

Camada de Transporte

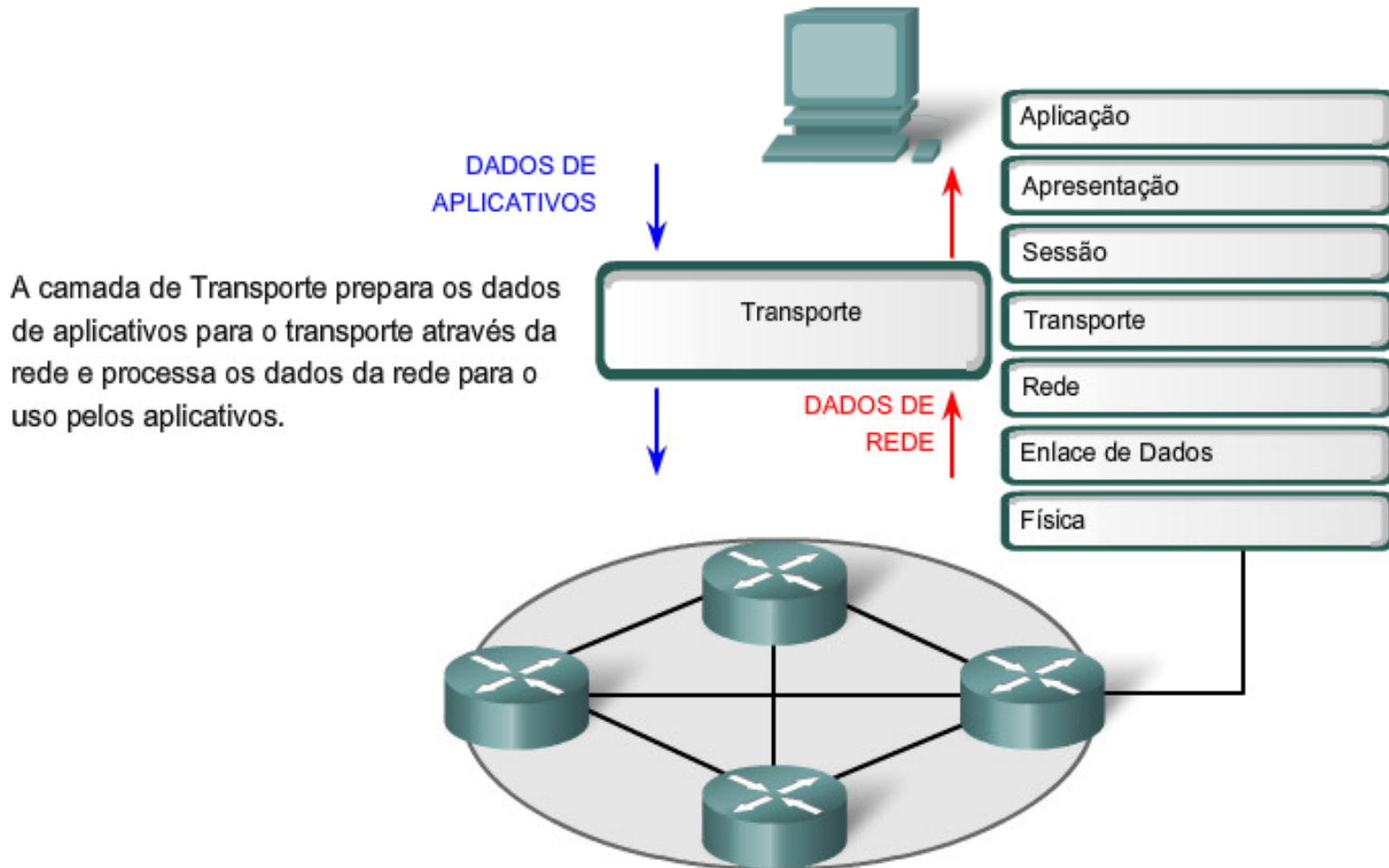
CCNA 1: Capítulo 4

Livro: Redes de Computadores e Internet. Autor: Kurose e Ross

Objetivos

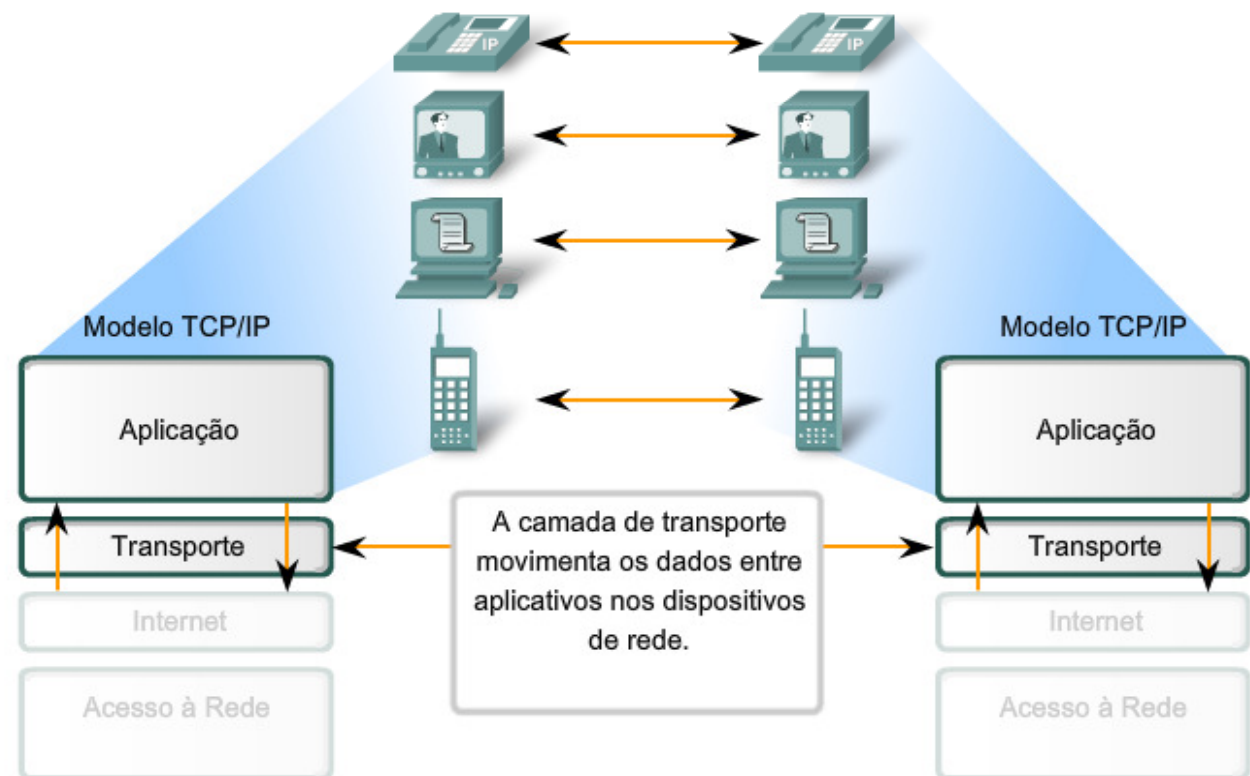
- Explicar o papel dos protocolos da camada de Transporte e os serviços que suportam a comunicação através de uma rede de dados
- Analisar a aplicação e operação de mecanismos do TCP que suportam a confiabilidade
- Analisar a aplicação e operação de mecanismos do TCP que suportam a multiplexação/demultiplexação e gerenciamento de perda de dados
- Analisar a operação do UDP que suportam a comunicação entre dois processos em sistemas finais diferentes.

Camada de Transporte

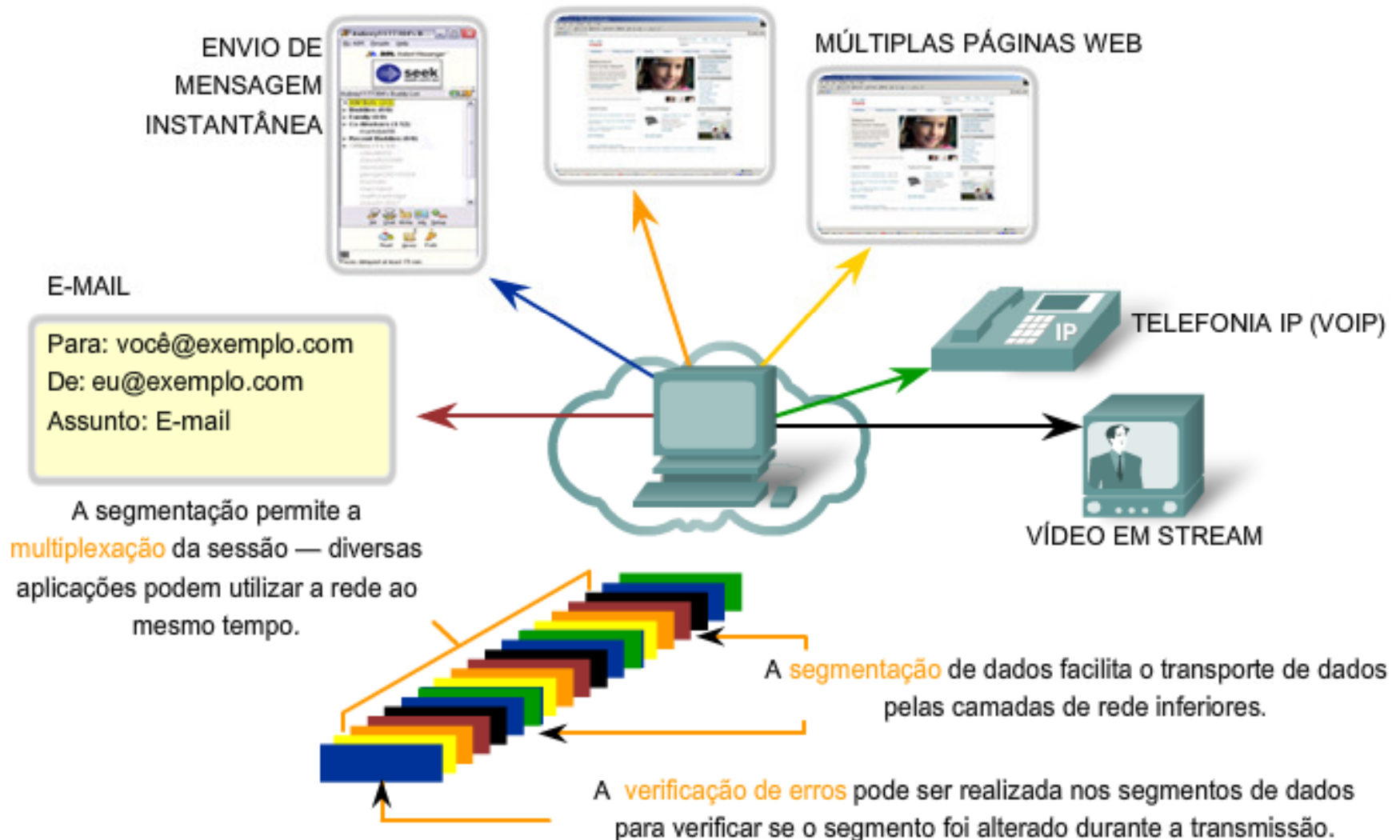


Funções da Camada de Transporte

- Segmentação de dados
- Multiplexação/Demultiplexação
- Comunicação entre processos
- Identificação de processos



Segmentação/Multiplexação/Demultiplexação



Comunicação entre processos

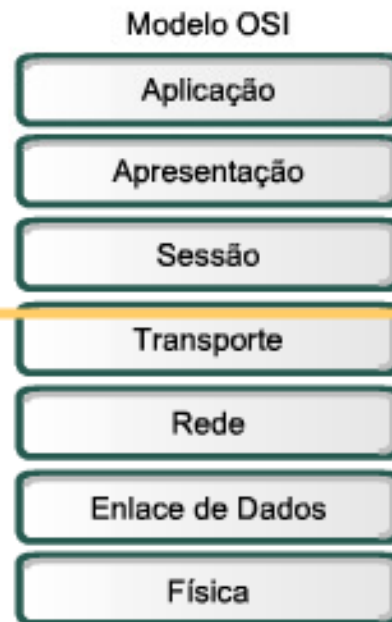


- Telefonia IP
- Vídeo Streaming

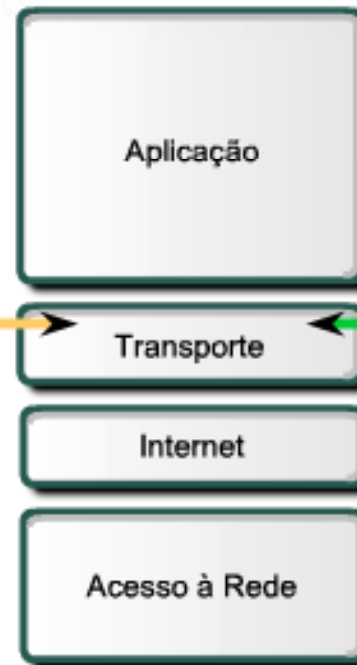
- SMTP/POP (E-mail)
- HTTP

Propriedades de Protocolo
Requisitadas

- Rápido
- Auxiliar baixo
- Não solicita confirmações
- Não re-envia dados perdidos
- Entrega dados como eles chegaram



Modelo TCP/IP



Propriedades de Protocolo
Requisitadas

- Confiável
- Confirma dados
- Re-envia dados perdidos
- Entrega dados na ordem que foram enviados

Os programadores de Aplicação escolhem o protocolo da Camada de Transporte adequado com base na natureza da aplicação.

