Protocolos de transporte a través de Internet para teleoperación

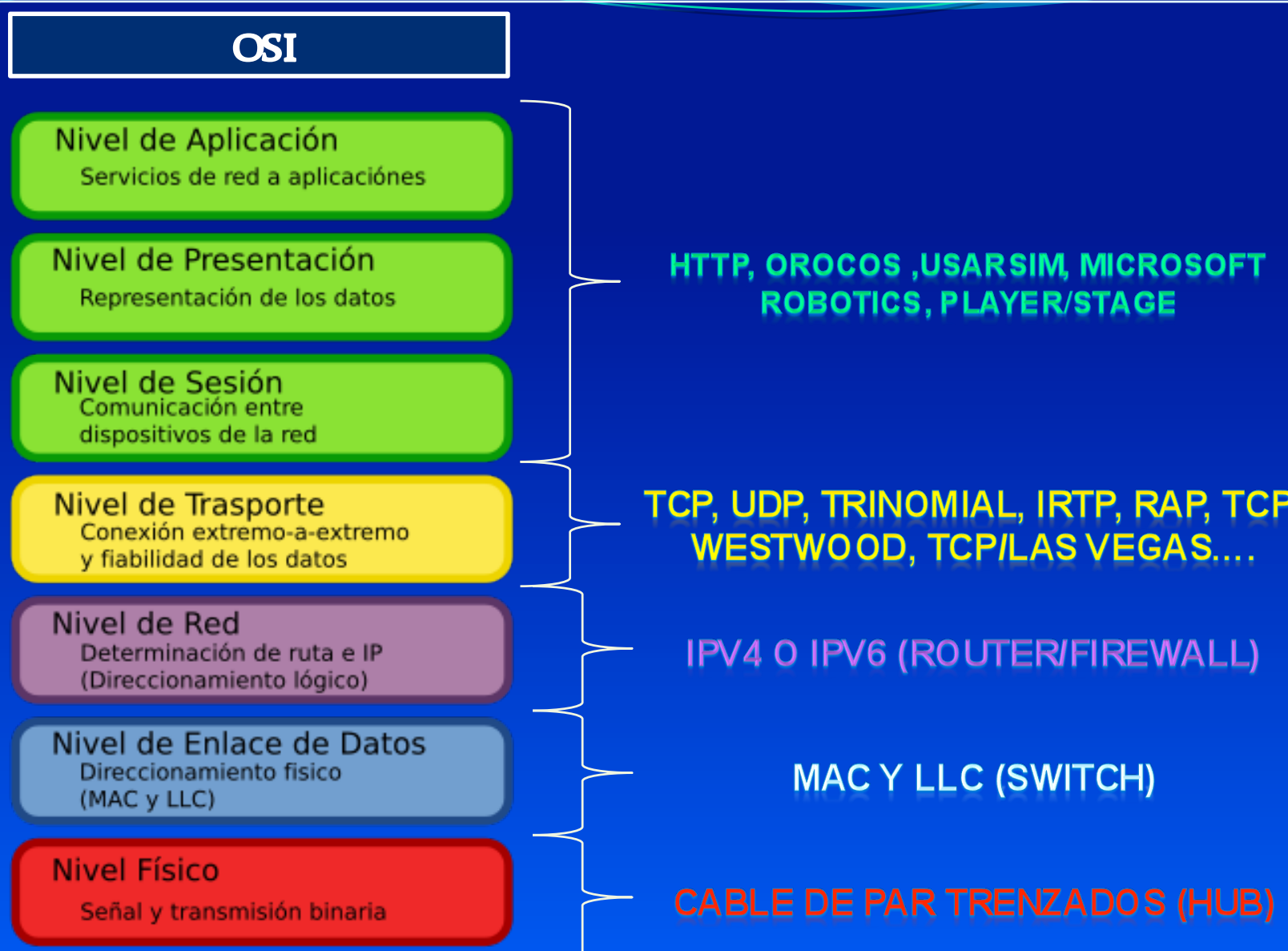
Raúl Wirz González
r.wirz@upm.es
http://www.romin.upm.es/rwirz

Centro de Automática y Robótica (CA.R.)
UPM-CISC

INDICE

INDICE

Niveles OSI



Wikipedia.org

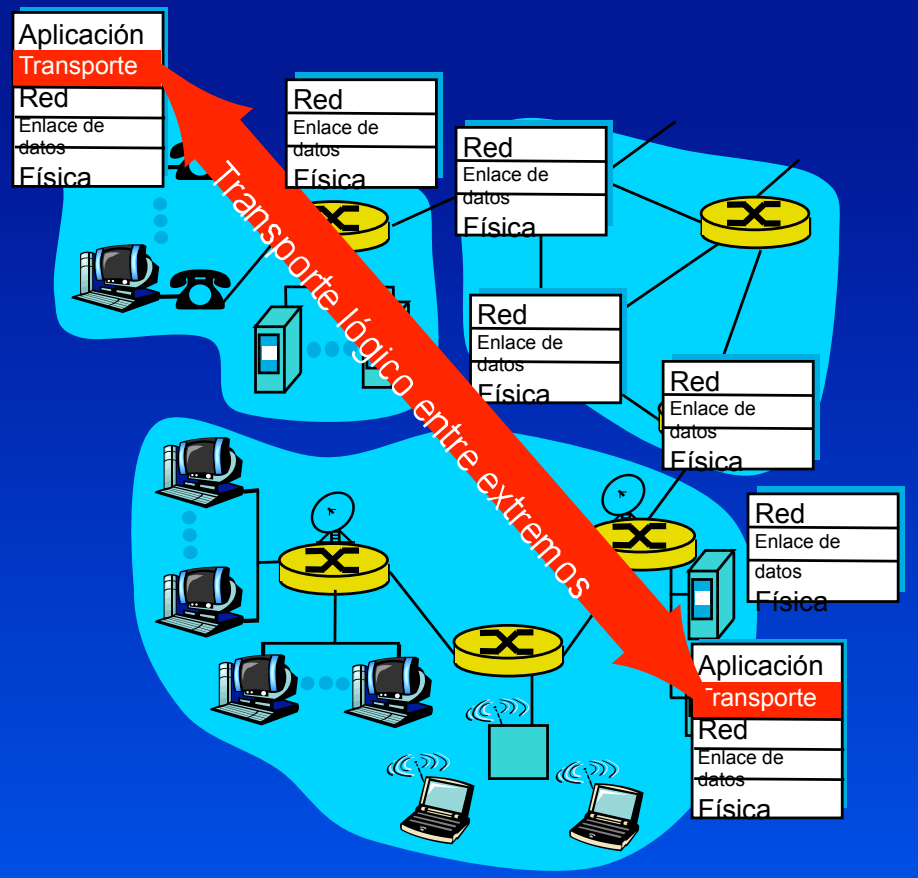
Capa Transporte

INTRODUCCION

Proporcionan *comunicación lógica* entre procesos de aplicación que se ejecutan en diferentes hosts.

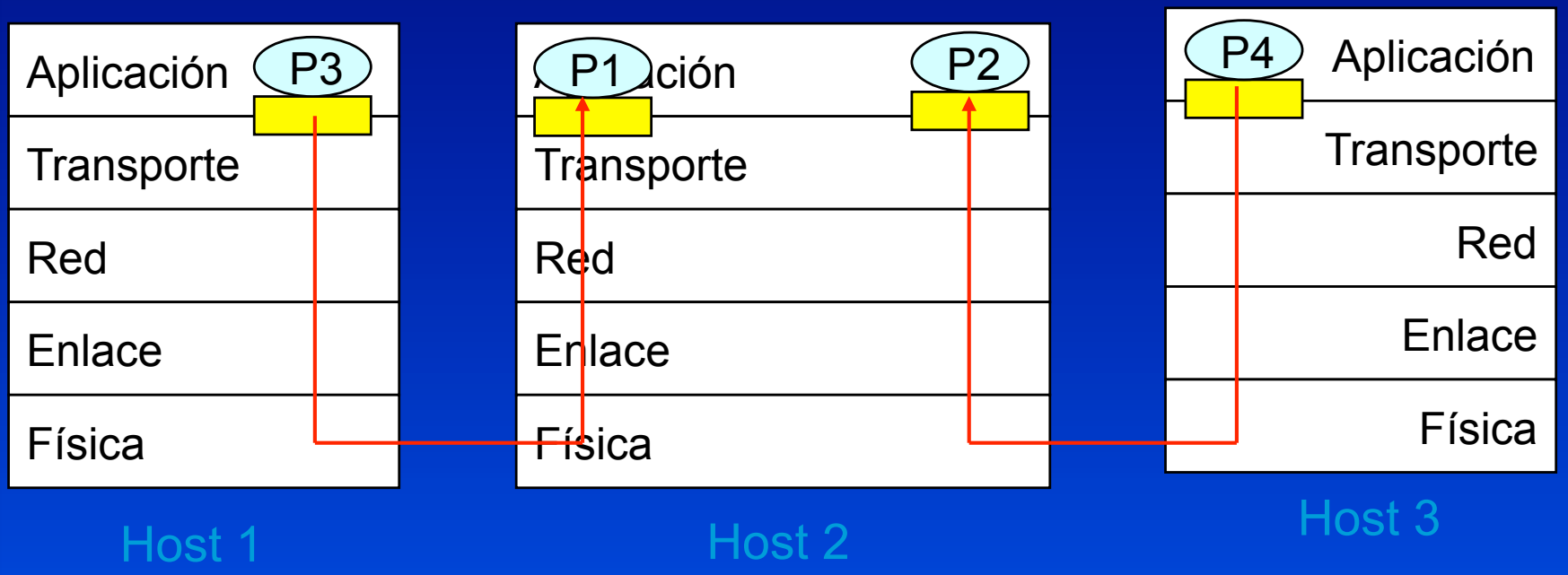
Los protocolos de transporte se ejecutan en sistemas finales.

- Lado emisor: fragmenta los mensajes de aplicación en segmentos y los pasa por la capa de red.
- Lado receptor: reagrupa los segmentos en mensajes y los pasa a una capa de aplicación.



Protocolos Transporte

INTRODUCCION



Trama UDP

INTRODUCCION

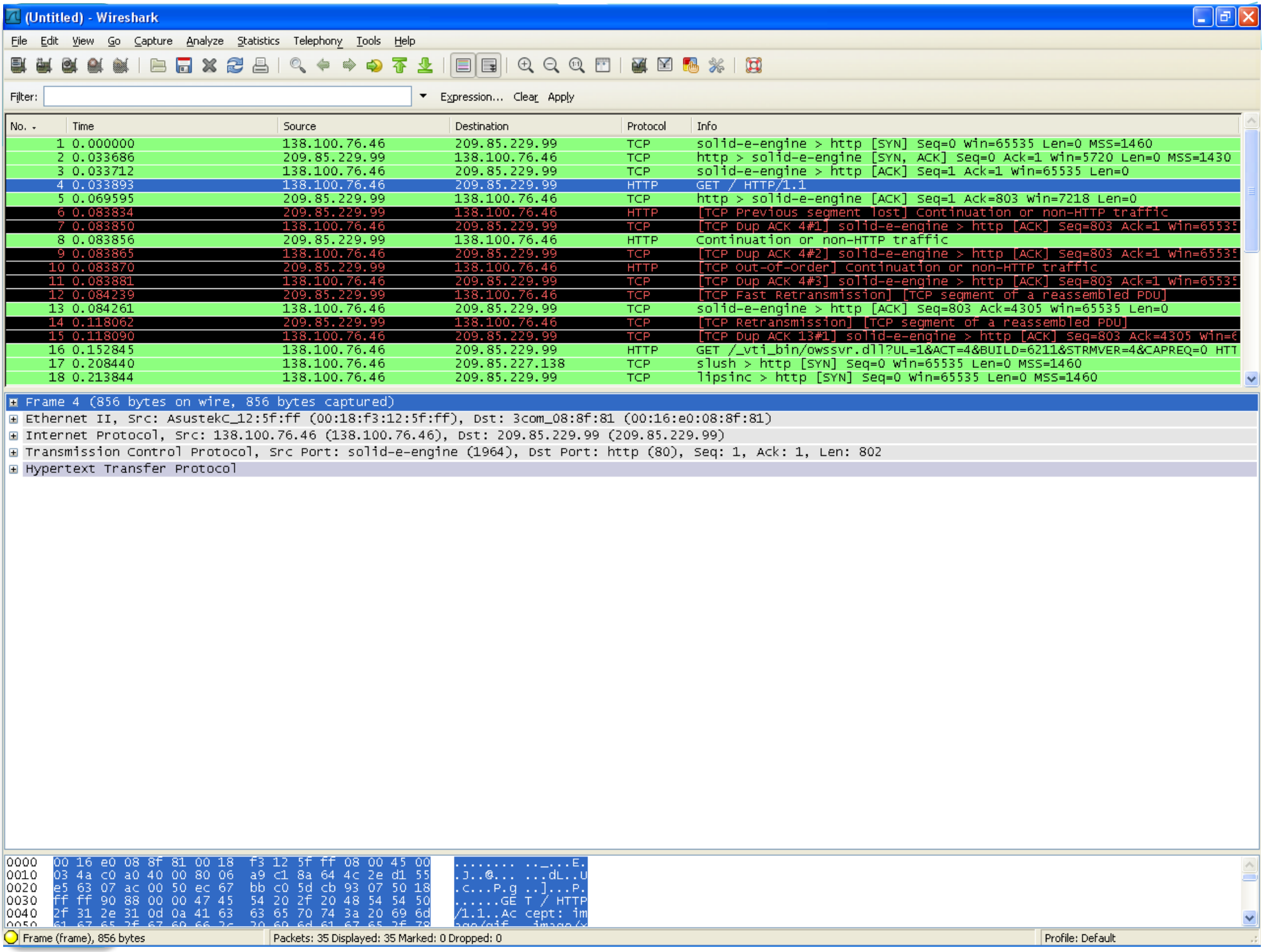
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Version				Header length				Type of service								Total length															
Identification																Flags			Fragment offset											
Time to life								Protocol								Header checksum															
Source IPv4 address																															
Destination IPv4 address																															
Options																															
Source port																Destination port															
Length																CheckSum															
Data….																															

