Network Monitoring and Wireshark

CSE 322

Acknowledgement

- https://studylib.net/doc/17803129/cap6135--malware-and-software-vulnerability-analysis-netw...
- https://slideplayer.com/slide/5692294/
- http://ilta.ebiz.uapps.net/ProductFiles/productfiles/672/**wireshark.ppt**

- UC Berkley course "EE 122: Intro to Communication Networks"
  - http://www.eecs.berkeley.edu/~jortiz/courses/ee122/presentations/Wireshark.ppt
- Other resources:
  - http://openmaniak.com/wireshark_filters.php

Motivation for Network Monitoring

- Essential for Network Management
  - Router and Firewall policy
  - Detecting abnormal/error in networking
  - Access control
- Security Management
  - Detecting abnormal traffic
  - Traffic log for future forensic analysis

Tools Overview

- Tcpdump
  - Unix-based command-line tool used to intercept packets
    - Including filtering just the packets of interest
  - Reads "live traffic" from interface specified using **`-i`** option …
  - … or from a previously recorded *trace file* specified using **`-r`** option
    - You create these when capturing live traffic using **`-w`** option
- Tshark
  - Tcpdump-like capture program
  - Very similar behavior and flags to tcpdump
- Wireshark
  - GUI for displaying tcpdump/tshark packet traces

tcpdump Example

- Ran tcpdump on a Unix machine
- Example: first few lines of the output

```
01:46:28.808262 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-
    7.dsl.pltn13.pacbell.net.2481: . 2513546054:2513547434(1380) ack
    1268355216 win 12816
01:46:28.808271 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-
    7.dsl.pltn13.pacbell.net.2481: P 1380:2128(748) ack 1 win 12816
01:46:28.808276 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-
    7.dsl.pltn13.pacbell.net.2481: . 2128:3508(1380) ack 1 win 12816
01:46:28.890021 IP adsl-69-228-230-7.dsl.pltn13.pacbell.net.2481 >
    danjo.CS.Berkeley.EDU.ssh: P 1:49(48) ack 1380 win 16560
```

What Does a Line Convey?

Timestamp

This is a TCP packet

Source host name

Source port number (22)

01:46:28.808262 IP danjo.CS.Berkeley.EDU.ssh >
adsl-69-228-230-7.dsl.pltn13.pacbell.net.2481: .
2513546054:2513547434(1380) ack 1268355216 win 12816

Destination host name

Destination port number

TCP specific information

- Different output formats for different packet types

Output from tcpdump

```
manav@ubuntulinux: ~
manav@ubuntulinux:~$ sudo tcpdump
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on wlo1, link-type EN10MB (Ethernet), capture size 262144 bytes
23:12:15.734637 IP6 fe80::6c8:7ff:fe26:ceac > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s), length 28
23:12:15.737024 IP ubuntulinux.60811 > b.resolvers.Level3.net.domain: 60873+ PTR? 6.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
23:12:15.937343 IP b.resolvers.Level3.net.domain > ubuntulinux.60811: 60873 NXDomain 0/1/0 (154)
23:12:15.939727 IP ubuntulinux.44132 > b.resolvers.Level3.net.domain: 7027+ PTR? c.a.e.c.6.2.e.f.f.f.7.0.8.c.6.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.e.f.ip6.arpa. (90)
23:12:16.142628 IP b.resolvers.Level3.net.domain > ubuntulinux.44132: 7027 NXDomain* 0/1/0 (149)
23:12:16.144477 IP ubuntulinux.54078 > b.resolvers.Level3.net.domain: 44074+ PTR? 2.2.2.4.in-addr.arpa. (38)
23:12:16.182985 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QU)? Android.local. ANY (QU)? Android.local. (81)
23:12:16.346940 IP b.resolvers.Level3.net.domain > ubuntulinux.54078: 44074 1/0/0 PTR b.resolvers.Level3.net. (74)
23:12:16.348070 IP ubuntulinux.40932 > b.resolvers.Level3.net.domain: 34612+ PTR? 102.0.168.192.in-addr.arpa. (44)
23:12:16.442544 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QM)? Android.local. ANY (QM)? Android.local. (81)
23:12:16.515363 IP b.resolvers.Level3.net.domain > ubuntulinux.40932: 34612 NXDomain* 0/1/0 (103)
23:12:16.517489 IP ubuntulinux.53519 > b.resolvers.Level3.net.domain: 3094+ PTR? b.f.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)
23:12:16.590778 IP6 fe80::6c8:7ff:fe26:ceac > ff02::16: HBH ICMP6, multicast listener report v2, 1 group record(s), length 28
23:12:16.680420 IP b.resolvers.Level3.net.domain > ubuntulinux.53519: 3094 NXDomain 0/1/0 (154)
23:12:16.683273 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0 [2q] [2n] ANY (QM)? Android.local. ANY (QM)? Android.local. (81)
23:12:16.961783 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0*- [0q] 4/0/3 (Cache flush) PTR Android.local., (Cache flush) PTR Android.local., (Cache flush) A 192.168.0.101, (Cache flush) AAAA fe80::6c8:7f
f:fe26:ceac (247)
23:12:17.986627 IP6 fe80::6c8:7ff:fe26:ceac.mdns > ff02::fb.mdns: 0*- [0q] 4/0/3 (Cache flush) PTR Android.local., (Cache flush) PTR Android.local., (Cache flush) A 192.168.0.101, (Cache flush) AAAA fe80::6c8:7f
f:fe26:ceac (247)
23:12:18.238252 IP ubuntulinux.35076 > a23-39-122-85.deploy.static.akamaitechnologies.com.https: Flags [.], ack 89677080, win 501, options [nop,nop,TS val 2010432084 ecr 2402112534], length 0
23:12:18.239048 IP ubuntulinux.55784 > b.resolvers.Level3.net.domain: 49805+ PTR? 85.122.39.23.in-addr.arpa. (43)
23:12:18.497830 IP a23-39-122-85.deploy.static.akamaitechnologies.com.https > ubuntulinux.35076: Flags [R], seq 89677080, win 0, length 0
23:12:18.497830 IP b.resolvers.Level3.net.domain > ubuntulinux.55784: 49805 1/0/0 PTR a23-39-122-85.deploy.static.akamaitechnologies.com. (107)
23:12:19.101841 IP ubuntulinux.55857 > 239.255.255.250.1900: UDP, length 171
23:12:19.102591 IP ubuntulinux.49879 > b.resolvers.Level3.net.domain: 54682+ PTR? 250.255.255.239.in-addr.arpa. (46)
23:12:19.317004 IP b.resolvers.Level3.net.domain > ubuntulinux.49879: 54682 NXDomain 0/1/0 (103)
23:12:19.636296 IP ubuntulinux.bootpc > _gateway.bootps: BOOTP/DHCP, Request from 84:fd:d1:e5:20:5e (oui Unknown), length 289
23:12:19.637071 IP ubuntulinux.51783 > b.resolvers.Level3.net.domain: 29099+ PTR? 1.0.168.192.in-addr.arpa. (42)
23:12:19.831442 IP _gateway.bootps > 255.255.255.255.bootpc: BOOTP/DHCP, Reply, length 295
23:12:19.931402 IP b.resolvers.Level3.net.domain > ubuntulinux.51783: 29099 NXDomain* 0/1/0 (101)
23:12:20.102446 IP ubuntulinux.55857 > 239.255.255.250.1900: UDP, length 171
^C23:12:20.239751 IP 192.168.0.3.mdns > 224.0.0.251.mdns: 16 [2q] PTR (QM)? _233637DE._sub._googlecast._tcp.local. PTR (QM)? _googlecast._tcp.local. (61)

30 packets captured
34 packets received by filter
0 packets dropped by kernel
manav@ubuntulinux:~$
```

- Link: https://www.geeksforgeeks.org/tcpdump-command-in-linux-with-examples/

Similar Output from Tshark

1190003744.940437 61.184.241.230 -> 128.32.48.169 SSH Encrypted request packet len=48

1190003744.940916 128.32.48.169 -> 61.184.241.230 SSH Encrypted response packet len=48

1190003744.955764 61.184.241.230 -> 128.32.48.169 TCP 6943 > ssh [ACK] Seq=48 Ack=48 Win=65514 Len=0 TSV=445871583 TSER=632535493

1190003745.035678 61.184.241.230 -> 128.32.48.169 SSH Encrypted request packet len=48

1190003745.036004 128.32.48.169 -> 61.184.241.230 SSH Encrypted response packet len=48

1190003745.050970 61.184.241.230 -> 128.32.48.169 TCP 6943 > ssh [ACK] Seq=96 Ack=96 Win=65514 Len=0 TSV=445871583 TSER=632535502

Demo 1 – Basic Run

- Syntax:

  tcpdump [options] [filter expression]

- To do -
  - $ sudo tcpdump –i eth0
    - Sudo command allows you to run tcpdump in root privilege
  - On your own Unix machine, you can run it using "sudo" or directly run "tcpdump" if you have root privilege
- Observe the output

Filters

- We are often not interested in all packets flowing through the network
- Use filters to capture only packets of interest to us

Demo 2

1. Capture only udp packets
   - tcpdump “udp”
2. Capture only tcp packets
   - tcpdump “tcp”

Demo 2 (contd.)

1. Capture only UDP packets with destination port 53 (DNS requests)
   - tcpdump “udp dst port 53”
2. Capture only UDP packets with source port 53 (DNS replies)
   - tcpdump “udp src port 53”
3. Capture only UDP packets with source or destination port 53 (DNS requests and replies)
   - tcpdump “udp port 53”

Demo 2 (contd.)

1. Capture only packets destined to buet.ac.bd
   - tcpdump “dst host buet.ac.bd”
2. Capture both DNS packets and TCP packets to/from buet.ac.bd
   - tcpdump “(tcp and host buet.ac.bd) or udp port 53”

How to write filters

- Refer the tcpdump/tshark man page
- Many example webpages on the Internet

Running tcpdump

- Requires superuser/administrator privileges on Unix
  - http://www.tcpdump.org/
  - You can do it on your own Unix machine
  - You can install a Linux OS in Vmware on your windows machine

- Tcpdump for Windows
  - WinDump: http://www.winpcap.org/windump/
    - Free software
    - Might have compatibility issues

So What is WireShark?

- Packet sniffer/protocol analyzer
- Open Source Network Tool
- Latest version of the ethereal tool

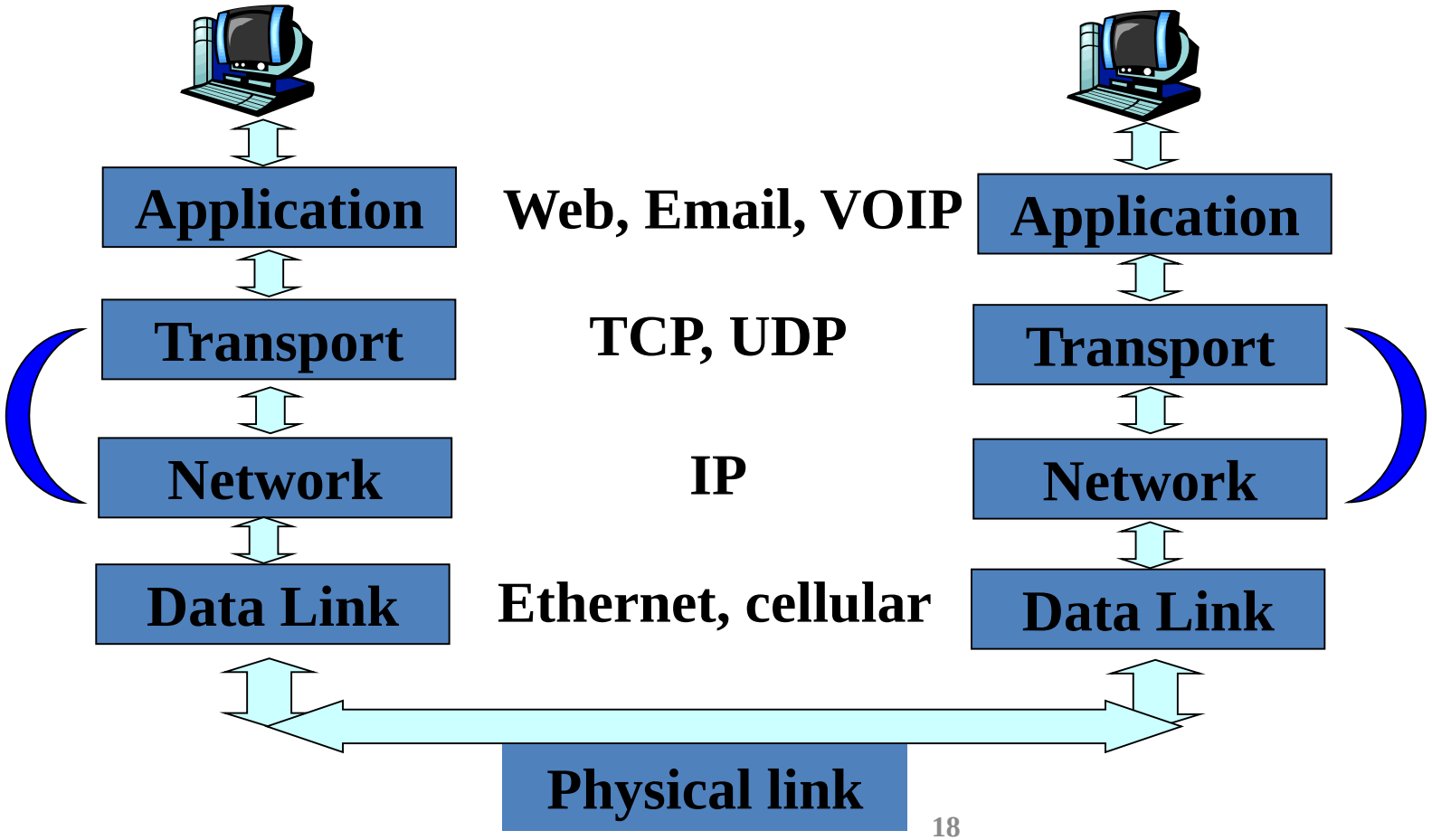
#	Wireshark	Tcpdump
1	A GUI tool to catch data packets	A CLI-based packet capturing tool
2	It does packet analysis, decode data payloads if the encryption keys are identified, and recognize data payloads from file transfers such as smtp, http, etc.	Tcpdump only provides do a simple analysis of such types of traffic, such as DNS queries
3	It has advanced network interfaces	It has conventional interfaces
4	Good for complex filters	Used for simple filters
5	It provides decoding of protocol-based packet capturing	It is less efficient in decoding compared to Wireshark

tShark
(An Unconventional Alternative)

