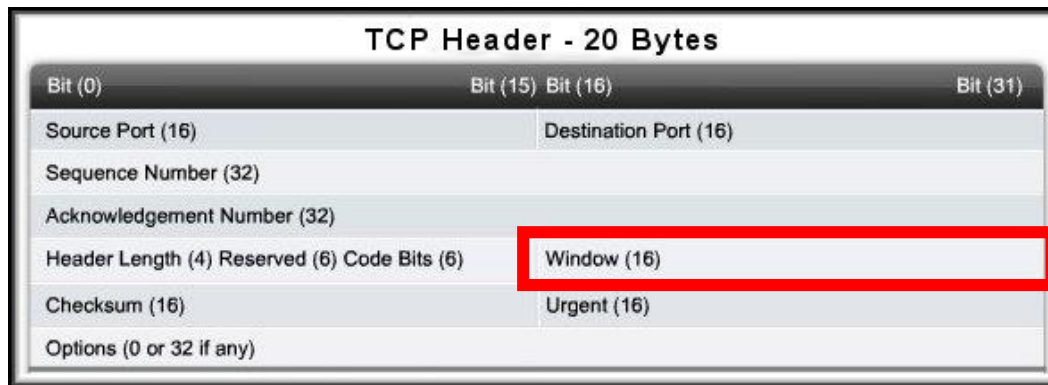
mais bytes iniciando com o byte 11

## Protocolos de Redes e de Computadores

### Confirmação TCP com Janelamento



- Com uma janela de tamanho 10, cada segmento carrega apenas 10 bytes de dados e deve aguardar pela confirmação de recebimento antes de outro segmento ser transmitido

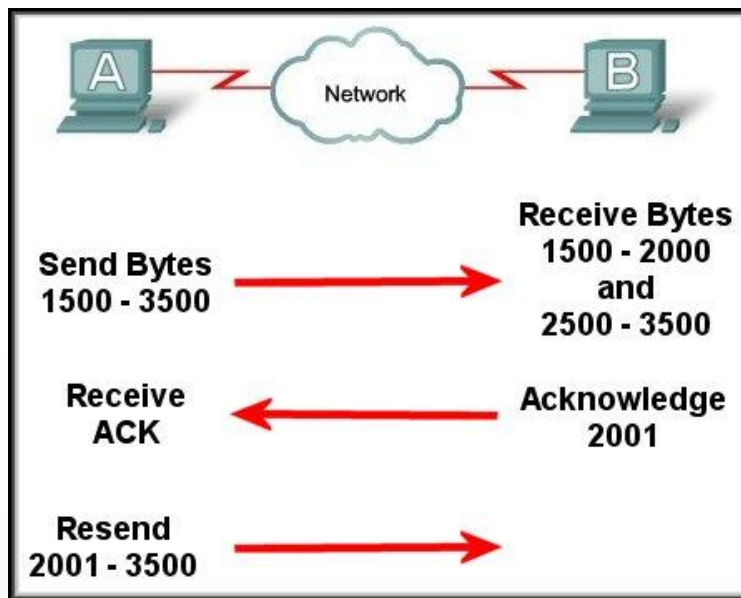


- Tamanho da Janela:
  - Quantidade de dados que pode ser enviado antes que uma confirmação seja recebida
  - Determinado pelo campo no cabeçalho TCP



### Retransmissão TCP

- Um serviço de host de destino usando TCP geralmente reconhece os dados apenas para bytes sequenciais contíguos
- Se estiver faltando um ou mais segmentos, apenas os dados nos segmentos que completam o fluxo serão confirmados

