COMPUTER NETWORKS

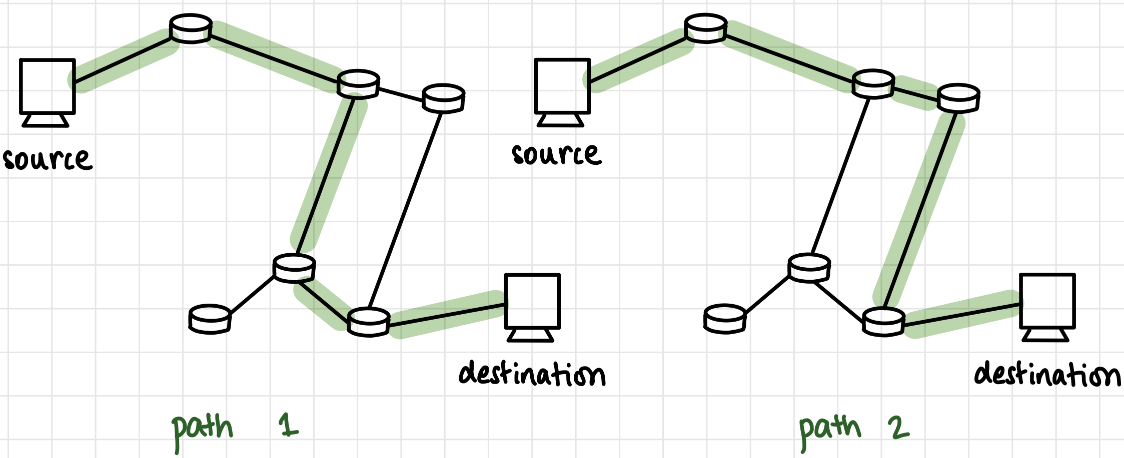
UNIT-4

network layer

feedback/corrections: vibha@pesu.pes.edu VIBHA MASTI

OVERVIEW of NETWORK LAYER

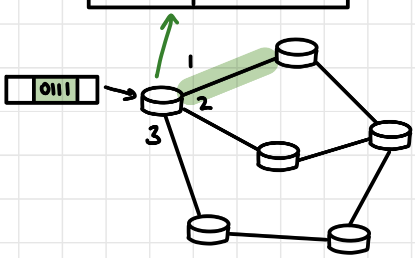
- Responsible for routing and forwarding
- Transports segment from sender to receiver
  - Sender: encapsulates segments into datagrams, passes to link layer
  - Receiver: delivers segment to transport layer
- Datagram sent from sender to receiver by routing through routers and end systems
- Multiple paths from Host 1 to Host 2: decisions taken at network layer
- Router examines header of packet and forwards it
- IP addresses assigned to interfaces, not devices



- Each router has 2 ports: input & output (additional ports through external interfaces)
- Forwarding: transfer packet from one port to another port
- Routers generally have more than 2 ports; therefore forwarding required
- Recall: hardware lab experiment
- Forwarding function uses only forwarding table, no algorithm
- Routing algorithms used for routing (tables cannot be used for large networks)

local forwarding table

header	output link
0100	3
0101	2
0111	1
⋮	⋮



connection setup

- some architectures: ATM, frame relay, X.25 require handshaking
- routers along chosen path handshake with each other; set up state

- network layer: connection between end devices; responsibility of routers
- transport layer: connection between processes (ports)

Service Model

- Default for internet: best-effort service model
- No guarantee in time frame, no guaranteed order, guaranteed data delivery, guaranteed bandwidth in best-effort service

no guarantee on: (1) successful delivery of datagram
(2) timing
(3) order
(4) bandwidth

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet	Diffserv (RFC 2475)	possible	possibly	possibly	no

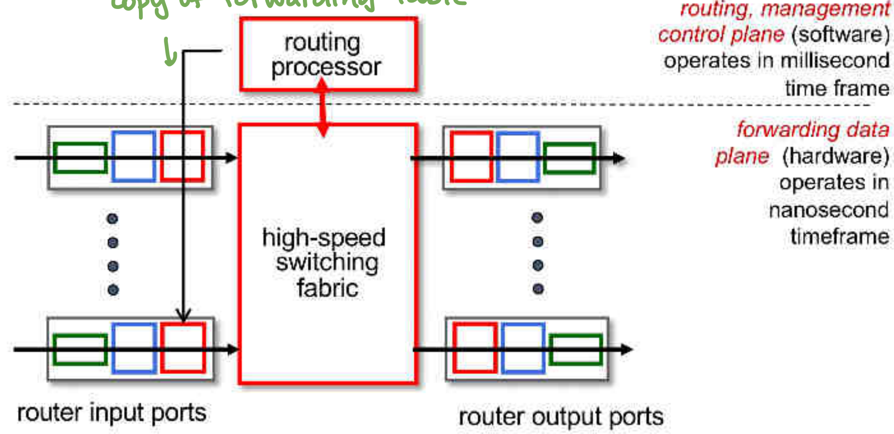
ROUTER

High Level Overview

Four router components

1. router input ports
2. router output ports
3. high speed switching fabric: connects i/p ports to o/p ports
4. routing processor: executes routing protocols, fills forwarding table