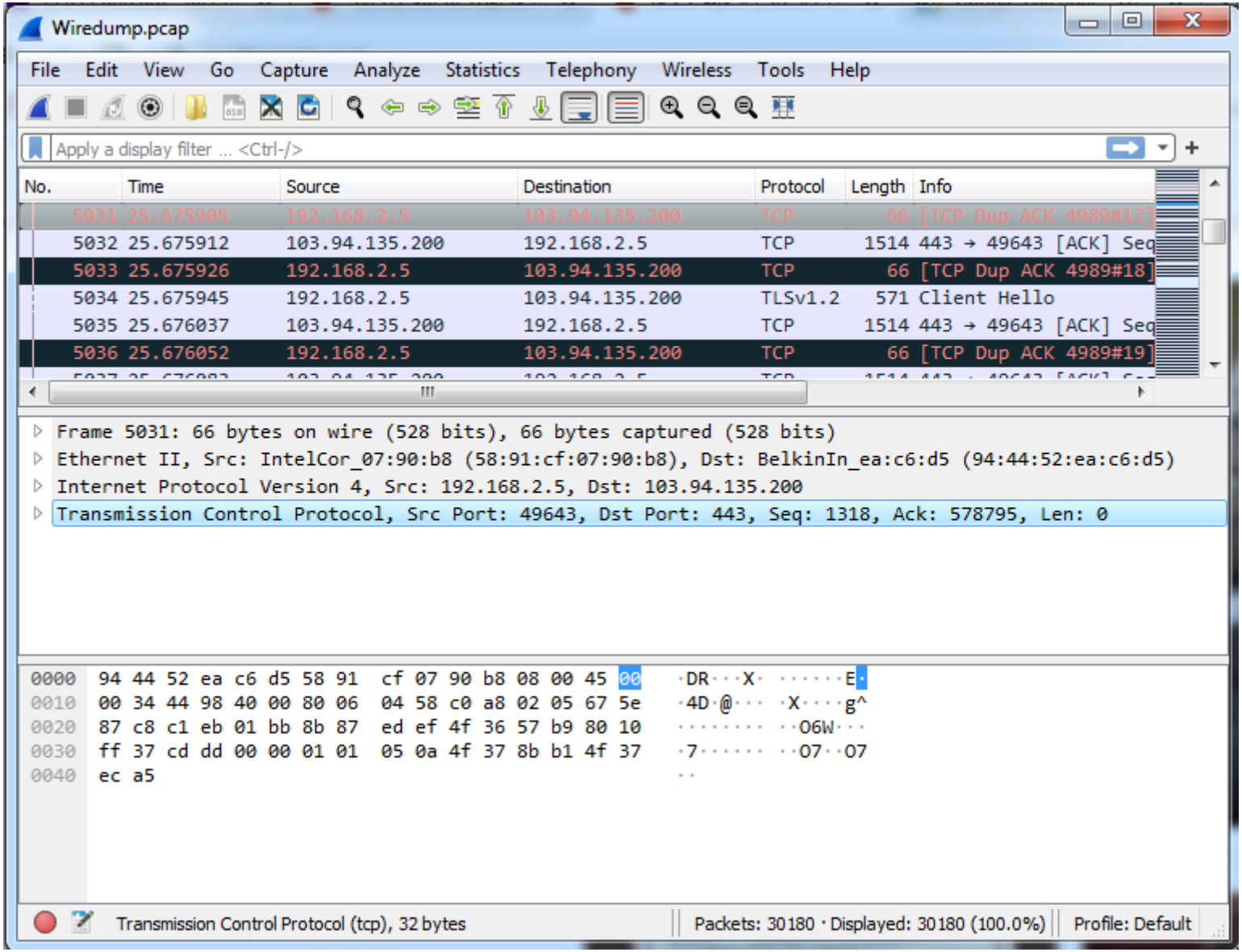
- Command-line based packet capture tool
- Equivalent to Wireshark

Scope of WireShark

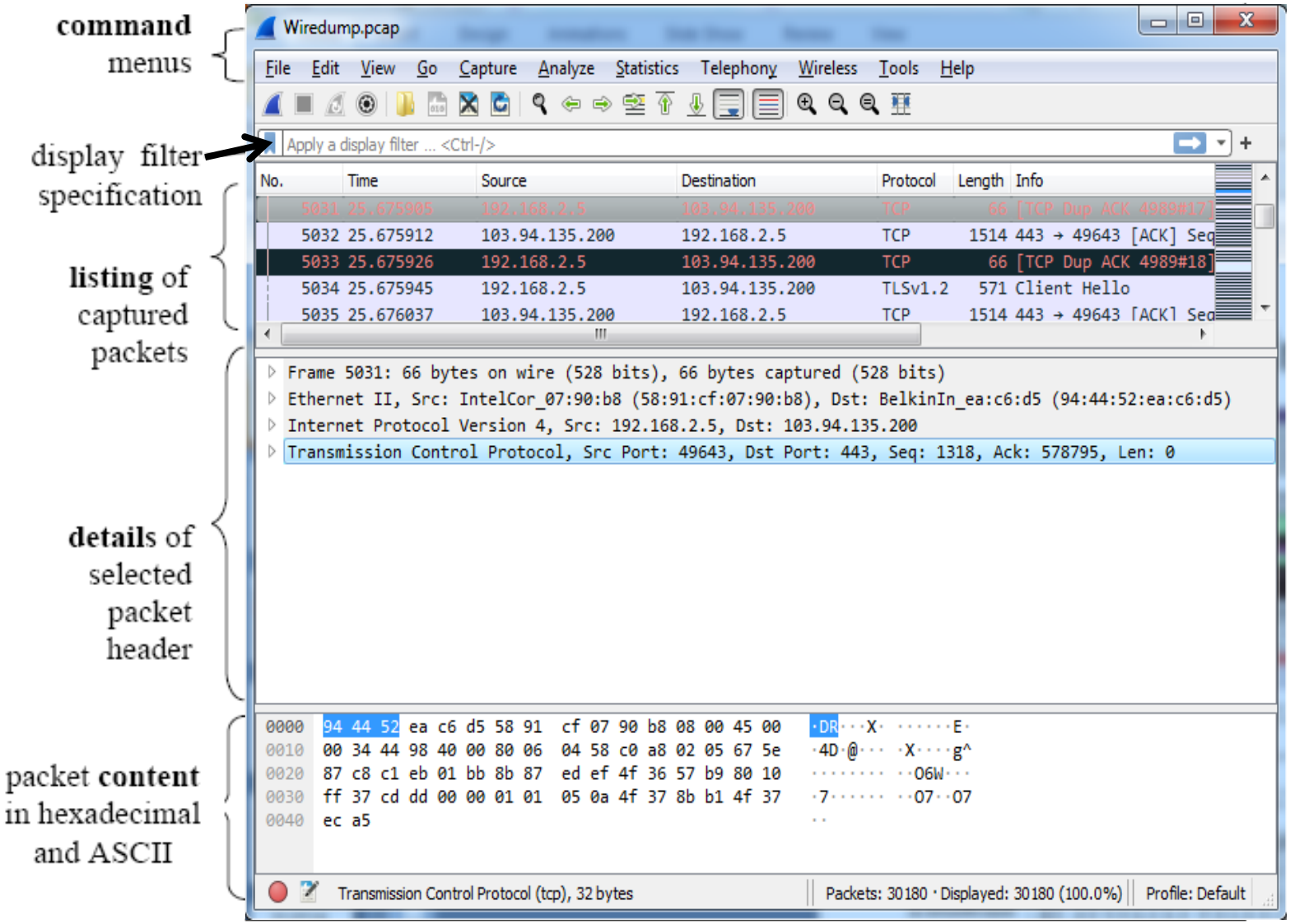
- Internet and all …



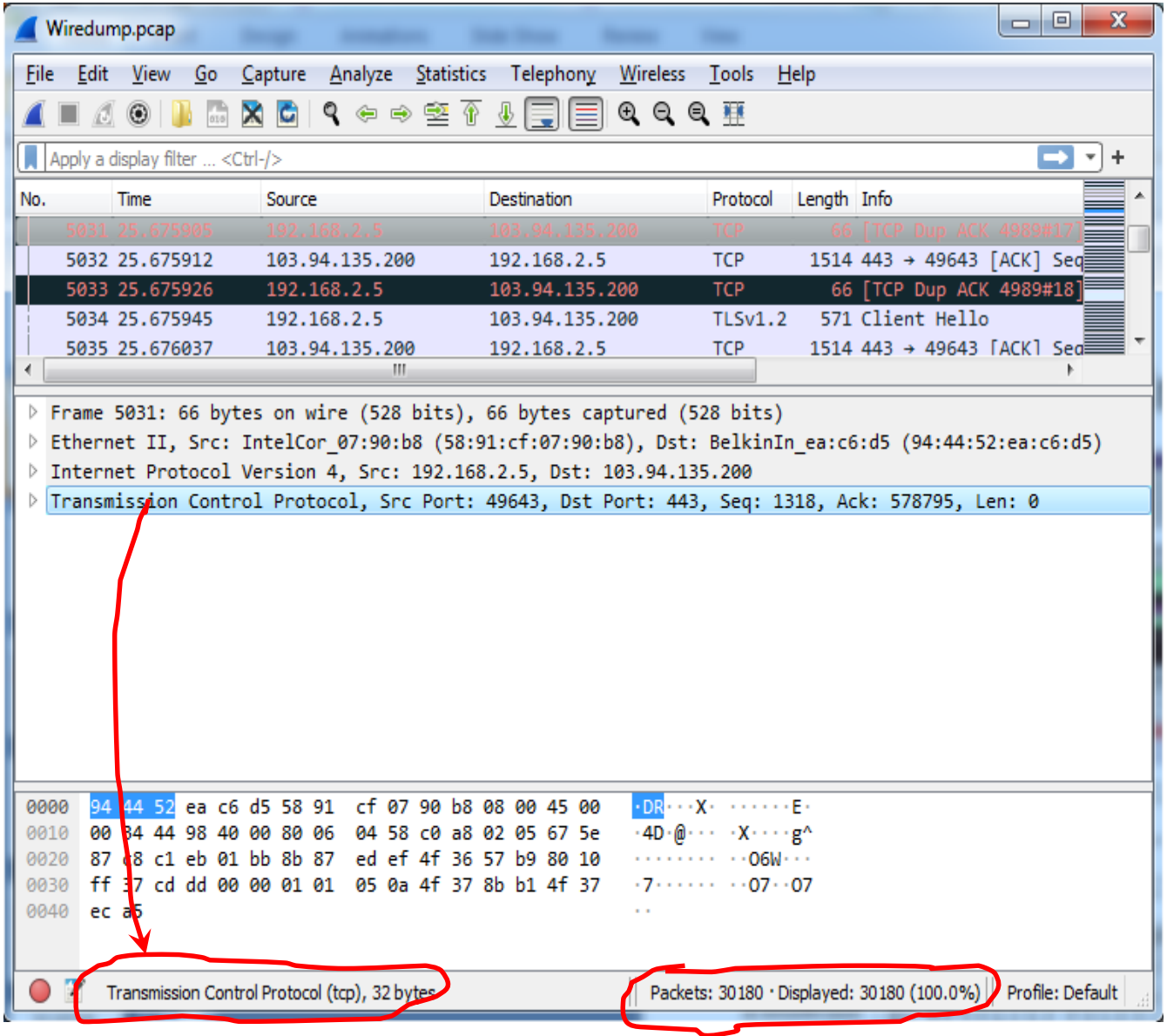
Wireshark Interface



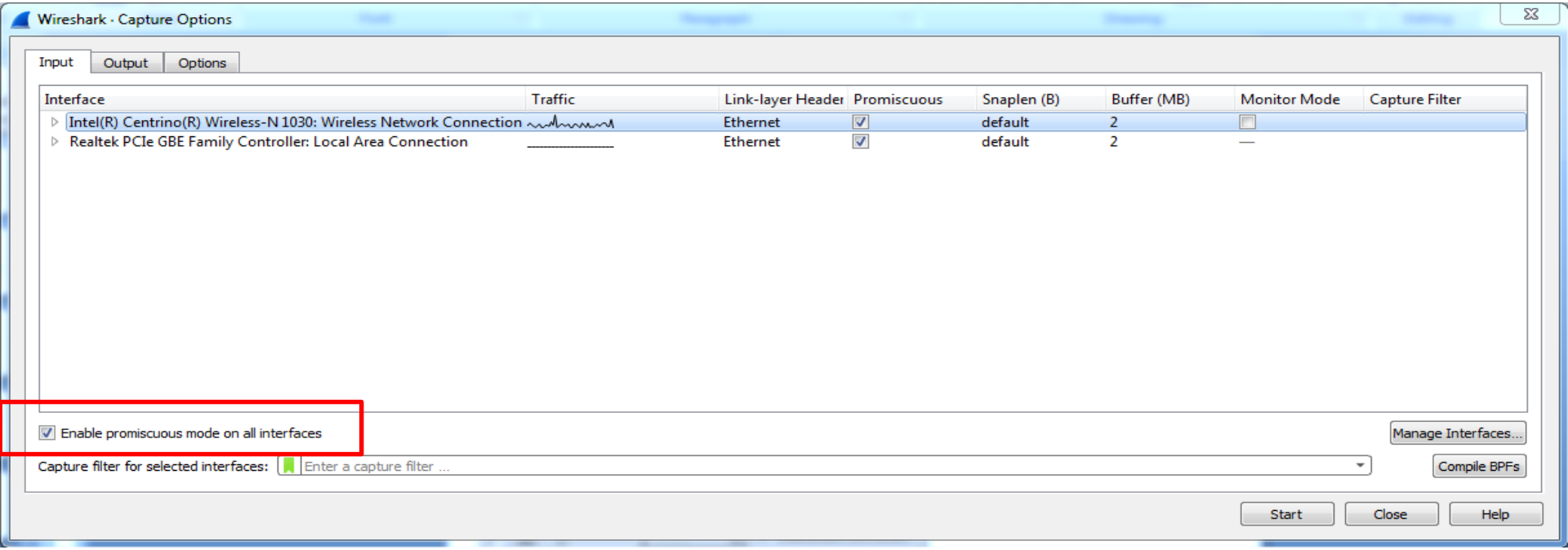
Wireshark Interface – Different Parts



Status Bar

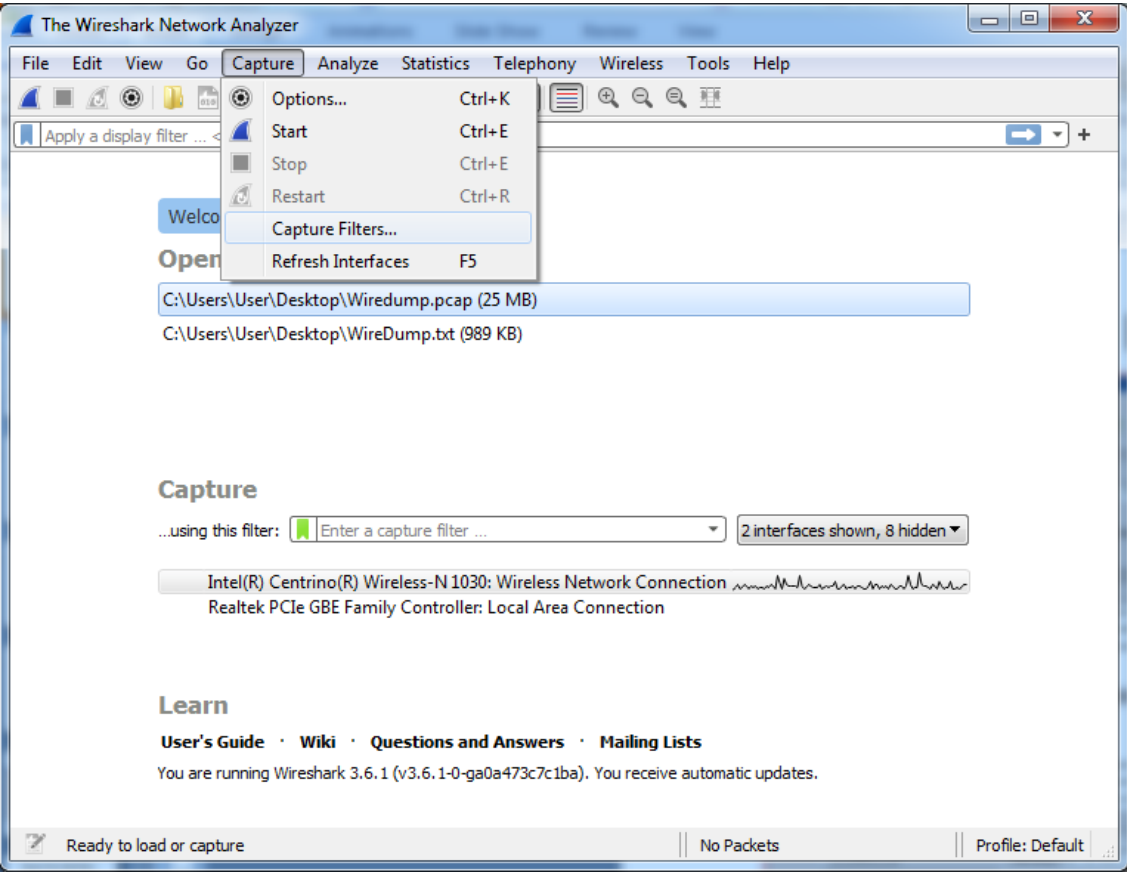

Wiredump.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
5031 25.675905 192.168.2.5 103.94.135.200 TCP 66 [TCP Dup ACK 4989#17]
5032 25.675912 103.94.135.200 192.168.2.5 TCP 1514 443 → 49643 [ACK] Seq
5033 25.675926 192.168.2.5 103.94.135.200 TCP 66 [TCP Dup ACK 4989#18]
5034 25.675945 192.168.2.5 103.94.135.200 TLSv1.2 571 Client Hello
5035 25.676037 103.94.135.200 192.168.2.5 TCP 1514 443 → 49643 [ACK] Seq
Frame 5031: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: IntelCor_07:90:b8 (58:91:cf:07:90:b8), Dst: BelkinIn_ea:c6:d5 (94:44:52:ea:c6:d5)
Internet Protocol Version 4, Src: 192.168.2.5, Dst: 103.94.135.200
Transmission Control Protocol, Src Port: 49643, Dst Port: 443, Seq: 1318, Ack: 578795, Len: 0
0000 94 44 52 ea c6 d5 58 91 cf 07 90 b8 08 00 45 00
0010 00 34 44 98 40 00 80 06 04 58 c0 a8 02 05 67 5e
0020 87 c8 c1 eb 01 bb 8b 87 ed ef 4f 36 57 b9 80 10
0030 ff 37 cd dd 00 00 01 01 05 0a 4f 37 8b b1 4f 37
0040 ec a5
Transmission Control Protocol (tcp), 32 bytes
Packets: 30180 · Displayed: 30180 (100.0%)
Profile: Default