Cabeçalho do TCP e UDP

Segmento TCP

Bit (0)	Bit (15)	Bit (16)	Bit (31)
Porta de Origem (16)		Porta de Destino (16)	
Número de Seqüência (32)			
Número de Reconhecimento (32)			
Comprimento do Cabeçalho (4) Reservado (6) Bits de Código (6)		Janela (16)	
Checksum (16)		Urgente (16)	
Opções (0 ou 32, se houver)			
DADOS DA CAMADA DE APLICATIVOS (Tamanho varia)			

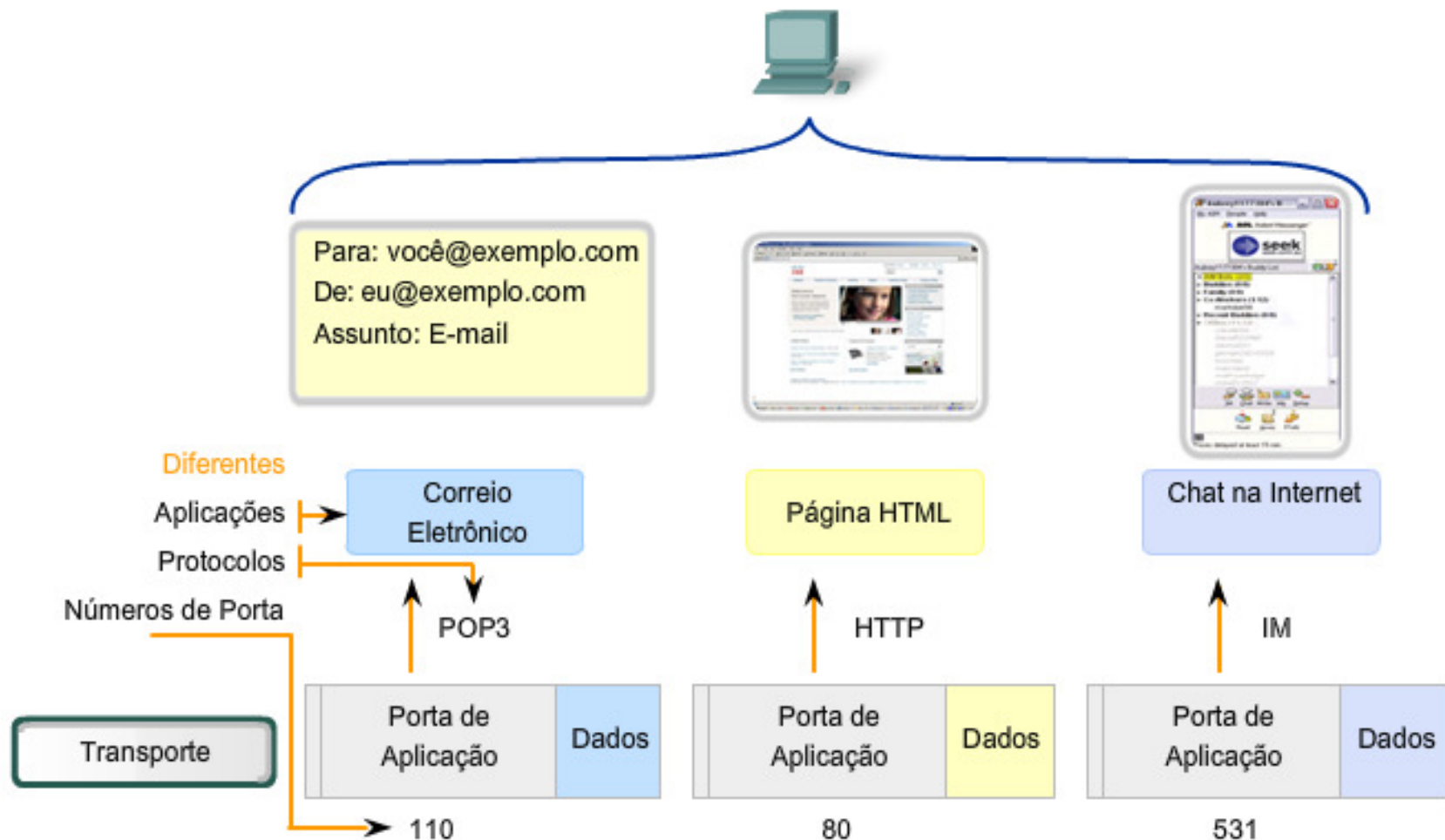
↑
20 Bytes
↓

Datagrama UDP

Bit (0)	Bit (15)	Bit (16)	Bit (31)
Porta de Origem (16)		Porta de Destino (16)	
Comprimento (16)		Checksum (16)	
DADOS DA CAMADA DE APLICATIVOS (Tamanho varia)			

↑
8 Bytes
↓

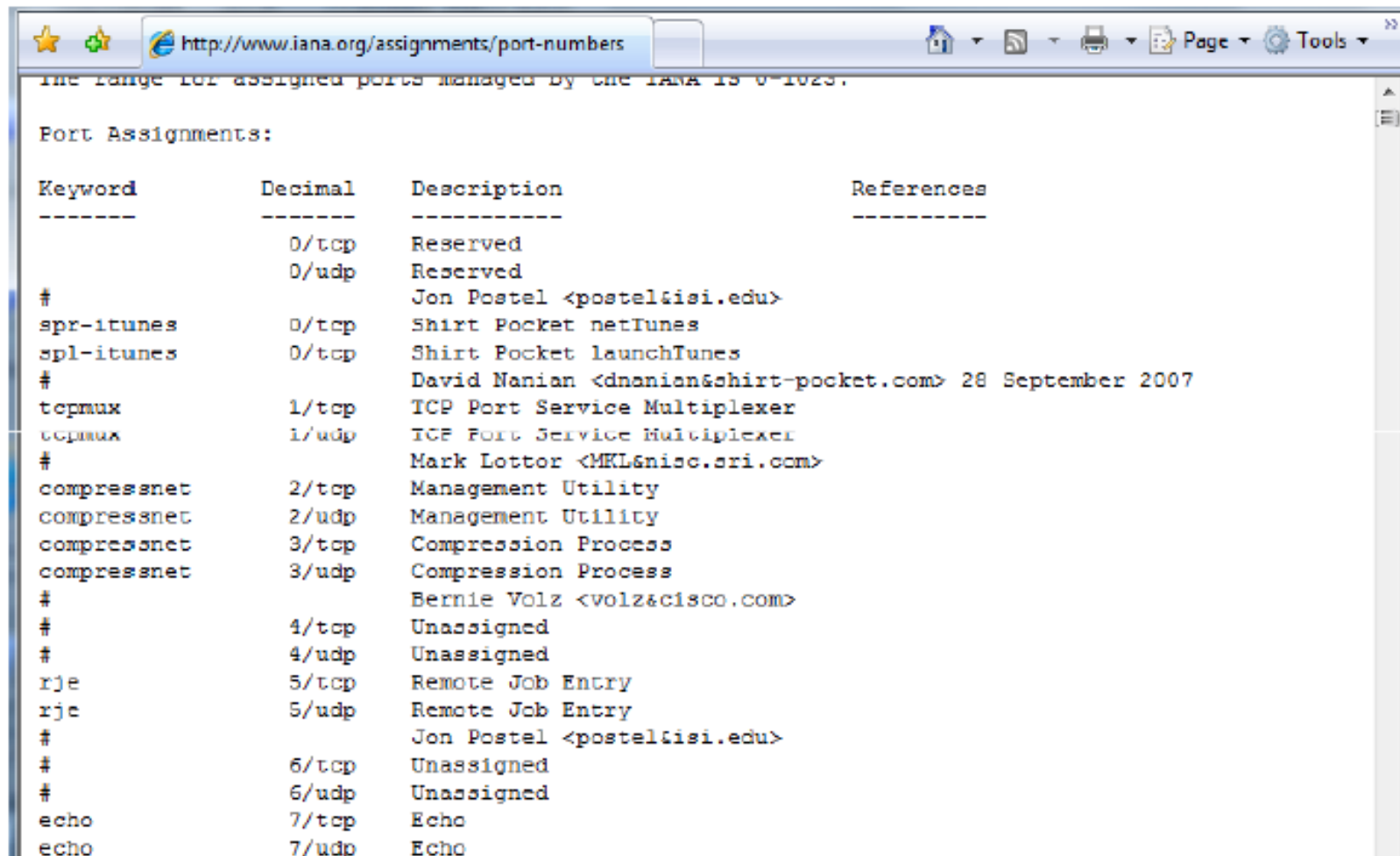
Endereçamento de Porta



Dados para diferentes aplicações são direcionados à aplicação correta porque cada aplicação tem um número de porta único.

Endereçamento de Porta

- IANA - The Internet Assigned Numbers Authority

A screenshot of a web browser displaying the IANA Port Assignments page. The browser's address bar shows the URL 'http://www.iana.org/assignments/port-numbers'. The page content includes a title 'Port Assignments:' and 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 TCP and UDP. The table is partially visible, showing entries from port 0 to 7.

The range for assigned ports managed by the IANA is 0-1023.