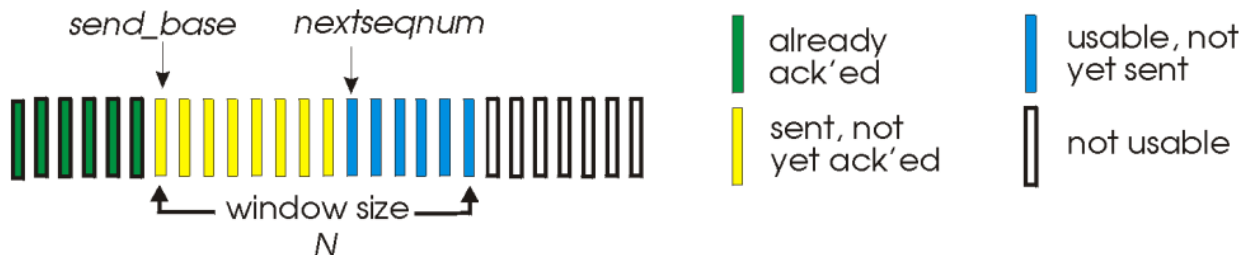


# Protocolos de retransmissão

## Volta-N

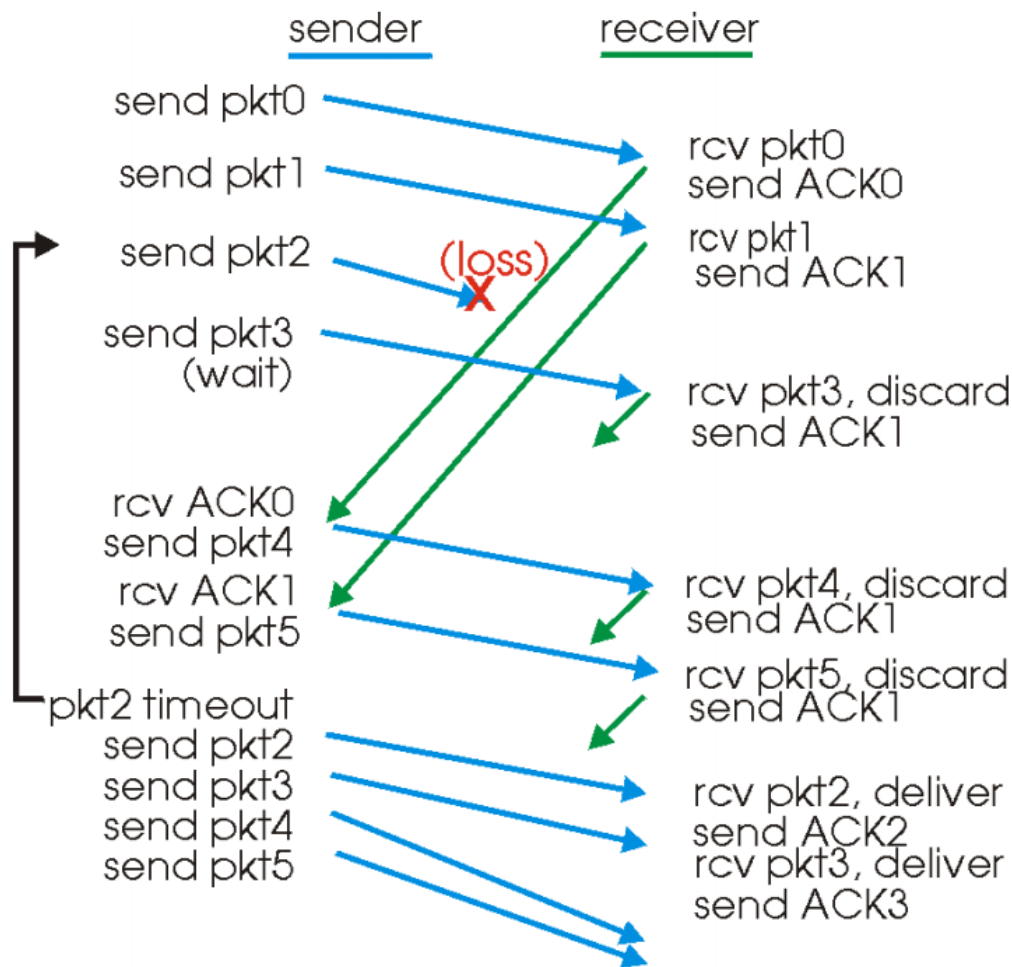
### Remetente:

- no. de seq. de k-bits no cabeçalho do pacote
- admite “janela” de até N pacotes consecutivos não reconhecidos



- **ACK(n):** reconhece todos pacotes, até e inclusive no. de seq n - “ACK cumulativo”
  - pode receber ACKs duplicados (veja receptor)
- temporizador para cada pacote em trânsito
- **timeout(n):** retransmite pacote n e todos os pacotes com no. de seq maiores na janela

