

# Ethernet 101

Paul Ferrill

ATAC/JT4

[James.Ferrill.ctr@us.af.mil](mailto:James.Ferrill.ctr@us.af.mil)

Paul.Ferrill@avtest.com

# Slides available on Github

- <https://github.com/ATAC/Presentations>
- Ethernet 101 11 Apr 22.pdf

# 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](http://www.ietf.org)
- Request for Comment (RFC)
  - [www.ietf.org/rfc.html](http://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/rfc137.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;
}
```

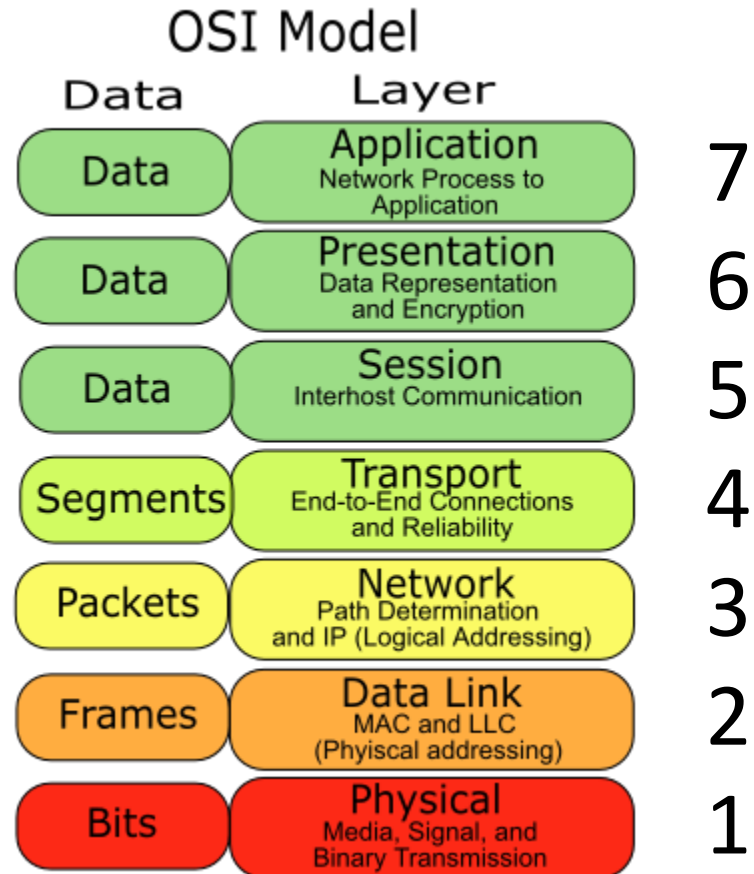
# Questions?

# Ethernet Basics

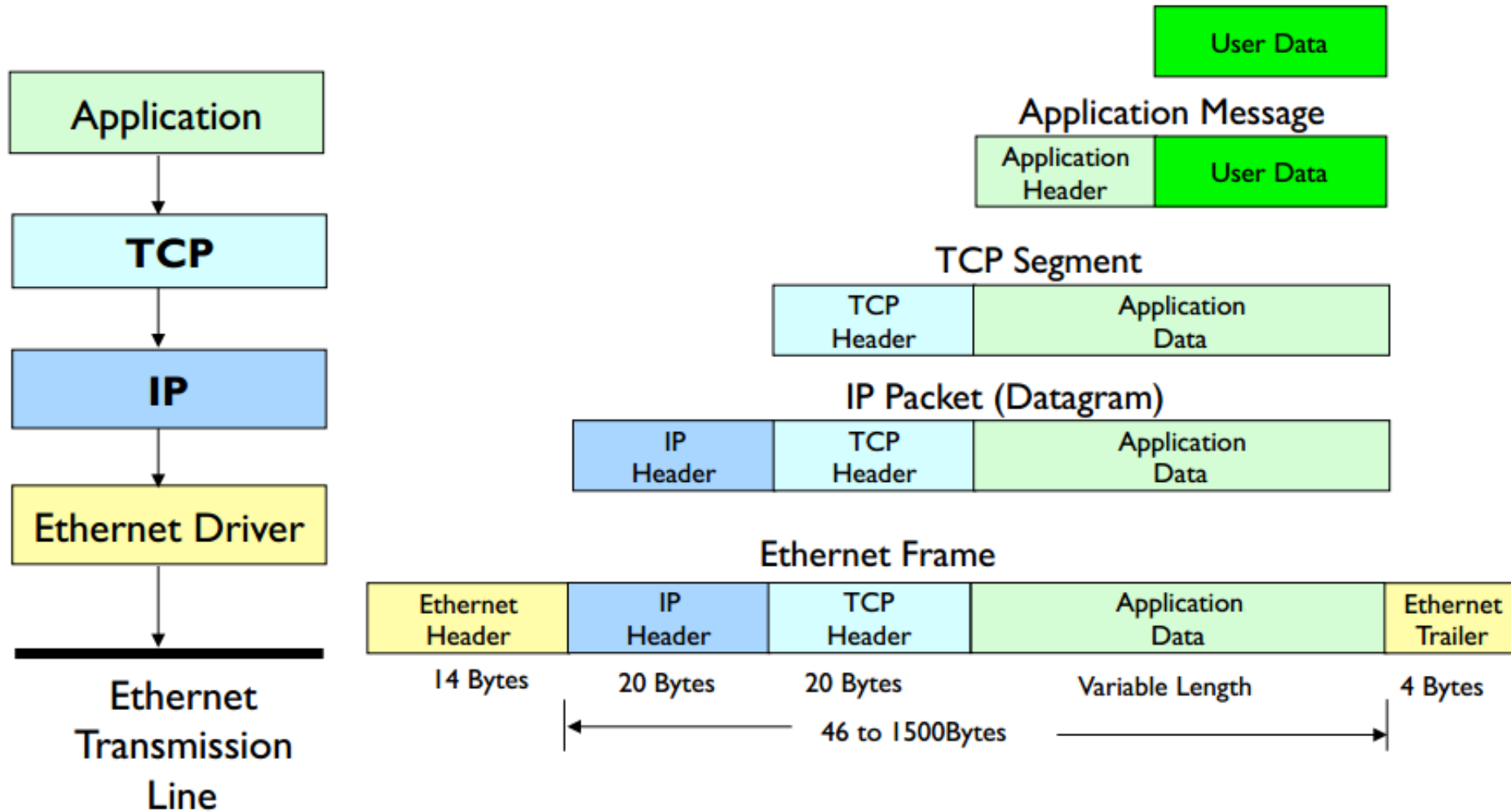
# Overview

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

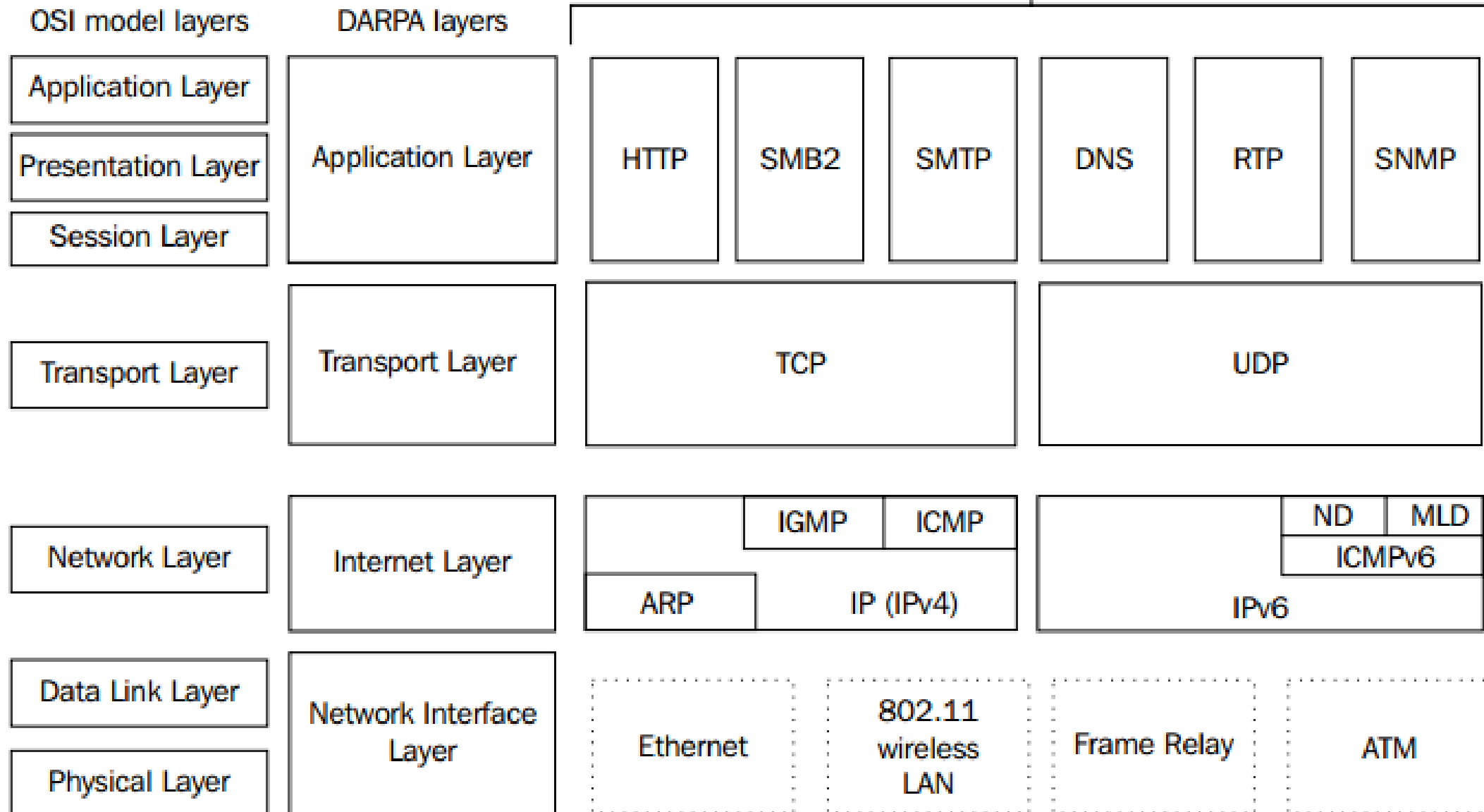
# OSI Model



# Data in Layers

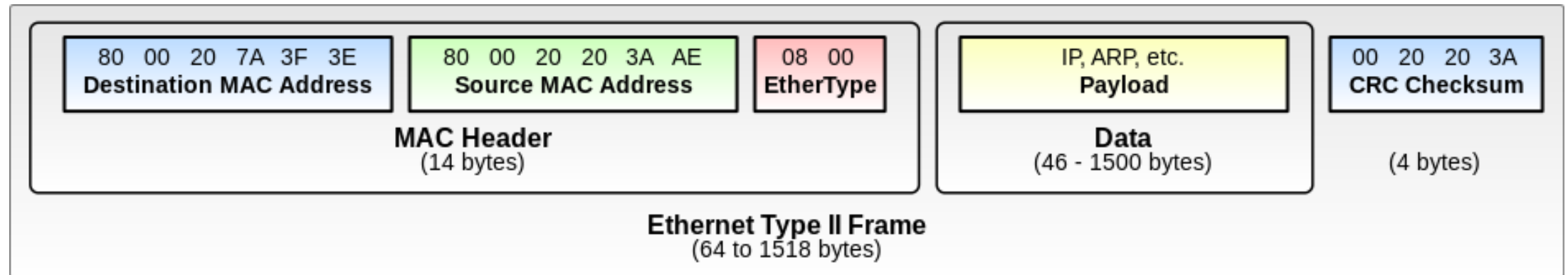


# TCP/IP Protocol Suite





# 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

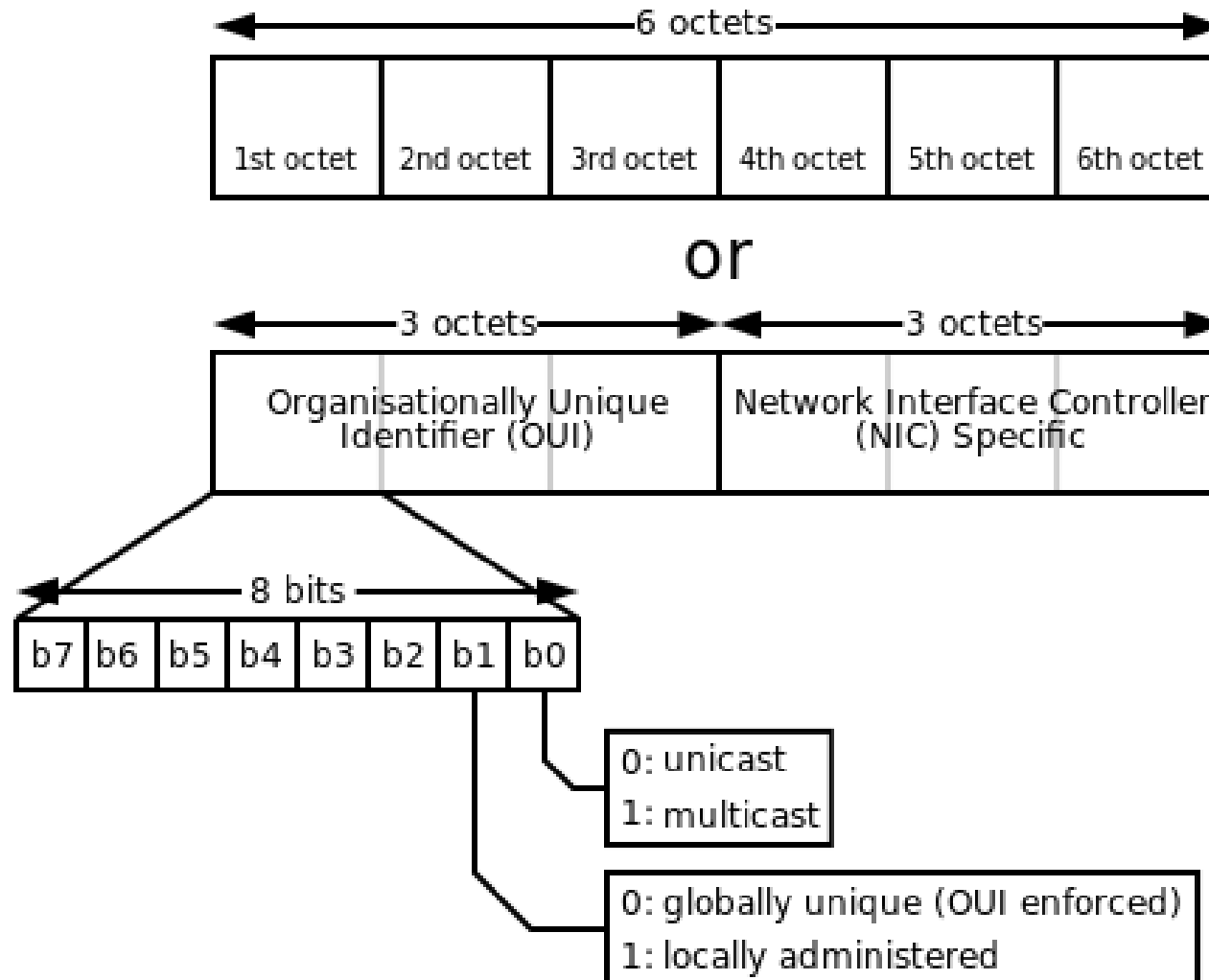
- ▼ Ethernet II, Src: Netgear\_3a:73:5d (c4:04:15:3a:73:5d), Dst: Dell\_60:65:ed (90:b1:1c:60:65:ed)
  - ▼ Destination: Dell\_60:65:ed (90:b1:1c:60:65:ed)
    - Address: Dell\_60:65:ed (90:b1:1c:60:65:ed)
    - .... ..0. .... = LG bit: Globally unique address (factory default)
    - .... ....0 .... = IG bit: Individual address (unicast)
  - ▼ Source: Netgear\_3a:73:5d (c4:04:15:3a:73:5d)
    - Address: Netgear\_3a:73:5d (c4:04:15:3a:73:5d)
    - .... ..0. .... = LG bit: Globally unique address (factory default)
    - .... ....0 .... = IG bit: Individual address (unicast)
- Type: IPv4 (0x0800)
- Padding: 00000000000000

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.....

