Traffic Analysis– Wireshark

CIS 6395, Incident Response Technologies

Fall 2016, Dr. Cliff Zou

czou@cs.ucf.edu

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 to just the packets of interest
- Wireshark
  - GUI for displaying tcpdump/tshark packet traces

Tcpdump example

- Ran tcpdump on a Unix machine
- 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
```

Filters

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

- How to write filters?
  - Refer the tcpdump/tshark man page
  - Many example webpages on the Internet

Example

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

Example (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"

Example (contd.)

1. Capture only packets destined to longwood.eecs.ucf.edu
   - tcpdump "dst host longwood.eecs.ucf.edu"
2. Capture both DNS packets and TCP packets to/from longwood.eecs.ucf.edu
   - tcpdump "(tcp and host longwood.eecs.ucf.edu) or udp port 53"

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

So What is WireShark?

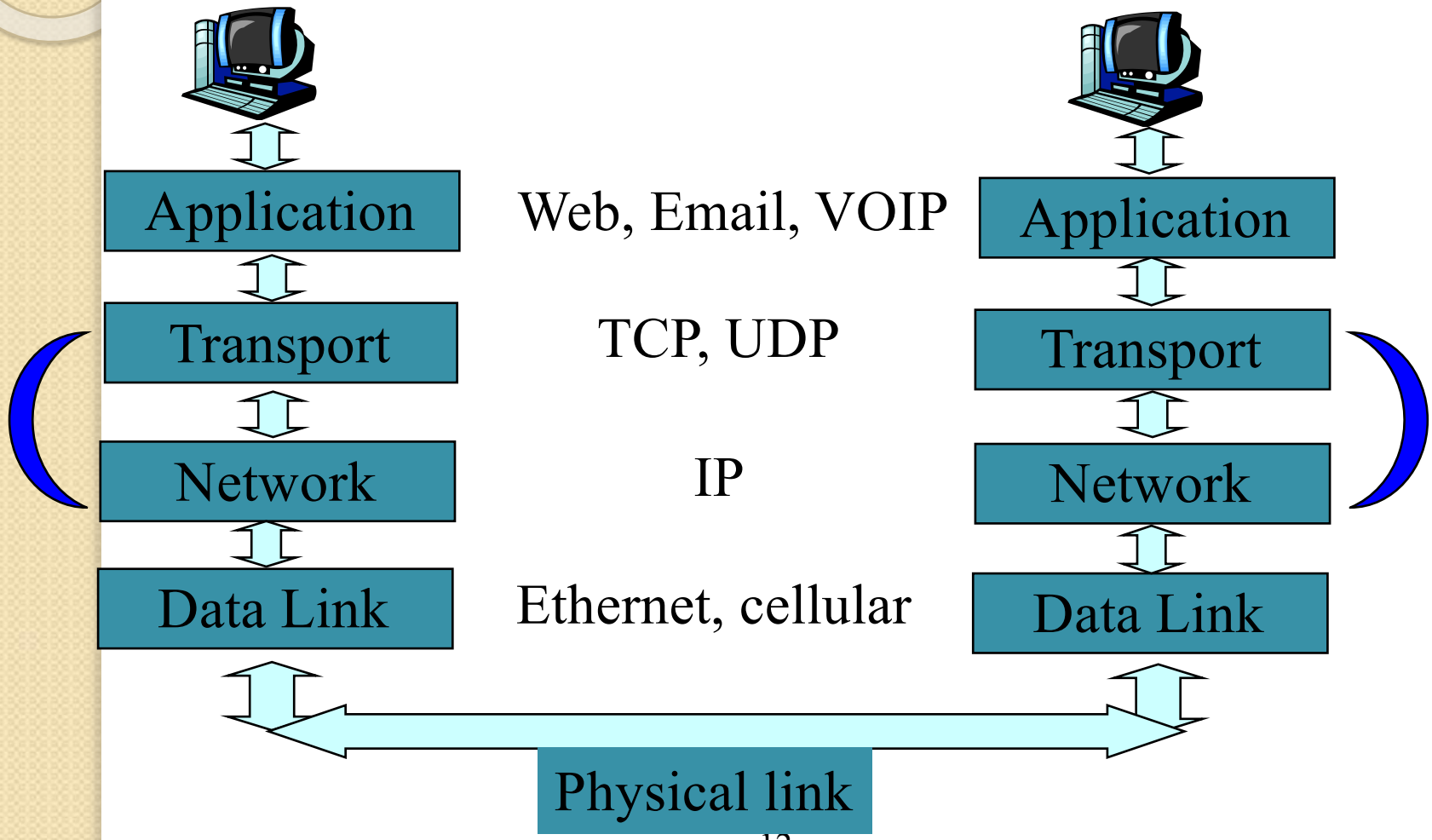
- Packet sniffer/protocol analyzer
- Open Source Network Tool
- Latest version of the ethereal tool

What is tShark?

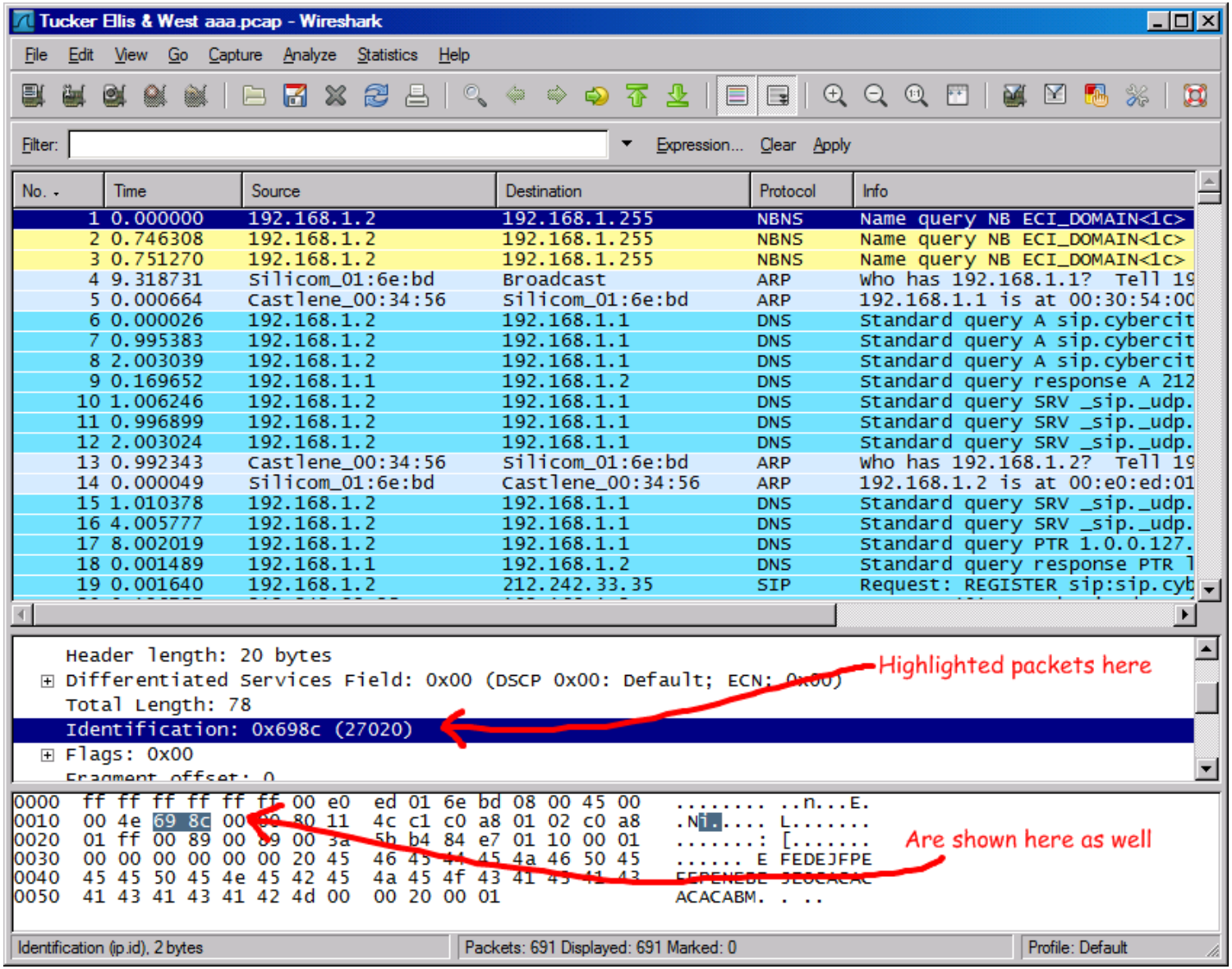
- The command-line based packet capture tool
- Equivalent to Wireshark

Network Layered Structure

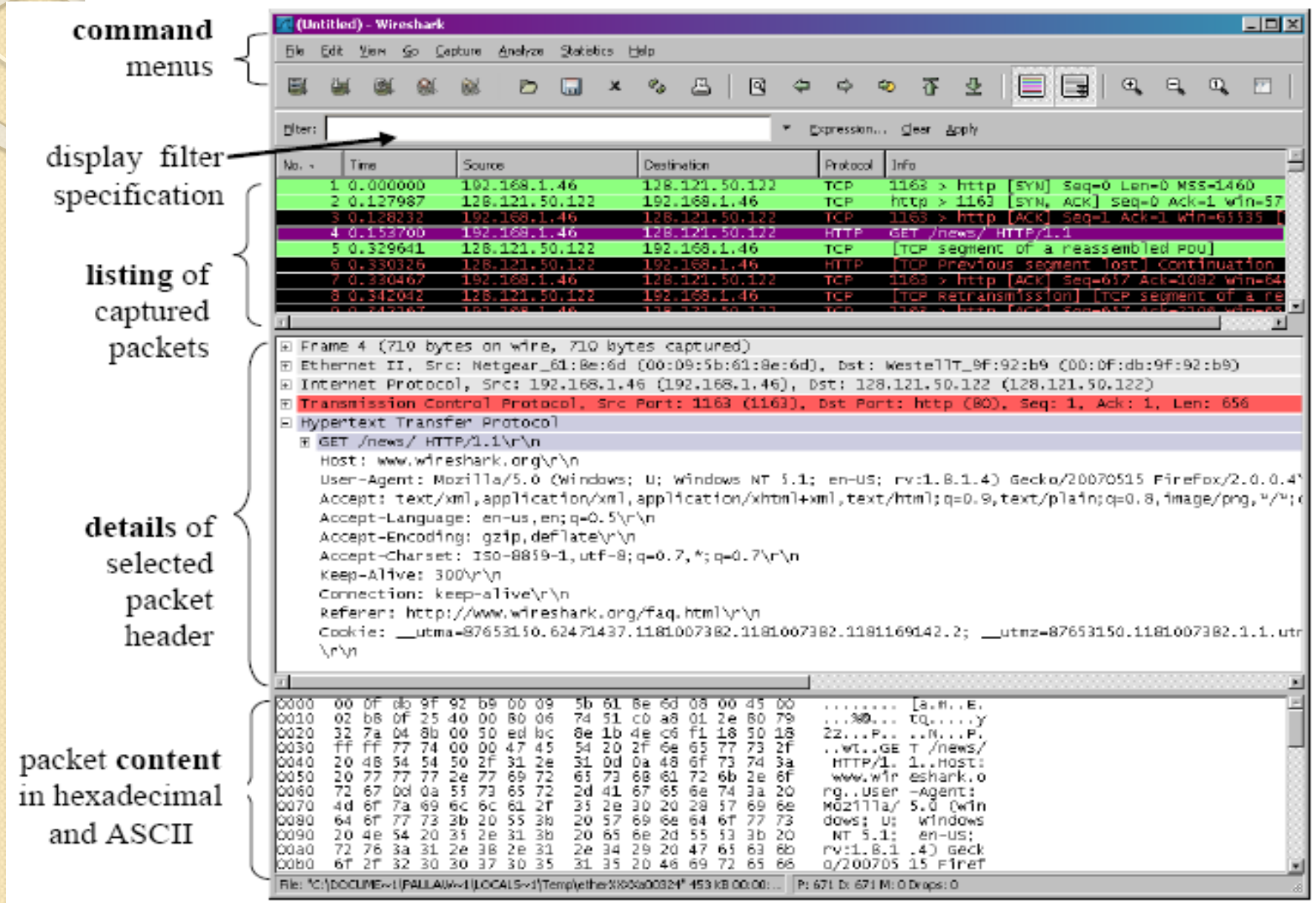
- What is the Internet?