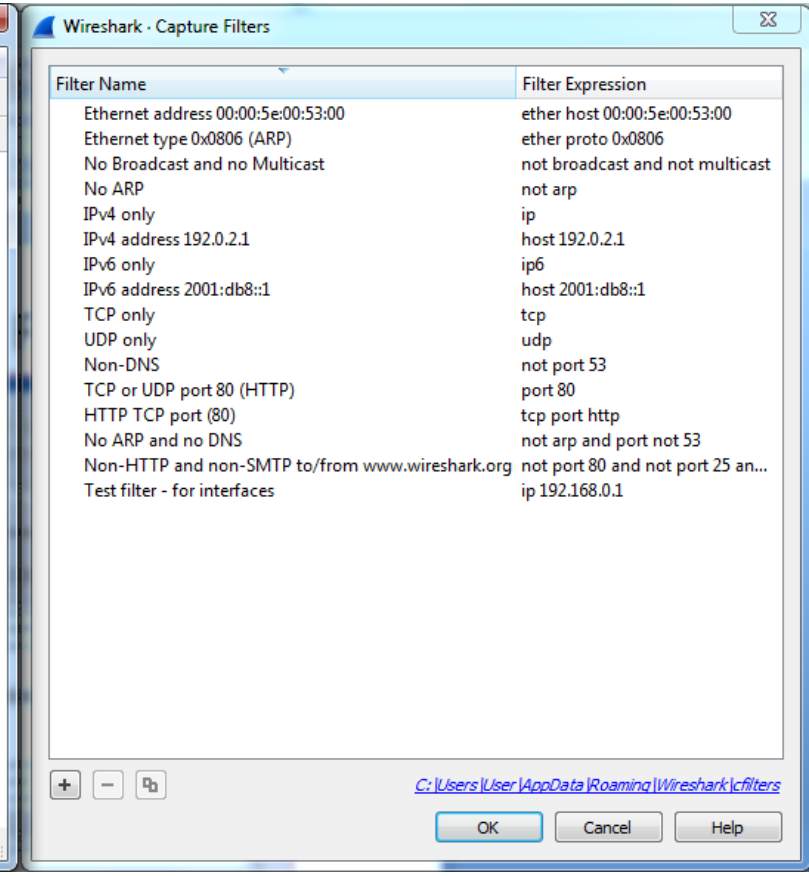
Capture Options



- Promiscuous mode is used to capture all traffic
- Sometime this does not work
  - Driver does not support
  - You are on a switched LAN

