Ethernet Basics

Paul Ferrill

ATAC/JT4

James.Ferrill.ctr@us.af.mil

Outline - Ethernet Basics

- Number Systems
  - Hex
  - Binary
- IP – RFC 791
- TCP – RFC 793
- UDP – RFC 768
- Port Numbers
- Datagram
- Frame

Outline - Hardware

- Switch
- Router
- Bridge
- Cabling
- Network Interface Card (NIC)

Outline – Free Tools

- HEXDUMP
- TCPDUMP
  - Basic operation
  - File format
  - Time
- Command line tools
- Chapter 10 Packet Viewer

Outline - Wireshark

- PCAP file contents
- Launch screen
- Time
- Capture options
- Real time vs offline
- Filtering
- Coloring
- Navigation
- Howto

Terminology

Glossary

- IEEE – Institute for Electrical and Electronic Engineers – standards
- CIDR – Classless Interdomain Routing
- IP – Internet Protocol
- RFC – Request for Comment – Internet standards
- TCP – Transmission Control Protocol
- UDP – User Datagram Protocol
- MTU – Maximum Transmission Unit
- PDU – Protocol Data Unit
  - Layer 1 PDU = bit
  - Layer 2 PDU = frame
  - Layer 3 PDU = packet
  - Layer 4 PDU = segment for TCP or datagram for UDP

Units

- Bit = 0 or 1 – smallest unit of information
- Nibble = 4 bits
- Byte = 8 bits (typically) – smallest addressable unit
- Octet = 8 bits by definition (Octal = base 8)
- Word = 16 bits
- Long = 32 bits

Number Systems

- Binary
  - Base 2
    - 0 or 1
    - 1001 in binary = 9 in decimal
- Byte = 8 bits or 2 hex digits
  - Range from 0 to 255 decimal or 0 to FF in hex
- Hexadecimal or hex
  - Base 16
    - 0 – 9 then a, b, c, d, e, f
    - ffff in hex = 1111 1111 1111 1111 in binary or 65535 in decimal

Converting between different bases

MSB															LSB
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Hex													Octal		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	3			6			2			4			5		
0	0	1	1	1	1	0	0	1	0	1	0	0	1	0	1
3				C				A				5			

Windows 10 Calculator

Radix to display

Word size to display

Binary operations

Internet Standards

- Internet Engineering Task Force (IETF)
  - www.ietf.org
- Request for Comment (RFC)
  - www.ietf.org/rfc.html
- Network byte order
  - Defined as big endian in RFC 1700

Definitions

- Header
  - Some number of bytes at the beginning of a file or packet which define the information
- Packet
  - A series of bytes containing control information and optionally data (payload)
- Octet
  - Exactly eight bits of information – commonly referred to as a byte
  - Historically used to precisely define the number of bits
  - Used throughout the RFC documents

Bit ordering or endianness

- Ordering of multi-byte data values
- Big endian vs little endian (http://www.ietf.org/rfc/ien/ien137.txt)
  - Most Significant Byte / Bit (MSB)
  - Least Significant Byte / Bit (LSB)
  - CPU architecture dependent
    - Motorola – big endian
    - Intel – little endian

Little Endian

A little endian number is ordered from the least significant byte (LSB) in a low memory address to the most significant byte (MSB) in a higher memory address.

Address offset	Data
0	byte0
1	byte1
2	byte2
3	byte3

Example

The hex number 0x0D0C0B0A will be represented in memory as shown :

Address offset	Data
0	0A
1	0B
2	0C
3	0D

Big Endian

A big endian number is ordered from the most significant byte (MSB) in a low memory address to the least significant byte (LSB) in a higher memory address.

Address offset	Data
0	byte3
1	byte2
2	byte1
3	byte0

Example

The hex number 0x0D0C0B0A will be represented in memory as shown :

Address offset	Data
0	0D
1	0C
2	0B
3	0A

Convert a 32-bit integer between big / little

```
unsigned long EndianSwap32(unsigned long x)
{
   unsigned long y=0;
   y += (x & 0x000000FF)<<24;
   y += (x & 0x0000FF00)<<8;
   y += (x & 0x00FF0000)>>8;
   y += (x & 0xFF000000)>>24;
   return y;
}
```

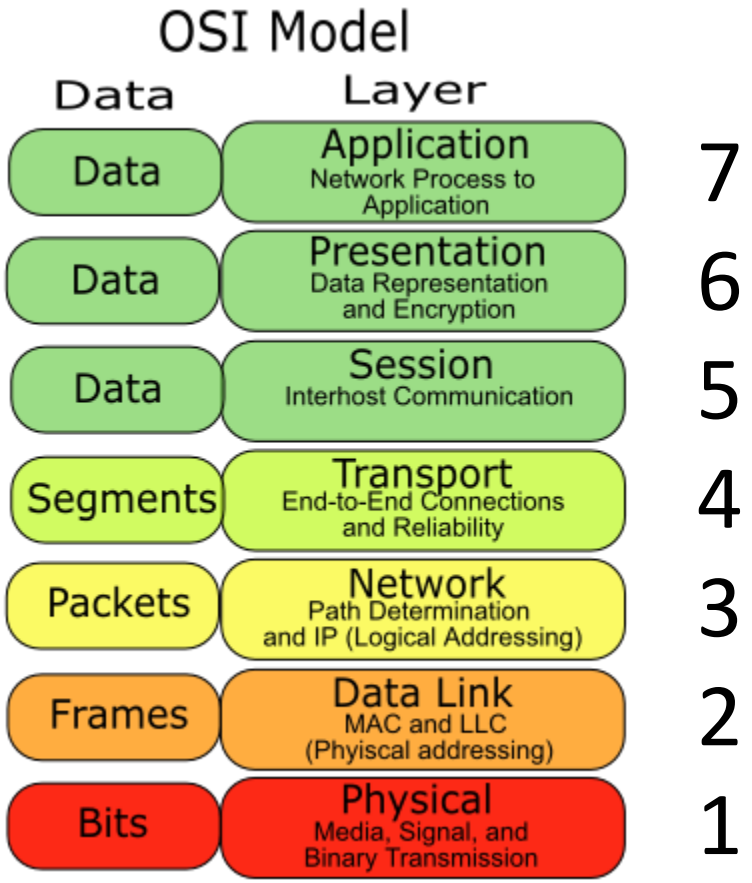
Questions?

Ethernet Basics

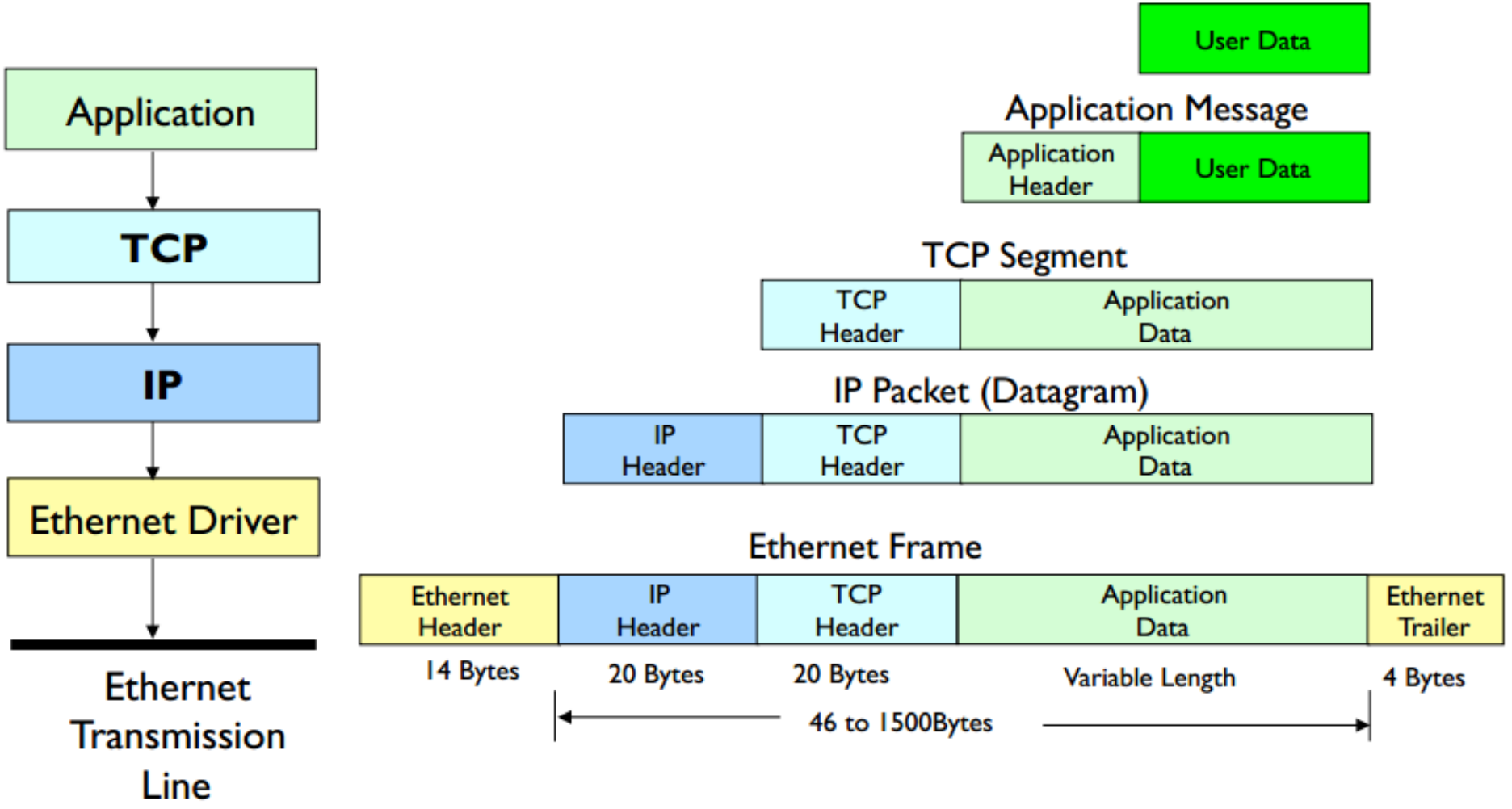
Overview

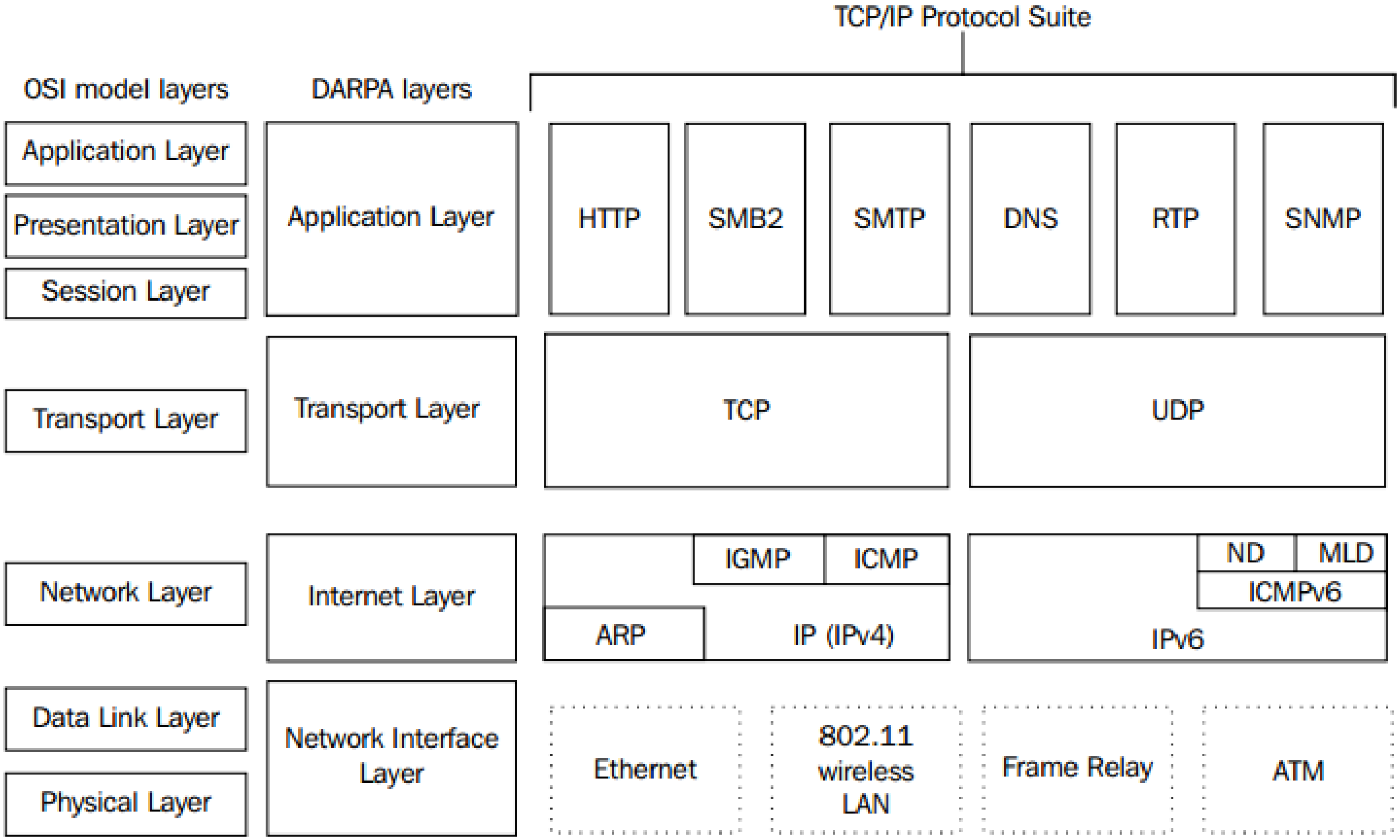
- IP
- TCP
- Private Address Ranges
- TCP/IP Tools
- Symbolic Name Translation
- Routers and Firewalls
- Analyzers

OSI Model

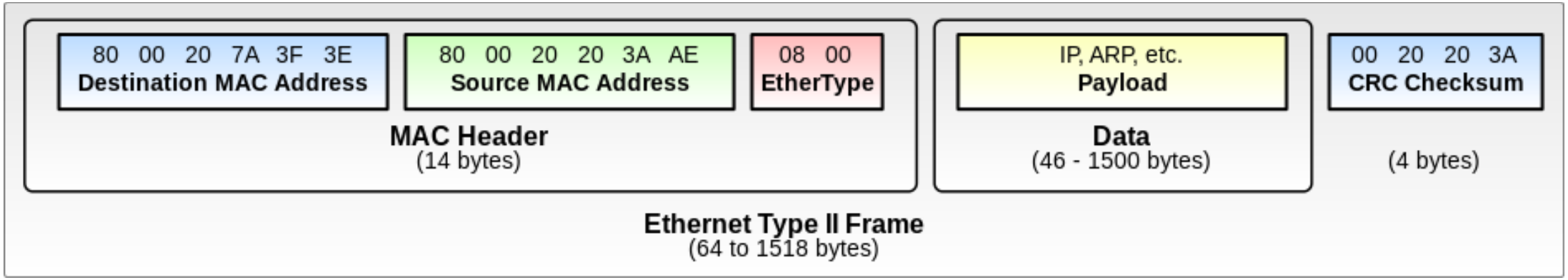

OSI Model
Data
Layer
Data
Application
Network Process to Application
7
Data
Presentation
Data Representation and Encryption
6
Data
Session
Interhost Communication
5
Segments
Transport
End-to-End Connections and Reliability
4
Packets
Network
Path Determination and IP (Logical Addressing)
3
Frames
Data Link
MAC and LLC (Phyiscal addressing)
2
Bits
Physical
Media, Signal, and Binary Transmission
1

Data in Layers




TCP/IP Protocol Suite
OSI model layers
DARPA layers
Application Layer
Presentation Layer
Session Layer
Application Layer
HTTP
SMB2
SMTP
DNS
RTP
SNMP
Transport Layer
Transport Layer
TCP
UDP
Network Layer
Internet Layer
IGMP
ICMP
ARP
IP (IPv4)
ND
MLD
ICMPv6
IPv6
Data Link Layer
Physical Layer
Network Interface Layer
Ethernet
802.11 wireless LAN
Frame Relay
ATM

Ethernet Type II Frame



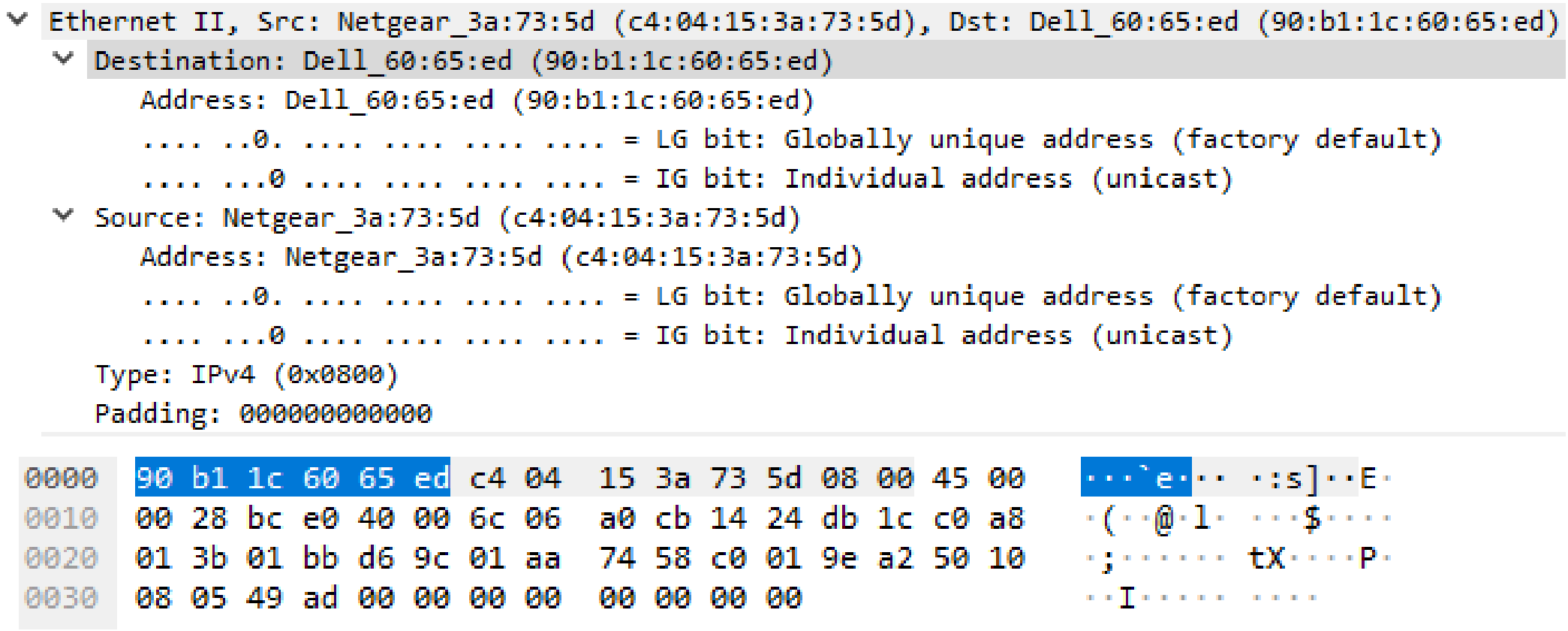
Layer 1 – Physical

- Standards define:
  - Signaling
  - Cabling
  - Connectors
- IEEE 802 is a family of standards covering the Data Link and Physical layer of the OSI networking reference model
- IEEE 802.3 defines Ethernet
- IEEE 802.11 defines Wireless LAN

Layer 2 – Data Link

- The Data Link layer is split into two sub layers
  - Logical Link Control (LLC)
  - Media Access Control (MAC)
- Addressing at this level is hardware unique – MAC address
- Channel access control mechanism
  - Most common is Carrier Sense Multiple Access / Carrier Detect (CSMA/CD) (802.3 standard)
  - Wireless uses CSMA/CA, ALOHA, TDMA, OFDMA
- Layer 2 Protocols
  - L2DP, LLDP, PPP, PPTP
- Layer 2 + 3 Protocols
  - ARP, RARP, SPB, X.25

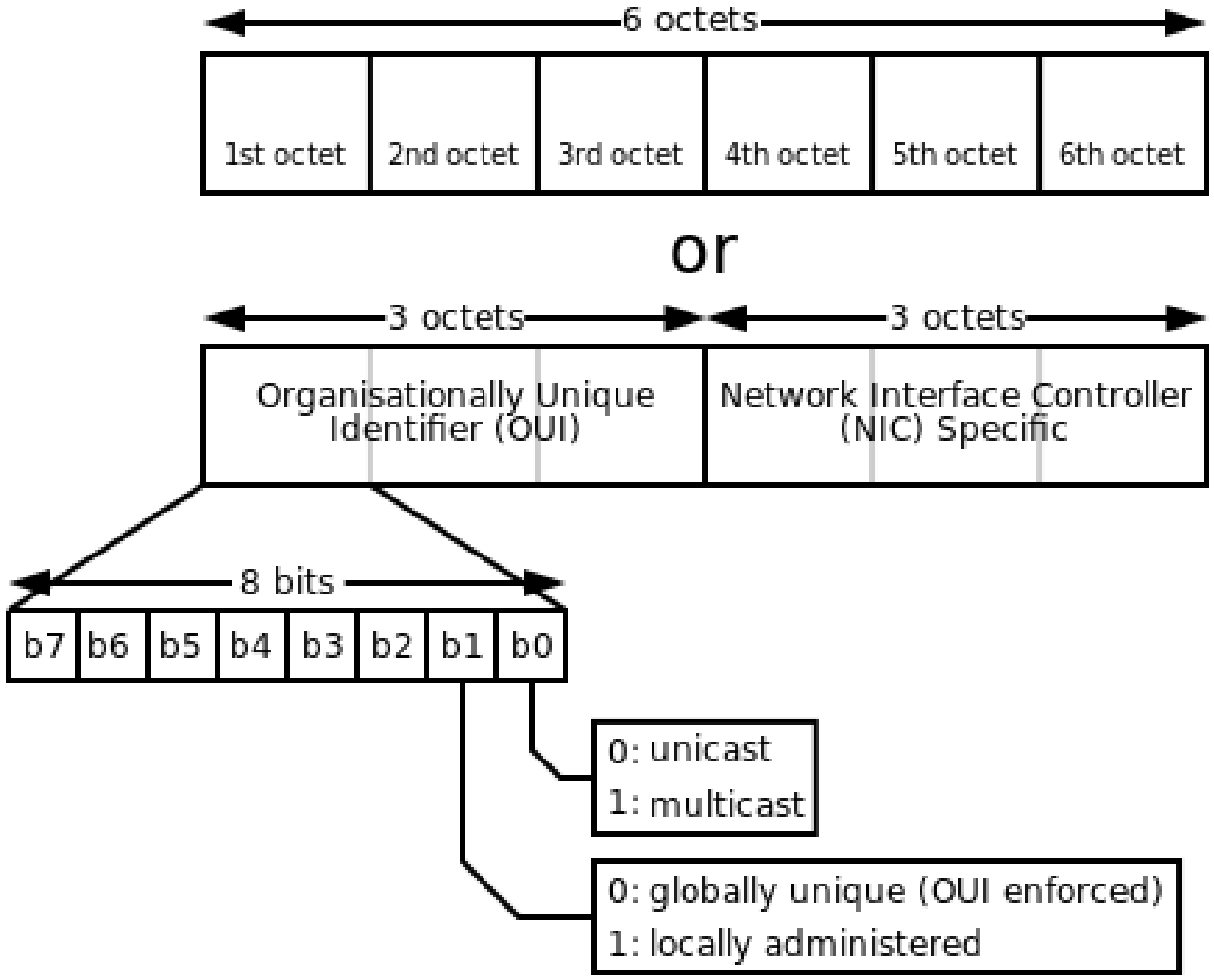
Wireshark data



MAC Addresses

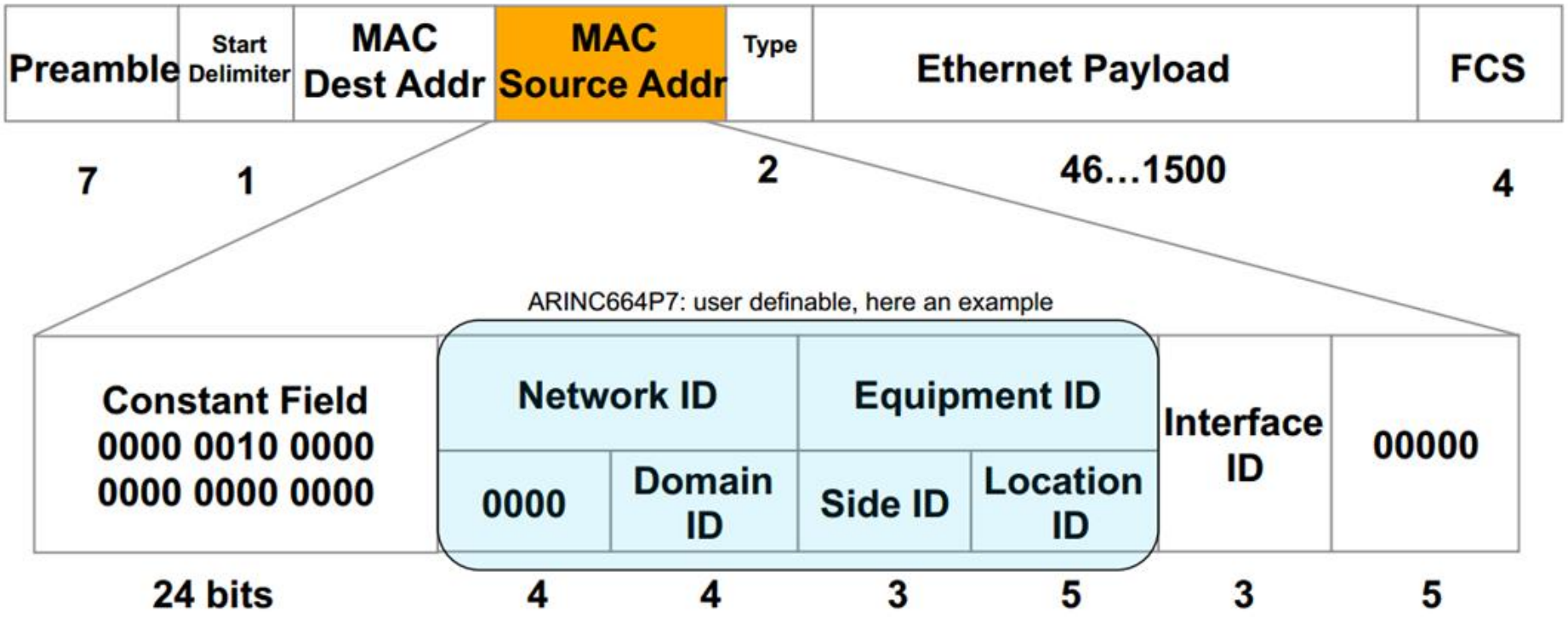
- Six bytes of information
  - 00-1D-92-98-36-8A
- Globally Unique
  - Conflicts not allowed
- First three bytes = OUI = Vendor ID
  - Organizationally Unique Identifier – assigned by IEEE
  - 00:1D:92 = Micro-star International
  - http://aruljohn.com/mac/001D92

