# ImmediaTV Corporation

Innovative video networking

## Innovations today for Broadcasters of tomorrow

# IP Protocol Review

Ciro A. Noronha, Ph.D.

Innovations today for Broadcasters of tomorrow

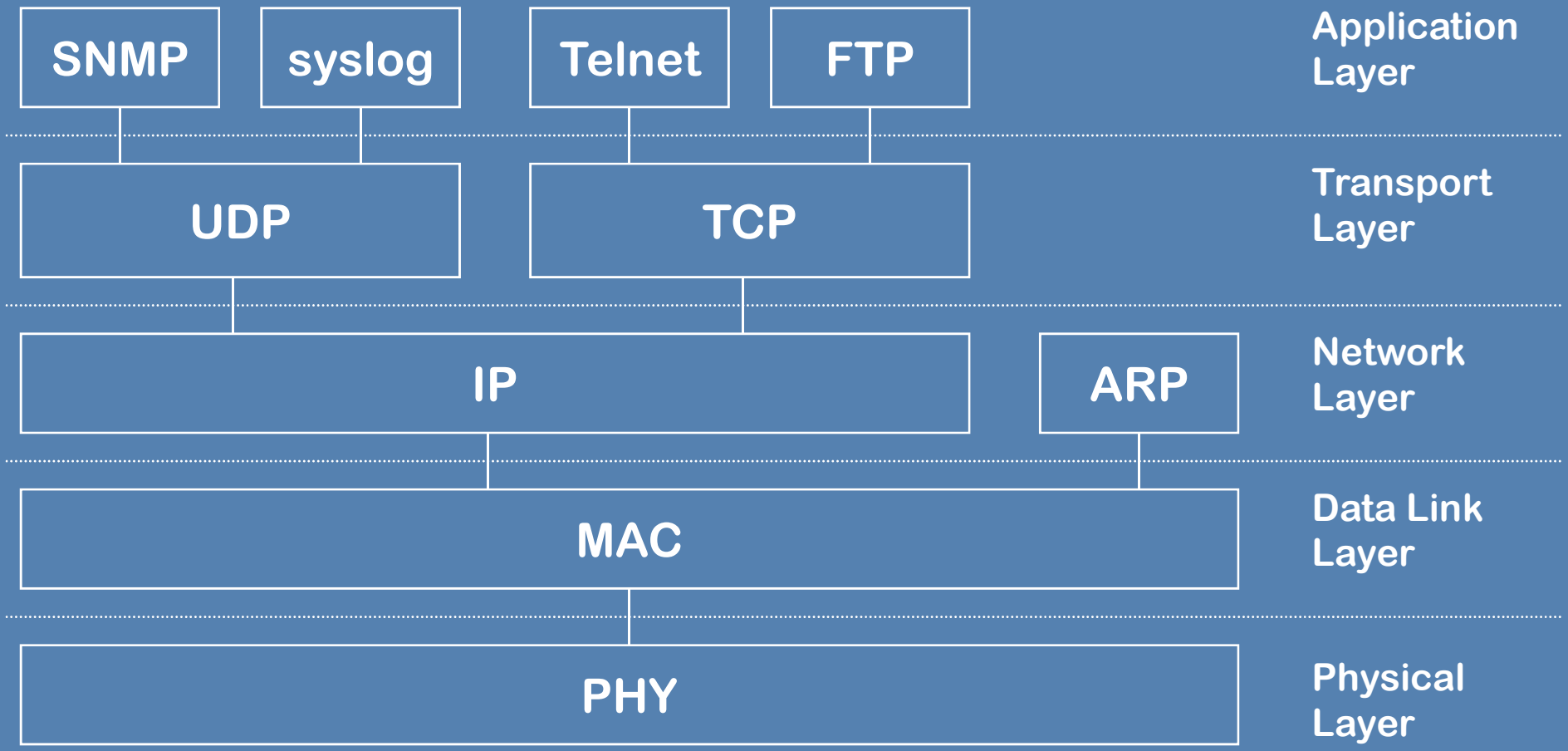
# Agenda

- Networking Layers
- Physical Layer
  - Ethernet Physical Layer
- Medium Access Control Layer
  - Ethernet MAC
  - Transparent Bridging
  - Virtual LANs
- Network Layer
  - IP Protocol
  - Multicast
  - Routing Basics

# Agenda (cont.)

- The Transport Layer
  - UDP/TCP
- The Application Layer
  - Selected Applications

# IP Protocol Roadmap



# Acronyms

- MAC: Media Access Control
- IP: Internet Protocol
- ARP: Address Resolution Protocol
- UDP: User Datagram Protocol
- TCP: Transmission Control Protocol
- SNMP: Simple Network Management Protocol
- RIP: Routing Information Protocol
- FTP: File Transfer Protocol
- RTP: Real Time transport Protocol

# Reference Material: IEEE Standards

- The IEEE standards for layers 1 and 2 are now free to download from their site.
- Download link:

  http://standards.ieee.org/getieee802/portfolio.html
- Standards of Interest:
  - IEEE 802.1: Bridging Standards
  - IEEE 802.3: CSMA/CD (Ethernet) Standards

# Reference Material: IP Stack

- Internet standards are published as RFCs (Request For Comments)
- RFCs go through a long review period before they become standards.
- The official site for RFCs is:

  http://www.ietf.org/rfc.html

- The subset of RFCs which have become approved standards can be found here:

  http://www.rfc-editor.org/rfcxx00.html

# Layer 1
# The Physical Layer

# What is in the Physical Layer?

- The Physical Layer defines the following:
  - Connectors and Cabling
  - Signal Levels
  - Line signaling and formatting
  - Timing, data rates
- Most of today's local area networks use **Ethernet** over **unshielded twisted pair** as the physical layer.
- It is also relatively common to find Ethernet over fiber, especially for Gigabit interfaces.

# A Brief History of Ethernet

- Ethernet was invented in the early 70s by Robert Metcalfe at Xerox PARC
  - Objective was to network computers to share a fast laser printer
  - Landmark paper describing architecture published in 76.
- Basic Idea:
  - Connect a whole bunch of computers in parallel to a coaxial cable
  - When a computer sends a packet, all others get it but only the destination accepts it
  - A computer checks the cable to see if a transmission is in progress before sending
  - If two computers send at the same time, there is a **collision** and they have to retry later

# Evolution of Ethernet

- From coaxial cable, Ethernet evolved into unshielded twisted pair
  - One pair to transmit, another independent pair to receive
  - Full-duplex operation is now possible!
- Speeds also evolved: 10 Mb/s, 100 Mb/s, 1 Gb/s, and now 10 Gb/s.
- Types of twisted pair cables:
  - Category 3: 16 MHz, good for 10 Mb/s
  - Category 4: 20 MHz, good for 10 Mb/s
  - Category 5: 100 MHz, good for 100 Mb/s
  - Category 5e: 100 MHz, good for 1 Gb/s
  - Category 6: 250 MHz, good for 1 Gb/s
  - Category 6a: 500 MHz, good for 10 Gb/s

# Facts to Remember

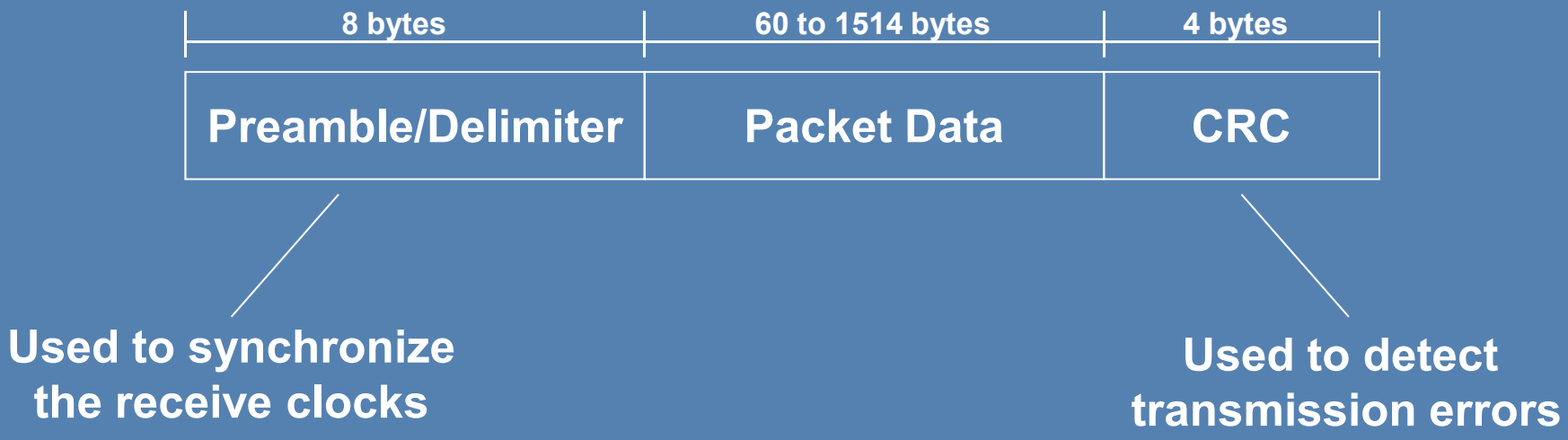
- If you use the proper cable for the speed range, UTP Ethernet can run up to 100m (300 ft)
- Ethernet UTP has 4 pairs (8 conductors)
  - 10 Mb/s and 100 Mb/s use 2 pairs (one pair to send, one pair to receive); the other 2 pairs are unused
  - 1 Gb/s uses all 4 pairs; data is divided into four 250Mb/s streams and each pair is used in both directions simultaneously
- Modern Ethernet devices can **auto-negotiate**; they will automatically set up:
  - Full duplex/half duplex
  - Speed (use the maximum compatible speed)
- Good tutorial:

  ftp://ftp.iol.unh.edu/pub/gec/training/ethernet_evolution.pdf

# Layer 2
# The Data Link Layer

# Introduction

- The Data Link Layer provides:
  - Data transfer services between nodes in the same network
  - Error Detection and correction for the physical layer
  - Medium access control as required.
- Ethernet Packet:

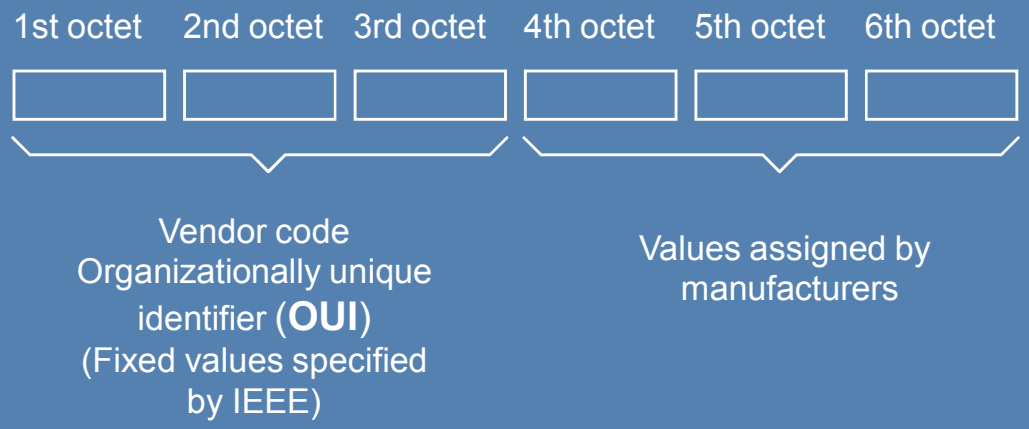


# LAN Addressing

- Ethernet devices have addresses
  - This is how a device knows "this packet is for me"
- IEEE 802 standardized length of address field
  - 16-bit addresses
  - 48-bit addresses
- 16-bit address option has not caught on and is not found in practice
- 48-bit (6-byte) addresses allow stations to be provided with globally unique identifiers - (Plug and Play).

