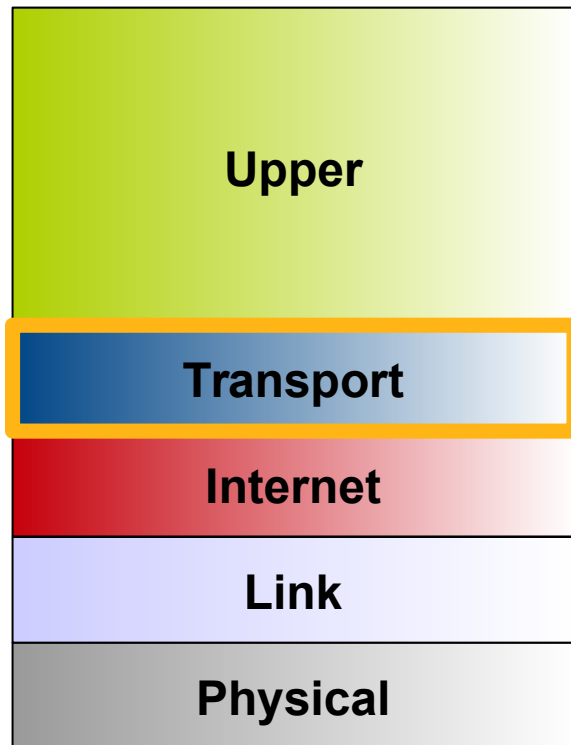


UDP & TCP

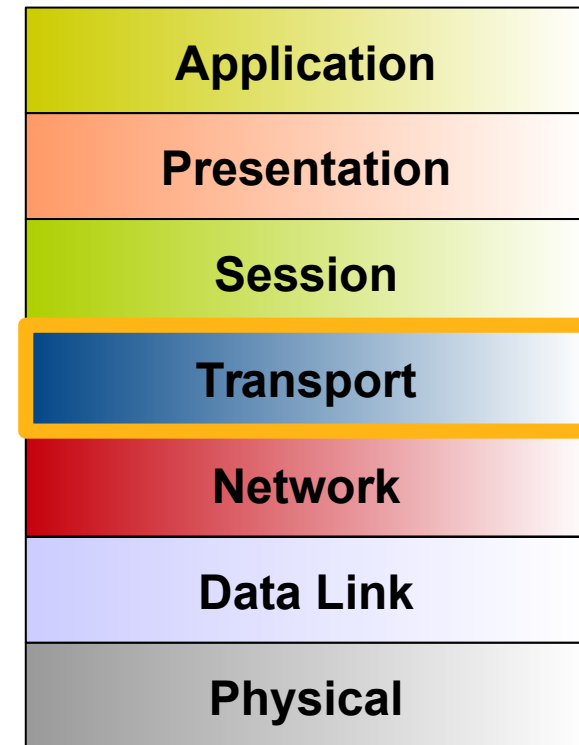
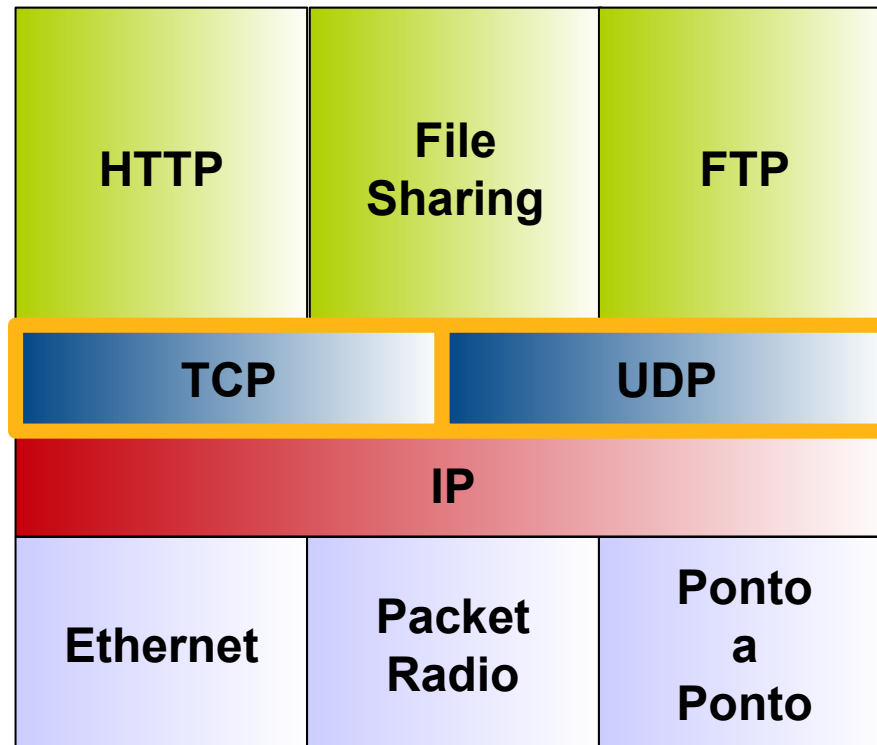
Redes e Serviços

