Networks: Tutorial 06

Shinnazar Seytnazarov, PhD

Innopolis University

s.seytnazarov@innopolis.ru

Febuary 04, 2022

Topic of the lecture

- Connection-oriented transport: TCP
  - Connection management
  - Segment structure
  - Reliable data transfer
  - Flow control
- Principles of congestion control
- TCP congestion control

Topic of the tutorial

- Understand Traffic Capture and Analysis
- Layers and Encapsulation
- Examine Common Protocols
    - TCP
    - HTTP
    - DNS
    - FTP
- Explore the Wireshark interface

Network Analysis

- Process of capturing, decoding, and analyzing network traffic
  - Why is the network slow?
  - What is the network traffic pattern?
  - How is the traffic being shared between nodes?

- Also known as:
  - Traffic analysis, protocol analysis, sniffing, packet analysis, eavesdropping*, etc.

*Listen secretly to what is said in private!

Network Analyzer

- A combination of hardware and software tools what can detect, decode, and manipulate traffic on the network
  - Passive monitoring (detection) – Difficult to detect
  - Active (attack)
- Tools availability:
  - Free
  - Commercially

Network Analyzer

- Mainly software-based (utilizing OS and NIC)
    - Also known as *sniffer*
    - A program that monitors the data traveling through the network *passively*

- Common network analyzers
    - Wireshark / Ethereal
    - tcpdump
    - Windump
    - Etherpeak
    - Dsniff
    - And much more….

Network Analyzer Components

- Hardware
  - Special hardware devices
    - Monitoring voltage fluctuation
    - Jitter (random timing variation)
    - Jabber (failure to handle electrical signals)
  - NIC Card
- Capture driver
  - Capturing the data
- Buffer
  - Memory or disk-based
- Real-time analysis
  - Analyzing the traffic in real time
- Decoder
  - Making data readable

Capturing the data is easy!
The question is what to do with it!

Who Uses Network Analyzers?

- System administrators
  - Understand system problems and performance

- Malicious individuals (intruders)
  - Capture clear text data
  - Passively collect data on vulnerable protocols
    - FTP, POP3, IMAP, SMTP, HTTP, etc.
    - Capture VoIP data
  - Traffic pattern discovery
  - Actively break into the network (backdoor techniques)

Network Abstraction

Layers and Encapsulation

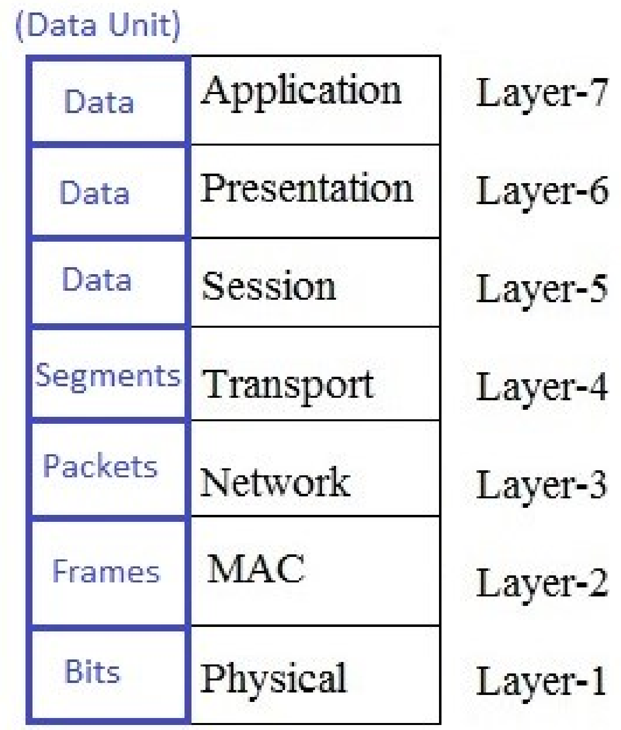
The OSI Model

To understand packet analysis you must understand the encapsulation process

The OSI Model

- A seven-layer representation
- How data changes as each layer provides services to the next layer
  - Data encapsulates
  - Data de-encapsulates

The OSI Model



(Data Unit)		
Data	Application	Layer-7
Data	Presentation	Layer-6
Data	Session	Layer-5
Segments	Transport	Layer-4
Packets	Network	Layer-3
Frames	MAC	Layer-2
Bits	Physical	Layer-1

OSI Layers

Source: http://www.rfwireless-world.com/Terminology/PHY-vs-MAC.html

Network Packet Analyzer – Wireshark

- Wireshark is a network packet analyzer.

- A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible

- You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable

- Wireshark is perhaps one of the best open source packet analyzers available today.

Some Intended Purposes

- Network administrators use it to troubleshoot network problems
- Network security engineers use it to examine security problems
- QA engineers use it to verify network applications
- Developers use it to debug protocol implementations
- People use it to learn network protocol internals

Features

- Capture live packet data from a network interface.
- Open files containing packet data captured with tcpdump/WinDump, Wireshark, and a number of other packet capture programs.
- Import packets from text files containing hex dumps of packet data.
- Display packets with very detailed protocol information.
- Save packet data captured.

Features

- Export some or all packets in a number of capture file formats.
- Filter packets on many criteria.
- Search for packets on many criteria.
- Colorize packet display based on filters.
- Create various statistics.
- …and a lot more!

What Wireshark is not!!

- Wireshark isn’t an intrusion detection system.
- It will not warn you when someone does strange things on your network that he/she isn’t allowed to do.
- However, if strange things happen, Wireshark might help you figure out what is really going on.
- Wireshark will not manipulate things on the network, it will only “measure” things from it.
- Wireshark doesn’t send packets on the network or do other active things.

Examine Common Protocols

TCP

A TCP Example

- Normal traffic
- Three-way handshake packets 1,2,3
- Review
  - Port numbers
  - Flags
  - SEQ ACK numbers
  - …. and so on.

Examine Common Protocols

UDP

UDP Example

- Connectionless Transport Layer service
- No handshake, sequencing or acknowledgement
- Few problems occur with UDP

UDP Applications

- Commonly used in video streaming and time-sensitive applications.
    - Domain Name System (DNS)
    - Routing Information Protocol (RIP)
    - Voice over IP (VoIP)
    - Trivial File Transfer Protocol (TFTP)
    - Domain Host Configuration Protocol (DHCP)

Examine Common Protocols

DNS

DNS

- DNS is essential to any network
- Converts host names (google.com) to an IP address (72.14.204.103)
- Client sends query to DNS server for an IP address
- Server responds with information
  - *or* asks other DNS servers for the information

DNS

- Transfers name information between DNS servers
  - DNS uses TCP in a zone transfer
- Look up other host names such as mail exchange (MX) records

DNS

- All DNS packets have four (4) sections:
  - Questions
  - Answer Resource Records
  - Authority Resources Records
  - Additional Resource Records

Examine Common Protocols

FTP

FTP – Grab a Pic

- Purpose of FTP is to transfer files over TCP

- Uses both ports 20 and 21
  - Command channel is designated on port 21 for the FTP server.
  - To transfer data like directory contents or files, a secondary channel, port 20 is used.

Reassemble the Streams

- Can reassemble and obtain content if data is not encrypted
- Filter ftp-data traffic
- Right click follow TCP stream and save the file as raw data and click save it.
- Go to where you saved the file and open it!

Examine Common Protocols

HTTP

Hypertext Transfer Protocol

- Actors in Web interaction
  - HTML
  - HTTP
  - Browser and the Web Server
- HTTP is a stateless protocol
- Two types of HTTP messages
  - Request and response

Hypertext Transfer Protocol

- Web page consists of objects
  - Identified by a URL or URI
- Request line (GET or POST methods)
- Additional information about the request
- Status code line
- Header Fields
- Data