Capture Filter





Capture Filter Examples

host 10.1.11.24

host 192.168.0.1 and host 10.1.11.1

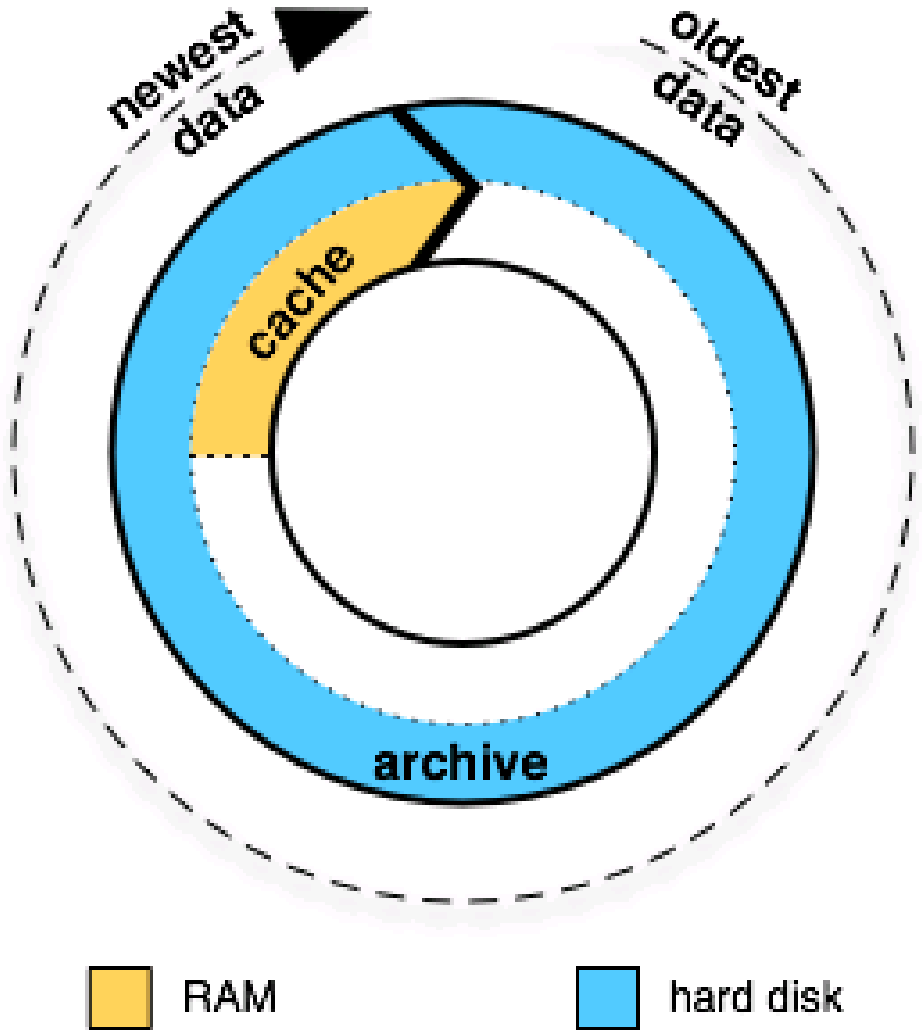
tcp port http

ip

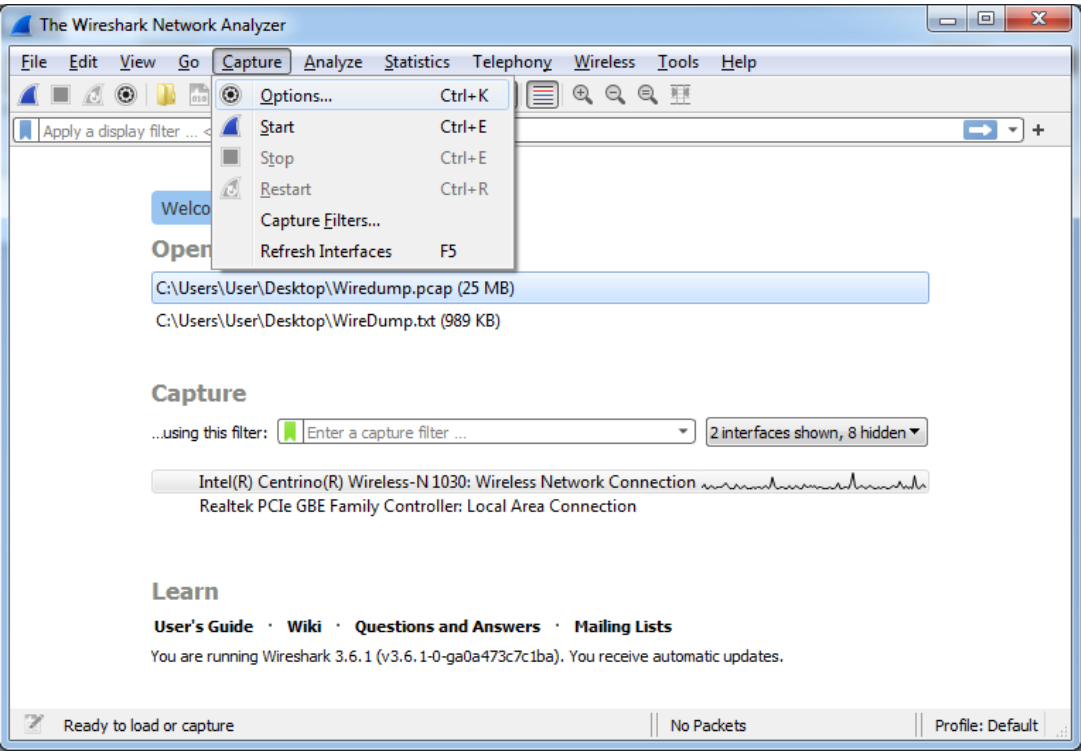
not broadcast not multicast

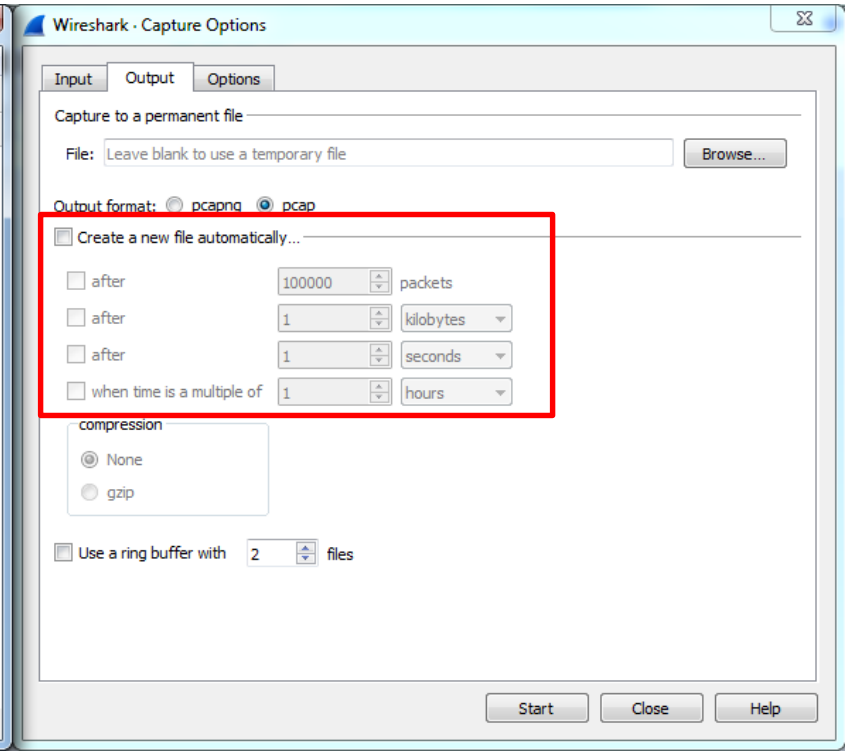
ether host 00:04:13:00:09:a3

Capture Buffer Usage



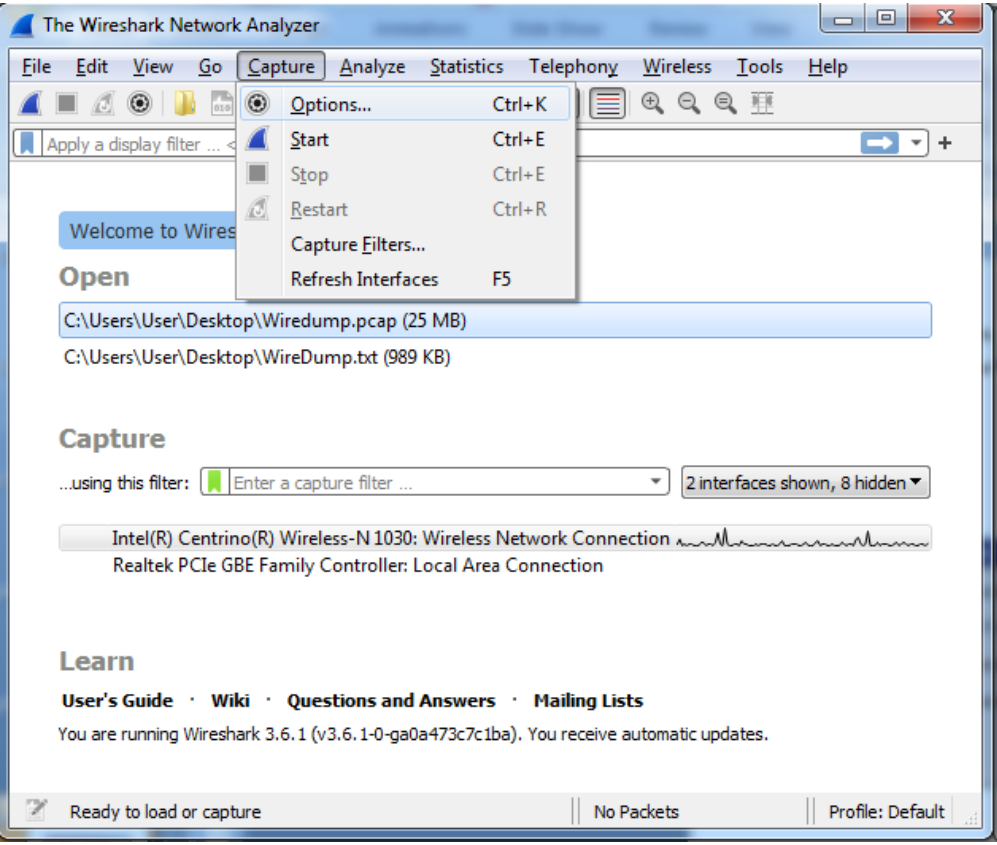
Capture Options

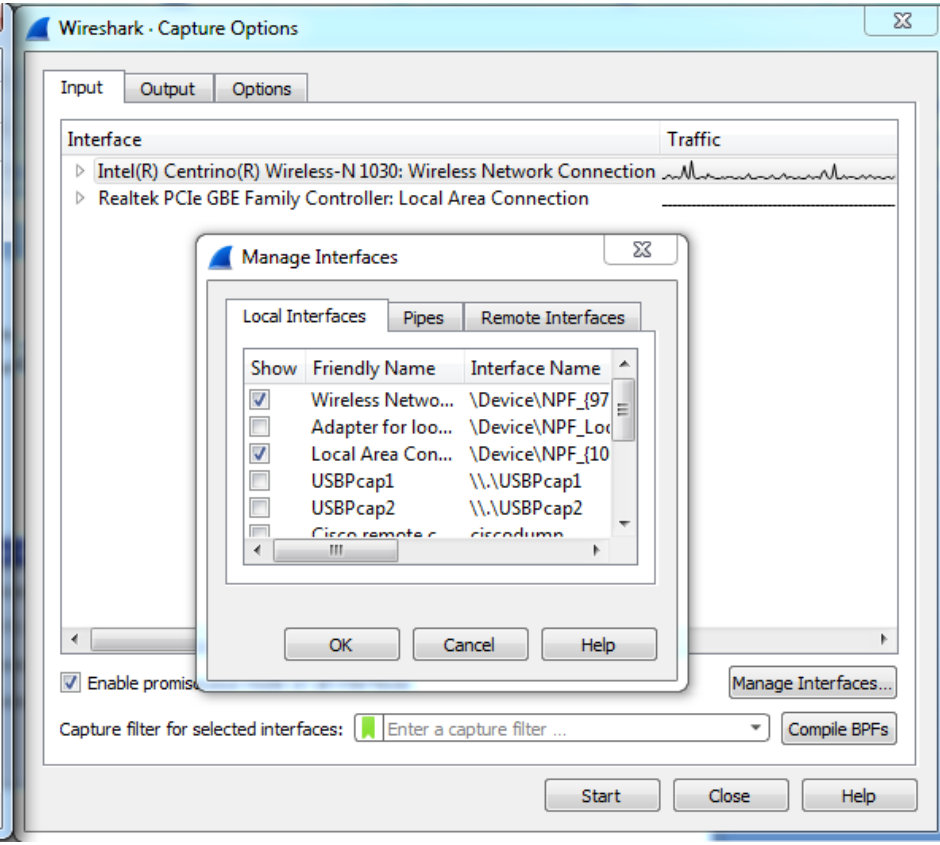




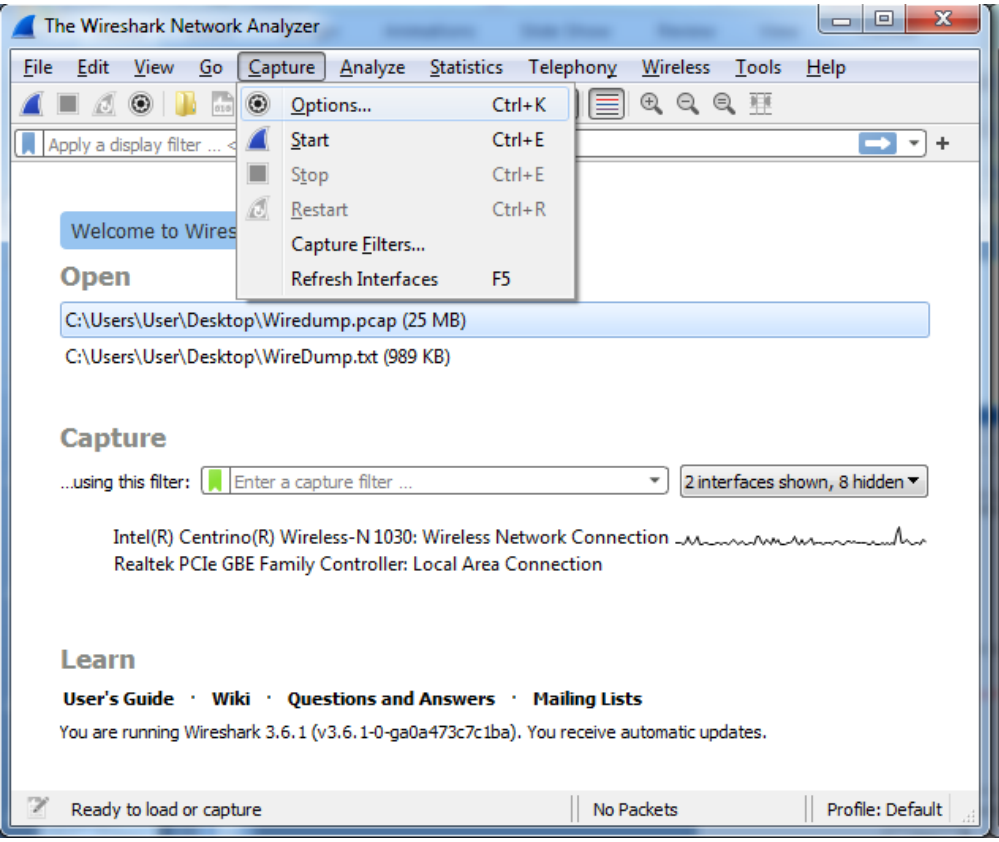
- Ring buffer
  - Addresses a common issue many analysts encounter when capturing packets - huge traces
  - When you use a Ring Buffer, you can define how many files you want to capture and various parameters that affects the file size (# of packets, bytes, and time)
- Link: https://www.networkcomputing.com/networking/working-ring-buffer-wireshark-files

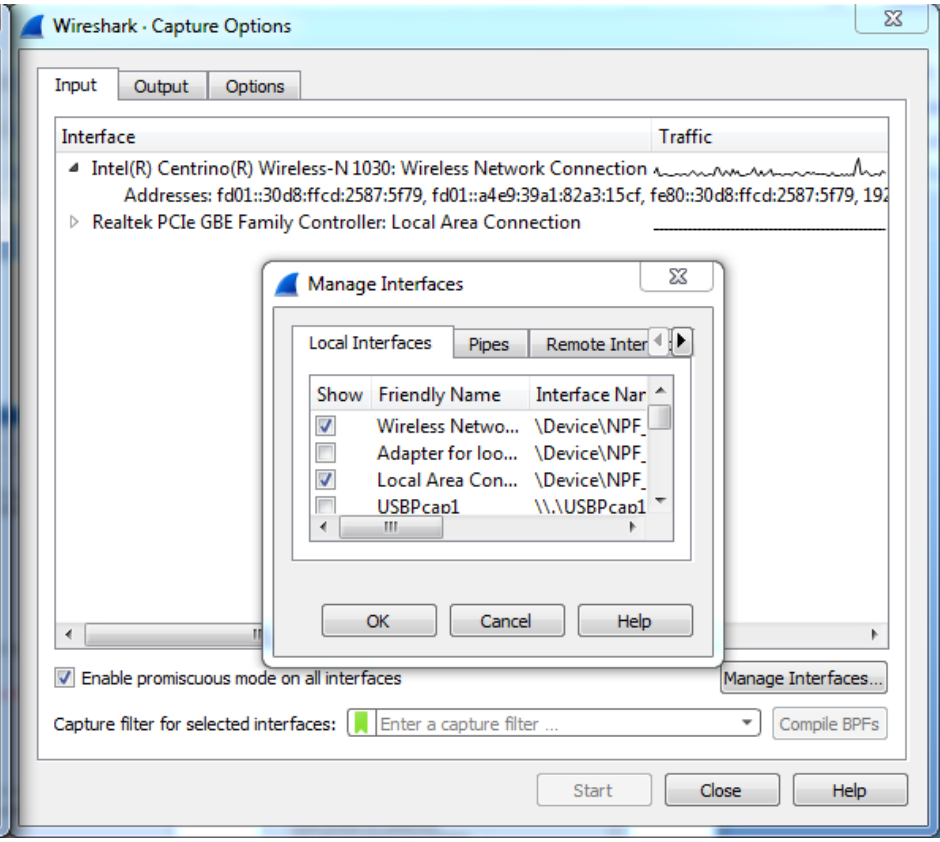
Capture Interfaces





Interface Details

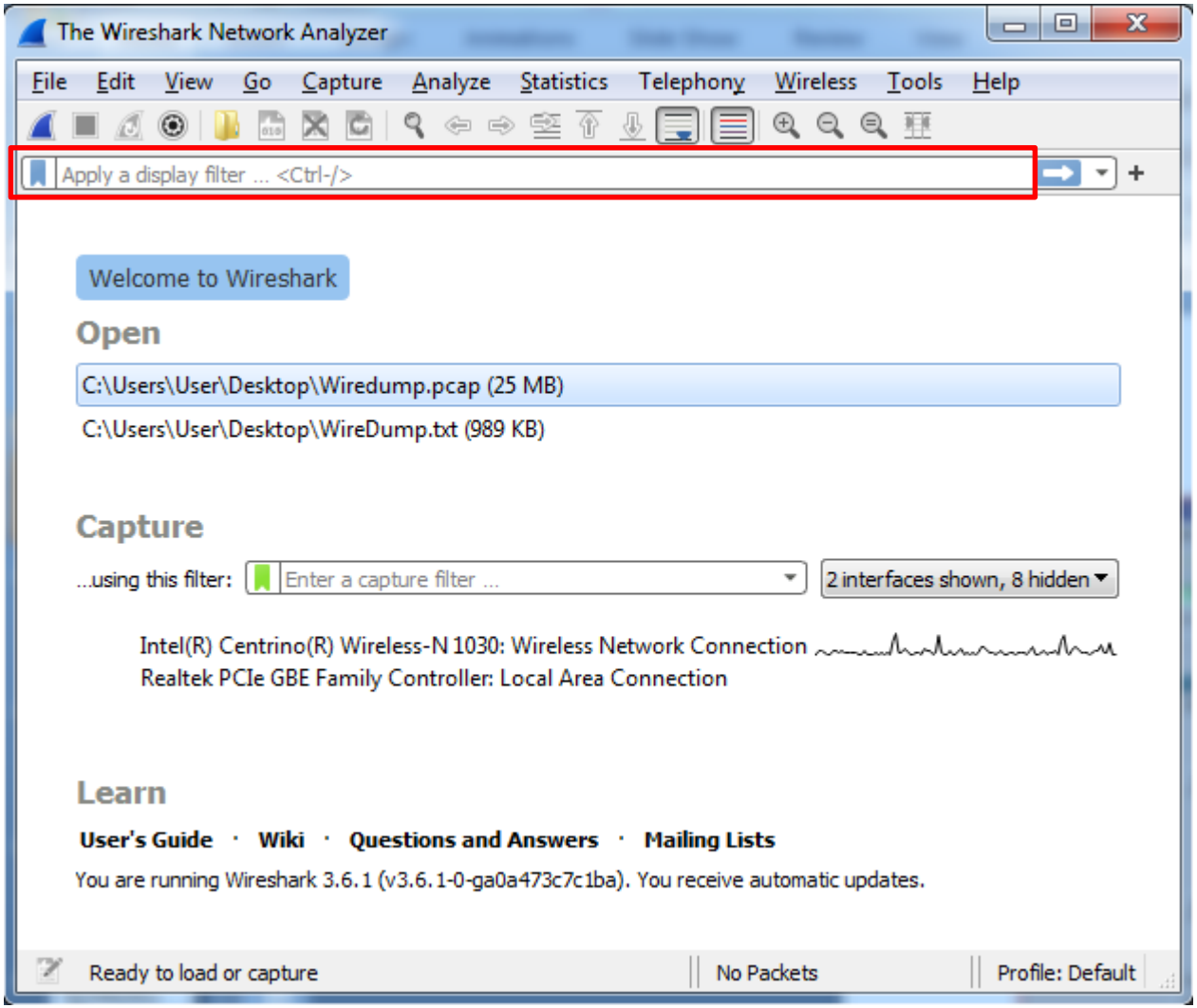




Display Filters (Post-Filters)

- Display filters (also called post-filters) only filter the view of what you are seeing
  - In analysis
  - All packets in the capture still exist in the trace
- Display filters use their own format and are much more powerful then capture filters

Display Filter



Display Filter Examples

ip.src==10.1.11.00/24

ip.addr==192.168.1.10 && ip.addr==192.168.1.20

tcp.port==80 || tcp.port==3389

!(ip.addr==192.168.1.10 && ip.addr==192.168.1.20)

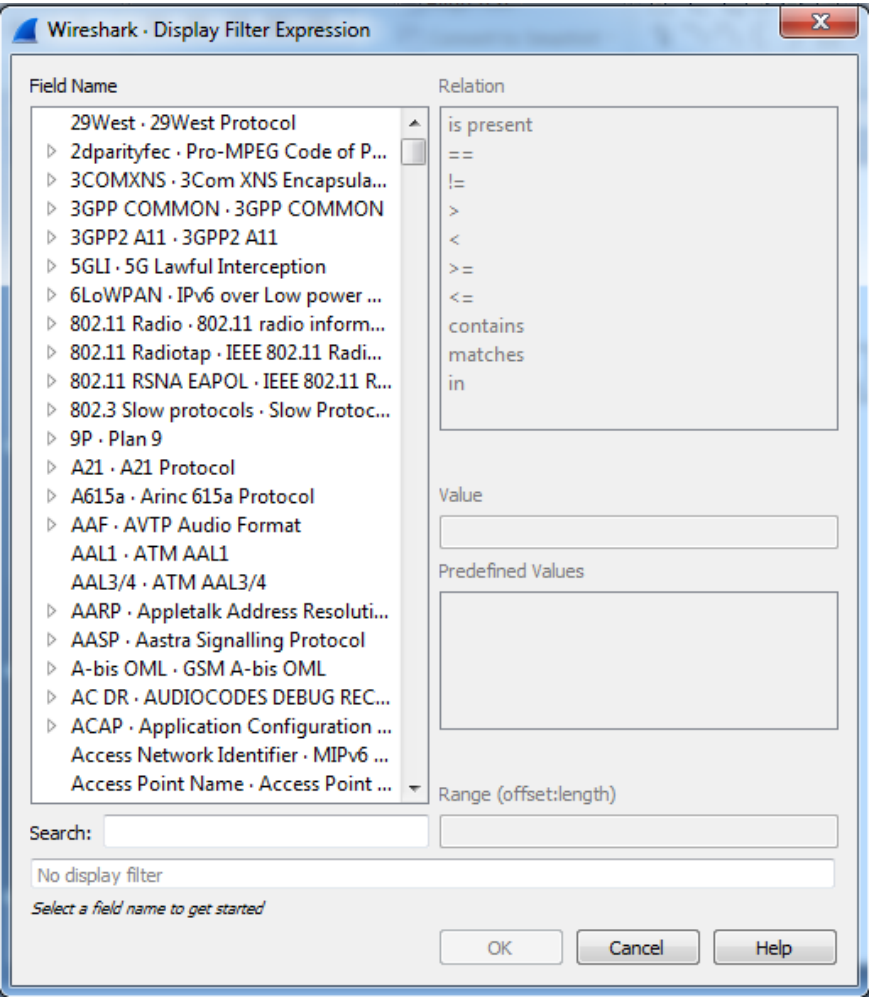
(ip.addr==192.168.1.10 && ip.addr==192.168.1.20) && (tcp.port==445
    || tcp.port==139)

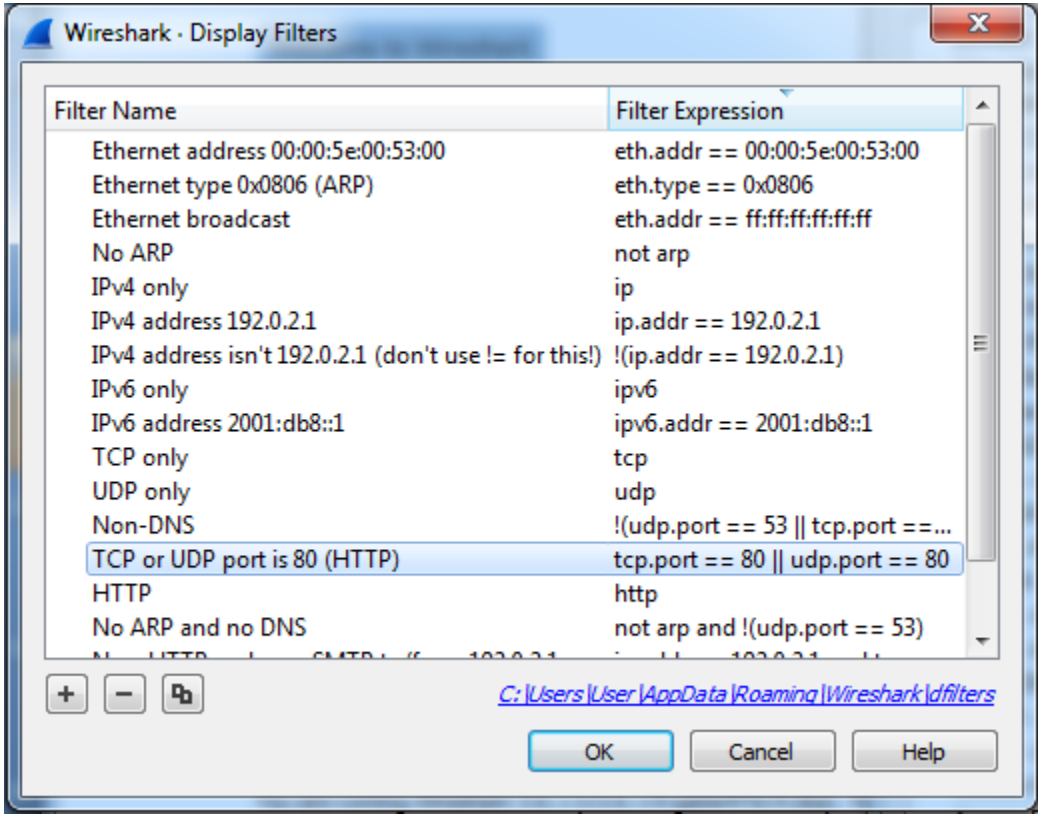
(ip.addr==192.168.1.10 && ip.addr==192.168.1.20) && (udp.port==67
    || udp.port==68)

