# 5 Computer Networks Communications Architecture and Protocols

Part 10

Internetworking and

Network Layer Concepts

# Internetworking Terms (1)

- internet : collection of networks interconnected by router and/or bridges
- The Internet - note upper case I
  - The global collection of thousands of individual machines and networks
- Intranet
  - Corporate internet operating within the organization
  - Uses Internet (TCP/IP and http) technology to deliver documents and resources

# Internetworking Terms(2)

- End System (ES)
  - Device attached to one of the networks of an internet
  - Supports end-user applications or services
  - ES sometimes called DTE
- Intermediate System (IS)
  - Device used to connect two networks
  - Permits communication between end systems attached to different networks
  - Examples: Routers and Bridges

# Internetworking Terms (3)

- Bridge
  - IS used to connect two LANs using similar LAN protocols
  - Address filter passing on packets to the required network only
  - OSI layer 2 (Data Link)
- Router
  - Connects two (possibly dissimilar) networks
  - Uses internet protocol present in each router and end system
  - OSI Layer 3 (Network)

# Network layer services

- The goals of the network layer services are:
  - Provide services to the transport layer that are independent of the subnet technology in use
  - Shield the transport layer from the vagaries of the underlying network
  - Provide uniform addresses to the transport layer
- To meet these goals, the primary function of the network layer is *routing*

# Network service Approaches

- Connection oriented
- Connectionless

# Connection Oriented

- A connection is established between ES’s that is used for duration of call
  - Call setup
  - Data transfer
  - Call  termination
- E.g: Virtual circuits at this layer
- IS’s connect two or more networks
  - IS appear as ES to each network
  - Logical connection set up between ESs
    - Concatenation of logical connections across networks
  - Individual network virtual circuits joined by IS

# Connectionless service

- Each packet sent independently
- Routing decisions made at every IS
- Corresponds to datagram service in packet switched network
- Network layer protocol common to all ES’s and routers
  - Known generically as the internet protocol
- Internet Protocol
  - One such internet protocol developed for ARPANET

# Connectionless service

- Advantages
  - Flexibility
  - Robust
  - No unnecessary overhead
- Unreliable
  - Not guaranteed delivery
  - Not guaranteed order of delivery
    - Packets can take different routes
  - Reliability is responsibility of next layer up (e.g. TCP)

# Routing

- Determine *path* or *route* that packets will follow
- Use *routing protocol* based on a *routing algorithm*
- “Good” path should be *least* cost path
- Cost
  - Average queuing delay
  - Propagation delay
  - Bandwidth, mean queue length, etc.
- End systems and routers maintain routing tables
- *Dynamic* or *static*

# *tracert*: Program to display the path a packet traverse

MS-DOS Prompt

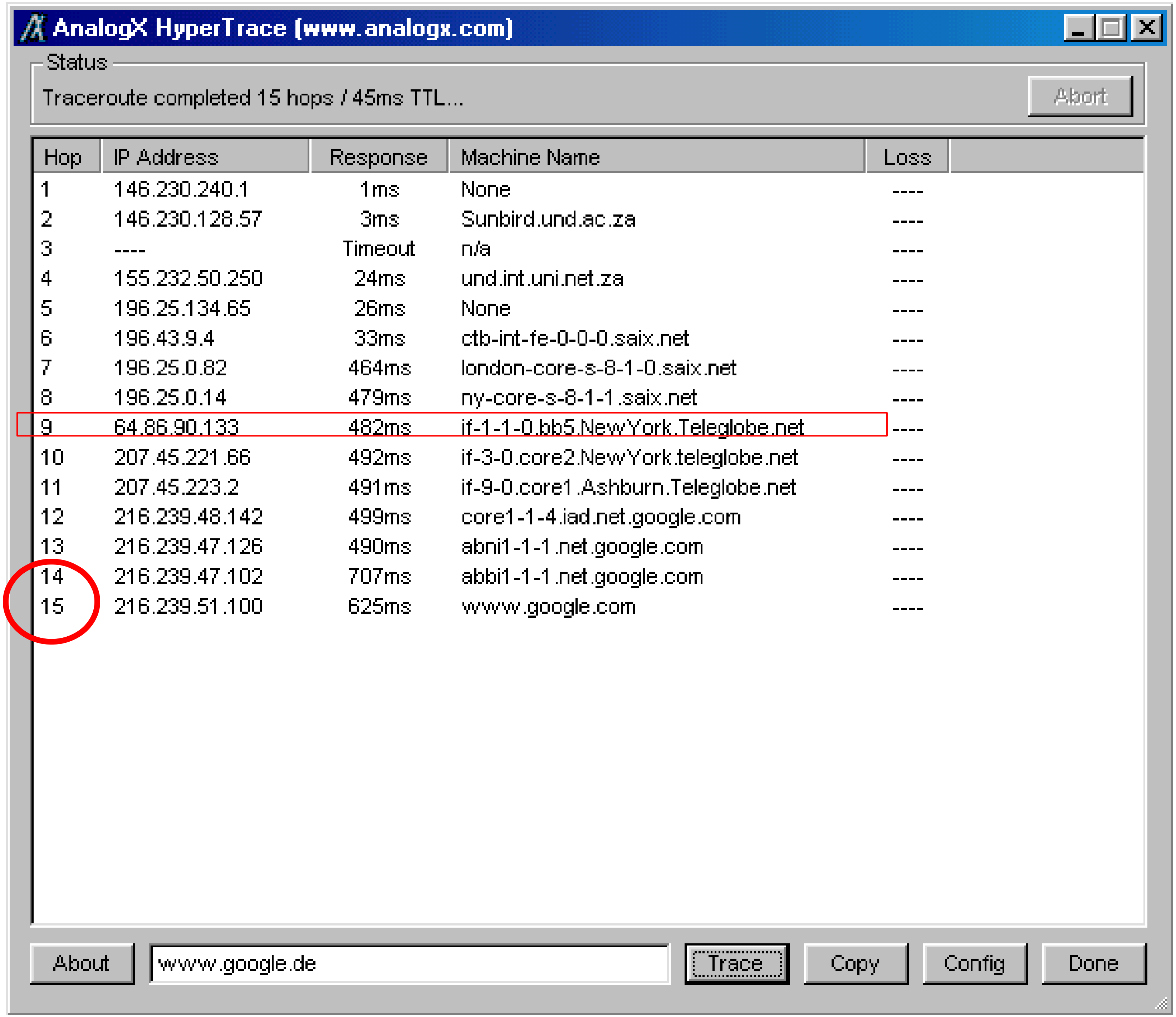
```
  4    26 ms    26 ms    25 ms  und.int.uni.net.za [155.232.50.250]
  5    48 ms    25 ms    27 ms  196.25.134.65
  6    26 ms    30 ms    25 ms  ctb-int-fe-0-0-0.saix.net [196.43.9.4]
  7   603 ms   608 ms   628 ms  london-core-s-8-1-0.saix.net [196.25.0.82]
  8   511 ms   503 ms   500 ms  ny-core-s-8-1-1.saix.net [196.25.0.14]
  9   499 ms   486 ms   493 ms  bar7-serial5-1-0-0.NewYork.cw.net [166.63.157.15
3]
 10   459 ms   465 ms   486 ms  agr2-loopback.NewYork.cw.net [206.24.194.102]
 11   518 ms   535 ms   534 ms  dcr1-so-7-1-0.NewYork.cw.net [206.24.207.69]
 12   541 ms   522 ms   526 ms  agr3-so-4-0-0.NewYork.cw.net [206.24.207.74]
 13   609 ms   591 ms   601 ms  acr1-loopback.Washington.cw.net [206.24.226.61]

 14   495 ms   498 ms   474 ms  pos4-0.per1.iad1.us.mfnx.net [216.200.254.145]
 15   469 ms   471 ms   506 ms  pos10-0.mpr1.iad1.us.mfnx.net [209.249.203.34]
 16   589 ms   567 ms   558 ms  so-6-1-0.mpr4.sjc2.us.mfnx.net [216.200.127.26]

 17   558 ms   566 ms   584 ms  so-1-0-0.cr2.sjc3.us.mfnx.net [208.184.233.50]
 18   568 ms   566 ms   553 ms  pos1-0.er2a.sjc3.us.mfnx.net [208.185.175.198]
 19   530 ms   524 ms   570 ms  sjni1-gige-1-1.google.com [64.124.78.29]
 20   558 ms   604 ms   563 ms  sjbi1-1-1.net.google.com [216.239.47.162]
 21   585 ms   591 ms   593 ms  www.google.com [216.239.35.101]

Trace complete.

C:\>
```


