



## **Capítulo 23**

# **Entrega Processo-a- Processo: UDP, TCP e SCTP**

# 23-1 ENTREGA PROCESSO-A-PROCESSO

*A Camada de Transporte é responsável pela entrega processo-a-processo, que é a entrega de um pacote, parte de uma mensagem, de um processo até outro. Dois processos podem se comunicar através do modelo cliente/servidor, como será visto na Camada de Aplicação.*

## **Topicos discutidos nessa sessão:**

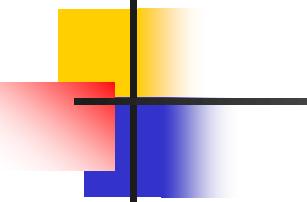
Paradigma Cliente/Servidor

Multiplexação e Demultiplexação

Serviço orientado a conexão vs serviço sem conexão

Serviço confiável vs serviço não confiável

Os três protocolos de transporte



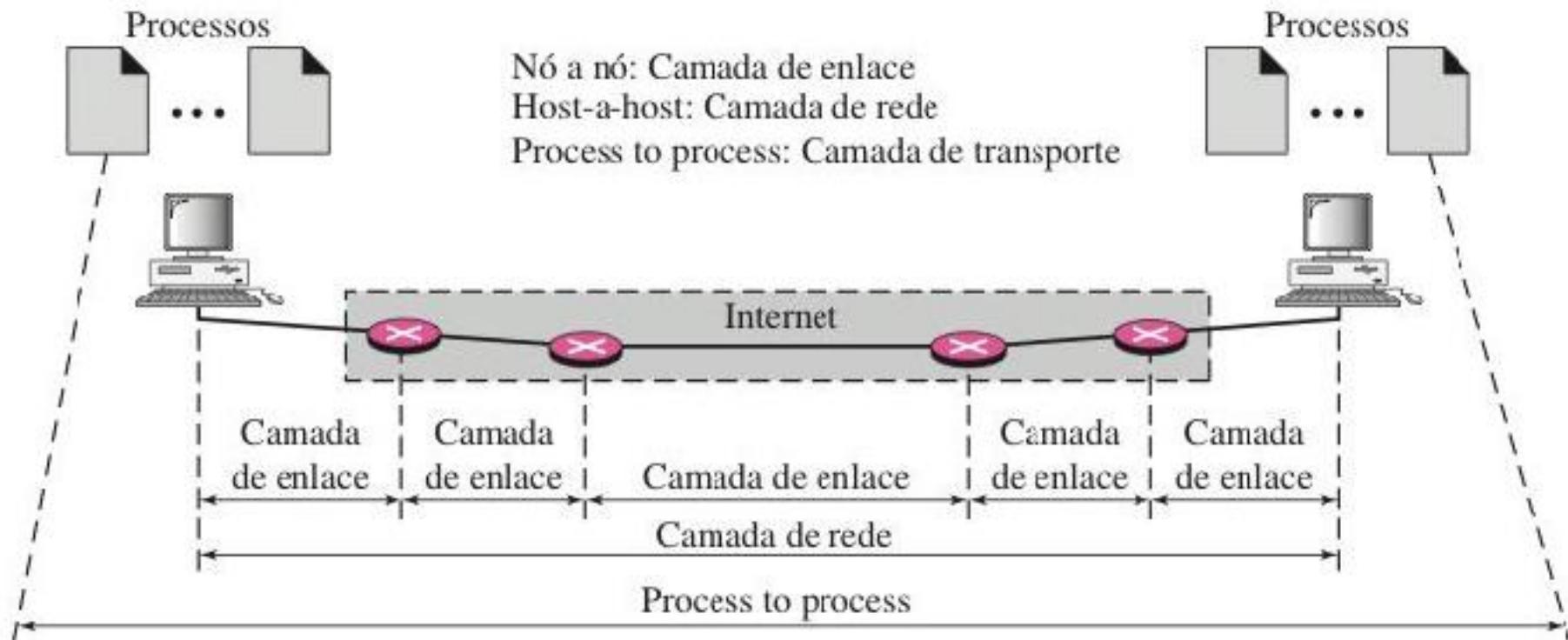
## *Nota*

---

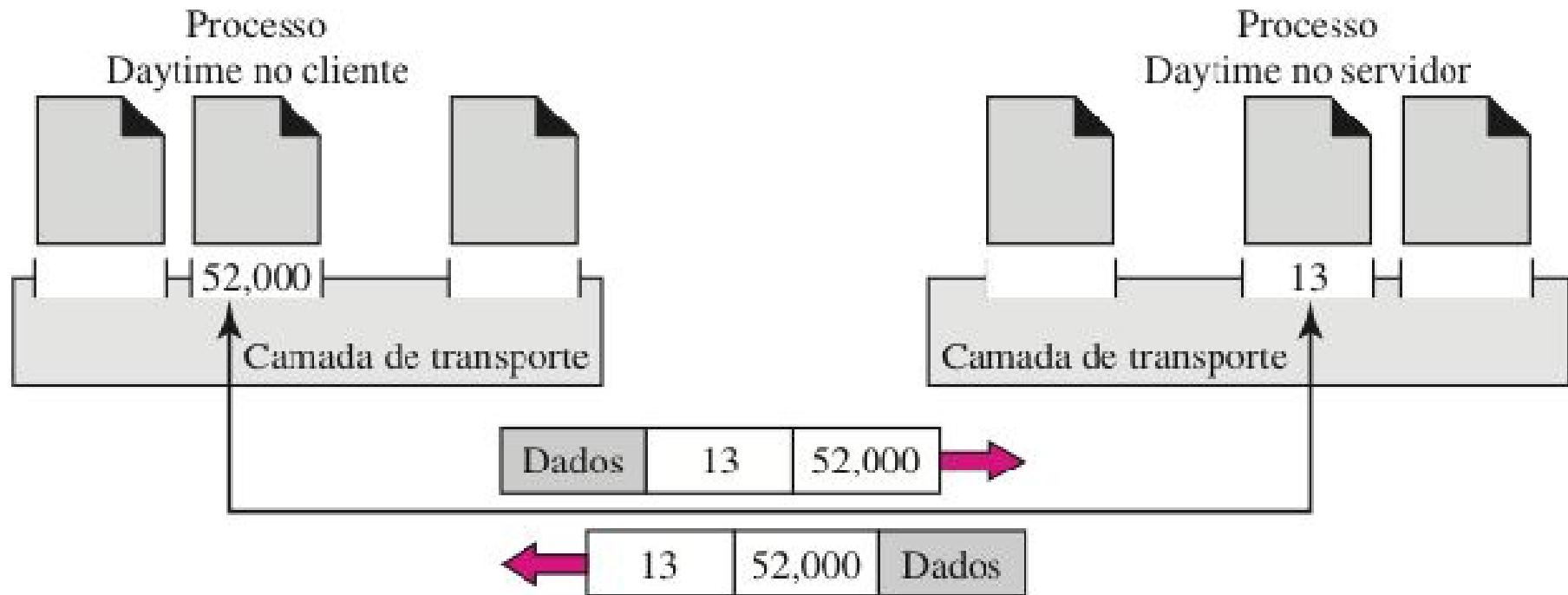
**A camada de transporte é responsável pela entrega processo-a-processo.**

---

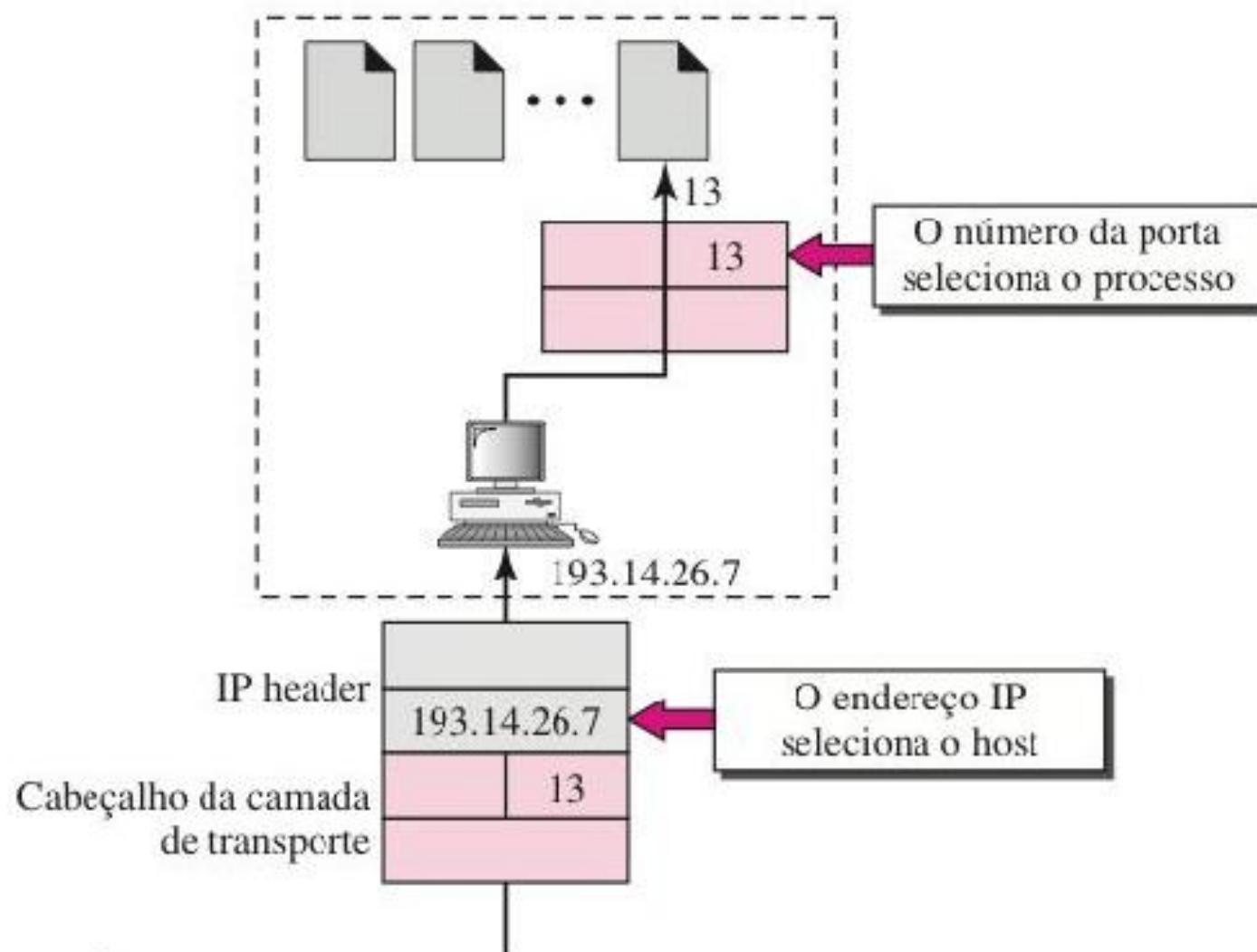
## Figura 23.1 *Tipos de entrega de dados*