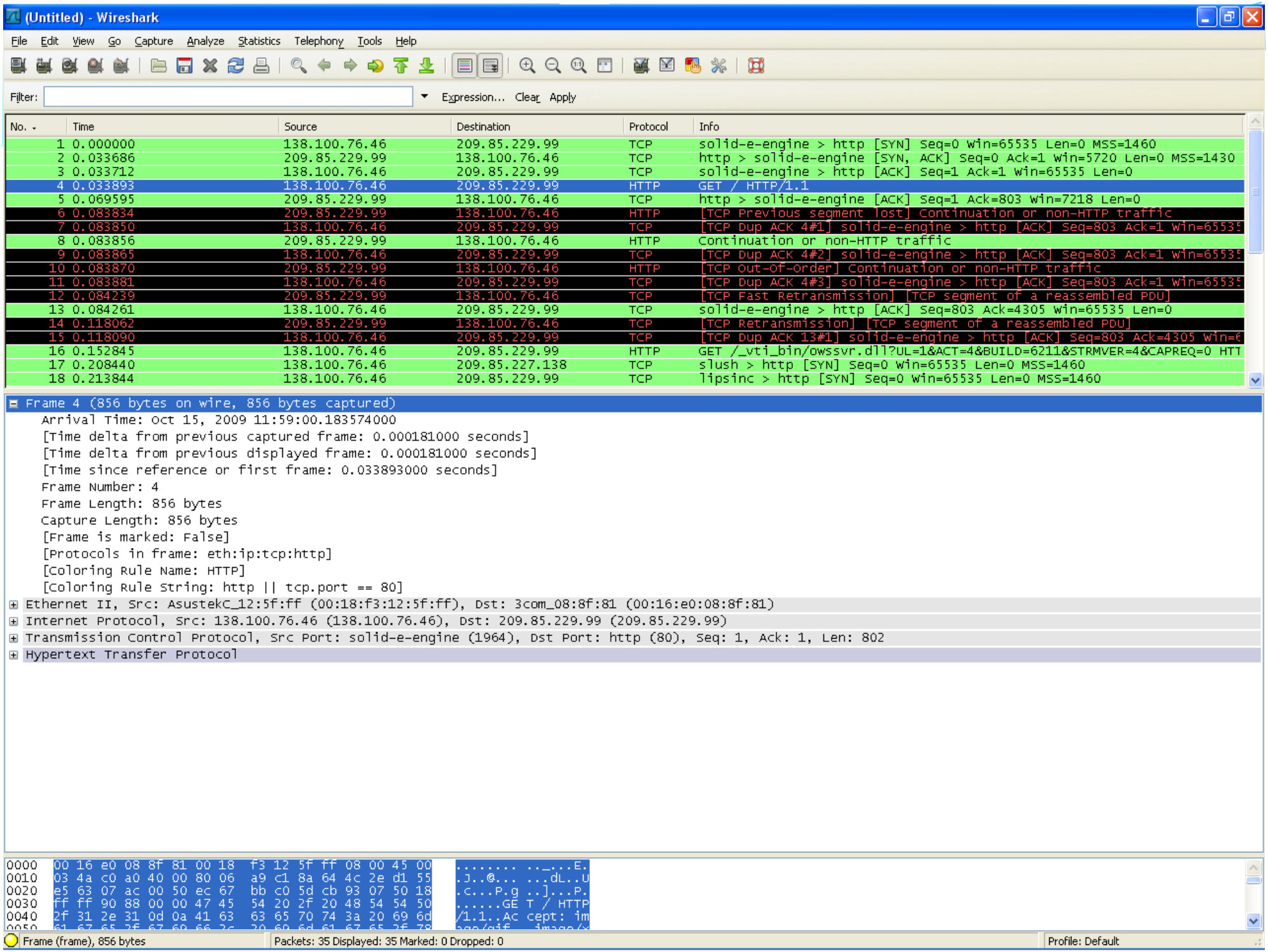
Practica

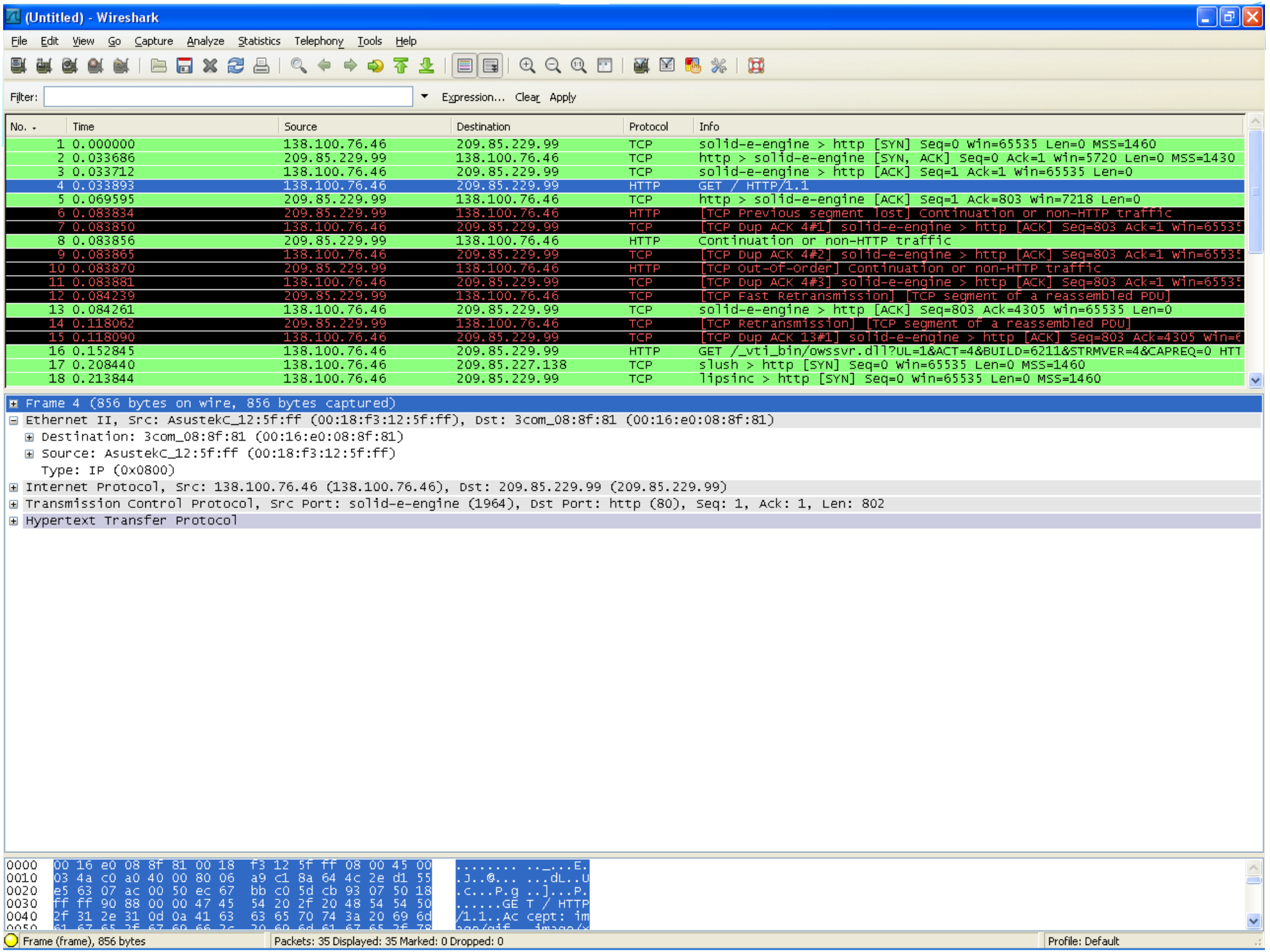
INTRODUCCION

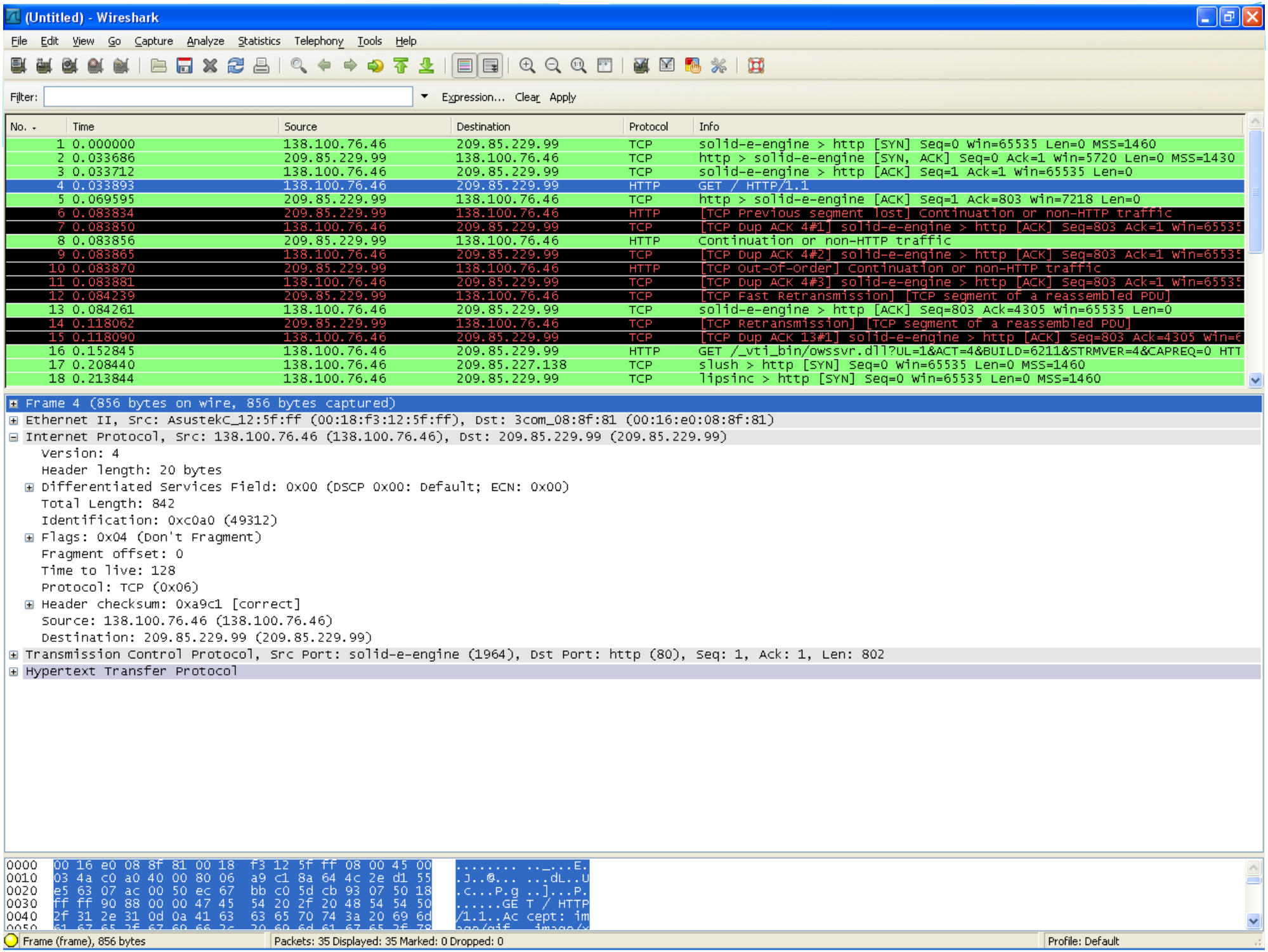
WireShark

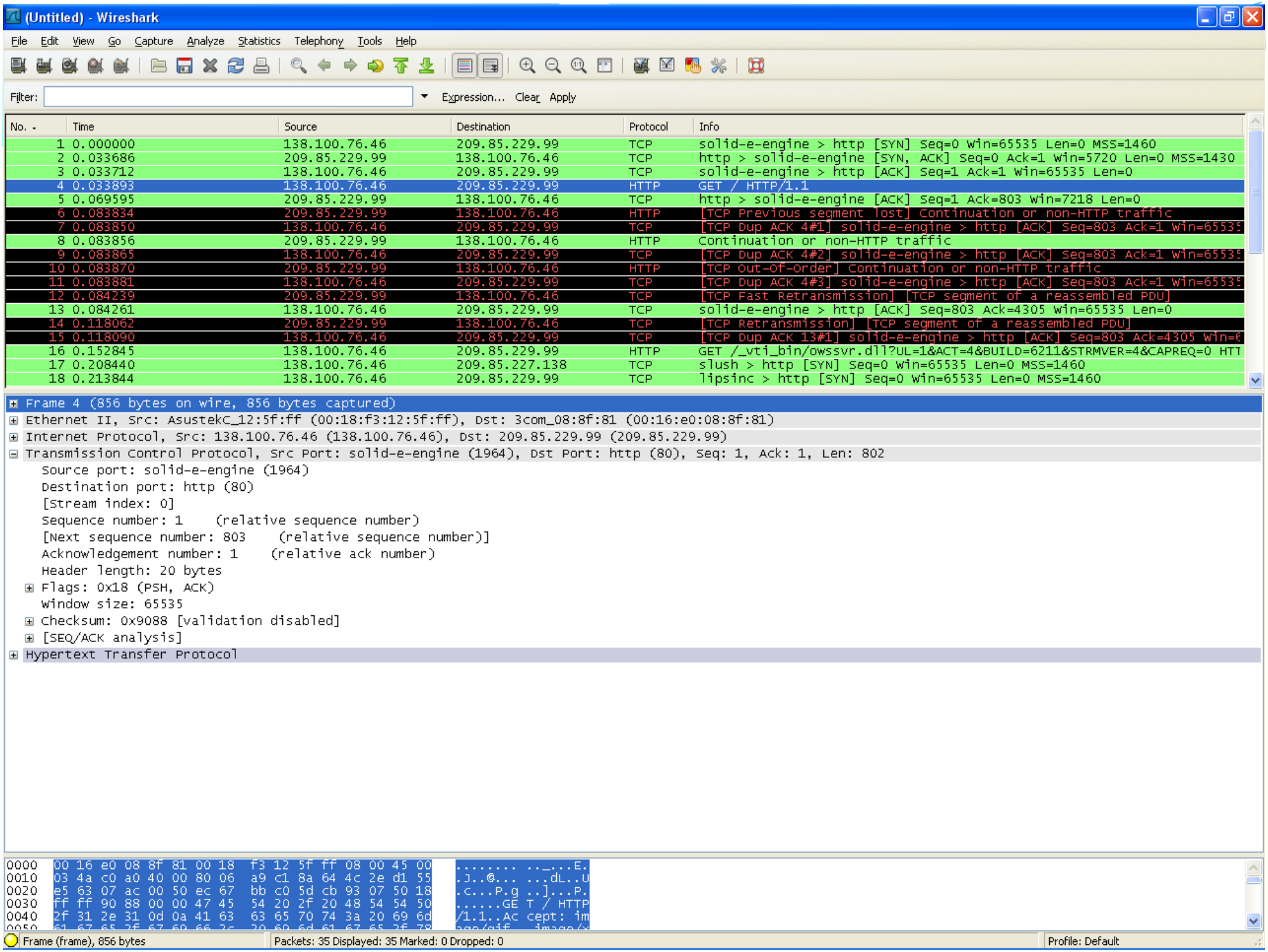
http://www.google.es

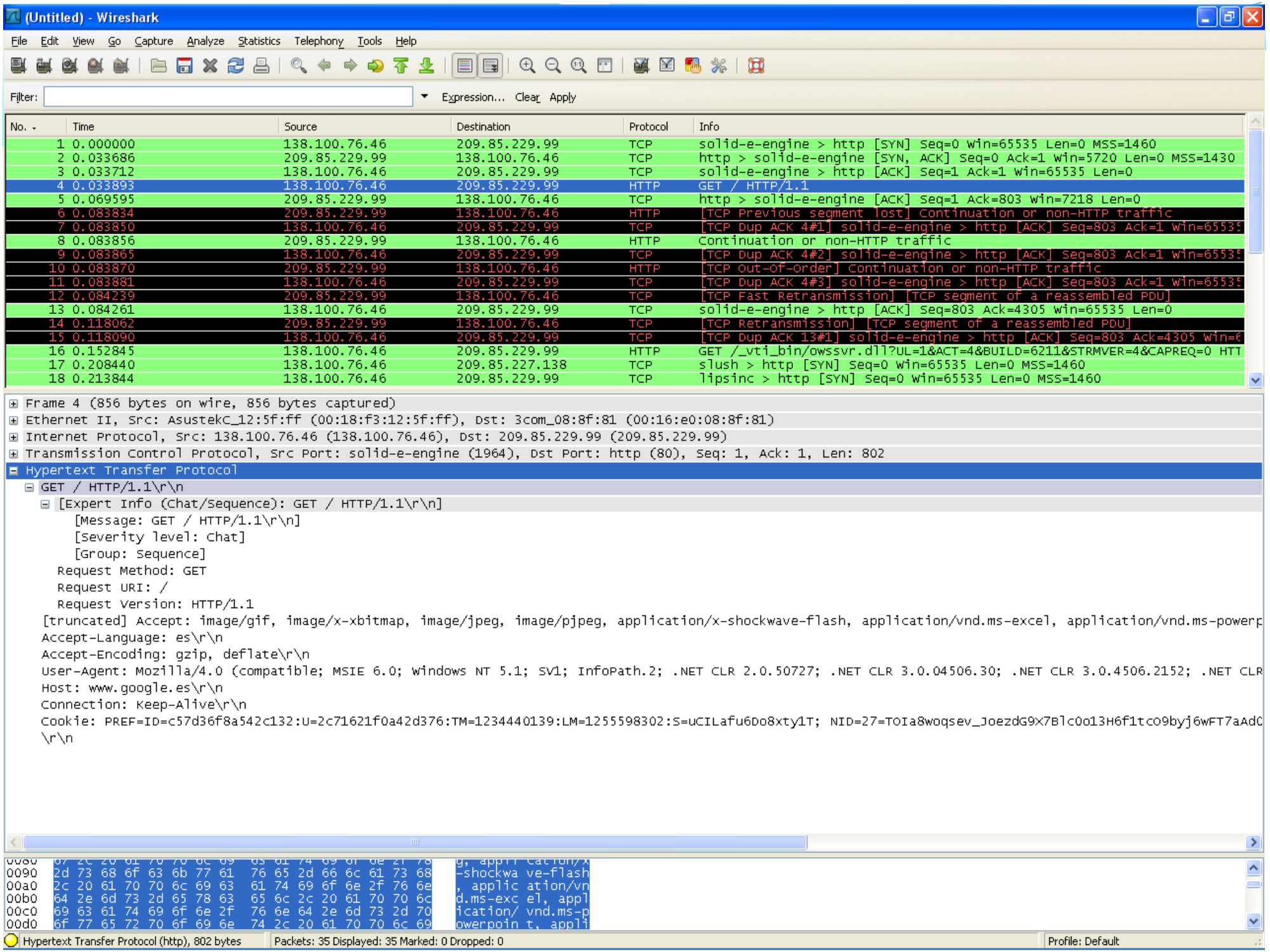

(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
2 0.033686 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
3 0.033712 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4 0.033893 138.100.76.46 209.85.229.99 HTTP GET / HTTP/1.1
5 0.069595 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [ACK] Seq=1 Ack=803 Win=7218 Len=0
6 0.083834 209.85.229.99 138.100.76.46 HTTP [TCP Previous segment lost] Continuation or non-HTTP traffic
7 0.083850 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#1] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
8 0.083856 209.85.229.99 138.100.76.46 HTTP Continuation or non-HTTP traffic
9 0.083865 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#2] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
10 0.083870 209.85.229.99 138.100.76.46 HTTP [TCP Out-Of-Order] Continuation or non-HTTP traffic
11 0.083881 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#3] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
12 0.084239 209.85.229.99 138.100.76.46 TCP [TCP Fast Retransmission] [TCP segment of a reassembled PDU]
13 0.084261 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=65535 Len=0
14 0.118062 209.85.229.99 138.100.76.46 TCP [TCP Retransmission] [TCP segment of a reassembled PDU]
15 0.118090 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 13#1] solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=6
16 0.152845 138.100.76.46 209.85.229.99 HTTP GET /_vti_bin/owssvr.dll?UL=1&ACT=4&BUILD=6211&STRMVER=4&CAPREQ=0 HTT
17 0.208440 138.100.76.46 209.85.227.138 TCP slush > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
18 0.213844 138.100.76.46 209.85.229.99 TCP lipsinc > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
Frame 4 (856 bytes on wire, 856 bytes captured)
Ethernet II, Src: AsustekC_12:5f:ff (00:18:f3:12:5f:ff), Dst: 3com_08:8f:81 (00:16:e0:08:8f:81)
Internet Protocol, Src: 138.100.76.46 (138.100.76.46), Dst: 209.85.229.99 (209.85.229.99)
Transmission Control Protocol, Src Port: solid-e-engine (1964), Dst Port: http (80), Seq: 1, Ack: 1, Len: 802
Hypertext Transfer Protocol
0000 00 16 e0 08 8f 81 00 18 f3 12 5f ff 08 00 45 00 ...._...E.
0010 03 4a c0 a0 40 00 80 06 a9 c1 8a 64 4c 2e d1 55 .J..@... ...dL..U
0020 e5 63 07 ac 00 50 ec 67 bb c0 5d cb 93 07 50 18 .c...P.g ..]...P.
0030 ff ff 90 88 00 00 47 45 54 20 2f 20 48 54 54 50 ......GE T / HTTP
0040 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d /1.1..Ac cept: im
Frame (frame), 856 bytes
Packets: 35 Displayed: 35 Marked: 0 Dropped: 0
Profile: Default


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
2 0.033686 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
3 0.033712 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4 0.033893 138.100.76.46 209.85.229.99 HTTP GET / HTTP/1.1
5 0.069595 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [ACK] Seq=1 Ack=803 Win=7218 Len=0
6 0.083834 209.85.229.99 138.100.76.46 HTTP [TCP Previous segment lost] Continuation or non-HTTP traffic
7 0.083850 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#1] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
8 0.083856 209.85.229.99 138.100.76.46 HTTP Continuation or non-HTTP traffic
9 0.083865 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#2] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
10 0.083870 209.85.229.99 138.100.76.46 HTTP [TCP Out-Of-Order] Continuation or non-HTTP traffic
11 0.083881 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#3] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
12 0.084239 209.85.229.99 138.100.76.46 TCP [TCP Fast Retransmission] [TCP segment of a reassembled PDU]
13 0.084261 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=65535 Len=0
14 0.118062 209.85.229.99 138.100.76.46 TCP [TCP Retransmission] [TCP segment of a reassembled PDU]
15 0.118090 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 13#1] solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=6
16 0.152845 138.100.76.46 209.85.229.99 HTTP GET /_vti_bin/owssvr.dll?UL=1&ACT=4&BUILD=6211&STRMVER=4&CAPREQ=0 HTT
17 0.208440 138.100.76.46 209.85.227.138 TCP slush > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
18 0.213844 138.100.76.46 209.85.229.99 TCP lipsinc > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
Frame 4 (856 bytes on wire, 856 bytes captured)
Arrival Time: Oct 15, 2009 11:59:00.183574000
[Time delta from previous captured frame: 0.000181000 seconds]
[Time delta from previous displayed frame: 0.000181000 seconds]
[Time since reference or first frame: 0.033893000 seconds]
Frame Number: 4
Frame Length: 856 bytes
Capture Length: 856 bytes
[Frame is marked: False]
[Protocols in frame: eth:ip:tcp:http]
[Coloring Rule Name: HTTP]
[Coloring Rule String: http || tcp.port == 80]
Ethernet II, Src: AsustekC_12:5f:ff (00:18:f3:12:5f:ff), Dst: 3com_08:8f:81 (00:16:e0:08:8f:81)
Internet Protocol, Src: 138.100.76.46 (138.100.76.46), Dst: 209.85.229.99 (209.85.229.99)
Transmission Control Protocol, Src Port: solid-e-engine (1964), Dst Port: http (80), Seq: 1, Ack: 1, Len: 802
Hypertext Transfer Protocol
0000 00 16 e0 08 8f 81 00 18 f3 12 5f ff 08 00 45 00 ...._...E.
0010 03 4a c0 a0 40 00 80 06 a9 c1 8a 64 4c 2e d1 55 .J..@... ...dL..U
0020 e5 63 07 ac 00 50 ec 67 bb c0 5d cb 93 07 50 18 .c...P.g ..]...P.
0030 ff ff 90 88 00 00 47 45 54 20 2f 20 48 54 54 50 ......GE T / HTTP
0040 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d /1.1..Ac cept: im
Frame (frame), 856 bytes
Packets: 35 Displayed: 35 Marked: 0 Dropped: 0
Profile: Default


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
2 0.033686 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
3 0.033712 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4 0.033893 138.100.76.46 209.85.229.99 HTTP GET / HTTP/1.1
5 0.069595 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [ACK] Seq=1 Ack=803 Win=7218 Len=0
6 0.083834 209.85.229.99 138.100.76.46 HTTP [TCP Previous segment lost] Continuation or non-HTTP traffic
7 0.083850 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#1] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
8 0.083856 209.85.229.99 138.100.76.46 HTTP Continuation or non-HTTP traffic
9 0.083865 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#2] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
10 0.083870 209.85.229.99 138.100.76.46 HTTP [TCP Out-Of-Order] Continuation or non-HTTP traffic
11 0.083881 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#3] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
12 0.084239 209.85.229.99 138.100.76.46 TCP [TCP Fast Retransmission] [TCP segment of a reassembled PDU]
13 0.084261 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=65535 Len=0
14 0.118062 209.85.229.99 138.100.76.46 TCP [TCP Retransmission] [TCP segment of a reassembled PDU]
15 0.118090 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 13#1] solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=6
16 0.152845 138.100.76.46 209.85.229.99 HTTP GET /_vti_bin/owssvr.dll?UL=1&ACT=4&BUILD=6211&STRMVER=4&CAPREQ=0 HTT
17 0.208440 138.100.76.46 209.85.227.138 TCP slush > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
18 0.213844 138.100.76.46 209.85.229.99 TCP lipsinc > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
Frame 4 (856 bytes on wire, 856 bytes captured)
Ethernet II, Src: AsustekC_12:5f:ff (00:18:f3:12:5f:ff), Dst: 3com_08:8f:81 (00:16:e0:08:8f:81)
Destination: 3com_08:8f:81 (00:16:e0:08:8f:81)
Source: AsustekC_12:5f:ff (00:18:f3:12:5f:ff)
Type: IP (0x0800)
Internet Protocol, Src: 138.100.76.46 (138.100.76.46), Dst: 209.85.229.99 (209.85.229.99)
Transmission Control Protocol, Src Port: solid-e-engine (1964), Dst Port: http (80), Seq: 1, Ack: 1, Len: 802
Hypertext Transfer Protocol
0000 00 16 e0 08 8f 81 00 18 f3 12 5f ff 08 00 45 00 ...._...E.
0010 03 4a c0 a0 40 00 80 06 a9 c1 8a 64 4c 2e d1 55 .J..@... ...dL..U
0020 e5 63 07 ac 00 50 ec 67 bb c0 5d cb 93 07 50 18 .c...P.g ..]...P.
0030 ff ff 90 88 00 00 47 45 54 20 2f 20 48 54 54 50 ......GE T / HTTP
0040 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d /1.1..Ac cept: im
Frame (frame), 856 bytes
Packets: 35 Displayed: 35 Marked: 0 Dropped: 0
Profile: Default


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
2 0.033686 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
3 0.033712 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4 0.033893 138.100.76.46 209.85.229.99 HTTP GET / HTTP/1.1
5 0.069595 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [ACK] Seq=1 Ack=803 Win=7218 Len=0
6 0.083834 209.85.229.99 138.100.76.46 HTTP [TCP Previous segment lost] Continuation or non-HTTP traffic
7 0.083850 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#1] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
8 0.083856 209.85.229.99 138.100.76.46 HTTP Continuation or non-HTTP traffic
9 0.083865 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#2] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
10 0.083870 209.85.229.99 138.100.76.46 HTTP [TCP Out-Of-Order] Continuation or non-HTTP traffic
11 0.083881 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#3] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
12 0.084239 209.85.229.99 138.100.76.46 TCP [TCP Fast Retransmission] [TCP segment of a reassembled PDU]
13 0.084261 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=65535 Len=0
14 0.118062 209.85.229.99 138.100.76.46 TCP [TCP Retransmission] [TCP segment of a reassembled PDU]
15 0.118090 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 13#1] solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=6
16 0.152845 138.100.76.46 209.85.229.99 HTTP GET /_vti_bin/owssvr.dll?UL=1&ACT=4&BUILD=6211&STRMVER=4&CAPREQ=0 HTT
17 0.208440 138.100.76.46 209.85.227.138 TCP slush > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
18 0.213844 138.100.76.46 209.85.229.99 TCP lipsinc > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
Frame 4 (856 bytes on wire, 856 bytes captured)
Ethernet II, Src: AsustekC_12:5f:ff (00:18:f3:12:5f:ff), Dst: 3com_08:8f:81 (00:16:e0:08:8f:81)
Internet Protocol, Src: 138.100.76.46 (138.100.76.46), Dst: 209.85.229.99 (209.85.229.99)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 842
Identification: 0xc0a0 (49312)
Flags: 0x04 (Don't Fragment)
Fragment offset: 0
Time to live: 128
Protocol: TCP (0x06)
Header checksum: 0xa9c1 [correct]
Source: 138.100.76.46 (138.100.76.46)
Destination: 209.85.229.99 (209.85.229.99)
Transmission Control Protocol, Src Port: solid-e-engine (1964), Dst Port: http (80), Seq: 1, Ack: 1, Len: 802
Hypertext Transfer Protocol
0000 00 16 e0 08 8f 81 00 18 f3 12 5f ff 08 00 45 00 ...._...E.
0010 03 4a c0 a0 40 00 80 06 a9 c1 8a 64 4c 2e d1 55 .J..@... ...dL..U
0020 e5 63 07 ac 00 50 ec 67 bb c0 5d cb 93 07 50 18 .c...P.g ..]...P.
0030 ff ff 90 88 00 00 47 45 54 20 2f 20 48 54 54 50 ......GE T / HTTP
0040 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d /1.1..Ac cept: im
Frame (frame), 856 bytes
Packets: 35 Displayed: 35 Marked: 0 Dropped: 0
Profile: Default


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
2 0.033686 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
3 0.033712 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4 0.033893 138.100.76.46 209.85.229.99 HTTP GET / HTTP/1.1
5 0.069595 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [ACK] Seq=1 Ack=803 Win=7218 Len=0
6 0.083834 209.85.229.99 138.100.76.46 HTTP [TCP Previous segment lost] Continuation or non-HTTP traffic
7 0.083850 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#1] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
8 0.083856 209.85.229.99 138.100.76.46 HTTP Continuation or non-HTTP traffic
9 0.083865 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#2] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
10 0.083870 209.85.229.99 138.100.76.46 HTTP [TCP Out-Of-Order] Continuation or non-HTTP traffic
11 0.083881 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#3] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
12 0.084239 209.85.229.99 138.100.76.46 TCP [TCP Fast Retransmission] [TCP segment of a reassembled PDU]
13 0.084261 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=65535 Len=0
14 0.118062 209.85.229.99 138.100.76.46 TCP [TCP Retransmission] [TCP segment of a reassembled PDU]
15 0.118090 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 13#1] solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=6
16 0.152845 138.100.76.46 209.85.229.99 HTTP GET /_vti_bin/owssvr.dll?UL=1&ACT=4&BUILD=6211&STRMVER=4&CAPREQ=0 HTT
17 0.208440 138.100.76.46 209.85.227.138 TCP slush > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
18 0.213844 138.100.76.46 209.85.229.99 TCP lipsinc > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
Frame 4 (856 bytes on wire, 856 bytes captured)
Ethernet II, Src: AsustekC_12:5f:ff (00:18:f3:12:5f:ff), Dst: 3com_08:8f:81 (00:16:e0:08:8f:81)
Internet Protocol, Src: 138.100.76.46 (138.100.76.46), Dst: 209.85.229.99 (209.85.229.99)
Transmission Control Protocol, Src Port: solid-e-engine (1964), Dst Port: http (80), Seq: 1, Ack: 1, Len: 802
Source port: solid-e-engine (1964)
Destination port: http (80)
[Stream index: 0]
Sequence number: 1 (relative sequence number)
[Next sequence number: 803 (relative sequence number)]
Acknowledgement number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x18 (PSH, ACK)
Window size: 65535
Checksum: 0x9088 [validation disabled]
[SEQ/ACK analysis]
Hypertext Transfer Protocol
0000 00 16 e0 08 8f 81 00 18 f3 12 5f ff 08 00 45 00 ...._...E.
0010 03 4a c0 a0 40 00 80 06 a9 c1 8a 64 4c 2e d1 55 .J..@... ...dL..U
0020 e5 63 07 ac 00 50 ec 67 bb c0 5d cb 93 07 50 18 .c...P.g ..]...P.
0030 ff ff 90 88 00 00 47 45 54 20 2f 20 48 54 54 50 ......GE T / HTTP
0040 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 69 6d /1.1..Ac cept: im
Frame (frame), 856 bytes
Packets: 35 Displayed: 35 Marked: 0 Dropped: 0
Profile: Default


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
2 0.033686 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430
3 0.033712 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4 0.033893 138.100.76.46 209.85.229.99 HTTP GET / HTTP/1.1
5 0.069595 209.85.229.99 138.100.76.46 TCP http > solid-e-engine [ACK] Seq=1 Ack=803 Win=7218 Len=0
6 0.083834 209.85.229.99 138.100.76.46 HTTP [TCP Previous segment lost] Continuation or non-HTTP traffic
7 0.083850 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#1] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
8 0.083856 209.85.229.99 138.100.76.46 HTTP Continuation or non-HTTP traffic
9 0.083865 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#2] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
10 0.083870 209.85.229.99 138.100.76.46 HTTP [TCP Out-Of-Order] Continuation or non-HTTP traffic
11 0.083881 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 4#3] solid-e-engine > http [ACK] Seq=803 Ack=1 Win=6553
12 0.084239 209.85.229.99 138.100.76.46 TCP [TCP Fast Retransmission] [TCP segment of a reassembled PDU]
13 0.084261 138.100.76.46 209.85.229.99 TCP solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=65535 Len=0
14 0.118062 209.85.229.99 138.100.76.46 TCP [TCP Retransmission] [TCP segment of a reassembled PDU]
15 0.118090 138.100.76.46 209.85.229.99 TCP [TCP Dup ACK 13#1] solid-e-engine > http [ACK] Seq=803 Ack=4305 Win=6
16 0.152845 138.100.76.46 209.85.229.99 HTTP GET /_vti_bin/owssvr.dll?UL=1&ACT=4&BUILD=6211&STRMVER=4&CAPREQ=0 HTT
17 0.208440 138.100.76.46 209.85.227.138 TCP slush > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
18 0.213844 138.100.76.46 209.85.229.99 TCP lipsinc > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460
Frame 4 (856 bytes on wire, 856 bytes captured)
Ethernet II, Src: AsustekC_12:5f:ff (00:18:f3:12:5f:ff), Dst: 3com_08:8f:81 (00:16:e0:08:8f:81)
Internet Protocol, Src: 138.100.76.46 (138.100.76.46), Dst: 209.85.229.99 (209.85.229.99)
Transmission Control Protocol, Src Port: solid-e-engine (1964), Dst Port: http (80), Seq: 1, Ack: 1, Len: 802
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
[Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]
[Message: GET / HTTP/1.1\r\n]
[Severity level: Chat]
[Group: Sequence]
Request Method: GET
Request URI: /
Request Version: HTTP/1.1
[truncated] Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerp
Accept-Language: es\r\n
Accept-Encoding: gzip, deflate\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.2; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.4506.2152; .NET CLR
Host: www.google.es\r\n
Connection: Keep-Alive\r\n
Cookie: PREF=ID=c57d36f8a542c132:U=2c71621f0a42d376:TM=1234440139:LM=1255598302:S=uCILafu6Do8xty1T; NID=27=TOIa8woqsev_JoezdG9X7Blc0o13H6f1tcO9byj6wFT7aAdC
\r\n
0090 2d 73 68 6f 63 6b 77 61 76 65 2d 66 6c 61 73 68 -shockwa ve-flash
00a0 2c 20 61 70 70 6c 69 63 61 74 69 6f 6e 2f 76 6e , applic ation/vn
00b0 64 2e 6d 73 2d 65 78 63 65 6c 2c 20 61 70 70 6c d.ms-exc el, appl
00c0 69 63 61 74 69 6f 6e 2f 76 6e 64 2e 6d 73 2d 70 ication/ vnd.ms-p
00d0 6f 77 65 72 70 6f 69 6e 74 2c 20 61 70 70 6c 69 owerpoin t, appli
Hypertext Transfer Protocol (http), 802 bytes
Packets: 35 Displayed: 35 Marked: 0 Dropped: 0
Profile: Default

Socket

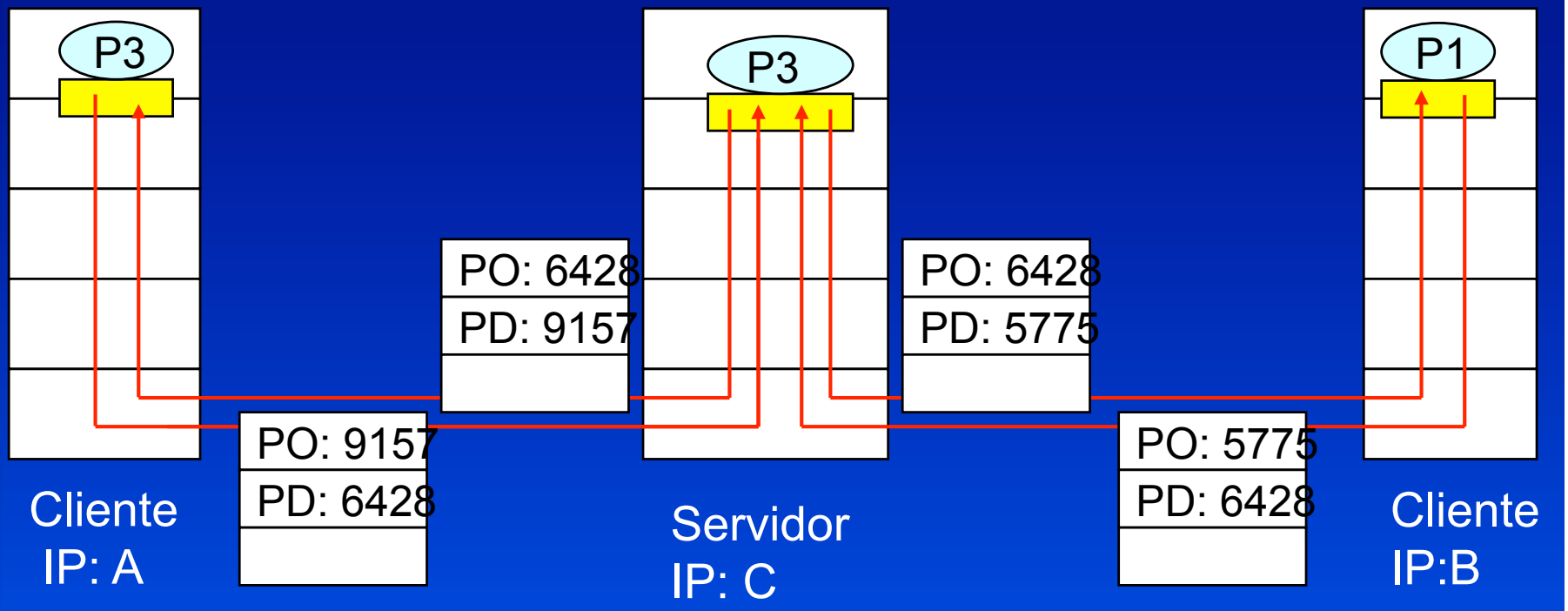
INTRODUCCION

El Socket (o Puerto) se define como la “puerta” por la que entrarán y saldrán datos del ordenador.

