Internet Traffic Monitoring and Analysis:
Wireshark Tutorial

Dr.S.Geetha

School of Computer Science and Engineering

VIT, Chennai

geetha.s@vit.ac.in

Outline

- What is Wireshark?
- Capturing Packets
- Analyzing Packets
- Filtering Packets
- Saving and Manipulating Packets
- Packet Statistics
- Colorizing Specific Packets
- References

What is Wireshark?

- The De-Facto Network Protocol Analyzer
  - Gerald Combs 1998
  - Open-Source (GNU Public License)
  - Multi-platform (Windows, Linux, OS X, Solaris, FreeBSD, NetBSD, and others)
  - Easily extensible
  - Large development group
- Previously Named “Ethereal”

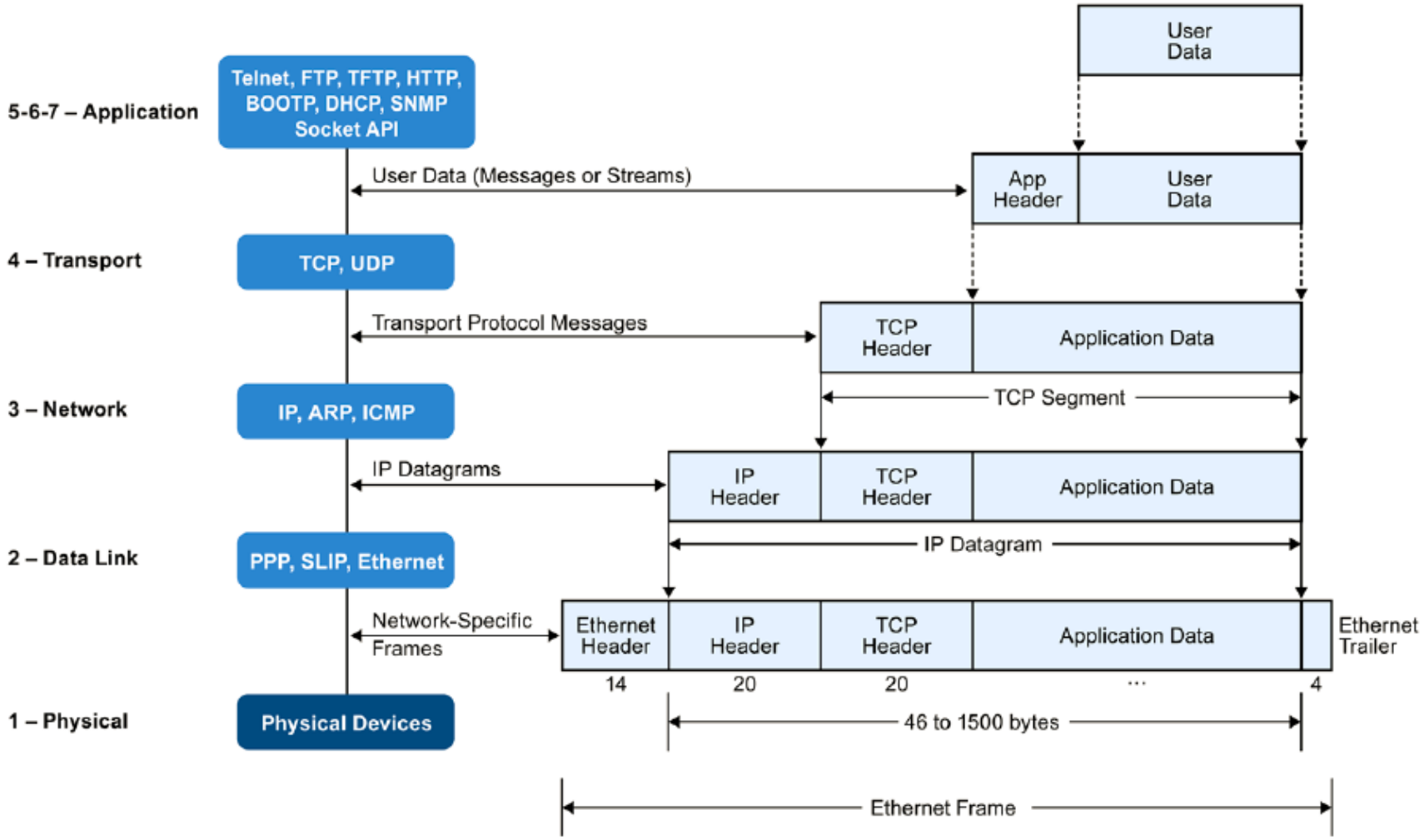
What is Wireshark?

- Features
  - Deep inspection of thousands of protocols
  - Powerful Packet Sniffer
  - Live capture and offline analysis
  - Standard three-pane packet browser
  - Captured network data can be browsed via a GUI, or via the TTY-mode TShark utility
  - The most powerful display filters in the industry
  - Rich VoIP analysis
  - Live data can be read from Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI, and others
  - Coloring rules can be applied to the packet list for quick, intuitive analysis
  - Output can be exported to XML, PostScript®, CSV, or plain text

What is Wireshark?

- What we can:
  - Capture network traffic
  - Decode packet protocols using dissectors
  - Define filters – capture and display
  - Watch smart statistics
  - Analyze problems
  - Interactively browse that traffic
- Some examples people use Wireshark for:
  - Network administrators: **troubleshoot network problems**
  - Network security engineers: **examine security problems**
  - Developers: **debug protocol implementations**
  - People: **learn network protocol internals**

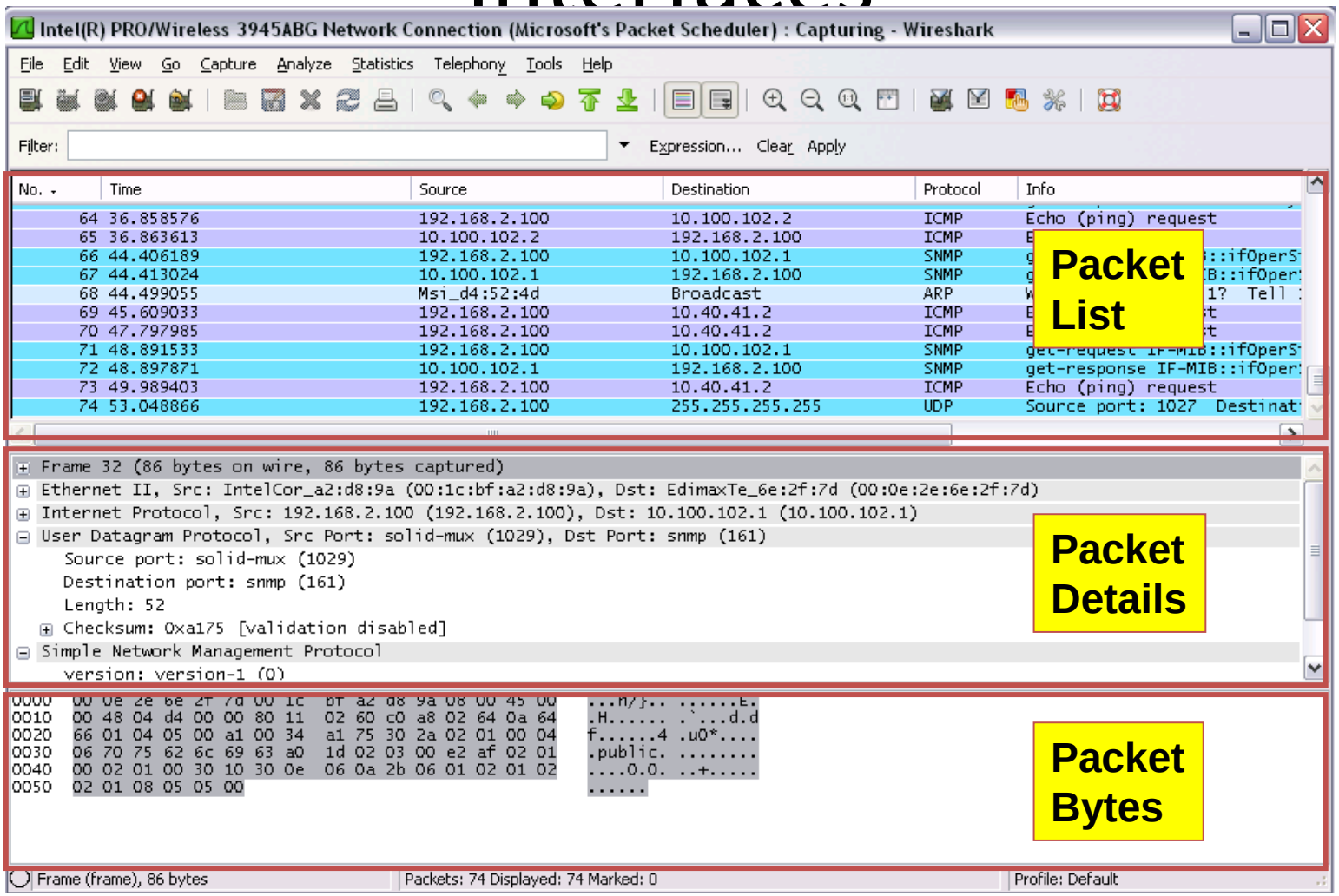
TCP/IP Traffic



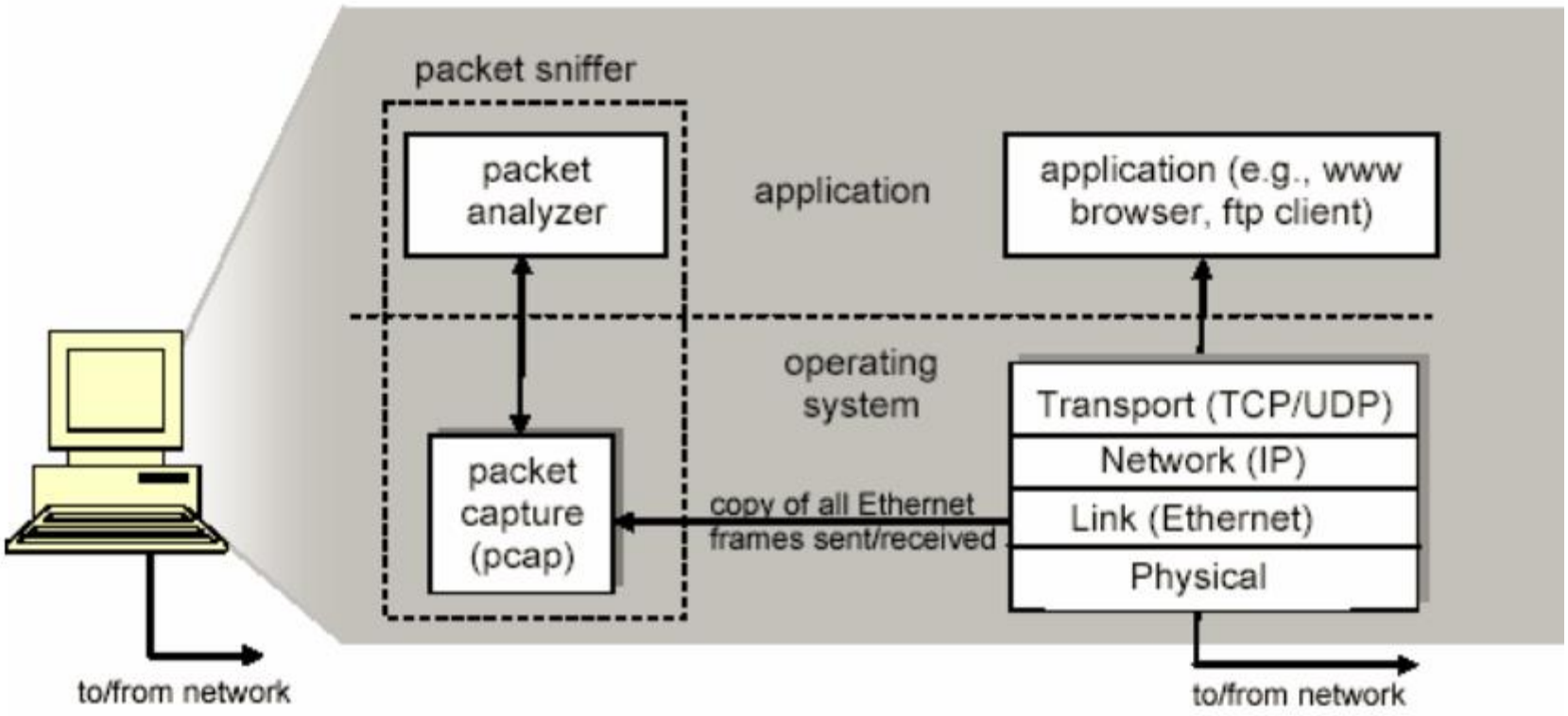
Encapsulation of Data in the TCP/IP Network Stack

WIRESHARK

Interfaces

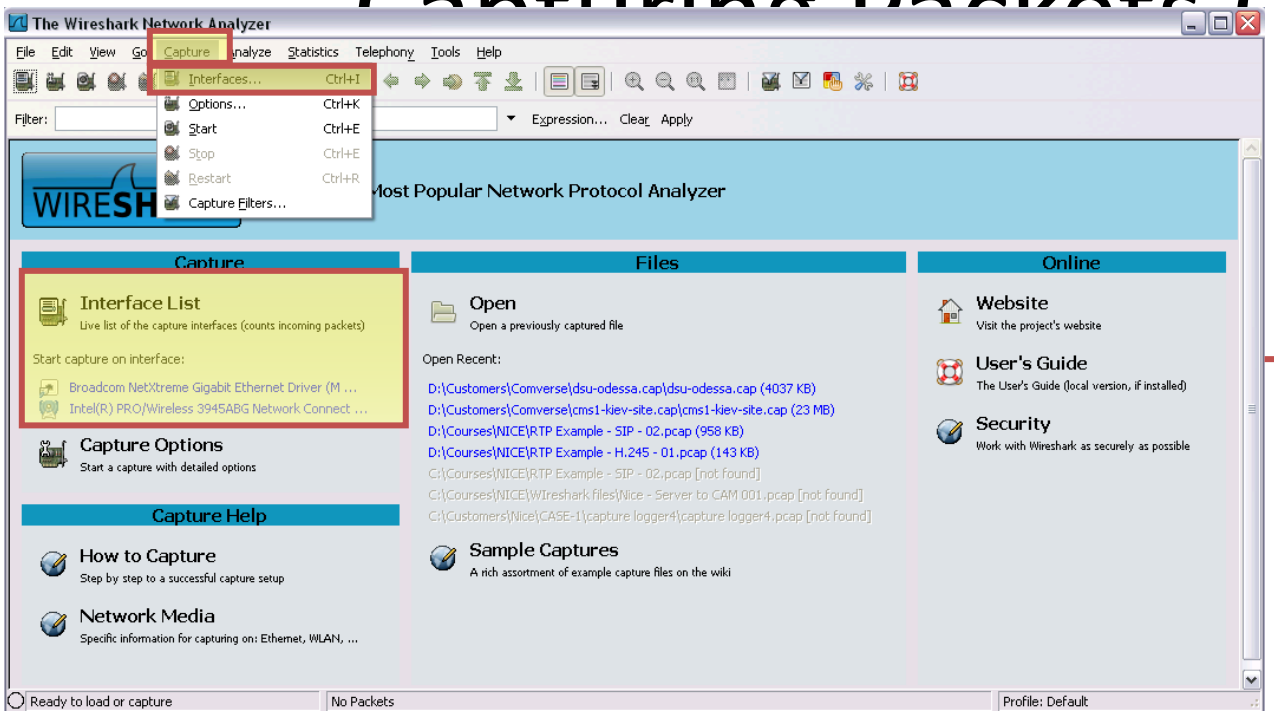


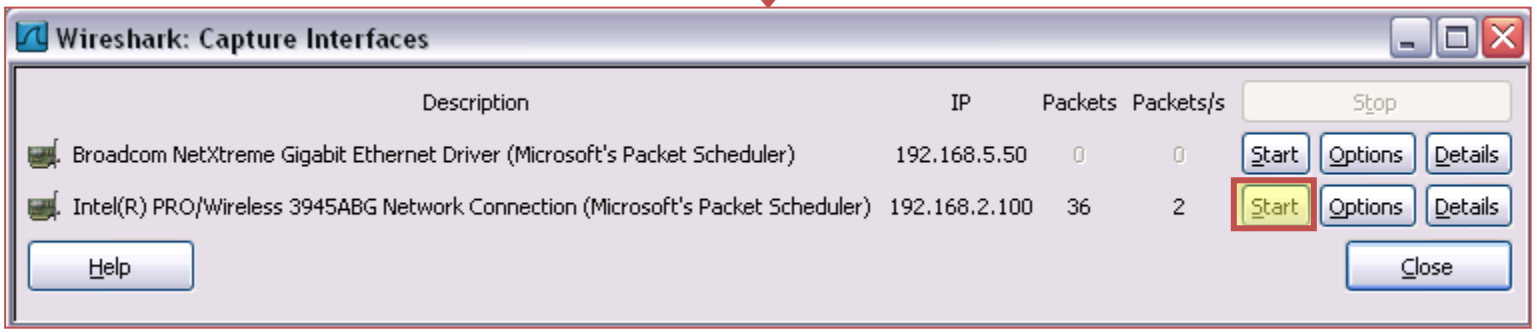
Packet Sniffer



WIRESHARK

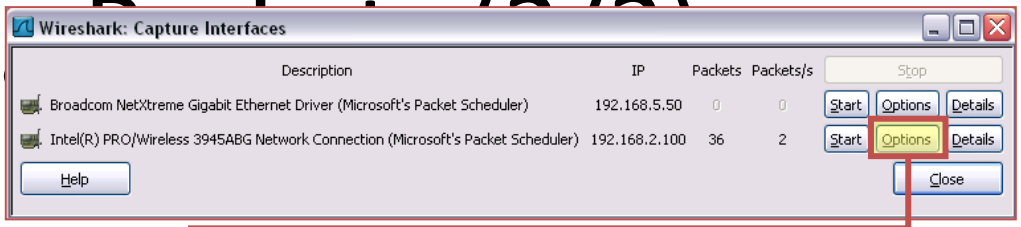
Capturing Packets (1/3)