Port Assignments:

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	
tcprmx	1/tcp	TCP Port Service Multiplexer	
tcprmx	1/udp	TCP Port Service Multiplexer	
#		Mark Lotter <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	

Endereçamento de Porta

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Well Known TCP Ports

21	FTP
23	Telnet
25	SMTP
80	HTTP
110	POP3
194	Internet Relay Chat (IRC)
443	Secure HTTP (HTTPS)

Well Known UDP Ports:

69	TFTP
520	RIP

Well Known TCP/UDP Common Ports:

53	DNS
161	SNMP
531	AOL Instant Messenger, IRC



Número de porta bem conhecida ou registrada

Portas bem conhecidas (0 a 1023)

- Reservada para serviços e aplicações comuns

- **Cliente:** TCP porta destino
- **Servidor:** TCP porta da fonte



Número de porta bem conhecida ou registrada

Endereçamento de Porta

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports

Registered TCP Ports:
 1863 MSN Messenger
 8008 Alternate HTTP
 8080 Alternate HTTP

Registered UDP Ports:
 1812 RADIUS Authentication Protocol
 2000 Cisco SCCP (VoIP)
 5004 RTP (Voice and Video Transport Protocol)
 5060 SIP (VoIP)

Registered TCP/UDP Common Ports:
 1433 MS SQL
 2948 WAP (MMS)



Número de porta bem conhecida ou registrada



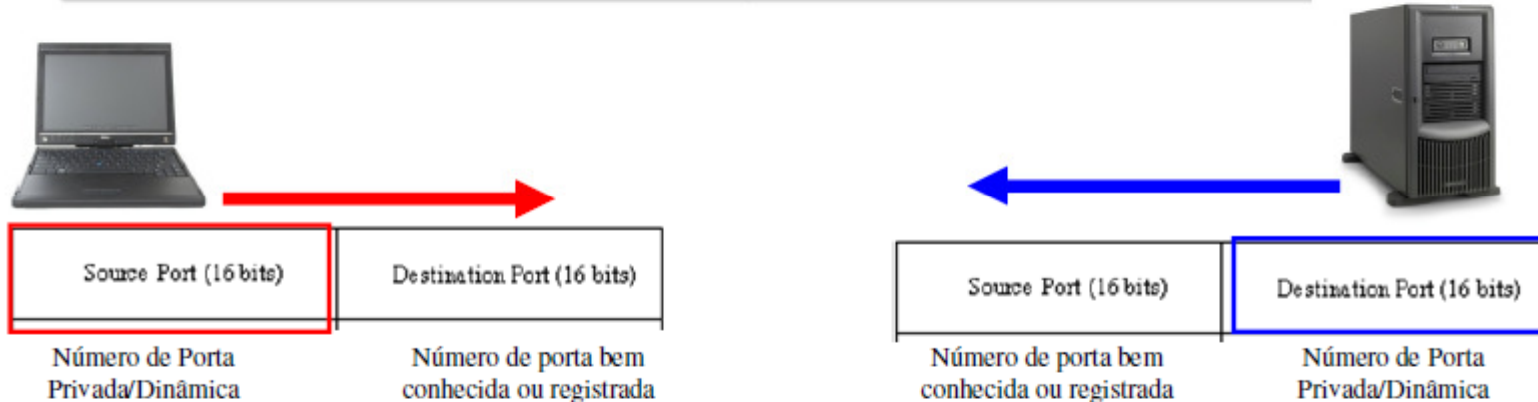
Número de porta bem conhecida ou registrada

Portas Registradas (1024 a 49151)

- Atribuídas a processos e aplicações de usuário.
- Aplicações não comuns.
 - **Cliente:** TCP destination port
 - **Servidor:** TCP source port
- Pode ser usada como portas dinâmicas ou privadas (próximo).

Endereçamento de Porta

Port Number Range	Port Group
0 to 1023	Well Known (Contact) Ports
1024 to 49151	Registered Ports
49152 to 65535	Private and/or Dynamic Ports



Portas Privadas ou Dinâmicas(49152 a 65535)

- Também conhecidas como portas efêmeras
- Geralmente atribuídas a aplicações clientes quando iniciam uma conexão
 - **Cliente:** TCP porta fonte
 - **Server:** TCP porta destino
- Podem também incluir a faixa de Portas Registradas (1024 a 49151)

Endereçamento de Porta

The screenshot displays a Windows XP desktop with three open applications:

- Google - Microsoft Internet Explorer:** Shows the Google homepage.
- Cisco Systems, Inc - Microsoft Internet Explorer:** Shows the Cisco Systems homepage.
- C:\WINDOWS\system32\cmd.exe:** A command prompt window showing the output of the command `netstat -n`.

The command prompt output is as follows:

```
C:\>netstat -n

Active Connections

Proto Local Address          Foreign Address         State
TCP   172.17.150.112:1033    172.16.1.44:524        ESTABLISHED
TCP   172.17.150.112:1034    172.16.1.44:524        ESTABLISHED
TCP   172.17.150.112:1042    205.188.9.73:5190      ESTABLISHED
TCP   172.17.150.112:1050    64.12.165.95:5190      ESTABLISHED
TCP   172.17.150.112:1069    207.62.185.140:143     ESTABLISHED
TCP   172.17.150.112:1332    198.133.219.25:80      TIME_WAIT
TCP   172.17.150.112:1333    198.133.219.25:80      ESTABLISHED
TCP   172.17.150.112:1334    198.133.219.25:80      ESTABLISHED
TCP   172.17.150.112:1335    64.154.80.254:80       ESTABLISHED
TCP   172.17.150.112:1336    66.102.7.99:80         ESTABLISHED
```

Red annotations highlight specific connections:

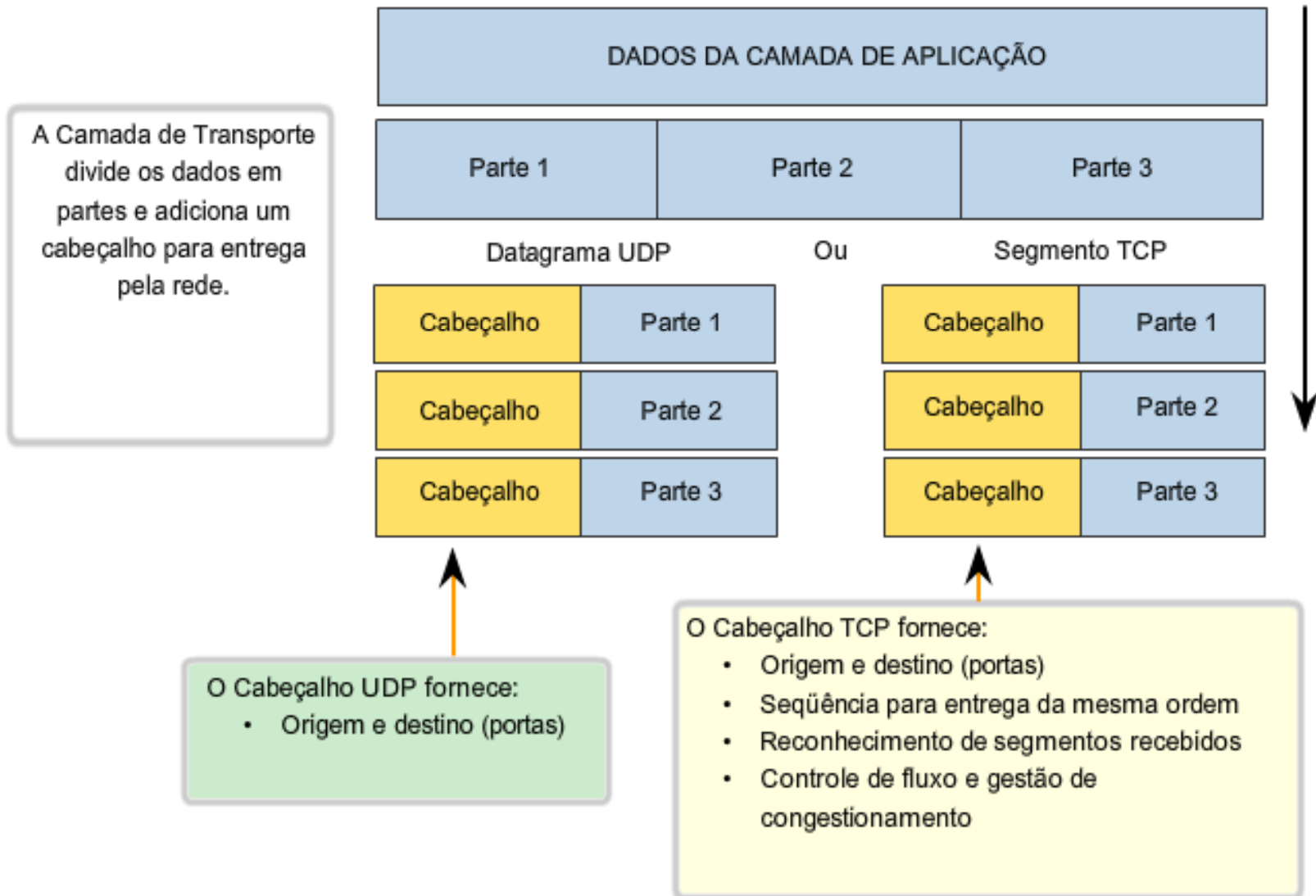
- A red arrow labeled "TCP or UDP" points to the "Proto" column.
- Red arrows labeled "Source IP" and "Destination IP" point to the "Local Address" and "Foreign Address" columns, respectively.
- Red arrows labeled "Source Port" and "Destination Port" point to the port numbers in the "Local Address" and "Foreign Address" columns, respectively.
- A red arrow labeled "Connection State" points to the "State" column.
- A blue box highlights the connection to 198.133.219.25:80 (TIME_WAIT state).
- A red box highlights the connection to 64.154.80.254:80 (ESTABLISHED state).

Below the command prompt, the following text is displayed:

www.google.com www.cisco.com netstat -n

Segmentação

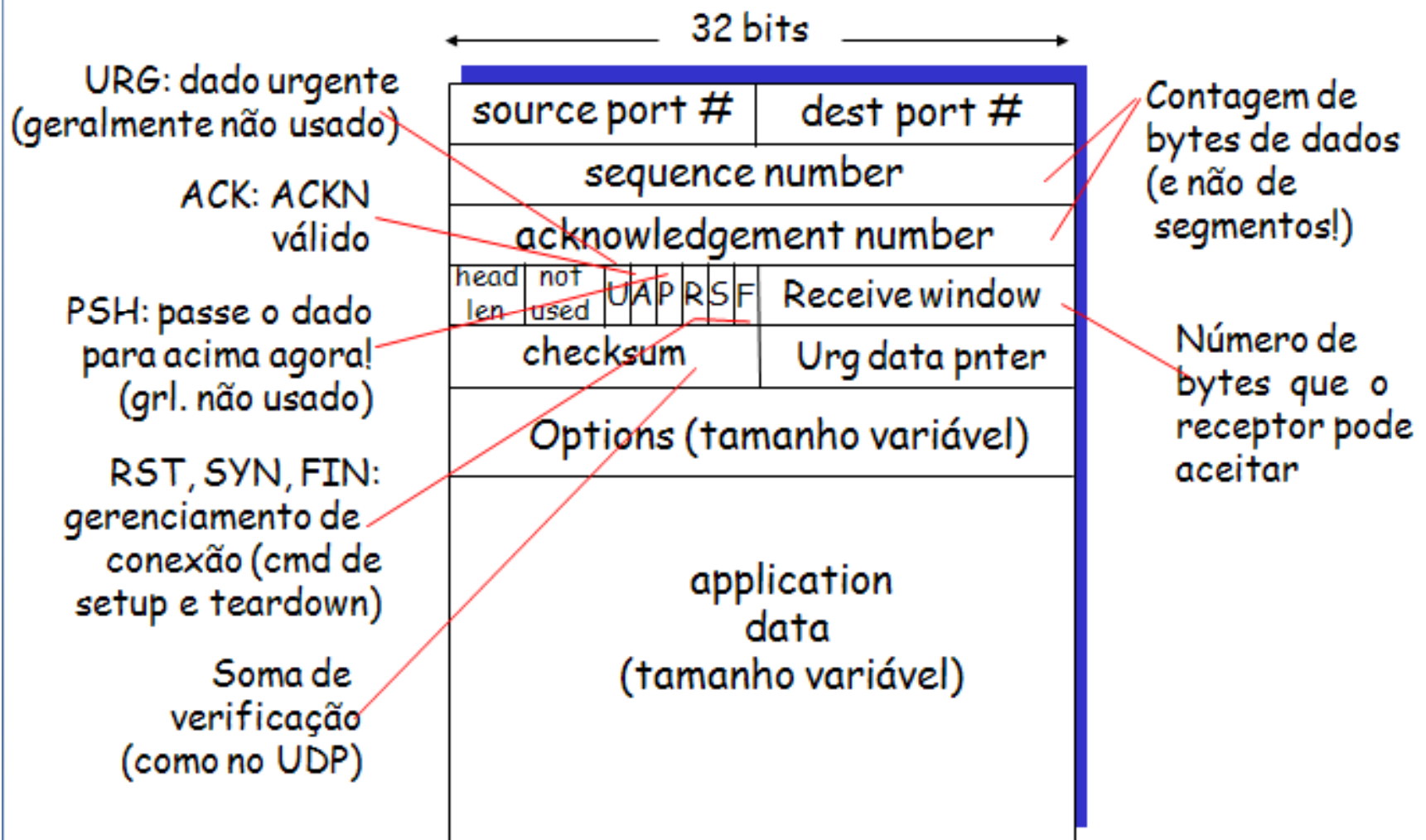
Funções da Camada de Transporte



TCP - Transmission Control Protocol

RFCs: 793, 1122, 1323, 2018, 2581

Estrutura do Segmento



Número de seq. ACKs

Número de sequencia

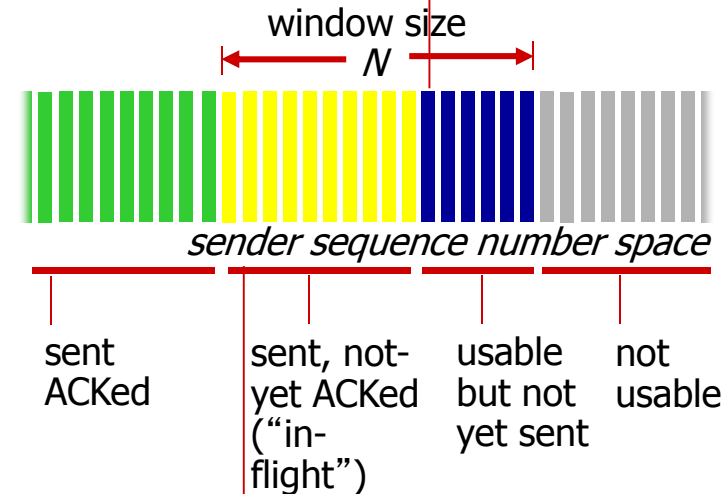
- Número do fluxo de bytes relacionado ao primeiro bytes de dados do segmento