HTTP Response Status Codes

- 2xx: Success
- 3xx: Redirection
- 4xx: Client Error
- 5xx: Server Error

Wireshark

Explore the Wireshark Interface

Capture Packets

- We will use pre-captured packets
- Review normal traffic

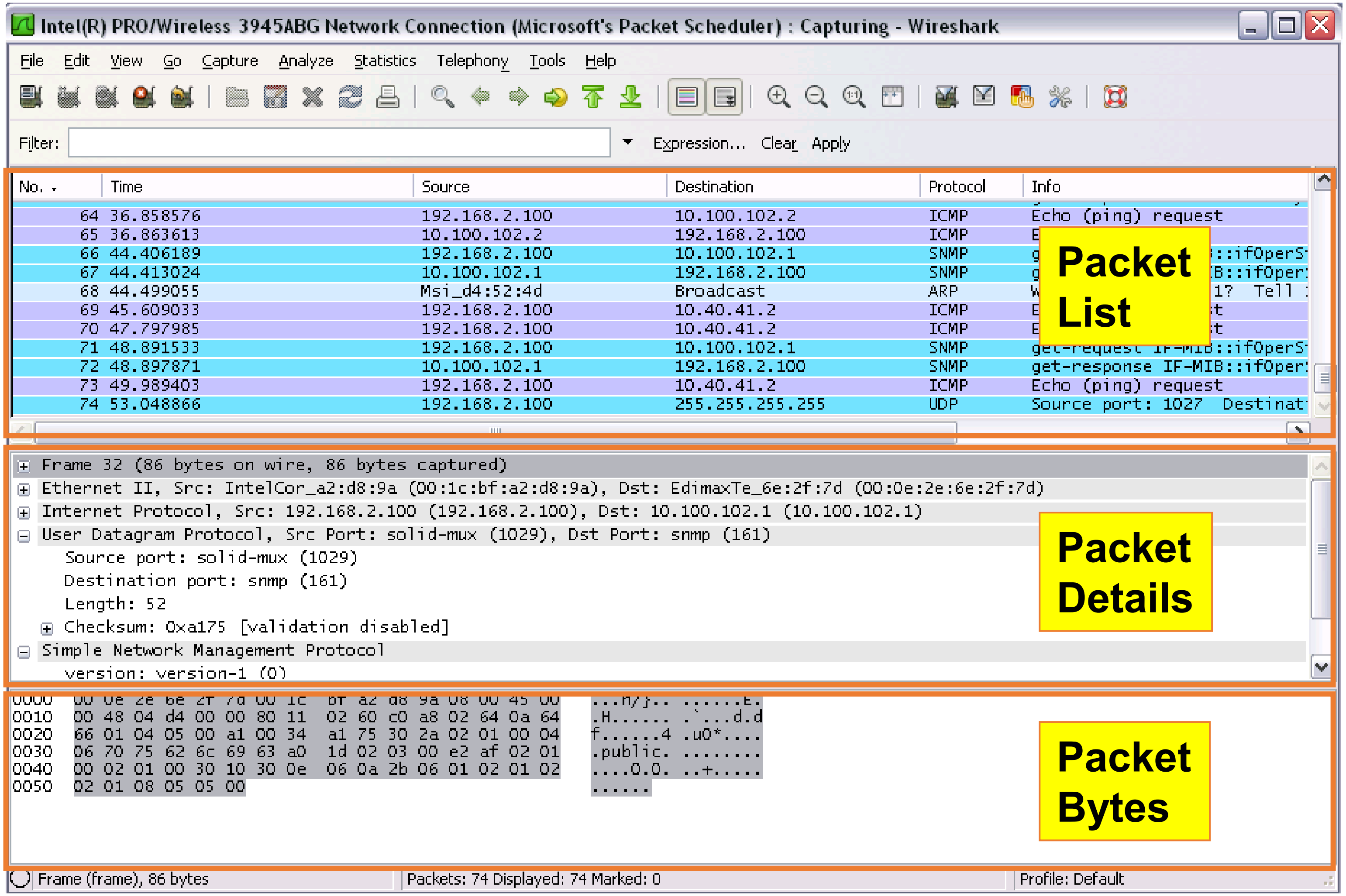
Capture Packets

- Once you open a capture you will see three panes:
  - **Top:** packet list of all of the packets received during the capture session
  - **Middle:** details of a single frame
  - **Bottom:** the bytes of a single frame

Capture Packets

- For a live capture
  - Launch Wireshark
  - Go to -> Capture Interfaces
  - Click the name of an interface
  - Start capturing packets on that interface

Interfaces



Intel(R) PRO/Wireless 3945ABG Network Connection (Microsoft's Packet Scheduler) : Capturing - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply

```
No. ▾  Time        Source          Destination      Protocol  Info
   64  36.858576   192.168.2.100   10.100.102.2     ICMP      Echo (ping) request
   65  36.863613   10.100.102.2    192.168.2.100    ICMP      E
   66  44.406189   192.168.2.100   10.100.102.1     SNMP      g              ::ifOperS
   67  44.413024   10.100.102.1    192.168.2.100    SNMP      g              B::ifOper
   68  44.499055   Msi_d4:52:4d    Broadcast        ARP       W              1?  Tell :
   69  45.609033   192.168.2.100   10.40.41.2       ICMP      E              t
   70  47.797985   192.168.2.100   10.40.41.2       ICMP      E              t
   71  48.891533   192.168.2.100   10.100.102.1     SNMP      get-request IF-MIB::ifOperS
   72  48.897871   10.100.102.1    192.168.2.100    SNMP      get-response IF-MIB::ifOper
   73  49.989403   192.168.2.100   10.40.41.2       ICMP      Echo (ping) request
   74  53.048866   192.168.2.100   255.255.255.255  UDP       Source port: 1027  Destinat
```

Packet List

```
⊞ Frame 32 (86 bytes on wire, 86 bytes captured)
⊞ Ethernet II, Src: IntelCor_a2:d8:9a (00:1c:bf:a2:d8:9a), Dst: EdimaxTe_6e:2f:7d (00:0e:2e:6e:2f:7d)
⊞ Internet Protocol, Src: 192.168.2.100 (192.168.2.100), Dst: 10.100.102.1 (10.100.102.1)
⊟ User Datagram Protocol, Src Port: solid-mux (1029), Dst Port: snmp (161)
    Source port: solid-mux (1029)
    Destination port: snmp (161)
    Length: 52
  ⊞ Checksum: 0xa175 [validation disabled]
⊟ Simple Network Management Protocol
    version: version-1 (0)
```

Packet Details

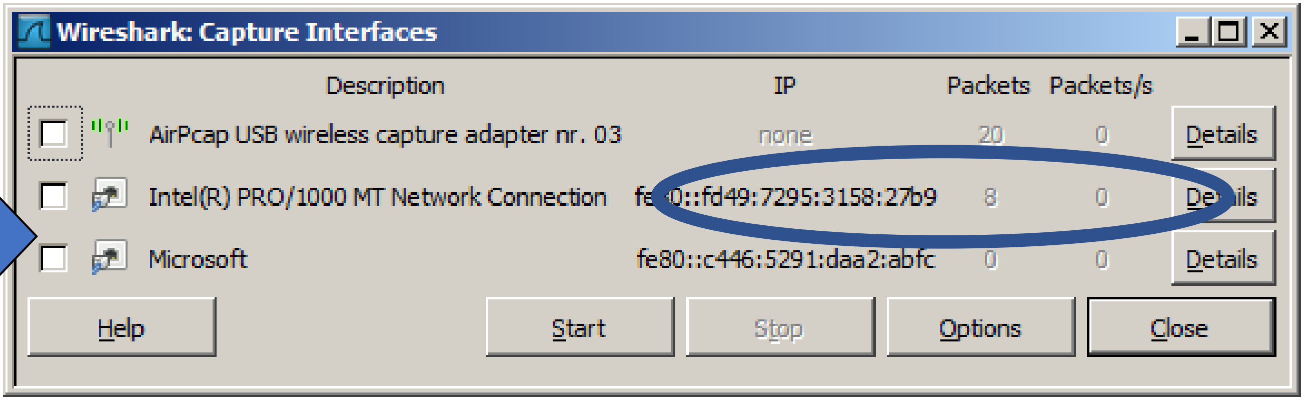
```
0000  00 0e 2e 6e 2f 7d 00 1c  bf a2 d8 9a 08 00 45 00   ...n/}.. ......E.
0010  00 48 04 d4 00 00 80 11  02 60 c0 a8 02 64 0a 64   .H...... .`...d.d
0020  66 01 04 05 00 a1 00 34  a1 75 30 2a 02 01 00 04   f......4 .u0*....
0030  06 70 75 62 6c 69 63 a0  1d 02 03 00 e2 af 02 01   .public. ........
0040  00 02 01 00 30 10 30 0e  06 0a 2b 06 01 02 01 02   ....0.0. ..+.....
0050  02 01 08 05 05 00                                  ......
```

