

Camada de Transporte

Protocolos UDP e TCP

Internetworking with TCP/IP
D. Comer



Mário Meireles Teixeira. UFMA-DEINF

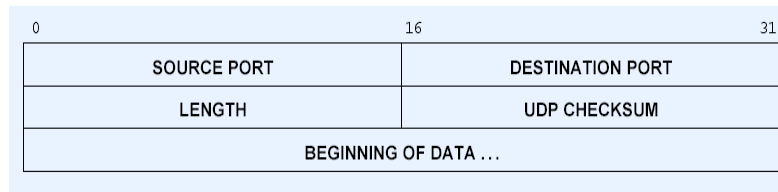
Protocolo UDP

- Protocolo de transporte não orientado a conexão
- Funciona sobre o serviço de datagramas do IP
- Mesma semântica do IP:
 - Não realiza controle de erros: mensagens podem ser perdidas, duplicadas ou chegar fora de ordem
 - Não realiza controle de fluxo
 - Não usa confirmações (ACKs), nem retransmissões
 - Confiabilidade é deixada a cargo dos programas
- O que o UDP faz?
 - Fornece uma abstração de *portas* que permite distinguir entre múltiplos programas executando no mesmo host
 - Indicado para aplicações C/S em redes locais e multimídia em tempo real





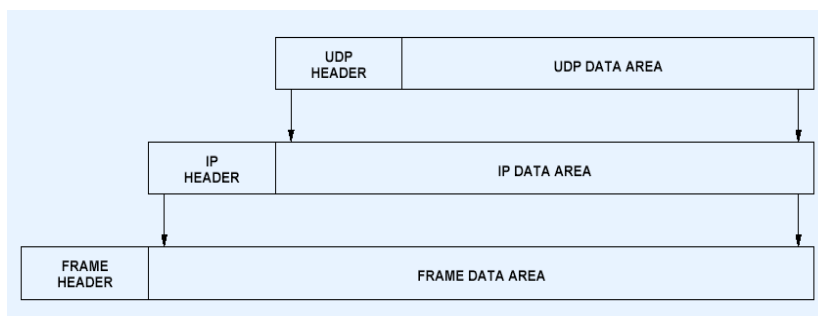
Formato do Datagrama UDP



- As portas identificam os pontos de origem e destino dos segmentos UDP em um host
- O campo LENGTH inclui o cabeçalho e os dados
- O campo CHECKSUM é opcional (valor "0" significa que não foi calculado)
- Atenção: o IP não calcula *checksum* para os dados!
- *Pseudo-header* UDP: *checksum* inclui endereços IP



Encapsulamento





Portas UDP Reservadas

Decimal	Keyword	UNIX Keyword	Description
0	-	-	Reserved
7	ECHO	echo	Echo
9	DISCARD	discard	Discard
11	USERS	systat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Network status program
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character Generator
37	TIME	time	Time
42	NAMESERVER	name	Host Name Server
43	NICNAME	whois	Who Is
53	DOMAIN	nameserver	Domain Name Server
67	BOOTPS	bootps	BOOTP or DHCP Server
68	BOOTPC	bootpc	BOOTP or DHCP Client
69	TFTP	tftp	Trivial File Transfer
88	KERBEROS	kerberos	Kerberos Security Service
111	SUNRPC	sunrpc	Sun Remote Procedure Call
123	NTP	ntp	Network Time Protocol
161	-	snmp	Simple Network Management Protocol
162	-	snmp-trap	SNMP traps
512	-	biff	UNIX comsat
513	-	who	UNIX rwho daemon
514	-	syslog	System log
525	-	timed	Time daemon



Protocolo TCP

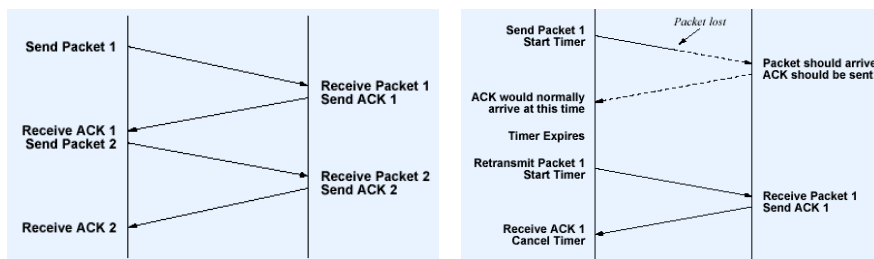
- Protocolo de transporte orientado a conexão
- Principal meio de transporte de mensagens na arquitetura TCP/IP, usado pela maioria das aplicações da Internet
- Características
 - Orientado a fluxo (stream)
 - Conexão de circuitos virtuais
 - Transmissão buferizada
 - Fluxos não estruturados
 - Conexão full duplex
 - Confiabilidade