# Globally Unique LAN Addresses

- IEEE is the official authority responsible for handing out blocks of addresses.
  - $1,750 for a block of 24 million addresses

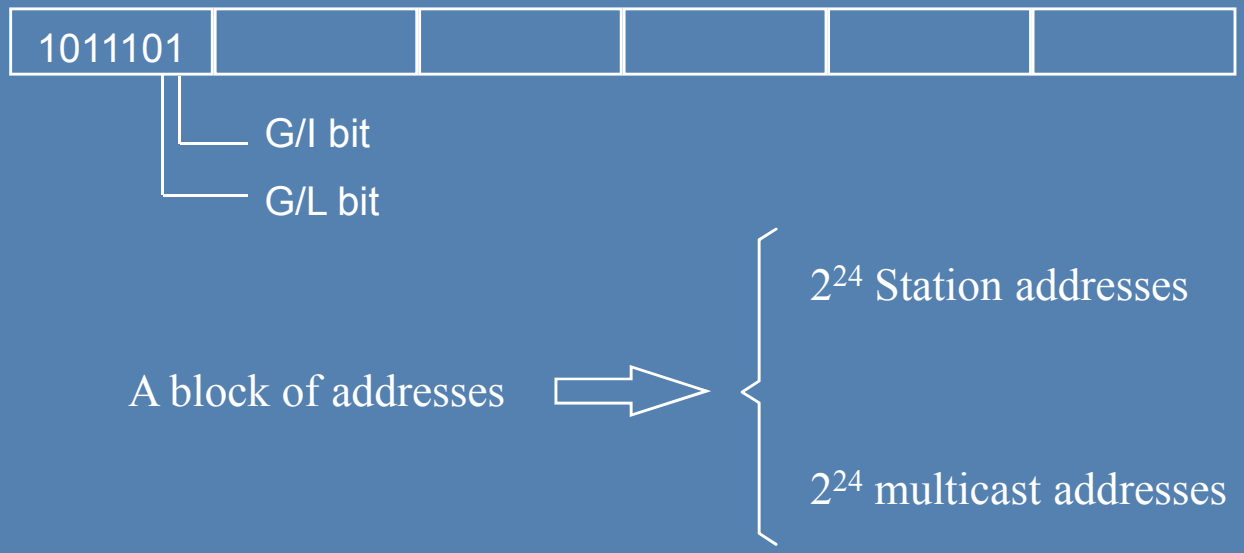



OUI Examples:

- **`00-23-02`**: Cobalt
- **`AC-83-F0`**: Magenta
- **`00-02-B3`**: Intel

# Group/Individual (G/I) Bit

- G/I bit is defined as the *first bit on the wire*
  - If G/I bit is 0, address refers to a particular station
  - If G/I bit is 1, address refers to a logical group of stations (or multicast address)
  - Special case: all 1's address means broadcast.



# Global/Local (G/L) Bit

- Blocks of addresses purchased from IEEE have G/L bit set to zero.
- Addresses with G/L bit set to 1 are freely available. It is up to network manager to assign these addresses & make sure that there are no address collisions.

# MAC Layer Packet (Ethernet)

| DST Address | SRC Address | Type/ Len | Layer-3 Datagram |
|---|---|---|---|
| 6 | 6 | 2 | 46 to 1500 |
| | | | |

Address: 3 bytes OUI, 3 bytes assigned by the manufacturer

- If LSB of the first byte is 1: Multicast packet
- If second LSB of the first byte is 1: Locally-administered address

Type/Length: Identifies the layer-3 protocol

- If Type/Length is less than 1536 (0x600), then the packet is an IEEE 802.3 packet and the field has the length of the packet.
- If Type/Length is 0x600 or more, the field is a protocol type (Ethertype)
- Common Ethertypes:
  - 0x800: IP
  - 0x806: ARP

# More Info

- IEEE 802-1990 tutorial:

  http://standards.ieee.org/regauth/oui/tutorials/lanman.html

- OUI Listing and information

  http://standards.ieee.org/regauth/oui/index.html

# Transparent Bridges (Switches)


- Transparent: hosts in segment 1 can talk to hosts in segment 2 as if they were in the same network and vice-versa.
- Hosts are not aware of the bridge.
- Traffic between hosts in segment 1 must not be transmitted to segment 2 and vice-versa.
- Switches are just bridges with lots of ports.

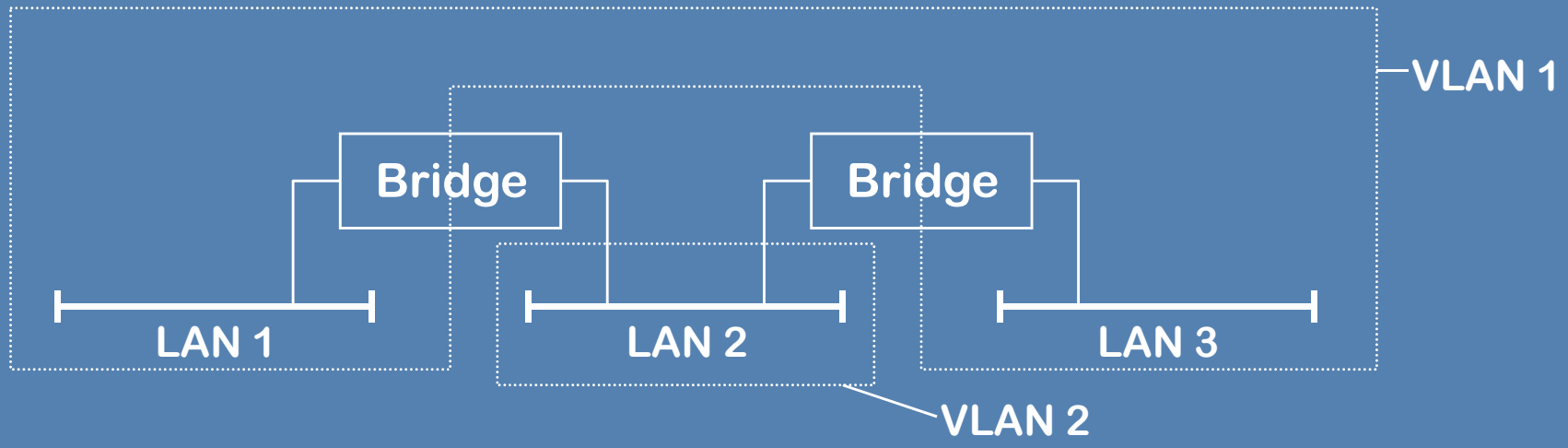
# Transparent Bridge Operation

- Transparent bridges work by monitoring the source MAC address
- Any time they receive a packet, they “learn” the location of that source
- Packets whose locations are unknown are “flooded”
  - “Flooding”: sending a packet on all ports except the one they came from
- Basic bridges will flood broadcast and multicast packets

# Virtual LANs

- A virtual LAN (VLAN) is a collection of LAN segments and the stations/devices connected to them within a bridged LAN that has exactly the same properties of an independent LAN.
- In a bridged LAN comprising several VLANs, traffic belonging to a VLAN is restricted from reaching users in other VLANs

# VLAN Example

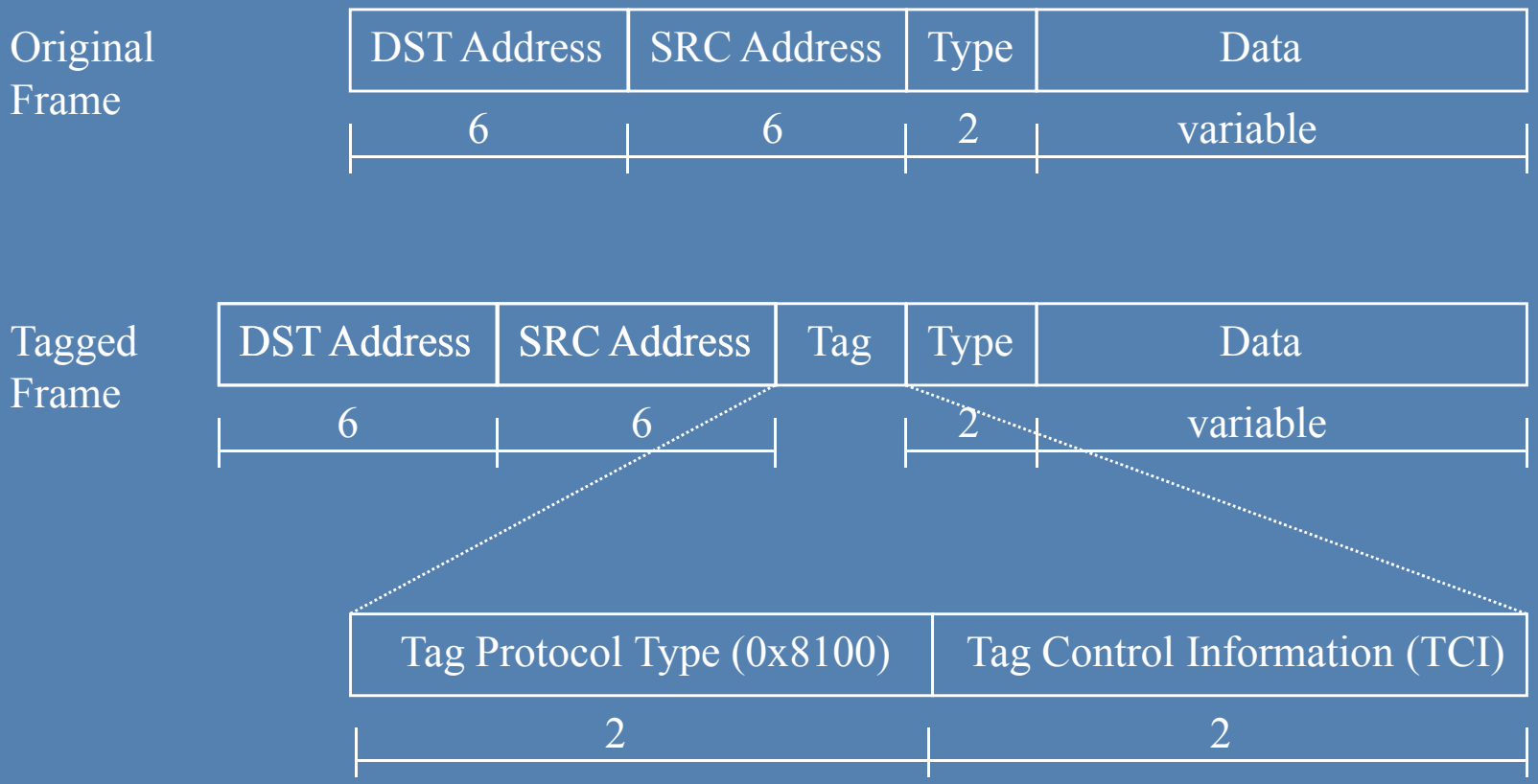



- Stations in LAN1 and LAN3 can freely communicate and "believe" to be in the same LAN
- Stations in LAN2 cannot talk to stations on LAN1 or LAN3.
- LAN2 is still used to convey traffic between LAN1 and 3