**Licenciatura em Engenharia Informática
DETI-UA**

Modelo de referência TCP/IP



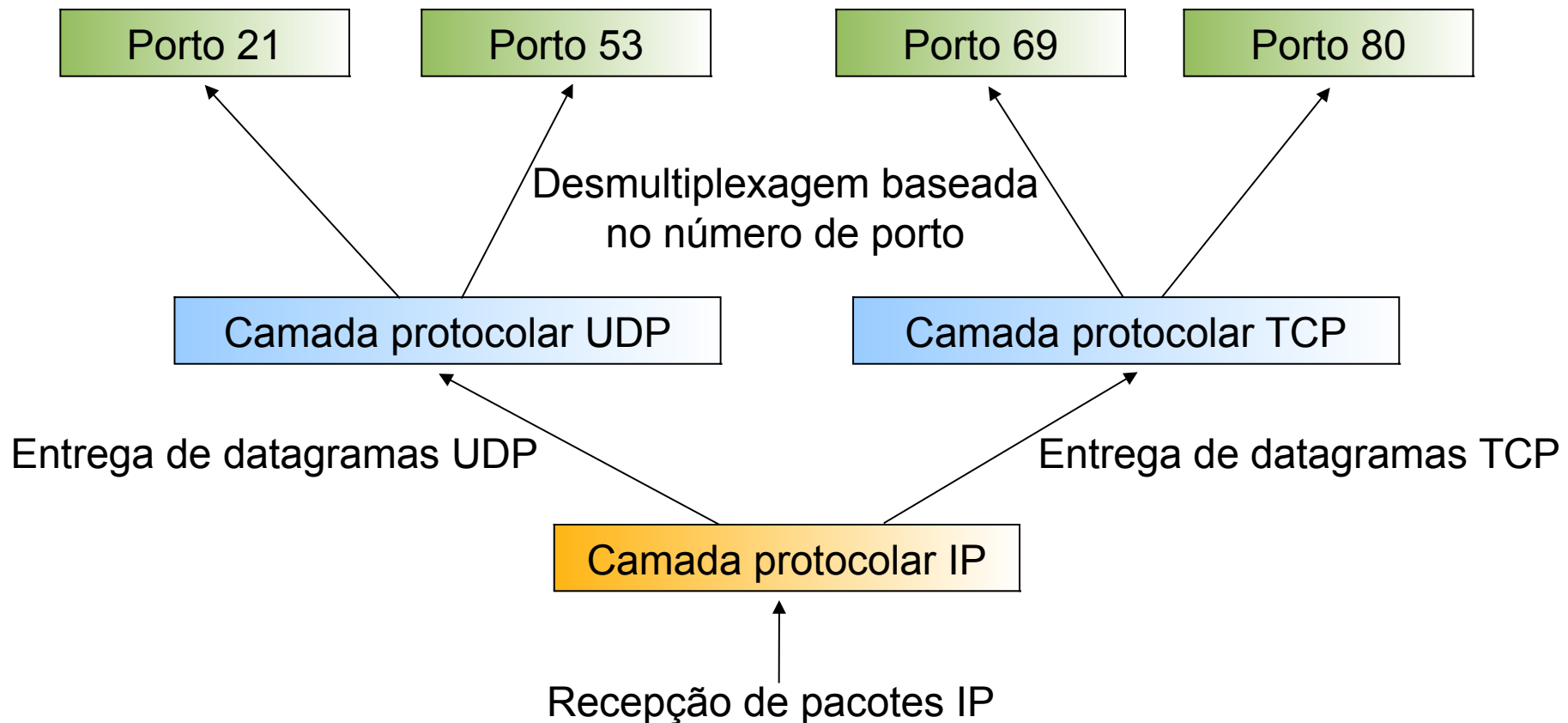
TCP/IP



OSI



Noção de Porto



- Um endereço IP identifica uma ligação de uma estação a uma rede IP
- O número de porto identifica uma aplicação em execução numa estação
- O sistema operativo assegura a atribuição de números diferentes a cada aplicação



Números de porto atribuídos

Decimal	Palavra Chave	Protocolo	Descrição
20	FTP-DATA	TCP	File Transfer Protocol (dados)
21	FTP-CONTROL	TCP	File Transfer Protocol (controlo)
23	TELNET	TCP	Terminal Connection
25	SMTP	TCP	Simple Mail Transport Protocol
67,68	BOOTP	UDP	Bootstrap Protocol
53	DNS	UDP/TCP	Domain Name System
69	TFTP	UDP	Trivial File Transfer Protocol
80	HTTP	TCP	Hypertext Transfer Protocol

- Muitos dos serviços existentes nas redes IP, pela sua utilidade, foram objecto de estudo pelo IETF que propôs protocolos de suporte adequados.
- Para esses serviços, o IETF fez acompanhar a especificação da proposta com a atribuição de um número de porto para o lado do servidor.
- Por exemplo, para o protocolo HTTP, o IETF recomenda a utilização do porto número 80. Desta forma, os Web browsers consideram por omissão este número de porto.