**Figura 23.2 Números de Portas**



**Figura 23.3 Endereços IP vs Números de portas**

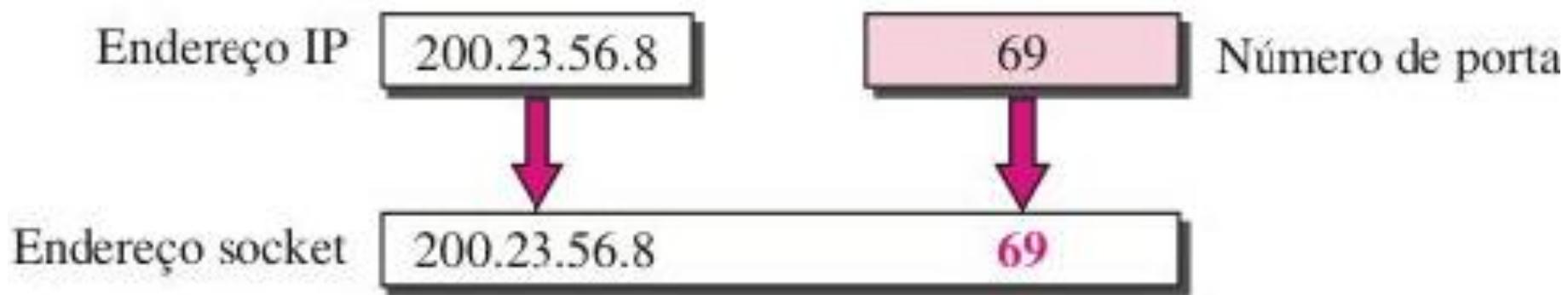


**Figura 23.4** Faixas de portas da IANA

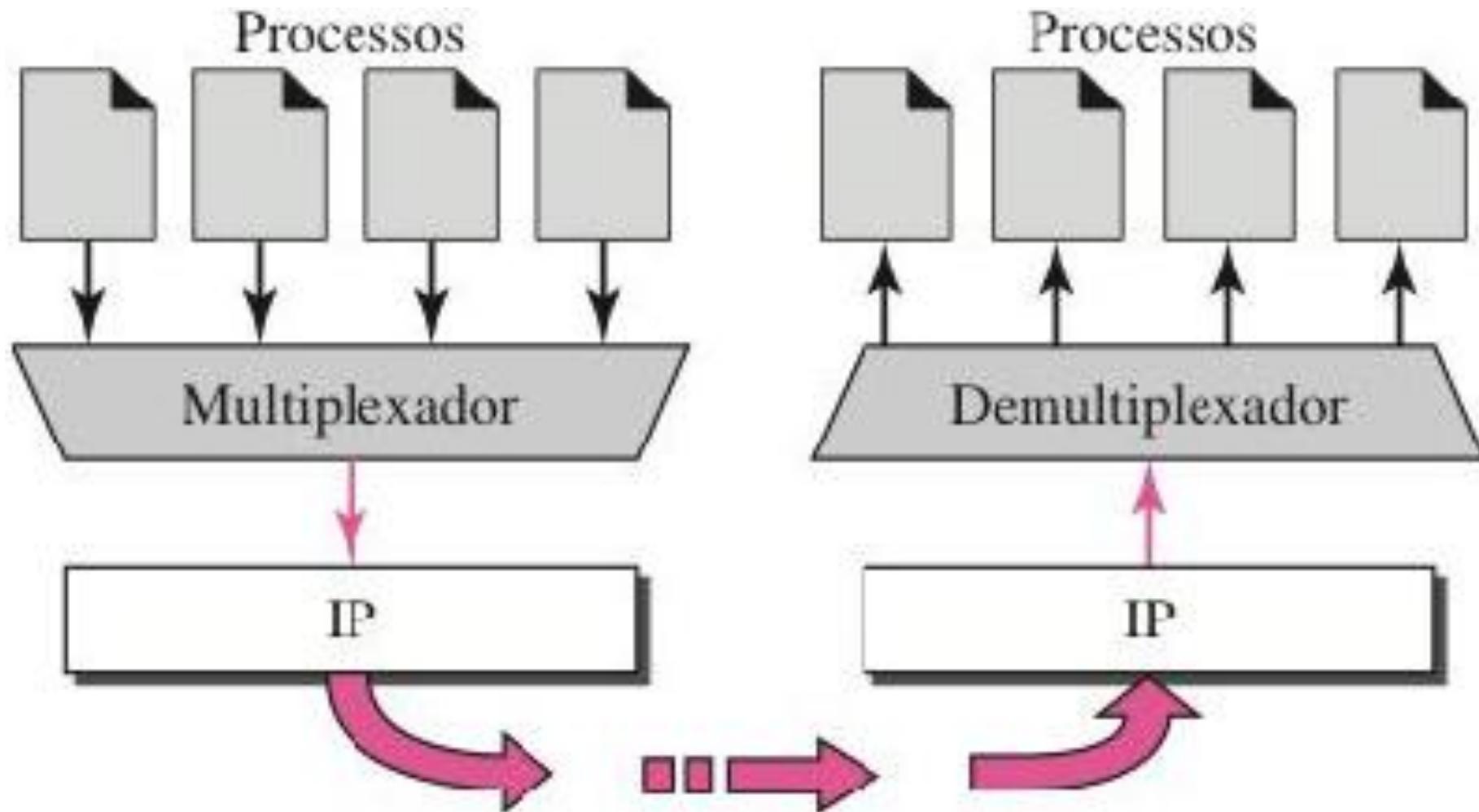


**Figura 23.5 Endereço de socket**

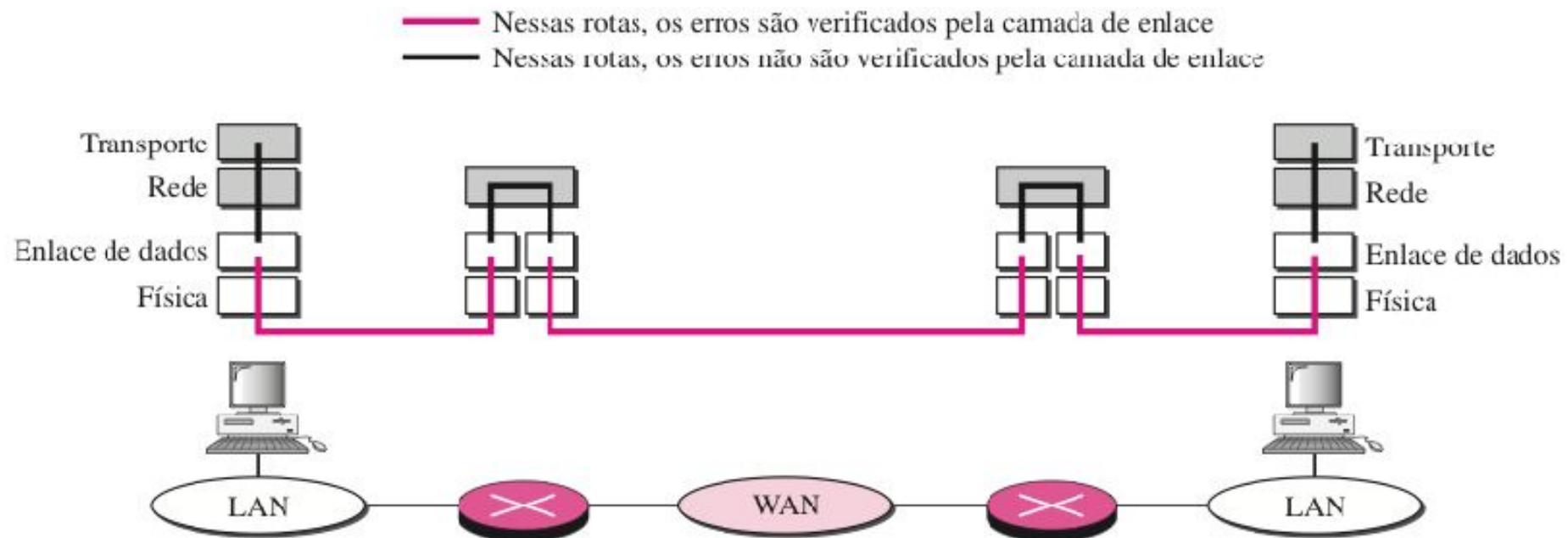
---



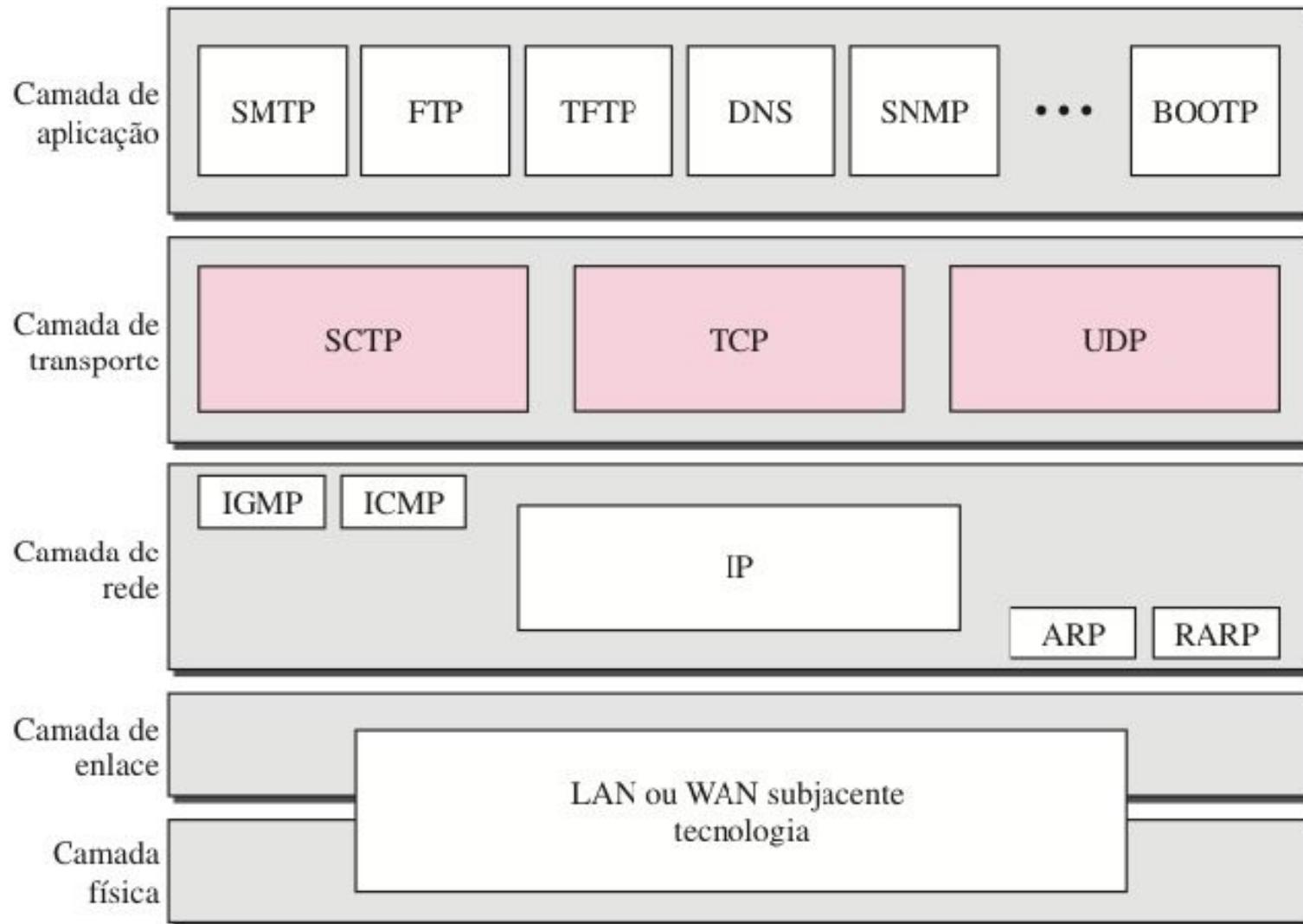
**Figura 23.6** Multiplexação e demultiplexação



**Figura 23.7** Controle de Erros



**Figura 23.8 Posição do UDP, TCP e SCTP no modelo TCP/IP**



## 23-2 PROTOCOLO DE DATAGRAMA DE USUÁRIO (UDP)

*O Protocolo de Datagrama de Usuário (UDP) é um protocolo de transporte sem conexão e não confiável. Ele não acrescenta nenhum controle adicional aos serviços de entrega IP, exceto pelo fato de implementar a comunicação entre processos.*

### Topicos discutidos nessa sessão:

Portas bem conhecidas no UDP

Datagrama de usuário

Checksum

Operação do UDP

Uso do UDP

**Tabela 23.1** *Algumas portas bem conhecidas no UDP*

Porta	Protocolo	Descrição
7	Echo	Ecoa um datagrama recebido de volta para o emissor
9	Discard	Descarta qualquer datagrama recebido
11	Users	Usuários ativos

Porta	Protocolo	Descrição
13	Daytime	Retorna data e hora
17	Quote	Retorna um comentário do dia
19	Chargen	Retorna uma string de caracteres
53	Nameserver	Domain Name Services
67	BOOTPs	Servidor bootstrap
68	BOOTPc	Cliente bootstrap
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

## *Exemplo 23.1*

*No UNIX, as portas bem conhecidas são armazenadas em um arquivo chamado /etc/services. Cada linha nesse arquivo armazena o nome do servidor e o número da porta bem conhecida. O utilitário grep pode ser usado para extrair a linha correspondente ao serviço desejado. O comando a seguir foi usado para mostrar o número de porta usado pelo FTP. Note que o FTP pode usar a porta 21 tanto com UDP como com o TCP.*

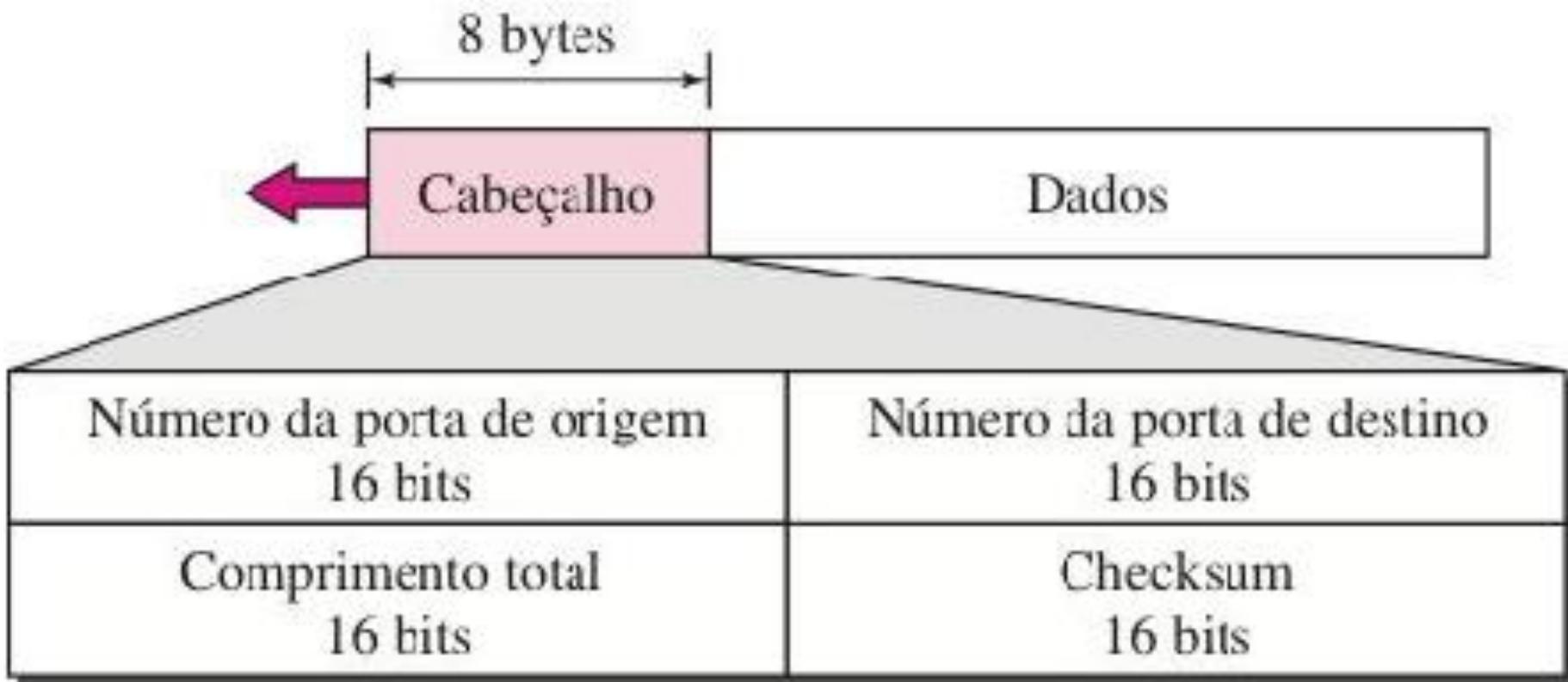
```
$ grep ftp /etc/services
ftp           21/tcp
ftp           21/udp
```

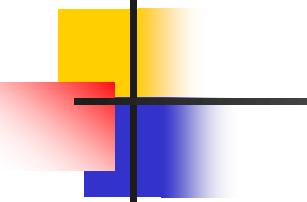
## *Exemplo 23.1 (continuação)*

*O SNMP usa dois números de portas (161 and 162), cada qual com uma finalidade específica.*

```
$ grep snmp /etc/services
snmp          161/tcp          #Simple Net Mgmt Proto
snmp          161/udp          #Simple Net Mgmt Proto
snmptrap      162/udp          #Traps for SNMP
```

**Figura 23.9 Formato do Datagrama de Usuário**



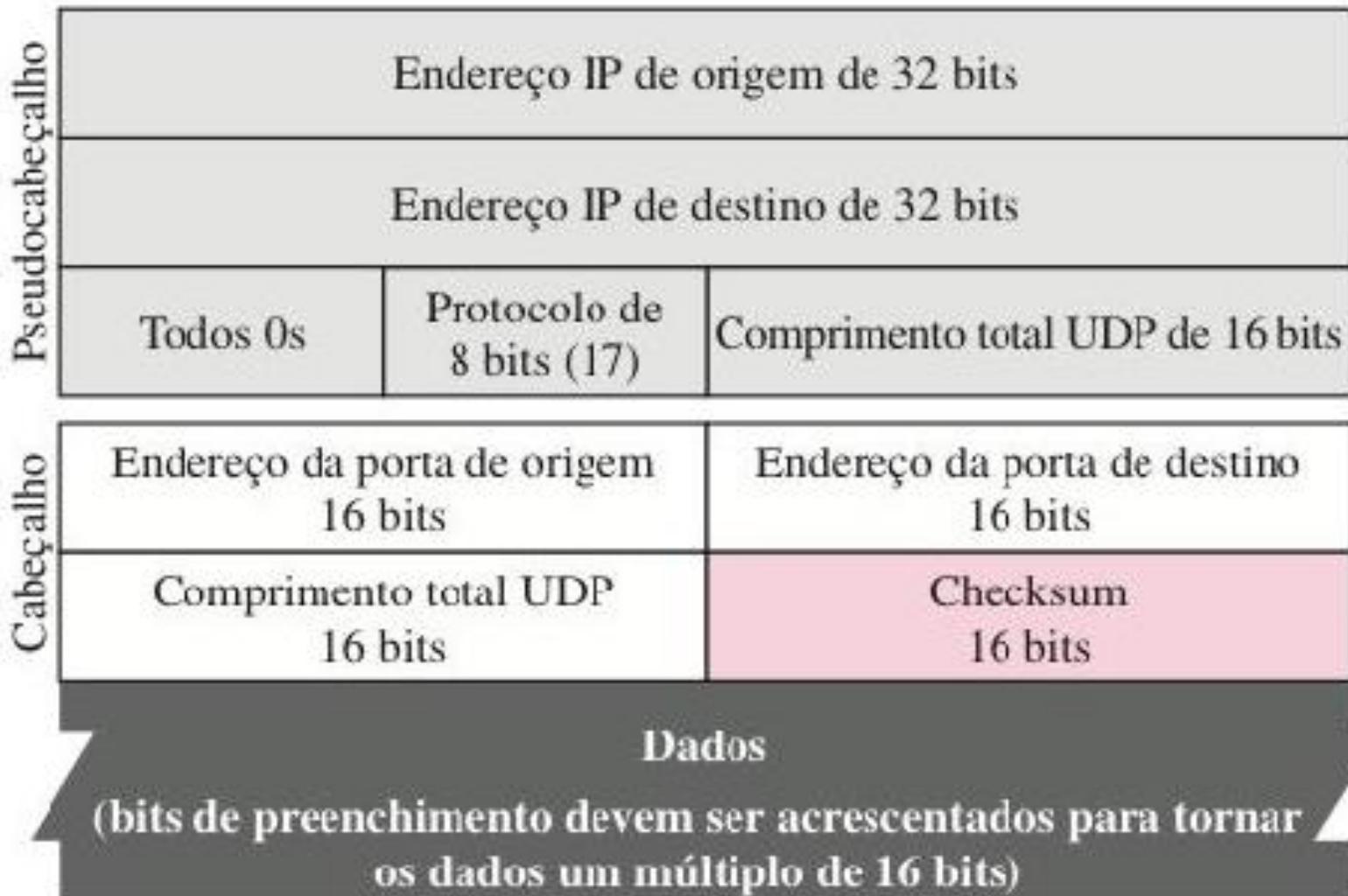


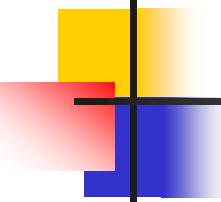
## *Nota*

---

**Comprimento do datagrama UDP**  
**= comprimento total do datagrama IP –**  
**comprimento do cabeçalho IP**

**Figura 23.10 Pseudocabeçalho para cálculo do checksum**





## *Exemplo 23.2*

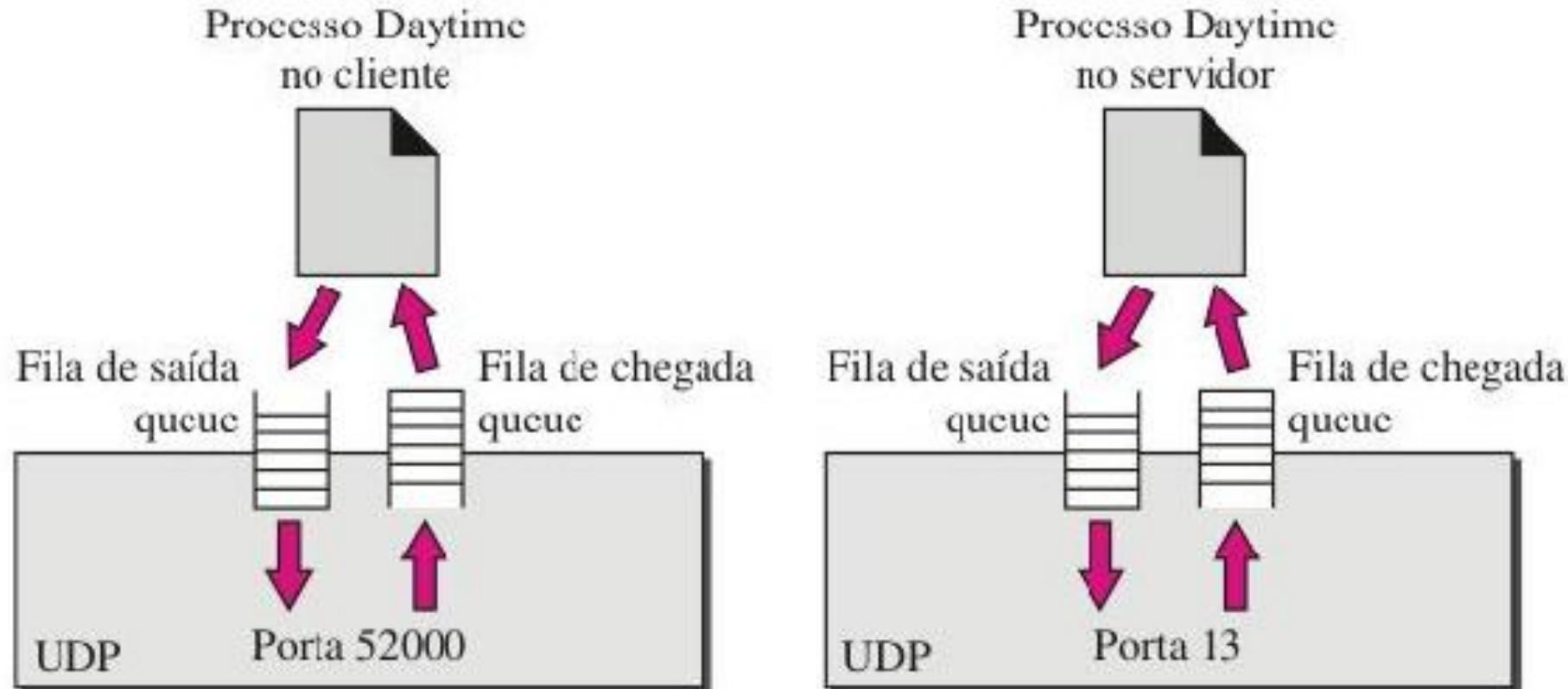
*A Figura 23.11 motra o cálculo do checksum para um datagrama UDP muito pequeno com apenas 7 bytes de dados. Como o número de bytes é ímpar, devemos acrecentar alguns bits de preenchimento antes de realizar o cálculo do checksum. O pseudocabeçalho, bem como os bits de preenchimento serão descartados quando o datagrama UDP for entregue a camada IP.*

