Fundamentos de Redes

Protocolos de Transporte

Djamel Sadok
Dênio Mariz

{jamel,dmts}@cin.ufpe.br

Cin/UFPE, JUN/2003

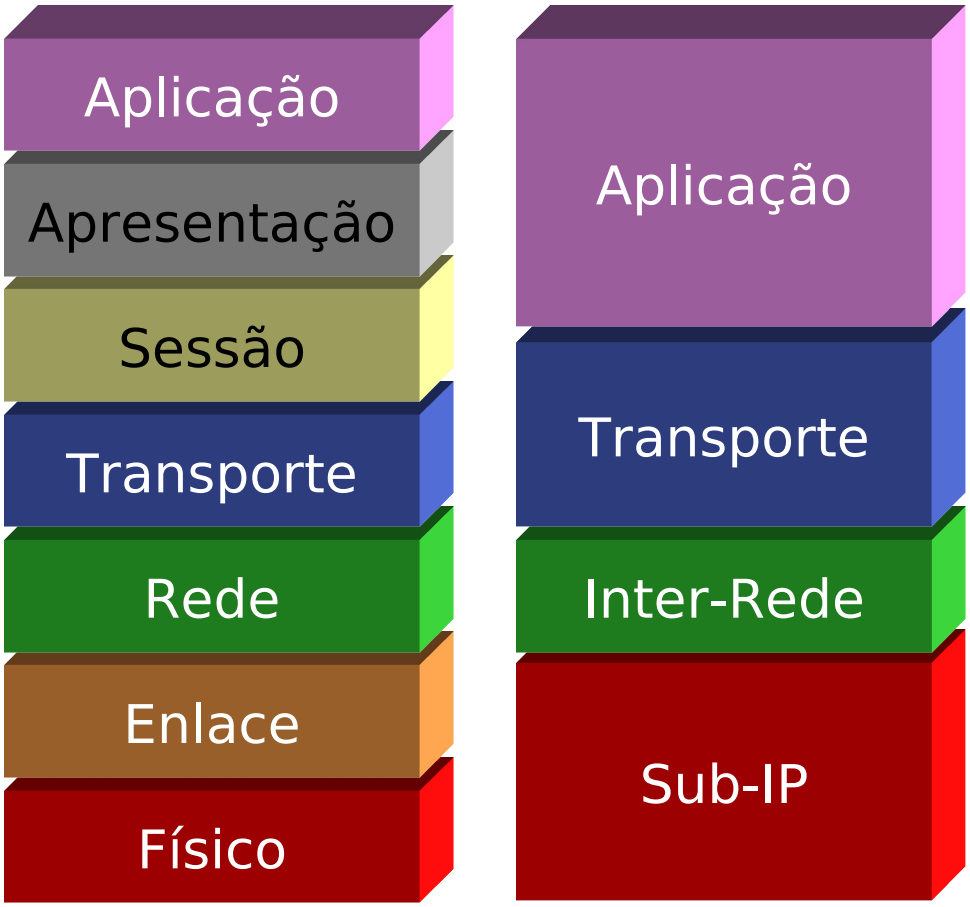
Internet e TCP/IP

- Internet
  - Agrupamento de grande quantidade de redes ao redor do mundo, ligadas pelo protocolo IP
  - Rede mundial de computadores

- TCP/IP
  - Família de protocolos de comunicação
  - Serviços e acesso independente de tecnologia
  - Permite a interconexão de redes físicas diferentes
  - Interconexão realizada por roteadores
- Protocolo IP
  - Não orientado a conexão, roteamento melhor esforço
  - Não confiável, sem controle de fluxo ou de erros → simples
  - Roteamento baseado no endereço da rede de destino