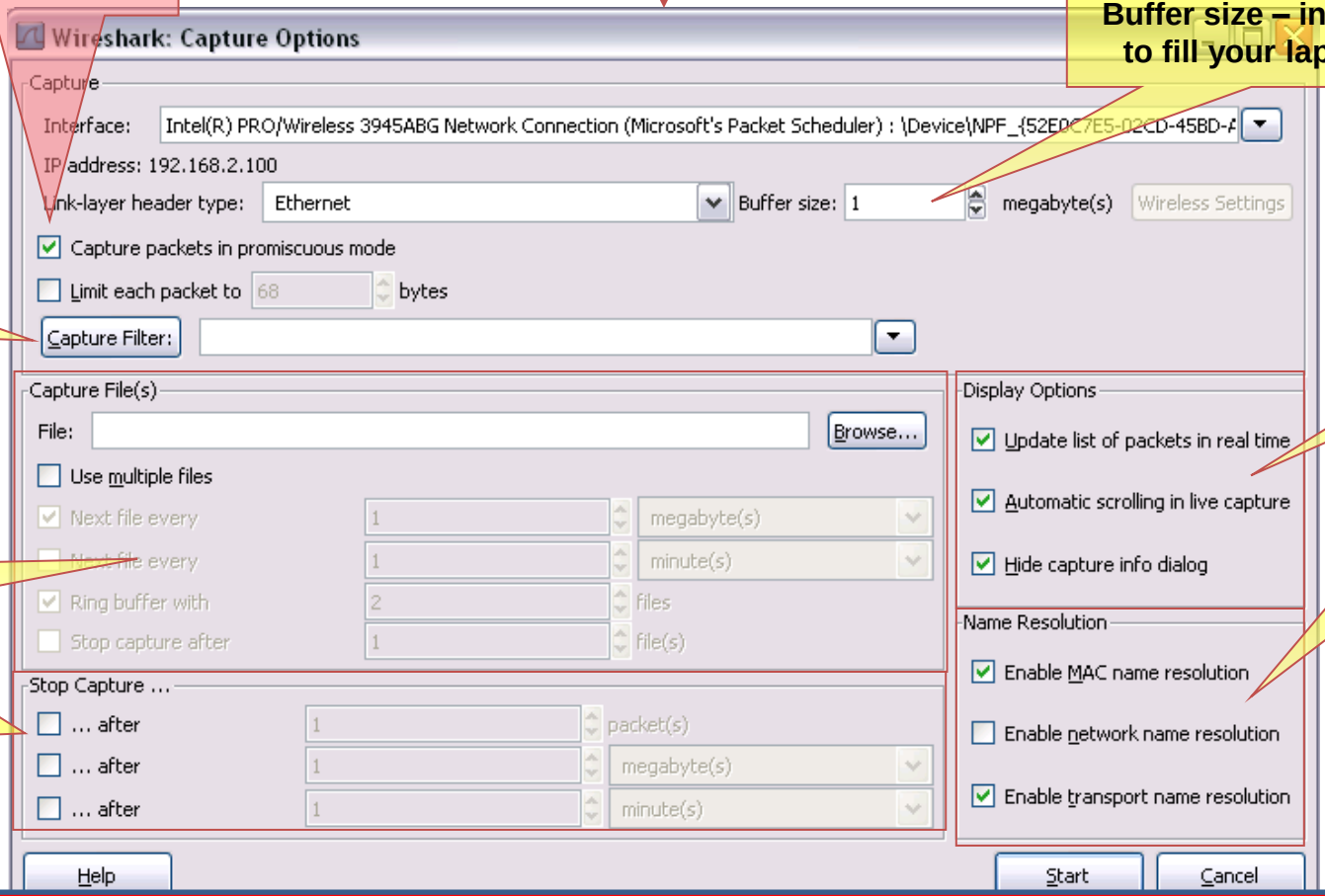
WIRESHARK

Capturin



Capture all packets on the network

Buffer size – in order not to fill your laptop disk



Capture filter

Display options

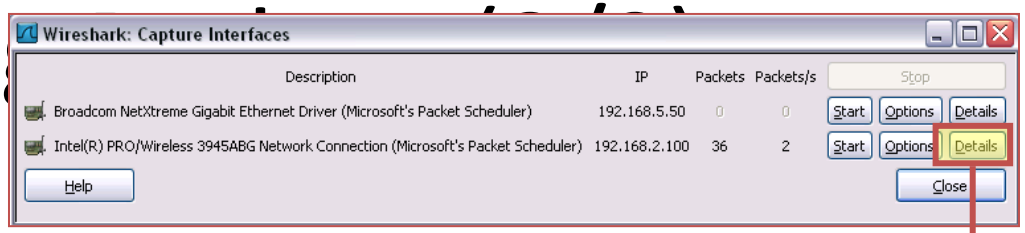
Capture in multiple files

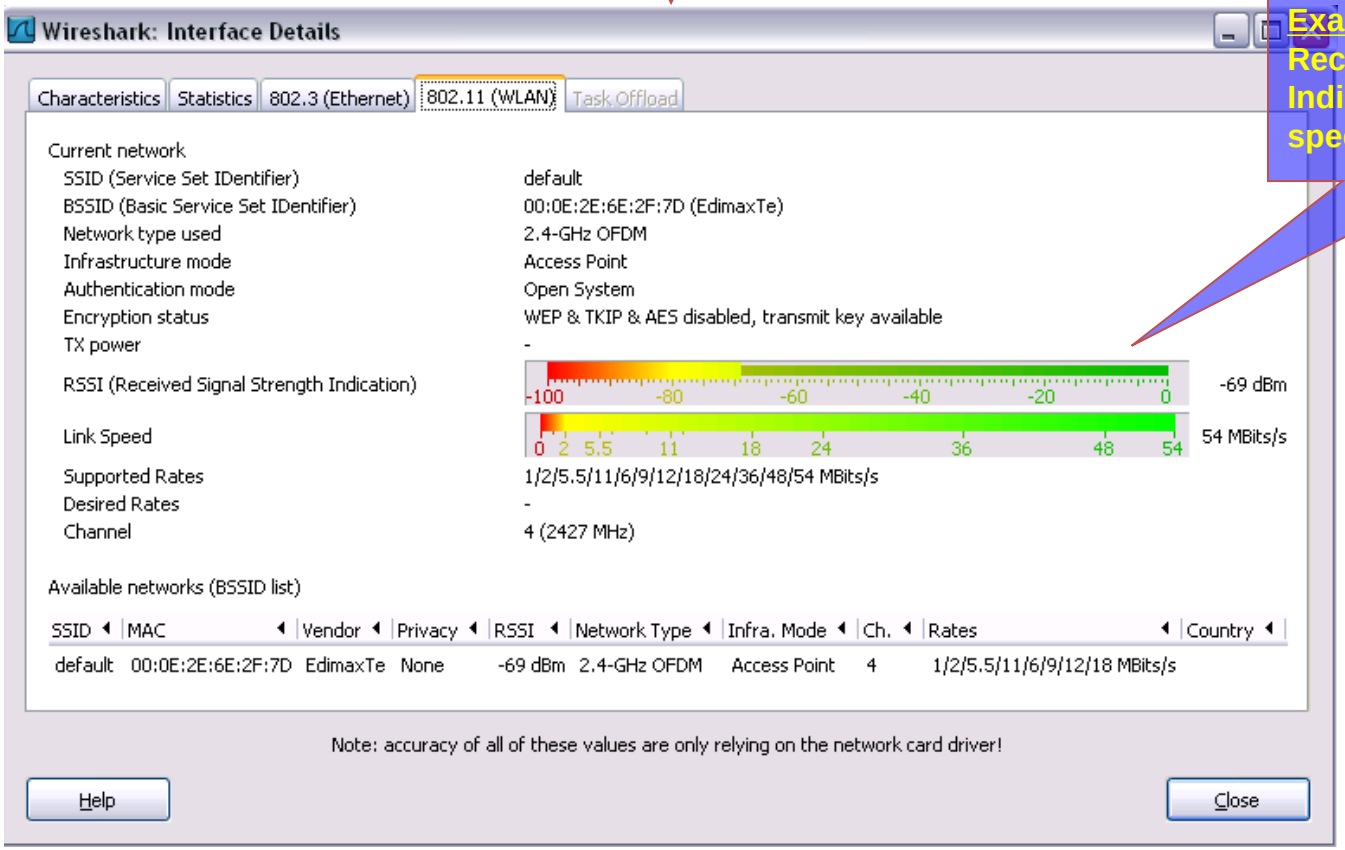
Name resolution options

When to automatically stop the capture

WIRESHARK

Capturin

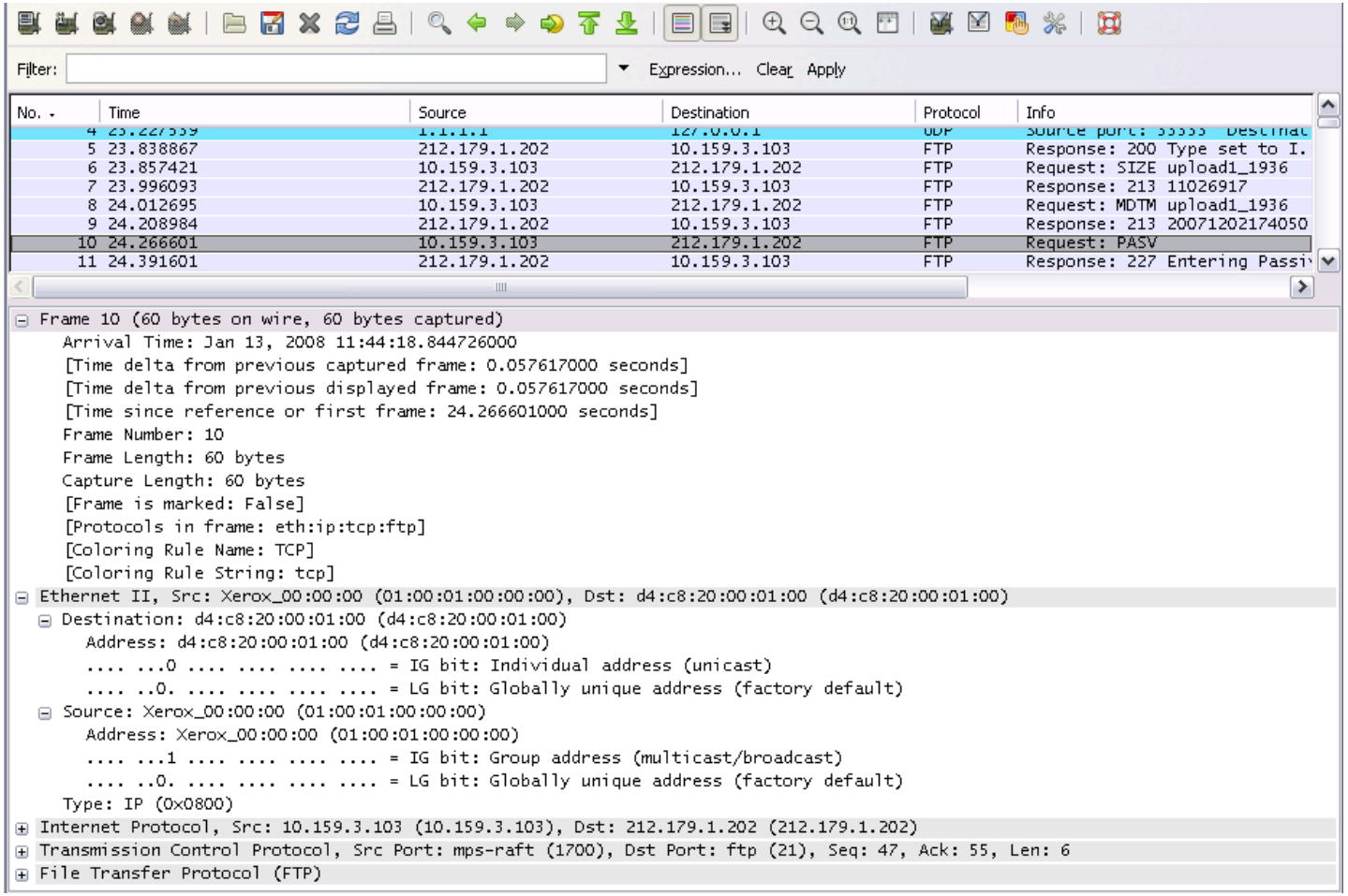




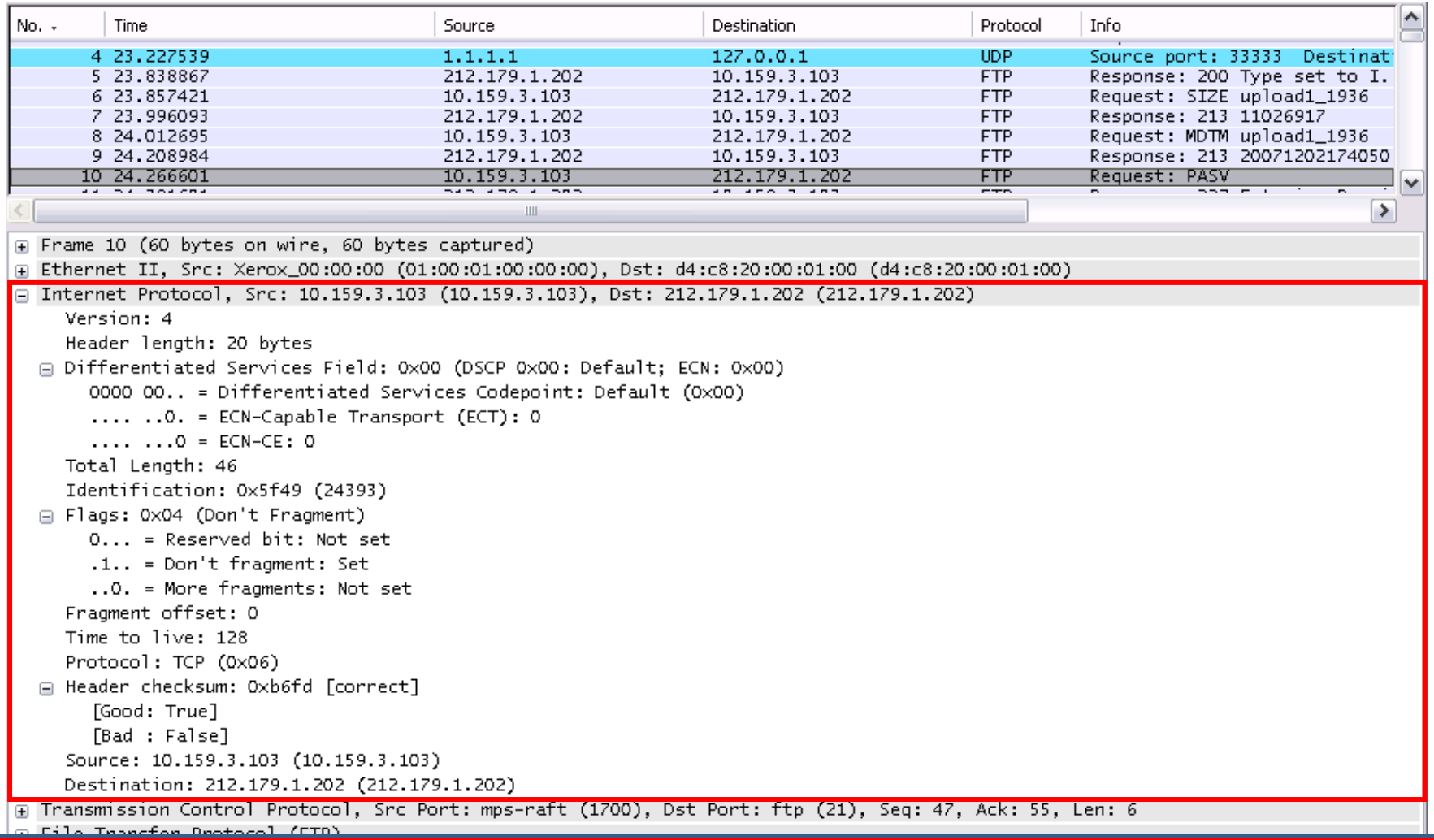
Example (W-LAN):
Received Signal Strength Indication (RSSI) and Link speed (BW)