AnalogX HyperTrace (www.analogx.com)
Status
Traceroute completed 15 hops / 45ms TTL...
Abort
Hop
IP Address
Response
Machine Name
Loss
1 146.230.240.1 1ms None ----
2 146.230.128.57 3ms Sunbird.und.ac.za ----
3 ---- Timeout n/a ----
4 155.232.50.250 24ms und.int.uni.net.za ----
5 196.25.134.65 26ms None ----
6 196.43.9.4 33ms ctb-int-fe-0-0-0.saix.net ----
7 196.25.0.82 464ms london-core-s-8-1-0.saix.net ----
8 196.25.0.14 479ms ny-core-s-8-1-1.saix.net ----
9 64.86.90.133 482ms if-1-1-0.bb5.NewYork.Teleglobe.net ----
10 207.45.221.66 492ms if-3-0.core2.NewYork.teleglobe.net ----
11 207.45.223.2 491ms if-9-0.core1.Ashburn.Teleglobe.net ----
12 216.239.48.142 499ms core1-1-4.iad.net.google.com ----
13 216.239.47.126 490ms abni1-1-1.net.google.com ----
14 216.239.47.102 707ms abbi1-1-1.net.google.com ----
15 216.239.51.100 625ms www.google.com ----
About
www.google.de
Trace
Copy
Config
Done

# Routing Algorithms

- Two general algorithms
  - Distance-vector
  - Link-state
- Goal of both is
  - to route a packet from one point of a network to another point through intermediate routers without “looping”
- Primary difference between the two algorithm ➔ manner in which they collect and propagate routing information

# Routing Algorithms: Distance vector routing

- Maintain table giving best known distance to each destination and line to use.
- Best known distance can be:
  - Hops
  - Delay in ms
  - Total packets queue along path
- Regularly update tables to reflect cost changes
- Problems:
  - Converges to correct answer slowly
  - Reacts fast to good news but slow to bad news – count-to-infinity problem
- Example of algorithm: Bellman-Ford Algorithm

# Distance vector routing
# **R**outing **I**nformation **P**rotocol (1)

- Used in many routing protocols in practice
  - TCP/IP suites (most of them)
  - Novell IPX
  - Used in the earliest Internet routing protocols
- hop count as cost metric
  - 1 hop per link, Max cost of path limited to 15
  - Thus can only be used within systems that are 15 hops away
- Queue length used as cost metric in

# Distance vector routing
# **R**outing **I**nformation **P**rotocol (2)

- Routing tables update messages are broadcast  every 30 seconds  using RIP messages.
- Routers update routing tables accordingly
- The *routed* process in UNIX is RIP
  - Try *netstat –rn*  at a host in UNIX
  - Try *netstat* in Windows
- Enhanced versions take several cost metrics into account (delay,bandwidth, load,reliability, etc.)
  - EIGRP from CISCO.
- 15 hop limit makes RIP unsuitable for large networks

# Routing Algorithm: Link State Routing

- Global routing algorithm – aware of cost of each link in network

Each router does the following:

- Discover neighbours and learn their IP addresses
- Measure delay or cost to each neighbour
- Construct packet with all info learnt
- Send packet to all routers
- Compute shortest path to every other router
  - Use Dijkstra’s Algorithm (RFC 1583)

# Link State Routing
# **O**pen **S**hortest **P**ath **F**irst (1)

- Problem with RIP - line bandwidth not taken into account when choosing routes
  - A 28kbps line Vs 10Mbps line quite different
- OSPF is a link state routing algorithm
  - Uses flooding to spread state info
- Advantages of OSPF
  - Does not have the count-to-infinity problem
  - Cost metric not limited to 16
  - Metric can be as large as 65535
  - Can use a variety of metrics
  - Causes less network traffic, updates every 30 minutes

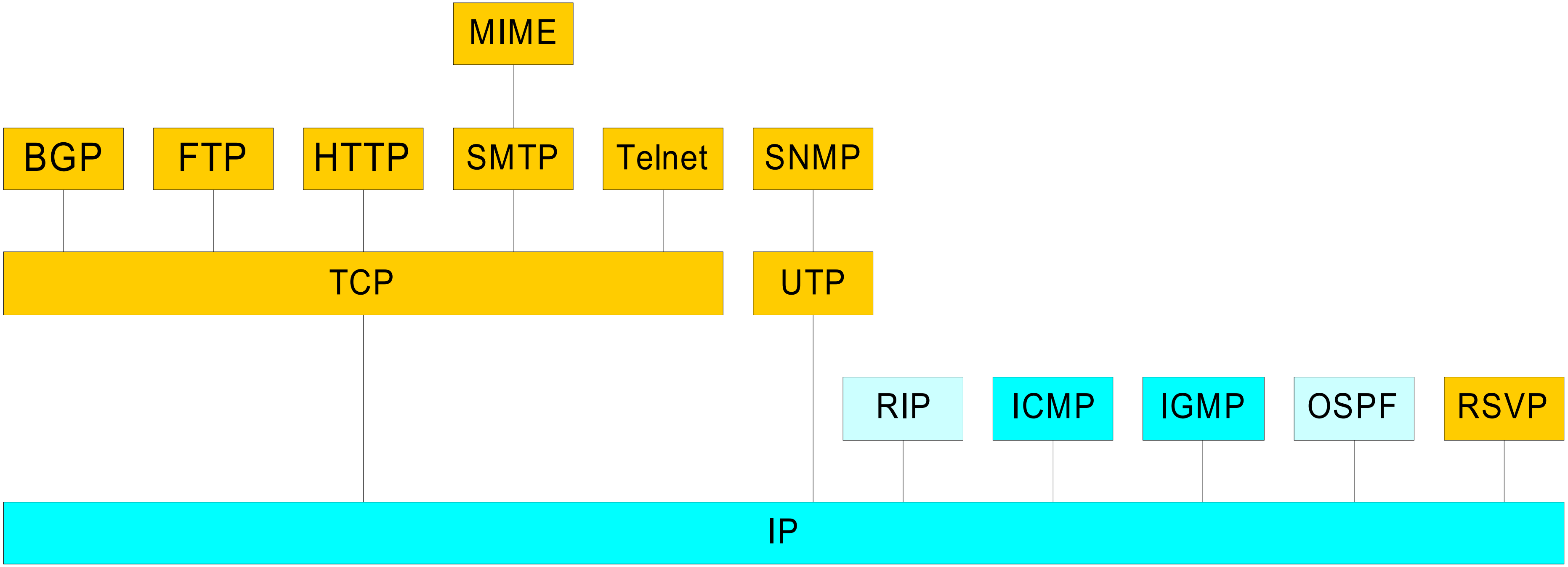
# Link State Routing
# **O**pen **S**hortest **P**ath **F**irst (2)