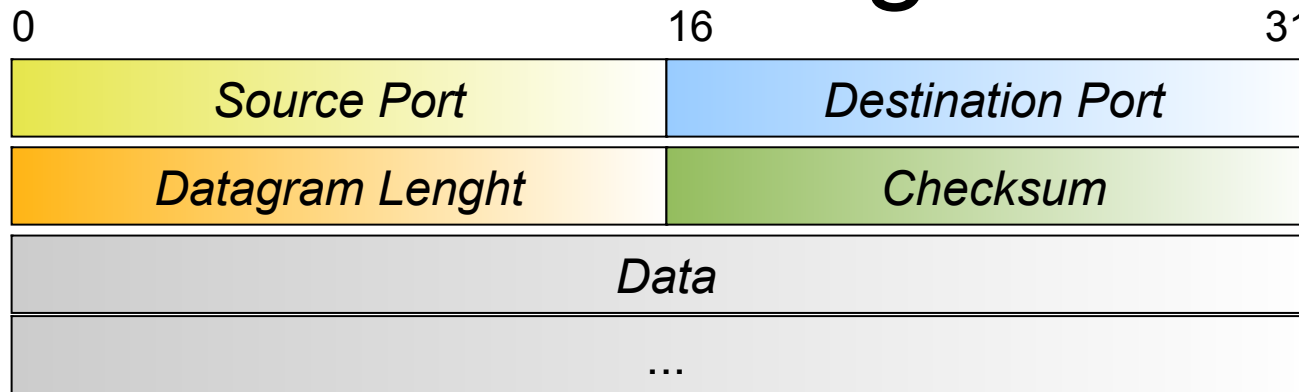


User Datagram Protocol (UDP)

- Proporciona um serviço de transporte de dados com as características de desempenho oferecidas pela rede IP
- Permite a troca de dados entre aplicações, e não apenas entre estações, através de introdução no seu cabeçalho de um campo identificador do porto
- Permite o envio de dados para múltiplos destinos (comunicações ponto-multiponto)



Formato do Datagrama UDP

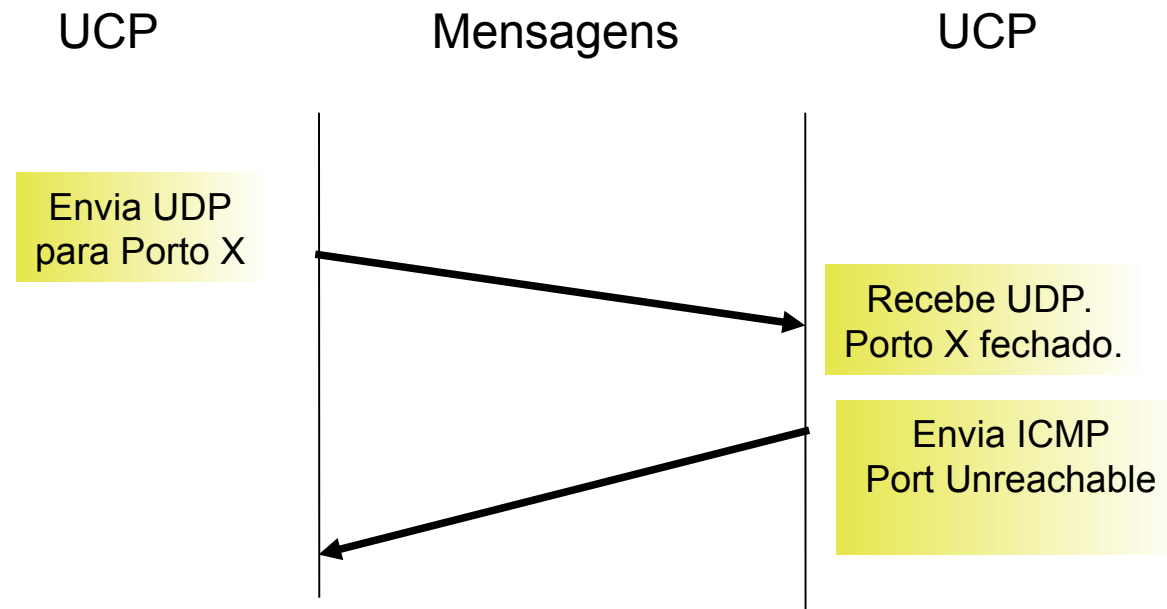


- Source Port (dois octetos): o número de porto da aplicação origem (este campo é opcional e se não for usado, deve ser preenchido com zeros)
- Destination Port (dois octetos): o número de porto da aplicação destino
- Datagram Lenght (dois octetos): o número de octetos do datagrama (cabeçalho + dados)
- Checksum (dois octetos): deteção de erros e validação dos extremos (este campo é opcional e se não for usado, deve ser preenchido com zeros)
 - ♦ o checksum é calculado com base no datagrama completo UDP mais o pseudoheader IP (identificador de protocolo IP, endereço IP origem, endereço IP destino e comprimento do datagrama IP)
 - ♦ a motivação para inclusão do pseudoheader é verificar se a mensagem foi enviada entre os extremos corretos



Porto UDP Fechado

- Quando um pacote UDP chega a um terminal de destino mas o porto UDP de destino não está aberto:
 - ♦ O terminal responde com um pacote ICMP *port unreachable*.