Puerto	Servicio o aplicación
21	FTP
23	Telnet
25	SMTP
53	Sistema de nombre de dominio
63	Whois
70	Gopher
79	Finger
80	HTTP
110	POP3
119	NNTP

¡ Del 0 al 1023 son reservados!

Sockets



El puerto origen proporciona “dirección de retorno”

Sockets

Tres tipos de sockets:

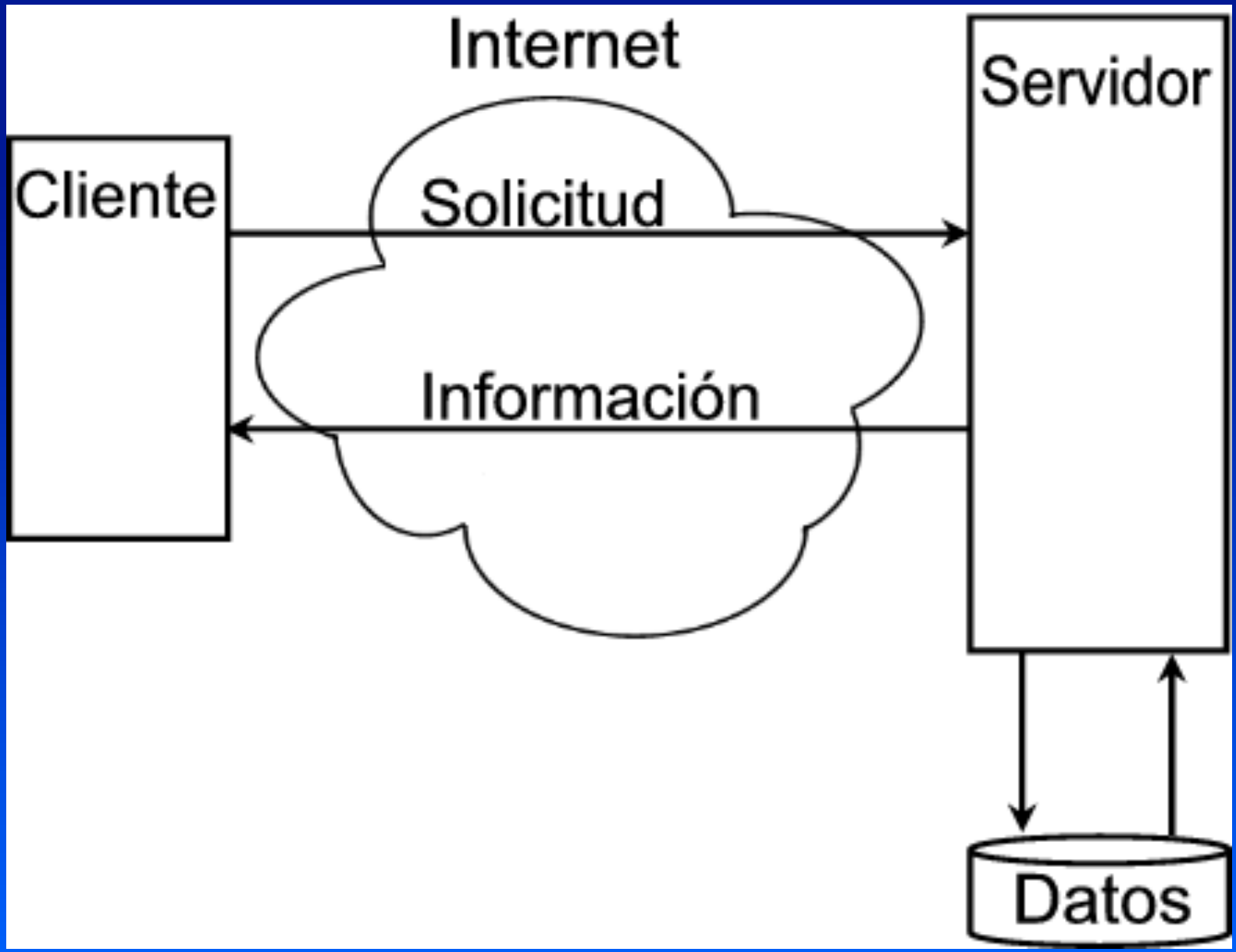
- Datagrama: Sin conexión (UDP)
- Stream: Con conexión (TCP)
- Raw: Bajo nivel

A la hora de programar:

- Sockets Bloqueantes
- Sockets no Bloqueantes

(Depende del lenguaje de programación)

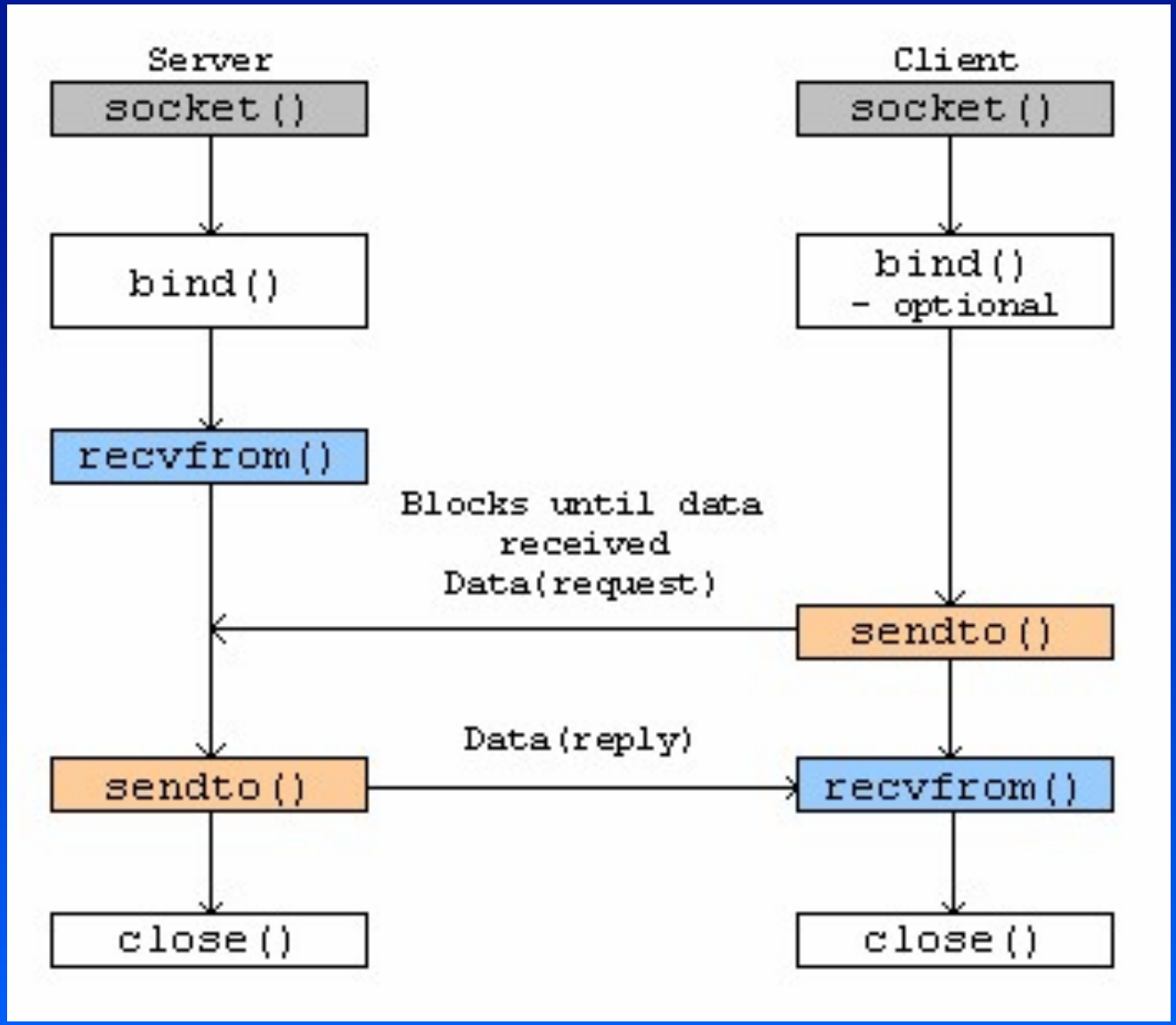
CLIENTE/SERVIDOR



Programación UDP

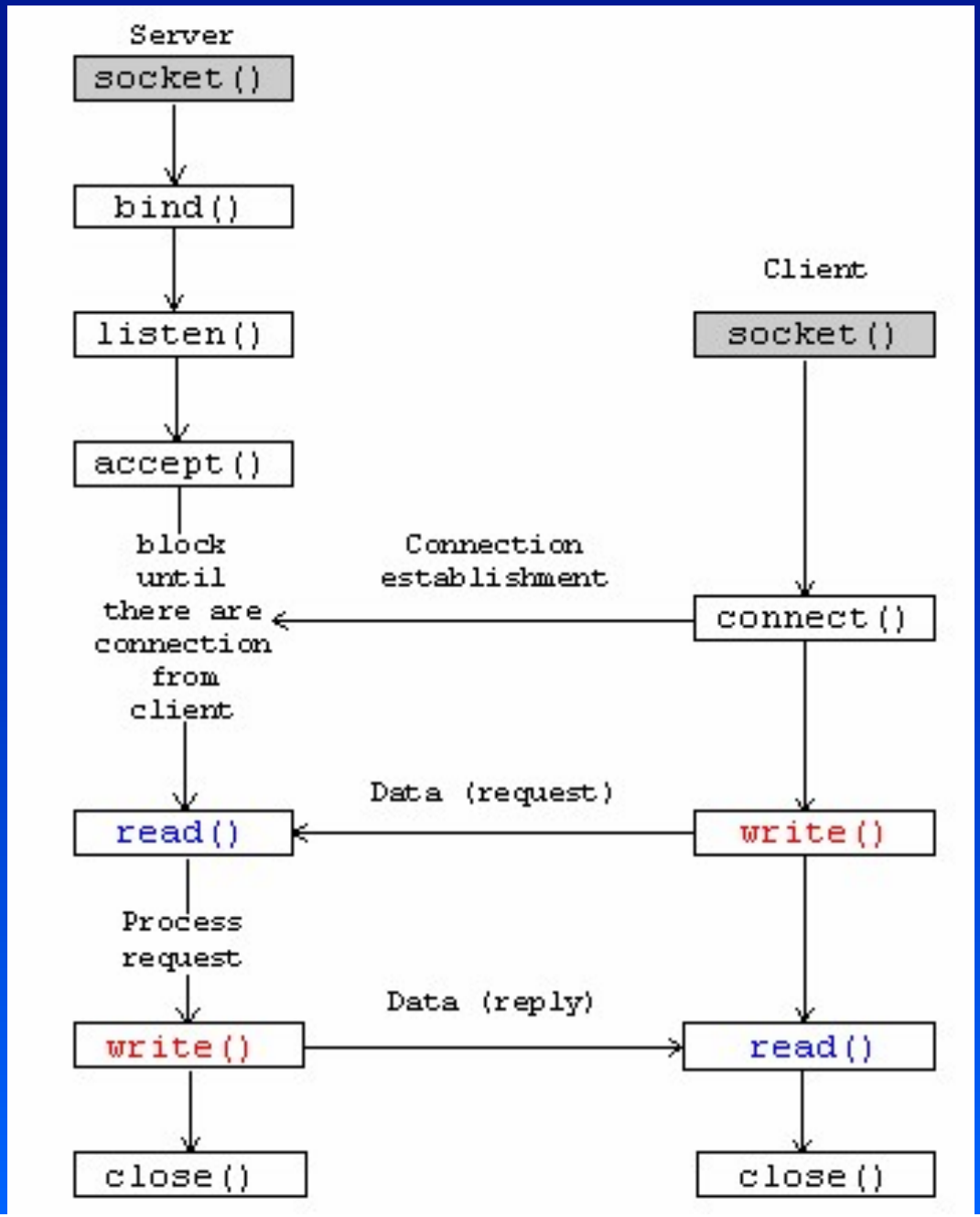
INTRODUCCION

UDP



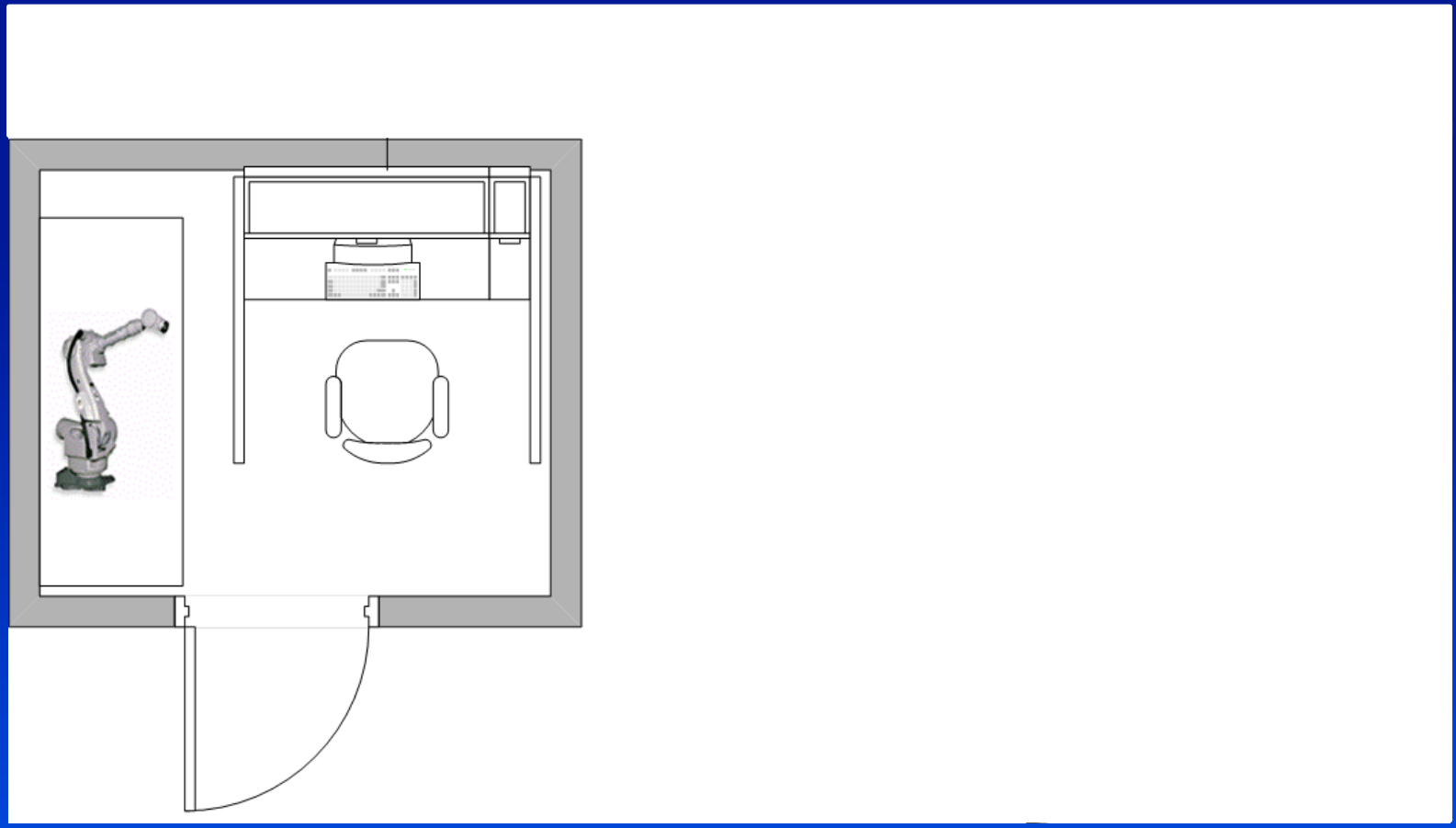
Programación TCP

INTRODUCCION



TCP

Introducción



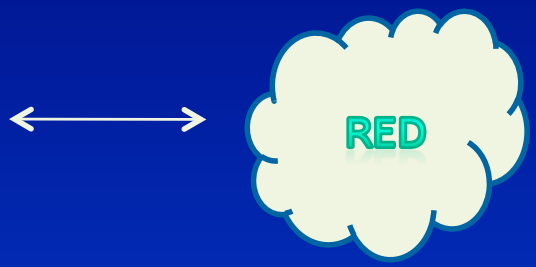
Tipos de Ordenes

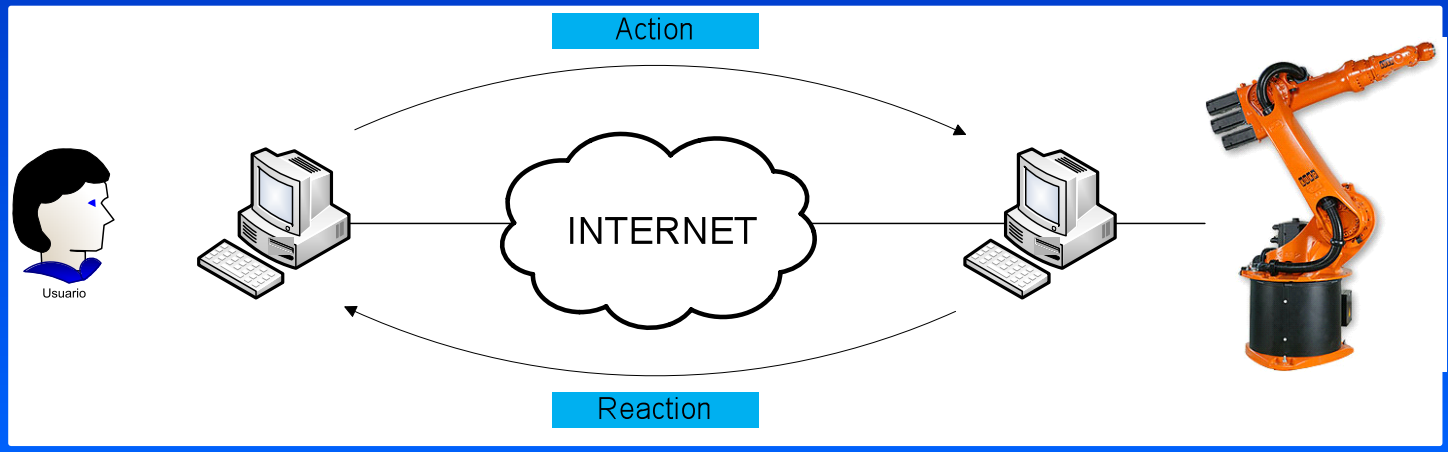
Ordenes de alto nivel: open grasp

Ordenes de bajo nivel: movejoin(128.,128, 128, 128,0,1)

Introducción

INTRODUCCION

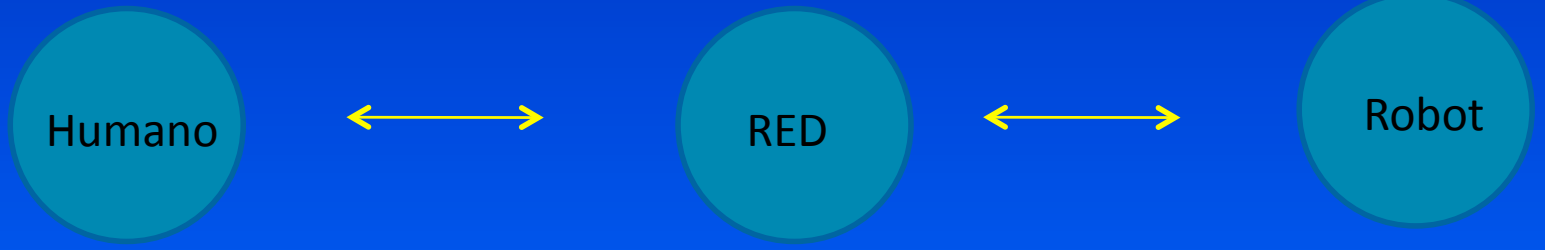




Introducción

Sistema Maestro/Esclavo

RED

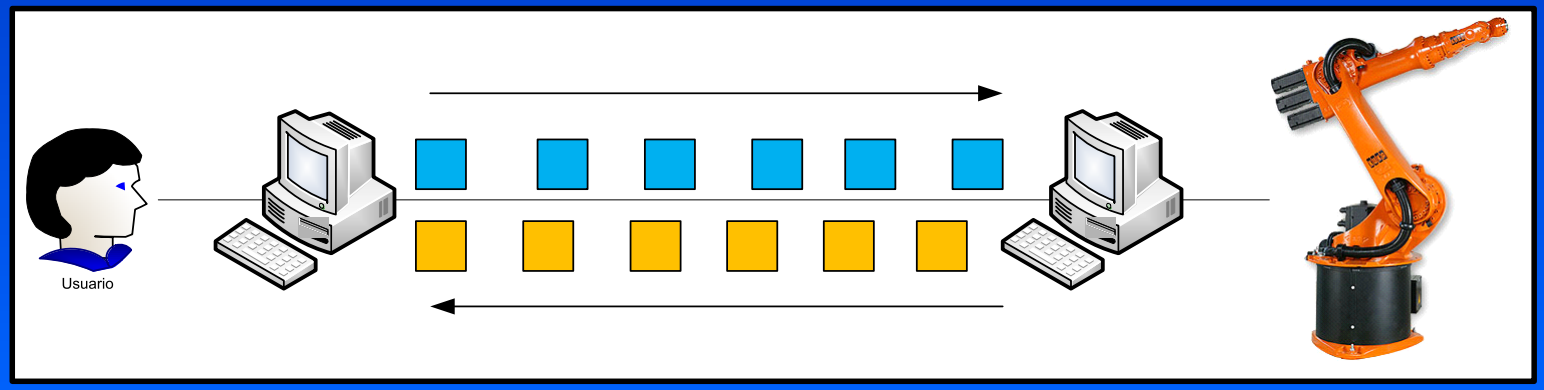


UDP

Telecontrol bilateral M/S

ETHERNET

¿TCP? ¿UDP?