copy of forwarding table



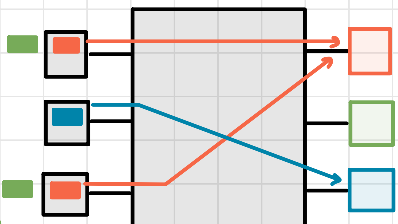
routing

forwarding

— INPUT PORT QUEUEING

- If packets come in at a speed greater than router's fabric, queuing will occur

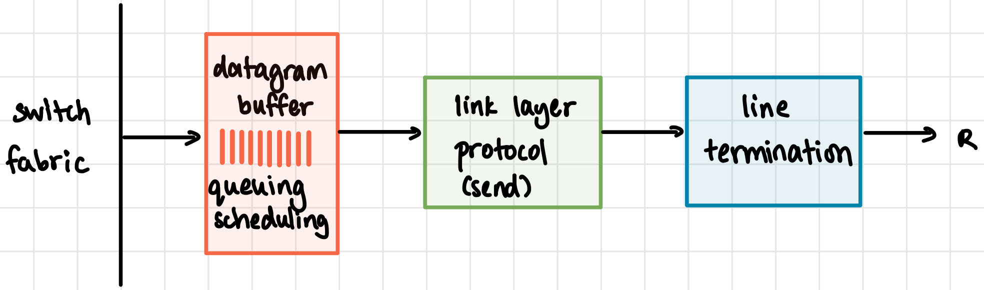
- If switch fabric slower than input ports combined, input port queuing occurs



HOL blocking

Head of the Line Blocking:
HOL blocking — datagram at head of queue blocks (green blocked by red)

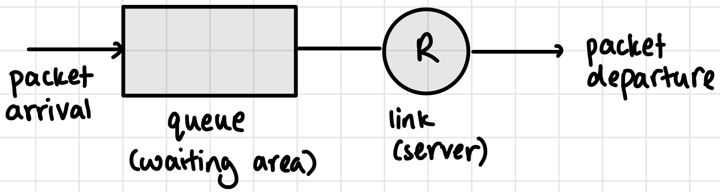
OUTPUT PORT QUEUEING



- Datagrams arrive from fabric faster than link transmission rate - buffering (o/p)
- Drop policy - what datagrams to drop if no free buffers; data loss
- Scheduling discipline chooses among queued datagrams for transmission (eg: priority scheduling)

Buffer Management

Abstraction: queue



DROP

- tail drop - drop arriving packet
- priority drop - drop based on priority

MARKING

- which packets to mark to signal congestion (ECN, RED)

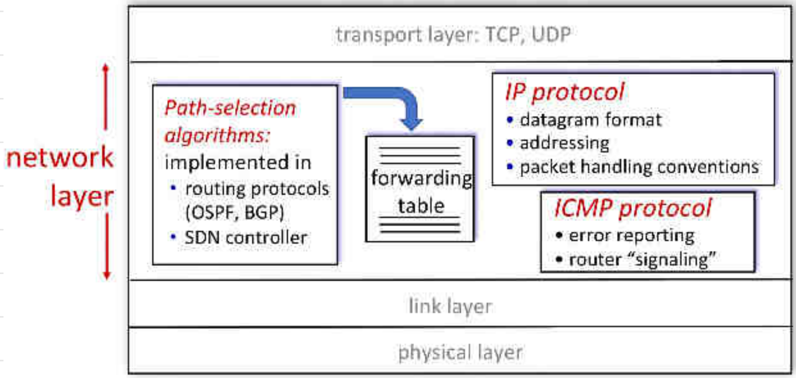
INTERNET PROTOCOL (IP)