- Features of OSPF
  - Security
  - Multiple same-cost paths
  - Support for unicast and multicast
  - Different cost metrics for different TOS traffic
  - Support for hierarchy within a domain
- Now used widely including on the Internet

# Distance-Vector/ Link-State Algorithm: Difference

- distance vector
  - each node talks only to its directly connected neighbors
  - neighbor tells them everything it has learned (i.e., distance to all nodes)
- link state
  - each node talks to all other nodes
  - each node tells only what it knows for sure (i.e., only the state of its directly connected links)

# Internetworking Protocols



# Internet Protocol (IP)

- Functions of IP include:
  - Routing
  - Fragmentation and reassembly
  - Error control and flow control
  - Support transparent internetworking

# Routing

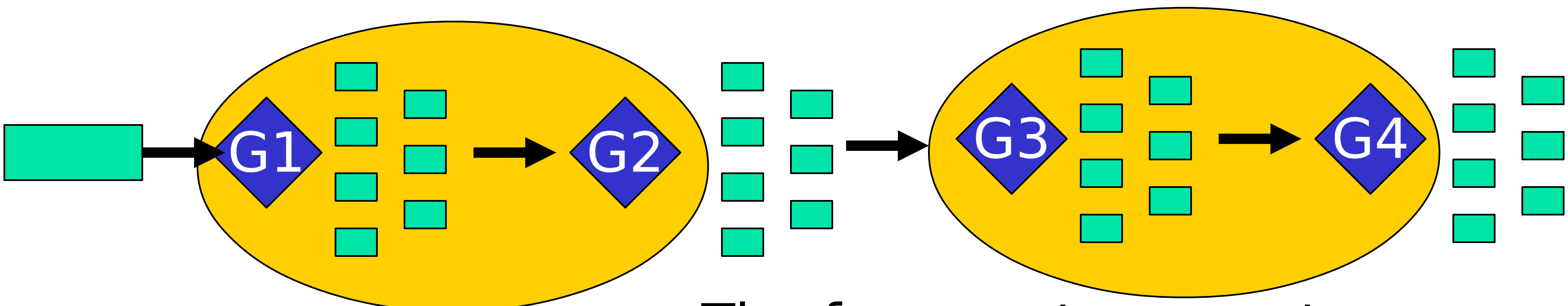
- End systems and routers maintain routing tables
  - Indicate next router to which datagram should be sent
  - Static
    - May contain alternative routes
  - Dynamic
    - Flexible response to congestion and errors
- Source routing
  - Source specifies route as sequential list of routers to be followed
  - Security
  - Priority
- Route recording

# Fragmentation and Re-assembly (1)

- Different packet sizes
  - Dictated by maximum transmission unit (MTU) of local LAN’s
    - Various along route: eg. 1500, 8174,etc.
  - Limited buffers etc.

# Fragmentation and Re-assembly (2)

- When to re-assemble
  - At destination
    - Results in packets getting smaller as data traverses internet

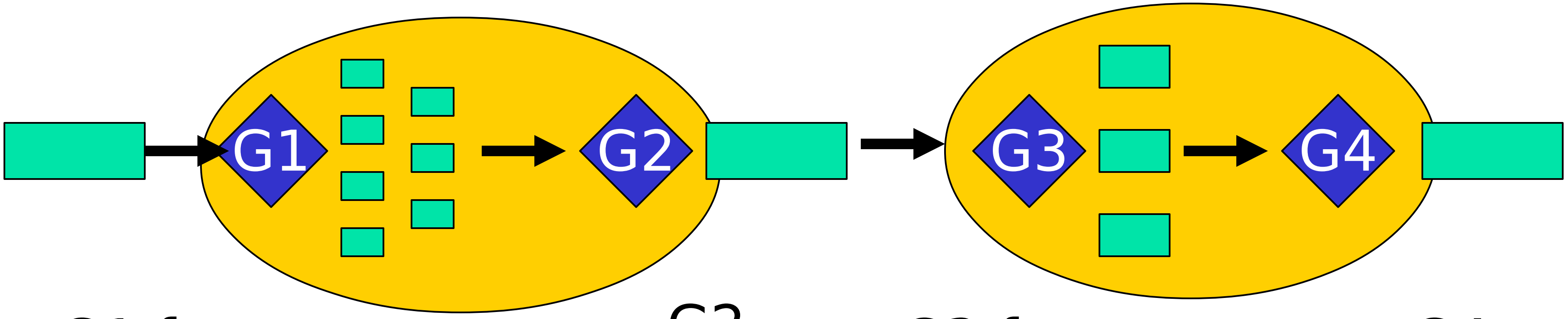



G1 fragments a large packet

The fragments are not reassembled until the final destination (host) is reached

# Fragmentation and Re-assembly (3)

- When to re-assemble
  - Intermediate re-assembly
    - Need large buffers at routers
    - Buffers may fill with fragments
    - All fragments must go through same router
      - Inhibits dynamic routing




G1 fragments a large packet

G2 reassembles

G3 fragments again

G4 reassembles

# IP Fragmentation (1)

- IP re-assembles at destination only
- Uses fields in header
  - Data Unit Identifier (ID)
    - Identifies end system originated datagram
      - Source and destination address
      - Protocol layer generating data (e.g. TCP)
      - Identification supplied by that layer
  - Data length
    - Length of user data in octets

# IP Fragmentation (2)

- Offset
  - Position of fragment of user data in original datagram
  - In multiples of 64 bits (8 octets)
- *More* flag
  - Indicates that this is not the last fragment
- D Flag : Do not fragment packet under any circumstances

# Fragmentation Example

- Need to send 300 bytes of data
  - D Flag not set
  - Maximum packet size is 128 bytes
  - Assume IP header in each IP datagram 20 bytes
- Fragmentation
  - All  packet fragments except the last must have a length divisible by 8
  - Result of fragmentation

| Fragment | Length(total) | offset | MF | ID |
|---|---|---|---|---|
| 1 | 124 | 0 | 1 | 2354 |
| 2 | 124 | 13 | 1 | 2354 |
| 3 | 112 | 26 | 0 | 2354 |

# Dealing with Failure

- Re-assembly may fail if some fragments get lost
- Need to detect failure
- Re-assembly time out
  - Assigned to first fragment to arrive
  - If timeout expires before all fragments arrive, discard partial data
- Use packet lifetime (time to live in IP)
  - If time to live runs out, kill partial data

# Error Control

- Not guaranteed delivery
- Router should attempt to inform source if packet discarded
  - For time to live expiring
  - Congestion
  - FCS error – notification may not possible
- Source may modify transmission strategy
- May inform high layer protocol
- Datagram identification needed
- (See ICMP later)

# Flow Control

- Allows routers and/or stations to limit rate of incoming data
- Limited in connectionless systems
- Send flow control packets
  - Requesting reduced flow
- ICMP can be used