INDICE

INTRODUCCION

UDP

CARACTERISTICAS

UDP

- Protocolo de transporte de Internet "sin adornos" y "muy limitado".
- Servicio de "mayor rendimiento"; los segmentos UDP puede que:
  - Se pierdan.
  - Se entreguen sin orden a la aplicación.
- *Sin conexión:*
  - No hay acuerdo entre el emisor UDP y el receptor.
  - Cada segmento UDP se maneja independientemente de los otros.

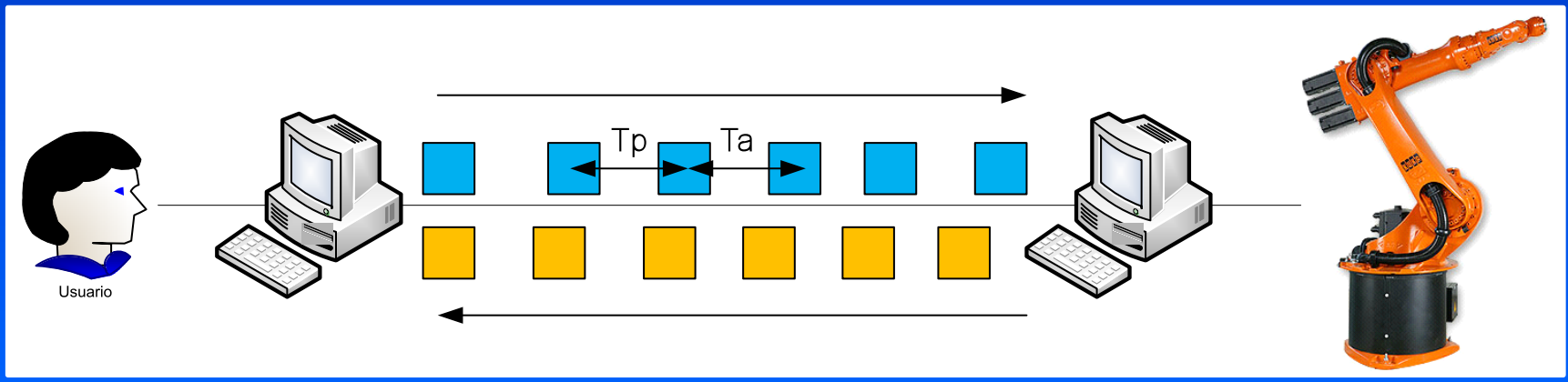
¿Por qué existe un UDP?

- Sin establecimiento de la conexión (lo que puede añadir un retardo).
- Simple: sin estado de conexión en el emisor y el receptor.
- Pequeño encabezamiento de segmento.
- Sin control de congestión: el UDP puede descongestionarse tan rápido como se desee.

Telecontrol bilateral M/S

UDP

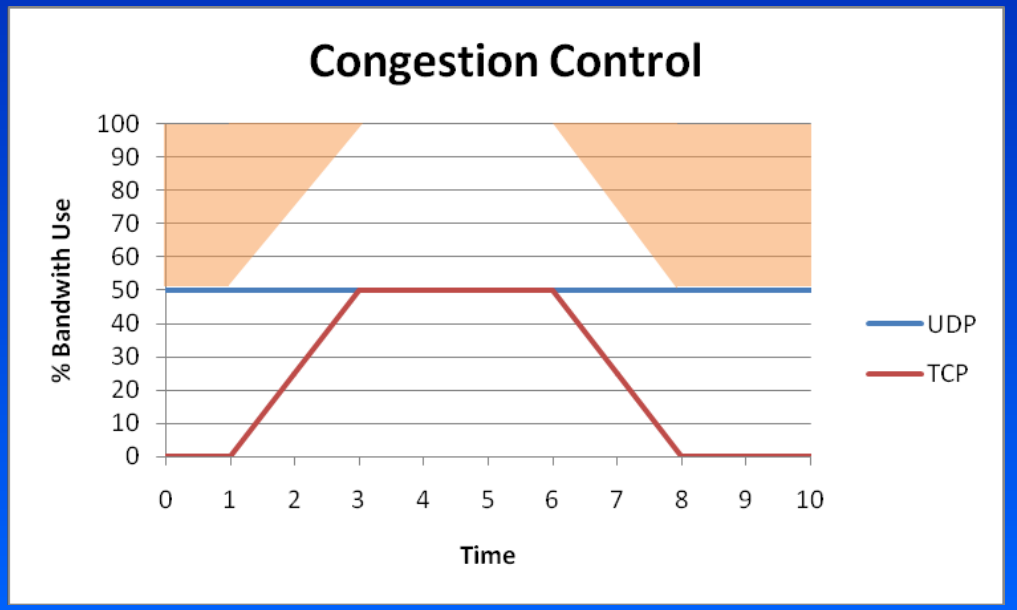
Basado en Ratio
Tp≈Ta



Telecontrol bilateral M/S

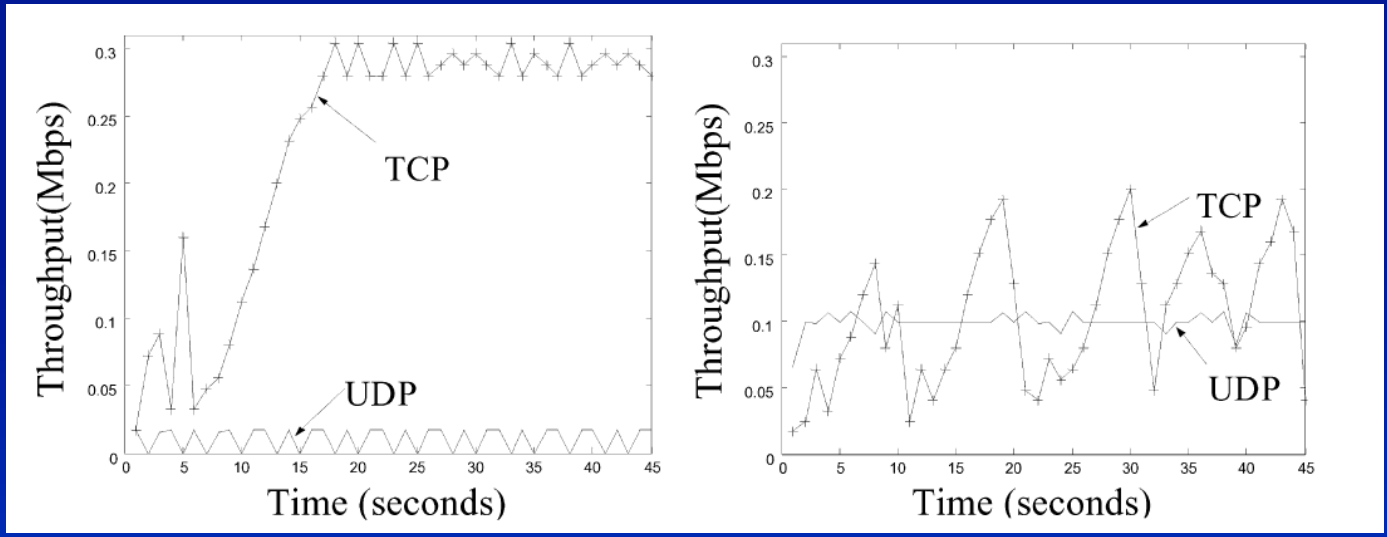
UDP

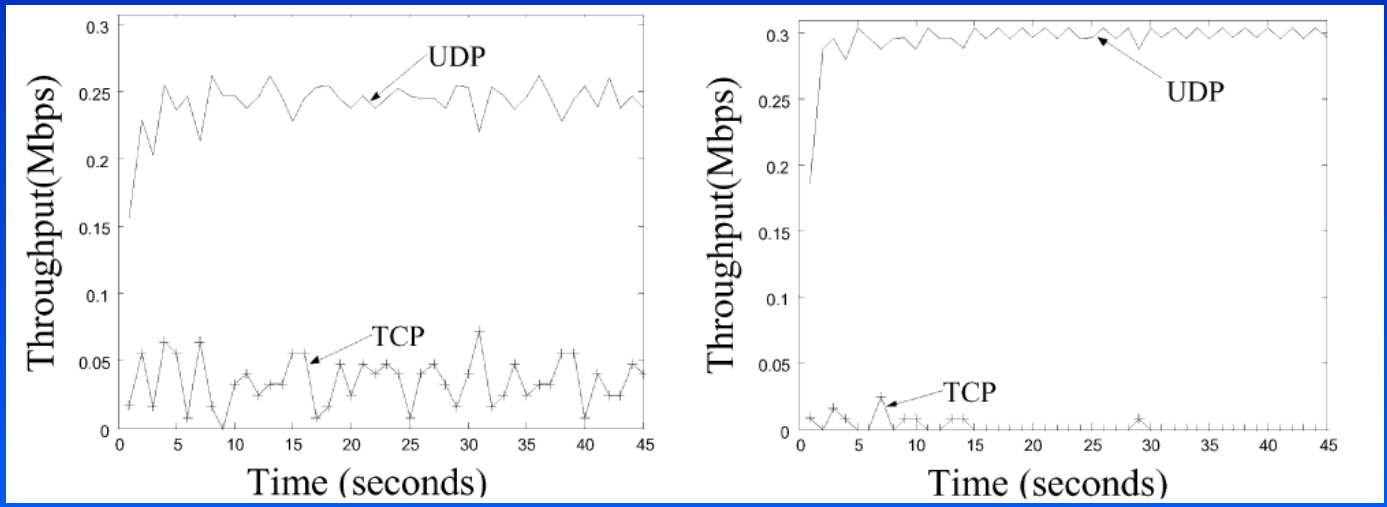
¿UDP?

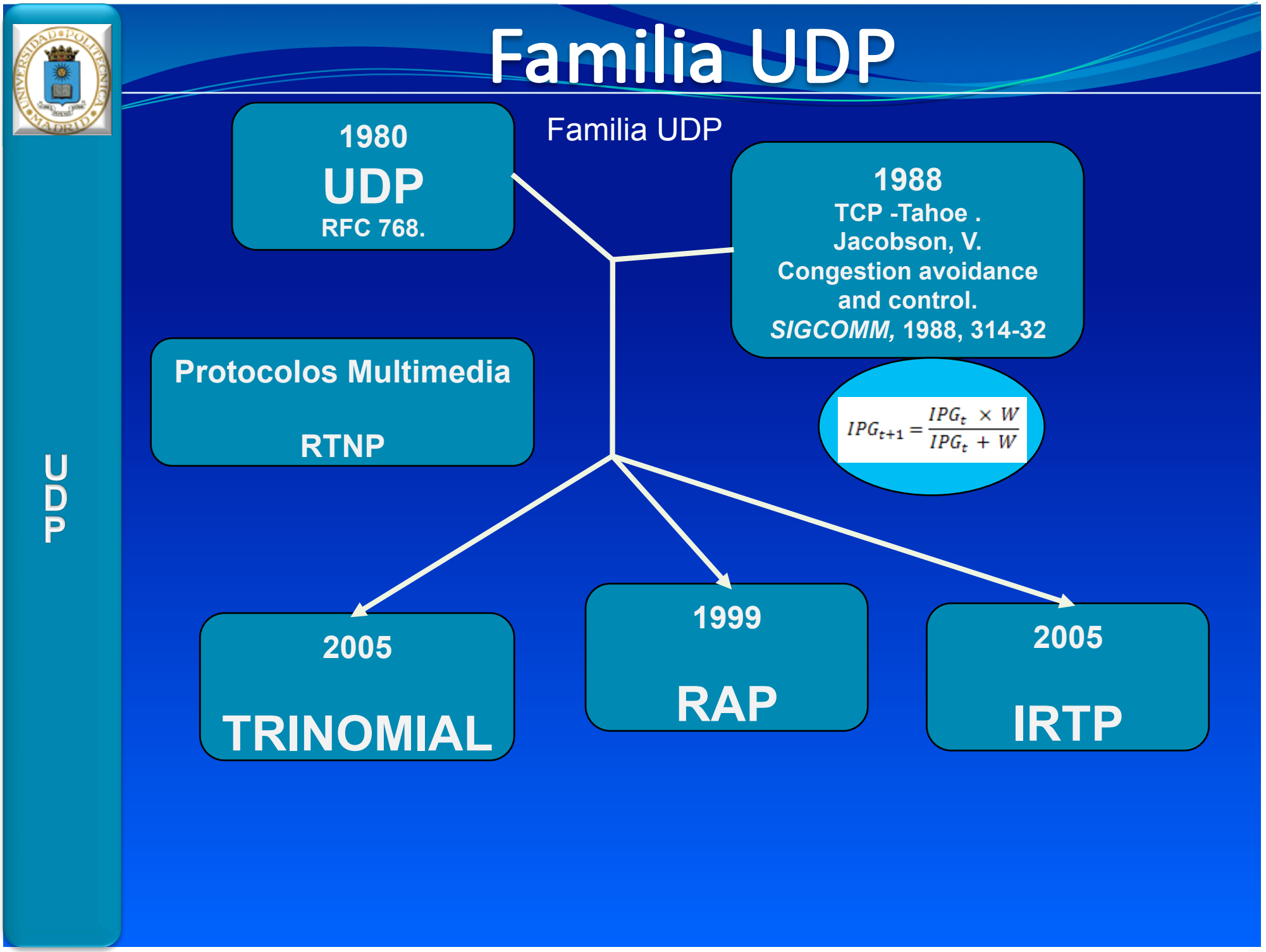


Problema UDP

UDP






Familia UDP
UDP
Familia UDP
1980
UDP
RFC 768.
1988
TCP -Tahoe .
Jacobson, V.
Congestion avoidance
and control.
SIGCOMM, 1988, 314-32
$IPG_{t+1} = \frac{IPG_t \times W}{IPG_t + W}$
Protocolos Multimedia
RTNP
2005
TRINOMIAL
1999
RAP
2005
IRTP

Trinomial

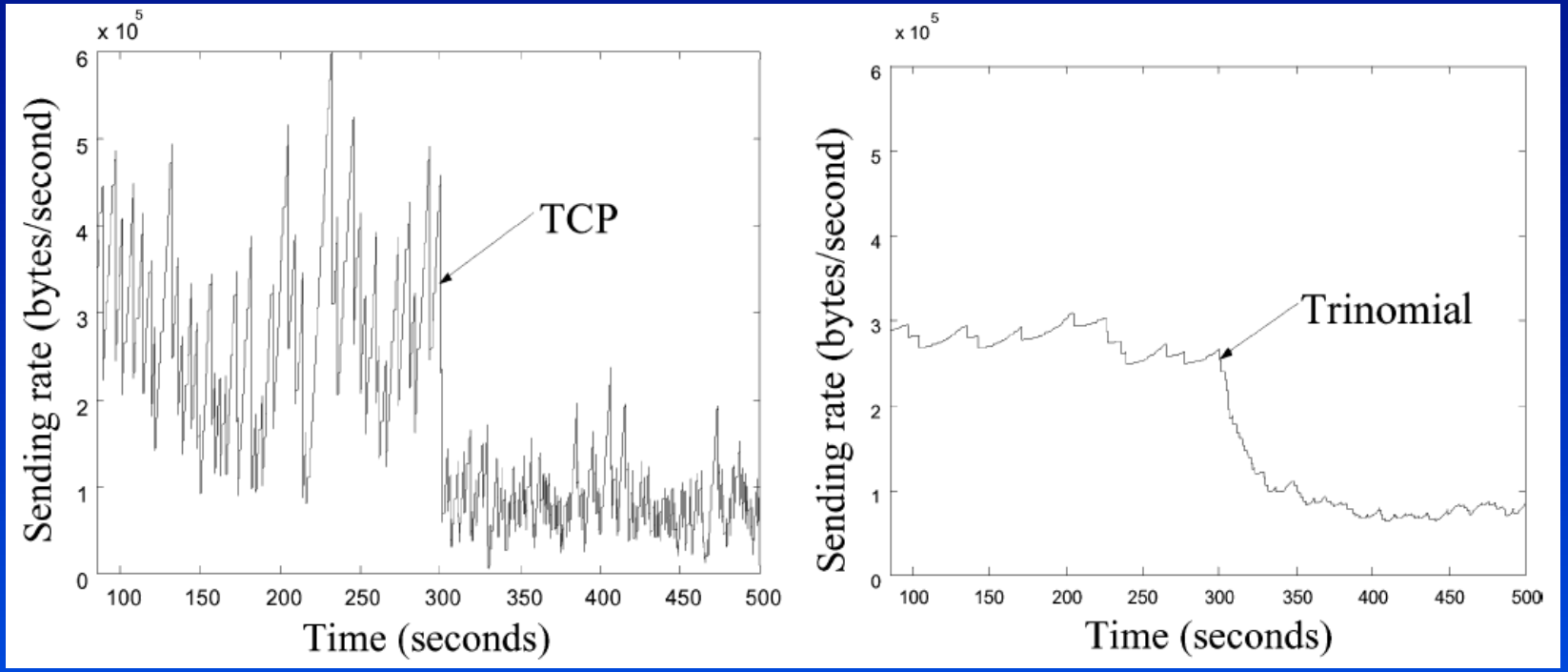
UDP

- Basado en UDP.
- Se usa ACK.
- El “sender” controla el protocolo.
- El IPG es calculado usando la formula de Jacobson (con el RTT)
- Aumenta el ratio muy rapido.

$$IPG_{t+1} = \frac{IPG_t \times W}{IPG_t + W}$$

Trinomial

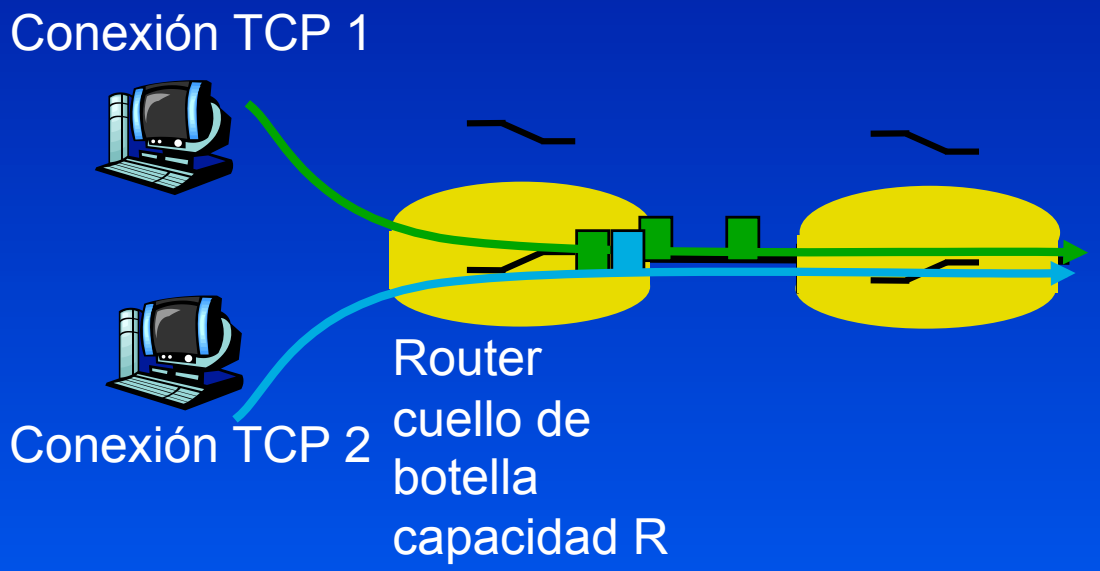
UDP



INDICE

TCP-Friendly (Imparcialidad)

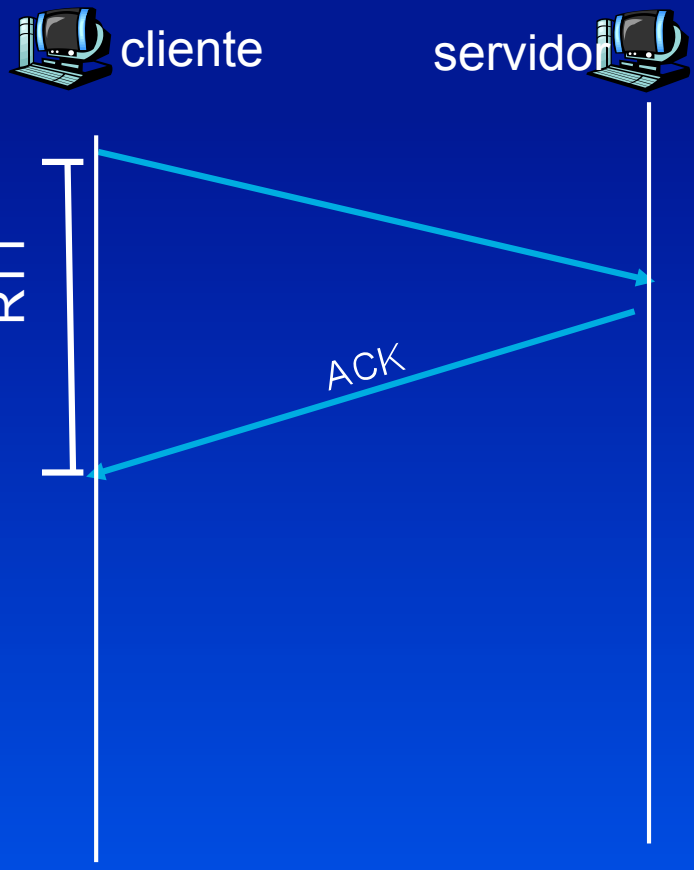
Objetivo de la imparcialidad: Si K sesiones TCP comparte el mismo enlace cuello de botella de banda R, cada una de ellas debe tener una tasa media de R/K.



RTT

RTT

ROUND TRIP TIME



TCP

TCP

- Punto a punto:
  - Un emisor, un receptor.
- Flujo de *bytes* fiable y ordenado:
  - Sin “límites de mensajes”.
- Entubado:
  - La congestión del TCP y el control de flujo determinan el tamaño de la ventana.
- *Almacén en el búfer de envío y recepción.*
- Datos *full duplex*:
  - Flujo de datos bidireccional en la misma conexión.
- Orientado a la conexión:
  - Acuerdo (intercambio del control de mensajes) inicaliza los estados del emisor y el receptor antes del intercambio de datos.
- Control de flujo:
  - El emisor no sobrecargará al receptor.

NUMERO SECUENCIA

TCP

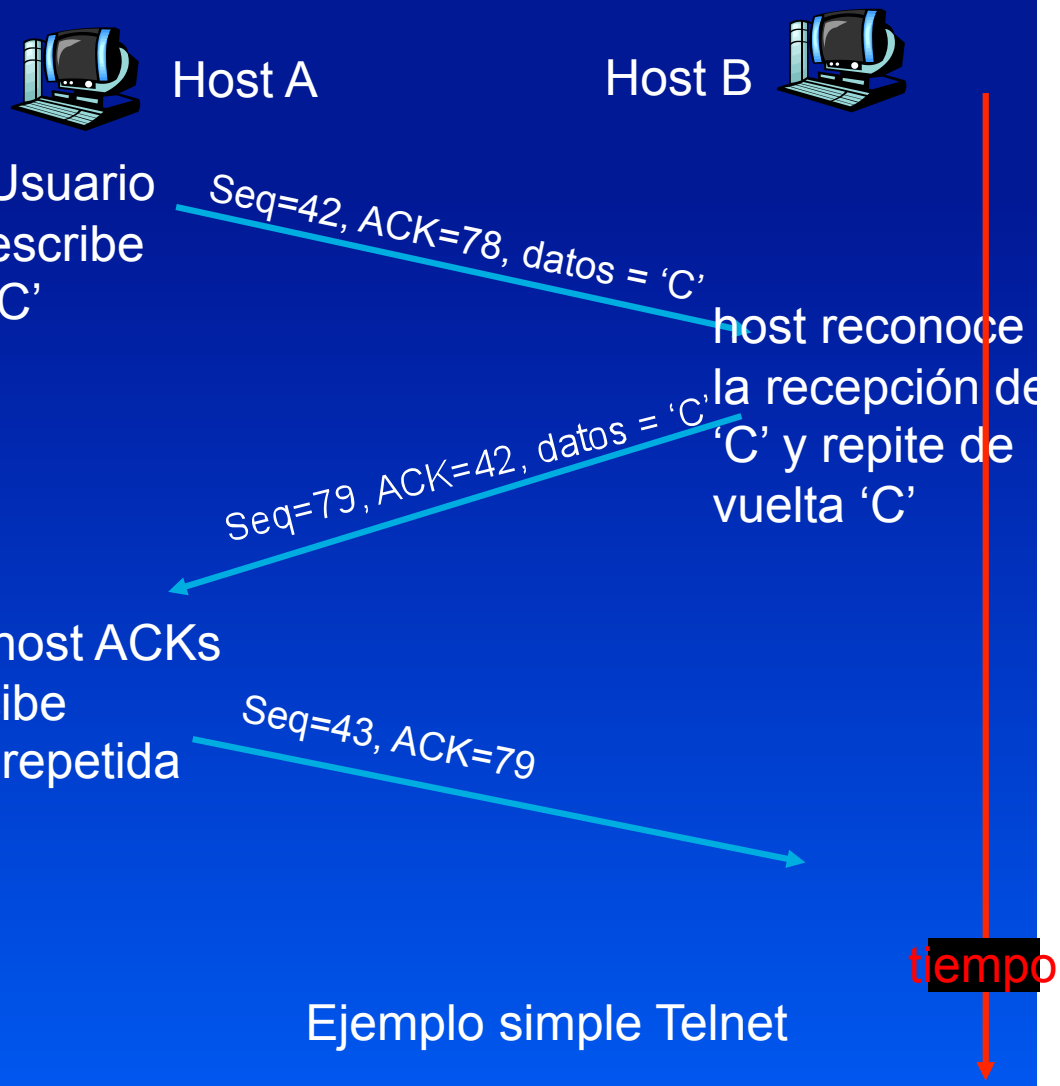
Números de secuencia:

"Número" de corriente de bytes del primer byte en los datos del segmento.

