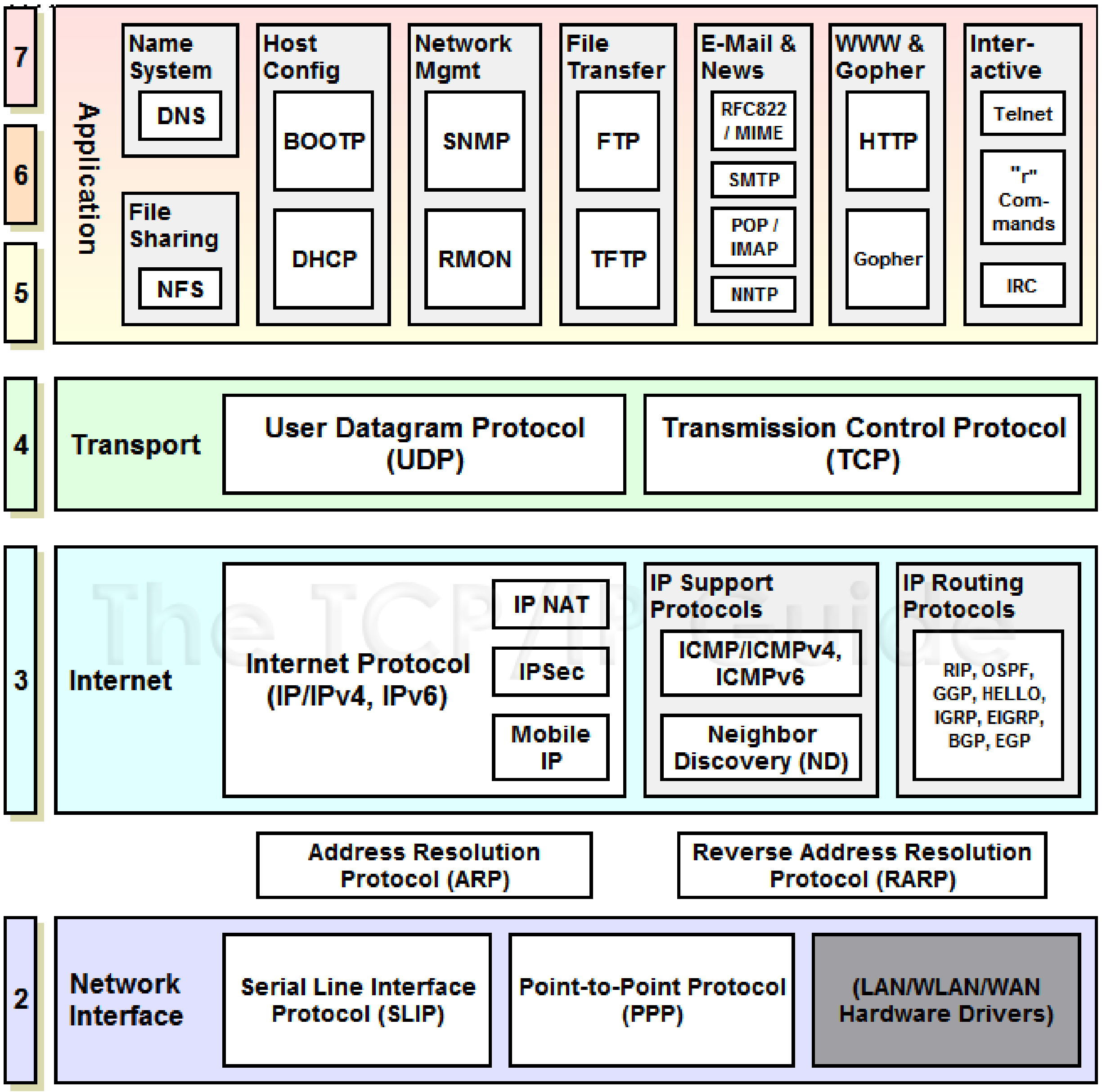
Chapter 04:
TCP/IP Protocol suite

http://www.tcpipguide.com

TCP/IP Protocol suite



7 6 5 Application

Name System: DNS

File Sharing: NFS

Host Config: BOOTP, DHCP

Network Mgmt: SNMP, RMON

File Transfer: FTP, TFTP

E-Mail & News: RFC822 / MIME, SMTP, POP / IMAP, NNTP

WWW & Gopher: HTTP, Gopher

Inter-active: Telnet, "r" Com-mands, IRC

4 Transport

User Datagram Protocol (UDP)

Transmission Control Protocol (TCP)

3 Internet

Internet Protocol (IP/IPv4, IPv6): IP NAT, IPSec, Mobile IP

IP Support Protocols: ICMP/ICMPv4, ICMPv6; Neighbor Discovery (ND)

IP Routing Protocols: RIP, OSPF, GGP, HELLO, IGRP, EIGRP, BGP, EGP

Address Resolution Protocol (ARP)

Reverse Address Resolution Protocol (RARP)

2 Network Interface

Serial Line Interface Protocol (SLIP)

Point-to-Point Protocol (PPP)

(LAN/WLAN/WAN Hardware Drivers)

TCP/IP protocol suite & RFC

RFC	Protocol	Description
Link layer		
1055	SLIP	**Serial Line IP**
1661	PPP	Peer to peer protocol
Internet layer		
826	ARP (Address Resolution Protocol)	Get IP address from MAC
903	RARP (Reverse ARP)	Get MAC address from IP
791, 950, 919, 992	IP	Internet Protocol
792	ICMP	Internet Control Message Protocol

Họ giao thức TCP/IP

RFC		
Transport layer		
793	TCP	Transmission Control Protocol
768	UDP	User Datagram Protocol
Application layer		
1034,1035	DNS	Domain Name Service
959	FTP	File Transfer Protocol
2131	DHCP	Dynamic Host Configuration Protocol
821	SMTP	Simple Mail Transfer Protocol
1157	SNMP	Simple Network Management Protocol
1939	POP-3	Post Office Protocol, version 3.
1945, 2068	HTTP	Web

Main protocols at lower layers

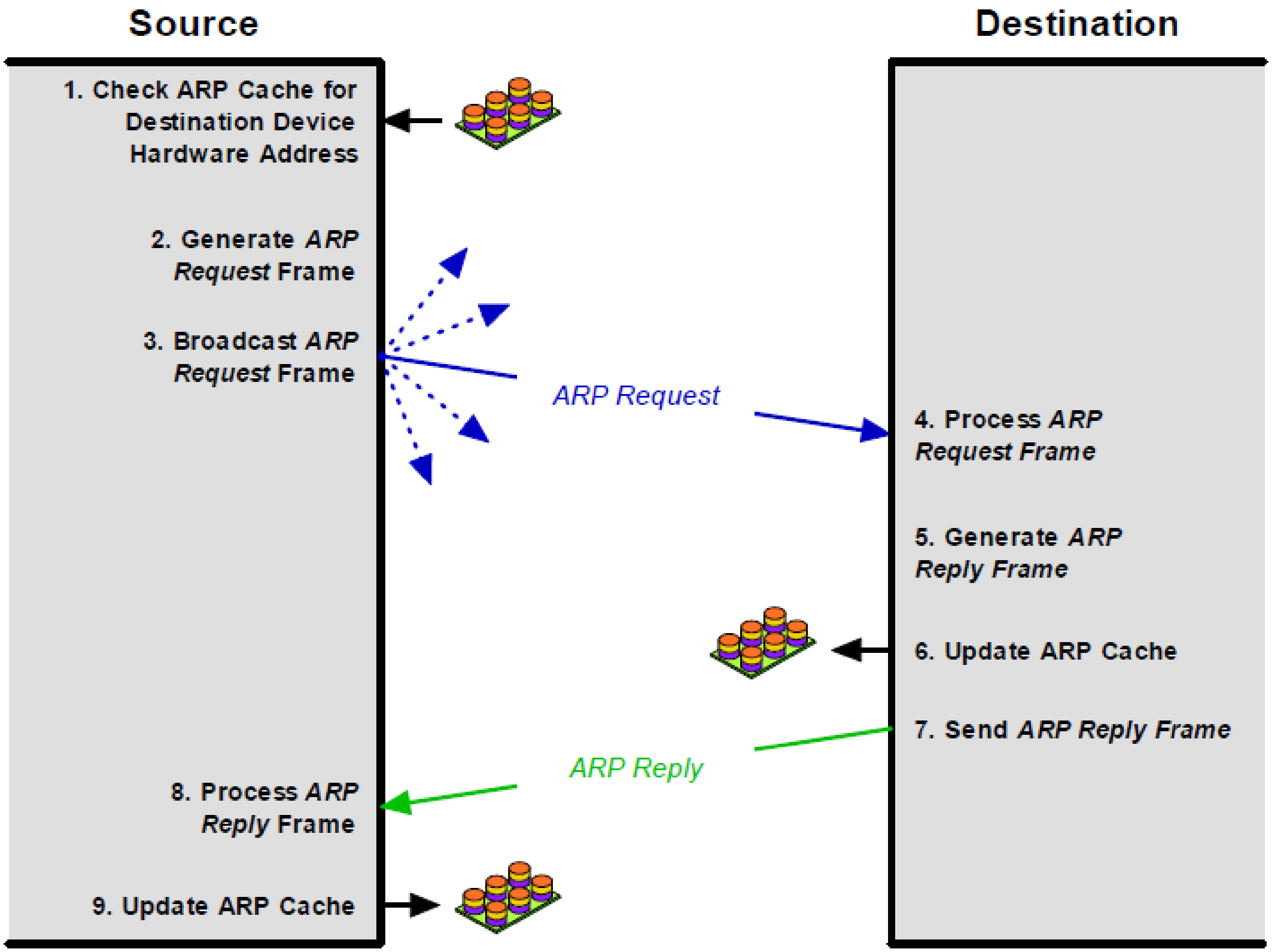
- ARP
- RARP
- IP
- ICMP
- TCP
- UDP

ARP & RARP

ARP

- ARP is a relatively simple request/reply protocol.
- The source device broadcasts an *ARP Request* looking for a particular device based on its IP address.
- That device responds with its hardware address in an *ARP Reply* message.