- IPv4 ⟶ 32 bits

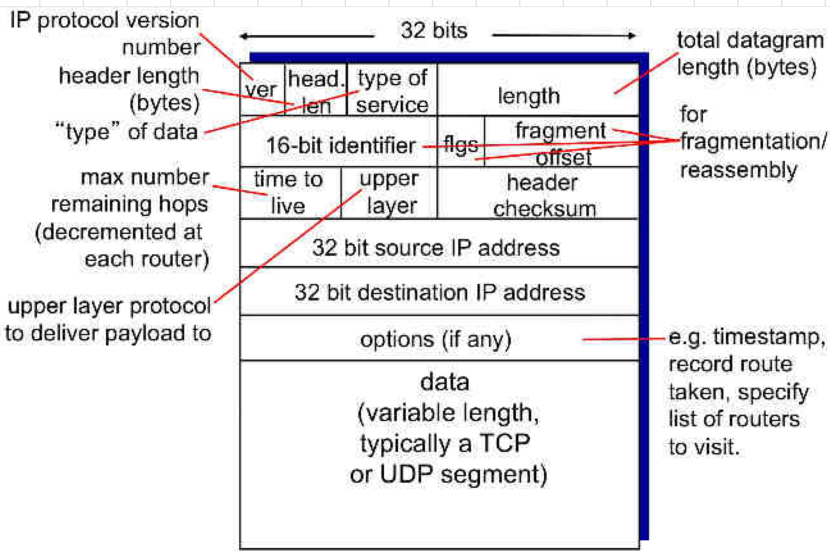
  0.0.0.0 to 255.255.255.255

- IPv6

- Addressing specified for interfaces, not devices.



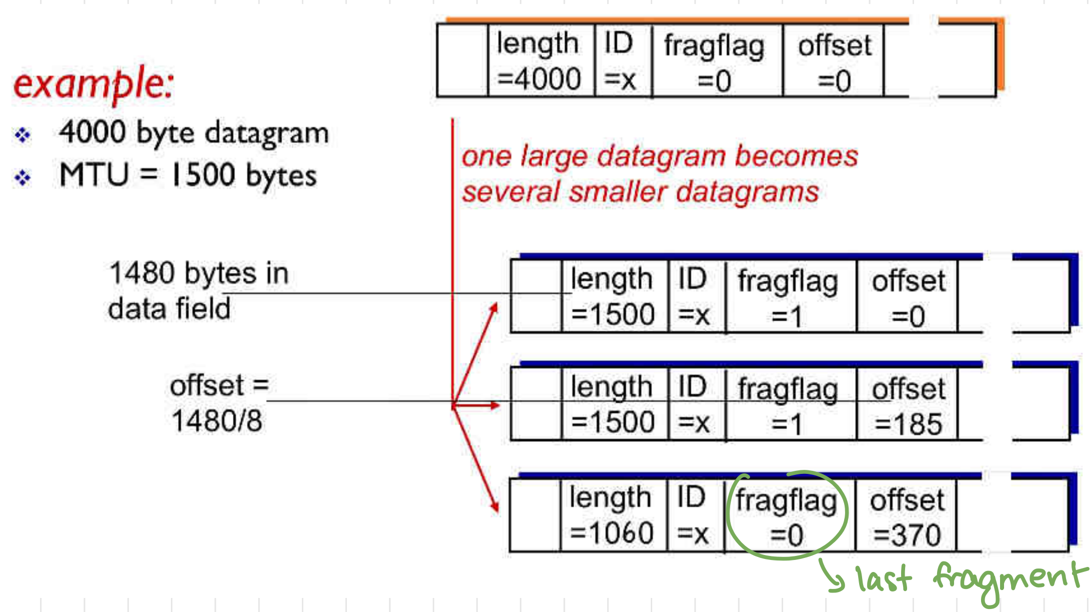
IP Datagram Format



IP Fragmentation

- Network links – MTU (max transfer size)
- Link-level frame

- Large datagram divided (fragmented)
- One datagram → several datagrams
- Reassembled at final destination (host), not routers