Números ACK:

Número de secuencia del siguiente byte que se espera del otro lado.

ACK acumulativo.



Ejemplo simple Telnet

CONEXION

TCP

Acuerdo en tres fases:

Paso 1: El host del cliente envía un segmento SYN TCP al servidor.

- Especifica el número de secuencia inicial.
- Sin datos.

Paso 2: El host del servidor recibe el SYN, y responde con un segmento SYNACK.

- El servidor asigna los búferes.
- Especifica el número de secuencia inicial del servidor.

Paso 3: El cliente recibe el SYNACK, responde con un segmento ACK, que puede contener datos.

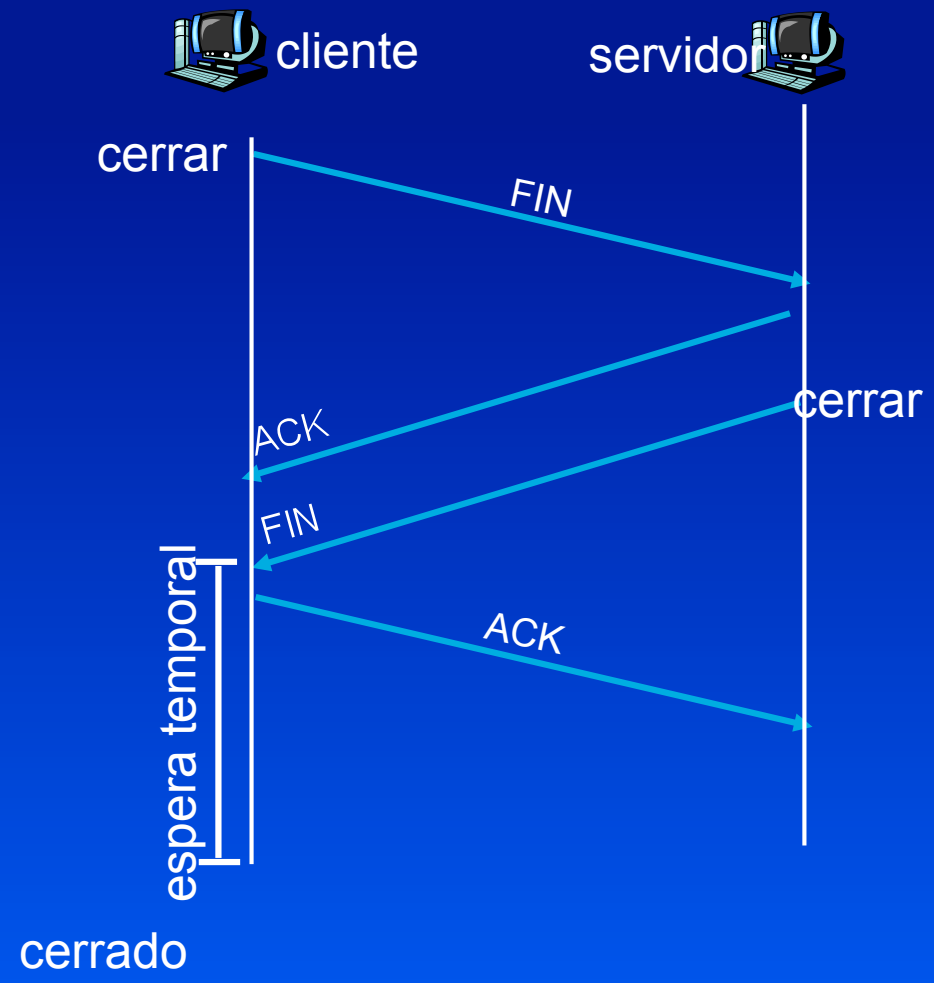
DESCONEXION

TCP

Cerrando una conexión:

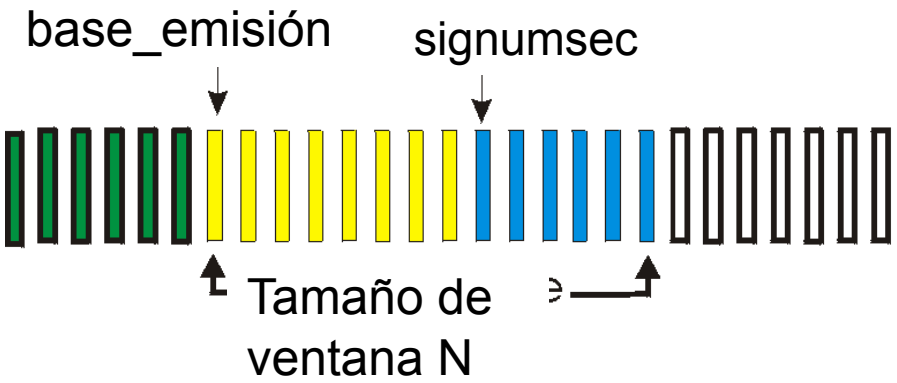
Paso 1: El sistema final del cliente envía un segmento de control TCP FIN al servidor.

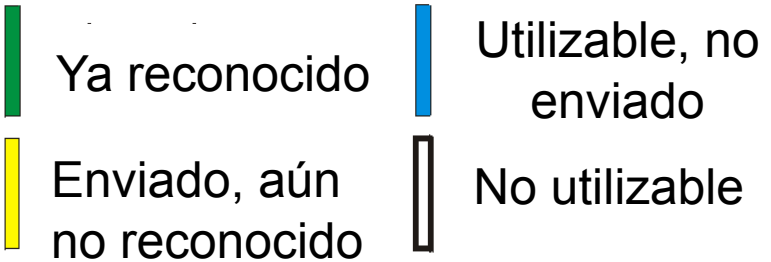
Step 2: El servidor recibe el FIN y responde con un ACK. Cierra la conexión y envía FIN.



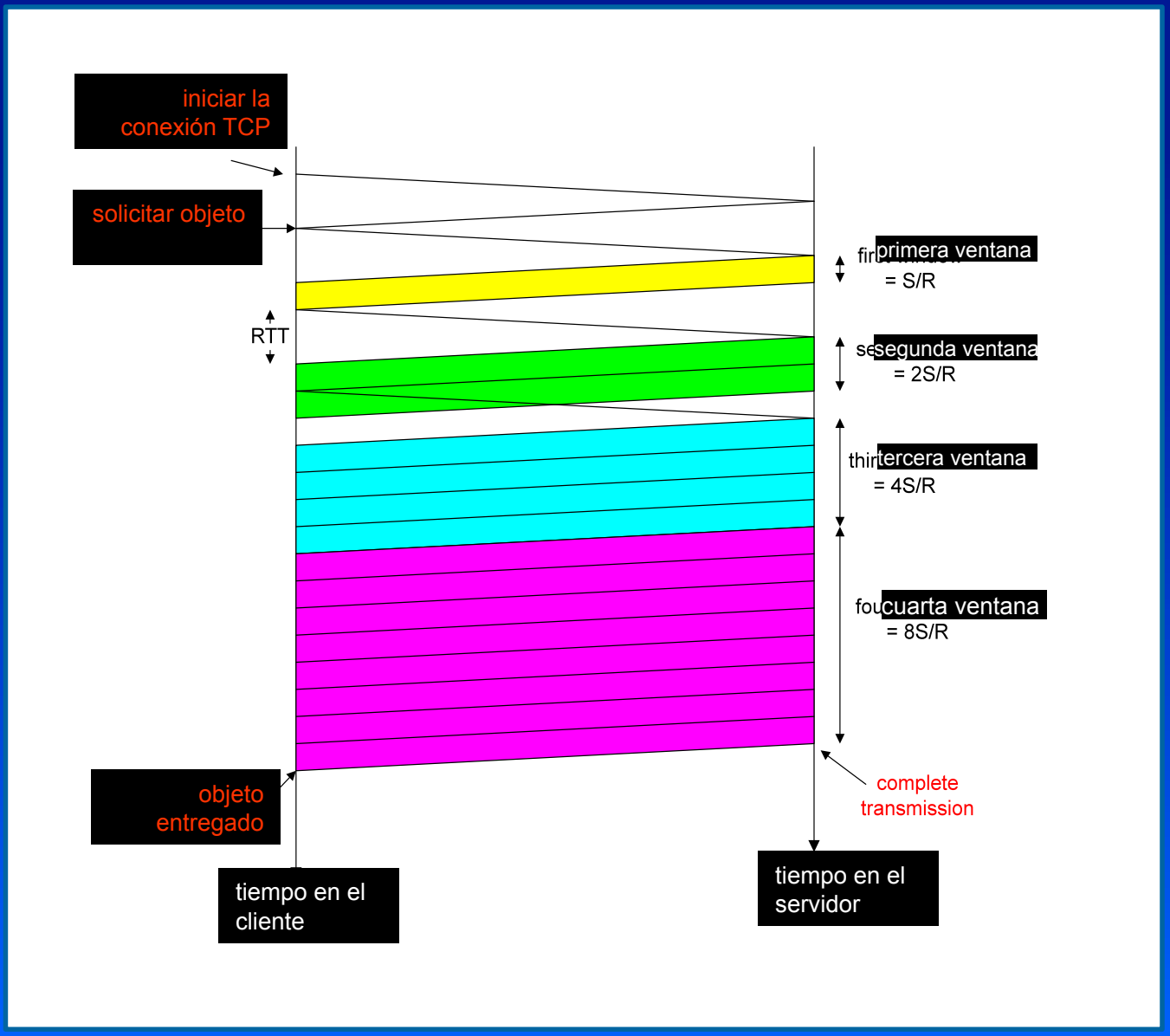
TCP

Basado en Ventana





TCP

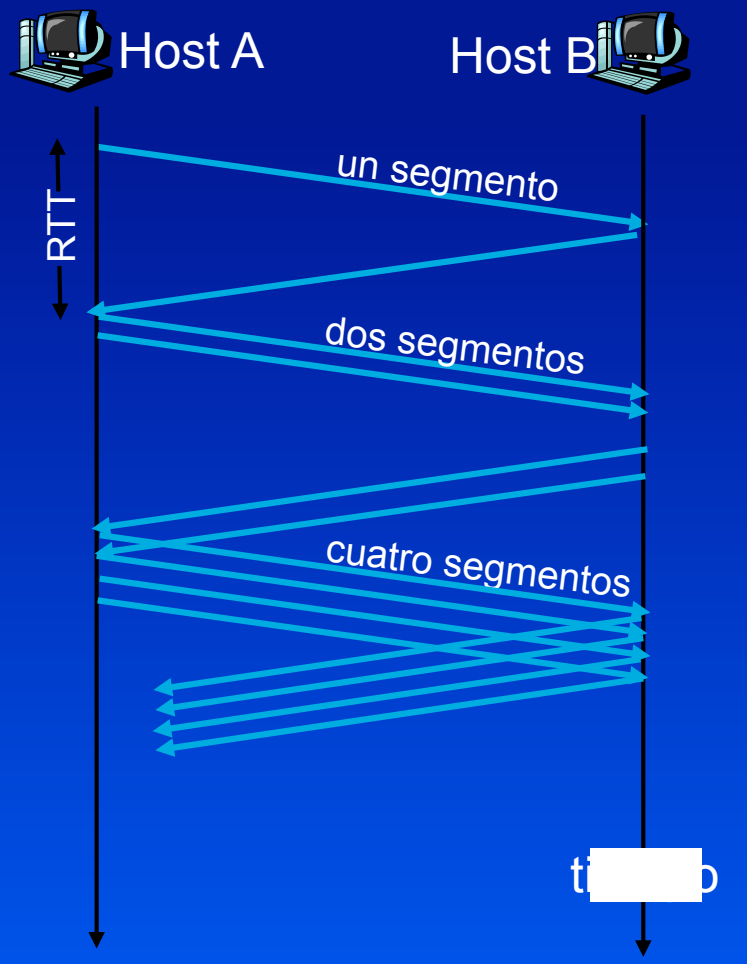

iniciar la conexión TCP
solicitar objeto
RTT
primera ventana = S/R
segunda ventana = 2S/R
tercera ventana = 4S/R
cuarta ventana = 8S/R
complete transmission
objeto entregado
tiempo en el cliente
tiempo en el servidor

FUNCIONAMIENTO

- Cuando la **`VentanaCong`** está por debajo del **`Umbral`**, el emisor está en la fase de arranque lento y la ventana crece exponencialmente.
- Cuando la **`VentanaCong`** está por encima del **`Umbral`**, el emisor está en una fase de evitar la congestión, la ventana crece linealmente.
- Cuando acontece un triple ACK duplicado, el **`Umbral`** se fija a la mitad de la **`VentanaCong`** y la **`VentanaCong`** se pone al valor del **`Umbral`**.
- Cuando se da el tiempo límite de espera, el **`Umbral`** se fija a la mitad de la **`VentanaCong`** y la **`VentanaCong`** se pone a 1 MSS.

FUNCIONAMIENTO

- Cuando la conexión comienza, la tasa aumenta exponencialmente rápido hasta que ocurre el primer evento de pérdida:
  - La **`VentanaCong`** se dobla en cada RTT.
  - La **`VentanaCong`** aumenta con cada ACK recibido.
- Resumen: la tasa inicial es lenta pero crece exponencialmente rápido.



TCP

CONGESTION

TCP

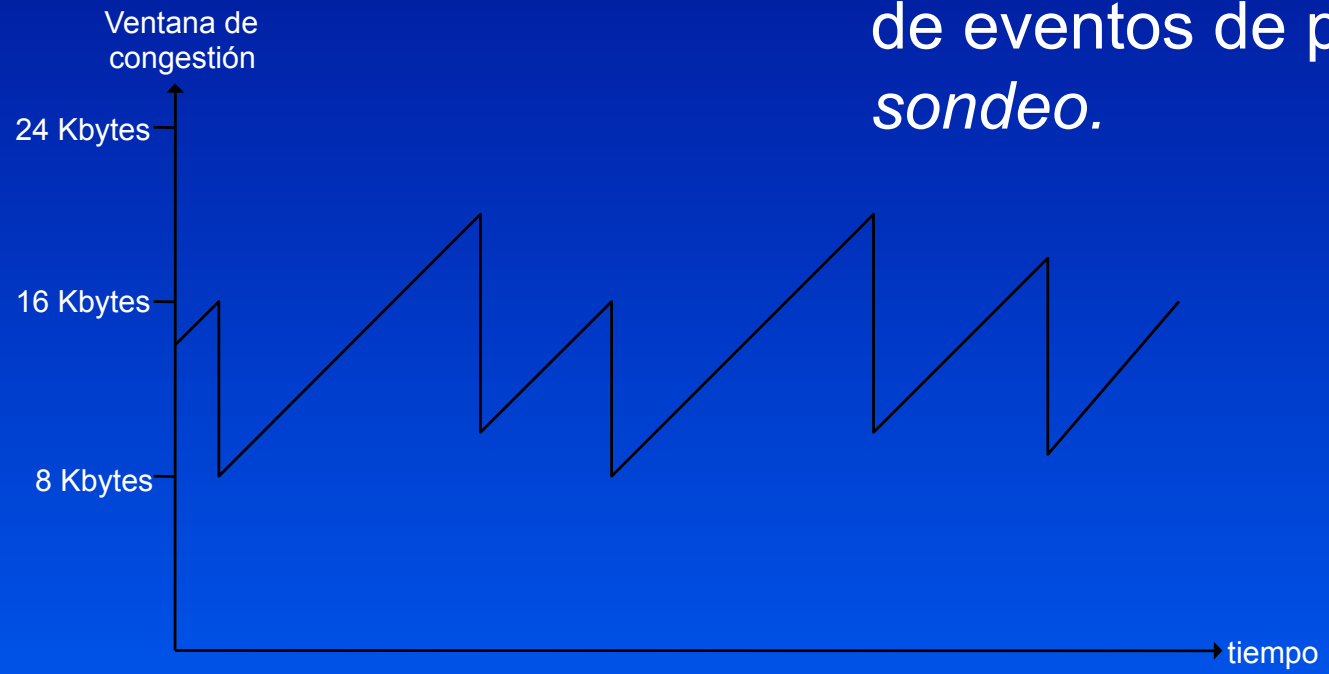
- Después de 3 ACKs duplicados:
  - La **`VentanaCong`** se divide a la mitad.
  - La ventana crece linealmente.
- Pero tras el evento del tiempo límite:
  - La **`VentanaCong`** se fija, en cambio, a 1 MSS.
  - La ventana entonces crece exponencialmente.
  - Al llegar a un umbral, entonces crece linealmente.

FUNCIONAMIENTO AIMD

TCP

Decremento multiplicativo: divide la **`VentanaCong`** a la mitad tras el evento de pérdida.

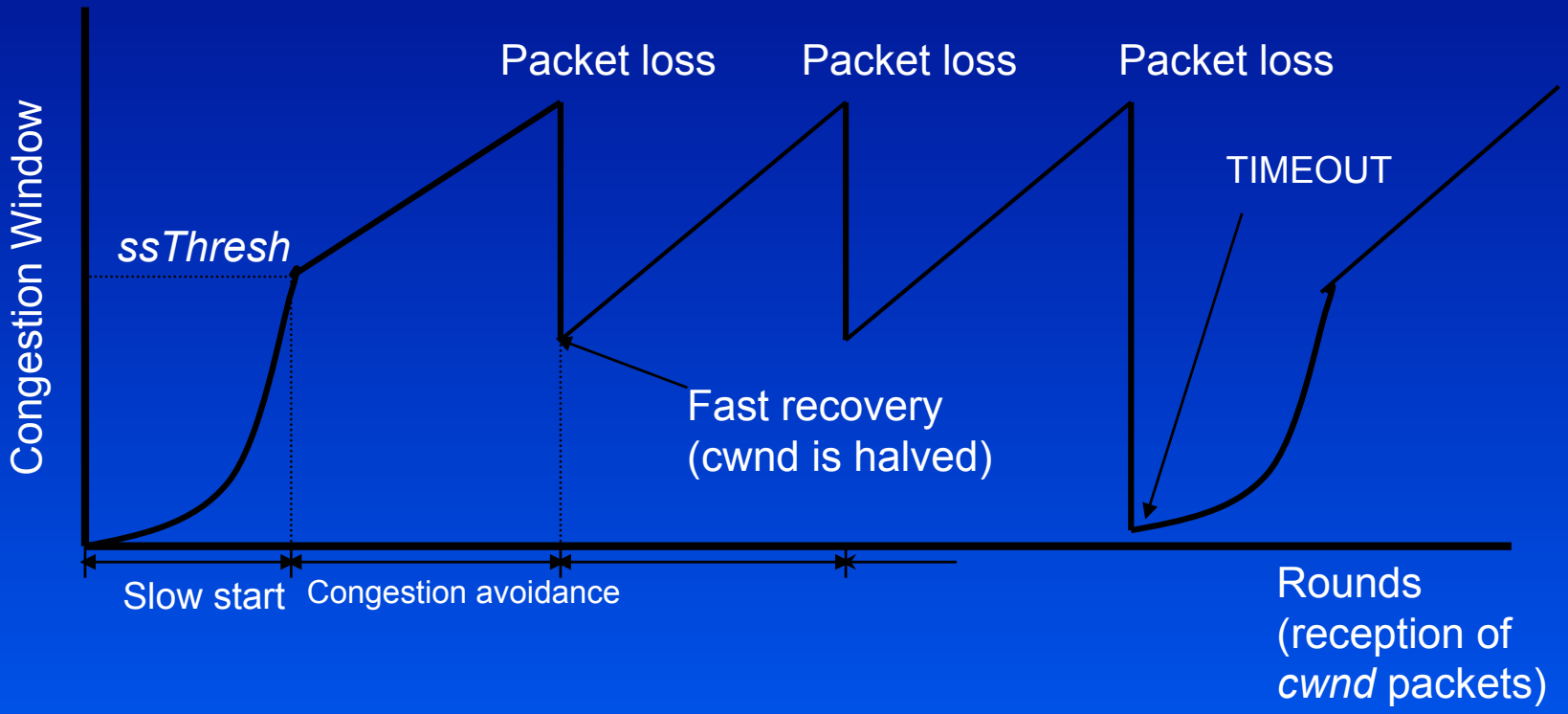
Incremento aditivo: aumenta la **`VentanaCong`** a 1 MSS cada RTT en ausencia de eventos de pérdida: *sondeo.*