## Figura 23.11 Cálculo do checksum para um datagrama de usuário UDP simples

153.18.8.105			
171.2.14.10			
<b>Todos 0s</b>	17	15	
1087		13	
15	<b>Todos 0s</b>		
T	E	S	T
I	N	G	<b>Todos 0s</b>

10011001	00010010	→ 153.18
00001000	01101001	→ 8.105
10101011	00000010	→ 171.2
00001110	00001010	→ 14.10
00000000	00010001	→ 0 e 17
00000000	00001111	→ 15
00000100	00111111	→ 1087
00000000	00001101	→ 13
00000000	00001111	→ 15
00000000	00000000	→ 0 (checksum)
01010100	01000101	→ T e E
01010011	01010100	→ S e T
01001001	01001110	→ I e N
01000111	00000000	→ G e 0 (preenchimento)
10010110		→ Sum
<b>01101001</b>		→ <b>Checksum</b>

**Figura 23.12 Filas no UDP**



## 23-3 PROTOCOLO DE CONTROLE DA TRANSMISSÃO (TCP)

*O TCP é um protocolo orientado a conexão; ele cria uma conexão virtual entre dois processos TCP para a transmissão de dados. Além disso, o TCP implementa mecanismos de controle de erros e controle de fluxo no nível de camada de transporte.*

### **Topicos discutidos nessa sessão:**

**Serviços do TCP**

**Recursos do TCP**

**Segmento**

**Um conexão TCP**

**Controle de fluxo**

**Controle de erros**

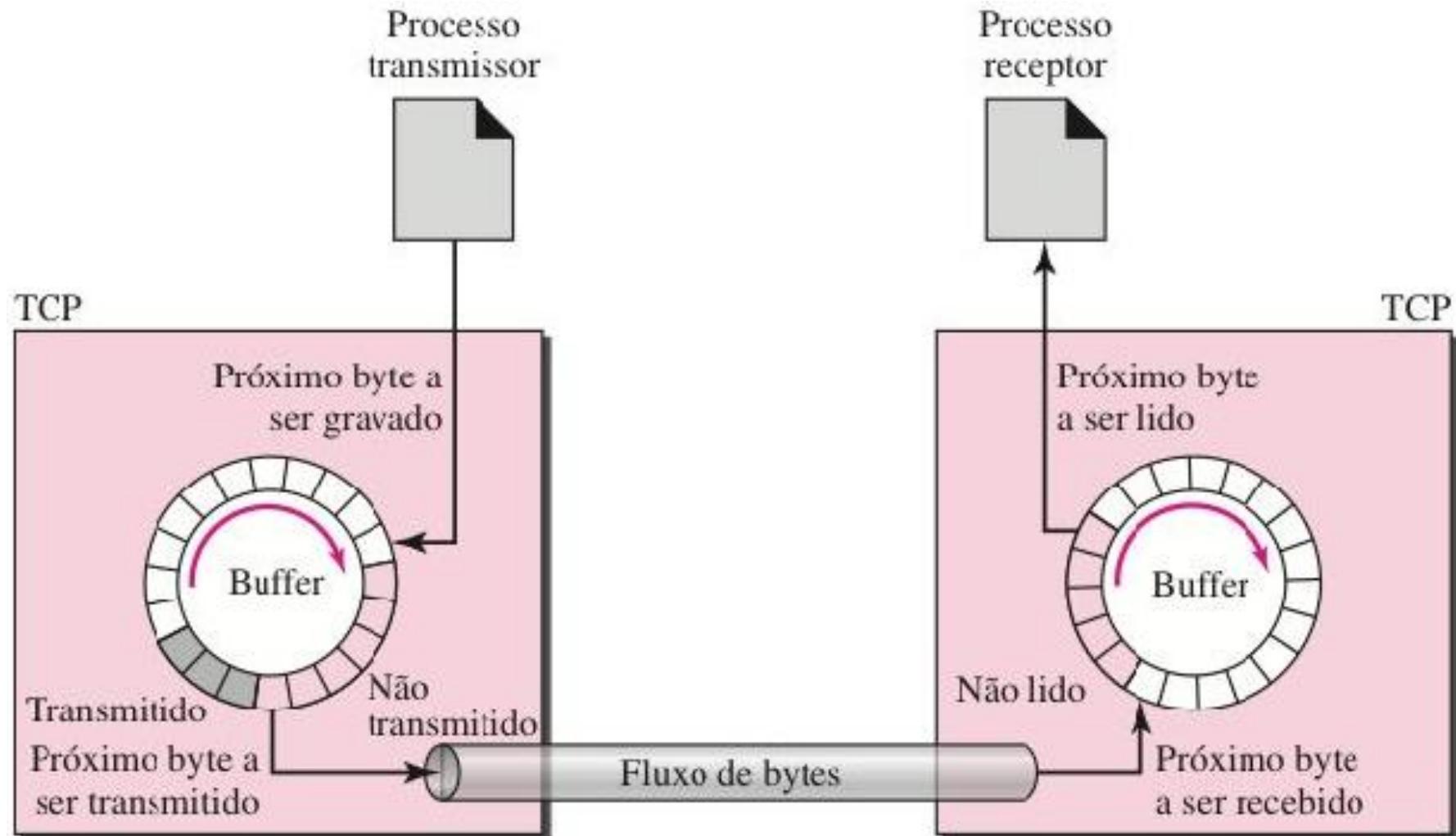
**Tabela 23.2** *Algumas portas bem conhecidas no TCP*

Porta	Protocolo	Descrição
7	Echo	Ecoa um datagrama recebido de volta para o emissor
9	Discard	Descarta qualquer datagrama recebido
11	Users	Usuários ativos
13	Daytime	Retorna a data e a hora
17	Quote	Retorna o comentário do dia
19	Chargen	Retorna uma string de caracteres
20	FTP, Dados	File Transfer Protocol (conexão de dados)
21	FTP, Control	File Transfer Protocol (conexão de controle)
23	TELNET	Telnet
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Services
67	BOOTP	Bootstrap protocol
79	Finger	Finger
80	HTTP	HyperText transfer protocol
111	RPC	Remote Procedure Call

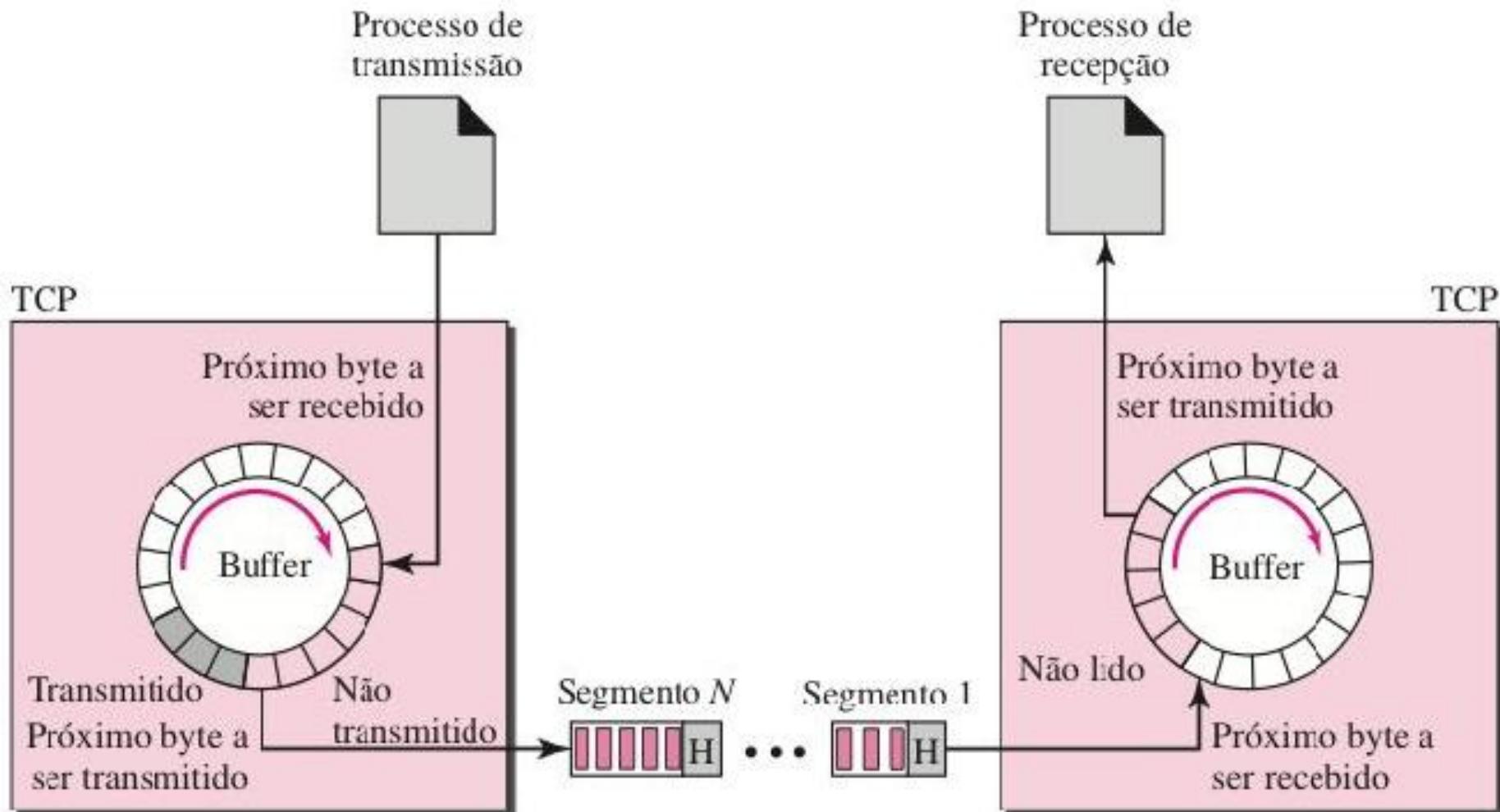
**Figura 23.13 Entrega de Fluxo de Dados**

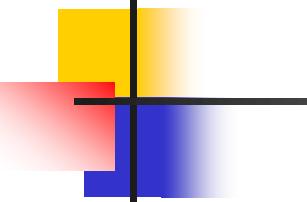


**Figura 23.14** *Buffers de transmissão e recepção*



**Figura 23.15 Segmentos TCP**





## *Nota*

**Os bytes de dados transmitidos em cada conexão são numerados pelo TCP. A numeração dos bytes começa com um número gerado randomicamente.**

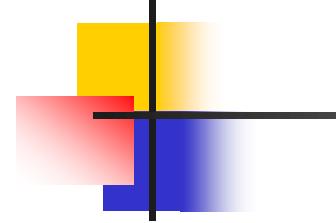
## Exemplo 23.3

*Suponha que uma conexão TCP esteja transferindo um arquivo de 5.000 bytes. O primeiro deles recebe a numeração 10.001. Quais são os número de sequência de cada segmento se os dados forem enviados em cinco segmentos, cada um transportando 1.000 bytes?*

**Solução:**

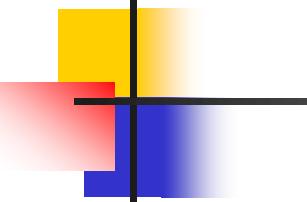
*A seguir, são apresentados os números de sequência de cada segmento:*

- Segmento 1 → Número de Seqüência: 10.001 (faixa: 10.001 a 11.000)
- Segmento 2 → Número de Seqüência: 11.001 (faixa: 11.001 a 12.000)
- Segmento 3 → Número de Seqüência: 12.001 (faixa: 12.001 a 13.000)
- Segmento 4 → Número de Seqüência: 13.001 (faixa: 13.001 a 14.000)
- Segmento 5 → Número de Seqüência: 14.001 (faixa: 14.001 a 15.000)



## *Nota*

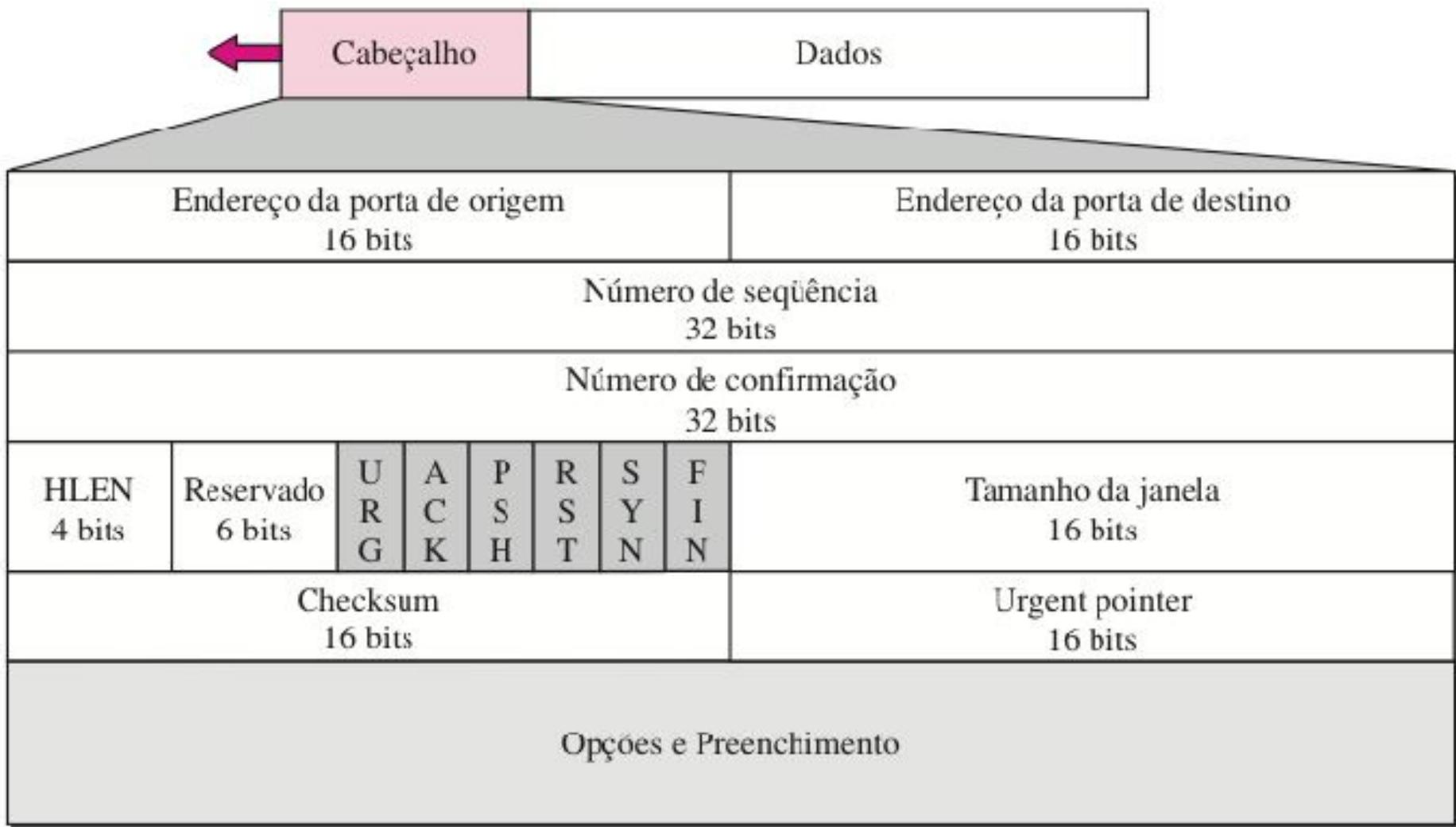
**O valor do campo número de sequência de um segmento define o número do primeiro byte de dados contido naquele segmento.**



## *Nota*

**O valor do campo reconhecimento (ACK) em um segmento define o número do próximo byte que o lado remoto espera receber. O número do reconhecimento é cumulativo.**