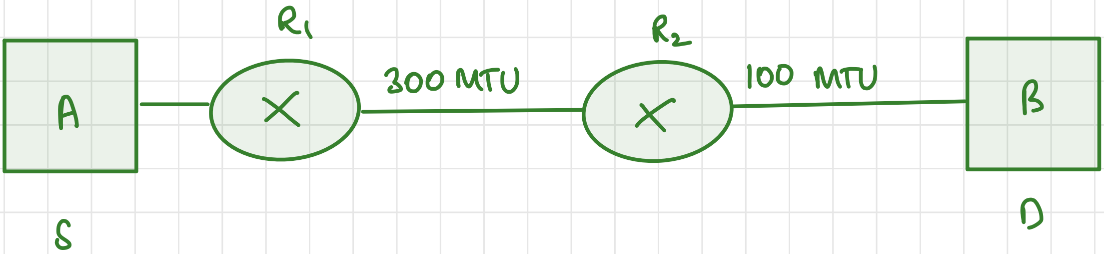
- 20 bytes header

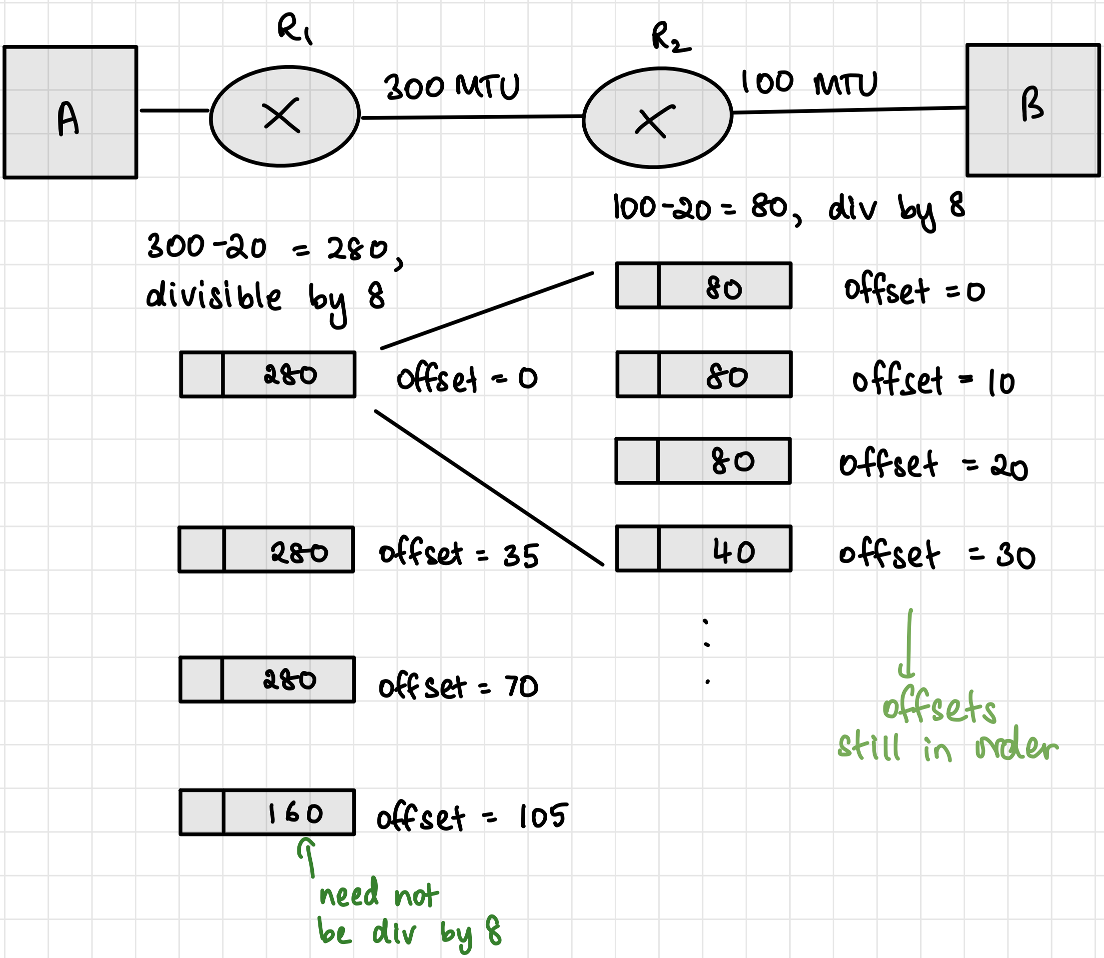
Q: An IP Router with MTU = 200 bytes, IP packet size = 520 bytes, header 20 bytes. Fragment size = ?

[20 | 176] ← 180 not div by 8 [20 | 176] [20 | 148]

offset	0	22	44
MF	1	1	0

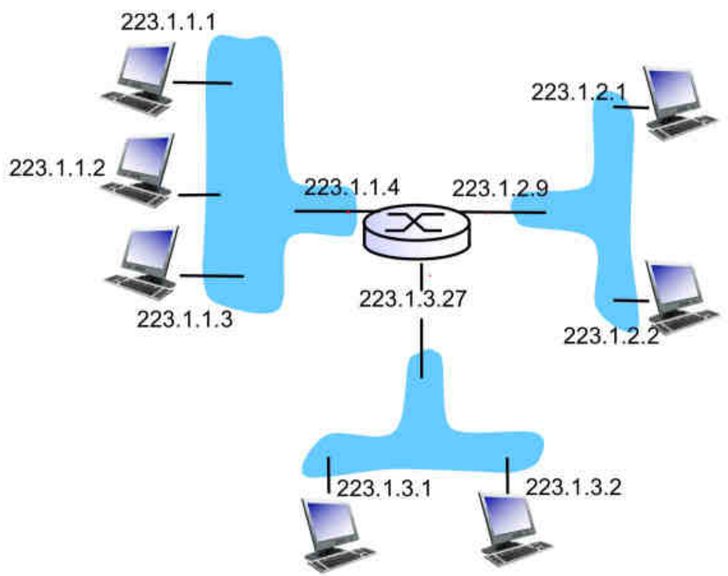
Q: How will the datagram be fragmented?