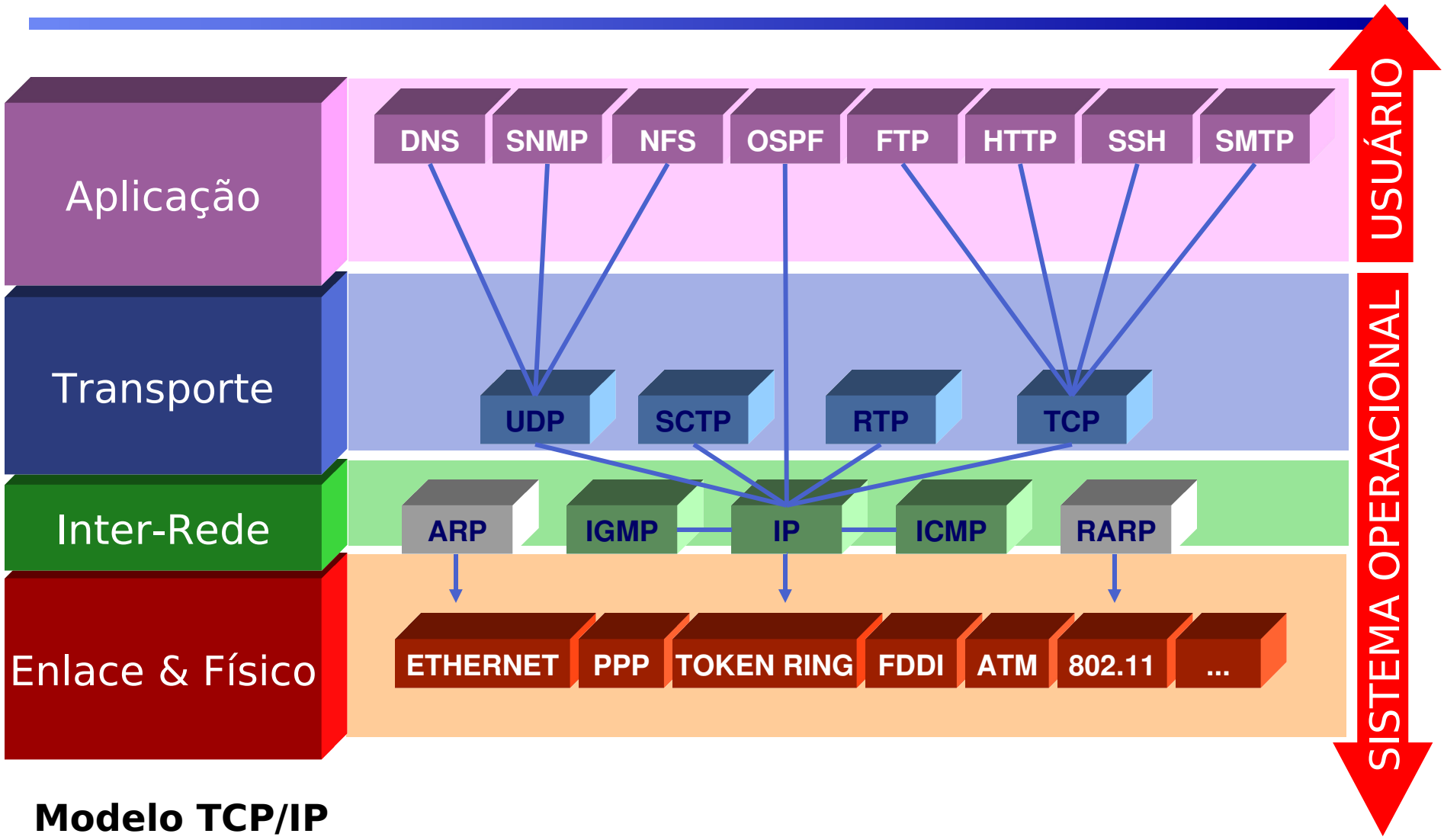
Protocolos, Serviços & Portas

Modelo de camadas OSI & TCP/IP



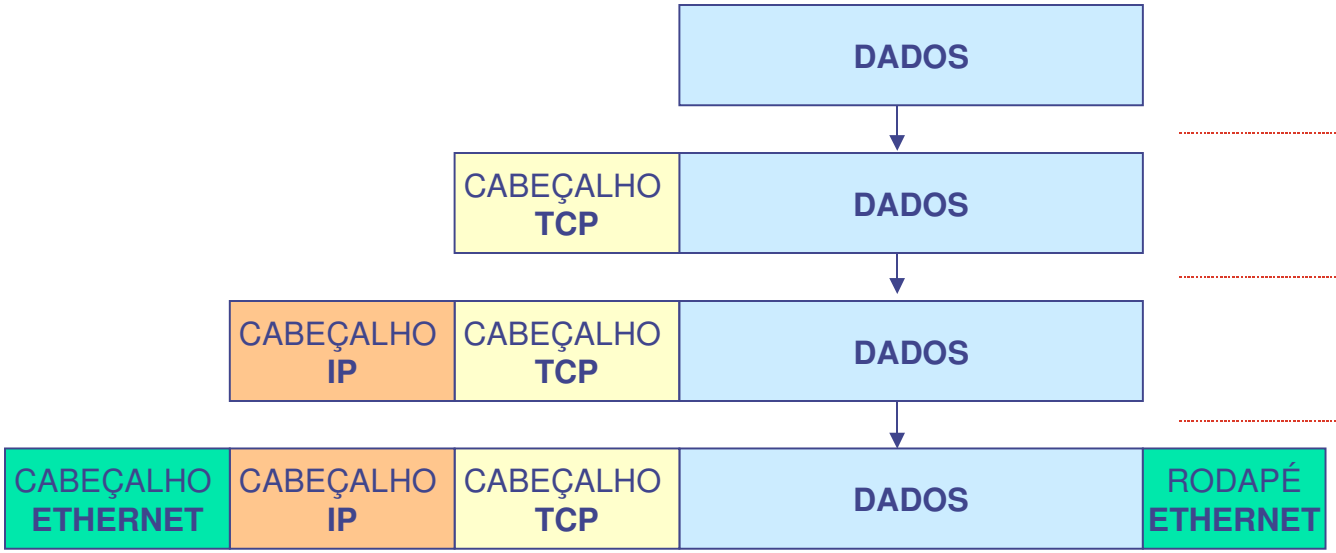
Modelo OSI **Modelo TCP/IP**

Família TCP/IP



Modelo TCP/IP

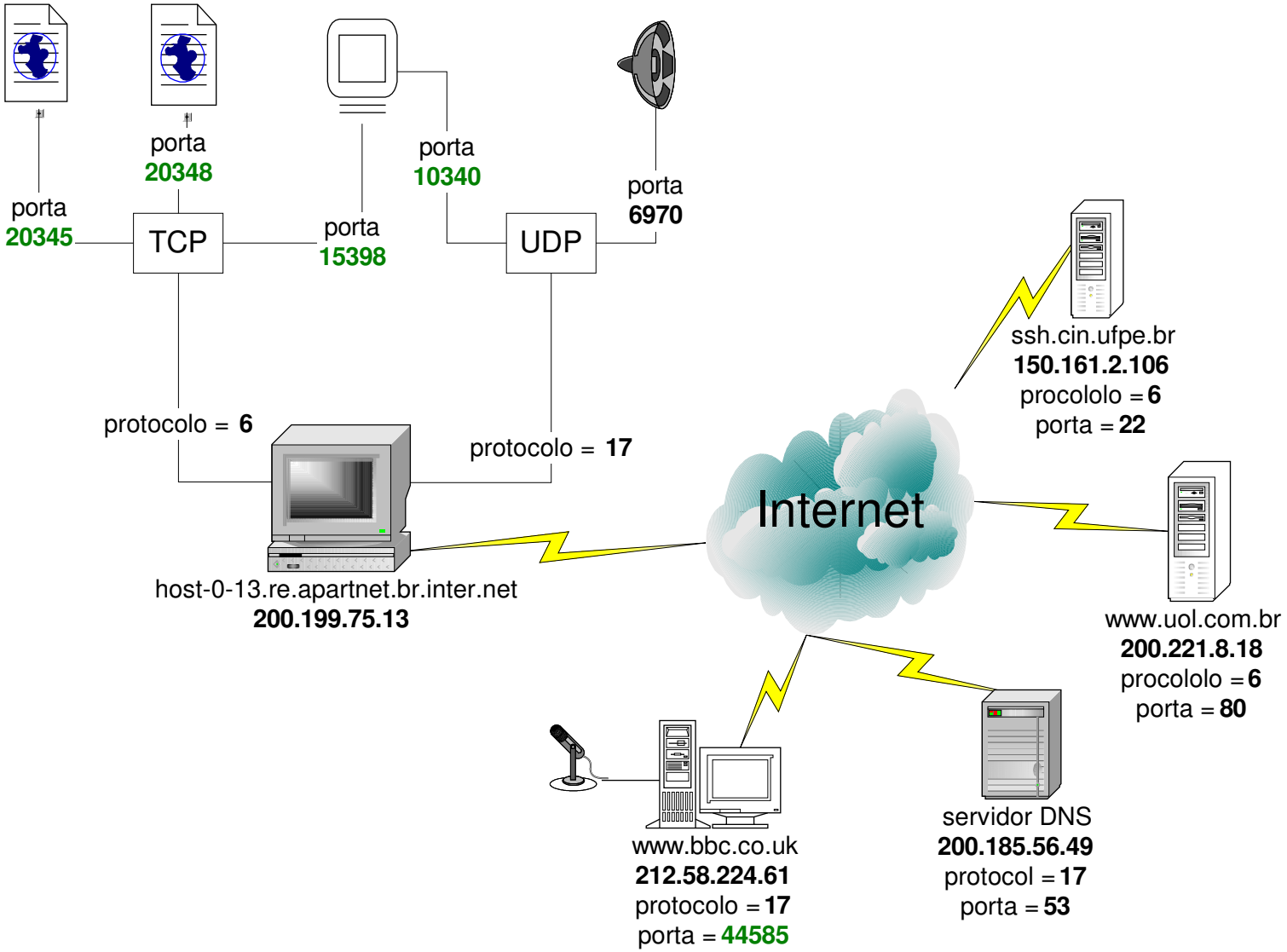
Encapsulamento



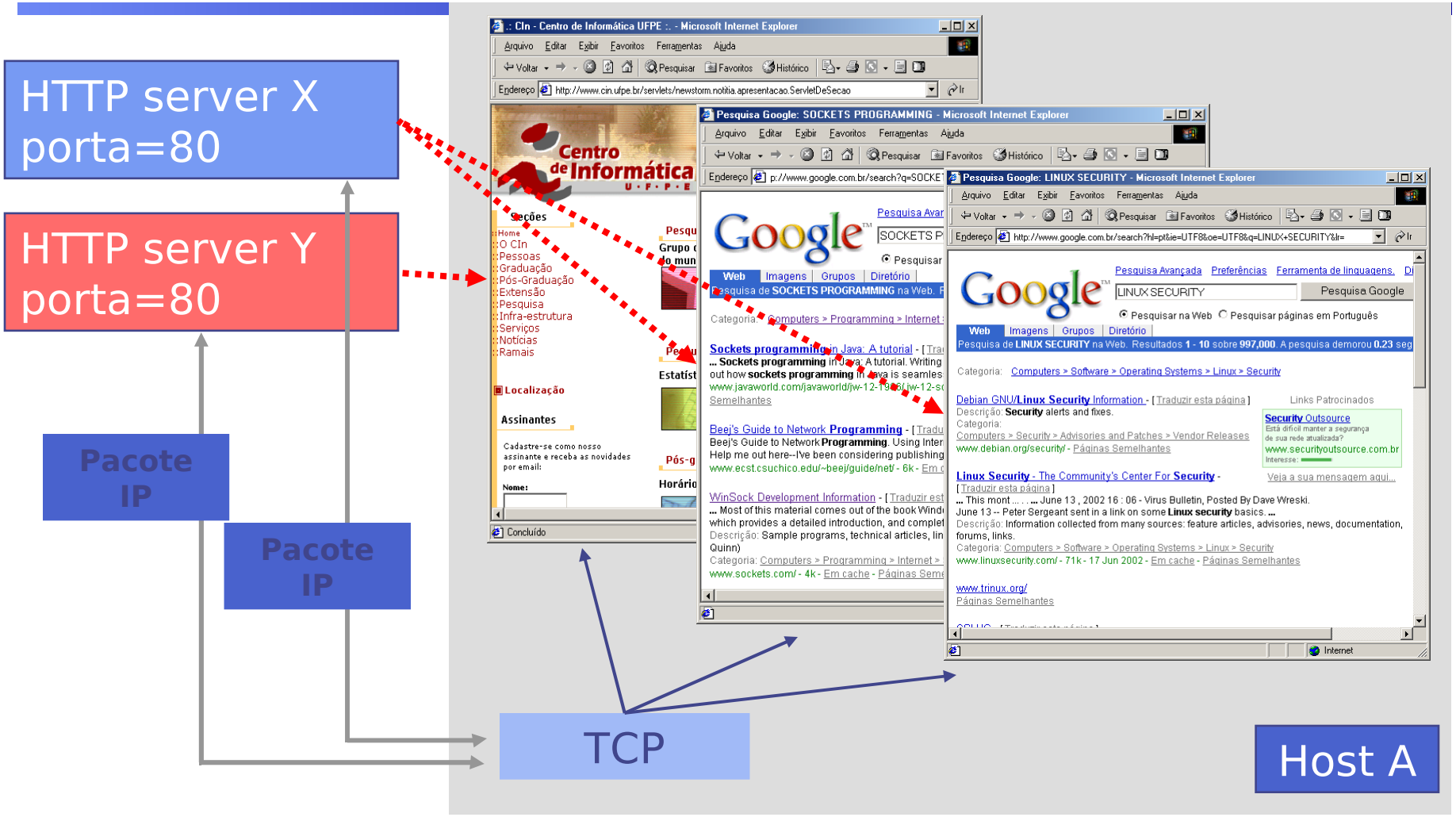
Identificação da aplicação no host

- Como cada máquina é identificada unicamente na Internet?
- Como a entidade de rede (IP) identifica qual o protocolo de transporte está sendo utilizado ?
- Dentro do host, como a entidade de transporte (TCP,UDP) sabe para qual aplicação entregar o pacote ?
- Como uma aplicação do cliente sabe qual é a aplicação dentro do servidor remoto para poder enviar pacotes?

Identificação da aplicação no host

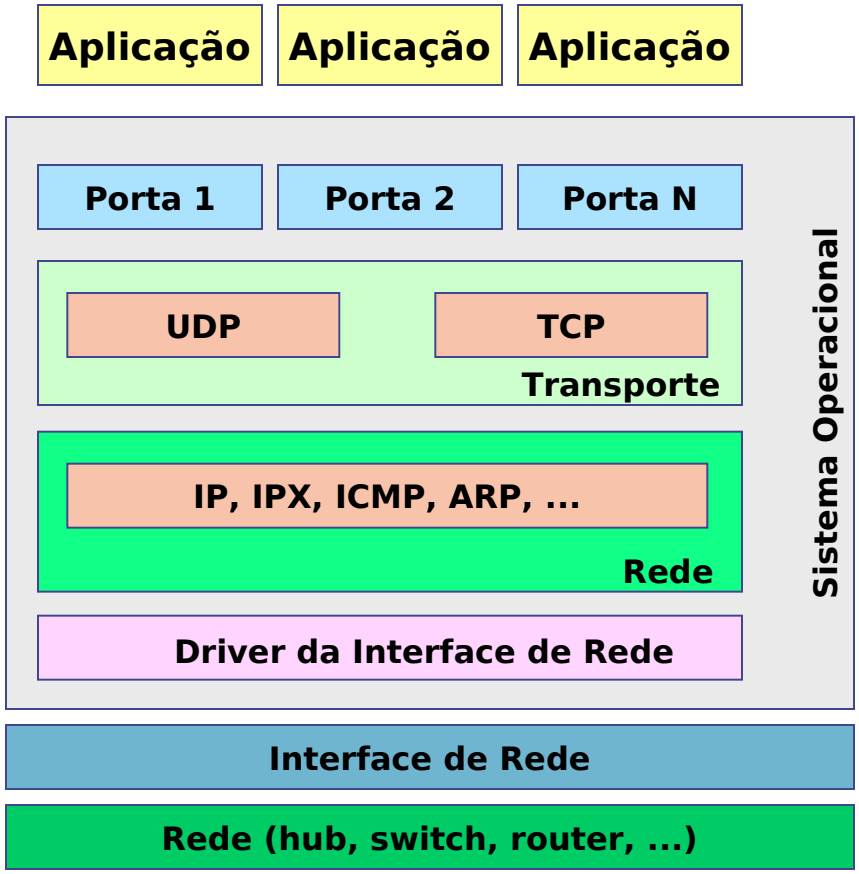


Identificação da aplicação no host

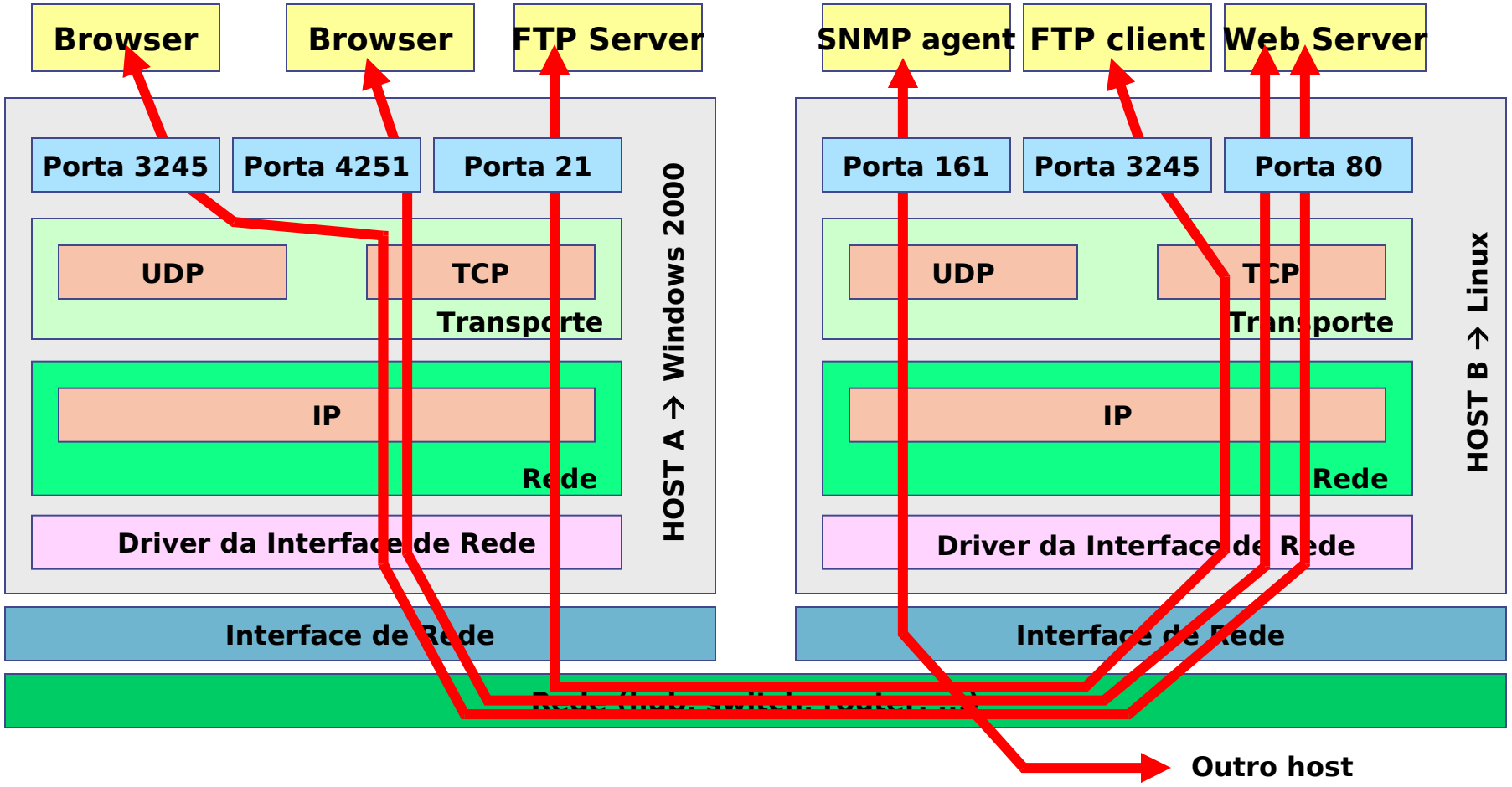


As aplicações não confundem os pacotes que chegam ?