Analyzing Packets (1/9)

❖ Ethernet Frame Example



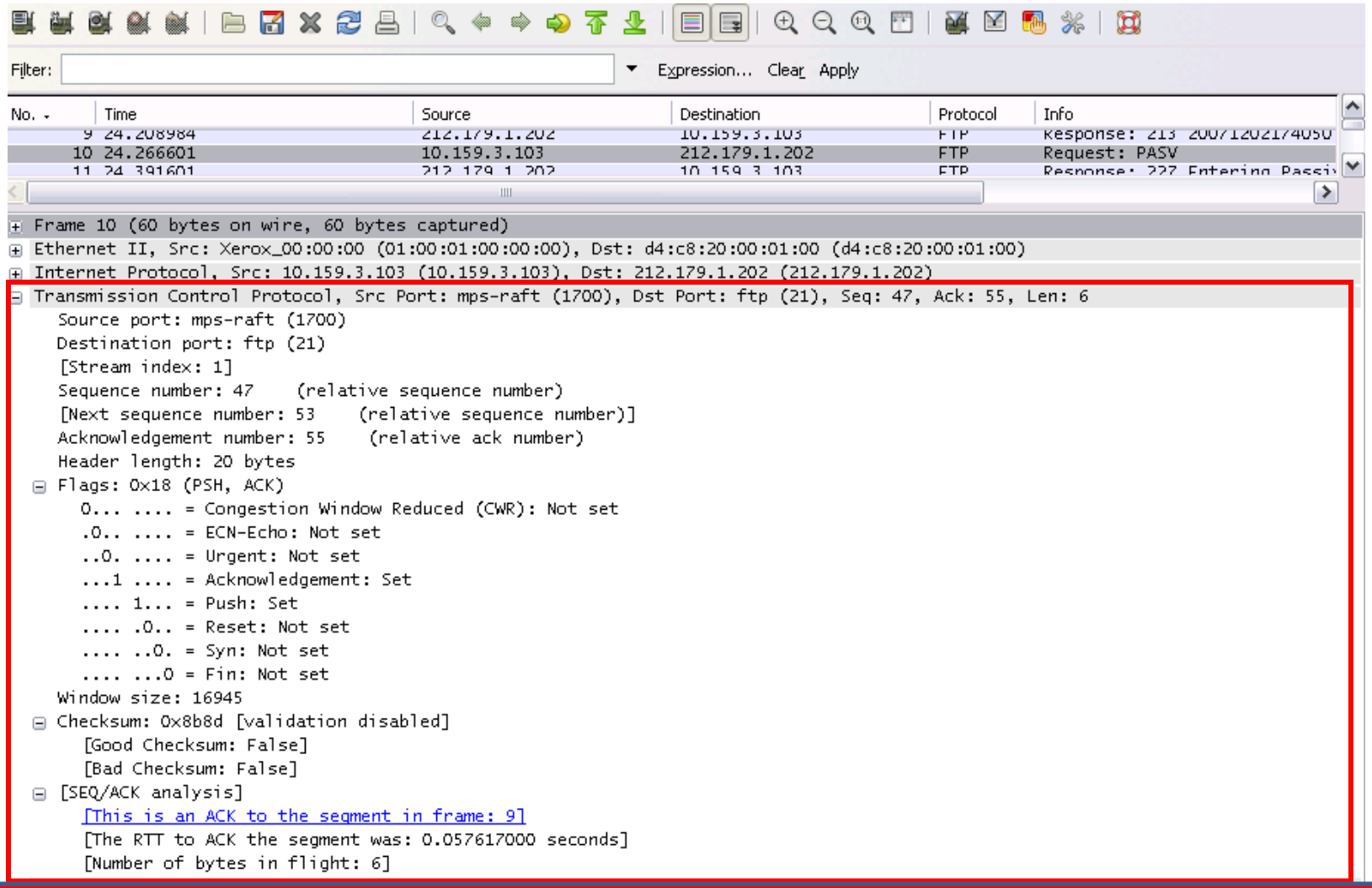
Analyzing Packets (2/9)



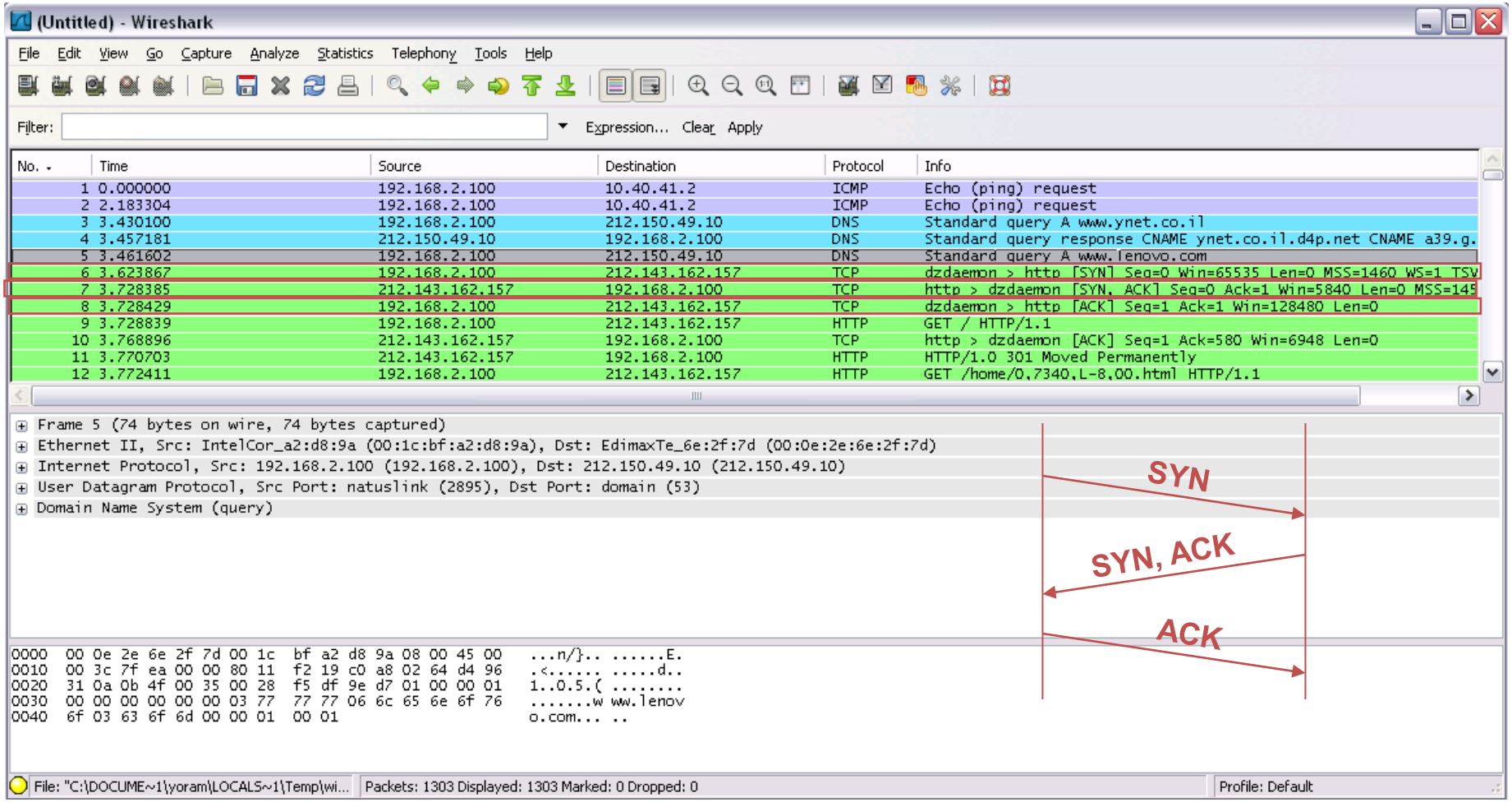
No.	Time	Source	Destination	Protocol	Info
4	23.227539	1.1.1.1	127.0.0.1	UDP	Source port: 33333 Destinat
5	23.838867	212.179.1.202	10.159.3.103	FTP	Response: 200 Type set to I.
6	23.857421	10.159.3.103	212.179.1.202	FTP	Request: SIZE upload1_1936
7	23.996093	212.179.1.202	10.159.3.103	FTP	Response: 213 11026917
8	24.012695	10.159.3.103	212.179.1.202	FTP	Request: MDTM upload1_1936
9	24.208984	212.179.1.202	10.159.3.103	FTP	Response: 213 20071202174050
10	24.266601	10.159.3.103	212.179.1.202	FTP	Request: PASV

```
⊞ Frame 10 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: Xerox_00:00:00 (01:00:01:00:00:00), Dst: d4:c8:20:00:01:00 (d4:c8:20:00:01:00)
⊟ Internet Protocol, Src: 10.159.3.103 (10.159.3.103), Dst: 212.179.1.202 (212.179.1.202)
    Version: 4
    Header length: 20 bytes
  ⊟ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
      0000 00.. = Differentiated Services Codepoint: Default (0x00)
      .... ..0. = ECN-Capable Transport (ECT): 0
      .... ...0 = ECN-CE: 0
    Total Length: 46
    Identification: 0x5f49 (24393)
  ⊟ Flags: 0x04 (Don't Fragment)
      0... = Reserved bit: Not set
      .1.. = Don't fragment: Set
      ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (0x06)
  ⊟ Header checksum: 0xb6fd [correct]
      [Good: True]
      [Bad : False]
    Source: 10.159.3.103 (10.159.3.103)
    Destination: 212.179.1.202 (212.179.1.202)
⊞ Transmission Control Protocol, Src Port: mps-raft (1700), Dst Port: ftp (21), Seq: 47, Ack: 55, Len: 6
```

Analyzing Packets (3/9)

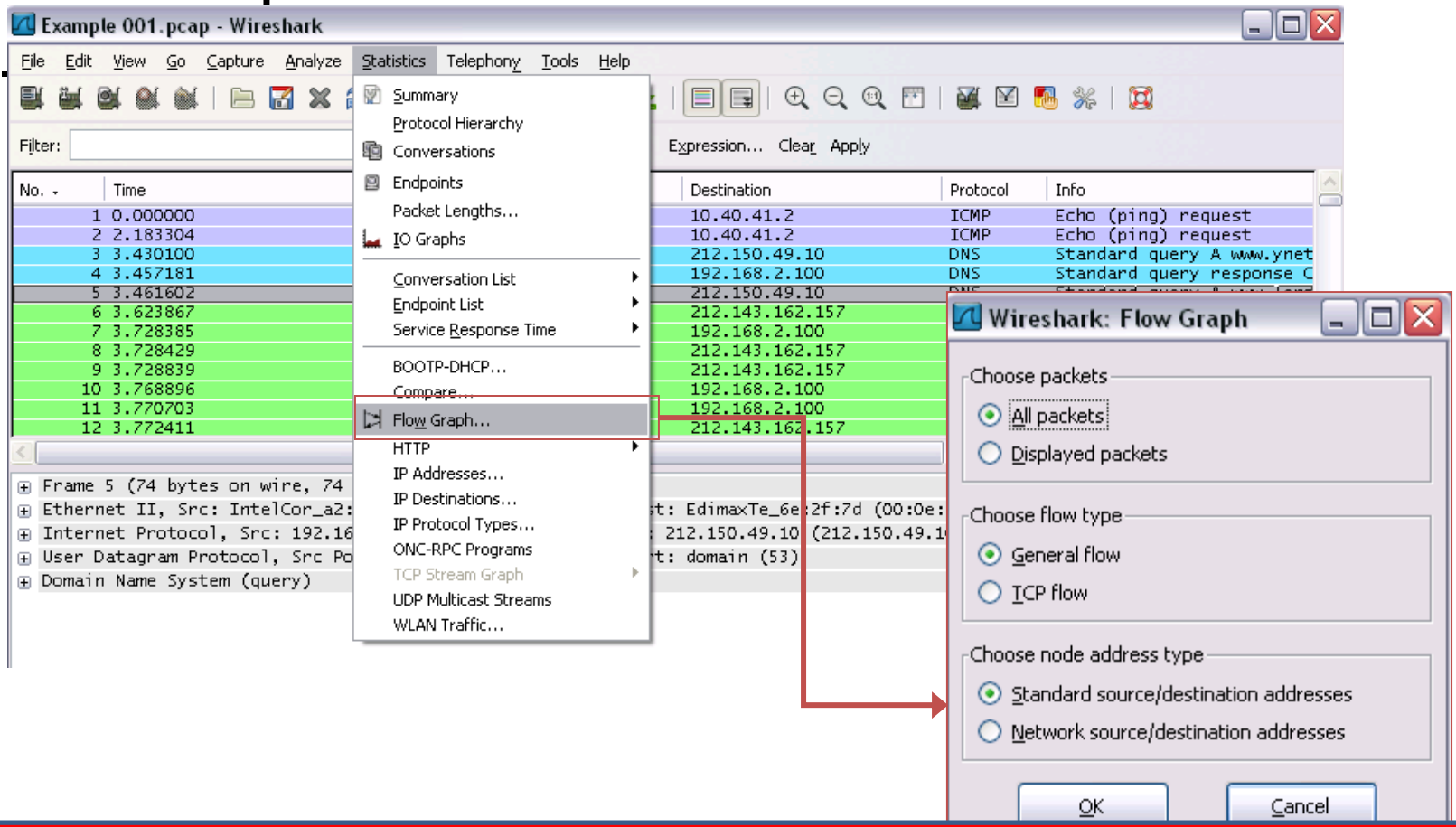


Analyzing Packets (4/9)