Conexión TCP de larga vida

FUNCIONAMIENTO

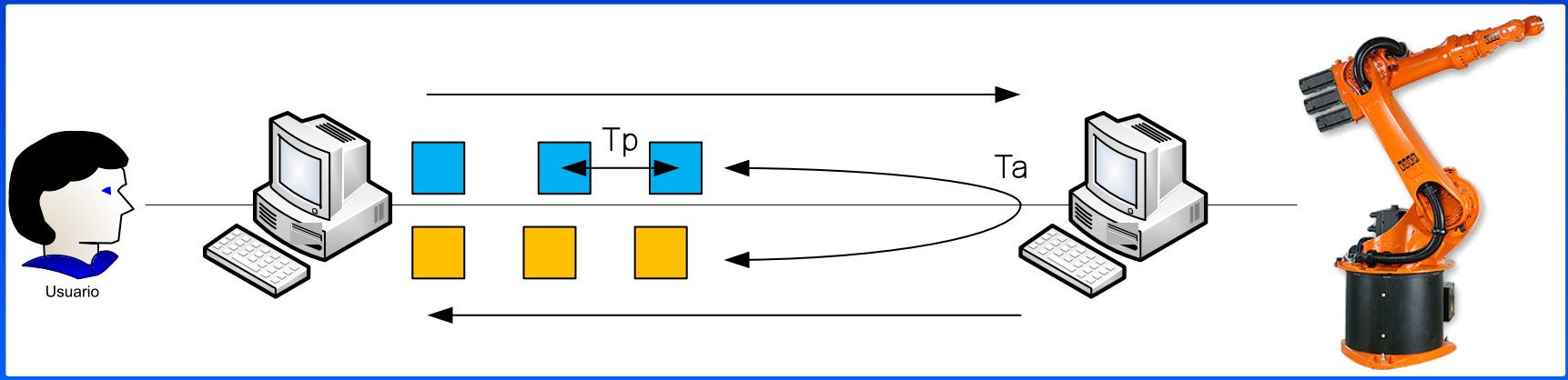
TCP



Telecontrol bilateral M/S

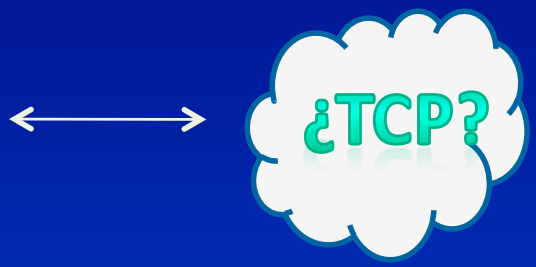
TCP

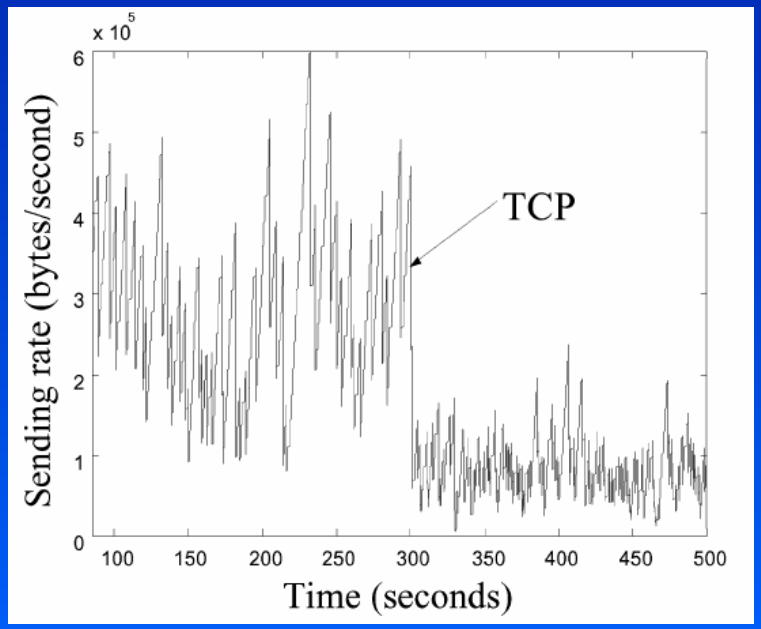
Basado en Ventana
Ta > Tp



Telecontrol bilateral M/S

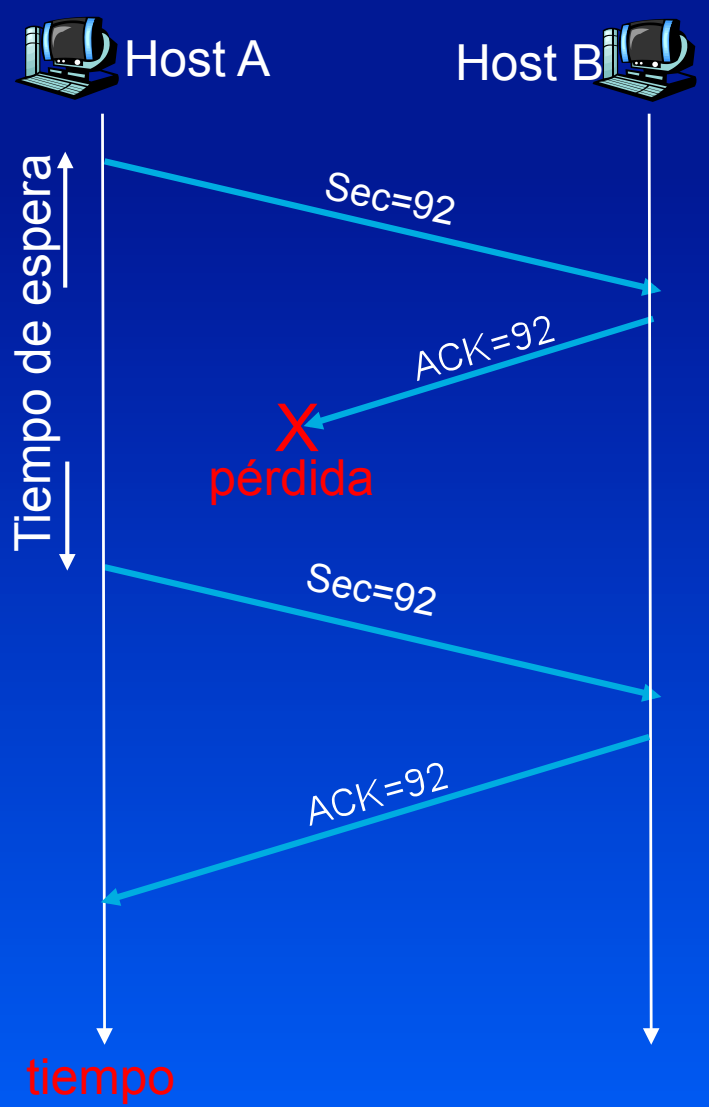
TCP





Jitter= Variación de retardo

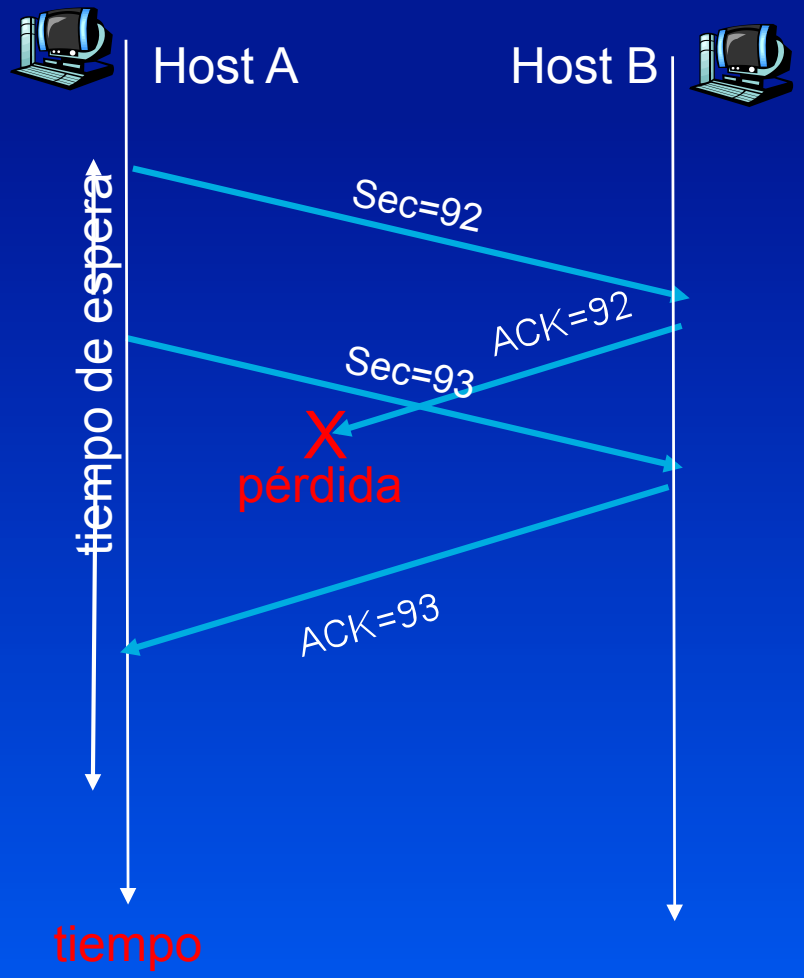
FUNCIONAMIENTO



Ejemplo de pérdida de ACK

FUNCIONAMIENTO

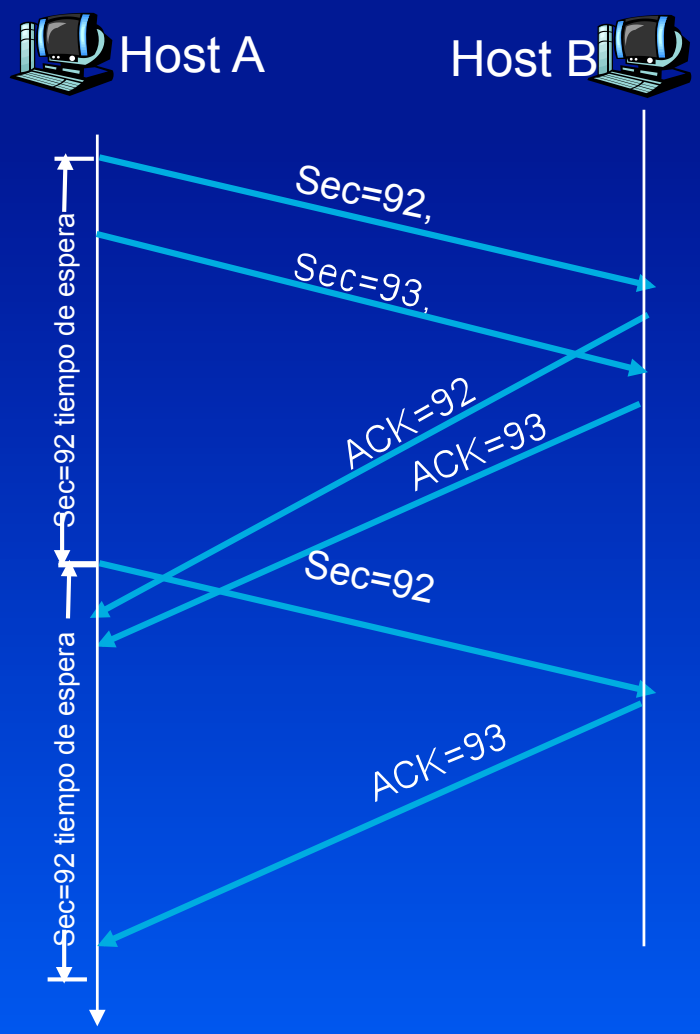
TCP



Ejemplo de ACK acumulado

FUINCIONAMIENTO

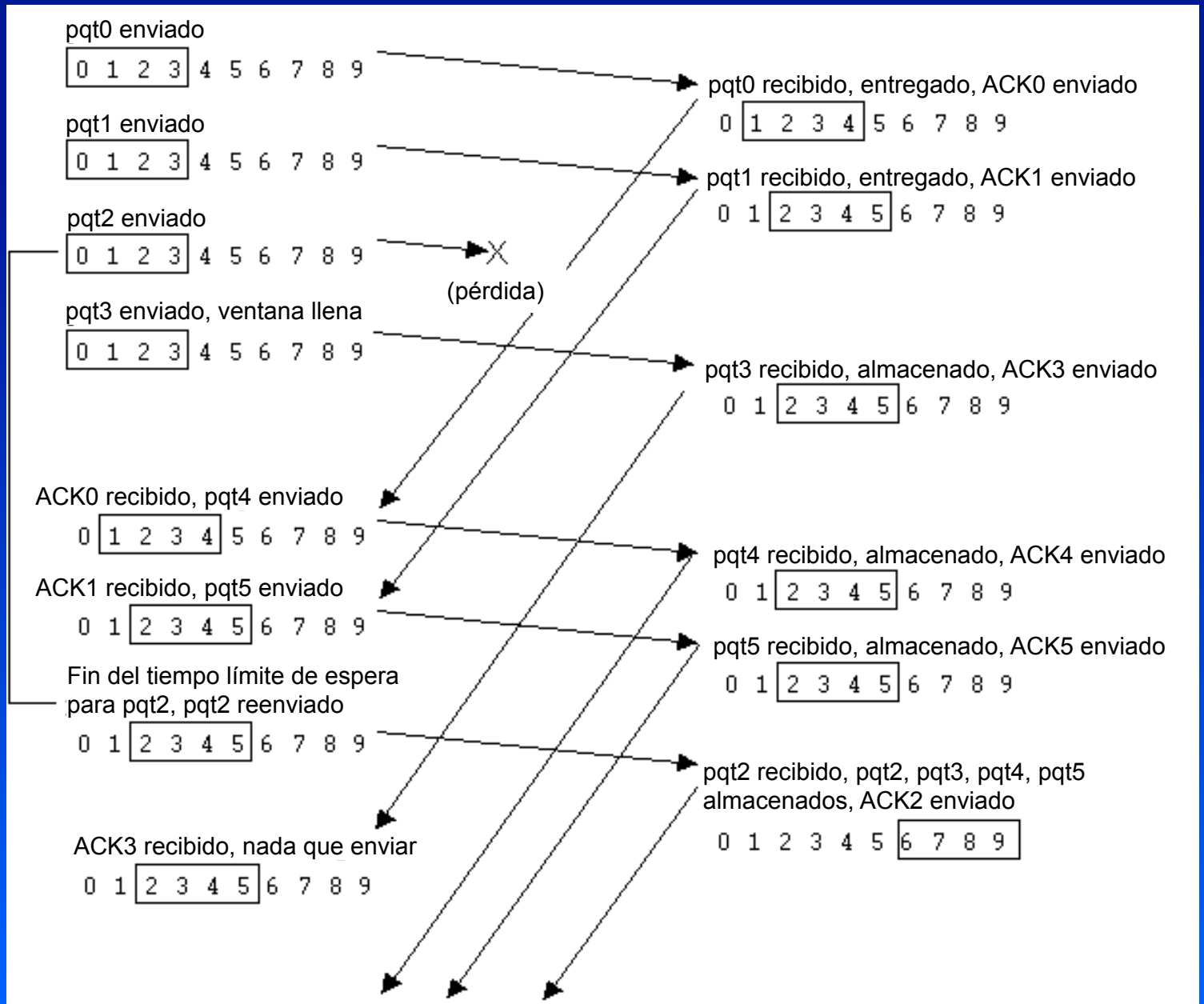
TCP



tiempo de espera prematuro

FUNCIONAMIENTO

TCP



DELAY

Pregunta: ¿Cuánto se tarda en recibir un objeto del servidor después de envíar la petición?

Sin contar la congestión, el retardo tiene su origen en:

- El establecimiento de conexión TCP.
- El retardo de transmisión de datos.
- El arranque lento.

TCP

Telecontrol bilateral M/S

TCP

FAMILIA TCP

1975
Primera demostración

1981
4.1aBSD RFC 793

1986
4.3BSD. Mejoras de rendimiento.

1988
4.3BSD-Tahoe . SlowStart, Congestion avoidance, fast retransmit

1990
4.3BSD-Reno. Predicción cabeceras, compresión, fast recovery, cabecera slip

1994
TCP/Las Vegas Detección de congestión por el RTT.

1996
SACK
ACK: Lista de paquetes.

1999
TCP New Reno
Mejora el fast retransmit y el fast recovery.

2000
TEAR
TCP . Receptor controla la congestion

2003
STCP
Formulas de TCP cambiadas para redes anchas

2003
HSTCP
Slow Start limitado, ventanas grandes de congestión.

2005
NACK
ACK de paquetes incorrectos

Telecontrol bilateral M/S

TCP Westwood

Redes wireless con alta probabilidad de perdida de paquetes.

El ancho de banda del “sender” viene decidido por el ratio de los ACK y por la informacion que contiene.

Ajusta el tamaño de Ventana según el ancho de banda estimado.

Tratamiento “especial” con la perdida de paquetes

TCP

Table 1
Internet throughput measurements.

Destination	Italy		Taiwan		Brazil	
RTT	170 ms		250 ms		450 ms	
Protocol	TCPW	Reno	TCPW	Reno	TCPW	Reno
Throughput (KB/s)	78.66	73.93	167.38	152	22.16	15.4

Telecontrol bilateral M/S

TEAR

TCP

TCP	TEAR
For each packet sent, one ack packet is received.	For each packet sent, none ack packet is received
Congestion Signals are detected by the sender (Packet loss, Timeouts and Packet Arrivals)	Congestion Signals are detected by the receiver (Packet loss, Timeouts and Packet Arrivals)
Sender control the protocol (cwnd and rate). Receiver only sends ack packets.	Receiver control the protocol (cwnd and rate). Sender only sends packets.

INDICE

PROTOCOLOS

OTROS PROTOCOLOS

TCP / Las Vegas
Trinomial Protocol
Tear
RTNP (Real-Time Network Protocol)
IRTP (Interactive Real-Time Protocol)
SMTCP
TFRC (TCP-Friendly Rate Control Protocol)
RAP (Rate Based Adaptation Protocol)
LDA (Loss-Delay Adjustment Protocol)
SIMD (Square-Increase/Multiplicative-Decrese Protocol)
RTP (Real Time Protocol)

TCP-Friendly (Imparcialidad)

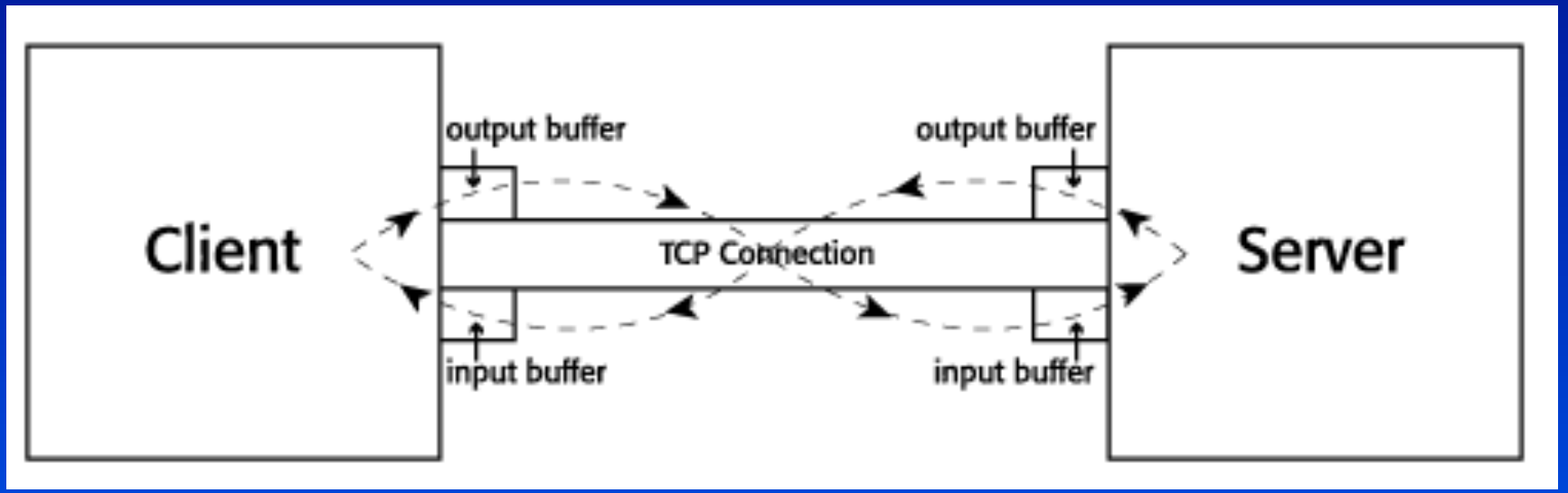
OTROS PROTOCOLOS

Protocolos Multimedia

- Las aplicaciones multimedia no suelen usar TCP:
  - No les conviene que la tasa se estrangule por control congestión.
- En su lugar, usan UDP:
  - Bombean audio/vídeo a una tasa constante, toleran pérdida de paquetes.
- Área de búsqueda: TCP compatible.

PROTOCOLOS BUFFER

OTROS PROTOCOLOS



INDICE

Teleoperación bilateral M/S

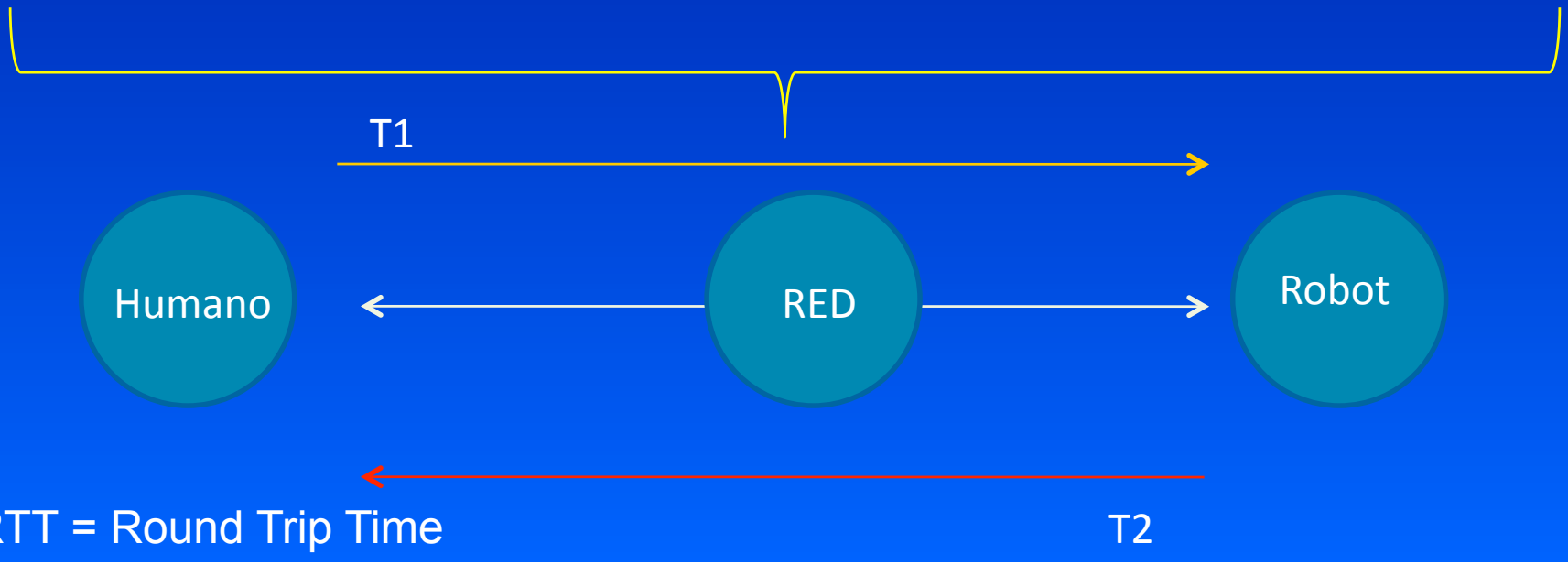
TELEOPERACION M-S

¿OBJETIVO?

El tiempo de transmisión entre el humano y el robot (RTT) debe de ser el mínimo

$\min(RTT) = \min(T_1) + \min(T_2)$

Solución: Que la red no se sature.



RTT = Round Trip Time

Teleoperación bilateral M/S

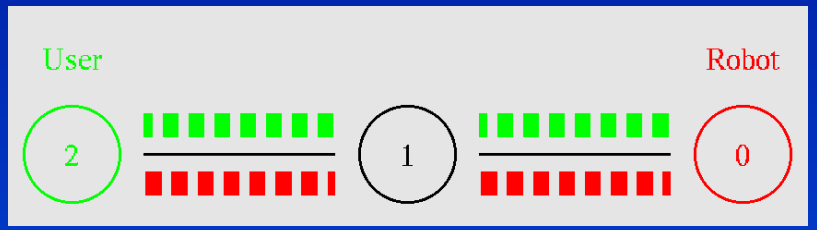
TELEOPERACION M-S

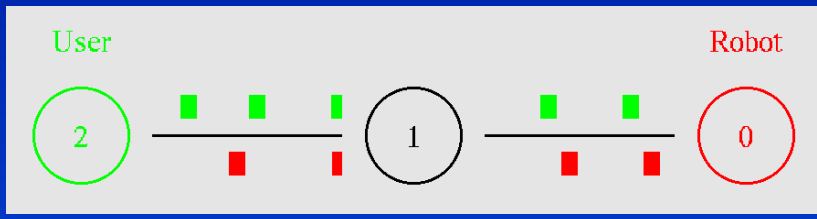
¿OBJETIVO?

El tiempo de transmisión entre el humano y el robot (RTT) debe de ser el mínimo

$\min(RTT) = \min(T_1) + \min(T_2)$

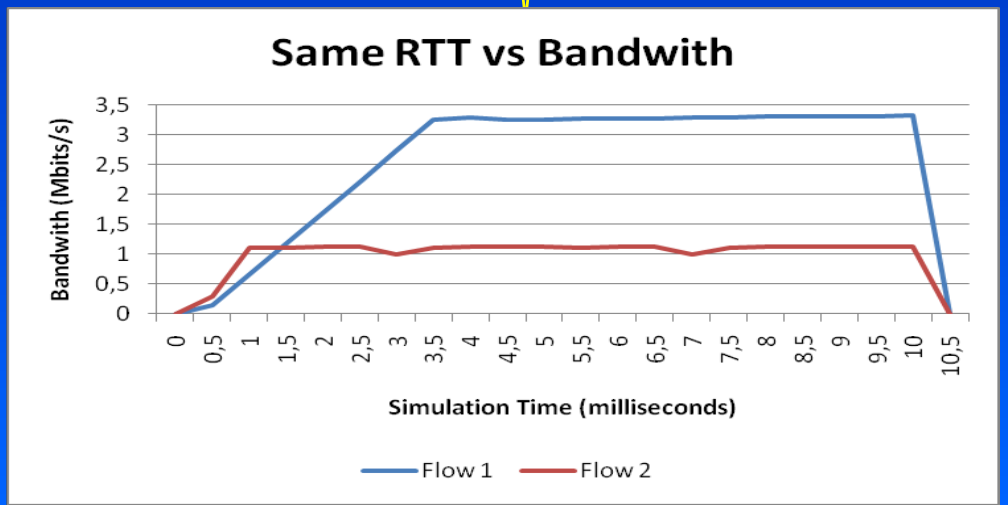
Solución: Que la red no se sature
Problemas: Distinto ancho de banda y Distinta frecuencia de llegada





Flow 1 Flow 2

Humano



Robot

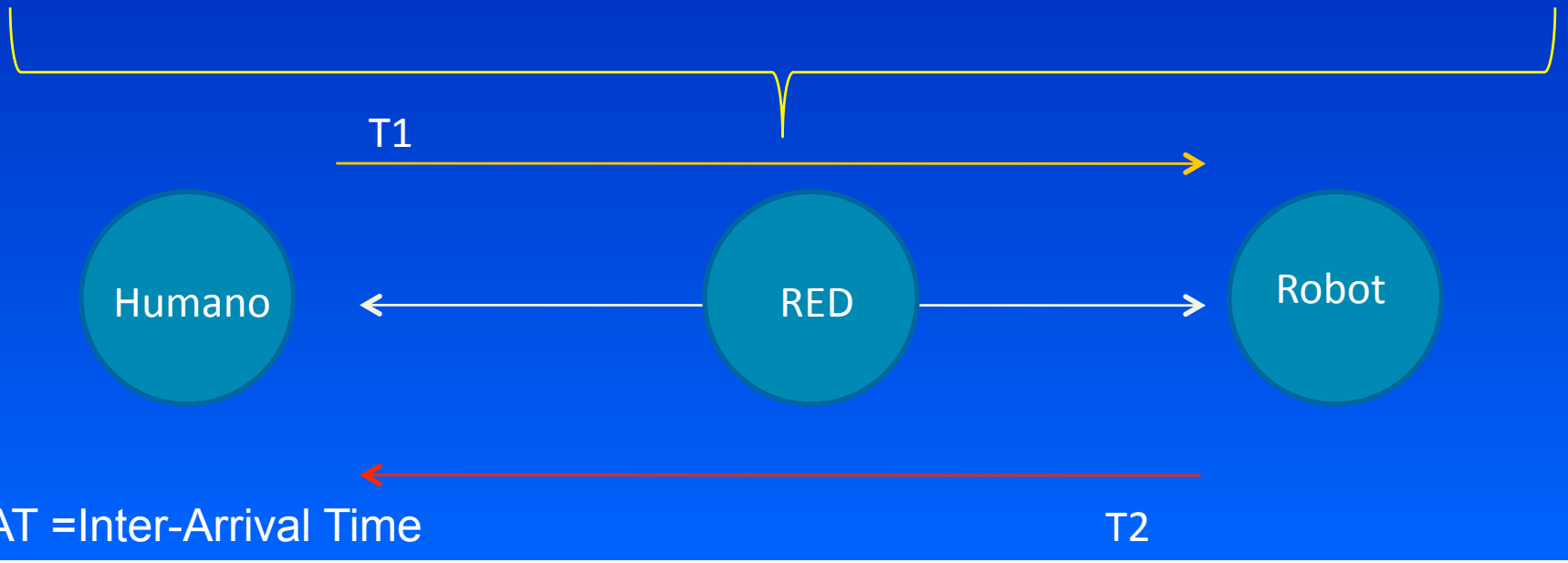
Teleoperación bilateral M/S

TELEOPERACION M-S

¿OTRO OBJETIVO?