Hardware (MAC) Address

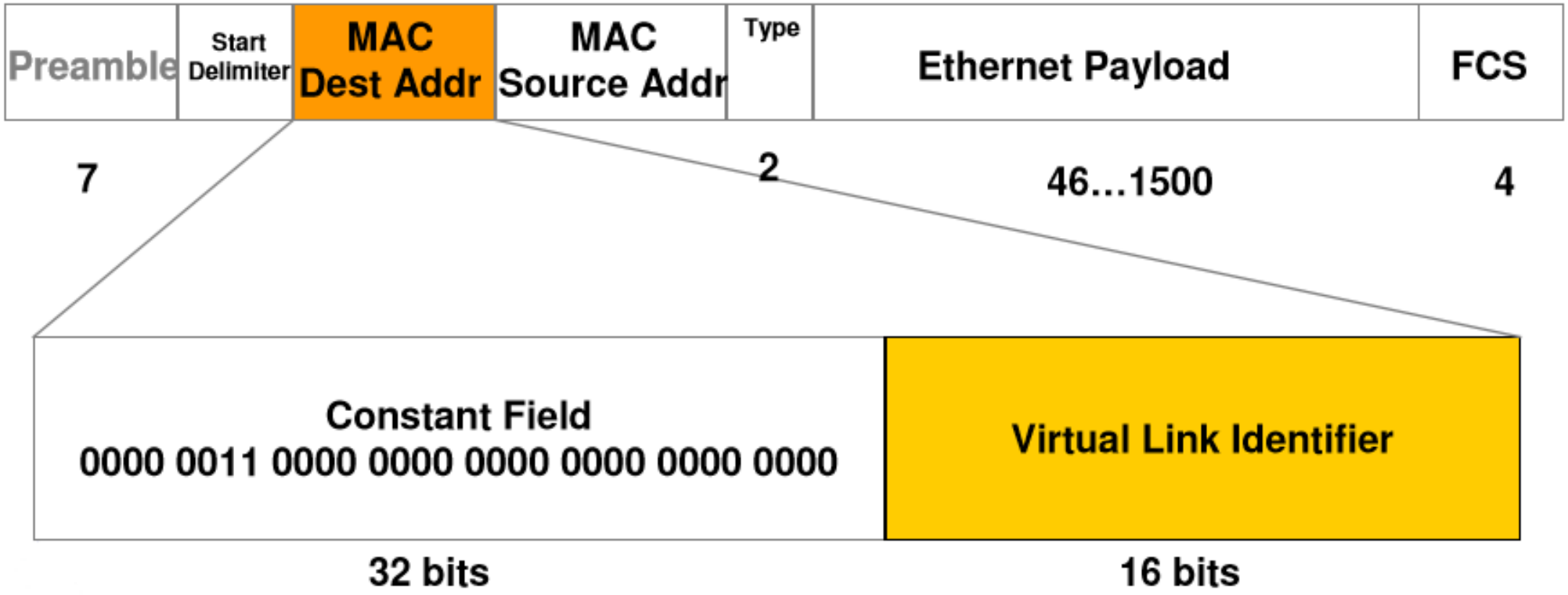


ARINC-664

- MAC Source Address encodes the unique "Source" of the frame



ARINC-664 MAC Destination Address



Layer 3 – Network

- IP lives here
- Protocols
  - ICMP – Internet Control Message Protocol (PING)
  - IGMP – Internet Group Management Protocol
  - IGRP – Interior Gateway Routing Protocol
  - IPv4 / IPv6 – Internet Protocol version 4 / 6
  - IPSec – Internet Protocol Security
  - IPX – Internetwork Packet Exchange
  - NDP – Neighbor Discovery Protocol
  - RIP – Routing Information Protocol

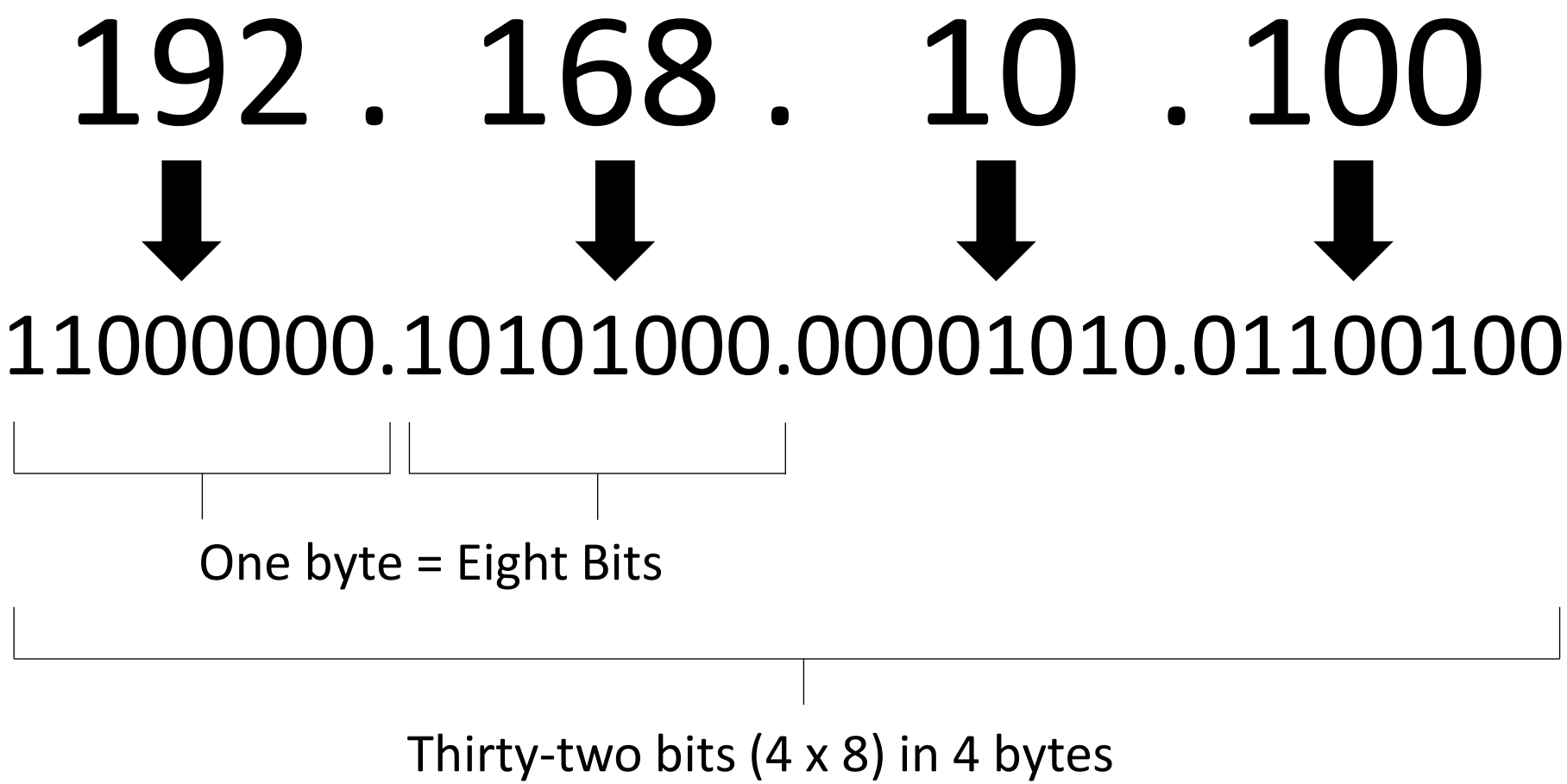
Packet Fundamentals

- IP Header = 24 Bytes
- TCP Header = minimum of 24 Bytes
- UDP Header = 8 Bytes exactly
- Maximum Transmission Unit (MTU) = 1500 bytes
  - Windows defaults to 1480 bytes
- Jumbo Frames
  - 9000-bytes long
  - Goal is to reduce packet overhead
  - CRC-based checksum

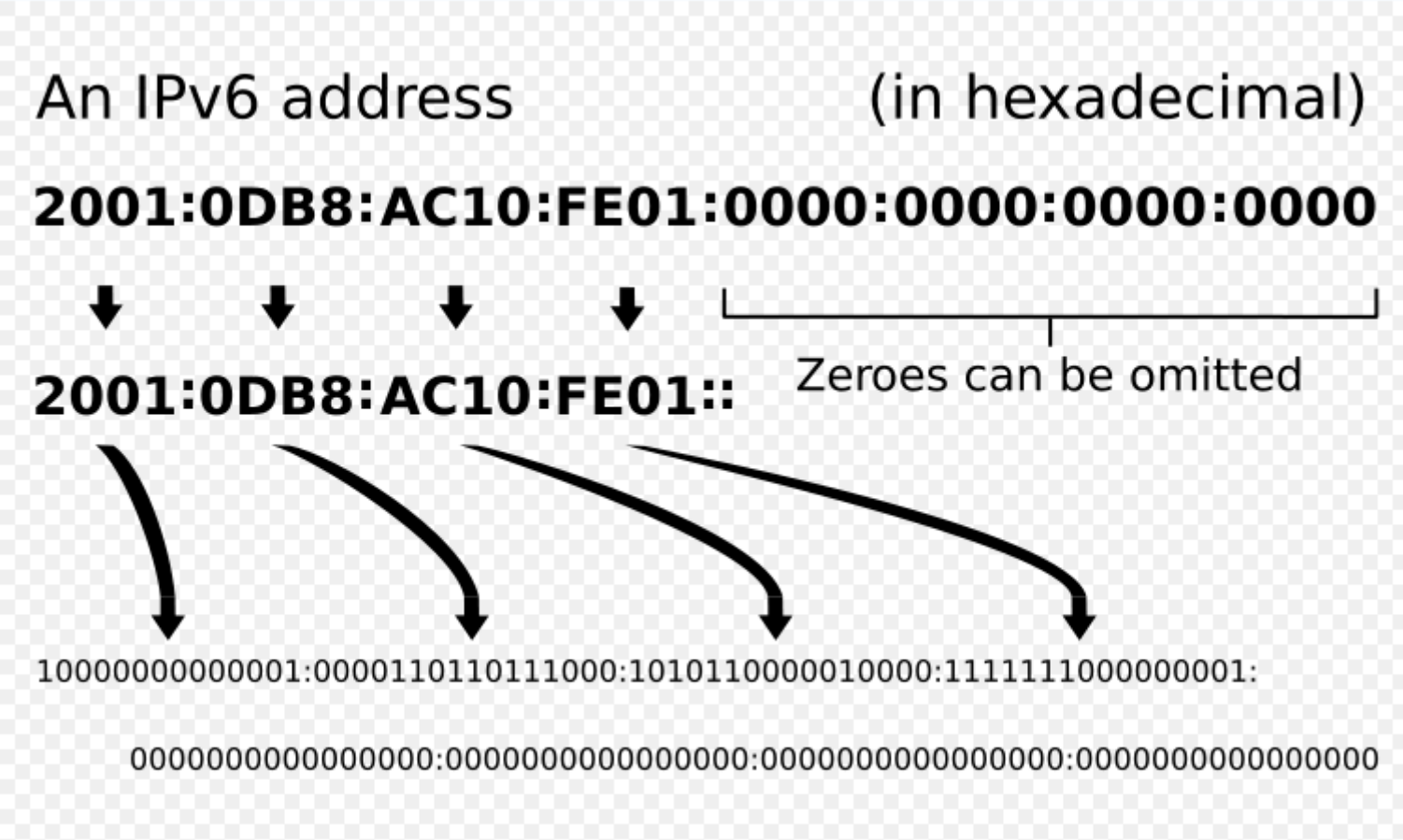
IP Addressing Basics

- IPv4 uses 32-bit addresses
- Class A (24-bit), B (20-bit) and C (16-bit)
- IPv4 addresses reserved in RFC 1918
- Dotted-decimal notation 192.168.1.1
- IPv6 uses 128-bit addresses
- IPv6 addresses reserved in RFC 4193
- Last octet may not be 0 or 255
  - 0 used for network identifier
  - 255 = broadcast address

IPv4 Addressing Details



IPv6 Addressing Details



IP – Internet Protocol

- Datagram
  - Send it let it rattle around to its destination
  - If it takes too long throw it away
  - Address Format (V4)
    - 192.168.0.188 4 Octets
- Sits on top of a Data Link Protocol
  - Ethernet
    - MAC Address Allocated by Card Manufacturer
      - http://aruljohn.com/mac.pl
  - But could be
      - IEEE 802-2, Token Ring, FDDI, SMDS,SDLC, LAPB, etc.

IP V4 Packet Format

<table>
<tr><td>Version</td><td>Header Length</td><td>Differentiated Services</td><td>Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Fragment Info</td></tr>
<tr><td colspan="2">Time to Live</td><td>Protocol</td><td>Header Checksum</td></tr>
<tr><td colspan="4">Source Address</td></tr>
<tr><td colspan="4">Destination Address</td></tr>
<tr><td colspan="4">Multiple 32 bit words of “Options”</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>

Wireshark data

```
Internet Protocol Version 4, Src: 20.36.219.28, Dst: 192.168.1.59
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      0000 00.. = Differentiated Services Codepoint: Default (0)
      .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 40
    Identification: 0xbce0 (48352)
  Flags: 0x4000, Don't fragment
      0... .... .... .... = Reserved bit: Not set
      .1.. .... .... .... = Don't fragment: Set
      ..0. .... .... .... = More fragments: Not set
      ...0 0000 0000 0000 = Fragment offset: 0
    Time to live: 108
    Protocol: TCP (6)
    Header checksum: 0xa0cb [validation disabled]
    [Header checksum status: Unverified]
    Source: 20.36.219.28
    Destination: 192.168.1.59
```

```
0000  90 b1 1c 60 65 ed c4 04  15 3a 73 5d 08 00 45 00    ···`e···  ·:s]··E·
0010  00 28 bc e0 40 00 6c 06  a0 cb 14 24 db 1c c0 a8    ·(··@·l·  ···$····
0020  01 3b 01 bb d6 9c 01 aa  74 58 c0 01 9e a2 50 10    ·;······  tX····P·
0030  08 05 49 ad 00 00 00 00  00 00 00 00               ··I·····  ····
```

IP Packet Header Details

- Version = 4 for IPv4
- Header length = number of 32-bit words in header
  - Min length = 5 words or 20 bytes
  - Max length = 15 words if all options present
- Header length can be used as an offset from the start of the header to the beginning of data
- Time to Live actually a hop count which is decremented by each gateway
- Identification – unique number for entire datagram – used to reassemble fragments

IP Packet Header Details (cont)

- Protocol
  - ICMP = 1
  - IGMP = 2
  - TCP = 6
  - UDP = 17
- Address
  - 32-bits with each octet representing one of four digits in address

IP Address Aspects

- The IP Address applies to a connection not a host a
- Two pieces of address identify subnet and host using mask or CIDR
- "Networks" and Subnets
  - Conceptual Class A, B, C
  - Actual implementation is Subnets
    - Defined by Subnet Mask 255.255.255.0
    - Works with IP Address
- Network Address Translation (NAT)
  - Routable address for public IP
  - Non-routable address behind firewall
  - http://en.Wikipedia.org/wiki/Private_network

Classless Interdomain Routing - CIDR

- The use of variable-length subnet masks to allow arbitrary length prefixes.
- Notation uses base address followed by the number of bits as in 192.168.1.0/24 which equates to a mask of 255.255.255.0.
- 192.168.78.0/23 would include both .78 and .79.
- CIDR boundaries must line up with class boundaries. Example is 192.168.78.0/22 crosses the boundary at .80.
- Binary 11111111.11111111.11111100.00000000 would translate in decimal to 255.255.252.0

Private IP Address

- Private IP Address Ranges (non-routable)
  - 10.0.0.0 to 10.255.255.255
  - 172.16.0.0 to 172.31.255.255
  - 192.168.0.0 to 192.168.255.255
- Gateway provides Address Translation (and other fire wall services)
  - Typically home router or Gateway Computer at .1 or .254 address
  - ISP provides global (WAN) IP address
  - For outgoing traffic NAT maintains a cross reference table
  - Incoming traffic must have handling rules (Port forwarding)

Automatic Private IP Addressing

- Defined in RFC 3927
  - Dynamic Configuration of of IPv4 Link-Local Addresses

  "This document describes how a host may automatically configure an interface with an IPv4 address within the 169.254/16 prefix that is valid for communication with other devices connected to the same physical (or logical) link."
- In the absence of a DHCP service an address in the 169.254/16 range may be assigned.
- Bonjour is Apple's implementation of RFC 3927
- Linux uses Avahi which implements the Apple Zeroconf specification

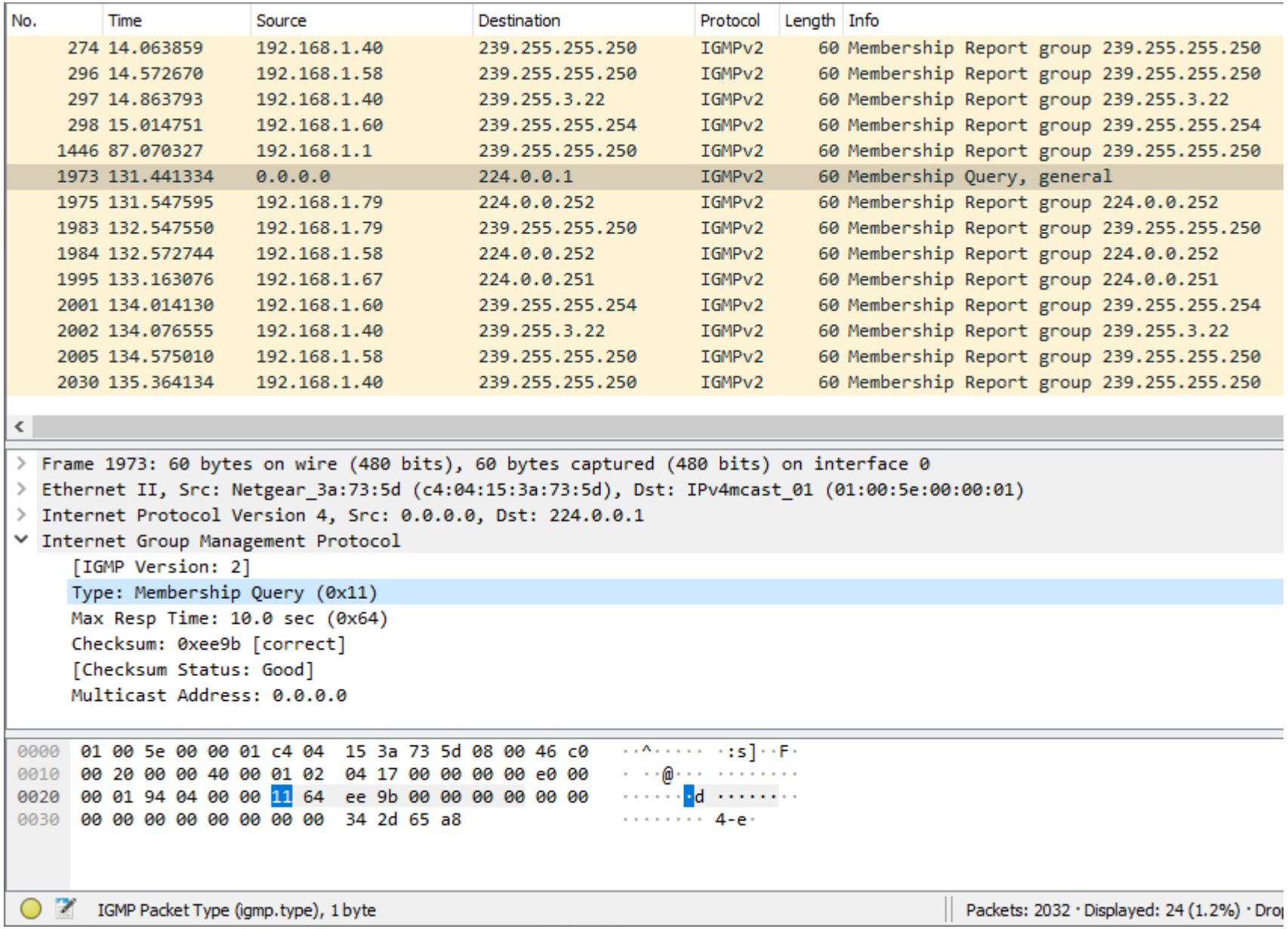
IP Multicast

- Definition from Wikipedia “a method of sending Internet Protocol (IP) datagrams to a group of interested receivers in a single transmission.”
- “It is a form of point-to-multipoint communication employed for streaming media and other applications on the Internet and private networks.”
- “IP multicast is the IP-specific version of the general concept of multicast networking. It uses specially reserved multicast address blocks in IPv4 and IPv6.”

Multicast IP Addresses

- Reserved range 224.0.0.0 to 239.255.255.255
- Well know addresses use 224.0 prefix
  - IGMPv3 uses 224.0.0.22
  - LLMNR uses 224.0.0.252 (Link Local Multicast Name Resolution)
  - PTP uses 224.0.0.107
  - NTP clients listen on 224.0.1.1
  - Zeroconf mDNS uses 224.0.0.251
- Ethernet multicast MAC addresses
  - FF:FF:FF:FF:FF:FF for broadcast
  - 01:80:C2:00:00:00, :03, :0E for Link Layer Discovery Protocol (LLDP)

Wireshark data



No.	Time	Source	Destination	Protocol	Length	Info
274	14.063859	192.168.1.40	239.255.255.250	IGMPv2	60	Membership Report group 239.255.255.250
296	14.572670	192.168.1.58	239.255.255.250	IGMPv2	60	Membership Report group 239.255.255.250
297	14.863793	192.168.1.40	239.255.3.22	IGMPv2	60	Membership Report group 239.255.3.22
298	15.014751	192.168.1.60	239.255.255.254	IGMPv2	60	Membership Report group 239.255.255.254
1446	87.070327	192.168.1.1	239.255.255.250	IGMPv2	60	Membership Report group 239.255.255.250
1973	131.441334	0.0.0.0	224.0.0.1	IGMPv2	60	Membership Query, general
1975	131.547595	192.168.1.79	224.0.0.252	IGMPv2	60	Membership Report group 224.0.0.252
1983	132.547550	192.168.1.79	239.255.255.250	IGMPv2	60	Membership Report group 239.255.255.250
1984	132.572744	192.168.1.58	224.0.0.252	IGMPv2	60	Membership Report group 224.0.0.252
1995	133.163076	192.168.1.67	224.0.0.251	IGMPv2	60	Membership Report group 224.0.0.251
2001	134.014130	192.168.1.60	239.255.255.254	IGMPv2	60	Membership Report group 239.255.255.254
2002	134.076555	192.168.1.40	239.255.3.22	IGMPv2	60	Membership Report group 239.255.3.22
2005	134.575010	192.168.1.58	239.255.255.250	IGMPv2	60	Membership Report group 239.255.255.250
2030	135.364134	192.168.1.40	239.255.255.250	IGMPv2	60	Membership Report group 239.255.255.250

```
> Frame 1973: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
> Ethernet II, Src: Netgear_3a:73:5d (c4:04:15:3a:73:5d), Dst: IPv4mcast_01 (01:00:5e:00:00:01)
> Internet Protocol Version 4, Src: 0.0.0.0, Dst: 224.0.0.1
v Internet Group Management Protocol
    [IGMP Version: 2]
    Type: Membership Query (0x11)
    Max Resp Time: 10.0 sec (0x64)
    Checksum: 0xee9b [correct]
    [Checksum Status: Good]
    Multicast Address: 0.0.0.0
```

```
0000  01 00 5e 00 00 01 c4 04  15 3a 73 5d 08 00 46 c0   ··^····· ·:s]··F·
0010  00 20 00 00 40 00 01 02  04 17 00 00 00 00 e0 00   · ··@··· ········
0020  00 01 94 04 00 00 11 64  ee 9b 00 00 00 00 00 00   ·······d ········
0030  00 00 00 00 00 00 00 00  34 2d 65 a8               ········ 4-e·
```

IGMP Packet Type (igmp.type), 1 byte | Packets: 2032 · Displayed: 24 (1.2%) · Dro

Address Resolution Protocol

- In IPv4 ARP is used to map IP network addresses to specific hardware addresses used by a data link protocol.
- Gratuitous ARP is used when a host chooses an IP address and then issues a query to make sure it does not conflict with another host.
- An ARP table on a host holds all known MAC / IP address pairs.
- ARP is also a program in both Linux and Windows used to display or modify the ARP table.