# IP Protocol

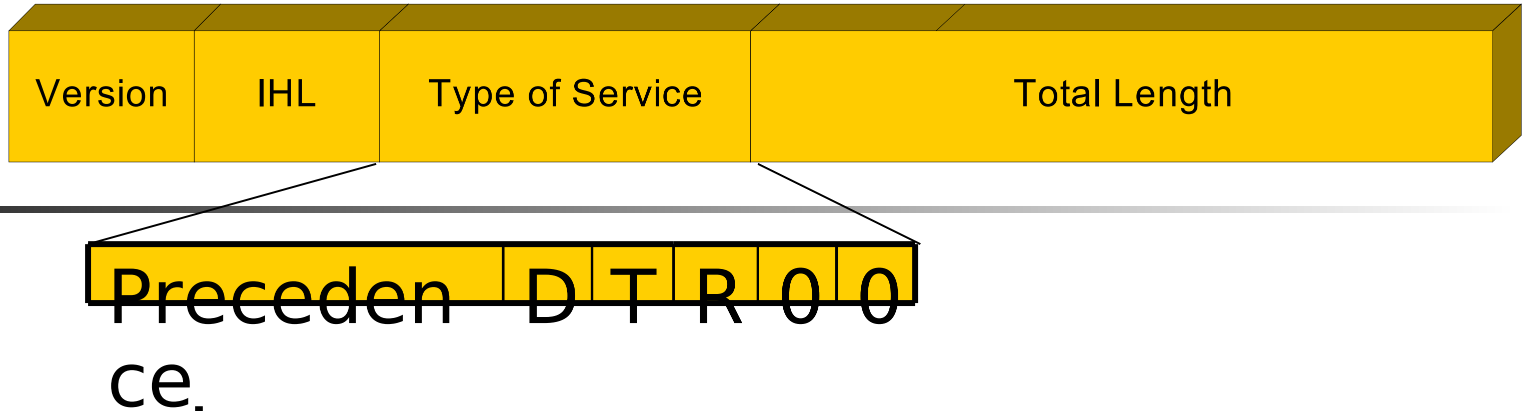
<table>
<tr><th>Bit: 0</th><th>4</th><th>8</th><th>16</th><th>19</th><th>31</th></tr>
<tr><td>Version</td><td>IHL</td><td>Type of Service</td><td colspan="3">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">Time to Live</td><td>Protocol</td><td colspan="3">Header Checksum</td></tr>
<tr><td colspan="6">Source Address</td></tr>
<tr><td colspan="6">Destination Address</td></tr>
<tr><td colspan="6">Option + Padding</td></tr>
</table>

20 Octets

| Version | IHL | Type of Service | Total Length |
|---|---|---|---|

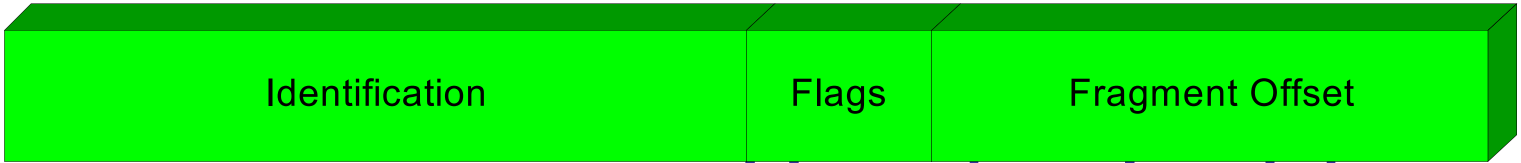
# Header Fields (1)

- Version
  - Currently 4
  - IP v6 - see later
- Internet header length
  - In 32 bit words
  - Including options
  - Minimum is 20 octets
- Type of service
  - Specify treatment of data unit during transmission through networks
- Total length
  - Of datagram, in octets=header length + user data



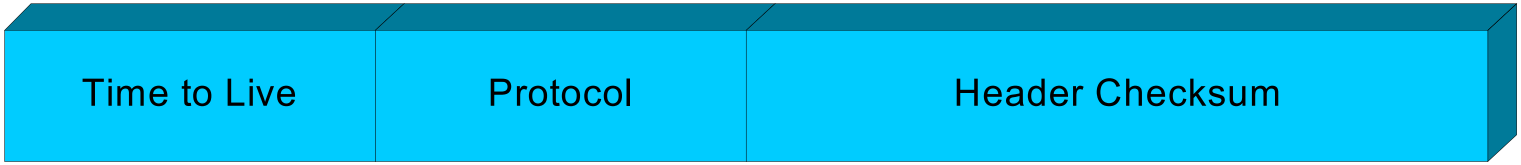

# Header field: Type of Service

- Precedence
  - 8 levels
- Delay
  - Normal or low
- Throughput
  - Normal or high
- Reliability
  - Normal or high
- unused fields



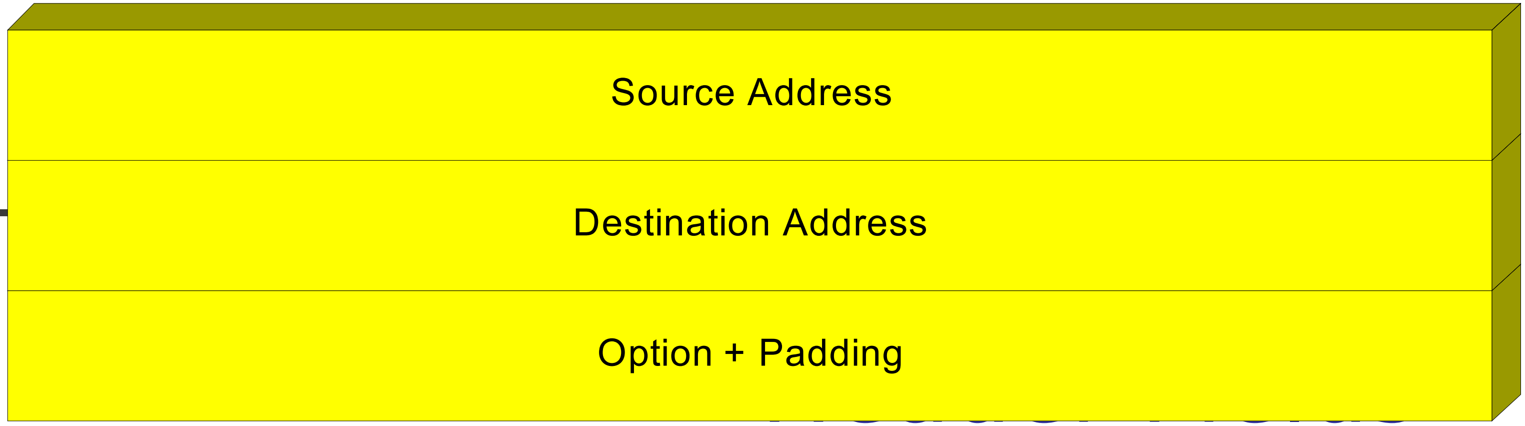

# Header Fields (2)

- Identification
  - Sequence number
  - Used with addresses and user protocol to identify datagram uniquely
  - Needed for re-assembly and error reporting
- Flags
  - More bit (M)
  - Don’t fragment (D)
- Fragmentation offset




# Header Fields (3)

- Time to live
- Protocol
  - Next higher layer to receive data field at destination
  - Can be TCP or UDP normally
- Header checksum
  - Re-verified and recomputed at each router
  - 16 bit ones complement sum of all 16 bit words in header
  - Set to zero during calculation




(4)

- Source address
- Destination address
- Options
- Padding
  - To fill to multiple of 32 bits long

# Header field: Options

- Source routing
  - *Strict source routing*: gives a complete path to be followed
  - *Loose source routing*: gives a list of routers not to be missed
- Route recording
  - Makes each router appends its IP address
- Timestamp
  - Makes each router appends its IP address and timestamp
- Security
  - Specifies how secret the datagram is
- Stream identification
  - Indicates the type of data in a datagram

# Header field: Data Field

- After options and padding field follows user data
- Carries user data from next layer up
- Integer multiple of 8 bits long (octet)
- Max length of datagram (header plus data) 65,536 octets

# IP – In Summary form

- Main attributes of IP are:
  - Connectionless protocols
  - Fragments packets if necessary
  - Addressing via 32-bit internet addresses(see next)
  - 8-bit transport protocol addresses
  - Maximum packet size of 65536 bytes
  - Only header checksum, no data checksum
  - Finite packet lifetime
  - ‘Best-effort’ delivery – no guarantees