La frecuencia de llegada de los paquetes debe de ser constante
IAT constante

Solución: Controlar la llegada de paquetes



IAT =Inter-Arrival Time

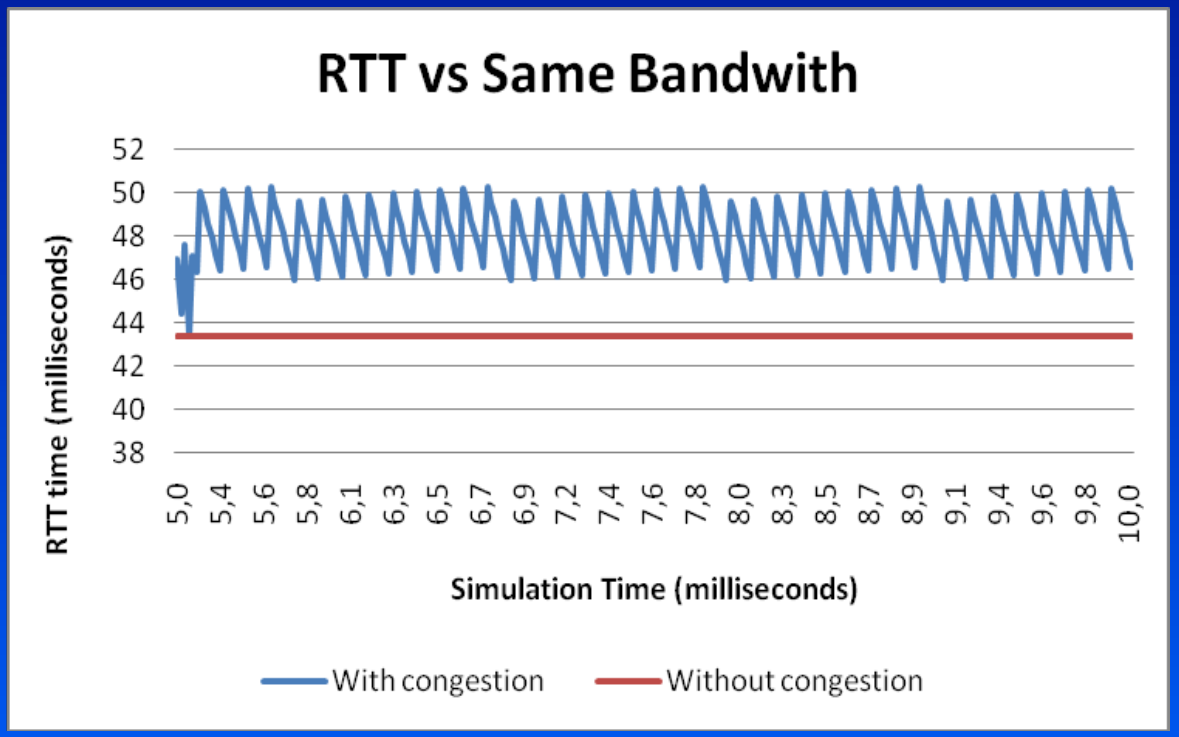
Teleoperación bilateral M/S

TELEOPERACION M-S

¿OBJETIVO?

La frecuencia de llegada de los paquetes debe de ser constante
IAT constante

Solución: Controlar la llegada de paquetes
Problema: La congestión provoca efecto acordeón



Dos flujos con mismo ancho de banda pero distinto RTT

IAT =Inter-Arrival Time

RTT = Round Trip Time

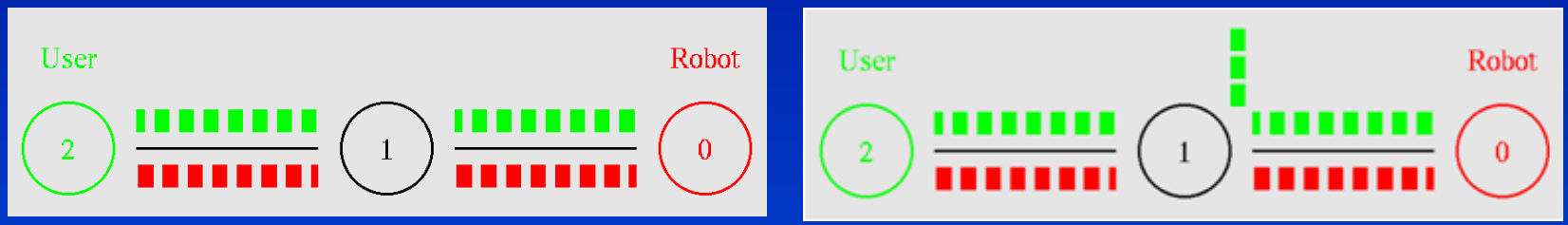
Teleoperación bilateral M/S

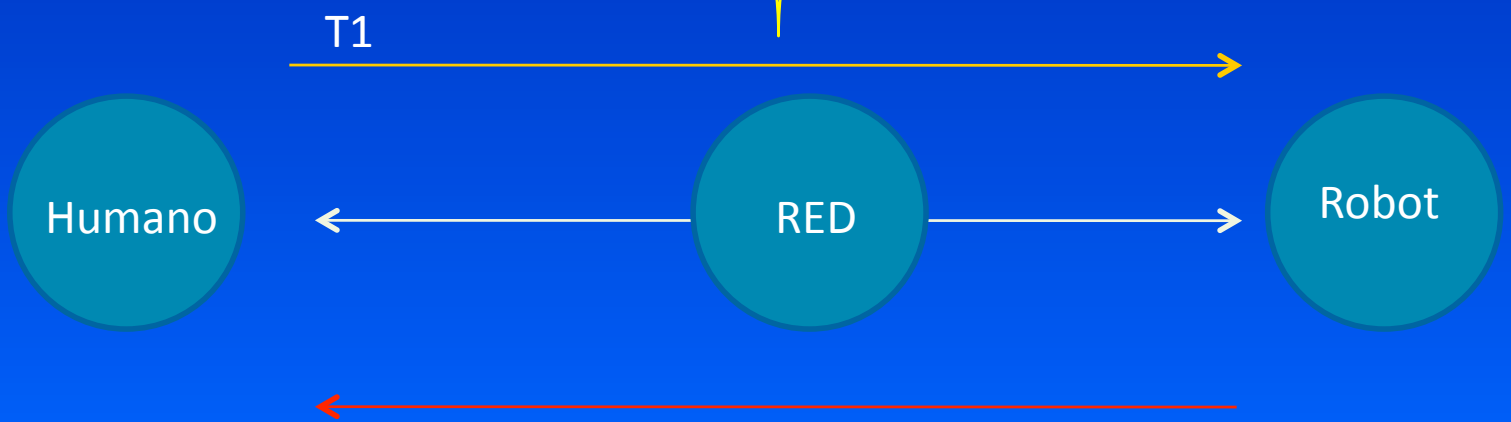
TELEOPERACION M-S

¿OBJETIVO?

La frecuencia de llegada de los paquetes debe de ser constante
IAT constante

Solución: Controlar la llegada de paquetes
Problema: Mismo tiempo de llegada (o ancho de banda), distinto tiempo de transmisión.





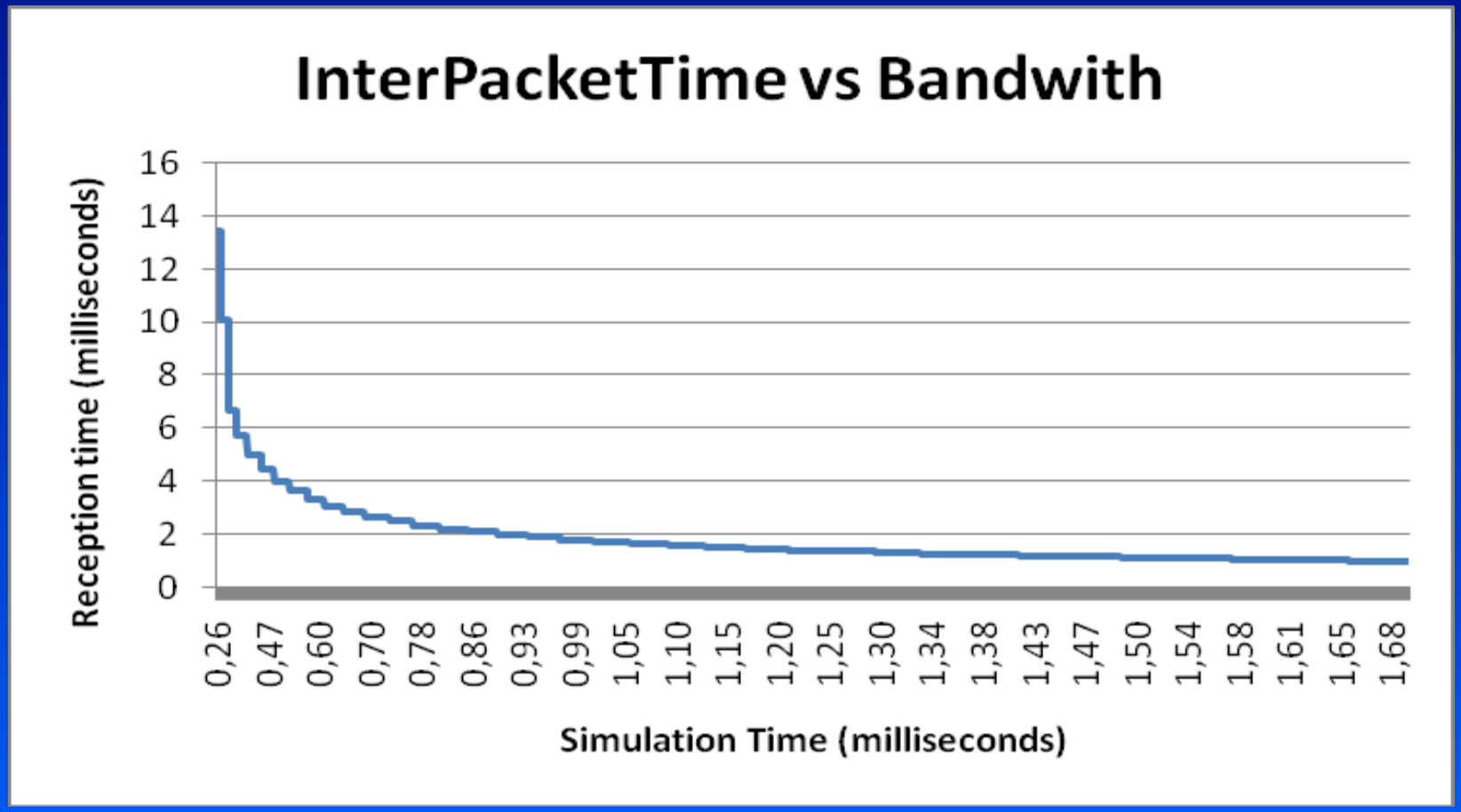
IAT =Inter-Arrival Time

Teleoperación bilateral M/S

PROPIEDAD DE LA RED

TELEOPERACION M-S

Aumentando el ancho de banda se disminuye el IAT



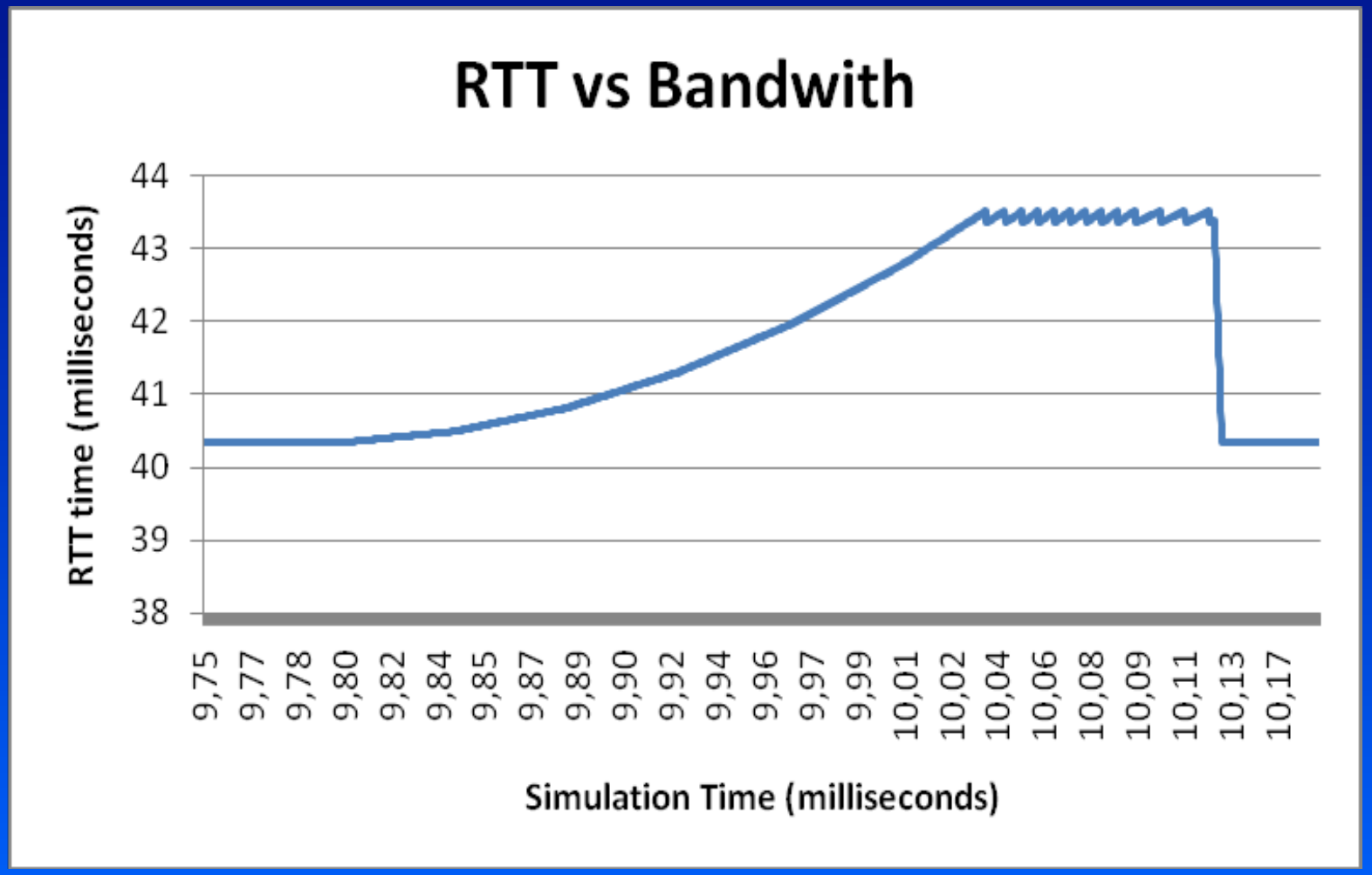
IAT =Inter-Arrival Time

Teleoperación bilateral M/S

PROPIEDAD DE LA RED

TELEOPERACION M-S

Aumentando el ancho de banda se aumenta el RTT

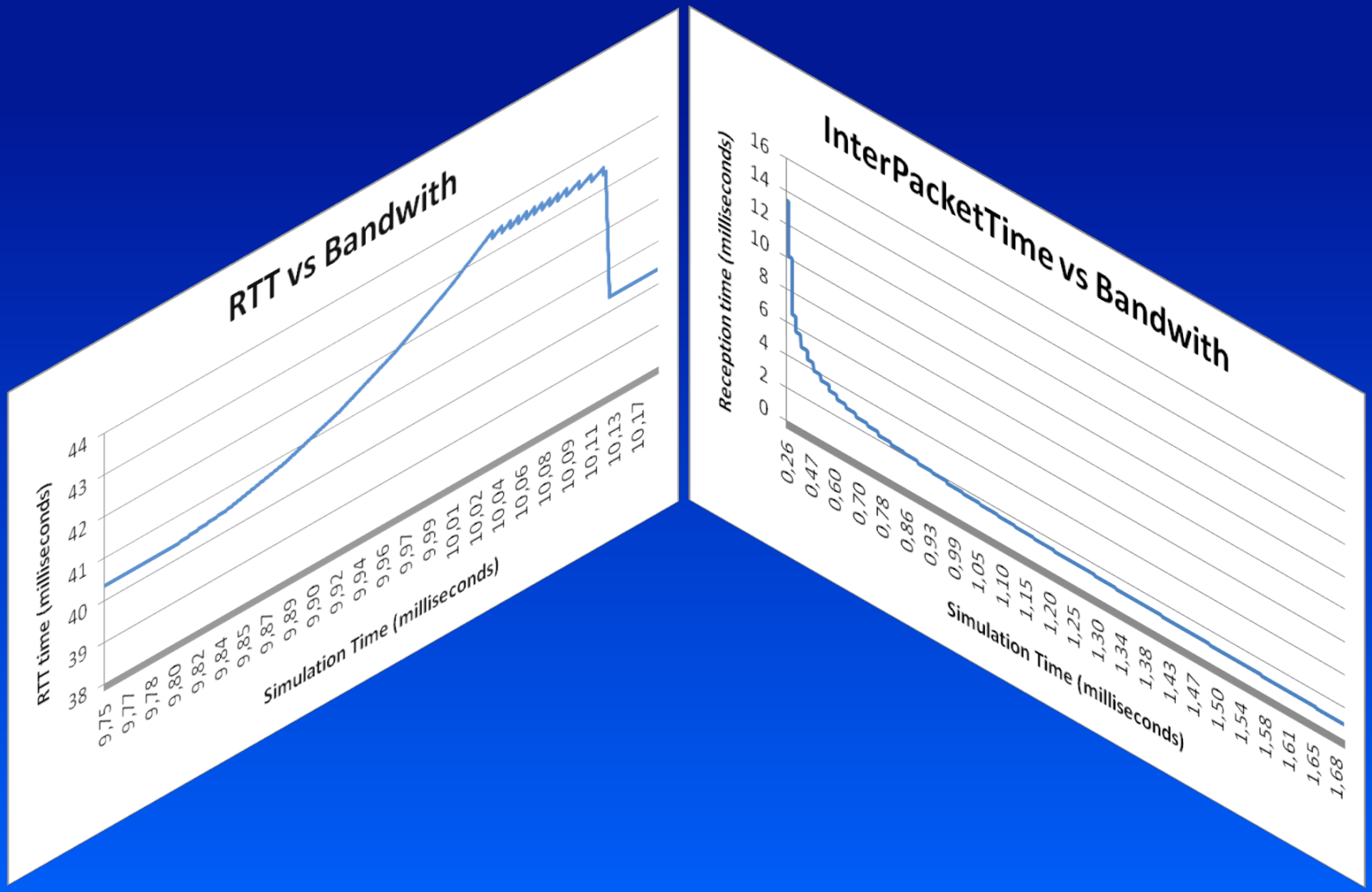


RTT = Round Trip Time

Teleoperación bilateral M/S

TELEOPERACION M-S

PROPIEDAD DE LA RED Y OBJETIVOS



Teleoperación bilateral M/S

TELEOPERACION M-S

OBJETIVOS

El tiempo de transmisión entre el humano y el robot (RTT) debe de ser el mínimo

$$\min(RTT) = \min(T_1) + \min(T_2)$$

Ofrecer un tiempo de retardo mínimo	con	un IAT mínimo.
Ofrecer un tiempo de transmisión mínimo		un IPG mínimo.
Ofrecer un RTT mínimo		una frecuencia de llegada máxima.

La frecuencia de llegada de los paquetes debe de ser constante
IAT constante

IAT =Inter-Arrival Time

RTT = Round Trip Time

Telecontrol bilateral M/S

¿UDP?

ETHERNET

¿TCP?

ESTADO ARTE

UDP

- No se establece una conexión entre origen y destino.
- No usa ACK.
- No se retransmite los paquetes perdidos.
- No controla la congestión del sistema.
- No tiene control de flujo.
- Paquetes no numerados.
- Header: 8 Bytes

TCP

- Se establece una conexión entre origen y destino.
- Usa ACK.
- Se retransmite los paquetes perdidos.
- Controla la congestión del sistema.
- Tiene control de flujo.
- Paquetes numerados.
- Header: 20 Bytes

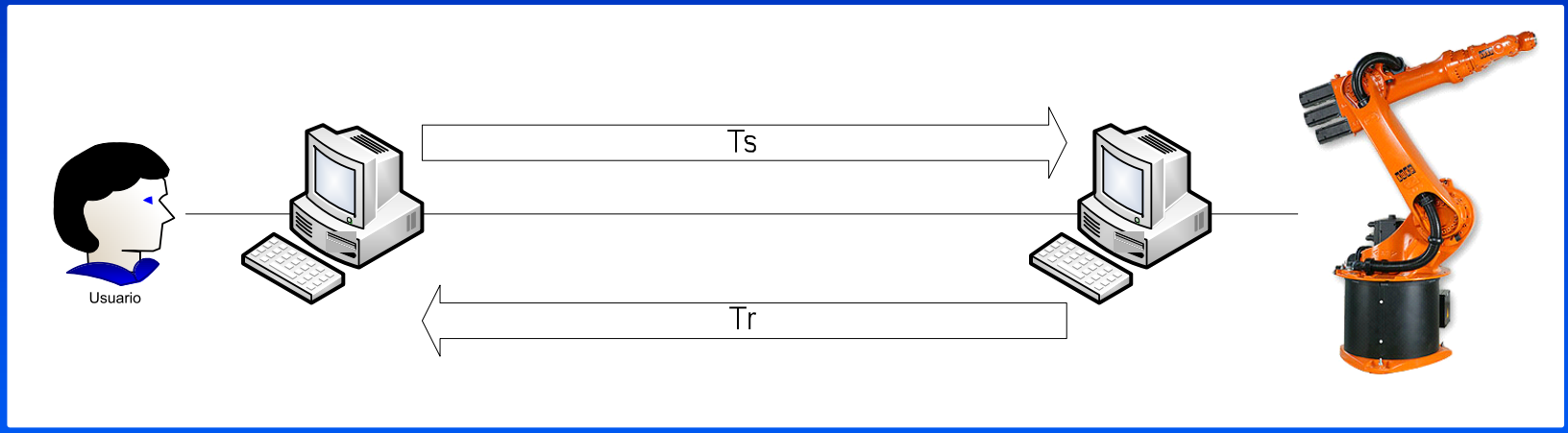
INDICE

Teleoperación bilateral M/S

TELEOPERACION M-S

Solución BTP

1. Dos canales de flujo independientes

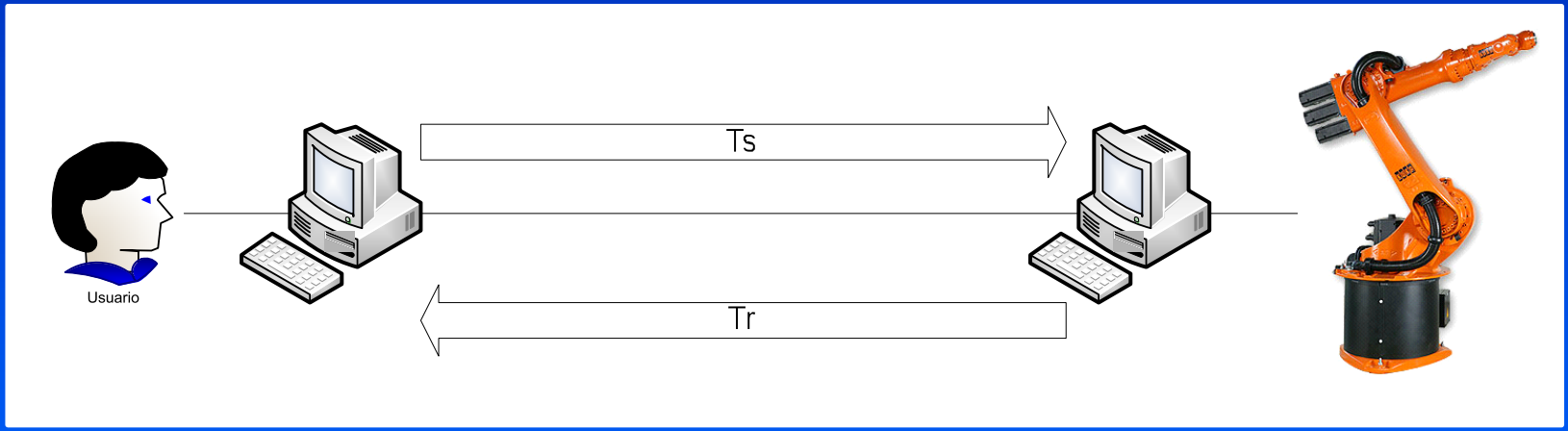


Teleoperacióm bilateral M/S

TELEOPERACION M-S

Solución BTP

1. Dos canales de flujo independientes
2. El lado que recibe los paquetes controla el IPG del lado que esta enviándolos



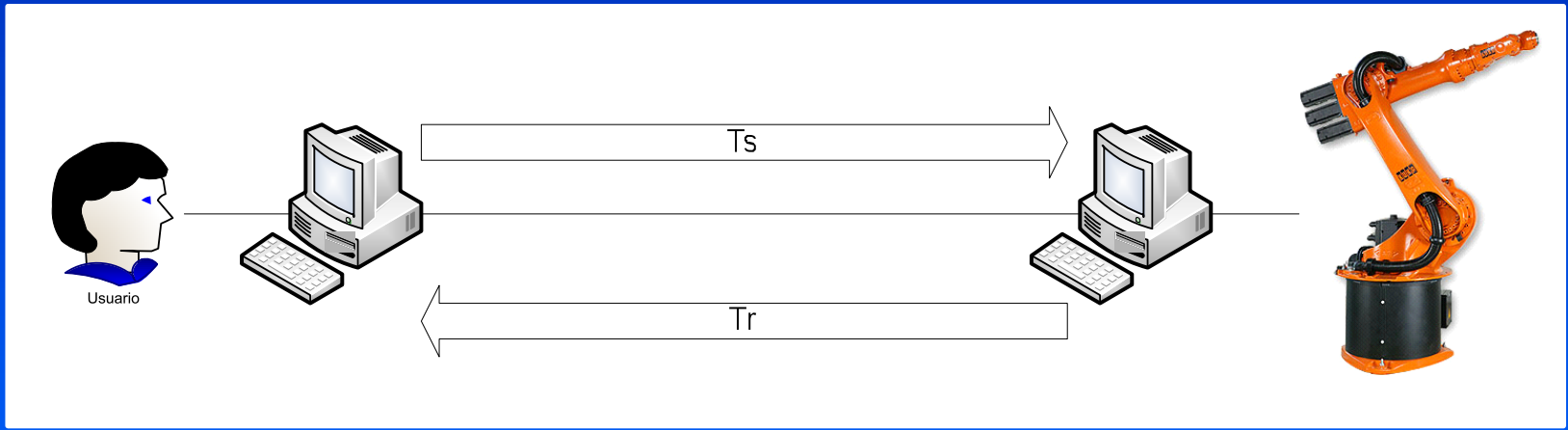
IPG = InterPacket Gap

Teleoperación bilateral M/S

TELEOPERACION M-S

Solución BTP

1. Dos canales de flujo independientes
2. El lado que recibe los paquetes controla el IPG del lado que esta enviándolos
3. Detección y control de la congestión por:
    - Control de tiempo de llegada entre paquetes (IAT)
    - Control del tiempo de transmisión
    - Control de la media del tiempo de llegada de paquetes