# MAC Addresses

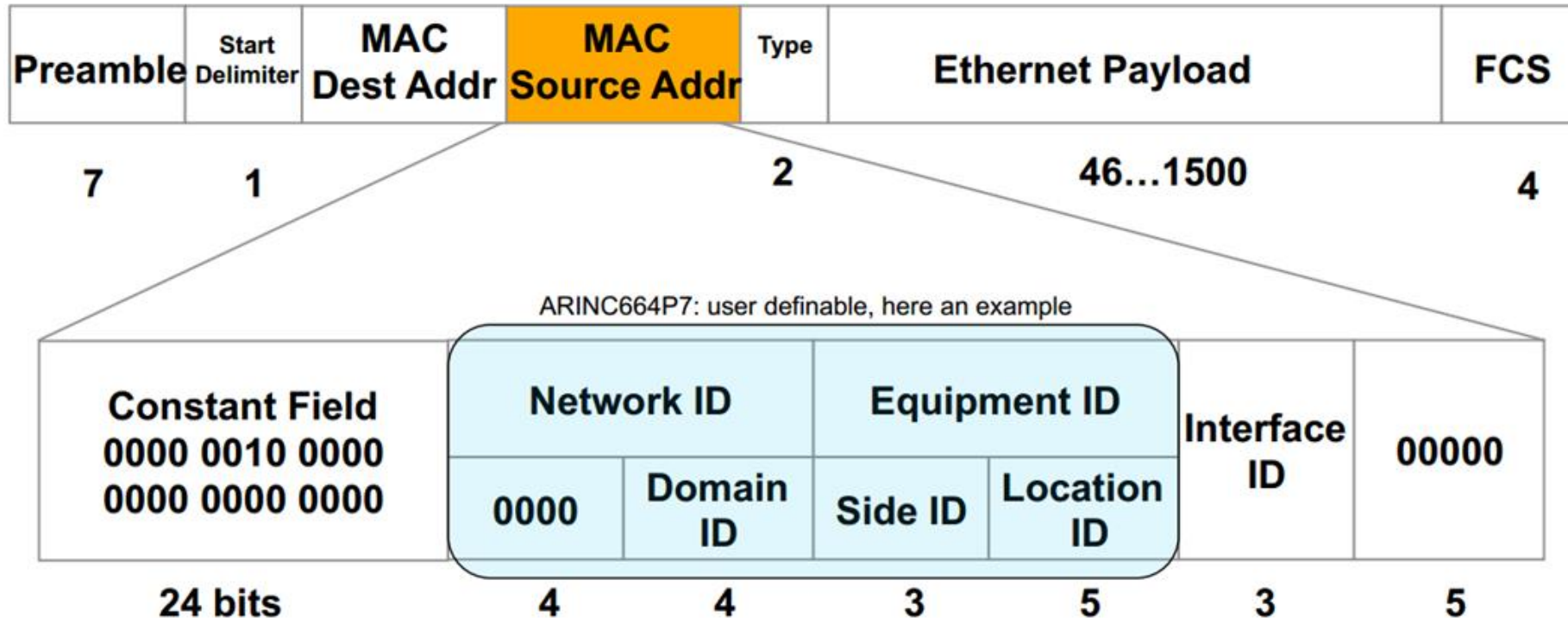
- Six bytes of information
  - 00-1D-92-98-36-8A
- Globally Unique
  - Conflicts not allowed
- First three bytes = OUI = Vendor ID
  - Organizationally Unique Identifier – assigned by IEEE
  - 00:1D:92 = Micro-star International
  - <http://aruljohn.com/mac/001D92>

# Hardware (MAC) Address

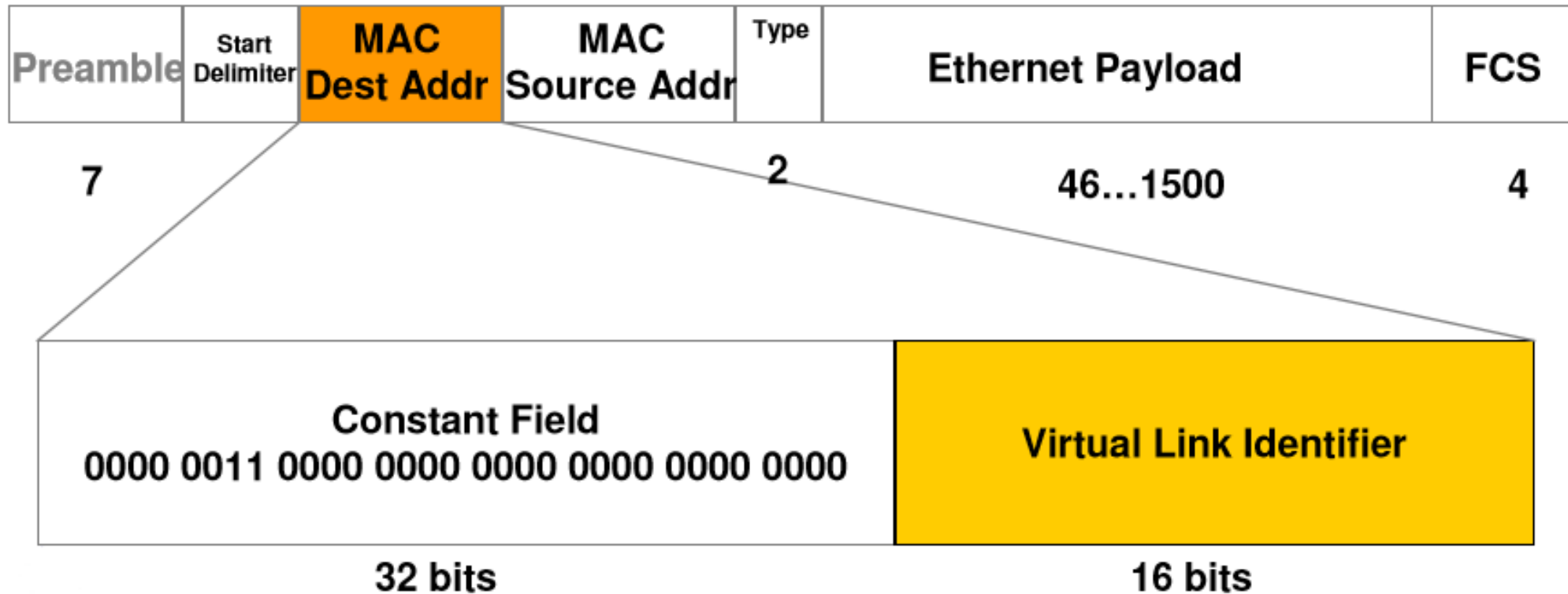


# ARINC-664

- **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

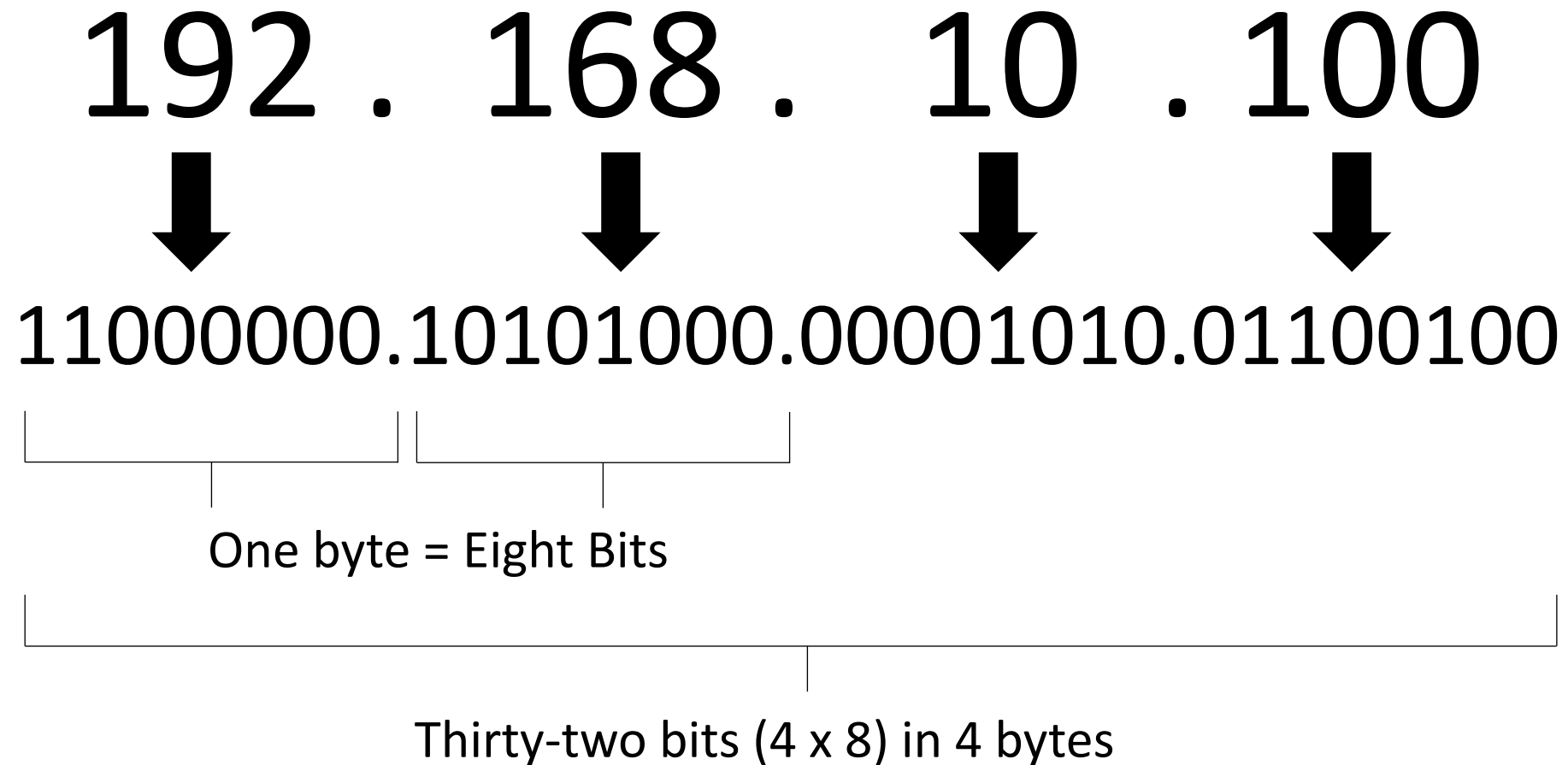
# Packet Fundamentals

- 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

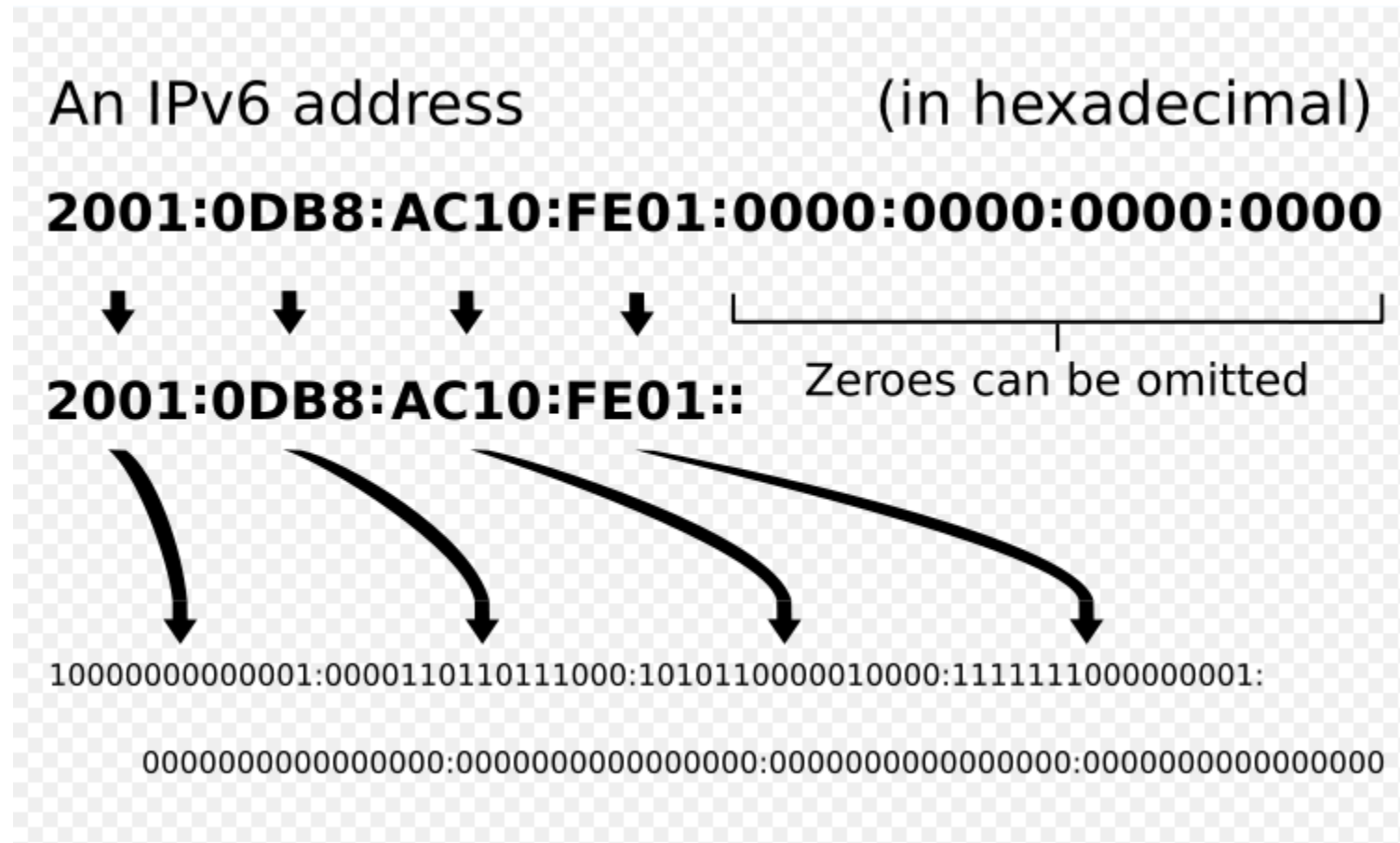
# IP Addressing Basics

- 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

Version	Header Length	Differentiated Services	Total Length
Identification			Fragment Info
Time to Live		Protocol	Header Checksum
Source Address			
Destination Address			
Multiple 32 bit words of "Options"			
Data			

# 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](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 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 known 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

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%)

Dropped: 0

# 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
▼ 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: 0000000000000000000000000000000000000000000000000000000000000000
▼ 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

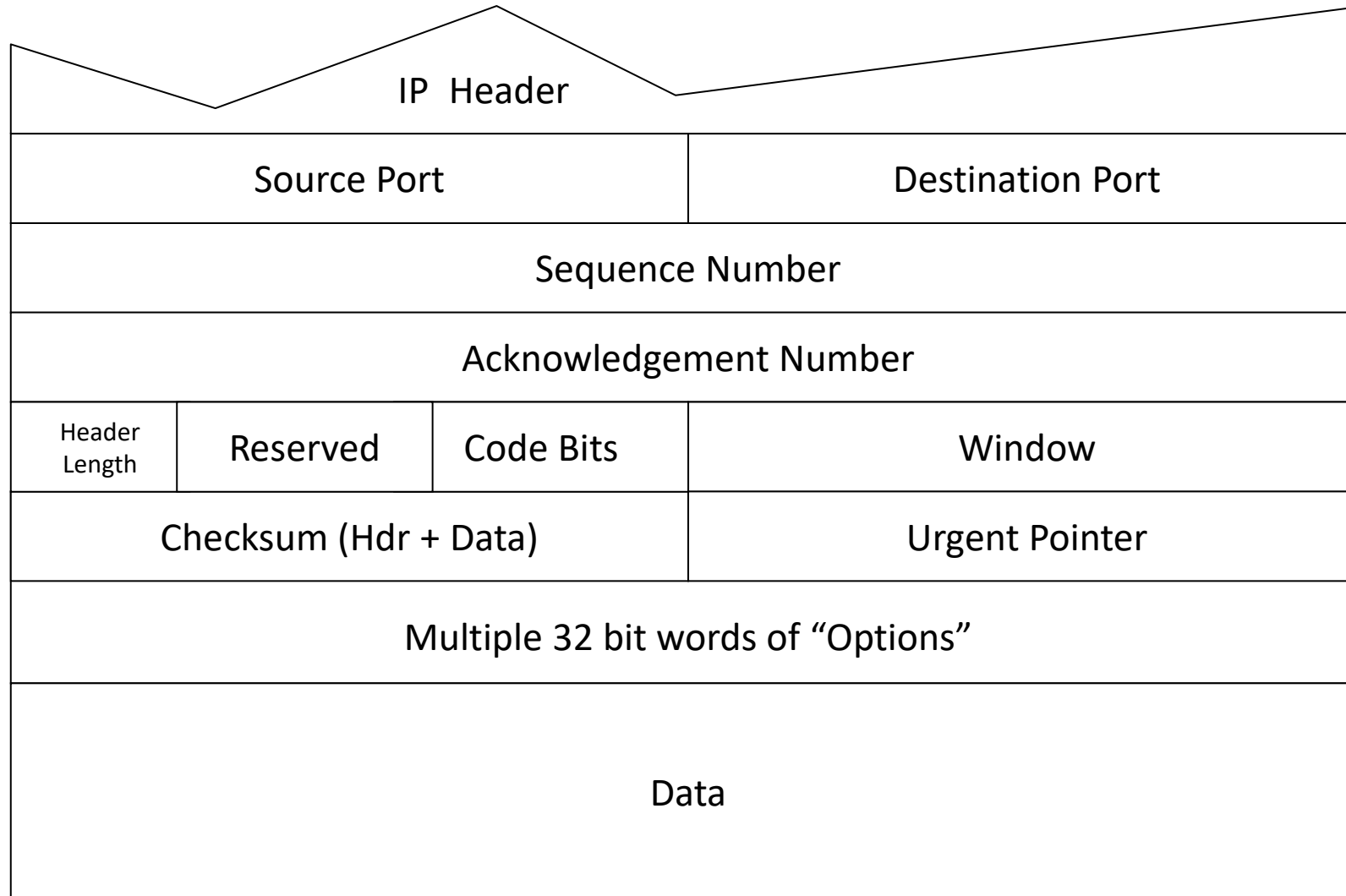
# TCP

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

# TCP Distinctions

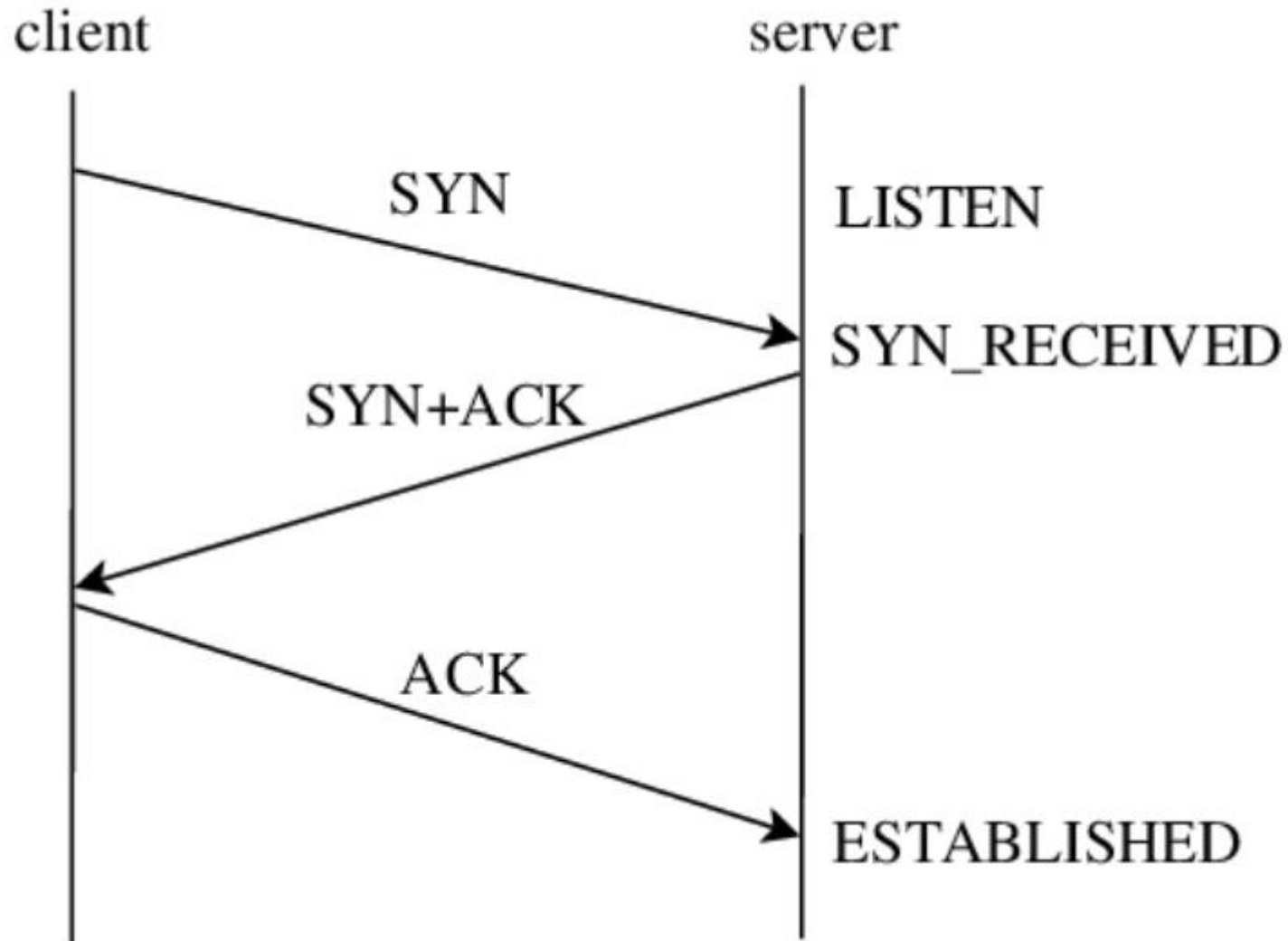
- Ordered data transfer – sequence number used to reassemble packets
- Retransmission of lost packets – not acknowledged packets resent
- Error-free data transfer – checksum used to ensure reliable transfer
- Flow control – limits transfer rate to ensure reliable delivery
- Congestion control

# TCP PDU Format

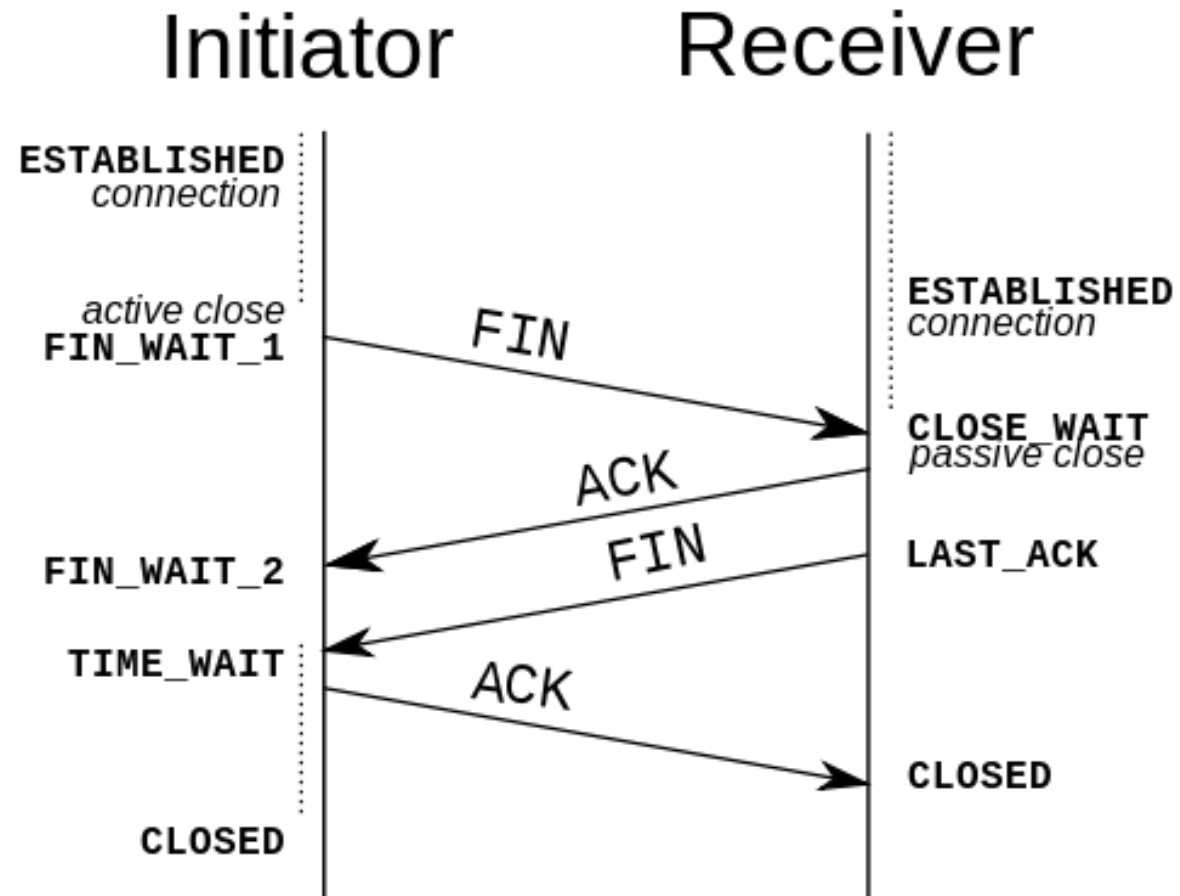




# Connection Establishment Diagram



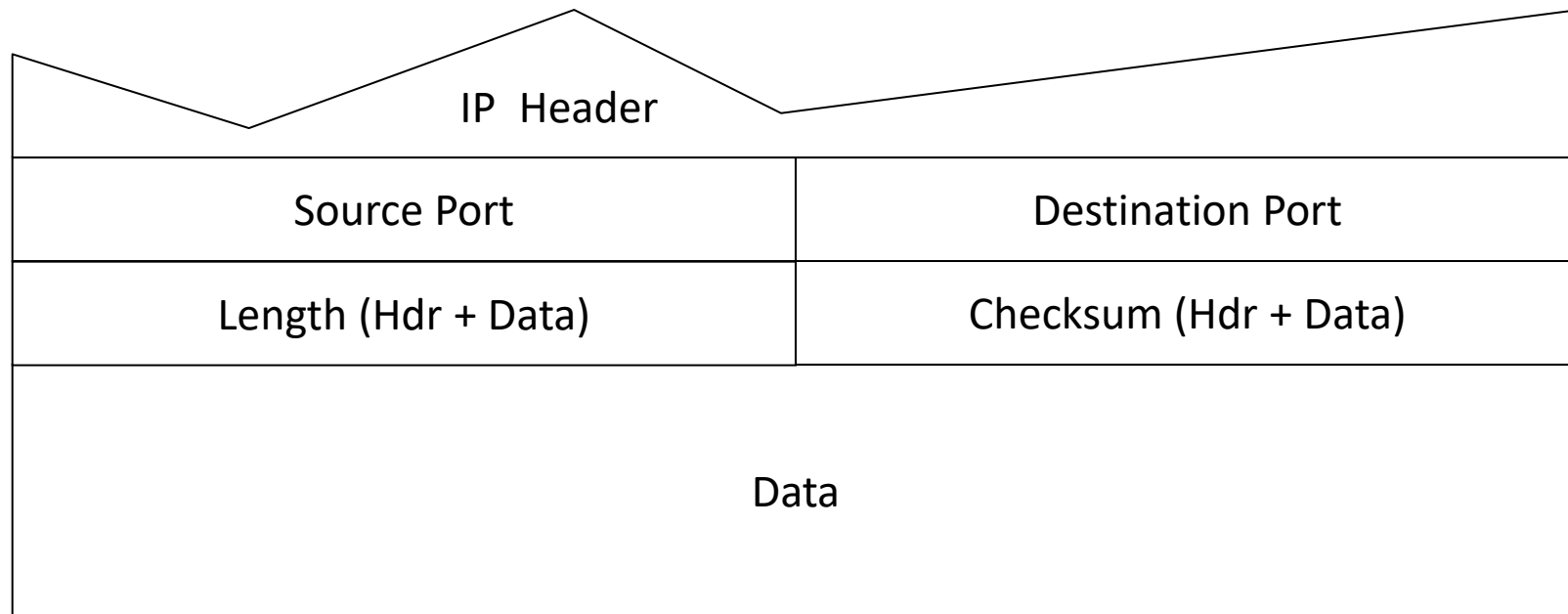
# Connection Termination Diagram



# 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

# UDP 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

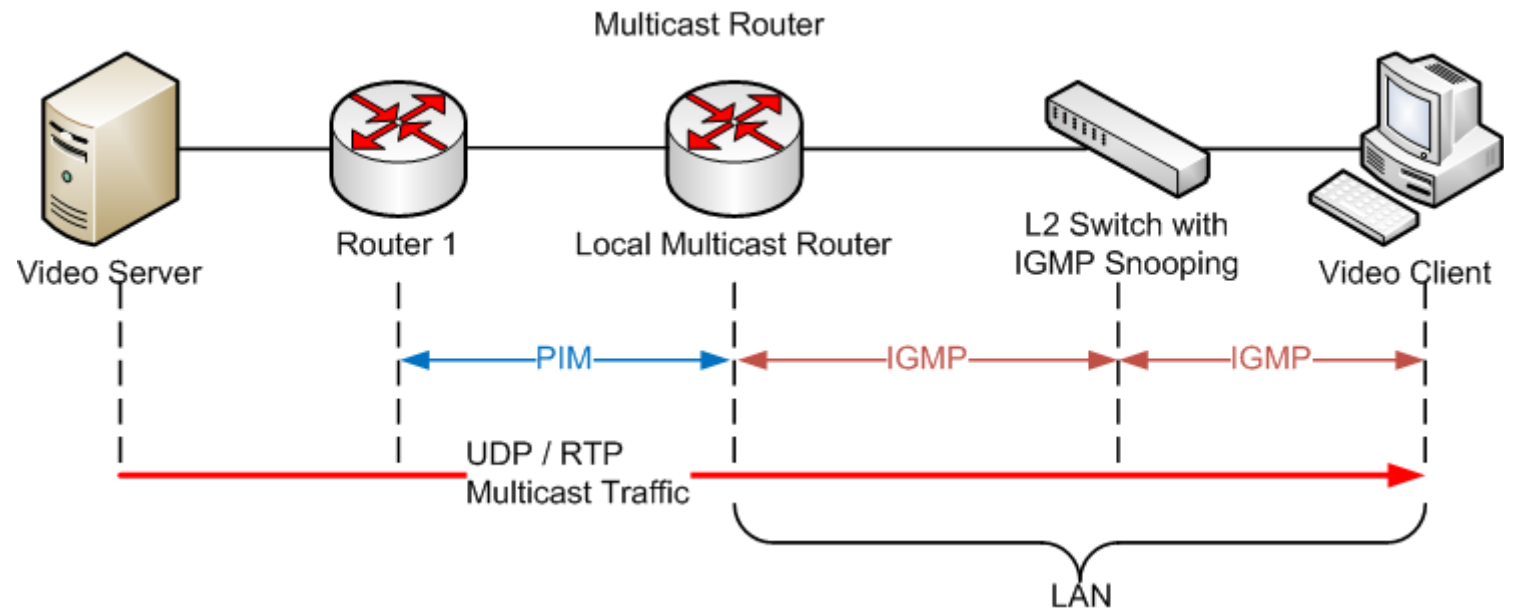
- The group includes the addresses from 224.0.0.0 to 239.255.255.255
- Addresses in the range of 224.0.0.0 to 224.0.0.255 are individually assigned by IANA and designated for multicasting on the local subnetwork only.
- Addresses in the range 224.0.1.0 to 224.0.1.255 are individually assigned by IANA 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

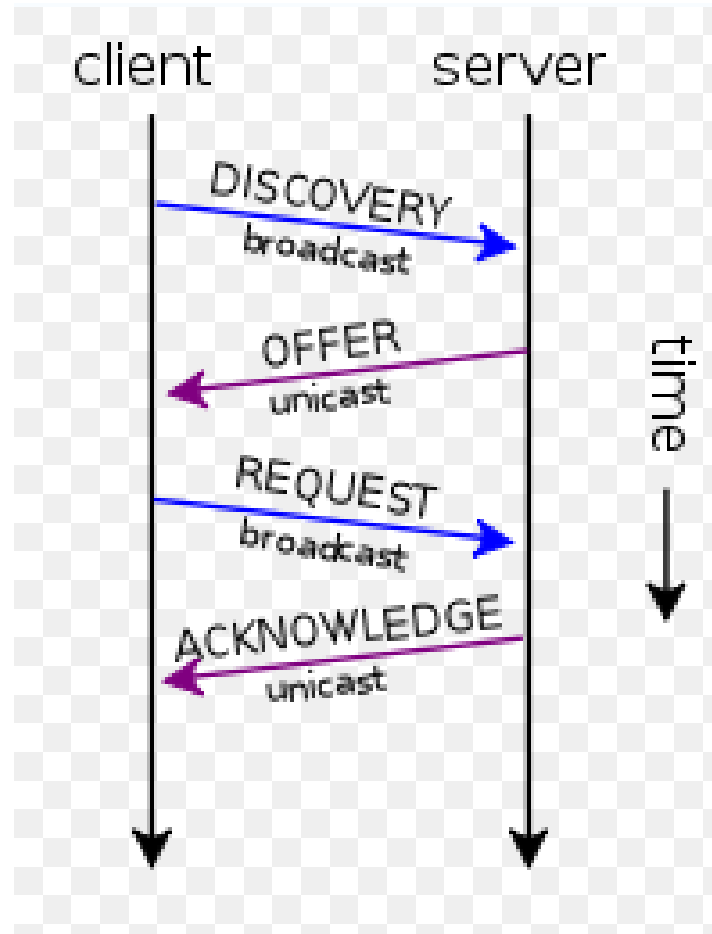
# Domain Name System (DNS)

- 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

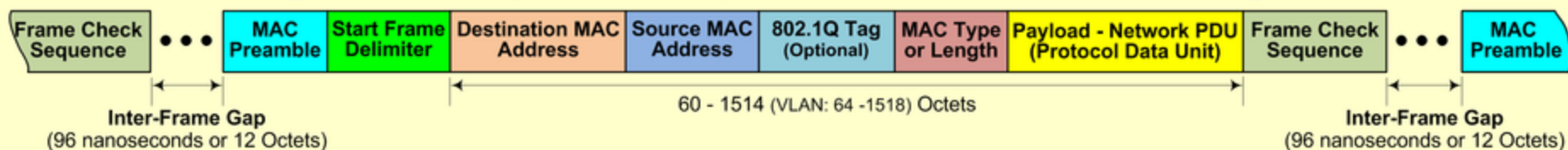
# DHCP

- 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

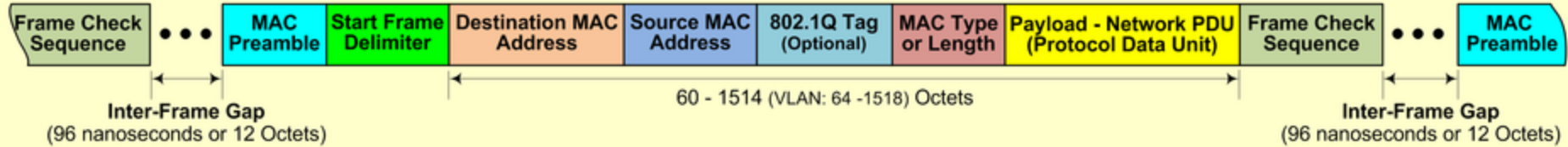


Gigabit Ethernet Frame Component Size with TCP/IP Datagram		
Frame Component	Component Size	
<b>MAC Preamble</b>	7 Octets of: <b>10101010</b>	
<b>Start Frame Delimiter</b>	1 Octet of: <b>10101011</b>	
<b>Destination MAC Address</b>	6 Octets	
<b>Source MAC Address</b>	6 Octets	
<b>802.1Q VLAN TAG ID (Optional)</b>	4 Octets (Optional)	
<b>MAC Type or Length</b>	2 Octets	
<b>MTU</b> (Maximum Transmission Unit)  <b>Payload</b> Network <b>PDU</b> Protocol Data Unit:	<b>IP Header</b>	20 Octets
	<b>TCP Header</b>	20 Octets
	<b>TCP Options/Data/Padding</b>	6 - 1460 Octets
	<b>***Total:</b>	<b>46 - 1500 Octets</b> (Max: 1504 - VLAN)