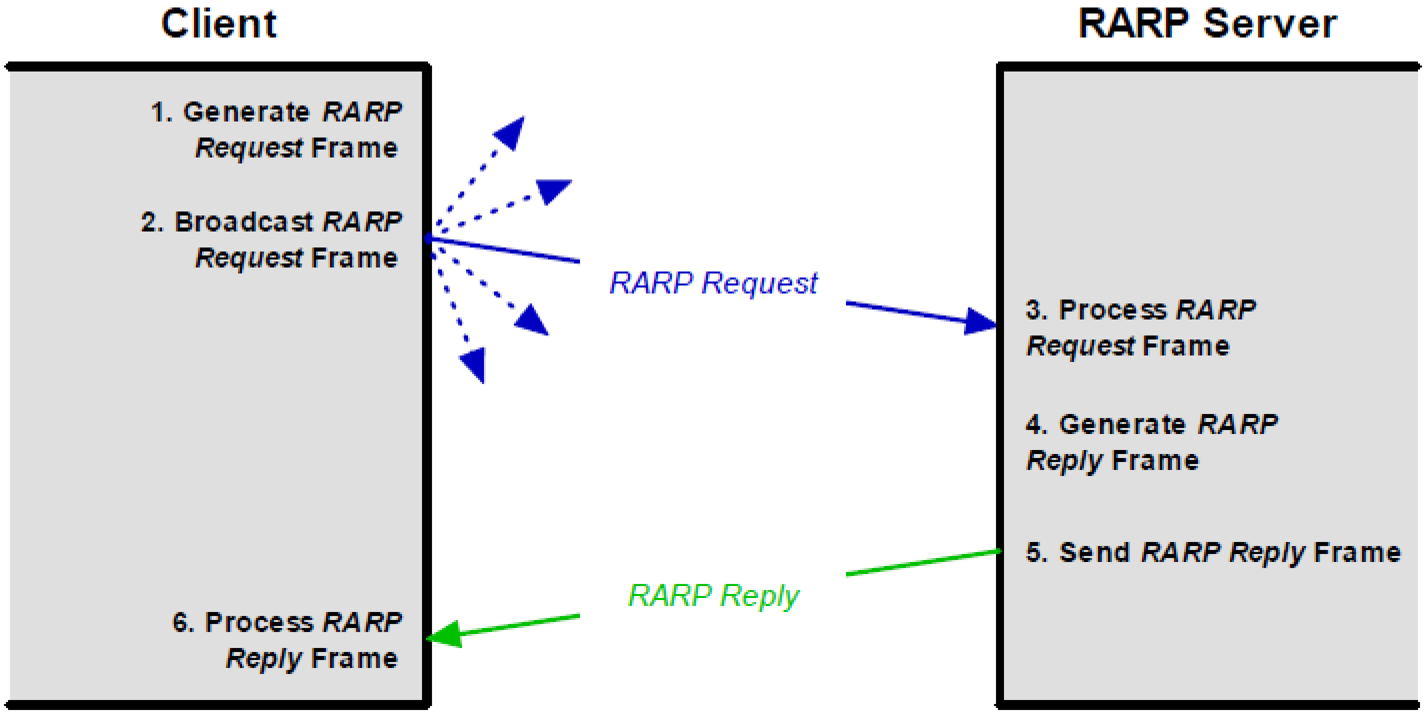
ARP Operation


Source
Destination
1. Check ARP Cache for Destination Device Hardware Address
2. Generate *ARP Request* Frame
3. Broadcast *ARP Request* Frame
ARP Request
4. Process *ARP Request Frame*
5. Generate *ARP Reply Frame*
6. Update ARP Cache
7. Send *ARP Reply Frame*
ARP Reply
8. Process *ARP Reply* Frame
9. Update ARP Cache

RARP

- The *Reverse Address Resolution Protocol (RARP)* is the earliest and simplest protocol designed to allow a device to obtain an IP address for use on a TCP/IP network.
- It is based directly on ARP and works in basically the same way, but in reverse: a device sends a request containing its hardware address and a device set up as an RARP server responds back with the device’s assigned IP address.

RARP Operation



Client

RARP Server

1. Generate *RARP Request* Frame

2. Broadcast *RARP Request* Frame

RARP Request

3. Process *RARP Request* Frame

4. Generate *RARP Reply* Frame

5. Send *RARP Reply* Frame

RARP Reply

6. Process *RARP Reply* Frame

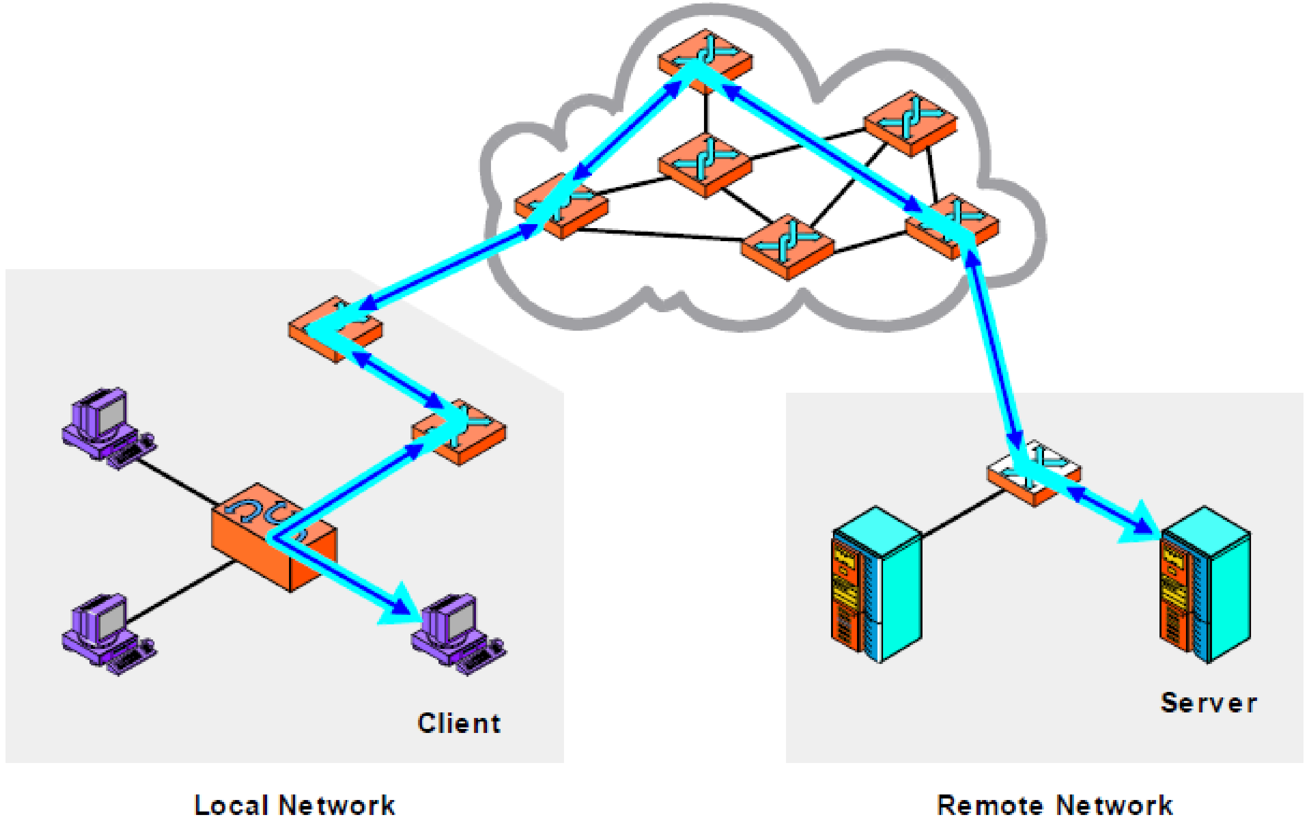
Limitation of RARP

- **Low-Level Hardware Orientation:** RARP works using hardware broadcasts. This means an RARP server is needed on ***every*** network segment.
- **Manual Assignment:** RARP allows hosts to configure themselves automatically, but the RARP server must still be set up with a manual table of bindings between hardware and IP addresses. These must be maintained for each server, which is again a lot of work on an administrator.
- **Limited Information:** RARP only provides a host with its IP address. It cannot provide other needed information such as, for example, a subnet mask or default gateway.

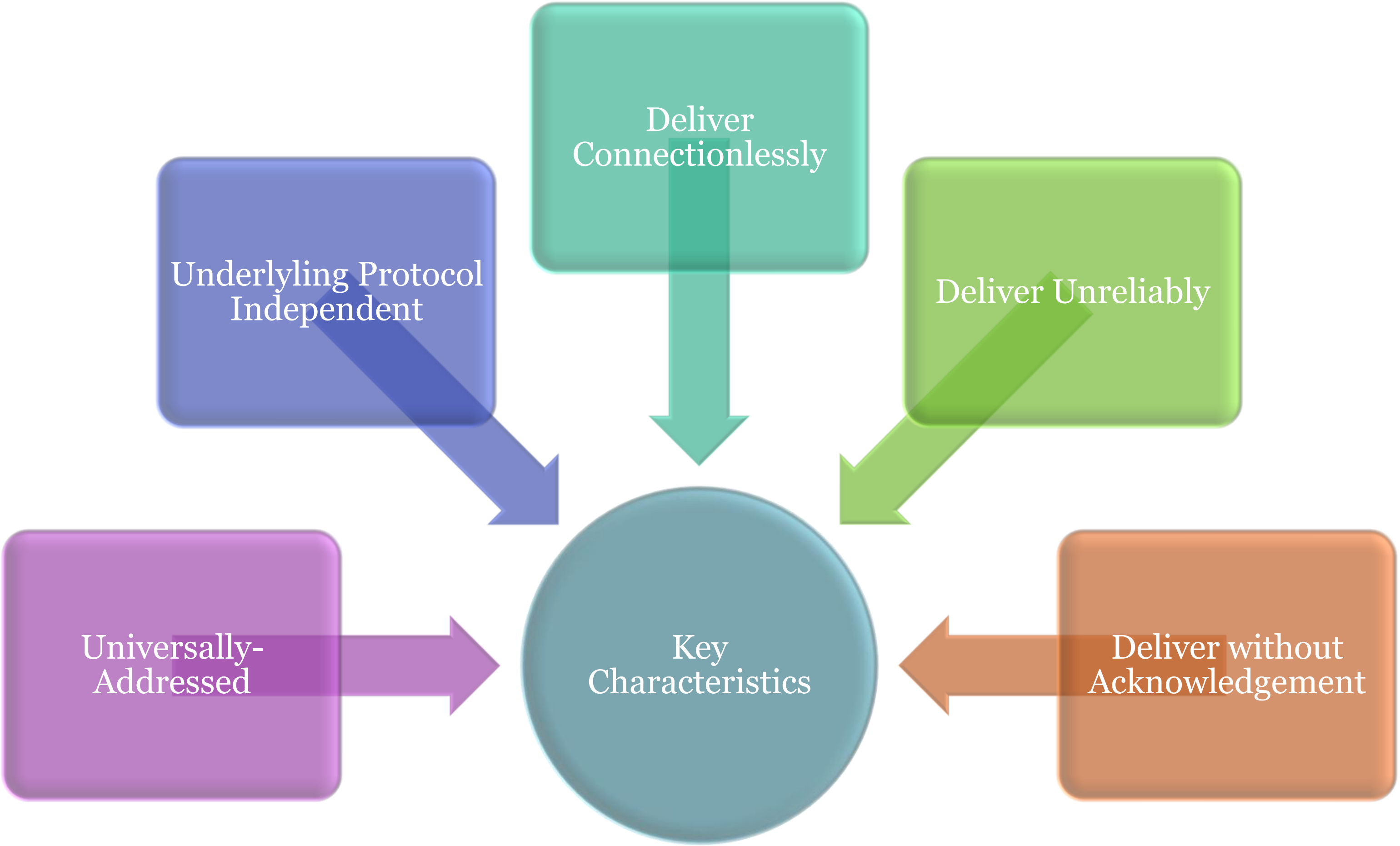
IP Overview & Key Operational characteristics

While the Internet Protocol has many functions and characteristics, it can be focused one primary purpose: the delivery of datagrams across an internetwork of connected networks

Delivery of datagrams



IP Key Characteristics



Deliver Connectionlessly

Underlyling Protocol Independent

Deliver Unreliably

Universally-Addressed

Key Characteristics

Deliver without Acknowledgement

IP Functions

- Addressing
- Data Encapsulation and Formatting/Packaging
- Fragmentation and Reassembly
- Routing / Indirect Delivery

IP versions & IP related Protocols

- IP Version : IPv4 (RFC 791) , IPv6
- Related protocols:
  - IP NAT (NAT)
  - IP Security (IPSec)
  - Mobile IP

IPv4

Even though the name seems to imply that it's the fourth iteration of the key Internet Protocol, version 4 of IP was the first that was widely used in modern TCP/IP.