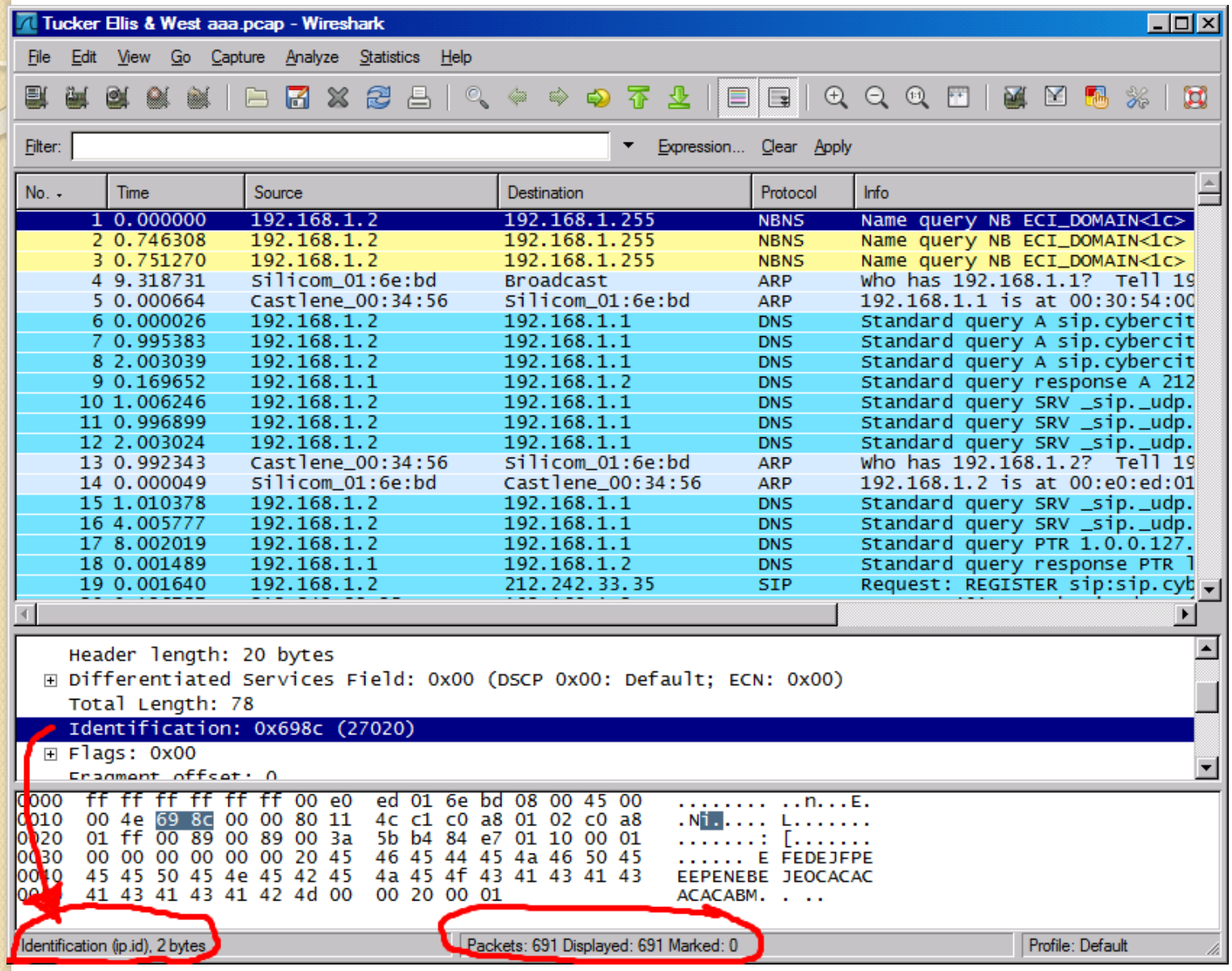
Wireshark Interface



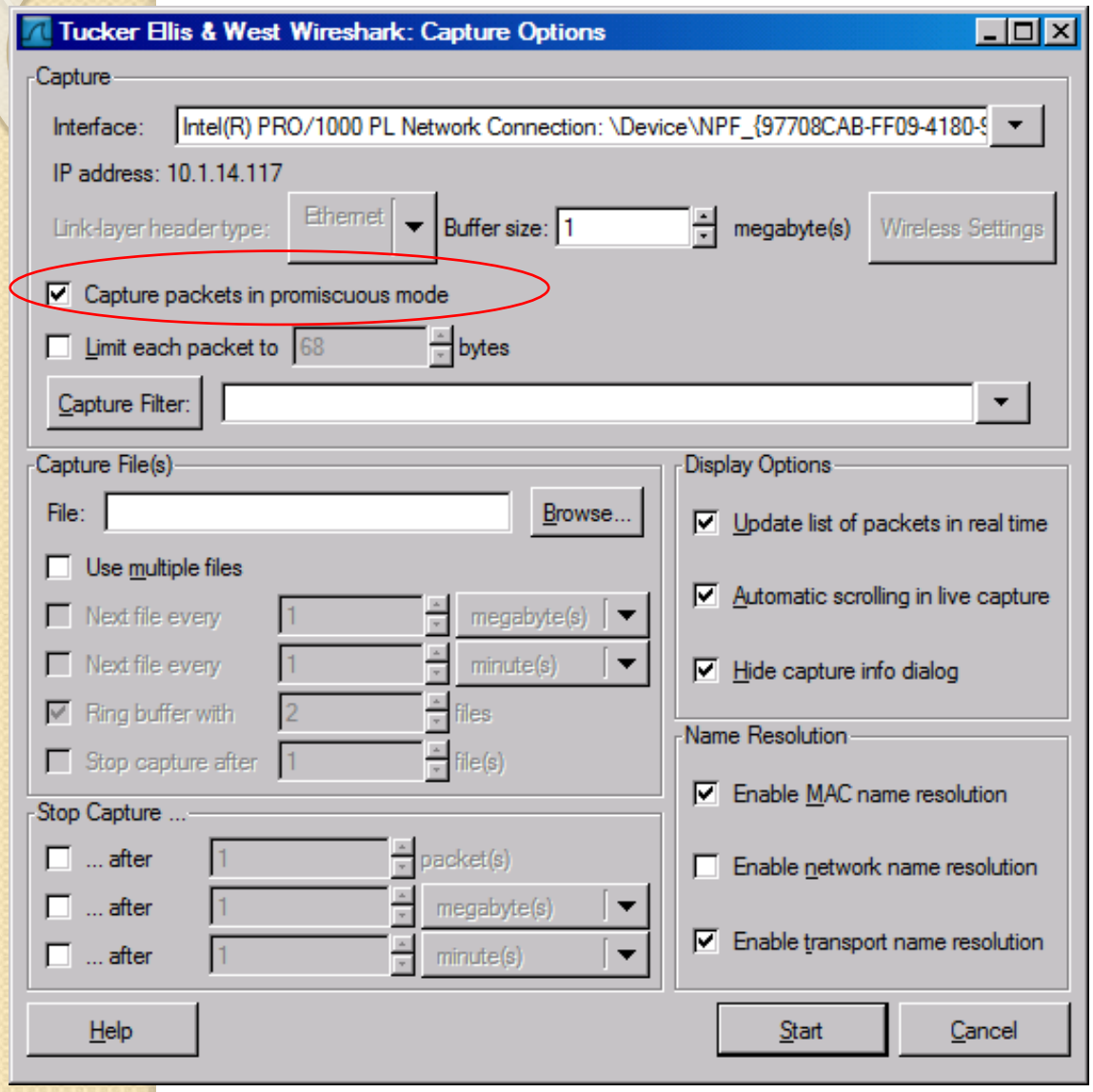
Wireshark Interface



Status Bar


Tucker Ellis & West aaa.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: Expression... Clear Apply
No. Time Source Destination Protocol Info
1 0.000000 192.168.1.2 192.168.1.255 NBNS Name query NB ECI_DOMAIN<1c>
2 0.746308 192.168.1.2 192.168.1.255 NBNS Name query NB ECI_DOMAIN<1c>
3 0.751270 192.168.1.2 192.168.1.255 NBNS Name query NB ECI_DOMAIN<1c>
4 9.318731 Silicom_01:6e:bd Broadcast ARP Who has 192.168.1.1? Tell 19
5 0.000664 Castlene_00:34:56 Silicom_01:6e:bd ARP 192.168.1.1 is at 00:30:54:00
6 0.000026 192.168.1.2 192.168.1.1 DNS Standard query A sip.cybercit
7 0.995383 192.168.1.2 192.168.1.1 DNS Standard query A sip.cybercit
8 2.003039 192.168.1.2 192.168.1.1 DNS Standard query A sip.cybercit
9 0.169652 192.168.1.1 192.168.1.2 DNS Standard query response A 212
10 1.006246 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
11 0.996899 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
12 2.003024 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
13 0.992343 Castlene_00:34:56 Silicom_01:6e:bd ARP Who has 192.168.1.2? Tell 19
14 0.000049 Silicom_01:6e:bd Castlene_00:34:56 ARP 192.168.1.2 is at 00:e0:ed:01
15 1.010378 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
16 4.005777 192.168.1.2 192.168.1.1 DNS Standard query SRV _sip._udp.
17 8.002019 192.168.1.2 192.168.1.1 DNS Standard query PTR 1.0.0.127.
18 0.001489 192.168.1.1 192.168.1.2 DNS Standard query response PTR l
19 0.001640 192.168.1.2 212.242.33.35 SIP Request: REGISTER sip:sip.cyb
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
Total Length: 78
Identification: 0x698c (27020)
Flags: 0x00
0000 ff ff ff ff ff ff 00 e0 ed 01 6e bd 08 00 45 00 ....n...E.
0010 00 4e 69 8c 00 00 80 11 4c c1 c0 a8 01 02 c0 a8 .Ni..... L.....
0020 01 ff 00 89 00 89 00 3a 5b b4 84 e7 01 10 00 01 ..... : [.....
0030 00 00 00 00 00 00 20 45 46 45 44 45 4a 46 50 45 ..... E FEDEJFPE
0040 45 45 50 45 4e 45 42 45 4a 45 4f 43 41 43 41 43 EEPENEBE JEOCACAC
0050 41 43 41 43 41 42 4d 00 00 20 00 01 ACACABM. . ..
Identification (ip.id), 2 bytes
Packets: 691 Displayed: 691 Marked: 0
Profile: Default

Capture Options



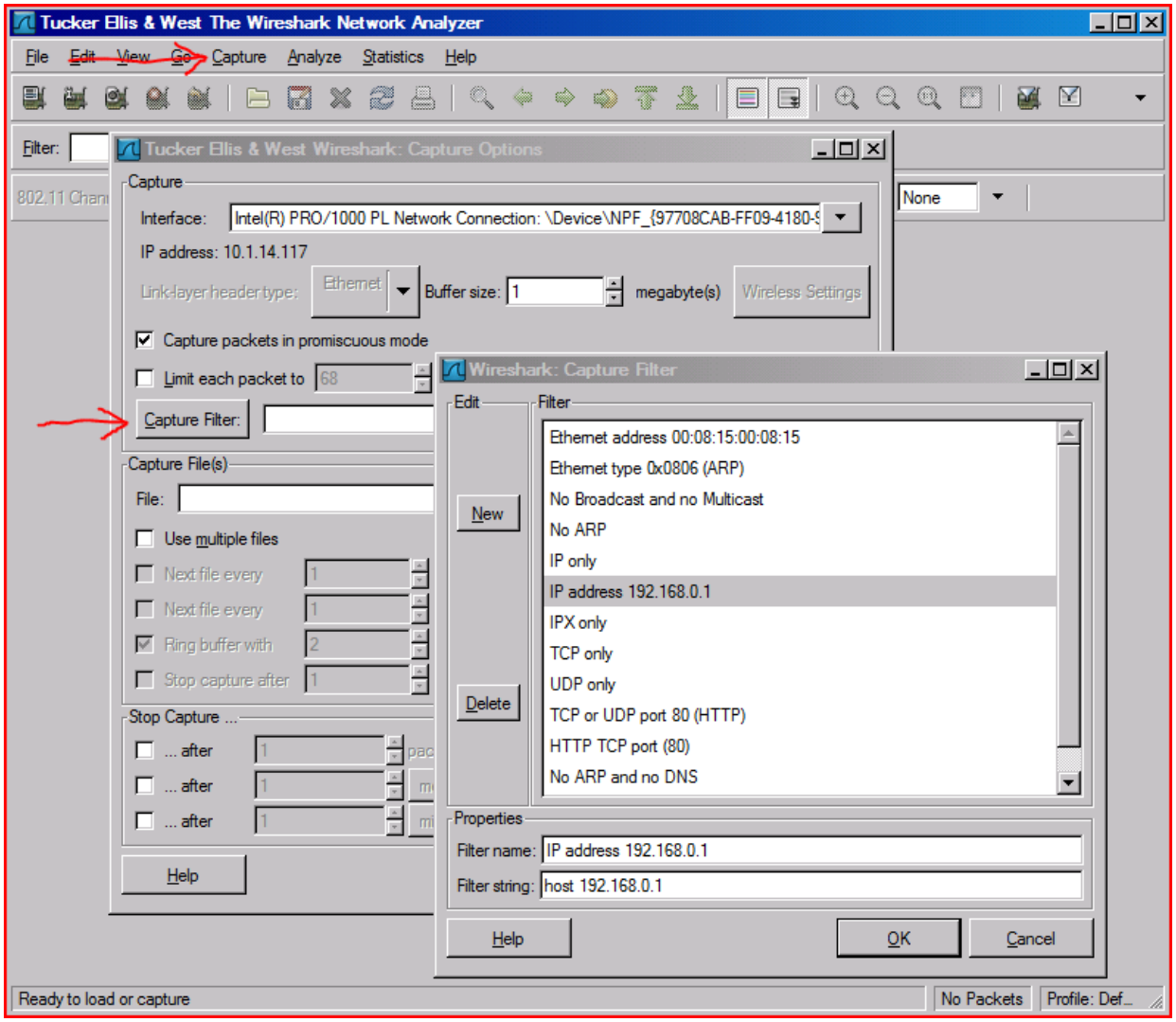
Promiscuous mode is used to Capture all traffic

In many cases this does not work:

- Network driver does not support
- You are on a switch LAN

Capture Filter

There are some pre-built capture filters that you can use:



Capture Filter examples

host 10.1.11.24

host 192.168.0.1 and host 10.1.11.1

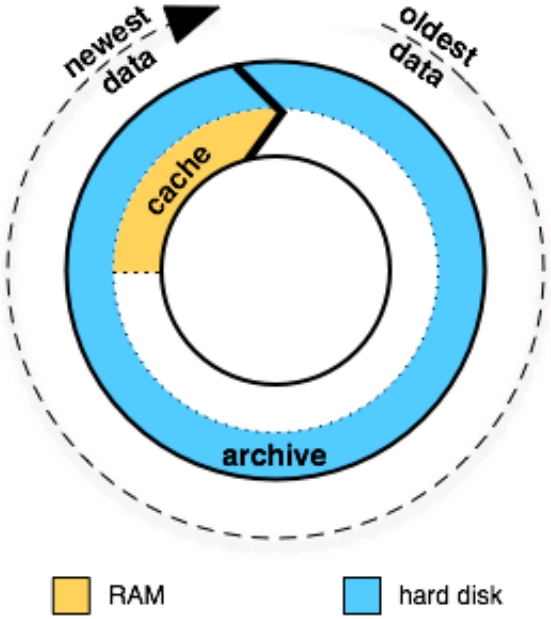
tcp port http

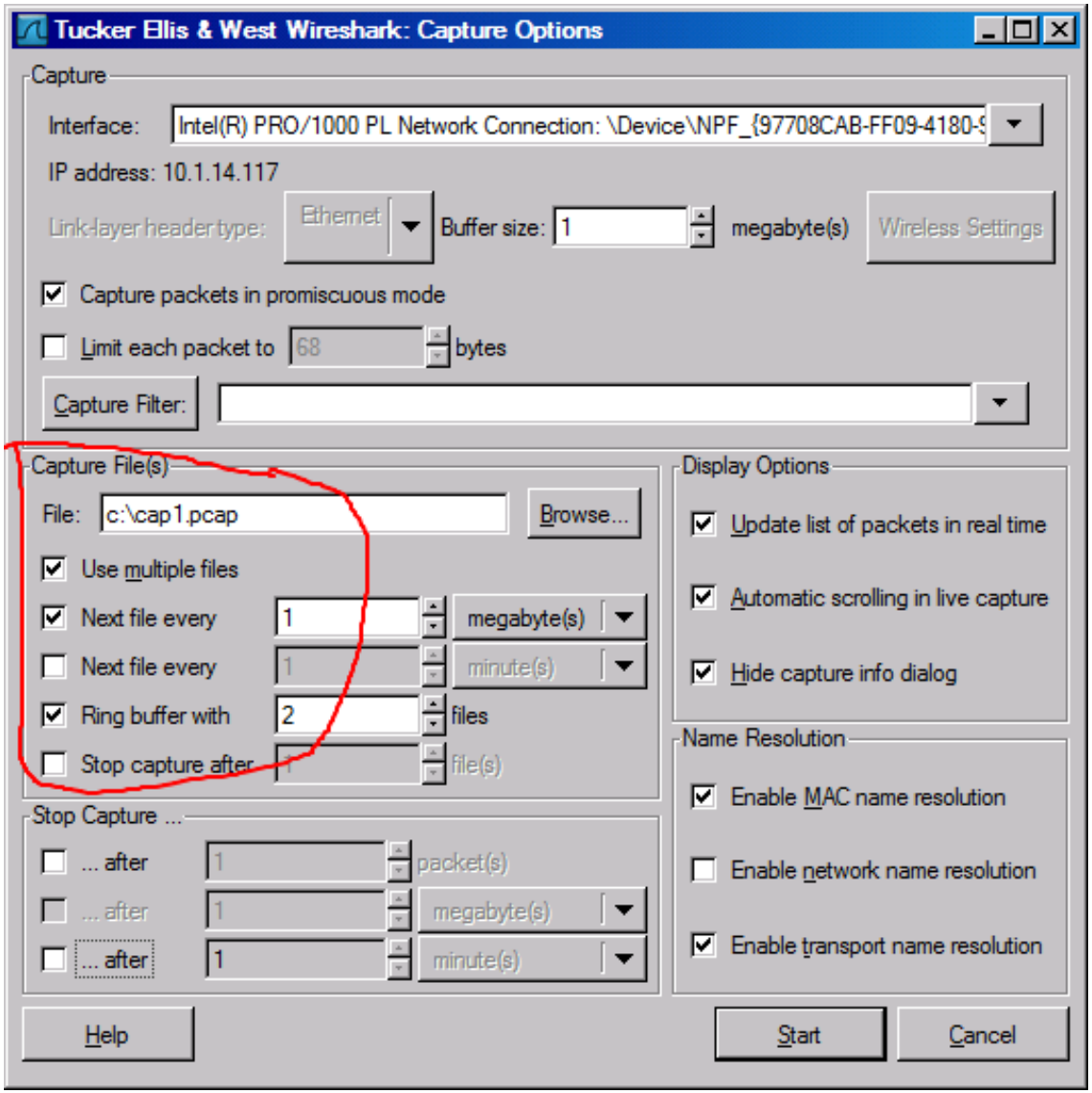
ip

not broadcast not multicast

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

Capture Buffer Usage




Tucker Ellis & West Wireshark: Capture Options
Capture
Interface:
Intel(R) PRO/1000 PL Network Connection: \Device\NPF_{97708CAB-FF09-4180-9
IP address: 10.1.14.117
Link-layer header type:
Ethernet
Buffer size:
1
megabyte(s)
Wireless Settings
Capture packets in promiscuous mode
Limit each packet to
68
bytes
Capture Filter:
Capture File(s)
File:
c:\cap1.pcap
Browse...