Confiabilidade



- Como o TCP pode garantir confiabilidade se a camada de rede fornece apenas um serviço de datagramas não-confiável?
- Técnica tradicional: *Positive Acknowledgement with Retransmissão* (PAR)
 - Receptor envia uma confirmação (ACK) ao receber o pacote
 - Transmissor aguarda a chegada do ACK antes de transmitir o próximo pacote. Mantém um timer para cada pacote transmitido
 - Transmissor retransmite se o timer expirar antes da chegada do ACK

Funcionamento do PAR



- Podem surgir duplicatas de dados e de confirmações. Solução:
 - Cada pacote recebe um número de seqüência
 - Os ACKs carregam consigo o número de seqüência, a fim de que o transmissor possa associá-los corretamente com os pacotes

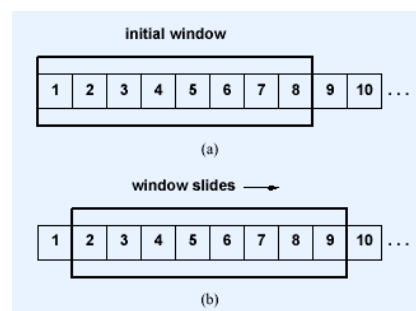


Desvantagem do PAR Simples

- Na sua forma mais simples, o esquema PAR desperdiça boa parte da largura de banda da rede
 - É preciso esperar até que o ACK anterior chegue, antes de enviar o próximo pacote
 - O problema se agrava se a rede tiver uma latência alta
- Solução:
 - Permitir vários pacotes pendentes (não confirmados) simultaneamente na rede
 - Mantém-se o esquema de ACK + Retransmissão
 - Abordagem de *Janela Deslizante* (sliding window)

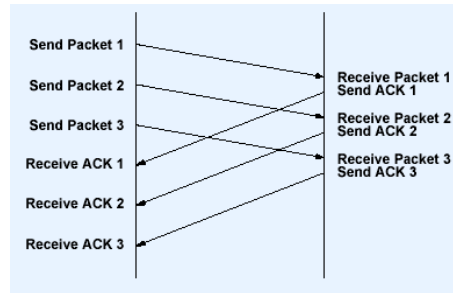
Janela Deslizante

- Tamanho da janela é **fixo**
- São transmitidos todos os pacotes dentro da janela, sem esperar por confirmações
- Ao receber o ACK para o **primeiro pacote** dentro da janela, o transmissor a move uma posição para a direita
- Mantém-se um timer para cada pacote não confirmado. Ao expirar, retransmite-se o pacote
- TamJanela = 1 (PAR simples)
- Receptor também mantém uma janela deslizante



- Três tipos de pacotes:
 - Confirmados (à esquerda)
 - Pendentes (dentro da janela)
 - A transmitir (à direita)

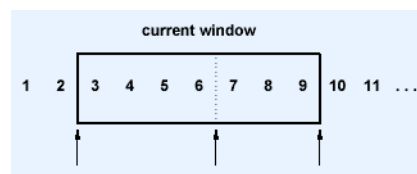
Janela Deslizante



- Um esquema de janela deslizante bem ajustado aproveita toda a largura de banda disponível na rede, obtendo um *throughput* bem maior que o do PAR simples
- O pacote de menor número na janela é o primeiro na seqüência que não foi confirmado

Janela Deslizante: TCP

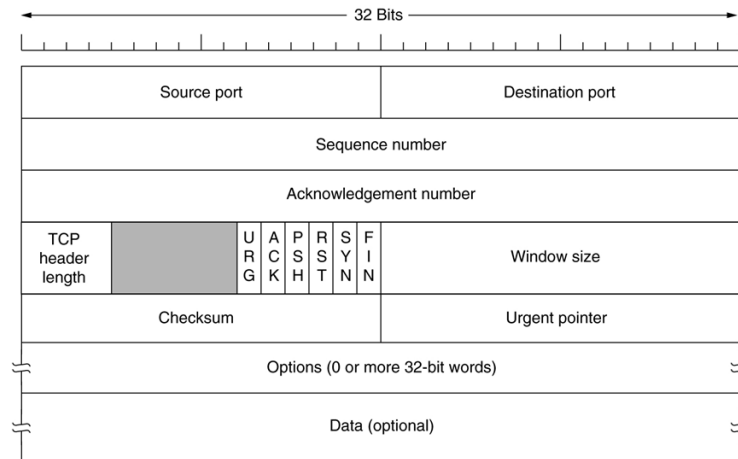
- Trabalha no nível de octetos
- Resolve o problema de **confiabilidade** e **controle de fluxo** fim-a-fim
- Mantém três ponteiros na janela
- Os octetos dentro da janela são enviados tão rápido quanto possível
- Tamanho de janela **variável** → *window advertisement*
- Cada ACK informa quantos octetos adicionais o receptor está preparado para receber → *controle de fluxo*
- TamJanela = 0 → *stop*