<b>Frame Check Sequence (CRC)</b>	4 Octets	
<b>Inter-Frame Gap</b> • • •	12 Octets (96 nanoseconds)	
<b>Total Physical Frame Size:</b>	<b>84 - 1538 Octets</b> (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
<b>Max Octet Rate</b>	(1000Mb/sec)/((8 Bits) = <b>125,000,000 Octets/sec</b>
<b>Max Frame Rate</b> (84 Octet Frames) Min Packet (60 Bytes + 4 Bytes CRC)	(1000Mb/sec)/((8 Bits)*(84 Octets/Frame)) = <b>1,488,095 Frames/sec (FPS)</b>
<b>Max TCP/IP Data Rate</b> (84 Octet Frames) Min TCP/IP Packet (60 Bytes + 4 Bytes CRC)	(1,488,095 Frames/sec)*(6 Bytes/Frame) = <b>8,928,571 Bytes/sec</b>
<b>Max Frame Rate</b> (1538 Octet Frames) Max Packet (1514 Bytes + 4 Bytes CRC)	(1000Mb/sec)/((8 Bits)*(1538 Octets/Frame)) = <b>81,274 Frames/sec (FPS)</b>
<b>Max TCP/IP Data Rate</b> (1538 Octet Frames) Max TCP/IP Packet (1514 Bytes + 4 Bytes CRC)	(81,274 Frames/sec)*(1448 Bytes/Frame) = <b>117,685,306 Bytes/sec (TCP/IP TimeStamp)</b>
<b>Max TCP/IP Data Rate</b> (1538 Octet Frames) Max TCP/IP Packet (1514 Bytes + 4 Bytes CRC)	(81,274 Frames/sec)*(1460 Bytes/Frame) = <b>118,660,598 Bytes/sec (no TCP/IP TimeStamp)</b>
<b>Max Gigabit Ethernet Frame Bandwidth</b> Max Packet (60 Bytes + 4 Bytes CRC) Max Packet (60 Bytes)	(1,488,095 Frames/sec)*(64 Bytes/Frame) = <b>95,238,080 Bytes/sec</b> ( 90.876031 MiB/s) (1,488,095 Frames/sec)*(60 Bytes/Frame) = <b>89,285,700 Bytes/sec</b> ( 85.149477 MiB/s)
<b>Max Gigabit Ethernet Frame Bandwidth</b> Max Packet (1514 Bytes + 4 Bytes CRC) Max Packet (1514 Bytes)	(81,274 Frames/sec)*(1518 Bytes/Frame) = <b>123,373,932 Bytes/sec</b> (117.658550 MiB/s) (81,274 Frames/sec)*(1514 Bytes/Frame) = <b>123,048,836 Bytes/sec</b> (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	
<b>MTU</b> (Maximum Transmission Unit)  <b>Payload</b> Network PDU Protocol Data Unit:	IP Header	20 Octets
	UDP Header	8 Octets
	Data/Padding	18 - 1472 Octets
	<b>***Total:</b>	<b>46 - 1500 Octets</b> (Max: 1504 - VLAN)
Frame Check Sequence (CRC)	4 Octets	
Inter-Frame Gap	12 Octets (96 nanoseconds)	
<b>Total Physical Frame Size:</b>	<b>84 - 1538 Octets</b> (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
<b>Max Octet Rate</b>	$(1000\text{Mb/sec}) / (8\text{ Bits}) = 125,000,000\text{ Octets/sec}$
<b>Max Frame Rate</b> (84 Octet Frames) Min Packet (60 Bytes + 4 Bytes CRC)	$(1000\text{Mb/sec}) / (8\text{ Bits}) * (84\text{ Octets/Frame}) = 1,488,095\text{ Frames/sec (FPS)}$
<b>Max UDP Data Rate</b> (84 Octet Frames) Min UDP Packet (60 Bytes + 4 Bytes CRC)	$(1,488,095\text{ Frames/sec}) * (18\text{ Bytes/Frame}) = 26,785,714\text{ Bytes/sec}$
<b>Max Frame Rate</b> (1538 Octet Frames) Max Packet (1514 Bytes + 4 Bytes CRC)	$(1000\text{Mb/sec}) / (8\text{ Bits}) * (1538\text{ Octets/Frame}) = 81,274\text{ Frames/sec (FPS)}$
<b>Max UDP Data Rate</b> (1538 Octet Frames) Max UDP Packet (1514 Bytes + 4 Bytes CRC)	$(81,274\text{ Frames/sec}) * (1472\text{ Bytes/Frame}) = 119,635,891\text{ Bytes/sec}$
<b>Max Gigabit Ethernet Frame Bandwidth</b> Max Packet (60 Bytes + 4 Bytes CRC) Max Packet (60 Bytes)	$(1,488,095\text{ Frames/sec}) * (64\text{ Bytes/Frame}) = 95,238,080\text{ Bytes/sec (90.876031 MiB/s)}$ $(1,488,095\text{ Frames/sec}) * (60\text{ Bytes/Frame}) = 89,285,700\text{ Bytes/sec (85.149477 MiB/s)}$
<b>Max Gigabit Ethernet Frame Bandwidth</b> Max Packet (1514 Bytes + 4 Bytes CRC) Max Packet (1514 Bytes)	$(81,274\text{ Frames/sec}) * (1518\text{ Bytes/Frame}) = 123,373,932\text{ Bytes/sec (117.658550 MiB/s)}$ $(81,274\text{ Frames/sec}) * (1514\text{ Bytes/Frame}) = 123,048,836\text{ Bytes/sec (117.348515 MiB/s)}$