**Figura 23.16 Formato do segmento TCP**



## Figura 23.17 *Campo Controle*

URG: Urgent Pointer é válido

ACK: Número de Confirmação é válido

PSH: Solicitação para empurrar

RST: Reinicia a conexão

SYN: Sincroniza números de seqüência

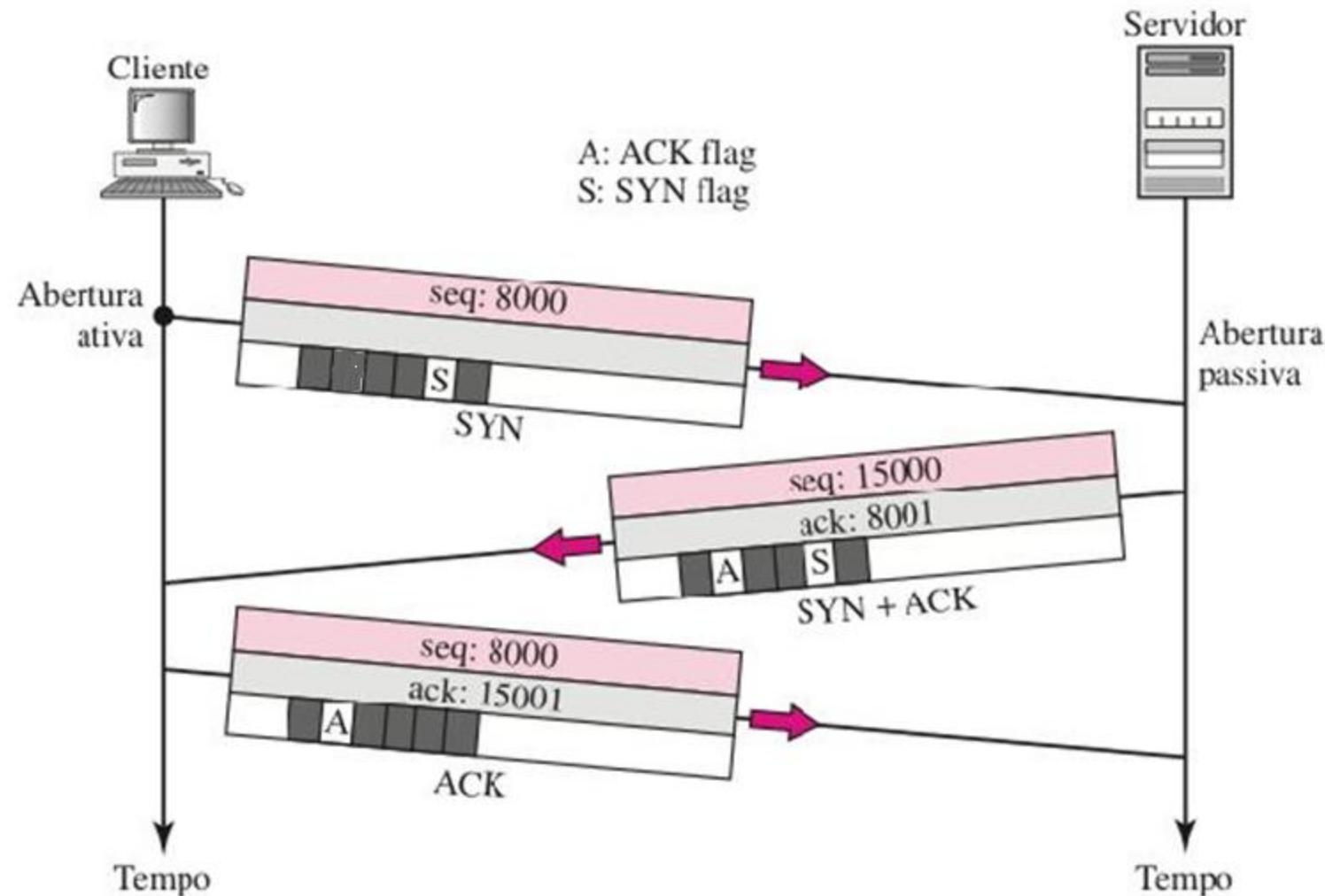
FIN: Encerra a conexão

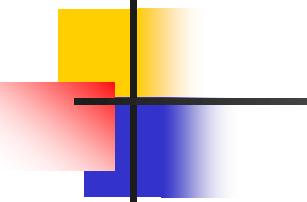
URG	ACK	PSH	RST	SYN	FIN
-----	-----	-----	-----	-----	-----

**Tabela 23.3** *Descrição dos flags do Campo Controle*

<i>Flag</i>	<i>Descrição</i>
URG	O valor do campo de Urgent Point é válido.
ACK	O valor do campo de confirmação é válido.
PSH	Empurra os dados.
RST	Reinicia a conexão.
SYN	Sincroniza números de seqüência durante a conexão.
FIN	Encerra a conexão.

**Figura 23.18** *Estabelecimento de conexão usando handshaking de 3 vias*

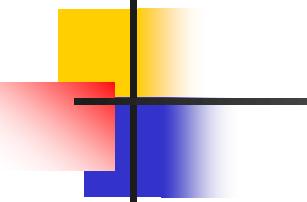




## *Nota*

---

**Apesar do segmento SYN não transportar dados, ele incrementa o número de sequência em 1 byte.**

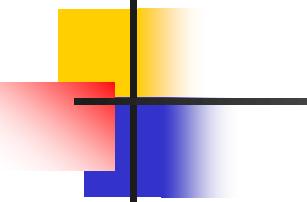


## *Nota*

---

**Apesar do segmento SYN + ACK não transportar dados, ele incrementa o número de sequência em 1 byte.**

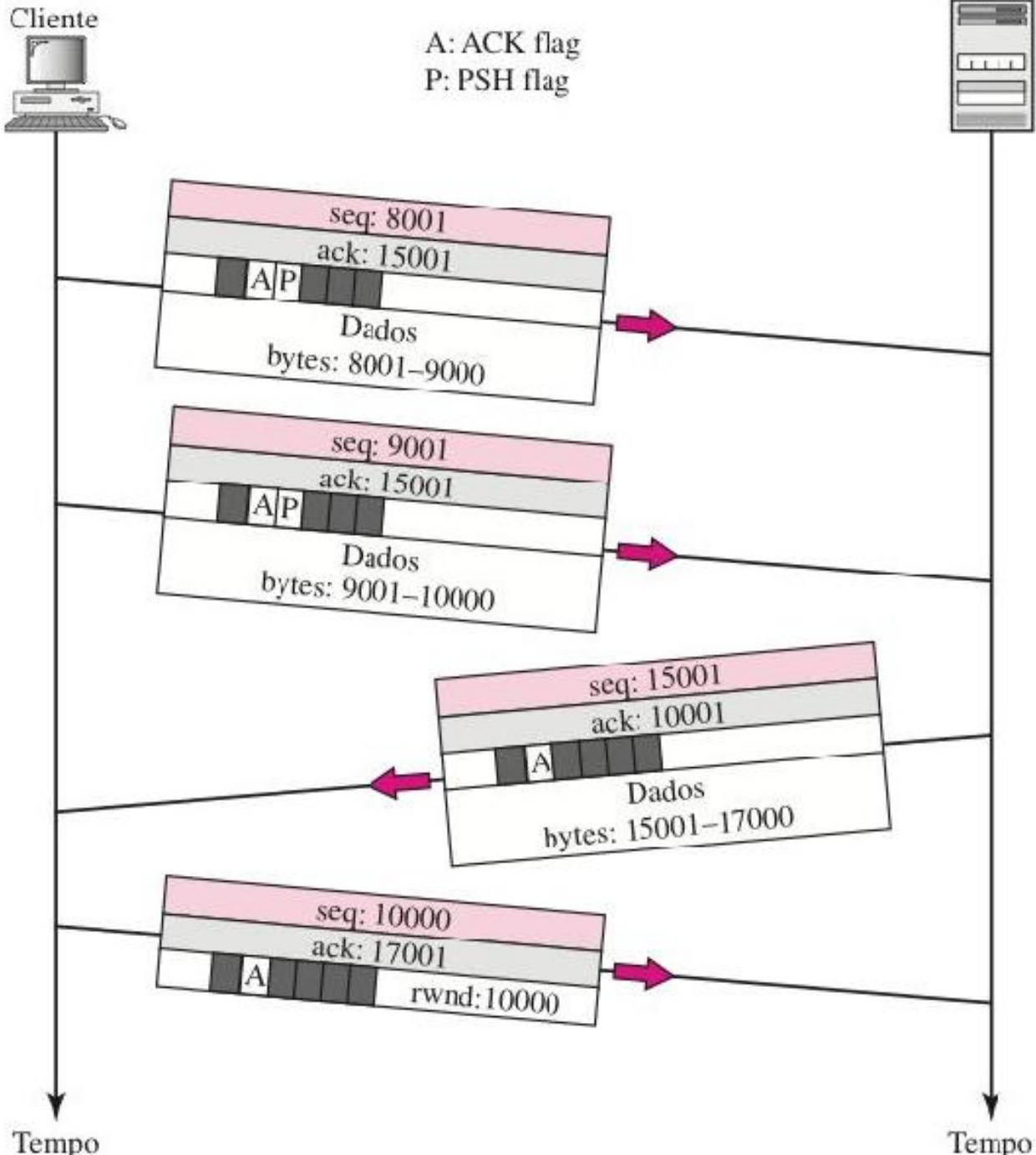
---



## *Nota*

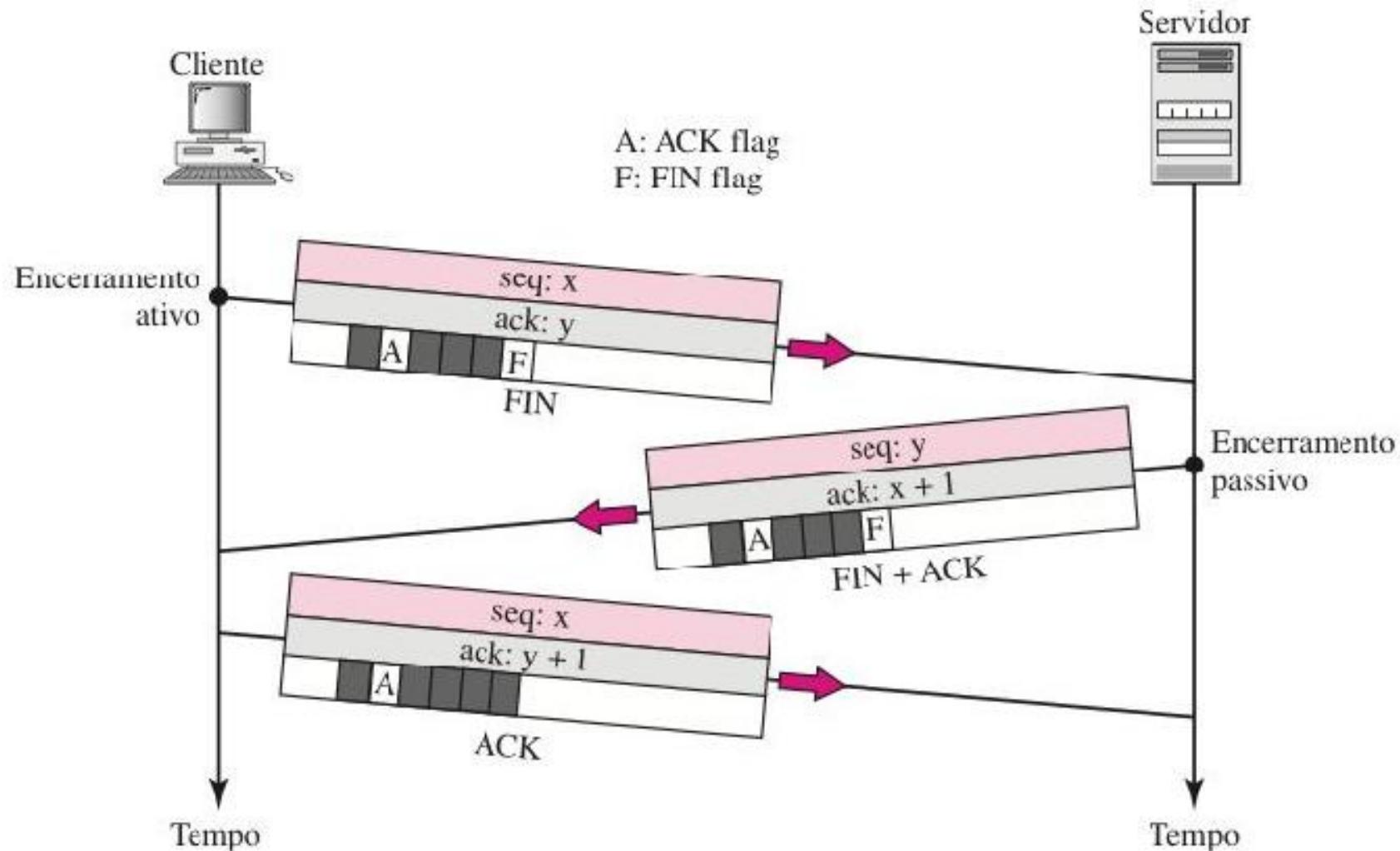
---

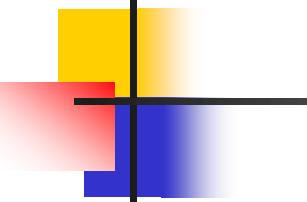
**Se o segmento ACK não transportar dados, ele não incrementa o número de sequência.**



**Figura 23.19**  
*Transferência de  
dados*

**Figura 23.20 Término da conexão usando handshaking de 3 vias**



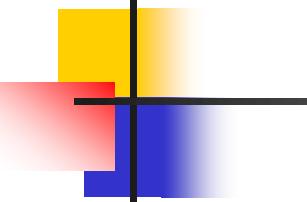


## *Nota*

---

**Mesmo quando o segmento FIN não transportar dados, ele incrementa o número de sequência em 1 byte.**

---

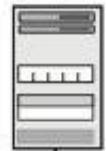


## *Nota*

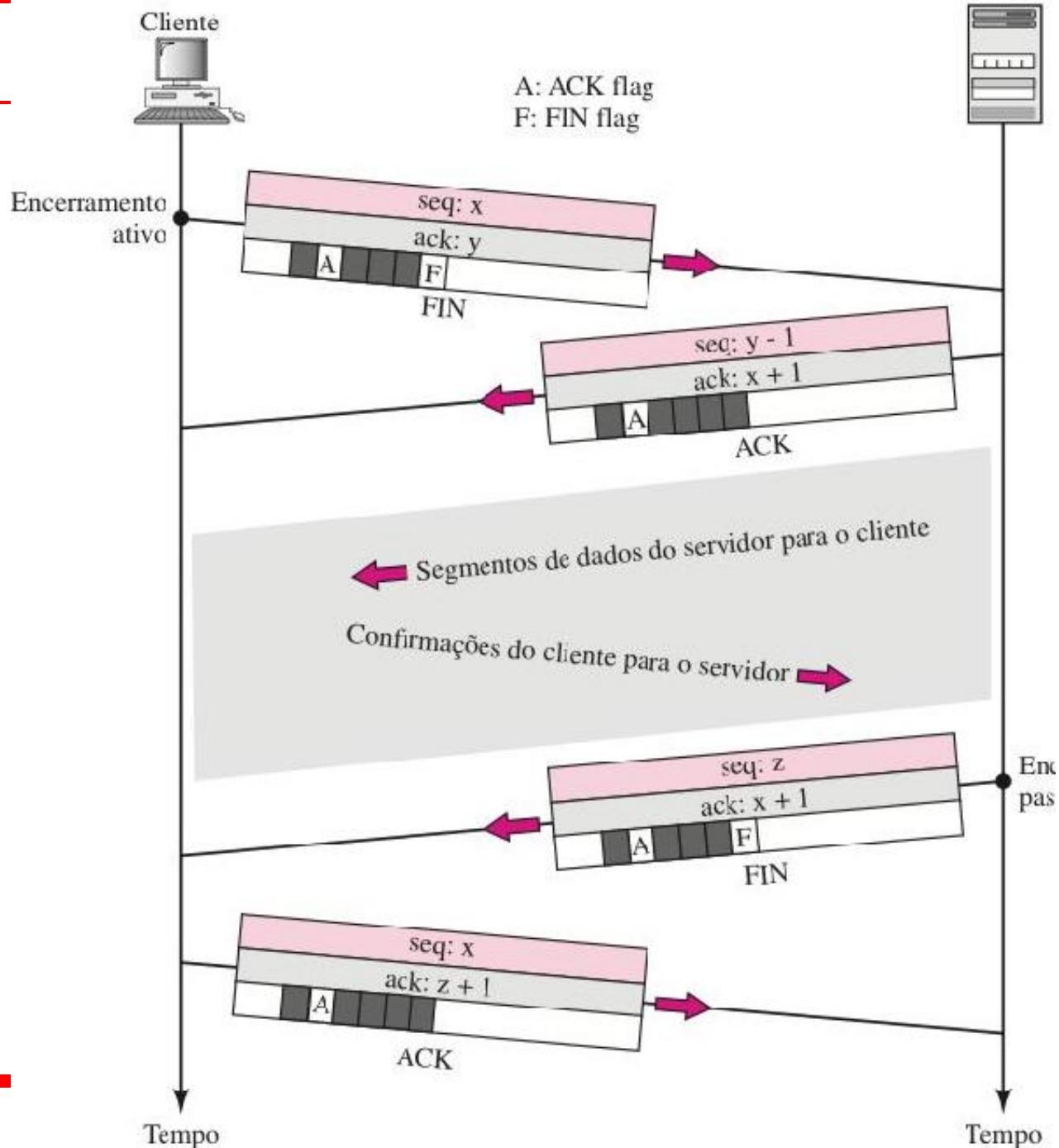
---

**Apesar do segmento FIN + ACK não transportar dados, ele incrementa o número de sequência em 1 byte.**

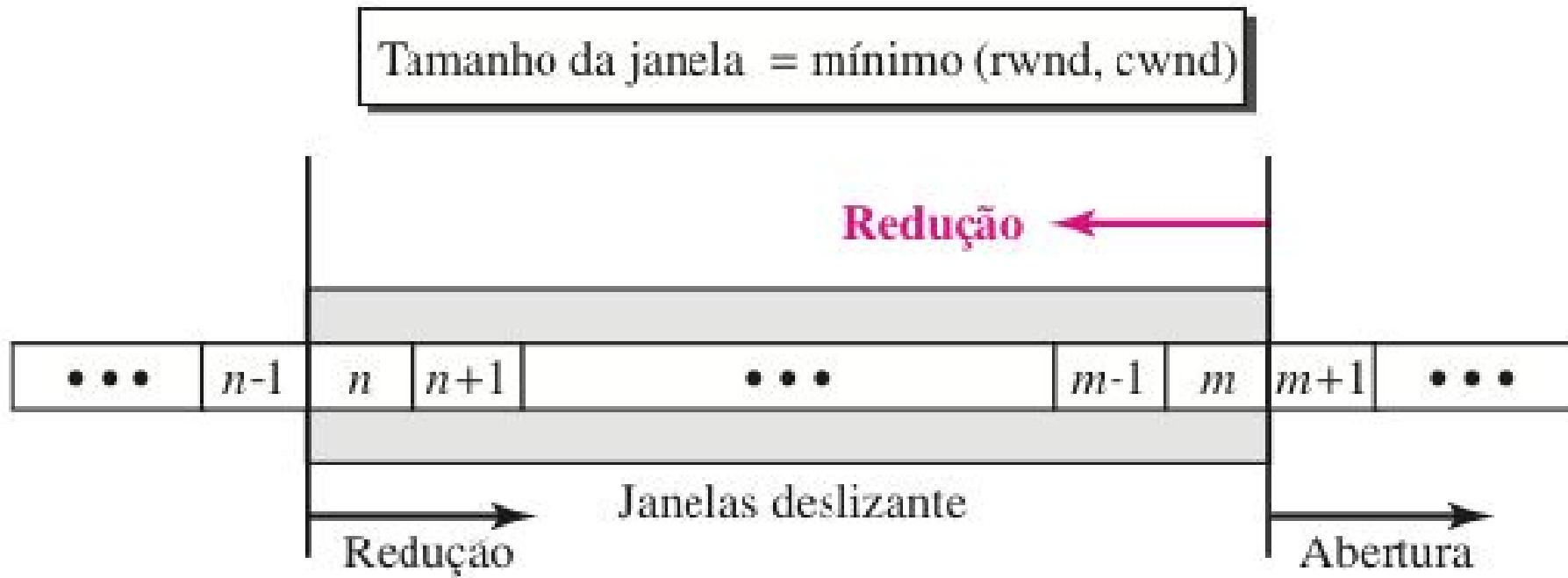
---



**Figura 23.21**  
*Semi-encerramento*

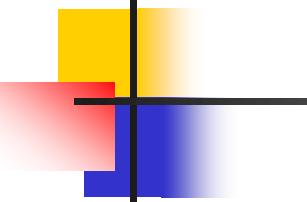


**Figura 23.22 Janela deslizante**



**rwnd: janela receptora**

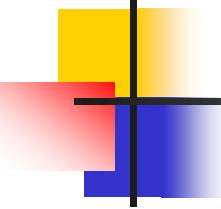
**cwnd: janela de congestionamento**



## *Nota*

**Uma janela deslizante é usada para implementar maior eficiência à transmissão, bem como o controle de fluxo de dados, de modo que o destino não fique sobrecarregado com dados.**