Wireshark data

No.	Time	Source	Destination	Protocol	Length	Info
61	40.981711	Raspberr_53:18:e9	Broadcast	ARP	60	Who has 169.254.137.172? Tell 0.0.0.0
72	42.671578	Raspberr_53:18:e9	Broadcast	ARP	60	Who has 169.254.137.172? Tell 0.0.0.0
73	44.473505	Raspberr_53:18:e9	Broadcast	ARP	60	Who has 169.254.137.172? Tell 0.0.0.0
75	46.476545	Raspberr_53:18:e9	Broadcast	ARP	60	Gratuitous ARP for 169.254.137.172 (Request)
83	48.477756	Raspberr_53:18:e9	Broadcast	ARP	60	Gratuitous ARP for 169.254.137.172 (Request)
87	64.003860	Raspberr_53:18:e9	Broadcast	ARP	60	Who has 192.168.22.1? Tell 169.254.137.172
89	65.031147	Raspberr_53:18:e9	Broadcast	ARP	60	Who has 192.168.22.1? Tell 169.254.137.172
90	66.072731	Raspberr_53:18:e9	Broadcast	ARP	60	Who has 192.168.22.1? Tell 169.254.137.172

```
> Frame 61: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
v Ethernet II, Src: Raspberr_53:18:e9 (b8:27:eb:53:18:e9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
    > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    > Source: Raspberr_53:18:e9 (b8:27:eb:53:18:e9)
      Type: ARP (0x0806)
      Padding: 000000000000000000000000000000000000
v Address Resolution Protocol (request)
      Hardware type: Ethernet (1)
      Protocol type: IPv4 (0x0800)
      Hardware size: 6
      Protocol size: 4
      Opcode: request (1)
      Sender MAC address: Raspberr_53:18:e9 (b8:27:eb:53:18:e9)
      Sender IP address: 0.0.0.0
      Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
      Target IP address: 169.254.137.172
```

Layer 4 - Transport

- TCP and UDP live here
- Also where encapsulation happens
  - GRE – Generic Routing Encapsulation for tunneling
- At this layer the data can be either connection oriented (TCP) or connectionless (UDP)
- A host operating system typically provides all services related to this layer
  - For a TCP connection the OS would handle all retransmit requests and return error status to the calling routine

TCP and UDP Port Numbers

- Destination Port # is the "application" or "service" address on the host
  - Applications/services register to listen for incoming data on the defined port
  - IANA port numbers: http://www.iana.org/assignments/port-numbers
  - 0 to 1023 Well Known ports managed by IANA
  - 1024 to 49151 Registered by IANA as a convenience
  - 49152 to 65535 Dynamic (used for source address)
  - C:\WINDOWS\system32\drivers\etc\services
  - Source Port number used with IP addresses and destination port number to create a unique identifier for the connection
  - Source port number incremented at each use in dynamic case

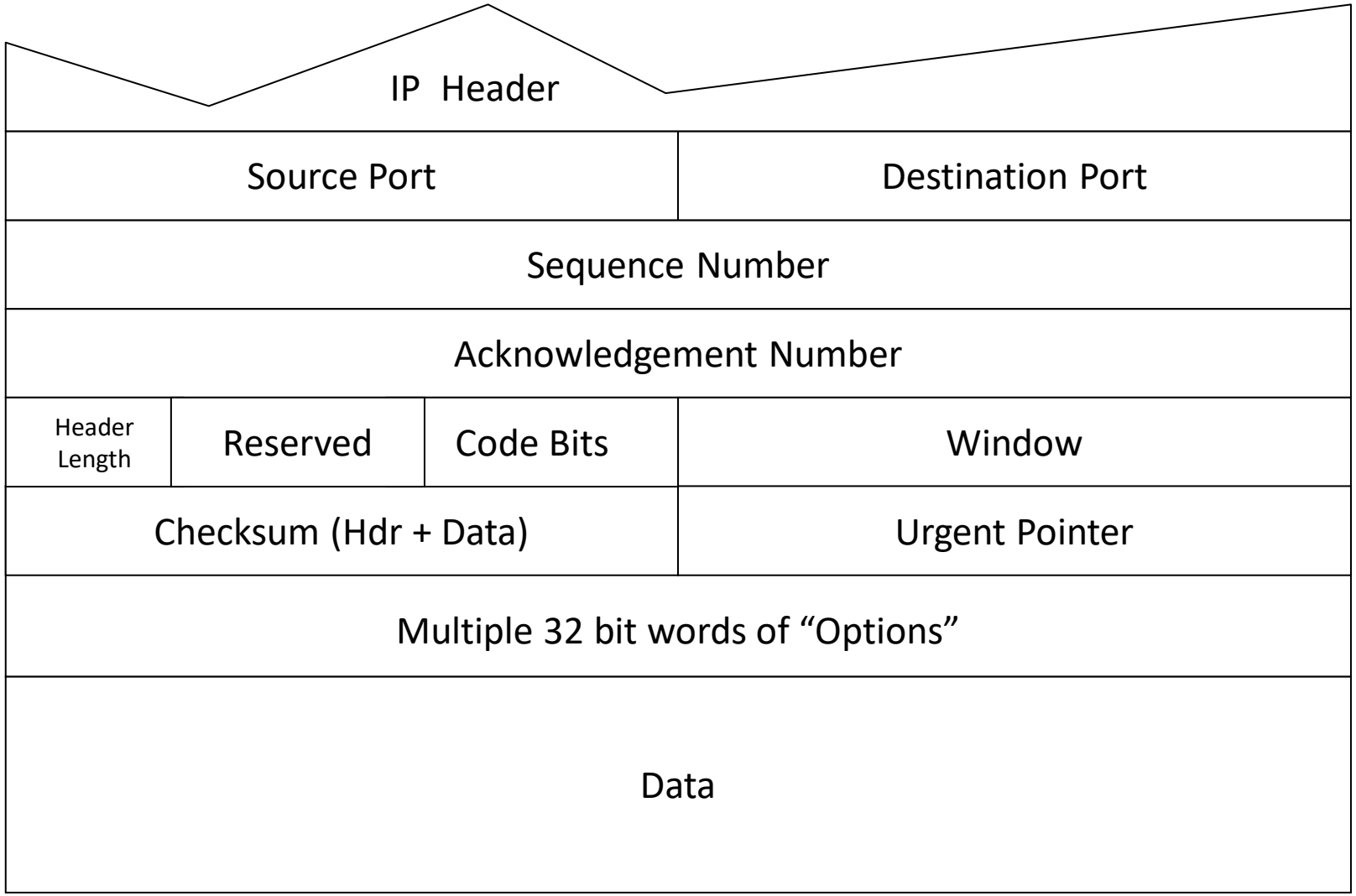
TCP

- Transmission Control Protocol
- Described in RFC 793
- Highly reliable
- Connection oriented
- Error detection through checksum
- ACK / NAK

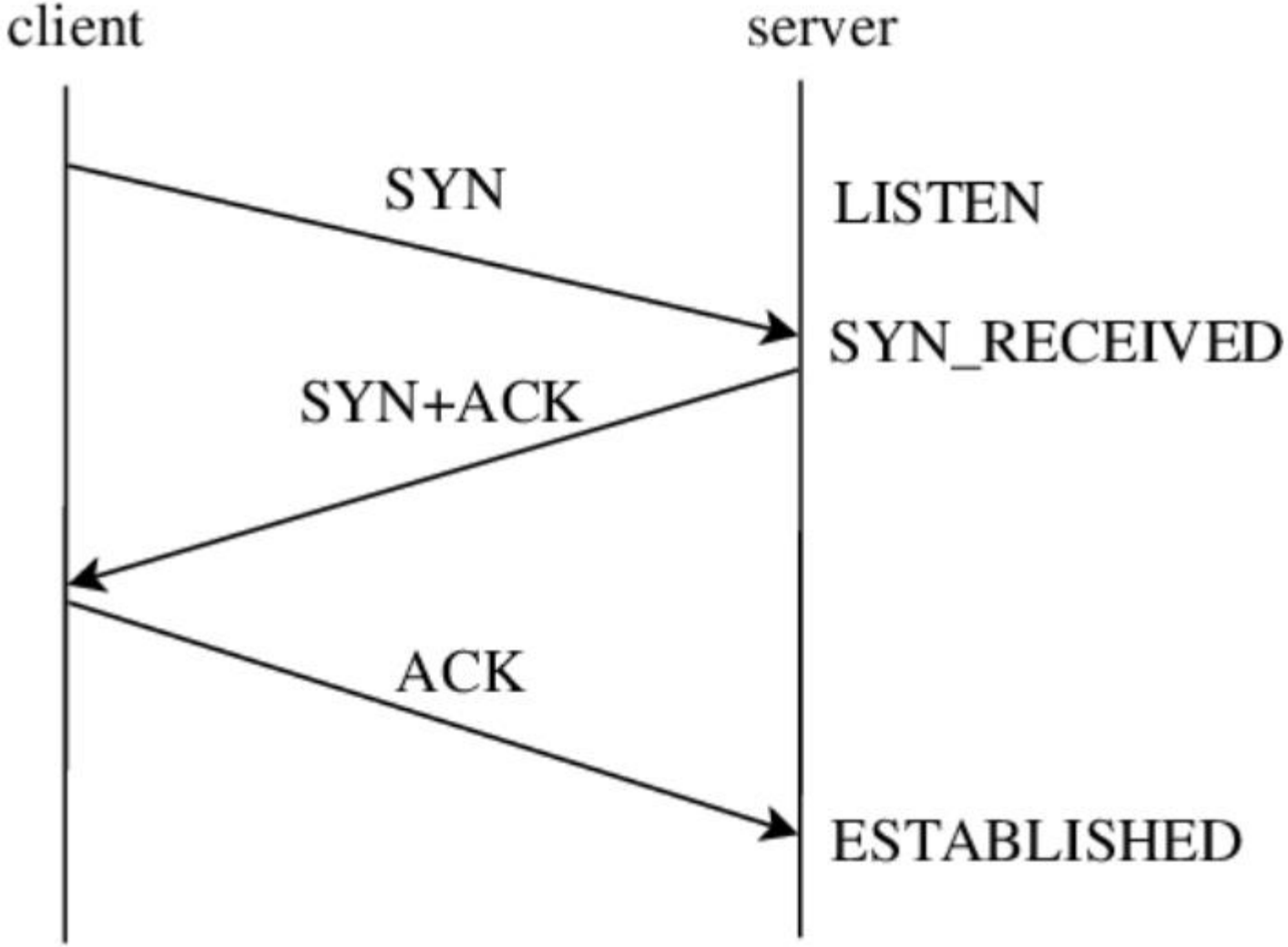
TCP Distinctions

- Ordered data transfer – sequence number used to reassemble packets
- Retransmission of lost packets – not acknowledged packets resent
- Error-free data transfer – checksum used to ensure reliable transfer
- Flow control – limits transfer rate to ensure reliable delivery
- Congestion control

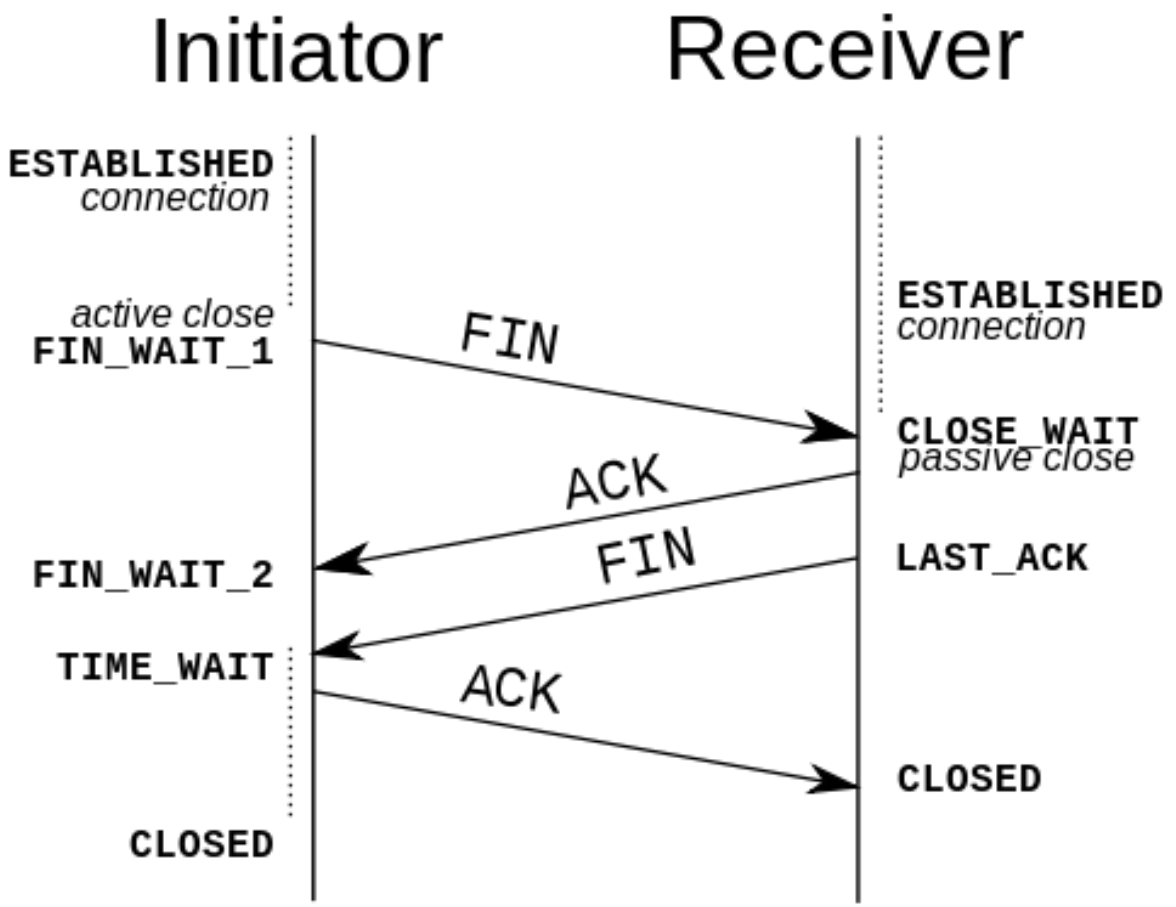
TCP PDU Format



Connection Establishment Diagram



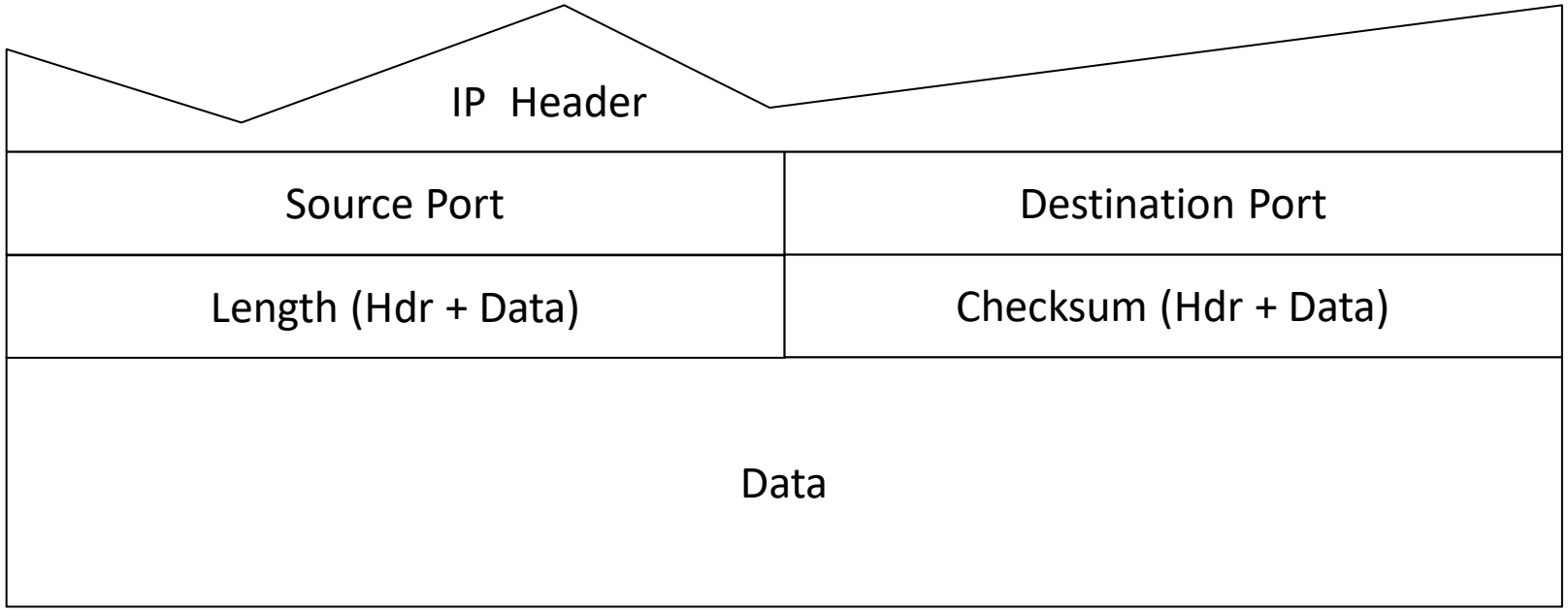
Connection Termination Diagram



UDP

- User Datagram Protocol
- Described in RFC 768
- Minimal overhead
- Transaction oriented
- Delivery and duplicate protection not guaranteed
- Stateless by design suitable for large numbers of clients
- Supports multicast for service discovery and information sharing

UPD PDU Format



UDP Traffic

- DNS
- SNMP – Simple Network Management Protocol
- Video / Audio streaming
- Broadcast – uses .255 in final octet of address
- Unicast – between two computers
- Multicast – from one to many

UDP Multicast

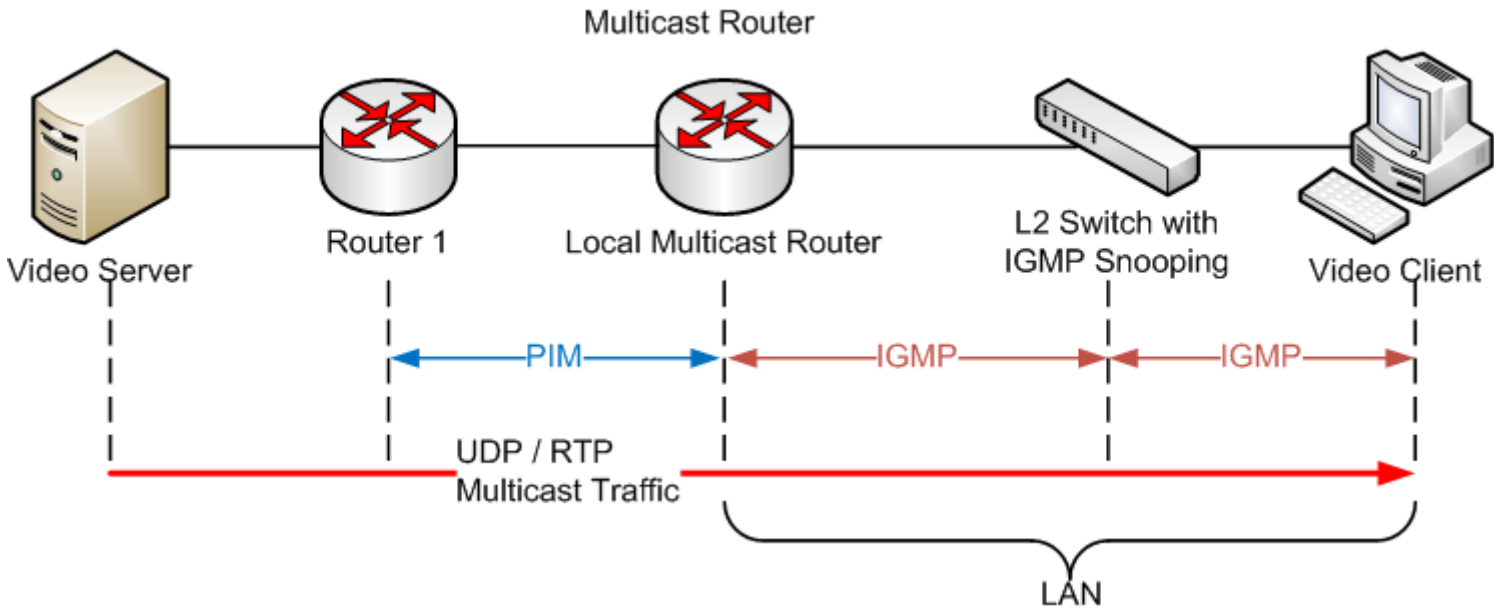
- The group includes the addresses from 224.0.0.0 to 239.255.255.255
- Addresses in the range of 224.0.0.0 to 224.0.0.255 are individually assigned by IANA and designated for multicasting on the local subnetwork only.
- Addresses in the range 224.0.1.0 to 224.0.1.255 are individually assigned by IANA and designated the Internetwork Control Block. Network Time Protocol (224.0.1.1)

Multicast Data Delivery

- By nature it uses UDP as the transport mechanism.
- Unicast packets are delivered to a specific recipient based on the MAC address.
- Broadcast packets use the broadcast MAC address of FF:FF:FF:FF:FF:FF
- IGMP used to join a multicast group.

IGMP

- Operates between a client computer and router
- IGMP snooping used to build maps of multicast streams
- Protocol Independent Multicast (PIM)
- Real-time Transport Protocol (RTP) RFC 3550
- Operates at the network layer (3)
- IP Protocol number = 2
- IGMPv2 messages include:
  - General query
  - Group-specific query
  - Membership report
  - Leave group
- IGMPv3 adds more messages



IGMP Traffic

- **Querier** – sends out messages asking devices connected to its network segments which devices are members of specific multicast groups
- **Receiver** – receives multicast traffic destined for a specific multicast address. Can be a client device or router, which then forwards the data on to other hosts and routers
- IGMP v1 uses 224.0.0.1 as a general query address
- IGMP v2 uses 224.0.0.2 as a general query address

Routing

- Routing is the act of moving information across an internetwork from source to destination. Along the way, at least one intermediate node typically is encountered. Routing occurs at Layer 3 (the network layer) of the OSI reference model.
- Routing algorithms
  - OSPF is the most common interior gateway protocol (IGP)
  - OSPF V2 defined in RFC 2328 for IPv4
  - OSPF V3 defined in RFC 5340 updated for IPv6
- Routing Information Protocol (RIP)
  - RFCs 1058, 1388, 1723

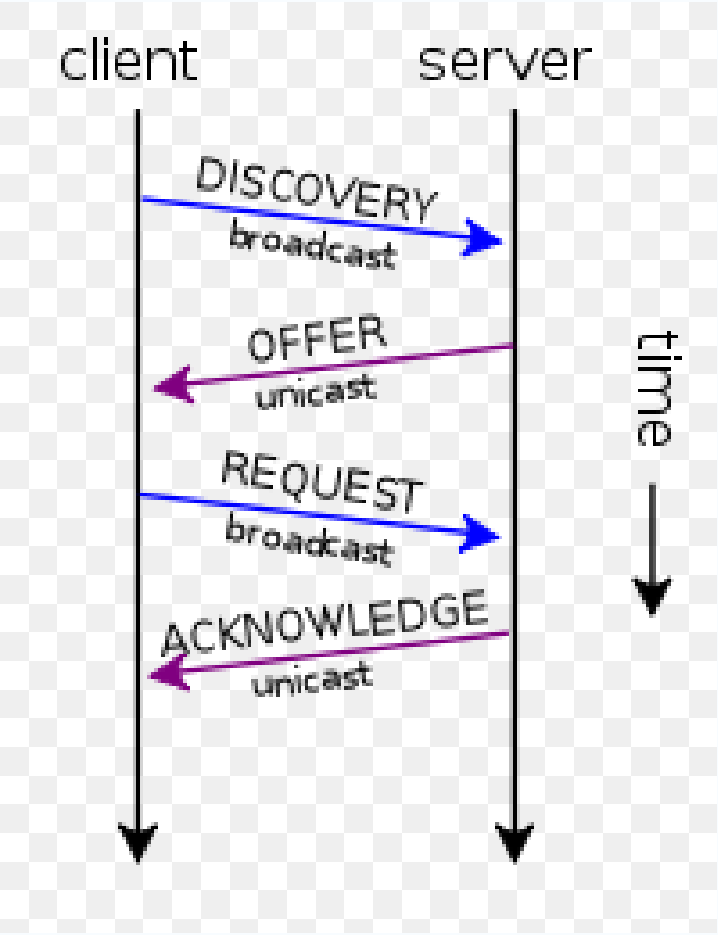
Domain Name System (DNS)

- Essentially a global phone book for the Internet
- Translates friendly names into IP addresses
- Original RFCs published in 1983 (882, 883)
- RFCs 1034, 1035 published in 1987 superseded previous versions
- Naming rules in RFCs 1035, 1123 and 2181
- Queries use UDP over port 53 using format specified in RFC 1035

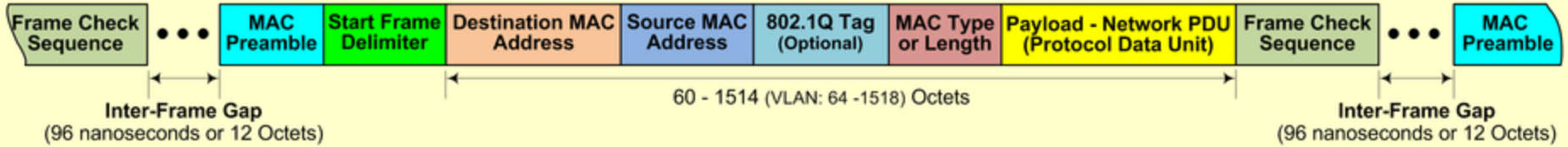
DHCP

- Dynamic Host Control Protocol
- Described in RFC 1531 and RFC 2131
- IPv6 extensions in RFC 3315
- DHCP uses the same two IANA assigned ports as BOOTP: 67/udp for the server side, and 68/udp for the client side.
- Four basic phases: IP discovery, IP lease offer, IP request, and IP lease acknowledgement.