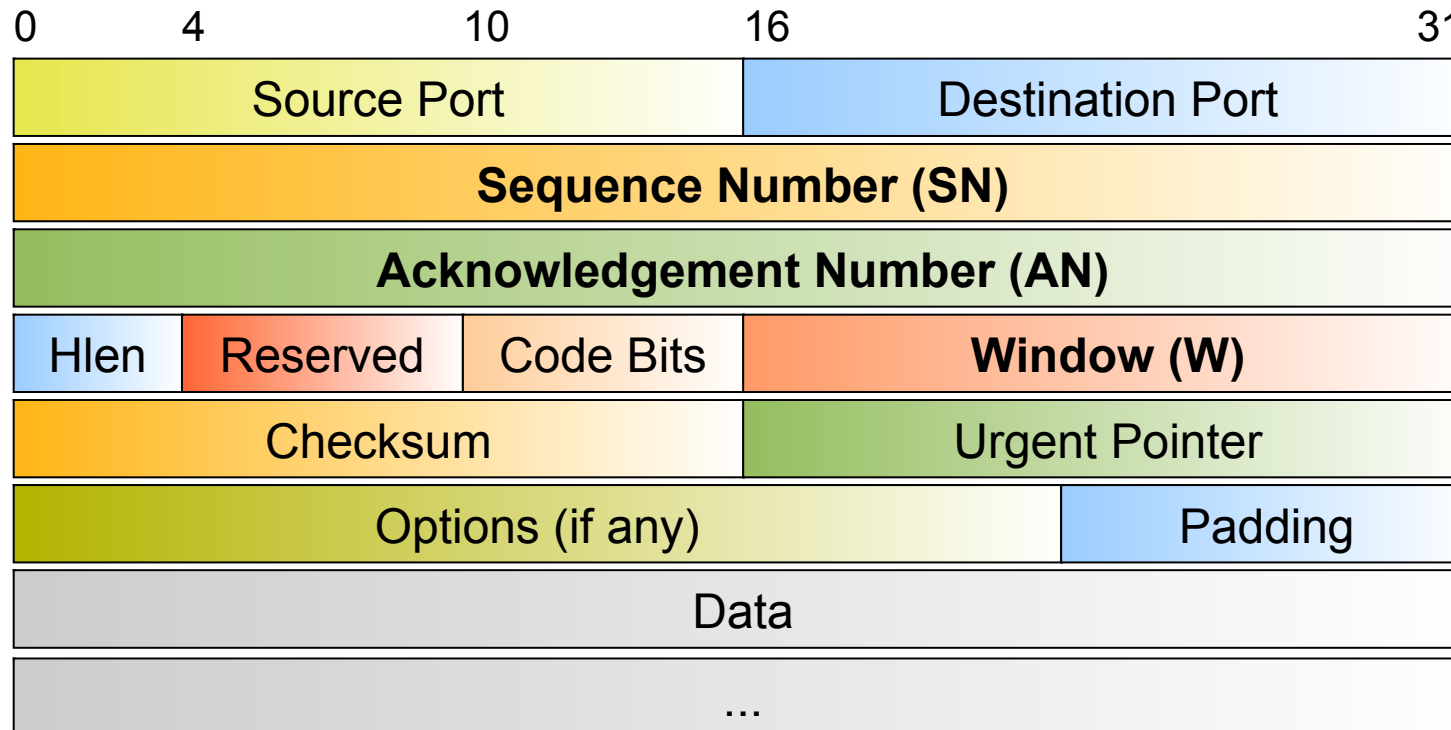


Transmission Control Protocol (TCP)

- Proporciona um serviço de transporte de dados fiável
 - os dados são recebidos pela aplicação destino sem falhas e pela ordem enviada
- É orientado à ligação
 - as estações estabelecem um canal lógico ao qual são atribuídos os identificadores das aplicações, bem como os recursos de memória necessários para que os dados sejam transmitidos de uma forma fiável
- É bi-direccional
 - São estabelecidos dois canais lógicos independentes, um em cada sentido
- Suporta apenas ligações ponto-a-ponto
- Faz uso da noção de fluxo de informação
 - o emissor secciona os dados de uma forma independente dos blocos de informação que lhe são entregues para envio pela aplicação origem
- Proporciona o estabelecimento e a terminação da ligação transparente
 - uma aplicação pede para estabelecer uma ligação: se o TCP responder à aplicação com uma mensagem de sucesso, significa que existe conectividade com a estação destino e que a aplicação destino está preparada para comunicar
 - uma aplicação pede para terminar uma ligação: o TCP assegura que se houver informação ainda por enviar, ela é transmitida antes de sinalizar a terminação da ligação