IPv4 Addressing

- IPv4 Address – 32 bit unique ID
- 4 octets separated by dots



223.1.1.1

= 11011111 00000001 00000001 00000001

11011111	00000001	00000001	00000001
223	1	1	1

- IP address is unique to every interface
- Interface: connection between host/router and physical link

Q: Convert to decimal IP Address

i)	10000001	00001011	00001011	11101111
	129	11	11	239
ii)	11000001	10000011	00011011	11111111
	193	131	27	255

Q: Convert to binary

i) 111.56.45.78

01101111 00111000 00101101 01001110

ii) 221.34.7.82

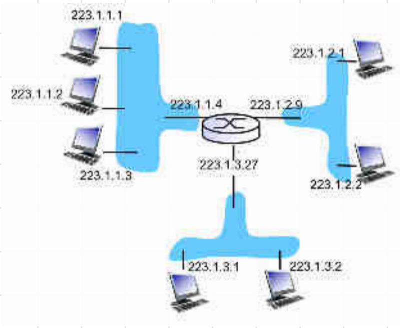
11011101 00100010 00000111 01010010

INTERFACE CONNECTION

- Interfaces connected through wires using switches (wired)
- Switch: interconnection between end devices and router

Subnet

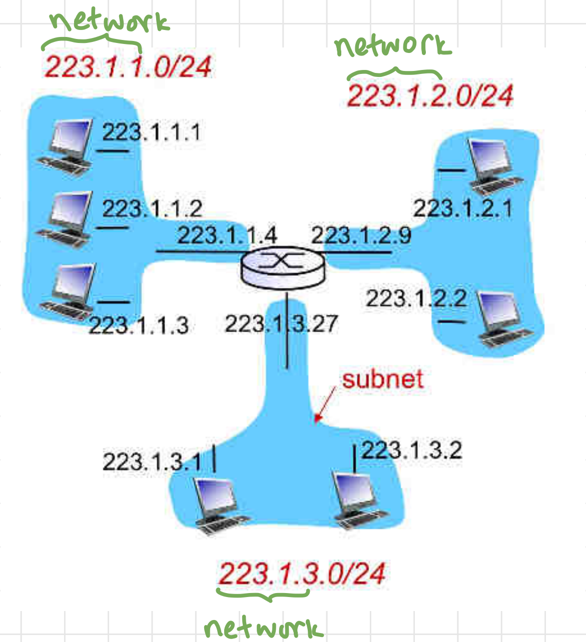
- Sub-networks created when a router is logically removed from a network
- Device interfaces that can physically reach each other without passing through a router



network consists of 3 subnets

Structure of IP Address

- Subnet part: devices in same subnet have common higher order bits
- Host part: remaining lower order bits
- Subnet IDs (network IDs)
- 24 — 8 x 3 bits ⟶ subnet mask: /24



ADDRESSING

- IPv4

CLASSFUL ADDRESSING

- Class A, Class B, Class C

- Network portion 8 — class A (/8)
  Network portion 16 — class B (/16)
  Network portion 24 - class C (/24)
  Multicast ———— class D
  Broadcast ———— class E

- 2 addresses reserved (127.0.0.1 - network address and broadcast address)

- Hosts Class C = $2^8 - 2 = 254$ } too small
  Hosts class B = $2^{16} - 2 = 65634$ } too large
  Hosts class A = $2^{24} - 2 = 16\,777\,214$

- Classful addressing, therefore, almost obsolete and classless used

Identifying Class from First Byte

	First byte	Second byte	Third byte	Fourth byte	
Class A	0				/8
Class B	10				/16
Class C	110				/24
Class D	1110				
Class E	1111				

binary representation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

assignable : 1-126

decimal representation

class A

	7 bit	24 bit
0	network	host

class B

		14 bit	16 bit
1	0	network	host

class C

			21 bit	8 bit
1	1	0	network	host

class D

				28 bit
1	1	1	0	host

class E

				28 bit
1	1	1	1	host

CLASSLESS ADDRESSING

- no restriction on subnetting
- CIDR - Classless InterDomain Routing "cider"
- format: a.b.c.d/x where x = no. of bits in subnet portion of address

Q: An organization is granted the block 214.17.160.0/24. The administrator wants to create 8 subnets.

a. Find the subnet mask.
b. Find the number of addresses in each subnet.
c. Find the last addresses in first subnet.
d. Find the first addresses in last subnet.