## Volta-N em ação

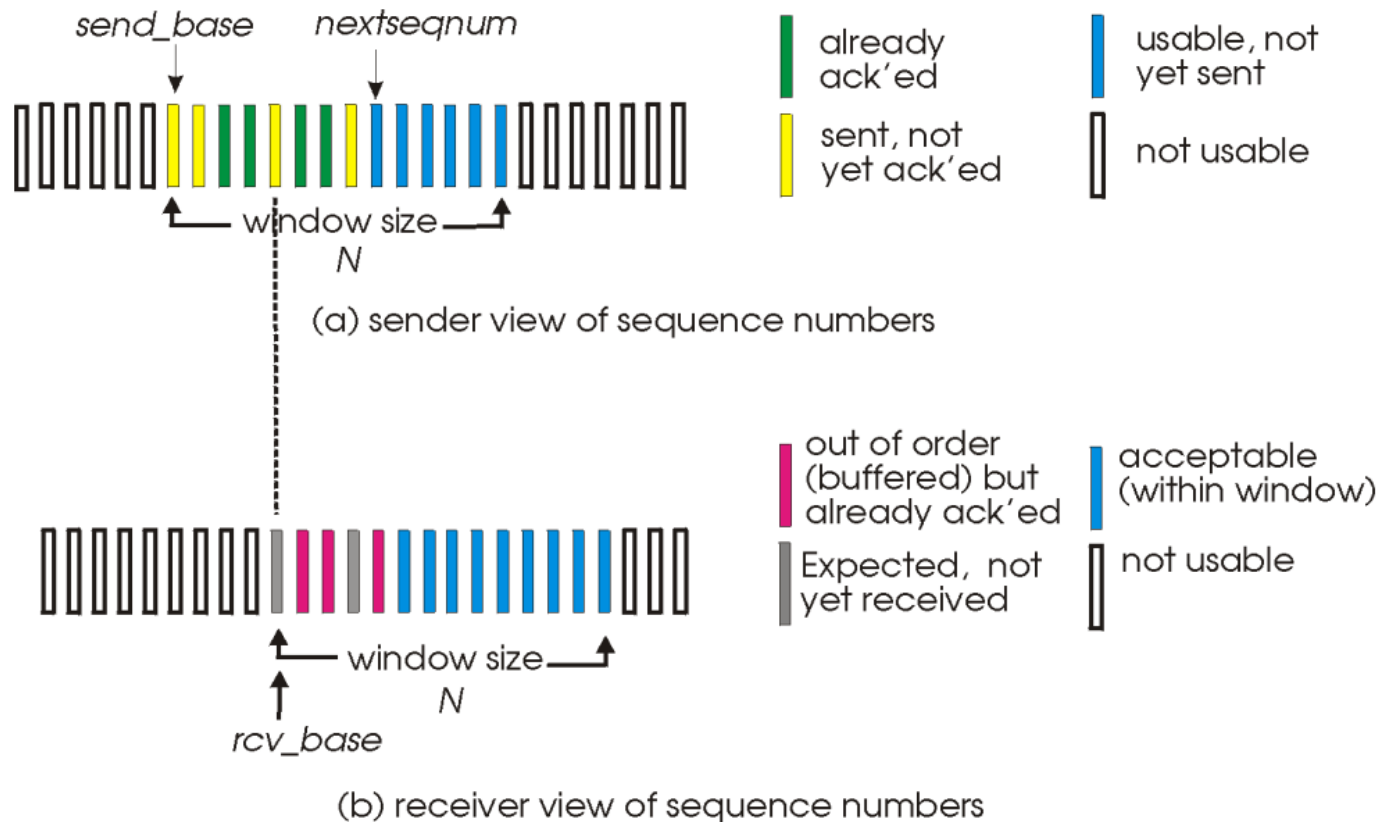




# Retransmissão seletiva

- ❑ receptor reconhece *individualmente* todos os pacotes recebidos corretamente
  - armazena pacotes no buffer, conforme precisa, para posterior entrega em-ordem à camada superior
- ❑ remetente apenas re-envia pacotes para os quais ACK não recebido
  - temporizador de remetente para cada pacote sem ACK
- ❑ janela do remetente
  - N nos. de seq consecutivos
  - outra vez limita nos. de seq de pacotes enviados, mas ainda não reconhecidos