DHCP Sequence



Gigabit Ethernet (IEEE 802.3ab) Frame Structure with TCP/IP Datagram



Gigabit Ethernet Frame Component Size with TCP/IP Datagram

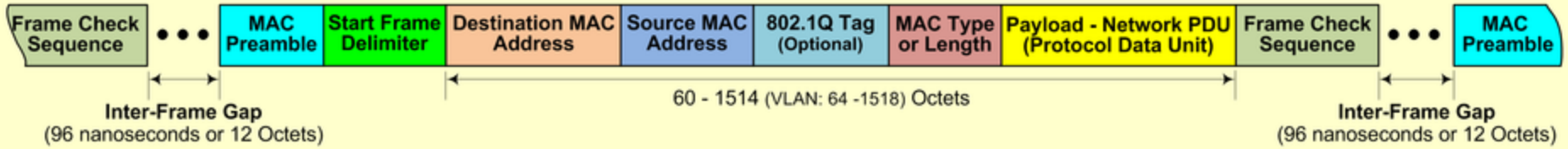
Frame Component	Component Size	
MAC Preamble	7 Octets of: **10101010**	
Start Frame Delimiter	1 Octet of: **10101011**	
Destination MAC Address	6 Octets	
Source MAC Address	6 Octets	
802.1Q VLAN TAG ID (Optional)	4 Octets (Optional)	
MAC Type or Length	2 Octets	
MTU (Maximum Transmission Unit) / **Payload** Network PDU Protocol Data Unit: (Packet / Segment)	**IP Header**	20 Octets
	TCP Header	20 Octets
	TCP Options/ Data/Padding	6 - 1460 Octets
	***Total:**	**46 - 1500 Octets** (Max: 1504 – VLAN)
Frame Check Sequence (CRC)	4 Octets	
Inter-Frame Gap • • •	12 Octets (96 nanoseconds)	
Total Physical Frame Size:	**84 – 1538 Octets** (Max: 1544 -VLAN)	

Gigabit Ethernet Maximum Frame and Data Throughput Rate Calculation with TCP/IP Datagram

Rate Term	Value
Gigabit Ethernet Bit Rate	1000 Mbit/sec -or- 1000Mb/sec
Gigabit Ethernet Bit Time	1 nanosecond (.000000001 seconds)
1 Octet (Byte)	8 Bits
Max Octet Rate	(1000Mb/sec)/((8 Bits) = **125,000,000 Octets/sec**
Max Frame Rate (84 Octet Frames) Min Packet (60 Bytes + 4 Bytes CRC)	(1000Mb/sec)/((8 Bits)*(84 Octets/Frame)) = **1,488,095 Frames/sec (FPS)**
Max TCP/IP Data Rate (84 Octet Frames) Min TCP/IP Packet (60 Bytes + 4 Bytes CRC)	(1,488,095 Frames/sec)*(6 Bytes/Frame) = **8,928,571 Bytes/sec**
Max Frame Rate (1538 Octet Frames) Max Packet (1514 Bytes + 4 Bytes CRC)	(1000Mb/sec)/((8 Bits)*(1538 Octets/Frame)) = **81,274 Frames/sec (FPS)**
Max TCP/IP Data Rate (1538 Octet Frames) Max TCP/IP Packet (1514 Bytes + 4 Bytes CRC)	(81,274 Frames/sec)*(1448 Bytes/Frame) = **117,685,306 Bytes/sec** (TCP/IP TimeStamp)
Max TCP/IP Data Rate (1538 Octet Frames) Max TCP/IP Packet (1514 Bytes + 4 Bytes CRC)	(81,274 Frames/sec)*(1460 Bytes/Frame) = **118,660,598 Bytes/sec** (no TCP/IP TimeStamp)
Max Gigabit Ethernet Frame Bandwidth Max Packet (60 Bytes + 4 Bytes CRC)	(1,488,095 Frames/sec)*(64 Bytes/Frame) = **95,238,080 Bytes/sec** (90.876031 MiB/s)
Max Packet (60 Bytes)	(1,488,095 Frames/sec)*(60 Bytes/Frame) = **89,285,700 Bytes/sec** (85.149477 MiB/s)
Max Gigabit Ethernet Frame Bandwidth Max Packet (1514 Bytes + 4 Bytes CRC)	(81,274 Frames/sec)*(1518 Bytes/Frame) = **123,373,932 Bytes/sec** (117.658550 MiB/s)
Max Packet (1514 Bytes)	(81,274 Frames/sec)*(1514 Bytes/Frame) = **123,048,836 Bytes/sec** (117.348515 MiB/s)

*** **Note 1:** IEEE 802.3ab – Gigabit Ethernet over copper twisted-pair cabling.
*** **Note 2:** Gigabit Ethernet allows for larger MTUs (Jumbo or Super Jumbo Frames).
*** **Note 3:** Units – **M:** 1,000,000 **Mi:** 1,048,576

Gigabit Ethernet (IEEE 802.3ab) Frame Structure with UDP Datagram



Gigabit Ethernet Frame Component Size With UDP Datagram

Frame Component	Component Size	
MAC Preamble	7 Octets of: **10101010**	
Start Frame Delimiter	1 Octet of: **10101011**	
Destination MAC Address	6 Octets	
Source MAC Address	6 Octets	
802.1Q VLAN TAG ID (Optional)	4 Octets (Optional)	
MAC Type or Length	2 Octets	
MTU (Maximum Transmission Unit) — Packet / Segment	IP Header	20 Octets
	UDP Header	8 Octets
Payload Network PDU Protocol Data Unit:	Data/Padding	18 - 1472 Octets
	***Total:	**46 - 1500 Octets** (Max: 1504 – VLAN)
Frame Check Sequence (CRC)	4 Octets	
Inter-Frame Gap • • •	12 Octets (96 nanoseconds)	
Total Physical Frame Size:	**84 – 1538 Octets** (Max: 1544 -VLAN)	

Gigabit Ethernet Maximum Frame and Data Throughput Rate Calculation with UDP Datagram

Rate Term	Value
Gigabit Ethernet Bit Rate	1000 Mbit/sec -or- 1000Mb/sec
Gigabit Ethernet Bit Time	1 nanosecond (.000000001 seconds)
1 Octet (Byte)	8 Bits
Max Octet Rate	(1000Mb/sec)/((8 Bits) = **125,000,000 Octets/sec**
Max Frame Rate (84 Octet Frames) Min Packet (60 Bytes + 4 Bytes CRC)	(1000Mb/sec)/((8 Bits)*(84 Octets/Frame)) = **1,488,095 Frames/sec (FPS)**
Max UDP Data Rate (84 Octet Frames) Min UDP Packet (60 Bytes + 4 Bytes CRC)	(1,488,095 Frames/sec)*(18 Bytes/Frame) = **26,785,714 Bytes/sec**
Max Frame Rate (1538 Octet Frames) Max Packet (1514 Bytes + 4 Bytes CRC)	(1000Mb/sec)/((8 Bits)*(1538 Octets/Frame)) = **81,274 Frames/sec (FPS)**
Max UDP Data Rate (1538 Octet Frames) Max UDP Packet (1514 Bytes + 4 Bytes CRC)	(81,274 Frames/sec)*(1472 Bytes/Frame) = **119,635,891 Bytes/sec**
Max Gigabit Ethernet Frame Bandwidth Max Packet (60 Bytes + 4 Bytes CRC)	(1,488,095 Frames/sec)*(64 Bytes/Frame) = **95,238,080 Bytes/sec** (90.876031 MiB/s)
Max Packet (60 Bytes)	(1,488,095 Frames/sec)*(60 Bytes/Frame) = **89,285,700 Bytes/sec** (85.149477 MiB/s)
Max Gigabit Ethernet Frame Bandwidth Max Packet (1514 Bytes + 4 Bytes CRC)	(81,274 Frames/sec)*(1518 Bytes/Frame) = **123,373,932 Bytes/sec** (117.658550 MiB/s)
Max Packet (1514 Bytes)	(81,274 Frames/sec)*(1514 Bytes/Frame) = **123,048,836 Bytes/sec** (117.348515 MiB/s)

*** **Note 1:** IEEE 802.3ab – Gigabit Ethernet over copper twisted-pair cabling.
*** **Note 2:** Gigabit Ethernet allows for larger MTUs (Jumbo or Super Jumbo Frames).
*** **Note 3:** Units – **M:** 1,000,000 **Mi:** 1,048,576

Questions?

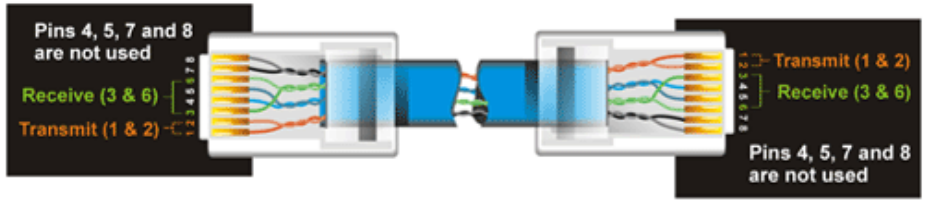
Hardware

Layer 1 Hardware

- A Network Interface Card or NIC provides the physical connection between a computer and the network
- Most common connector / cabling uses an RJ-45 and CAT 5
- An Ethernet Hub simply provides a connection at the physical layer
- Not suitable for higher network speeds
- Limited number of ports possible on a single hub

Network Interface Card (NIC)

Cabling

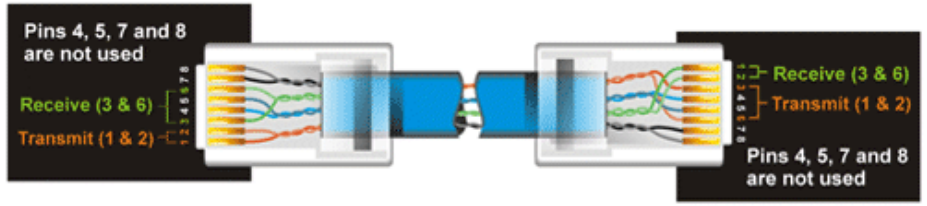


Pin number	Wire Color
Pin 1 ==>	Orange/White
Pin 2 ==>	Orange
Pin 3 ==>	Green/White
Pin 4 ==>	Blue
Pin 5 ==>	Blue/White
Pin 6 ==>	Green
Pin 7 ==>	Brown/White
Pin 8 ==>	Brown

Straight-Through

Wire	Becomes
1	1
2	2
3	3
6	6

Pin number	Wire Color
Pin 1 ==>	Orange/White
Pin 2 ==>	Orange
Pin 3 ==>	Green/White
Pin 4 ==>	Blue
Pin 5 ==>	Blue/White
Pin 6 ==>	Green
Pin 7 ==>	Brown/White
Pin 8 ==>	Brown



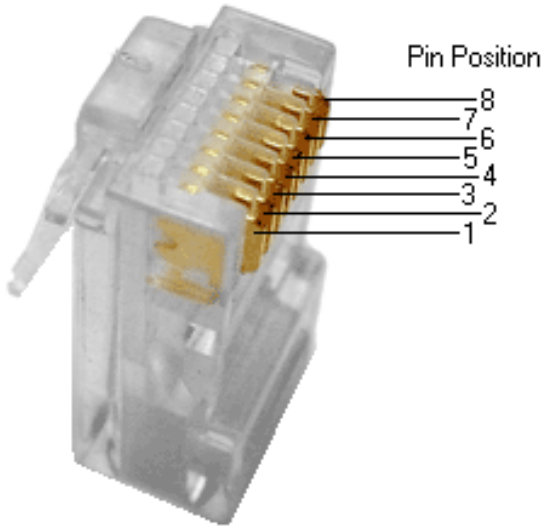
Pin number	Wire Color
Pin 1 ==>	Orange/White
Pin 2 ==>	Orange
Pin 3 ==>	Green/White
Pin 4 ==>	Blue
Pin 5 ==>	Blue/White
Pin 6 ==>	Green
Pin 7 ==>	Brown/White
Pin 8 ==>	Brown

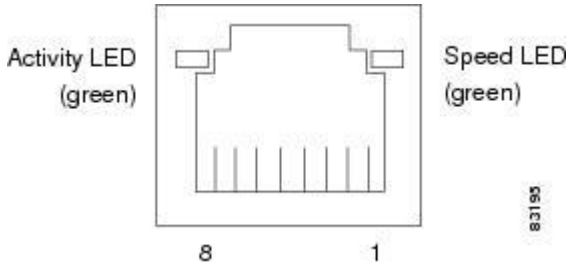
Crossed-Over

Wire	Becomes
1	3
2	6
3	1
6	2

Pin number	Wire Color
Pin 1 ==>	Green/White
Pin 2 ==>	Green
Pin 3 ==>	Orange/White
Pin 4 ==>	Blue
Pin 5 ==>	Blue/White
Pin 6 ==>	Orange
Pin 7 ==>	Brown/White
Pin 8 ==>	Brown

Standards
EIA/TIA T568A/B





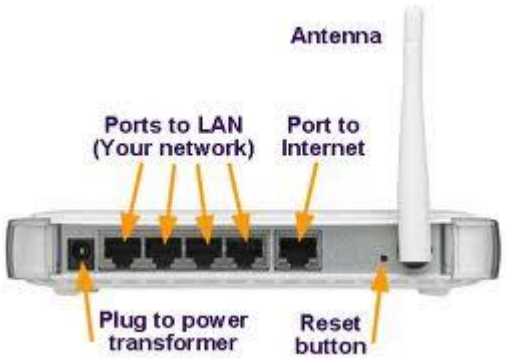
Ethernet Hub

Layer 2 Hardware

- An unmanaged switch functions at the Datalink layer of the OSI model
- The most basic function of a switch is to connect two
- A multi-function devices such as a home router can operate in more than one level of the OSI model
- An Ethernet Bridge provides a physical connection between two networks
- Bridges operate on MAC-layer addresses and are protocol independent

Home Switch / Router

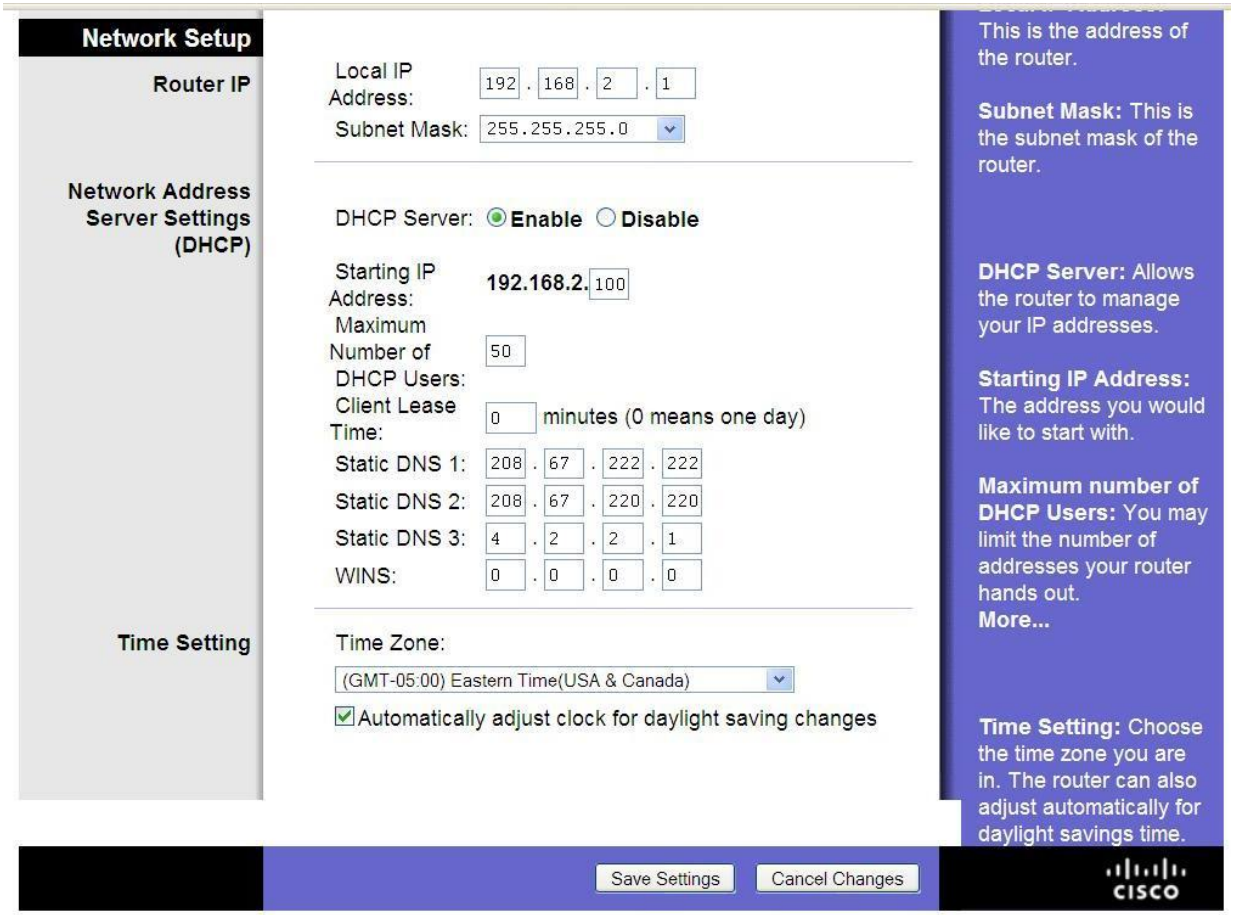
Linksys WRT-54G



Netgear WGR61

Cisco / Linksys WRT-310N

Router Setup



Layer 3 Hardware

- A managed switch functions at the Network layer of the OSI model as does a router
- Source and destination address are needed for the router to do its job
- IP address is logical and independent of hardware

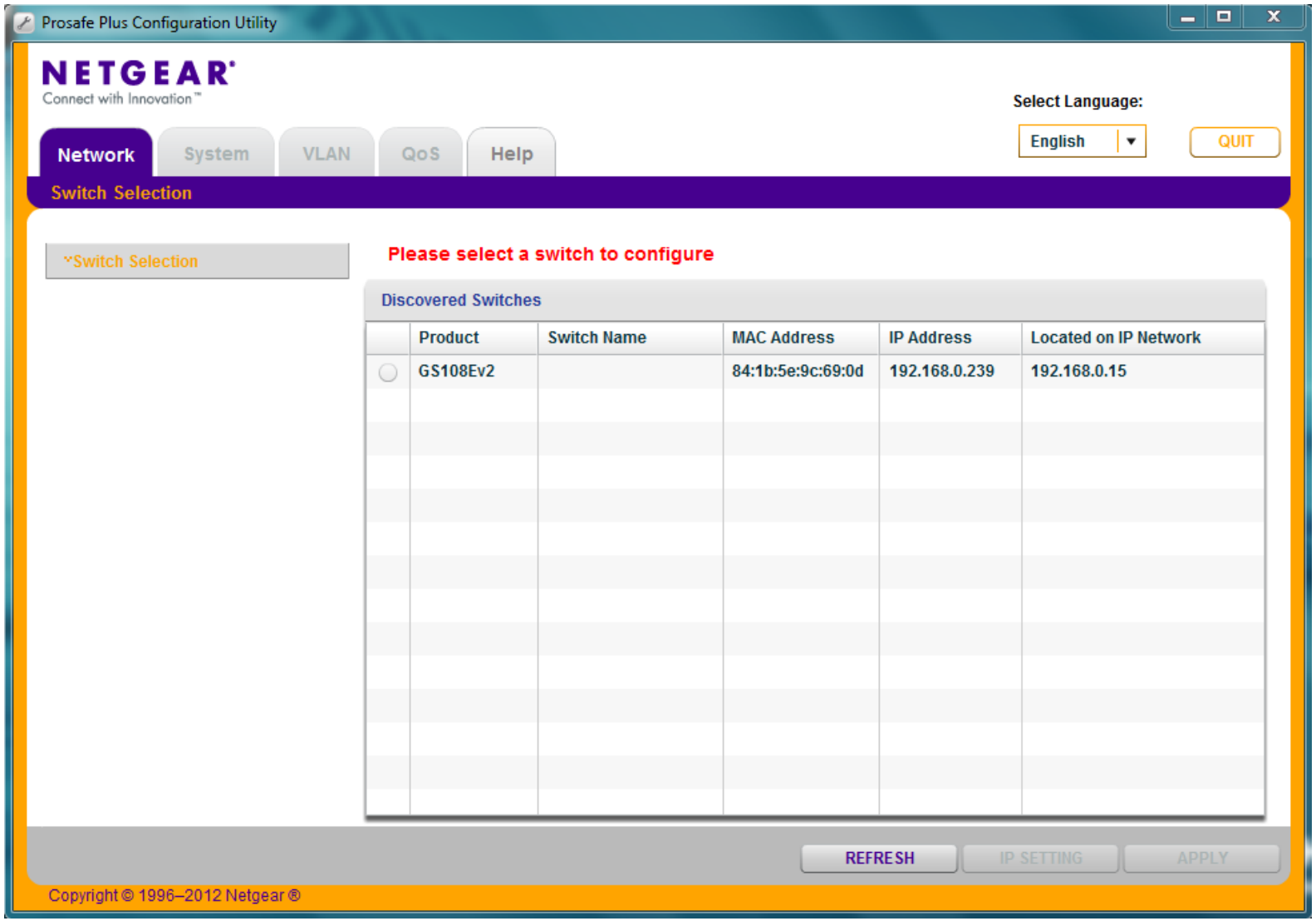
Managed Switch

Netgear GS108E Plus Switch

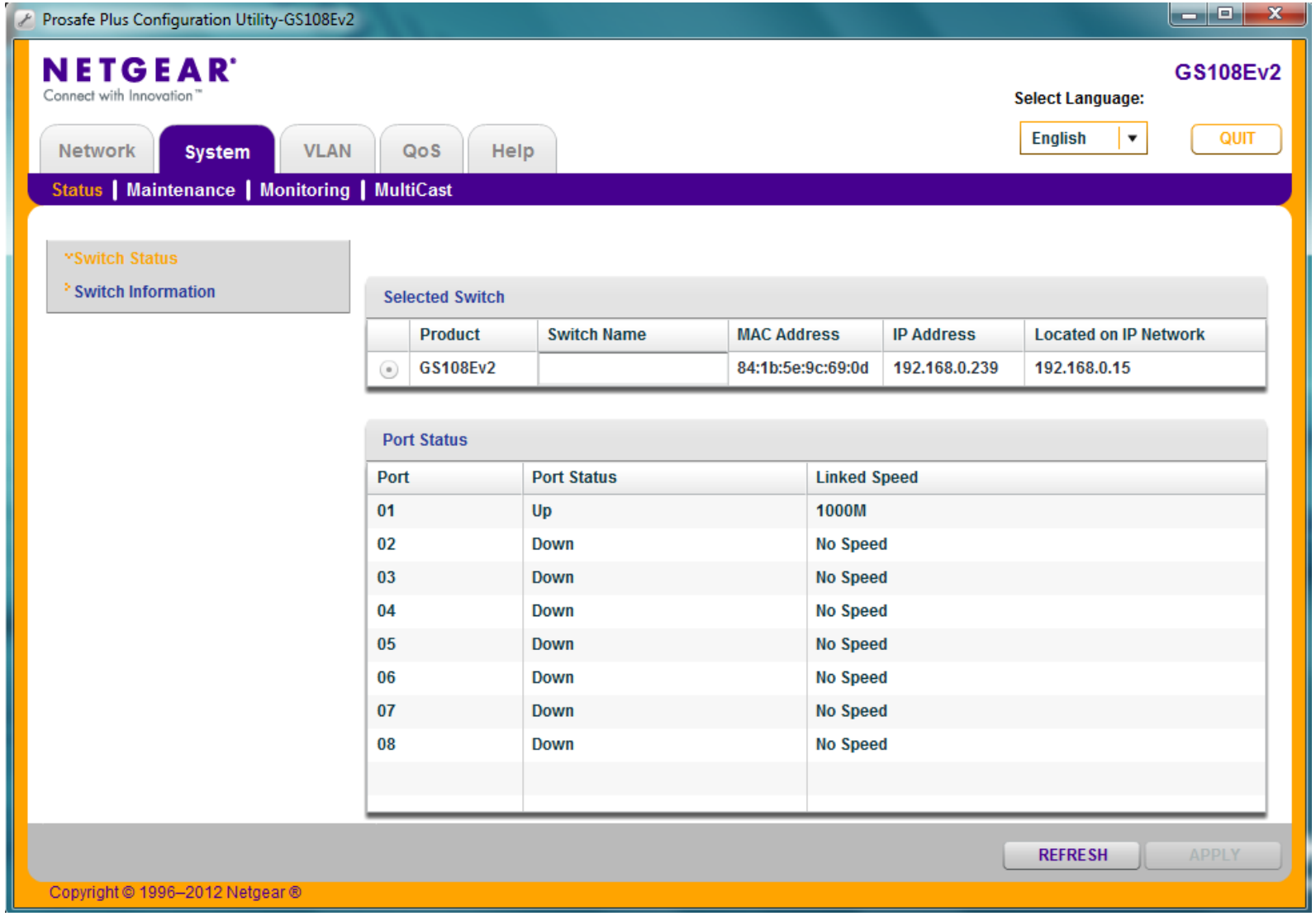
D-Link DGS-3200-10 Switch

HP PS1810-8G Switch

Discovery application



Switch configuration



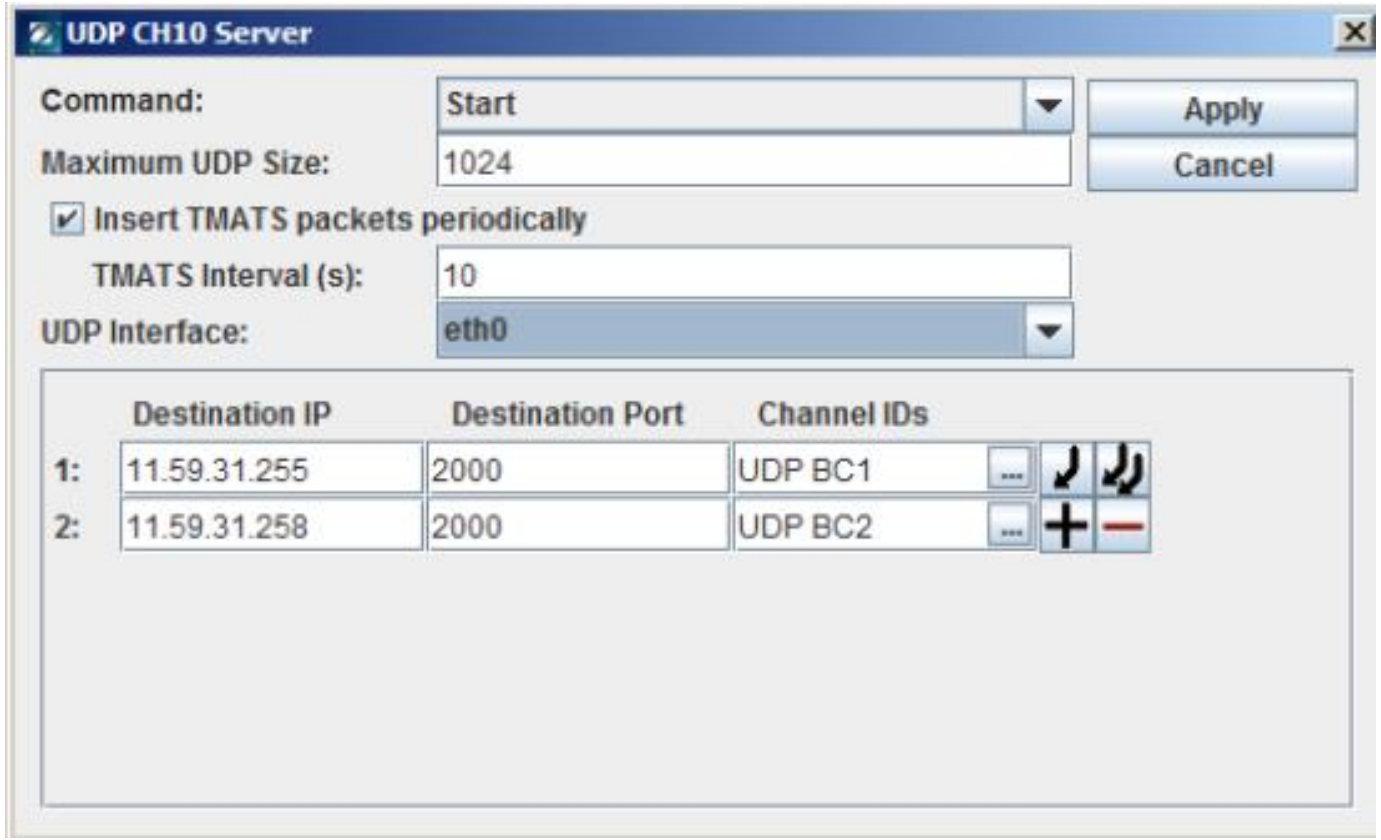
VLANs

- Virtual LAN – capability of higher end switches and routers
- Made up of networked devices logically grouped into separate networks
- Port-based or IEEE 802.1Q
- All addressing unique and independent within VLAN