Formato de um segmento TCP



Code Bits/Flag

URG

ACK

PSH

RST

SYN

FIN

Significado quando estão a 1

Campo *Urgent Pointer* válido

Campo Acknowledgement válido

Os dados requerem um *Push*

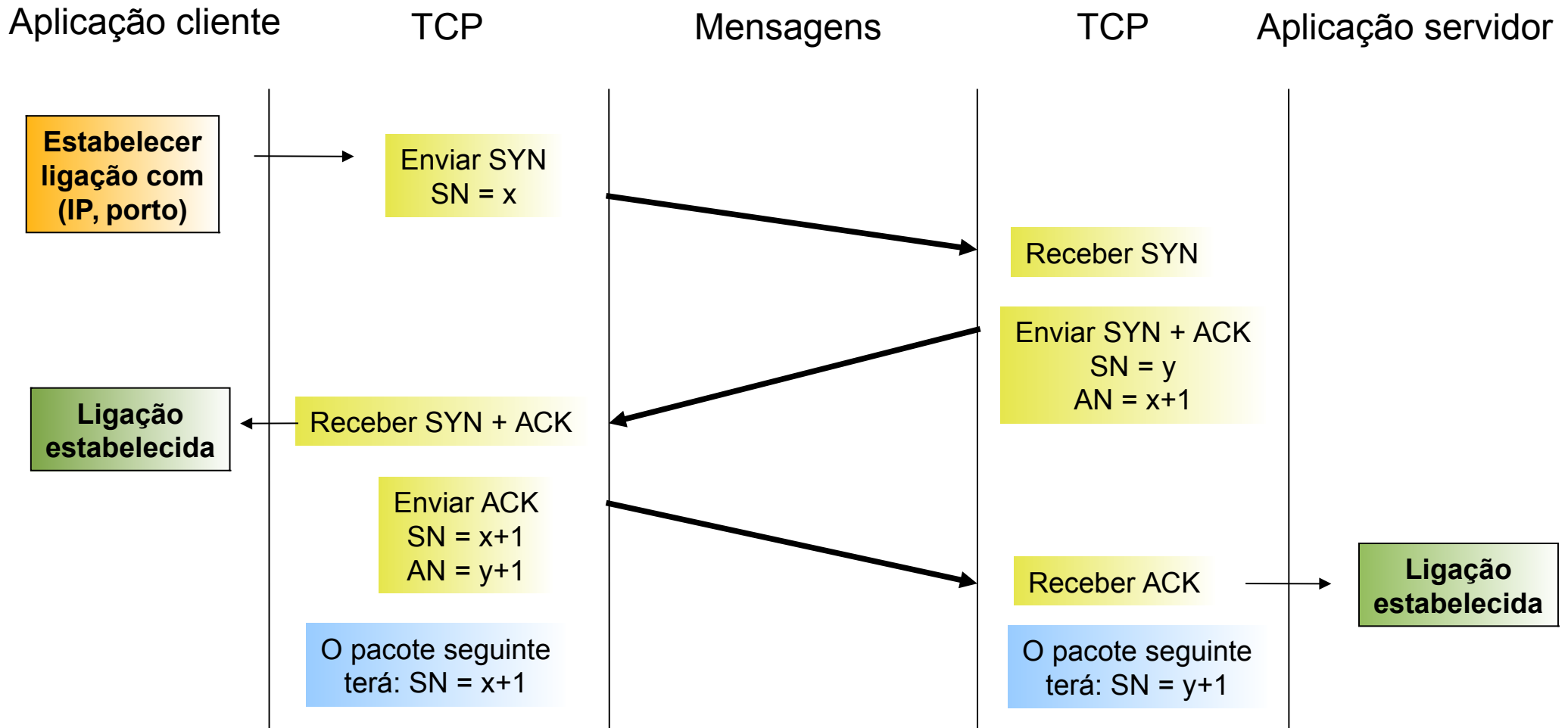
Fazer *Reset* à ligação

Sincronizar Sequence Number

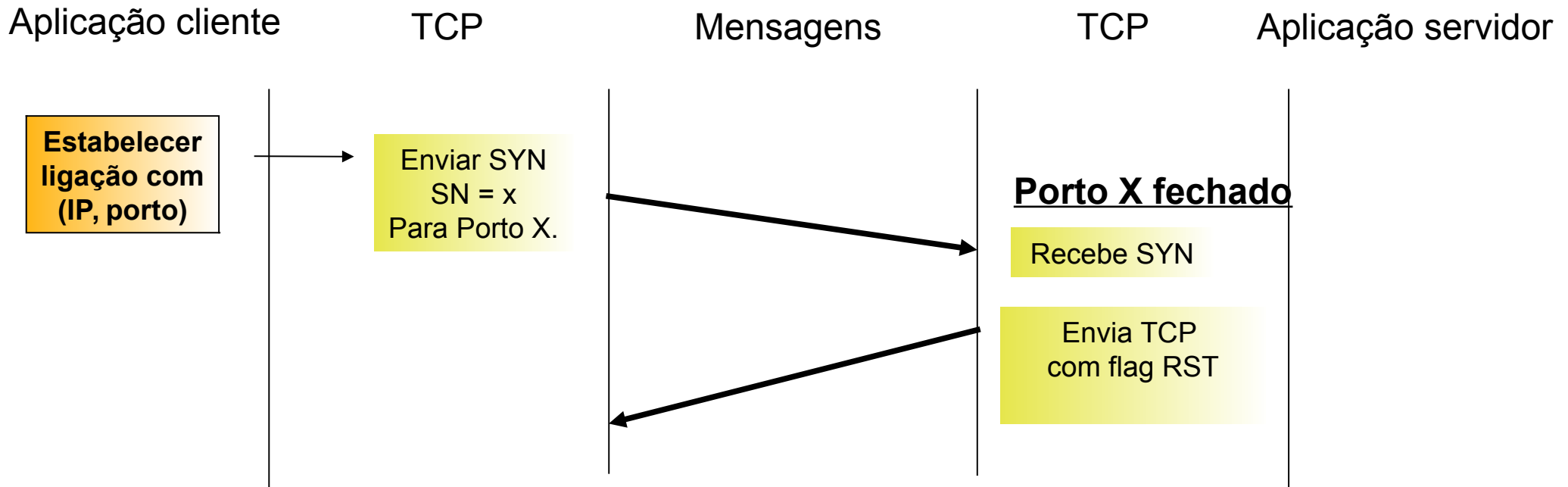
Origem terminou o envio de informação



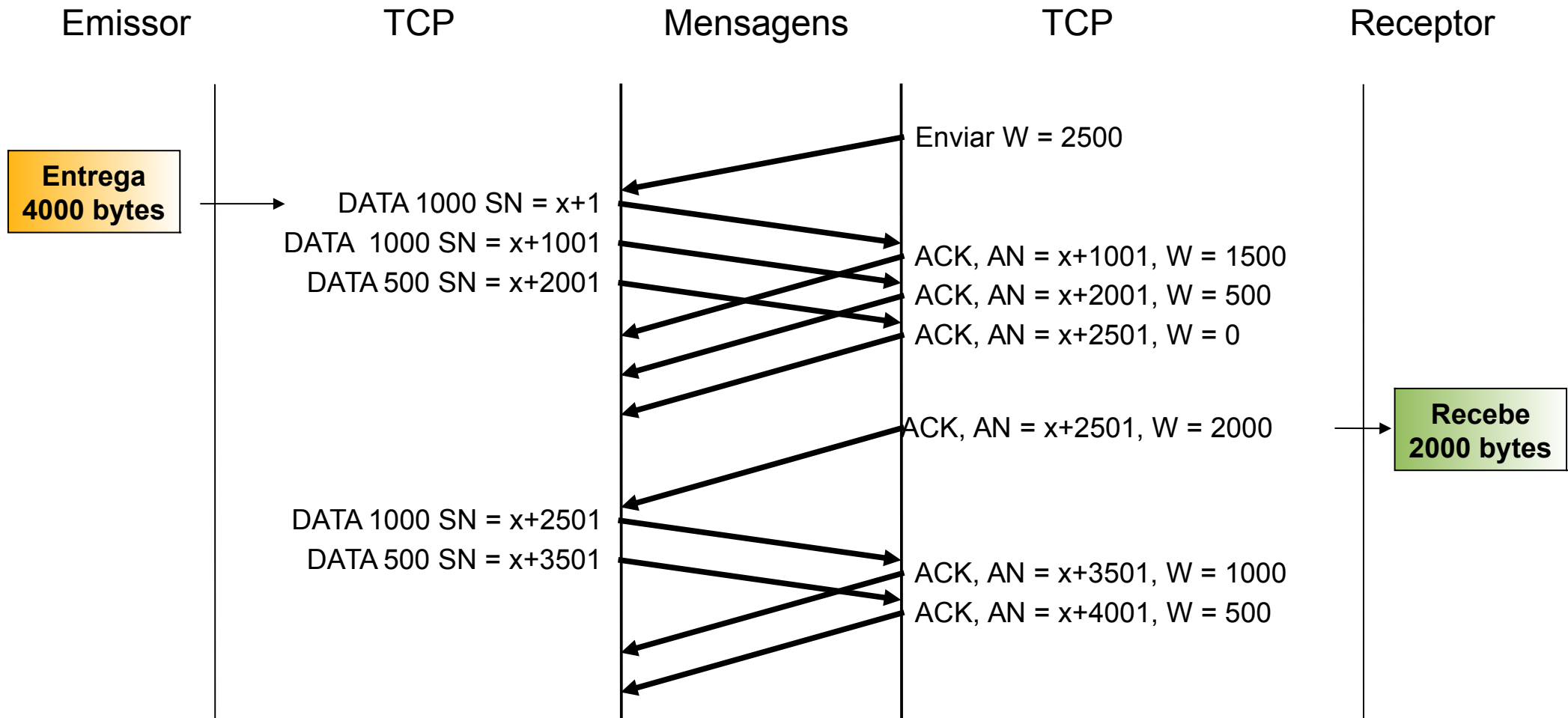
Estabelecimento de ligação em TCP



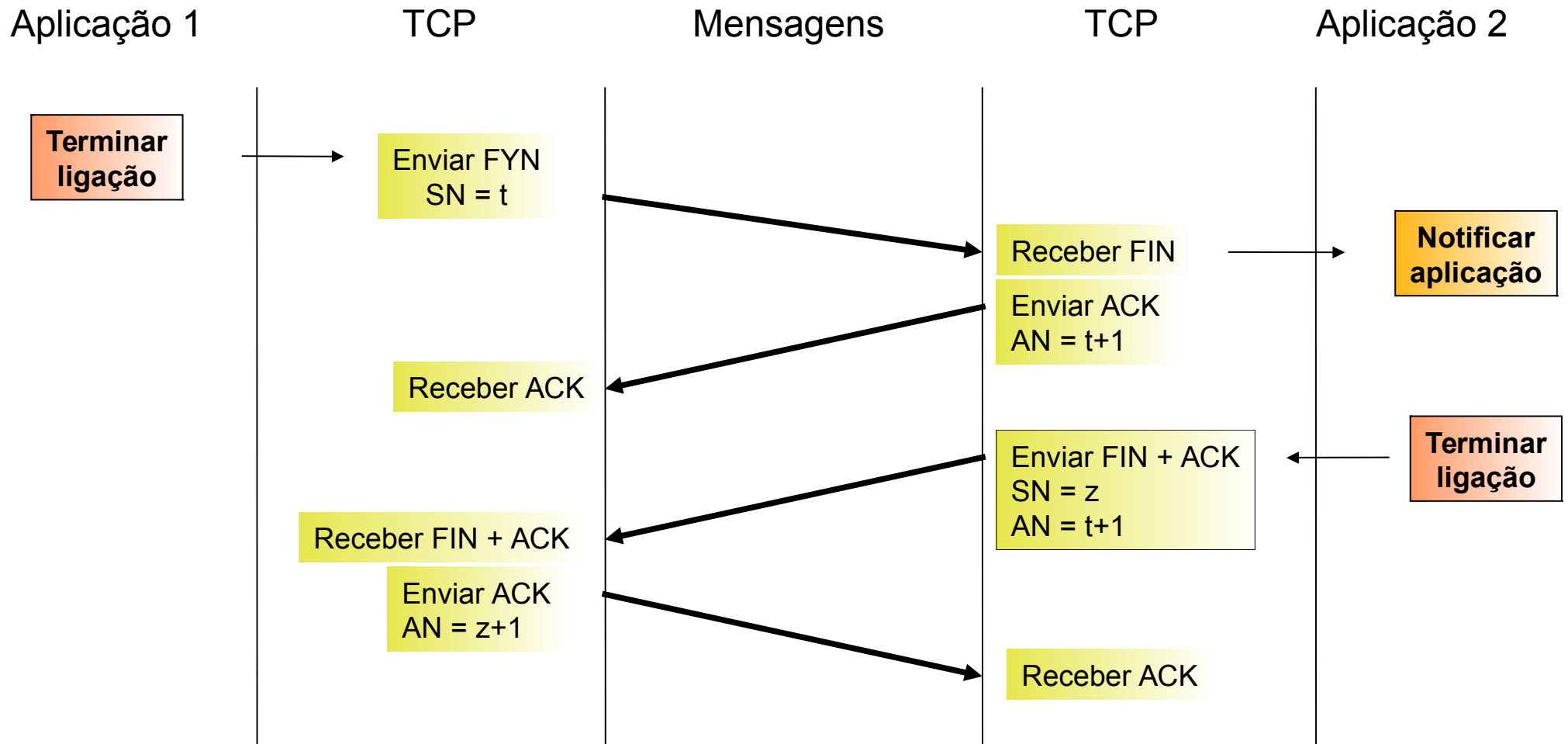
Tentativa de estabelecimento de uma sessão para um porto fechado



Controlo de fluxo (TCP)



Terminação de ligação em TCP



Campos do cabeçalho TCP

- O cabeçalho TCP é de tamanho variável porque podem ser incluídos alguns campos no campo *Options*
- O campo *Hlen* indica o tamanho do cabeçalho em múltiplos de 4 octetos (quando há opções, o campo *Padding* acrescenta octetos para que o cabeçalho tenha um tamanho múltiplo de 4 octetos)
- O *Sequence Number* sinaliza os dados já enviados
- O *Acknowledge Number* sinaliza os dados já recebidos