Pilha de Protocolos, na prática



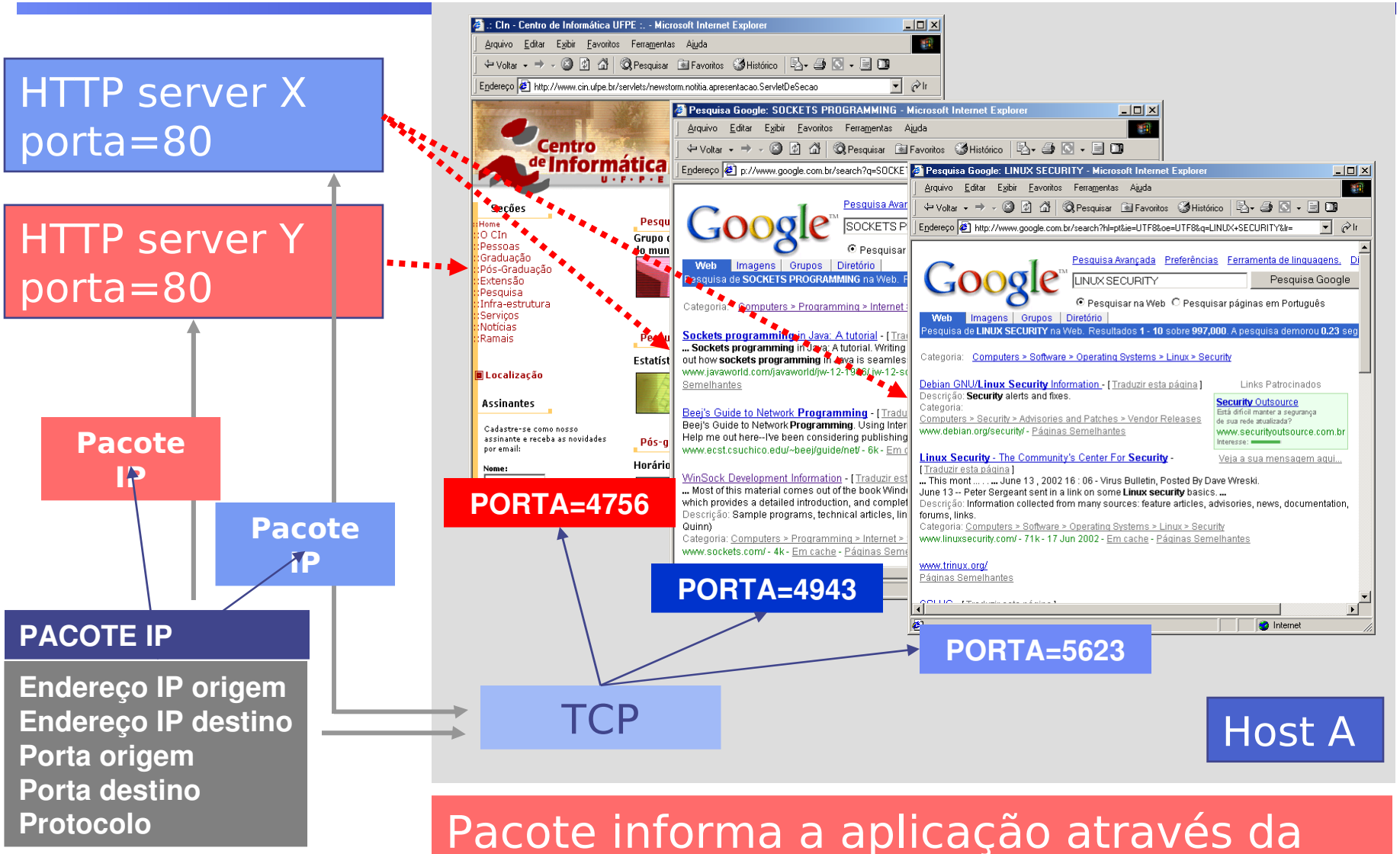
Pilha de Protocolos, na prática



O TCP/IP sabe para qual aplicação entregar o pacote olhando a TUPLA:

Endereço IP origem, Endereço IP destino, Porta origem, Porta destino, Protocolo

Identificação da aplicação no host



Pacote informa a aplicação através da PORTA

Identificação da aplicação no host

- Como cada máquina é identificada unicamente na Internet?
  - Número IP
  - IPv4 → 2^32 = 4.294.967.296 hosts
  - IPv6 → 2^128 = 340.282.366.920.938.463.463.374.607.431.768.211.456 hosts
- Como a entidade de rede (IP) identifica qual o protocolo de transporte está sendo utilizado ?
  - Tipo de protocolo está indicado no cabeçalho IP
- Dentro do host, como a entidade de transporte (TCP,UDP) sabe para qual aplicação entregar o pacote ?
  - Cada aplicação tem uma “Porta” única no host
  - Porta é identificada no pacote IP
- Como uma aplicação do cliente sabe qual é a aplicação dentro do servidor remoto para poder enviar pacotes?
  - Alguns serviços têm números de portas já convencionadas (portas “bem conhecidas”)

Identificação de aplicações

- Como cada máquina é identificada unicamente na Internet ?
  - Número IP
- Como a entidade de rede (IP) identifica qual o protocolo de transporte está sendo utilizado ?
  - Tipo de protocolo está indicado no cabeçalho IP
- Dentro do host, como a entidade de transporte identifica qual aplicação está sendo utilizada ?
  - Cada aplicação tem uma “Porta” única no host
  - Porta é identificada no pacote IP
- Como uma aplicação cliente sabe qual a porta de uma aplicação servidora para poder enviar pacotes?
  - Alguns serviços têm números de portas já convencionadas (portas “bem conhecidas”)

Números de portas

- 1-255 — reservadas para serviços padrão portas “bem conhecidas”
- 256-1023 — reservado para serviços Unix
- 1-1023 — Somente podem ser usadas por usuários privilegiados (super-usuário)
- 1024-4999 — Usadas por processos de sistema e de usuário
- 5000- — Usadas somente por processos de usuário

Algumas Portas Bem Conhecidas

- 21 → FTP
- 22 → SSH
- 23 → Telnet
- 25 → SMTP
- 53 → DNS
- 69 → TFTP
- 79 → Finger
- 80 → HTTP
- 88 → KERBEROS
- 110 → POP3
- 135-139 → NetBIOS Services
- 161-162 → SNMP
- 443 → HTTPS (HTTP+SSL)
- 995 → POP3S (POP3+SSL)
- 1433 → MS-SQL Server
- 2049 → NFS
- 3006 → MySQL
- 6000 → X Windows

Detalhes em www.iana.org/assignments/port-numbers

Nível de Transporte

- Usuários de nível de transporte são “processos” de aplicação no host
- Entidade de transporte: pode ser no kernel do SO, um processo separado, biblioteca, NIC, etc.
- Tipos de serviços de transporte: orientados ou não a conexão
- Nível de transporte => usuário querendo controlar a conexão
- Oferece uma interface única para diferentes subredes => portabilidade das aplicações

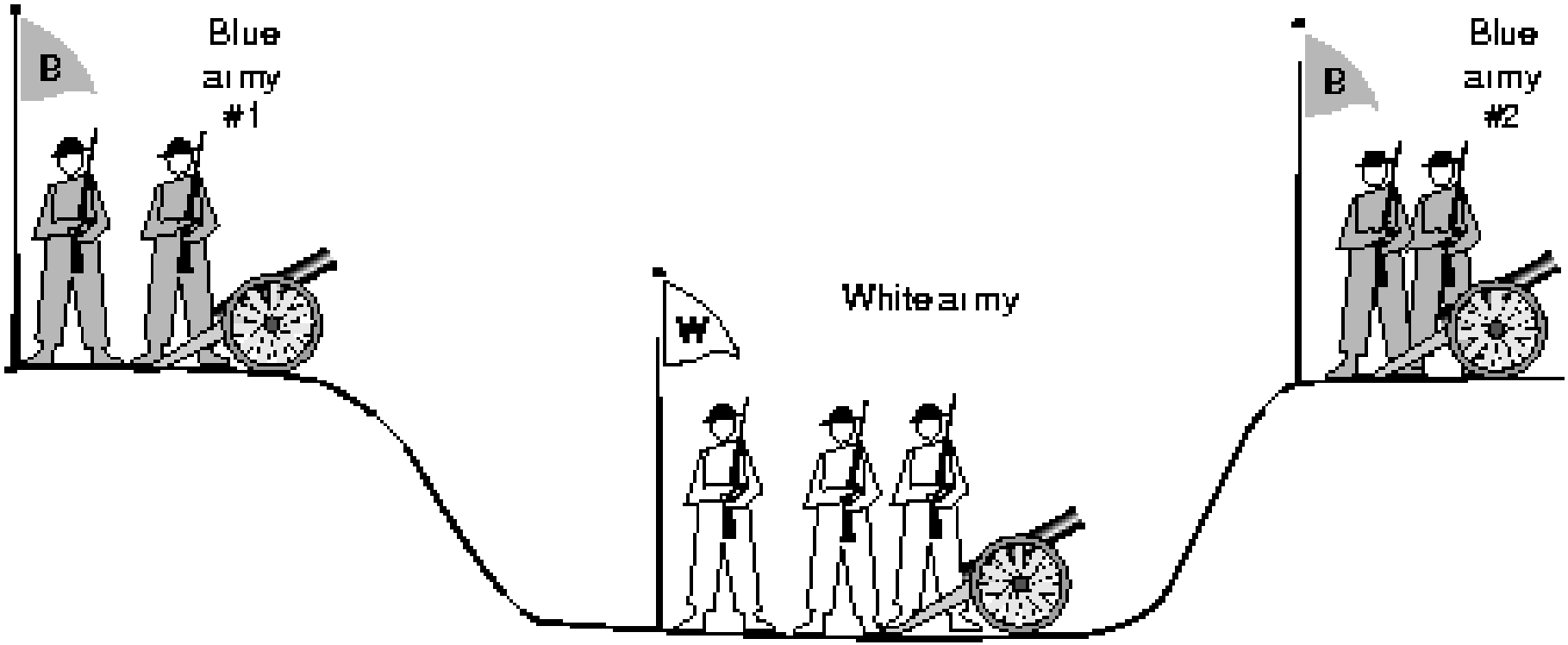
Exemplo de Primitivas de Transporte (Sockets)

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

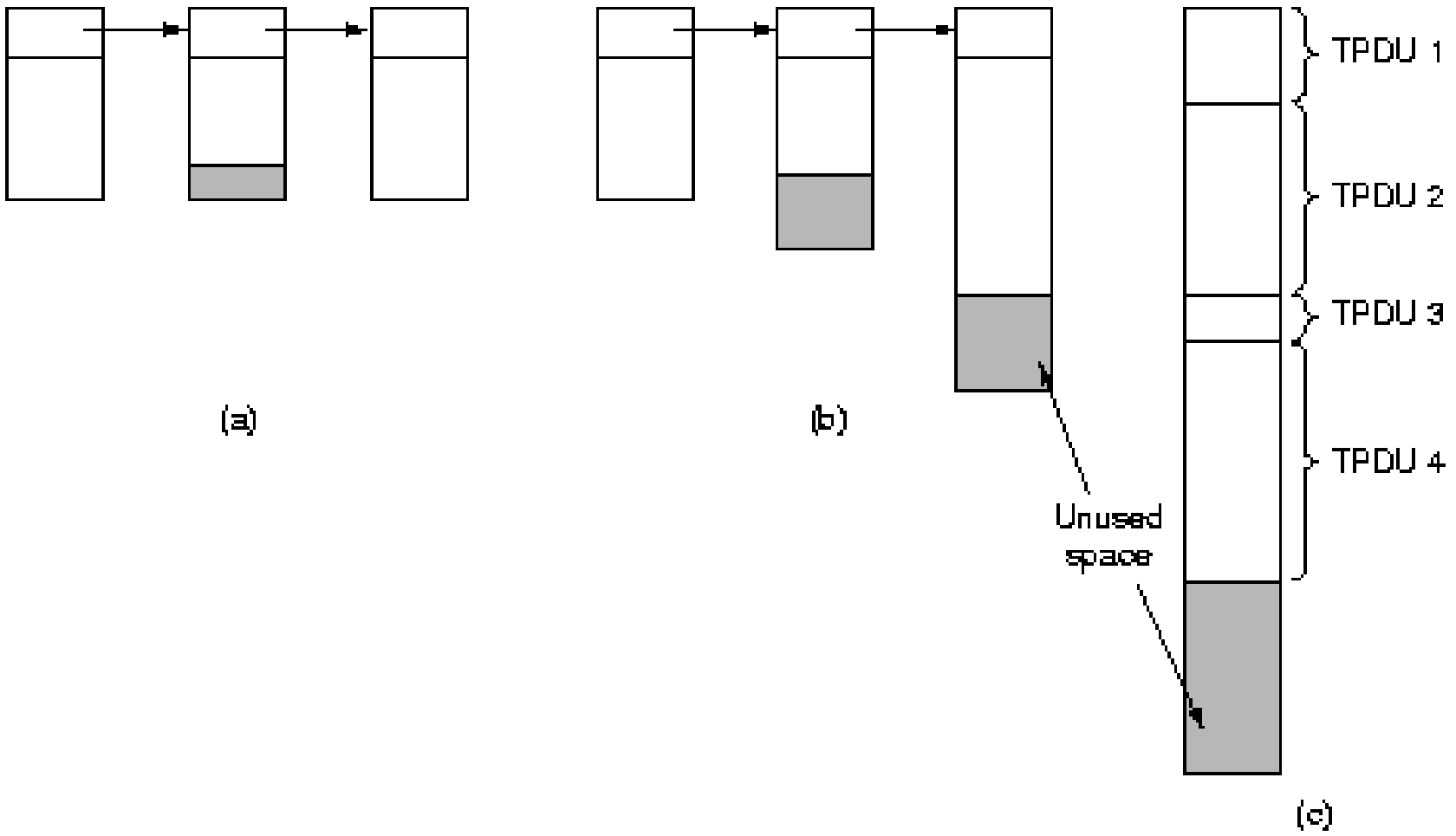
Estabelecimento de Conexões

- Perda e duplicação de pacotes de pedidos de conexões -problema de duplicatas atrasadas
- Mecanismos para retirar da rede pacotes velhos
  - projeto limitado da subrede
  - uso de contadores (hop counter) nos pacotes
  - uso de informação de tempo nos pacotes (timestamps)
- O Three-Way Handshake

Liberação de Conexão sem Perda - Problema dos dois exércitos



Alocação de Buffers - Problema Do



Recuperação de Falhas -> não é transparente!!