Chapter 10 Recorders

- Heim / Safran D4Recorder software for MDR8 recorder
- Configure recorder for real-time Chapter 10 packet streaming, FTP file access, and as an RTP (video) server
- Target for Chapter 10 streaming can be single, broadcast or multicast, broadcast is the default (11.59.31.255)

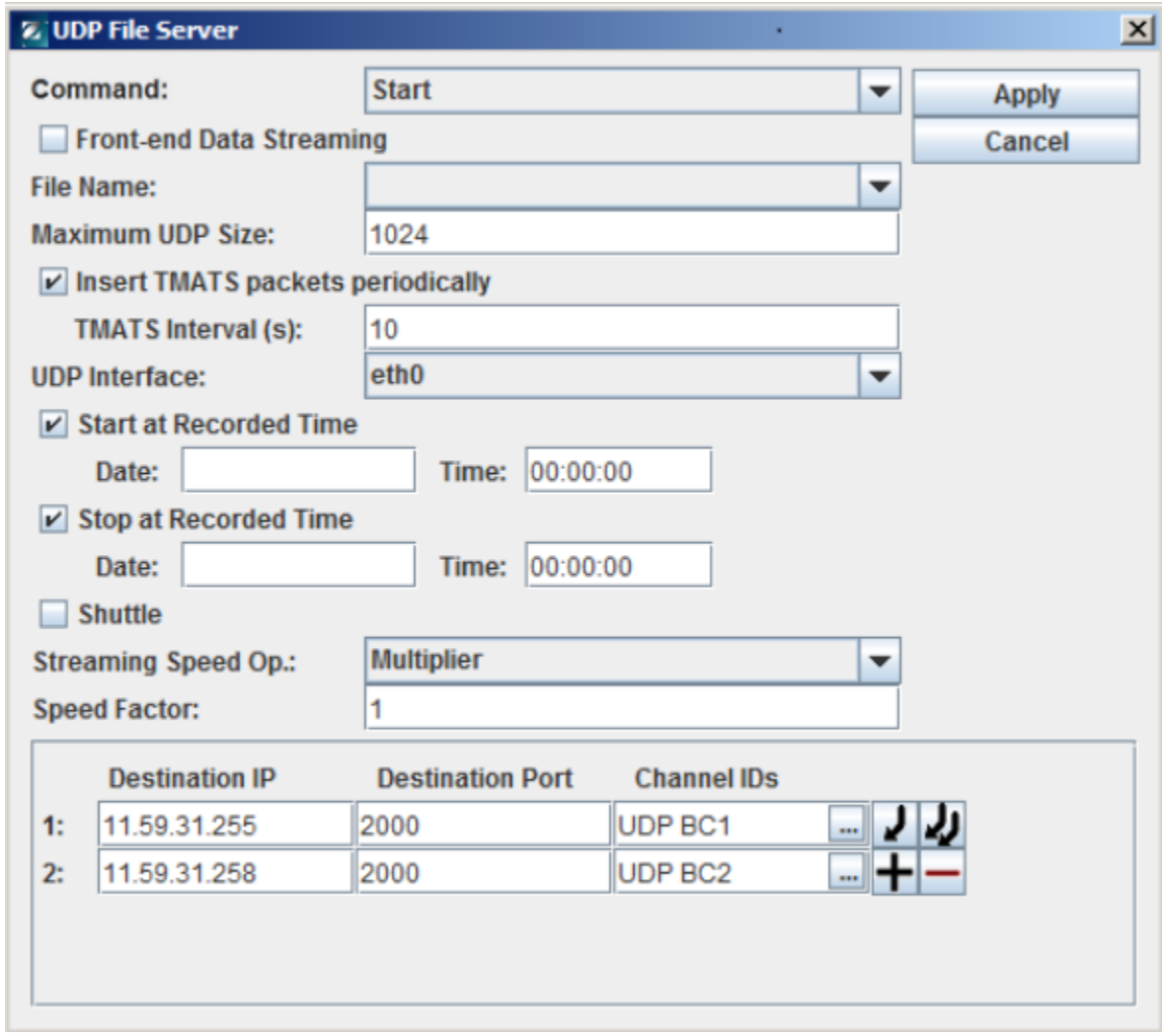
UDP CH10 Server Config



Multiple destination IP addresses allowed

Dialog has an error – can you spot it?

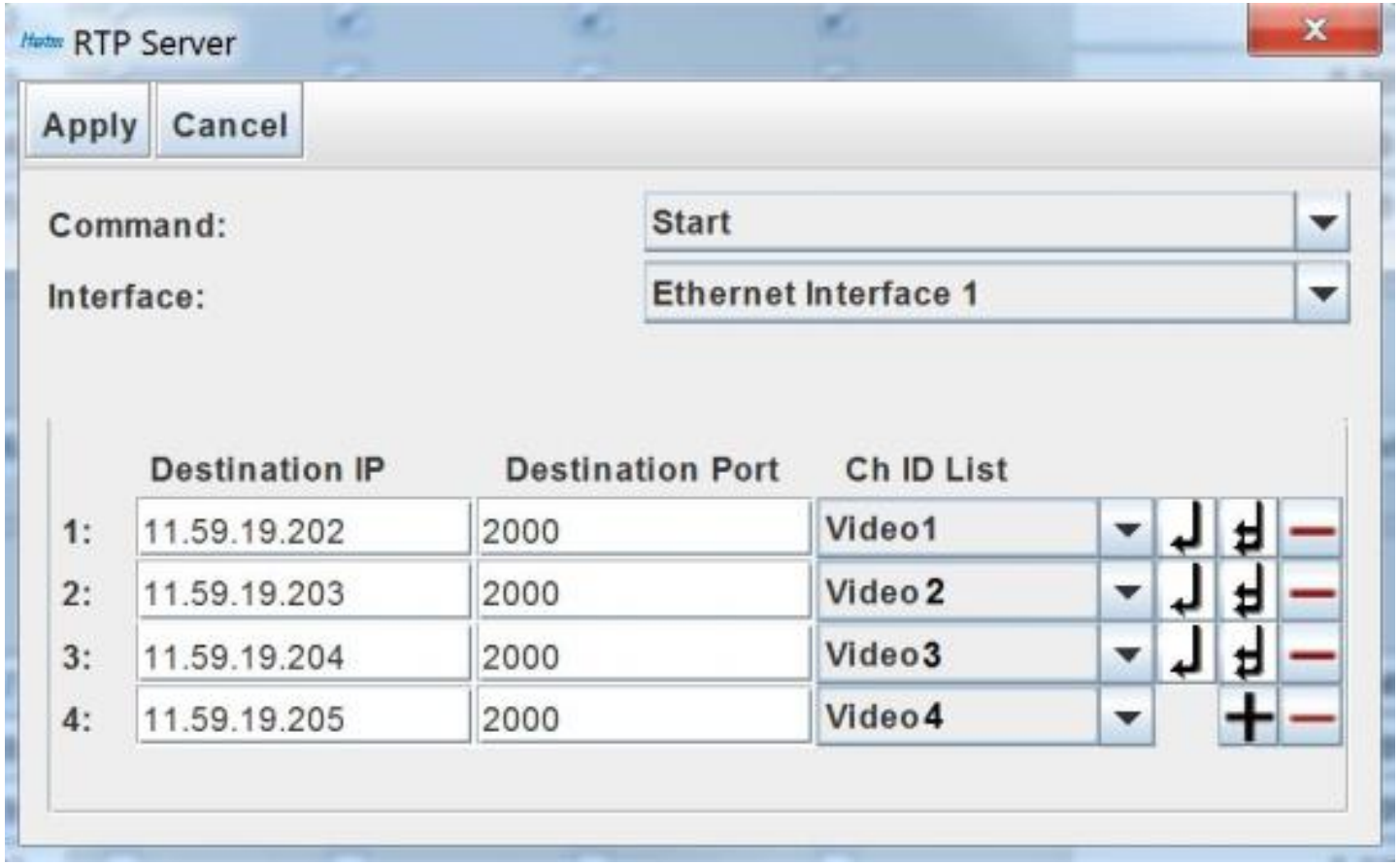
UDP File Server



File Server uses UDP for transfer, not FTP

Same IP address typo

RTP Server



Destination address can be single, broadcast or multicast

Standard FTP also supported

Free Tools

Open Source

- NMAP
- TCPDUMP
- Cryping

Command Line Tools

- Windows
  - Getmac
  - ipconfig
- Linux
  - Ifconfig
- Both
  - netstat
  - ping
  - arp

```
C:\Program Files\ConEmu>getmac /?

GETMAC [/S system [/U username [/P [password]]]] [/FO format] [/NH] [/V]

Description:
    This tool enables an administrator to display the MAC address
    for network adapters on a system.

Parameter List:
    /S     system           Specifies the remote system to connect to.

    /U     [domain\]user    Specifies the user context under
                            which the command should execute.

    /P     [password]       Specifies the password for the given
                            user context. Prompts for input if omitted.

    /FO    format           Specifies the format in which the output
                            is to be displayed.
                            Valid values: "TABLE", "LIST", "CSV".

    /NH                     Specifies that the "Column Header" should
                            not be displayed in the output.
                            Valid only for TABLE and CSV formats.

    /V                      Specifies that verbose output is displayed.

    /?                      Displays this help message.

Examples:
    GETMAC /?
    GETMAC /FO csv
    GETMAC /S system /NH /V
    GETMAC /S system /U user
    GETMAC /S system /U domain\user /P password /FO list /V
    GETMAC /S system /U domain\user /P password /FO table /NH
```

```
C:\Program Files\ConEmu>arp

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

  -a            Displays current ARP entries by interrogating the current
                protocol data.  If inet_addr is specified, the IP and Physical
                addresses for only the specified computer are displayed.  If
                more than one network interface uses ARP, entries for each ARP
                table are displayed.
  -g            Same as -a.
  -v            Displays current ARP entries in verbose mode.  All invalid
                entries and entries on the loop-back interface will be shown.
  inet_addr     Specifies an internet address.
  -N if_addr    Displays the ARP entries for the network interface specified
                by if_addr.
  -d            Deletes the host specified by inet_addr. inet_addr may be
                wildcarded with * to delete all hosts.
  -s            Adds the host and associates the Internet address inet_addr
                with the Physical address eth_addr.  The Physical address is
                given as 6 hexadecimal bytes separated by hyphens. The entry
                is permanent.
  eth_addr      Specifies a physical address.
  if_addr       If present, this specifies the Internet address of the
                interface whose address translation table should be modified.
                If not present, the first applicable interface will be used.
Example:
  > arp -s 157.55.85.212   00-aa-00-62-c6-09  .... Adds a static entry.
  > arp -a                                    .... Displays the arp table.
```

```
C:\Program Files\ConEmu>ping /?

Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]
            [-r count] [-s count] [[-j host-list] | [-k host-list]]
            [-w timeout] [-R] [-S srcaddr] [-4] [-6] target_name

Options:
    -t             Ping the specified host until stopped.
                   To see statistics and continue - type Control-Break;
                   To stop - type Control-C.
    -a             Resolve addresses to hostnames.
    -n count       Number of echo requests to send.
    -l size        Send buffer size.
    -f             Set Don't Fragment flag in packet (IPv4-only).
    -i TTL         Time To Live.
    -v TOS         Type Of Service (IPv4-only. This setting has been deprecated
                   and has no effect on the type of service field in the IP Header).
    -r count       Record route for count hops (IPv4-only).
    -s count       Timestamp for count hops (IPv4-only).
    -j host-list   Loose source route along host-list (IPv4-only).
    -k host-list   Strict source route along host-list (IPv4-only).
    -w timeout     Timeout in milliseconds to wait for each reply.
    -R             Use routing header to test reverse route also (IPv6-only).
    -S srcaddr     Source address to use.
    -4             Force using IPv4.
    -6             Force using IPv6.
```

Netstat

NETSTAT [-a] [-b] [-e] [-f] [-n] [-o] [-p proto] [-r] [-s] [-t] [interval]

-a Displays all connections and listening ports.

-b Displays the executable involved in creating each connection or listening port. In some cases well-known executables host multiple independent components, and in these cases the sequence of components involved in creating the connection or listening port is displayed. In this case the executable name is in [] at the bottom, on top is the component it called, and so forth until TCP/IP was reached. Note that this option can be time-consuming and will fail unless you have sufficient permissions.

-e Displays Ethernet statistics. This may be combined with the -s option.

-f Displays Fully Qualified Domain Names (FQDN) for foreign addresses.

-n Displays addresses and port numbers in numerical form.

-o Displays the owning process ID associated with each connection.

Netstat (cont)

-p proto	Shows connections for the protocol specified by proto; proto may be any of: TCP, UDP, TCPv6, or UDPv6. If used with the -s option to display per-protocol statistics, proto may be any of: IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, or UDPv6.
-r	Displays the routing table.
-s	Displays per-protocol statistics. By default, statistics are shown for IP, IPv6, ICMP, ICMPv6, TCP, TCPv6, UDP, and UDPv6; the -p option may be used to specify a subset of the default.
-t	Displays the current connection offload state.
interval	Redisplays selected statistics, pausing interval seconds between each display. Press CTRL+C to stop redisplaying statistics. If omitted, netstat will print the current configuration information once.

Sysinternals tools on Windows

- Channel9 Video
  - https://channel9.msdn.com/Shows/Defrag-Tools/Defrag-Tools-160-PsPing
- PsPing
  - https://docs.microsoft.com/en-us/sysinternals/downloads/psping

ICMP ping usage

```
psping [[-6]|[-4]] [-h [buckets]] [-i <interval>] [-l <requestsize> [-q]
[-t|-n <count>] [-w <count>] <destination>

  -h    Print histogram (default bucket count is 20).
  -i    Interval in seconds. Specify 0 for fast ping.
  -l    Request size.
  -n    Number of pings.
  -q    Don't output during pings.
  -t    Ping until stopped with Ctrl+C and type Ctrl+Break
        for statistics.
  -w    Warmup with the specified number of iterations (default is 1).
  -4    Force using IPv4.
  -6    Force using IPv6.

For high-speed ping tests use -q and -i 0.
```

TCP ping usage

```
psping [[-6]|[-4]] [-h [buckets]] [-i <interval>] [-l <requestsize> [-q]
[-t|-n <count>] [-w <count>] <destination:destport>

  -h    Print histogram (default bucket count is 20).
  -i    Interval in seconds. Specify 0 for fast ping.
  -l    Request size.
  -n    Number of pings.
  -q    Don't output during pings.
  -t    Ping until stopped with Ctrl+C and type Ctrl+Break
        for statistics.
  -w    Warmup with the specified number of iterations (default is 1).
  -4    Force using IPv4.
  -6    Force using IPv6.

For high-speed ping tests use -q and -i 0.
```

TCP latency usage

```
server: psping [[-6]|[-4]] <-s source:sourceport>
client: psping [[-6]|[-4]] [-h [buckets]] [-r] <-l requestsize>]
<-n count> [-w <count>] <destination:destport>

  -h    Print histogram (default bucket count is 20).
  -l    Request size.
  -n    Number of sends/receives.
  -r    Receive from the server instead of sending.
  -w    Warmup with the specified number of iterations (default is 5).
  -4    Force using IPv4.
  -6    Force using IPv6.

The server can serve both latency and bandwidth tests and remains active
until you terminate it with Control-C.
```

TCP bandwidth usage

```
server: psping [[-6]|[-4]] <-s source:sourceport>

client: psping [[-6]|[-4]] -b [-h [buckets]] [-r] <-l requestsize> <-n count> [-i
<outstanding>] [-w <count>] <destination:destport>


  -b    Bandwidth test.
  -h    Print histogram (default bucket count is 20).
  -i    Number of outstanding I/Os (default is min of 16 and 2x CPU cores).
  -l    Request size.
  -n    Number of sends/receives.
  -r    Receive from the server instead of sending.
  -w    Warmup for the specified iterations (default is 2x CPU cores).
  -4    Force using IPv4.
  -6    Force using IPv6.


The server can serve both latency and bandwidth tests and remains active until
you terminate it with Control-C.
```

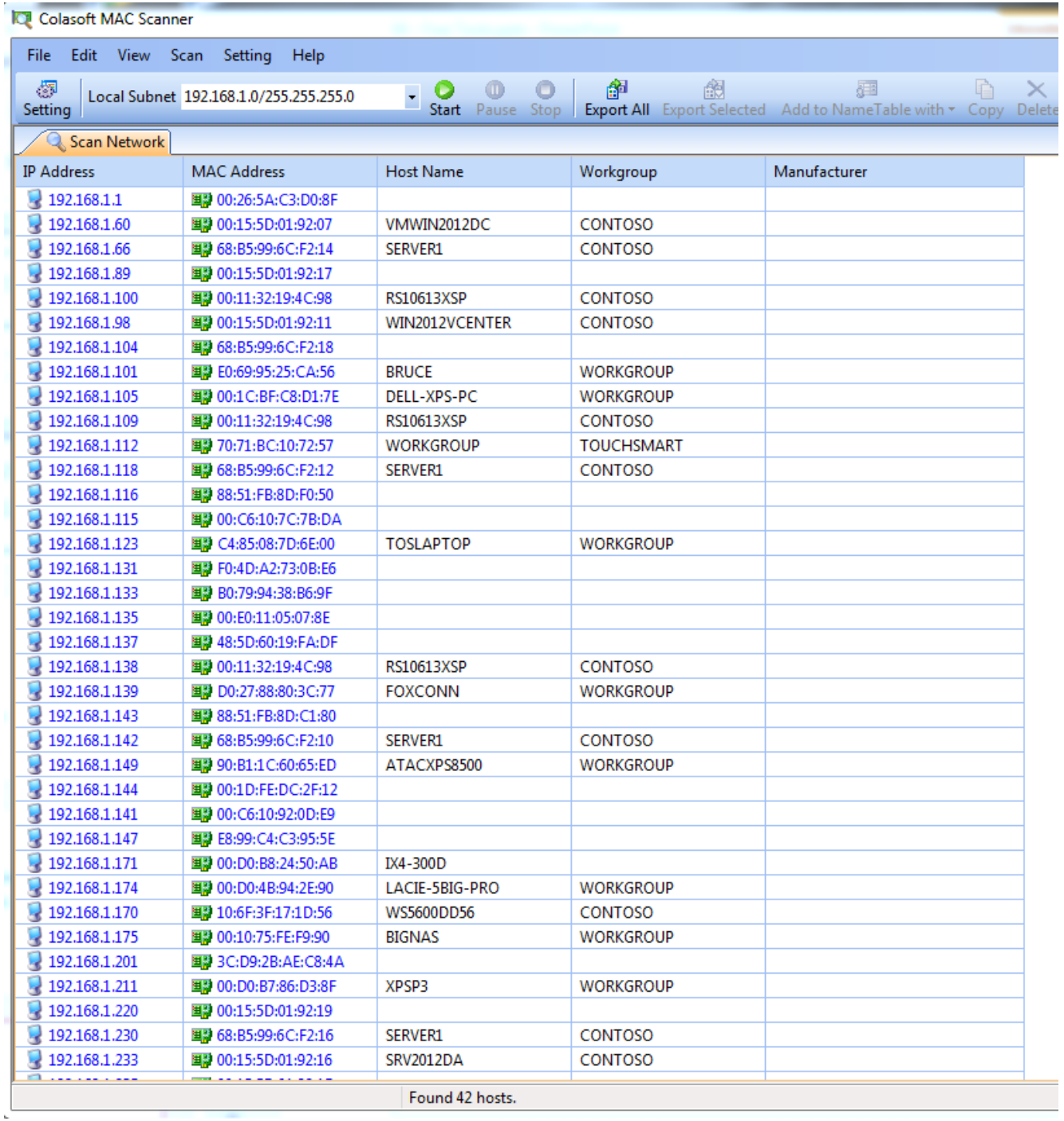
Packet Utilities

- PCAP Replay
  - PlayCap
    - https://github.com/signal11/PlayCap
  - TCP Replay
    - https://github.com/appneta/tcpreplay
- Scapy
  - https://scapy.net/
- Ngrep
  - https://github.com/jpr5/ngrep

Colasoft

- Has a number of free tools
  - https://www.colasoft.com/products/freeware.php
- Colasoft Packet Player
  - https://www.colasoft.com/packet_player/
- Capsa Free
- Colasoft MAC Scanner
- Colasoft Ping Tool
- Colasoft Packet Builder

Colasoft MAC Scanner

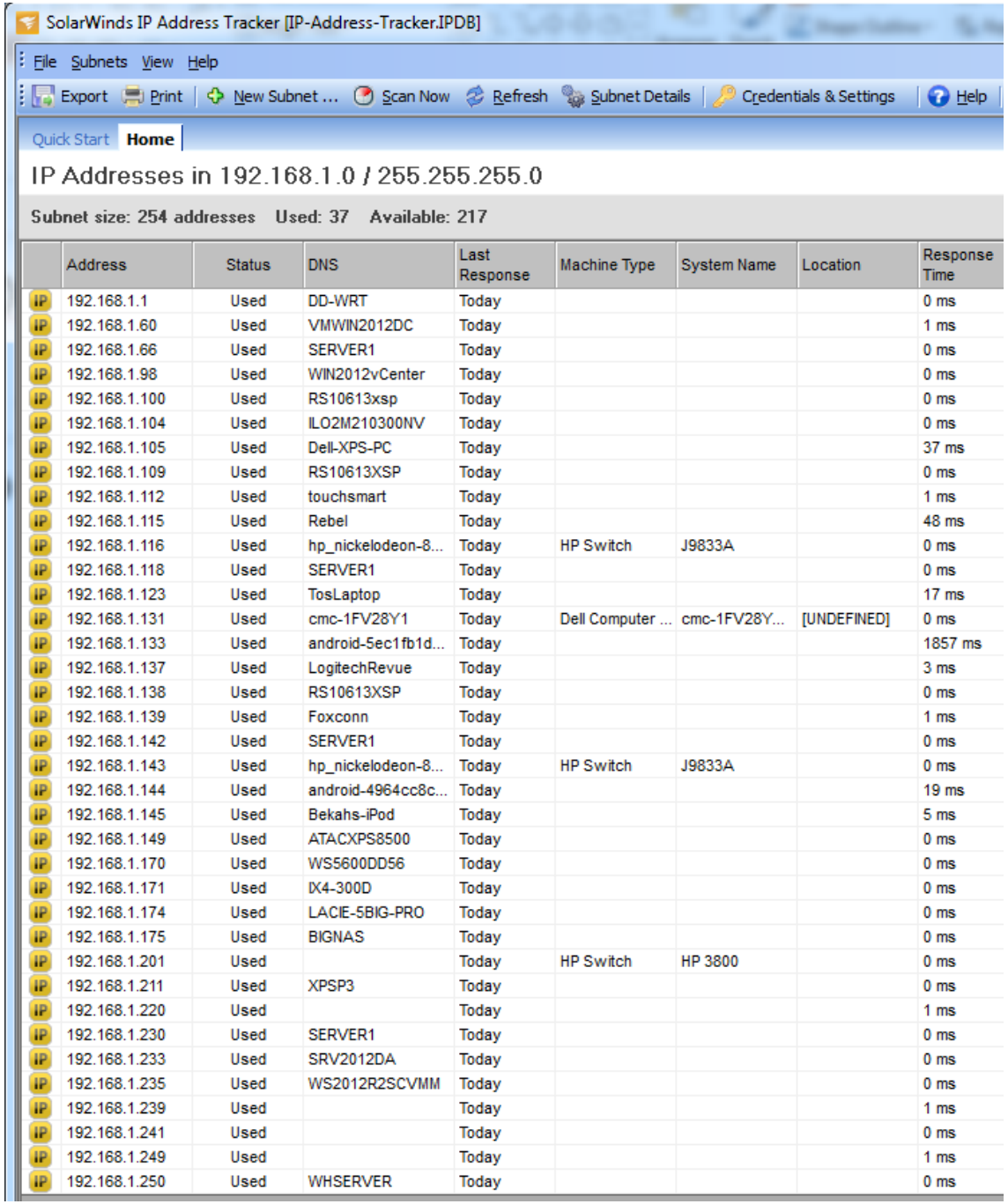


SolarWinds

- Lots of free and trial tools:
  - https://www.solarwinds.com/downloads/
- IP Address Manager
  - Free and trial version
- Subnet Calculator