IP Address Overview & Fundamentals

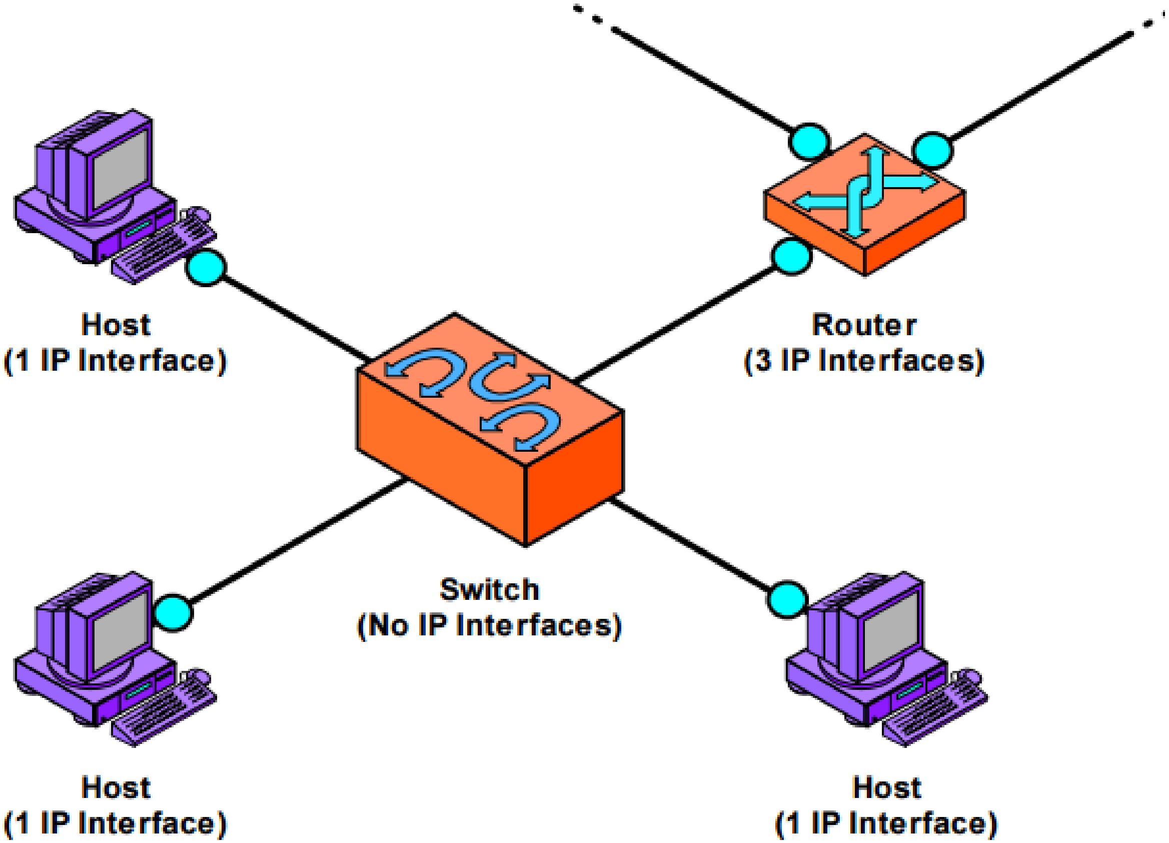
Facilitates the delivery of datagrams across an Internetwork

Functions:

- **Network Interface Identification**: the IP address provides unique identification of the interface between a device and the network.
- **Routing**: When the source and destination of an IP datagram are not on the same network, the datagram must be delivered "indirectly" using intermediate systems, a process called *routing*.

Questions

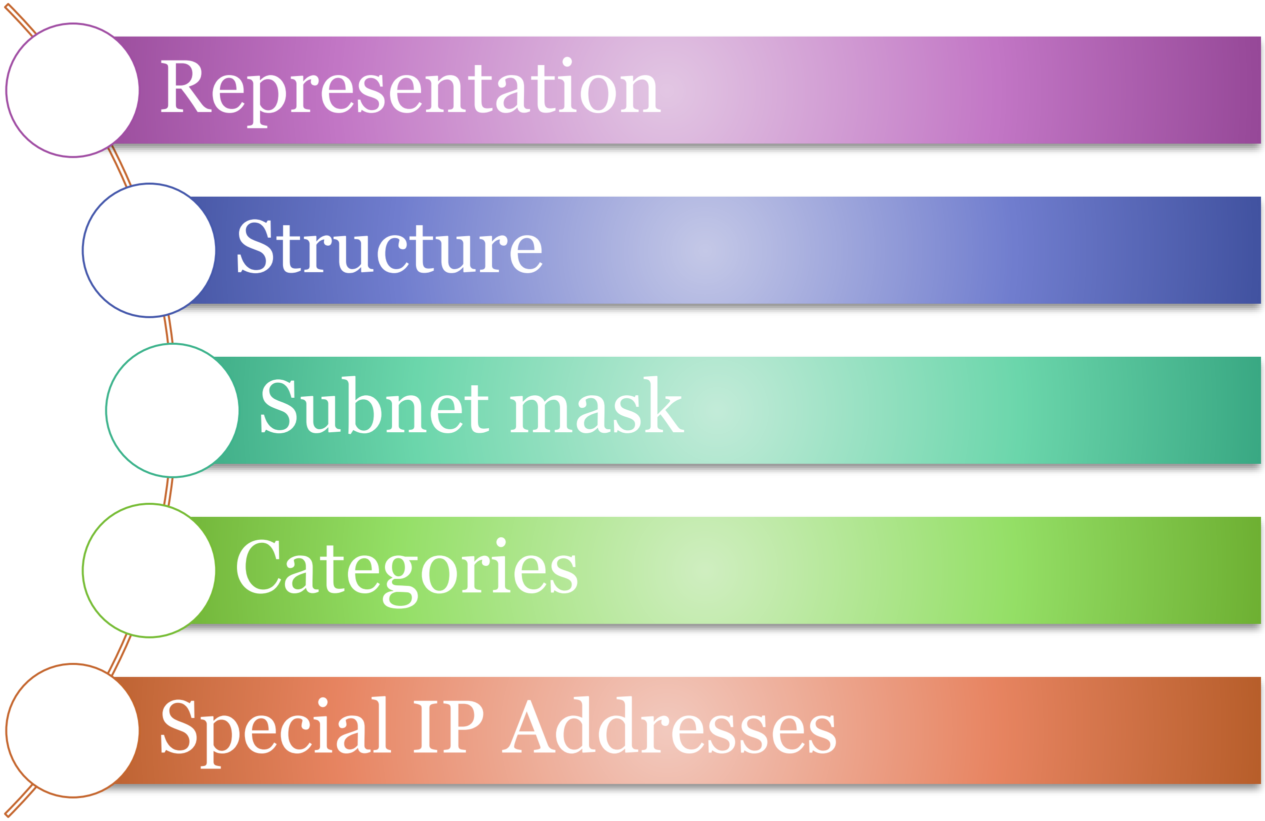
- Number of IP Addresses Per Device ?
- Which devices require an IP address ?



IP Address fundamentals

- Address Uniqueness
- Network-Specificity of IP Addresses
- Contrasting IP Addresses and Data Link Layer Addresses
- Private and Public IP Network Addresses
- IP Address Configuration

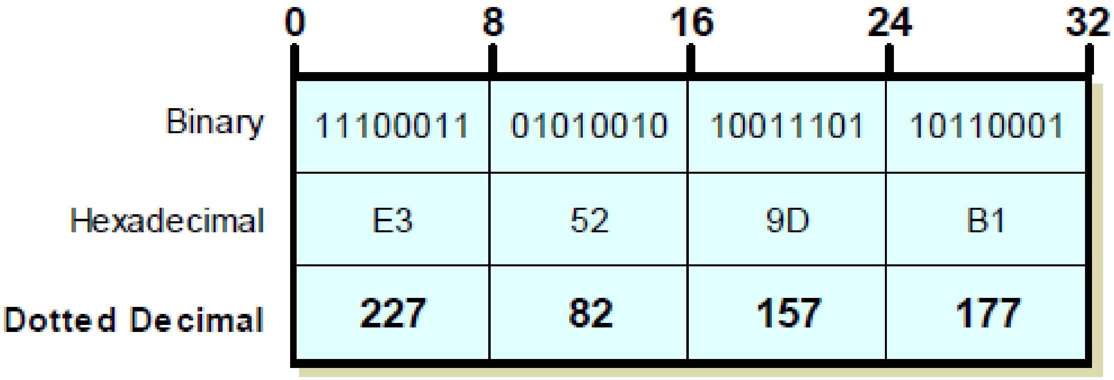
IP Address



- Representation
- Structure
- Subnet mask
- Categories
- Special IP Addresses

IP Address Representations

- 32 bits long = 4,294,967,296 addresses
- Representations:

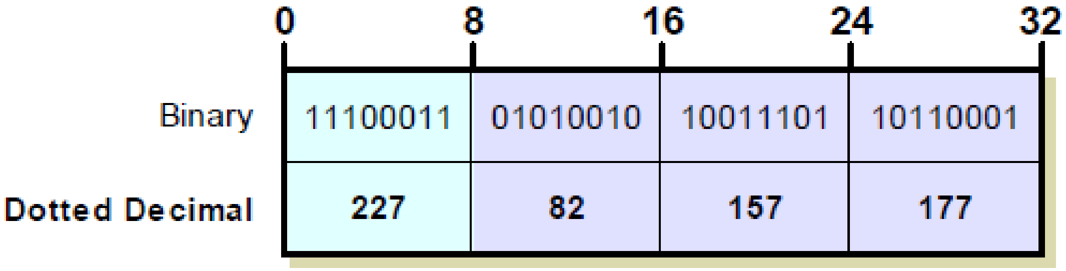


Binary	11100011	01010010	10011101	10110001
Hexadecimal	E3	52	9D	B1
Dotted Decimal	**227**	**82**	**157**	**177**

IP Address Structure

- **Network Identifier (Network ID):** A certain number of bits, starting from the left-most bit, is used to identify the network where the host or other network interface is located.
- **Host Identifier (Host ID):** The remainder of the bits are used to identify the host on the network.

The dividing point of the 32-bit address is not fixed, but rather, depends on a number of factors, and can occur in a variety of places, including in the middle of a dotted-decimal octet.