Packet Bytes

Frame (frame), 86 bytes | Packets: 74 Displayed: 74 Marked: 0 | Profile: Default

Interfaces

Checkmark the interface you want to capture



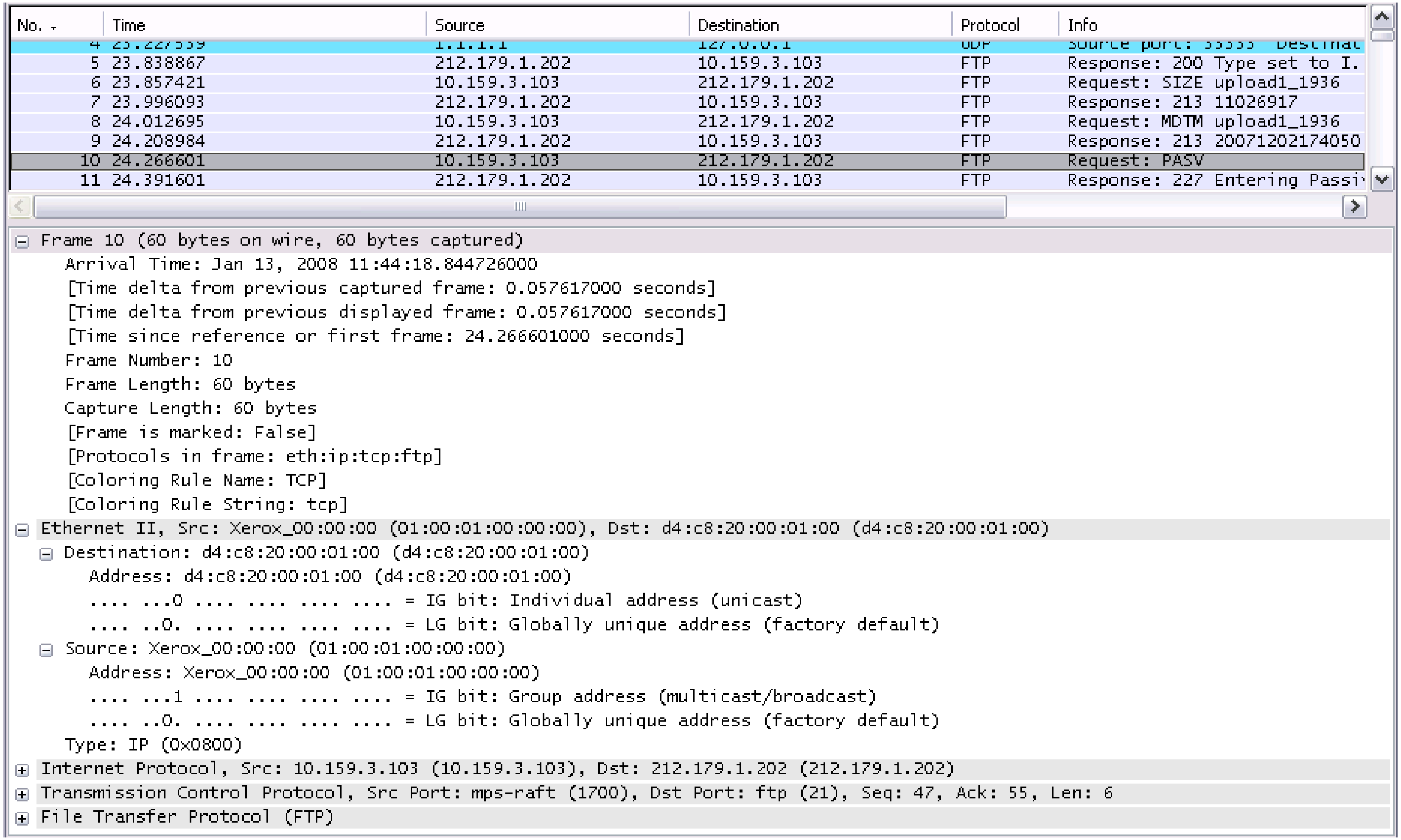
Wireshark: Capture Interfaces

		Description	IP	Packets	Packets/s	
□		AirPcap USB wireless capture adapter nr. 03	none	20	0	Details
□		Intel(R) PRO/1000 MT Network Connection	fe80::fd49:7295:3158:27b9	8	0	Details
□		Microsoft	fe80::c446:5291:daa2:abfc	0	0	Details

- Configure advanced features by clicking Options
- Select the interface with active packet exchange

Analyzing Packets (1/9)

- **Ethernet Frame Example**

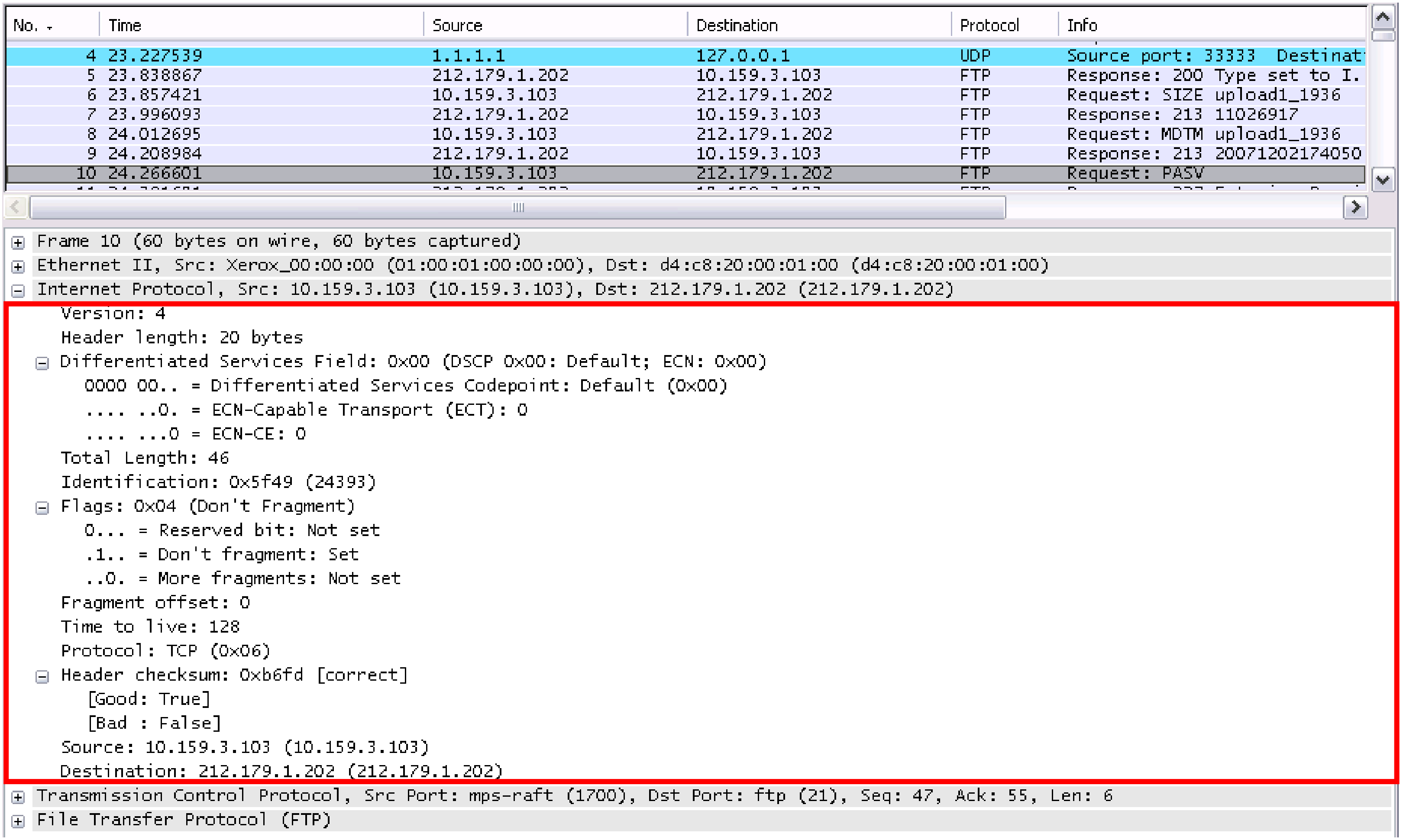


No.	Time	Source	Destination	Protocol	Info
4	23.227339	1.1.1.1	127.0.0.1	UDP	Source port: 33333 Destinat
5	23.838867	212.179.1.202	10.159.3.103	FTP	Response: 200 Type set to I.
6	23.857421	10.159.3.103	212.179.1.202	FTP	Request: SIZE upload1_1936
7	23.996093	212.179.1.202	10.159.3.103	FTP	Response: 213 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
11	24.391601	212.179.1.202	10.159.3.103	FTP	Response: 227 Entering Passi

```
⊟ Frame 10 (60 bytes on wire, 60 bytes captured)
    Arrival Time: Jan 13, 2008 11:44:18.844726000
    [Time delta from previous captured frame: 0.057617000 seconds]
    [Time delta from previous displayed frame: 0.057617000 seconds]
    [Time since reference or first frame: 24.266601000 seconds]
    Frame Number: 10
    Frame Length: 60 bytes
    Capture Length: 60 bytes
    [Frame is marked: False]
    [Protocols in frame: eth:ip:tcp:ftp]
    [Coloring Rule Name: TCP]
    [Coloring Rule String: tcp]
⊟ 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)
  ⊟ Destination: d4:c8:20:00:01:00 (d4:c8:20:00:01:00)
      Address: d4:c8:20:00:01:00 (d4:c8:20:00:01:00)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
  ⊟ Source: Xerox_00:00:00 (01:00:01:00:00:00)
      Address: Xerox_00:00:00 (01:00:01:00:00:00)
      .... ...1 .... .... .... .... = IG bit: Group address (multicast/broadcast)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
    Type: IP (0x0800)
⊞ Internet Protocol, Src: 10.159.3.103 (10.159.3.103), Dst: 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
⊞ File Transfer Protocol (FTP)
```

Analyzing Packets (2/9)

- IP Packet Example

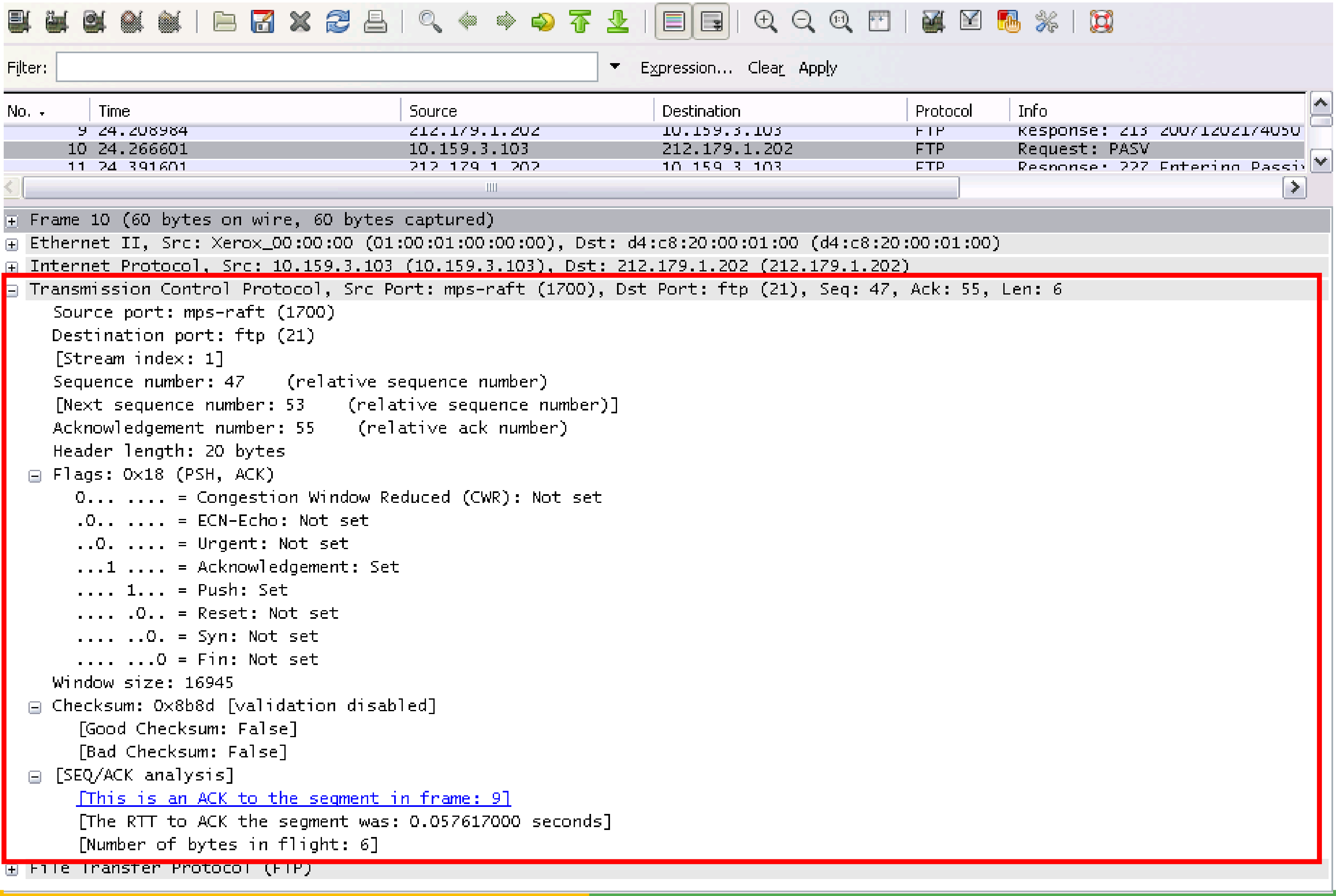


```
No.   Time        Source          Destination     Protocol  Info
   4  23.227539   1.1.1.1         127.0.0.1       UDP       Source port: 33333  Destinat
   5  23.838867   212.179.1.202   10.159.3.103    FTP       Response: 200 Type set to I.
   6  23.857421   10.159.3.103    212.179.1.202   FTP       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
⊞ File Transfer Protocol (FTP)
```