Reconhecimentos (acknowledgements):

- Num. seq. do próximo byte esperado do outro lado
- ACK cumulativo

outgoing segment from sender

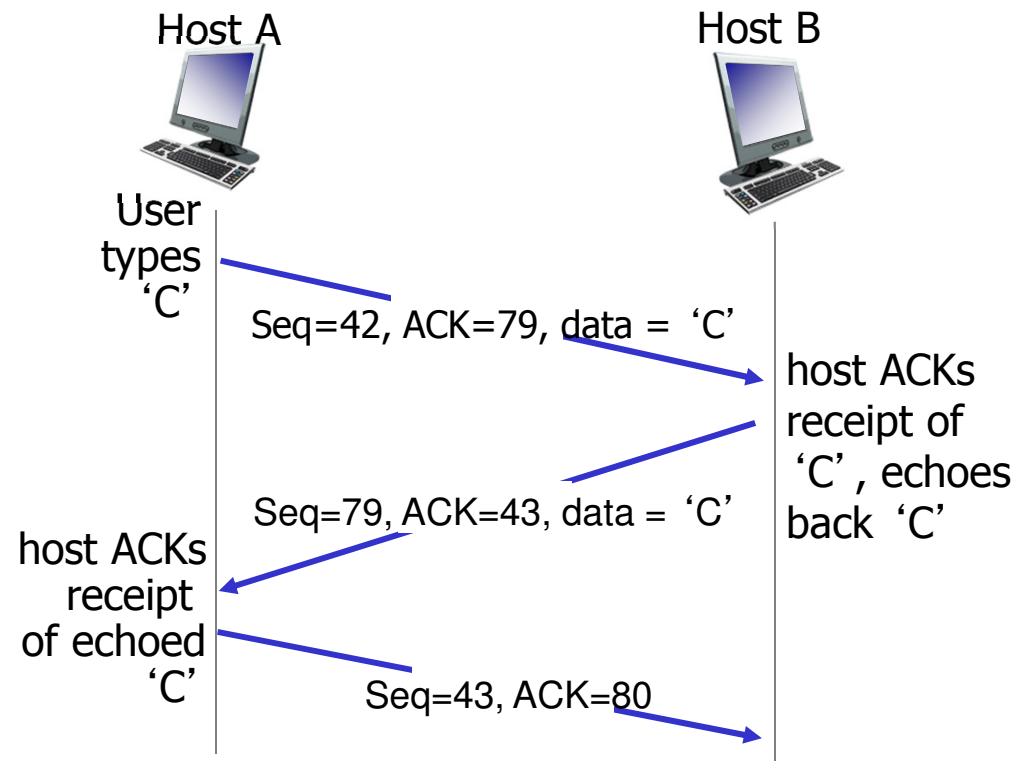
source port #	dest port #
sequence number	
acknowledgement number	
	rwnd
checksum	urg pointer



incoming segment to sender

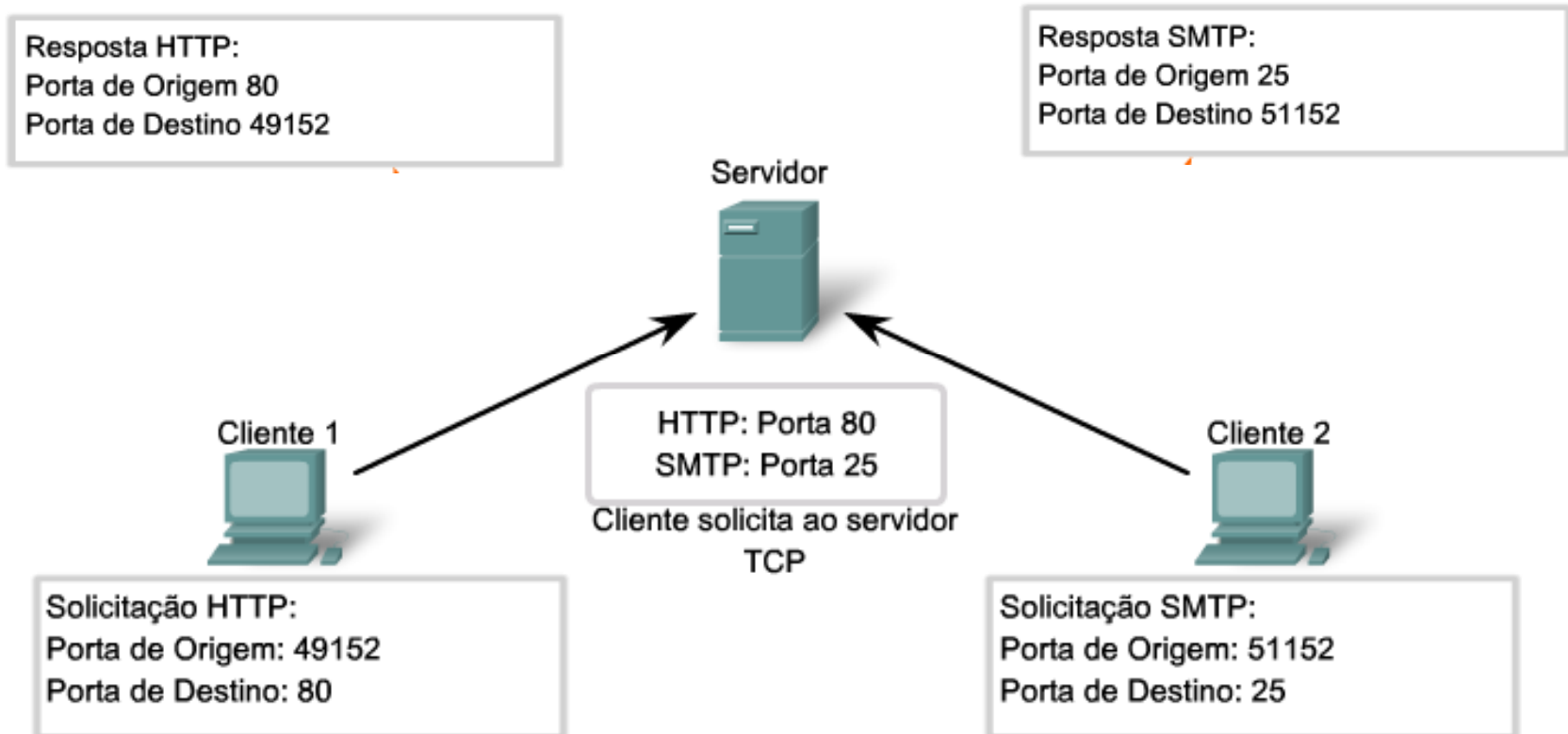
source port #	dest port #
sequence number	
acknowledgement number	
	A
checksum	urg pointer