tcp.dstport == 80

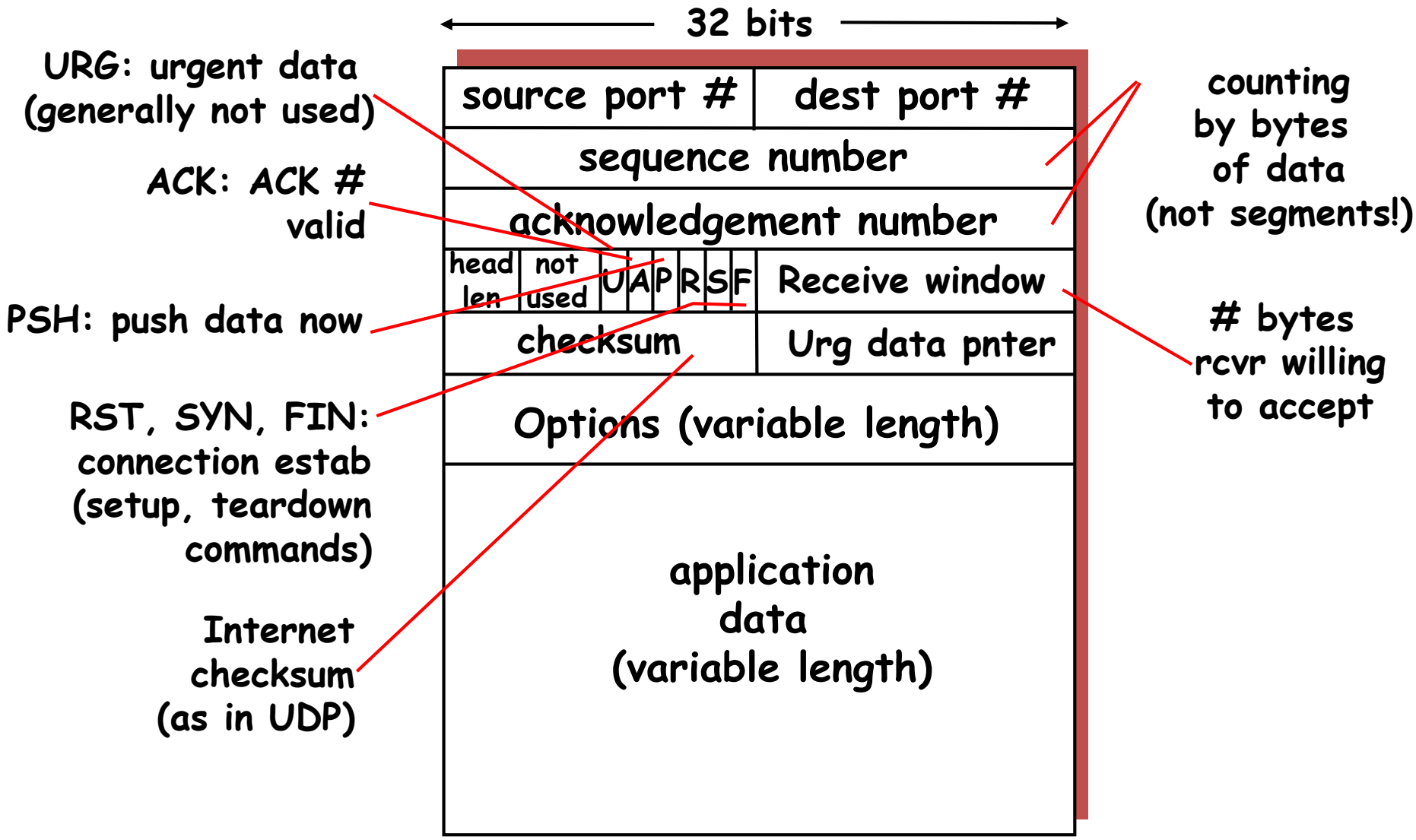
Display Filter

Syntax:	**Protocol** .	**String 1** .	**String 2**	**Comparison operator**	**Value**	**Logical Operations**	**Other expression**
Example:	ftp	passive	ip	==	10.2.3.4	xor	icmp.type





TCP Segment Structure

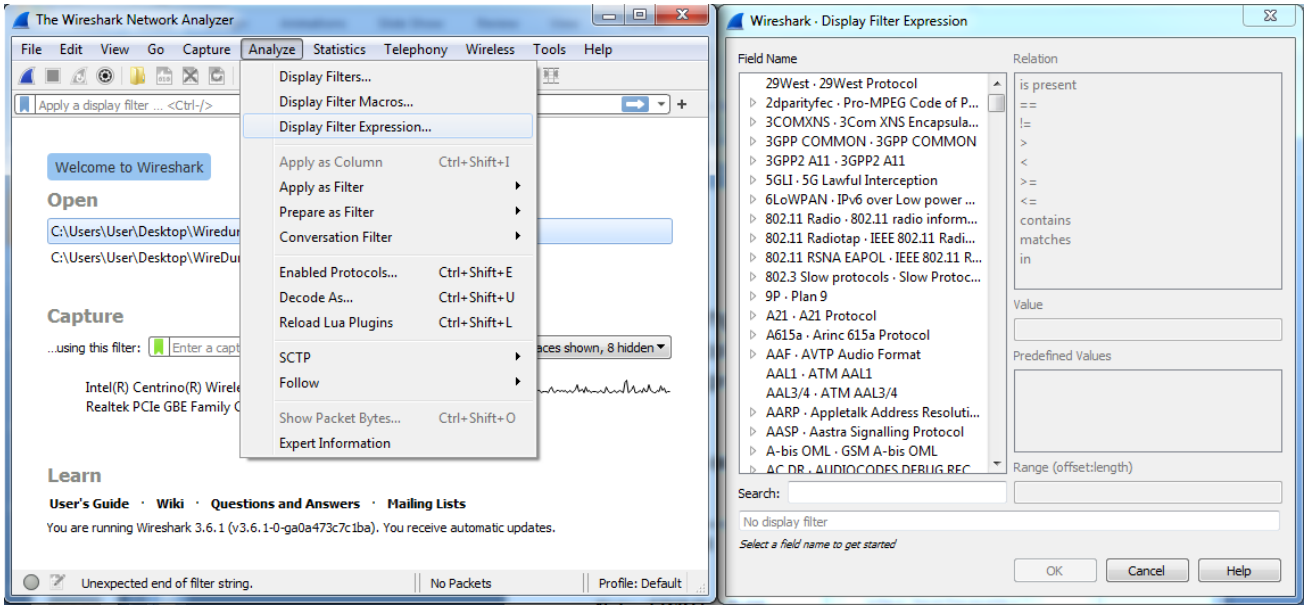


Display Filter

- String1, String2 (Optional settings):
  - Sub protocol categories inside the protocol.
  - Look for a protocol and then click on the "+" character.
  - Example:
  - **tcp.srcport == 80**
  - **tcp.flags == 2**
    - **SYN packet**
    - **Tcp.flags.syn==1**
  - **tcp.flags == 18**
    - **SYN/ACK**

  - **TCP Flag field:**

U	A	P	R	S	F
R	C	S	S	Y	I
G	K	H	T	N	N



Display Filter Expressions

- snmp || dns || icmp
  - Display the SNMP or DNS or ICMP traffics.
- tcp.port == 25
  - Display packets with TCP source or destination port 25.
- tcp.flags
  - Display packets having a TCP flag.
- tcp.flags == 0x02
  - Display packets with a TCP SYN flag.

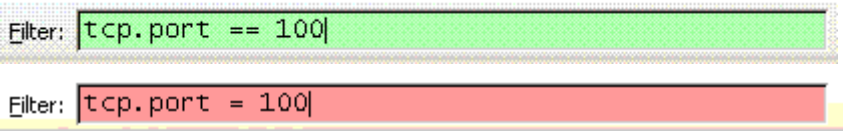
Six comparison operators are available:

English format:	C like format:	Meaning:
eq	==	Equal
ne	!=	Not equal
gt	>	Greater than
lt	<	Less than
ge	>=	Greater or equal
le	<=	Less or equal

Logical expressions:

English format:	C like format:	Meaning:
and	&&	Logical AND
or	||	Logical OR
xor	^^	Logical XOR
not	!	Logical NOT

If the filter syntax is correct, it will be highlighted in green, otherwise if there is a syntax mistake it will be highlighted in red.



Correct syntax

Wrong syntax

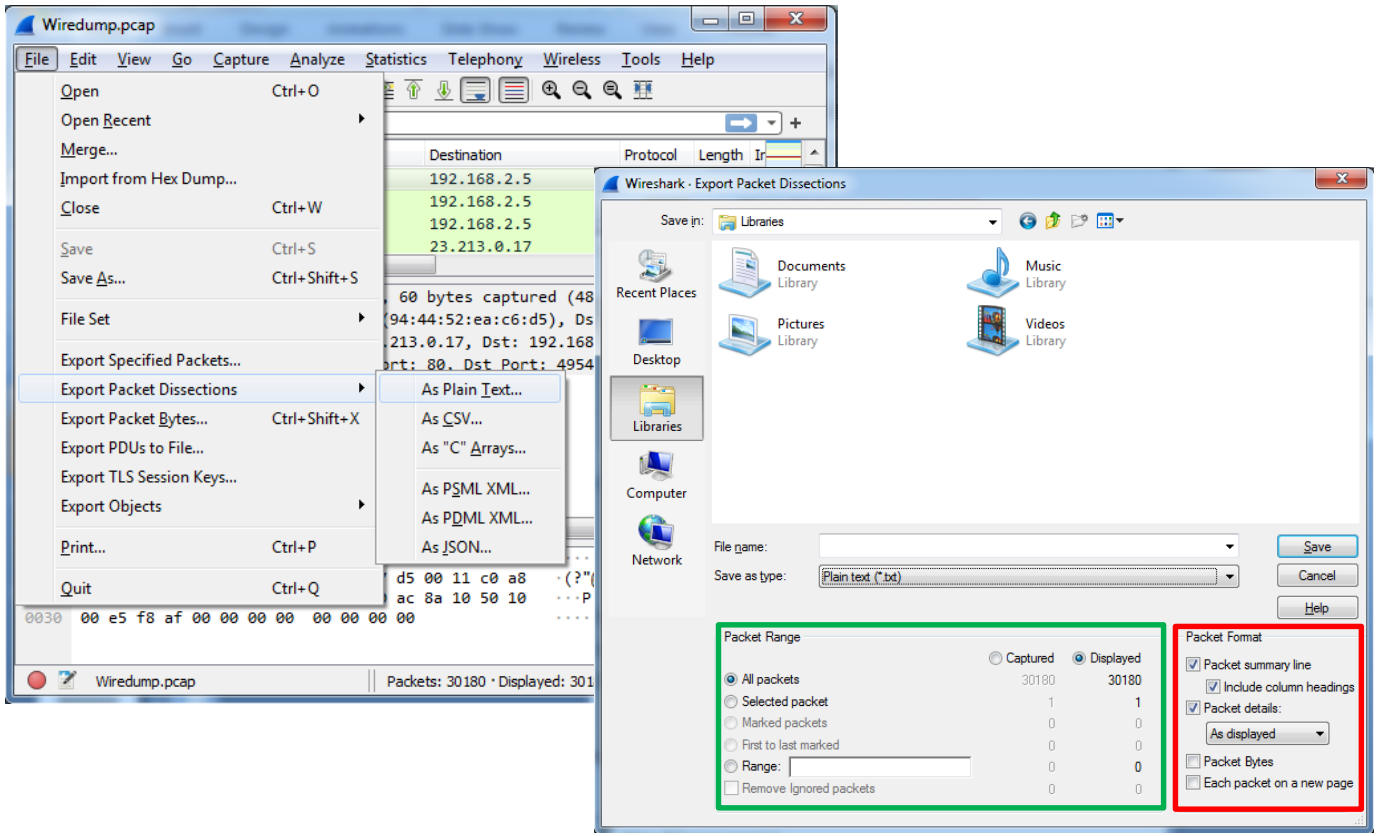
Save Filtered Packets after Using Display Filter

- We can also save all filtered packets in text file for further analysis
- Operation:

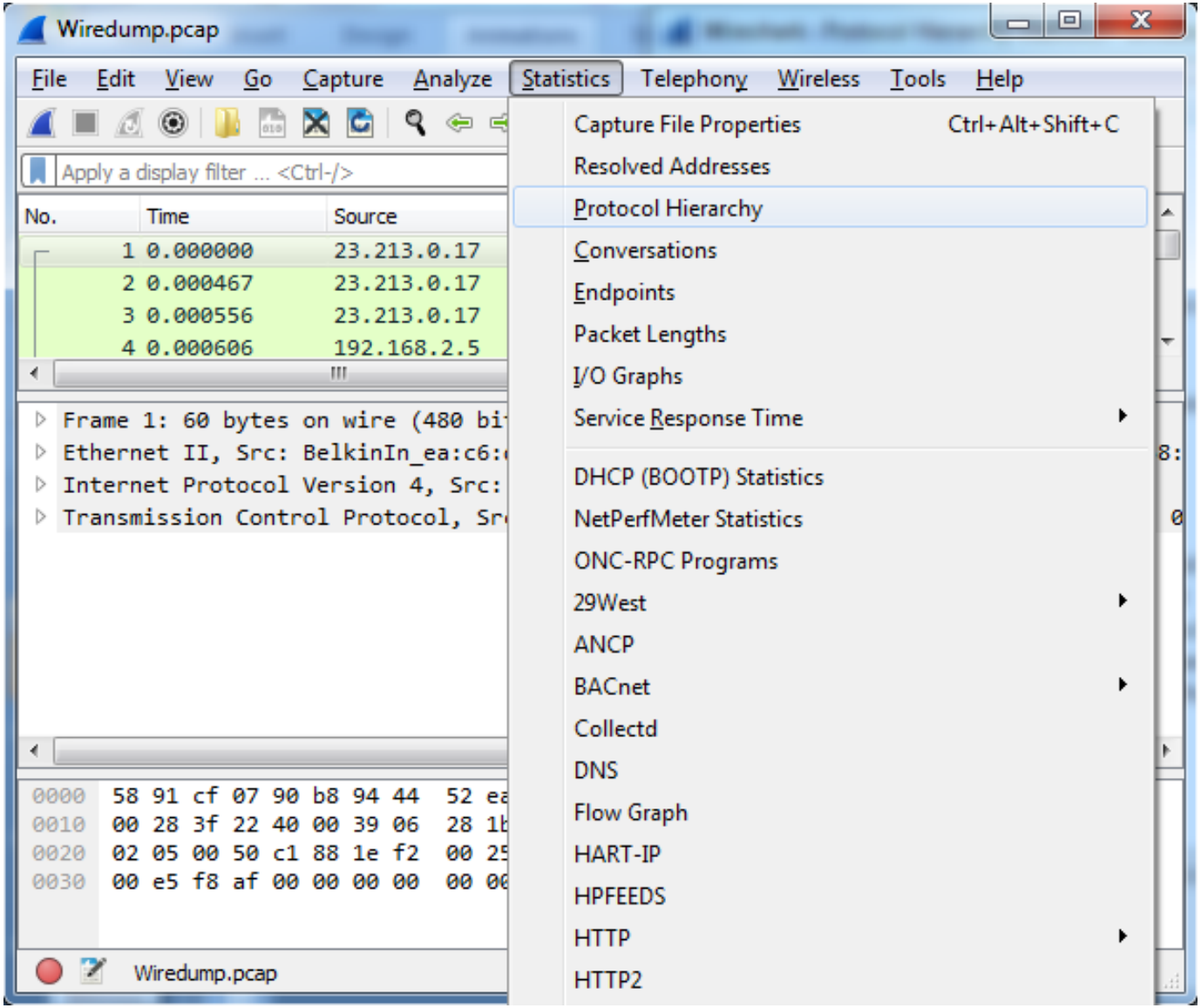
File→Export packet dissections →as "plain text" file

1). In "packet range" option, select "Displayed"

2). Choose "summary line" or "detail"



Protocol Hierarchy



Protocol Hierarchy (contd.)