All possible addresses must fall in 214.17.160.0/24

(a) Must be divided into 8 subnets

$\therefore \log_2 8 = 3$ additional bits req

$\therefore$ subnet mask = /27

(b) No. of host bits = 32-27 = 5 ⇒ no. of addresses = $2^5 = 32$

no. of assignable addresses = 32-2 = 30 (hosts)

(c) Last address in first subnet = 214.17.160.$(000\ 11111)_2$

= 214.17.160.31

(d) First address in last subnet = 214.17.160.$(111\ 00000)_2$

= 214.17.160.224

Q: Find the class of the following IP addresses

(i) 208.34.54.12 — C (192-223)
(ii) 238.34.2.1 — D (224-239)
(iii) 114.34.2.8 — A (0-127)

Q: Find network ID and host ID

(i) 114.34.2.8 class A: /8 net ID: 114 host ID: 34.2.8
(ii) 132.56.8.6 class B: /16 net ID: 132.56 host ID: 8.6
(iii) 222.35.4.1 class C: /24 net ID: 222.35.4 host ID: 1

Q: In a block of addresses, IP address of one host is 25.34.12.56/16. What are the first & last addresses in this block?

25.34.0.0 ⟶ 25.34.255.255

first address = 25.34.12.56 && mask

last address = 25.34.12.56 || ~mask

Q: An organisation is granted the block 16.0.0.0/8. The admin wants to create 500 fixed length subnets.
i) Find subnet mask
ii) Find no. of addresses in subnet each
iii) Find first & last addresses in subnet 1
iv) Find first & last addresses in subnet 500

additional bits = $\lceil \log_2 500 \rceil = 9$

(i) subnet mask = 9 + 8 = 17

(ii) no. of addresses/subnet $= 2^{32-17} = 2^{15}$

(iii) subnet 1 = 16.0000 0000. 00000000. 00000000

first addr = 16.0.0.0

last addr = 16.0.01111111.255
= 16.0.127.255

(iv) subnet 512 = 16.1111 1111 .1 0000000. 00000000

500 = 512 − 12 =

```
   1111  1111   1
 −         101  0
 ----------------
   1111  1010   1
```