Use multiple files
Next file every
1
megabyte(s)
Next file every
1
minute(s)
Ring buffer with
2
files
Stop capture after
1
file(s)
Display Options
Update list of packets in real time
Automatic scrolling in live capture
Hide capture info dialog
Name Resolution
Enable MAC name resolution
Enable network name resolution
Enable transport name resolution
Stop Capture ...
... after
1
packet(s)
... after
1
megabyte(s)
... after
1
minute(s)
Help
Start
Cancel

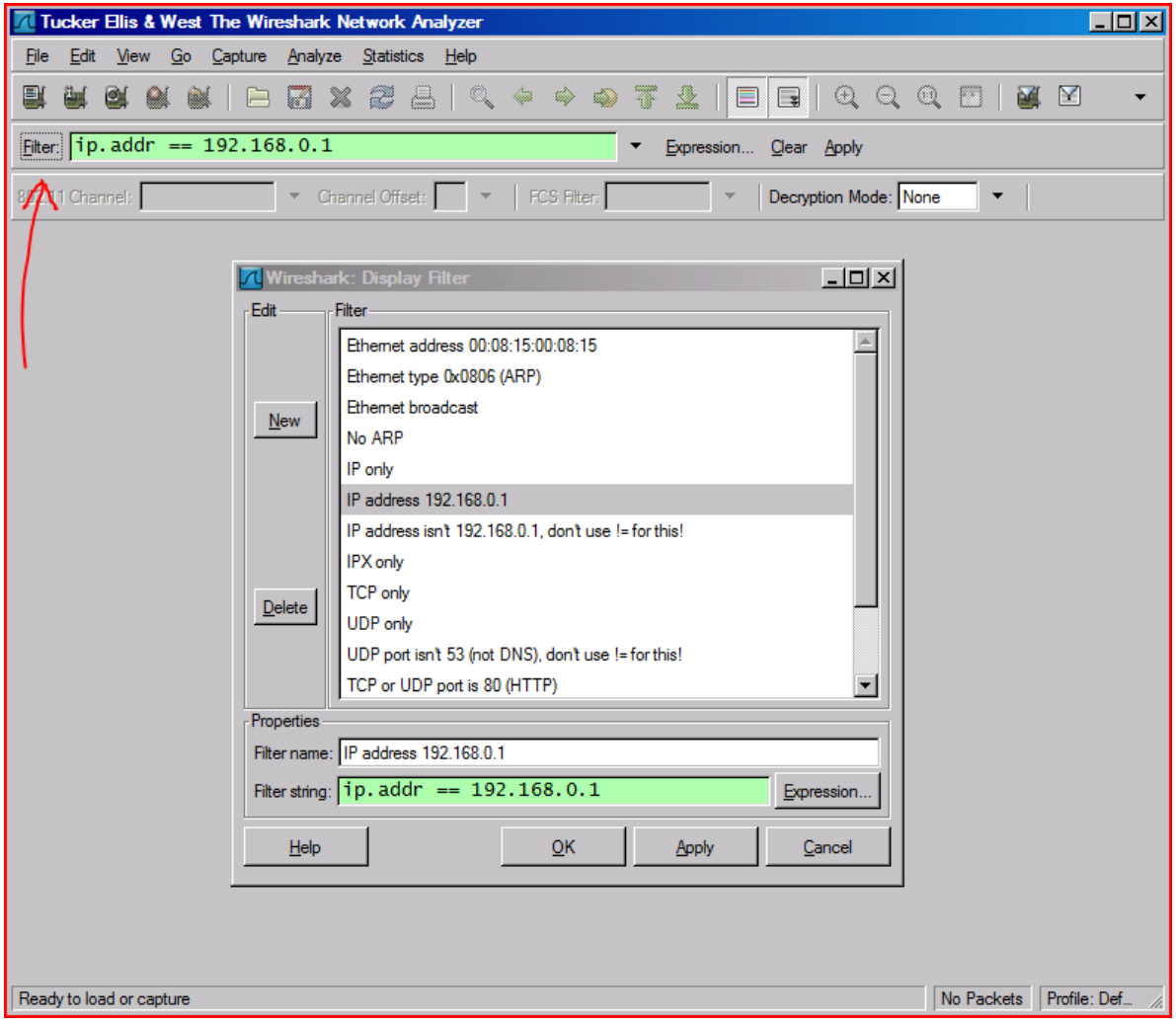
Display Filters (Post-Filters)

- Display filters (also called post-filters)
  - Only filter the view of what you are seeing
  - 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

There are some basic pre-built display filters, too



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)

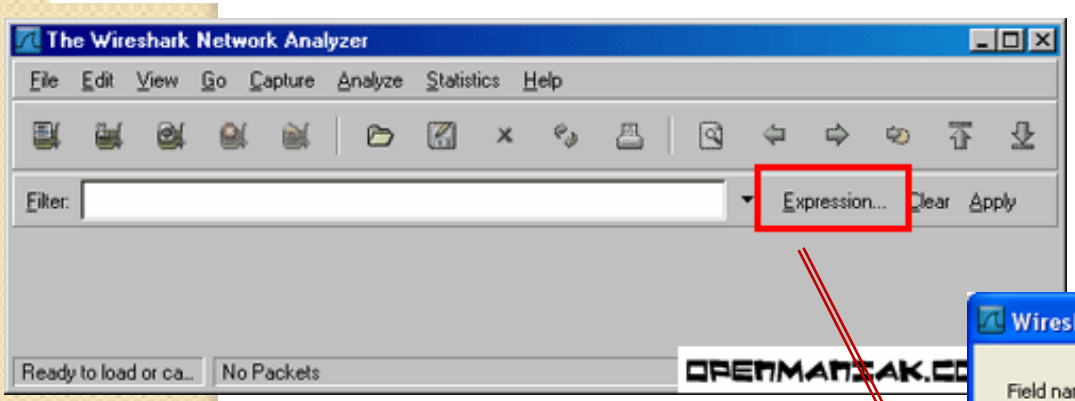
(ip.addr==192.168.1.10 && ip.addr==192.168.1.20) && (tcp.port==445 || tcp.port==139)

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

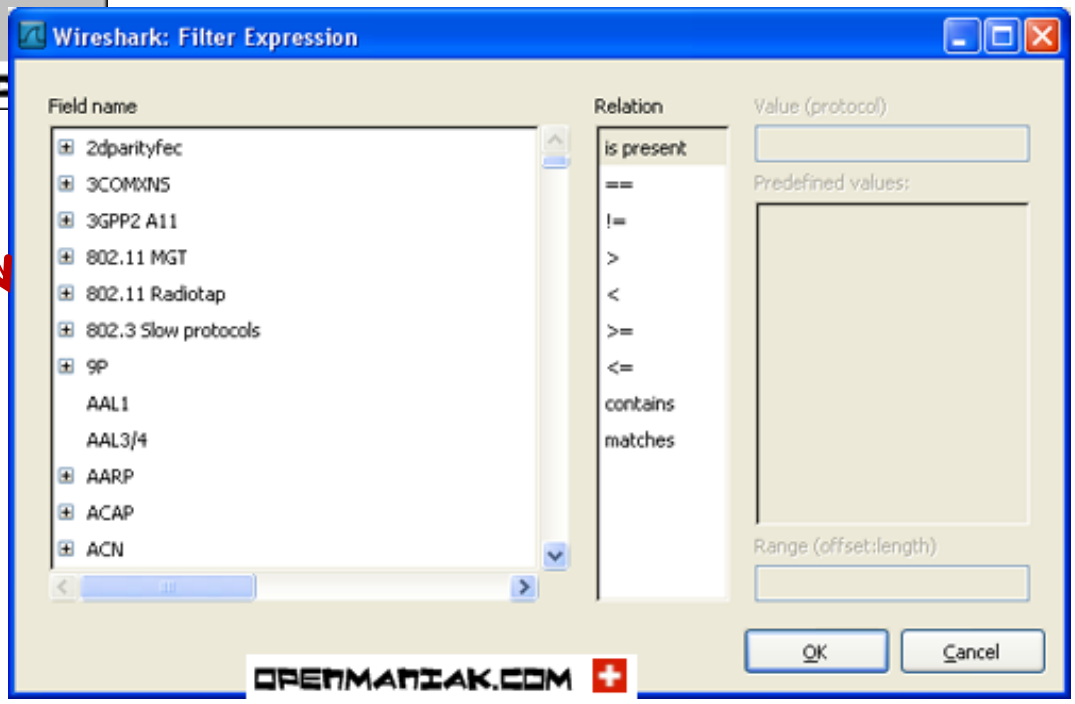
tcp.dstport == 80

Display Filter

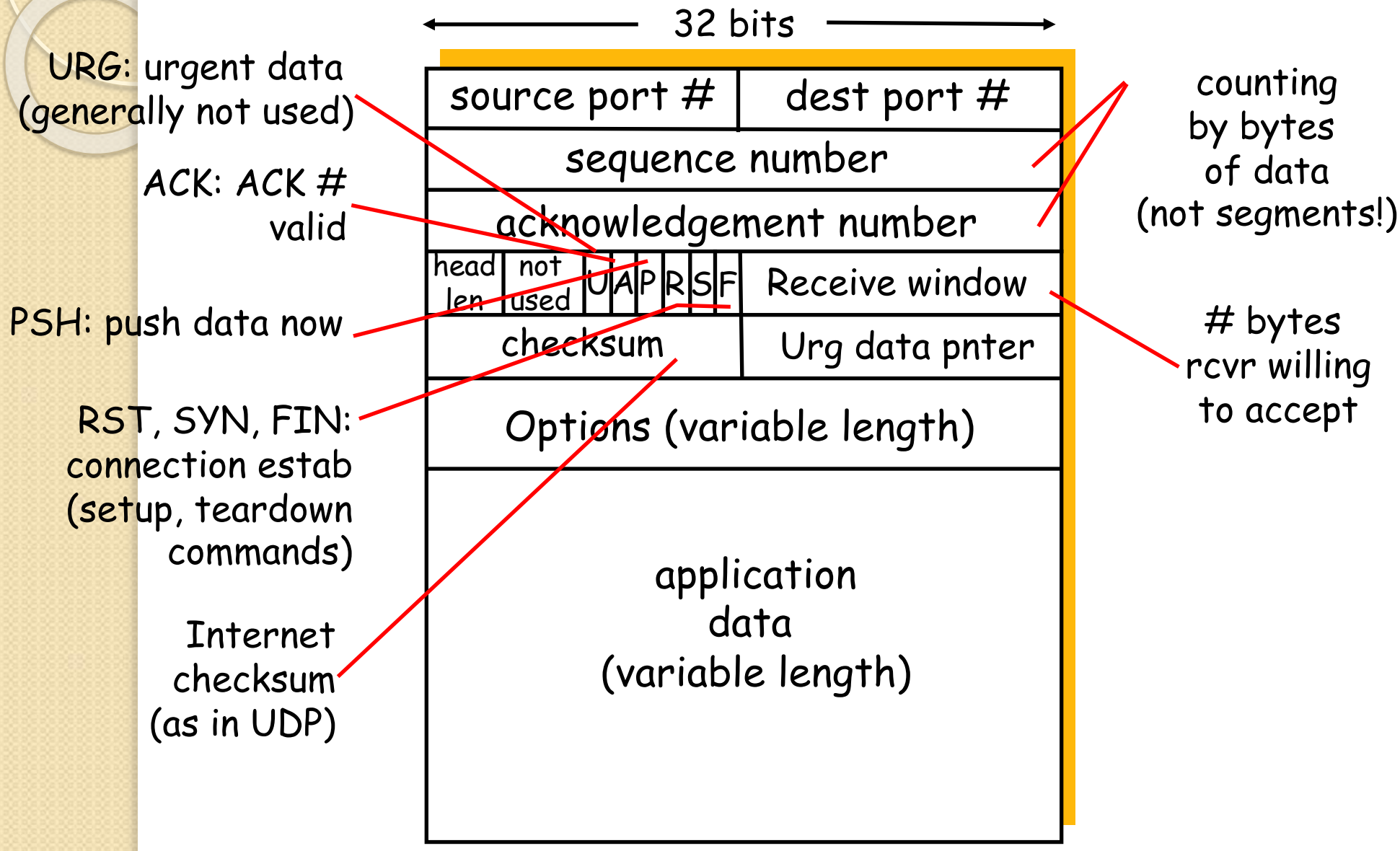
Syntax:	Protocol	.	String 1	.	String 2	Comparison operator	Value	Logical Operations	Other expression
Example:	ftp		passive		ip	==	10.2.3.4	xor	icmp.type



There are thousands of pre-defined protocol fields that
You can use in the display filter!