# Retransmissão seletiva: janelas de remetente, receptor





# Retransmissão seletiva

## remetente

### **dados de cima:**

- ❑ se próx. no. de seq na janela, envia pacote

### **timeout(n):**

- ❑ reenvia pacote n, reiniciar temporizador

**ACK(n)** em  $[sendbase, sendbase + N]$ :

- ❑ marca pacote n “recebido”
- ❑ se n for menor pacote não reconhecido, avança base da janela ao próx. no. de seq não reconhecido

## receptor

### **pacote n em**

$[rcvbase, rcvbase + N - 1]$

- ❑ envia ACK(n)
- ❑ fora de ordem: buffer
- ❑ em ordem: entrega (tb. entrega pacotes em ordem no buffer), avança janela p/ próxima pacote ainda não recebido

### **pacote n em**

$[rcvbase - N, rcvbase - 1]$

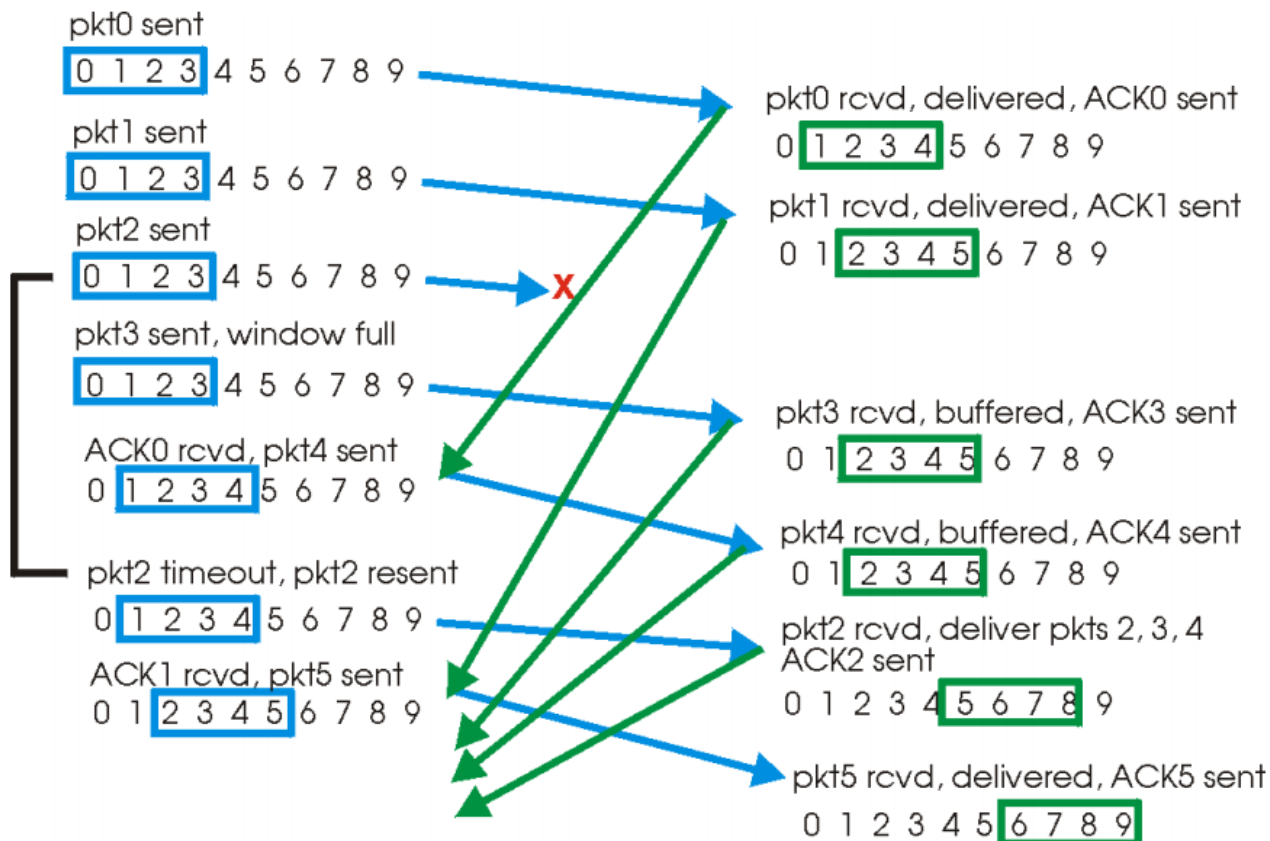
- ❑ ACK(n)