SolarWinds IP Address Tracker

- IP Address Management



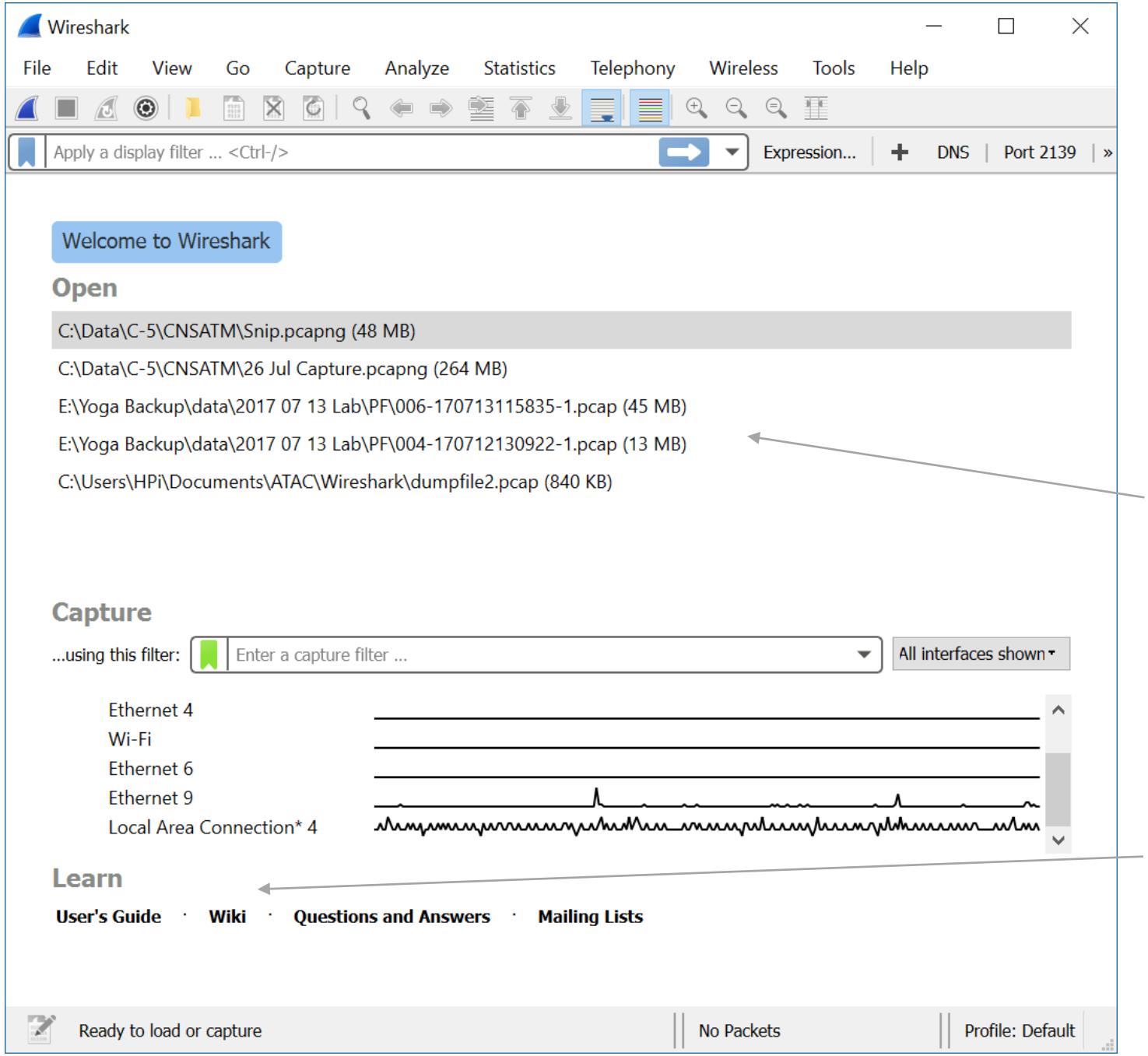
Address	Status	DNS	Last Response	Machine Type	System Name	Location	Response Time
192.168.1.1	Used	DD-WRT	Today				0 ms
192.168.1.60	Used	VMWIN2012DC	Today				1 ms
192.168.1.66	Used	SERVER1	Today				0 ms
192.168.1.98	Used	WIN2012vCenter	Today				0 ms
192.168.1.100	Used	RS10613xsp	Today				0 ms
192.168.1.104	Used	ILO2M210300NV	Today				0 ms
192.168.1.105	Used	Dell-XPS-PC	Today				37 ms
192.168.1.109	Used	RS10613XSP	Today				0 ms
192.168.1.112	Used	touchsmart	Today				1 ms
192.168.1.115	Used	Rebel	Today				48 ms
192.168.1.116	Used	hp_nickelodeon-8...	Today	HP Switch	J9833A		0 ms
192.168.1.118	Used	SERVER1	Today				0 ms
192.168.1.123	Used	TosLaptop	Today				17 ms
192.168.1.131	Used	cmc-1FV28Y1	Today	Dell Computer ...	cmc-1FV28Y...	[UNDEFINED]	0 ms
192.168.1.133	Used	android-5ec1fb1d...	Today				1857 ms
192.168.1.137	Used	LogitechRevue	Today				3 ms
192.168.1.138	Used	RS10613XSP	Today				0 ms
192.168.1.139	Used	Foxconn	Today				1 ms
192.168.1.142	Used	SERVER1	Today				0 ms
192.168.1.143	Used	hp_nickelodeon-8...	Today	HP Switch	J9833A		0 ms
192.168.1.144	Used	android-4964cc8c...	Today				19 ms
192.168.1.145	Used	Bekahs-iPod	Today				5 ms
192.168.1.149	Used	ATACXPS8500	Today				0 ms
192.168.1.170	Used	WS5600DD56	Today				0 ms
192.168.1.171	Used	IX4-300D	Today				0 ms
192.168.1.174	Used	LACIE-5BIG-PRO	Today				0 ms
192.168.1.175	Used	BIGNAS	Today				0 ms
192.168.1.201	Used		Today	HP Switch	HP 3800		0 ms
192.168.1.211	Used	XPSP3	Today				0 ms
192.168.1.220	Used		Today				1 ms
192.168.1.230	Used	SERVER1	Today				0 ms
192.168.1.233	Used	SRV2012DA	Today				0 ms
192.168.1.235	Used	WS2012R2SCVMM	Today				0 ms
192.168.1.239	Used		Today				1 ms
192.168.1.241	Used		Today				0 ms
192.168.1.249	Used		Today				1 ms
192.168.1.250	Used	WHSERVER	Today				0 ms

Wireshark

PCAP File in Hex

```
00000000  D4 C3 B2 A1 02 00 04 00 00 00 00 00 00 00 00 00   Magic Number
00000010  FF FF 00 00 01 00 00 00 69 DF B6 00 37 A1 07 00   ts_sec ts_usec
00000020  52 00 00 00 52 00 00 00 01 00 5E 01 09 2A 02 00   Length
00000030  00 01 02 08 08 00 45 00 00 40 A5 E3 00 00 40 11
00000040  CA 0A 0A 92 08 02 EF 01 09 2A 5E 1F 23 B2 00 2C
00000050  00 00 01 00 00 00 00 00 00 08 81 AC 48 40 00 00
00000060  24 37 83 84 52 9D 00 00 00 08 81 AC 38 40 00 00
00000070  24 BE 31 33 38 37 98 06 A1 12 69 DF B6 00 6E A1
00000080  07 00 52 00 00 00 52 00 00 00 01 00 5E 01 09 0C
00000090  02 00 00 01 02 08 08 00 45 00 00 40 A5 E4 00 00
000000A0  40 11 CA 27 0A 92 08 02 EF 01 09 0C 5E 20 23 AF
000000B0  00 2C 00 00 01 00 00 00 00 00 00 08 81 A9 50 40
000000C0  00 00 0A 52 E0 00 11 4E 00 00 00 08 81 A9 E8 40
000000D0  00 00 0A 54 6F FE 01 C7 E3 F1 52 61 69 DF B6 00
000000E0  9F A1 07 00 82 00 00 00 82 00 00 00 01 00 5E 01
000000F0  09 16 02 00 00 01 02 08 08 00 45 00 00 70 A5 E5
```

Launch Screen



Go button

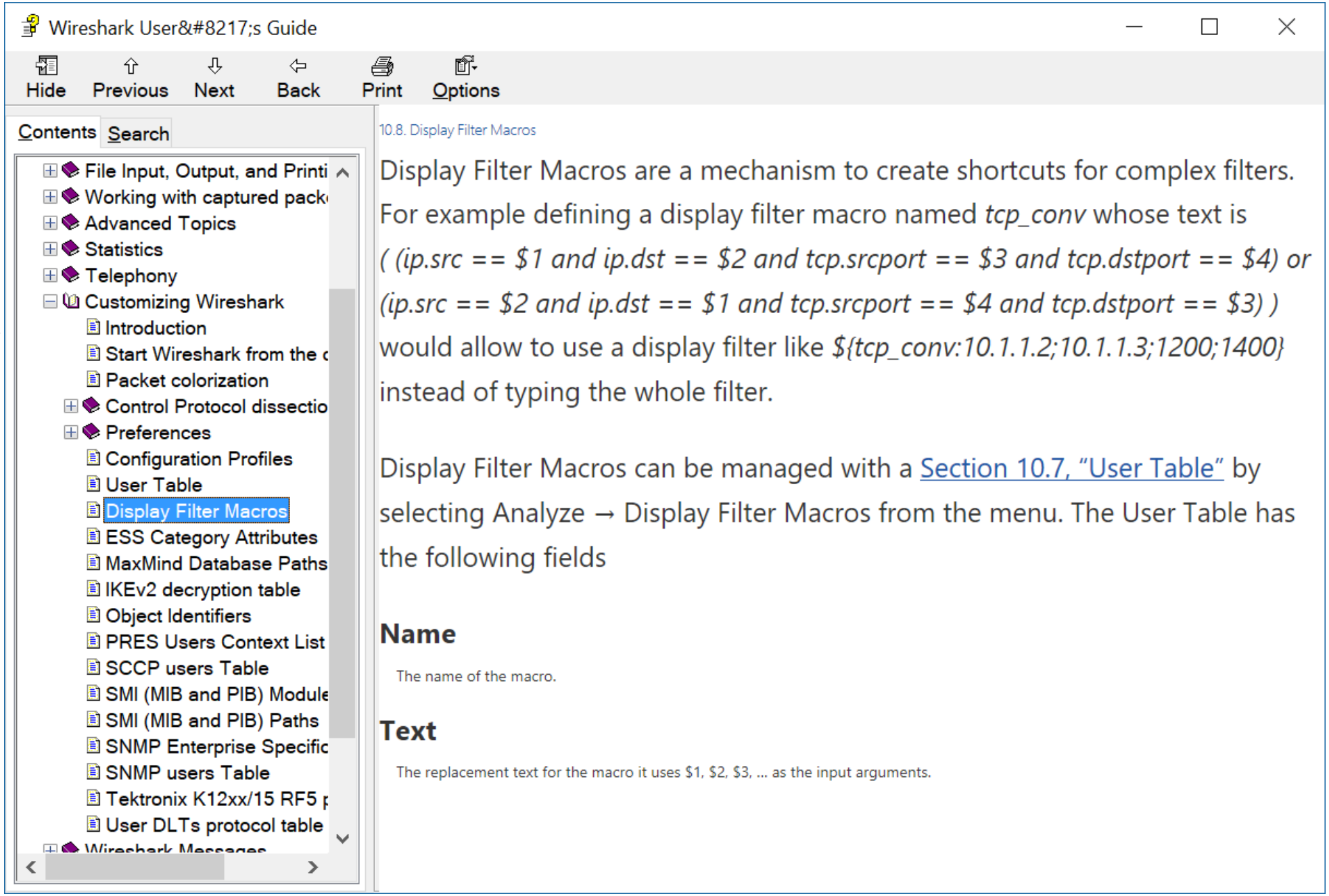
Interface list

Recent files

Takes you to wiki.wireshark.org/SampleCaptures

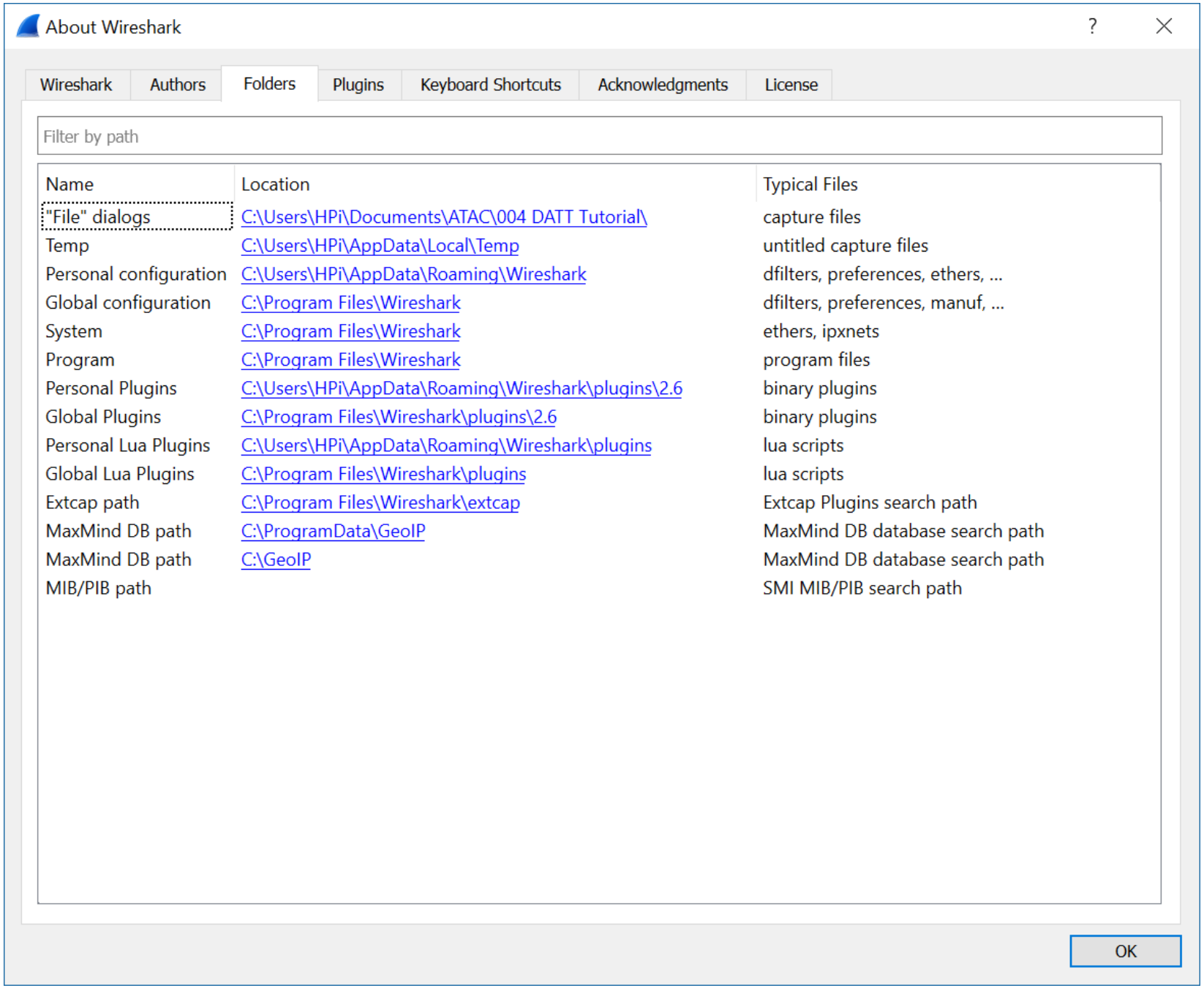
Help

Stand alone help



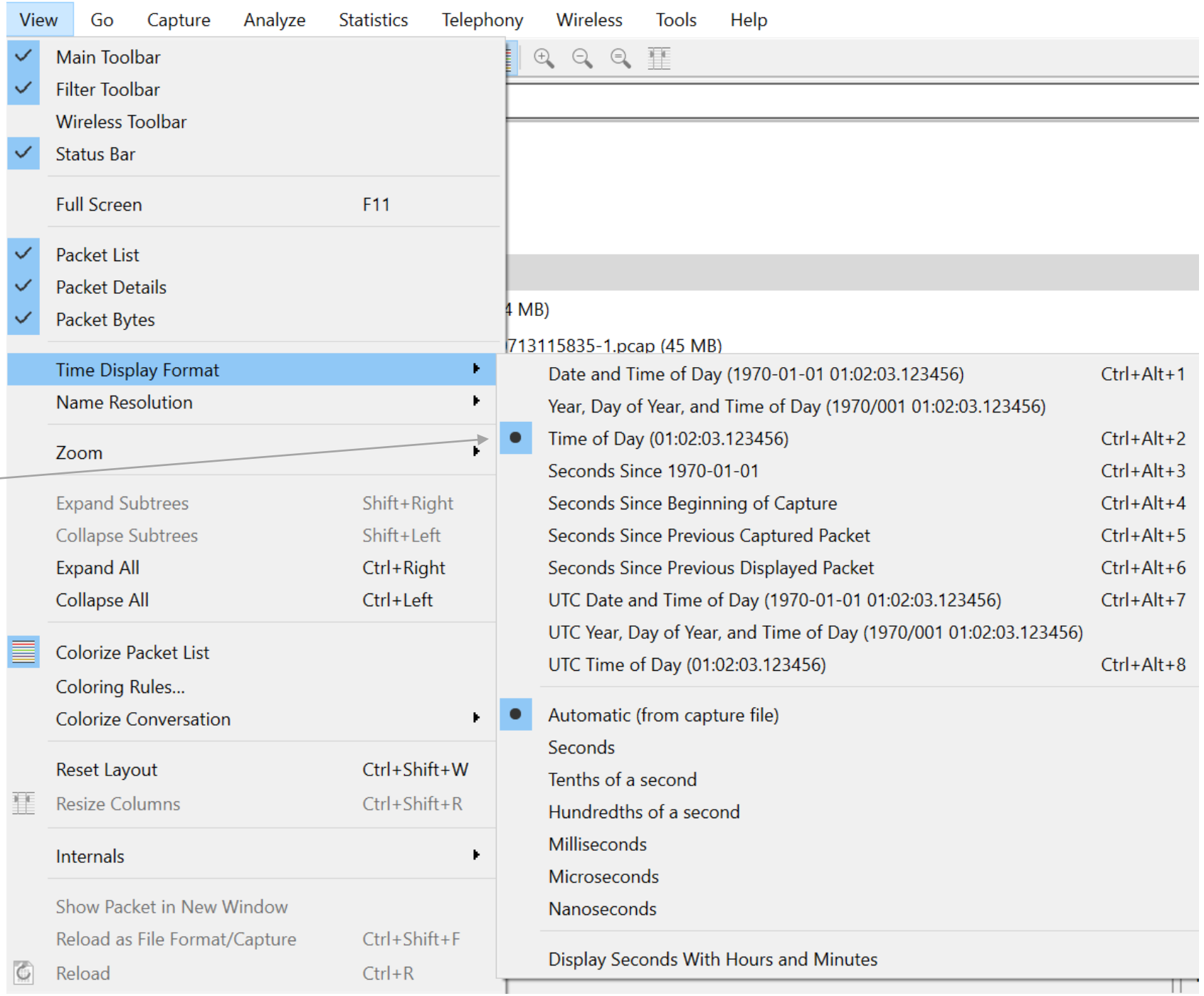
Folders

From About menu

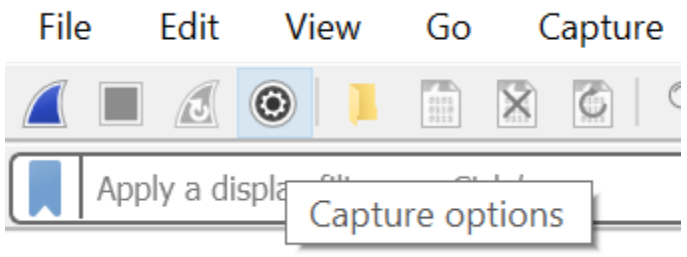


Time Format

Default time setting



Capture options



Wireshark · Capture Interfaces

Input | Output | Options

Interface	Traffic	Link-layer Header	Promiscuous	Snaplen (B)	Buffer (MB)	Monitor Mode	Capture Filter
NETGEAR		Ethernet	☑	default	2	—	
Bluetooth Network Connection		Ethernet	☑	default	2	—	
Ethernet 4		Ethernet	☑	default	2	—	
Wi-Fi		Ethernet	☑	default	2	—	
Ethernet 6		Ethernet	☑	default	2	—	
Ethernet 9		Ethernet	☑	default	2	—	
Local Area Connection* 4		Ethernet	☑	default	2	—	

☑ Enable promiscuous mode on all interfaces

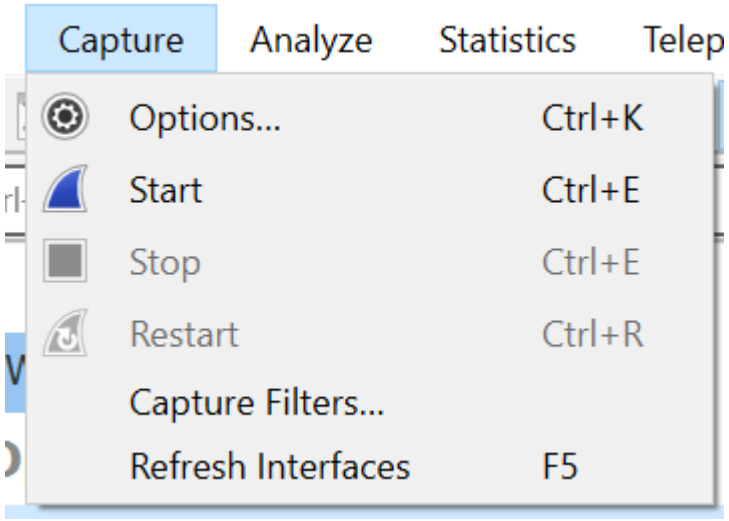
Manage Interfaces...

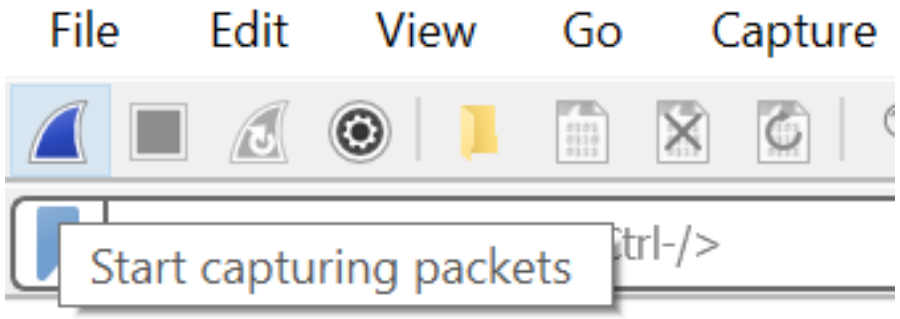
Capture filter for selected interfaces: Enter a capture filter ...