Strategy used by sending host	Strategy used by receiving host: First ACK, then write			Strategy used by receiving host: First write, then ACK		
	AC(W)	AWC	C(AW)	C(WA)	W AC	WC(A)
Always retransmit	OK	DUP	OK	OK	DUP	DUP
Never retransmit	LOST	OK	LOST	LOST	OK	OK
Retransmit in S0	OK	DUP	LOST	LOST	DUP	OK
Retransmit in S1	LOST	OK	OK	OK	OK	DUP

Protocolos de Transporte da Internet

Transmission Control Protocol

- orientado a conexão (RFC 793, 1122 e 1323)
- oferece um serviço fim-a-fim confiável sobre subredes com diferentes (velocidades, topologias, atrasos, tamanho de pacotes, etc.)
- executa no host na forma de um processo de usuário ou no kernel
- divisão dos mensagens em pacotes (segmentos) de ate 64KB (normalmente 1500 bytes)

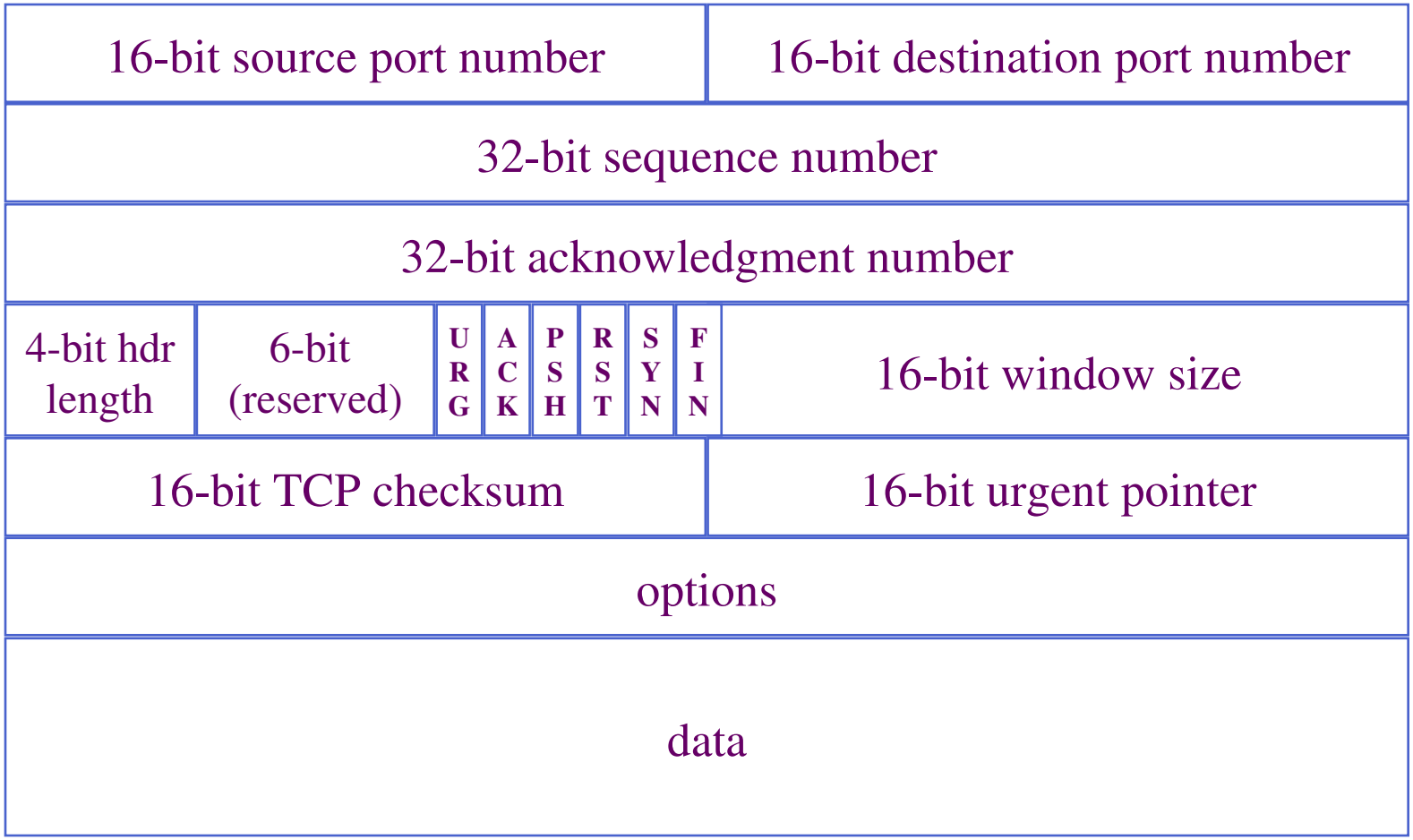
Modelo de Serviço do TCP

- Cada entidade de transporte cria um socket com endereço IP + numero de porta (16 bits)
- Mais de uma conexão pode terminar no mesmo socket
- Identificação de uma conexão através da dupla (socket1, socket2)
- Números de portas ate 256 são conhecidos e reservados (ftp-21, telnet 23)
- Conexões TCP são full duplex e ponto a ponto
- TCP não suporta comunicação muticast ou broadcast
- TCP e baseado em um stream de bytes (não é baseado em mensagens)

Modelo de Serviço do TCP

- Uso de um flag para indicar dados urgentes (a entidade para de acumular dados para mandar o mensagem)
- CRC (checksum) do cabeçalho é baseado no complemento de 1 => resultado deve ser 0
- Segmentos podem ser sem dados (ACKs, mensagens de controle)
- Default payload = 536 bytes + 20 do cabeçalho do segmento = 556 bytes

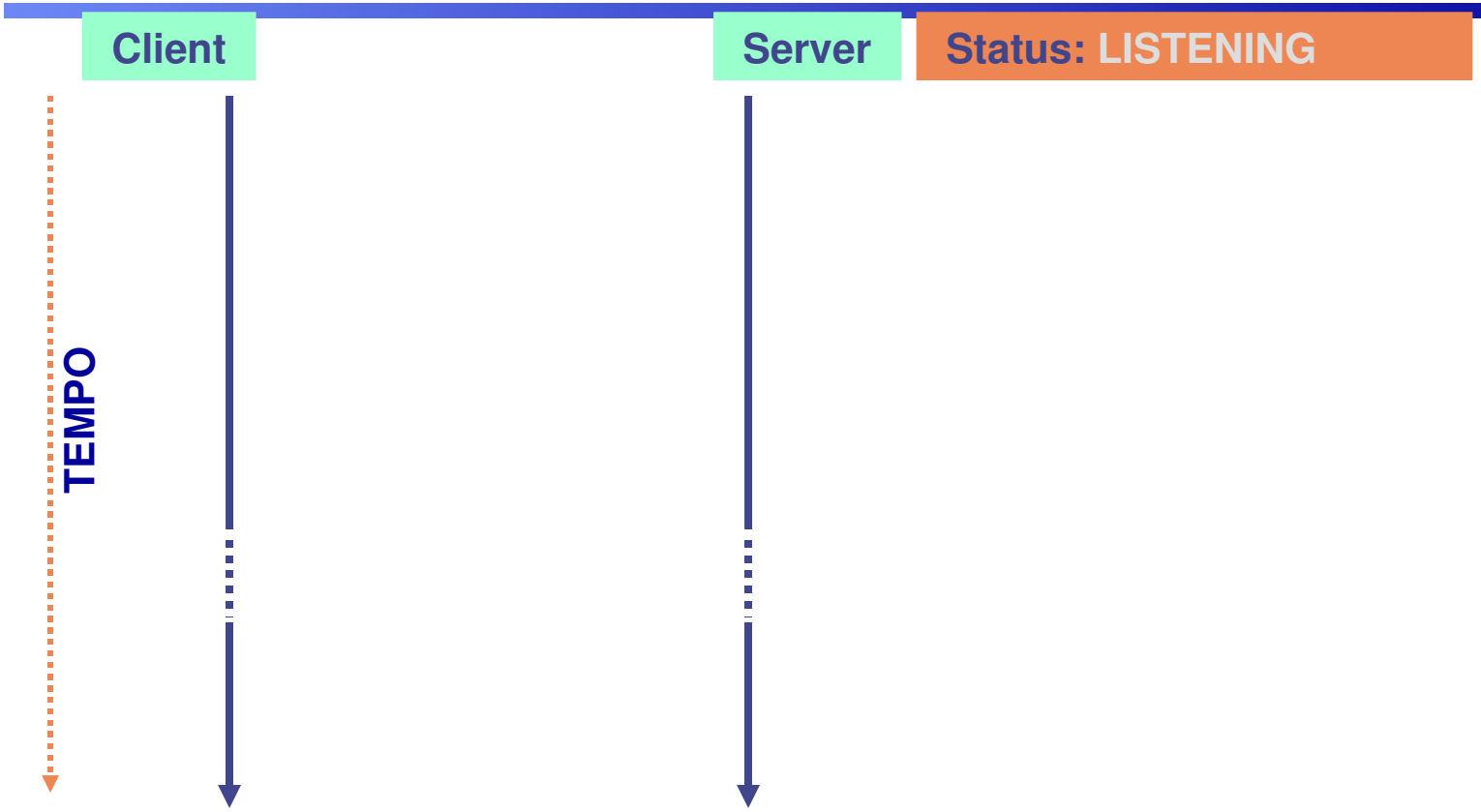
O Cabeçalho TCP



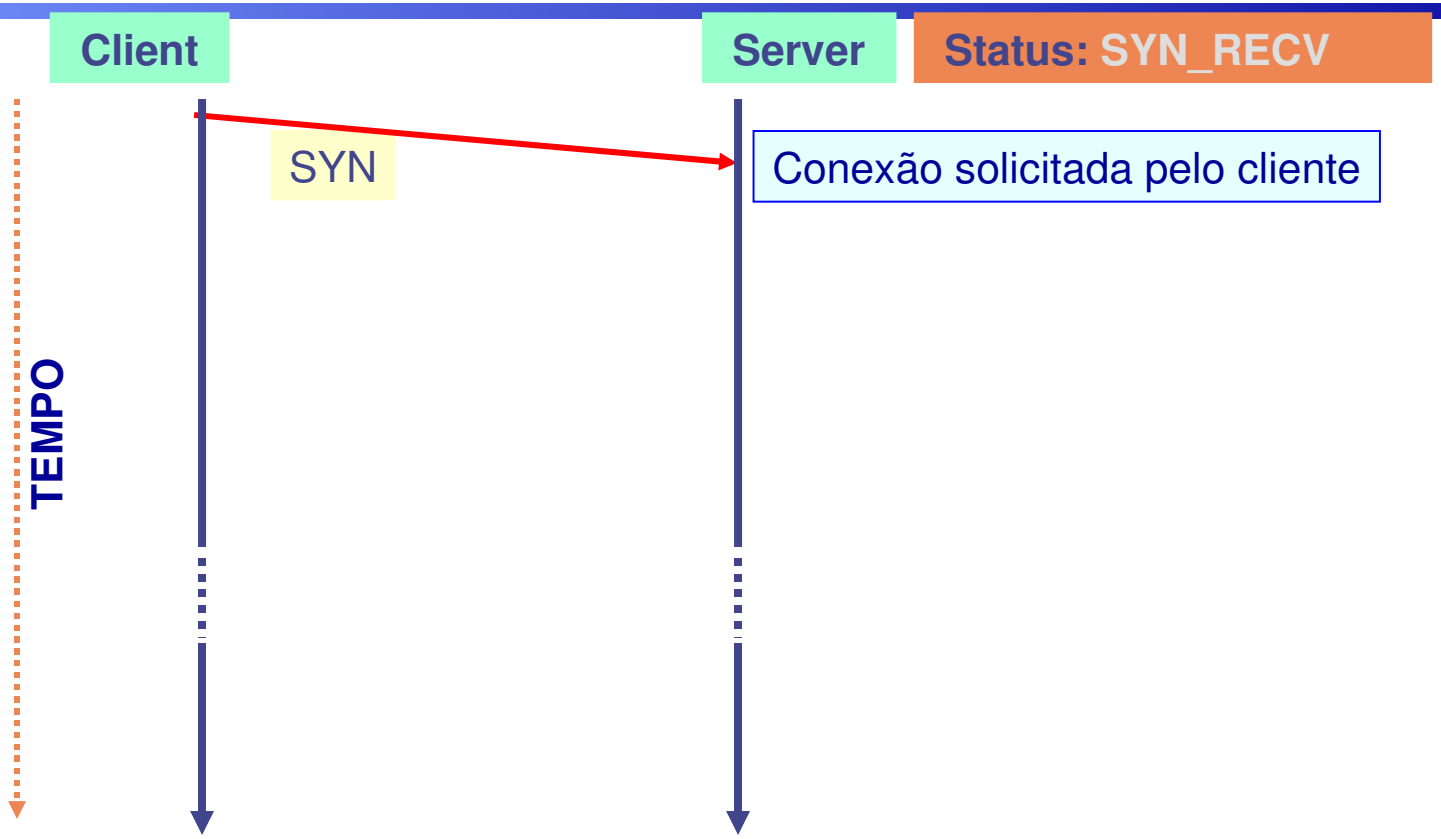
O Protocolo TCP

- Estabelecimento de Conexão
  - Protocolo
    - **Passo 1: o cliente envia um segmento SYN especificando a porta do servidor ao qual deseja se conectar e seu número de sequência inicial**
    - **Passo 2: o servidor responde enviando outro segmento SYN com o ACK do segmento recebido e o seu próprio número de sequência**
    - **Passo 3: o cliente retorna um ACK e a conexão se estabelece**
  - O tamanho máximo de segmento (MSS) que cada lado se propõe a aceitar também é definido no momento do estabelecimento da conexão
  - Pode acontecer um “half open”