- O *Window* permite ao receptor informar o emissor de quantos octetos ele está preparado para receber
- O *Sequence Number* diz respeito ao controlo de fluxo no sentido da transmissão dos dados, enquanto que o *Acknowledge Number* e o *Window* dizem respeito ao controlo de fluxo no sentido contrário



Exemplo

1. Considere uma ligação TCP estabelecida da estação A para a estação B:
 - Em ambas as estações, o TCP: (i) considera um *buffer* de recepção de tamanho fixo de 2000 bytes e (ii) segmenta a informação em pacotes com no máximo 1000 bytes de dados.
 - No estabelecimento da ligação, a estação A escolhe um *Sequence Number* inicial de 1515 e a estação B escolhe um *Sequence Number* inicial de 502.
 - Durante o tempo de vida da ligação: a estação A entrega um bloco de 530 bytes de dados para envio e a estação B não envia dados.

Desenhe o diagrama temporal de todos os segmentos TCP trocados incluindo o estabelecimento e a terminação da ligação. Para cada segmento, indique justificando o *Sequence Number* (SN), o *Acknowledgement Number* (AN), a Janela de Recepção (W) e as *flags* activas.



Controlo de congestão no TCP

- Situações de congestão e/ou falha temporária de conectividade da rede IP provocam atrasos variáveis e perdas de pacotes.
- O TCP inclui algoritmos que permitem reagir de uma forma eficiente a estas situações.
- Os algoritmos podem ser genericamente classificados em dois tipos:
 - Gestão do tempo de retransmissão de pacotes (*timeout*):
 - O emissor estima o RTT (*Round Trip Time*) com base nos ACK recebidos anteriormente
 - Considera um *timeout* com base no RTT estimado cujo valor é $\beta \times \text{RTT}$
 - O IETF RFC 793 propõe $1.3 \leq \beta \leq 2.0$
 - Gestão da janela de transmissão de pacotes
 - Próximos slides



Gestão da janela de transmissão do TCP

- Em qualquer instante, a transmissão de segmentos TCP é restringida por:

$$awnd = \min(credit, cwnd)$$

awnd – (allowed window) janela permitida, em segmentos.

credit – crédito não utilizado que foi autorizado pelo ACK mais recente, em segmentos.

cwnd – (congestion window) janela de congestionamento, em segmentos; janela usada

pelo TCP durante o start-up e em períodos de congestionamento.