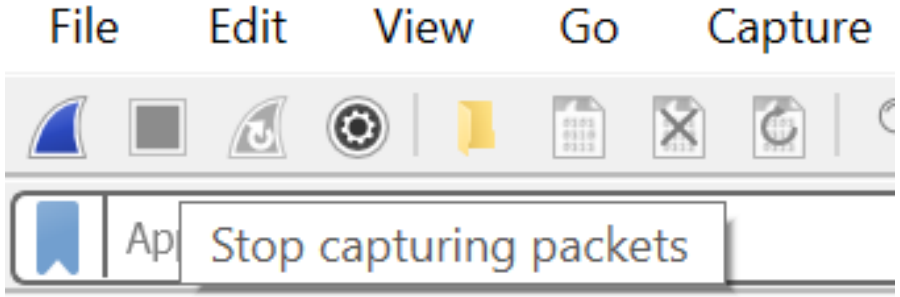
Compile BPFs

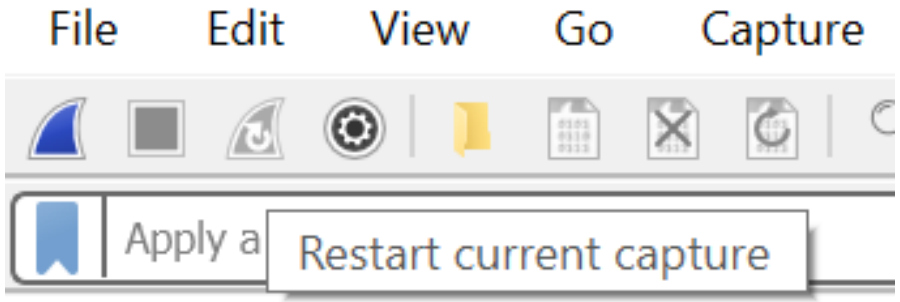
Start | Close | Help

Start / Stop Capture

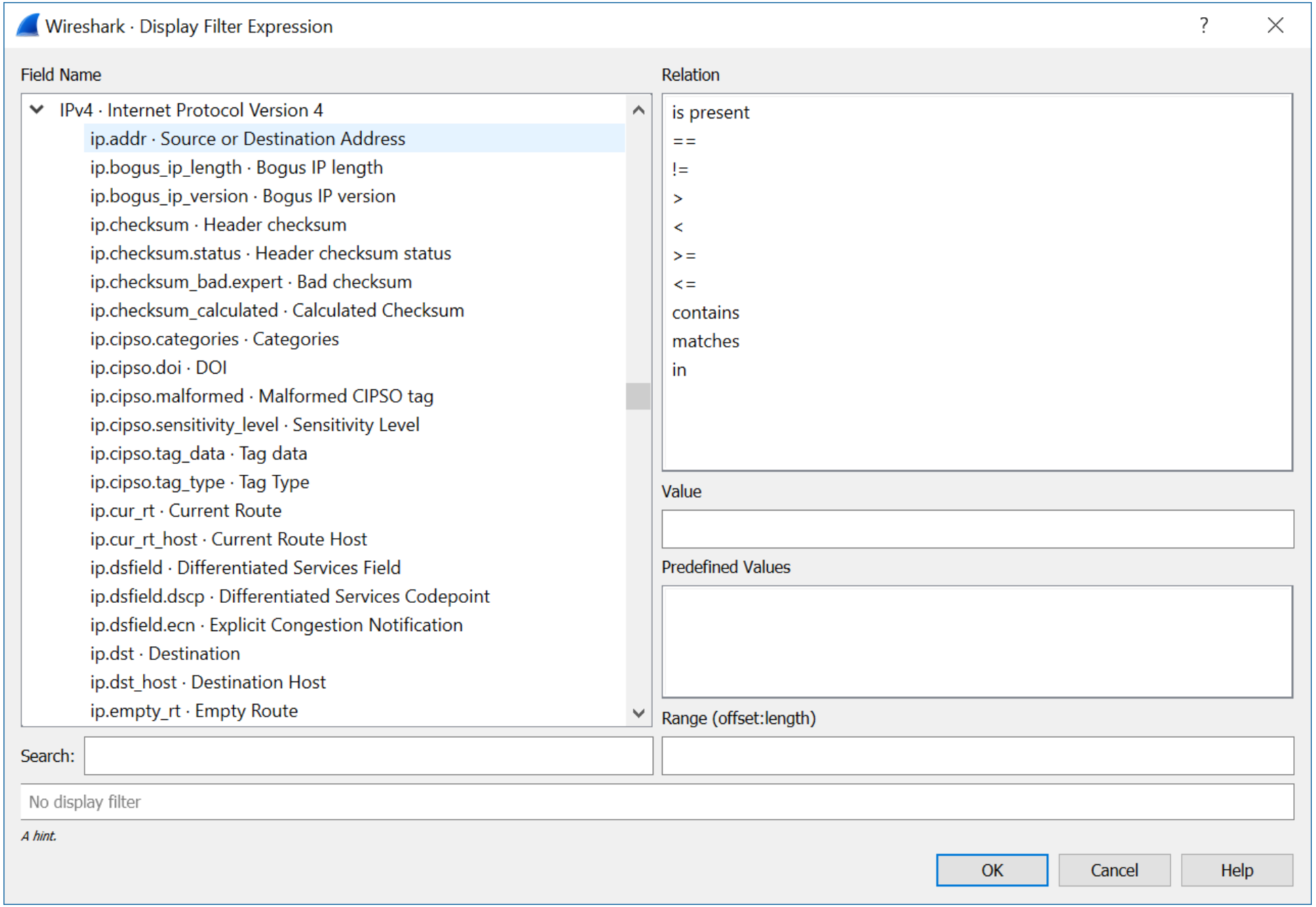








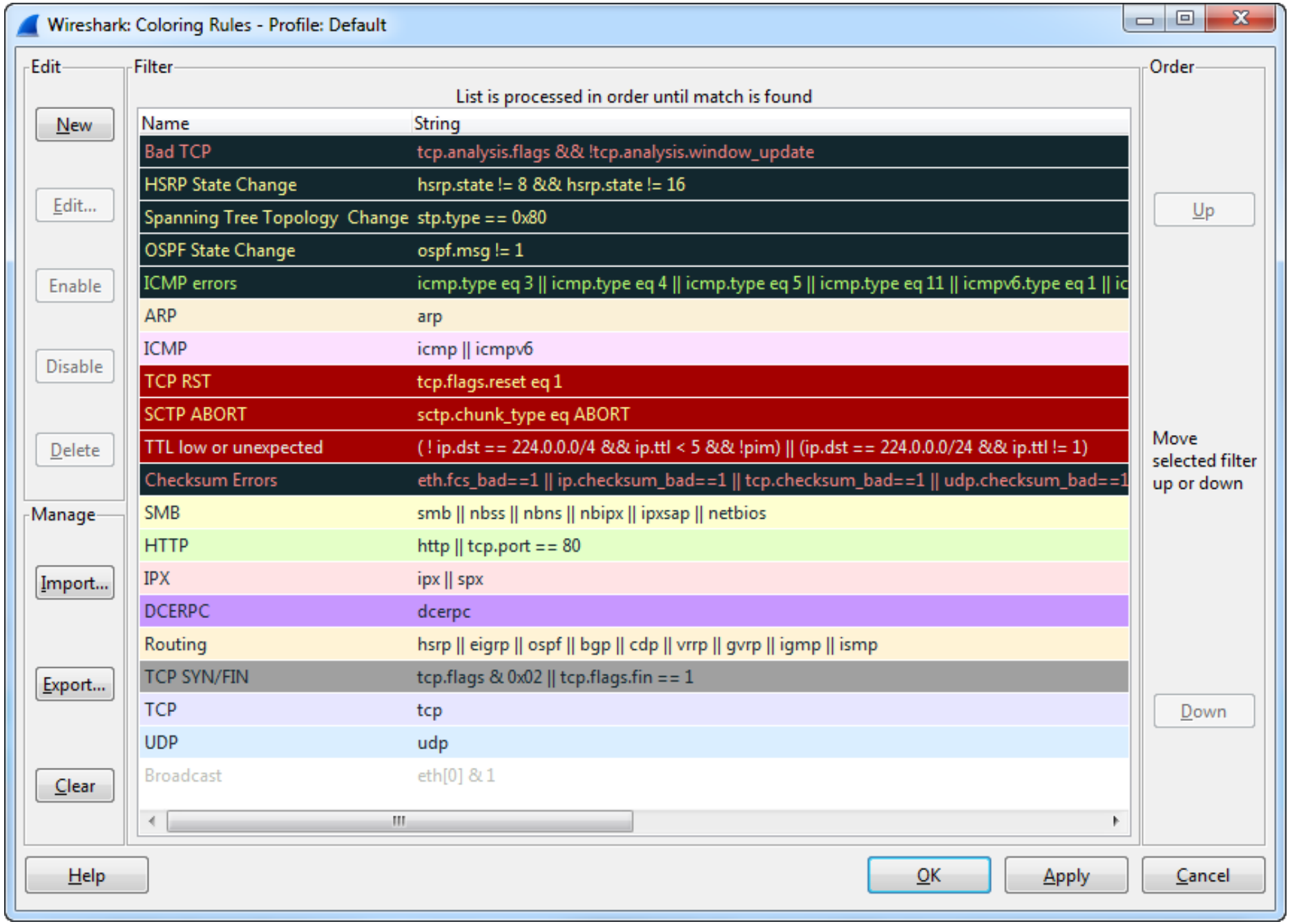
Filter Dialog



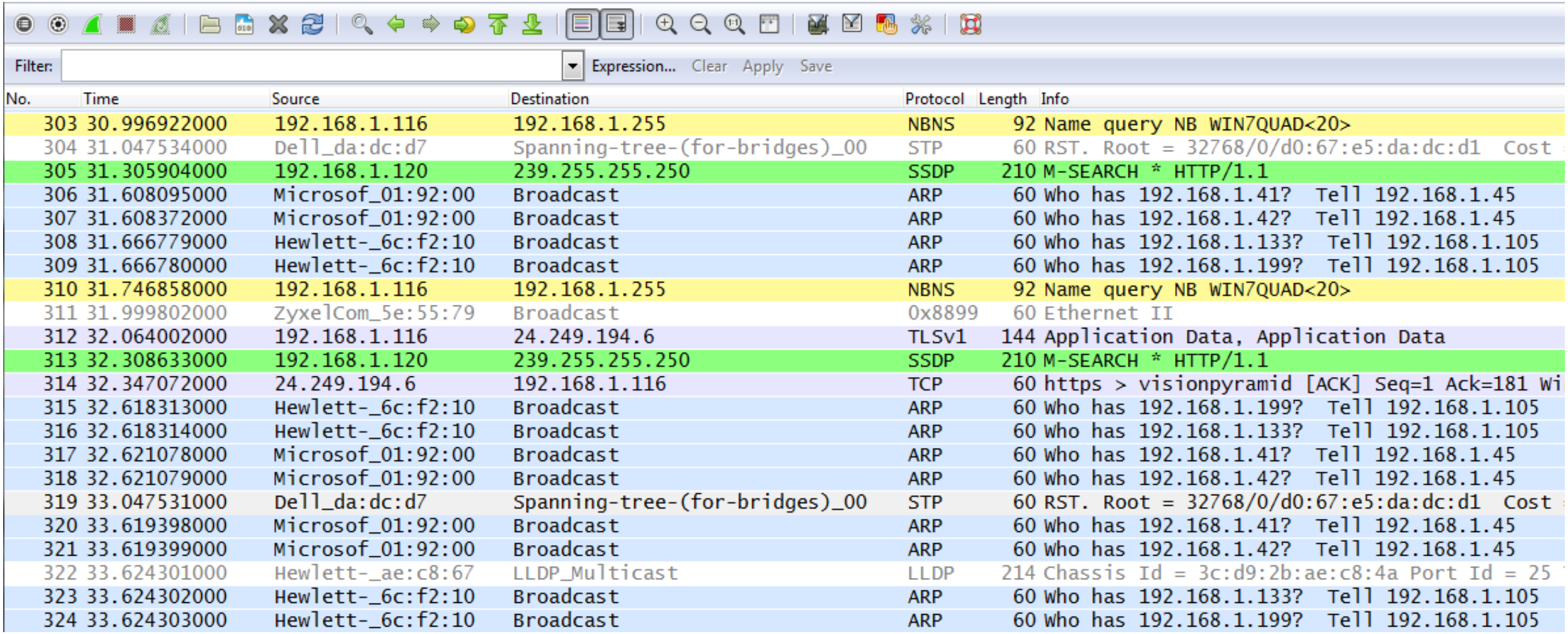
Filter Rules

- By MAC address
  - eth.dst == 30:46:9a:7f:fd:6d
- By IP address
  - tcp.src == 192.168.1.1
- From a packet
  - Right-click and add filter

Coloring Rules



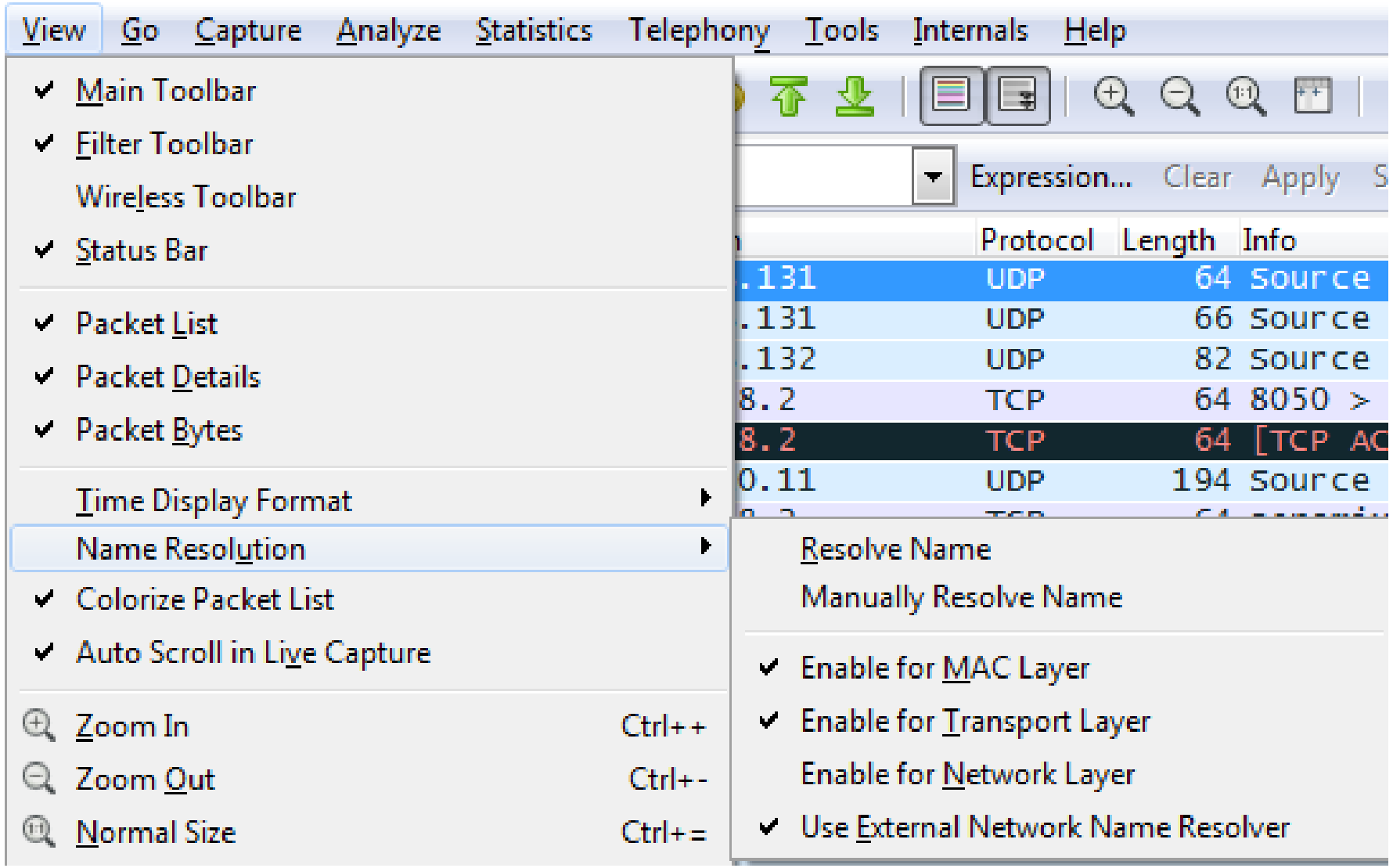
Coloring example



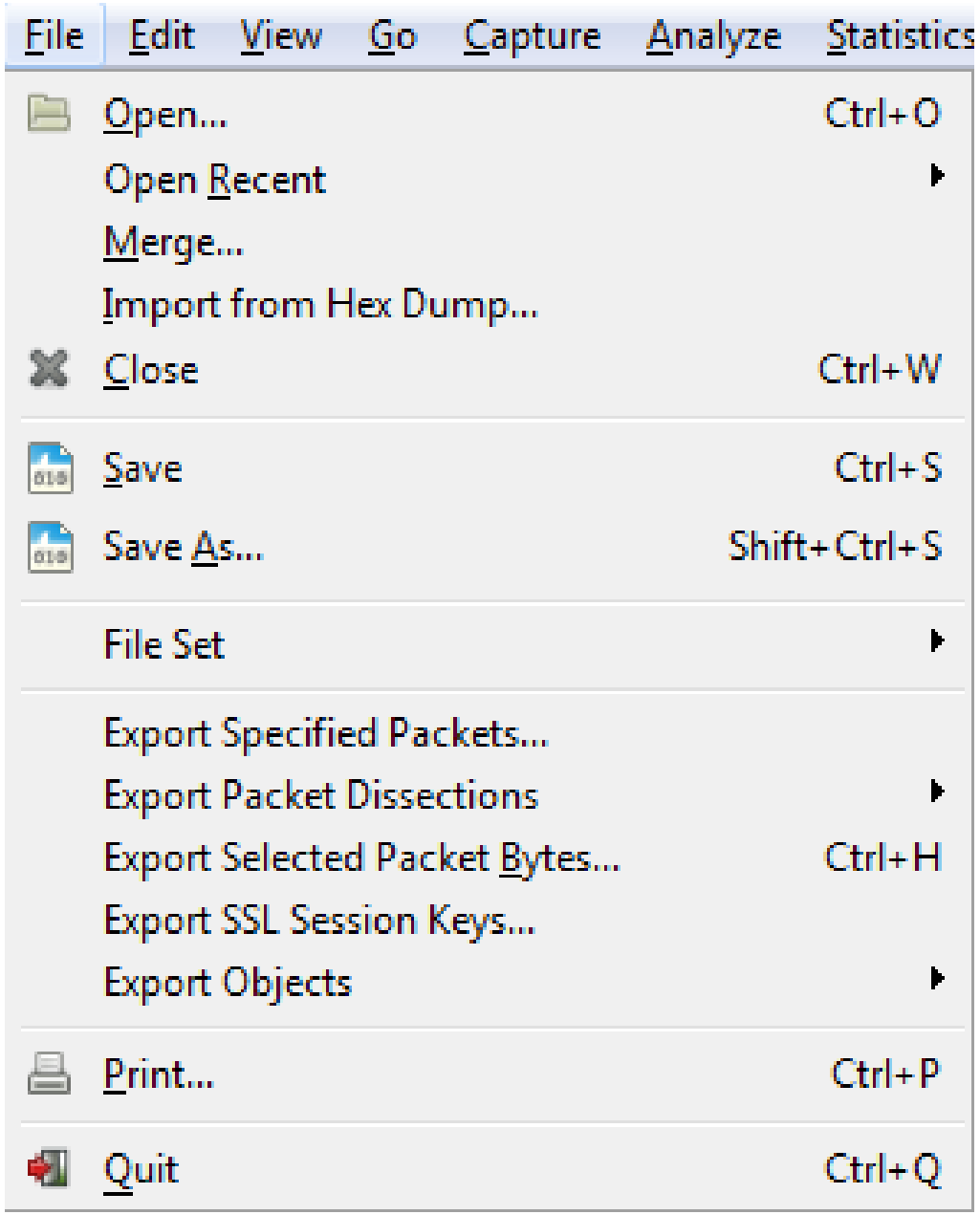
Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
303	30.996922000	192.168.1.116	192.168.1.255	NBNS	92	Name query NB WIN7QUAD<20>
304	31.047534000	Dell_da:dc:d7	Spanning-tree-(for-bridges)_00	STP	60	RST. Root = 32768/0/d0:67:e5:da:dc:d1 Cost
305	31.305904000	192.168.1.120	239.255.255.250	SSDP	210	M-SEARCH * HTTP/1.1
306	31.608095000	Microsof_01:92:00	Broadcast	ARP	60	Who has 192.168.1.41? Tell 192.168.1.45
307	31.608372000	Microsof_01:92:00	Broadcast	ARP	60	Who has 192.168.1.42? Tell 192.168.1.45
308	31.666779000	Hewlett-_6c:f2:10	Broadcast	ARP	60	Who has 192.168.1.133? Tell 192.168.1.105
309	31.666780000	Hewlett-_6c:f2:10	Broadcast	ARP	60	Who has 192.168.1.199? Tell 192.168.1.105
310	31.746858000	192.168.1.116	192.168.1.255	NBNS	92	Name query NB WIN7QUAD<20>
311	31.999802000	ZyxelCom_5e:55:79	Broadcast	0x8899	60	Ethernet II
312	32.064002000	192.168.1.116	24.249.194.6	TLSv1	144	Application Data, Application Data
313	32.308633000	192.168.1.120	239.255.255.250	SSDP	210	M-SEARCH * HTTP/1.1
314	32.347072000	24.249.194.6	192.168.1.116	TCP	60	https > visionpyramid [ACK] Seq=1 Ack=181 Wi
315	32.618313000	Hewlett-_6c:f2:10	Broadcast	ARP	60	Who has 192.168.1.199? Tell 192.168.1.105
316	32.618314000	Hewlett-_6c:f2:10	Broadcast	ARP	60	Who has 192.168.1.133? Tell 192.168.1.105
317	32.621078000	Microsof_01:92:00	Broadcast	ARP	60	Who has 192.168.1.41? Tell 192.168.1.45
318	32.621079000	Microsof_01:92:00	Broadcast	ARP	60	Who has 192.168.1.42? Tell 192.168.1.45
319	33.047531000	Dell_da:dc:d7	Spanning-tree-(for-bridges)_00	STP	60	RST. Root = 32768/0/d0:67:e5:da:dc:d1 Cost
320	33.619398000	Microsof_01:92:00	Broadcast	ARP	60	Who has 192.168.1.41? Tell 192.168.1.45
321	33.619399000	Microsof_01:92:00	Broadcast	ARP	60	Who has 192.168.1.42? Tell 192.168.1.45
322	33.624301000	Hewlett-_ae:c8:67	LLDP_Multicast	LLDP	214	Chassis Id = 3c:d9:2b:ae:c8:4a Port Id = 25
323	33.624302000	Hewlett-_6c:f2:10	Broadcast	ARP	60	Who has 192.168.1.133? Tell 192.168.1.105
324	33.624303000	Hewlett-_6c:f2:10	Broadcast	ARP	60	Who has 192.168.1.199? Tell 192.168.1.105

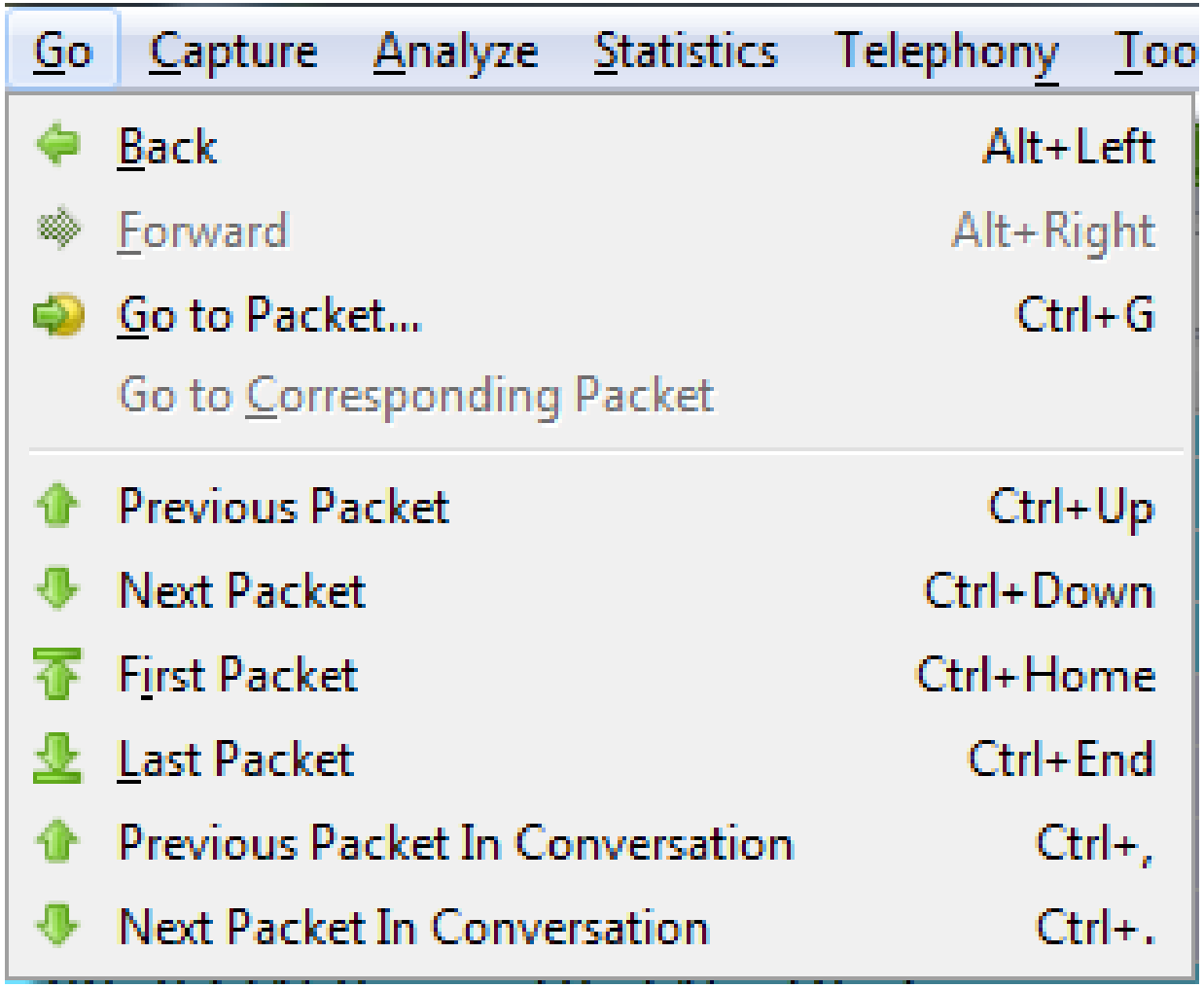
Name Resolution



File Open / Save Options

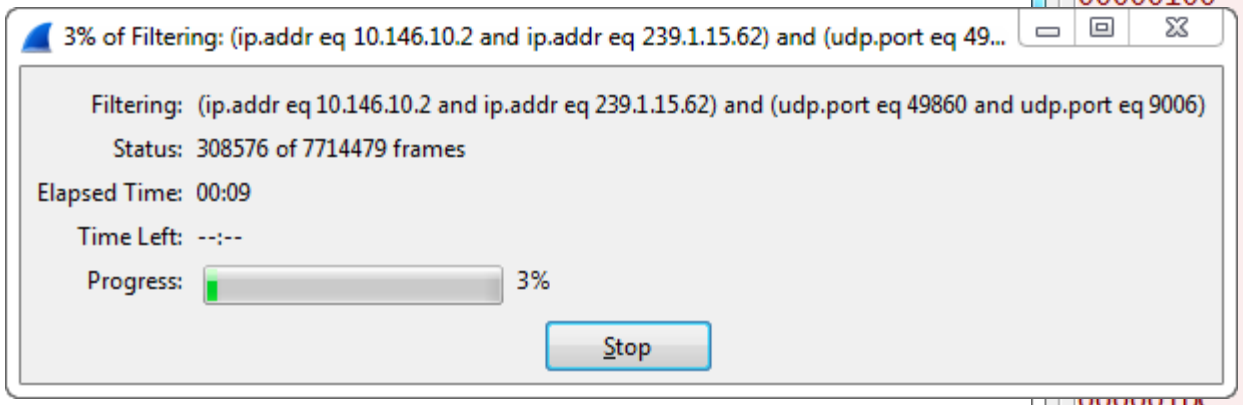


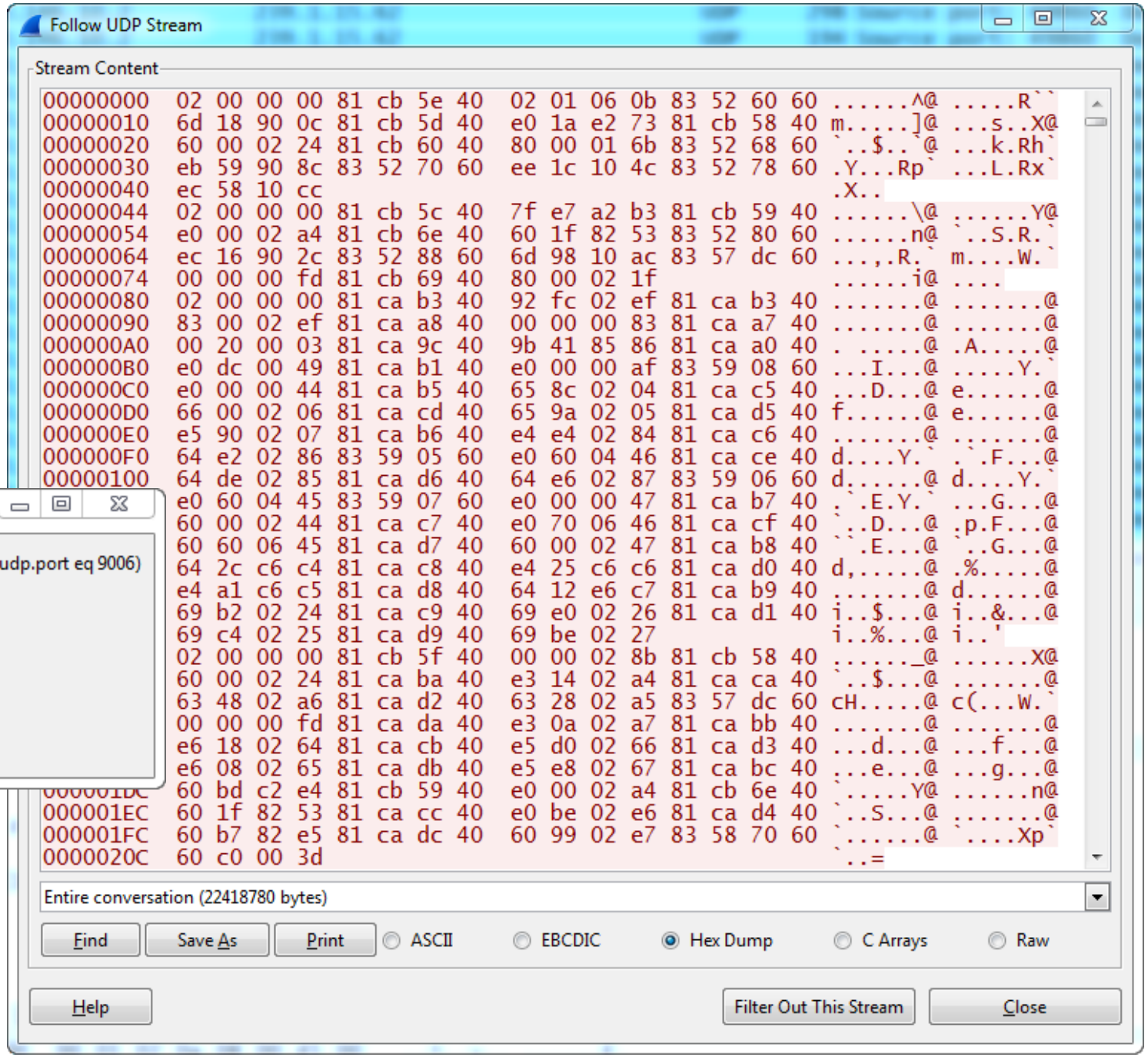
Navigation


Go
Capture
Analyze
Statistics
Telephony
Too
Back
Alt+Left
Forward
Alt+Right
Go to Packet...
Ctrl+G
Go to Corresponding Packet
Previous Packet
Ctrl+Up
Next Packet
Ctrl+Down
First Packet
Ctrl+Home
Last Packet
Ctrl+End
Previous Packet In Conversation
Ctrl+,
Next Packet In Conversation
Ctrl+.