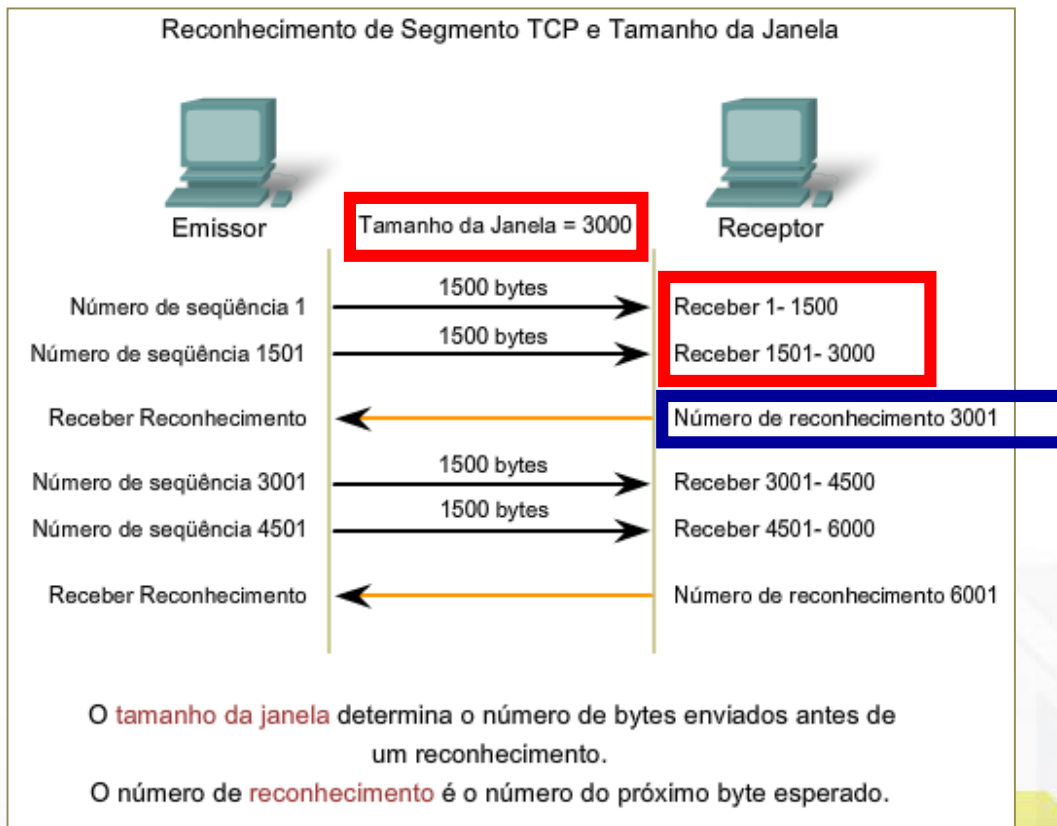
### **senão:**

- ❑ ignora



# Retransmissão seletiva em ação







### Controle de Fluxo TCP

Packet 1: source: 130.57.20.10 dest.:130.57.20.1

TCP: ----- TCP header -----

TCP: Source port = 1026

TCP: Destination port = 524

TCP: Initial sequence number = 12952

TCP: Next expected Seq number= 12953

TCP: .....1. = SYN

**TCP: Window = 8192**

TCP: Checksum = 1303 (correct)

TCP:

Packet 2: source: 130.57.20.1 dest: 130.57.20.10

TCP: ----- TCP header -----

TCP: Source port = 524

TCP: Destination port = 1026

TCP: Initial sequence number = 2744080

TCP: Next expected Seq number= 2744081

TCP: Acknowledgment number = 12953

TCP: .....1. = SYN

**TCP: Window = 32768**

TCP: Checksum = 9387 (correct)