Analyzing Packets (3/9)

- TCP Packet Example

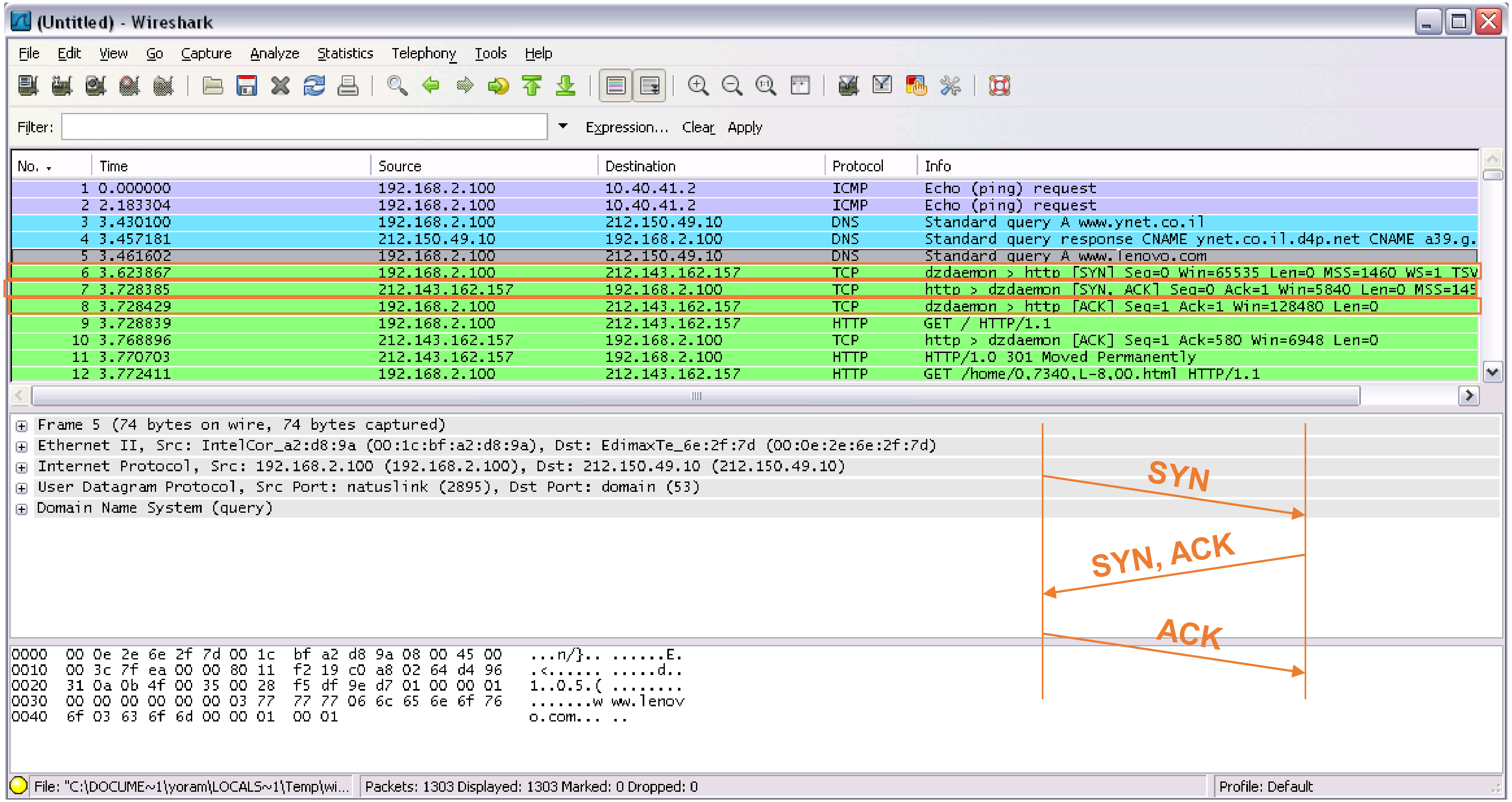


```
No. .   Time          Source          Destination     Protocol  Info
      9 24.208984     212.179.1.202   10.159.3.103    FTP       Response: 213 20071202174050
     10 24.266601     10.159.3.103    212.179.1.202   FTP       Request: PASV
     11 24.391601     212.179.1.202   10.159.3.103    FTP       Response: 227 Entering Passi
```

```
⊞ 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)
⊟ Transmission Control Protocol, Src Port: mps-raft (1700), Dst Port: ftp (21), Seq: 47, Ack: 55, Len: 6
    Source port: mps-raft (1700)
    Destination port: ftp (21)
    [Stream index: 1]
    Sequence number: 47    (relative sequence number)
    [Next sequence number: 53    (relative sequence number)]
    Acknowledgement number: 55    (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
      ...1 .... = Acknowledgement: Set
      .... 1... = Push: Set
      .... .0.. = Reset: Not set
      .... ..0. = Syn: Not set
      .... ...0 = Fin: Not set
    Window size: 16945
  ⊟ Checksum: 0x8b8d [validation disabled]
      [Good Checksum: False]
      [Bad Checksum: False]
  ⊟ [SEQ/ACK analysis]
      [This is an ACK to the segment in frame: 9]
      [The RTT to ACK the segment was: 0.057617000 seconds]
      [Number of bytes in flight: 6]
⊞ File Transfer Protocol (FTP)
```

Analyzing Packets (4/9)

- TCP 3-way Handshake


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter:
Expression... Clear Apply
SYN
SYN, ACK
ACK

```
No. .  Time        Source            Destination       Protocol  Info
     1 0.000000    192.168.2.100     10.40.41.2        ICMP      Echo (ping) request
     2 2.183304    192.168.2.100     10.40.41.2        ICMP      Echo (ping) request
     3 3.430100    192.168.2.100     212.150.49.10     DNS       Standard query A www.ynet.co.il
     4 3.457181    212.150.49.10     192.168.2.100     DNS       Standard query response CNAME ynet.co.il.d4p.net CNAME a39.g.
     5 3.461602    192.168.2.100     212.150.49.10     DNS       Standard query A www.lenovo.com
     6 3.623867    192.168.2.100     212.143.162.157   TCP       dzdaemon > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1 TSV
     7 3.728385    212.143.162.157   192.168.2.100     TCP       http > dzdaemon [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=145
     8 3.728429    192.168.2.100     212.143.162.157   TCP       dzdaemon > http [ACK] Seq=1 Ack=1 Win=128480 Len=0
     9 3.728839    192.168.2.100     212.143.162.157   HTTP      GET / HTTP/1.1
    10 3.768896    212.143.162.157   192.168.2.100     TCP       http > dzdaemon [ACK] Seq=1 Ack=580 Win=6948 Len=0
    11 3.770703    212.143.162.157   192.168.2.100     HTTP      HTTP/1.0 301 Moved Permanently
    12 3.772411    192.168.2.100     212.143.162.157   HTTP      GET /home/0,7340,L-8,00.html HTTP/1.1
```