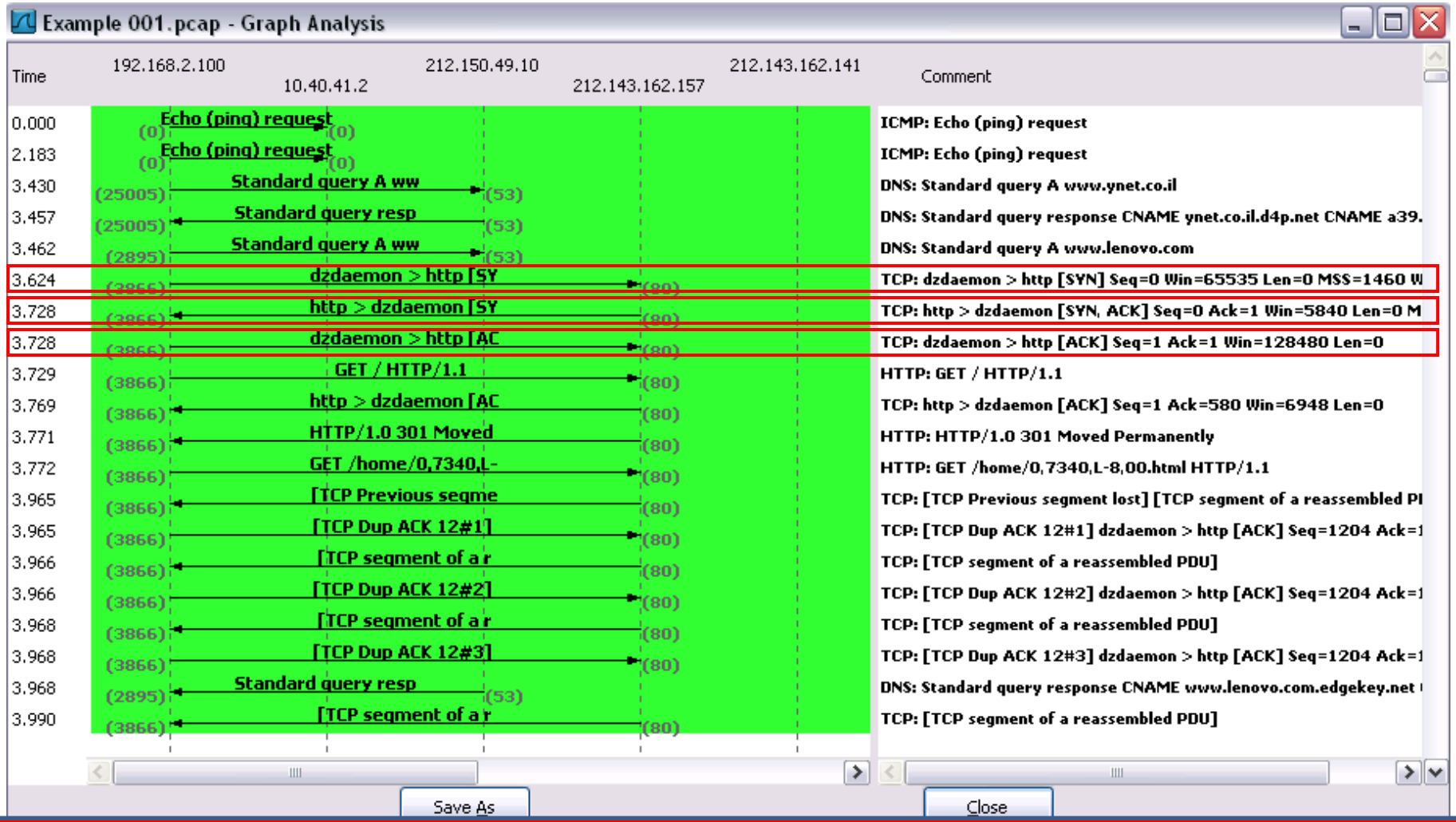


Analyzing Packets (5/9)

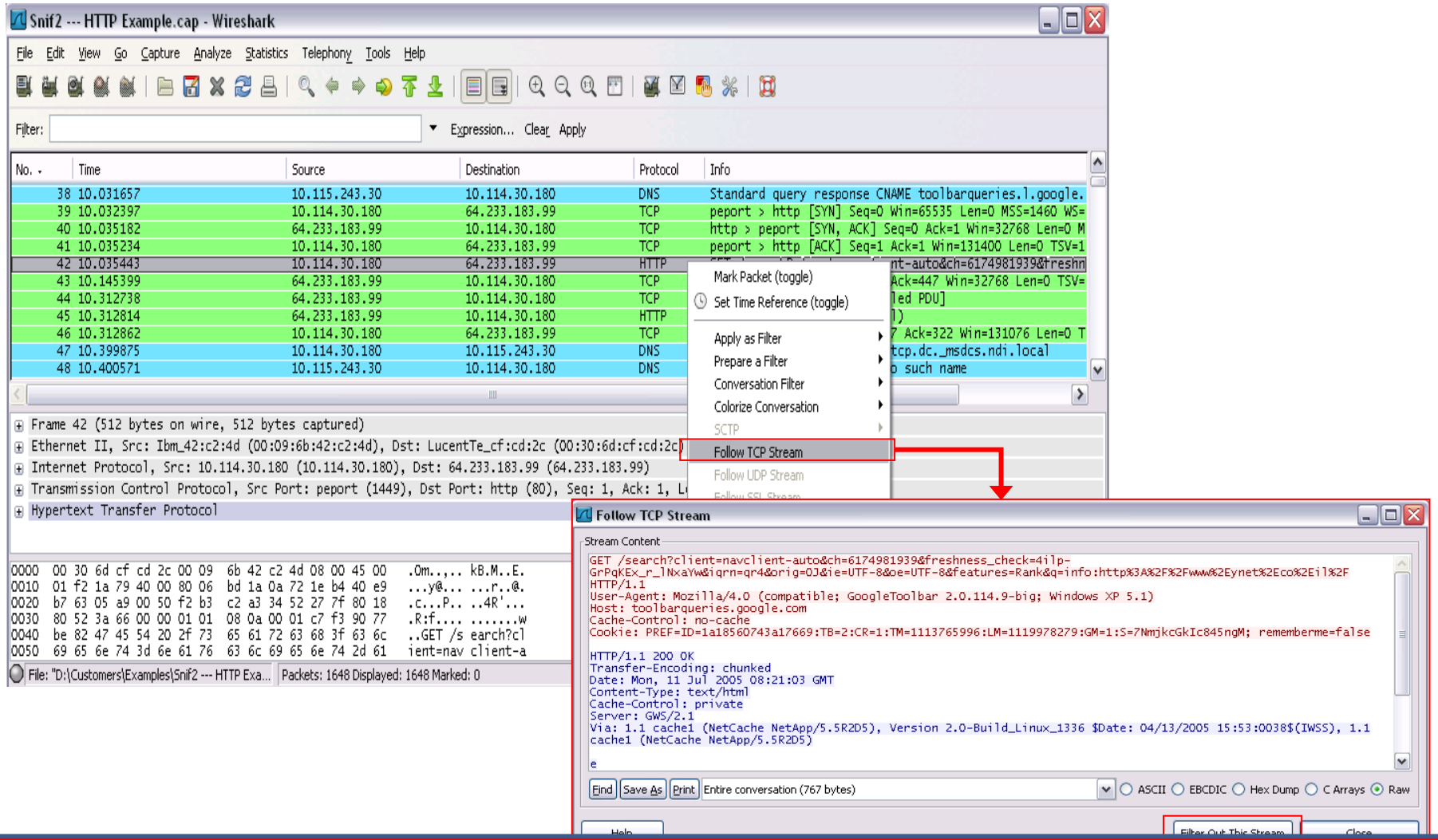
- Flow Graph



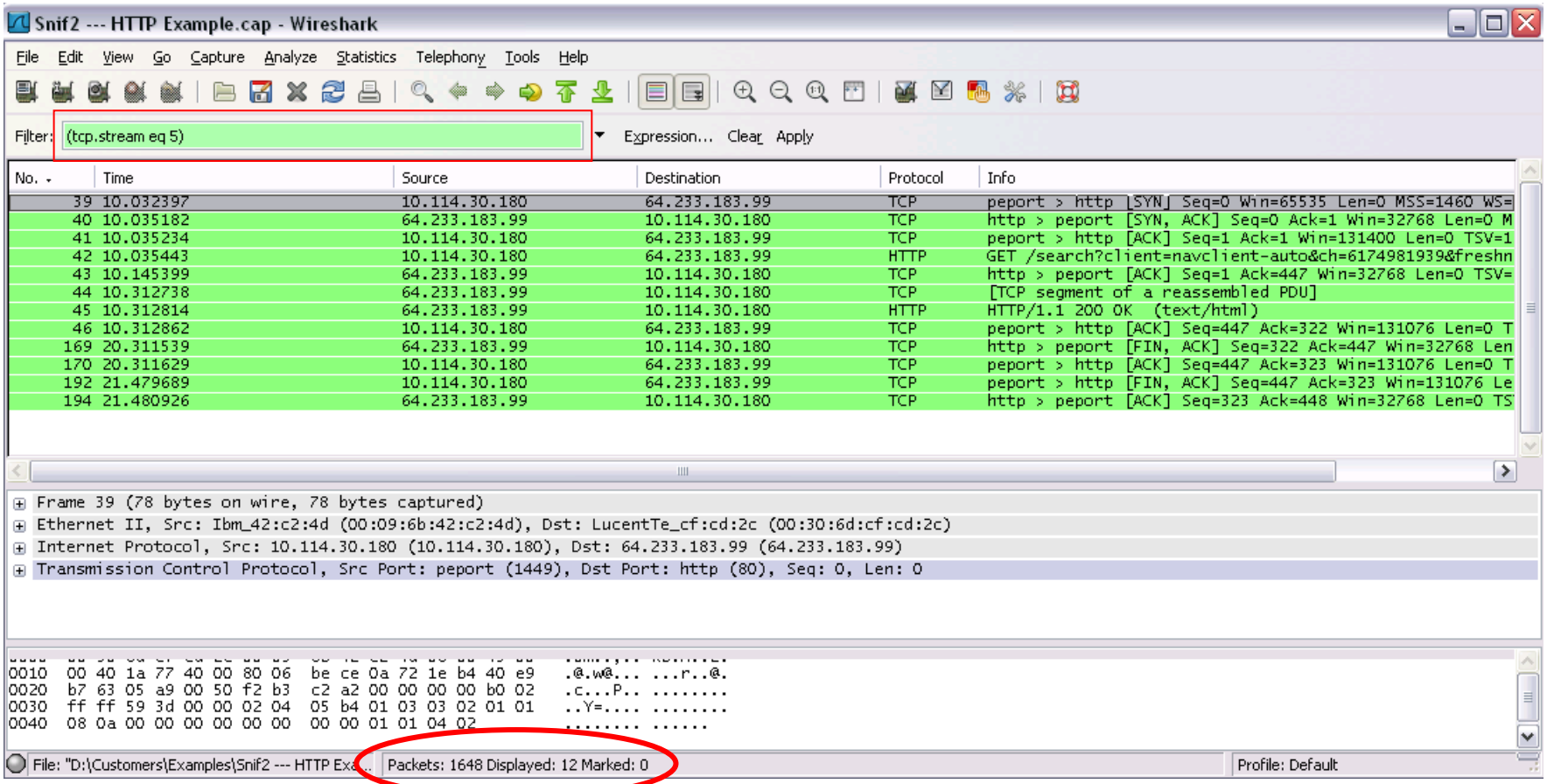
Analyzing Packets (6/9)



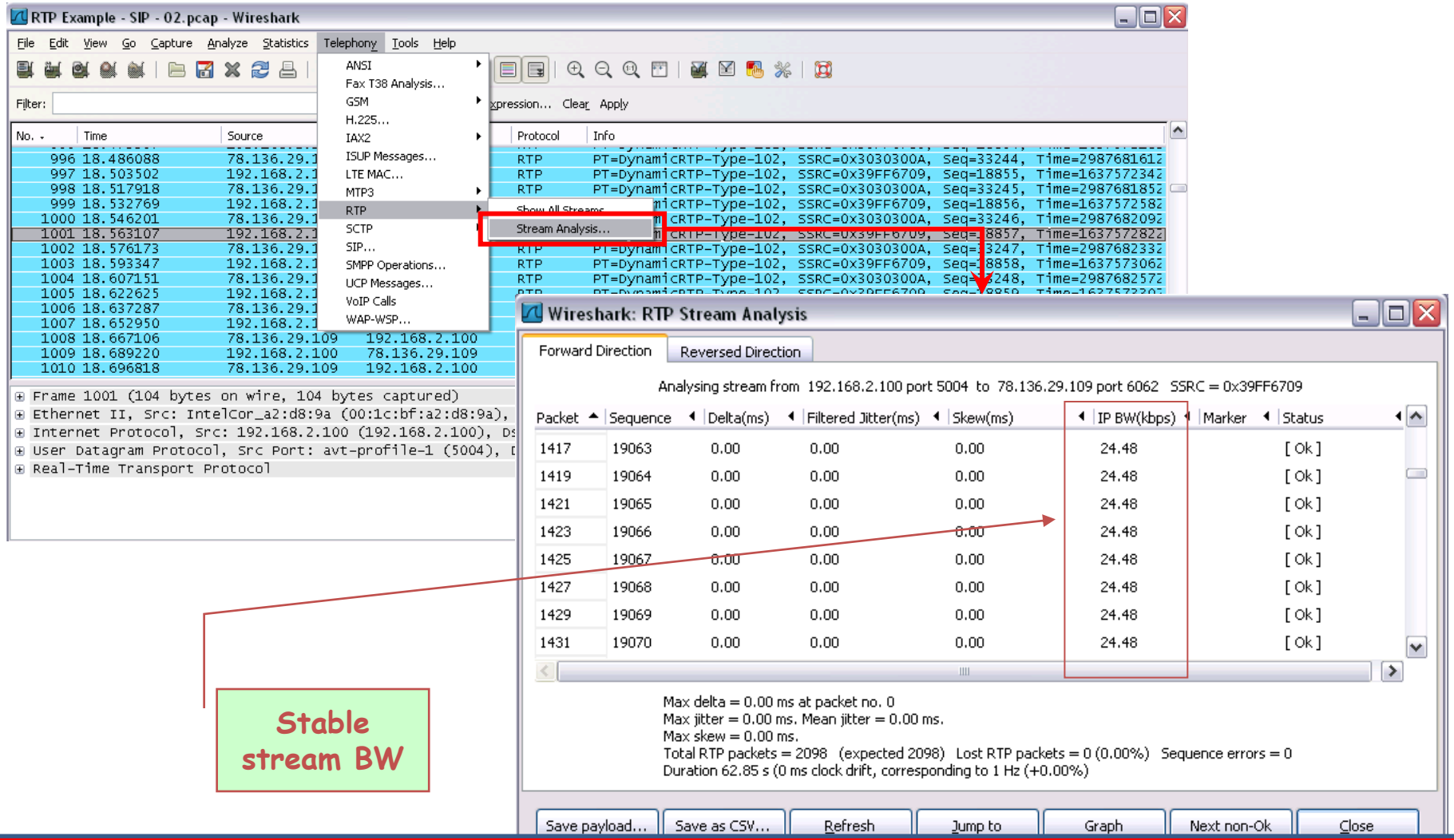
Analyzing Packets (7/9)



Analyzing Packets (8/9)



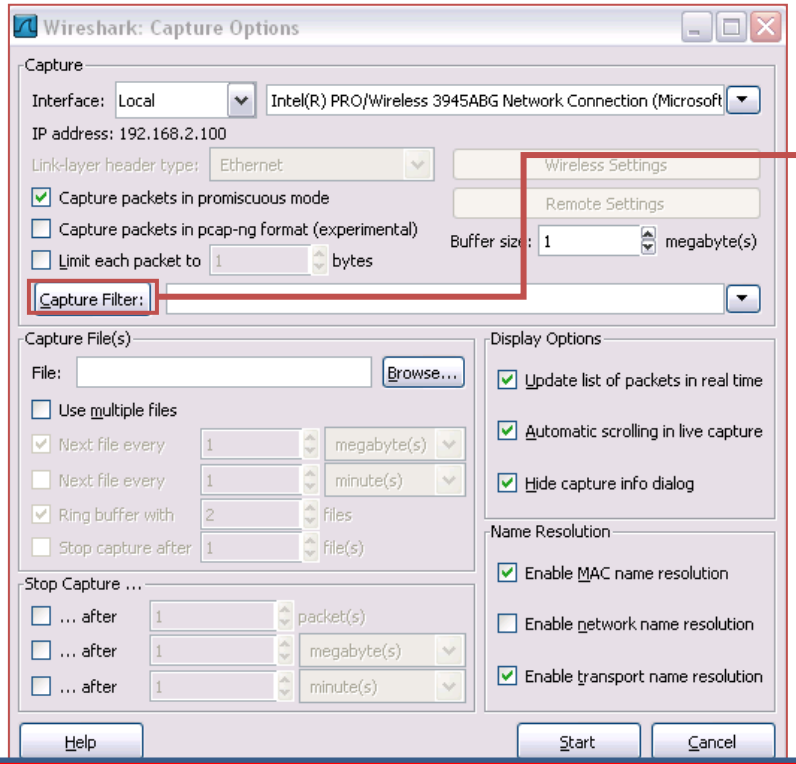
Analyzing Packets (9/9)