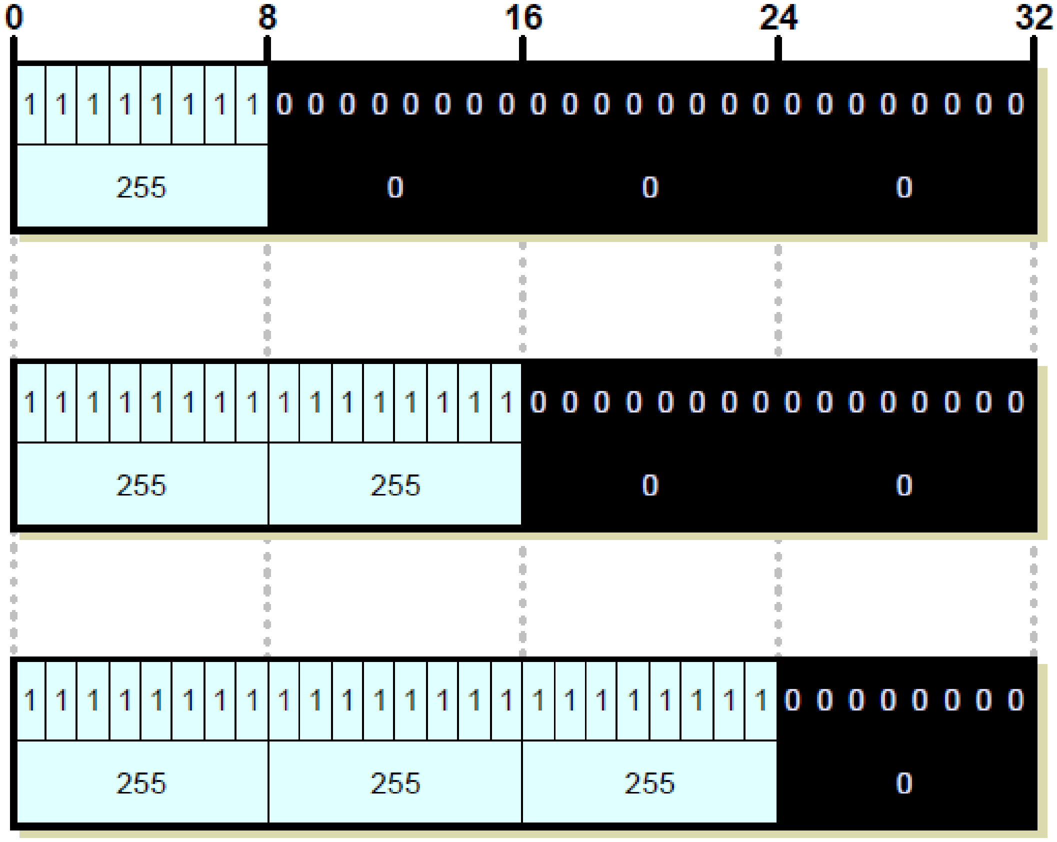
Binary	11100011	01010010	10011101	10110001
Dotted Decimal	227	82	157	177

IP Address: 227.82.157.177
Split Into 8-Bit Network ID and 24-Bit Host ID

NetID & HostID examples

Subnet mask

- A 32-bits long value which is used to identify the network id of an IP address.

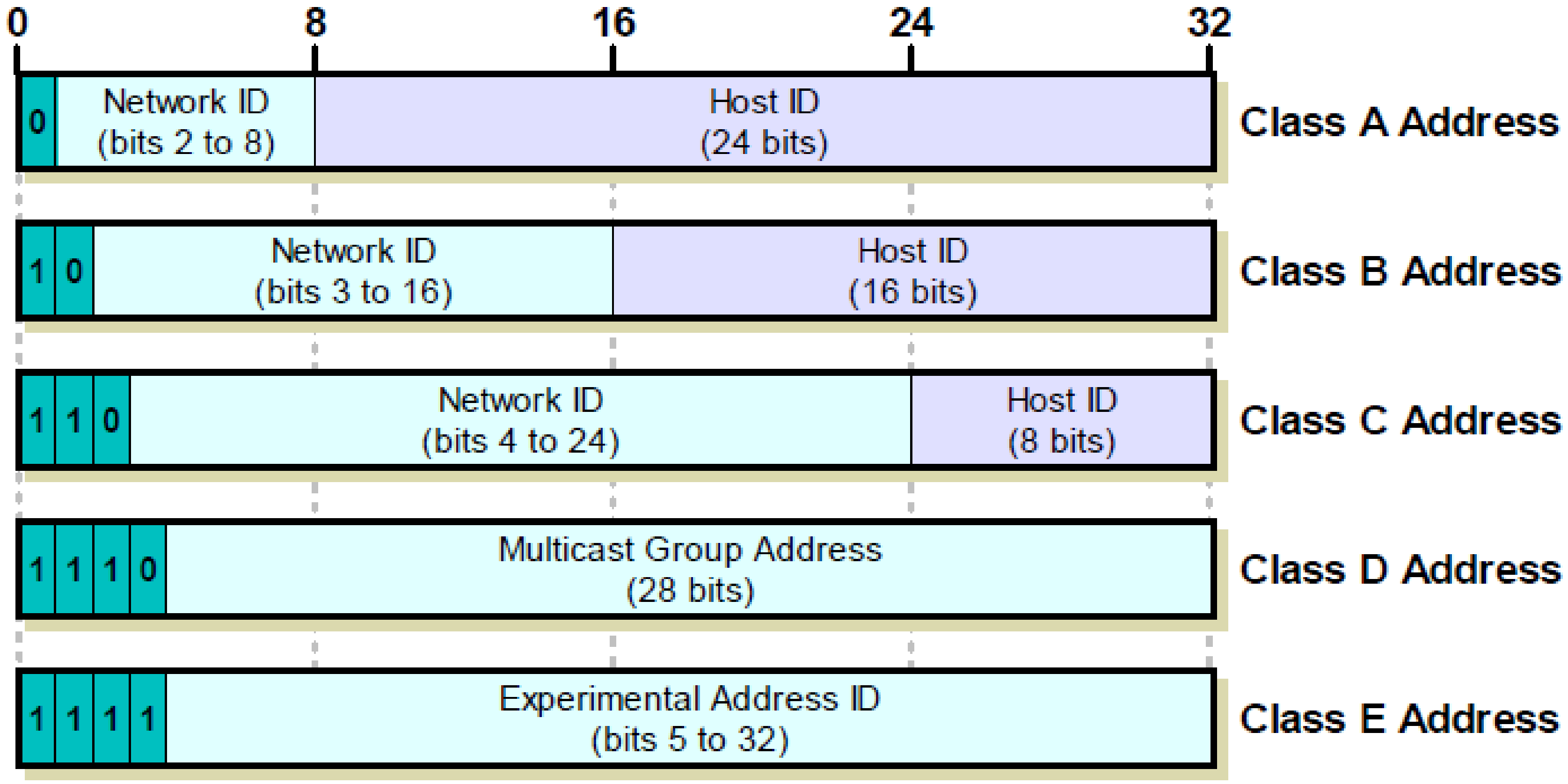


Subnet mask notation

- Dotted Decimal:
  - 10.5.6.7/255.0.0.0,
  - 172.16.32.1/255.255.0.0,
  - 192.168.10.5/255.255.255.0
- Slash notation:
  - 10.5.6.7/8,
  - 172.16.32.1/16,
  - 192.168.10.5/24