Número de seq. ACKs



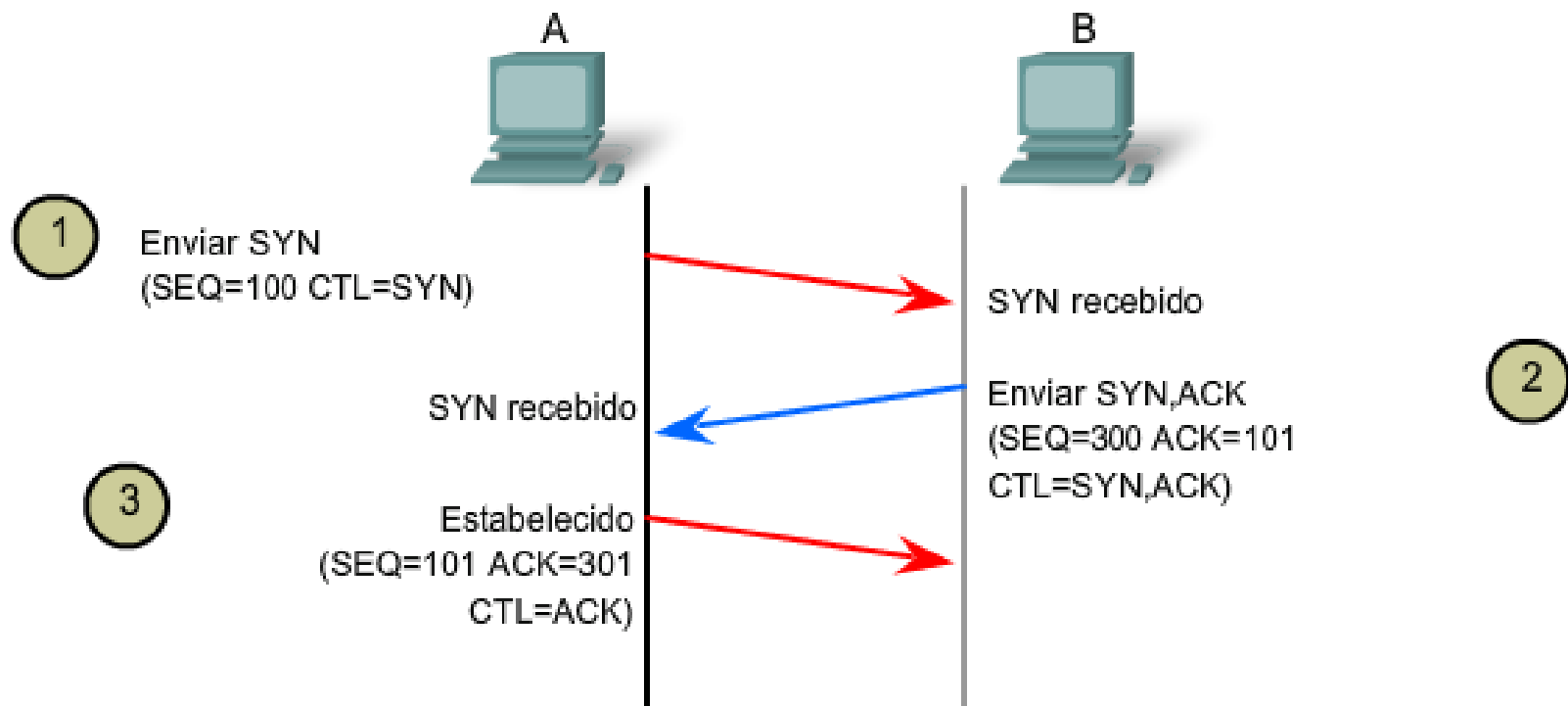
simple telnet scenario

TCP - Solicitações e Respostas



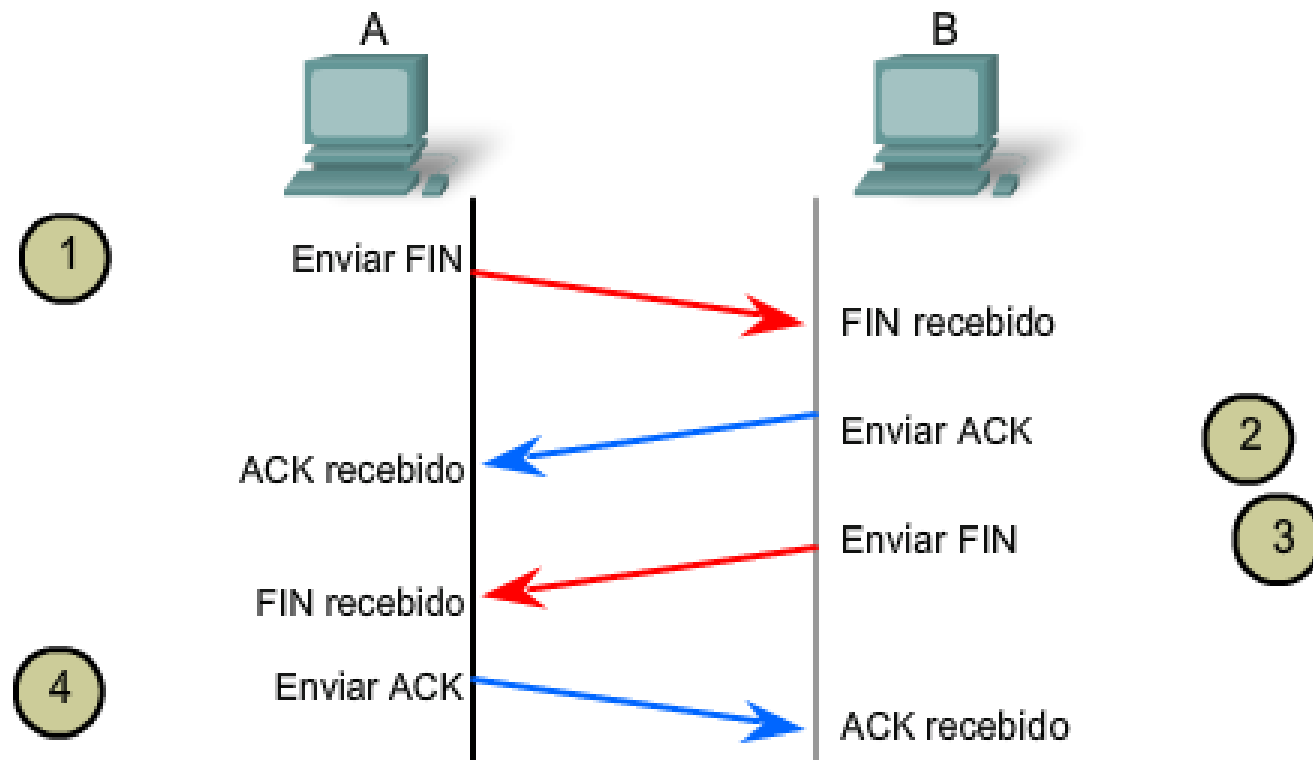
TCP - Gerenciamento de Conexões

- Estabelecimento de Conexões - 3 Way Handshake

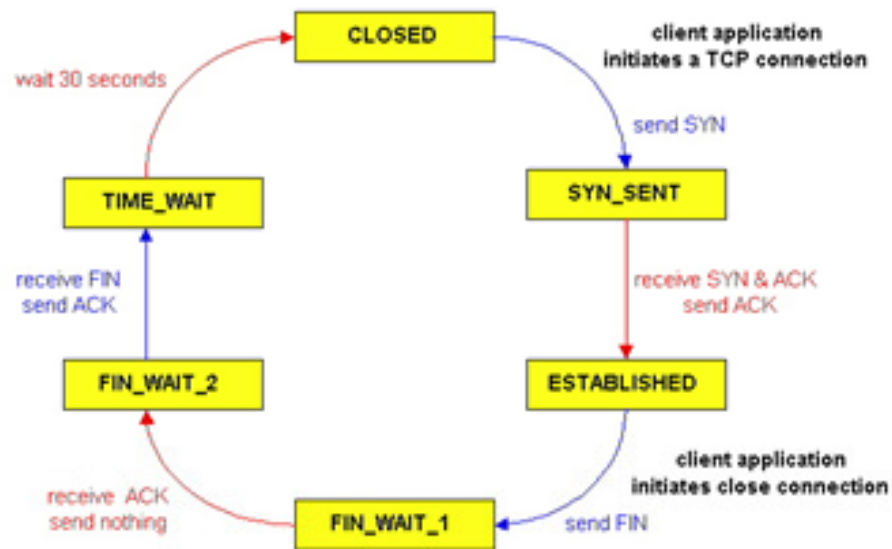


TCP - Gerenciamento de Conexões

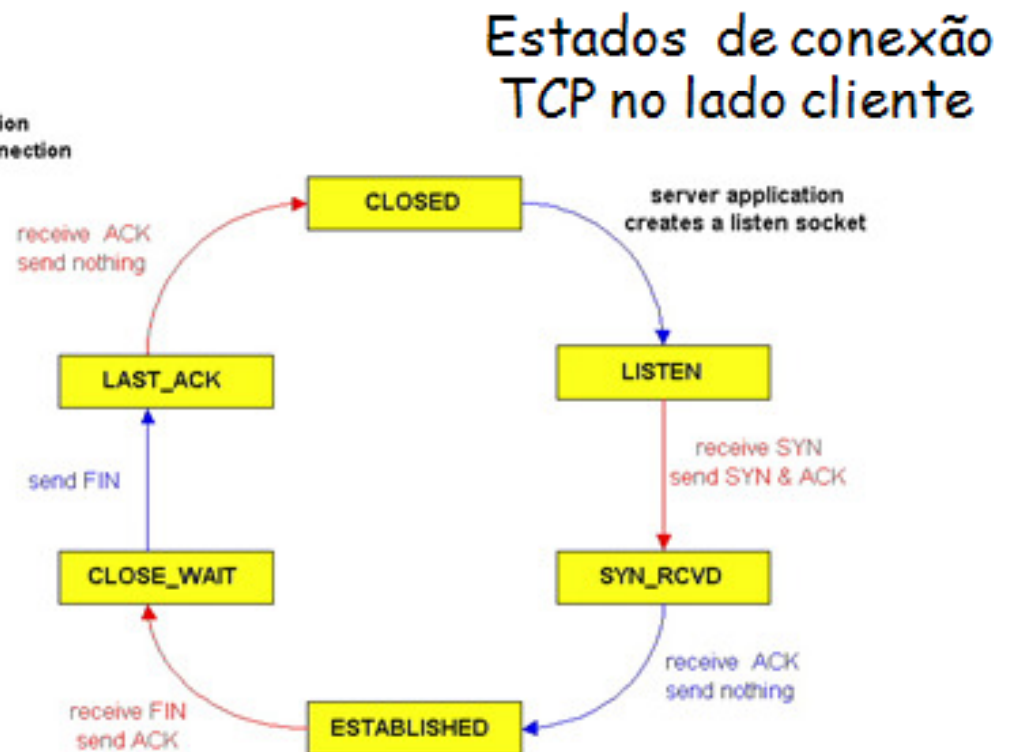
- Finalização de Conexões



TCP - Gerenciamento de Conexões

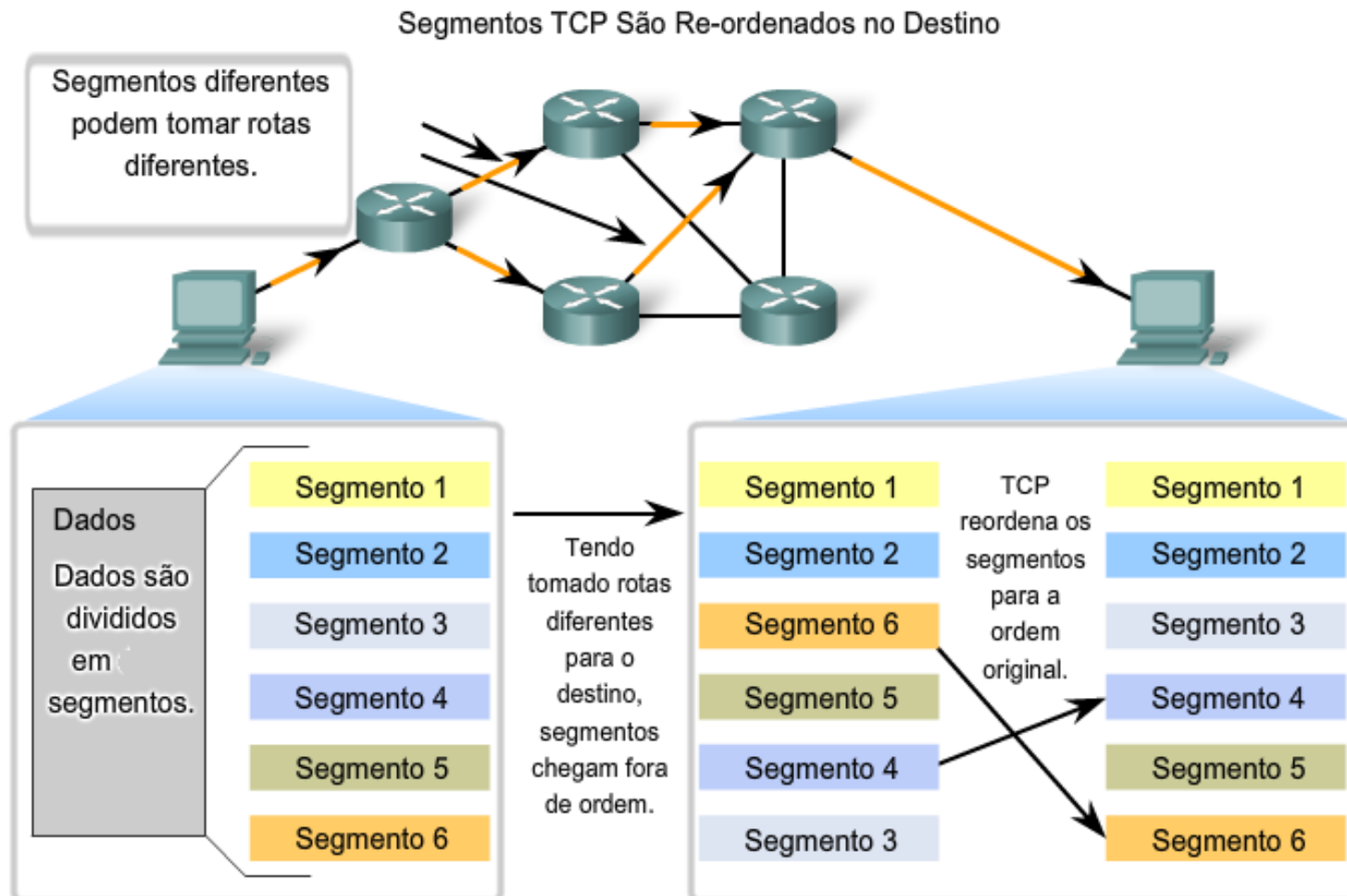


Estados de conexão
TCP no lado cliente



Estados de conexão
TCP no lado cliente

TCP - Entrega Confiável de Dados



Tempo de ida e volta e timeout

Q: como configurar o valor do timeout do TCP?