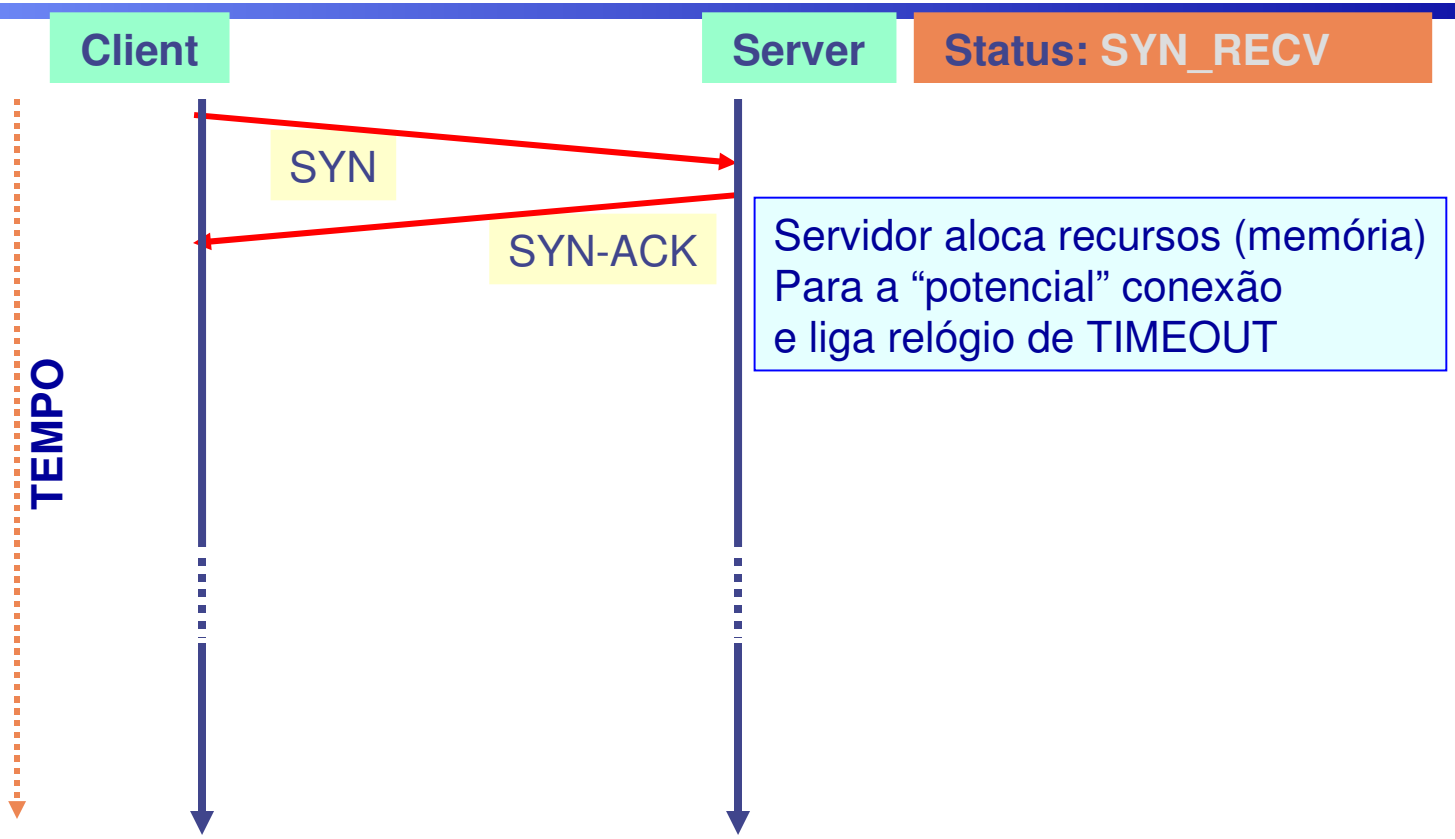
TCP – Como Funciona



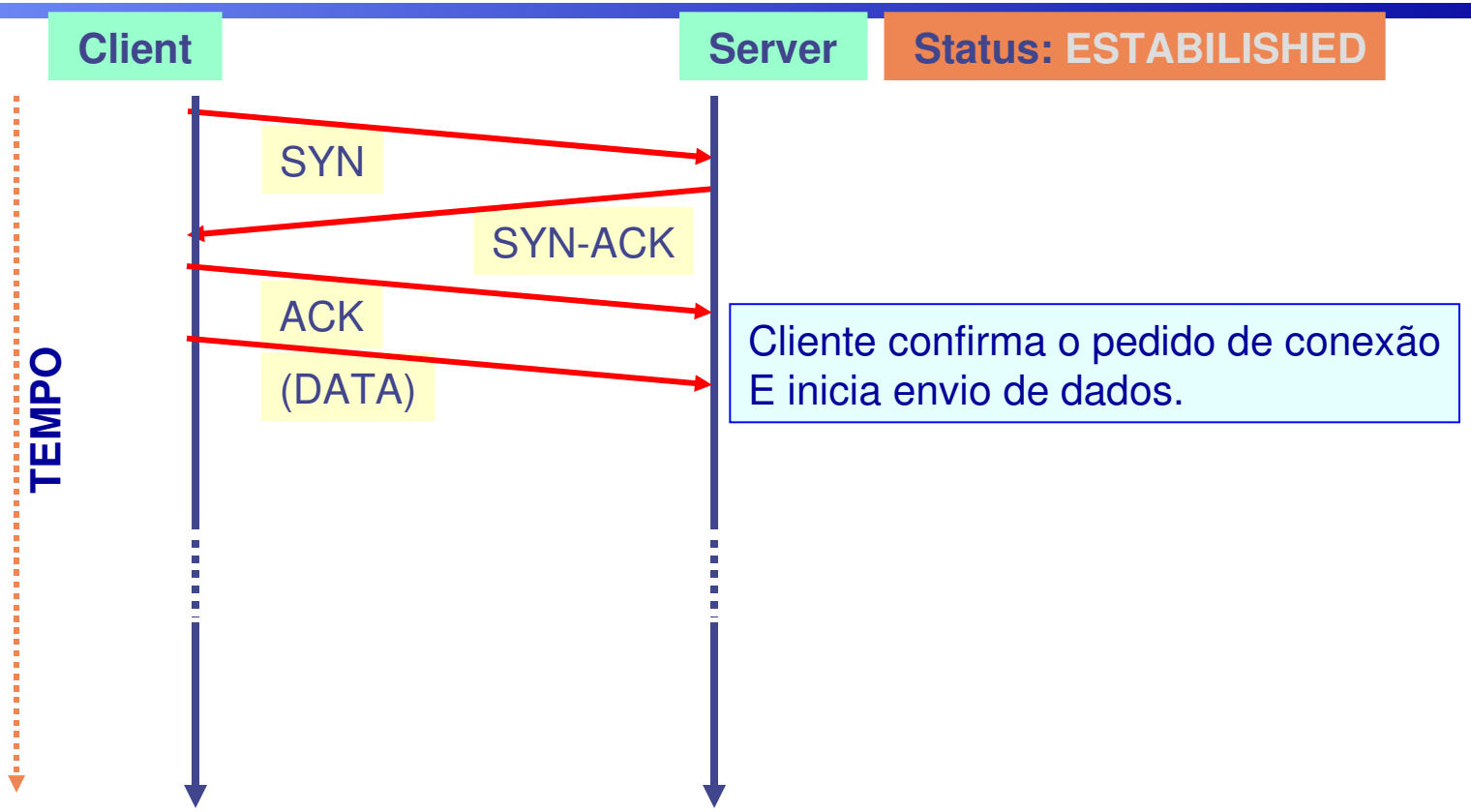
TCP – Como Funciona



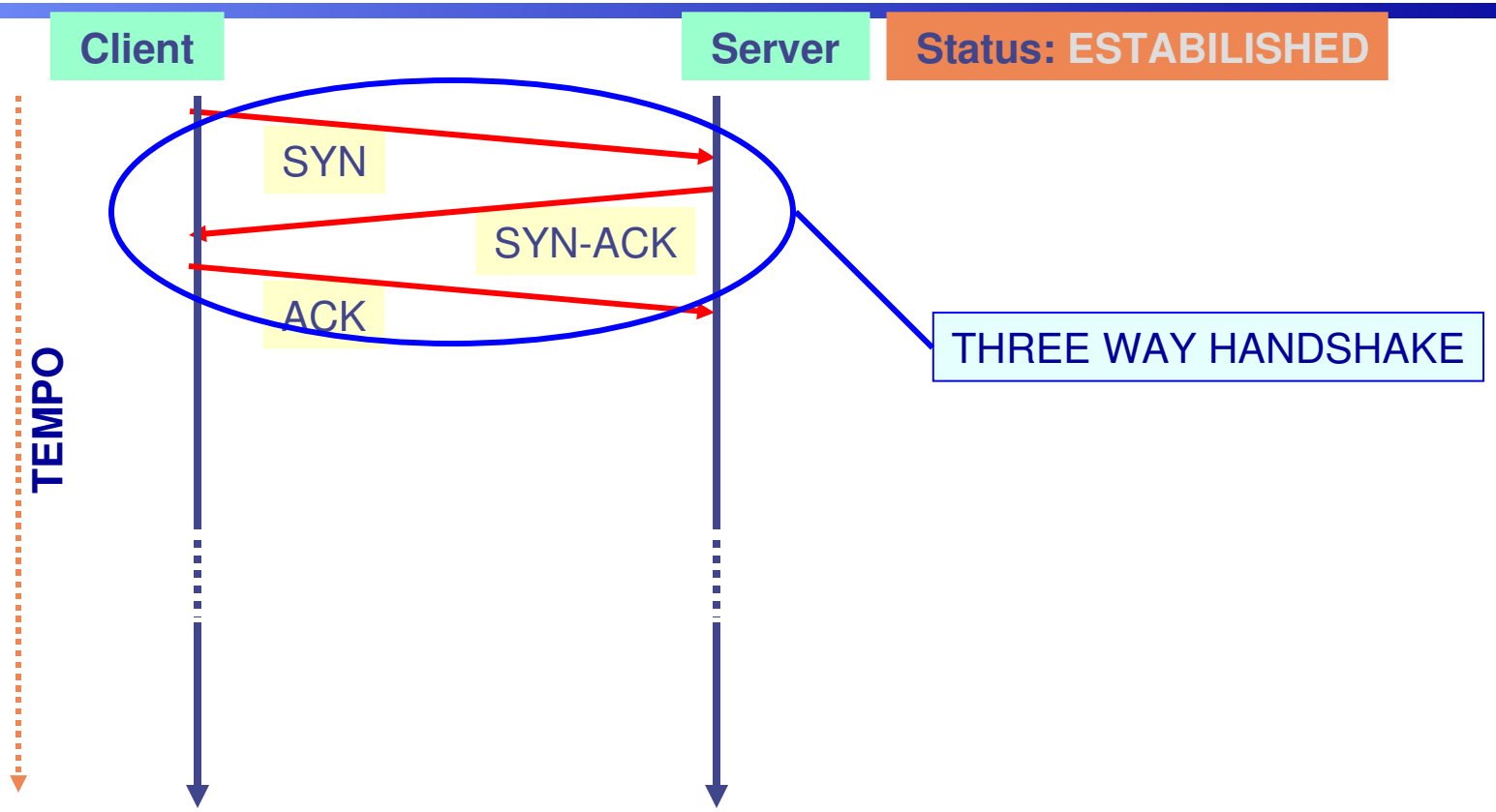
TCP – Como Funciona



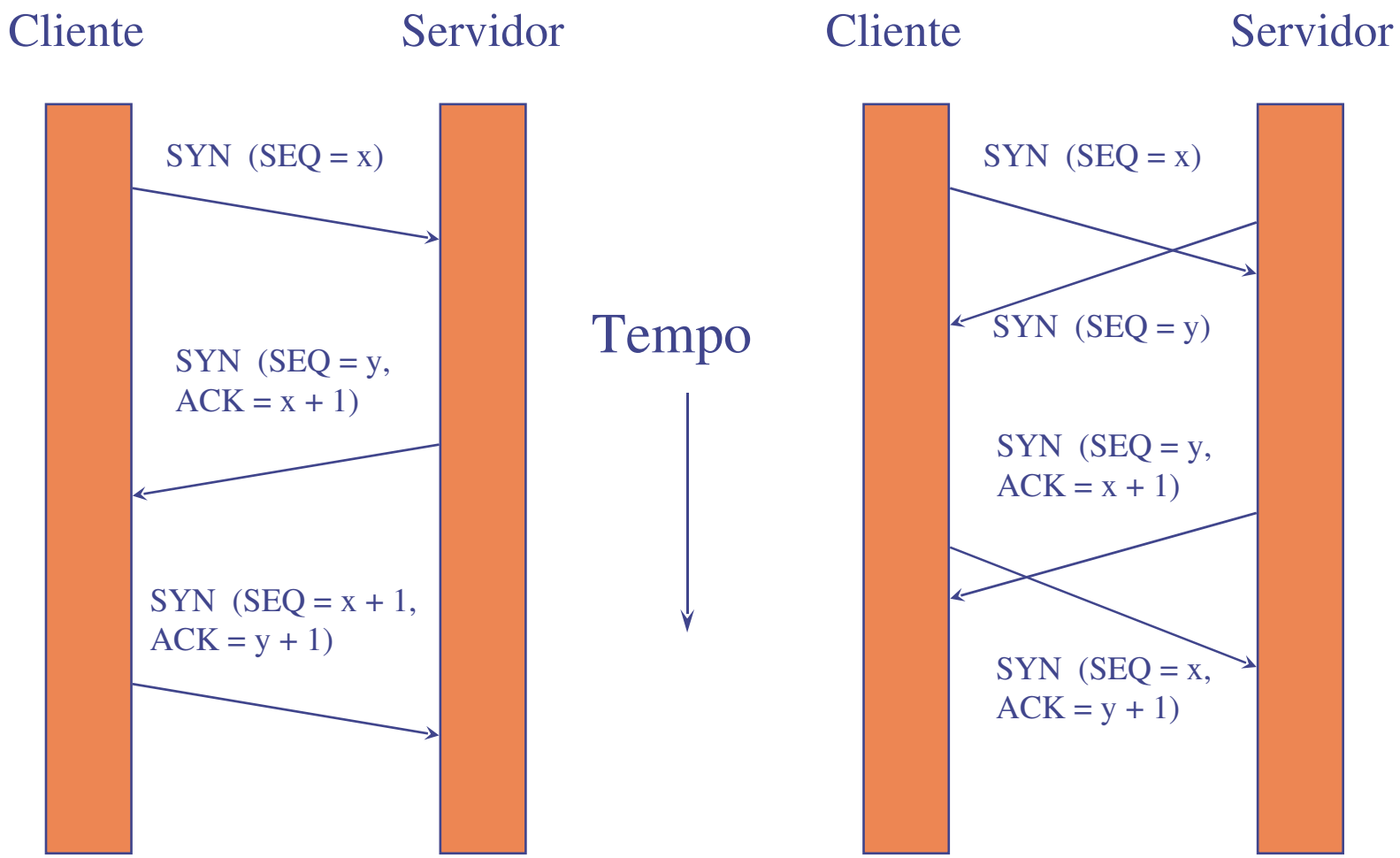
TCP – Como Funciona



TCP – Como Funciona



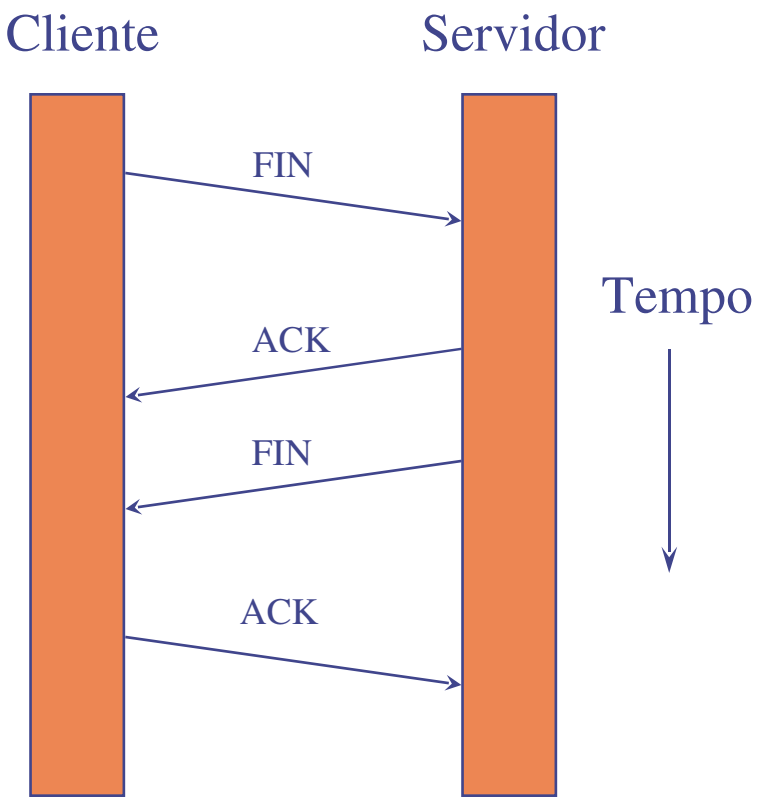
O Protocolo TCP-Estabelecimento de Conexão (cont.)



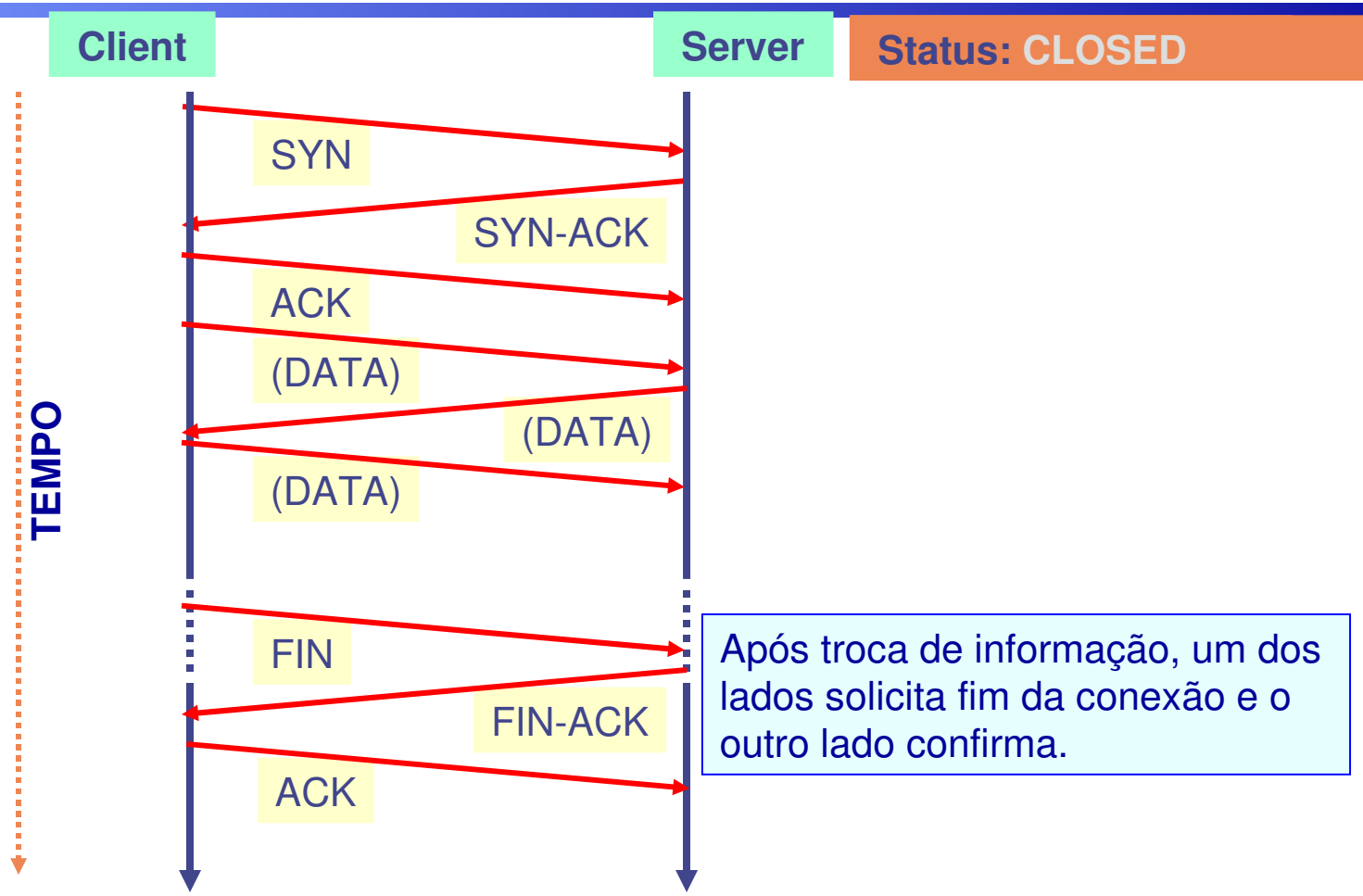
O Protocolo TCP

➔Término de Conexão

- Cada direção da conexão é encerrada independentemente