- \*\*\* **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

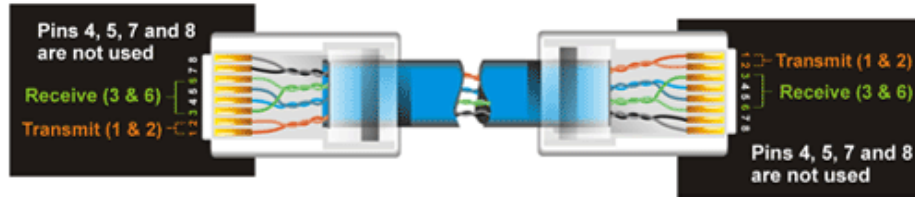
# Layer 1 Hardware

- 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

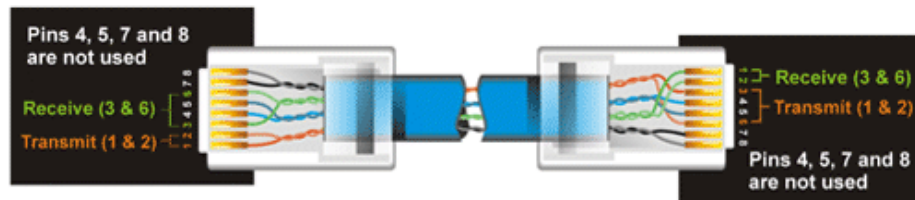
# Network Interface Card (NIC)



# Cabling

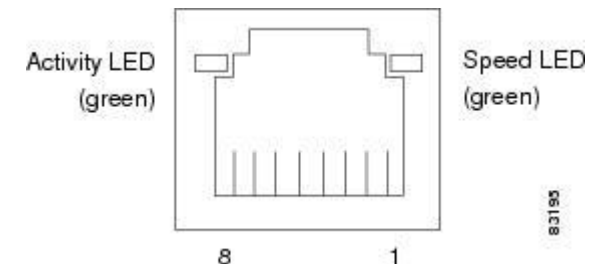
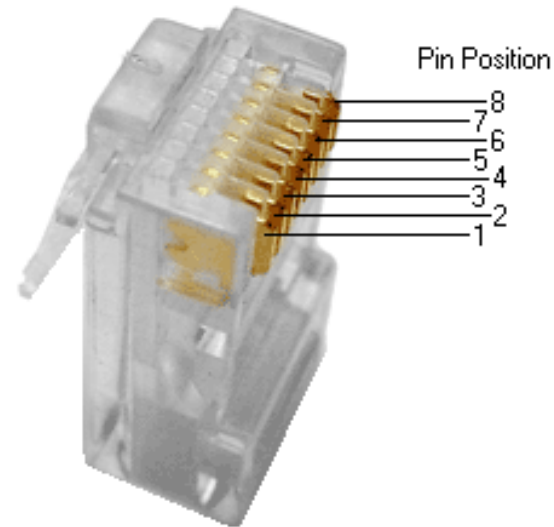


Pin number	Wire Color	Straight-Through		Pin number	Wire Color
Pin 1 ==>	Orange/White	Wire	Becomes	Pin 1 ==>	Orange/White
Pin 2 ==>	Orange	1	1	Pin 2 ==>	Orange
Pin 3 ==>	Green/White	2	2	Pin 3 ==>	Green/White
Pin 4 ==>	Blue	3	3	Pin 4 ==>	Blue
Pin 5 ==>	Blue/White	6	6	Pin 5 ==>	Blue/White
Pin 6 ==>	Green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

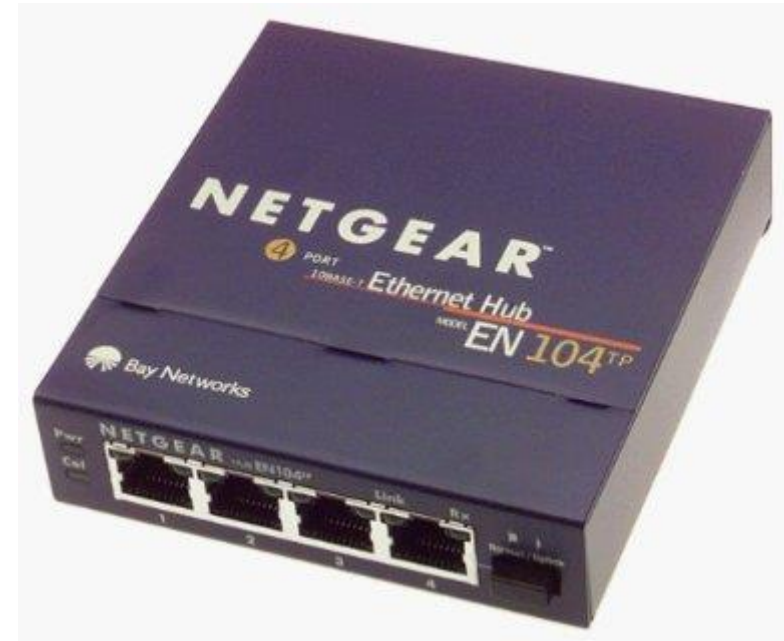


Pin number	Wire Color	Crossed-Over		Pin number	Wire Color
Pin 1 ==>	Orange/White	Wire	Becomes	Pin 1 ==>	Green/White
Pin 2 ==>	Orange	1	3	Pin 2 ==>	Green
Pin 3 ==>	Green/White	2	6	Pin 3 ==>	Orange/White
Pin 4 ==>	Blue	3	1	Pin 4 ==>	Blue
Pin 5 ==>	Blue/White	6	2	Pin 5 ==>	Blue/White
Pin 6 ==>	Green			Pin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

Standards  
EIA/TIA T568A/B



# Ethernet Hub



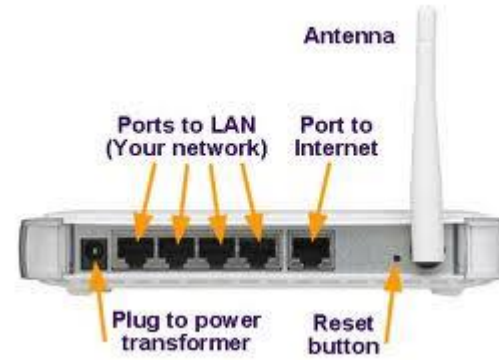
# Layer 2 Hardware

- 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

Network Setup

Router IP

Network Address  
Server Settings  
(DHCP)

Time Setting

Local IP  
Address: 192 . 168 . 2 . 1

Subnet Mask: 255 . 255 . 255 . 0

DHCP Server: ☒ Enable ☐ Disable

Starting IP  
Address: 192.168.2.100

Maximum  
Number of  
DHCP Users: 50

Client Lease  
Time: 0 minutes (0 means one day)

Static DNS 1: 208 . 67 . 222 . 222

Static DNS 2: 208 . 67 . 220 . 220

Static DNS 3: 4 . 2 . 2 . 1

WINS: 0 . 0 . 0 . 0

Time Zone:  
(GMT-05:00) Eastern Time(USA & Canada)

☒ Automatically adjust clock for daylight saving changes

This is the address of  
the router.

**Subnet Mask:** This is  
the subnet mask of the  
router.

**DHCP Server:** Allows  
the router to manage  
your IP addresses.


**Starting IP Address:**  
The address you would  
like to start with.

**Maximum number of  
DHCP Users:** You may  
limit the number of  
addresses your router  
hands out.  
**More...**

**Time Setting:** Choose  
the time zone you are  
in. The router can also  
adjust automatically for  
daylight savings time.

Save Settings

Cancel Changes





# 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

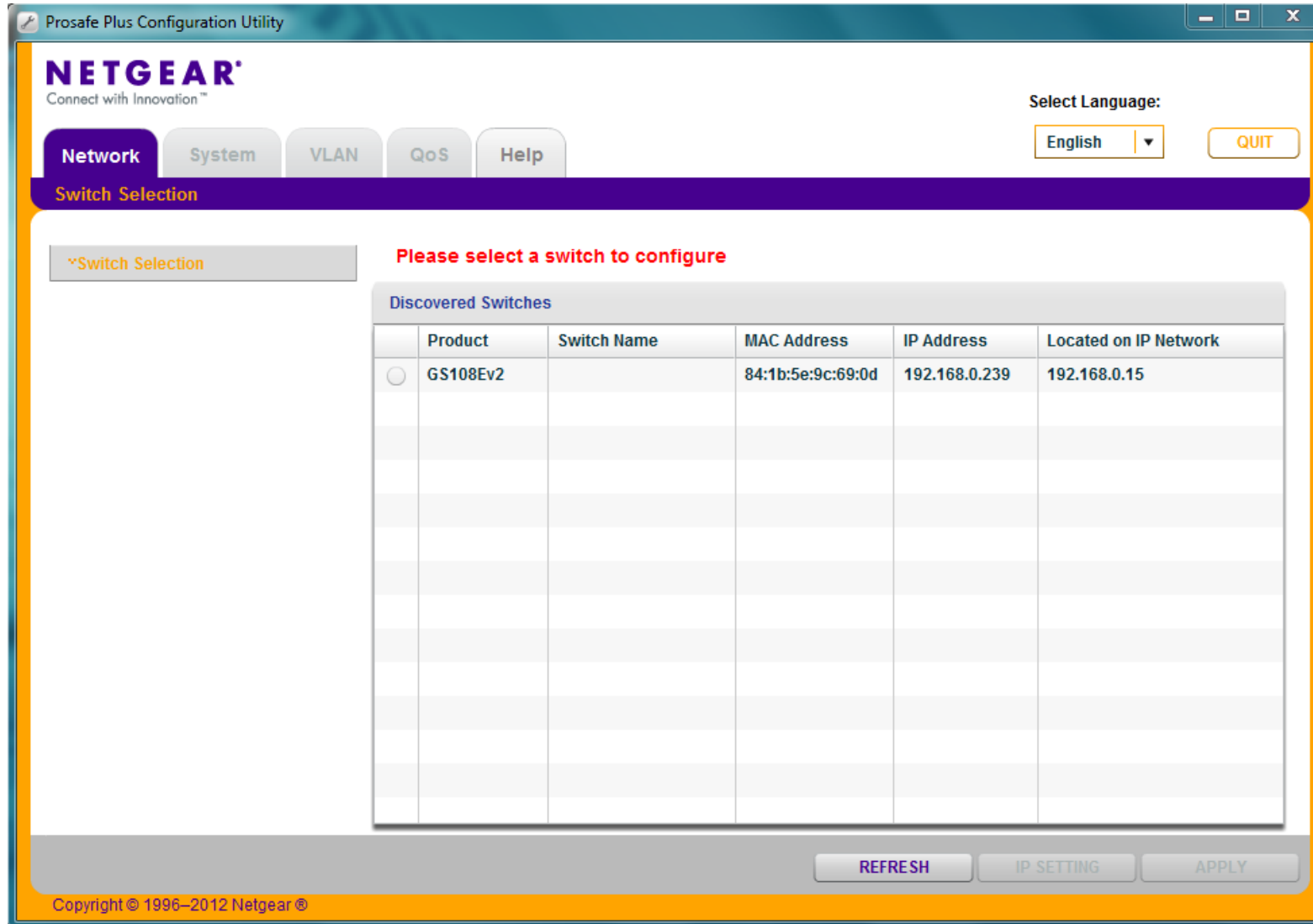


D-Link DGS-3200-10 Switch



HP PS1810-8G Switch

# Discovery application



# Switch configuration

ProSAFE Plus Configuration Utility-GS108Ev2

**NETGEAR**  
Connect with Innovation™

GS108Ev2

Select Language: English

QUIT

Network System VLAN QoS Help

Status Maintenance Monitoring MultiCast

Switch Status  
Switch Information

Selected Switch

	Product	Switch Name	MAC Address	IP Address	Located on IP Network
<input checked="" type="radio"/>	GS108Ev2		84:1b:5e:9c:69:0d	192.168.0.239	192.168.0.15

Port Status

Port	Port Status	Linked Speed
01	Up	1000M
02	Down	No Speed
03	Down	No Speed
04	Down	No Speed
05	Down	No Speed
06	Down	No Speed
07	Down	No Speed
08	Down	No Speed

REFRESH APPLY

Copyright © 1996–2012 Netgear ©

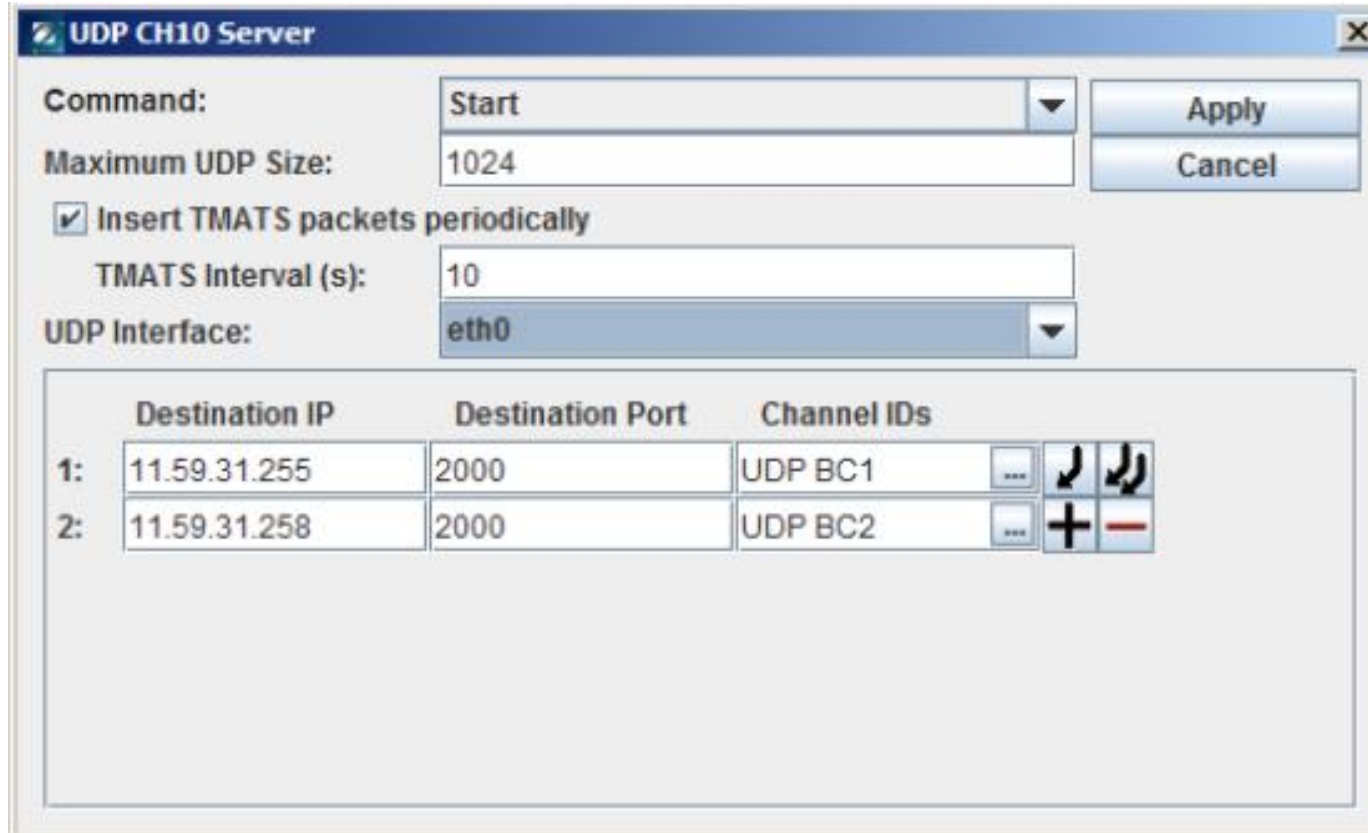
# VLANs

- 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



The dialog box is titled "UDP CH10 Server". It contains the following fields and controls:

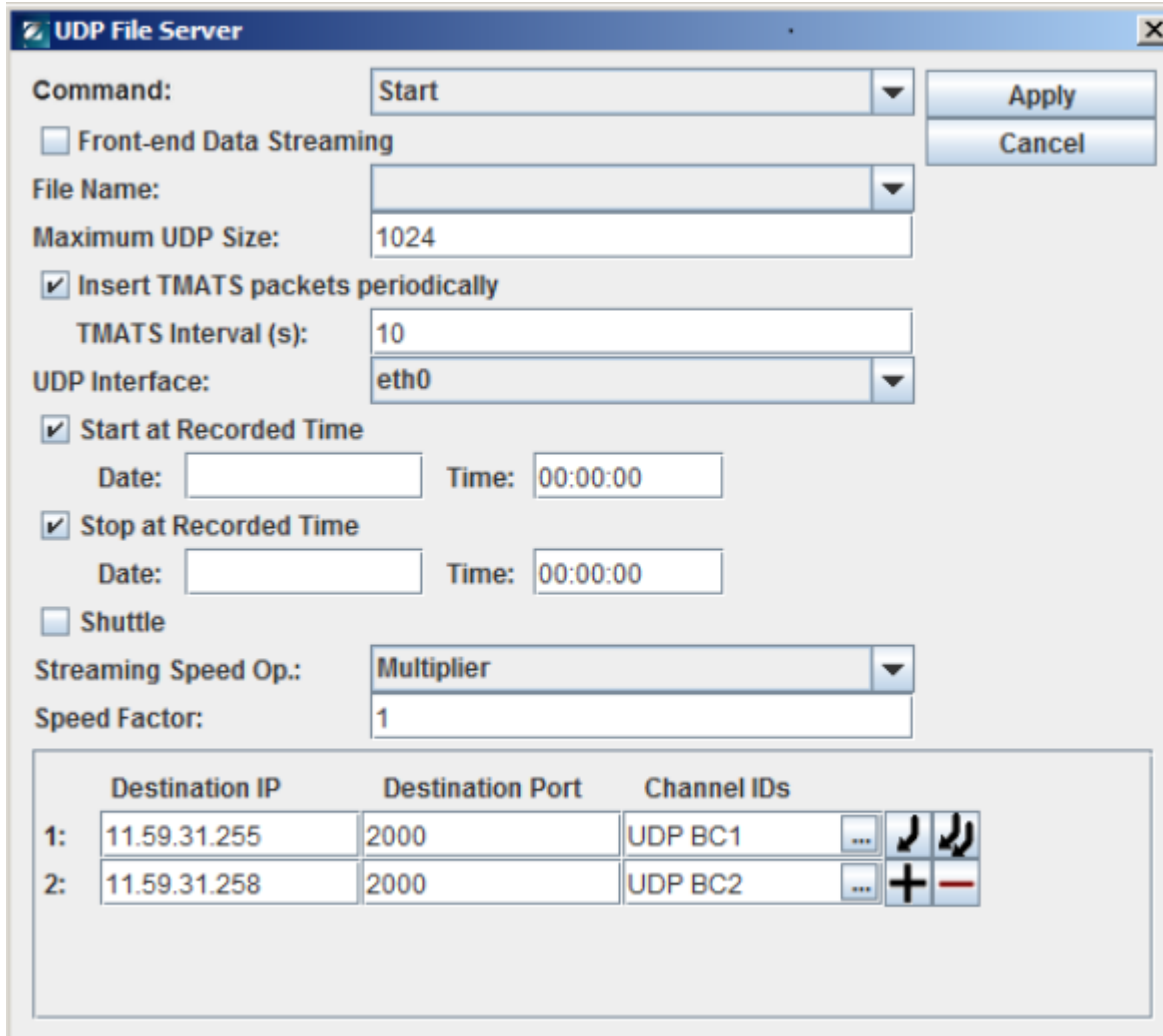
- Command:** A dropdown menu with "Start" selected.