- ❖ Maior que RTT
 - mas RTT varia
- ❖ *Muito pequeno:* timeout prematuro, retransmissões desnecessárias
- ❖ *Mais longo:* reação lenta a perda do segmento

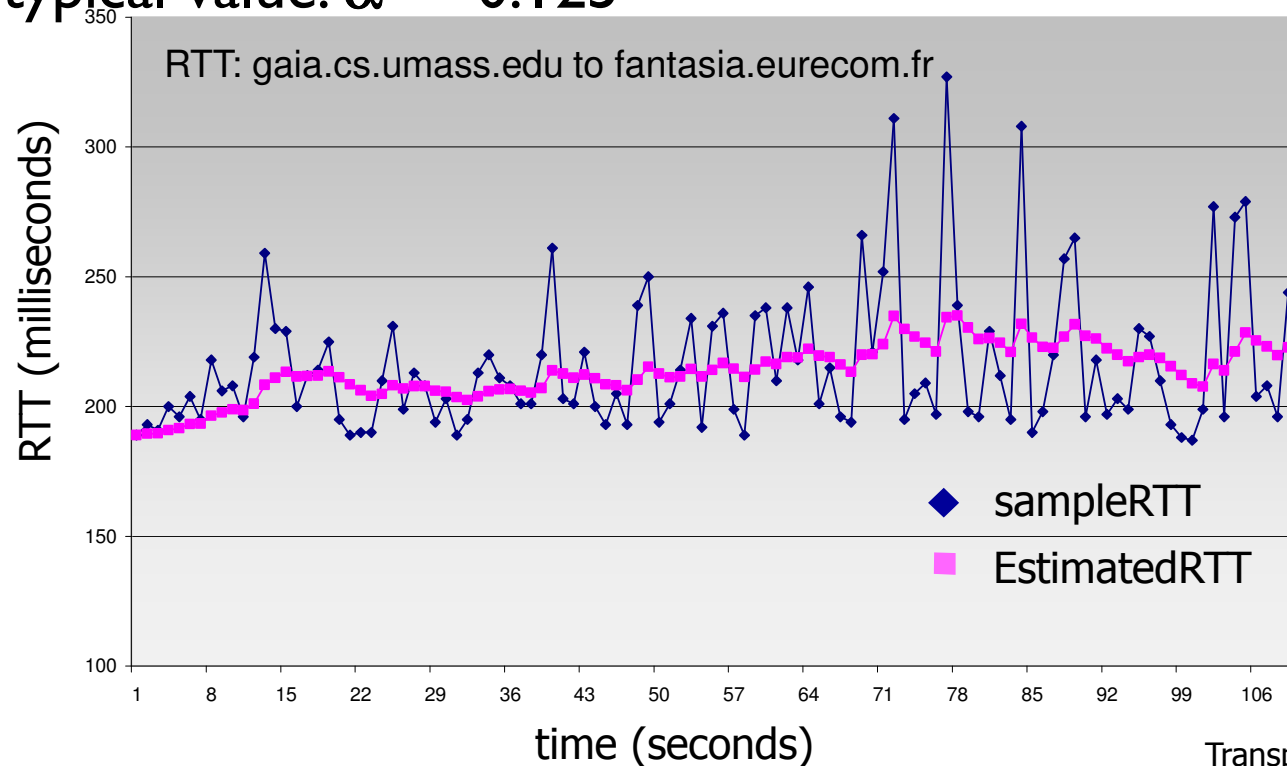
Q: como estimar RTT?

- ❖ **SampleRTT:** tempo medido desde a transmissão do segmento até um ACK recebido
 - ignore retransmissões
- ❖ **SampleRTT** pode variar, logo é necessário que o RTT estimado seja suavizado
 - Média de várias medições recentes, não apenas o **SampleRTT** atual

Tempo de ida e volta e timeout

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

- ❖ Média móvel ponderada exponencial
- ❖ Influencia das amostras passadas diminui exponencialmente rápido
- ❖ typical value: $\alpha = 0.125$



Tempo de ida e volta e timeout

- ❖ **timeout interval:** `EstimatedRTT` mais “margem segura”
 - Variação grande em `EstimatedRTT` → margem segura maior
- ❖ Estimao desvio de `SampleRTT` a partir de `EstimatedRTT`:

$$\text{DevRTT} = (1-\beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$$

(typically, $\beta = 0.25$)

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$



↑
RTT estimado

↑
“margem de segurança”

Transferencia confiável do TCP

- ❖ TCP cria um serviço confiável acima do IP
 - Fluxo de segmentos
 - Acks cumulativos
 - Único timer de retransmissão
- ❖ Retransmissões disparadas por:
 - Eventos de timeout
 - Acks duplicados

Inicialmente considere um transmissão simplificado

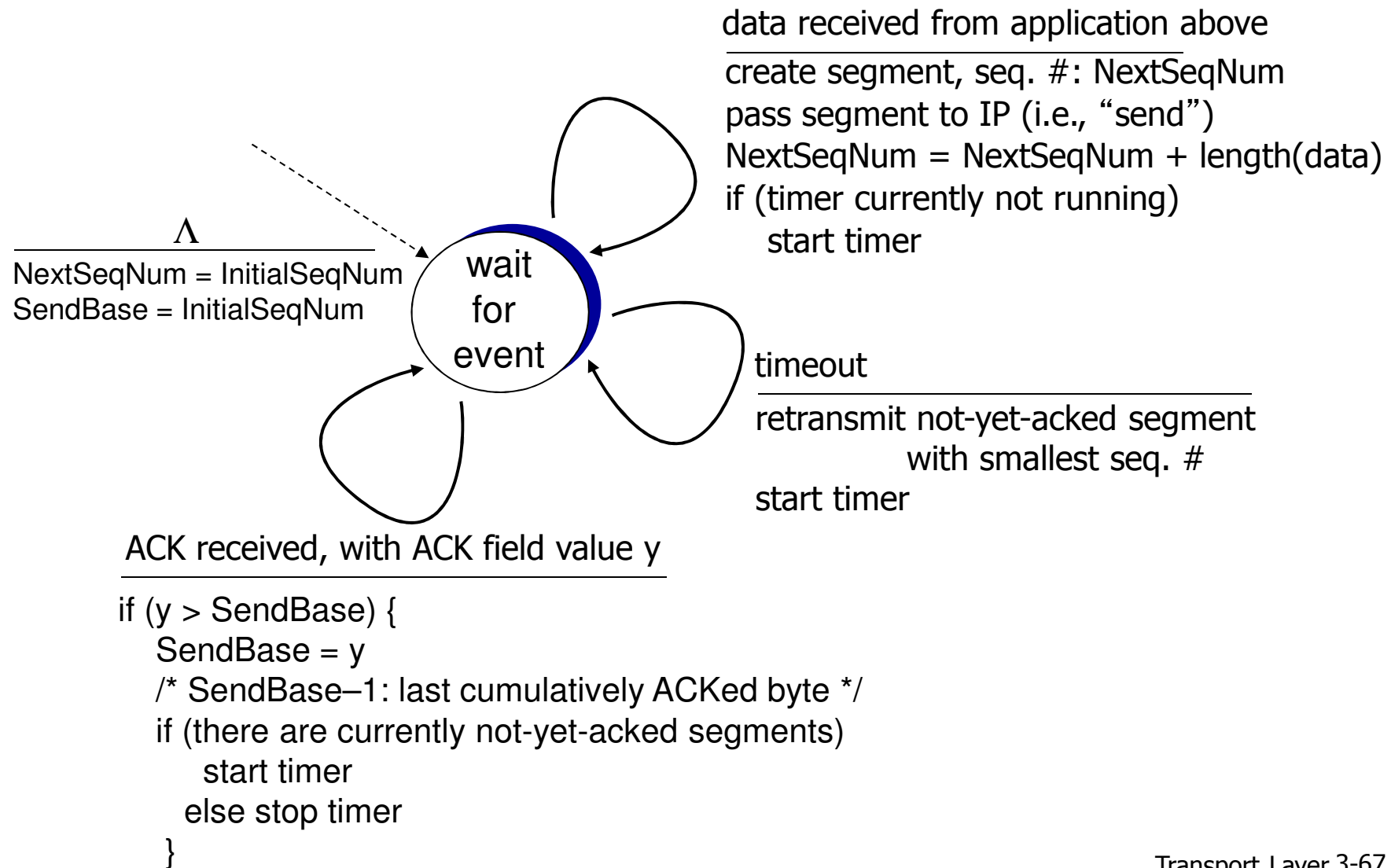
- Ignore acks duplicados
- ignore controle de fluxo, controle de congestionamento

Eventos no Transmissor TCP:

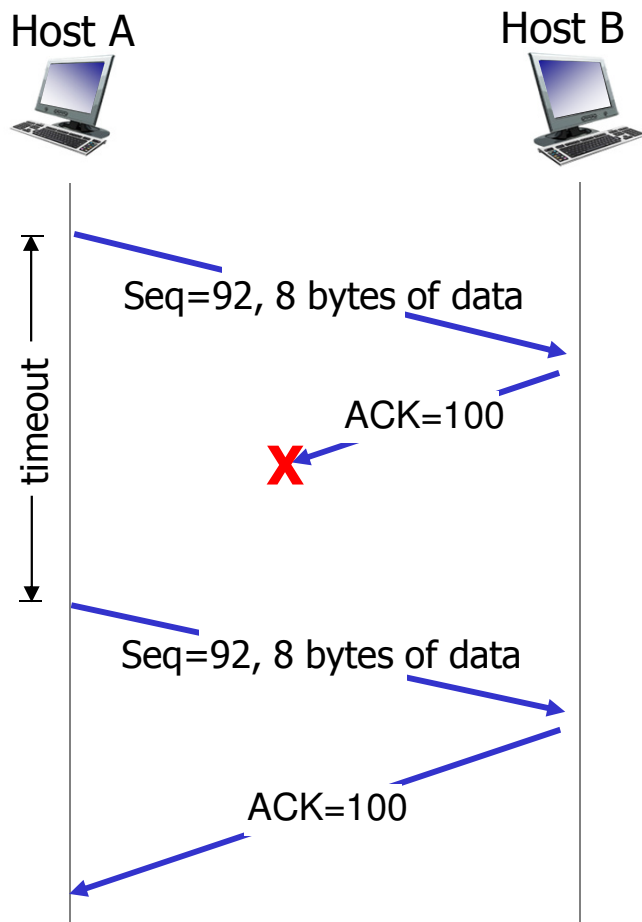
Dados recebidos pela app: timeout:

- ❖ Cria um segmento com num. seq
 - ❖ Num seq. é o número do fluxo de bytes do primeiro byte do segmento
 - ❖ Inicia o timer se ele não estiver já rodando
 - Intervalo de expiração: `TimeoutInterval`
 - ❖ Retransmite o segmento que causou o timeout
 - ❖ Reinicia o timer
- ## *ack recebido:*
- ❖ Se o ack reconhece segmentos recebidos previamente sem ACK
 - Atualize o que é pra ser reconhecido
 - Inicie o timer existir segmentos sem reconhecimento

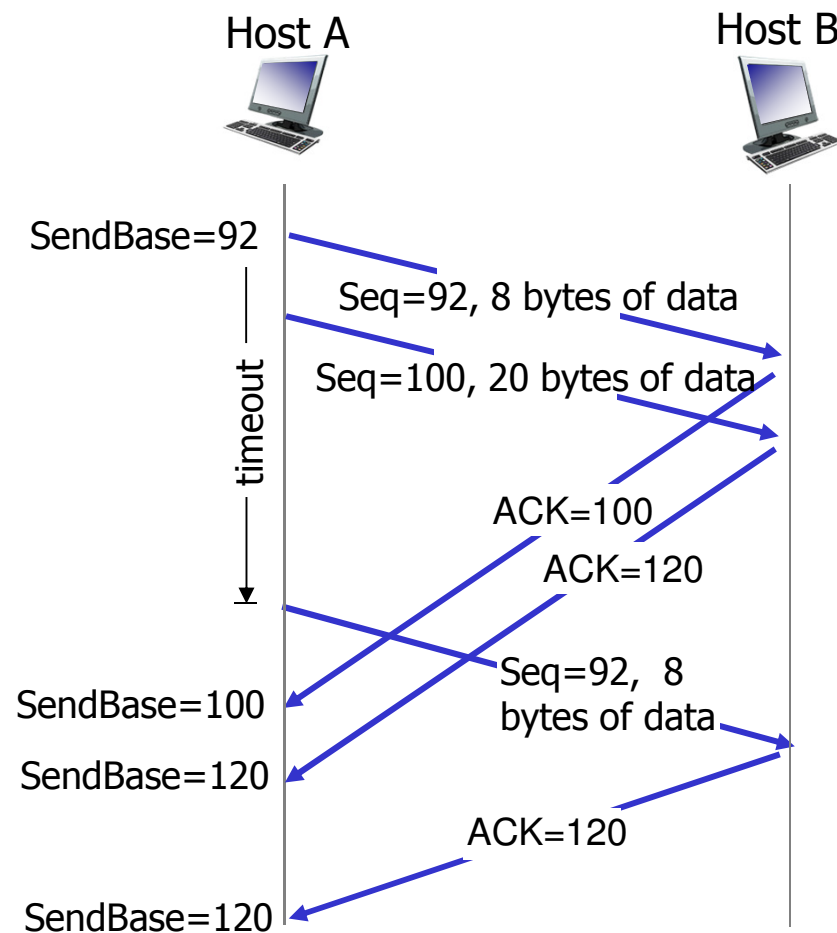
Transmissor (simplificado)