TCP: Maximum

Packet 3: source: 130.57.20.10 dest: 130.57.20.1

TCP: ----- TCP header -----

TCP: Source port = 1026

TCP: Destination port = 524

TCP: Sequence number = 12953

TCP: Next expected Seq number= 12953

TCP: Acknowledgment number = 2744081

TCP: .....1. = Acknowledgment

**TCP: Window = 8760**

TCP: Checksum = 493D (correct)

TCP: No TCP options

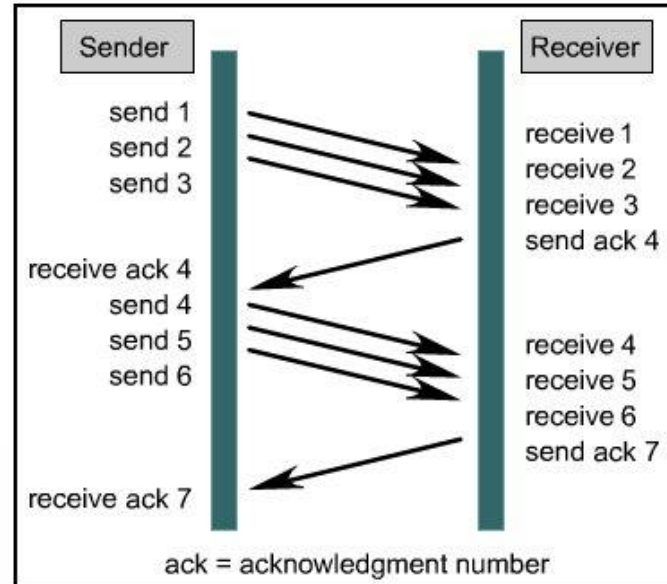
Quantidade de dados que  
pode ser enviado antes de  
um reconhecimento

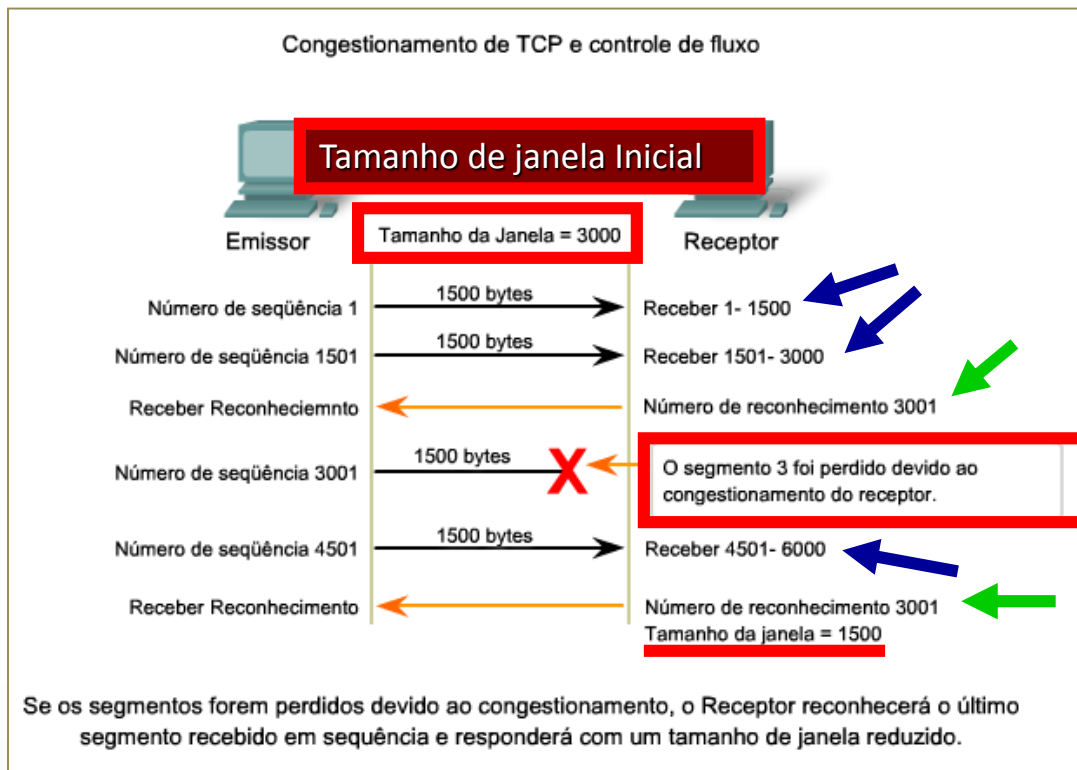
O tamanho da janela inicial  
é determinado durante o  
handshake triplo