# IPv4 Addresses

- 32 bit global internet address
- Partitioned into four groups of eight bits (called octets)
- Expressed in decimal form
  - Decimal point separates the octets.
  - Example: 204.163.25.37
- Each octet treated as independent unit
- Addresses are organized into one of 5 classes:
  - A,B,C,D or E
  - Classification determined by the value of the first four bits (bits 0 through 3)

# IP Addresses - Class A

- Network part and host part
- Class A
  - Start with binary 0
  - Network part is next 7 bits, host part  rest
  - 00000000 and 01111111 (127) reserved
  - Range 1.x.x.x to 126.x.x.x
  - Network mask (netmask)255.0.0.0
  - Allows few networks with many hosts
  - All allocated

# IP Addresses - Class B

- Start 10
- Second Octet also included in network address
- Range 128.x.x.x to 191.x.x.x
- Network mask 255.255.0.0
- $2^{14}$ = 16,384 class B addresses
- All allocated

# IP Addresses - Class C

- Start 110
- Range 192.x.x.x to 223.x.x.x
- Network mask 255.255.255.0
- Second and third octet also part of network address
- $2^{21}$ = 2,097,152 addresses
- Nearly all allocated
  - See IPv6

# IP Addresses – Class D & E

**Class D**

- Start 1110
- Second, third and fourth octet also part of network address (I.e. no host part)
- Range 224.x.x.x to 239.x.x.x
- $2^{28}$ = 268, 435, 456 addresses
- Used for multicasting

**Class E**

- Start 11110
- Second, third and fourth octets also part of the network address
- For future use

# IP Addresses – Summary

| IP class | Format | Purpose | High order bit(s) | Address range | No. bits Network/ host | Max. hosts | netmask |
|---|---|---|---|---|---|---|---|
| **A** | N.H.H.H | Few large organizations | 0 | 1.x.x.x to 126.x.x.x | 7 / 24 | $2^{24}$-2 | 255.0.0.0 |
| **B** | N.N.H.H | Medium-size organizations | 10 | 128.x.x.x to 191.x.x.x | 14 / 16 | $2^{16}$-2 | 255.255.0.0 |
| **C** | N.N.N.H | relatively small organizations | 110 | 192.x.x.x to 223.x.x.x | 21 / 8 | $2^{8}$-2 | 255.255.255.0 |
| **D** | N/A | Multicast groups | 1110 | 224.x.x.x to 239.x.x.x | Not for com. use | N/A | |
| **E** | N/A | Experimental | 1111 | 240.x.x.x to 255.x.x.x | N/A | N/A | |

# Subnets and subnet masks (1)

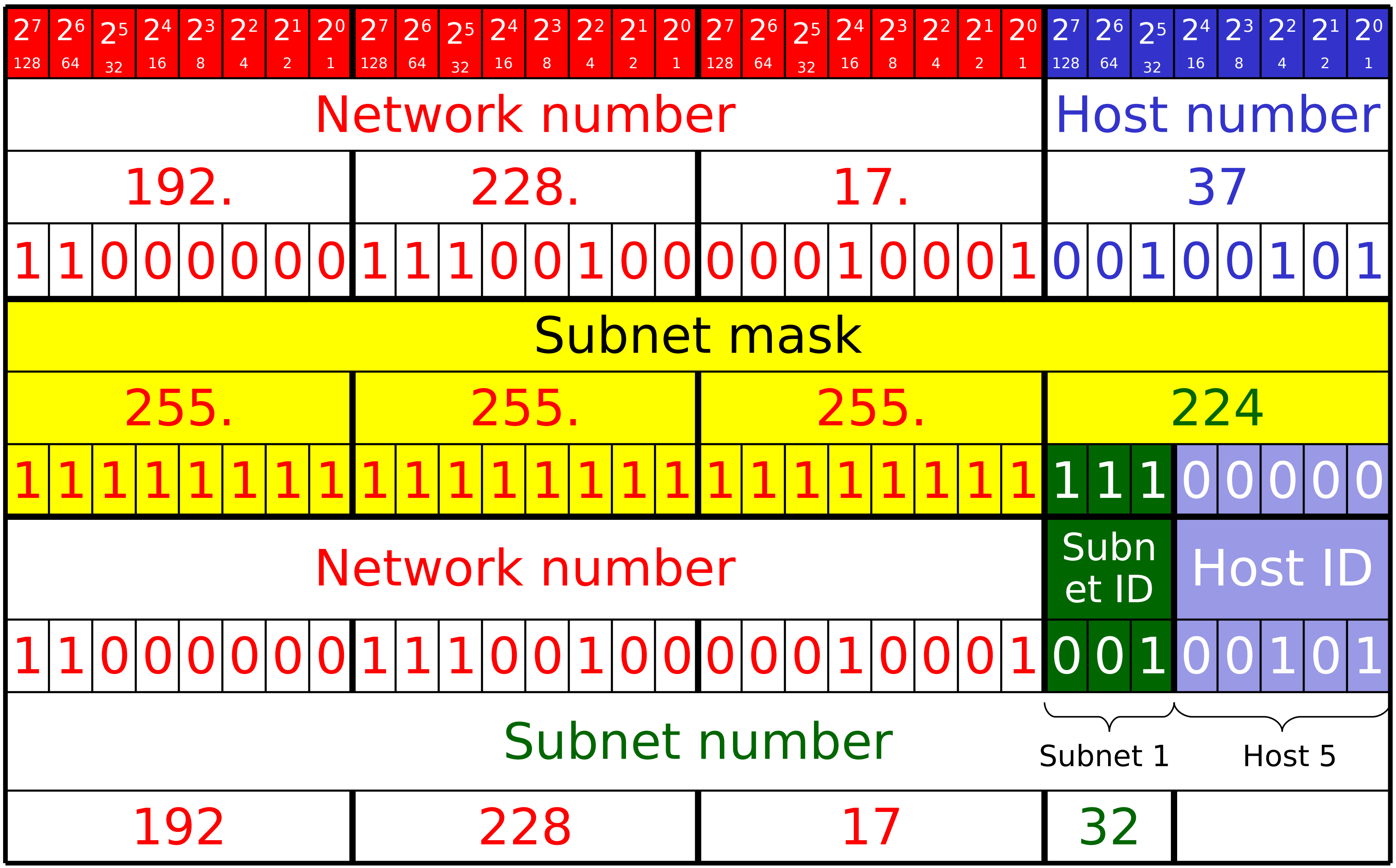
- Host portion of address partitioned into subnet number and host number
- Local routers route within subnetted network
- Subnet mask indicates which bits are subnet number and which are host number
- Each LAN assigned subnet number

# Subnets - Example

- 192.228.17.x is a class C address
- Subnet mask = 255.255.255.224
- Reserve 3 bits of fourth byte for subnet ID, up to 7 subnets
- 5 bits left for host ID, up to 30 hosts per subnet
- E.g.: 3 subnets: 001  00000 = 192.228.17.32, 010  00000 = 192.228.17.64, 011 00000 = 192.228.17.96

C

Host 5 on Subnet 1 (192.228.17.32)