- Quatro janelas por conexão
 - Transmissor + Receptor
 - Conexão full duplex



Formato do Segmento TCP



Formato do Segmento TCP

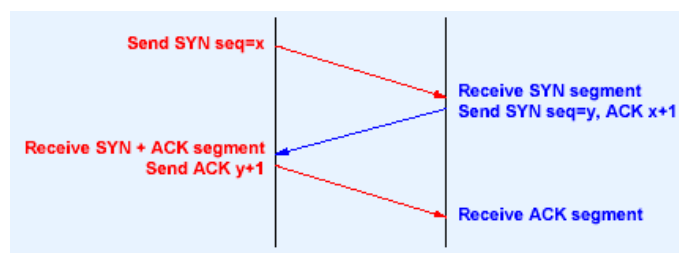
Source/Dest. Port	Números de porta TCP
Sequence Number	Número seqüencial (posição) do segmento no stream
ACK	Número do próximo octeto esperado
HLEN	Tamanho do cabeçalho
Reserved	Reservado para uso futuro (6 bits)
Code bits	URG : Campo <i>urgent pointer</i> é válido ACK : Campo ACK é válido PSH : Requisição de push RST : Reseta a conexão SYN : Pedido de início de conexão FIN : Pedido unilateral de desconexão

Formato do Segmento TCP



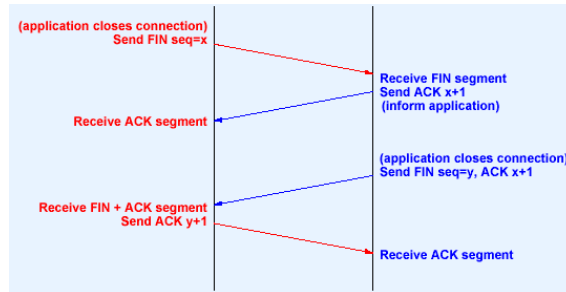
Window	Tamanho da janela (buffer)
Checksum	Checação de integridade do segmento (cabeçalho e dados)
Urgent Pointer	Identifica a posição do dado urgente no segmento (<i>out-of-band data</i>)
Options	Usado para negociação com o par do outro lado da conexão (p.ex., MSS)
Padding	Preenchimento para tornar o tamanho do segmento múltiplo de 32 bits

Estabelecimento de Conexões



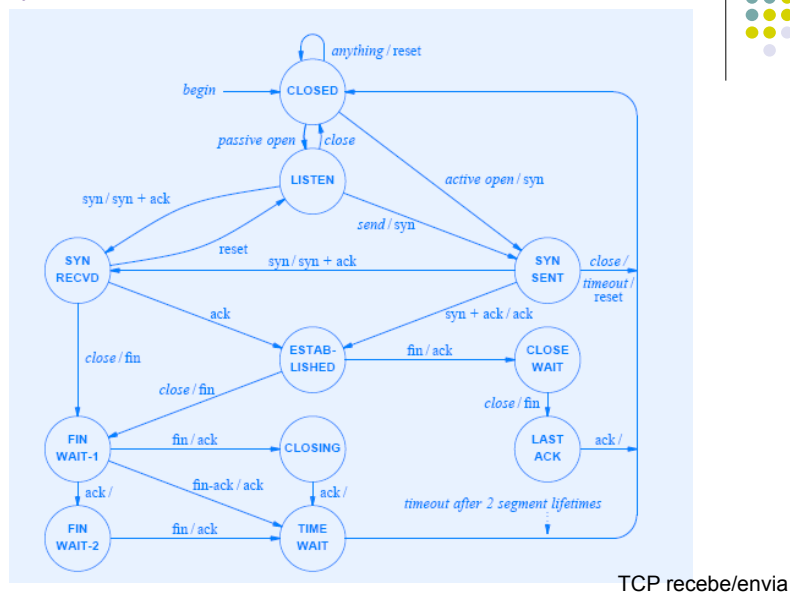
- Handshake de Três Vias
- Open passivo e ativo
- Uma vez estabelecida a conexão, os dados podem fluir em ambas as direções. Não existe relação de mestre-escravo