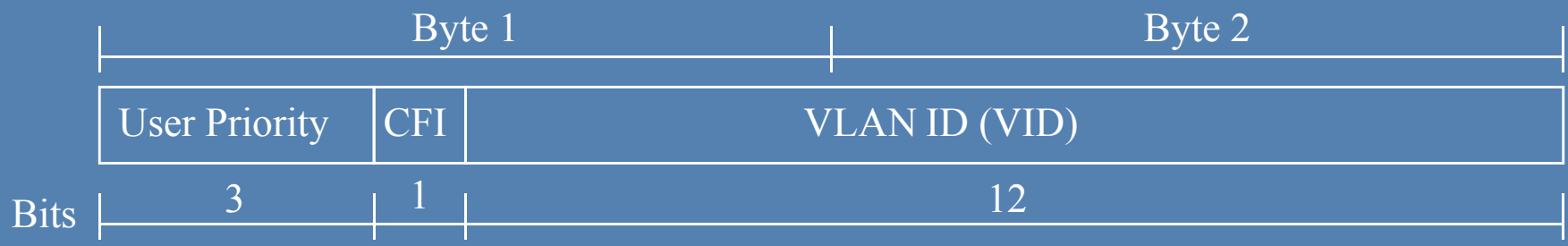
# VLAN Tags

- Depending on the network topology, traffic between different physical parts of the same VLAN may need to be “tunneled” through segments not belonging to that VLAN.
- These “transit packets” must contain some additional information:
  - The VLAN they belong to (this is encoded as a 12-bit number, assigned by the network manager)
  - The frame priority (to convey priority information over LANs that do not support it)
  - Information about MAC addressing (for packets travelling between different kinds of LANs)

# Inserting the VLAN tag (Ethernet)



# TCI Format




- User Priority: as defined in IEEE 802.1D
- CFI (Canonical Format Indicator): Specifies the bit ordering of the MAC addresses and the presence of source-routing data.
- VID: Identifies the VLAN. Values 0, 1 and 0xFFF are reserved.

# Jumbo Frames

- Standard Ethernet packets are limited to 1514 bytes
- CPU normally take a fixed amount of time to process a packet, regardless of its size
  - Increasing packet size can increase performance!
- For Ethernet at high speeds (1 Gb/s and higher), it is possible to increase the Ethernet frame size to about 9000 bytes – “Jumbo Frames”
- This is a configuration option; all devices in that network need to be able to support it

# Layer 3
# The Network Layer

# The Network Layer

- The key function of the Network Layer is end-to-end **routing**, possibly across multiple network segments.
- Depending on the technology, the Network Layer is also responsible for maintaining Quality of Service.
- In summary:
  - Data Link Layer (Layer 2) is responsible for hop-to-hop data communication
  - Network Layer (Layer 3) is responsible for end-to-end data communication
- The IP protocol “starts” at Layer 3

# The Internet Protocol (IP)

- The *Internet Protocol* defines an unreliable, connectionless, best-effort delivery mechanism for the Internet.
  - Unreliable: packet delivery is not guaranteed
  - Connectionless: packets are treated independently; multiple packets between two nodes may take different paths and arrive out-of-order
  - Best Effort: packets are discarded when underlying networks fail or resources are exhausted
- The protocol specifies packet formats, processing, and error-handling.
- Reference: RFC 791

# The Internet Datagram

- The *Internet Datagram* is the basic unit of information transfer in the Internet Protocol. It is divided into a datagram header and data area:

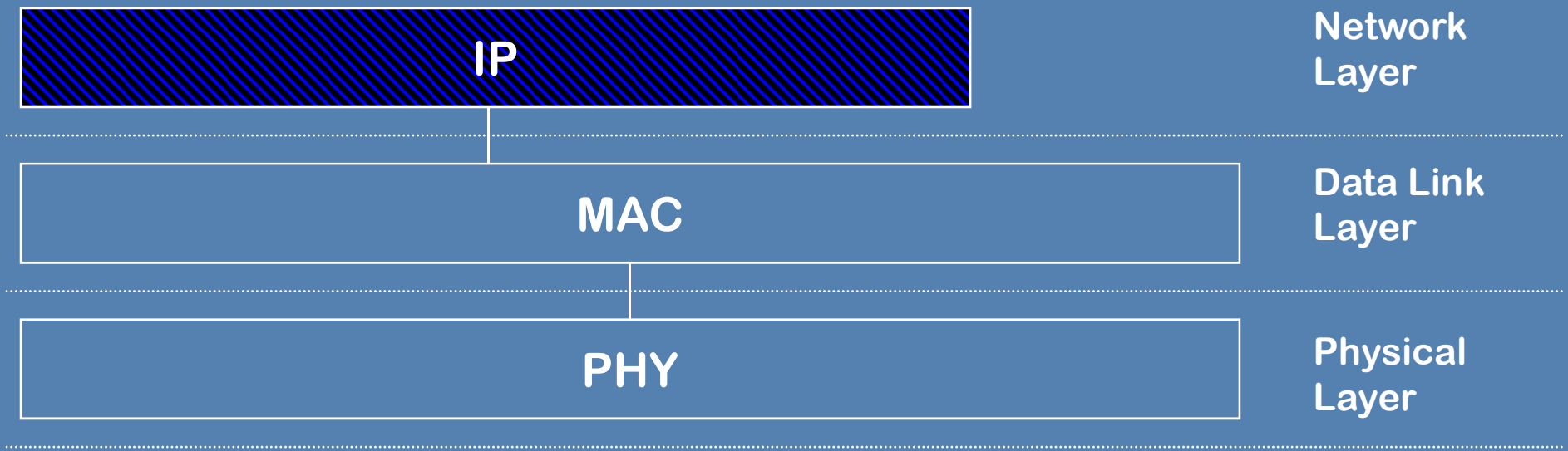

- The datagram is transported from source to destination through one or more intermediate networks. Within a *particular network*, it is *encapsulated* into a network frame:


- Sometimes a complete internet datagram will not fit into the frame size of an intermediate physical network. IP allows its datagrams to be *fragmented* . Once a datagram is fragmented, its fragments travel as separate datagrams all the way to the final destination.

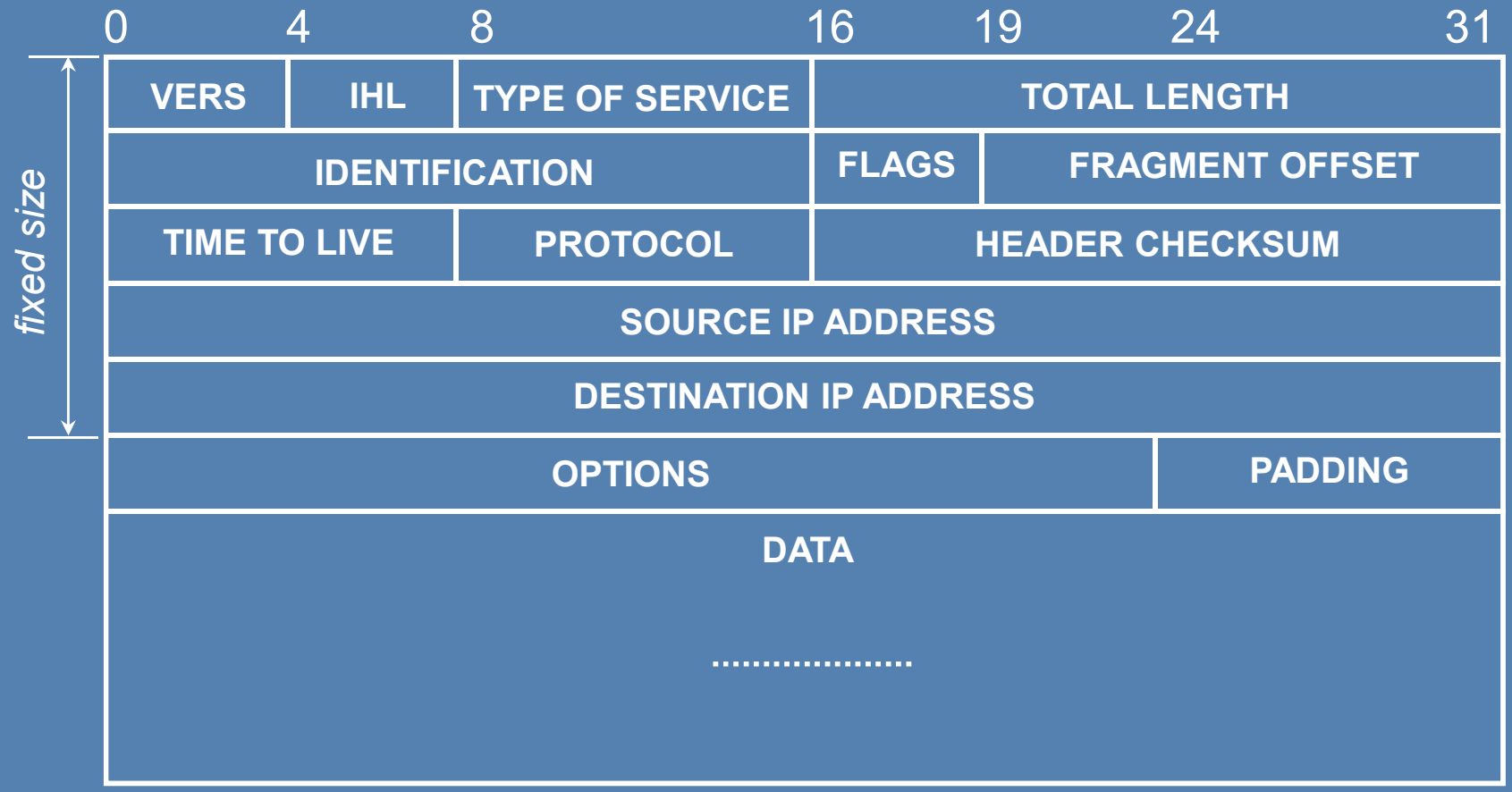
# Where does it fit?

| DST Address | SRC Address | Type/ Len | IP Packet |
|---|---|---|---|
| 6 | 6 | 2 | 46 to 1500 |