TCP segment structure

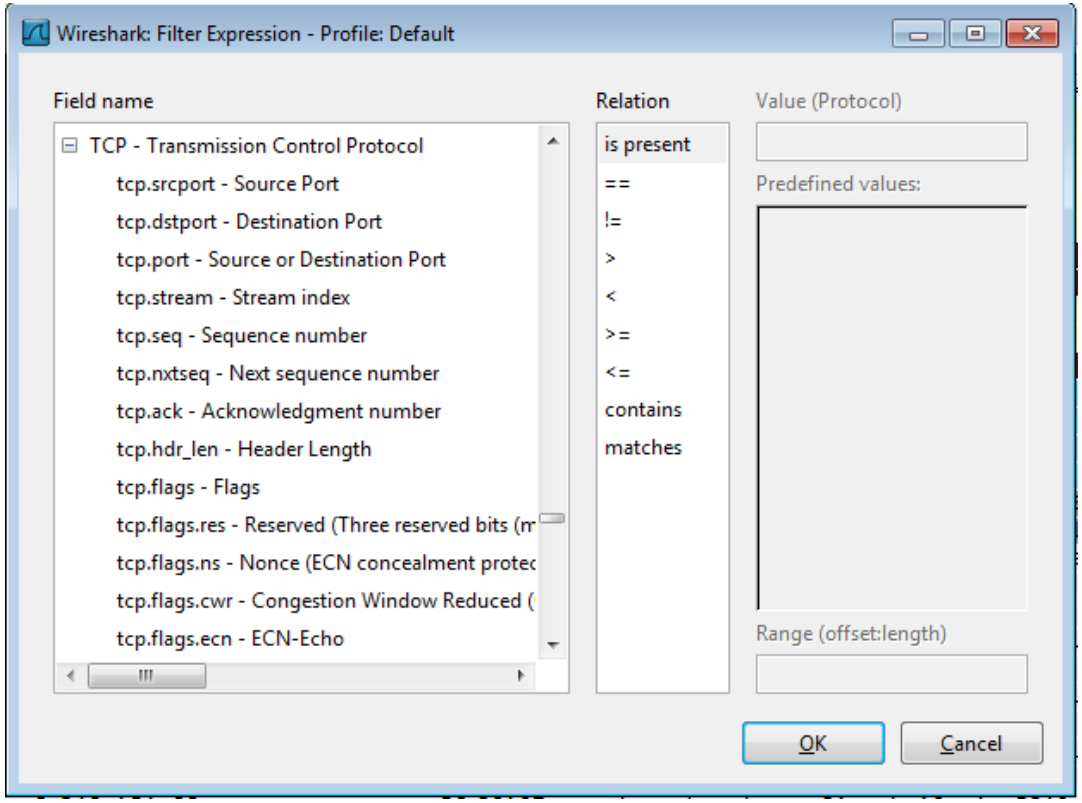


Display Filter

- 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**
    - **Or use "Tcp.flags.syn==1"**
  - **tcp.flags == 18**
    - **SYN/ACK**

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

  - **Note of TCP Flag field:**



Display Filter Expressions

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

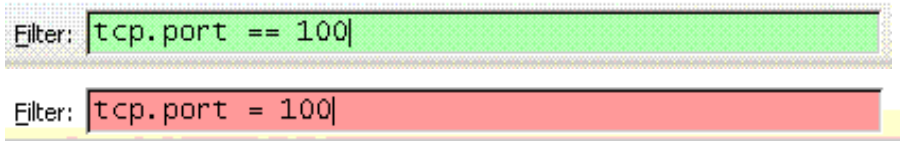
Six comparison operators are available:

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

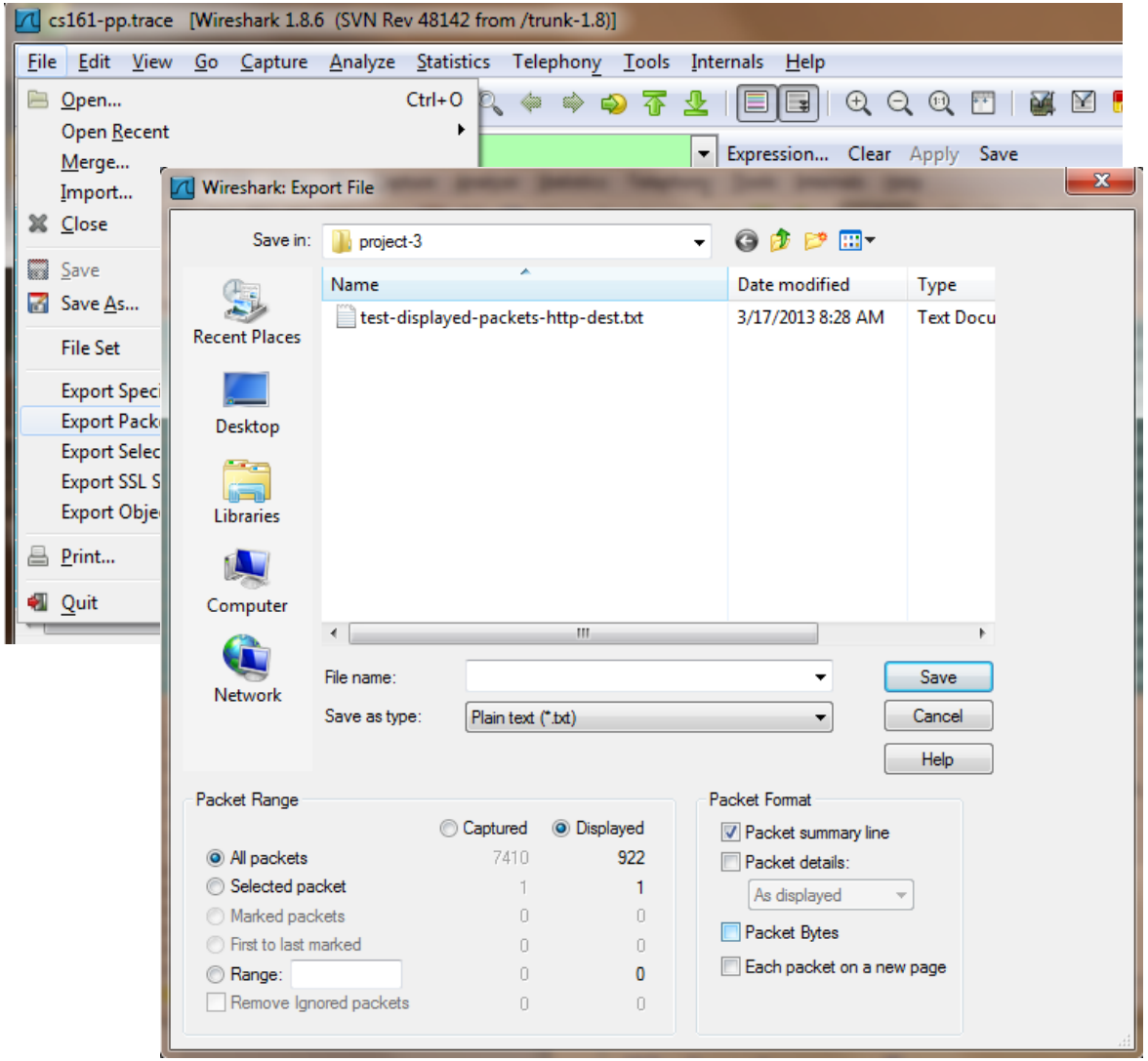
Save Filtered Packets as Text After Using Display Filter

- We can 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”

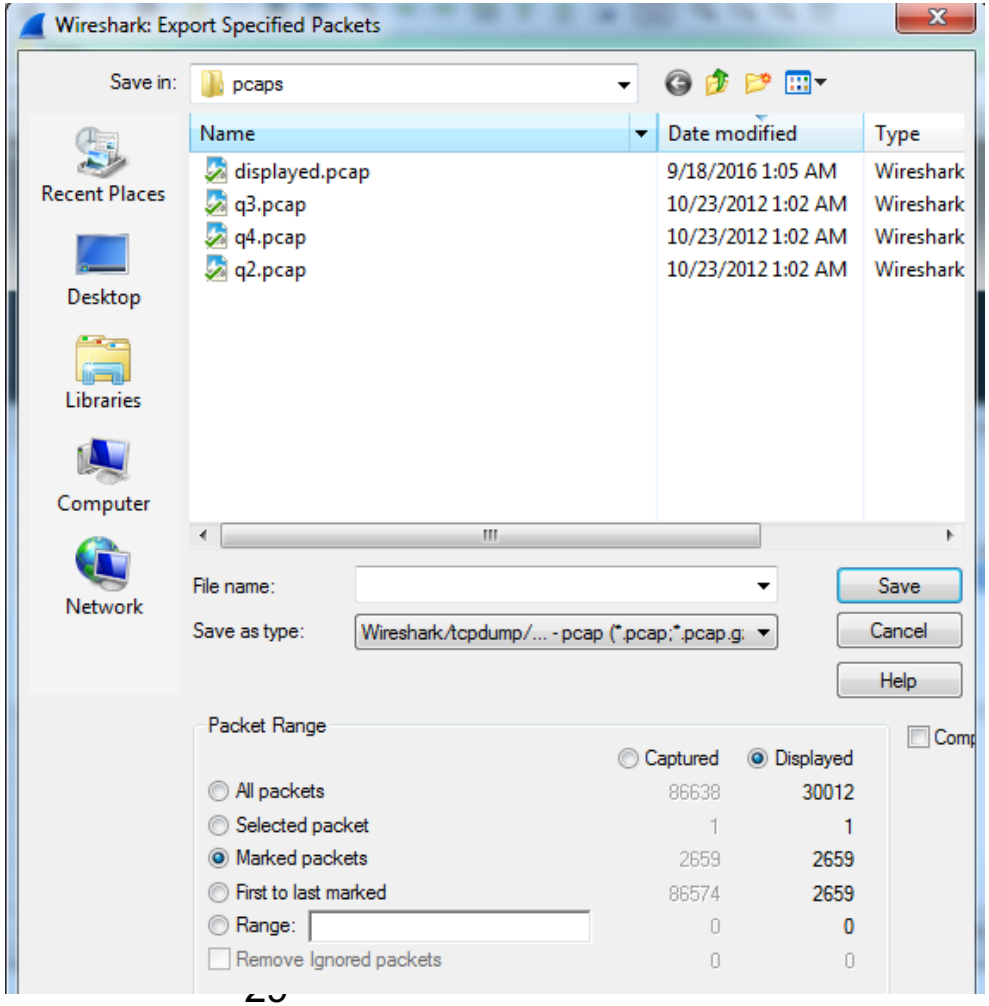
2). In choose “summary line” or “detail”



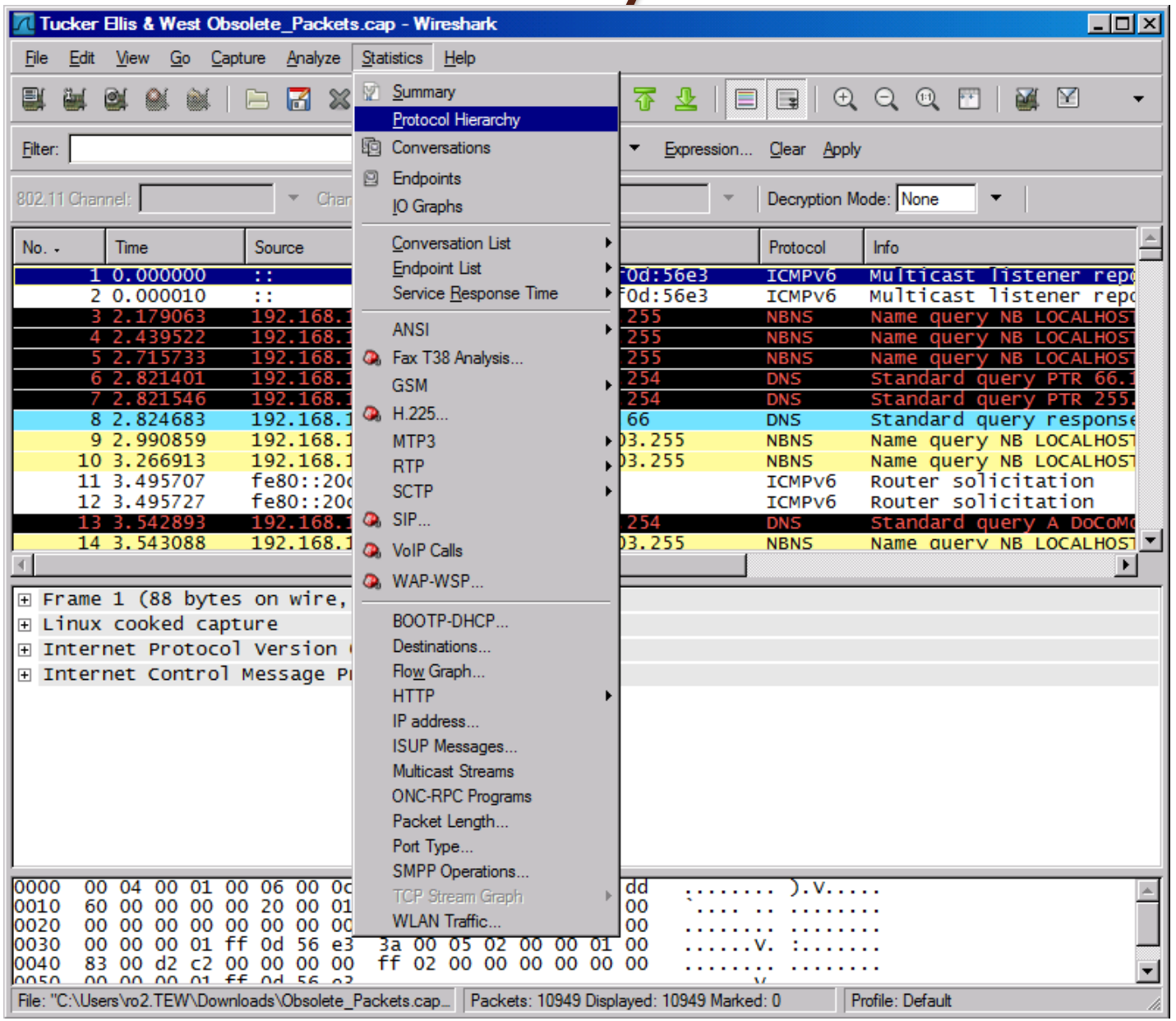
Save Filtered Packets in Wireshark format After Using Display Filter

- We can also save all filtered packets in the original wireshark format for further analysis
- Operation:

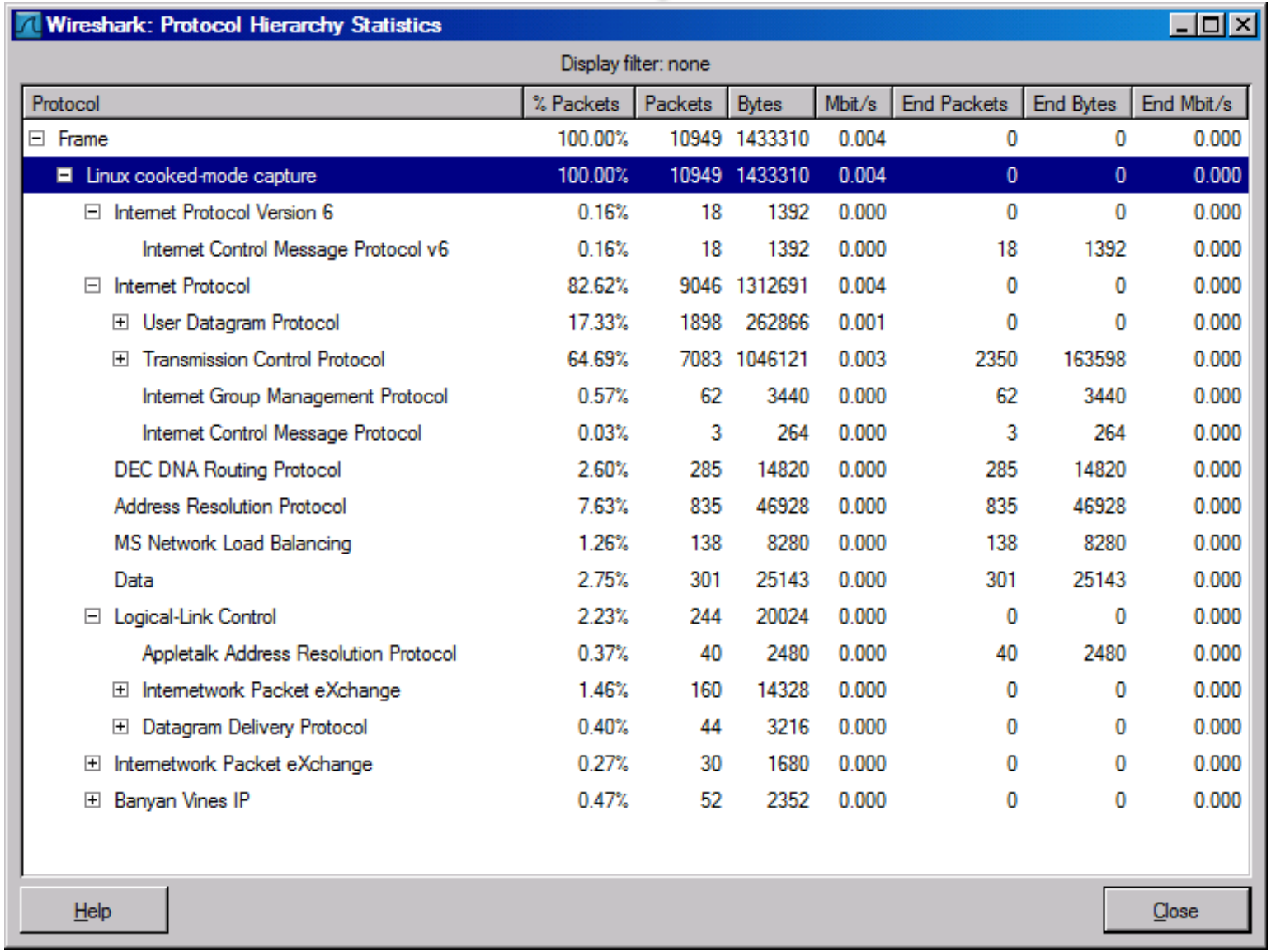
1. Enter Display filter to show packets you want
2. Go to "Edit>" and choose "Mark all displayed packets“
3. Go to “File” → Export specific packets…
4. Choose the option “Marked packets” to save the file



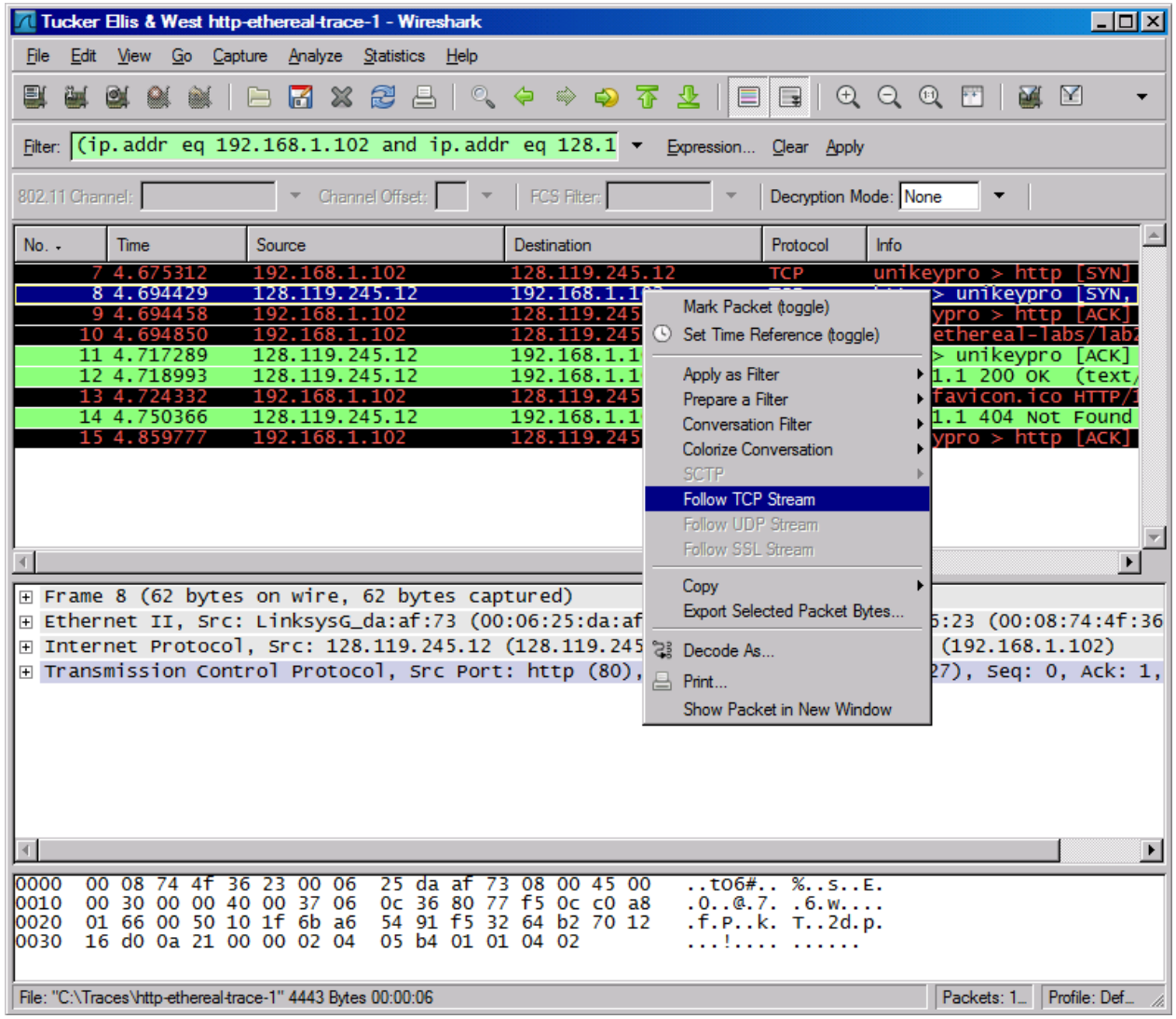
Protocol Hierarchy



Protocol Hierarchy

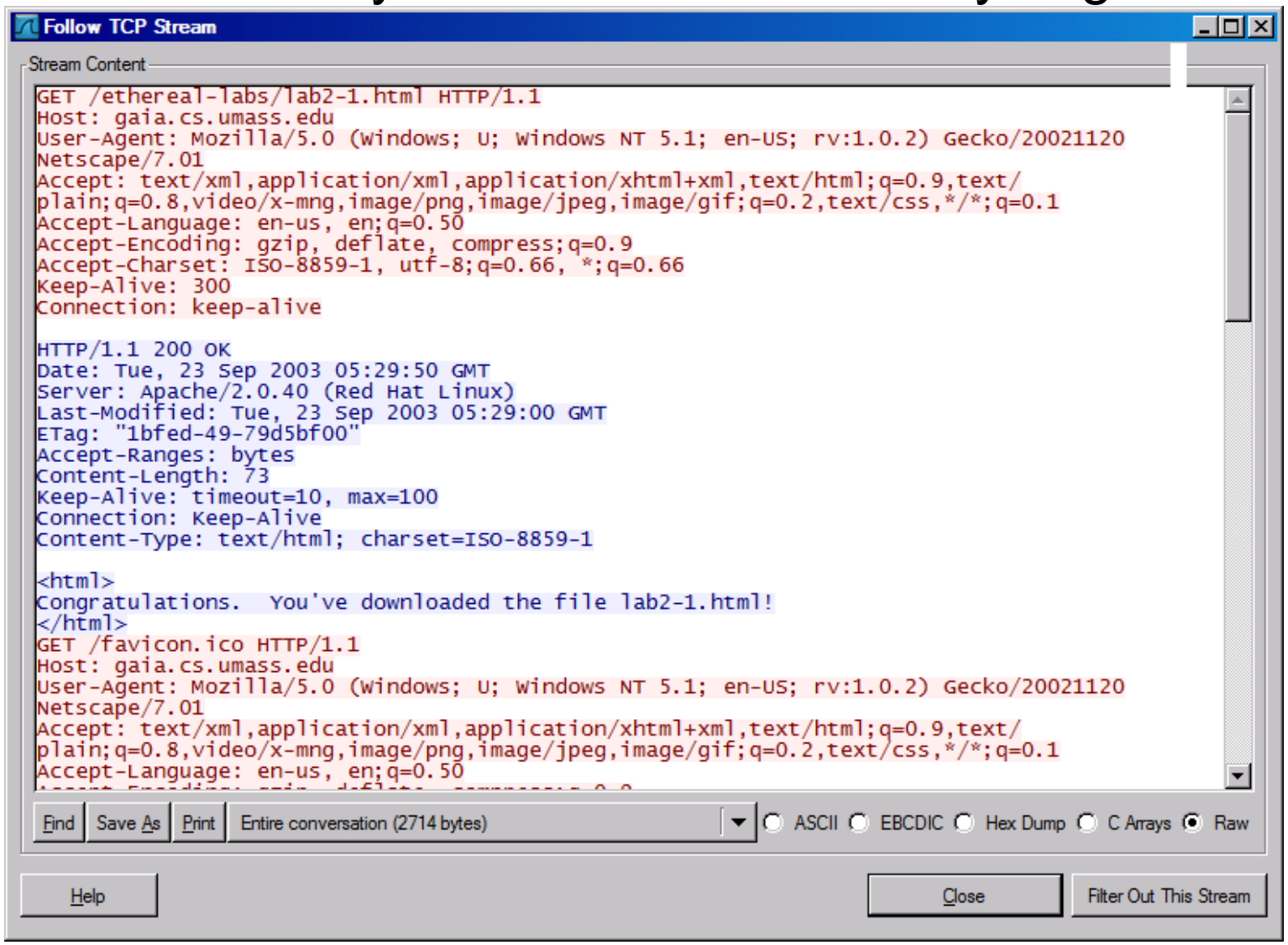


Follow TCP Stream



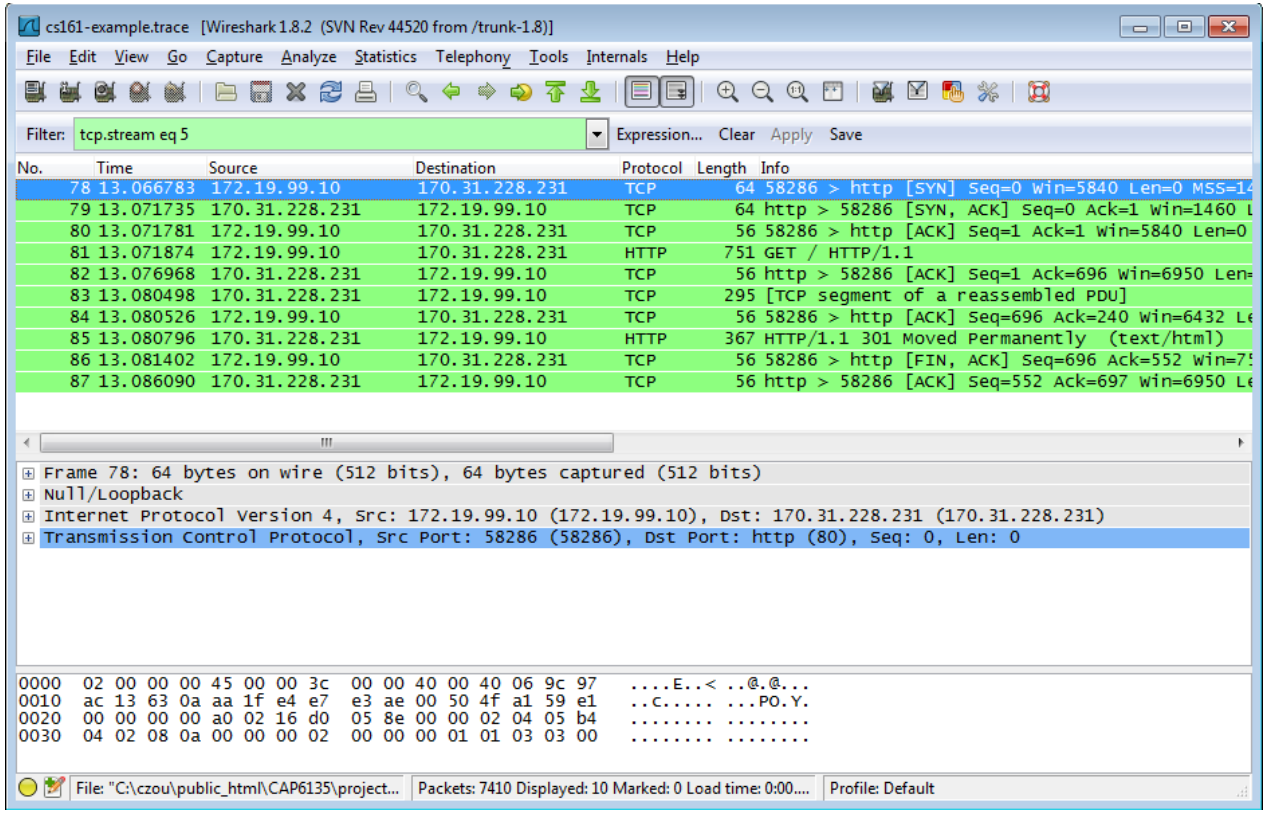
Follow TCP Stream