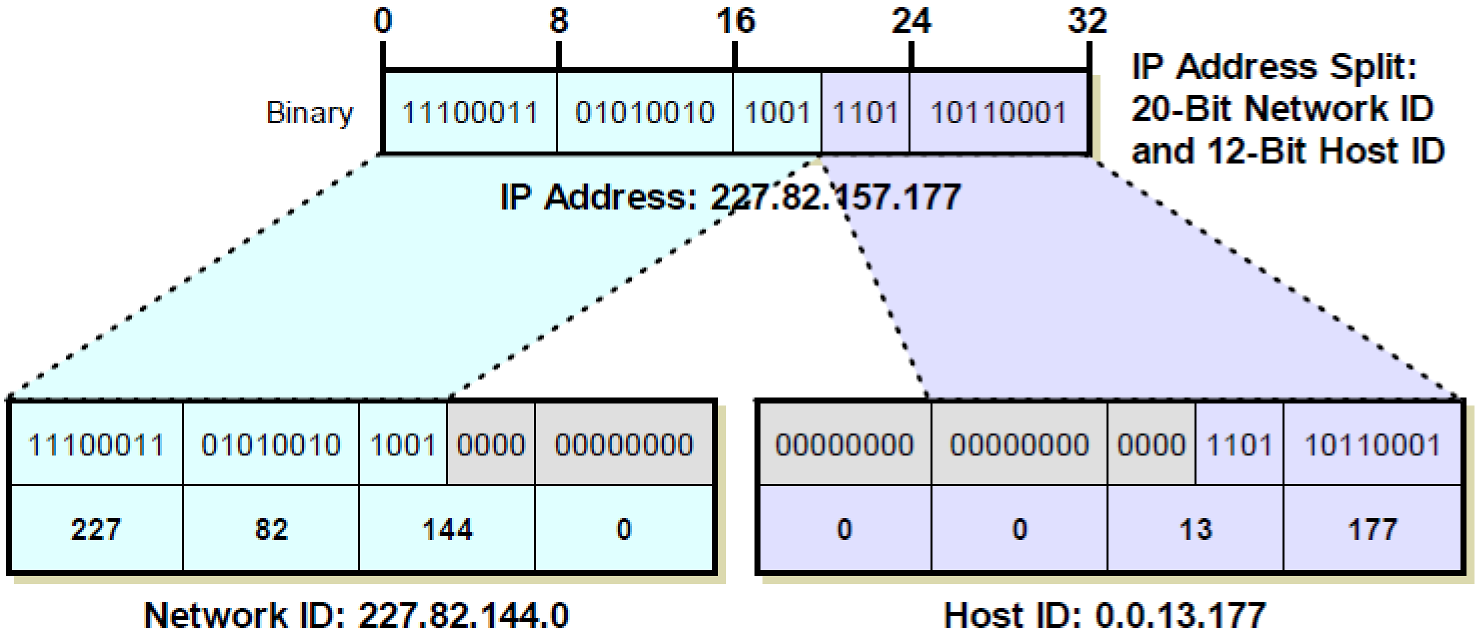
IP Address Categories

- Classful Addressing



IP Address Categories

- Classless Addressing

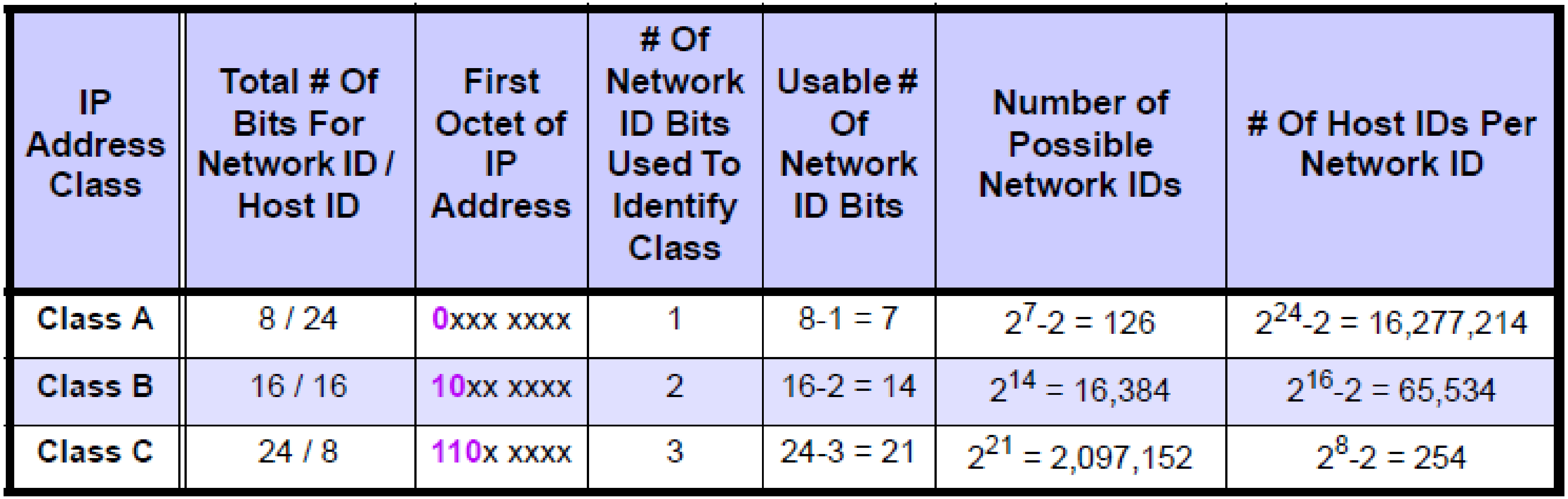


Binary	11100011	01010010	1001	1101	10110001

11100011	01010010	1001	0000	00000000
227	82	144		0

00000000	00000000	0000	1101	10110001
0	0	13		177

IP Class and Host Capablilities



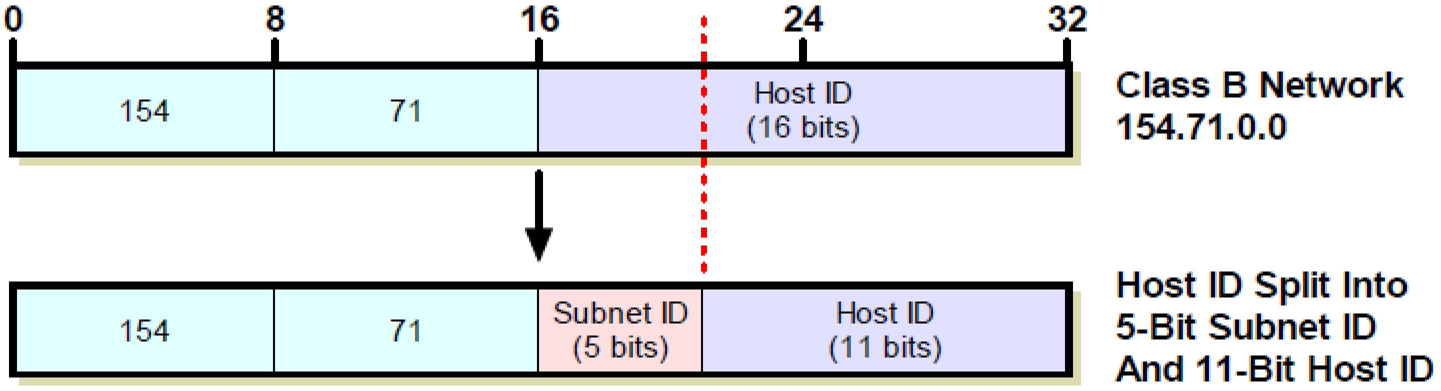
IP Address Class	Total # Of Bits For Network ID / Host ID	First Octet of IP Address	# Of Network ID Bits Used To Identify Class	Usable # Of Network ID Bits	Number of Possible Network IDs	# Of Host IDs Per Network ID
Class A	8 / 24	0xxx xxxx	1	8-1 = 7	2^{7}-2 = 126	2^{24}-2 = 16,277,214
Class B	16 / 16	10xx xxxx	2	16-2 = 14	2^{14} = 16,384	2^{16}-2 = 65,534
Class C	24 / 8	110x xxxx	3	24-3 = 21	2^{21} = 2,097,152	2^{8}-2 = 254

Special IP Addresses

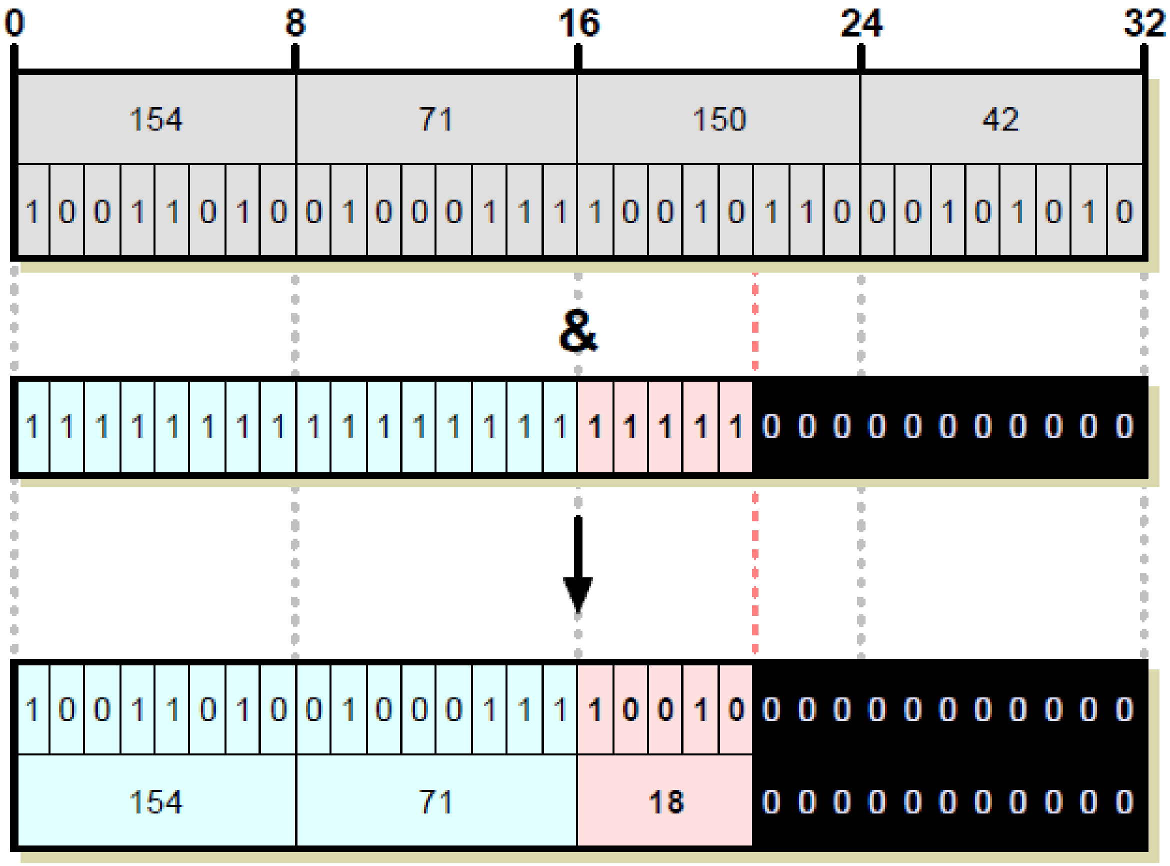
- Loopback : 127.0.0.0 to 127.255.255.255
- All Zeroes (0.0.0.x, 192.168.10.0),
- All Ones (196.254.255.255)

Subnetting

- A “classful” network is subnetted by dividing its host ID portion, leaving some of the bits for the host ID while allocating others to a new *subnet ID*.
- These bits are then used to identify individual subnets within the network, into which hosts are assigned.



Determining the Subnet Mask of a Subnetted Network

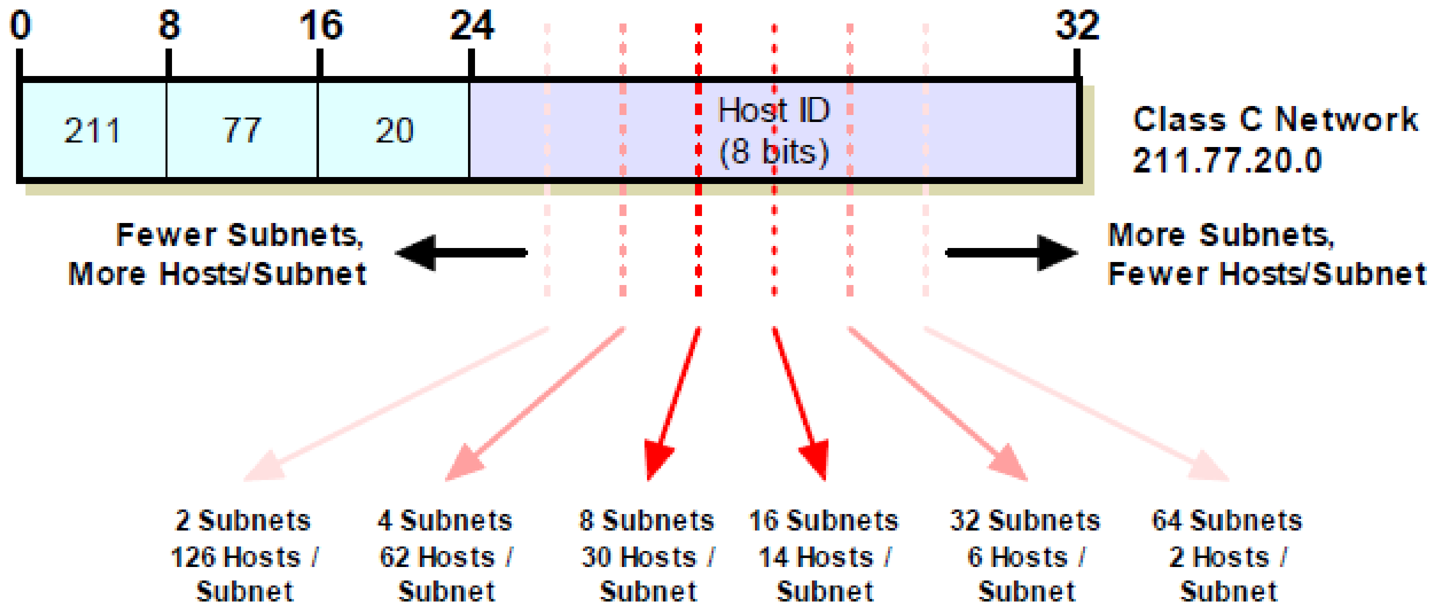


IP Address 154.71.150.42 In Dotted Decimal And Binary

Subnet Mask For Class B Network With 5 Subnet ID Bits

Masked IP Address Showing Network ID (154.71.0.0) And Subnet ID (18)