- Procedimento Slow Start de uma nova ligação TCP:
 - é feito $cwnd = MSS$ (Maximum Segment Size);
 - para que o atraso até ser atingida uma taxa de transmissão razoável não seja muito elevado, o TCP aumenta a janela mais rapidamente nesta fase
 - cada vez que um ACK é recebido o valor de *cwnd* é incrementado de um segmento de tamanho máximo (MSS), até um valor máximo;
 - neste procedimento, *cwnd* cresce exponencialmente.



Gestão da Janela de Transmissão do TCP

- O procedimento de slow start é demasiado agressivo em situações de recuperação de timeout.
- Regras para gestão de janelas quando ocorre um timeout (algoritmo de Jacobson):
 - Colocar o limiar de slow start *ssthresh* (slow start threshold) igual a metade da janela actual:

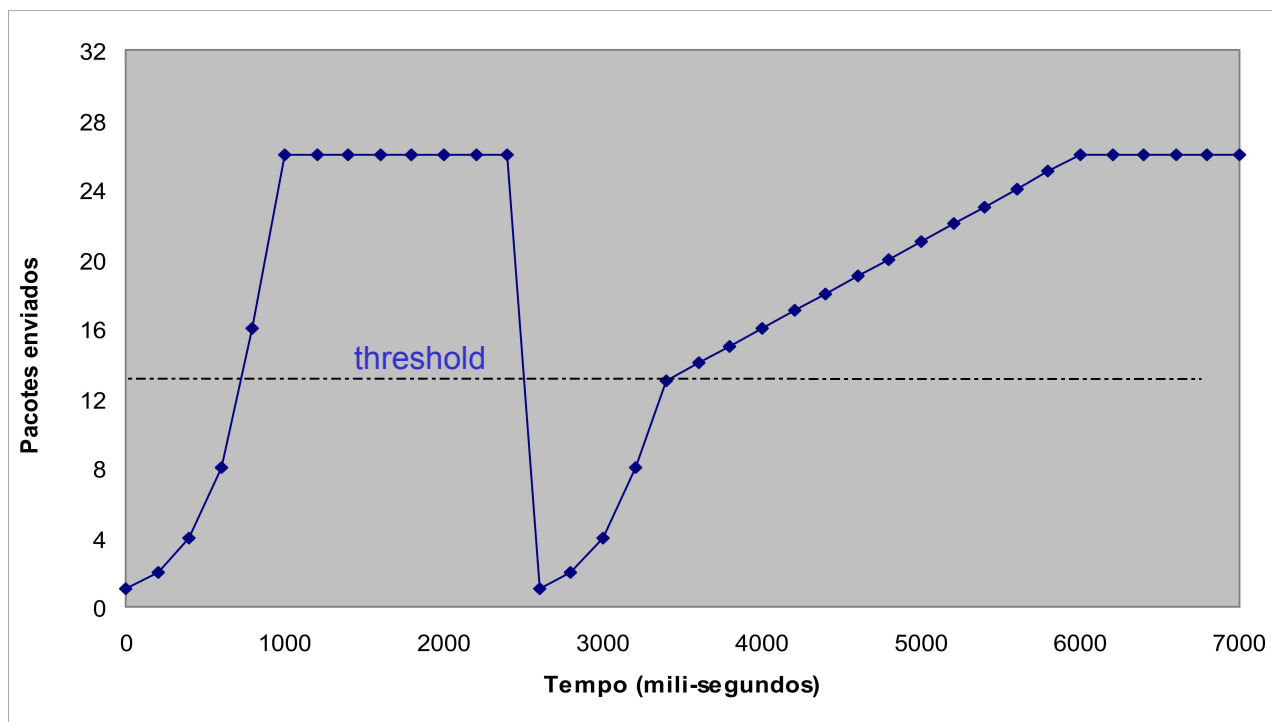
$$ssthresh = awnd/2$$

- Fazer $cwnd = MSS$
- Efectuar o procedimento slow start até que $cwnd = ssthresh$. Nesta fase, $awnd$ é incrementada de MSS de cada vez que é recebido um ACK.
- Para $cwnd \geq ssthresh$, incrementar $cwnd$ linearmente, isto é, incrementar $cwnd$ de MSS em cada RTT.



Gestão da janela de transmissão do TCP

- RTT = 200 mili-segundos
- *credit* = 26 segmentos
- Ocorreu um *timeout* no instante 2,6 segundos:
 $ssthresh = awnd/2 = 26/2 = 13$



Gestão da janela de transmissão do TCP

- RTT = 200 mili-segundos
- *credit* = 26 segmentos
- Ocorreu um *timeout* no instante 7,2 segundos:
 $ssthresh = awnd/2 = 26/2 = 13$
- Ocorreu um *timeout* no instante 8,8 segundos:
 $ssthresh = awnd/2 = 16/2 = 8$