red - stuff you sent blue - stuff you get

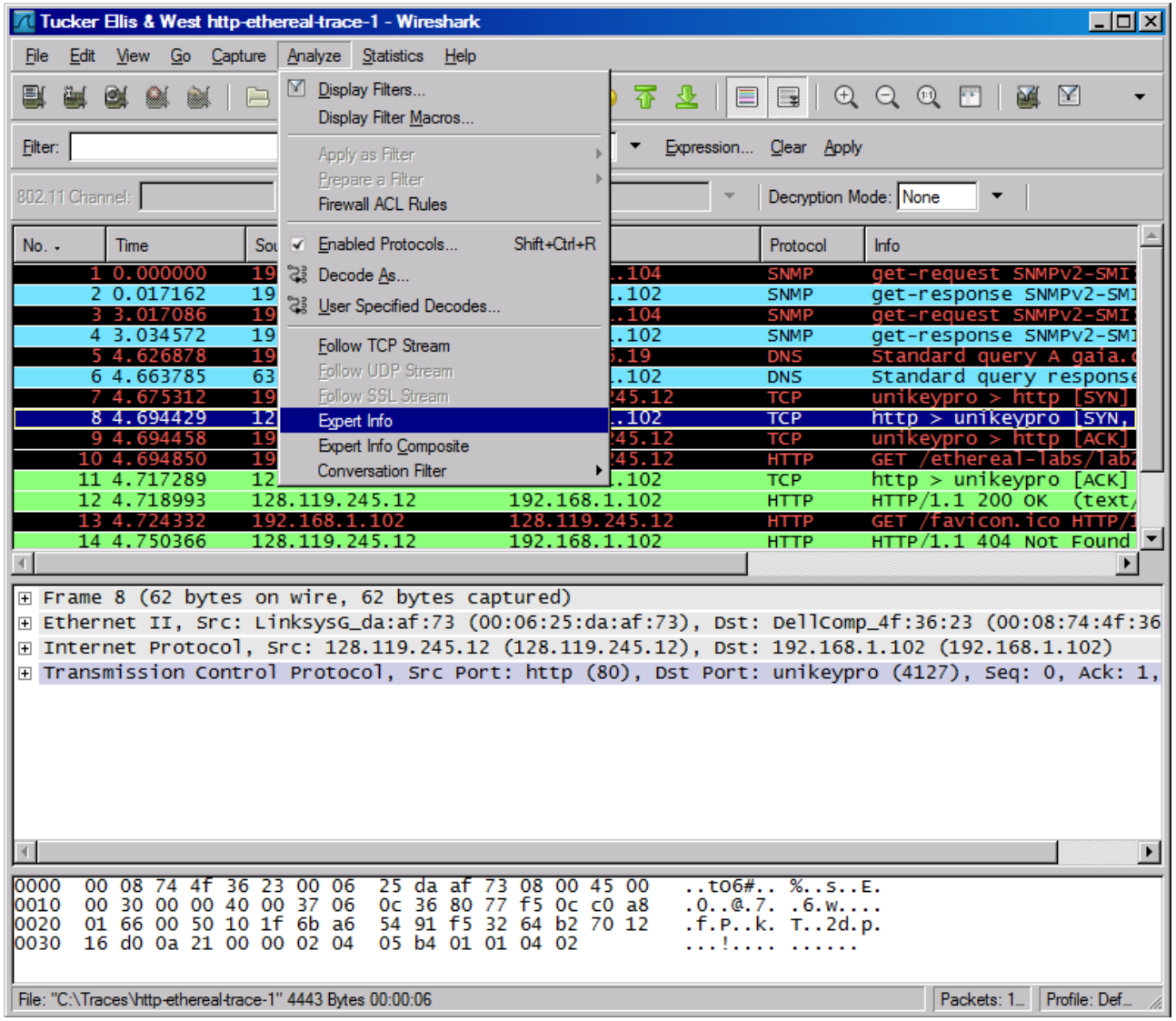


Filter out/in Single TCP Stream

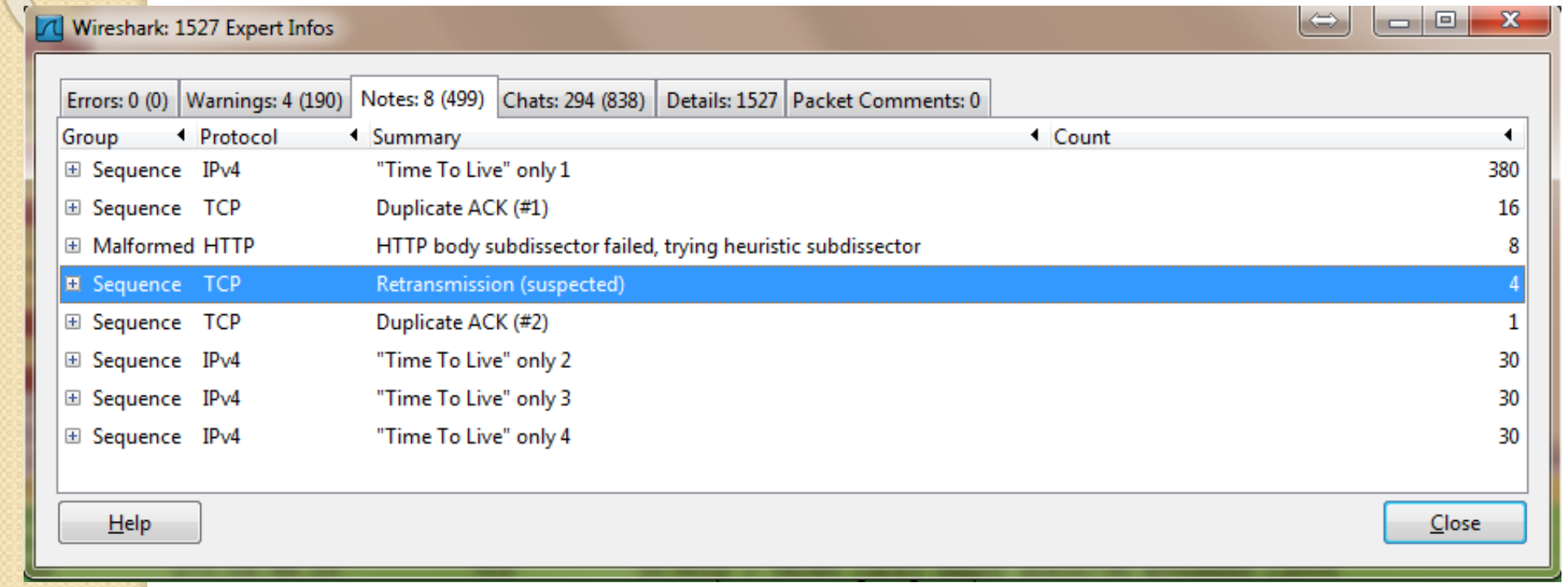
- When click "filter out this TCP stream" in previous page's box, new filter string will contain like:
  - http and !(tcp.stream eq 5)
- So, if you use "tcp.stream eq 5" as filter string, you keep this HTTP session



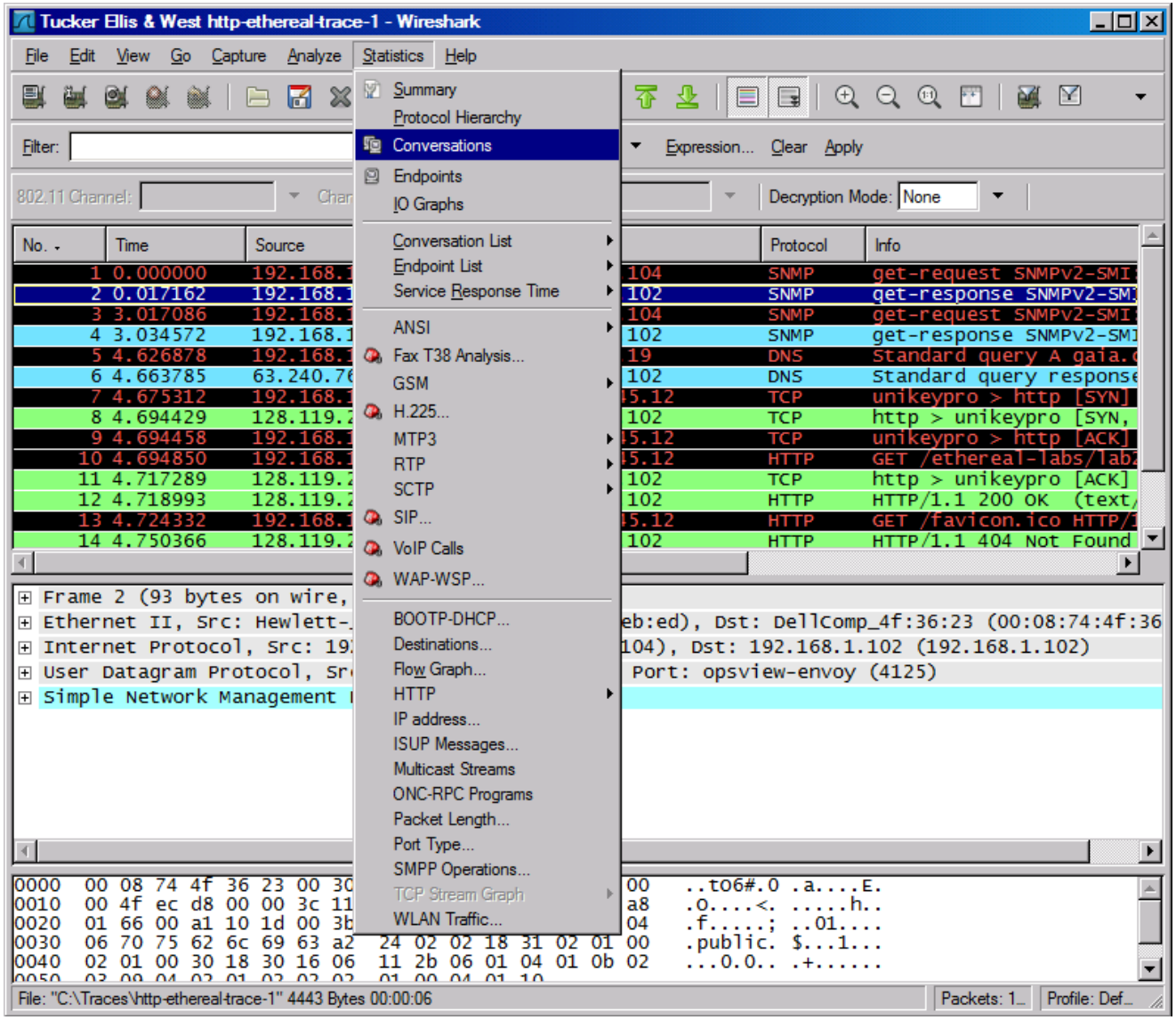
Expert Info


Tucker Ellis & West http-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Display Filters...
Display Filter Macros...
Apply as Filter
Prepare a Filter
Firewall ACL Rules
Enabled Protocols... Shift+Ctrl+R
Decode As...
User Specified Decodes...
Follow TCP Stream
Follow UDP Stream
Follow SSL Stream
Expert Info
Expert Info Composite
Conversation Filter
Filter:
Expression... Clear Apply
802.11 Channel:
Decryption Mode: None
No. Time Protocol Info
1 0.000000 SNMP get-request SNMPv2-SMI
2 0.017162 SNMP get-response SNMPv2-SMI
3 3.017086 SNMP get-request SNMPv2-SMI
4 3.034572 SNMP get-response SNMPv2-SMI
5 4.626878 DNS Standard query A gaia.
6 4.663785 DNS Standard query response
7 4.675312 TCP unikeypro > http [SYN]
8 4.694429 TCP http > unikeypro [SYN,
9 4.694458 TCP unikeypro > http [ACK]
10 4.694850 HTTP GET /ethereal-labs/lab
11 4.717289 TCP http > unikeypro [ACK]
12 4.718993 128.119.245.12 192.168.1.102 HTTP HTTP/1.1 200 OK (text/
13 4.724332 192.168.1.102 128.119.245.12 HTTP GET /favicon.ico HTTP/1