TCP: cenários de retransmissão

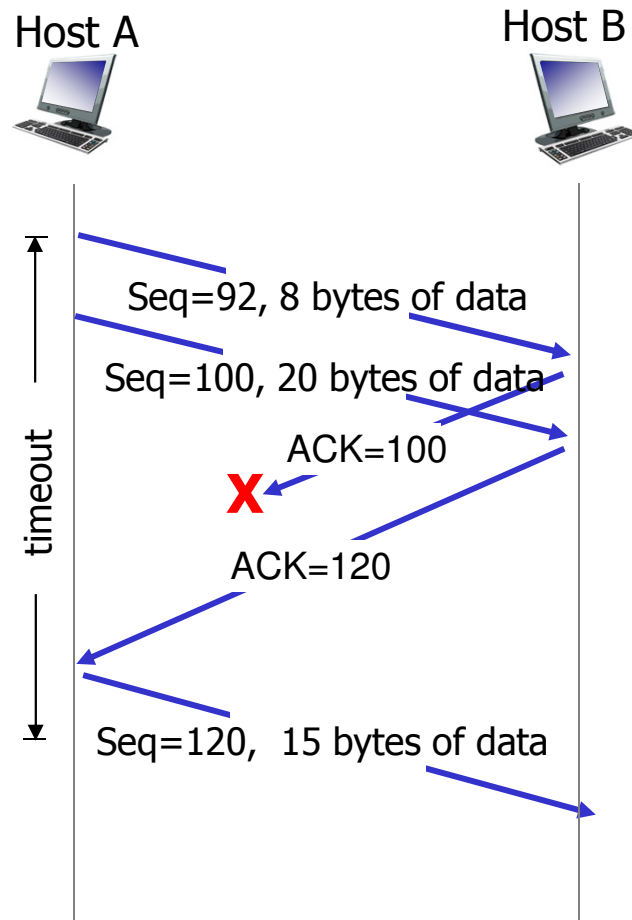


Cenário do ack perdido



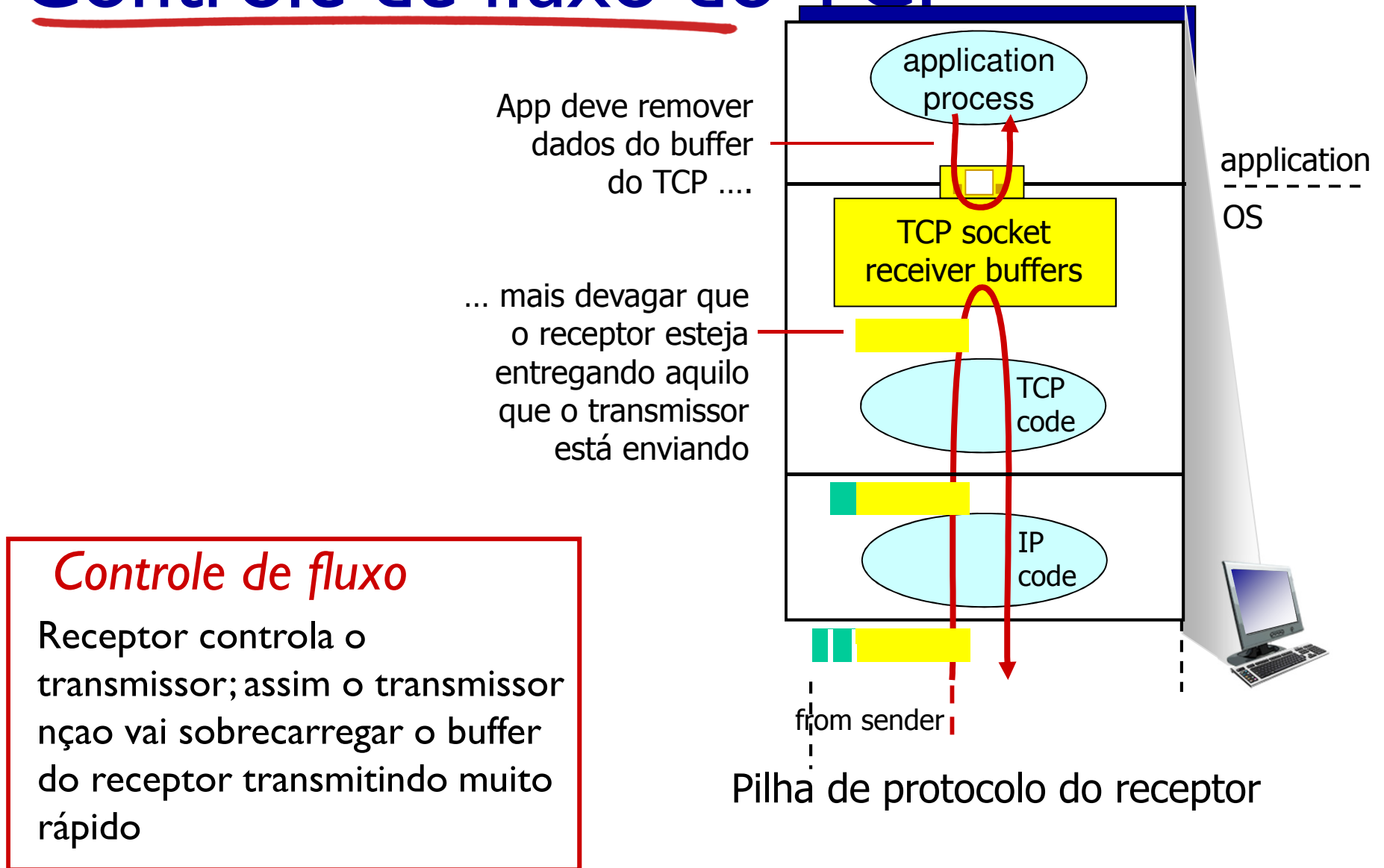
timeout prematuro

TCP: cenários de retransmissão



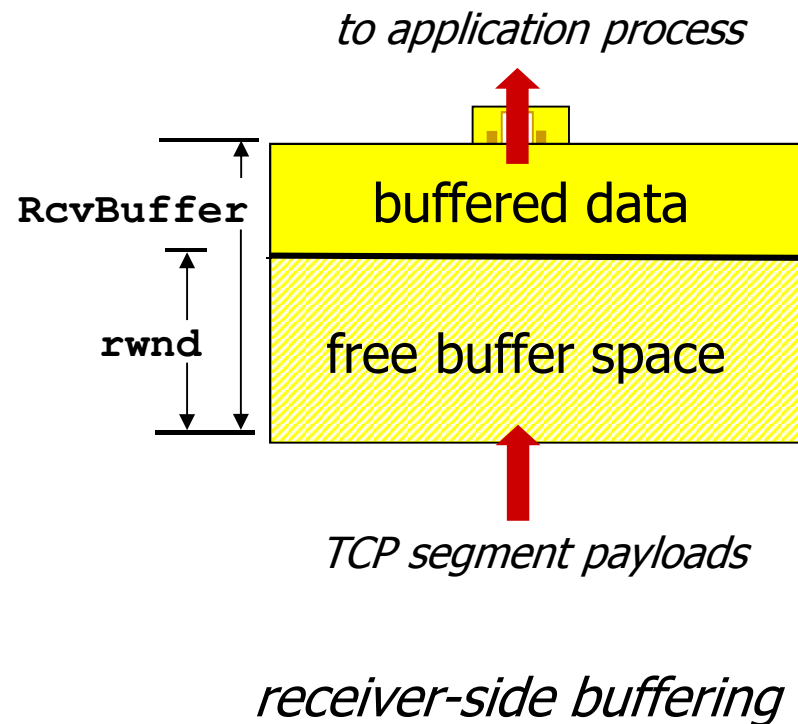
ACK cumulativo

Controle de fluxo do TCP



Controle de fluxo do TCP

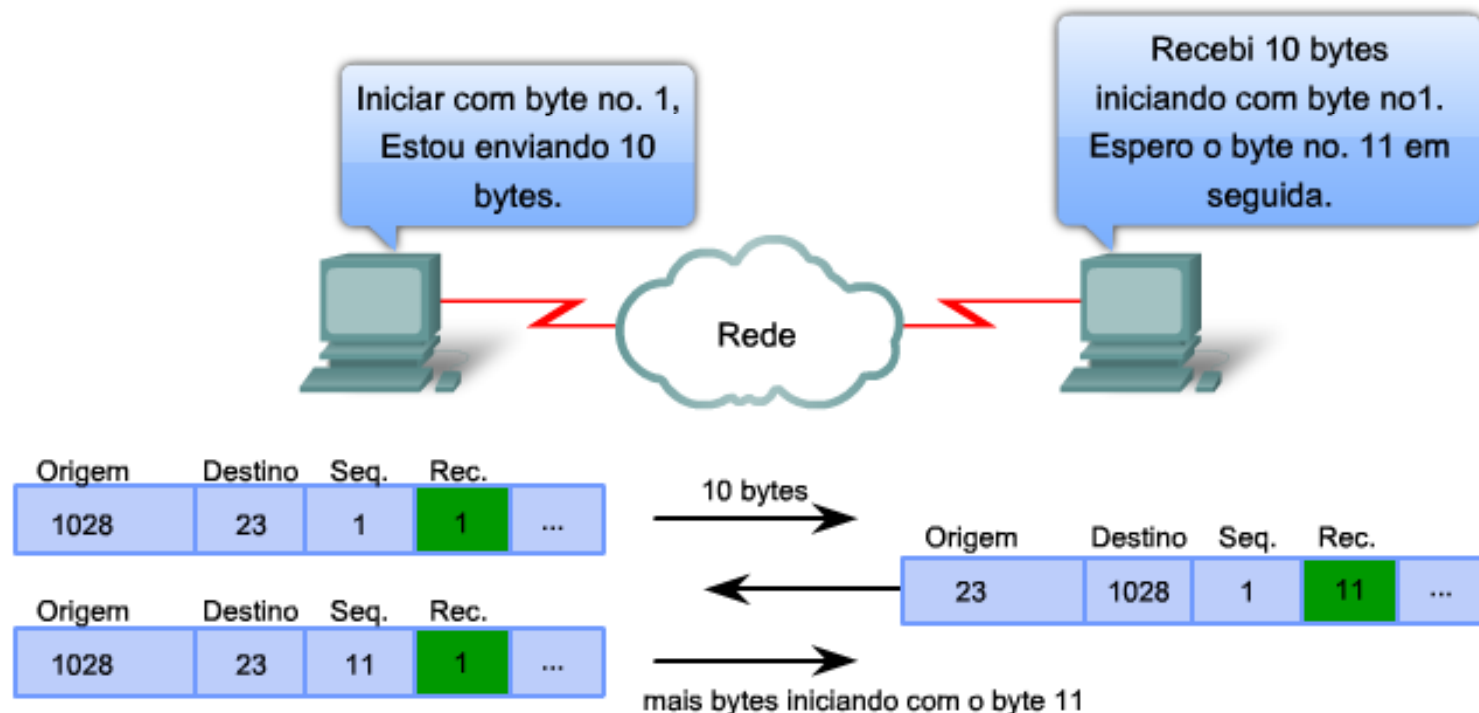
- ❖ receptor “avisa” sobre o espaço vazio no buffer pelo valor de **rwnd** no cabeçaho de segmentos enviados para o transmissor
 - **RcvBuffer** tamanho configurado nas opções de socket (valor default é 4096 bytes)
 - Muitos sistemas operacionais auto-ajustam **RcvBuffer**
- ❖ O transmissor limita a quantidade de dados sem reconhecimento para o valor da **rwnd** do receptor
- ❖ Garante que o buffer do receptor não vai estourar