08-00



# Datagram Format



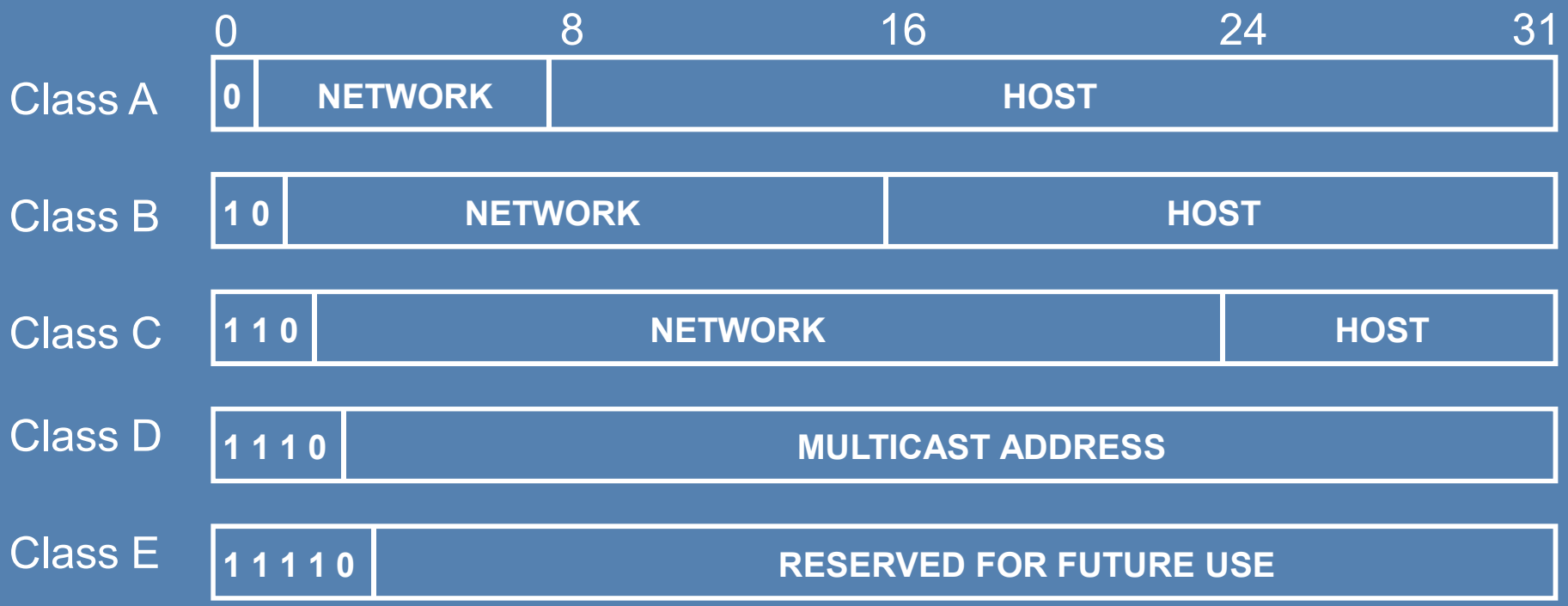
# Fragmenting IP Packets

- If an IP packet is too large to fit into the Layer-2 datagram, it can be “fragmented” (broken into pieces that fit)
- Each “piece” is an IP packet of its own, with a full IP header
- The header has information that allows the receiver to put together the original packet
- An IP packet can be fragmented “on route” (crossing network boundaries) or by the sender

# Internet Addresses

- IP addresses are unique over the whole Internet
- IP addresses are 32-bit long and consist of a *network address* part and a *host address* part. They are represented by the *dotted decimal* notation, where each byte is written in decimal values (from 0 to 255).
  - e.g. 10000000 00001010 00000010 00011110
  - 128. 10. 2. 30.
- IP addresses used to be grouped into classes depending on the size of the host part of the address
- The use of classes is currently mostly obsolete, although people in the business still refer to them

# Internet Address Classes




IP address formats

# Private Internets

- Private Internets have no direct connection to the Internet
- Blocks of addresses have been reserved for Private Internets (RFC1918); these addresses will never be used in the Internet.
- Reserved blocks:
  - 10.0.0.0 - 10.255.255.255 (1 class-A network)
  - 172.16.0.0 - 172.31.255.255 (16 class-B networks)
  - 192.168.0.0 - 192.168.255.255 (256 class-C networks)

# Translating Between IP Addresses and MAC Addresses

- Each interface has an IP address at Layer 3, and a MAC address at Layer 2.
- Assume that host A wants to send a packet to host B.
- Host A knows the IP address of host B; however, in order to transmit the packet, host A must somehow know or find out what is the MAC (layer 2) address of host B.
- Solution: the Address Resolution Protocol (ARP), RFC826

ARP

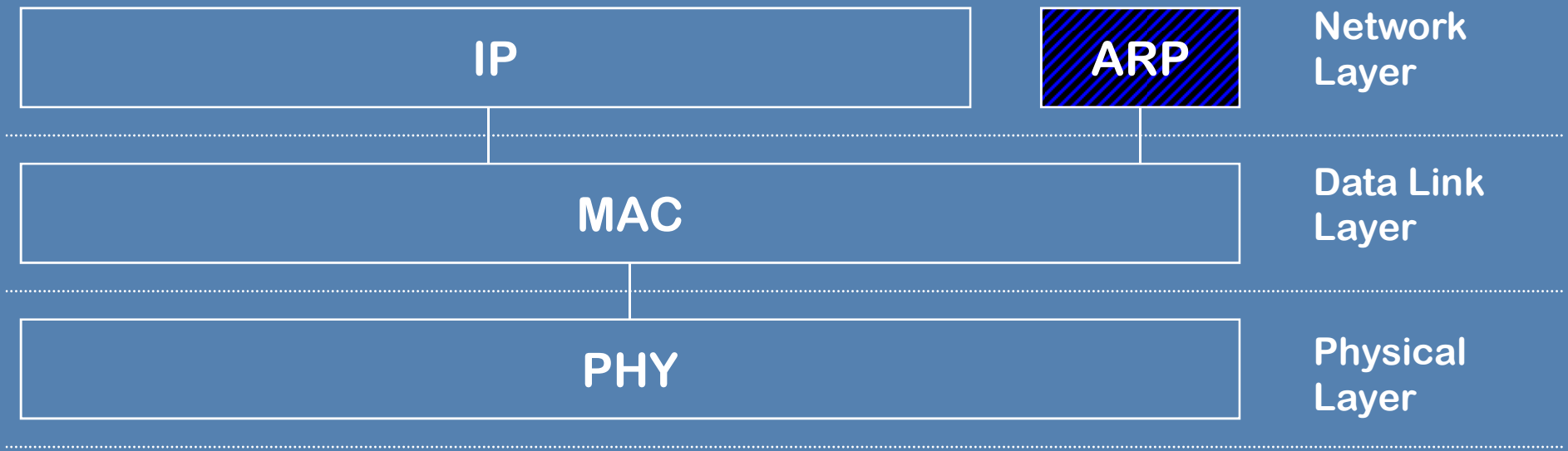
# Address Resolution Protocol

- Used to find the physical address of a target host on the local physical network, given only the host's IP address
- Mechanism:
  - The source broadcasts a special packet asking host with the target IP address to respond with a message carrying the (IP address, physical address) mapping
  - All hosts on the local physical network receive the broadcast, but only the target recognizes its IP address and responds to the request
  - When the source receives the reply, it sends the packet to the target using the target's physical address and places the mapping in its cache (a cache is used to prevent repeated broadcasts for the same destination)

# Where does it fit?

| DST Address | SRC Address | Type/ Len | ARP Packet |
|---|---|---|---|
| 6 | 6 | 2 | 46 |

08-06



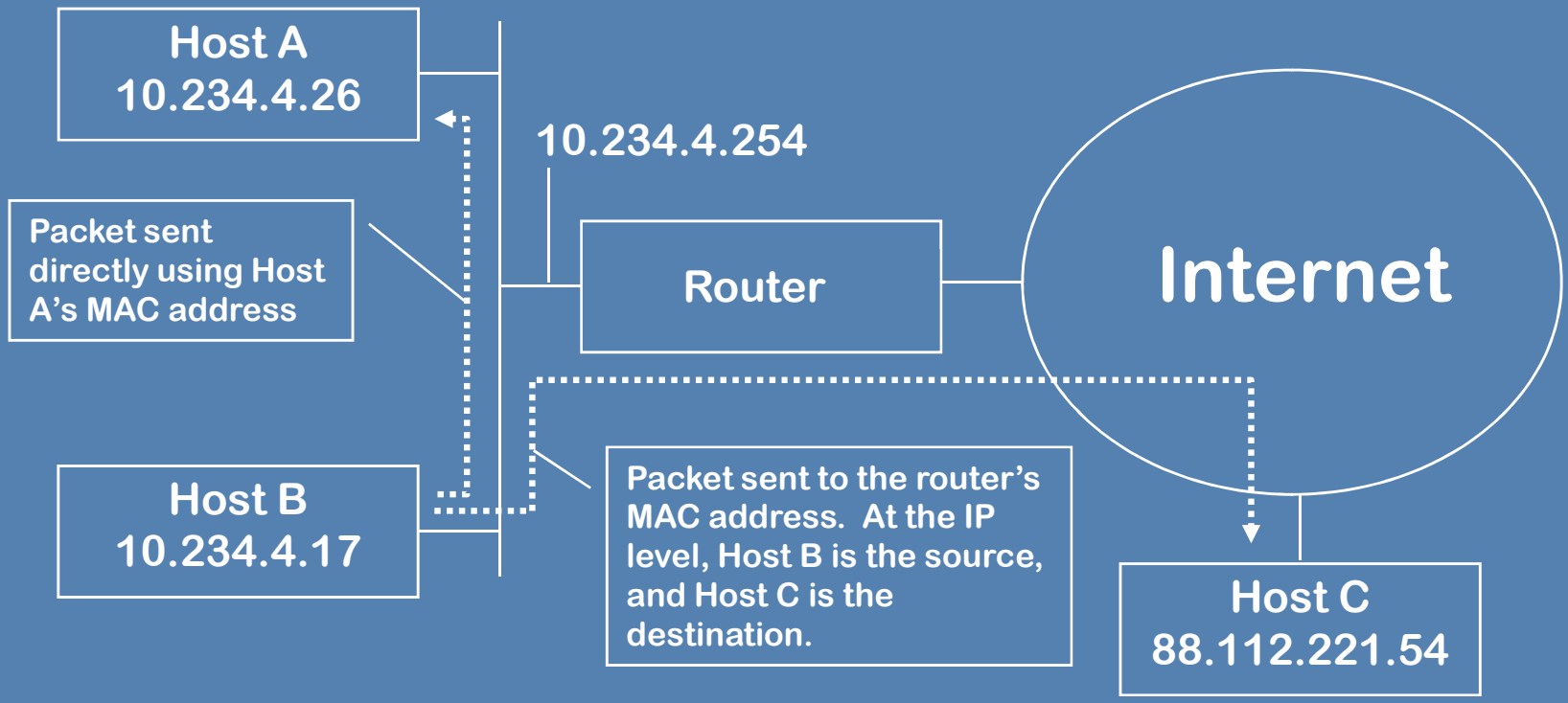
# Routing IP Datagrams (1)

- Direct Delivery :
  - Transmission of an IP datagram between two machines on a single physical network does not involve routers
  - The sender encapsulates the datagram in a physical frame, binds the destination IP address to a physical hardware address, and sends the resulting frame directly to the destination.