14 4.750366 128.119.245.12 192.168.1.102 HTTP HTTP/1.1 404 Not Found
Frame 8 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: DellComp_4f:36:23 (00:08:74:4f:36
Internet Protocol, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.1.102 (192.168.1.102)
Transmission Control Protocol, Src Port: http (80), Dst Port: unikeypro (4127), Seq: 0, Ack: 1,
0000 00 08 74 4f 36 23 00 06 25 da af 73 08 00 45 00 ..to6#.. %..s..E.
0010 00 30 00 00 40 00 37 06 0c 36 80 77 f5 0c c0 a8 .0..@.7. .6.w....
0020 01 66 00 50 10 1f 6b a6 54 91 f5 32 64 b2 70 12 .f.P..k. T..2d.p.
0030 16 d0 0a 21 00 00 02 04 05 b4 01 01 04 02 ...!.... ......
File: "C:\Traces\http-ethereal-trace-1" 4443 Bytes 00:00:06
Packets: 1...
Profile: Def...

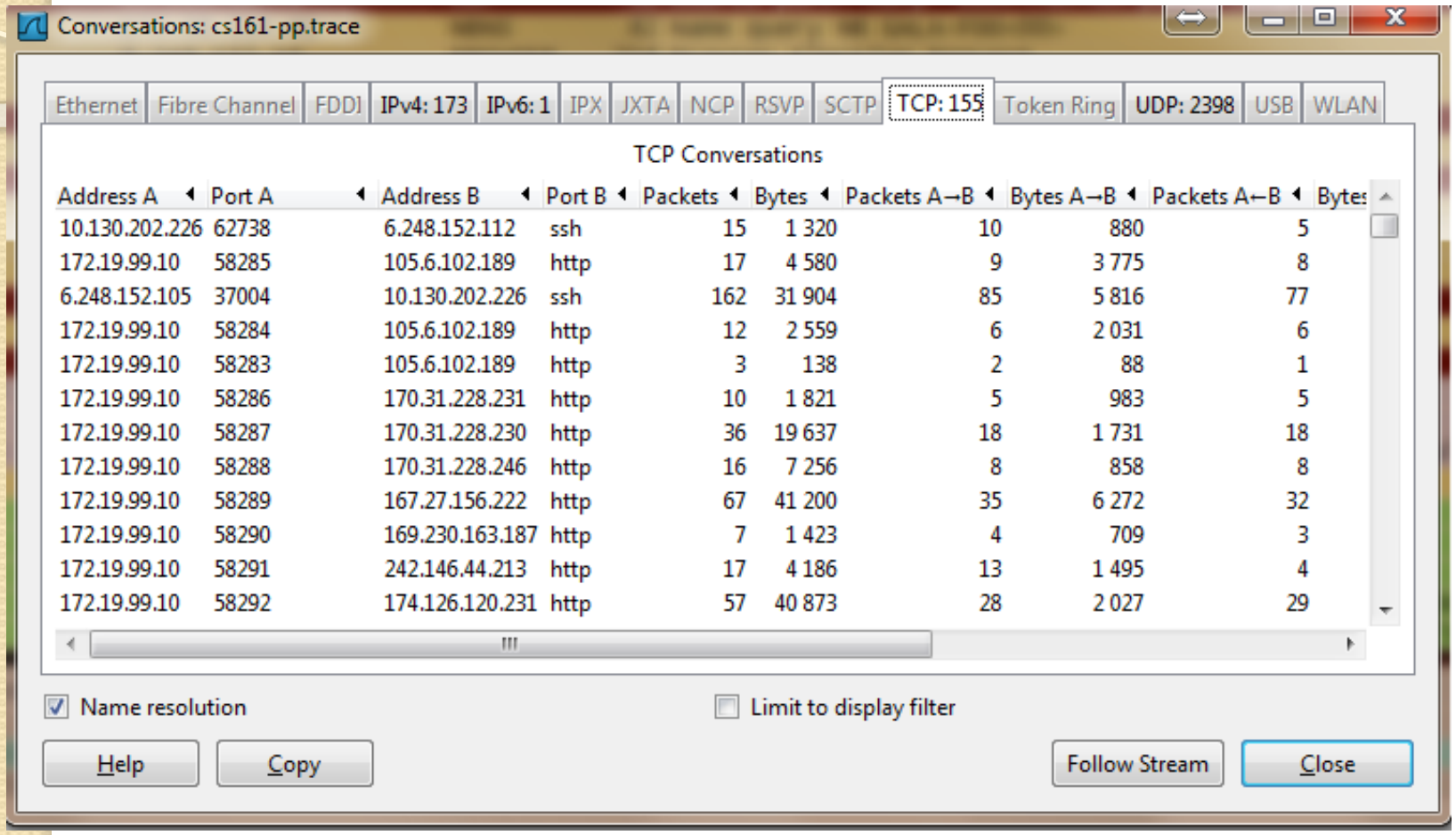
Expert Info


Wireshark: 1527 Expert Infos
Errors: 0 (0)
Warnings: 4 (190)
Notes: 8 (499)
Chats: 294 (838)
Details: 1527
Packet Comments: 0
Group
Protocol
Summary
Count
Sequence IPv4 "Time To Live" only 1 380
Sequence TCP Duplicate ACK (#1) 16
Malformed HTTP HTTP body subdissector failed, trying heuristic subdissector 8
Sequence TCP Retransmission (suspected) 4
Sequence TCP Duplicate ACK (#2) 1
Sequence IPv4 "Time To Live" only 2 30
Sequence IPv4 "Time To Live" only 3 30
Sequence IPv4 "Time To Live" only 4 30
Help
Close

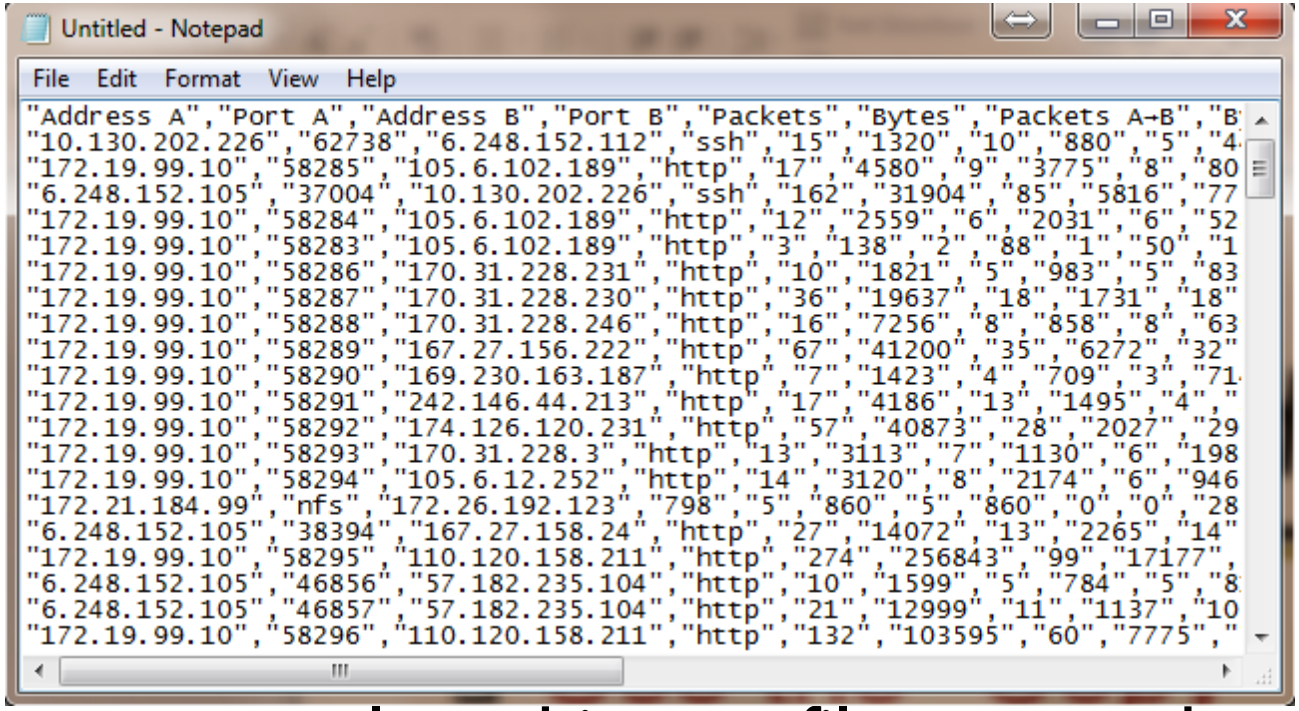
Conversations



Conversations



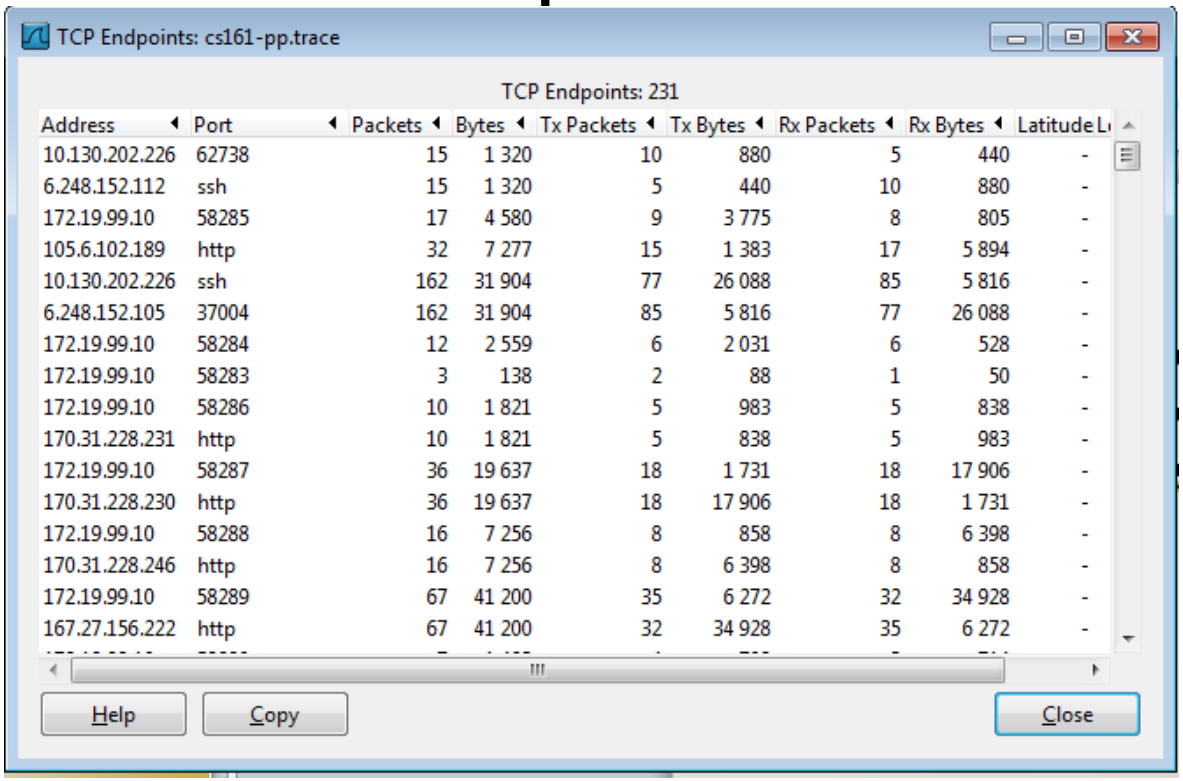
- Use the "Copy" button to copy all text into clipboard