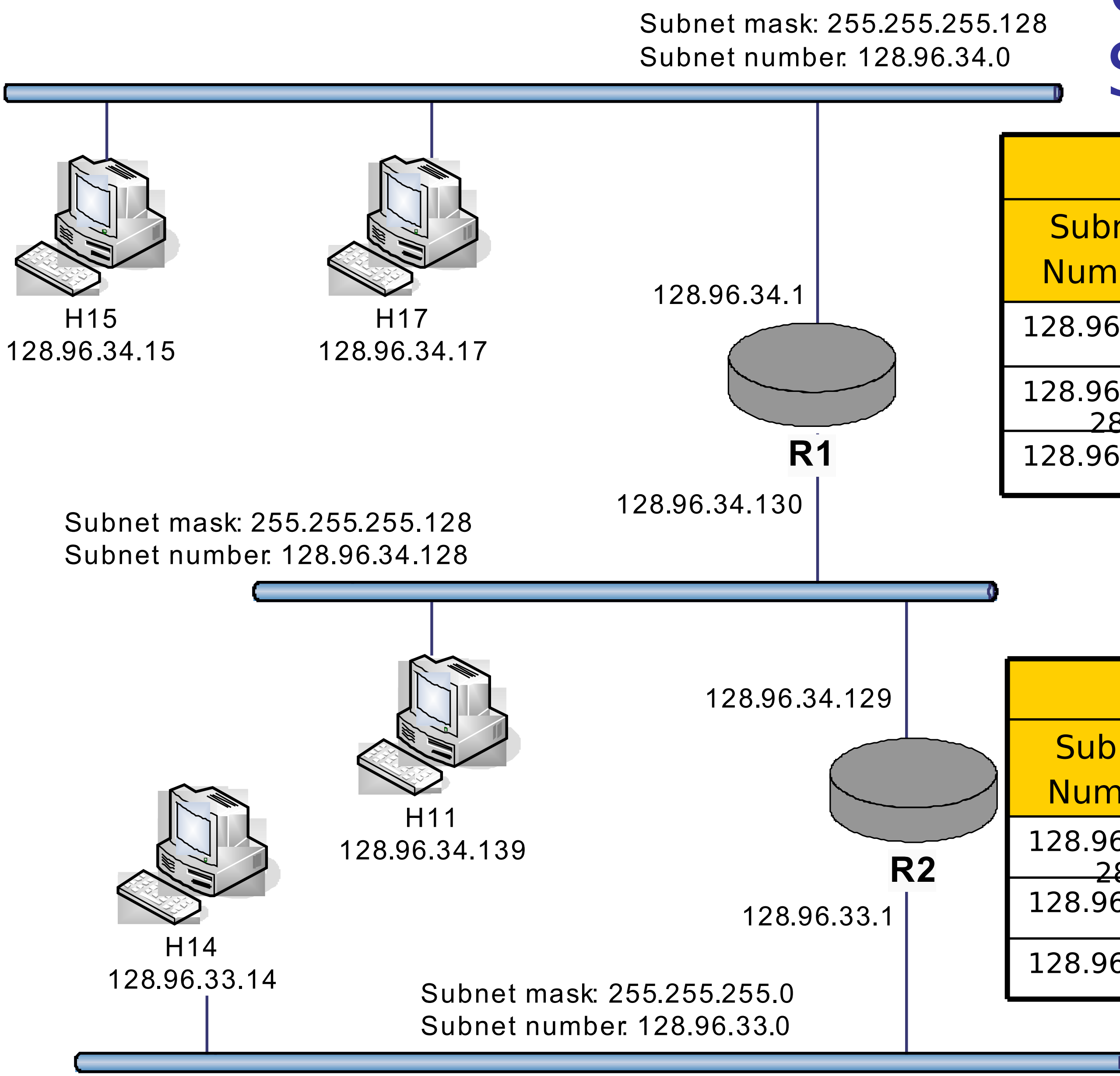
# Subnets and Subnet Masks (2)

- Improves address assignment efficiency
  - An entire class C or class B address is not wasted every time we add a new network
- From distance a complex collection of physical networks can be made to look like a single network
  - Amount of information that routers need to store to deliver datagrams to those networks can be reduced

# Routers and IP addressing

- IP address depends on network address
- IP address specifies an interface, or network attachment point, not a computer
- What about a router that is connected to two networks?
- Router has multiple IP addresses – one for each interface

# Routing Using Subnets




| Routing table **R1** | | |
|---|---|---|
| Subnet Number | Subnet Mask | Next Hop |
| 128.96.34.0 | 255.255.255.128 | Interface 0 |
| 128.96.34.128 | 255.255.255.128 | Interface 1 |
| 128.96.33.0 | 255.255.255.0 | R2 |

| Routing table **R2** | | |
|---|---|---|
| Subnet Number | Subnet Mask | Next Hop |
| 128.96.34.128 | 255.255.255.128 | Interface 0 |
| 128.96.33.0 | 255.255.255.0 | Interface 1 |
| 128.96.34.0 | 255.255.255.128 | R1 |

Modified after Peterson and Davie (2003): Computer Networks, A Systems Approach

# Datagram forwarding algorithm

D = destination IP address

For each forwarding table entry {SubnetNumber, SubnetMask, NextHop}

    D1 = SubnetMask & D

        if D1 = SubnetNumber

          if NextHop is an interface

            deliver datagram directly to destination

        Else

            Deliver datagram to NextHop {a router}

# ICMP

- Internet Control Message Protocol (RFC 792)
- Transfer of (control) messages from routers and hosts to hosts
- Feedback about problems
  - e.g. time to live expired
  - Packet congestion at gateway
  - Inoperative nodes and gateways
  - Routing redirect
  - Echo requests to test connectivity- *ping*
- Encapsulated in IP datagram (data portion)
  - Not reliable

# ARP and RARP

- ARP returns the Ethernet address of a host given the IP addresses.
  - Try *arp –a*
- RARP returns the IP addresses of a host given the Ethernet address.
  - Intended for diskless workstations
  - Need a  RARP server
- Only meaningful within  a given LAN environment.

# IP v6 - Version Number

- IP v 1-3 defined and replaced
- IP v4 - current version
- IP v5 - streams protocol
- IP v6 - replacement for IP v4
  - During development it was called IPng
  - Next Generation

# Why Change IP?

- Address space exhaustion
  - Two level addressing (network and host) wastes space though convenient
  - Network addresses used even if not connected to Internet
  - Growth of networks and the Internet
  - Extended use of TCP/IP
  - Single address per host
- Requirements for new types of service

# IPv6 RFCs

- 1752 - Recommendations for the IP Next Generation Protocol
- 2460 - Overall specification
- 2373 - addressing structure
- others

# PIv6 Enhancements (1)

- Expanded address space
  - 16 bytes =128 bit
- Improved option mechanism
  - Separate optional headers between IPv6 header and transport layer header
  - Most are not examined by intermediate routes
    - Improved speed and simplified router processing
    - Easier to extend options
- Address auto configuration
  - Dynamic assignment of addresses