Subnetting Design Trade-Off

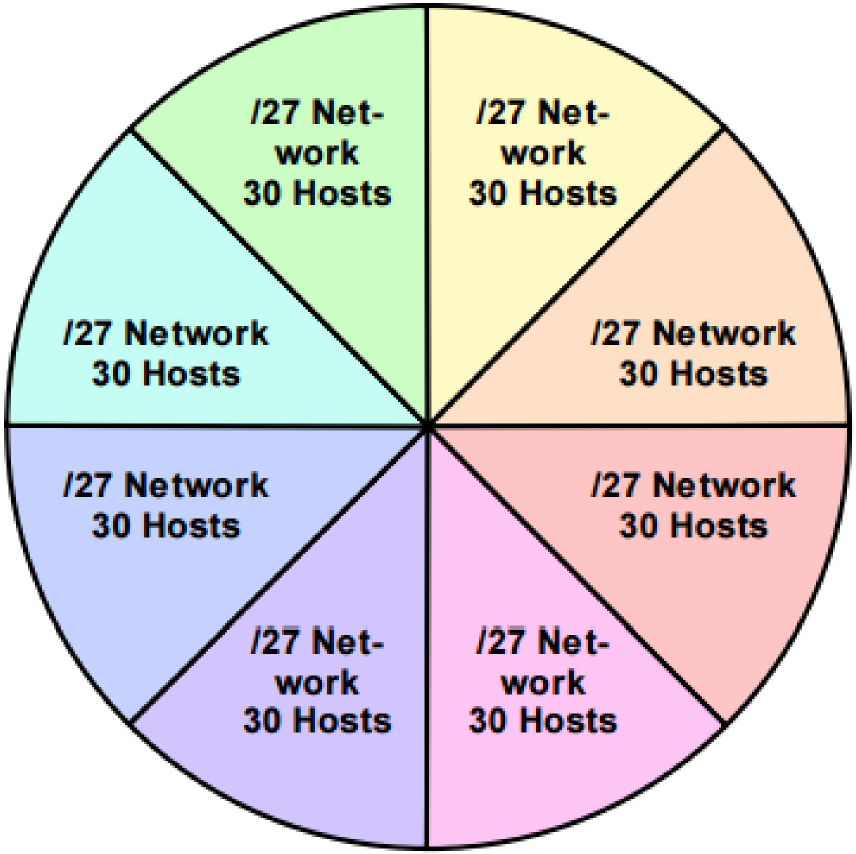


Variable Length Subnet Masking

Problem: a company with class C network 201.11.55.0/24 has 6 subnetworks in which:

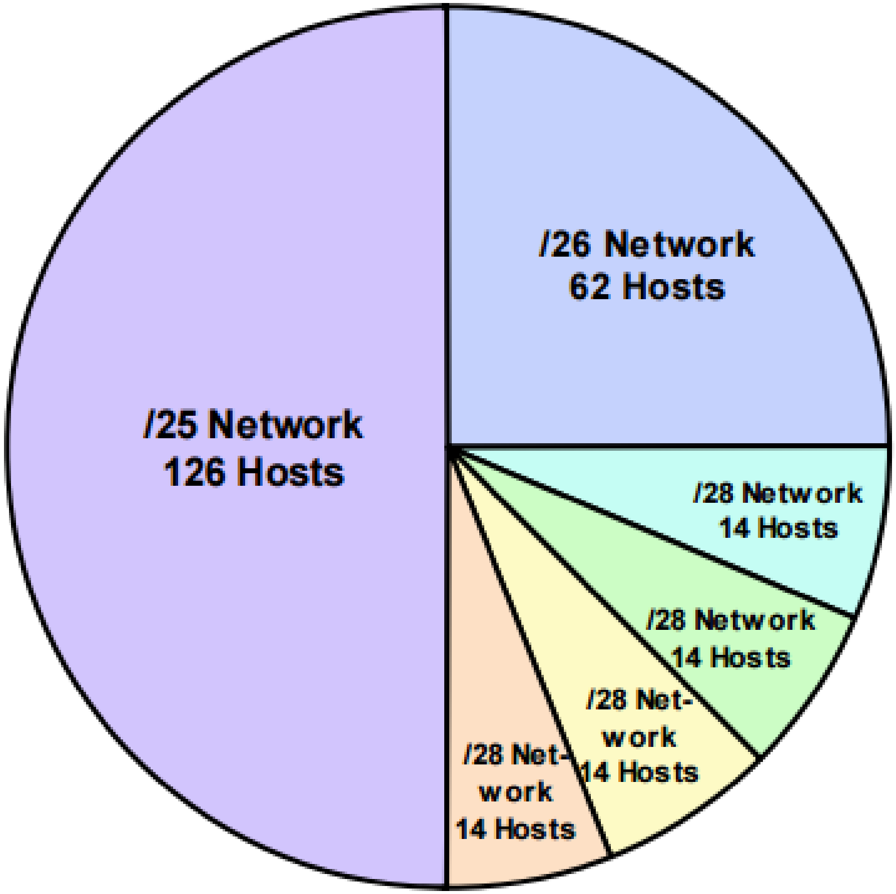
- The first 4 subnets (S1, S2, S3, S4): 10 hosts each,
- The fifth subnet (S5): 50 hosts,
- The last subnet (S6): 100 hosts

Conventional subnetting

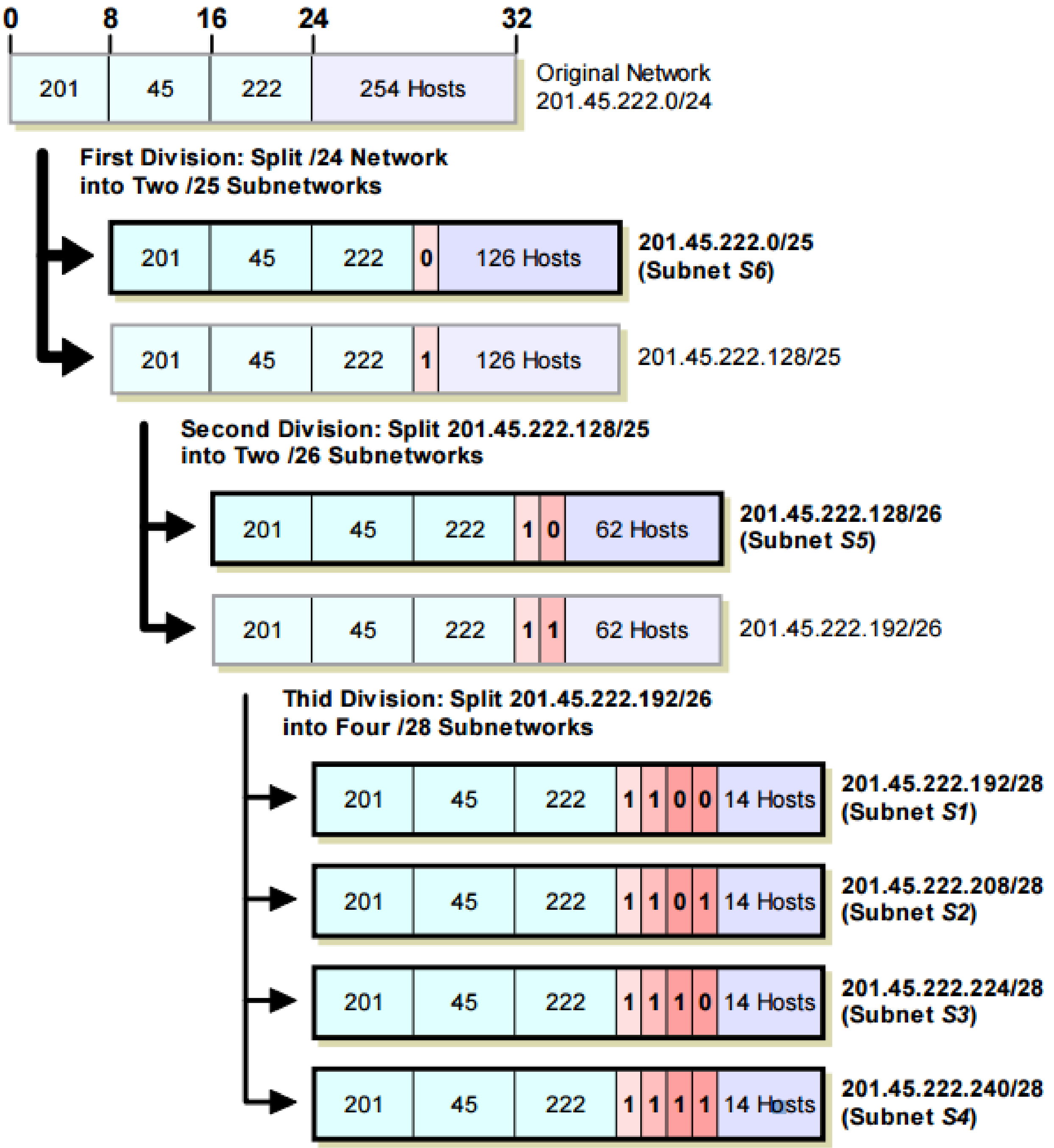


Class C (/24) Network (254 Hosts)

VLSM



Class C (/24) Network (254 Hosts)



0 8 16 24 32

201 | 45 | 222 | 254 Hosts

Original Network
201.45.222.0/24

First Division: Split /24 Network into Two /25 Subnetworks

201 | 45 | 222 | 0 | 126 Hosts

201.45.222.0/25 (Subnet *S6*)

201 | 45 | 222 | 1 | 126 Hosts

201.45.222.128/25

Second Division: Split 201.45.222.128/25 into Two /26 Subnetworks

201 | 45 | 222 | 1 | 0 | 62 Hosts

201.45.222.128/26 (Subnet *S5*)

201 | 45 | 222 | 1 | 1 | 62 Hosts

201.45.222.192/26

Thid Division: Split 201.45.222.192/26 into Four /28 Subnetworks

201 | 45 | 222 | 1 | 1 | 0 | 0 | 14 Hosts

201.45.222.192/28 (Subnet *S1*)

201 | 45 | 222 | 1 | 1 | 0 | 1 | 14 Hosts

201.45.222.208/28 (Subnet *S2*)

201 | 45 | 222 | 1 | 1 | 1 | 0 | 14 Hosts

201.45.222.224/28 (Subnet *S3*)