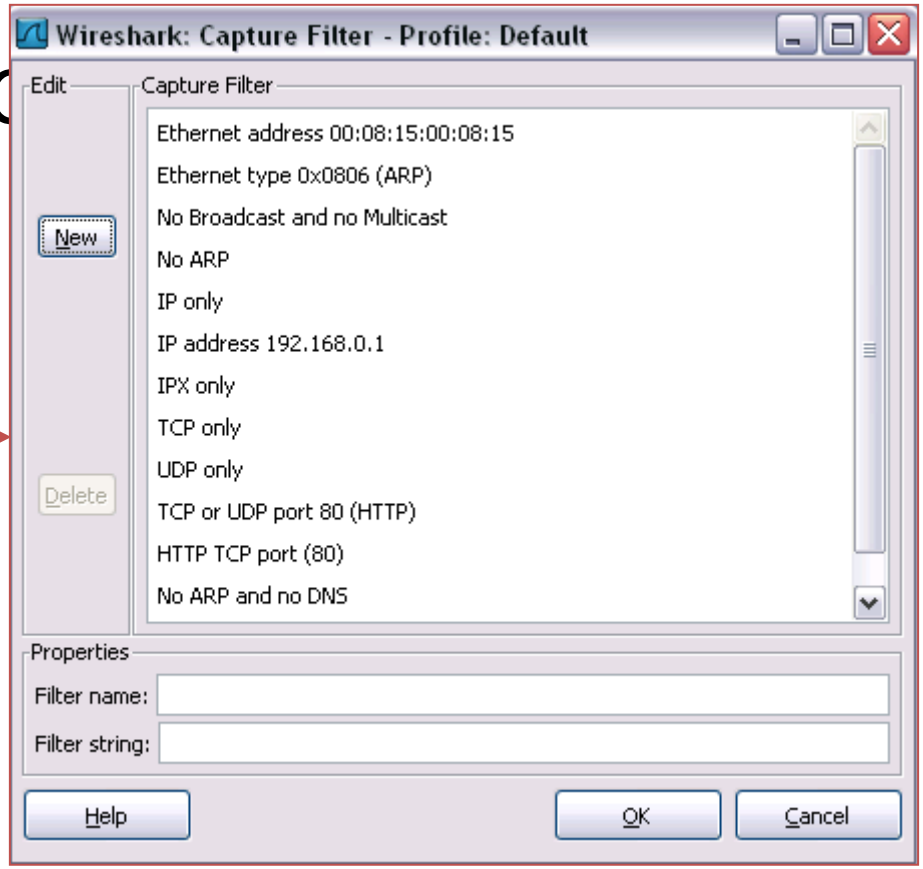


Filtering Packets (1/4)

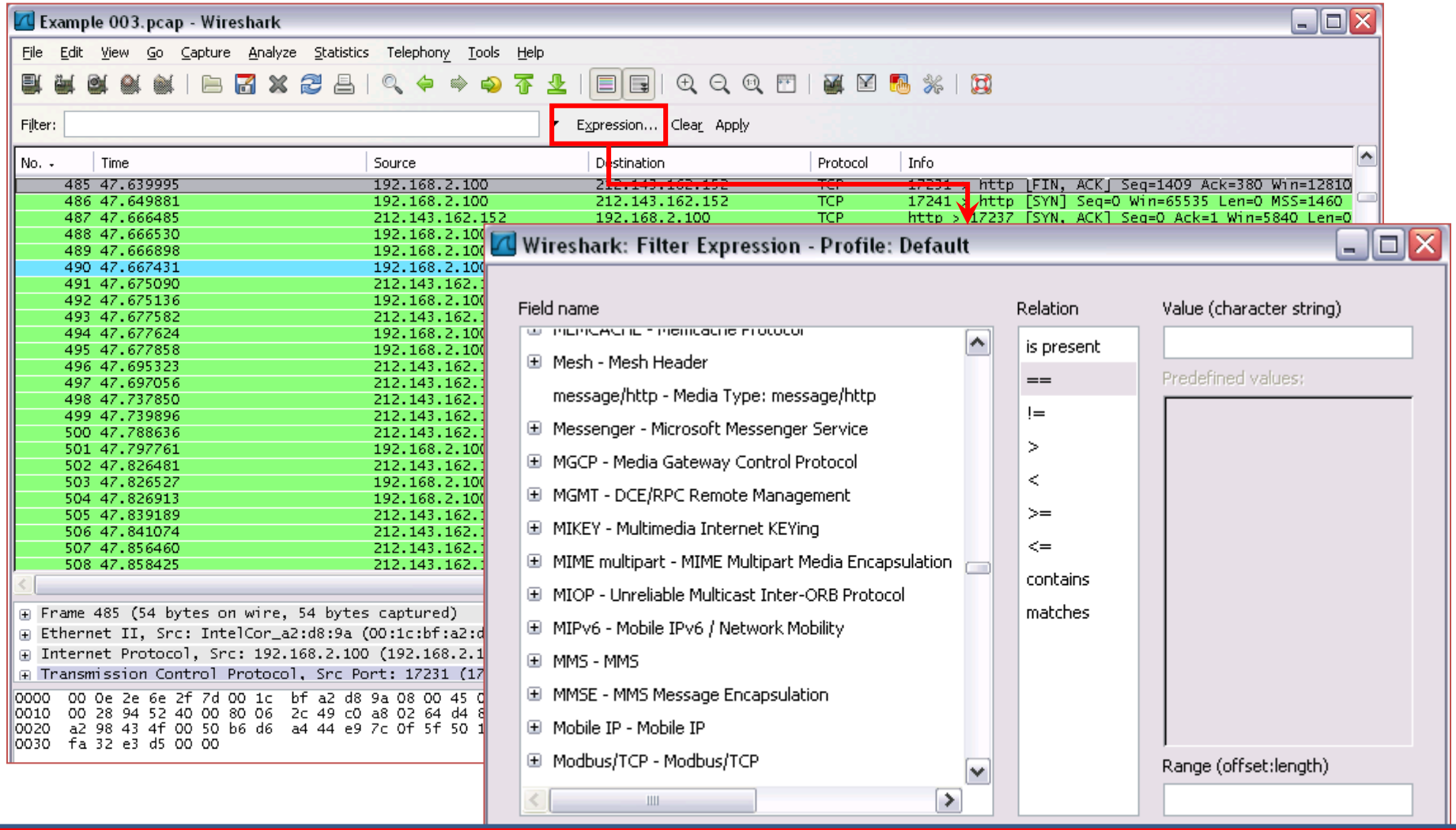
- Applying Filter when C

Capture → Interfaces → Options:





Filtering Packets (2/4)



Filtering Packets (3/4)

- Examples:
  - Capture only traffic to or from IP address 172.18.5.4
    - **host 172.18.5.4**
  - Capture traffic to or from a range of IP addresses
    - **net 192.168.0.0/24**
    - **net 192.168.0.0 mask 255.255.255.0**
  - Capture traffic from a range of IP addresses
    - **src net 192.168.0.0/24**
    - **src net 192.168.0.0 mask 255.255.255.0**
  - Capture traffic to a range of IP addresses
    - **dst net 192.168.0.0/24**
    - **dst net 192.168.0.0 mask 255.255.255.0**
  - Capture only DNS (port 53) traffic
    - **port 53**
  - Capture non-HTTP and non-SMTP traffic on your server
    - **host www.example.com and not (port 80 or port 25)**
    - **host www.example.com and not port 80 and not port 25**

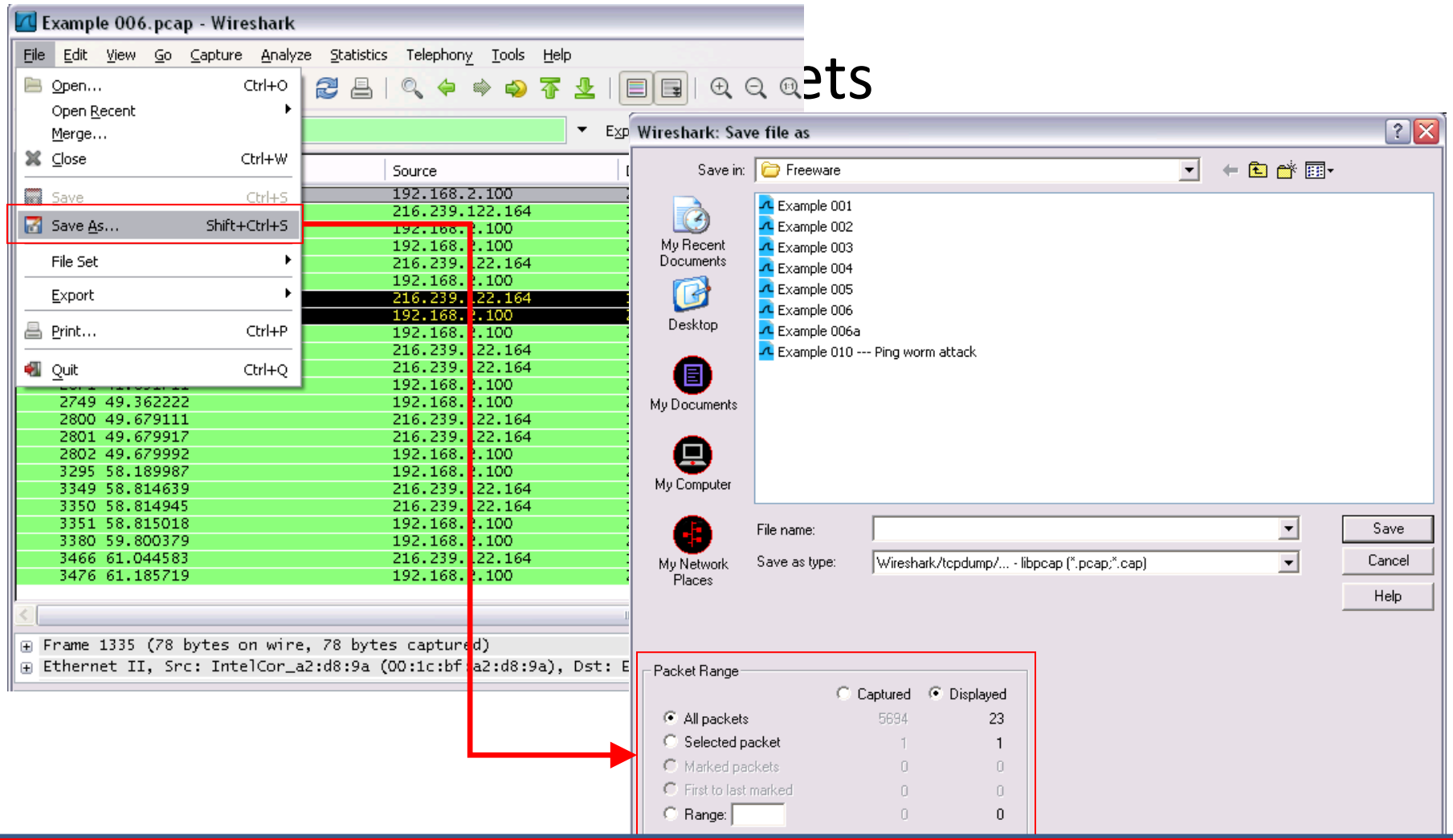
Filtering Packets (4/4)

- Examples:
  - Capture except all ARP and DNS traffic
    - **port not 53 and not arp**
  - Capture traffic within a range of ports
    - **(tcp[2:2] > 1500 and tcp[2:2] < 1550) or (tcp[4:2] > 1500 and tcp[4:2] < 1550)**
    - **tcp portrange 1501-1549**
  - Capture only Ethernet type EAPOL
    - **ether proto 0x888e**
  - Capture only IP traffic

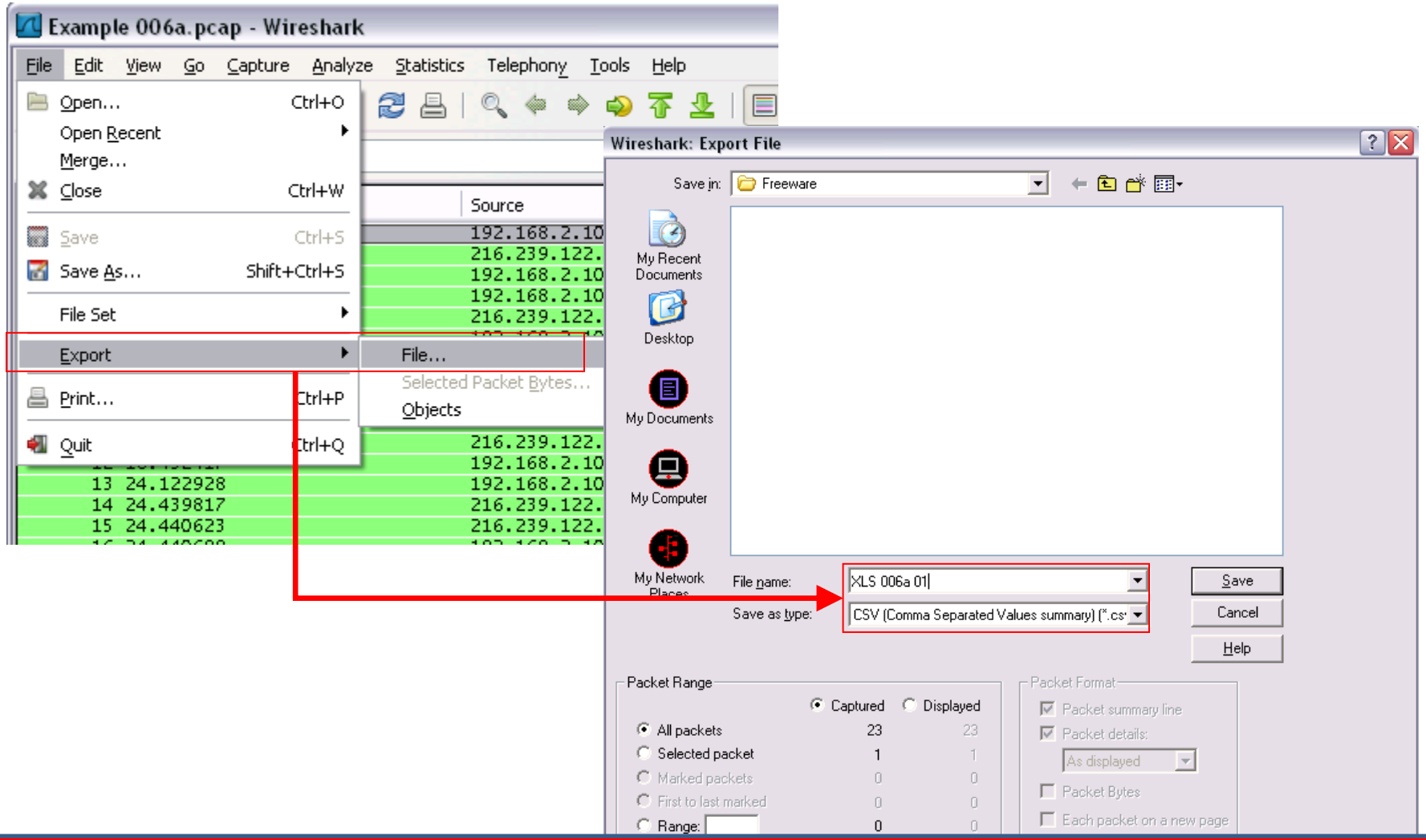
    (the shortest filter, but sometimes very useful to get rid of lower layer protocols like ARP and STP)
    - **ip**
  - Capture only unicast traffic

    (useful to get rid of noise on the network if you only want to see traffic to and from your machine, not, for example, broadcast and multicast announcements)
    - **not broadcast and not multicast**

Saving and Manipulating Packets (1/3)



Saving and Manipulating Packets (2/3)