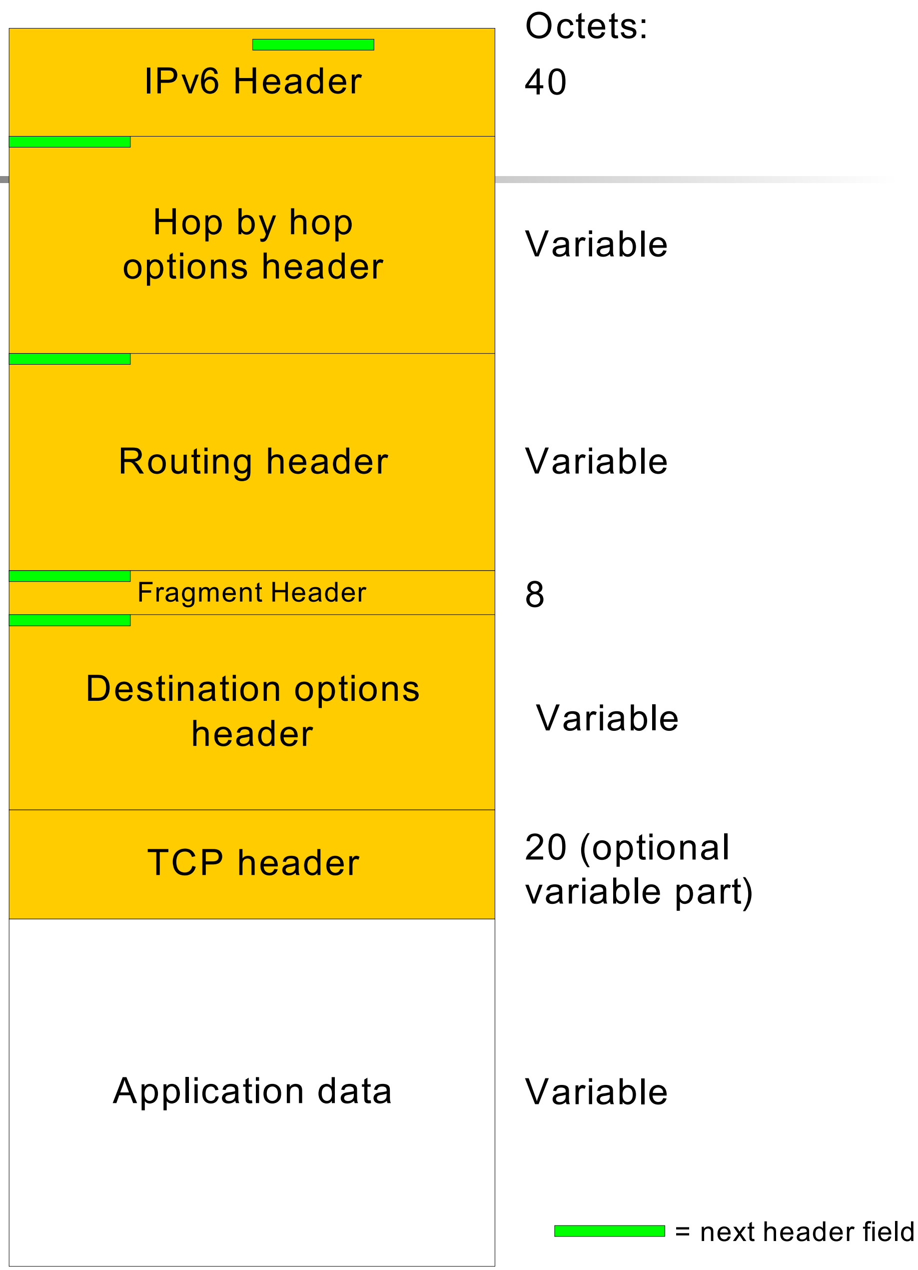
# IPv6 Enhancements (2)

- Increased addressing flexibility
  - Anycast - delivered to one of a set of nodes
  - Improved scalability of multicast addresses
- Support for resource allocation
  - Replaces type of service
  - Labeling of packets to particular traffic flow
  - Allows special handling; priorities up to 8
  - e.g. real time video, real time audio can be flows
  - Can have priorities within a flow (e.g ICMP datagram)
- Enhanced security mechanisms
  - Authentication and encryption an integral part

# IPv6 Enhancements (3)

- Fragmentation/Reassembly
  - Only done at source and destination, not at intermediate routers
  - Source must perform path discovery to find smallest MTU of intermediate networks, and then use that MTU, enables faster processing
- Checksum done away with
  - Already done at link and transport layers, considered enough
  - Faster processing
- M and F bits now in Fragment extension header

# Structure

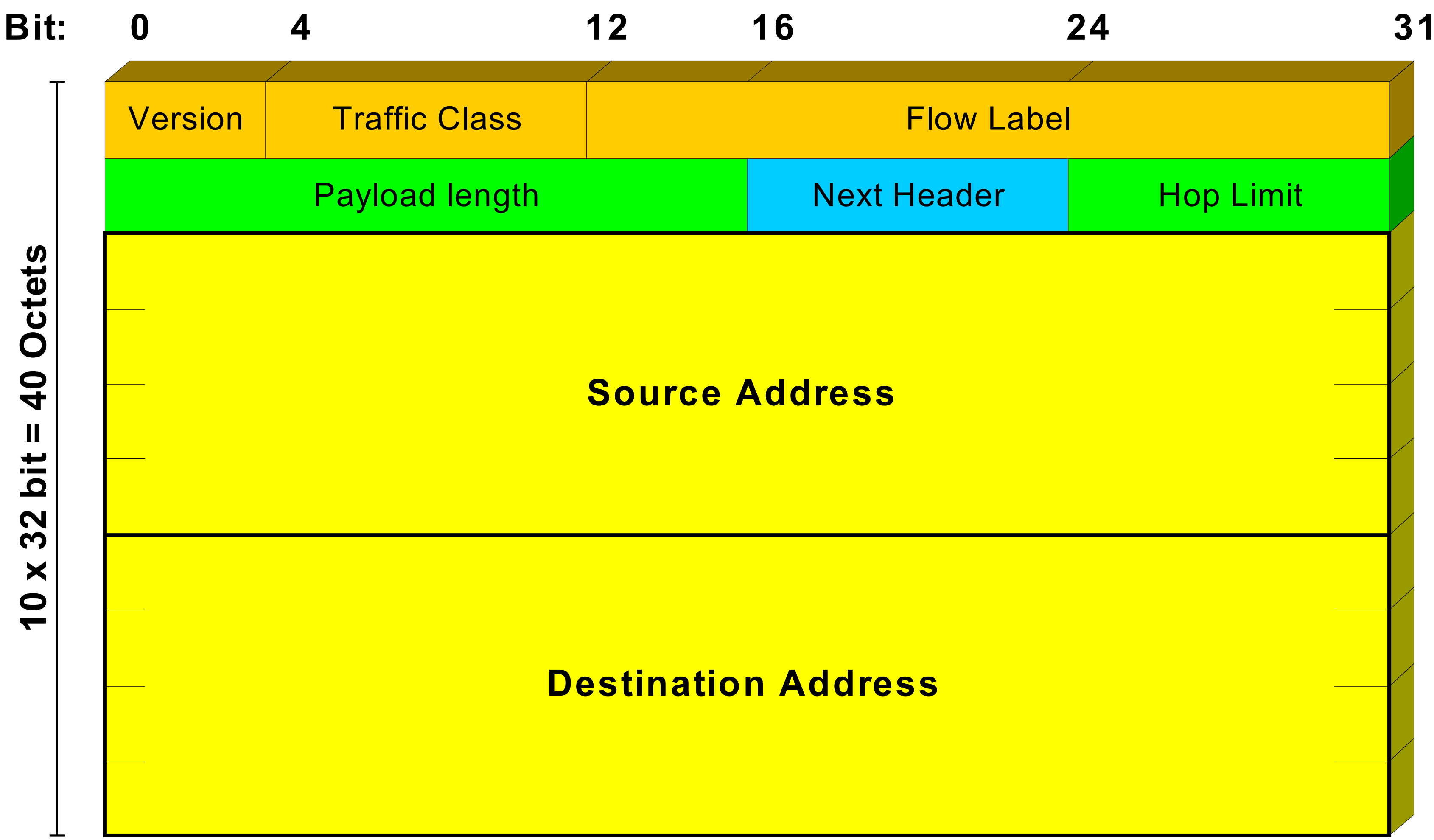


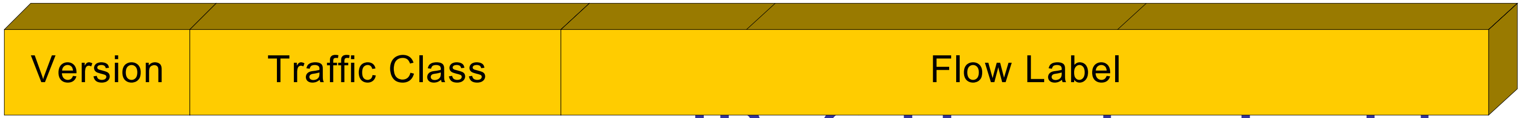
Octets:
IPv6 Header
40
Hop by hop options header
Variable
Routing header
Variable
Fragment Header
8
Destination options header
Variable
TCP header
20 (optional variable part)
Application data
Variable
= next header field