# Routing IP Datagrams (2)

- Indirect delivery
  - through intermediate routers
  - At the MAC layer, the packet gets sent to the router's MAC address
  - routing decisions are made based on network prefixes (not full IP address)
  - "Next hop" refers to next router IP address on the route to destination

# Routing IP Datagrams (3)



# Actual Example

**Direct delivery: both at layer 2 and at layer 3, the packet is sent directly to the destination.**

```
⊞ Frame 94 (182 bytes on wire, 182 bytes captured)
⊟ Ethernet II, Src: Dell_88:6e:ee (00:19:b9:88:6e:ee), Dst: Intel_09:4c:b4 (00:d0:b7:09:4c:b4)
  ⊞ Destination: Intel_09:4c:b4 (00:d0:b7:09:4c:b4)
  ⊞ Source: Dell_88:6e:ee (00:19:b9:88:6e:ee)
    Type: IP (0x0800)
⊞ Internet Protocol, Src: 10.234.4.17 (10.234.4.17), Dst: 10.234.4.26 (10.234.4.26)
⊞ Transmission Control Protocol, Src Port: 3695 (3695), Dst Port: netbios-ssn (139), Seq: 2688, Ack: 2
⊞ NetBIOS Session Service
⊞ SMB (Server Message Block Protocol)
```

**Indirect delivery: at layer 2, packet is sent to the router; at layer 3, to the destination.**

```
⊞ Frame 92 (154 bytes on wire, 154 bytes captured)
⊟ Ethernet II, Src: Dell_88:6e:ee (00:19:b9:88:6e:ee), Dst: ExtremeN_85:c1:00 (00:e0:2b:85:c1:00)
  ⊞ Destination: ExtremeN_85:c1:00 (00:e0:2b:85:c1:00)
  ⊞ Source: Dell_88:6e:ee (00:19:b9:88:6e:ee)
    Type: IP (0x0800)
⊞ Internet Protocol, Src: 10.234.4.17 (10.234.4.17), Dst: 88.112.221.54 (88.112.221.54)
⊞ User Datagram Protocol, Src Port: 3334 (3334), Dst Port: 54025 (54025)
  Data (112 bytes)
```

# Deciding between direct and indirect delivery

- Packets should be sent directly only to hosts “in the same sub-network”.
- Hosts need to know which range of addresses are “local”, to be sent directly, and which ranges are not, to be sent to a router or gateway.
- Previous example:
  - Hosts with addresses between 10.234.4.0 and 10.234.4.255 are “local”, and packets can be sent directly.
  - Hosts with addresses outside this range are “not local”, and packets must be sent to the Extreme Networks router at IP address 10.234.4.254.
    - Using ARP, we find that this router is at 00:E0:2B:84:C1:00

# Subnet Masks

- The number of addresses in a subnetwork is always a power of 2 (i.e., 8, 16, 256, etc.)
- This allows the IP address to be split into two parts:
  - A "Subnetwork Address", which is common for all addresses in that subnet (this is the part that "does not change" from host to host)
  - A "Host Address", which identifies that particular host
- In our example, the Subnet Address is 10.234.4 and the host address is 17 for one node (making it 10.234.4.17) and 26 for the other
- So, a way to denote which part of the address is the subnet is by using a **mask**
- The mask can be expressed as the IP address as X.Y.Z.W or as /N, where N is the number of bits "1" in the mask

# Subnet Mask Examples

**Range: 10.234.4.0 to 10.234.4.255, 256 addresses**
**10.234.4.0/24**

| | | | | |
|---|---|---|---|---|
| 10 | 234 | 4 | 17 | |
| 11111111 | 11111111 | 11111111 | 00000000 | **/24** |
| 255 | 255 | 255 | 0 | |

**Range: 172.17.0.0 to 172.17.255.255, 65536 addresses**
**172.17.0.0/16**

| | | | | |
|---|---|---|---|---|
| 172 | 17 | 240 | 11 | |
| 11111111 | 11111111 | 00000000 | 00000000 | **/16** |
| 255 | 255 | 0 | 0 | |

**Range: 63.193.215.216 to 63.193.215.223, 8 addresses**
**63.193.215.216/29**

| | | | | |
|---|---|---|---|---|
| 63 | 193 | 215 | 218 | |
| 11111111 | 11111111 | 11111111 | 11111000 | **/29** |
| 255 | 255 | 255 | 248 | |

# “Special” IP addresses

- 0.0.0.0 is sometimes used by the host before it knows its IP address
- 255.255.255.255 is the broadcast address, sent to MAC address FF:FF:FF:FF:FF:FF
- 127.0.0.0/8 is the **loopback** range – the packets come back to the same host (software loopback, they are not actually transmitted)
- The first address of a subnet is reserved and means “this subnet”. It must not be used. Example: 10.234.4.0 for the 10.234.4.0/24 subnet.
- The last address of a subnet is the “subnet broadcast” address; packets sent to it also use the FF:FF:FF:FF:FF:FF address. It must not be assigned to any host. Example: 10.234.4.255 for the 10.234.4.0/24 subnet.
  - Was intended to be used to send a broadcast packet to a remote subnet, but this is rarely enabled in the routers.

# Multicast: Basic Concepts

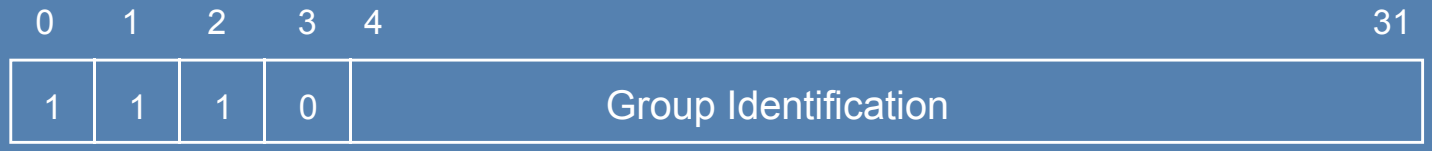
- Unicast:
  - Sent to a specific host.
- Broadcast:
  - Sent to all hosts
  - Restricted to a single LAN (local or remote)
- Multicast:
  - Sent to a group (not everybody)
  - Unlike broadcast, can go across routers
  - Main reason: efficiency, saves on bandwidth over possible alternatives:
    - Multiple unicast: packet sent many times.
    - Broadcast: hosts that are not interested in the packet get it.

# IP Multicast

- IP multicasting is the “Internet abstraction” of LAN multicasting.
- Membership of an IP- multicast group is dynamic.
  - A host may join or leave at any time
  - A host may be member of an arbitrary number of multicast groups.
- Membership determines whether the host will receive datagrams sent to the multicast group.
- A host may send datagrams to a multicast group without being a member.
- Basic definition: RFC 1112

# IP Multicast Addresses

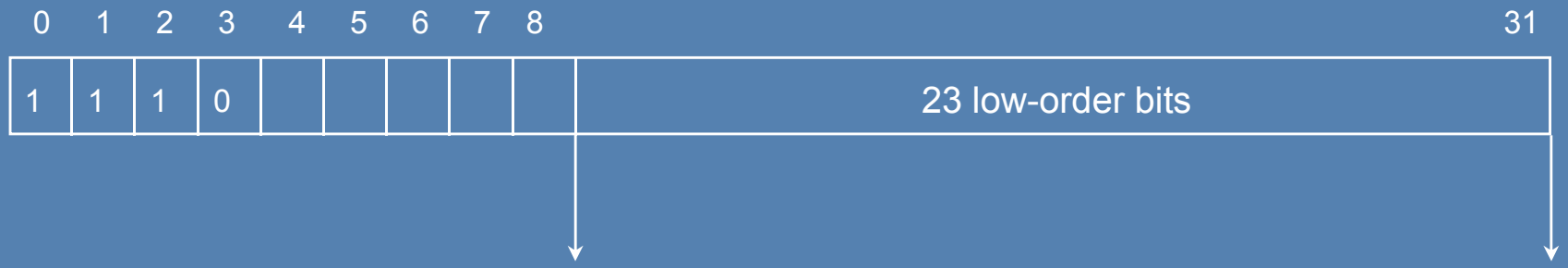
- A multicast group has a unique class D address




*Addresses range from 224.0.0.0 through 239.255.255.255*
*224.0.0.0 reserved.*
*224.0.0.1 all hosts group (all hosts & routers participating IP multicast)*

- Multicast addresses can be used only as destinations

# Mapping IP Multicast Address to Ethernet Multicast Address




*Ethernet special multicast MAC address*

$01.00.5E.00.00.00_{16}$

*with 23 low-order bits replaced by 23 low-order bits of IP multicast address*

*Note that 32 Class-D addresses map into the same MAC address, so software must be prepared to filter.*

*Example: 225.1.2.3 maps to 01-00-5E-01-02-03 (also 225.129.1.2.3, 226.1.2.3, ...)*

***When planning a multicast distribution system, make sure that the addresses differ in the last three bytes!!!***

# The IGMP Protocol

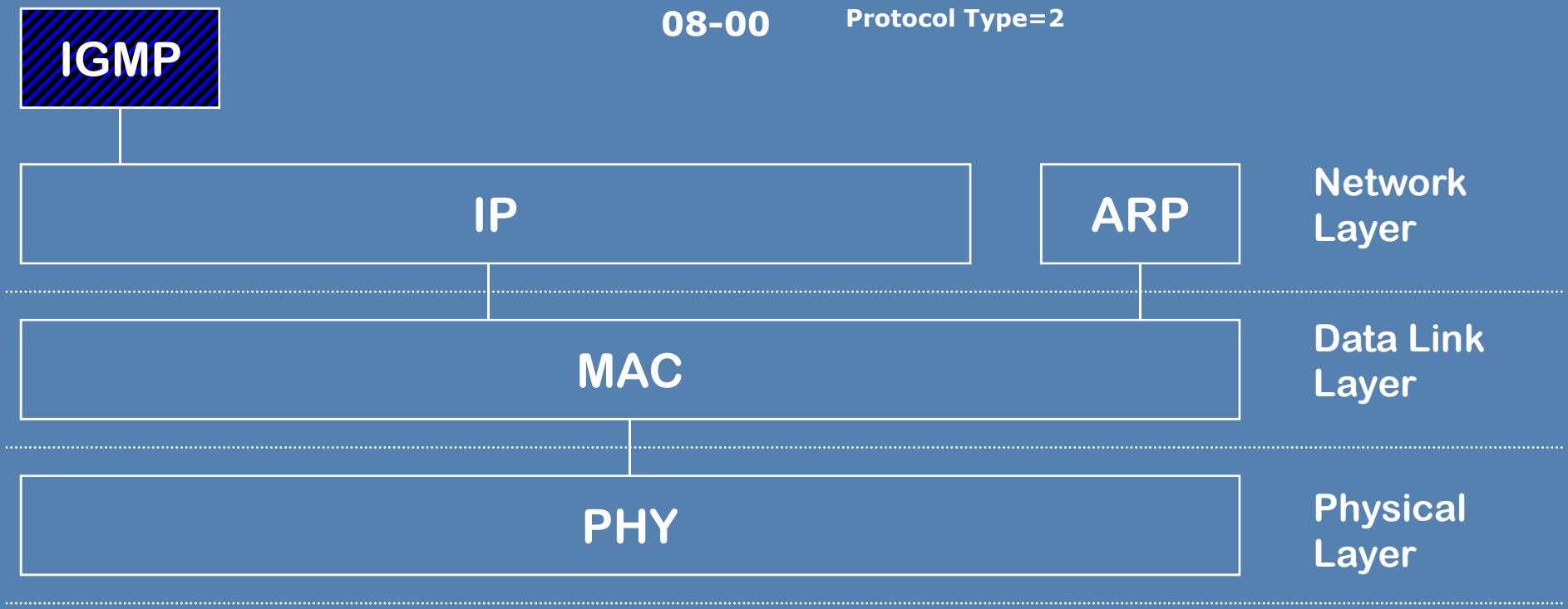
- The Internet Group Management Protocol (IGMP) is used by hosts to report group membership data to neighboring routers.
- Asymmetric protocol.
- Currently supported by most operating systems.
- IGMP Version 1: specified in RFC 1112.
- IGMP Version 2: specified in RFC 2236
- IGMP Version 3: specified in RFC 3376

# Where does it fit?

| DST Address | SRC Address | Type/ Len | IP Header | IGMP Packet |
|---|---|---|---|---|
| 6 | 6 | 2 | 20 | variable |

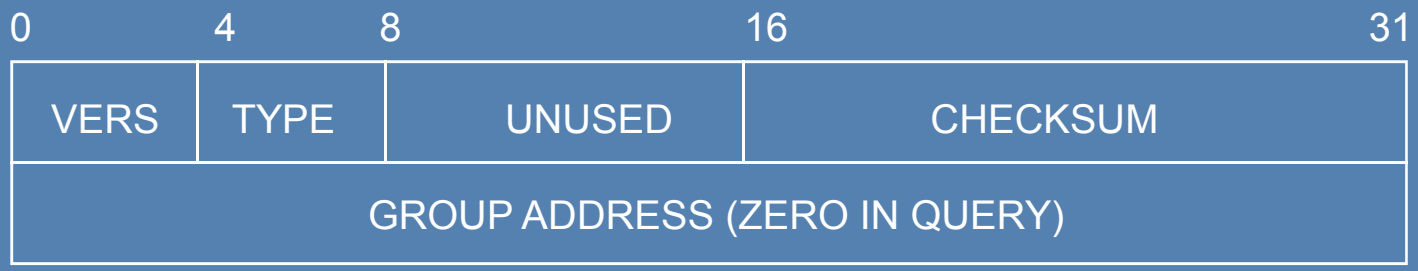
08-00

Protocol Type=2



# IGMP Version 1

- Defined in RFC 1112; carried over IP with protocol # 2
- Two messages:
  - Host membership query (Type = 1)
  - Host membership reports (Type = 2)
- Message Format:



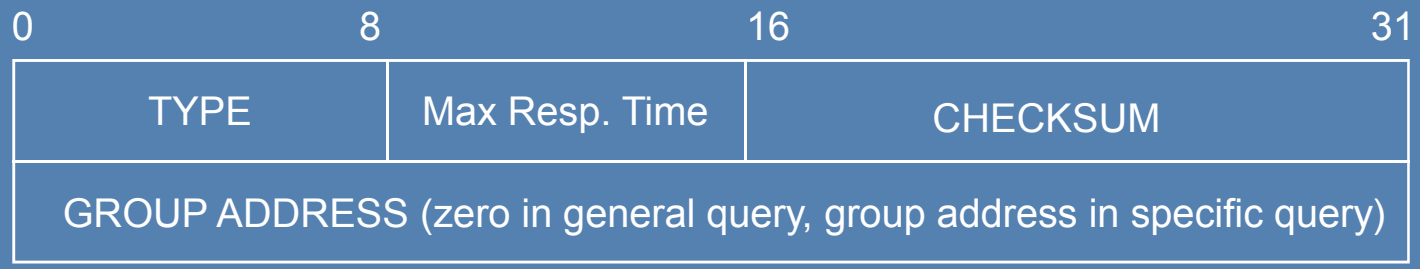

*Version = 1*

# IGMP V1 Operation

- When a host joins a group, it immediately sends a group membership report
- Multicast router send periodic queries to 224.0.0.1 (All-systems) with TTL = 1, group address = 0.
  - Hosts reply with one report message per group, sent to the group address.
  - Hosts replies are staggered using random delays
    - If within chosen delay, no report for the same group is heard, report is sent.
    - Otherwise, canceled.
- To leave a group, the host just stops responding to queries; routers drop the group if nobody responds

# IGMP V2 Additions

- Key addition: group membership leave message (to speed up the group leaving process).
- Also adds a group-specific query, in addition to the general membership query.

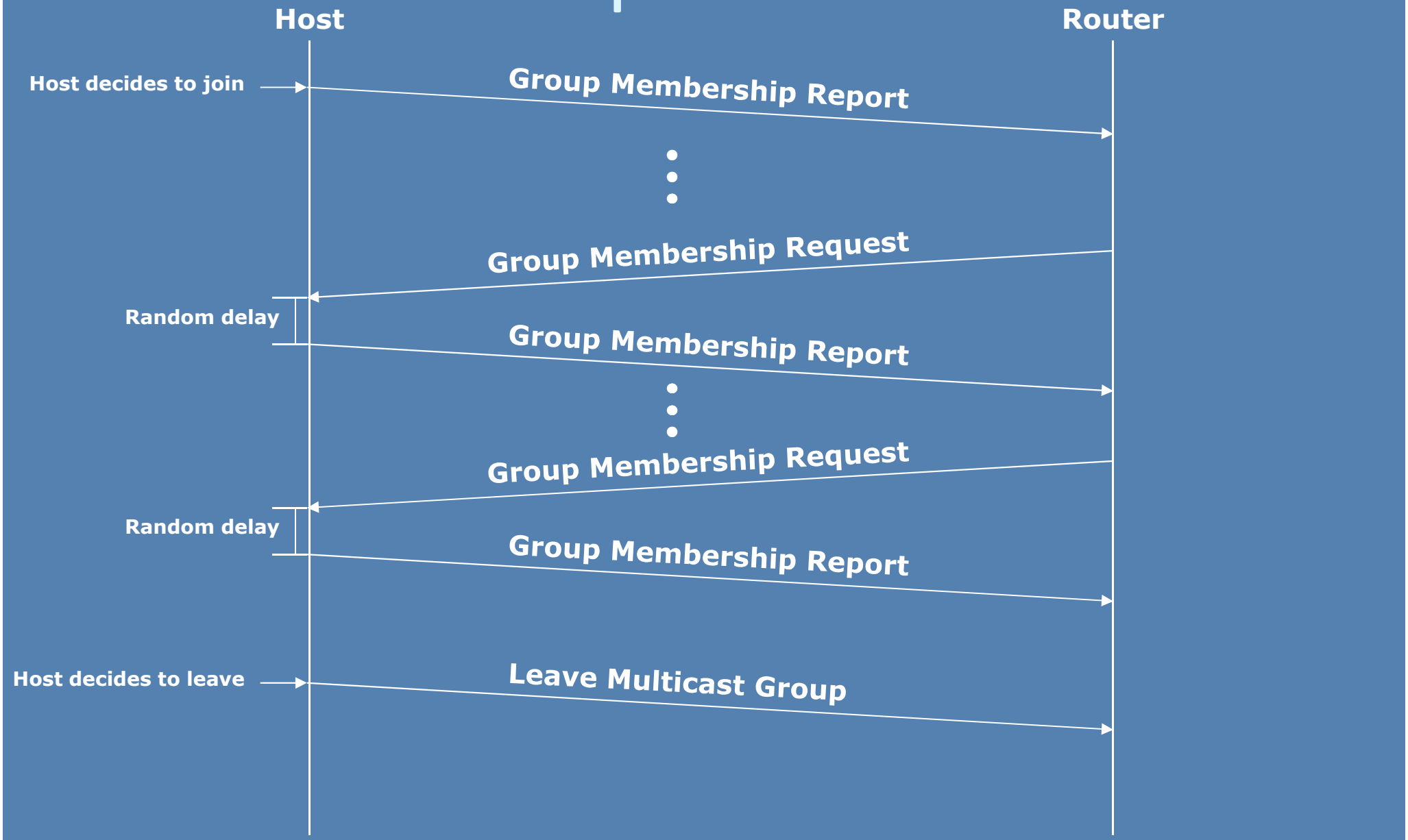



Type field:

0x11: Membership Query
0x16: Version 2 Membership Report
0x17: Leave Group
0x12: Version 1 Membership Report

IGMP V2 interoperates with V1

# IGMP V2 Example



# IGMP V3 Additions

- Introduces mechanisms to allow a host to:
  - Elect to receive traffic only from certain sources in the multicast group.
  - Explicitly identify sources in the multicast from which it does not want to receive.
  - Leave a whole multicast group, or leave (stop receiving) from certain sources in the group.
- IGMP message enhanced to include a listing of sources.

# Reading

- Very good tutorial paper on IP Multicast technology, including the various multicast routing protocols:

  "Introduction to IP Multicast Routing", by Chuck Semeria and Tom Maufer, 3Com Corp.

- Available on-line from:

  **http://www.kohala.com/start/papers.others/draft-ietf-mboned-intro-multicast-03.txt**

# Layer 4
# The Transport Layer

# User Datagram Protocol (UDP)

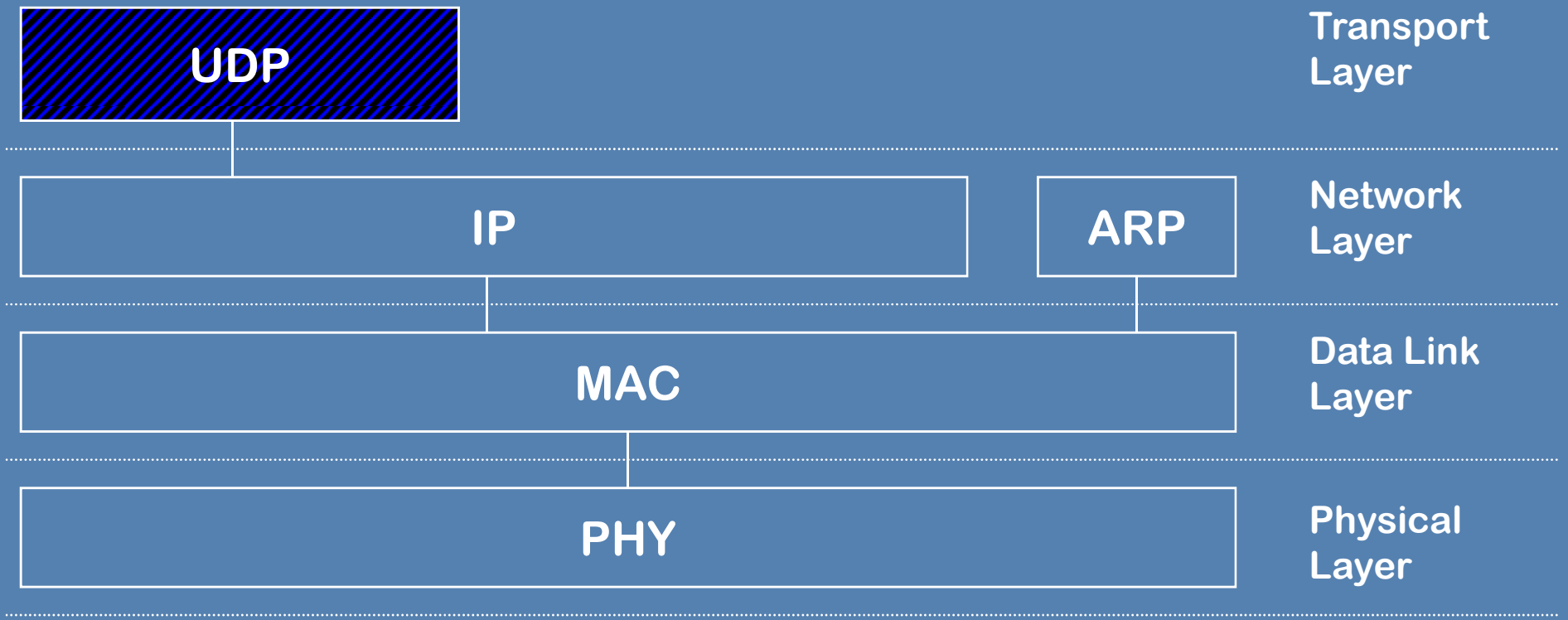
- Basic Idea: provide the same basic service as IP to the application layer, with the following additions:
  - Data checksum to increase “confidence” in the received bytes; packets failing checksum are dropped.
  - Application demultiplexing: introduce the concept of ports to allow a packet to be addressed to a specific process in the host.
- UDP is a very “thin” layer over IP; only 8 bytes of header.
- Each UDP datagram becomes one IP packet (which may be fragmented).
- UDP is described in RFC 768.

# Where does it fit?

| DST Address | SRC Address | Type/ Len | IP Header | UDP Header | UDP Packet |
|---|---|---|---|---|---|
| 6 | 6 | 2 | 20 | 8 | variable |

08-00

Protocol Type=0x11



# UDP Header

| 2 bytes | 2 bytes | 2 bytes | 2 bytes |
|---|---|---|---|
| Source Port | Destination Port | Length | Checksum |

- Destination Port identifies the process (program) in the destination host who should receive the data.
- Source Port identifies the sending process (program), in case the receiver wants to answer – can be set to zero if no answer is required or expected.
- Length gives the total length of the datagram.
- Checksum protects the data.
  - Can be set to zero if the sender did not compute this and fill in the field.