- Then, you can analyze this text file to get what statistics you want

Find EndPoint Statistics

- Menu “statistics” → “endpoint list” → “TCP”

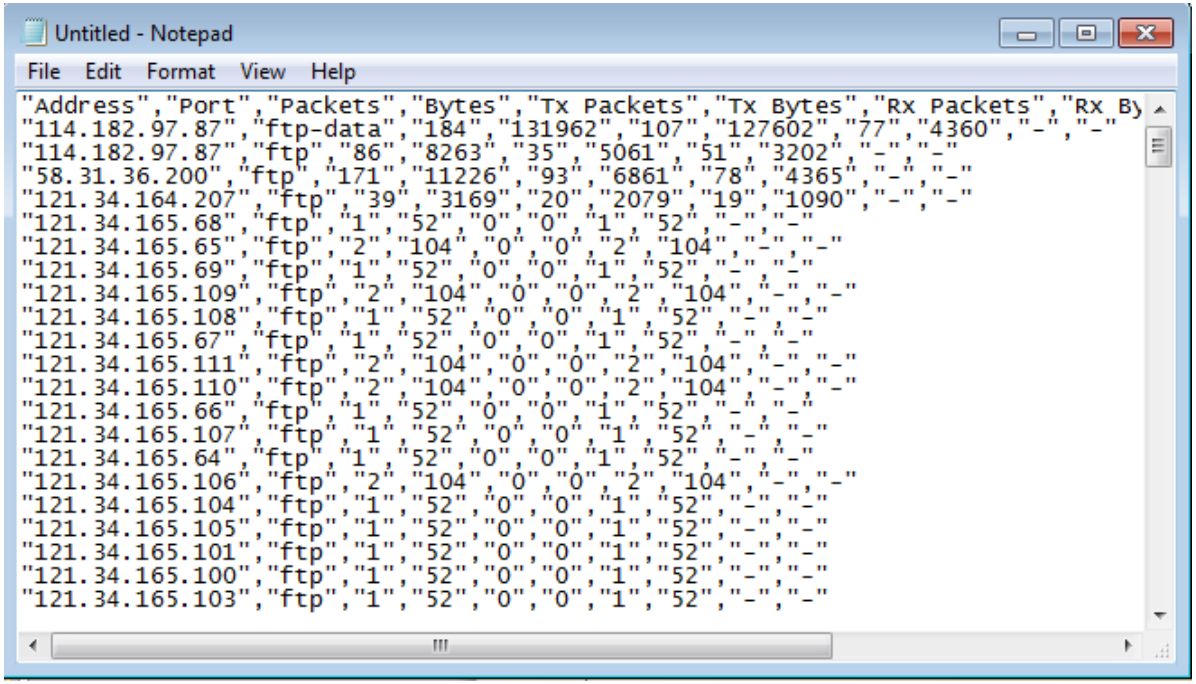


Address	Port	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Latitude
10.130.202.226	62738	15	1 320	10	880	5	440	-
6.248.152.112	ssh	15	1 320	5	440	10	880	-
172.19.99.10	58285	17	4 580	9	3 775	8	805	-
105.6.102.189	http	32	7 277	15	1 383	17	5 894	-
10.130.202.226	ssh	162	31 904	77	26 088	85	5 816	-
6.248.152.105	37004	162	31 904	85	5 816	77	26 088	-
172.19.99.10	58284	12	2 559	6	2 031	6	528	-
172.19.99.10	58283	3	138	2	88	1	50	-
172.19.99.10	58286	10	1 821	5	983	5	838	-
170.31.228.231	http	10	1 821	5	838	5	983	-
172.19.99.10	58287	36	19 637	18	1 731	18	17 906	-
170.31.228.230	http	36	19 637	18	17 906	18	1 731	-
172.19.99.10	58288	16	7 256	8	858	8	6 398	-
170.31.228.246	http	16	7 256	8	6 398	8	858	-
172.19.99.10	58289	67	41 200	35	6 272	32	34 928	-
167.27.156.222	http	67	41 200	32	34 928	35	6 272	-

- You can sort by field
- “Tx” : transmit “Rx” : receive

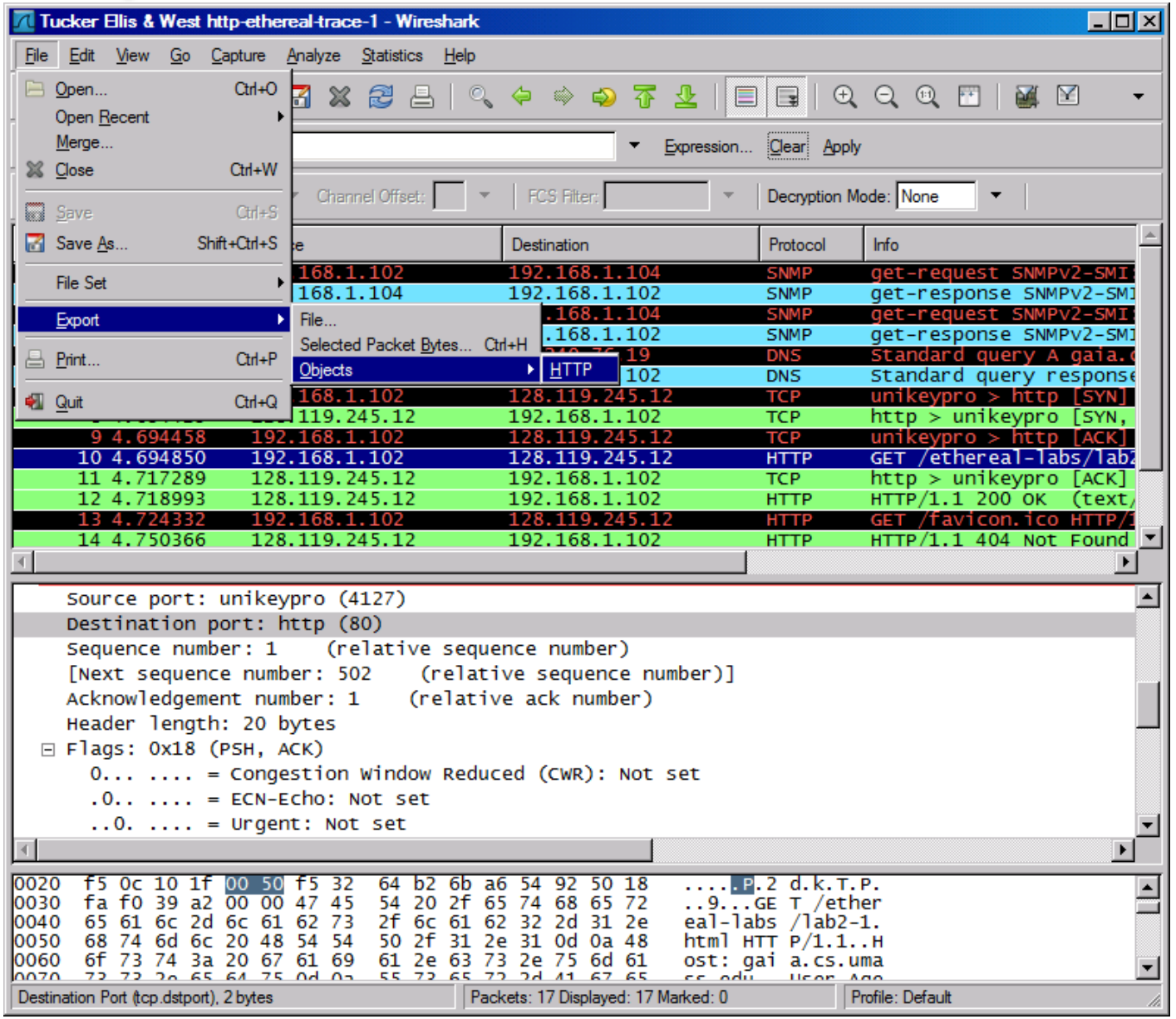
Find EndPoint Statistics

- Use the "Copy" button to copy all text into clipboard



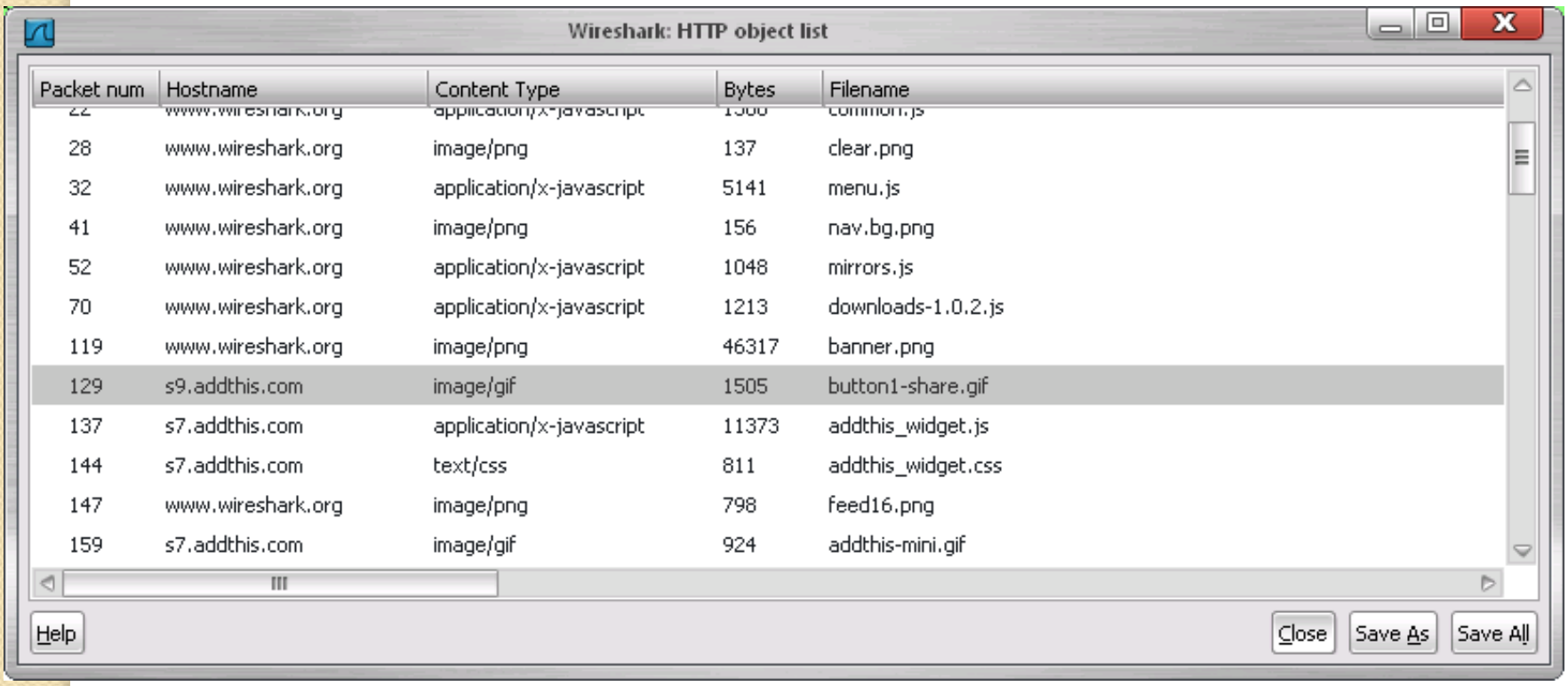
- Then, you can analyze this text file to get what statistics you want

Export HTTP

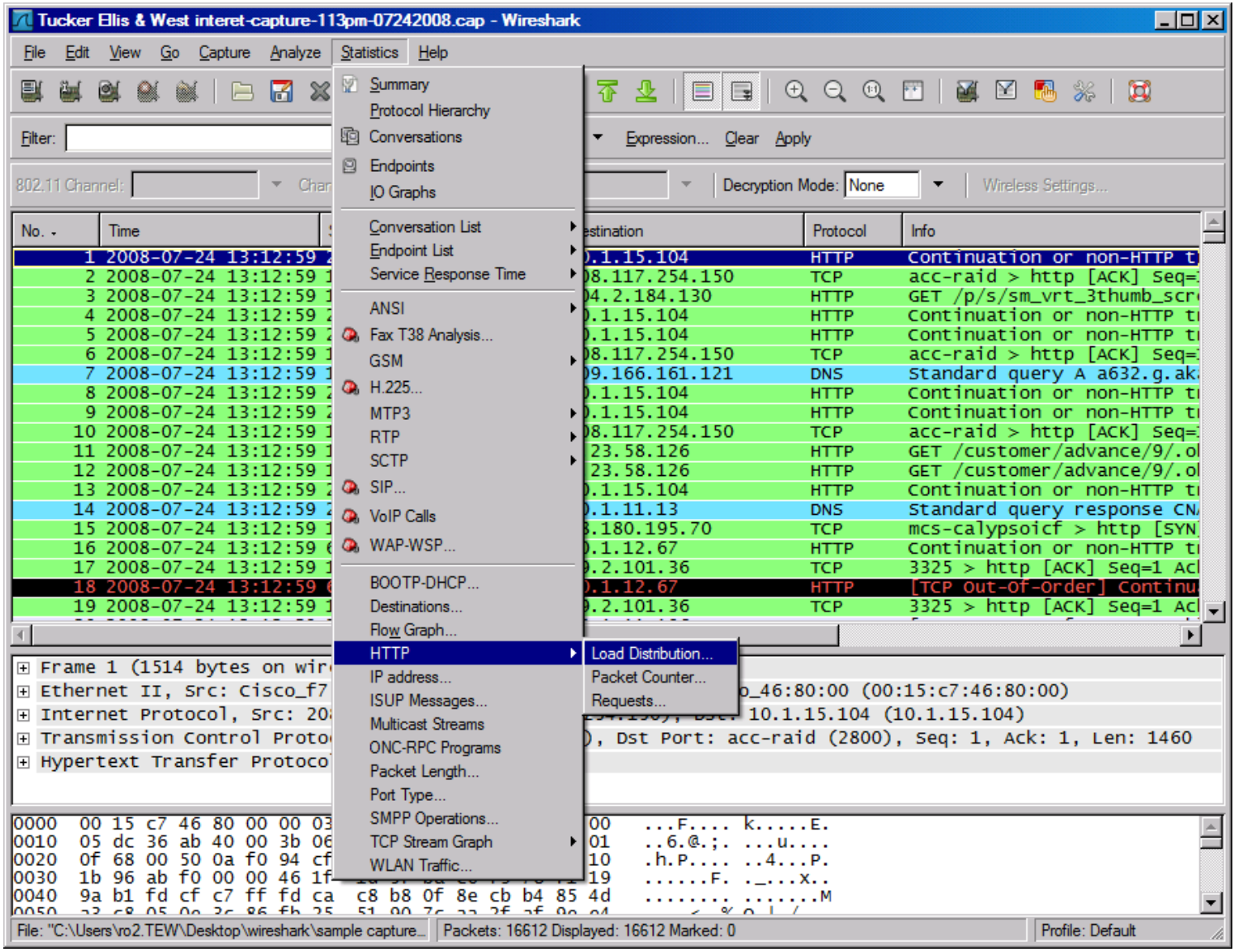

Tucker Ellis & West http-ethereal-trace-1 - Wireshark
File Edit View Go Capture Analyze Statistics Help
Open... Ctrl+O
Open Recent
Merge...