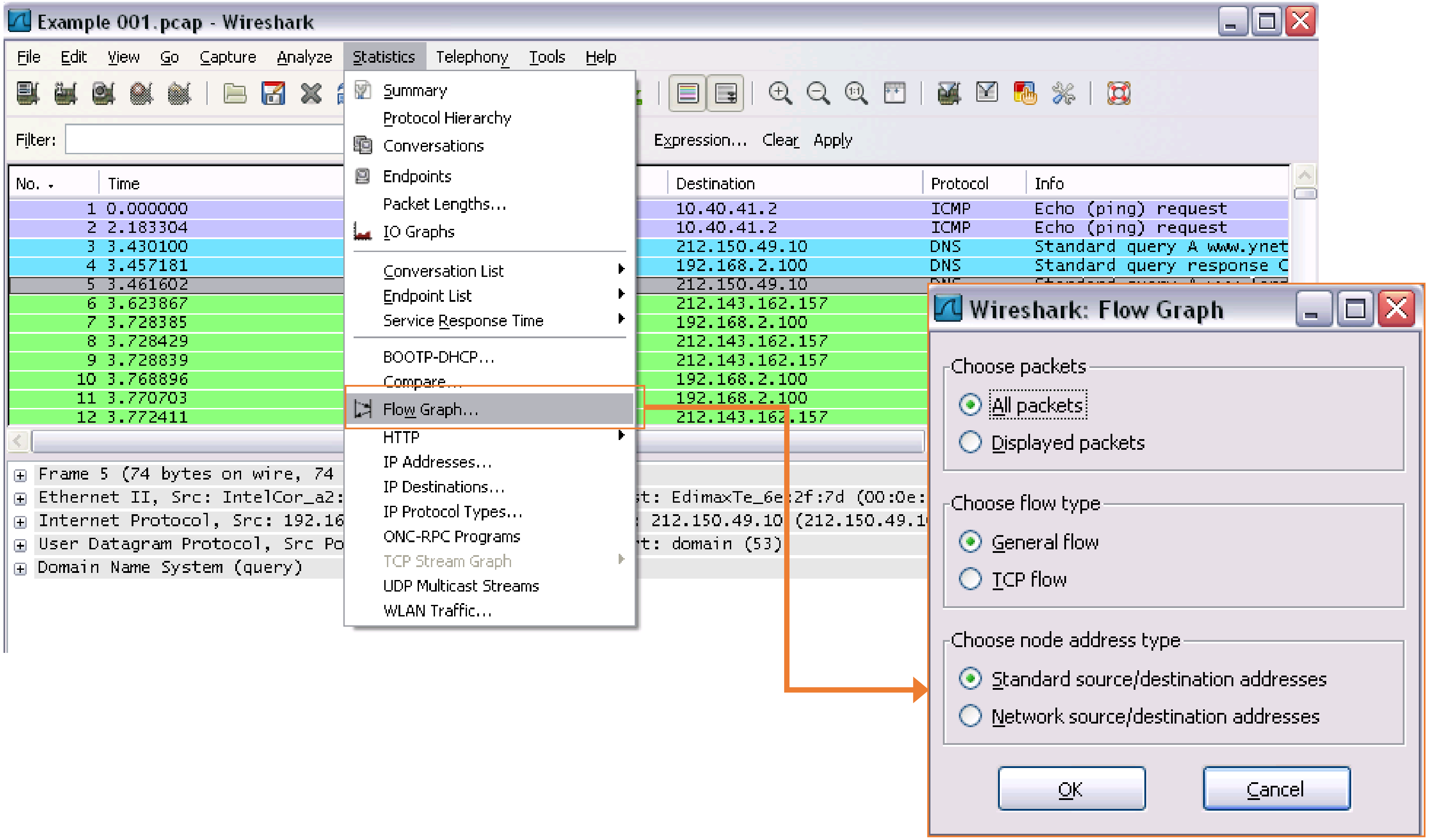
```
⊞ Frame 5 (74 bytes on wire, 74 bytes captured)
⊞ Ethernet II, Src: IntelCor_a2:d8:9a (00:1c:bf:a2:d8:9a), Dst: EdimaxTe_6e:2f:7d (00:0e:2e:6e:2f:7d)
⊞ Internet Protocol, Src: 192.168.2.100 (192.168.2.100), Dst: 212.150.49.10 (212.150.49.10)
⊞ User Datagram Protocol, Src Port: natuslink (2895), Dst Port: domain (53)
⊞ Domain Name System (query)
```

```
0000  00 0e 2e 6e 2f 7d 00 1c  bf a2 d8 9a 08 00 45 00   ...n/}.. ......E.
0010  00 3c 7f ea 00 00 80 11  f2 19 c0 a8 02 64 d4 96   .<...... .....d..
0020  31 0a 0b 4f 00 35 00 28  f5 df 9e d7 01 00 00 01   1..O.5.( ........
0030  00 00 00 00 00 00 03 77  77 77 06 6c 65 6e 6f 76   .......w ww.lenov
0040  6f 03 63 6f 6d 00 00 01  00 01                     o.com... ..
```

File: "C:\DOCUME~1\yoram\LOCALS~1\Temp\wi... Packets: 1303 Displayed: 1303 Marked: 0 Dropped: 0 Profile: Default

Analyzing Packets (5/9)

- Flow Graph
  - Giving us a graphical flow, for better understanding of what we see



No.	Time	Destination	Protocol	Info
1	0.000000	10.40.41.2	ICMP	Echo (ping) request
2	2.183304	10.40.41.2	ICMP	Echo (ping) request
3	3.430100	212.150.49.10	DNS	Standard query A www.ynet
4	3.457181	192.168.2.100	DNS	Standard query response C
5	3.461602	212.150.49.10		
6	3.623867	212.143.162.157		
7	3.728385	192.168.2.100		
8	3.728429	212.143.162.157		
9	3.728839	212.143.162.157		
10	3.768896	192.168.2.100		
11	3.770703	192.168.2.100		
12	3.772411	212.143.162.157		

```
⊞ Frame 5 (74 bytes on wire, 74
⊞ Ethernet II, Src: IntelCor_a2:          t: EdimaxTe_6e:2f:7d (00:0e:
⊞ Internet Protocol, Src: 192.16          : 212.150.49.10 (212.150.49.1
⊞ User Datagram Protocol, Src Po          t: domain (53)
⊞ Domain Name System (query)
```