**As janelas deslizantes no TCP são orientadas a bytes.**

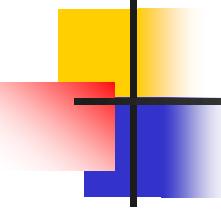


## Exemplo 23.4

*Qual é o valor da janela receptora (rwnd) para o host A, se o receptor, o host B, tiver um tamanho de buffer de 5000 bytes e 1000 bytes de dados recebidos e não processados?*

**Solução:**

*O valor de rwnd = 5000 – 1000 = 4000. O host B pode receber apenas 4000 bytes de dados para não estourar o seu buffer. O host B anuncia esse valor no próximo segmento enviado para o host A.*



## *Exemplo 23.5*

*Qual é o valor da janela de um host A se o valor de rwnd e o valor de é 3500 bytes?*

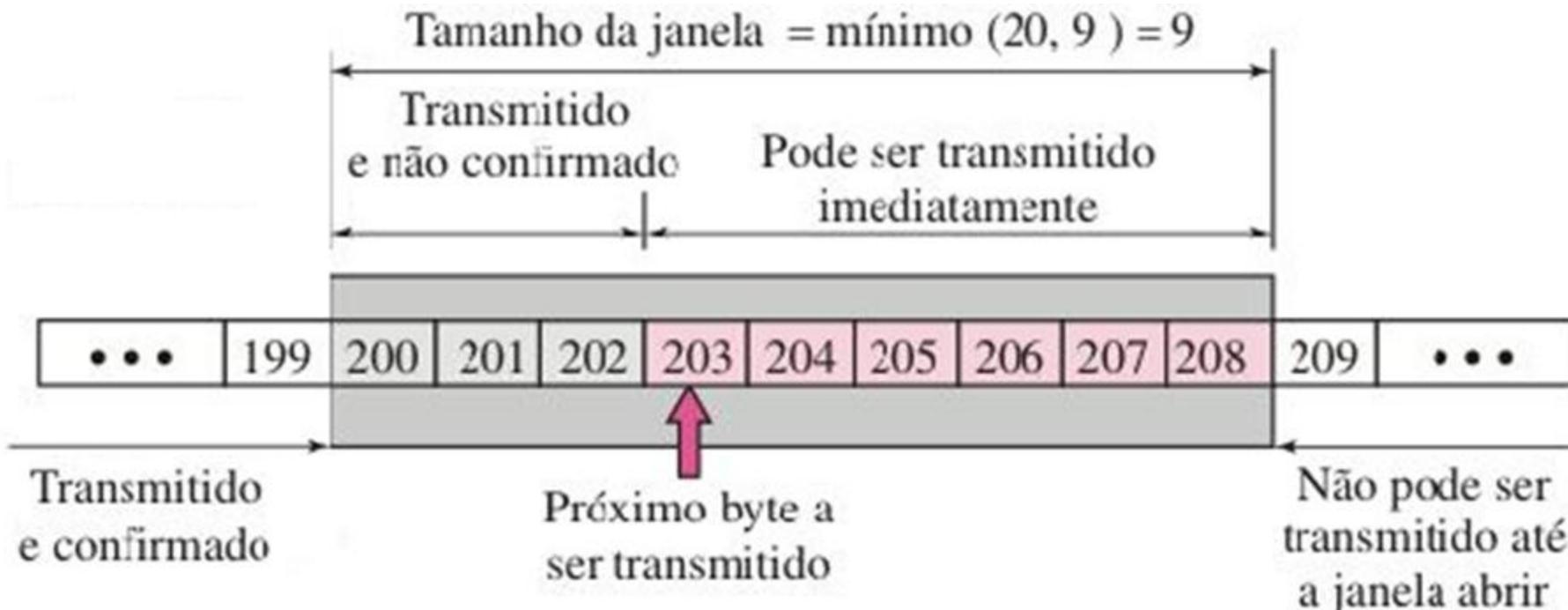
*Solução:*

*O tamanho da janela é o menor valor entre rwnd e cwnd, que é 3000 bytes.*

## Exemplo 23.6

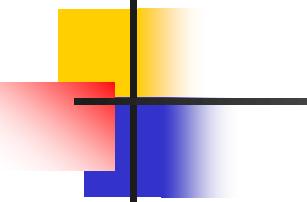
*A Figura 23.23 mostra um valor fictício de uma janela deslizante de um host. O transmissor enviou bytes acima de 202. Estamos supondo que cwnd seja 20 (na prática, esse valor pode ser de milhares de bytes). O receptor enviou o número de confirmação 200 com rwnd de 9 bytes. O tamanho da janela no transmissor é o menor valor entre rwnd e cwnd, ou 9 bytes. Os bytes 200 a 202 são enviados, mas não são confirmados. Os bytes 203 a 208 podem ser enviados antes de receber o reconhecimento (ACK). Os bytes acima de 209 ainda não podem ser enviados.*

**Figura 23.23 Exemplo 23.6**



## Alguns comentários sobre janelas deslizantes no TCP:

- ❑ O tamanho da janela é o menor valor entre rwnd e cwnd.
- ❑ O transmissor não deve enviar uma janela de dados completa.
- ❑ A janela pode ser aberta ou fechada pelo receptor, porém, não pode ser reduzida.
- ❑ O receptor pode enviar uma confirmação a qualquer instante, desde que não resulte na redução da janela.
- ❑ O receptor pode fechar temporariamente uma janela; o transmissor, entretanto, sempre pode transmitir um segmento de 1 byte após a janela ter sido fechada.

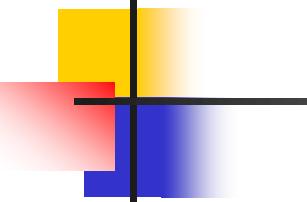


## *Nota*

---

**Os segmentos ACK não incrementam o número de sequência e não podem ser confirmados.**

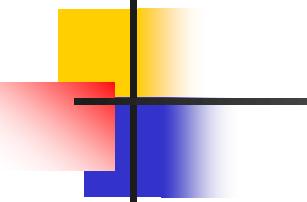
---



## *Nota*

---

**Nas implementações modernas, ocorre retransmissão caso o timer expire ou se tiverem chegados 3 segmentos ACK duplicados.**

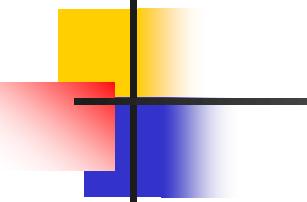


## *Nota*

---

**Não é ativado nenhum timer de retransmissão para o segmento ACK.**

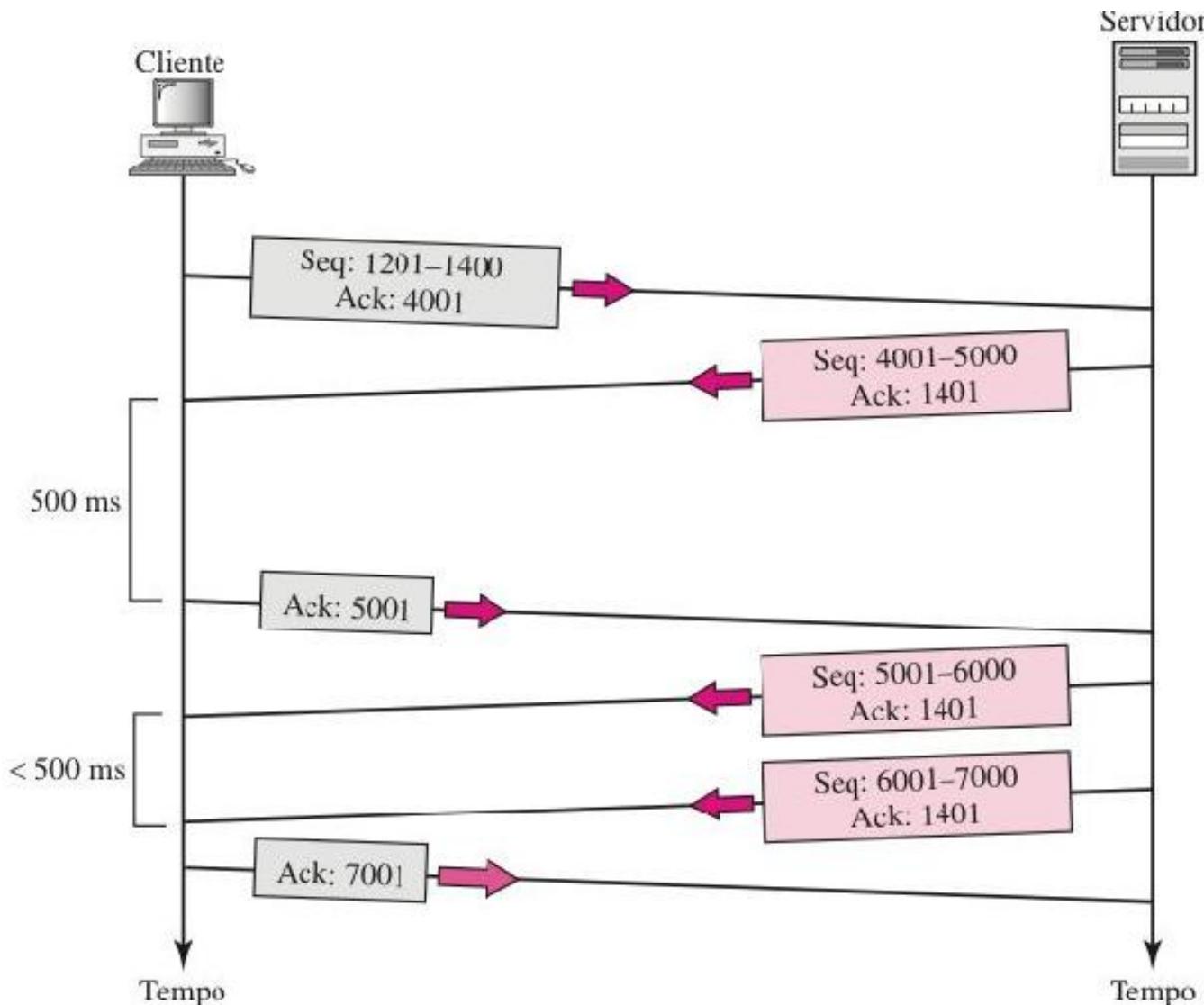
---



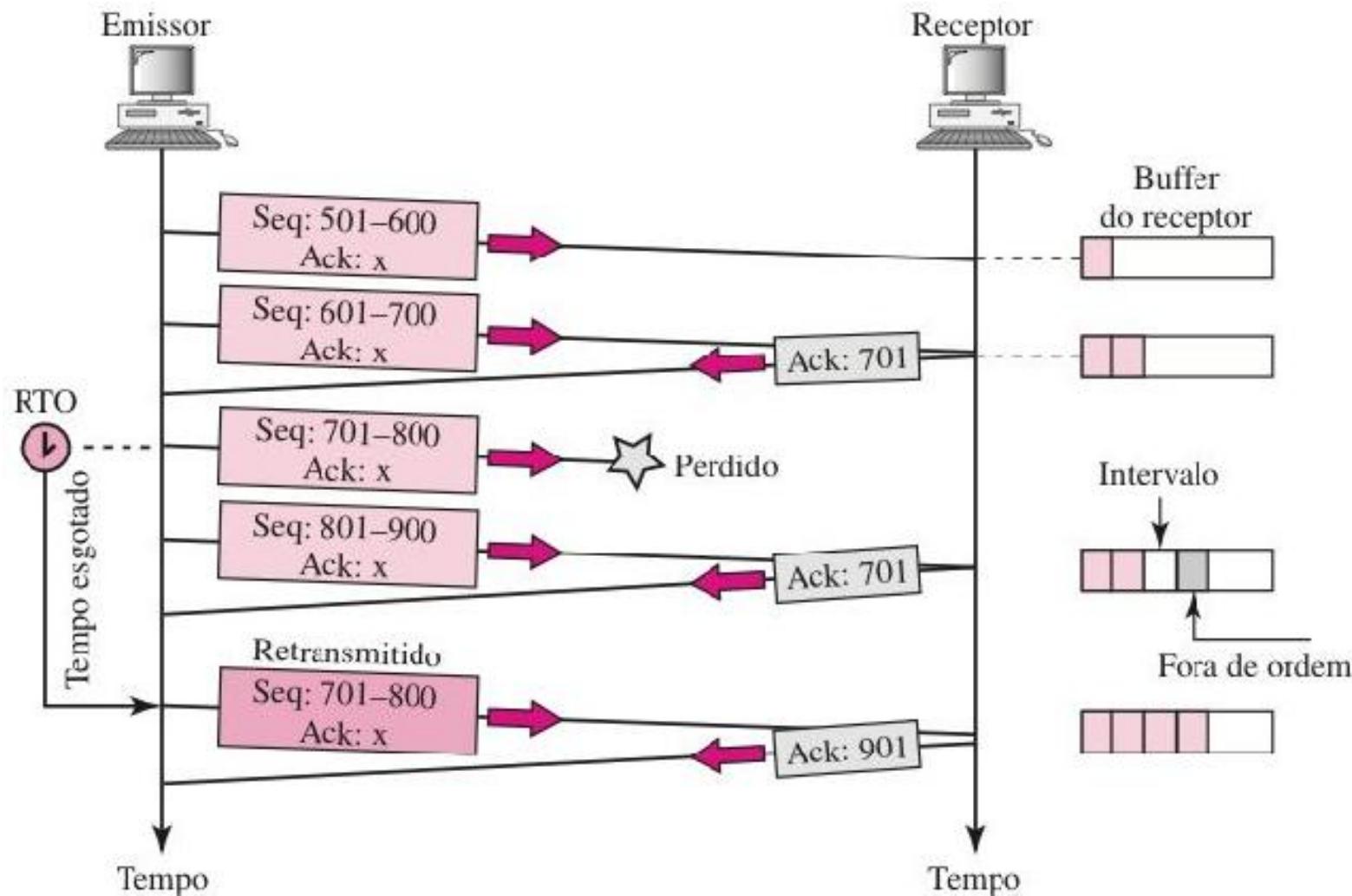
## **Nota**

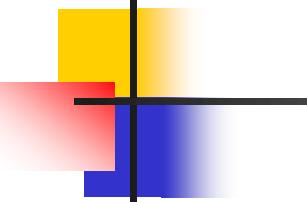
**Existe a possibilidade dos dados chegarem fora de ordem e serem armazenados temporariamente pelo receptor TCP, porém o TCP garante que nenhum segmento fora de ordem será entregue ao processo receptor da camada superior.**