- Protocolo
  - **Passo 1: o cliente envia um segmento FIN**
  - **Passo 2: o servidor retorna um FIN e um ACK para o cliente**
  - **Passo 3: o cliente envia um ACK e a conexão se encerra**
- É possível efetuar um “half close”, mantendo-se apenas uma conexão simplex



TCP – Como Funciona



Após troca de informação, um dos lados solicita fim da conexão e o outro lado confirma.

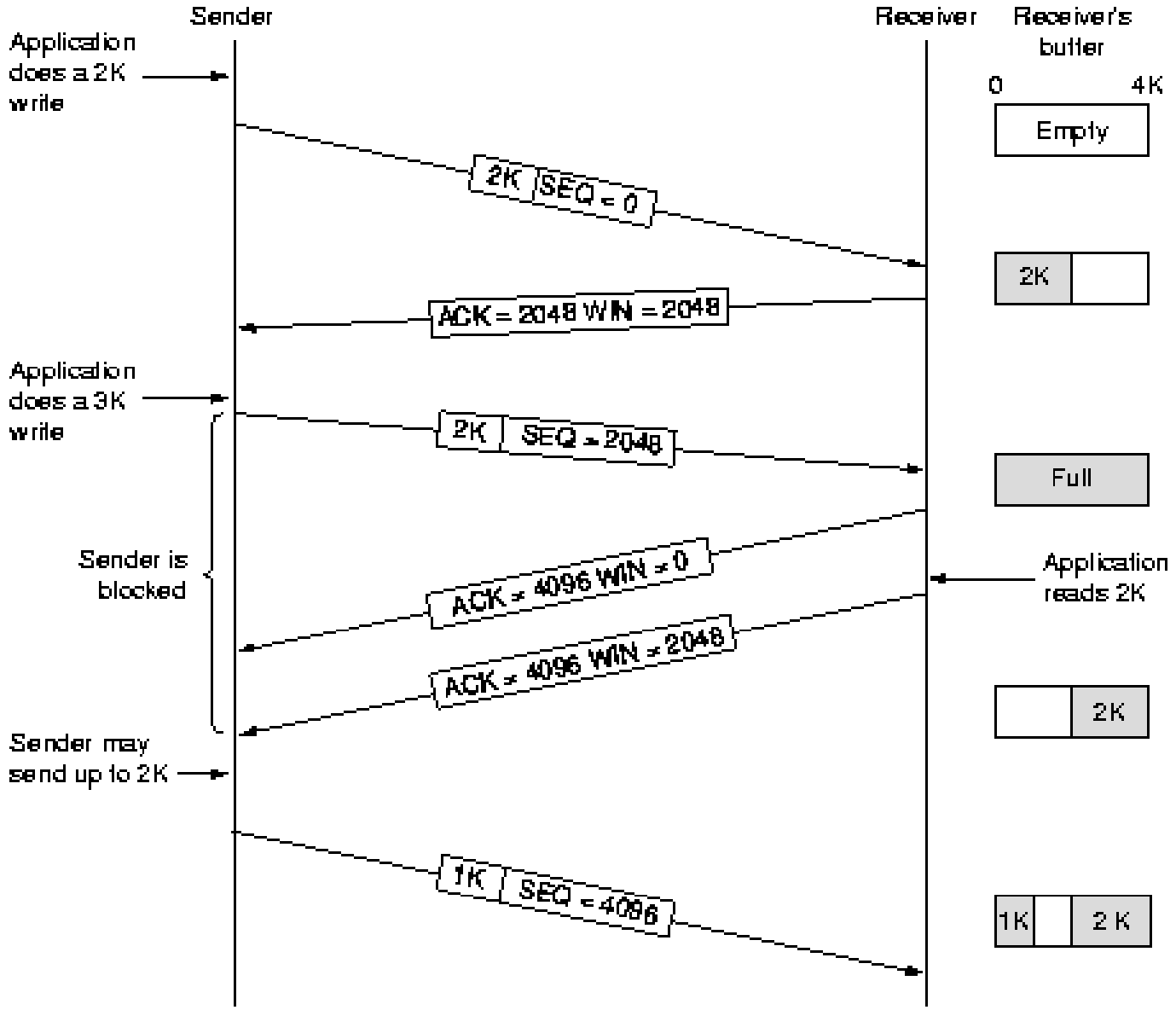
Protocolos de Transporte da Internet (cont.)