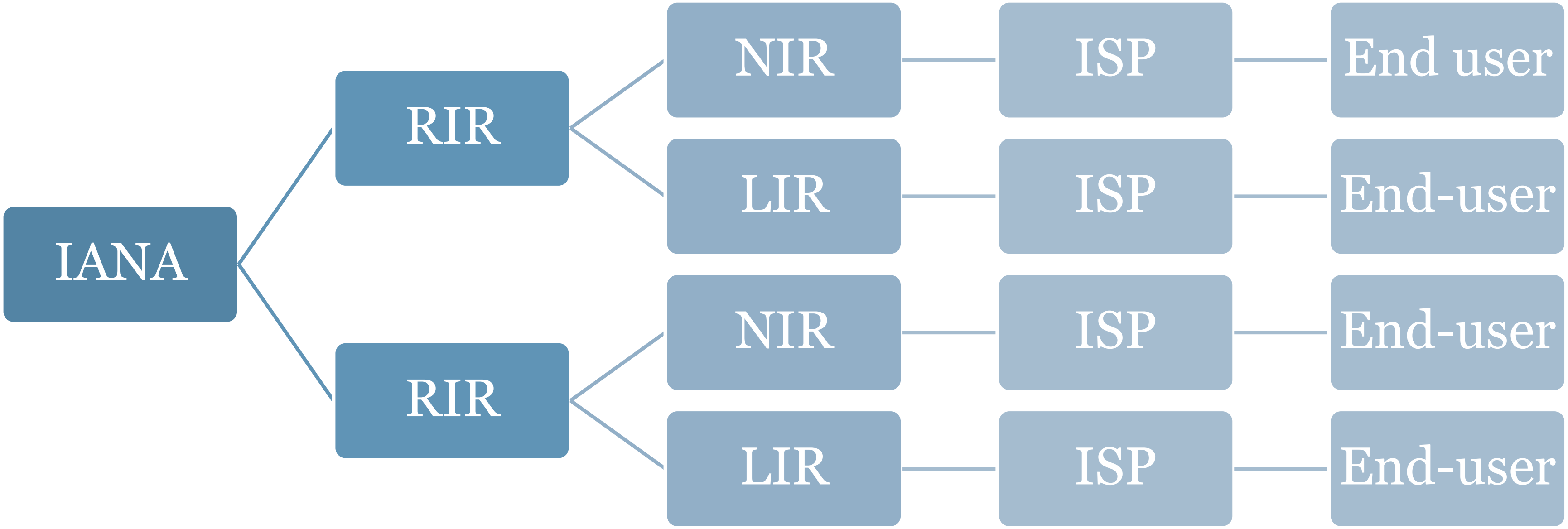
201 | 45 | 222 | 1 | 1 | 1 | 1 | 14 Hosts

201.45.222.240/28 (Subnet *S4*)

IP Address Management & Authorities

- The Need for Centralized Registration
- The Original IP Address Authority: IANA
- In the late 1990s, a new organization called the Internet Corporation for Assigned Names and Numbers (ICANN) was created

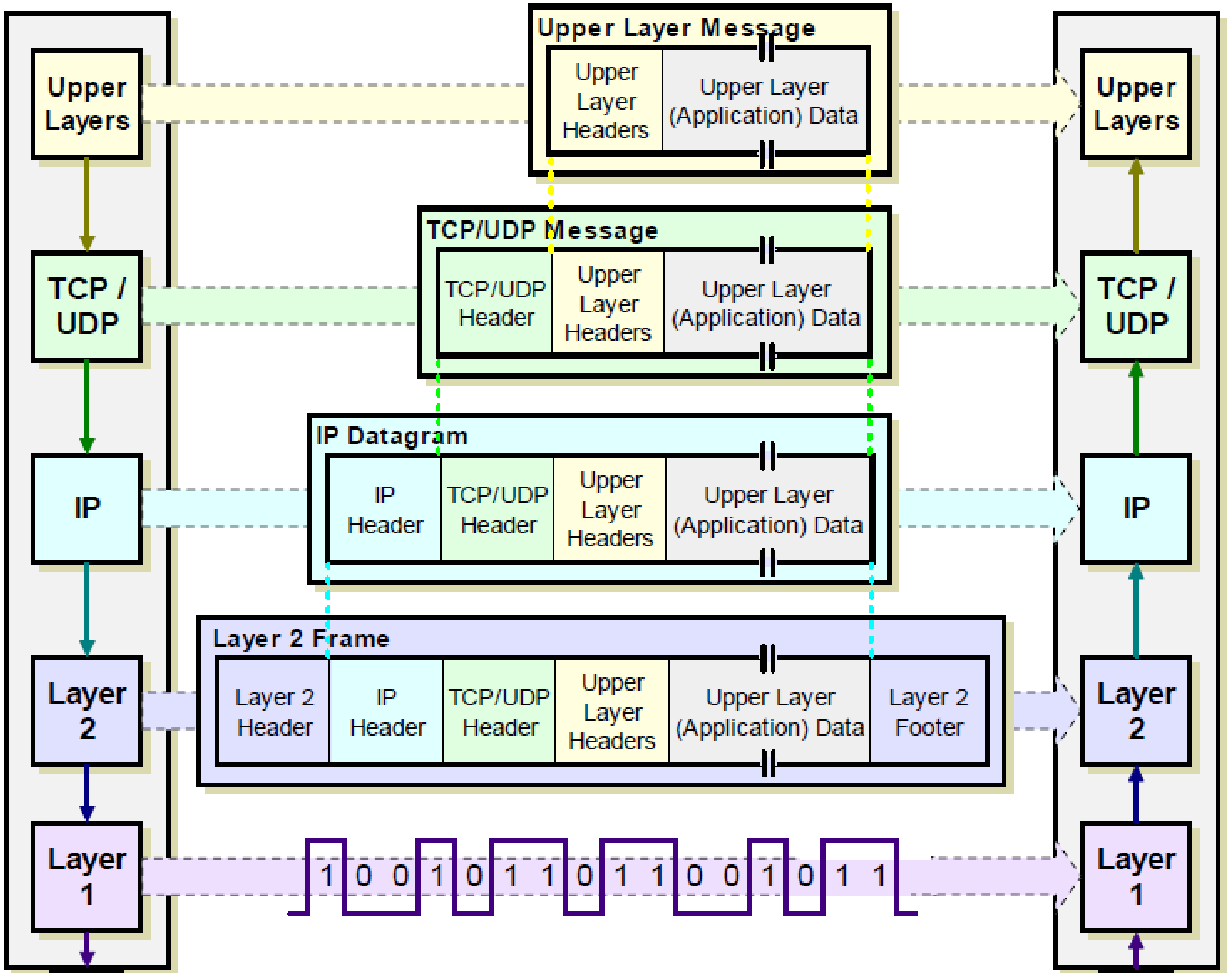
Modern IP Address Registration & Authorities



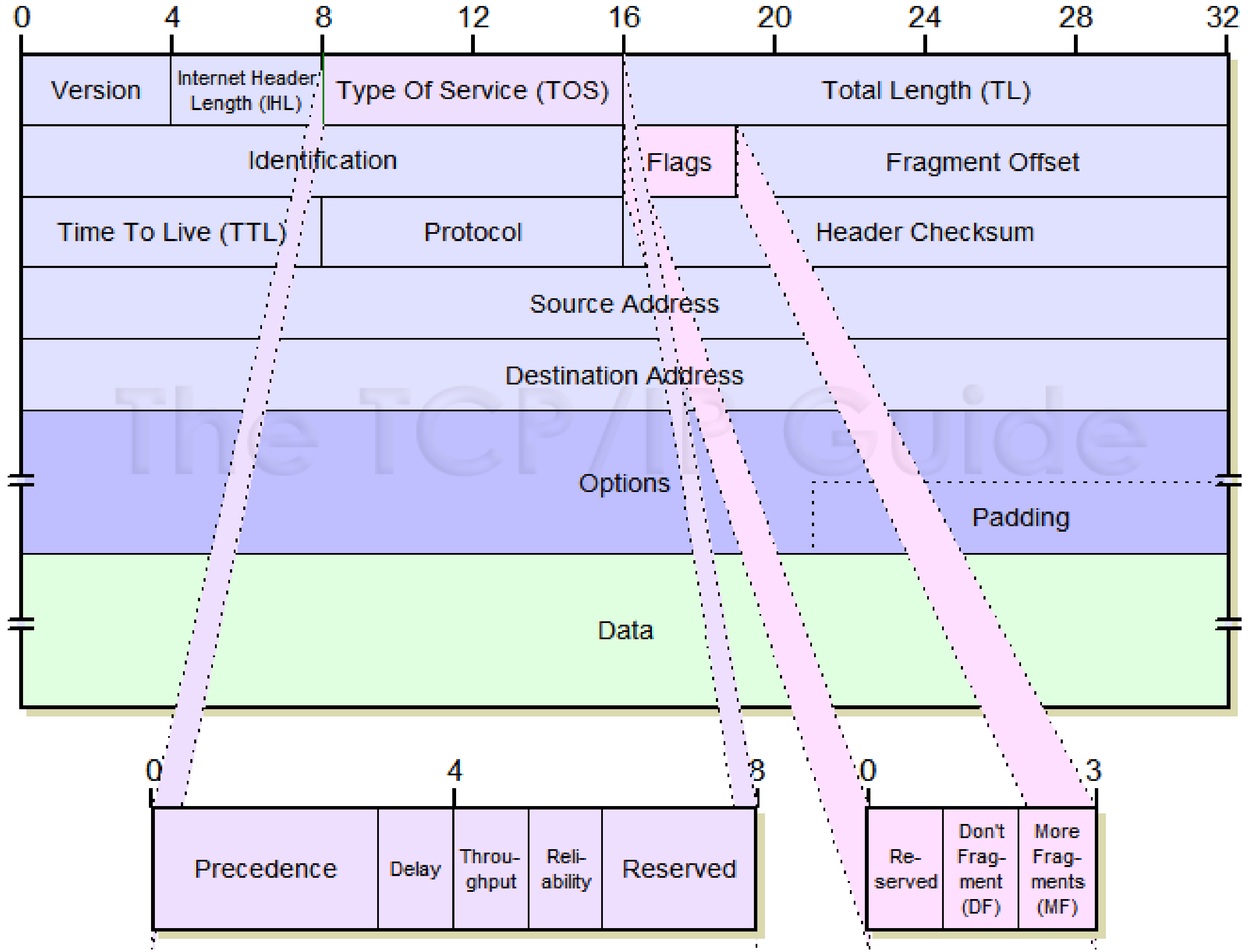
IP Datagram

The IPv4 datagram is conceptually divided into two pieces: the *header* and the *payload*. The header contains addressing and control fields, while the payload carries the actual data to be sent over the internetwork. Unlike some message formats, IP datagrams do not have a footer following the payload.