- Maximum UDP Size:** A text field containing "1024".
- Insert TMATS packets periodically:** A checked checkbox.
- TMATS Interval (s):** A text field containing "10".
- UDP Interface:** A dropdown menu with "eth0" selected.
- Buttons:** "Apply" and "Cancel" buttons are located to the right of the "Maximum UDP Size" field.
- Table:** A table with three columns: "Destination IP", "Destination Port", and "Channel IDs". It contains two rows of data.

	Destination IP	Destination Port	Channel IDs
1:	11.59.31.255	2000	UDP BC1
2:	11.59.31.258	2000	UDP BC2

Multiple destination IP addresses allowed

Dialog has an error – can you spot it?

# UDP File Server



The screenshot shows the 'UDP File Server' configuration window. It includes fields for Command (Start), File Name, Maximum UDP Size (1024), and UDP Interface (eth0). There are checkboxes for 'Front-end Data Streaming', 'Insert TMATS packets periodically' (checked), 'Start at Recorded Time' (checked), and 'Stop at Recorded Time' (checked). The 'Shuttle' checkbox is unchecked. The 'Streaming Speed Op.' is set to 'Multiplier' and the 'Speed Factor' is 1. At the bottom, there is a table with two rows of destination IP, port, and channel ID information.

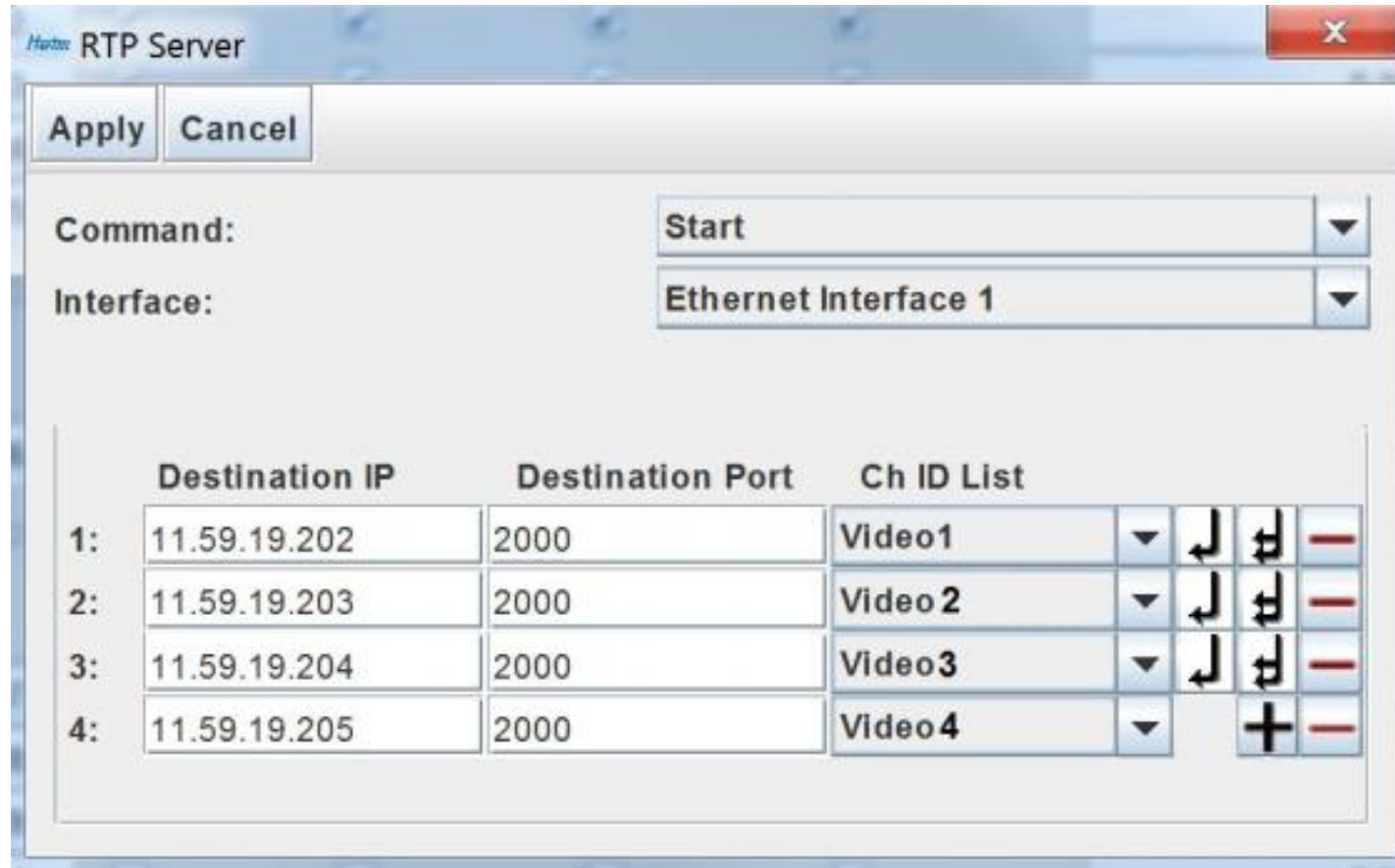
	Destination IP	Destination Port	Channel IDs
1:	11.59.31.255	2000	UDP BC1
2:	11.59.31.258	2000	UDP BC2

File Server uses UDP for transfer, not FTP

Same IP address typo



# RTP Server



	Destination IP	Destination Port	Ch ID List				
1:	11.59.19.202	2000	Video1	▼	↩	⌂	—
2:	11.59.19.203	2000	Video 2	▼	↩	⌂	—
3:	11.59.19.204	2000	Video3	▼	↩	⌂	—
4:	11.59.19.205	2000	Video4	▼		+	—

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

- PSTools Docs
  - <https://docs.microsoft.com/en-us/sysinternals/downloads/pstools>
- 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

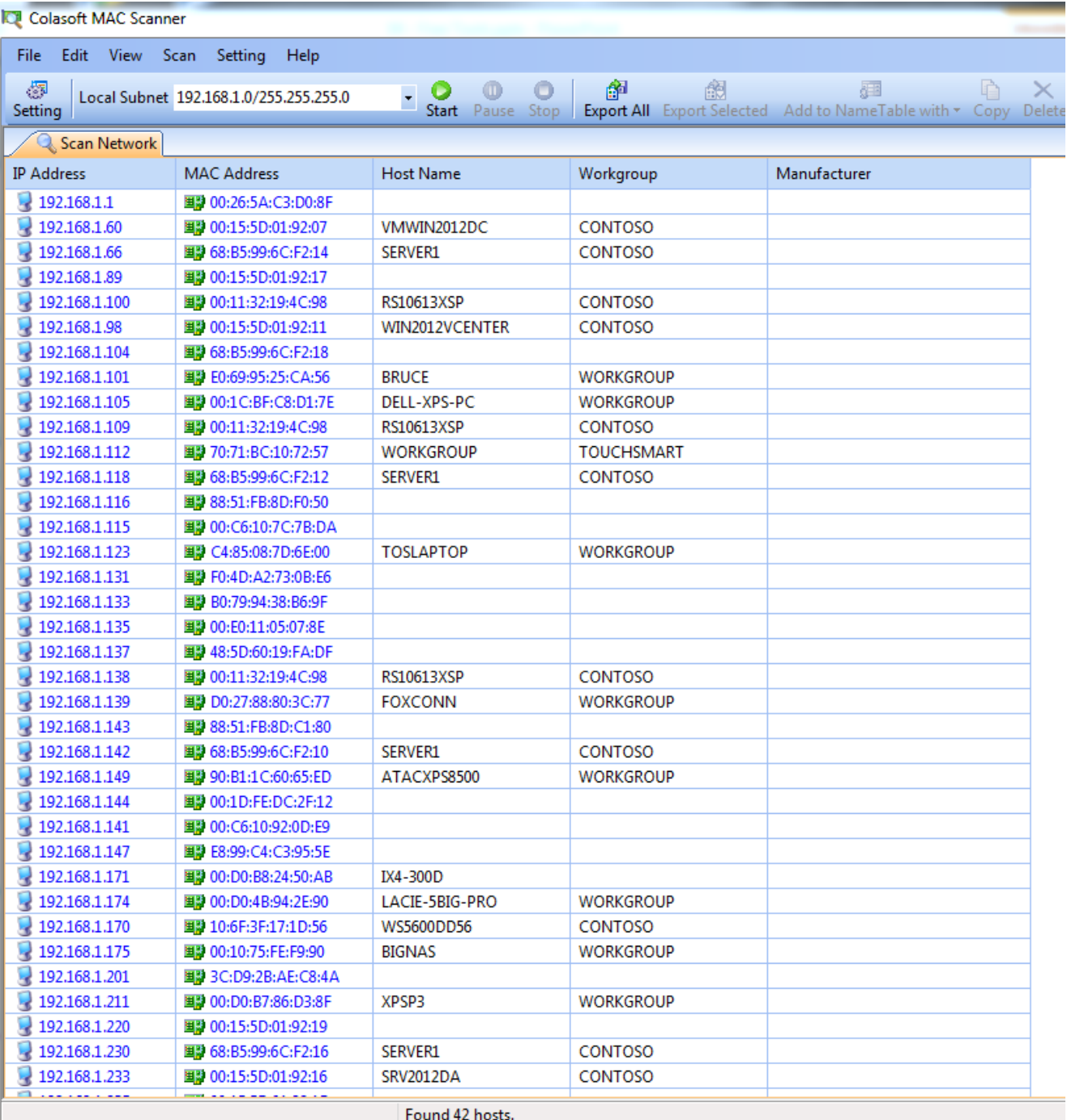
- PCAP Replay
  - PlayCap
    - <https://github.com/signal11/PlayCap>
  - TCP Replay
    - <https://github.com/appneta/tcpreplay>
- Scapy
  - <https://scapy.net/>
- Ngrep
  - <https://github.com/jpr5/ngrep>

# Colasoft

- Has a number of free tools
  - <https://www.colasoft.com/products/freeware.php>
- Colasoft Packet Player
  - [https://www.colasoft.com/packet\\_player/](https://www.colasoft.com/packet_player/)
- Capsa Free – network analyzer
- Colasoft MAC Scanner
- Colasoft Ping Tool
- Colasoft Packet Builder



# Colasoft MAC Scanner



The screenshot shows the Colasoft MAC Scanner application window. The title bar reads "Colasoft MAC Scanner". The menu bar includes "File", "Edit", "View", "Scan", "Setting", and "Help". The toolbar contains icons for "Setting", "Local Subnet" (set to 192.168.1.0/255.255.255.0), "Start", "Pause", "Stop", "Export All", "Export Selected", "Add to NameTable with", "Copy", and "Delete". Below the toolbar is a tab labeled "Scan Network". The main area displays a table of scanned hosts with columns for IP Address, MAC Address, Host Name, Workgroup, and Manufacturer. The table lists 42 hosts. At the bottom, a status bar indicates "Found 42 hosts."

IP Address	MAC Address	Host Name	Workgroup	Manufacturer
192.168.1.1	00:26:5A:C3:D0:8F			
192.168.1.60	00:15:5D:01:92:07	VMWIN2012DC	CONTOSO	
192.168.1.66	68:B5:99:6C:F2:14	SERVER1	CONTOSO	
192.168.1.89	00:15:5D:01:92:17			
192.168.1.100	00:11:32:19:4C:98	RS10613XSP	CONTOSO	
192.168.1.98	00:15:5D:01:92:11	WIN2012VCENTER	CONTOSO	
192.168.1.104	68:B5:99:6C:F2:18			
192.168.1.101	E0:69:95:25:CA:56	BRUCE	WORKGROUP	
192.168.1.105	00:1C:BF:C8:D1:7E	DELL-XPS-PC	WORKGROUP	
192.168.1.109	00:11:32:19:4C:98	RS10613XSP	CONTOSO	
192.168.1.112	70:71:BC:10:72:57	WORKGROUP	TOUCHSMART	
192.168.1.118	68:B5:99:6C:F2:12	SERVER1	CONTOSO	
192.168.1.116	88:51:FB:8D:F0:50			
192.168.1.115	00:C6:10:7C:7B:DA			
192.168.1.123	C4:85:08:7D:6E:00	TOSLAPTOP	WORKGROUP	
192.168.1.131	F0:4D:A2:73:0B:E6			
192.168.1.133	B0:79:94:38:B6:9F			
192.168.1.135	00:E0:11:05:07:8E			
192.168.1.137	48:5D:60:19:FA:DF			
192.168.1.138	00:11:32:19:4C:98	RS10613XSP	CONTOSO	
192.168.1.139	D0:27:88:80:3C:77	FOXCONN	WORKGROUP	
192.168.1.143	88:51:FB:8D:C1:80			
192.168.1.142	68:B5:99:6C:F2:10	SERVER1	CONTOSO	
192.168.1.149	90:B1:1C:60:65:ED	ATACXPS8500	WORKGROUP	
192.168.1.144	00:1D:FE:DC:2F:12			
192.168.1.141	00:C6:10:92:0D:E9			
192.168.1.147	E8:99:C4:C3:95:5E			
192.168.1.171	00:D0:B8:24:50:AB	IX4-300D		
192.168.1.174	00:D0:4B:94:2E:90	LACIE-5BIG-PRO	WORKGROUP	
192.168.1.170	10:6F:3F:17:1D:56	WS5600DD56	CONTOSO	
192.168.1.175	00:10:75:FE:F9:90	BIGNAS	WORKGROUP	
192.168.1.201	3C:D9:2B:AE:C8:4A			
192.168.1.211	00:D0:B7:86:D3:8F	XPSP3	WORKGROUP	
192.168.1.220	00:15:5D:01:92:19			
192.168.1.230	68:B5:99:6C:F2:16	SERVER1	CONTOSO	
192.168.1.233	00:15:5D:01:92:16	SRV2012DA	CONTOSO	

Found 42 hosts.