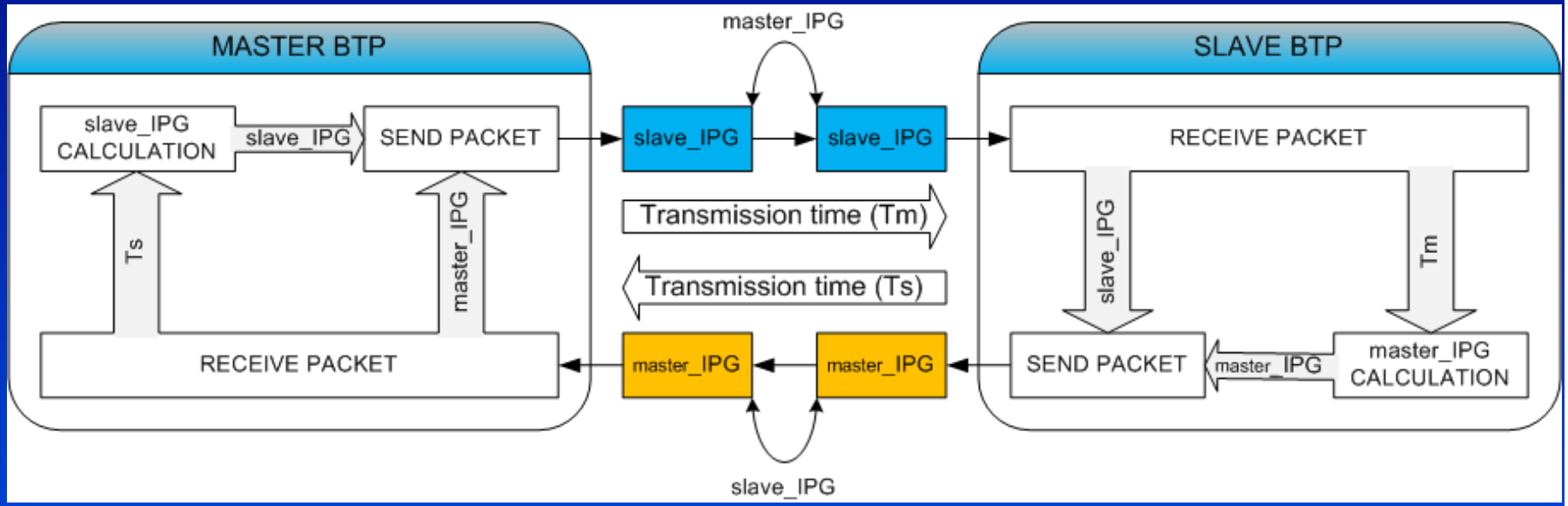


IPG = InterPacket Gap IAT =Inter-Arrival Time

Teleoperación bilateral M/S

TELEOPERACION M-S

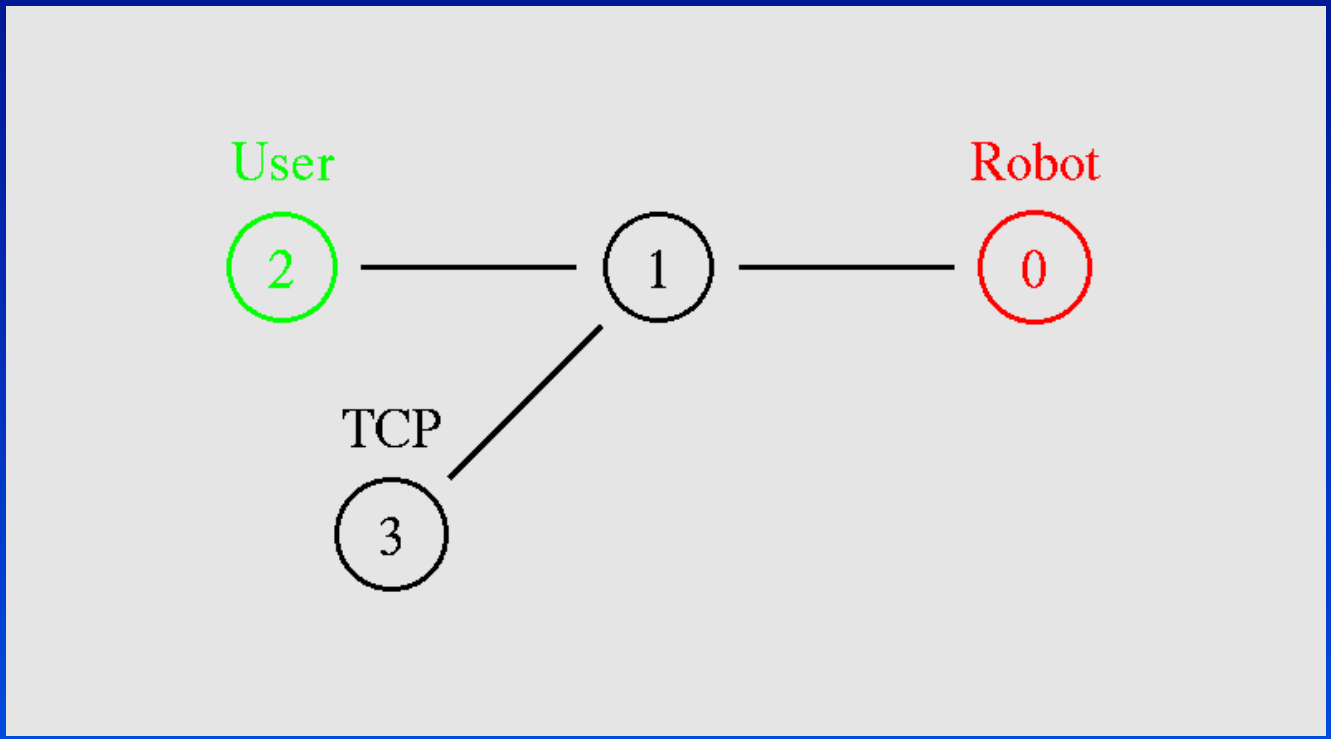
Solución BTP



Teleoperaciónm bilateral M/S

TELEOPERACION M-S

Simulación BTP

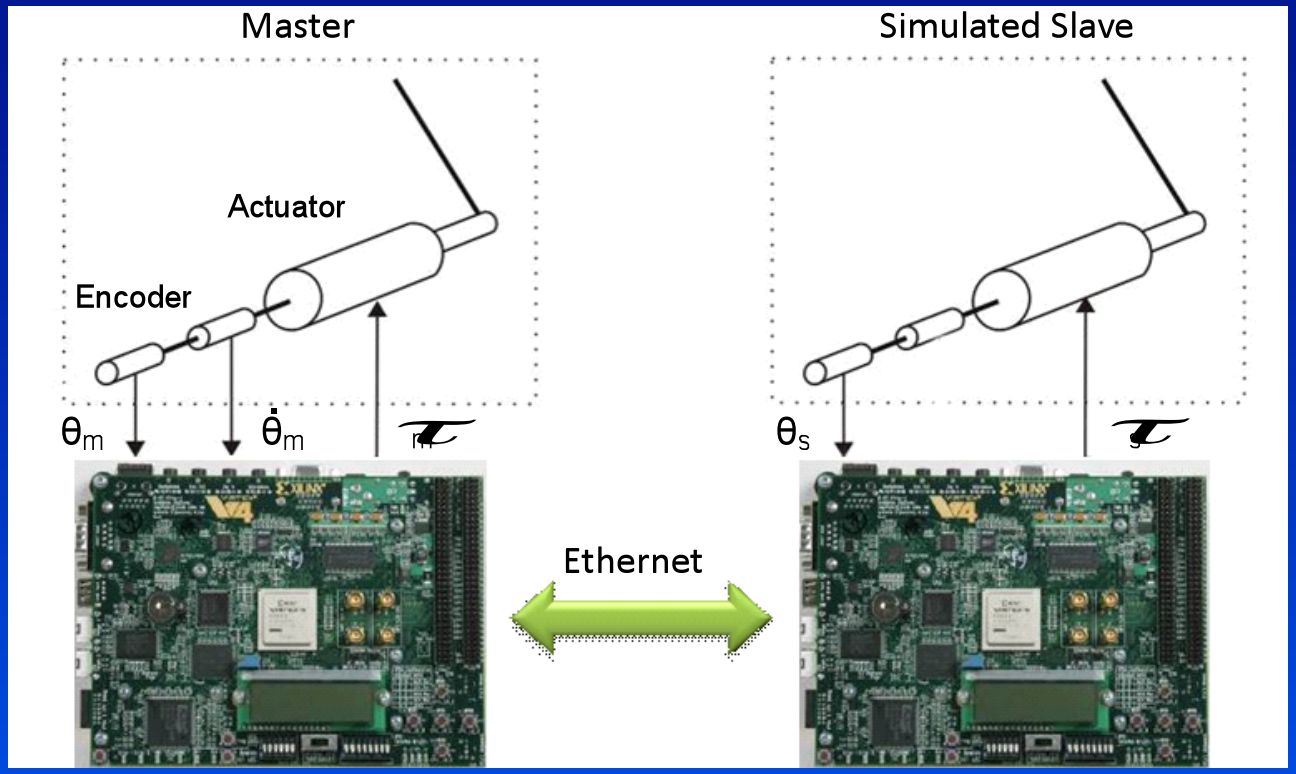


User(2)- > Robot (0): BTP
Robot (0) -> User (2): BTP
TCP (3) <–> Robot (0): TCP

Teleoperación bilateral M/S

TELEOPERACION M-S

Experimento Real



Simulación de Muelle

$$if\,(\theta_m \geq \theta_{0,s}) \Rightarrow \tau_m = 0$$

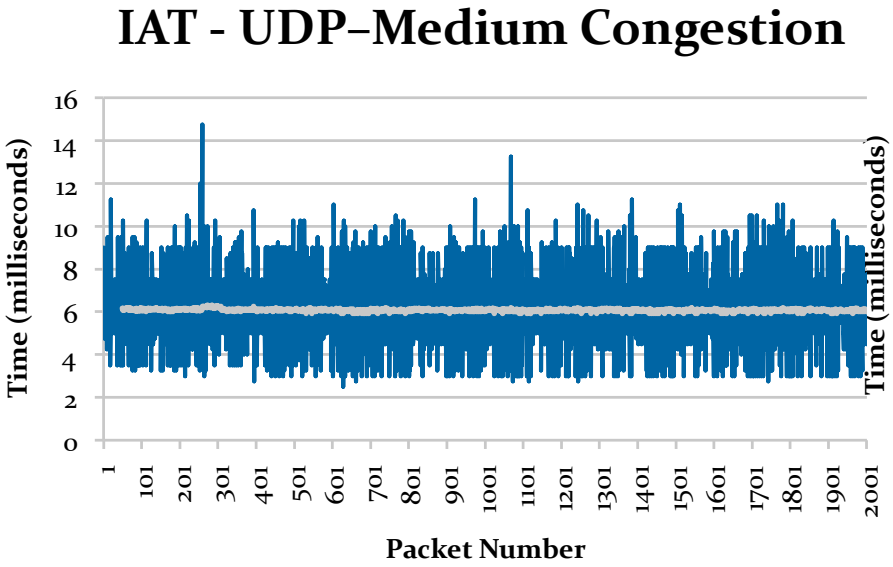
$$else\,\tau_m = k_s(\theta_m - \theta_{0,s})$$

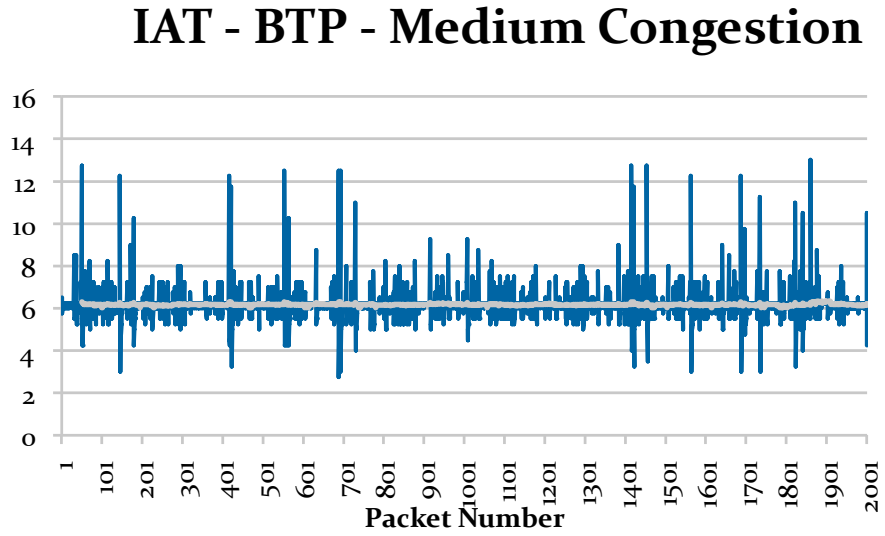
R. Wirz, R. Marin, M. Ferre, J. Barrio, J.M. Claver y J. Ortego *Bi-directional Transport Protocol for Teleoperated Robots.* **TRANSACTIONS ON INDUSTRIAL ELECTRONICS.** Pendiente Publicación

Teleoperación bilateral M/S

TELEOPERACION M-S

Experimento Real



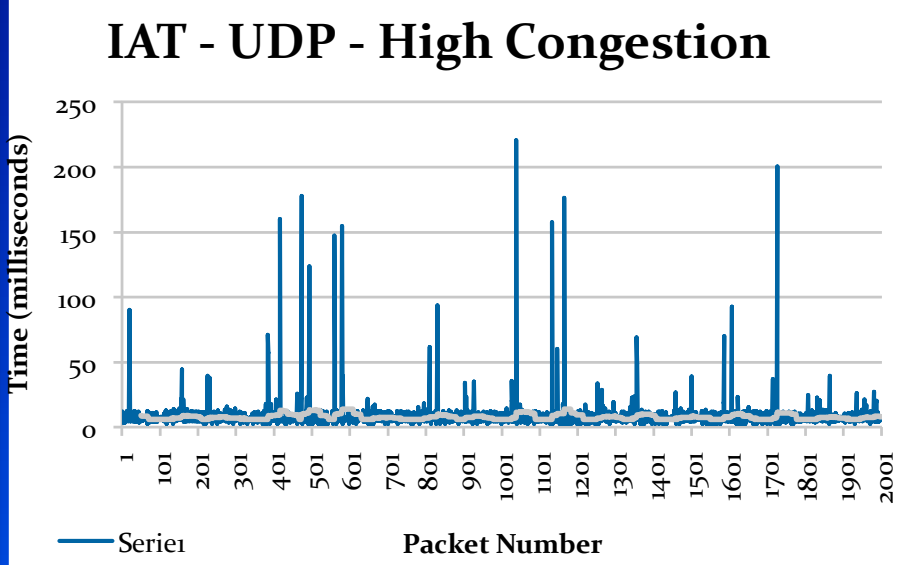


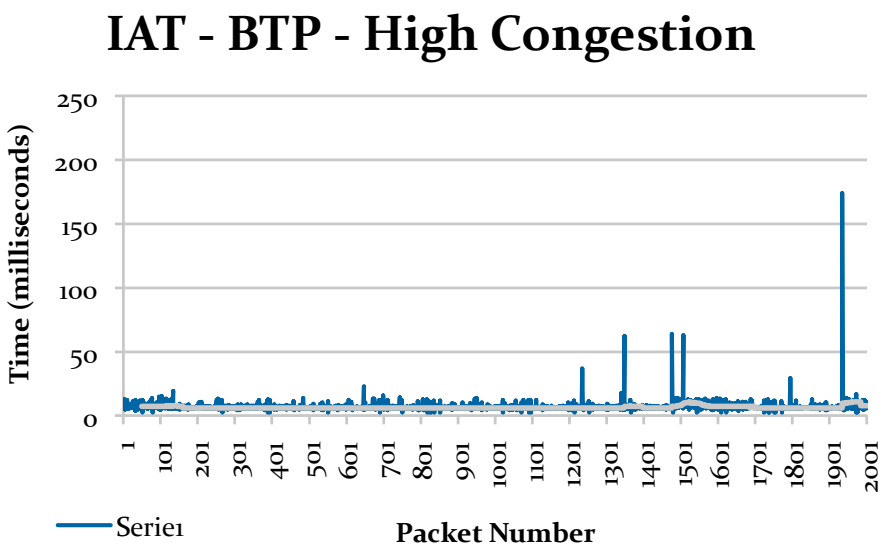
Media Congestión = 40% Ancho de Banda Disponible

Teleoperación bilateral M/S

TELEOPERACION M-S

Experimento Real



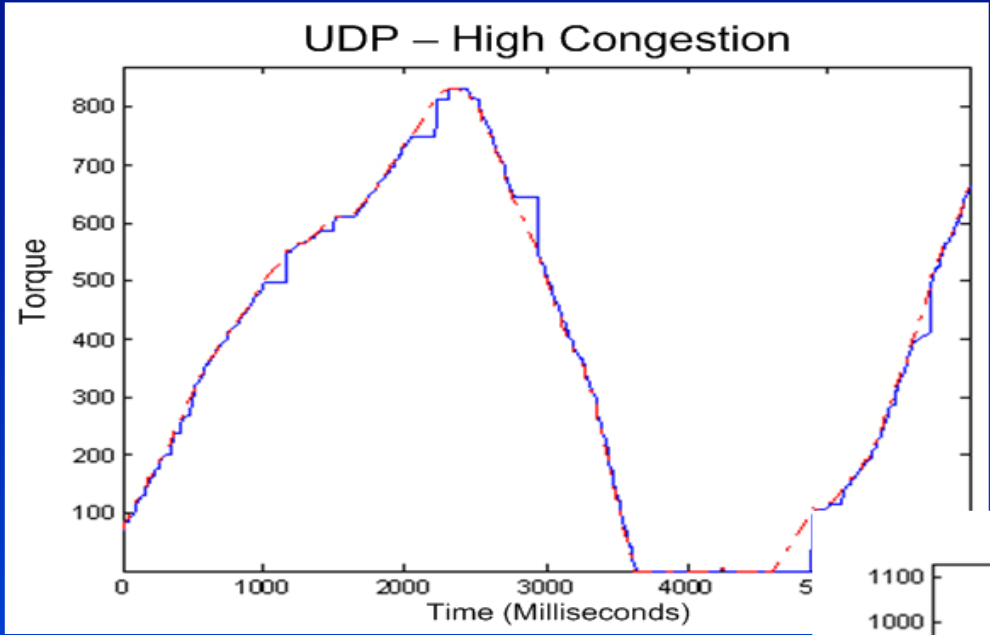


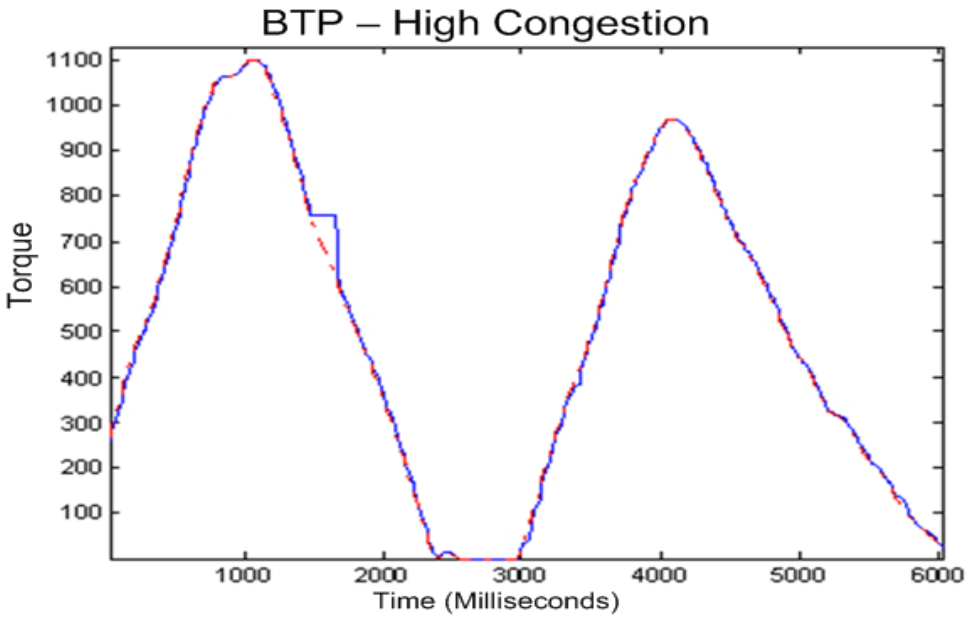
Alta Congestión = 70% Ancho de Banda Disponible

Teleoperación bilateral M/S

TELEOPERACION M-S

Experimento Real



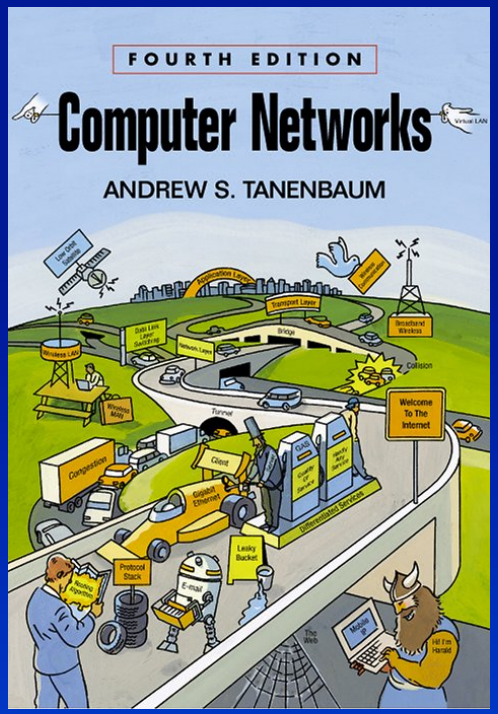


Línea Roja = Fuerza calculada por el esclavo
Línea Azul = Fuerza reflejada por el maestro

INDICE

BIBLIOGRAFIA

Redes de computadores: un enfoque descendente basado en Internet
2ª Edición
Jim Kurose, Keith Ross



Computer Networks
4ª Edición
Andrew S. Tanenbaum

GRACIAS