# Transmission Control Protocol (TCP)

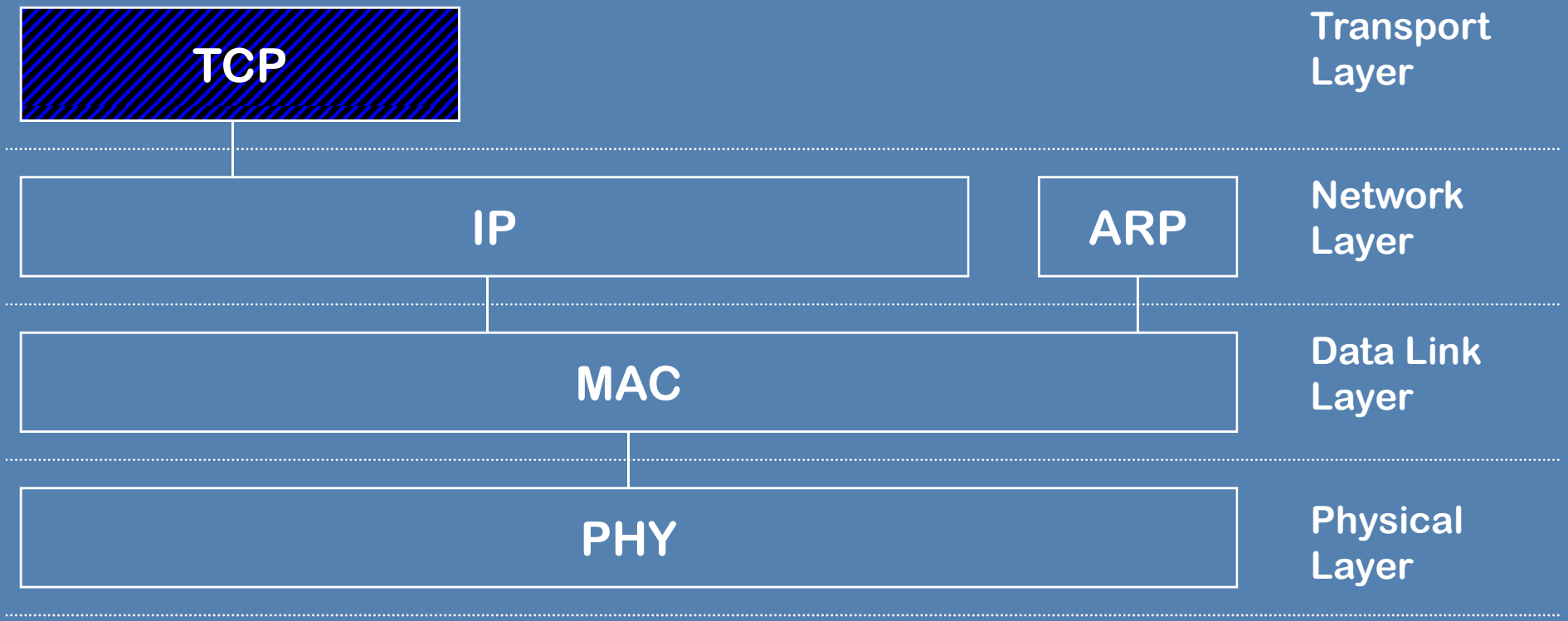
- TCP creates a service over IP as follows:
  - Connection oriented: a connection needs to be established before hosts can communicate. Data is delivered strictly in the order sent.
  - Reliable: the underlying protocol takes care of data reliability, by using checksums, timeouts, and retransmission.
  - Data stream transfer: the data to be sent is seen as a stream of bytes. No packet boundaries are visible to the application. Data is packetized by the underlying protocol.
  - Flow Control from the receiver to the sender.
  - Process demultiplexing: TCP also has the concept of ports, to allow data delivery to individual processes inside the host.
  - Latency: TCP has no guarantees or specification for latency, data may take “a long time” to transfer.
  - TCP is defined in RFC 793.

# Where does it fit?

| DST Address | SRC Address | Type/ Len | IP Header | TCP Header | TCP Packet |
|---|---|---|---|---|---|
| 6 | 6 | 2 | 20 | 20 | variable |

08-00

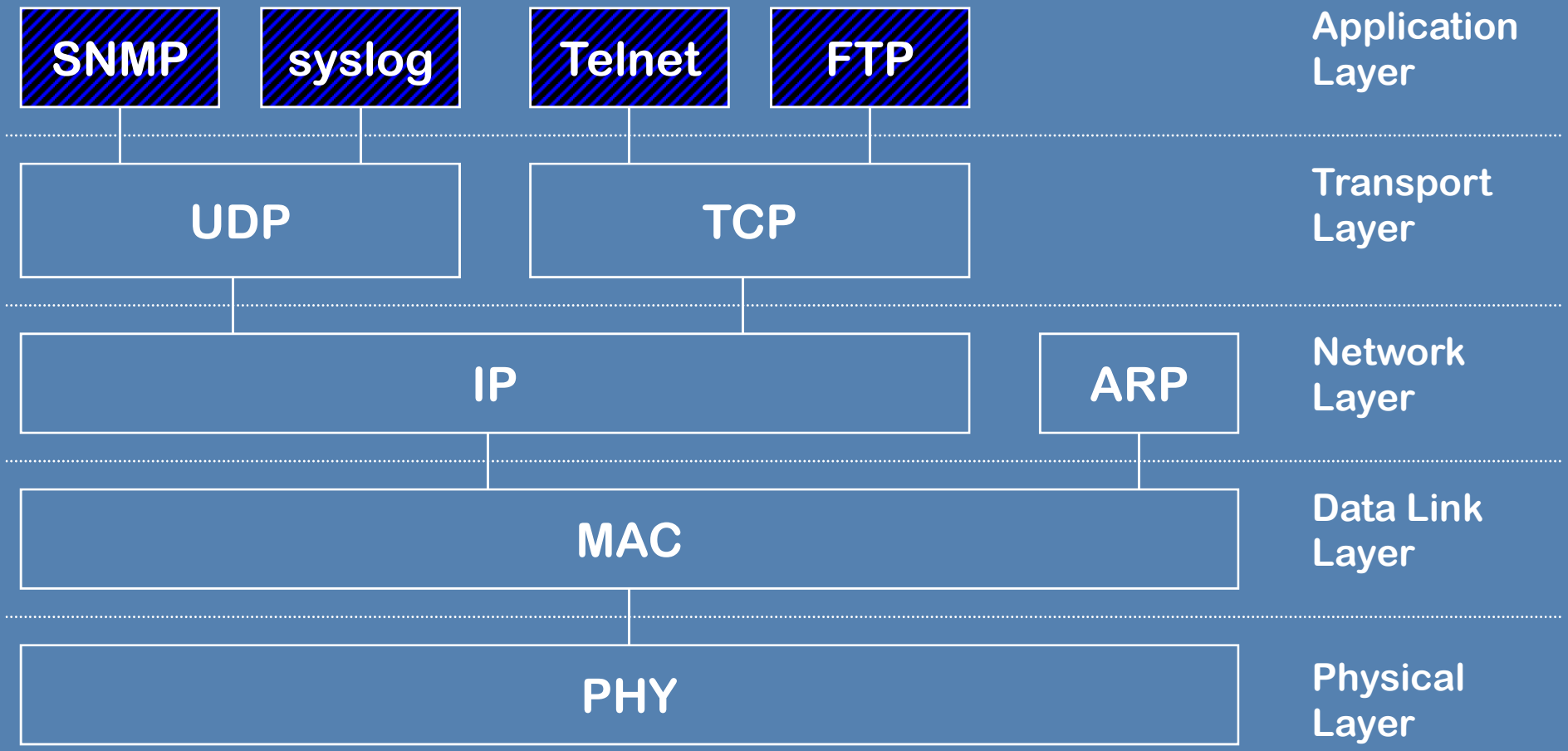
Protocol Type=6



# The Application Layer

Discussion of Selected Applications

# Where does it fit?



# Application Overview

- Virtually all Internet applications run on top of TCP or UDP.
- Applications that require reliable data transfer run on top of TCP:
  - Web, Telnet, FTP, SSH
  - Data delivery timing is non-critical
- Applications where timing is important or reliable data transfer is not required run on top of UDP
  - SNMP, syslog, real-time streaming (voice and video)

# Applications on top of TCP

- Common applications on top of TCP:
  - HTTP: used by web servers to serve pages. Uses TCP and runs on port **80**.
  - Telnet: used to get terminal connections using the network, without encryption. Uses port **23**.
  - SSH: same as telnet, but with encryption. Uses port **22**.
  - FTP: provides file transfer. Uses two TCP connections, one for control, and another for data. The control connection uses port **21**; the data connection uses a dynamically-chosen port negotiated in the control connection.
  - SMTP: e-mail transfer. Uses port **25**.
  - SMB: Microsoft file services ("Windows Shares"). Uses TCP port **139**.
  - DashBoard: (proprietary "CAN over IP"). Uses TCP port 5253.
- Many of these applications implement an application-specific protocol on top of TCP.

# Applications on top of UDP

- Streaming applications:
  - Voice over IP
  - MPEG over IP
- Other common applications over UDP:
  - SNMP (Simple Network Management Protocol): this is a “remote control” protocol. Devices expose read-only or read-write parameters, and managers can read or modify these parameters. SNMP runs over UDP ports **161** and **162**.
  - DHCP: Protocol used to automatically configure network nodes. Uses UDP ports **67** (server) and **68** (client).

# A Closer Look at Streaming

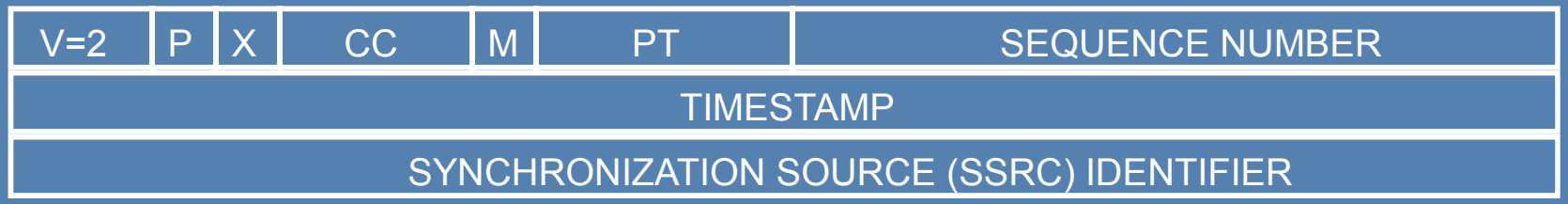
- In the broadcast world, real-time streaming is always done on top of UDP, because:
  - No flow control in the protocol
  - Support for multicast
- However, UDP has deficiencies:
  - No re-ordering support – if packets get out of order, UDP cannot do anything about it
  - No mechanism to recover from packet loss
- UDP streaming variants:
  - Transport stream over UDP
  - Transport stream over RTP/UDP

# Transport Streaming over UDP

- Simplest form of streaming – take an integral number of transport packets, place them in the UDP payload
- Supported by virtually all equipment
- No actual standard for it
- Industry practices:
  - Integral number of packets per UDP datagram – no splitting a transport packet between datagrams
  - Number of transport packets per UDP datagram: 7
    - Less than 7: inefficiency (short packets)
    - More than 7: packets become fragmented – many devices do not support that

# The Real-Time Transport Protocol (RTP)

- Runs on top of UDP, on even UDP ports
- Defined by RFC 1889, updated by RFC 3350
- Adds a 12-byte header to the UDP payload:




- **PT:** Payload Type, MPEG TS is 33
- **Sequence Number:** 16-bit packet count, increments by one for each packet
  - Allows for detecting lost and out-of-order packets
- **Timestamp:** 32-bit timestamp

# MPEG Transport over RTP

- Defined in RFC 2250, further detailed on SMPTE 2022-2
- Highlights:
  - Integral number of transport packets per RTP packet
  - Timestamp frequency: 90 kHz
  - P, X, M and CC fields in the header are set to zero
  - No fragmentation (maximum of 7 transport packets per RTP packet)
    - Recommended values are 1, 4 and 7
  - TS packet size: 188 bytes
    - Support for 204-byte packets over IP is optional (and extremely uncommon in practice)

# SMPTE 2022 FEC

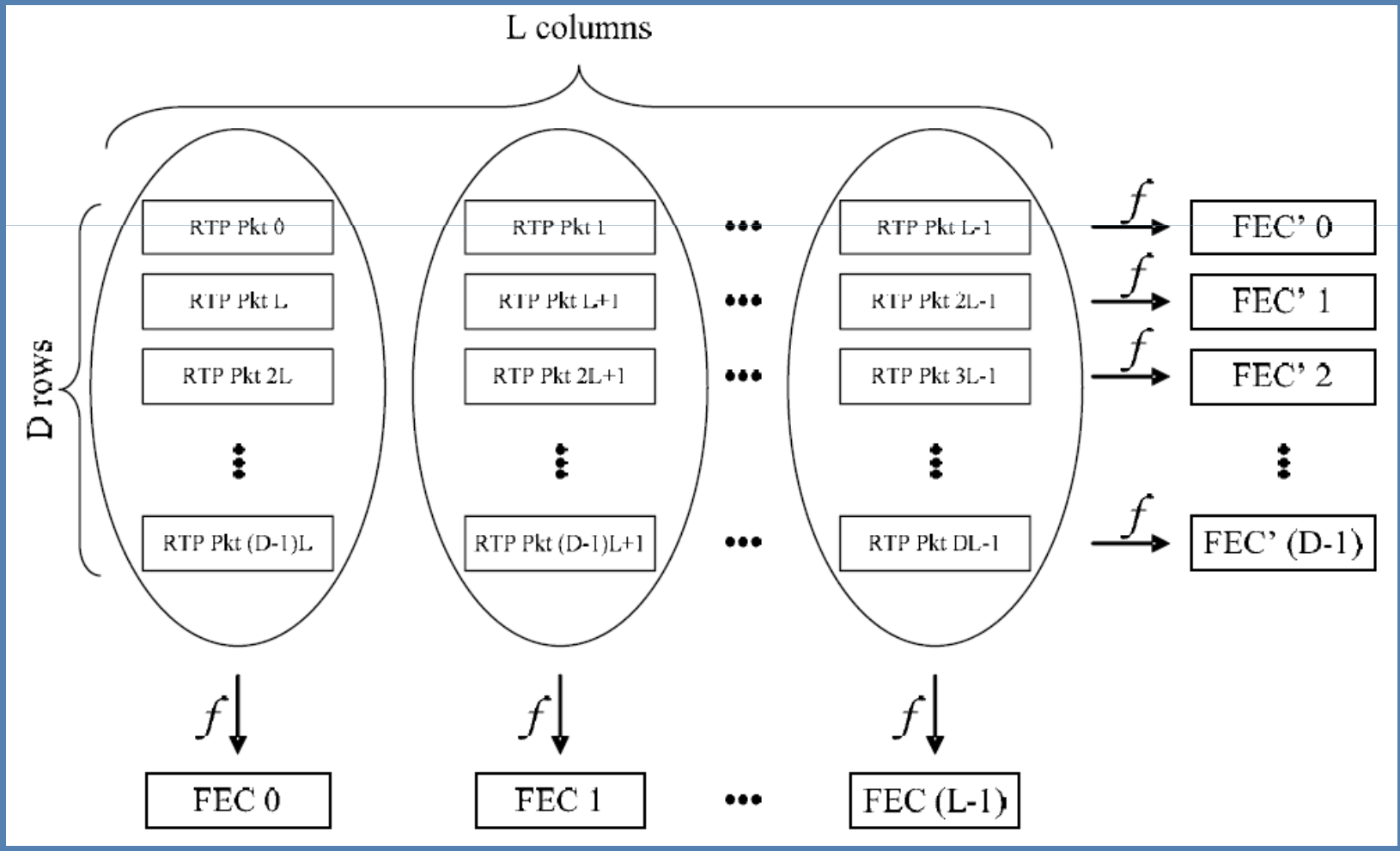
- RTP allows for detection of lost packets
- Need a way to recover lost packets under the following conditions:
  - One sender, possibly a large number of receivers
  - One-way communication channels
- Solution: send a certain amount of "redundant" data
  - Lost packets may be recovered from the received packets plus the redundant data
  - Use a class of error-correcting codes called "erasure codes"
- Standard: SMPTE 2022 FEC (previously Pro-MPEG COP3 FEC)

# XOR-based FEC Basics

- Basic theory: send a block of N packets, and then a packet with their "sum".
- If one packet is lost, it can be recovered by "subtracting" the received packets from the "sum".
- Example:
  - Send: 2 3 5 10 (sum)
  - Receive: 2 loss 5 10
  - Recover: 10 – 5 – 2 = 3
- In the practical implementation, the "sum" and "subtraction" are implemented as bit XOR operations, just like in disk arrays.

# Matrix-Based FEC (SMPTE 2022)

- Packet losses typically come in bursts
- Solution: arrange the packets in a matrix, do the FEC per column



# Implementation Details

- MPEG Transport Bitstream is sent on UDP Port P
- Column FEC is sent on UDP Port P+2
- Row FEC, if enabled, is sent on UDP Port P+4
  - Receivers will only "see" the packets corresponding to the level of FEC they support
- FEC packets have an extra 16-byte header from which all the parameters can be derived
  - No additional configuration is required on receivers
- Constraints:
  - Rows × Columns ≤ 100
  - 1 ≤ Columns ≤ 20
  - 4 ≤ Rows ≤ 20

# A Closer Look at SNMP

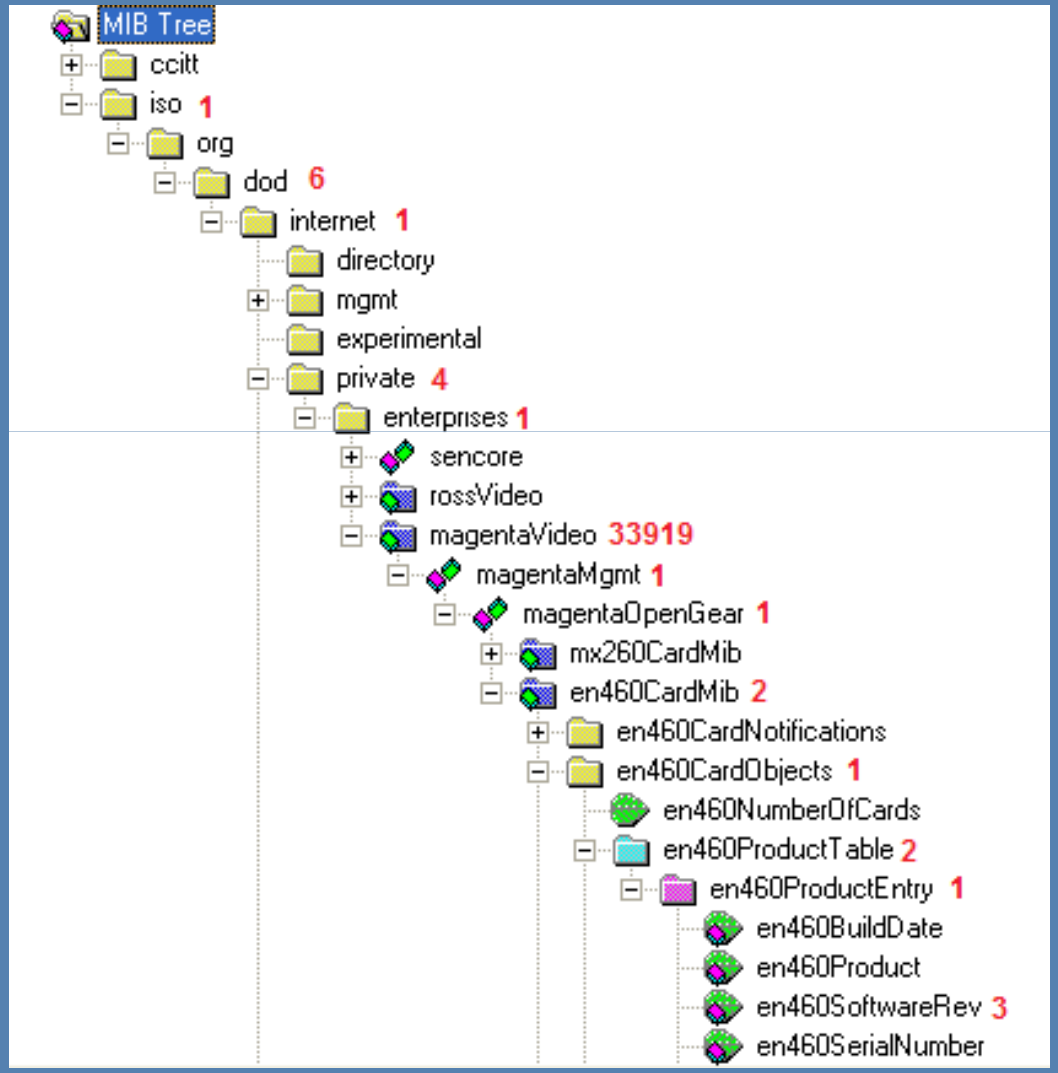
- SNMP was introduced in 1988 as a short-term interim solution to manage network devices
  - However, the "final solution", CMIP, never caught up…
- SNMP uses five basic messages:
  - The **GET** and **GET-NEXT** messages allow a manager to request the value of a given available variable
  - The **GET-RESPONSE** message is the device's response to the manager's request; either contains the value or an error code
  - The **SET** message is used by the manager to request a change to a specific variable. The device uses the GET-RESPONSE message to either acknowledge the SET or provide an error code.
  - The **TRAP** message is issued by the device to alert the manager to an event that has just happened; TRAP is sent on UDP port 162.

# SNMP MIBs

- The manageable variables in a device are described by one or more “Management Information Base” (MIB)
- The MIB is a hierarchical list of all the parameters that can monitored and/or controlled in the device
- Variables are identified by a string of numbers
- Example: an encoder software rev is accessible through:

  1.3.6.1.4.1.33919.1.1.2.1.2.1.3
- MIBs are communicated using text files in the ASN.1 format
- SNMP managers “compile” the MIB into some internal format

# MIB Example




Example: encoder software rev base OID:

1.3.6.1.4.1.33919.1.1.2.1.2.1.3

In the openGear chassis, the final OID for any variable includes **.1.N** at the end, where **N** is the slot number.

Software rev for encoder at slot 4:

1.3.6.1.4.1.33919.1.1.2.1.2.1.3**.1.4**

# SNMP Authentication

- SNMP has the concept of “communities”
- Communities can have “read-only” or “read-write” rights.
- The “Community Name” is the same as a password – if you know it, you can access.
- Default community names in common use:
  - Read-Only: **public**
  - Read-Write: **private**
- This mechanism is not secure at all – community names are sent in the clear.
- SNMP V3 adds crypto authentication
  - Not supported in openGear