Choose packets

- ● All packets
- ○ Displayed packets

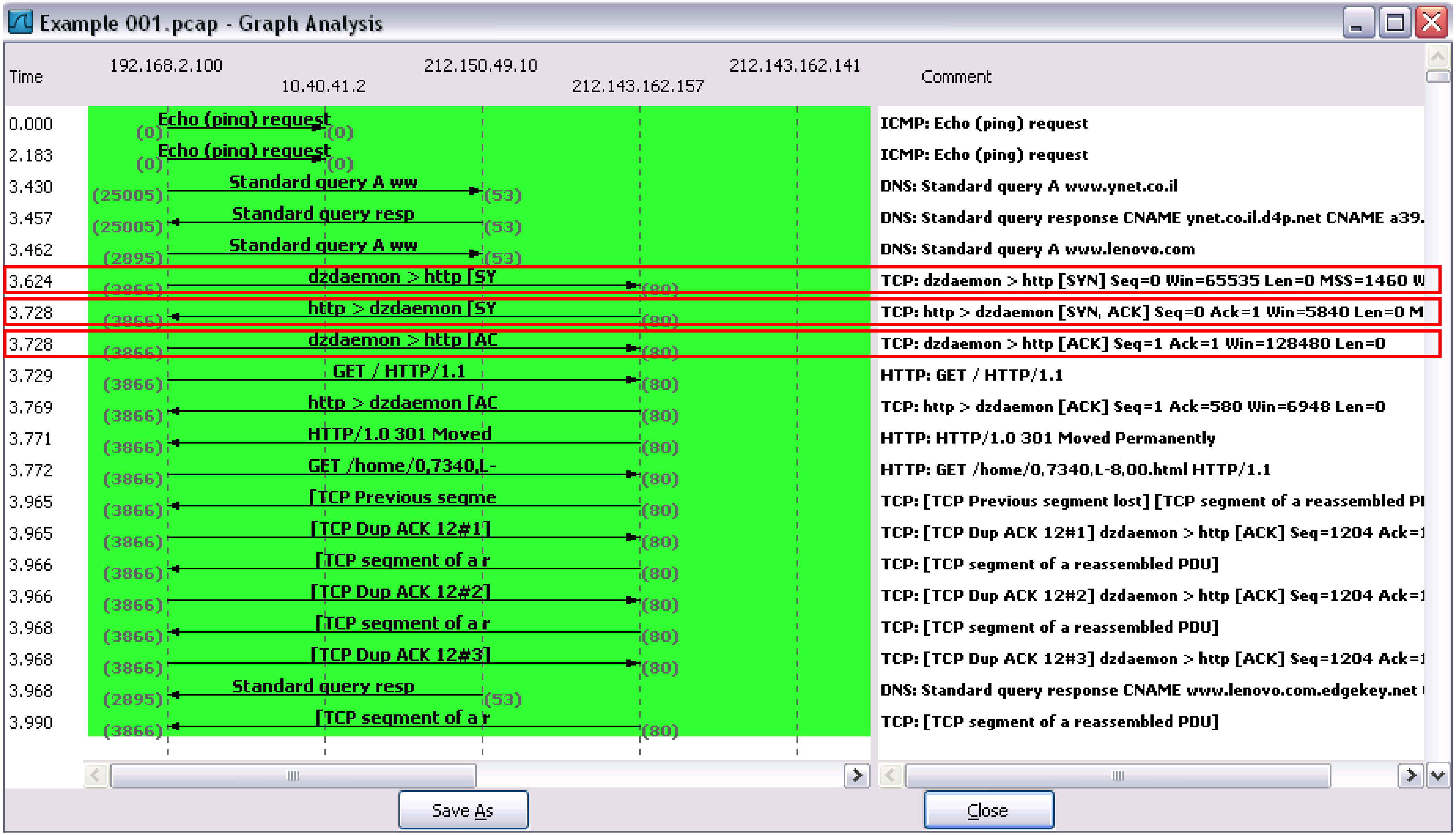
Choose flow type

- ● General flow
- ○ TCP flow

Choose node address type

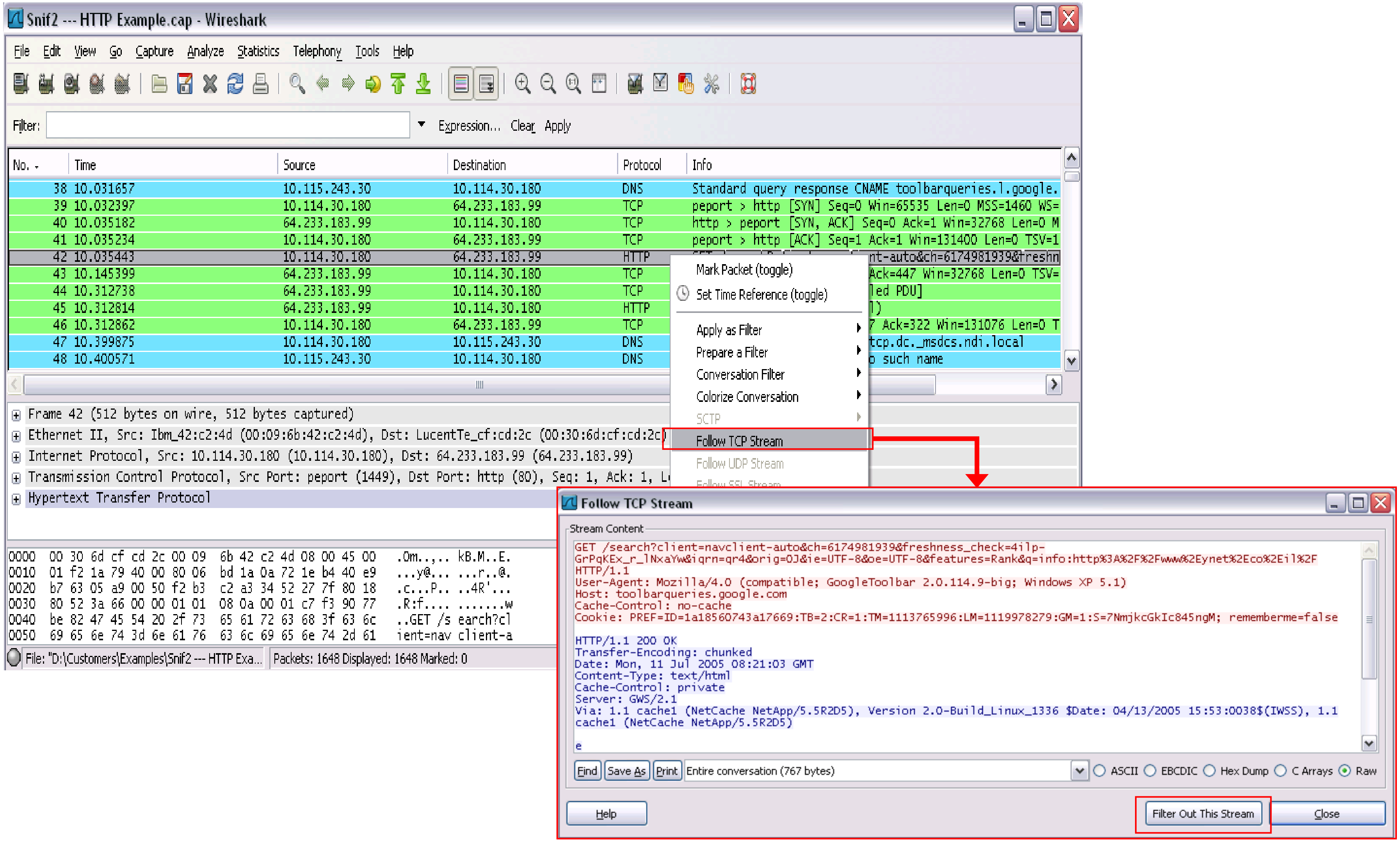
- ● Standard source/destination addresses
- ○ Network source/destination addresses

Analyzing Packets (6/9)



Time	Flow	Comment
0.000	(0) Echo (ping) request (0)	ICMP: Echo (ping) request
2.183	(0) Echo (ping) request (0)	ICMP: Echo (ping) request
3.430	(25005) Standard query A ww (53)	DNS: Standard query A www.ynet.co.il
3.457	(25005) Standard query resp (53)	DNS: Standard query response CNAME ynet.co.il.d4p.net CNAME a39.
3.462	(2895) Standard query A ww (53)	DNS: Standard query A www.lenovo.com
3.624	(3866) dzdaemon > http [SY (80)	TCP: dzdaemon > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 W
3.728	(3866) http > dzdaemon [SY (80)	TCP: http > dzdaemon [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 M
3.728	(3866) dzdaemon > http [AC (80)	TCP: dzdaemon > http [ACK] Seq=1 Ack=1 Win=128480 Len=0
3.729	(3866) GET / HTTP/1.1 (80)	HTTP: GET / HTTP/1.1
3.769	(3866) http > dzdaemon [AC (80)	TCP: http > dzdaemon [ACK] Seq=1 Ack=580 Win=6948 Len=0
3.771	(3866) HTTP/1.0 301 Moved (80)	HTTP: HTTP/1.0 301 Moved Permanently
3.772	(3866) GET /home/0,7340,L- (80)	HTTP: GET /home/0,7340,L-8,00.html HTTP/1.1
3.965	(3866) [TCP Previous segme (80)	TCP: [TCP Previous segment lost] [TCP segment of a reassembled PI
3.965	(3866) [TCP Dup ACK 12#1] (80)	TCP: [TCP Dup ACK 12#1] dzdaemon > http [ACK] Seq=1204 Ack=1
3.966	(3866) [TCP segment of a r (80)	TCP: [TCP segment of a reassembled PDU]
3.966	(3866) [TCP Dup ACK 12#2] (80)	TCP: [TCP Dup ACK 12#2] dzdaemon > http [ACK] Seq=1204 Ack=1
3.968	(3866) [TCP segment of a r (80)	TCP: [TCP segment of a reassembled PDU]
3.968	(3866) [TCP Dup ACK 12#3] (80)	TCP: [TCP Dup ACK 12#3] dzdaemon > http [ACK] Seq=1204 Ack=1
3.968	(2895) Standard query resp (53)	DNS: Standard query response CNAME www.lenovo.com.edgekey.net
3.990	(3866) [TCP segment of a r (80)	TCP: [TCP segment of a reassembled PDU]

Analyzing Packets (7/9)



Snif2 --- HTTP Example.cap - Wireshark
File Edit View Go Capture Analyze Statistics Telephony Tools Help
Filter: Expression... Clear Apply

```
No. -  Time        Source          Destination     Protocol  Info
   38  10.031657   10.115.243.30   10.114.30.180   DNS       Standard query response CNAME toolbarqueries.l.google.
   39  10.032397   10.114.30.180   64.233.183.99   TCP       peport > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=
   40  10.035182   64.233.183.99   10.114.30.180   TCP       http > peport [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 M
   41  10.035234   10.114.30.180   64.233.183.99   TCP       peport > http [ACK] Seq=1 Ack=1 Win=131400 Len=0 TSV=1
   42  10.035443   10.114.30.180   64.233.183.99   HTTP                          nt-auto&ch=6174981939&freshn
   43  10.145399   64.233.183.99   10.114.30.180   TCP                           Ack=447 Win=32768 Len=0 TSV=
   44  10.312738   64.233.183.99   10.114.30.180   TCP                           led PDU]
   45  10.312814   64.233.183.99   10.114.30.180   HTTP                          l)
   46  10.312862   10.114.30.180   64.233.183.99   TCP                           7 Ack=322 Win=131076 Len=0 T
   47  10.399875   10.114.30.180   10.115.243.30   DNS                           tcp.dc._msdcs.ndi.local
   48  10.400571   10.115.243.30   10.114.30.180   DNS                           o such name
```

```
⊞ Frame 42 (512 bytes on wire, 512 bytes captured)
⊞ Ethernet II, Src: Ibm_42:c2:4d (00:09:6b:42:c2:4d), Dst: LucentTe_cf:cd:2c (00:30:6d:cf:cd:2c)
⊞ Internet Protocol, Src: 10.114.30.180 (10.114.30.180), Dst: 64.233.183.99 (64.233.183.99)
⊞ Transmission Control Protocol, Src Port: peport (1449), Dst Port: http (80), Seq: 1, Ack: 1, Le
⊞ Hypertext Transfer Protocol
```

```
0000  00 30 6d cf cd 2c 00 09  6b 42 c2 4d 08 00 45 00   .0m..,.. kB.M..E.
0010  01 f2 1a 79 40 00 80 06  bd 1a 0a 72 1e b4 40 e9   ...y@... ...r..@.
0020  b7 63 05 a9 00 50 f2 b3  c2 a3 34 52 27 7f 80 18   .c...P.. ..4R'...
0030  80 52 3a 66 00 00 01 01  08 0a 00 01 c7 f3 90 77   .R:f.... .......w
0040  be 82 47 45 54 20 2f 73  65 61 72 63 68 3f 63 6c   ..GET /s earch?cl
0050  69 65 6e 74 3d 6e 61 76  63 6c 69 65 6e 74 2d 61   ient=nav client-a
```

File: "D:\Customers\Examples\Snif2 --- HTTP Exa... | Packets: 1648 Displayed: 1648 Marked: 0