TCP - Entrega Confiável de Dados

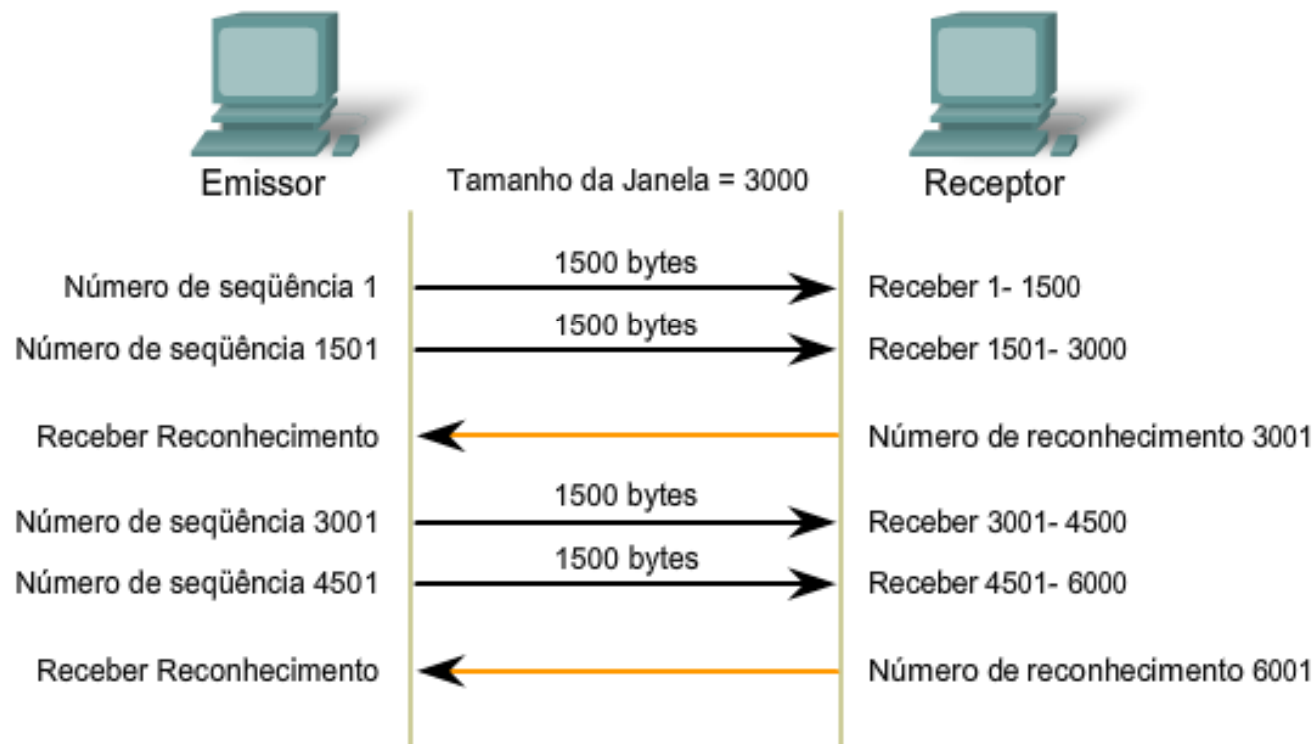
Reconhecimento de segmentos TCP

Porta de Origem	Porta de Destino	Número de Sequência	Números de reconhecimento	...
-----------------	------------------	---------------------	---------------------------	-----



TCP - Controle de Fluxo

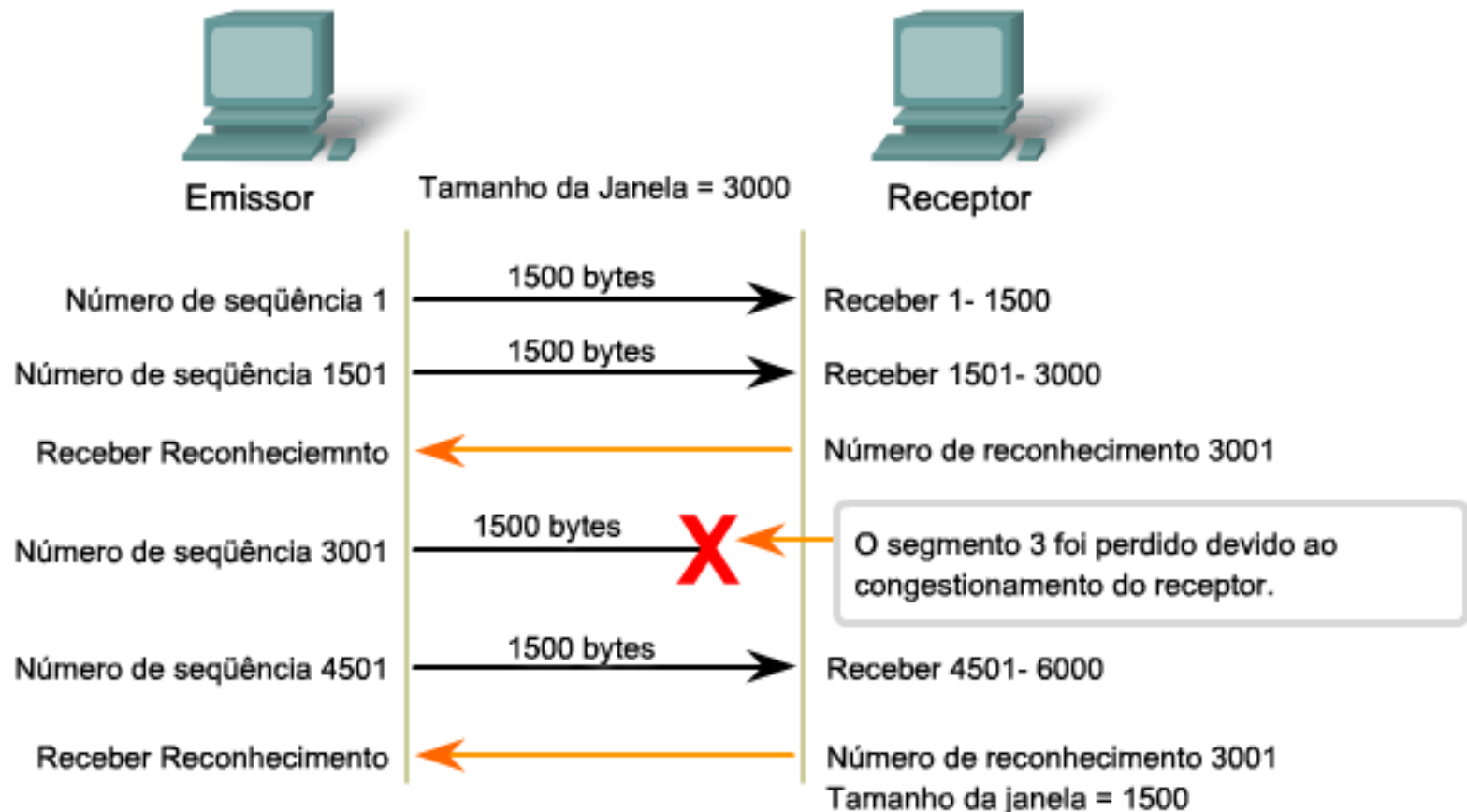
Reconhecimento de Segmento TCP e Tamanho da Janela



O **tamanho da janela** determina o número de bytes enviados antes de um reconhecimento.

O número de **reconhecimento** é o número do próximo byte esperado.

TCP - Controle de Congestionamento



Se os segmentos forem perdidos devido ao congestionamento, o Receptor reconhecerá o último segmento recebido em sequência e responderá com um tamanho de janela reduzido.

Prática

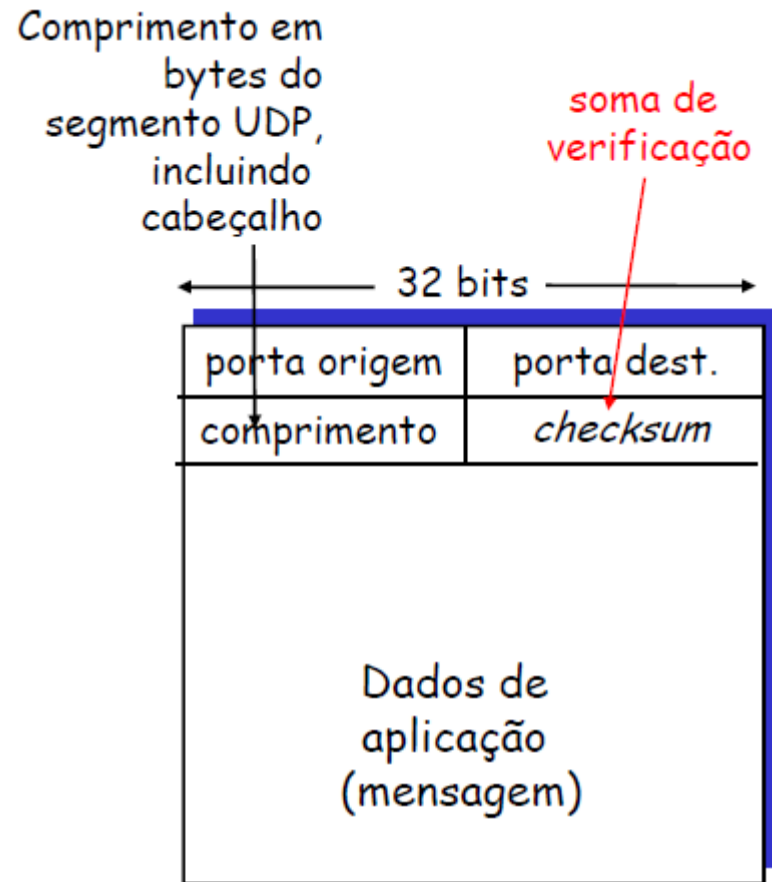
- Segmentação de mensagens HTTP
 - Prática Wireshark e HTTP
 - Questões 12 a 15

UDP- User Datagram Protocol [RFC 768]

- Serviço do melhor esforço, datagramas podem ser:
 - Perdidos
 - Entregues na ordem errada
- Sem conexão
 - Não há "setup" entre remetente e receptor
 - Tratamento independente para cada datagrama
- Elimina estabelecimento de conexão (o que pode causar retardo)
- Simples: não se mantém o estado da conexão nem no remetente, nem no receptor
- Não há controle de congestionamento: UDP pode transmitir tão rápido quanto desejado (e possível)

UDP- User Datagram Protocol [RFC 768]

- Muito utilizado para aplicações de dados contínuos (voz e vídeo)
 - Tolerantes a perda
 - Sensíveis à taxa de Transmissão
- Outros usos do UDP
 - DNS
 - SNMP
- Transferência confiável com UDP: acrescentar confiabilidade na camada de aplicação
 - Recuperação de erro específica a aplicação



Formato do segmento UDP

Checksum

Objetivo: detectar "erros" (ex.: bits trocados) no segmento transmitido

Transmissor:

- Trata o conteúdo do datagrama como sequência de inteiro de 16 bits
- Campo checksum zerado
- Checksum: soma (adição usando complemento de 1) do conteúdo do segmento
- Transmissor coloca complemento de valor da soma no campo checksum da UDP

Receptor:

- Calcula checksum de todo datagrama incluindo o próprio campo checksum
- Verifica se o checksum calculado possui zero
 - SIM: erro detectado
 - NÃO: nenhum erro. Mas ainda pode ter erros? Veja depois...

Exemplo de Checksum da Internet

- Note que:
 - Ao adicionar números , o transbordo (vai um) do bit mais significativo deve ser adicionado ao resultado
 - Exemplo: adição de dois inteiros de 16 bits

[illegible]