# 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		

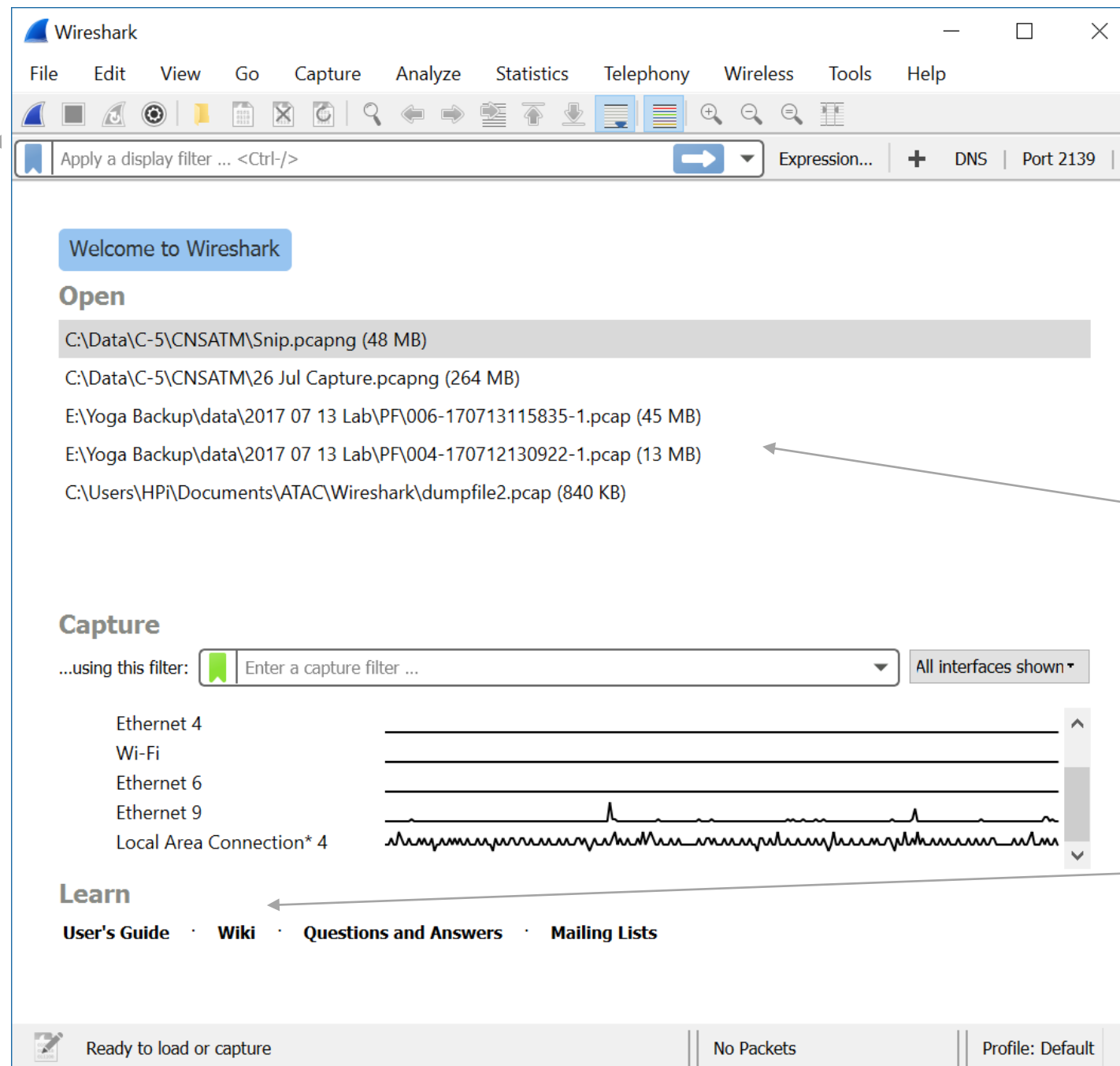
# Launch Screen

Go button

Interface list

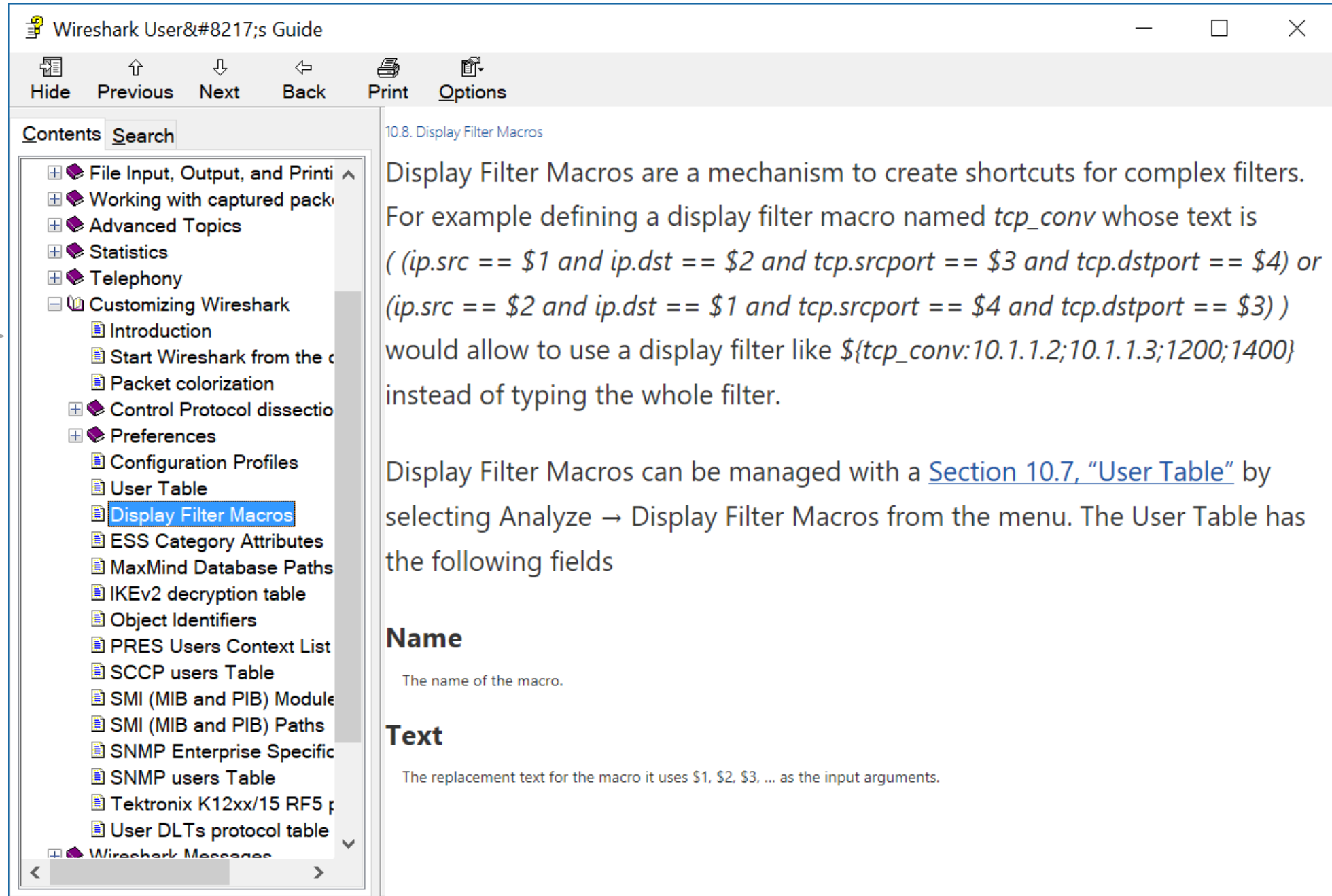
Recent files

Takes you to  
[wiki.wireshark.org/  
SampleCaptures](https://wiki.wireshark.org/SampleCaptures)



# Help

Stand alone help



Wireshark User's Guide

Hide Previous Next Back Print Options

Contents Search

- File Input, Output, and Printing
- Working with captured packets
- Advanced Topics
- Statistics
- Telephony
- Customizing Wireshark
  - Introduction
  - Start Wireshark from the command line
  - Packet colorization
- Control Protocol dissection
- Preferences
  - Configuration Profiles
  - User Table
  - Display Filter Macros**
  - ESS Category Attributes
  - MaxMind Database Paths
  - IKEv2 decryption table
  - Object Identifiers
  - PRES Users Context List
  - SCCP users Table
  - SMI (MIB and PIB) Modules
  - SMI (MIB and PIB) Paths
  - SNMP Enterprise Specific
  - SNMP users Table
  - Tektronix K12xx/15 RF5 p
  - User DLTs protocol table
- Wireshark Messages

## 10.8. Display Filter Macros

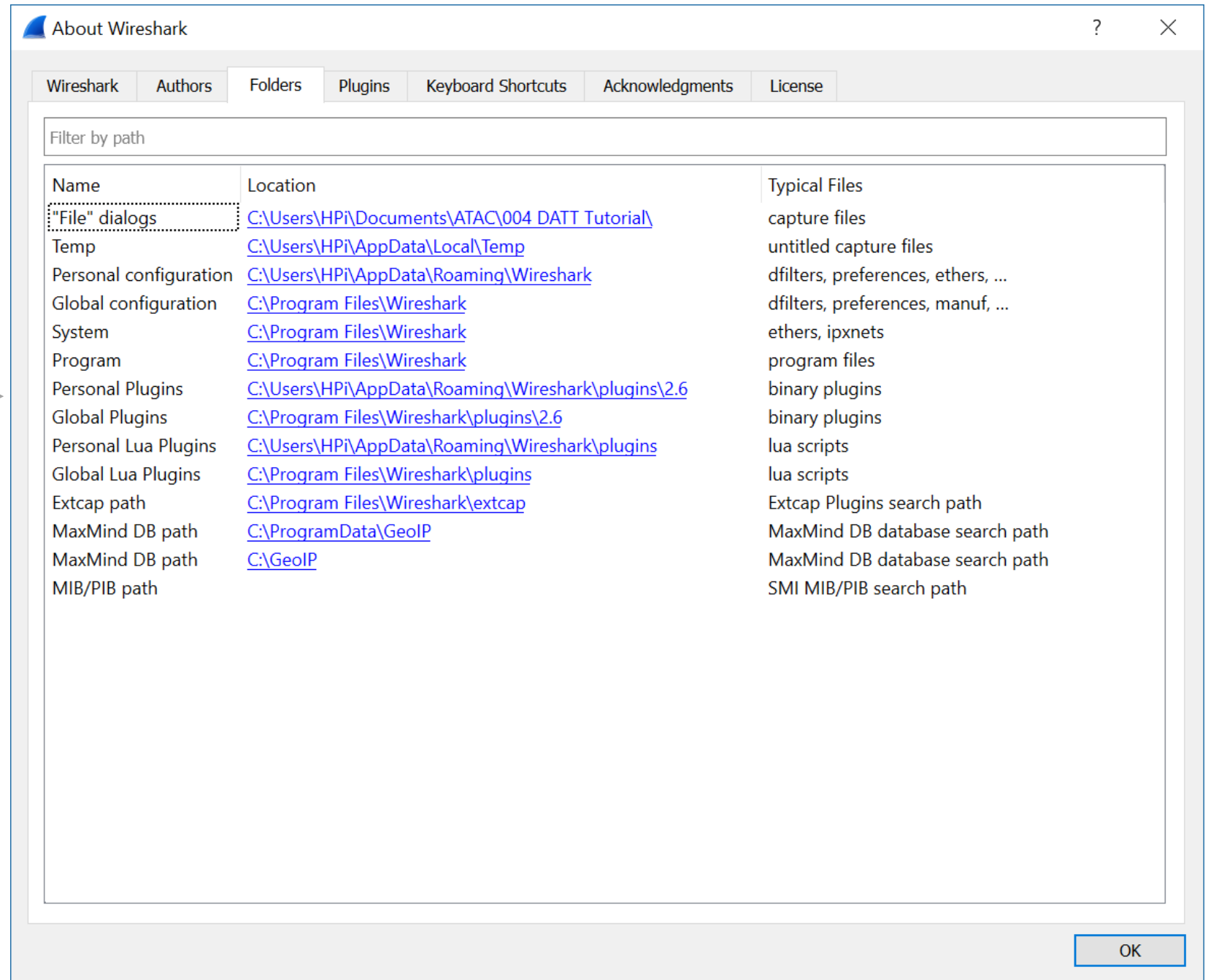
Display Filter Macros are a mechanism to create shortcuts for complex filters. For example defining a display filter macro named `tcp_conv` whose text is `(( ip.src == $1 and ip.dst == $2 and tcp.srcport == $3 and tcp.dstport == $4) or (ip.src == $2 and ip.dst == $1 and tcp.srcport == $4 and tcp.dstport == $3) )` would allow to use a display filter like `${tcp_conv:10.1.1.2;10.1.1.3;1200;1400}` instead of typing the whole filter.

Display Filter Macros can be managed with a [Section 10.7, "User Table"](#) by selecting Analyze → Display Filter Macros from the menu. The User Table has the following fields

Name
The name of the macro.
Text
The replacement text for the macro it uses \$1, \$2, \$3, ... as the input arguments.

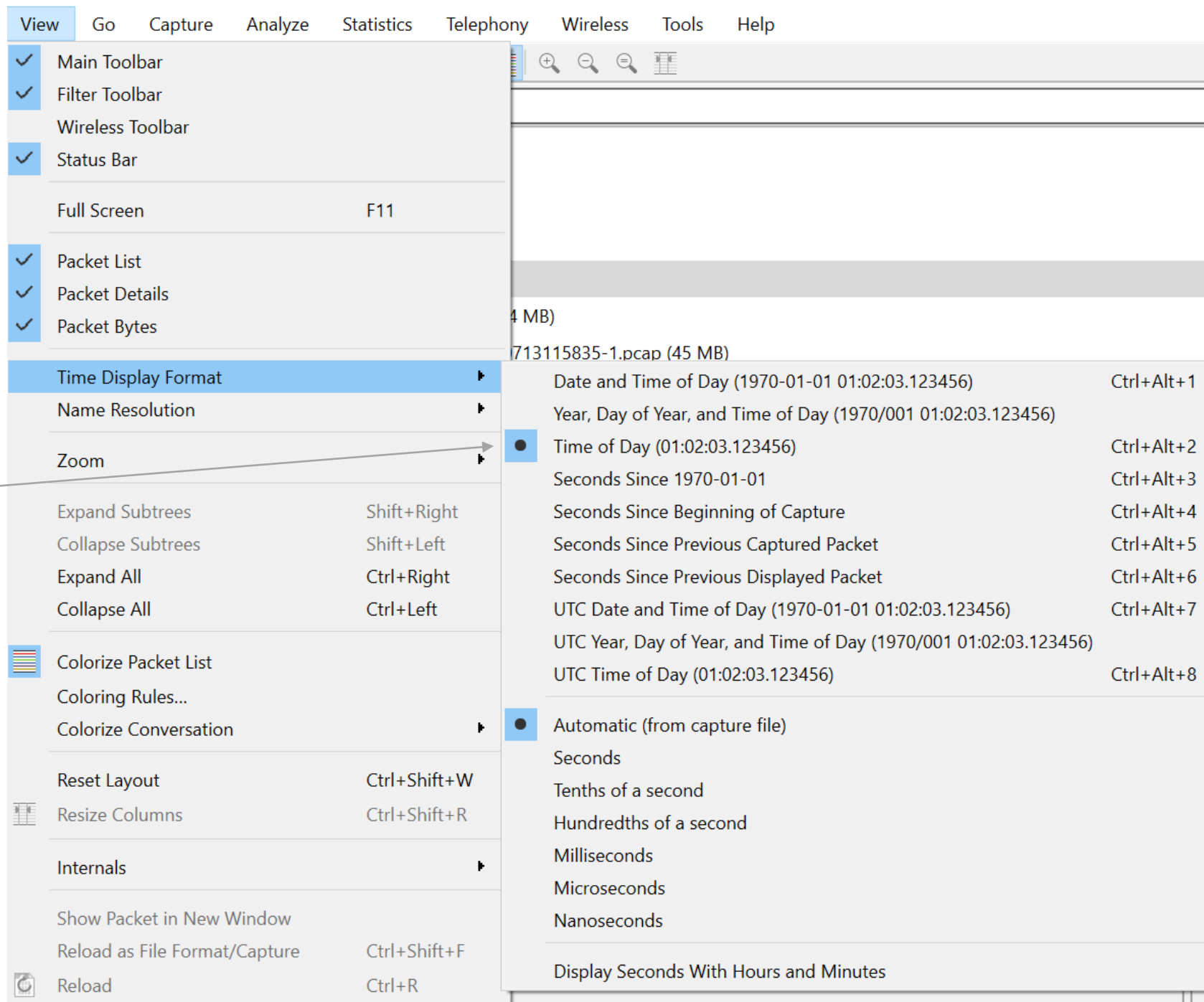
# Folders

From About  
menu



# Time Format

Default time  
setting



The image shows the Wireshark application's 'View' menu. The 'Time Display Format' option is selected, opening a submenu. In the submenu, 'Automatic (from capture file)' is selected. A callout box labeled 'Default time setting' points to the 'Automatic (from capture file)' option in the submenu.

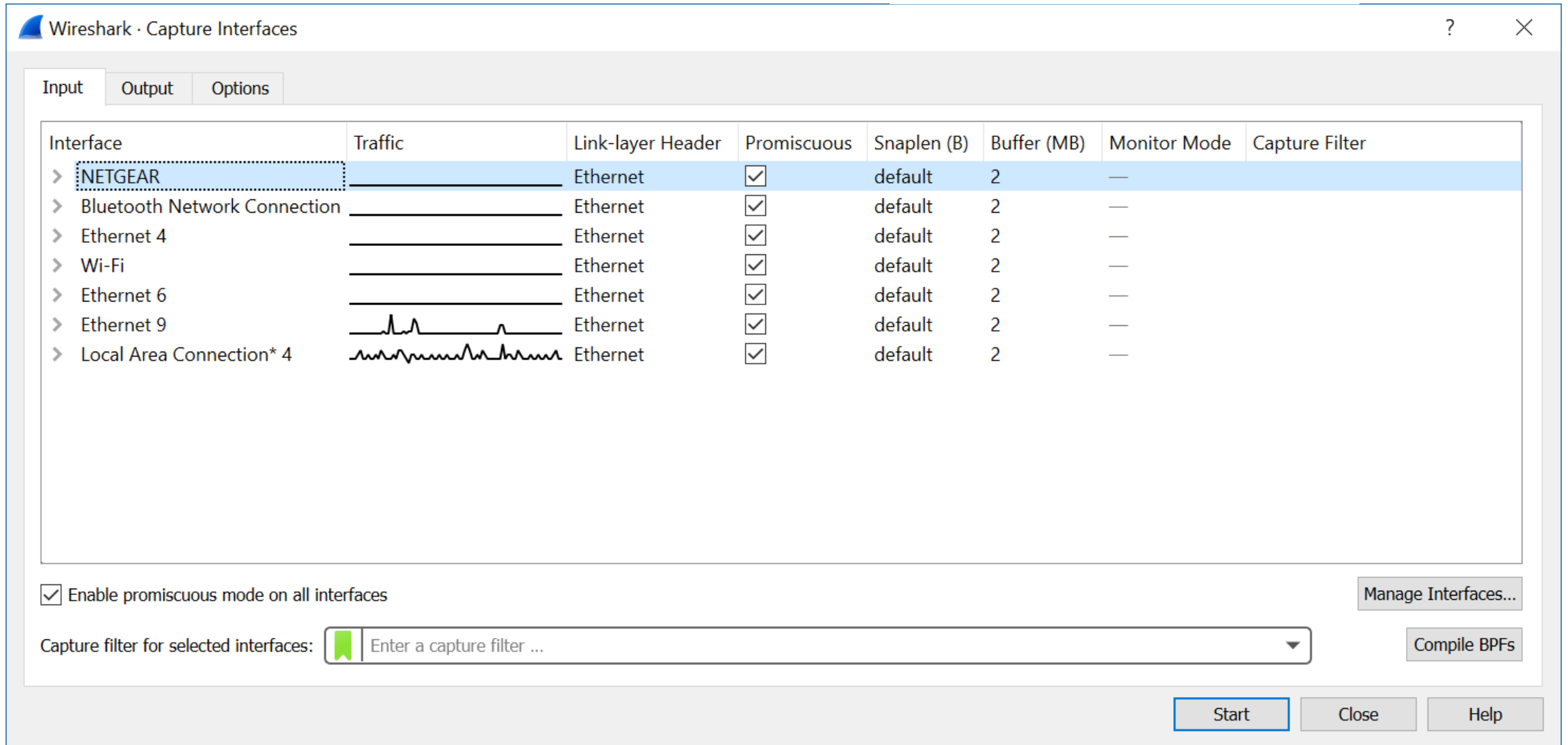
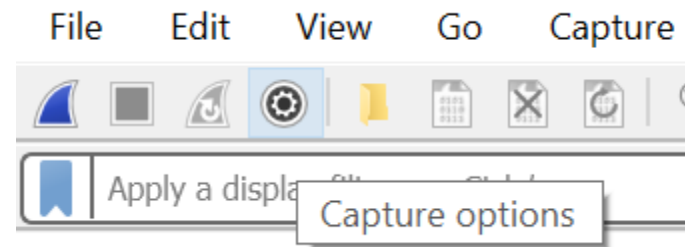
**View Menu Items:**

- ✓ Main Toolbar
- ✓ Filter Toolbar
- Wireless Toolbar
- ✓ Status Bar
- Full Screen F11
- ✓ Packet List
- ✓ Packet Details
- ✓ Packet Bytes
- Time Display Format**
- Name Resolution
- Zoom
- Expand Subtrees Shift+Right
- Collapse Subtrees Shift+Left
- Expand All Ctrl+Right
- Collapse All Ctrl+Left
- Colorize Packet List
- Coloring Rules...