**Figura 23.24 Operação Normal**



## Figura 23.25 Segmento Perdido



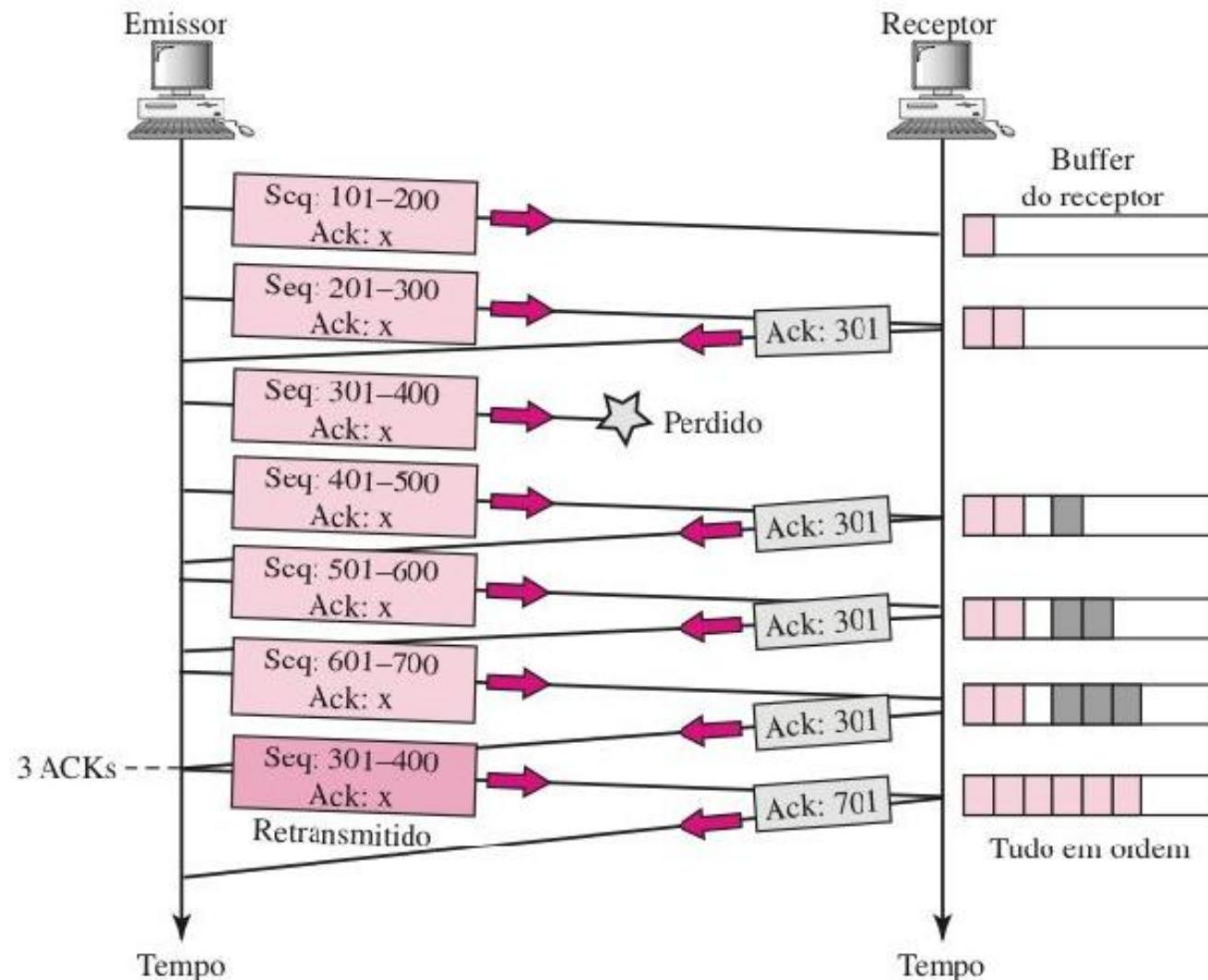


## *Nota*

---

**O receptor TCP entrega apenas dados ordenados para o processo na camada superior.**

## Figura 23.26 Retransmissão rápida



## 23-4 SCTP

*O SCTP (Protocolo de Controle da Transmissão em Fluxo) é um novo protocolo da camada de transporte, confiável e orientado a mensagens. O SCTP, entretanto, foi especificado, em sua maior parte, para atender os requisitos das novas especificações das novas aplicações da Internet recentemente introduzidas, como o VoIP, que necessitam de serviços mais sofisticados do que os que são oferecidos pelo TCP.*

### **Tópicos discutidos nessa sessão:**

**Serviços e recursos do SCTP**

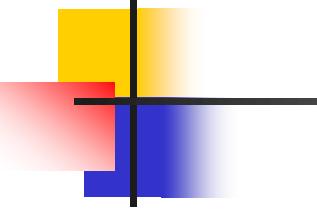
**Formato do pacote**

**Uma associação SCTP**

**Controle de fluxo e controle de erros**

## UDP vs TCP vs SCTP

UDP	TCP	SCTP
<b>Orientado a mensagens</b>	<b>Orientado a bytes</b>	<b>Orientado a mensagens</b>
<b>Datagrama</b>	<b>Segmento</b>	<b>Pacotes</b>
<b>Não confiável</b>	<b>Confiável</b>	<b>Confiável</b>
<b>Sem conexão lógica</b>	<b>Conexão lógica</b>	<b>Associação</b>

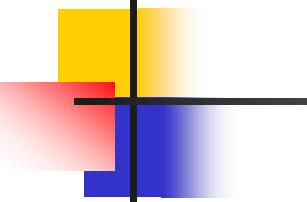


*Nota*

---

**Uma conexão SCTP é chamada de associação.**

---



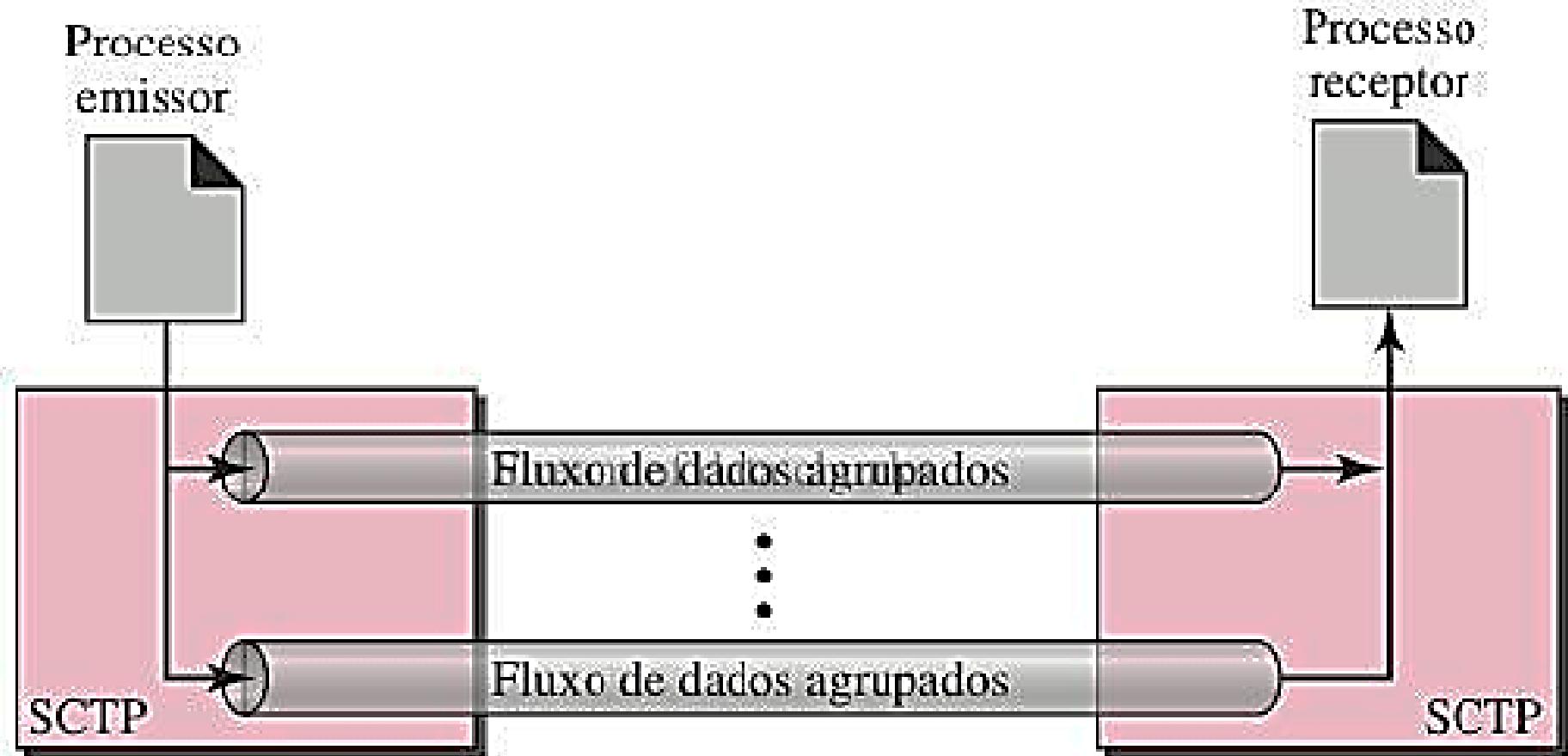
## *Nota*

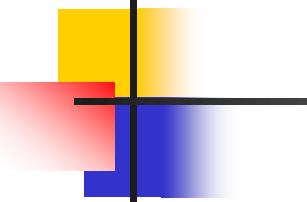
**O SCTP é um protocolo confiável orientado a mensagens que combina as melhores características do UDP e do TCP.**

**Tabela 23.4 Aplicações do SCTP**

<i>Protocolo</i>	<i>Número da porta</i>	<i>Descrição</i>
IUA	9990	ISDN sobre IP
M2UA	2904	Sinalização SS7 para telefonia
M3UA	2905	Sinalização SS7 para telefonia
H.248	2945	<i>Media gateway control</i>
H.323	1718, 1719, 1720, 11720	Telefonia IP
SIP	5060	Telefonia IP

**Figura 23.27 Conceito de múltiplos fluxos de dados**





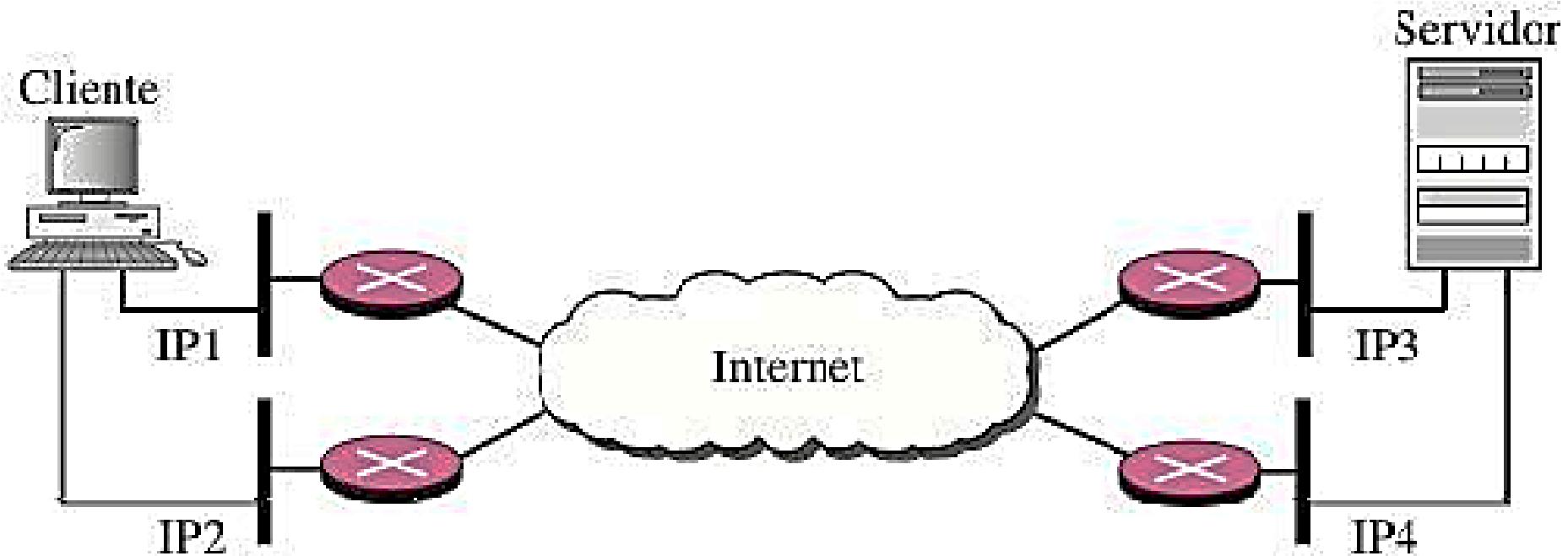
## *Nota*

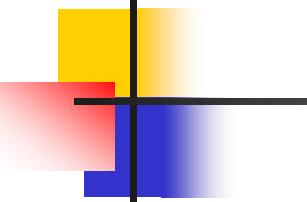
---

**Uma associação no SCTP pode envolver múltiplos fluxos de dados.**

---

**Figura 23.28 Conceito de Multihoming**



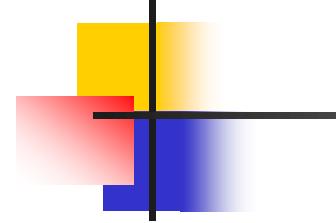


## *Nota*

---

**Uma associação SCTP permite vários endereços IP para cada ponto terminal.**

---

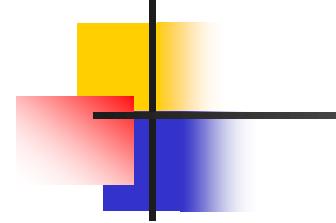


## *Nota*

---

**Os conjuntos de blocos de dados possuem os identificadores: TSN, SI e SSN.**

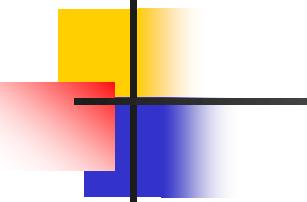
- **TSN**: número cumulativo que identifica a associação;
- **SI**: define o fluxo
- **SSN**: identifica o bloco em um fluxo de dados.



## **Nota**

---

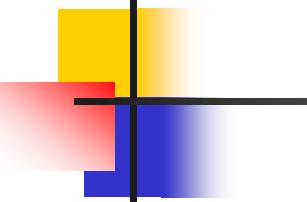
**No SCTP um bloco de dados é numerado usando um TSN (Número de Sequência da Transmissão).**



## *Nota*

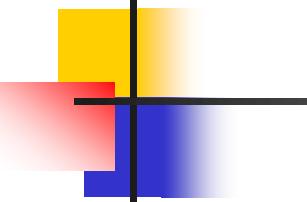
---

**Para fazer a distinção entre fluxos diferentes, o SCTP usa um SI (Identificador de Fluxo).**



## **Nota**

**Para distinguir diferentes conjuntos de blocos de dados pertencentes a um mesmo fluxo de dados, o SCTP usa SSNs (Números de Sequência de Fluxos).**