# Extension Headers

- Hop-by-Hop Options
  - Require processing at each router
- Routing
  - Similar to v4 source routing
- Fragment
- Authentication
- Encapsulating security payload
- Destination options
  - For destination node

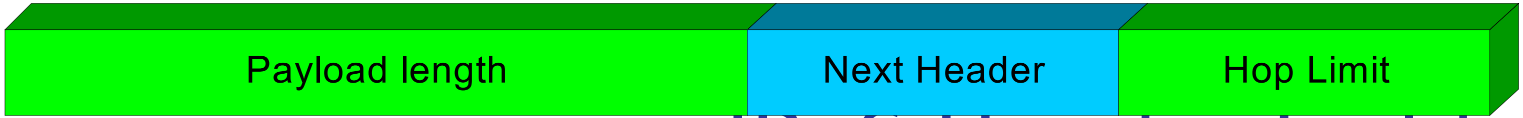
# IP v6 Header


Bit: 0 4 12 16 24 31
Version
Traffic Class
Flow Label
Payload length
Next Header
Hop Limit
Source Address
Destination Address
10 x 32 bit = 40 Octets




# IPv6 Header Fields (1)

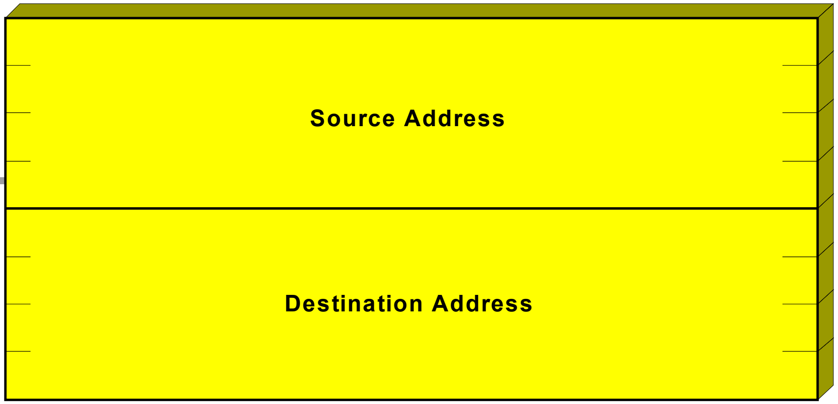
- Version
  - 6
- Traffic Class
  - Classes or priorities of packet
  - Still under development
  - See RFC 2460
- Flow Label
  - Used by hosts requesting special handling




# IPv6 Header Fields (2)

- Payload length
  - Includes all extension headers plus user data
- Next Header
  - Identifies type of header
    - Extension or next layer up
- Hop limit
  - Similar to time-to-live

# IPv6 Addresses




- Source Address
- Destination Address
  - 128 bits long
    - Expressed in hexadecimal form
    - Written as eight 16-bit integers
    - Uses colons as delimiter (2A01:0000:0000:0000:12FB:071C:04DE:689E)
  - Assigned to interface
  - Single interface may have multiple unicast addresses
  - Three types of address

# Types of address

- Unicast
  - Single interface
- Anycast
  - Set of interfaces (typically different nodes)
  - Delivered to any one interface
  - the “nearest”
  - E.g. send HTTP GET to any one of the mirror sites containing a document
- Multicast
  - Set of interfaces
  - Delivered to all interfaces identified

# Multicasting

- Addresses that refer to group of hosts on one or more networks
- Uses
  - Multimedia “broadcast” –live lecture to many locations
  - Teleconferencing
  - Database – software upgrades to a number of sites
    - Data feeds –stock quotes
  - Distributed computing
  - Real time workgroups
  - Interactive gaming, etc

# Multicast approaches

- Multiple unicast
  - Sender sets up a unicast connection to each member of the intended recipients
- Explicit multicast support at the network layer – true multicast
  - This mechanism supported on the internet through address indirection

# True Multicast

- Determine least cost path to each network that has host in group
  - Gives spanning tree configuration containing networks with group members
- Transmit single packet along spanning tree
- Routers replicate packets at branch points of spanning tree

# Requirements for Multicasting (1)

- Router may have to forward more than one copy of packet
- Convention needed to identify multicast addresses
  - IPv4 - Class D - start 1110
  - IPv6 - 8 bit prefix, all 1, 4 bit flags field, 4 bit scope field, 112 bit group identifier
- Nodes must translate between IP multicast addresses and list of networks containing group members
- Router must translate between IP multicast address and network multicast address

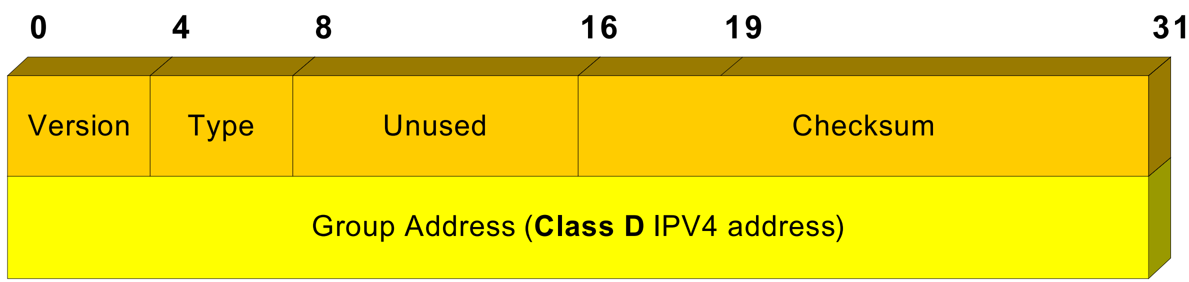
# Requirements for Multicasting (2)

- Mechanism required for hosts to join and leave multicast group
- Routers must exchange info
  - Which networks include members of given group
  - Sufficient info to work out shortest path to each network
  - Routing algorithm to work out shortest path
    - Multicast routing algorithms (MDVRP, MOSPF)
  - Routers must determine routing paths based on source and destination addresses

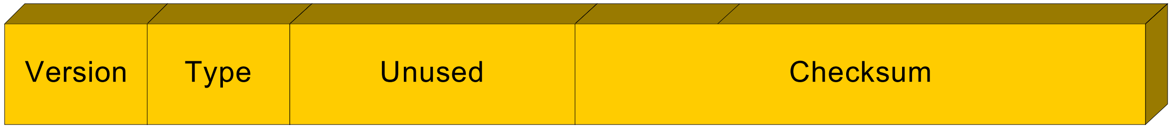
# IGMP

- Internet Group Management Protocol
- RFC 1112
- Host and router exchange of multicast group info
- Use broadcast LAN to transfer info among multiple hosts and routers

# IGMP Format


0
4
8
16
19
31
Version
Type
Unused
Checksum
Group Address (**Class D** IPV4 address)

# IGMP Fields




- Version
  - 1
- Type
  - 1 - query sent by router
  - 0 - report sent by host
- Checksum
- Group address
  - Zero in request message
  - Valid group address in report message

# IGMP Operation

- To join a group, hosts sends report message
  - Group address of group to join
  - In IP datagram to same multicast destination address
  - All hosts in group receive message
  - Routers listen to all multicast addresses to hear all reports
- Routers periodically issue request message
  - Sent to all-hosts multicast address
  - Host that want to stay in groups must read all-hosts messages and respond with report for each group it is in

# Group Membership in IPv6

- Function of IGMP included in ICMP v6
- New group membership termination message to allow host to leave group