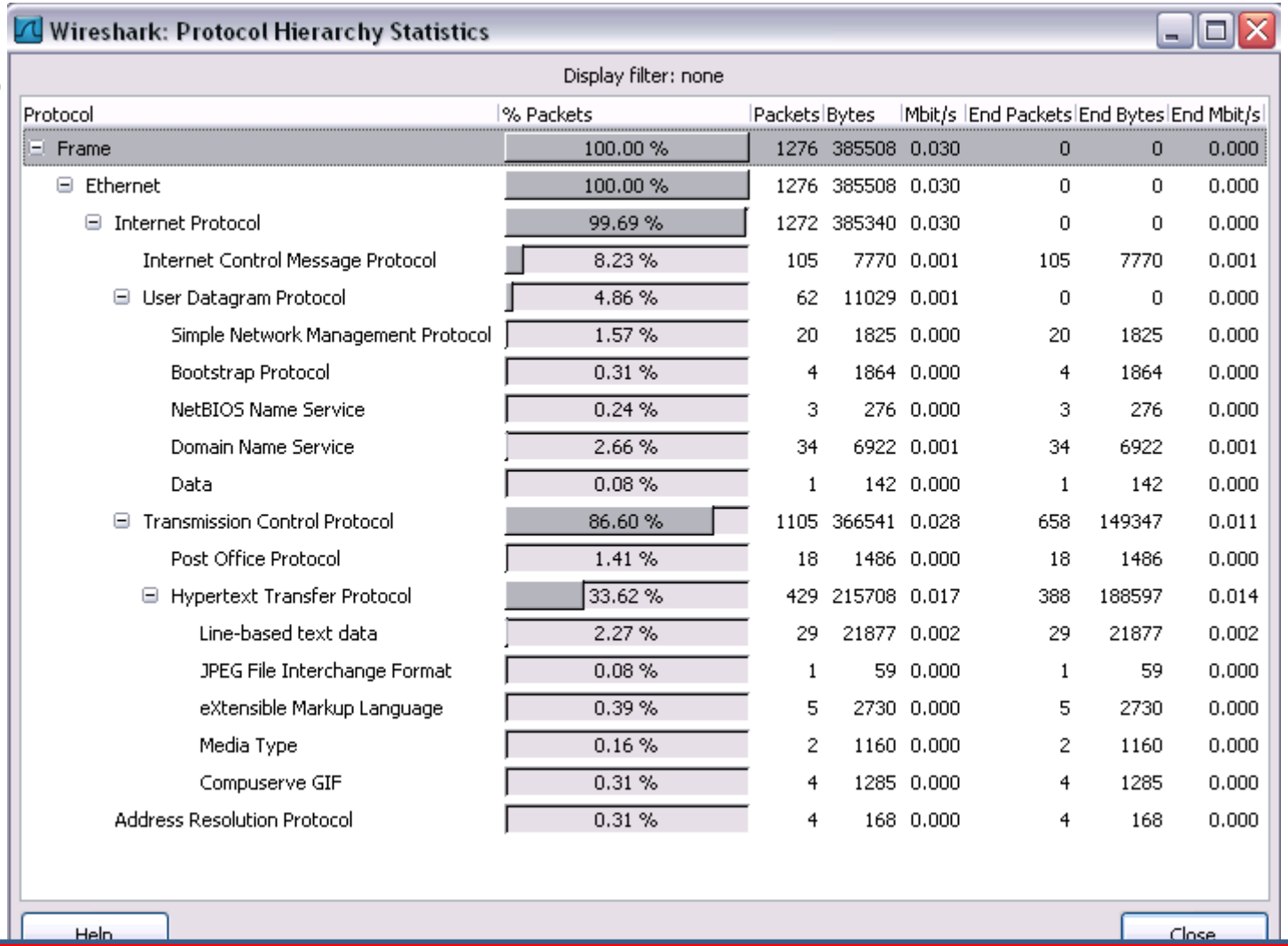


Saving and Manipulating Packets (3/3)

No.	Time	Time Variation	Source	Destination	Protocol	Info
1	0	0	192.168.2.100	216.239.122.164	TCP	27837 > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1 TSV=0 TSER=0
2	0.226724	0.226724	216.239.122.164	192.168.2.100	TCP	http > 27837 [SYN, ACK] Seq=0 Ack=1 Win=8190 Len=0 MSS=1380
3	0.226772	4.8E-05	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
4	0.227146	0.227098	192.168.2.100	216.239.122.164	HTTP	GET /i/b.jpg HTTP/1.1
5	0.700674	0.473576	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
6	0.883533	0.409957	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=649 Ack=767 Win=64769 Len=0
7	**1.161312**	**0.751355**	**216.239.122.164**	**192.168.2.100**	**HTTP**	**[TCP Retransmission] HTTP/1.1 200 OK (JPEG JFIF image)**
8	1.161361	0.410006	192.168.2.100	216.239.122.164	TCP	[TCP Dup ACK 6#1] 27837 > http [ACK] Seq=649 Ack=767 Win=64769 Len=0
9	16.211468	15.801462	192.168.2.100	216.239.122.164	HTTP	GET /i/b.jpg HTTP/1.1
10	16.452024	0.650562	216.239.122.164	192.168.2.100	TCP	[TCP segment of a reassembled PDU]
11	16.452343	15.801781	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
12	16.452417	0.650636	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=1539 Ack=1533 Win=65535 Len=0
13	24.122928	23.472292	192.168.2.100	216.239.122.164	HTTP	GET /i/b.jpg HTTP/1.1
14	24.439817	0.967525	216.239.122.164	192.168.2.100	TCP	[TCP segment of a reassembled PDU]
15	24.440623	23.473098	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
16	24.440698	0.9676	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=2384 Ack=2299 Win=64769 Len=0
17	32.950693	31.983093	192.168.2.100	216.239.122.164	HTTP	GET /i/b.jpg HTTP/1.1
18	33.575345	1.592252	216.239.122.164	192.168.2.100	TCP	[TCP segment of a reassembled PDU]
19	33.575651	31.983399	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (JPEG JFIF image)
20	33.575724	1.592325	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=3269 Ack=3065 Win=65535 Len=0
21	34.561085	32.96876	192.168.2.100	216.239.122.164	HTTP	GET /b.gif HTTP/1.1
22	35.805289	2.836529	216.239.122.164	192.168.2.100	HTTP	HTTP/1.1 200 OK (GIF89a)
23	35.946425	33.109896	192.168.2.100	216.239.122.164	TCP	27837 > http [ACK] Seq=4080 Ack=3567 Win=65033 Len=0

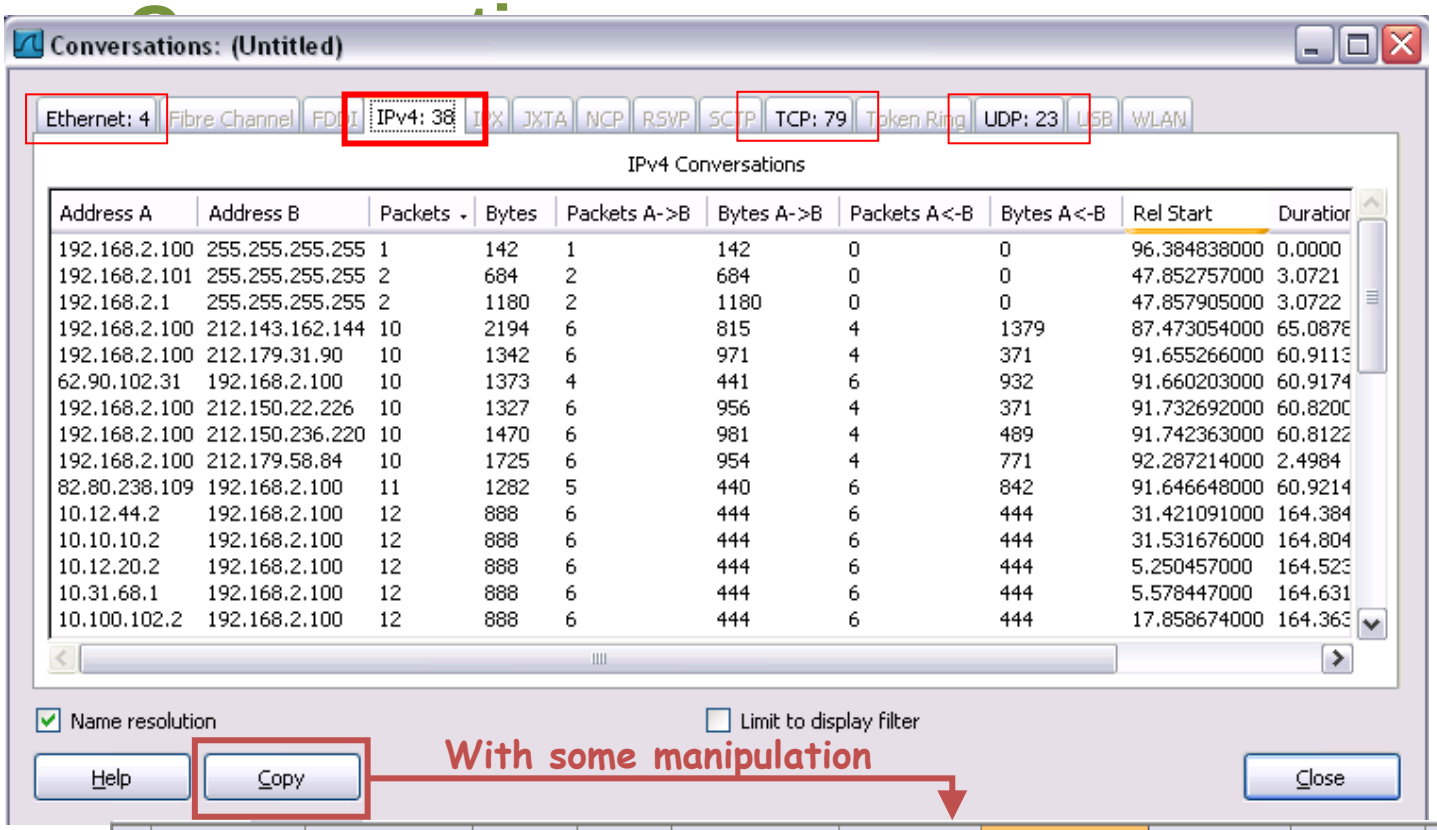
Packet Statistics (1/8)

- P



Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00 %	1276	385508	0.030	0	0	0.000
Ethernet	100.00 %	1276	385508	0.030	0	0	0.000
Internet Protocol	99.69 %	1272	385340	0.030	0	0	0.000
Internet Control Message Protocol	8.23 %	105	7770	0.001	105	7770	0.001
User Datagram Protocol	4.86 %	62	11029	0.001	0	0	0.000
Simple Network Management Protocol	1.57 %	20	1825	0.000	20	1825	0.000
Bootstrap Protocol	0.31 %	4	1864	0.000	4	1864	0.000
NetBIOS Name Service	0.24 %	3	276	0.000	3	276	0.000
Domain Name Service	2.66 %	34	6922	0.001	34	6922	0.001
Data	0.08 %	1	142	0.000	1	142	0.000
Transmission Control Protocol	86.60 %	1105	366541	0.028	658	149347	0.011
Post Office Protocol	1.41 %	18	1486	0.000	18	1486	0.000
Hypertext Transfer Protocol	33.62 %	429	215708	0.017	388	188597	0.014
Line-based text data	2.27 %	29	21877	0.002	29	21877	0.002
JPEG File Interchange Format	0.08 %	1	59	0.000	1	59	0.000
eXtensible Markup Language	0.39 %	5	2730	0.000	5	2730	0.000
Media Type	0.16 %	2	1160	0.000	2	1160	0.000
Compuserve GIF	0.31 %	4	1285	0.000	4	1285	0.000
Address Resolution Protocol	0.31 %	4	168	0.000	4	168	0.000

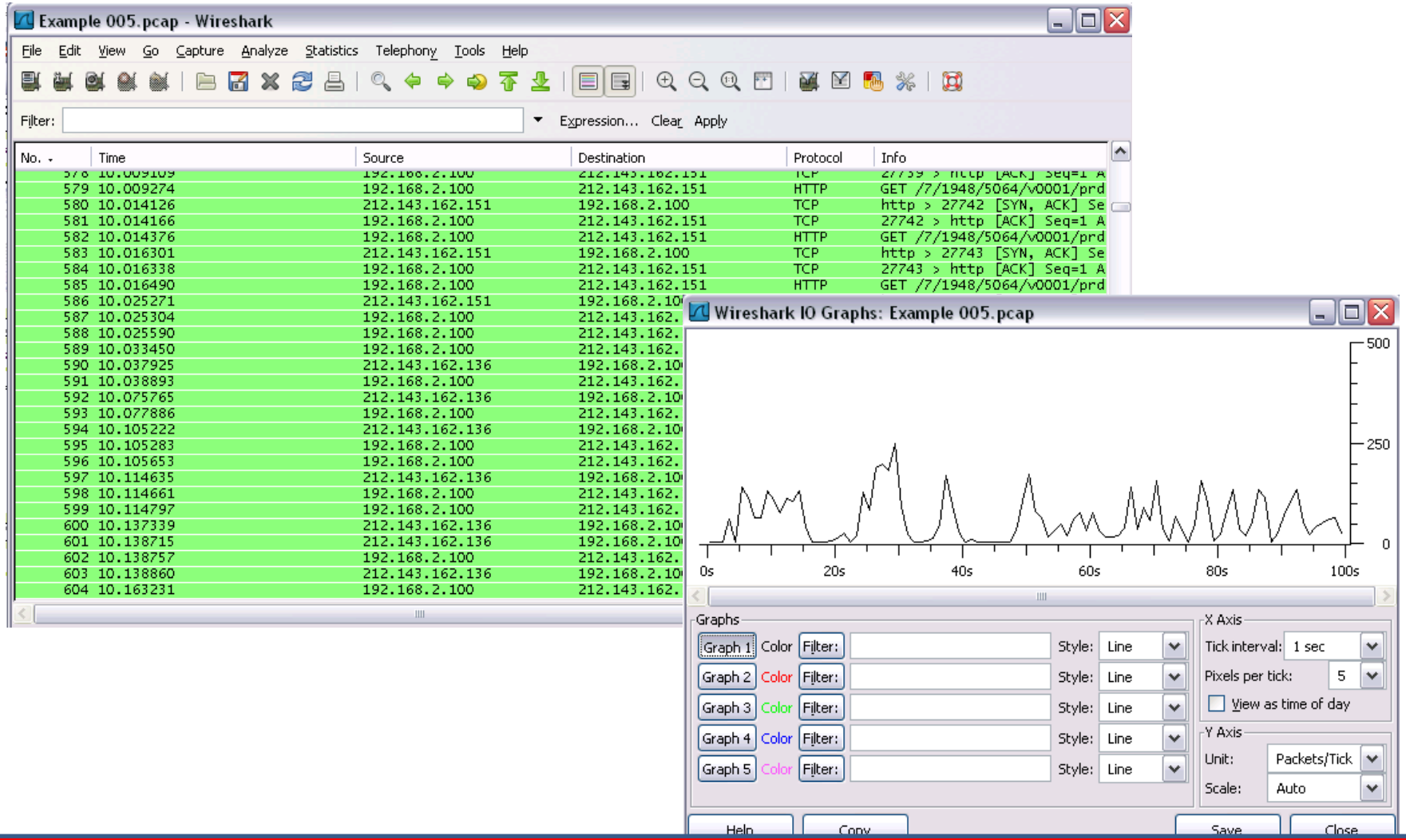
Packet Statistics (2/8)



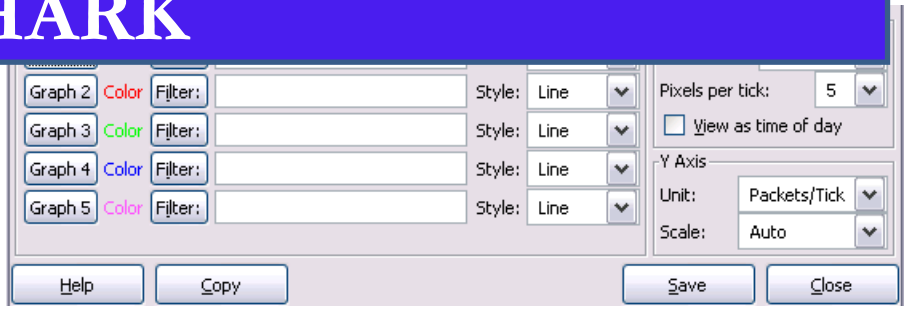
Address A	Address B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration
192.168.2.100	255.255.255.255	1	142	1	142	0	0	96.384838000	0.0000
192.168.2.101	255.255.255.255	2	684	2	684	0	0	47.852757000	3.0721
192.168.2.1	255.255.255.255	2	1180	2	1180	0	0	47.857905000	3.0722
192.168.2.100	212.143.162.144	10	2194	6	815	4	1379	87.473054000	65.0878
192.168.2.100	212.179.31.90	10	1342	6	971	4	371	91.655266000	60.9113
62.90.102.31	192.168.2.100	10	1373	4	441	6	932	91.660203000	60.9174
192.168.2.100	212.150.22.226	10	1327	6	956	4	371	91.732692000	60.8200
192.168.2.100	212.150.236.220	10	1470	6	981	4	489	91.742363000	60.8122
192.168.2.100	212.179.58.84	10	1725	6	954	4	771	92.287214000	2.4984
82.80.238.109	192.168.2.100	11	1282	5	440	6	842	91.646648000	60.9214
10.12.44.2	192.168.2.100	12	888	6	444	6	444	31.421091000	164.384
10.10.10.2	192.168.2.100	12	888	6	444	6	444	31.531676000	164.804
10.12.20.2	192.168.2.100	12	888	6	444	6	444	5.250457000	164.523
10.31.68.1	192.168.2.100	12	888	6	444	6	444	5.578447000	164.631
10.100.102.2	192.168.2.100	12	888	6	444	6	444	17.858674000	164.363