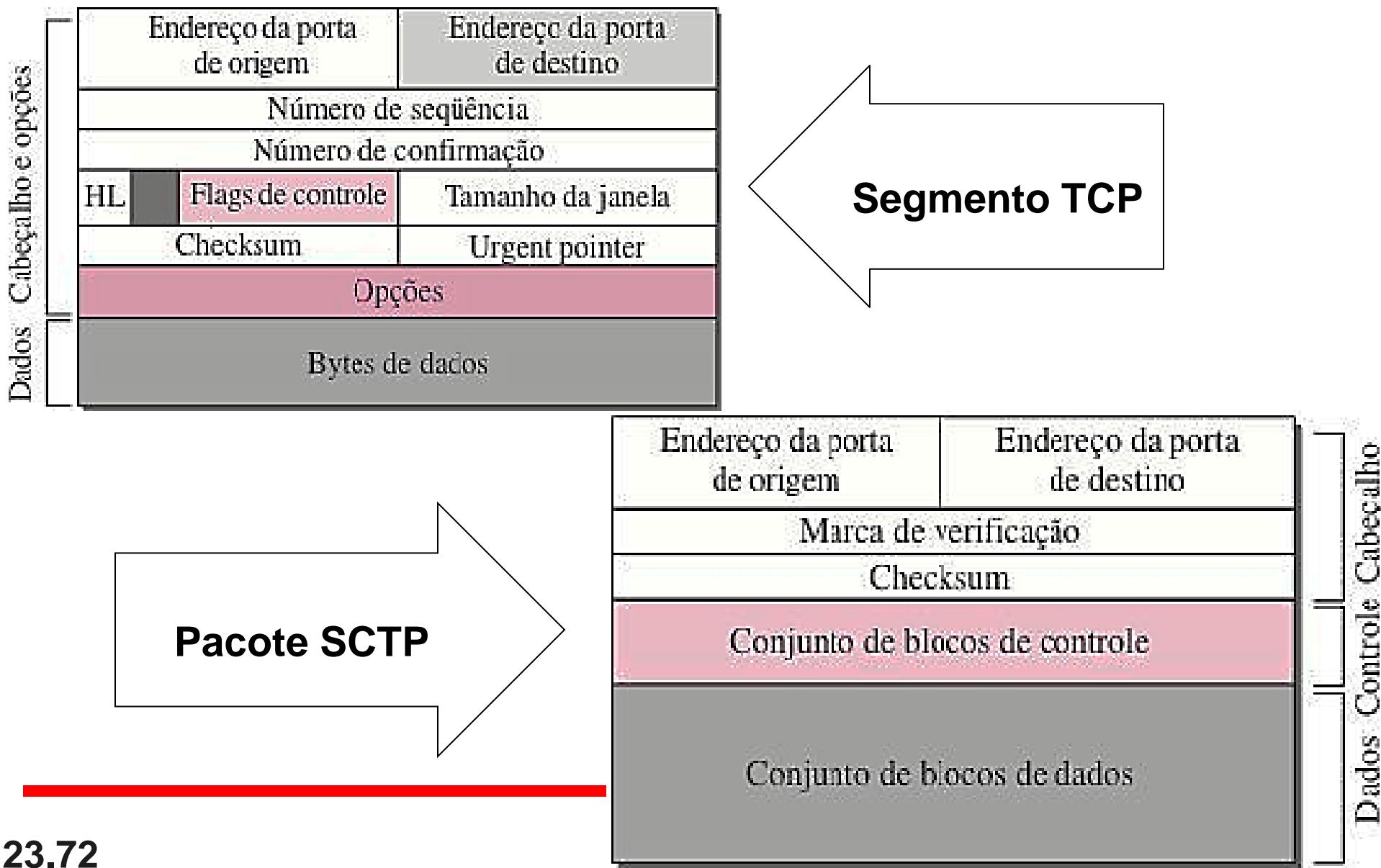


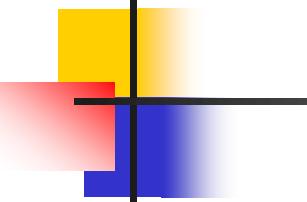
## *Nota*

---

**Enquanto o TCP possui segmentos, o  
SCTP possui pacotes.**

**Figura 23.29 Comparação entre um segmento TCP e um pacote SCTP**





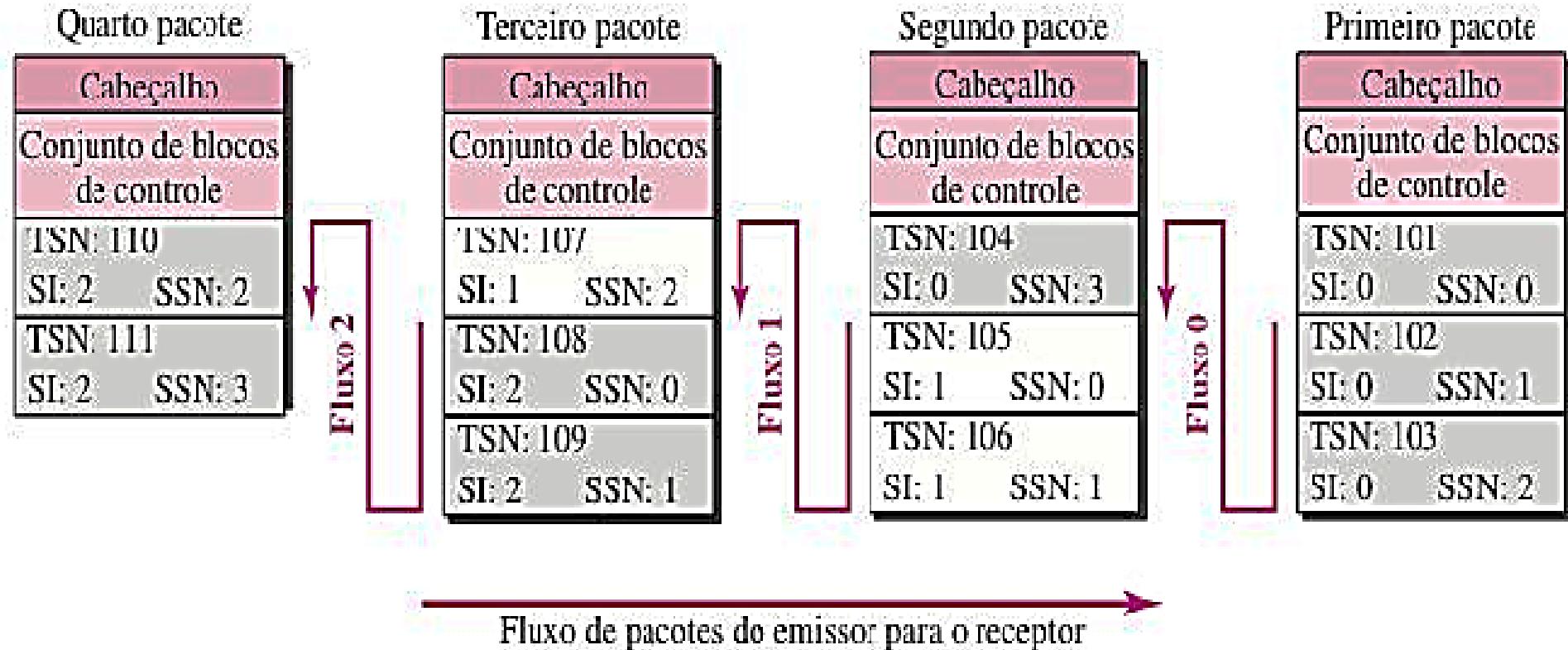
## *Nota*

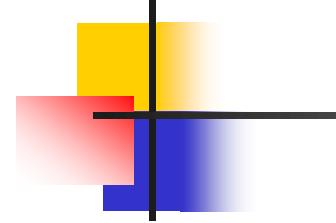
---

**No SCTP as informações de controle e os dados são transportados em conjuntos de blocos separados.**

---

**Figura 23.30 Pacote, conjunto de blocos e fluxos de dados**



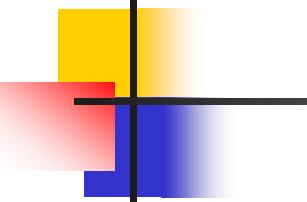


## *Nota*

---

**Os conjuntos de blocos de dados possuem identificadores: TSN, SI e SSN.**

- **TSN**: número cumulativo que identifica a associação;
- **SI**: define o fluxo
- **SSN**: identifica o bloco em um fluxo de dados.

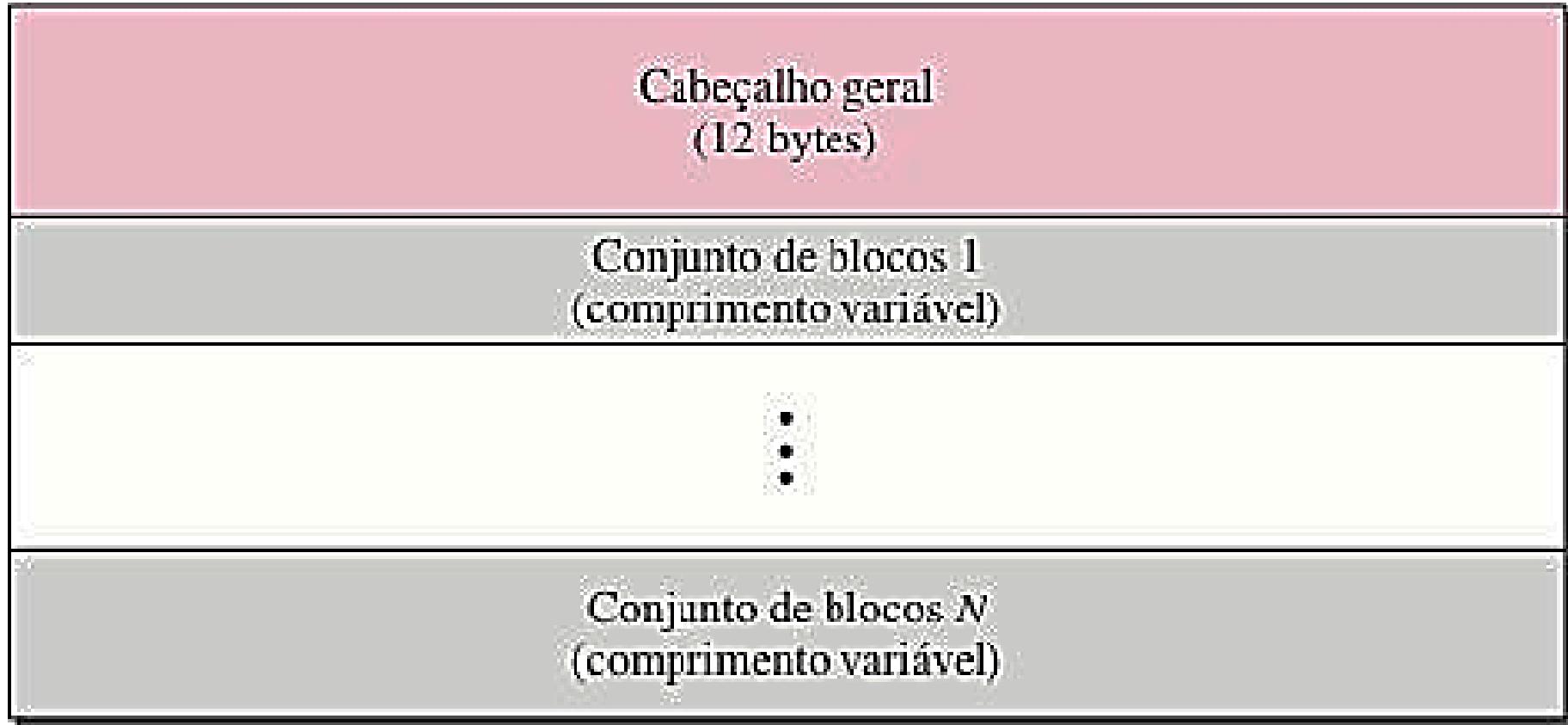


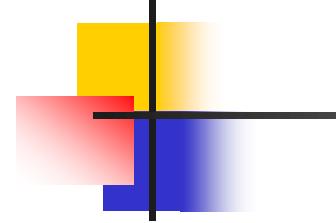
## **Nota**

**No SCTP os números de confirmação (ACK) são usados para confirmar apenas conjuntos de blocos de dados; os conjuntos de blocos de controle são confirmados por outros conjuntos de blocos de controle, se necessários.**

**Figura 23.31 Formato do pacoteSCTP**

---





## **Nota**

---

**Em um pacote SCTP os conjuntos de blocos de controle são transmitidos antes dos conjuntos de blocos de dados.**

**Figura 23.32 Cabeçalho Geral**

---



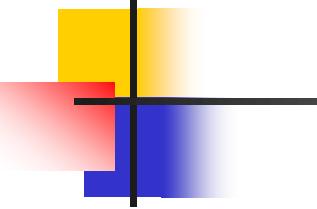
## *Campos do cabeçalho geral*

---

- ❑ **Endereço da porta de origem.** Trata-se de um campo de 16 bits que define o número da porta do processo que está enviando o pacote.
- ❑ **Endereço da porta de destino.** Trata-se de um campo de 16 bits que estabelece o número da porta do processo que receberá o pacote.
- ❑ **Marca de verificação (*Verification Tag*).** Este é um número que correlaciona um pacote a uma associação. Isso impede um pacote de uma associação anterior ser confundida com um pacote dessa associação. Ele serve como identificador da associação; ele é repetido em todos os pacotes de uma associação. Existe um processo de verificação distinto para cada direção da associação.
- ❑ **Checksum.** Esse campo de 32 bits armazena o valor calculado do checksum CRC-32. Note que o tamanho do campo checksum foi aumentado de 16 (no UDP, TCP e IP) para 32 bits, permitindo o uso do CRC-32.

**Tabela 23.5 Conjuntos de blocos**

<i>Descrição</i>	<i>conjunto de blocos</i>	<i>Tipo</i>
Dados de usuário	<b>DATA</b>	0
Ativa uma associação	<b>INIT</b>	1
Confirma o conjunto de blocos INIT	<b>INIT ACK</b>	2
Confirmação seletiva	<b>SACK</b>	3
Testa o par para ver se este está vivo	<b>HEARTBEAT</b>	4
Confirma o conjunto de blocos HEARTBEAT	<b>HEARTBEAT ACK</b>	5
Aborta uma associação	<b>ABORT</b>	6
Encerra uma associação	<b>SHUTDOWN</b>	7
Confirma o conjunto de blocos SHUTDOWN	<b>SHUTDOWN ACK</b>	8
Informa erros sem desligar	<b>ERROR</b>	9
Terceiro pacote do estabelecimento da associação	<b>COOKIE ECHO</b>	10
Confirma o conjunto de blocos COOKIE ECHO	<b>COOKIE ACK</b>	11
Terceiro pacote do encerramento de uma associação	<b>SHUTDOWN COMPLETE</b>	14
Permite ajustar TSNs acumulativos	<b>FORWARD TSN</b>	192

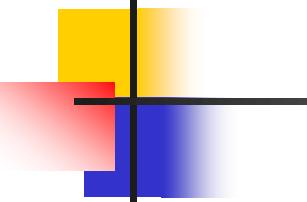


*Nota*

---

**Uma conexão SCTP é chamada de associação.**

---

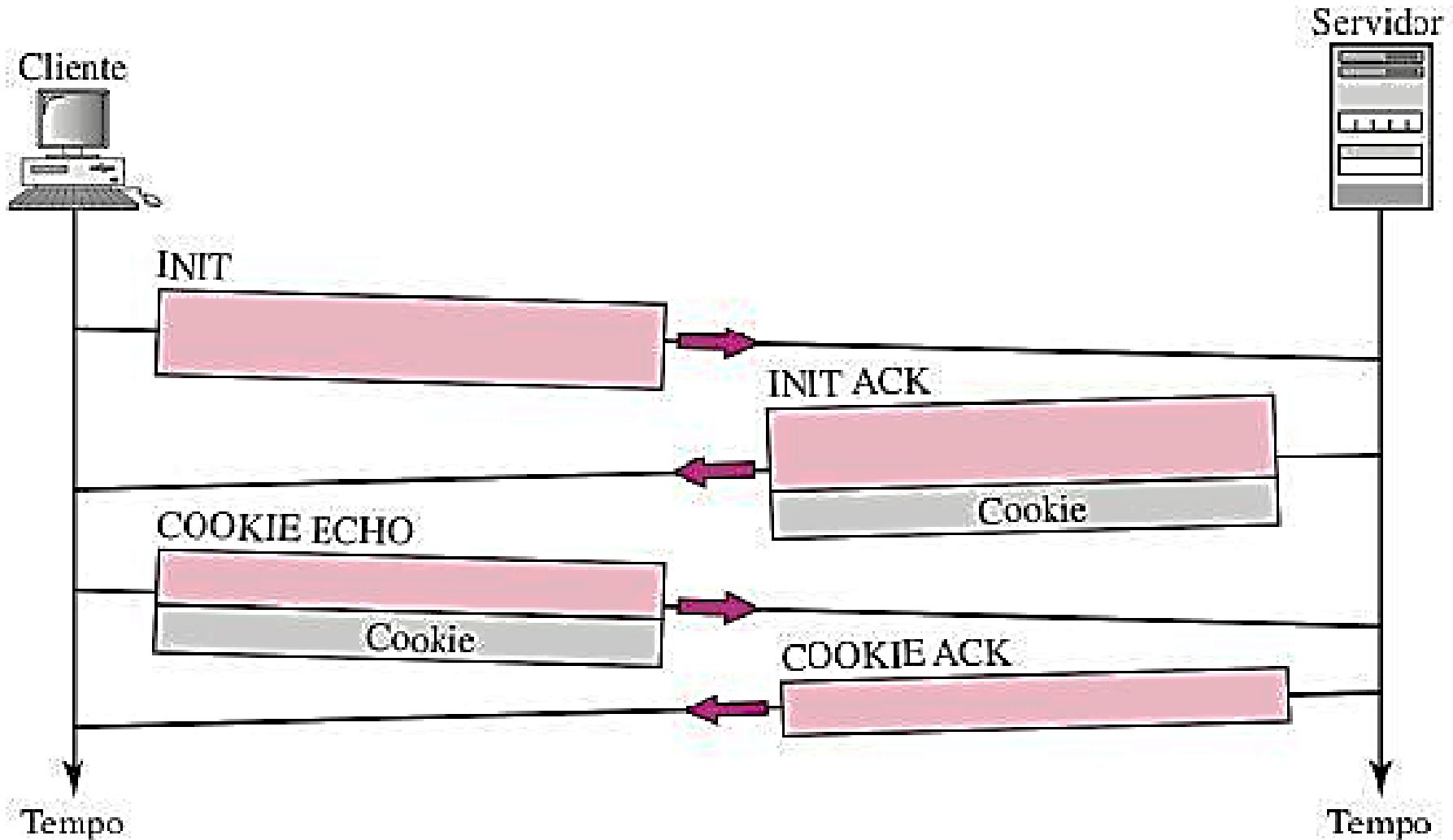


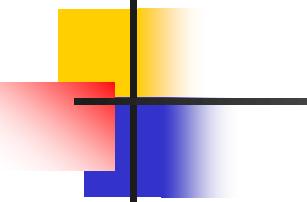
## **Nota**

**Não é permitido nenhum outro conjunto de blocos INIT ou INIT ACK.**

**Um conjunto de blocos COOKIE ECHO ou um COOKIE ACK pode transportar simultaneamente conjuntos de blocos de dados.**

Figura 23.33 *Handshaking de quatro vias*

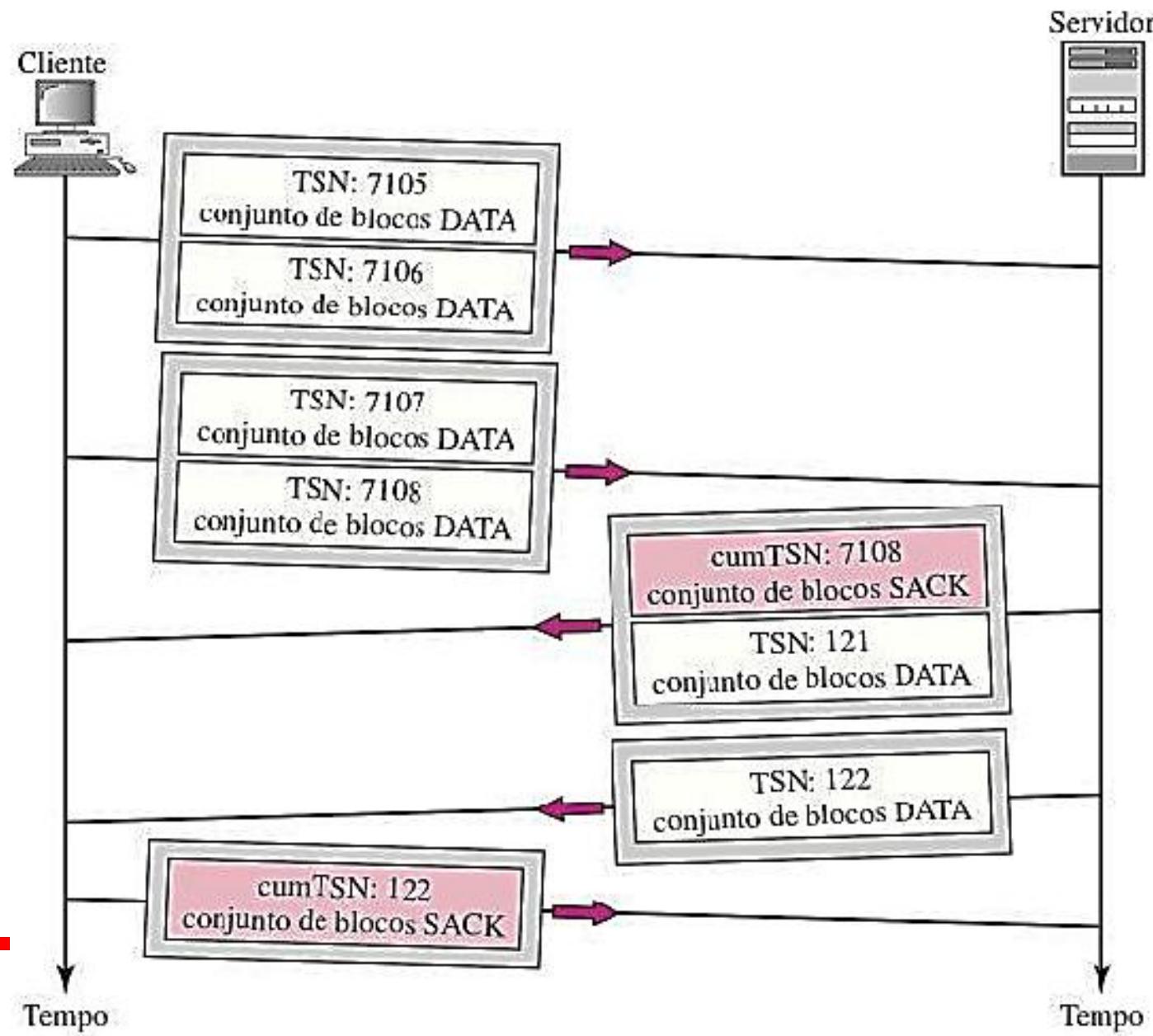


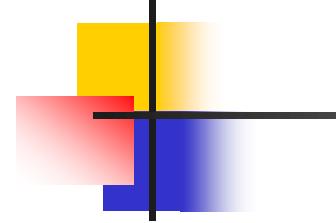


## **Nota**

**No SCTP apenas conjuntos de bloco de DATA consomem TSNs;  
Conjuntos de bloco DATA são os únicos conjuntos de blocos que podem ser confirmados (ACK).**