- Uso de uma tamanho de janela de 16 bits => problema em linhas de grande atraso (satélite ou transcontinental)
- RFC 1323 define um extensão ao tamanho da janela a ser 32 bits (suportado pelas implementações)
- RFC 1106 propõe o uso de uma repetição seletivo no lugar do “go back n” usando NAKs

Gerenciamento de Conexões em TCP

- Estabelecimento de conexão TCP usando o “Three-way handshake”
- Na liberação de conexão (FIN), temporizadores são usados para evitar o problema dos exércitos

Gerenciamento de buffers no TCP



Política de Transmissão em TCP

- O gerenciamento da janela TCP não é direitamente ligado aos reconhecimentos (ACKs)
- Quando o tamanho == 0, o transmissor pode enviar dados urgentes, ou um segmento de 1 byte pedindo ao receptor re-anunciar o próximo byte esperado (para evitar deadlock no caso da perda de um ACK!)
- Atrasar (de 500 msec) ACKs e updates das janelas para economizar na troca de pacotes

O Protocolo TCP - Tráfego Interativo

- Cada letra digitada gera um segmento: Telnet, Rlogin
- Muitos pacotes pequenos podem provocar congestionamento em redes lentas (WAN)
- O TCP tenta minimizar o número de pacotes trocados:
  - Reconhecimentos (ACKs) atrasados
  - Algoritmo Nagle
    - **Apenas um segmento pequeno pode ficar pendente (sem reconhecimento) a cada momento**
    - **A taxa de envio dos pacotes se adequar a velocidade da rede**
    - **Nem sempre é desejável. Ex: X Windows**

O Protocolo TCP - Tráfego constante

- FTP, SMTP
- Controle de fluxo é efetuado através do protocolo de janela deslizante (“Sliding Window Protocol”):
  - O servidor (receptor) fica enviando o tamanho de sua janela de recepção (nº de bytes disponíveis no buffer de entrada)
  - A janela de envio do cliente evolui da seguinte forma:
    - **a janela fecha quando segmentos são enviados e reconhecidos**
    - **quando o servidor lê os dados recebidos liberando o buffer do TCP, este sinaliza a nova janela de recepção para o cliente, que abre sua janela de envio**
  - O cliente estabelece um “persist timer” para ficar inquirindo o servidor quanto ao tamanho de sua janela de recepção mesmo que ela esteja aparentemente fechada para evitar deadlocks
- O problema da “Silly Window Sindrome”(aplicação lendo 1 byte cada vez!)

Controle de Congestionamento no TCP

Segmentos TCP

- Um segmento consiste em um cabeçalho seguido de bytes de dados.
- O software TCP decide qual deve ser o tamanho dos segmentos.
- Porém, cada segmento deve caber na carga útil do IP, que é 64KB.

Temporizadores

- Quando um segmento é enviado, o transmissor ativa um temporizador.

- Quando o segmento chega ao destino, a entidade TCP receptora retorna um segmento com o próximo número de seqüência que espera receber.

- Se o temporizador do transmissor expirar antes da confirmação chegar, haverá uma retransmissão.

Janela de Congestionamento

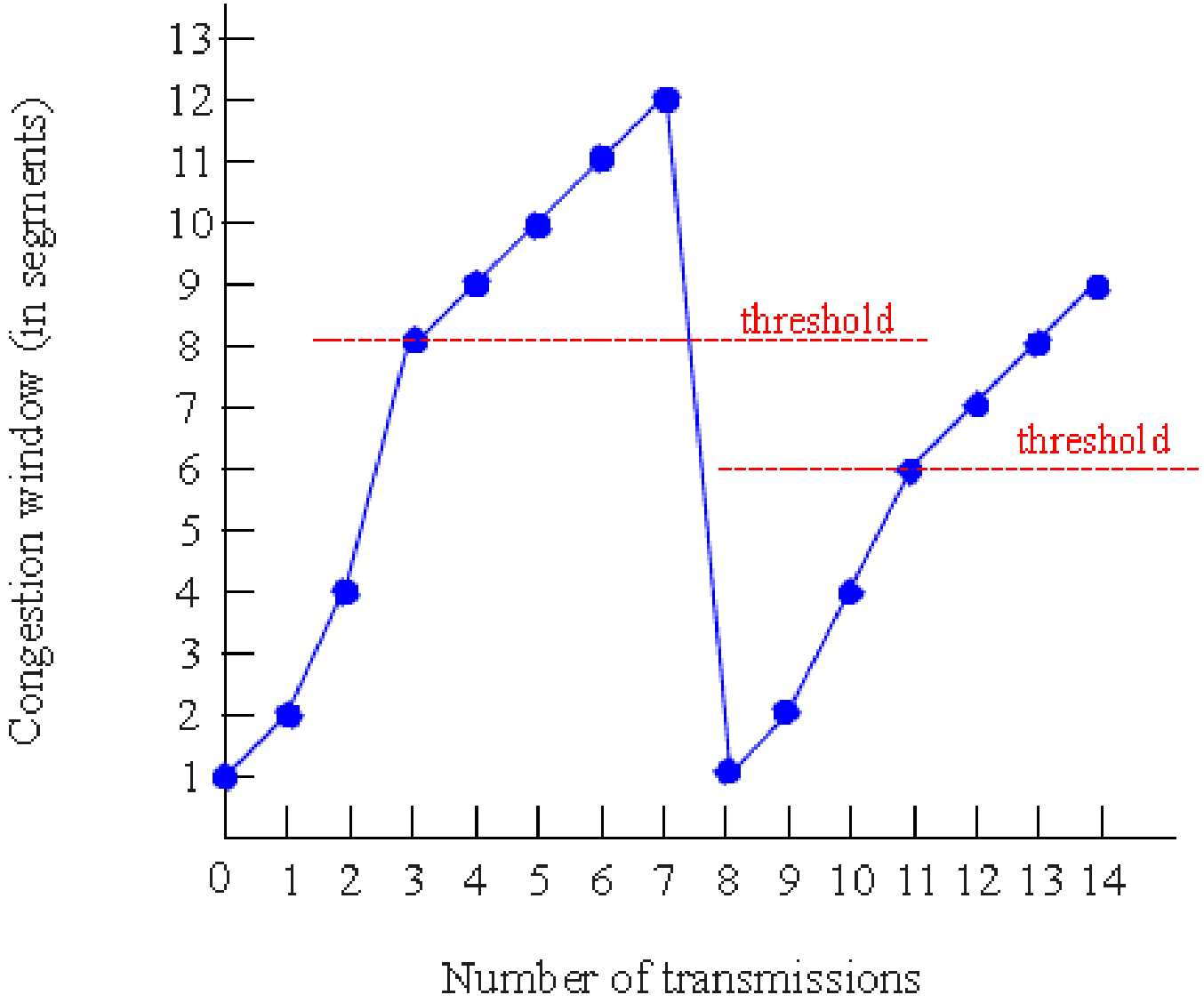
- Uma conexão TCP controla sua taxa de transmissão limitando o seu número de segmentos que podem ser transmitidos sem que uma confirmação seja recebida.

- Esse número é chamado o tamanho da janela do TCP (w)
- Uma conexão TCP começa com um pequeno valor de *w* e então o incrementa arriscando que exista mais largura de banda.
- Isso continua a ocorrer até que algum segmento seja perdido.
- Nesse momento, a conexão TCP reduz *w* para um valor seguro, e então continua a arriscar o crescimento.

Controle de Congestionamento

- **O controle é feito através de duas variáveis adicionadas em cada lado da conexão:**
  - ***Janela de Congestionamento;***
    - Janela do TCP explicada anteriormente.
  - ***Limiar;***
    - Serve para controlar o crescimento da janela de congestionamento.

Graficamente ...



Janela do Receptor

- **O número máximo de segmentos não confirmados é dado pelo mínimo entre os tamanhos das janelas de congestionamento e do receptor.**
  - *Ou seja, mesmo que haja mais largura de banda, o receptor também pode ser um gargalo.*

Evolução de uma Conexão TCP

- No início, a janela de congestionamento tem o tamanho de um segmento.
  - Tal segmento tem o tamanho do maior segmento suportado.
- O primeiro segmento é enviado e então é esperado seu reconhecimento.
  - Se o mesmo chegar antes que ocorra o timeout, o transmissor duplica o tamanho da janela de congestionamento e envia dois segmentos.
  - Se esses dois segmentos também forem reconhecidos antes de seus timeouts, o transmissor duplica novamente sua janela, enviando agora quatro segmentos.

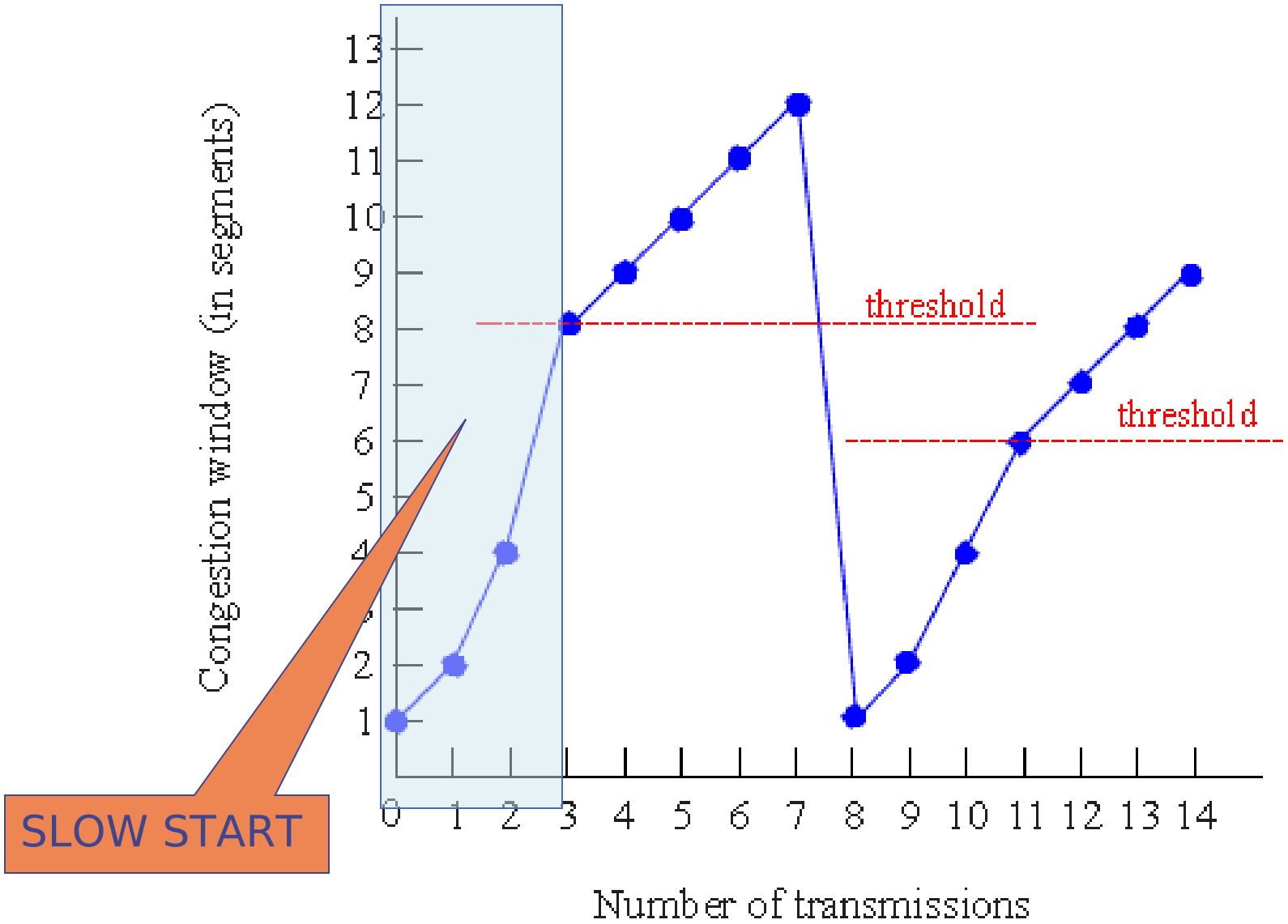
Evolução de uma Conexão TCP

- Esse processo continua até que:
  - O tamanho da janela de congestionamento seja maior que o limiar, ou maior que o tamanho da janela do receptor;
  - Ocorra algum timeouts antes da confirmação.