- Colorize Conversation
- Reset Layout Ctrl+Shift+W
- Resize Columns Ctrl+Shift+R
- Internals
- Show Packet in New Window
- Reload as File Format/Capture Ctrl+Shift+F
- Reload Ctrl+R

**Time Display Format Submenu Items:**

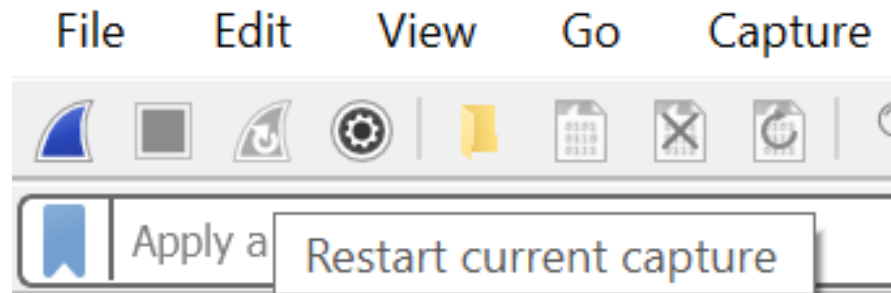
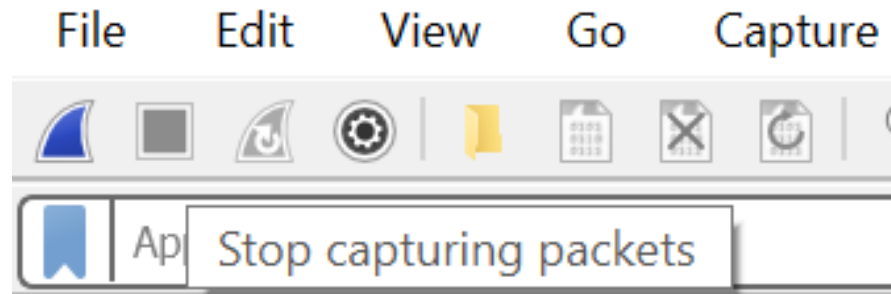
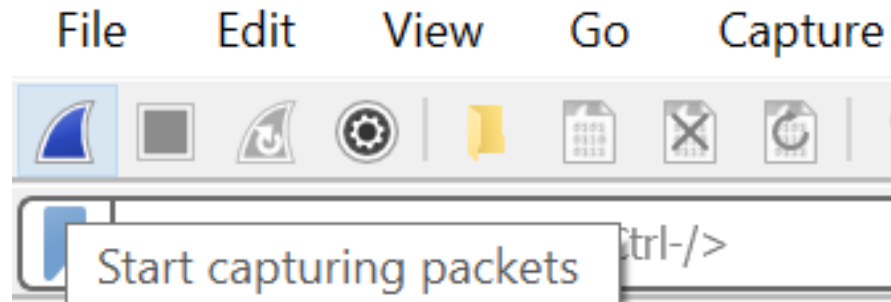
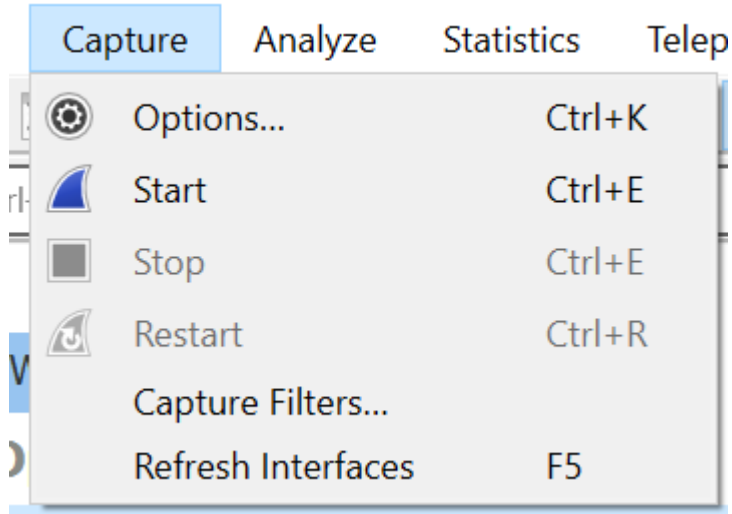
- Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+1
- Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
- Time of Day (01:02:03.123456) Ctrl+Alt+2**
- Seconds Since 1970-01-01 Ctrl+Alt+3
- Seconds Since Beginning of Capture Ctrl+Alt+4
- Seconds Since Previous Captured Packet Ctrl+Alt+5
- Seconds Since Previous Displayed Packet Ctrl+Alt+6
- UTC Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+7
- UTC Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
- UTC Time of Day (01:02:03.123456) Ctrl+Alt+8
- Automatic (from capture file)**
- Seconds
- Tenths of a second
- Hundredths of a second
- Milliseconds
- Microseconds
- Nanoseconds
- Display Seconds With Hours and Minutes

# Capture options





# Start / Stop Capture



# Filter Dialog

Wireshark · Display Filter Expression

Field Name

IPv4 · Internet Protocol Version 4

ip.addr · Source or Destination Address

ip.bogus\_ip\_length · Bogus IP length

ip.bogus\_ip\_version · Bogus IP version

ip.checksum · Header checksum

ip.checksum.status · Header checksum status

ip.checksum\_bad.expert · Bad checksum

ip.checksum\_calculated · Calculated Checksum

ip.cipso.categories · Categories

ip.cipso.doi · DOI

ip.cipso.malformed · Malformed CIPSO tag

ip.cipso.sensitivity\_level · Sensitivity Level

ip.cipso.tag\_data · Tag data

ip.cipso.tag\_type · Tag Type

ip.cur\_rt · Current Route

ip.cur\_rt\_host · Current Route Host

ip.dsfield · Differentiated Services Field

ip.dsfield.dscp · Differentiated Services Codepoint

ip.dsfield.ecn · Explicit Congestion Notification

ip.dst · Destination

ip.dst\_host · Destination Host

ip.empty\_rt · Empty Route

Relation

is present

==

!=

>

<

>=

<=

contains

matches

in

Value

Predefined Values

Range (offset:length)

Search:

No display filter

A hint.

OK

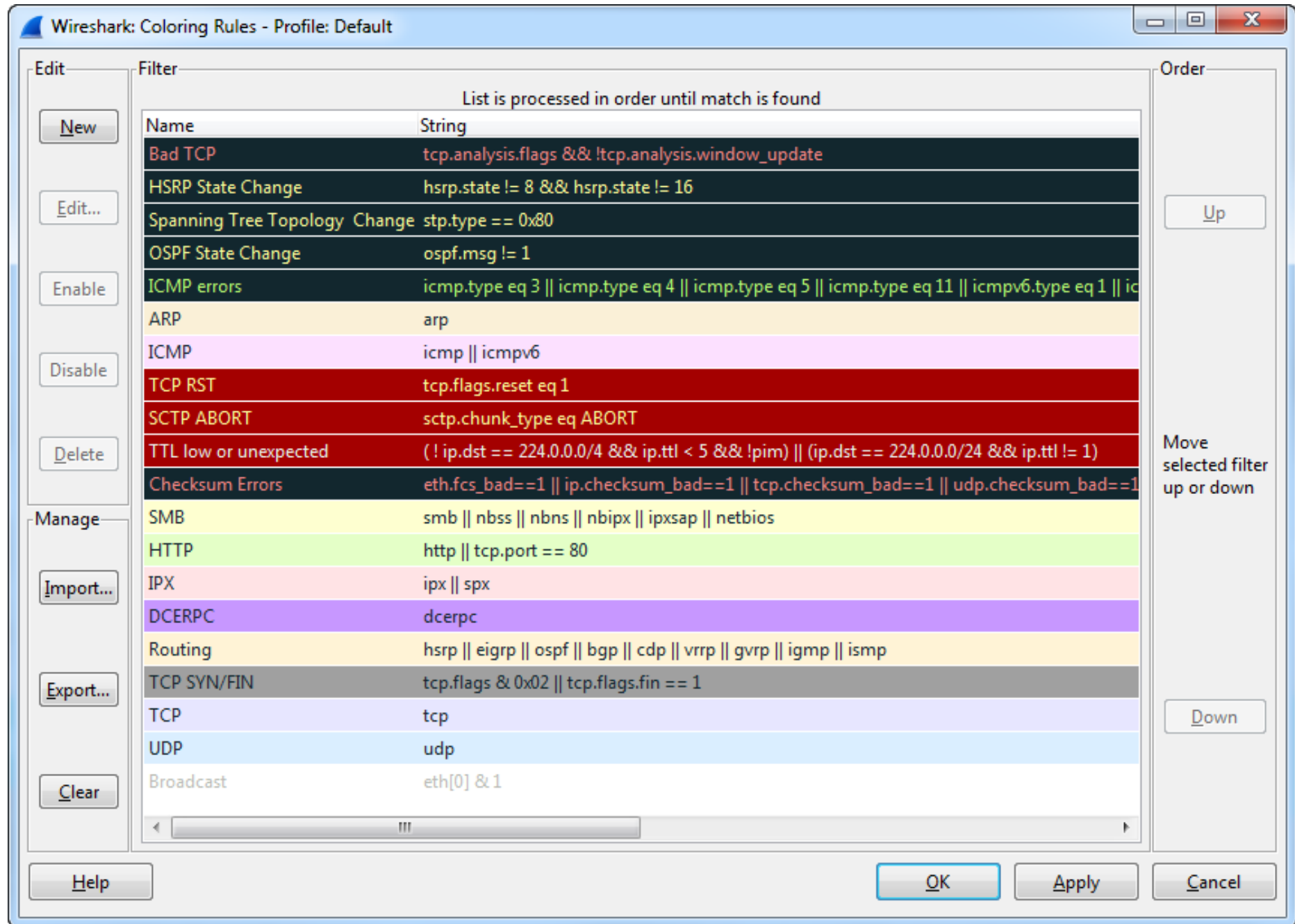
Cancel

Help

# 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

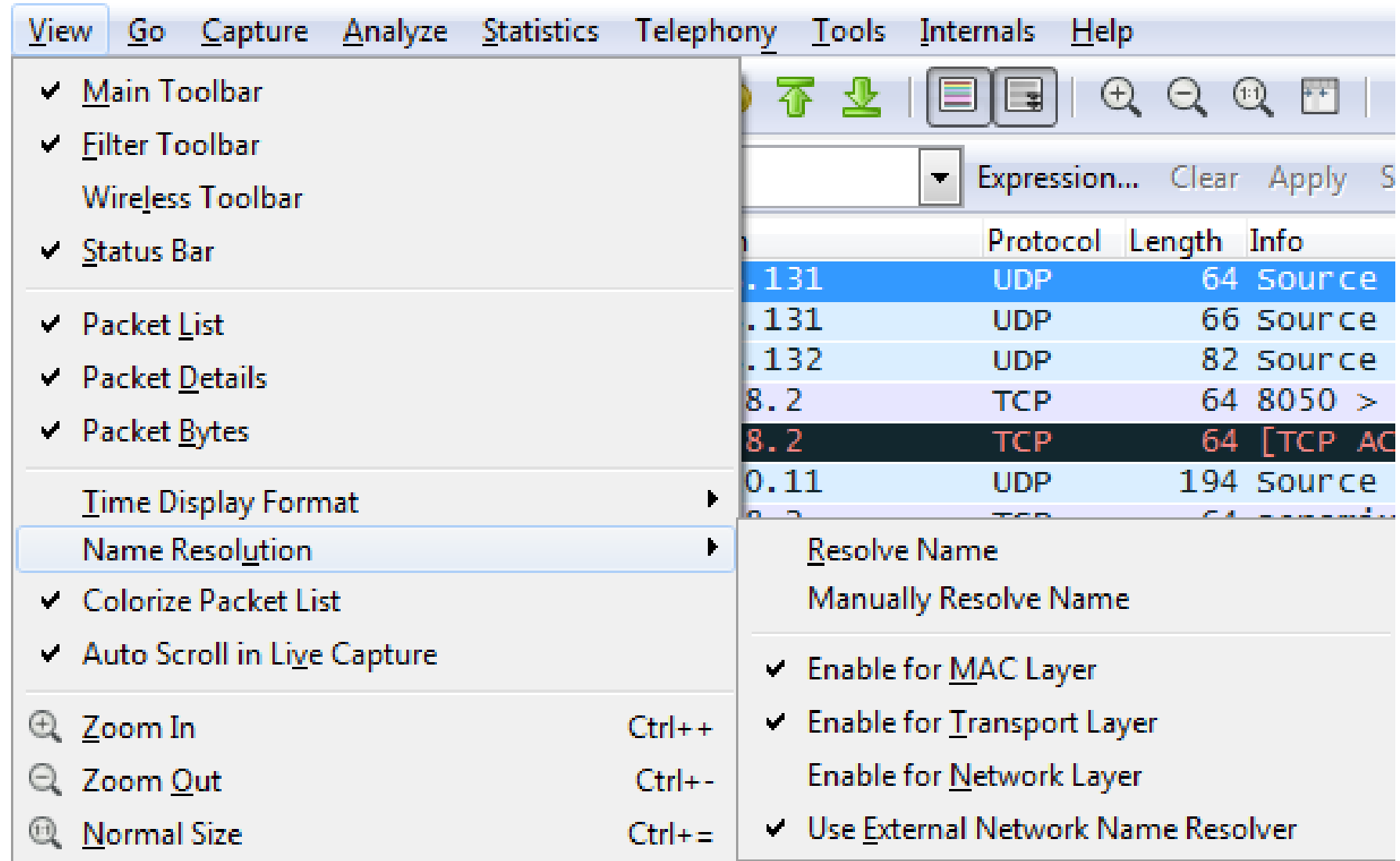
# Coloring Rules



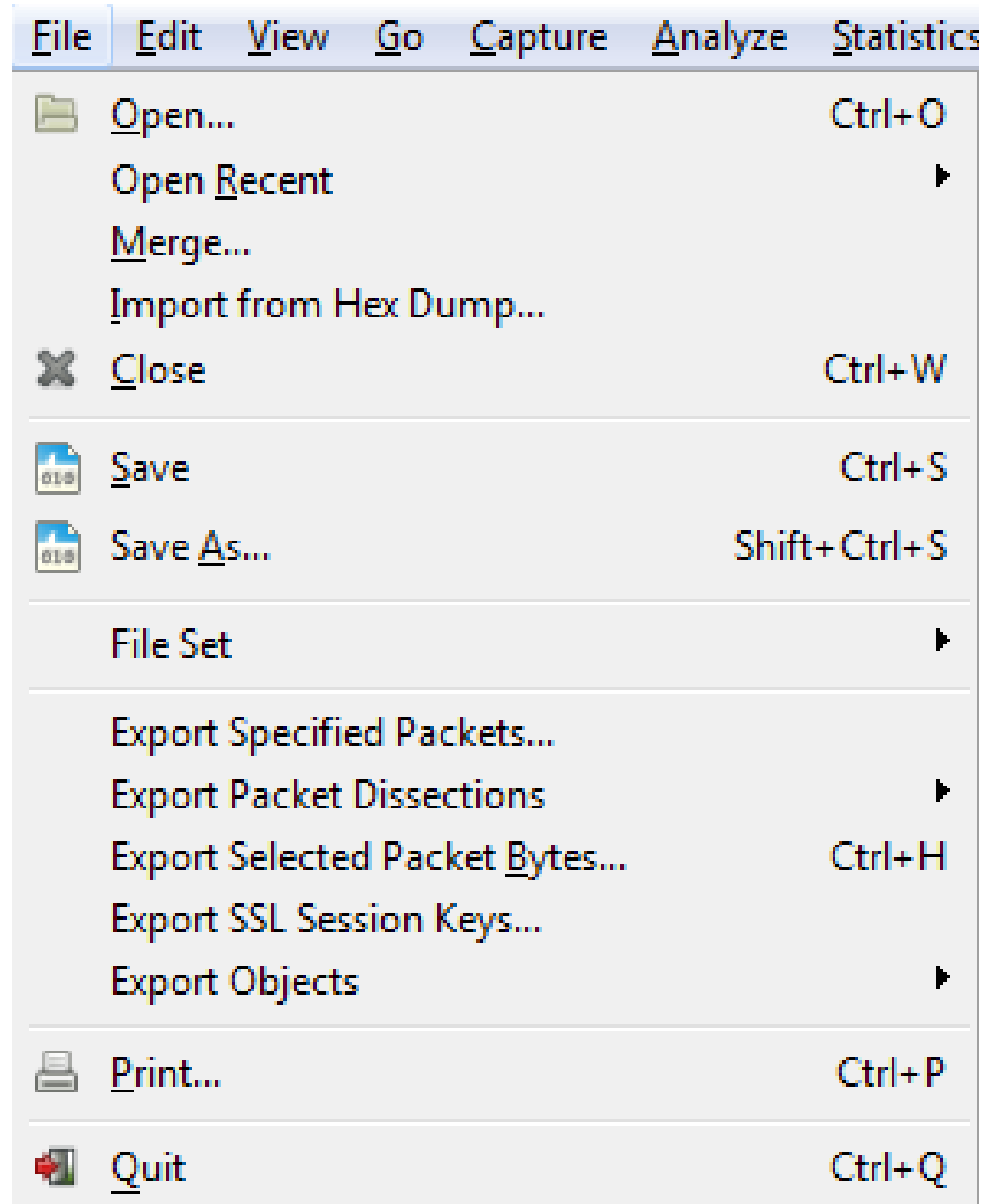
# Coloring example

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	ZyxeCom_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

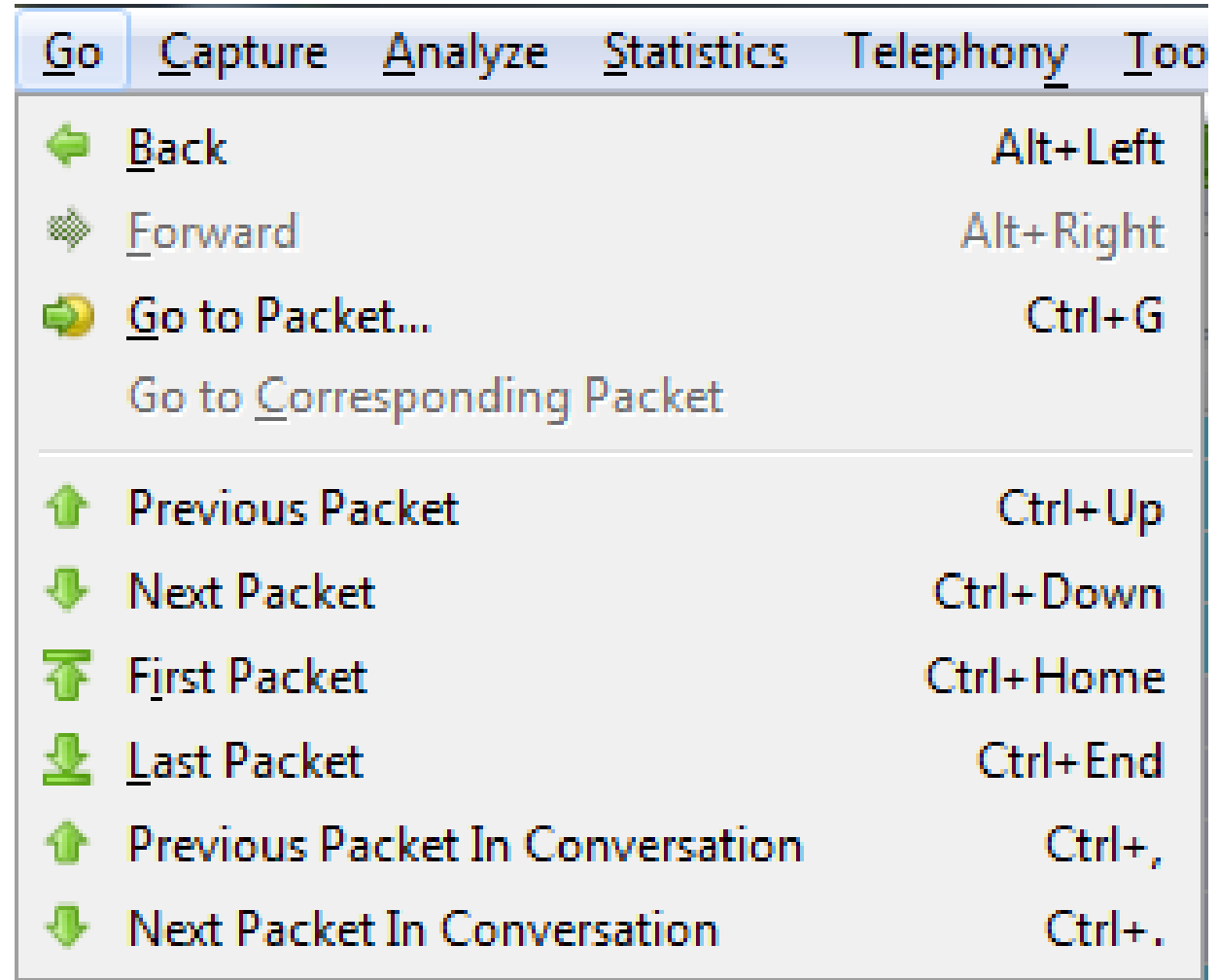
# Name Resolution



# File Open / Save Options



# Navigation





# Follow UDP Stream

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

Follow UDP Stream

Stream Content

```
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 ...$. .@ ...k.Rh`
00000030 eb 59 90 8c 83 52 70 60 ee 1c 10 4c 83 52 78 60 .Y...Rp` ...L.Rx`
00000040 ec 58 10 cc .X.