Close Ctrl+W
Save Ctrl+S
Save As... Shift+Ctrl+S
File Set
Export
Print... Ctrl+P
Quit Ctrl+Q
File...
Selected Packet Bytes... Ctrl+H
Objects
HTTP
Expression... Clear Apply
Channel Offset: FCS Filter: Decryption Mode: None
Destination Protocol Info
192.168.1.104 SNMP get-request SNMPv2-SMI
192.168.1.102 SNMP get-response SNMPv2-SMI
128.119.245.12 TCP unikeypro > http [SYN]
192.168.1.102 TCP http > unikeypro [SYN,
9 4.694458 192.168.1.102 128.119.245.12 TCP unikeypro > http [ACK]
10 4.694850 192.168.1.102 128.119.245.12 HTTP GET /ethereal-labs/lab2
11 4.717289 128.119.245.12 192.168.1.102 TCP http > unikeypro [ACK]
12 4.718993 128.119.245.12 192.168.1.102 HTTP HTTP/1.1 200 OK (text/
13 4.724332 192.168.1.102 128.119.245.12 HTTP GET /favicon.ico HTTP/1
14 4.750366 128.119.245.12 192.168.1.102 HTTP HTTP/1.1 404 Not Found
Source port: unikeypro (4127)
Destination port: http (80)
Sequence number: 1 (relative sequence number)
[Next sequence number: 502 (relative sequence number)]
Acknowledgement number: 1 (relative ack number)
Header length: 20 bytes
Flags: 0x18 (PSH, ACK)
0... .... = Congestion Window Reduced (CWR): Not set
.0.. .... = ECN-Echo: Not set
..0. .... = Urgent: Not set
0020 f5 0c 10 1f 00 50 f5 32 64 b2 6b a6 54 92 50 18 .....P.2 d.k.T.P.
0030 fa f0 39 a2 00 00 47 45 54 20 2f 65 74 68 65 72 ..9...GE T /ether
0040 65 61 6c 2d 6c 61 62 73 2f 6c 61 62 32 2d 31 2e eal-labs /lab2-1.
0050 68 74 6d 6c 20 48 54 54 50 2f 31 2e 31 0d 0a 48 html HTT P/1.1..H
0060 6f 73 74 3a 20 67 61 69 61 2e 63 73 2e 75 6d 61 ost: gai a.cs.uma
Destination Port (tcp.dstport), 2 bytes
Packets: 17 Displayed: 17 Marked: 0
Profile: Default

Export HTTP Objects

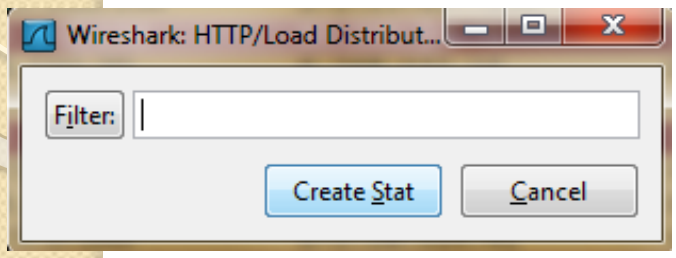
Now you can save all files transmitted in Web traffic!



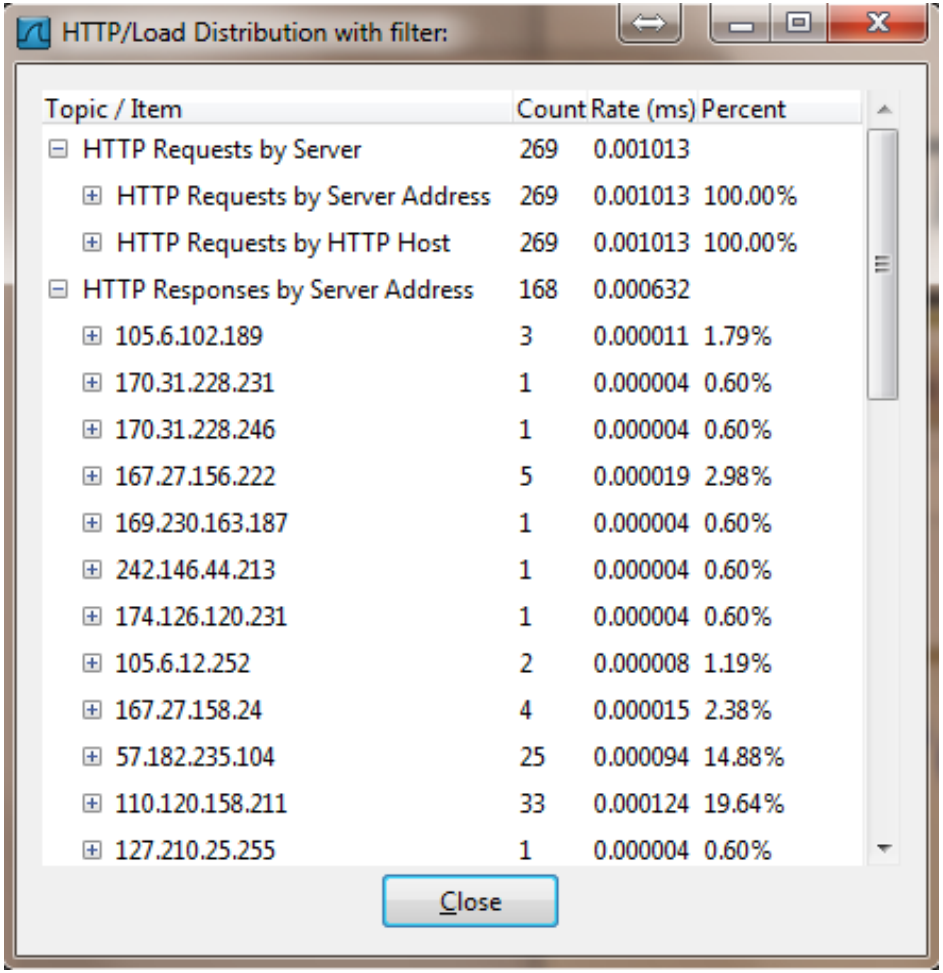
HTTP Analysis



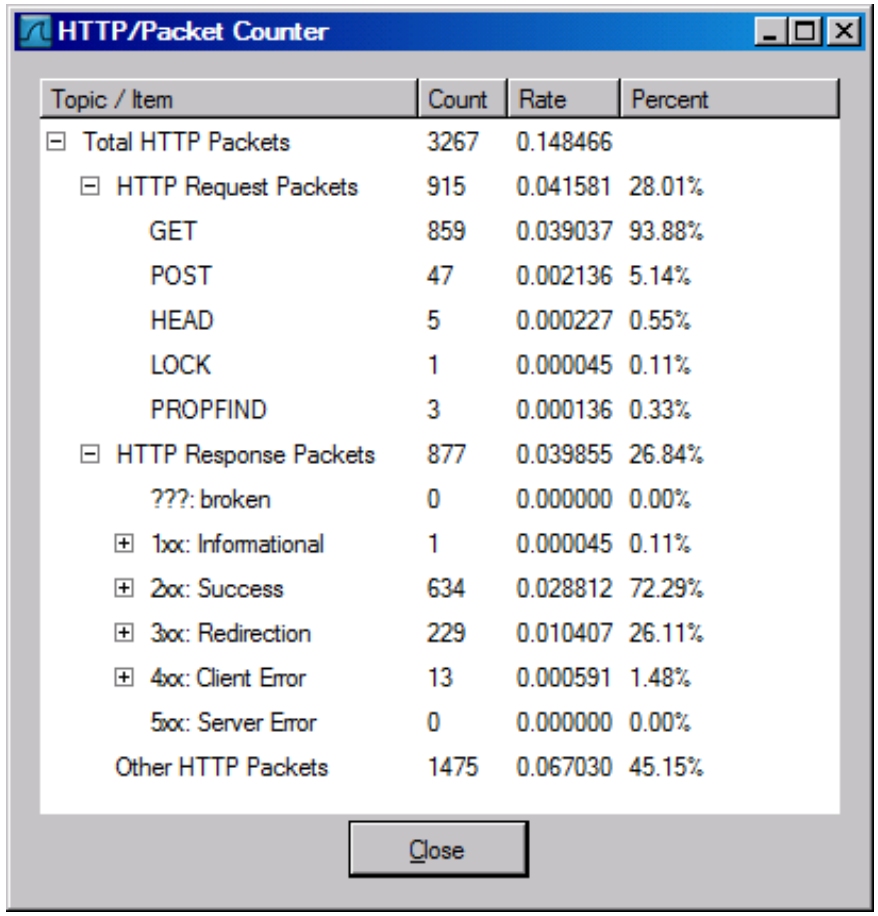
HTTP Analysis – Load Distribution



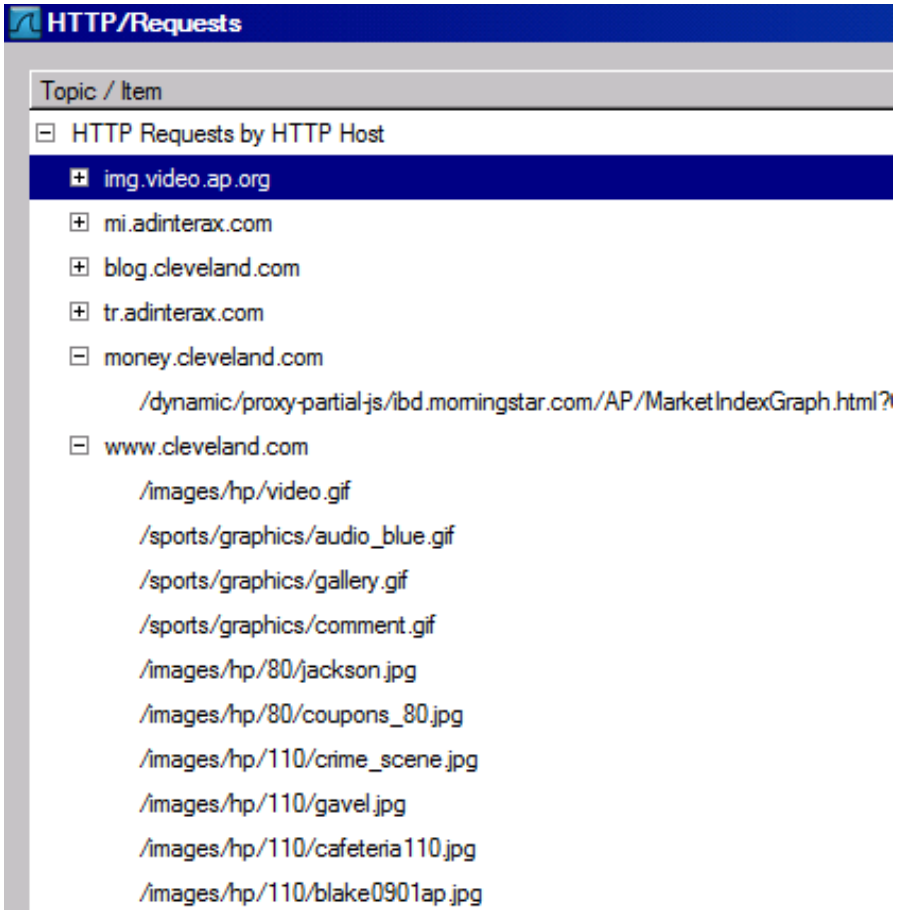
Click “Create Stat” button
You can add “filter” to only Show selected traffic



HTTP Analysis – Packet Counter



HTTP Analysis – Requests



Improving WireShark Performance

- Don’t use capture filters
- Increase your read buffer size
- Don’t update the screen dynamically
- Get a faster computer
- Use a TAP
- Don’t resolve DNS hostnames

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