Duas Fases dessa Evolução

- A primeira fase, em que a janela de congestionamento cresce exponencialmente é chamada de inicialização lenta (slow start), pelo fato de começar com um segmento.
  - A taxa de transmissão começa pequena porém cresce muito rapidamente.

Graficamente ...

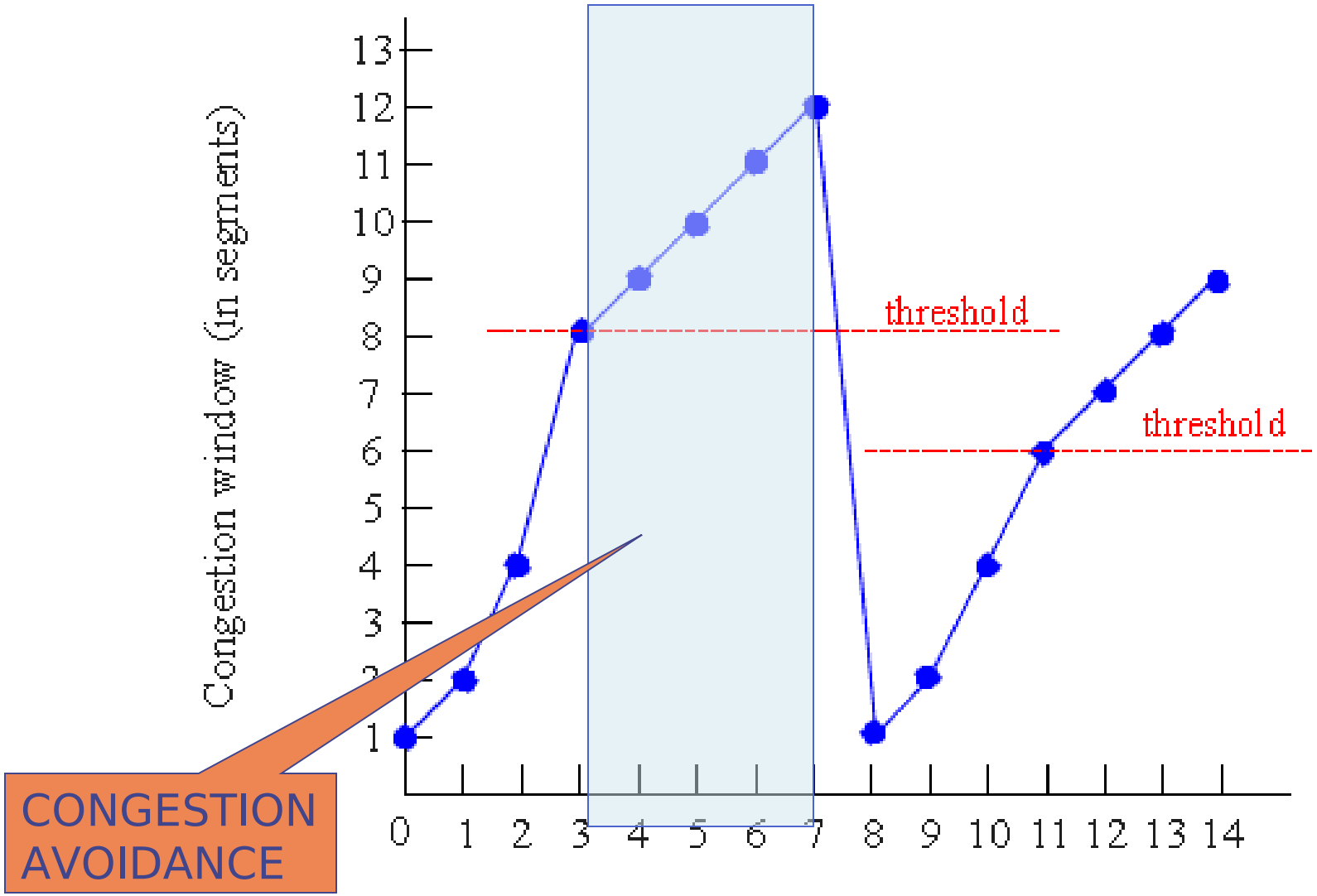


Duas Fases dessa Evolução

- **Uma vez ultrapassado o limiar, e a janela do receptor ainda não seja um limitante, o crescimento da janela passa a ser linear.**

- **Essa *segunda fase* é chamada de prevenção de congestionamento (congestion avoidance).**
  - **Sua duração também depende da não ocorrência timeouts, e da aceitação do fluxo por parte do receptor.**

Graficamente ...



E quando ocorrer um problema?

Evolução de uma Conexão TCP

- **Na ocorrência de um timeout, as seguintes atitudes são tomadas:**
  - **O valor do limiar passa a ser a metade do do atual tamanho da janela de congestionamento.**
  - **O tamanho da janela de congestionamento volta ser do tamanho de um segmento.**
  - **O tamanho da janela de congestionamento volta a crescer exponencialmente.**

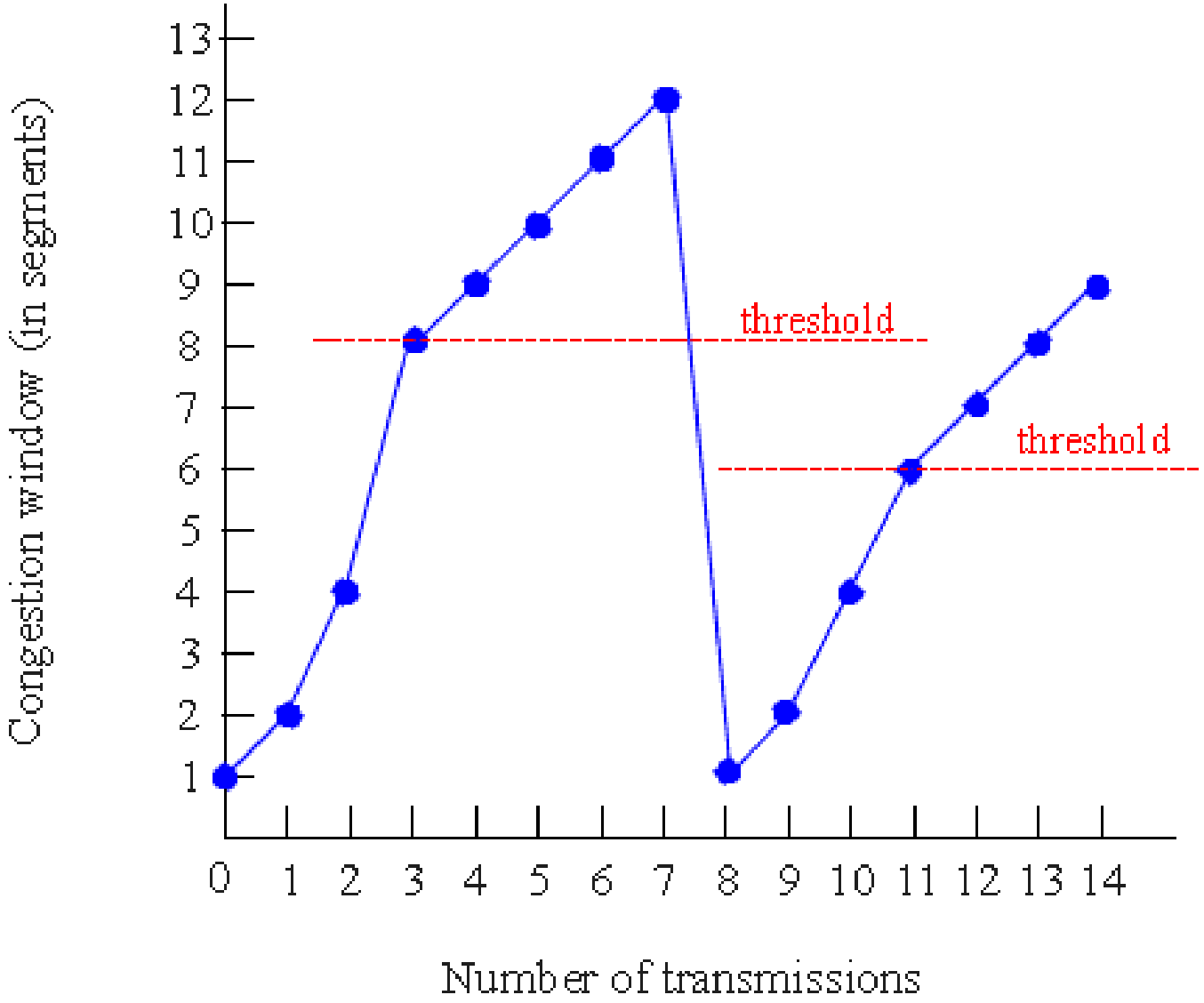
Resumo

- Quando o tamanho da janela de congestionamento está abaixo do limiar, seu crescimento é exponencial.

- Quando este tamanho está acima do limiar, o crescimento é linear.

- Todas as vezes que ocorrer um timeout, o limiar é modificado para a metade do tamanho da janela e o tamanho da janela passa a ser 1.

Graficamente ...



O Header UDP

Pseudo-header

32-bit source IP address		
32-bit destination IP address		
0 0 0 0 0 0 0 0	protocol	length

Header

Source port	Destination port
UDP length	UDP checksum

Portas UDP

- Processos requerem portas UDP para transmissão e recepção de datagramas
- Sistemas que suportam multicast permitem que mais de um processo compartilhe a mesma porta
- Os servidores que usam UDP podem restringir o recebimento de datagramas a nível local e remoto

TCP vs UDP

TCP

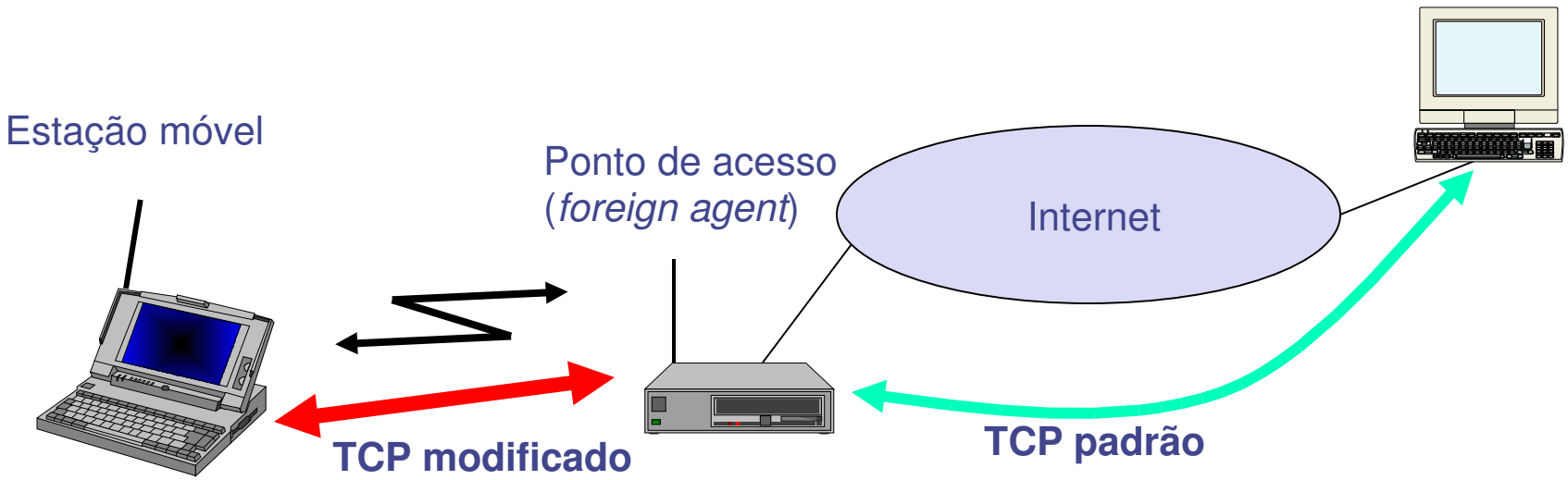
- Confiável, faz tratamento de erros (perdas) e garante a entrega dos dados
- Adaptativo: faz controle de congestionamento variando a taxa de envio. Adapta a taxa de transmissão à banda disponível
- Usa ACKs
- Aplicação não se preocupa com controle de perdas de pacotes e sequenciamento
- Controle de fluxo: janela deslizante temporização e timeout
- Usado pelos protocolos FTP, HTTP, POP, SMTP
- Requer tempo adicional para estabelecer conexão

UDP

- Não Confiável - não faz tratamento de perda de pacotes
- Não Garante a entrega dos dados
- Usado pelo SNMP, DHCP, DNS
- Não Adaptativo: toma o máximo da banda disponível
- Não faz controle de fluxo
- Exige que a aplicação execute controle de perdas de pacotes e sequenciamento
- Não requer tempo adicional para estabelecer conexão

TCP Móvel

- I-TCP Indirect TCP



TCP Móvel

- *Snooping* TCP