00000044 02 00 00 00 81 cb 5c 40 7f e7 a2 b3 81 cb 59 40 ..... \@ .....Y@
00000054 e0 00 02 a4 81 cb 6e 40 60 1f 82 53 83 52 80 60 .....n@ ...S.R.
00000064 ec 16 90 2c 83 52 88 60 6d 98 10 ac 83 57 dc 60 ...R. m...W.
00000074 00 00 00 fd 81 cb 69 40 80 00 02 1f .....i@
00000080 02 00 00 00 81 ca b3 40 92 fc 02 ef 81 ca b3 40 .....@ .....@
00000090 83 00 02 ef 81 ca a8 40 00 00 00 83 81 ca a7 40 .....@ .....@
000000A0 00 20 00 03 81 ca 9c 40 9b 41 85 86 81 ca a0 40 .....@ .A.....@
000000B0 e0 dc 00 49 81 ca b1 40 e0 00 00 af 83 59 08 60 ...I...@ .....Y.
000000C0 e0 00 00 44 81 ca b5 40 65 8c 02 04 81 ca c5 40 ...D...@ e.....@
000000D0 66 00 02 06 81 ca cd 40 65 9a 02 05 81 ca d5 40 f.....@ e.....@
000000E0 e5 90 02 07 81 ca b6 40 e4 e4 02 84 81 ca c6 40 .....@ .....@
000000F0 64 e2 02 86 83 59 05 60 e0 60 04 46 81 ca ce 40 d...Y. .F...@
00000100 64 de 02 85 81 ca d6 40 64 e6 02 87 83 59 06 60 d.....@ d...Y.
e0 60 04 45 83 59 07 60 e0 00 00 47 81 ca b7 40 .E.Y. .G...@
60 00 02 44 81 ca c7 40 e0 70 06 46 81 ca cf 40 .D...@ .p.F...@
60 60 06 45 81 ca d7 40 60 00 02 47 81 ca b8 40 .E...@ .G...@
64 2c c6 c4 81 ca c8 40 e4 25 c6 c6 81 ca d0 40 d.....@ .....@
e4 a1 c6 c5 81 ca d8 40 64 12 e6 c7 81 ca b9 40 .....@ d.....@
69 b2 02 24 81 ca c9 40 69 e0 02 26 81 ca d1 40 i...$. .@ i...&...@
69 c4 02 25 81 ca d9 40 69 be 02 27 i...%. .@ i...'
02 00 00 00 81 cb 5f 40 00 00 02 8b 81 cb 58 40 .....@ .....X@
60 00 02 24 81 ca ba 40 e3 14 02 a4 81 ca ca 40 ...$. .@ .....@
63 48 02 a6 81 ca d2 40 63 28 02 a5 83 57 dc 60 cH.....@ c(...W.
00 00 00 fd 81 ca da 40 e3 0a 02 a7 81 ca bb 40 .....@ .....@
e6 18 02 64 81 ca cb 40 e5 d0 02 66 81 ca d3 40 ...d...@ ...f...@
e6 08 02 65 81 ca db 40 e5 e8 02 67 81 ca bc 40 ...e...@ ...g...@
0000010C 60 bd c2 e4 81 cb 59 40 e0 00 02 a4 81 cb 6e 40 .....Y@ .....n@
0000011C 60 1f 82 53 81 ca cc 40 e0 be 02 e6 81 ca d4 40 ...S...@ .....@
0000012C 60 b7 82 e5 81 ca dc 40 60 99 02 e7 83 58 70 60 ...=.
0000013C 60 c0 00 3d ...=
```

Entire conversation (22418780 bytes)

Find Save As Print ☐ ASCII ☐ EBCDIC ☒ Hex Dump ☐ C Arrays ☐ Raw

Help Filter Out This Stream Close

3% of Filtering: (ip.addr eq 10.146.10.2 and ip.addr eq 239.1.15.62) and (udp.port eq 49860 and udp.port eq 9006)

Filtering: (ip.addr eq 10.146.10.2 and ip.addr eq 239.1.15.62) and (udp.port eq 49860 and udp.port eq 9006)

Status: 308576 of 7714479 frames

Elapsed Time: 00:09

Time Left: --:--

Progress: 3%

Stop

# 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.`

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...19.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

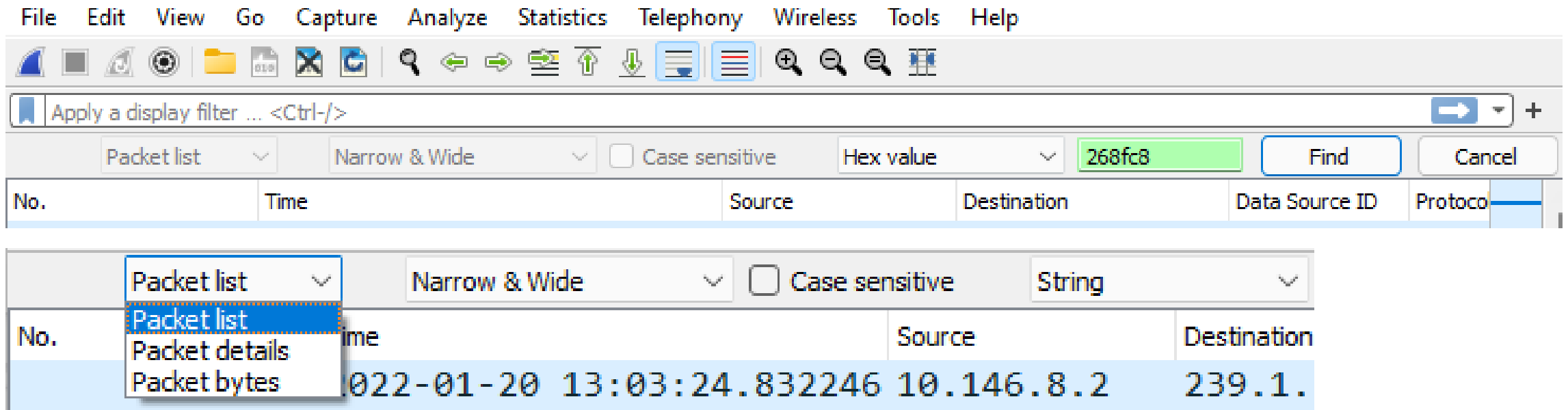
Profile: Classic

11

Fragments  
reassembled  
in Pkt #15

# Find Hex Values

- Launch with Control-F
- Brings up dialog to search Display filter, Hex value, String, Reg expr



data.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

Packet list Narrow & Wide Case sensitive Hex value 268fc8 Find Cancel

No.	Time	Source	Destination	Data Source ID	Protocol	Length	Info
26	2022-01-20 13:03:24.832246	10.146.8.2	239.1.11.24		UDP	110	31045 → 9100 Len=68
27	2022-01-20 13:03:24.832721	10.146.8.2	239.1.8.132		UDP	150	30995 → 9270 Len=108
28	2022-01-20 13:03:24.832941	10.146.8.2	239.1.8.131		UDP	60	31051 → 9281 Len=12
29	2022-01-20 13:03:24.832942	10.146.8.2	239.1.8.132		UDP	60	31052 → 9281 Len=12
30	2022-01-20 13:03:24.833900	10.145.8.2	239.1.8.131		UDP	60	16456 → 9270 Len=12
31	2022-01-20 13:03:24.833901	10.145.8.2	239.1.8.131		UDP	62	16457 → 9271 Len=20
32	2022-01-20 13:03:24.834194	10.145.8.2	239.1.8.31		UDP	86	16481 → 9120 Len=44
33	2022-01-20 13:03:24.835061	10.145.10.2	239.1.15.11		UDP	342	28735 → 9001 Len=300
34	2022-01-20 13:03:24.835610	10.146.11.1	239.1.0.10		UDP	378	31382 → 9027 Len=336
35	2022-01-20 13:03:24.836164	10.145.11.2	239.1.12.31		UDP	270	55016 → 9065 Len=228
36	2022-01-20 13:03:24.838472	10.145.8.3	239.1.10.91		UDP	174	16525 → 9248 Len=132

> Frame 36: 174 bytes on wire (1392 bits), 174 bytes captured (1392 bits)

> Ethernet II, Src: 02:00:00:01:01:08 (02:00:00:01:01:08), Dst: IPv4mcast\_01:0a:5b (01:00:5e:01:0a:5b)

> Internet Protocol Version 4, Src: 10.145.8.3, Dst: 239.1.10.91

> User Datagram Protocol, Src Port: 16525, Dst Port: 9248

▼ Data (132 bytes)

Data: 02000000268fc02000000019268f002082008401268f402000404411268f8020120a3809...

[Length: 132]

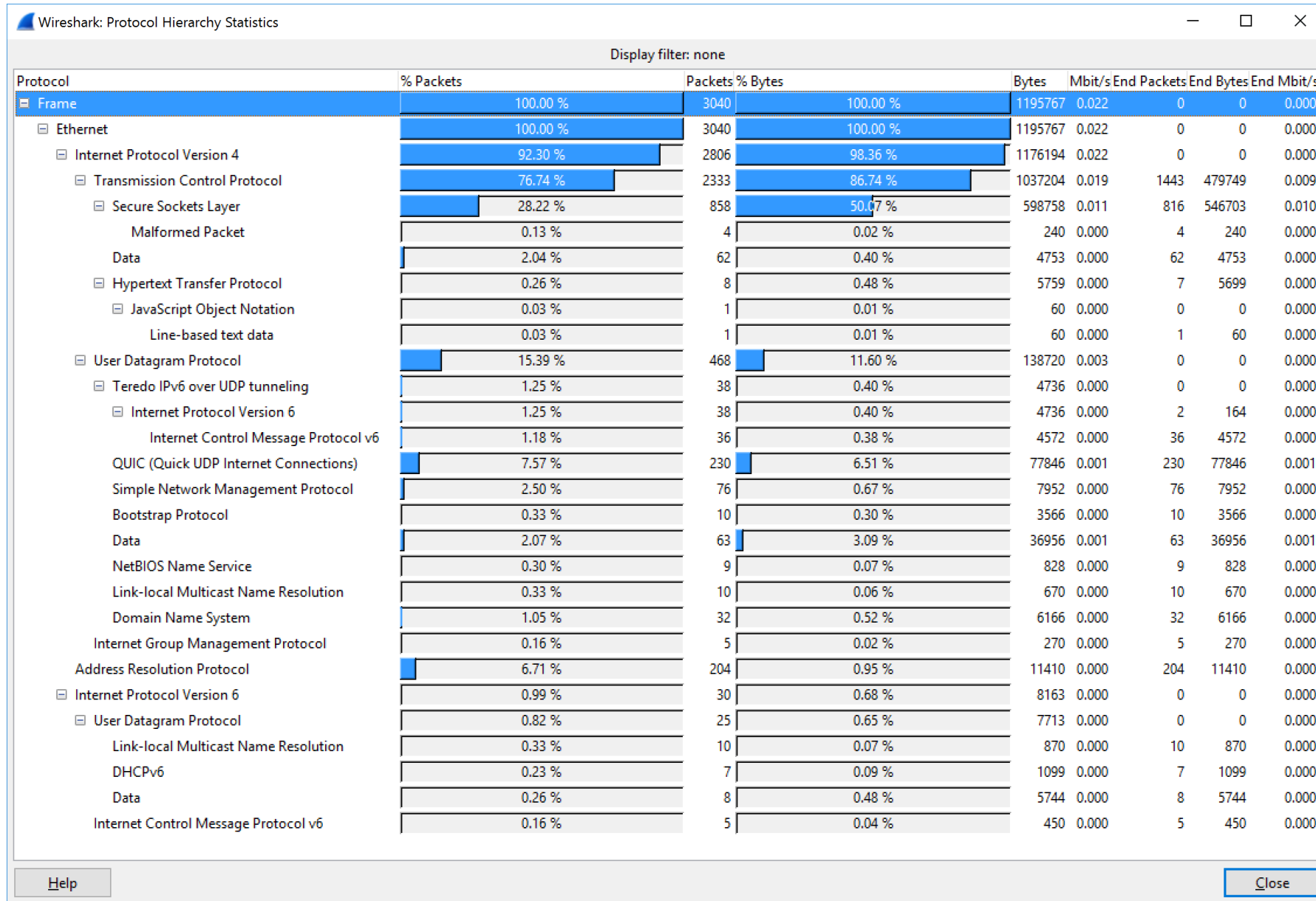
```
0000 01 00 5e 01 0a 5b 02 00 00 01 01 08 08 00 45 00  --^--[. . . . .E-
0010 00 a0 c0 82 00 00 40 11 ad da 0a 91 08 03 ef 01  -...-@- . . . . .
0020 0a 5b 40 8d 24 20 00 8c 00 00 02 00 00 00 26 8f  -[@-$ - . . . . .&-
0030 c0 20 00 00 00 19 26 8f 00 20 82 00 84 01 26 8f  -...-&- . . . . .&-
0040 40 20 00 40 44 11 26 8f 80 20 12 0a 38 09 26 8f  @-@D-&- . -8-&-
0050 c8 20 80 00 00 99 26 8f 18 20 00 00 00 c1 26 91  -...-&- . . . . .&-
0060 40 20 e0 00 80 13 26 91 48 20 60 00 80 93 26 91  @-...-&- H `...-&-
0070 00 20 60 05 e0 03 26 91 08 20 e0 05 e0 83 26 91  -`...-&- . . . . .&-
0080 b8 20 60 00 20 eb 26 8d 20 20 60 00 04 22 26 8d  -`. -&- `..."&-
0090 18 20 63 c0 00 c2 26 8d 30 20 61 90 00 62 26 8d  -c- -&- 0 a-b&-
00a0 10 20 62 13 40 42 26 8d 07 20 60 05 e0 26      -b-@B&- . `...&-
```

Data (data.data), 132 bytes

Packets: 3972360 · Displayed: 3972360 (100.0%) Profile: Default

# Bad Behavior

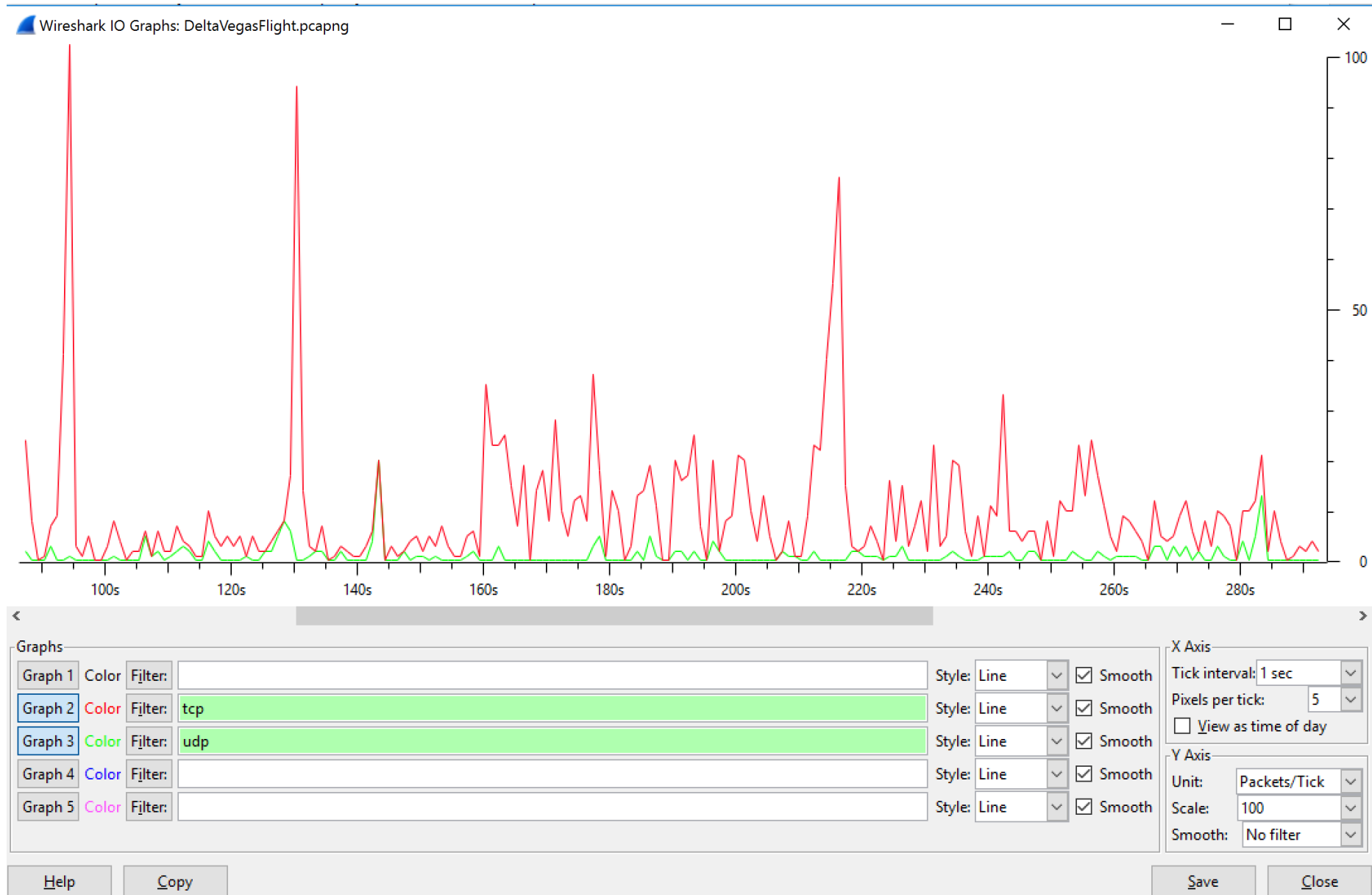
- Statistics -> Show Address Resolution
- Statistics -> Protocol Hierarchy
- Statistics -> Conversations
- Statistics -> Endpoints
- Statistics -> Flow Graph
- IP Statistics -> Source and Dest IP Addresses



# 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





# Expert Infos

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

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

- Home page - <https://www.wireshark.org/>
- Wiki - <https://gitlab.com/wireshark/wireshark/-/wikis/home>
- Sharkfest user conference videos -  
<https://www.youtube.com/c/SharkFestWiresharkDeveloperandUserConference>

Questions?