**Figura 23.34 Transferência de dados simples**

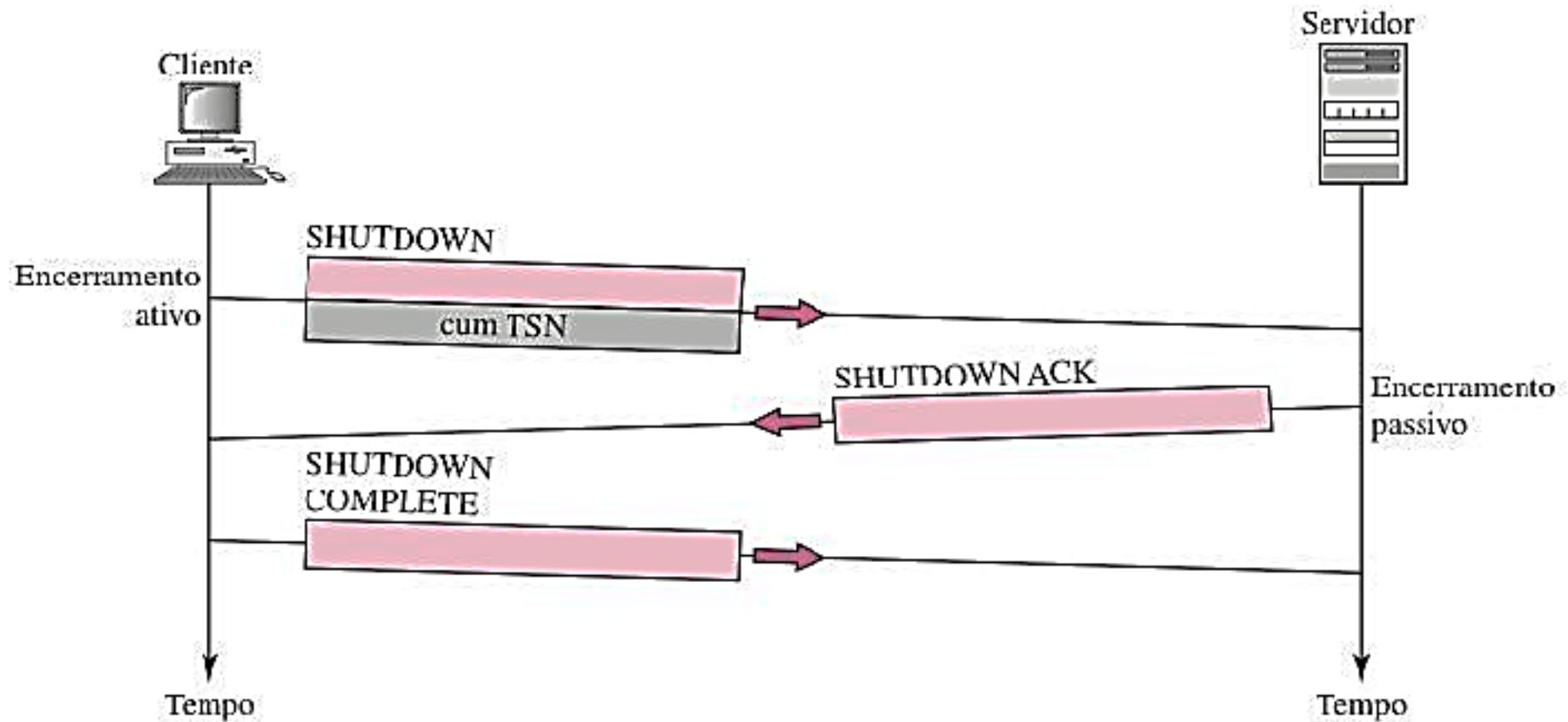




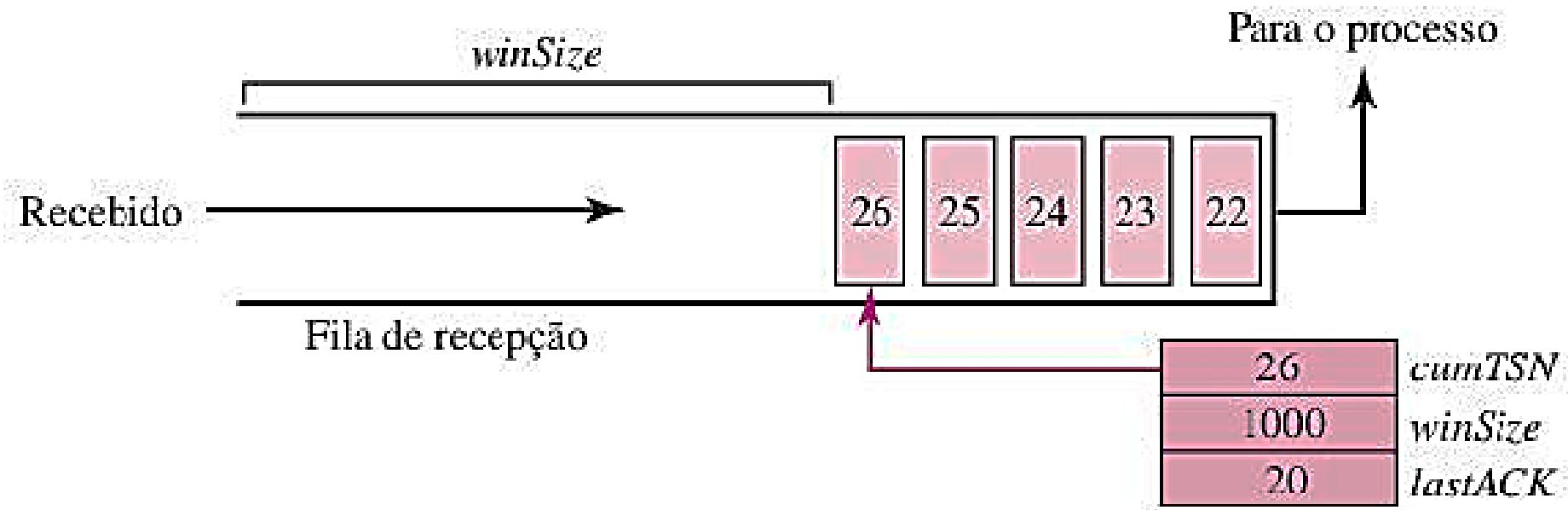
## *Nota*

**Uma confirmação (ACK) SCTP informa o valor do TSN acumulado, o TSN do último bloco de dados recebido em ordem.**

Figura 23.35 Término da associação



**Figura 23.36 Controle de fluxo, lado do receptor**



**Figura 23.37 Controle de fluxo, lado do transmissor**

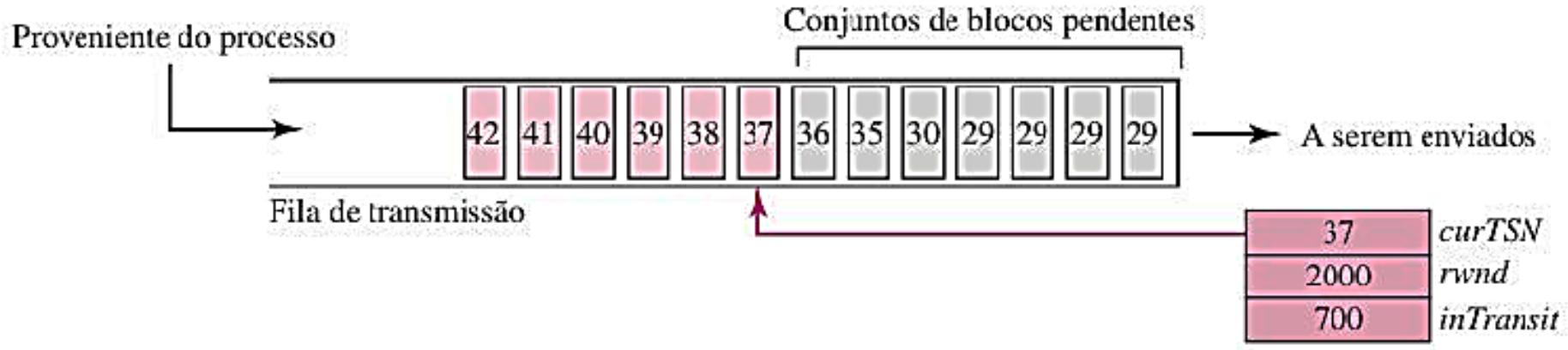
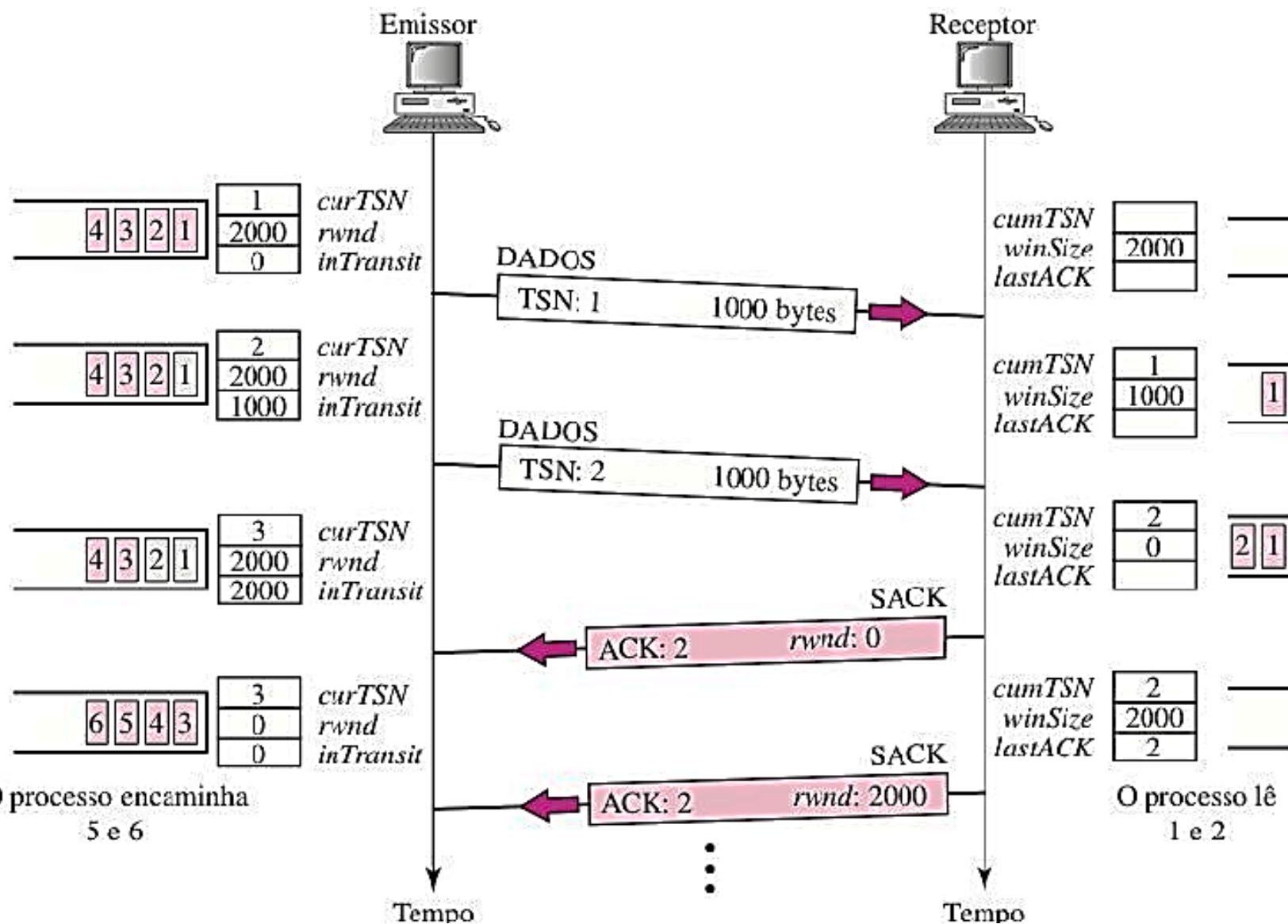
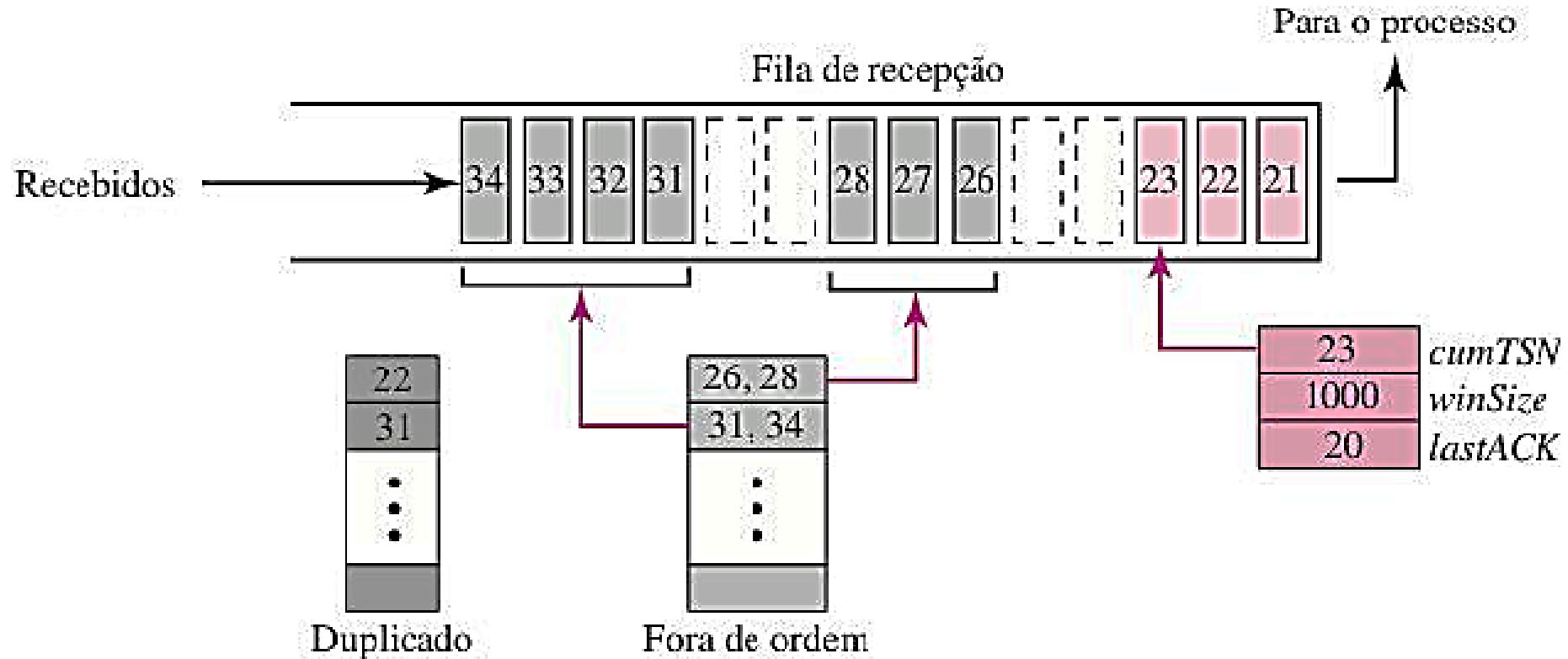


Figura 23.38 Cenário para estudo do controle de fluxo



**Figura 23.39 Controle de erros, lado do receptor**



**Figura 23.40 Controle de erros, lado do transmissor**