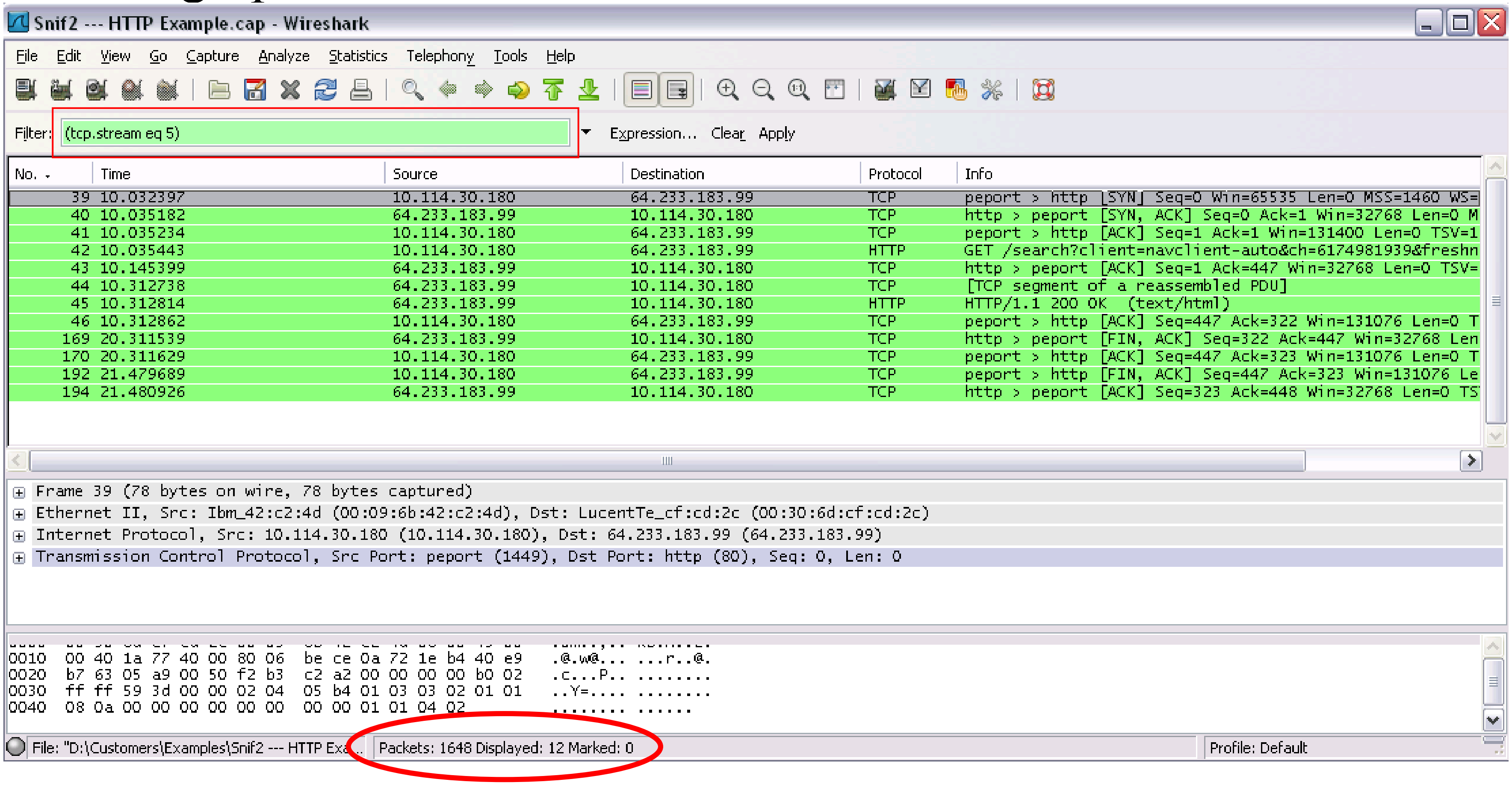
```
GET /search?client=navclient-auto&ch=6174981939&freshness_check=4ilp-
GrPqKEx_r_lNxaYw&iqrn=qr4&orig=0J&ie=UTF-8&oe=UTF-8&features=Rank&q=info:http%3A%2F%2Fwww%2Eynet%2Eco%2Eil%2F
HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; GoogleToolbar 2.0.114.9-big; Windows XP 5.1)
Host: toolbarqueries.google.com
Cache-Control: no-cache
Cookie: PREF=ID=1a18560743a17669:TB=2:CR=1:TM=1113765996:LM=1119978279:GM=1:S=7NmjkcGkIc845ngM; rememberme=false

HTTP/1.1 200 OK
Transfer-Encoding: chunked
Date: Mon, 11 Jul 2005 08:21:03 GMT
Content-Type: text/html
Cache-Control: private
Server: GWS/2.1
Via: 1.1 cache1 (NetCache NetApp/5.5R2D5), Version 2.0-Build_Linux_1336 $Date: 04/13/2005 15:53:0038$(IWSS), 1.1
cache1 (NetCache NetApp/5.5R2D5)

e
```

Entire conversation (767 bytes)

○ ASCII ○ EBCDIC ○ Hex Dump ○ C Arrays ● Raw

Analyzing Packets (8/9)

- Filtering Specific TCP Stream



Filter: (tcp.stream eq 5)

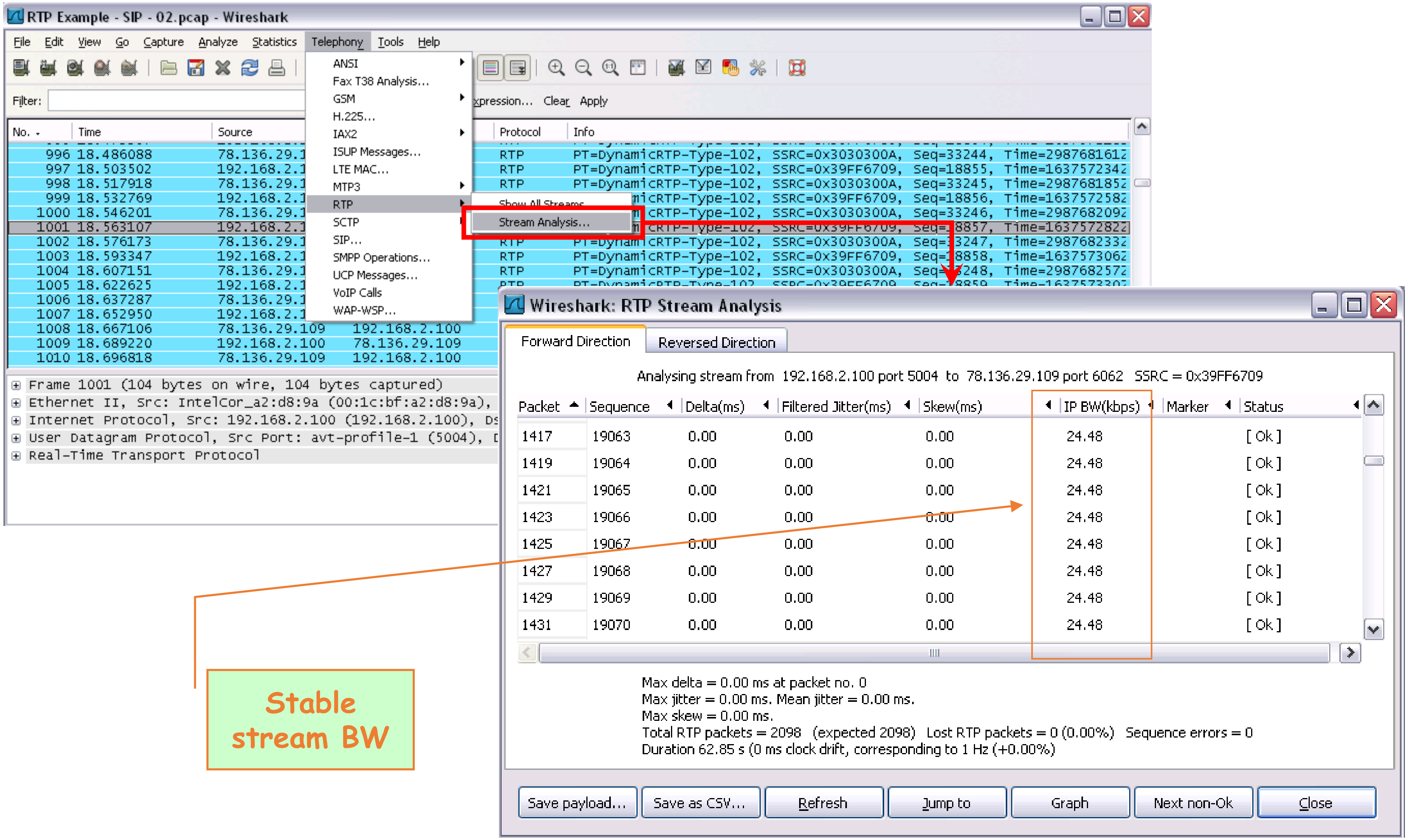
```
No. .  Time         Source          Destination     Protocol  Info
   39  10.032397    10.114.30.180   64.233.183.99   TCP       peport > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=
   40  10.035182    64.233.183.99   10.114.30.180   TCP       http > peport [SYN, ACK] Seq=0 Ack=1 Win=32768 Len=0 M
   41  10.035234    10.114.30.180   64.233.183.99   TCP       peport > http [ACK] Seq=1 Ack=1 Win=131400 Len=0 TSV=1
   42  10.035443    10.114.30.180   64.233.183.99   HTTP      GET /search?client=navclient-auto&ch=6174981939&freshn
   43  10.145399    64.233.183.99   10.114.30.180   TCP       http > peport [ACK] Seq=1 Ack=447 Win=32768 Len=0 TSV=
   44  10.312738    64.233.183.99   10.114.30.180   TCP       [TCP segment of a reassembled PDU]
   45  10.312814    64.233.183.99   10.114.30.180   HTTP      HTTP/1.1 200 OK  (text/html)
   46  10.312862    10.114.30.180   64.233.183.99   TCP       peport > http [ACK] Seq=447 Ack=322 Win=131076 Len=0 T
  169  20.311539    64.233.183.99   10.114.30.180   TCP       http > peport [FIN, ACK] Seq=322 Ack=447 Win=32768 Len
  170  20.311629    10.114.30.180   64.233.183.99   TCP       peport > http [ACK] Seq=447 Ack=323 Win=131076 Len=0 T
  192  21.479689    10.114.30.180   64.233.183.99   TCP       peport > http [FIN, ACK] Seq=447 Ack=323 Win=131076 Le
  194  21.480926    64.233.183.99   10.114.30.180   TCP       http > peport [ACK] Seq=323 Ack=448 Win=32768 Len=0 TS
```

```
⊞ Frame 39 (78 bytes on wire, 78 bytes captured)
⊞ Ethernet II, Src: Ibm_42:c2:4d (00:09:6b:42:c2:4d), Dst: LucentTe_cf:cd:2c (00:30:6d:cf:cd:2c)
⊞ Internet Protocol, Src: 10.114.30.180 (10.114.30.180), Dst: 64.233.183.99 (64.233.183.99)
⊞ Transmission Control Protocol, Src Port: peport (1449), Dst Port: http (80), Seq: 0, Len: 0
```

```
0010  00 40 1a 77 40 00 80 06  be ce 0a 72 1e b4 40 e9   .@.w@... ...r..@.
0020  b7 63 05 a9 00 50 f2 b3  c2 a2 00 00 00 00 b0 02   .c...P.. ........
0030  ff ff 59 3d 00 00 02 04  05 b4 01 03 03 02 01 01   ..Y=.... ........
0040  08 0a 00 00 00 00 00 00  00 00 01 01 04 02         ........ ......
```