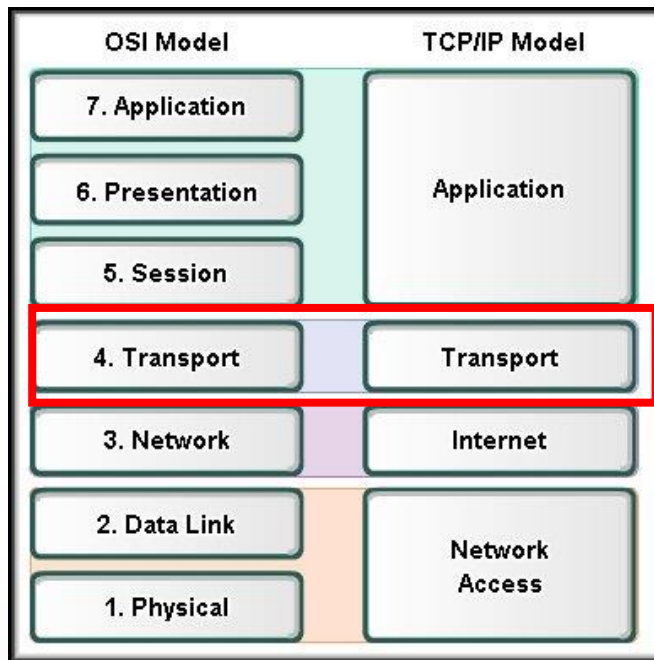
## Protocolos de Redes e de Computadores

### Tamanho Dinâmico da Janela TCP

- Serviço TCP é Full-duplex:
  - Significa que dados podem trafegar em cada direção, simultaneamente
  - Tamanhos de janela, número de sequência e número de reconhecimento são independentes de cada fluxo de dados
- O host de recebimento envia o valor do tamanho de janela ao remetente para indicar o número de bytes que ele está preparado para receber como parte desta sessão
  - Se o destino precisar diminuir a velocidade da taxa de comunicação, **ele pode enviar um valor de tamanho de janela pequeno para a origem como parte de uma confirmação**
  - Se não há restrição (nem perda), **o tamanho de janela continuará a aumentar**