IP Datagram Encapsulation



IP Datagram



0 4 8 12 16 20 24 28 32

Version | Internet Header Length (IHL) | Type Of Service (TOS) | Total Length (TL)

Identification | Flags | Fragment Offset

Time To Live (TTL) | Protocol | Header Checksum

Source Address

Destination Address

Options | Padding

Data

0 4 8

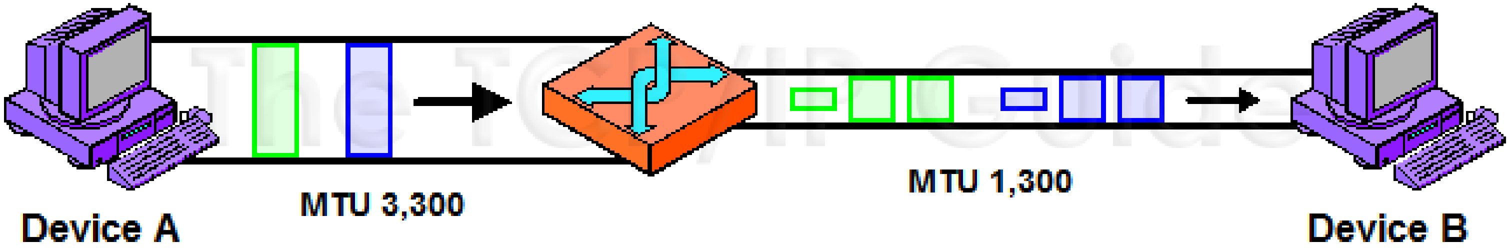
Precedence | Delay | Throughput | Reliability | Reserved

0 3

Reserved | Don't Fragment (DF) | More Fragments (MF)

IP Datagram size, MTU

- Datagram size = IP Header + TCP Header + Upper layer Header + Data
- MTU: The size of the largest IP datagram that can be transmitted over a physical network

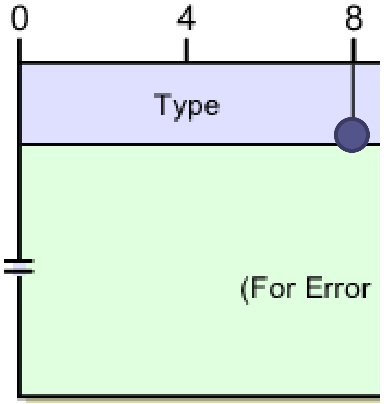


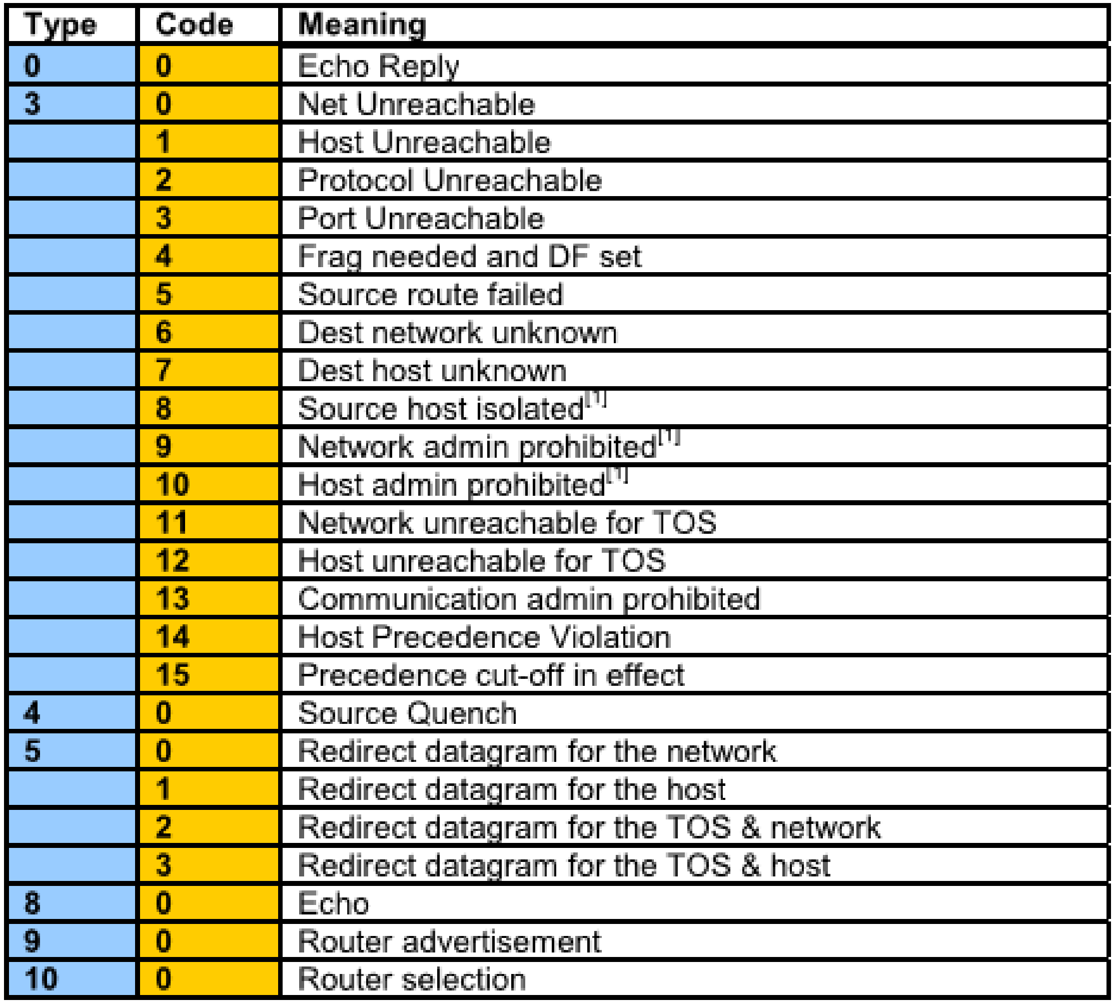
Internet Control Message Protocol (ICMP)

ICMP Message classes

- Error Messages:
  - that are used to report problem conditions
- Informational messages
  - that are used for diagnostics, testing and other purposes

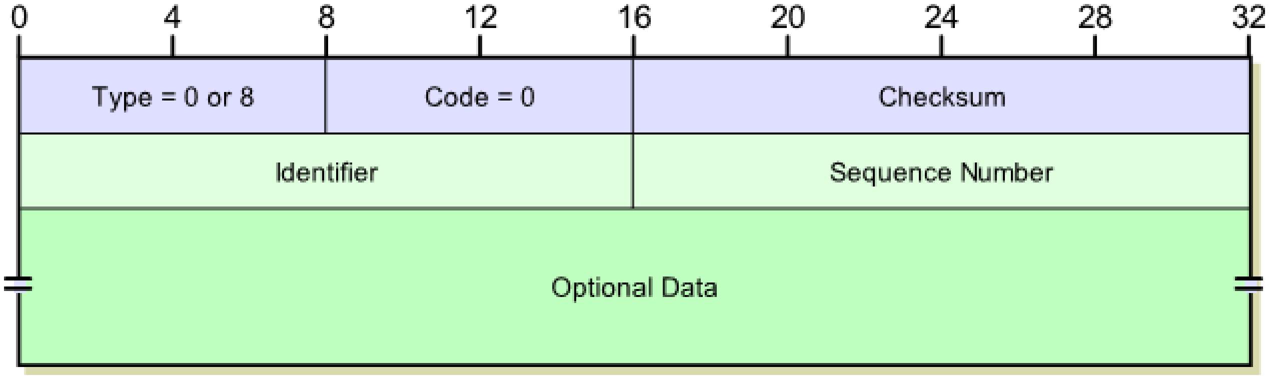
ICMP Common Message Format





Type	Code	Meaning
0	0	Echo Reply
3	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Frag needed and DF set
	5	Source route failed
	6	Dest network unknown
	7	Dest host unknown
	8	Source host isolated[1]
	9	Network admin prohibited[1]
	10	Host admin prohibited[1]
	11	Network unreachable for TOS
	12	Host unreachable for TOS
	13	Communication admin prohibited
	14	Host Precedence Violation
	15	Precedence cut-off in effect
4	0	Source Quench
5	0	Redirect datagram for the network
	1	Redirect datagram for the host
	2	Redirect datagram for the TOS & network
	3	Redirect datagram for the TOS & host
8	0	Echo
9	0	Router advertisement
10	0	Router selection

ICMP Echo & Echo Reply Message



Type = 0 or 8

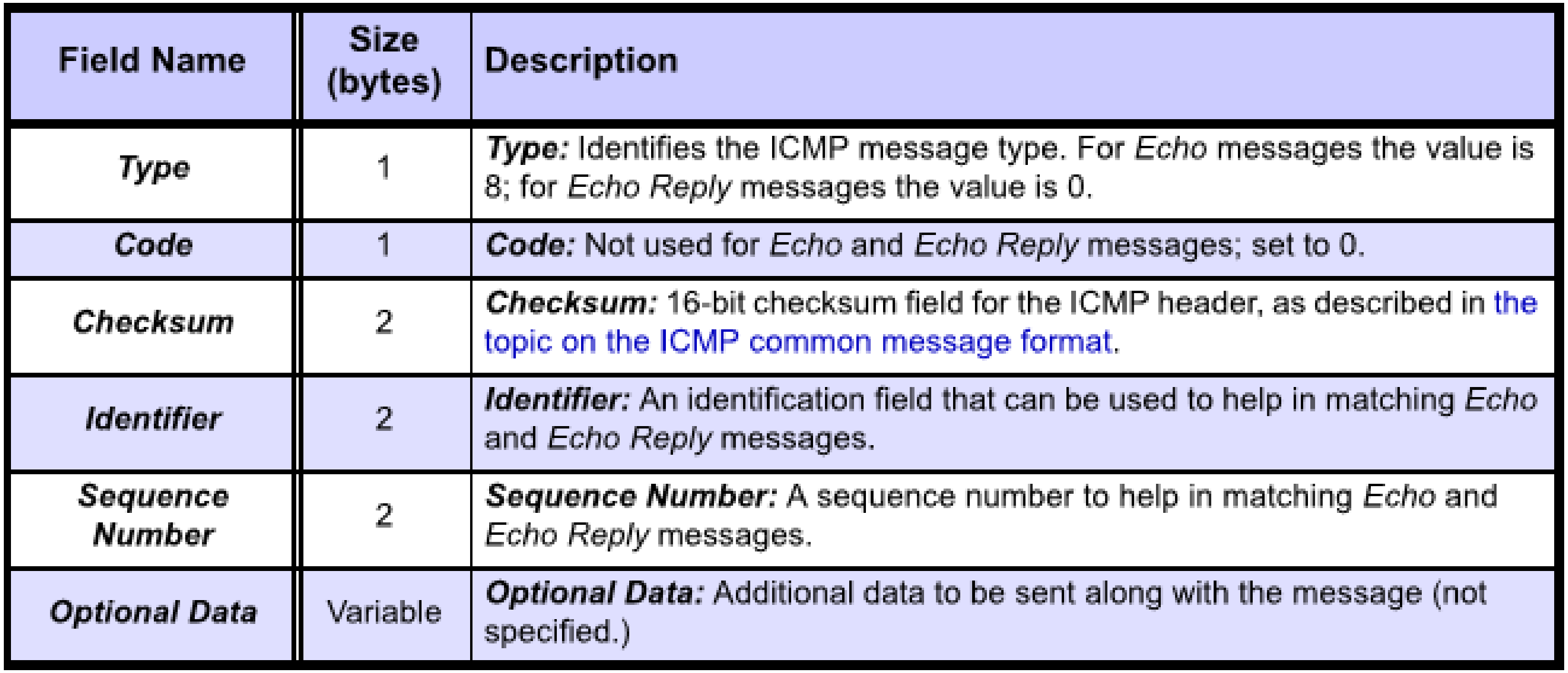
Code = 0

Checksum

Identifier

Sequence Number

Optional Data



Field Name	Size (bytes)	Description
Type	1	***Type:*** Identifies the ICMP message type. For *Echo* messages the value is 8; for *Echo Reply* messages the value is 0.
Code	1	***Code:*** Not used for *Echo* and *Echo Reply* messages; set to 0.
Checksum	2	***Checksum:*** 16-bit checksum field for the ICMP header, as described in the topic on the ICMP common message format.
Identifier	2	***Identifier:*** An identification field that can be used to help in matching *Echo* and *Echo Reply* messages.
Sequence Number	2	***Sequence Number:*** A sequence number to help in matching *Echo* and *Echo Reply* messages.
Optional Data	Variable	***Optional Data:*** Additional data to be sent along with the message (not specified.)