Encerramento de Conexões



- Handshake de Três Vias modificado (conexões TCP são full duplex)
- Pedido de desconexão (FIN) é unilateral: solicitante não mais enviará dados naquela direção (exceção aos ACKs), porém ainda pode receber dados (direção oposta)
- **RST**: Termina a conexão imediatamente (ambos os lados param a transmissão; recursos liberados)

Máquina de estados do TCP





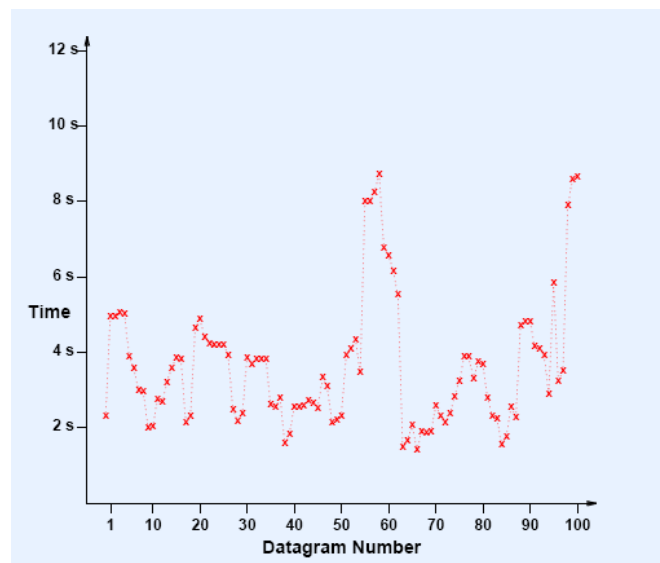
Acknowledgements

- ACKs não se referem a segmentos, mas a uma posição específica no stream de dados
- Um ACK refere-se ao próximo octeto esperado pelo receptor (*esquema cumulativo*)
 - Simples de gerenciar e não ambíguo
 - Porém, o transmissor não recebe informações sobre cada transmissão, mas sim sobre uma posição particular no stream
- Exemplo:
 - Transmissor envia 5 segmentos
 - Chegam os 4 últimos, mas não o primeiro segmento
 - ACK refere-se ao primeiro segmento. O que fazer?
 - *PAR; Go-Back-N; Selective ACK*



Retransmissão

- Para cada segmento transmitido, o TCP inicia um *timer de retransmissão* e aguarda o ACK. Se o timer expira, assume que o segmento foi perdido e o retransmite
- No ambiente da Internet:
 - Atrasos em cada conexão variam com o tempo
 - Atrasos variam bastante entre conexões
- Portanto, um valor fixo para o timer não é adequado:
 - Muito curto → retransmissões desnecessárias
 - Muito longo → atraso desnecessário
- Solução: *retransmissão adaptativa*



- Tempos de ida e volta (RTT) medidos para 100 datagramas IP sucessivos (Comer, 2006)

Retransmissão Adaptativa

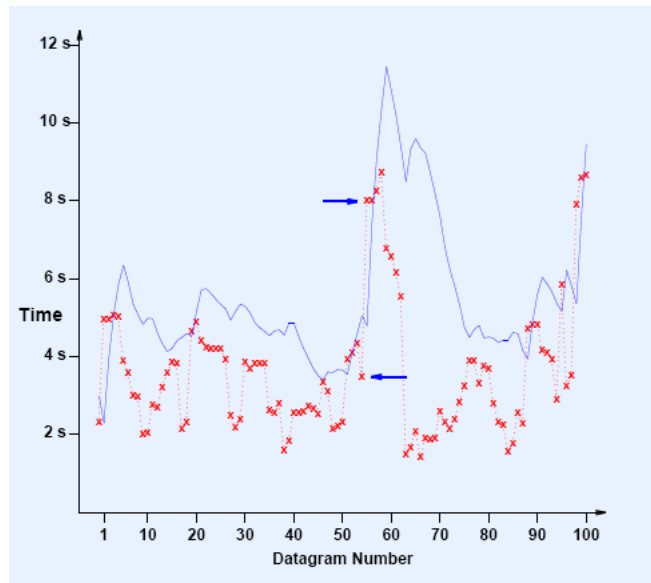
- Faz-se uma estimativa do *round-trip time* (RTT) em cada conexão:

$$RTT = (\alpha * RTT) + (1 - \alpha) * New_RTT, 0 < \alpha < 1$$

- Valor do timeout baseia-se na estimativa do RTT:

$$Timeout = \beta * RTT$$

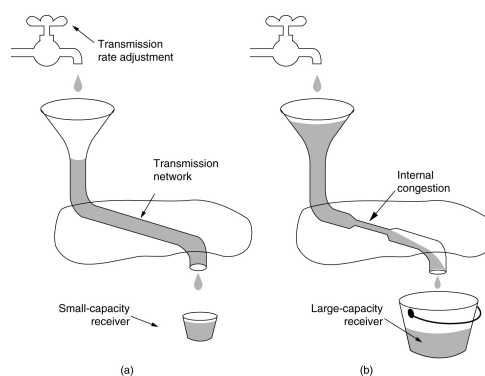
- Algoritmo de Karn
 - Não atualizar o RTT para segmentos retransmitidos
 - Duplicar o valor do timeout a cada falha
 - Timeout volta a ser função de RTT para o próximo segmento que chegar ao destino da primeira vez



- O timer de retransmissão procura acompanhar o RTT (as setas indicam quando o timeout dobra)

Controle de Congestionamento

- Carga oferecida à rede maior que capacidade da rede → **congestionamento**
- Atualmente, perda de pacotes devido a erros de transmissão é relativamente raro
- Portanto, o TCP assume que os timeouts são causados por congestionamentos!
- Dois problemas potenciais: a capacidade do receptor e a capacidade da rede





Controle de Congestionamento

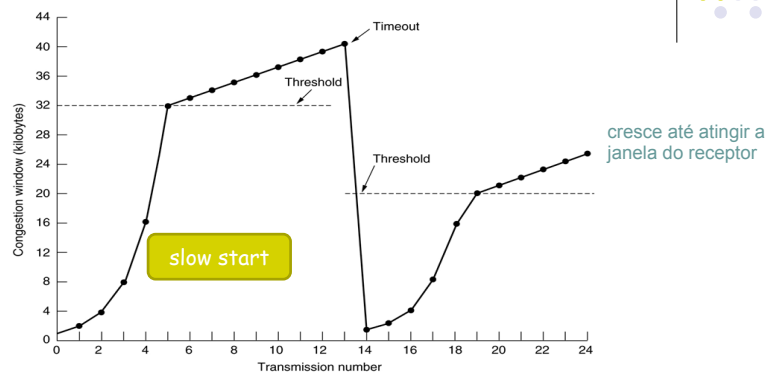
- Cada transmissor precisa manter duas janelas:
 - Tamanho da janela do receptor (*window advertisement*)
 - Janela de congestionamento

janela atual = min (janela_receptor, janela_congest)

- Quando a transmissão inicia, o transmissor faz a janela de congestionamento igual ao *MSS* (*max. segment size*) em uso na conexão



Controle de Congestionamento



- Transmissor usa:
 - Janela do receptor
 - Janela de congestionamento
- Internet usa:
 - *Threshold* (limiar)

Portas TCP Reservadas



Decimal	Keyword	UNIX Keyword	Description
0			Reserved
1	TCPMUX	-	TCP Multiplexor
7	ECHO	echo	Echo
9	DISCARD	discard	Discard
11	USERS	sysstat	Active Users
13	DAYTIME	daytime	Daytime
15	-	netstat	Network status program
17	QUOTE	qotd	Quote of the Day
19	CHARGEN	chargen	Character Generator
20	FTP-DATA	ftp-data	File Transfer Protocol (data)
21	FTP	ftp	File Transfer Protocol
22	SSH	ssh	Secure Shell
23	TELNET	telnet	Terminal Connection
25	SMTP	smtp	Simple Mail Transport Protocol
37	TIME	time	Time
43	NICNAME	whois	Who Is
53	DOMAIN	nameserver	Domain Name Server
67	BOOTPS	bootps	BOOTP or DHCP Server
77	-	rje	any private RJE service
79	FINGER	finger	Finger
80	WWW	www	World Wide Web Server
88	KERBEROS	kerberos	Kerberos Security Service
95	SUPDUP	supdup	SUPDUP Protocol
101	HOSTNAME	hostnames	NIC Host Name Server
102	ISO-TSAP	iso-tsap	ISO-TSAP
103	X400	x400	X.400 Mail Service
104	X400-SND	x400-snd	X.400 Mail Sending
110	POP3	pop3	Post Office Protocol Vers. 3
111	SUNRPC	sunrpc	SUN Remote Procedure Call