	A	B	C	D	E	F	G	H	I	J	K	L
1	Address A	Address B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	Rel Start	Duration	bps A->B	bps A<-B
2	10.10.10.1	10.159.3.103	2	124	0	0	2	124	0	42.0039	N/A	23.62
3	1.1.1.1	10.159.3.103	2	120	0	0	2	120	24.49414	1.1885	N/A	807.76

Packet Statistics (3/8)

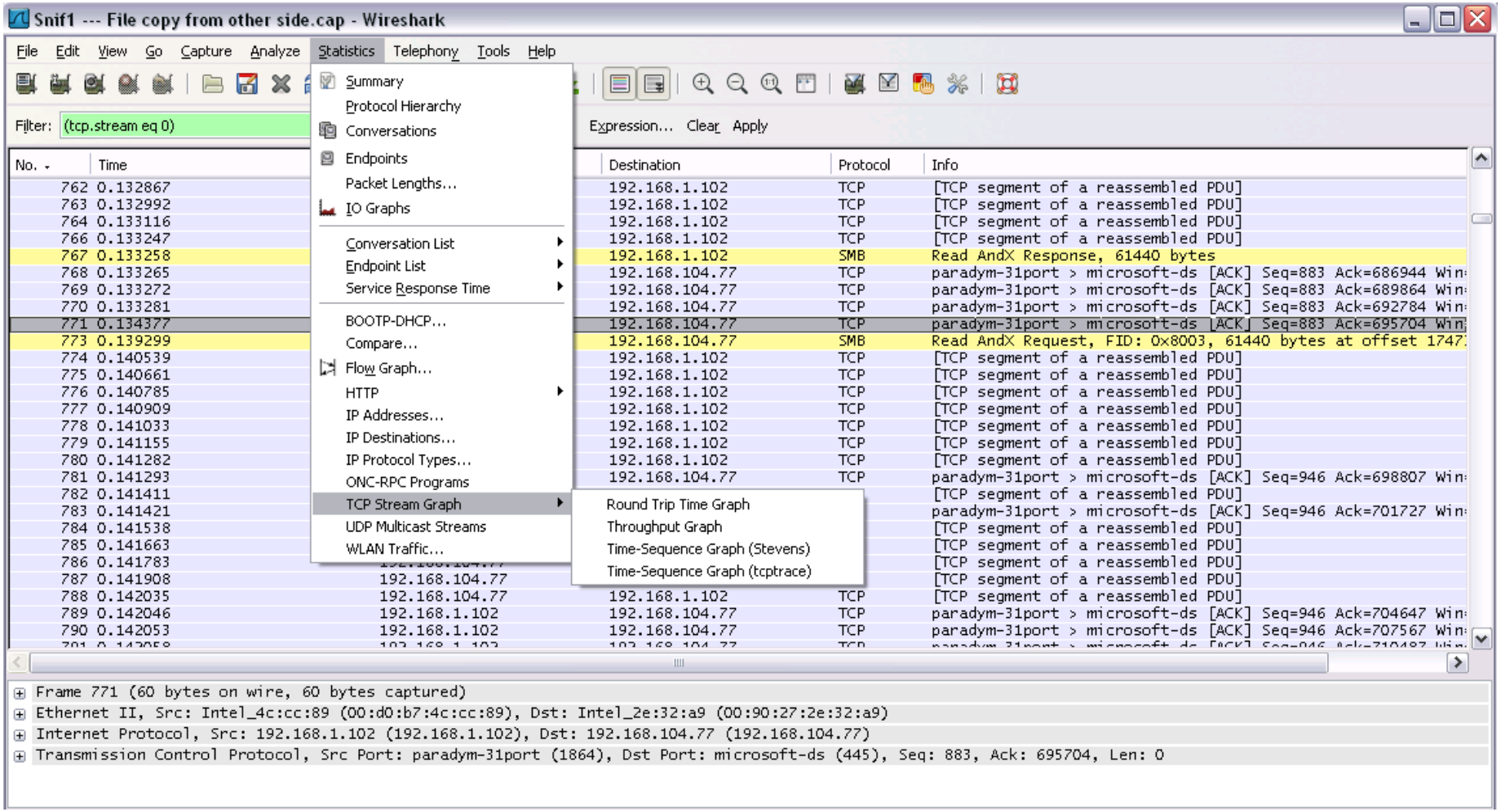


Packet Stati

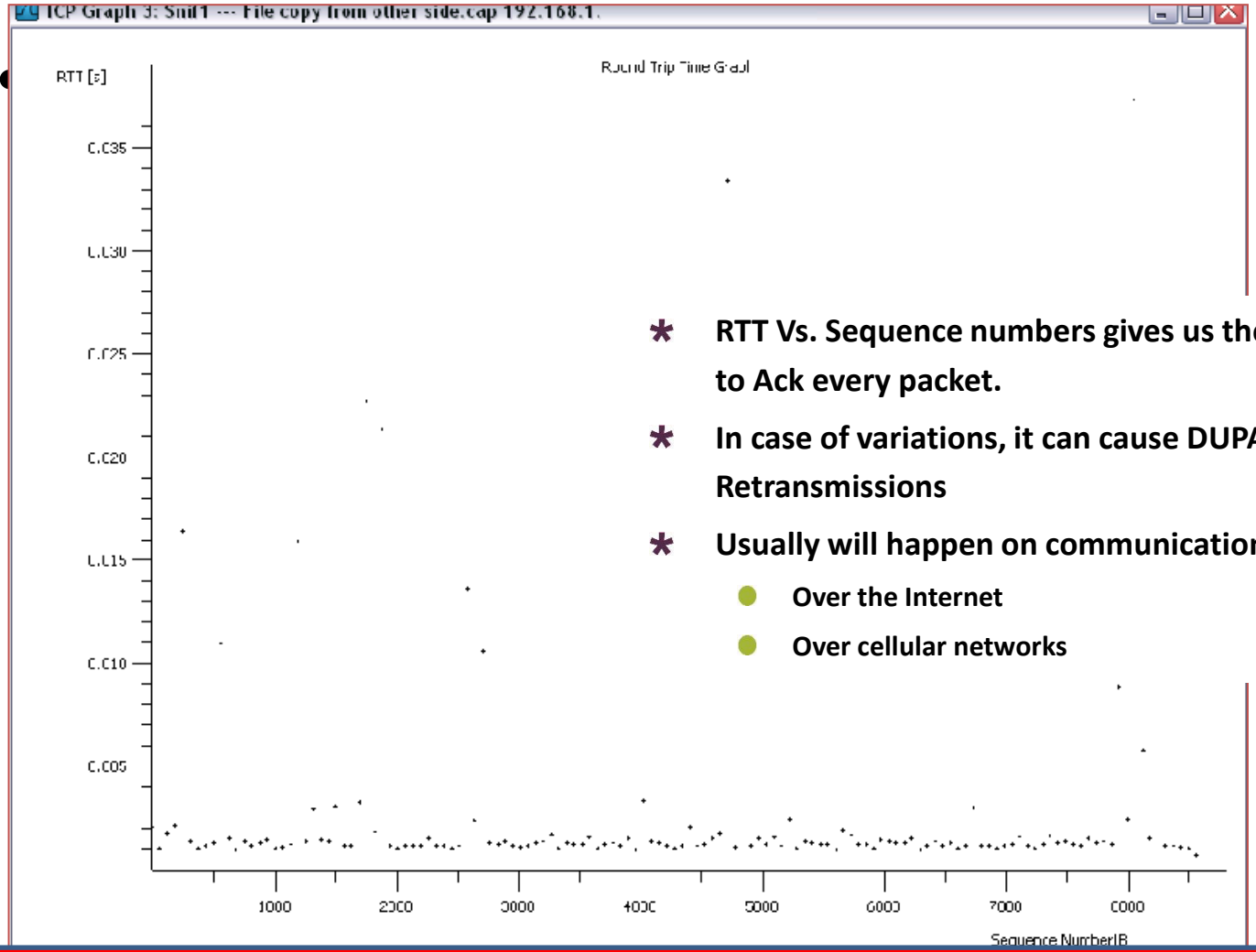


- Configurable Options
  - I/O Graphs
    - **Graph 1-5**: enable the specific graph 1-5 (graph 1 by default)
    - **Filter**: a display filter for this graph (only the packets that pass this filter will be taken into account for this graph)
    - **Style**: the style of the graph (Line/Impulse/FBar/Dot)
  - X Axis
    - **Tick interval**: an interval in x direction lasts (10/1 minutes or 10/1/0.1/0.01/0.001 seconds)
    - **Pixels per tick**: use 10/5/2/1 pixels per tick interval
    - **View as time of day**: option to view x direction labels as time of day instead of seconds or minutes since beginning of capture
  - Y Axis
    - **Unit**: the unit for the y direction (Packets/Tick, Bytes/Tick, Bits/Tick, Advanced...)
    - **Scale**: the scale for the y unit (Logarithmic,Auto,10,20,50,100,200,...)

Packet Statistics (5/8)

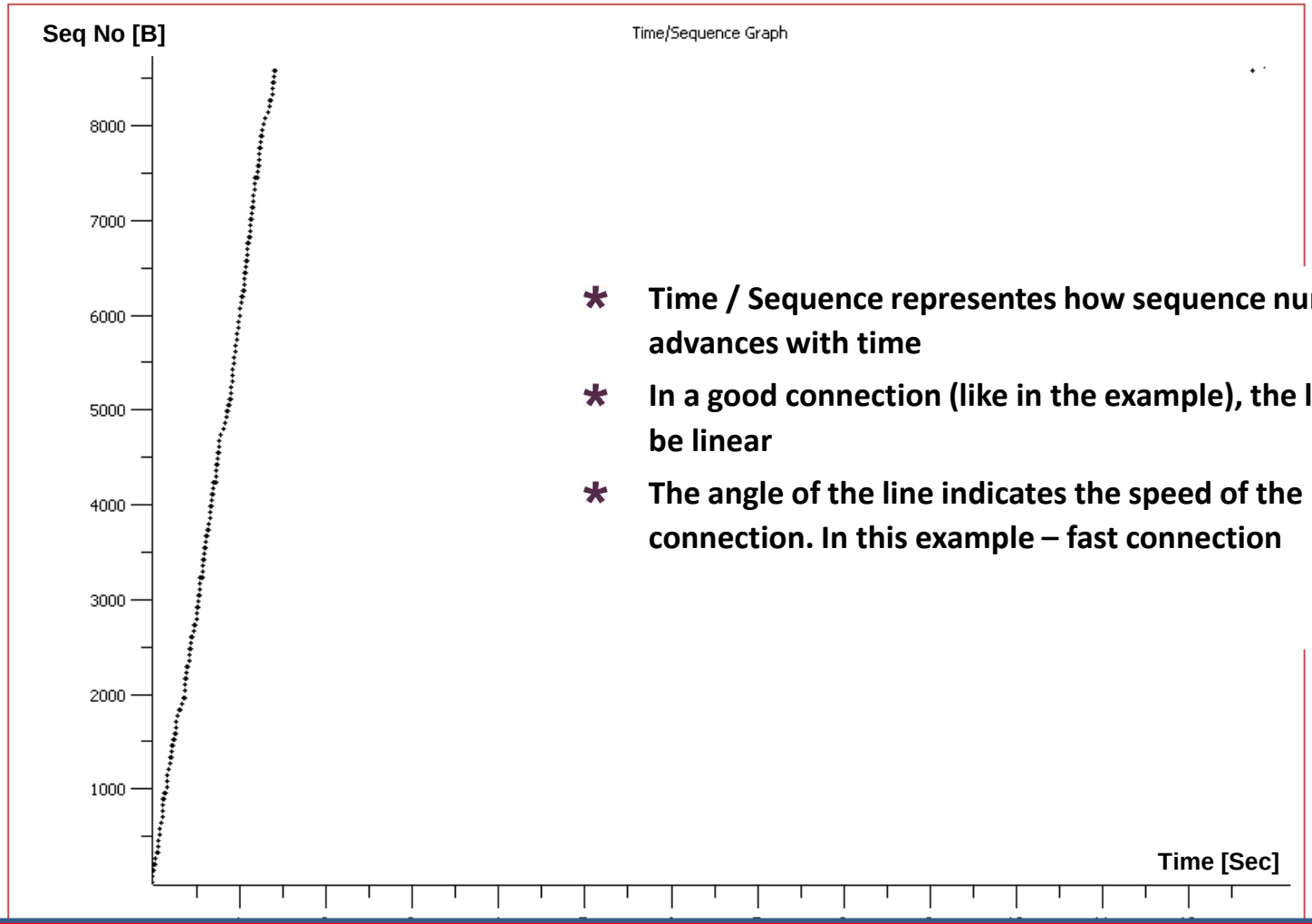


Packet Statistics (6/8)



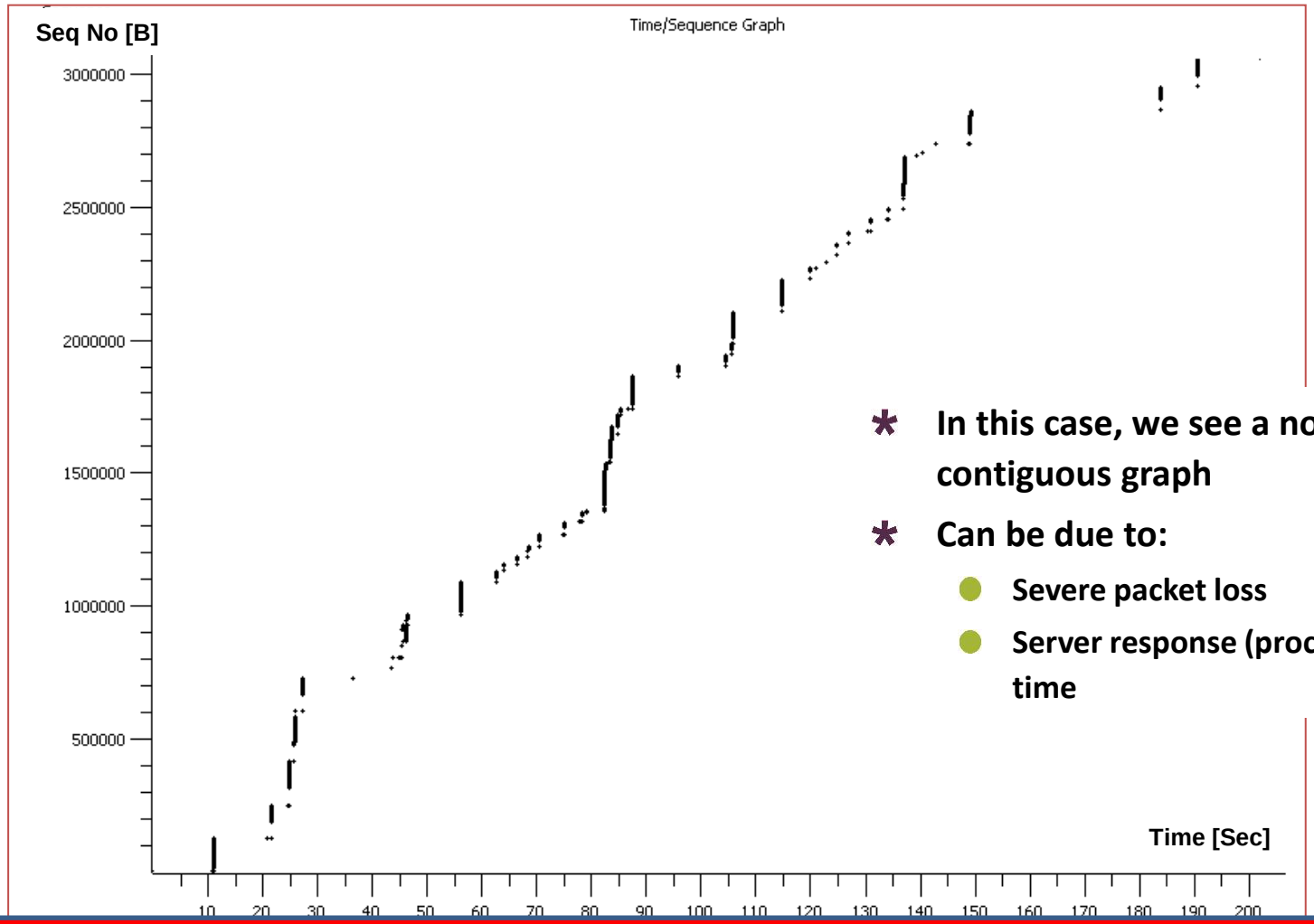
- **RTT Vs. Sequence numbers gives us the time that take to Ack every packet.**
- **In case of variations, it can cause DUPACKs and even Retransmissions**
- **Usually will happen on communications lines:**
  - **Over the Internet**
  - **Over cellular networks**