12 = 1010

first = 16.255.128.0

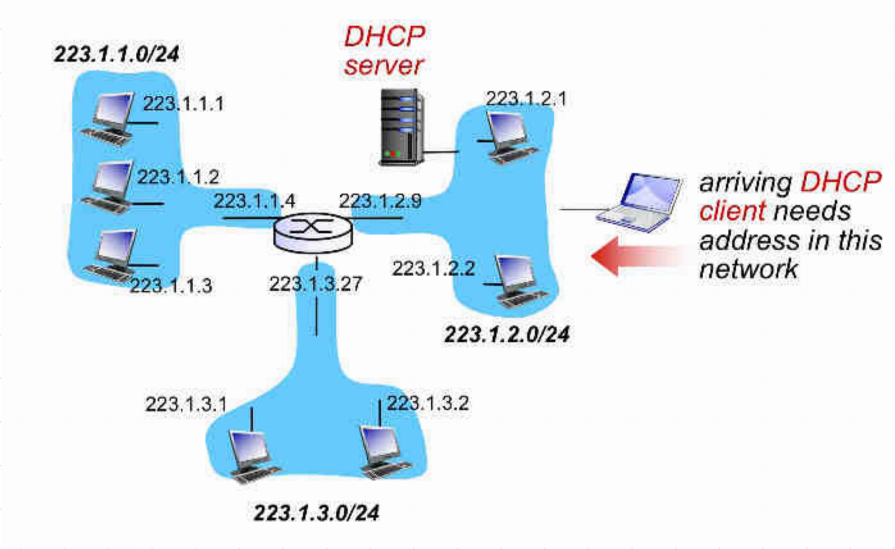
last = 16.255.255.255

DHCP

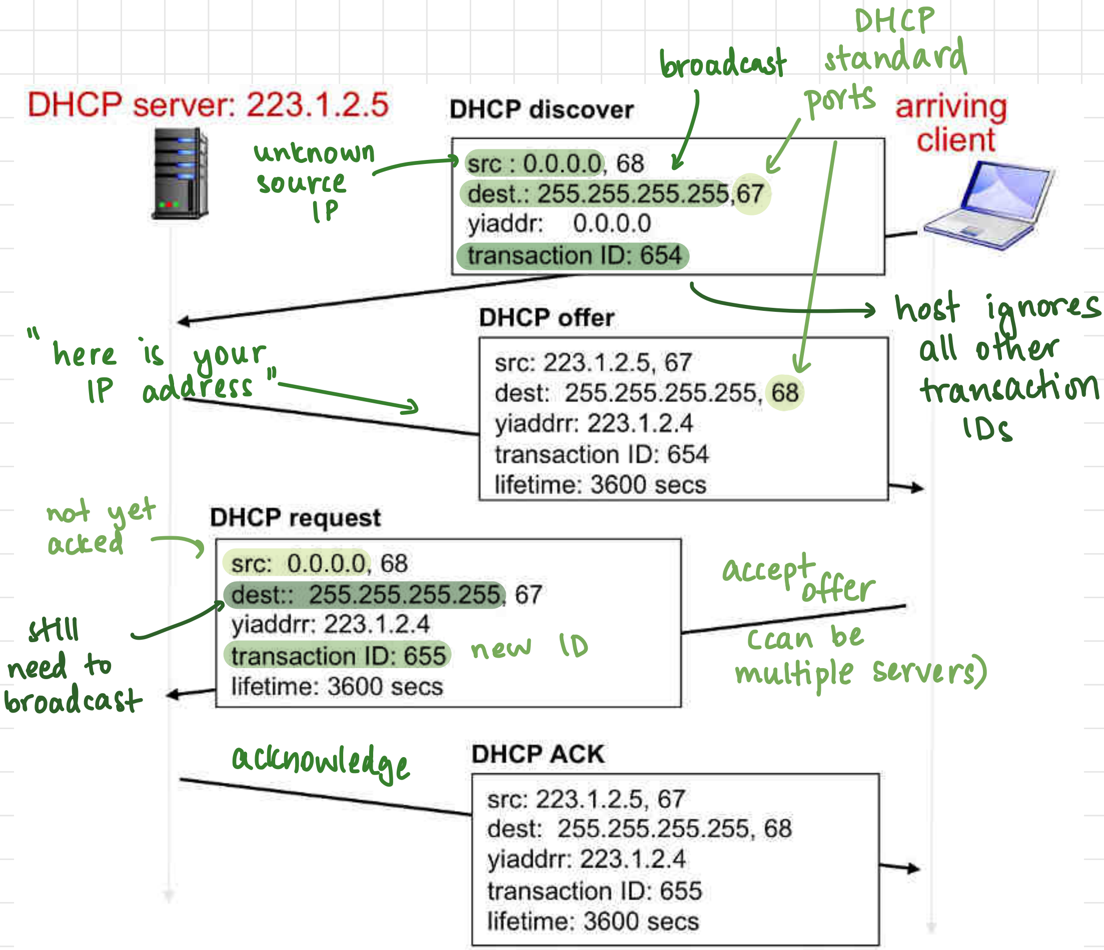
- Application layer protocol, not network layer
- Dynamic Host Configuration Protocol
- Dynamically obtains IP address from network server when it joins network
- Allows reuse of addresses, support for mobile users who join/leave network

Overview

- Host entering network broadcasts DHCP discover message
- DHCP server in network responds with DHCP offer message
- Host requests IP address: DHCP request message
- DHCP server sends IP address: DHCP ack message

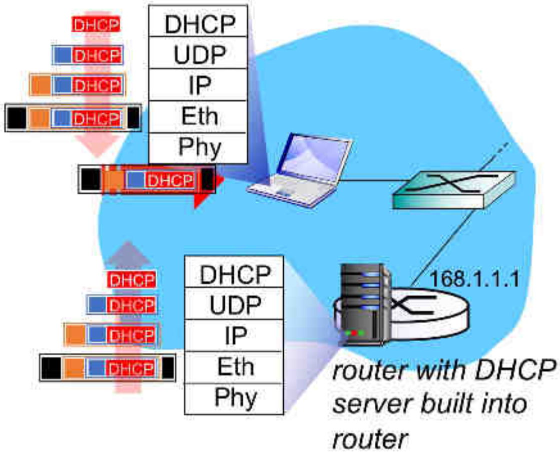


PROTOCOL



- host can renew lease
- can return address of first-hop router, name & IP address of DNS server, network mask
- ICANN for ISPs

Layers



ICMP PROTOCOL

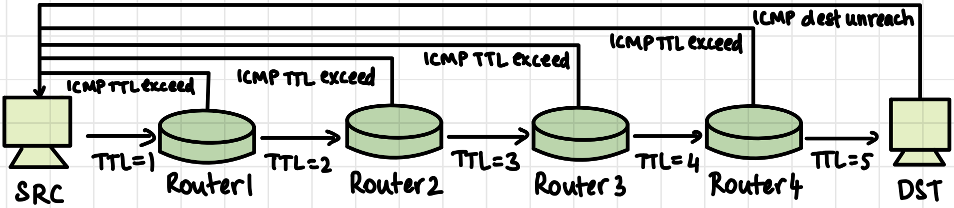
- Internet Control Message Protocol
- Communicate control messages (network layer information) between hosts and routers
- Network layer protocol
- Typically used for error reporting
- Architecturally lies above IP and below transport, but often considered to be a part of IP
- IP datagrams contain ICMP msg