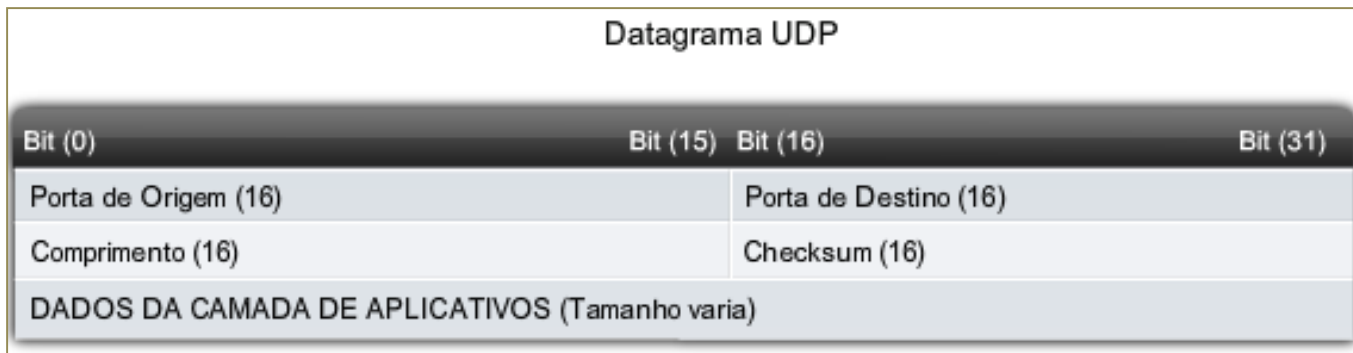






## Protocolos de Redes e de Computadores

### User Datagram Protocol (UDP)

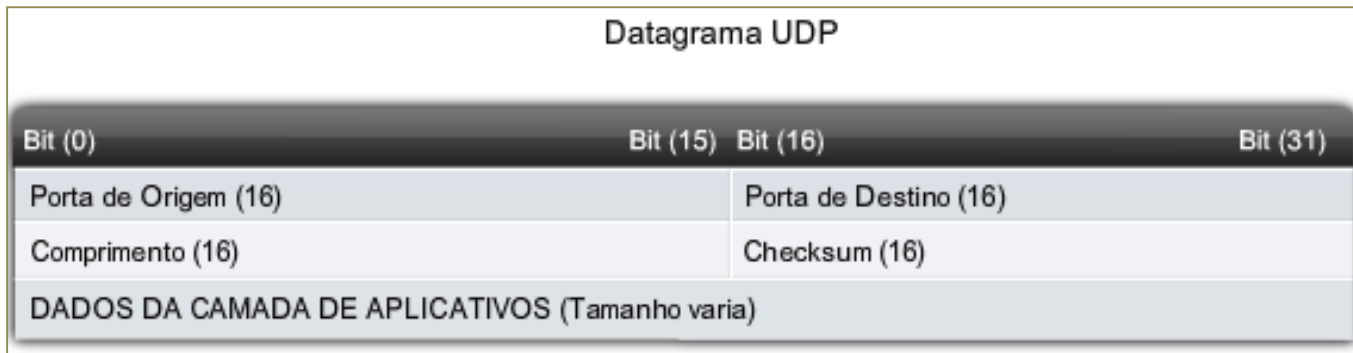


- Sem-conexão
- Entrega de melhor esforço (Best Effort)

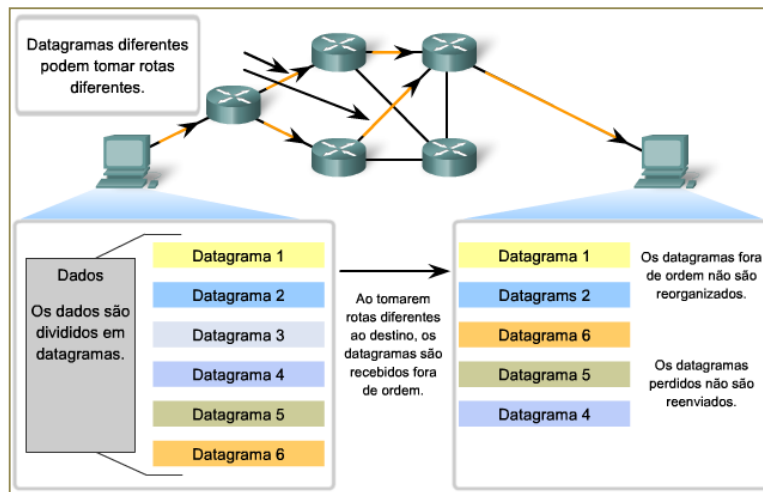
Sem Handshake triplo

#### Exemplos de Aplicações

- Domain Name System (DNS)
- Jogos online
- Voice over IP (VoIP)
- Dynamic Host Configuration Protocol (DHCP)
- Trivial File Transfer Protocol (TFTP)



- Baixo Overhead:
  - Sem-Conexão:
- Sem estabelecimento de conexão, como o TCP
  - Não confiável ou entrega de “Melhor Esforço”:
  - Sem detecção de erro
  - Sem controle de Fluxo
  - Sem controle de congestionamento
  - Sem número de sequência para entrega ordenada
  - Quando uma aplicação tem dado a enviar, ela simplesmente envia o dado



- Quando múltiplos datagramas são enviados a um destino, eles podem tomar diferentes caminhos e chegar na ordem errada.
- O UDP não rastreia os números de sequência da forma que o TCP faz.
- O UDP não tem um modo para reordenar os datagramas na sua ordem de transmissão.
- UDP simplesmente reagrupa os dados na ordem que eles foram recebidos e os encaminha para a aplicação

Port Number	Application	Layer 4 Protocol	Description
20	FTP	TCP	File Transfer Protocol – Data
21	FTP	TCP	File Transfer Protocol – Control Commands
23	TELNET	TCP	Terminal connection
25	SMTP	TCP	Simple Mail Transfer Protocol - Email
53	DNS	UDP	Domain Name System
67,68	DHCP	UDP	Dynamic Host Configuration Protocol
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP	Hypertext Transfer Protocol