File: "D:\Customers\Examples\Snif2 --- HTTP Exa... | Packets: 1648 Displayed: 12 Marked: 0 | Profile: Default

Analyzing Packets (9/9)



```
No.   Time       Source        Protocol Info
 996 18.486088  78.136.29.1    RTP      PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33244, Time=2987681612
 997 18.503502  192.168.2.1    RTP      PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18855, Time=1637572342
 998 18.517918  78.136.29.1    RTP      PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33245, Time=2987681852
 999 18.532769  192.168.2.1             micRTP-Type-102, SSRC=0x39FF6709, Seq=18856, Time=1637572582
1000 18.546201  78.136.29.1             micRTP-Type-102, SSRC=0x3030300A, Seq=33246, Time=2987682092
1001 18.563107  192.168.2.1             cRTP-Type-102, SSRC=0x39FF6709, Seq=18857, Time=1637572822
1002 18.576173  78.136.29.1    RTP      PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33247, Time=2987682332
1003 18.593347  192.168.2.1    RTP      PT=DynamicRTP-Type-102, SSRC=0x39FF6709, Seq=18858, Time=1637573062
1004 18.607151  78.136.29.1    RTP      PT=DynamicRTP-Type-102, SSRC=0x3030300A, Seq=33248, Time=2987682572
1005 18.622625  192.168.2.1
1006 18.637287  78.136.29.1
1007 18.652950  192.168.2.1
1008 18.667106  78.136.29.109  192.168.2.100
1009 18.689220  192.168.2.100  78.136.29.109
1010 18.696818  78.136.29.109  192.168.2.100

⊞ Frame 1001 (104 bytes on wire, 104 bytes captured)
⊞ Ethernet II, Src: IntelCor_a2:d8:9a (00:1c:bf:a2:d8:9a),
⊞ Internet Protocol, Src: 192.168.2.100 (192.168.2.100), Ds
⊞ User Datagram Protocol, Src Port: avt-profile-1 (5004), D
⊞ Real-Time Transport Protocol
```

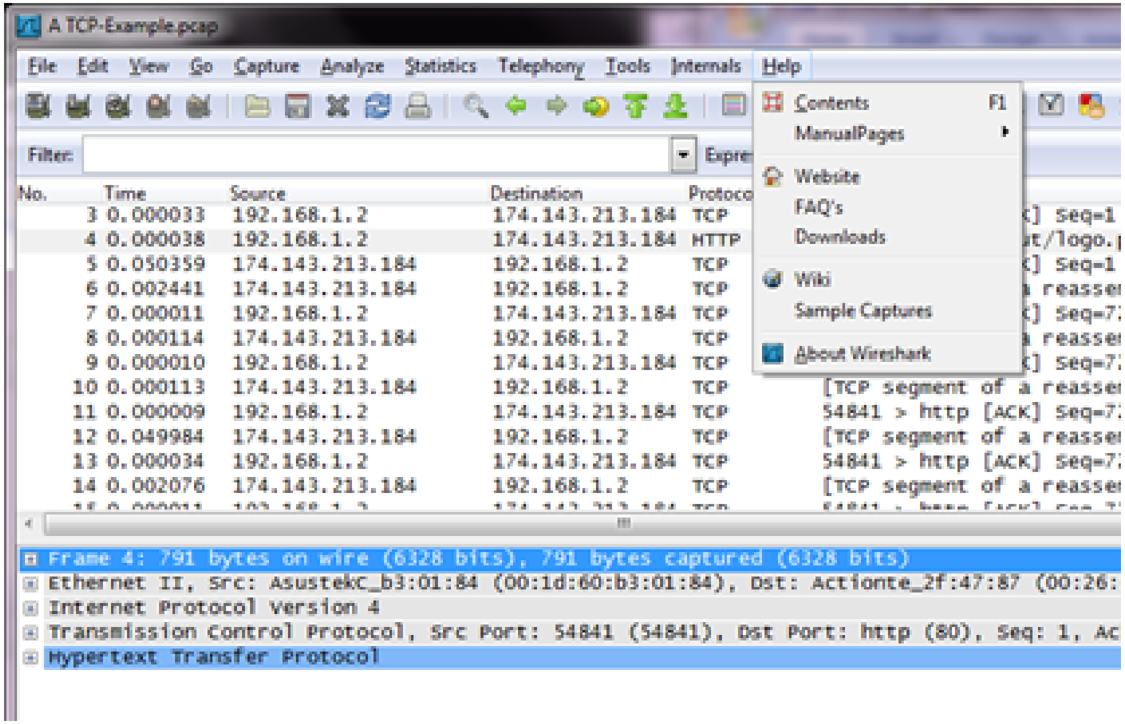
Analysing stream from 192.168.2.100 port 5004 to 78.136.29.109 port 6062 SSRC = 0x39FF6709

Packet	Sequence	Delta(ms)	Filtered Jitter(ms)	Skew(ms)	IP BW(kbps)	Marker	Status
1417	19063	0.00	0.00	0.00	24.48		[Ok]
1419	19064	0.00	0.00	0.00	24.48		[Ok]
1421	19065	0.00	0.00	0.00	24.48		[Ok]
1423	19066	0.00	0.00	0.00	24.48		[Ok]
1425	19067	0.00	0.00	0.00	24.48		[Ok]
1427	19068	0.00	0.00	0.00	24.48		[Ok]
1429	19069	0.00	0.00	0.00	24.48		[Ok]
1431	19070	0.00	0.00	0.00	24.48		[Ok]

Max delta = 0.00 ms at packet no. 0
Max jitter = 0.00 ms. Mean jitter = 0.00 ms.
Max skew = 0.00 ms.
Total RTP packets = 2098 (expected 2098) Lost RTP packets = 0 (0.00%) Sequence errors = 0
Duration 62.85 s (0 ms clock drift, corresponding to 1 Hz (+0.00%)

Stable stream BW

Help in Wireshark



```
No.   Time      Source           Destination      Protoco
  3 0.000033  192.168.1.2      174.143.213.184  TCP       ] Seq=1
  4 0.000038  192.168.1.2      174.143.213.184  HTTP      t/logo.
  5 0.050359  174.143.213.184  192.168.1.2      TCP       ] Seq=1
  6 0.002441  174.143.213.184  192.168.1.2      TCP       reasse
  7 0.000011  192.168.1.2      174.143.213.184  TCP       ] Seq=7
  8 0.000114  174.143.213.184  192.168.1.2      TCP       reasse
  9 0.000010  192.168.1.2      174.143.213.184  TCP       ] Seq=7
 10 0.000113  174.143.213.184  192.168.1.2      TCP       [TCP segment of a reasse
 11 0.000009  192.168.1.2      174.143.213.184  TCP       54841 > http [ACK] Seq=7
 12 0.049984  174.143.213.184  192.168.1.2      TCP       [TCP segment of a reasse
 13 0.000034  192.168.1.2      174.143.213.184  TCP       54841 > http [ACK] Seq=7
 14 0.002076  174.143.213.184  192.168.1.2      TCP       [TCP segment of a reasse
 15 0.000011  192.168.1.2      174.143.213.184  TCP       54841 > http [ACK] Seq=7
```

```
⊞ Frame 4: 791 bytes on wire (6328 bits), 791 bytes captured (6328 bits)
⊞ Ethernet II, Src: AsustekC_b3:01:84 (00:1d:60:b3:01:84), Dst: Actionte_2f:47:87 (00:26:
⊞ Internet Protocol Version 4
⊞ Transmission Control Protocol, Src Port: 54841 (54841), Dst Port: http (80), Seq: 1, Ac
⊞ Hypertext Transfer Protocol
```

Easily find help in Wireshark-including Sample Captures

Acknowledgements

- Most part of this tutorial was prepared by M.Fahim, G.Succi, and A.Tormasov

Reference

- This tutorial is based on the on the following resources as well as relevant material over the internet.

- https://www.wireshark.org/download/docs/user-guide.pdf
- 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