Packet Statistics (7/8)



- **Time / Sequence representes how sequence numbers advances with time**
- **In a good connection (like in the example), the line will be linear**
- **The angle of the line indicates the speed of the connection. In this example – fast connection**

Packet Statistics (8/8)

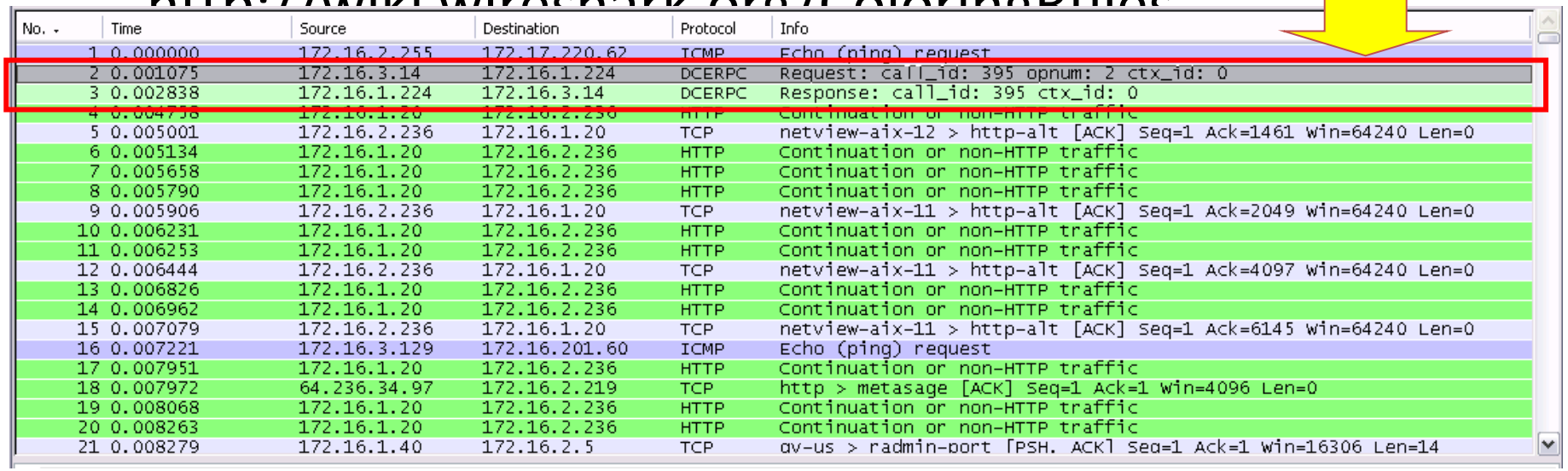


- In this case, we see a non-contiguous graph
- Can be due to:
  - Severe packet loss
  - Server response (processing) time

Colorizing Specific Packets (1/4)

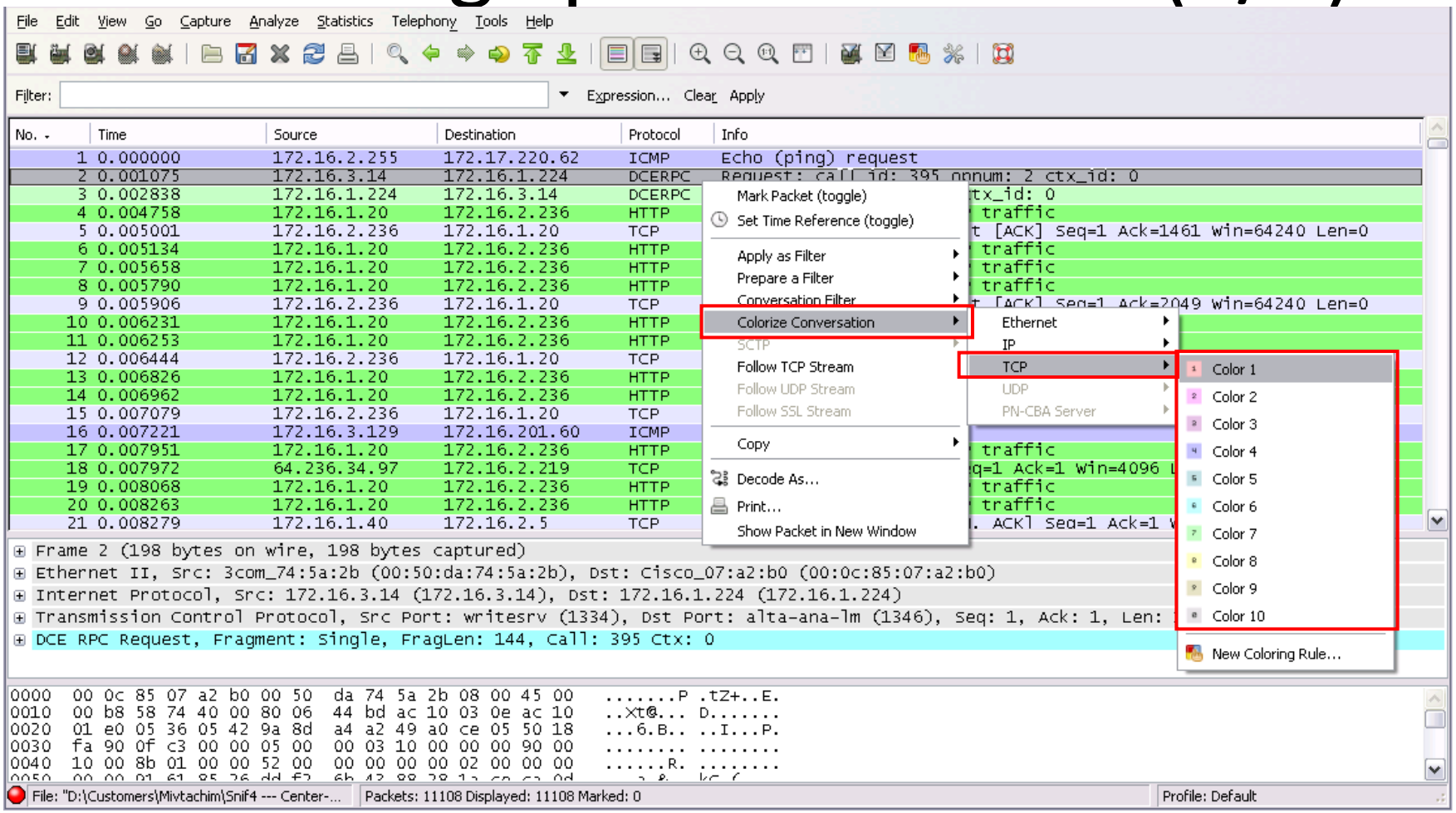
- Packet Colorization
  - Colorize packets according to a filter
  - Allow to emphasize the packets interested in
  - A lot of Coloring Rule examples at the Wireshark Wiki Coloring Rules page at http://wiki.wireshark.org/ColoringRules

We want to watch a specific protocol through out the capture file

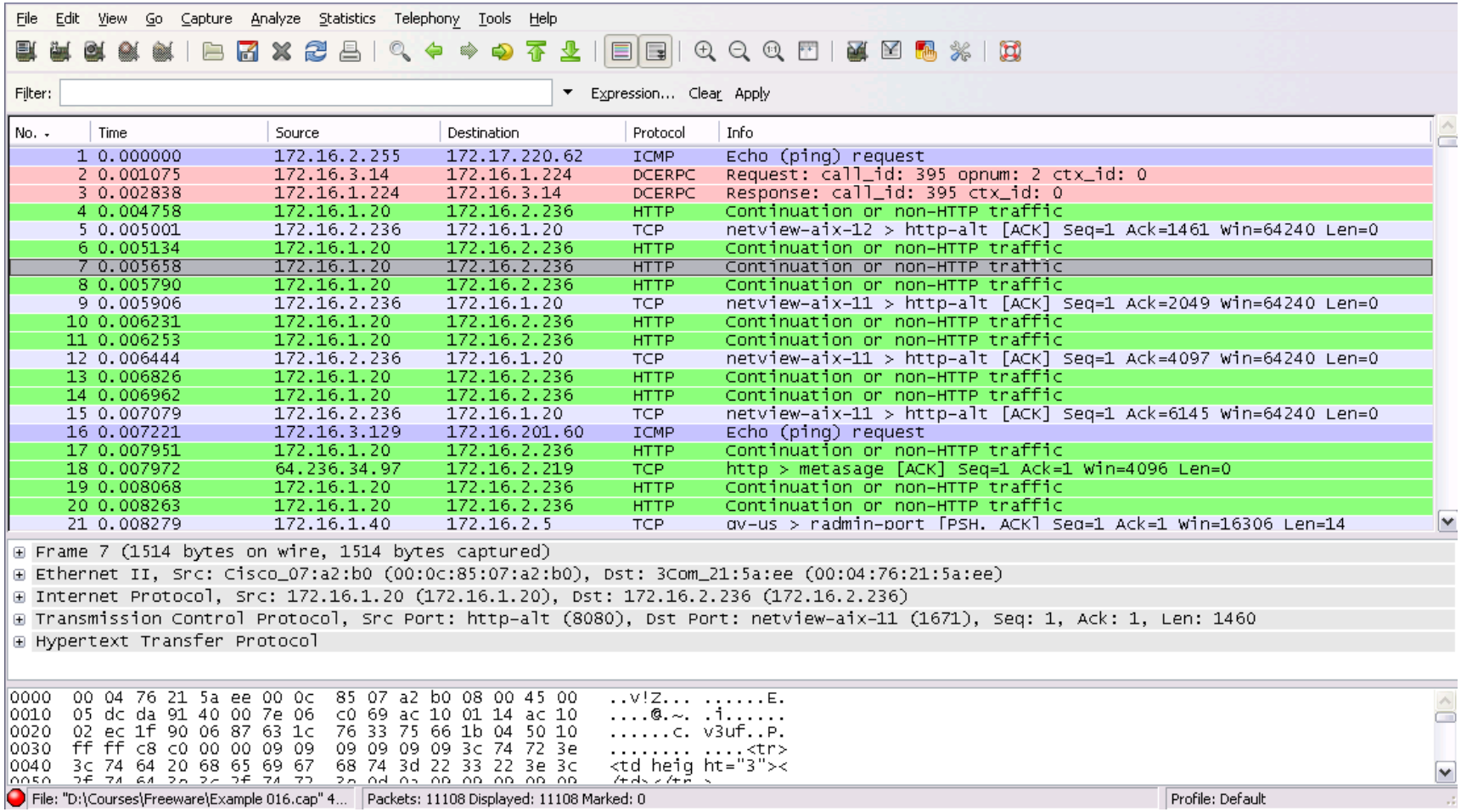


No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.2.255	172.17.220.62	ICMP	Echo (ping) request
2	0.001075	172.16.3.14	172.16.1.224	DCERPC	Request: call_id: 395 opnum: 2 ctx_id: 0
3	0.002838	172.16.1.224	172.16.3.14	DCERPC	Response: call_id: 395 ctx_id: 0
4	0.004758	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
5	0.005001	172.16.2.236	172.16.1.20	TCP	netview-aix-12 > http-alt [ACK] Seq=1 Ack=1461 Win=64240 Len=0
6	0.005134	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
7	0.005658	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
8	0.005790	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
9	0.005906	172.16.2.236	172.16.1.20	TCP	netview-aix-11 > http-alt [ACK] Seq=1 Ack=2049 Win=64240 Len=0
10	0.006231	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
11	0.006253	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
12	0.006444	172.16.2.236	172.16.1.20	TCP	netview-aix-11 > http-alt [ACK] Seq=1 Ack=4097 Win=64240 Len=0
13	0.006826	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
14	0.006962	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
15	0.007079	172.16.2.236	172.16.1.20	TCP	netview-aix-11 > http-alt [ACK] Seq=1 Ack=6145 Win=64240 Len=0
16	0.007221	172.16.3.129	172.16.201.60	ICMP	Echo (ping) request
17	0.007951	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
18	0.007972	64.236.34.97	172.16.2.219	TCP	http > metasage [ACK] Seq=1 Ack=1 Win=4096 Len=0
19	0.008068	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
20	0.008263	172.16.1.20	172.16.2.236	HTTP	Continuation or non-HTTP traffic
21	0.008279	172.16.1.40	172.16.2.5	TCP	av-us > radmin-port [PSH, ACK] Seq=1 Ack=1 Win=16306 Len=14

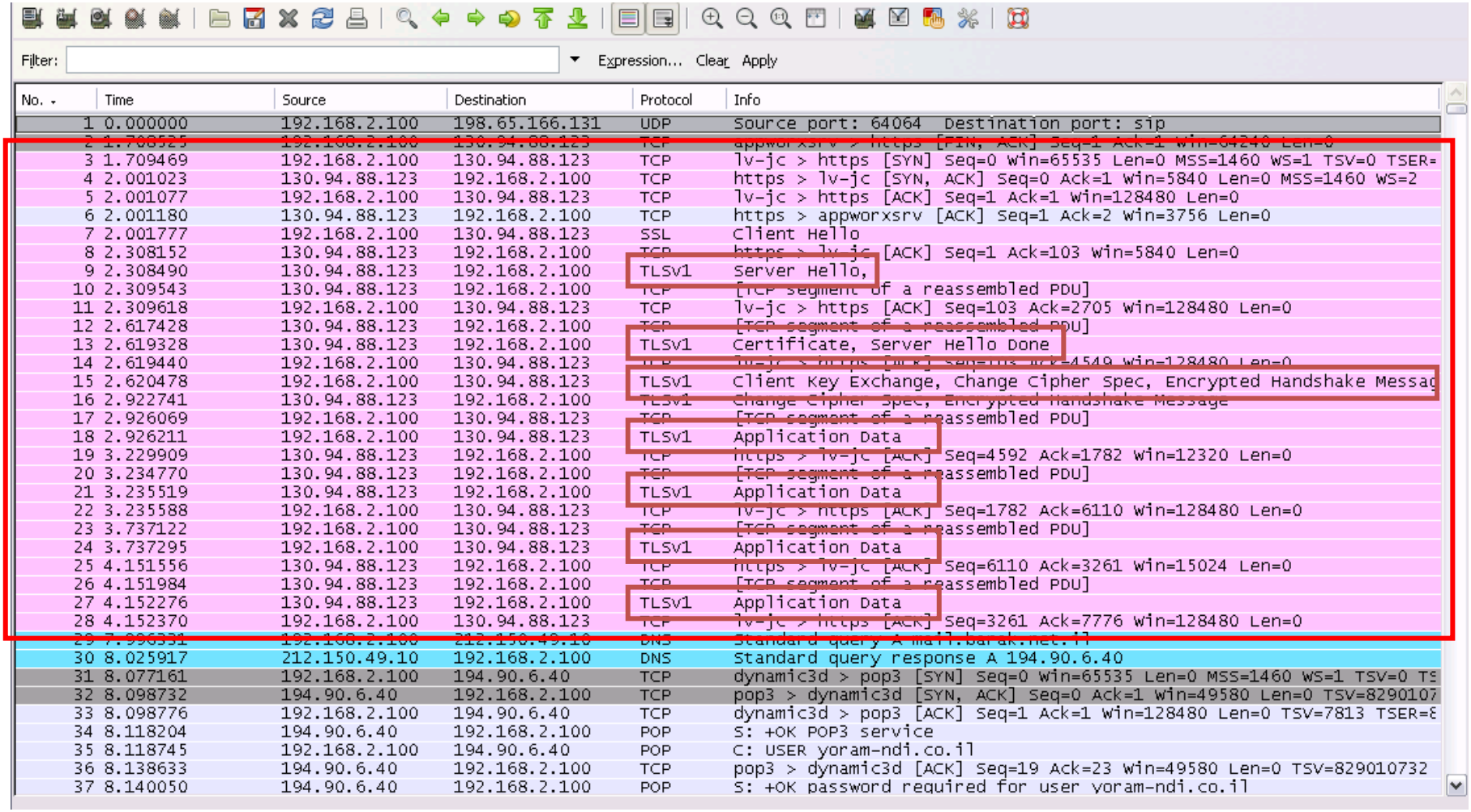
Colorizing Specific Packets (2/4)



Colorizing Specific Packets (3/4)



Colorizing Specific Packets (4/4)



References

- Wireshark Website
  - http://www.wireshark.org
- Wireshark Documentation
  - http://www.wireshark.org/docs/
- Wireshark Wiki
  - http://wiki.wireshark.org
- Network analysis Using Wireshark Cookbook
  - http://www.amazon.com/Network-Analysis-Using-Wireshark-Cookbook/dp/1849517649

WIRESHARK

Q&A