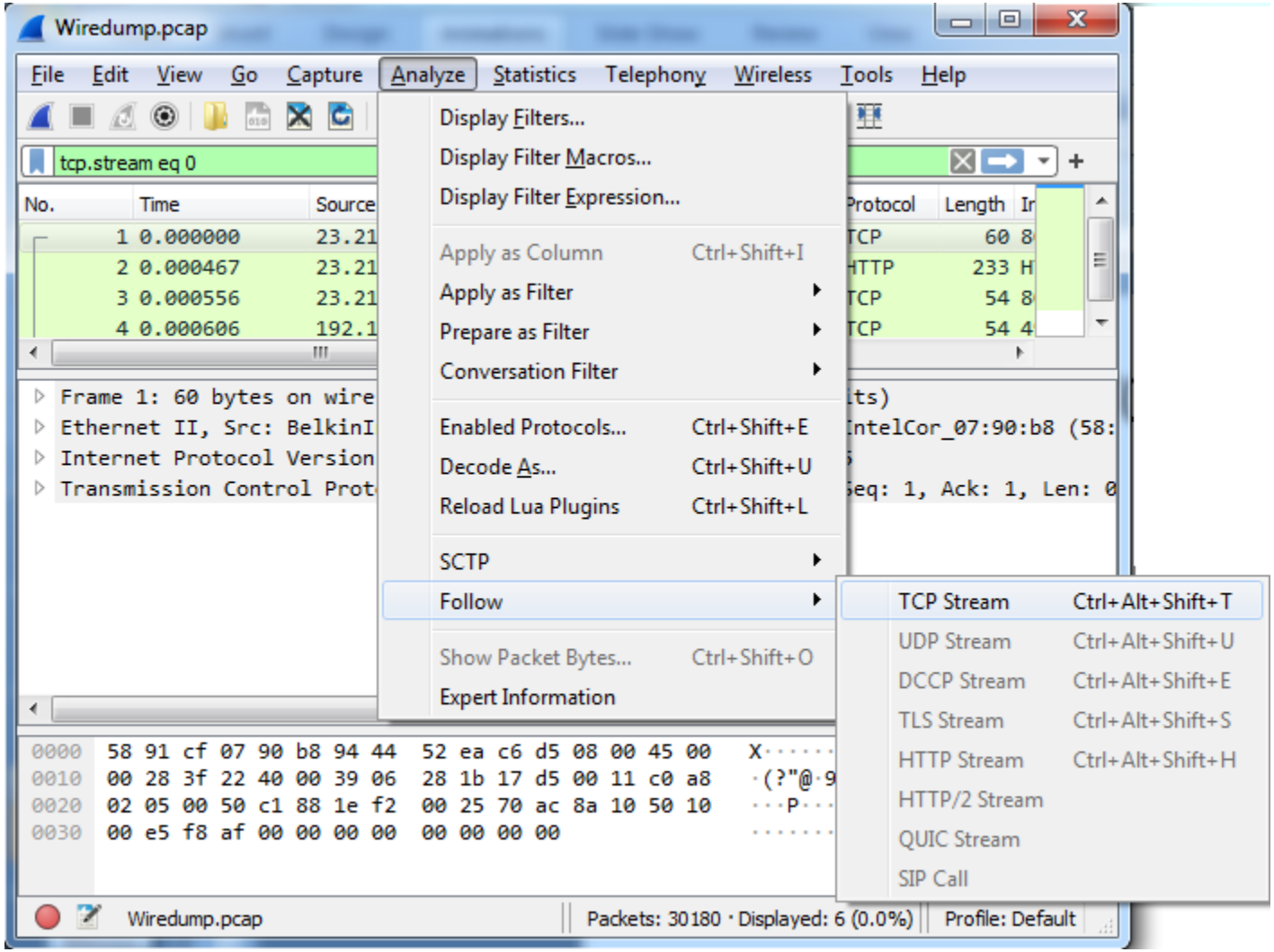
Wireshark · Protocol Hierarchy Statistics · Wiredump.pcap

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	30180	100.0	26368284	5445 k	0	0	0
Ethernet	100.0	30180	1.6	422520	87 k	0	0	0
Internet Protocol Version 6	0.0	9	0.0	360	74	0	0	0
User Datagram Protocol	0.0	3	0.0	24	4	0	0	0
DHCPv6	0.0	3	0.0	261	53	3	261	53
Internet Control Message Protocol v6	0.0	6	0.0	192	39	6	192	39
Internet Protocol Version 4	99.9	30160	2.3	603228	124 k	0	0	0
User Datagram Protocol	8.9	2691	0.1	21528	4445	0	0	0
Simple Service Discovery Protocol	0.2	63	0.1	16611	3430	63	16611	3430
QUIC IETF	8.3	2496	5.8	1517524	313 k	2439	1471948	303 k
Malformed Packet	0.0	2	0.0	0	0	2	0	0
NetBIOS Name Service	0.1	23	0.0	1240	256	23	1240	256
NetBIOS Datagram Service	0.0	4	0.0	765	157	0	0	0
SMB (Server Message Block Protocol)	0.0	4	0.0	437	90	0	0	0
SMB MailSlot Protocol	0.0	4	0.0	100	20	0	0	0
Microsoft Windows Browser Protocol	0.0	4	0.0	93	19	4	93	19
Multicast Domain Name System	0.0	4	0.0	244	50	4	244	50
Dynamic Host Configuration Protocol	0.0	3	0.0	900	185	3	900	185
Domain Name System	0.3	95	0.1	14561	3006	95	14561	3006
Data	0.2	58	0.0	2278	470	58	2278	470
Transmission Control Protocol	91.0	27457	90.2	23775639	4909 k	20838	15200882	3139 k
Transport Layer Security	22.1	6658	82.3	21701465	4481 k	6513	18459368	3812 k

No display filter.

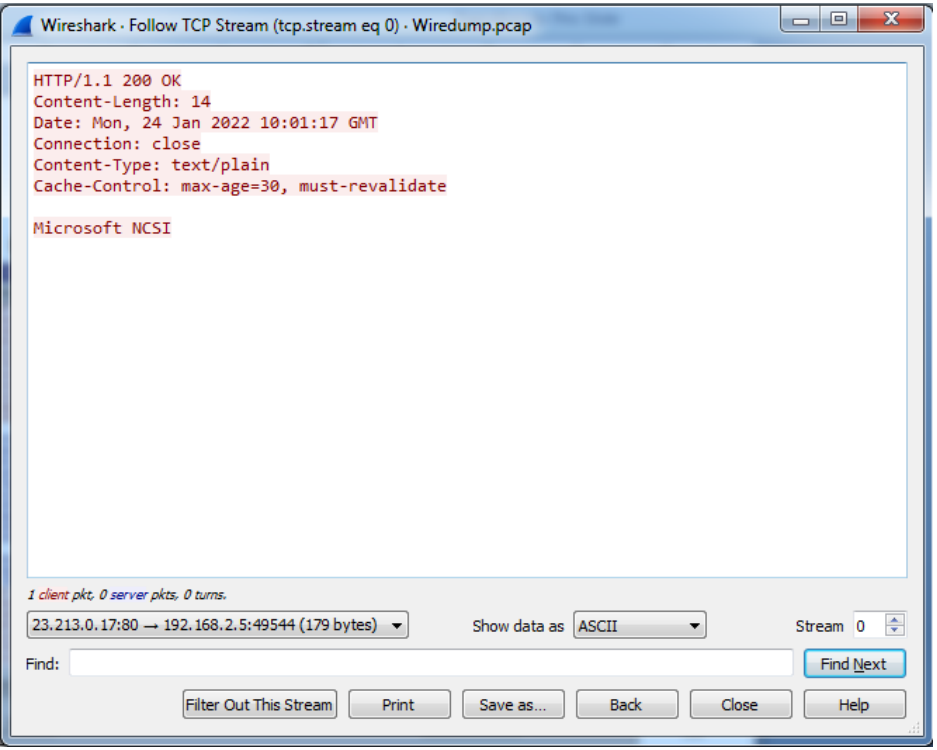
Close Copy Help

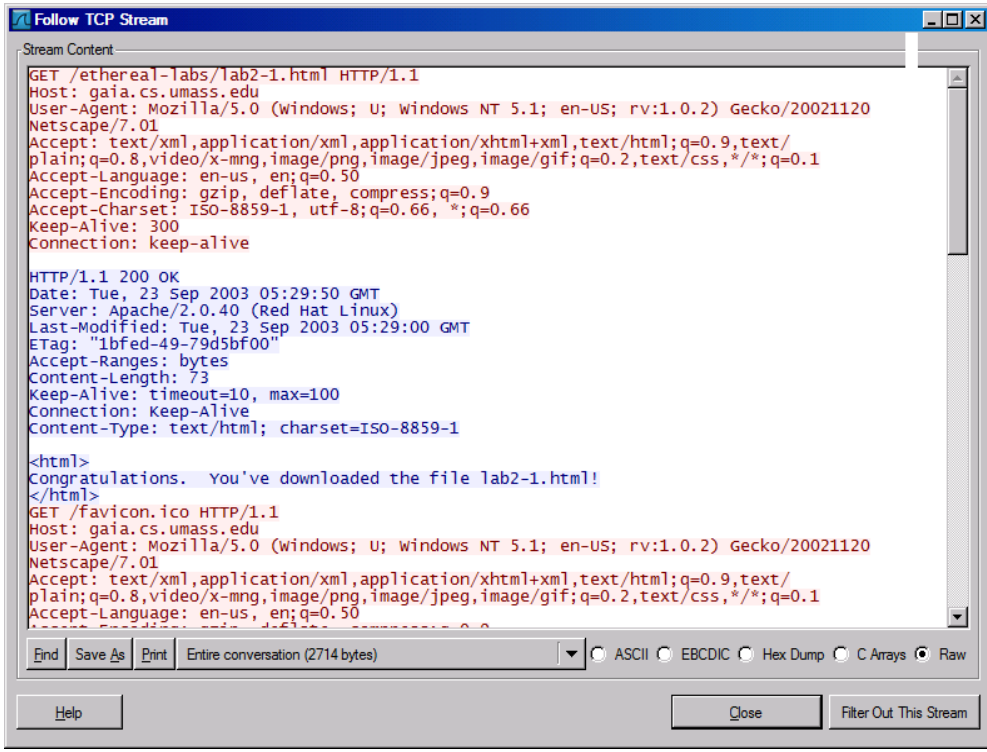
Follow TCP Stream



Follow TCP Stream

Red - stuff you sent **Blue - stuff you get**

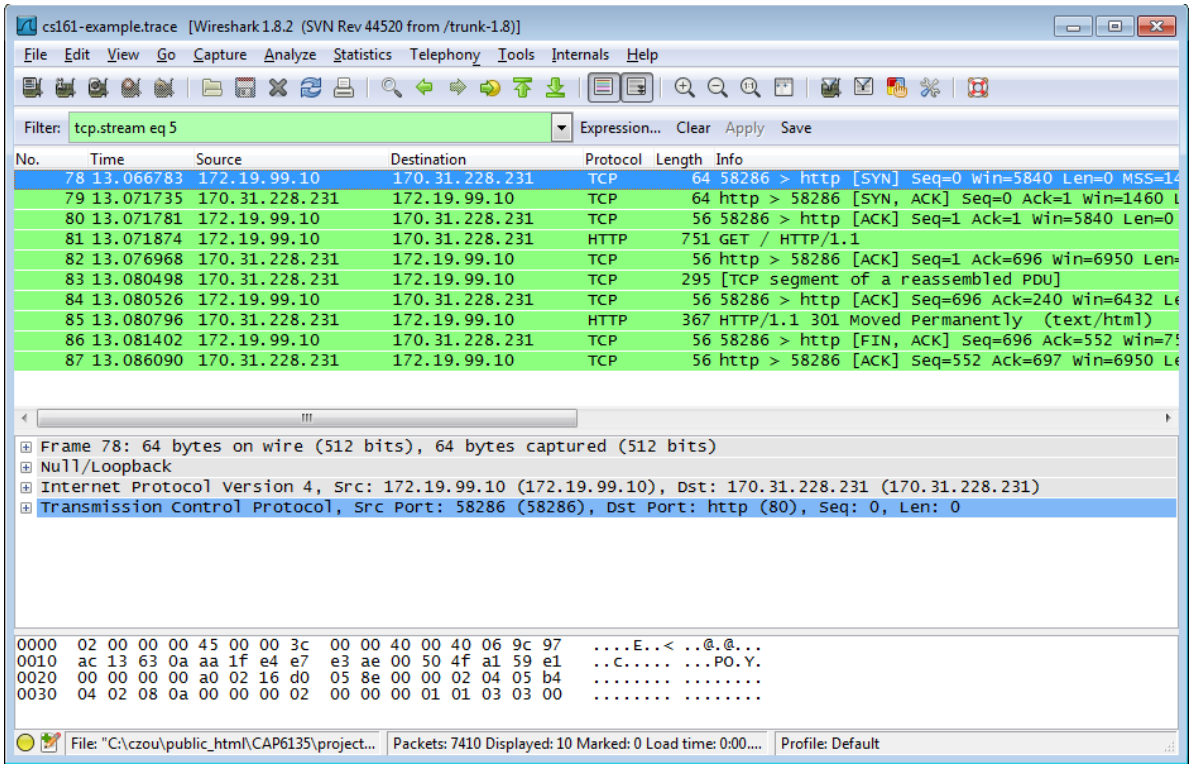




Old version

Filter out/in Single TCP Stream

- When click “filter out this TCP stream” in previous page’s box, new filter string will contain like:
  - http and !(tcp.stream eq 5)
- So, if you use “tcp.stream eq 5” as filter string, you keep this HTTP session



Old version

Expert Info

Wiredump.pcap

File Edit View Go Capture Analyze Statistics Telephony

Apply a display filter ... <Ctrl-/>

No.	Time	Source
3	0.000556	23.21
4	0.000606	192.1
5	0.000634	192.1
6	0.014648	192.1

Frame 5: 54 bytes on wire
Ethernet II, Src: IntelCo
Internet Protocol Version
Transmission Control Prot

```
0000  94 44 52 ea c6 d5 58 91  cf 07 90 b8 08 00 45 00
0010  00 28 3c 90 40 00 80 06  e3 ac c0 a8 02 05 17 d5
0020  00 11 c1 88 00 50 70 ac  8a 11 1e f2 00 d9 50 10
0030  10 3b e8 a4 00 00
```

Wiredump.pcap | Packets:

- Display Filters...
- Display Filter Macros...
- Display Filter Expression...
- Apply as Column Ctrl
- Apply as Filter
- Prepare as Filter
- Conversation Filter
- Enabled Protocols... Ctrl
- Decode As... Ctrl
- Reload Lua Plugins Ctrl
- SCTP
- Follow
- Show Packet Bytes... Ctrl
- Expert Information

Wireshark · Expert Information · Wiredump.pcap

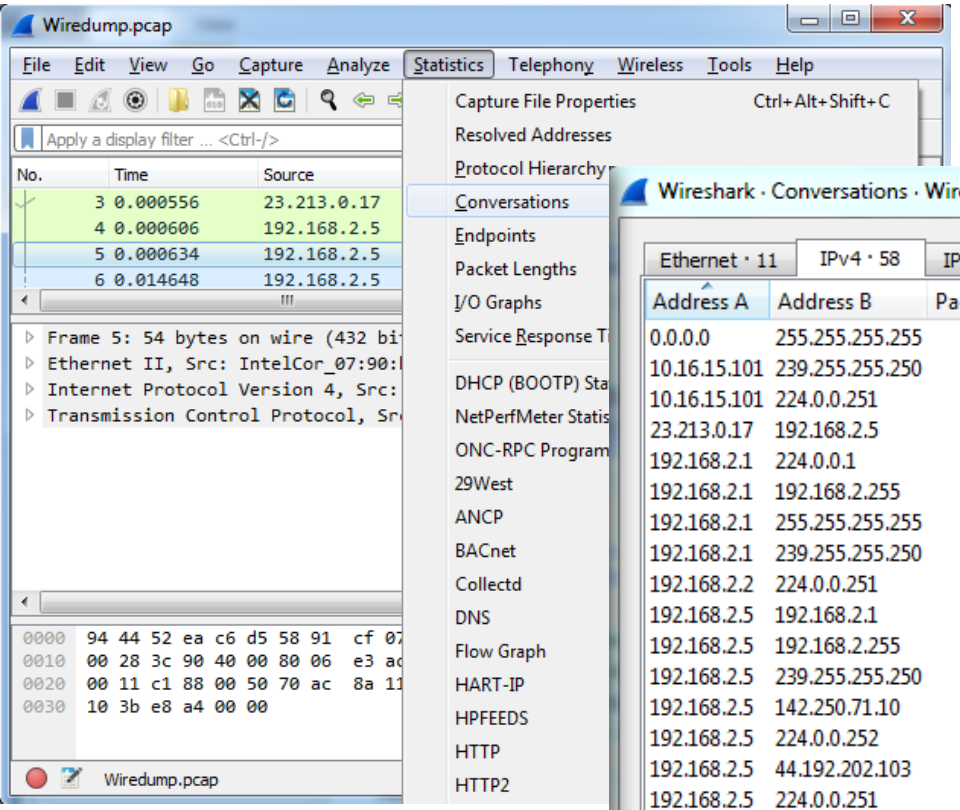
Severity	Summary	Group	Protocol
Error	TLSCiphertext length MUST NOT exceed 2^14 + 2048	Protocol	TLS
Error	Malformed Packet (Exception occurred)	Malformed	TLS
Error	Vector length 185 is too large, truncating it to 7	Malformed	TLS
Error	C:\gitlab-builds\builds\-fyeYoMP\1\wireshark\wireshark\...	Malformed	TLS
Warning	DNS query retransmission. Original request in frame 19250	Protocol	DNS
Warning	Connection reset (RST)	Sequence	TCP
Warning	ACKed segment that wasn't captured (common at capture...	Sequence	TCP
Warning	This frame is a (suspected) out-of-order segment	Sequence	TCP
Warning	Ignored Unknown Record	Protocol	TLS
Warning	TCP Zero Window segment	Sequence	TCP
Warning	TCP window specified by the receiver is now completely full	Sequence	TCP
Warning	Previous segment(s) not captured (common at capture sta...	Sequence	TCP
Warning	D-SACK Sequence	Sequence	TCP
Warning	Failed to create decryption context: Secrets are not available	Decryption	QUIC
Note	This frame is a (suspected) fast retransmission	Sequence	TCP
Note	"Time To Live" != 255 for a packet sent to the Local Netwo...	Sequence	IPv4
Note	Duplicate ACK (#1)	Sequence	TCP
Note	This frame is a (suspected) spurious retransmission	Sequence	TCP
Note	This frame is a (suspected) retransmission	Sequence	TCP
Note	This session reuses previously negotiated keys (Session res...	Sequence	TLS
Note	"Time To Live" only 4	Sequence	IPv4