Follow UDP Stream

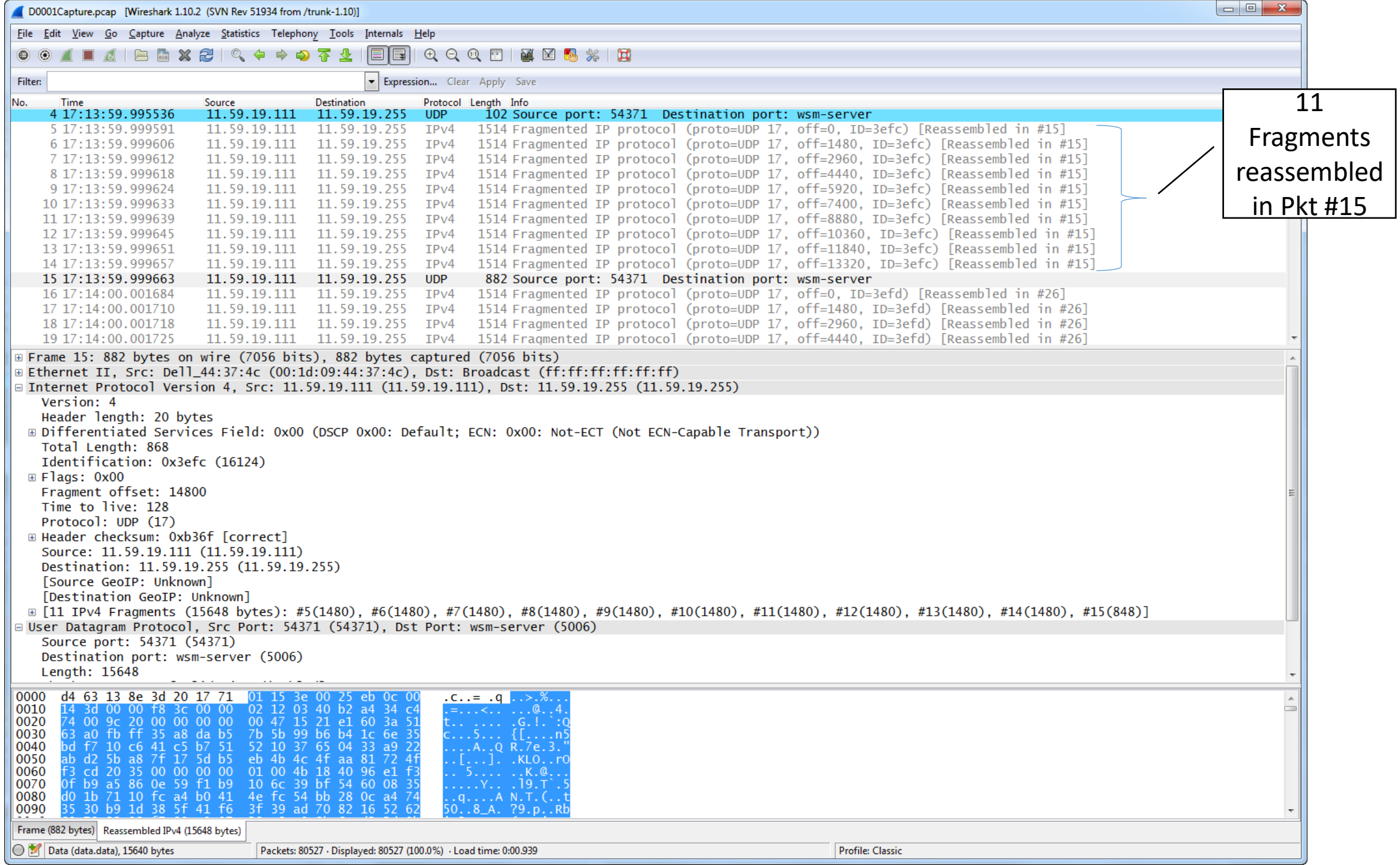
- Choose a packet first
- Launch from Analyze
- Default display Raw
- Save





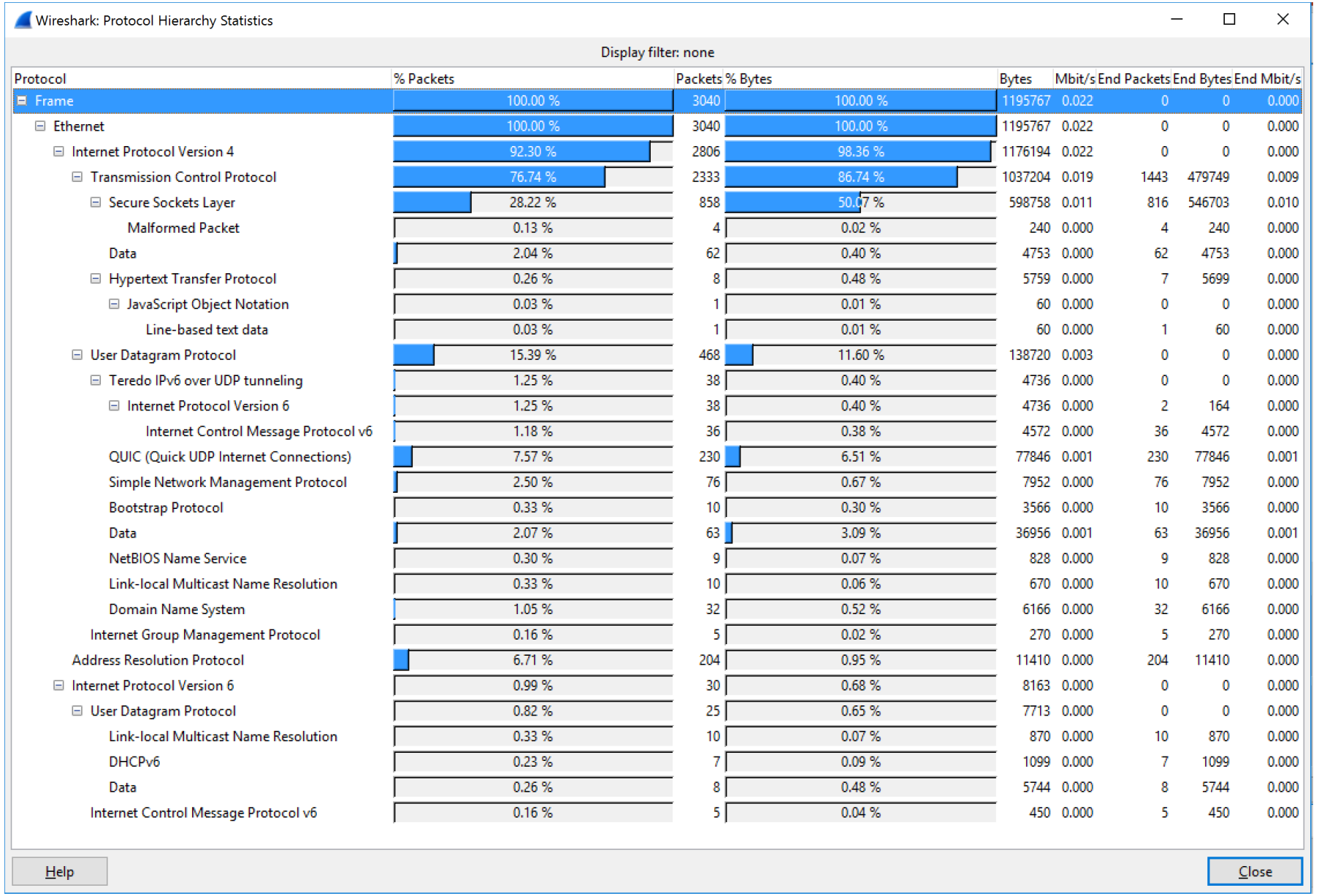
Save file output

```
00000000  02 00 00 00 81 cb 5e 40  02 01 06 0b 83 52 60 60 ......^@ .....R``
00000010  6d 18 90 0c 81 cb 5d 40  e0 1a e2 73 81 cb 58 40 m.....]@ ...s..X@
00000020  60 00 02 24 81 cb 60 40  80 00 01 6b 83 52 68 60 `..$..`@ ...k.Rh`
00000030  eb 59 90 8c 83 52 70 60  ee 1c 10 4c 83 52 78 60 .Y...Rp` ...L.Rx`
00000040  ec 58 10 cc                                     .X..
00000044  02 00 00 00 81 cb 5c 40  7f e7 a2 b3 81 cb 59 40 ......\@ ......Y@
00000054  e0 00 02 a4 81 cb 6e 40  60 1f 82 53 83 52 80 60 ......n@ `..S.R.`
00000064  ec 16 90 2c 83 52 88 60  6d 98 10 ac 83 57 dc 60 ...,.R.` m....W.`
00000074  00 00 00 fd 81 cb 69 40  80 00 02 1f              ......i@ ....
00000080  02 00 00 00 81 ca b3 40  92 fc 02 ef 81 ca b3 40 .......@ .......@
00000090  83 00 02 ef 81 ca a8 40  00 00 00 83 81 ca a7 40 .......@ .......@
000000A0  00 20 00 03 81 ca 9c 40  9b 41 85 86 81 ca a0 40 . .....@ .A.....@
000000B0  e0 dc 00 49 81 ca b1 40  e0 00 00 af 83 59 08 60 ...I...@ .....Y.`
000000C0  e0 00 00 44 81 ca b5 40  65 8c 02 04 81 ca c5 40 ...D...@ e......@
000000D0  66 00 02 06 81 ca cd 40  65 9a 02 05 81 ca d5 40 f......@ e......@
000000E0  e5 90 02 07 81 ca b6 40  e4 e4 02 84 81 ca c6 40 .......@ .......@
000000F0  64 e2 02 86 83 59 05 60  e0 60 04 46 81 ca ce 40 d....Y.` .`.F...@
00000100  64 de 02 85 81 ca d6 40  64 e6 02 87 83 59 06 60 d......@ d....Y.`
```


D0001Capture.pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
4 17:13:59.995536 11.59.19.111 11.59.19.255 UDP 102 Source port: 54371 Destination port: wsm-server
5 17:13:59.999591 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=3efc) [Reassembled in #15]
6 17:13:59.999606 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=3efc) [Reassembled in #15]
7 17:13:59.999612 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=2960, ID=3efc) [Reassembled in #15]
8 17:13:59.999618 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=4440, ID=3efc) [Reassembled in #15]
9 17:13:59.999624 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=5920, ID=3efc) [Reassembled in #15]
10 17:13:59.999633 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=7400, ID=3efc) [Reassembled in #15]
11 17:13:59.999639 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=8880, ID=3efc) [Reassembled in #15]
12 17:13:59.999645 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=10360, ID=3efc) [Reassembled in #15]
13 17:13:59.999651 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=11840, ID=3efc) [Reassembled in #15]
14 17:13:59.999657 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=13320, ID=3efc) [Reassembled in #15]
15 17:13:59.999663 11.59.19.111 11.59.19.255 UDP 882 Source port: 54371 Destination port: wsm-server
16 17:14:00.001684 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=3efd) [Reassembled in #26]
17 17:14:00.001710 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=3efd) [Reassembled in #26]
18 17:14:00.001718 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=2960, ID=3efd) [Reassembled in #26]
19 17:14:00.001725 11.59.19.111 11.59.19.255 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=4440, ID=3efd) [Reassembled in #26]
Frame 15: 882 bytes on wire (7056 bits), 882 bytes captured (7056 bits)
Ethernet II, Src: Dell_44:37:4c (00:1d:09:44:37:4c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 11.59.19.111 (11.59.19.111), Dst: 11.59.19.255 (11.59.19.255)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 868
Identification: 0x3efc (16124)
Flags: 0x00
Fragment offset: 14800
Time to live: 128
Protocol: UDP (17)
Header checksum: 0xb36f [correct]
Source: 11.59.19.111 (11.59.19.111)
Destination: 11.59.19.255 (11.59.19.255)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
[11 IPv4 Fragments (15648 bytes): #5(1480), #6(1480), #7(1480), #8(1480), #9(1480), #10(1480), #11(1480), #12(1480), #13(1480), #14(1480), #15(848)]
User Datagram Protocol, Src Port: 54371 (54371), Dst Port: wsm-server (5006)
Source port: 54371 (54371)
Destination port: wsm-server (5006)
Length: 15648
0000 d4 63 13 8e 3d 20 17 71 01 15 3e 00 25 eb 0c 00 .c..= .q ..>.%...
0010 14 3d 00 00 f8 3c 00 00 02 12 03 40 b2 a4 34 c4 .=...<.. ...@..4.
0020 74 00 9c 20 00 00 00 00 00 47 15 21 e1 60 3a 51 t.. .... .G.!.`:Q
0030 63 a0 fb ff 35 a8 da b5 7b 5b 99 b6 b4 1c 6e 35 c...5... {[....n5
0040 bd f7 10 c6 41 c5 b7 51 52 10 37 65 04 33 a9 22 ....A..Q R.7e.3."
0050 ab d2 5b a8 7f 17 5d b5 eb 4b 4c 4f aa 81 72 4f ..[...]. .KLO..rO
0060 f3 cd 20 35 00 00 00 00 01 00 4b 18 40 96 e1 f3 .. 5.... ..K.@...
0070 0f b9 a5 86 0e 59 f1 b9 10 6c 39 bf 54 60 08 35 .....Y.. .l9.T`.5
0080 d0 1b 71 10 fc a4 b0 41 4e fc 54 bb 28 0c a4 74 ..q....A N.T.(..t
0090 35 30 b9 1d 38 5f 41 f6 3f 39 ad 70 82 16 52 62 50..8_A. ?9.p..Rb
Frame (882 bytes)
Reassembled IPv4 (15648 bytes)
Data (data.data), 15640 bytes
Packets: 80527 · Displayed: 80527 (100.0%) · Load time: 0:00.939
Profile: Classic
11 Fragments reassembled in Pkt #15

Bad Behavior

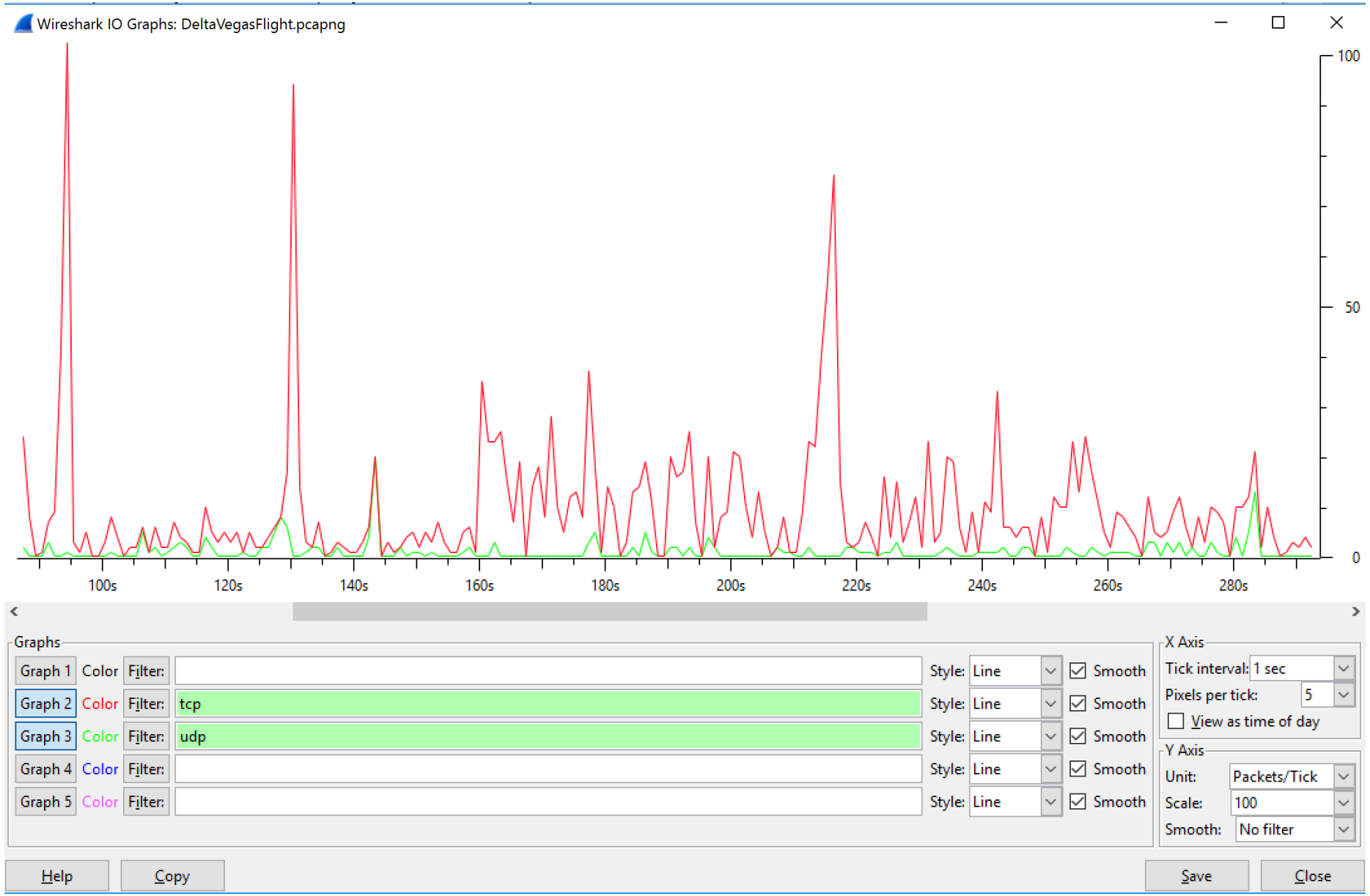
- Statistics -> Show Address Resolution
- Statistics -> Protocol Hierarchy
- Statistics -> Conversations
- Statistics -> Endpoints
- Statistics -> Flow Graph
- IP Statistics -> Source and Dest IP Addresses



Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00 %	3040	100.00 %	1195767	0.022	0	0	0.000
Ethernet	100.00 %	3040	100.00 %	1195767	0.022	0	0	0.000
Internet Protocol Version 4	92.30 %	2806	98.36 %	1176194	0.022	0	0	0.000
Transmission Control Protocol	76.74 %	2333	86.74 %	1037204	0.019	1443	479749	0.009
Secure Sockets Layer	28.22 %	858	50.07 %	598758	0.011	816	546703	0.010
Malformed Packet	0.13 %	4	0.02 %	240	0.000	4	240	0.000
Data	2.04 %	62	0.40 %	4753	0.000	62	4753	0.000
Hypertext Transfer Protocol	0.26 %	8	0.48 %	5759	0.000	7	5699	0.000
JavaScript Object Notation	0.03 %	1	0.01 %	60	0.000	0	0	0.000
Line-based text data	0.03 %	1	0.01 %	60	0.000	1	60	0.000
User Datagram Protocol	15.39 %	468	11.60 %	138720	0.003	0	0	0.000
Teredo IPv6 over UDP tunneling	1.25 %	38	0.40 %	4736	0.000	0	0	0.000
Internet Protocol Version 6	1.25 %	38	0.40 %	4736	0.000	2	164	0.000
Internet Control Message Protocol v6	1.18 %	36	0.38 %	4572	0.000	36	4572	0.000
QUIC (Quick UDP Internet Connections)	7.57 %	230	6.51 %	77846	0.001	230	77846	0.001
Simple Network Management Protocol	2.50 %	76	0.67 %	7952	0.000	76	7952	0.000
Bootstrap Protocol	0.33 %	10	0.30 %	3566	0.000	10	3566	0.000
Data	2.07 %	63	3.09 %	36956	0.001	63	36956	0.001
NetBIOS Name Service	0.30 %	9	0.07 %	828	0.000	9	828	0.000
Link-local Multicast Name Resolution	0.33 %	10	0.06 %	670	0.000	10	670	0.000
Domain Name System	1.05 %	32	0.52 %	6166	0.000	32	6166	0.000
Internet Group Management Protocol	0.16 %	5	0.02 %	270	0.000	5	270	0.000
Address Resolution Protocol	6.71 %	204	0.95 %	11410	0.000	204	11410	0.000
Internet Protocol Version 6	0.99 %	30	0.68 %	8163	0.000	0	0	0.000
User Datagram Protocol	0.82 %	25	0.65 %	7713	0.000	0	0	0.000
Link-local Multicast Name Resolution	0.33 %	10	0.07 %	870	0.000	10	870	0.000
DHCPv6	0.23 %	7	0.09 %	1099	0.000	7	1099	0.000
Data	0.26 %	8	0.48 %	5744	0.000	8	5744	0.000
Internet Control Message Protocol v6	0.16 %	5	0.04 %	450	0.000	5	450	0.000

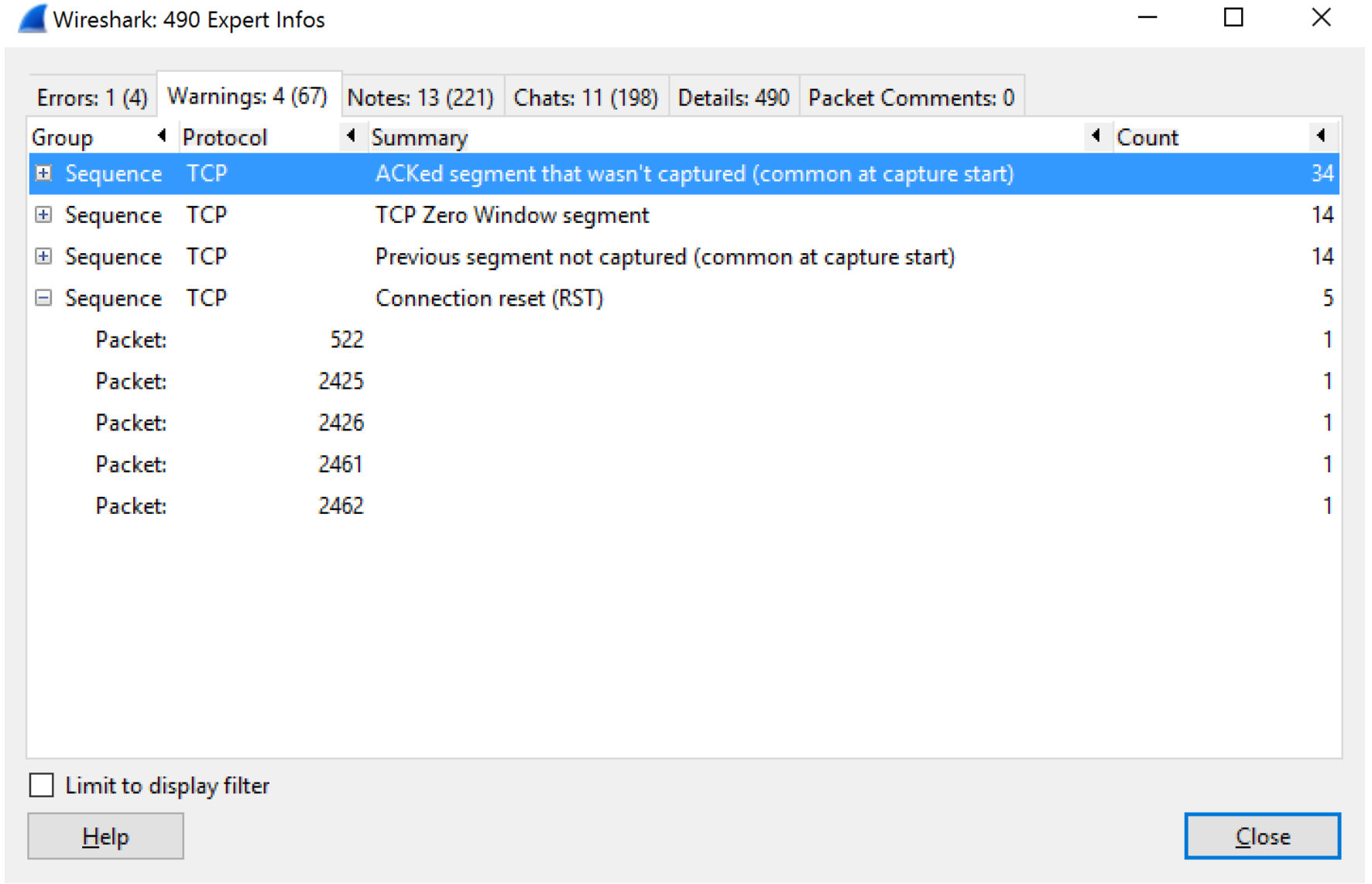
Wireshark IO Graphs

- Click in graph jumps to packet in main display
- Use Filter to show specifics
  - tcp shows just TCP traffic
- Click on Graph 1 button to show / hide
- Y-axis units
  - Packets / tick as default
  - Bytes, bits / tick available


Wireshark IO Graphs: DeltaVegasFlight.pcapng
100
50
0
100s
120s
140s
160s
180s
200s
220s
240s
260s
280s
Graphs
Graph 1 Color Filter: Style: Line Smooth
Graph 2 Color Filter: tcp Style: Line Smooth
Graph 3 Color Filter: udp Style: Line Smooth
Graph 4 Color Filter: Style: Line Smooth
Graph 5 Color Filter: Style: Line Smooth
X Axis
Tick interval: 1 sec
Pixels per tick: 5
View as time of day
Y Axis
Unit: Packets/Tick
Scale: 100
Smooth: No filter
Help
Copy
Save
Close

Expert Infos

- Identifies potential problems
- Warnings Tab
  - Connection reset
  - Duplicate IP address
- Click on entry and jump to packet display


Wireshark: 490 Expert Infos
Errors: 1 (4)
Warnings: 4 (67)
Notes: 13 (221)
Chats: 11 (198)
Details: 490
Packet Comments: 0
Group
Protocol
Summary
Count
Sequence TCP ACKed segment that wasn't captured (common at capture start) 34
Sequence TCP TCP Zero Window segment 14
Sequence TCP Previous segment not captured (common at capture start) 14
Sequence TCP Connection reset (RST) 5
Packet: 522 1
Packet: 2425 1
Packet: 2426 1
Packet: 2461 1
Packet: 2462 1
Limit to display filter
Help
Close

Wireshark Info

- Wiki - https://gitlab.com/wireshark/wireshark/-/wikis/home

Questions?