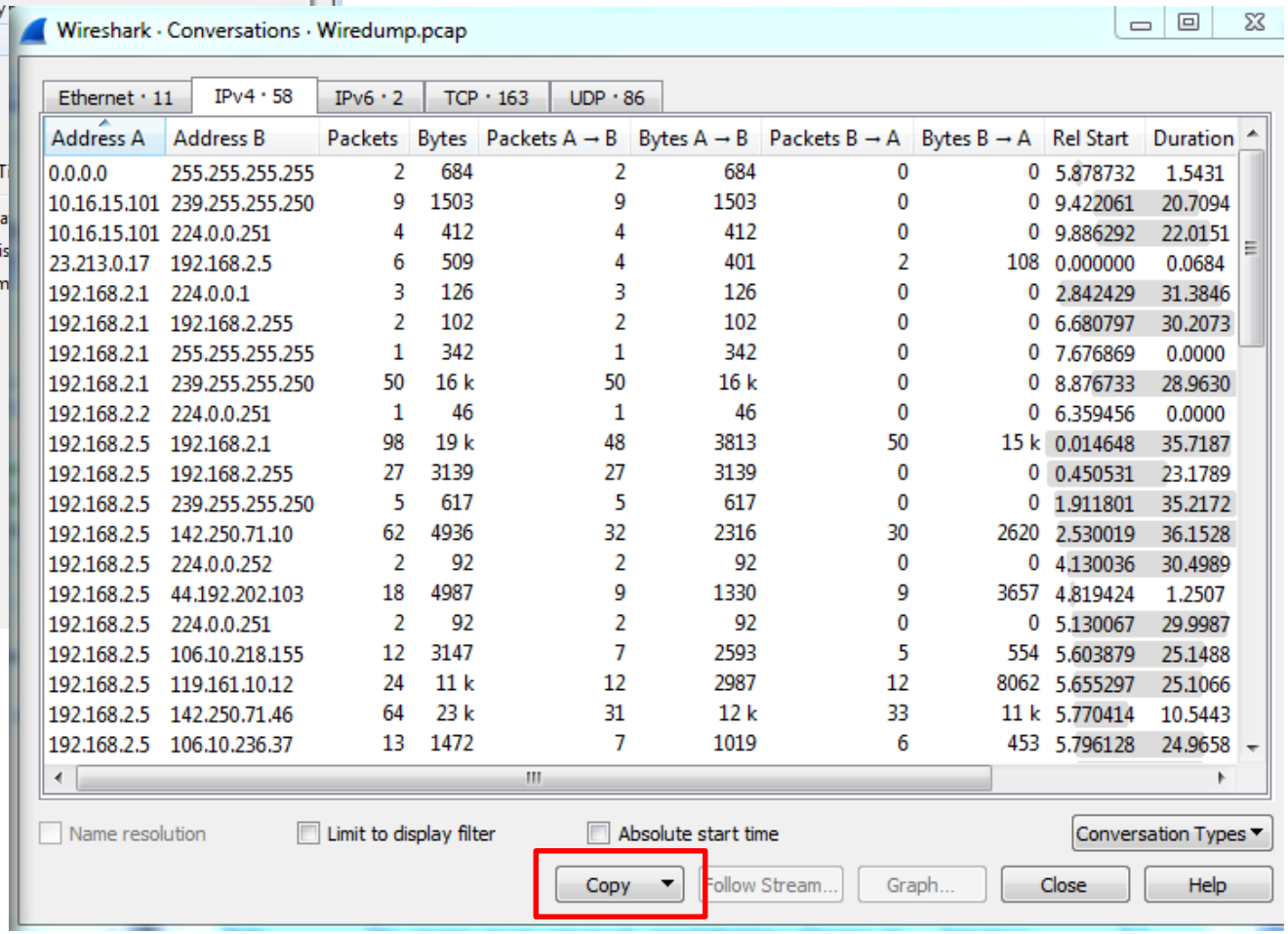
No display filter set.

Limit to Display Filter | Group by summary | Search: | Show...

Close | Help

Conversations

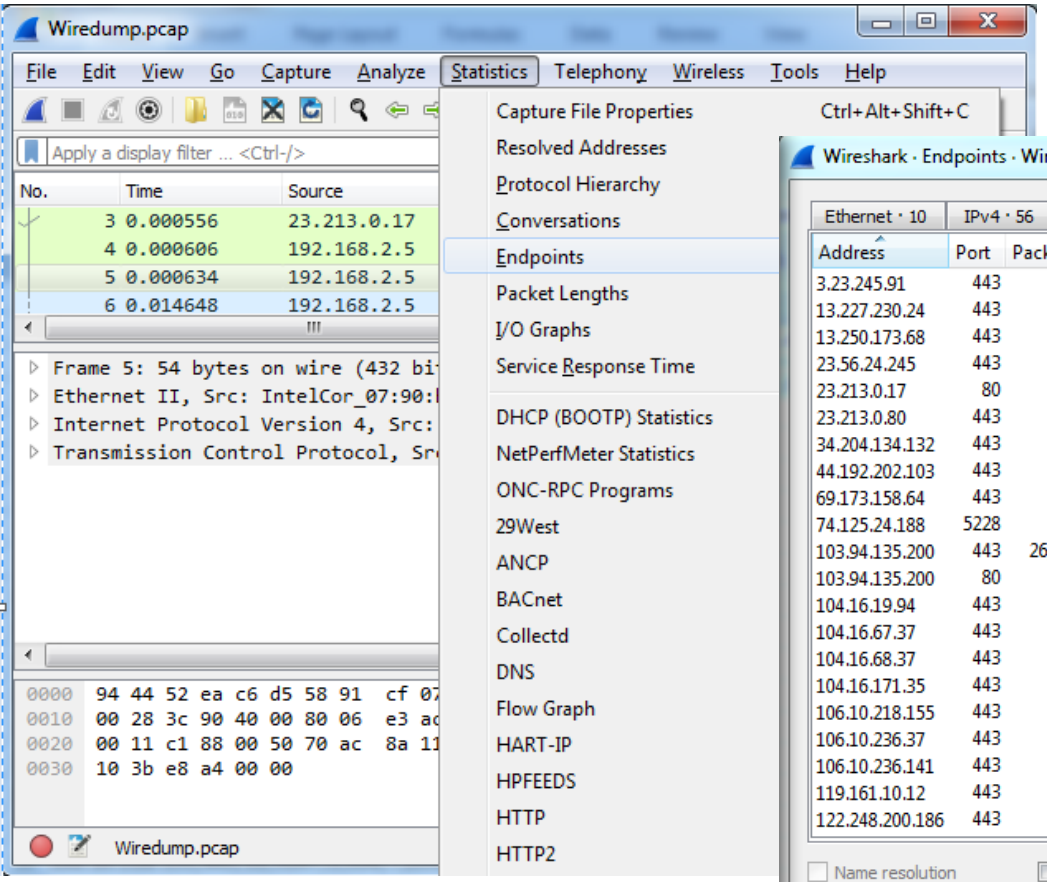


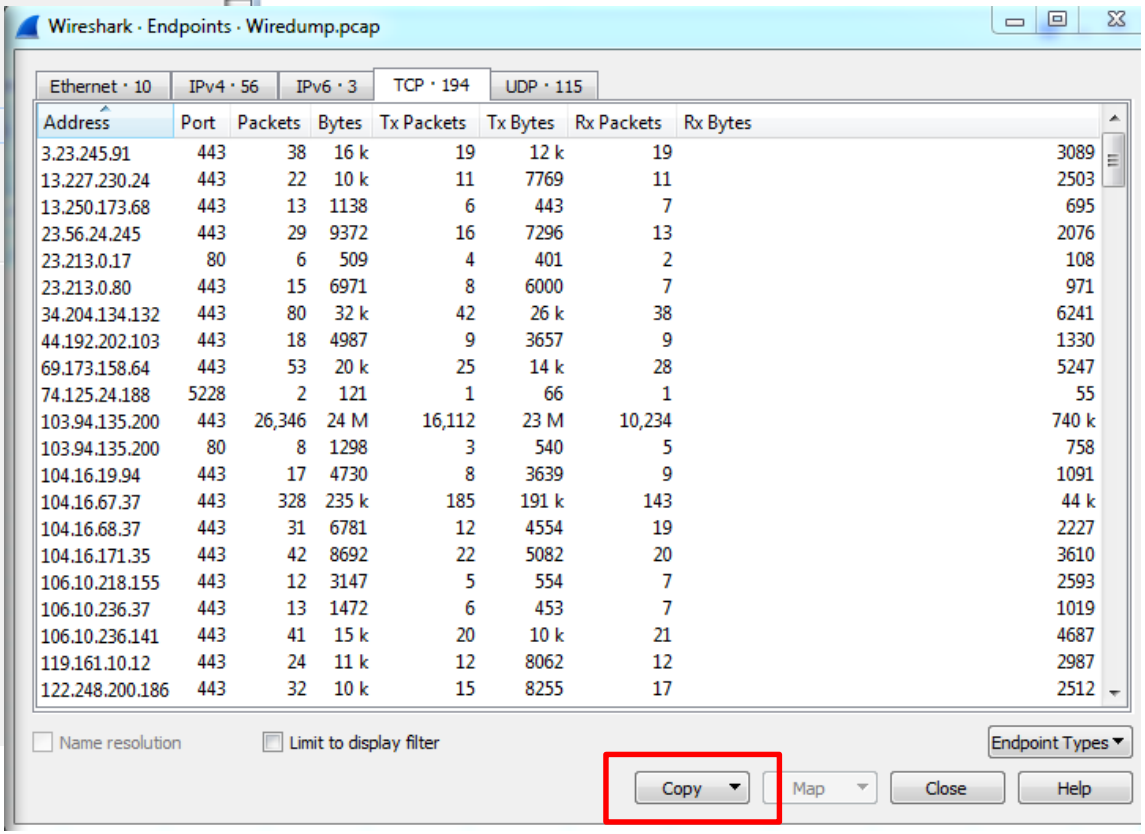


Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
0.0.0.0	255.255.255.255	2	684	2	684	0	0	5.878732	1.5431
10.16.15.101	239.255.255.250	9	1503	9	1503	0	0	9.422061	20.7094
10.16.15.101	224.0.0.251	4	412	4	412	0	0	9.886292	22.0151
23.213.0.17	192.168.2.5	6	509	4	401	2	108	0.000000	0.0684
192.168.2.1	224.0.0.1	3	126	3	126	0	0	2.842429	31.3846
192.168.2.1	192.168.2.255	2	102	2	102	0	0	6.680797	30.2073
192.168.2.1	255.255.255.255	1	342	1	342	0	0	7.676869	0.0000
192.168.2.1	239.255.255.250	50	16 k	50	16 k	0	0	8.876733	28.9630
192.168.2.2	224.0.0.251	1	46	1	46	0	0	6.359456	0.0000
192.168.2.5	192.168.2.1	98	19 k	48	3813	50	15 k	0.014648	35.7187
192.168.2.5	192.168.2.255	27	3139	27	3139	0	0	0.450531	23.1789
192.168.2.5	239.255.255.250	5	617	5	617	0	0	1.911801	35.2172
192.168.2.5	142.250.71.10	62	4936	32	2316	30	2620	2.530019	36.1528
192.168.2.5	224.0.0.252	2	92	2	92	0	0	4.130036	30.4989
192.168.2.5	44.192.202.103	18	4987	9	1330	9	3657	4.819424	1.2507
192.168.2.5	224.0.0.251	2	92	2	92	0	0	5.130067	29.9987
192.168.2.5	106.10.218.155	12	3147	7	2593	5	554	5.603879	25.1488
192.168.2.5	119.161.10.12	24	11 k	12	2987	12	8062	5.655297	25.1066
192.168.2.5	142.250.71.46	64	23 k	31	12 k	33	11 k	5.770414	10.5443
192.168.2.5	106.10.236.37	13	1472	7	1019	6	453	5.796128	24.9658

- Use the "Copy" button to copy all text into clipboard (file)
  - Analyze the file to get what statistics you want

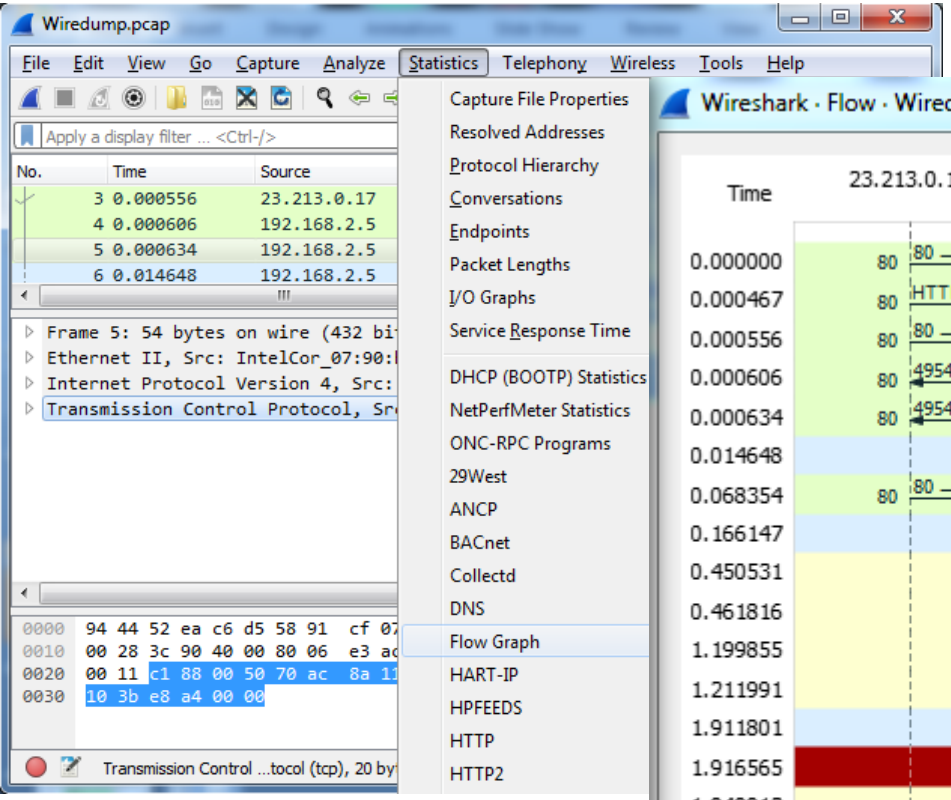
Find EndPoint Statistics

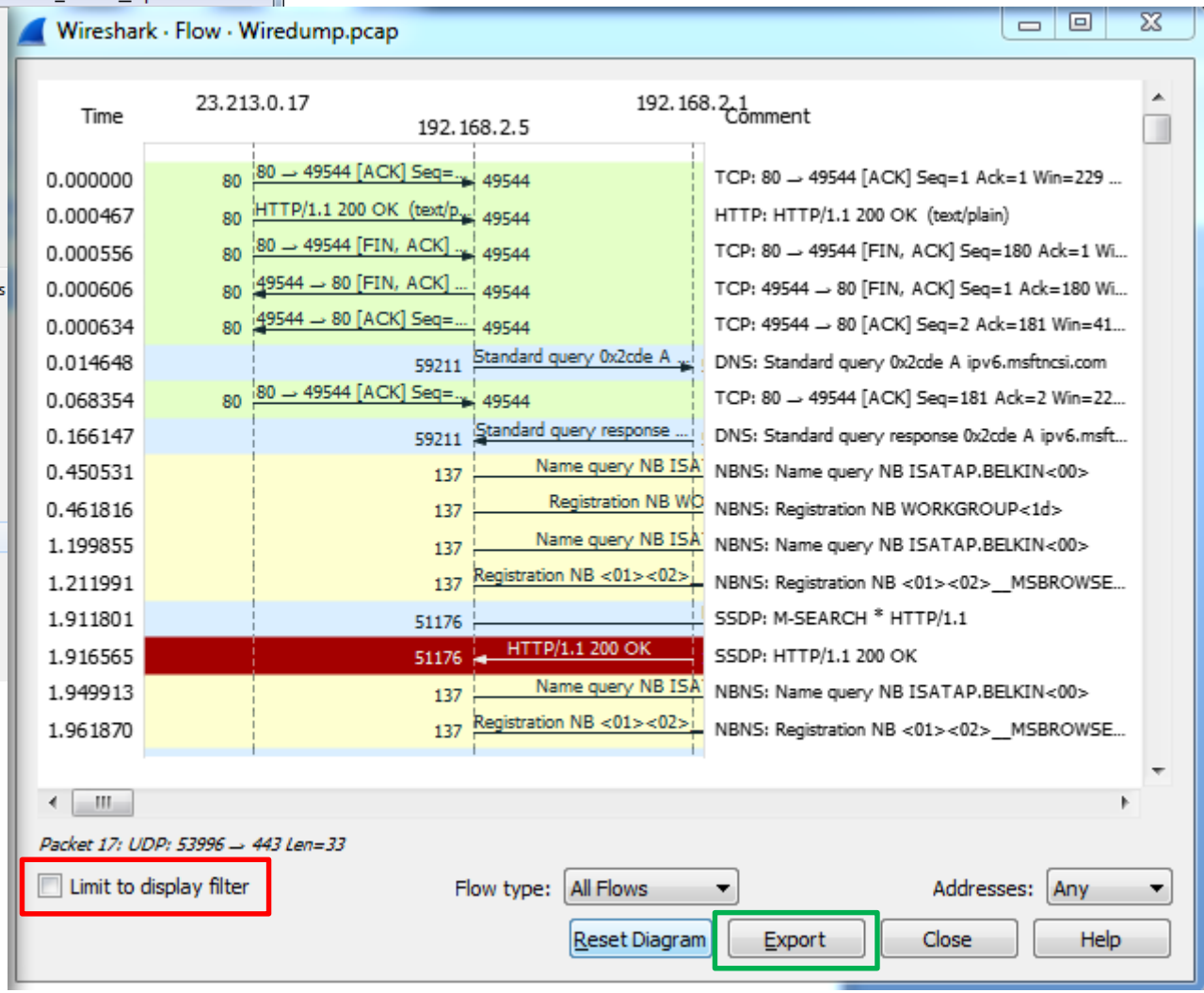




- You can sort by field
- Use the "Copy" button to copy all text into clipboard (file)
  - Analyze the file to get what statistics you want

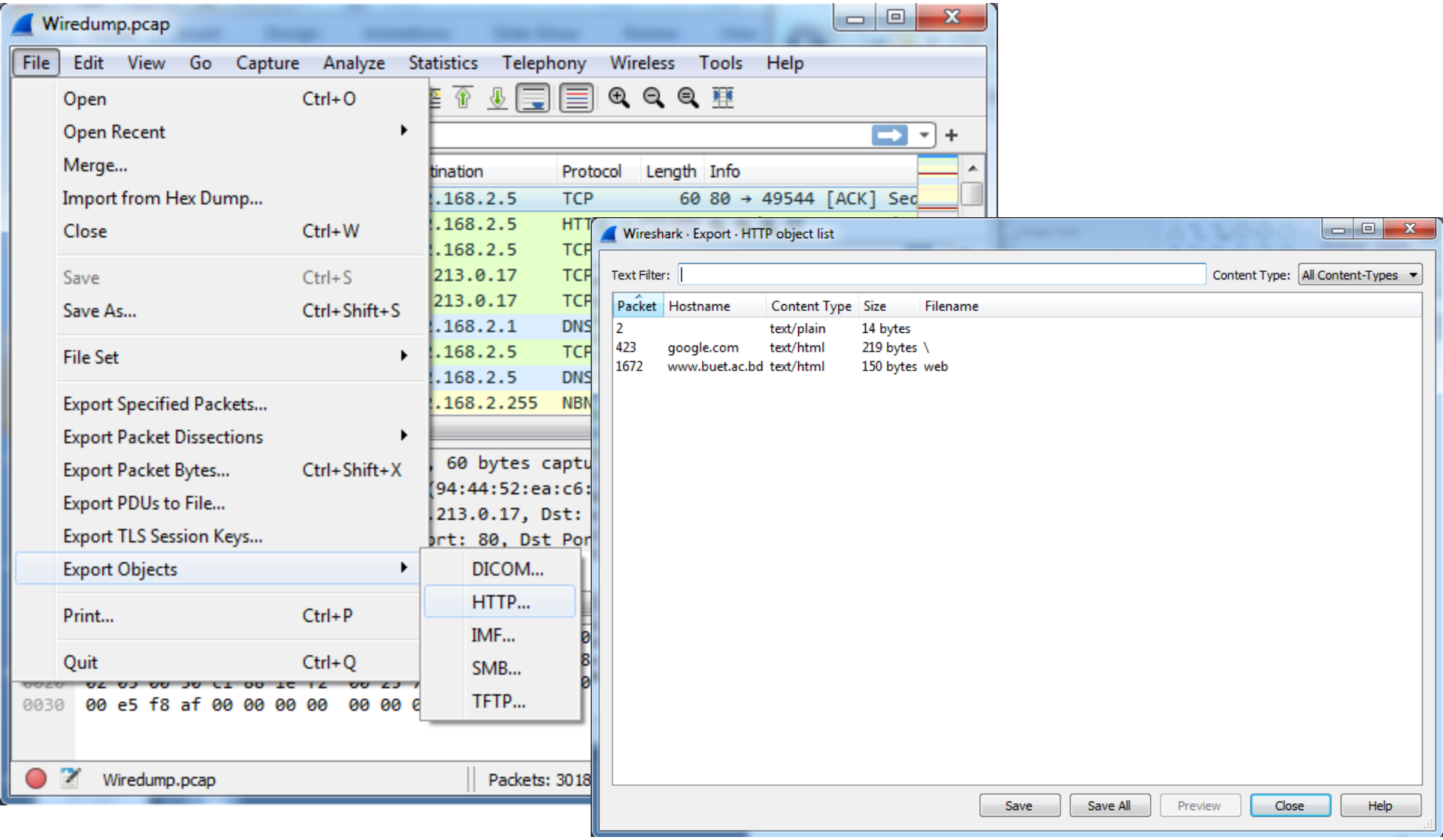
Flow Graphs



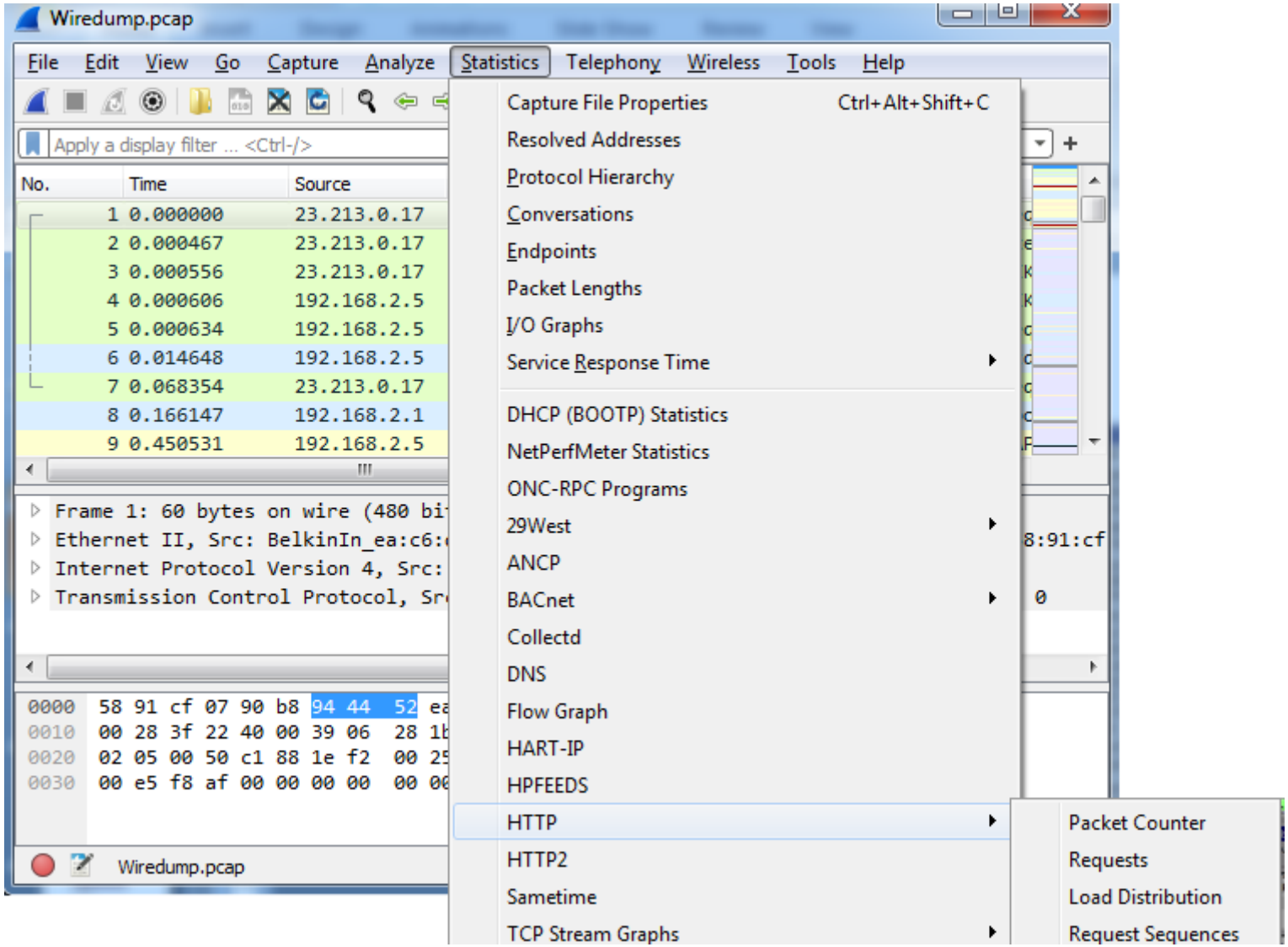


- "Limit to display filter" option enable showing the flow of packets displayed in the main interface
- "Export" enables exporting the flow graph shown above

Export HTTP


Wiredump.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Open Ctrl+O
Open Recent
Merge...
Import from Hex Dump...
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
File Set
Export Specified Packets...
Export Packet Dissections
Export Packet Bytes... Ctrl+Shift+X
Export PDUs to File...
Export TLS Session Keys...
Export Objects
Print... Ctrl+P
Quit Ctrl+Q
DICOM...
HTTP...
IMF...
SMB...
TFTP...
Protocol Length Info
Wireshark · Export · HTTP object list
Text Filter:
Content Type: All Content-Types
Packet Hostname Content Type Size Filename
2 text/plain 14 bytes
423 google.com text/html 219 bytes \
1672 www.buet.ac.bd text/html 150 bytes web
Save Save All Preview Close Help

HTTP Analysis



HTTP Analysis – Load Distribution

Wireshark · Load Distribution · Wiredump.pcap

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
HTTP Responses by Server Address	6				0.0002	100%	0.0100	0.000
23.213.0.17	1				0.0000	16.67%	0.0100	0.000
OK	1				0.0000	100.00%	0.0100	0.000
192.168.2.1	3				0.0001	50.00%	0.0100	1.917
OK	3				0.0001	100.00%	0.0100	1.917
142.250.182.142	1				0.0000	16.67%	0.0100	9.990
OK	1				0.0000	100.00%	0.0100	9.990
103.94.135.200	1				0.0000	16.67%	0.0100	23.858
OK	1				0.0000	100.00%	0.0100	23.858
HTTP Requests by Server	62				0.0021	100%	0.1600	9.024
HTTP Requests by Server Address	62				0.0021	100.00%	0.1600	9.024
239.255.255.250	60				0.0020	96.77%	0.1600	9.024
239.255.255.250:1900	60				0.0020	100.00%	0.1600	9.024
142.250.182.142	1				0.0000	1.61%	0.0100	9.912
google.com	1				0.0000	100.00%	0.0100	9.912
103.94.135.200	1				0.0000	1.61%	0.0100	23.855
www.buet.ac.bd	1				0.0000	100.00%	0.0100	23.855
HTTP Requests by HTTP Host	62				0.0021	100.00%	0.1600	9.024
www.buet.ac.bd	1				0.0000	1.61%	0.0100	23.855
103.94.135.200	1				0.0000	100.00%	0.0100	23.855
google.com	1				0.0000	1.61%	0.0100	9.912

Display filter: Apply

Copy Save as... Close

HTTP Analysis – Requests

Wireshark · Requests · Wiredump.pcap

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
HTTP Requests by HTTP Host	62				0.0021	100%	0.1600	9.024
www.buet.ac.bd	1				0.0000	1.61%	0.0100	23.855
/web/	1				0.0000	100.00%	0.0100	23.855
google.com	1				0.0000	1.61%	0.0100	9.912
/	1				0.0000	100.00%	0.0100	9.912
239.255.255.250:1900	60				0.0020	96.77%	0.1600	9.024
*	60				0.0020	100.00%	0.1600	9.024

Display filter: Apply

Copy Save as... Close

HTTP Analysis – Packet Counter

Wireshark · Packet Counter · Wiredump.pcap

Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
Total HTTP Packets	68				0.0023	100%	0.1700	9.990
Other HTTP Packets	0				0.0000	0.00%	-	-
HTTP Response Packets	6				0.0002	8.82%	0.0100	0.000
???: broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	0				0.0000	0.00%	-	-
3xx: Redirection	2				0.0001	33.33%	0.0100	9.990
301 Moved Permanently	2				0.0001	100.00%	0.0100	9.990
2xx: Success	4				0.0001	66.67%	0.0100	0.000
200 OK	4				0.0001	100.00%	0.0100	0.000
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	62				0.0021	91.18%	0.1600	9.024
SEARCH	12				0.0004	19.35%	0.0100	1.912
NOTIFY	48				0.0016	77.42%	0.1600	9.024
GET	2				0.0001	3.23%	0.0100	9.912

Display filter: Apply

Copy Save as... Close

Improving WireShark Performance

- Don’t use capture filters
- Increase your read buffer size
- Get a faster computer
- Don’t resolve names

Post-Processing Text File

- For saved text-format packet files, further analysis needs coding or special tools
- One useful tool on Unix: Grep
  - On Windows: PowerGrep http://www.powergrep.com/
  - Command-line based utility for searching plain-text data sets for lines matching a regular expression

Basic usage of Grep

- Command-line text-search program in Linux
- Some useful usage:
  - Grep ‘word’ filename # find lines with ‘word’
  - Grep –v ‘word’ filename # find lines without ‘word’
  - Grep ‘^word’ filename # find lines beginning with ‘word’
  - Grep ‘word’ filename > file2 # output lines with ‘word’ to file2
  - ls -l | grep rwxrwxrwx # list files that have ‘rwxrwxrwx’ feature
  - grep '^[0-4]‘ filename # find lines beginning with any of the numbers from 0-4
  - Grep –c ‘word’ filename # find lines with ‘word’ and print out the number of these lines
  - Grep –i ‘word’ filename # find lines with ‘word’ regardless of case
- Many tutorials on grep online
  - http://www.cyberciti.biz/faq/howto-use-grep-command-in-linux-unix/
  - http://www.thegeekstuff.com/2009/03/15-practical-unix-grep-command-examples/

On-line Wireshark Trace Files

- Public available .pcap files:
  - http://www.netresec.com/?page=PcapFiles

- http://www.tp.org/jay/nwanalysis/traces/Lab%20Trace%20Files/

- Wiki Sample capture
  - https://wiki.wireshark.org/SampleCaptures

Example Trace File and Questions

- Network Forensic Puzzle Contests
  - http://forensicscontest.com/2010/02/03/puzzle-4-the-curious-mr-x

- SharkFest'15 Packet Challenge
  - https://sharkfest.wireshark.org/assets/presentations15/packetchallenge.zip

Source of Installer

- https://www.wireshark.org/download.html

Thank You

Acknowledgement: Web sources