Type & Code

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

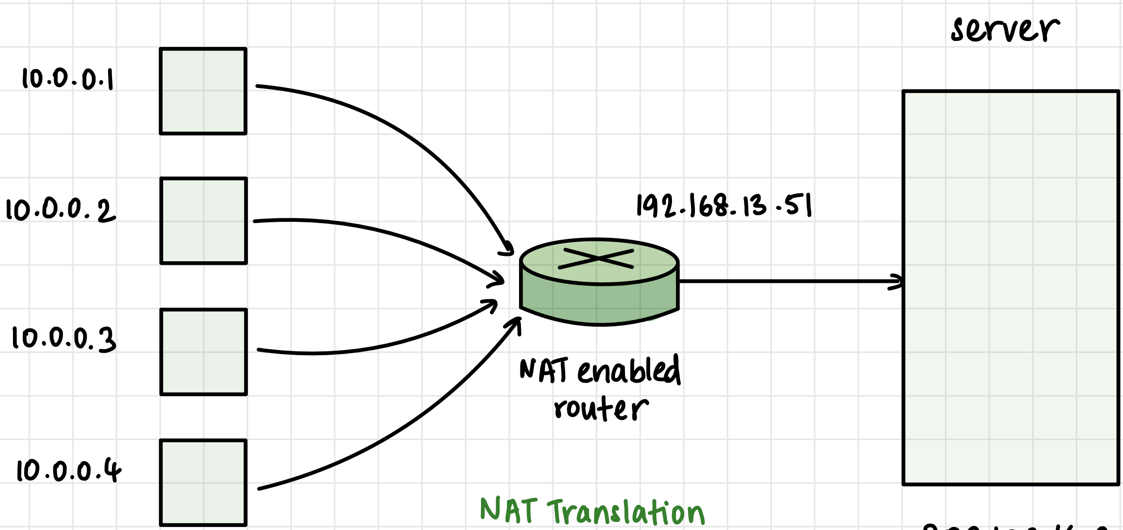
- Utilities — traceroute & ping

Traceroute



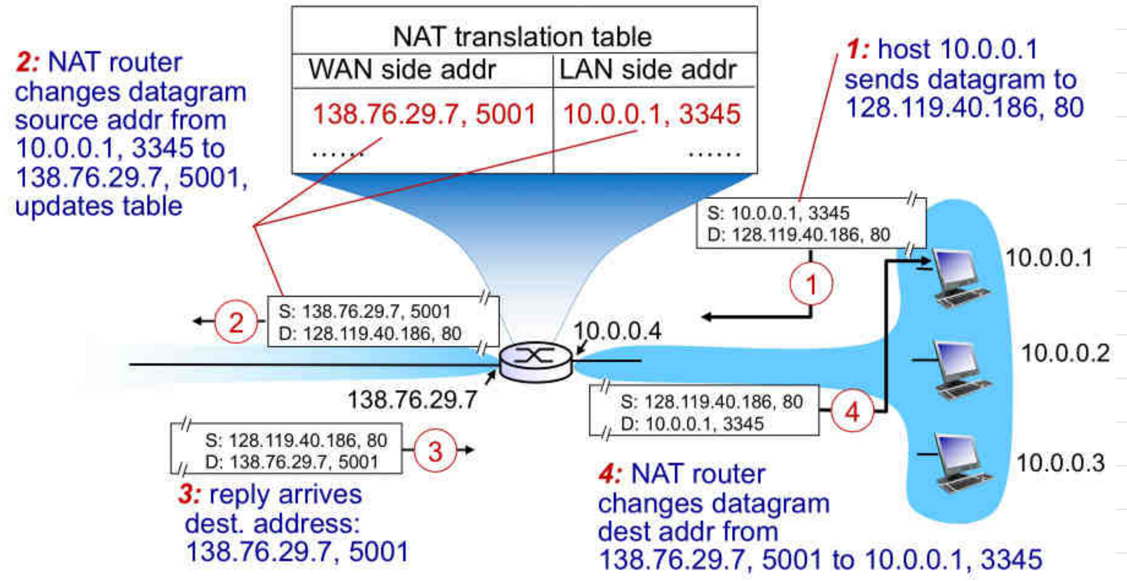
NAT - Network Address Translation

- Entire local network connected to the outside world with a single IP address
- Internally, possesses local addresses that are unique only within local network but not unique over multiple networks
- NAT Translation table from WAN side addr to LAN side address



NAT Translation Table:
port numbers

- Translates LAN side addresses to WAN side addresses and WAN side addresses to LAN side addresses



- At network layer, operating on port numbers (violates layered policy)
- Transport & application layers at end systems only
- Using router address ; violates host-to-host communication as hosts are hidden
- Controversial

IPv6 Addressing

- 2^{32} — addresses in IPv4 — insufficient space
- claims that address space will never get depleted
- 128 bits IP addresses — 2^{128} addresses $= 2^{96} \times$ IPv4 address space
- According to textbook, no. of grains of sand is of the order of 2^{128}
- IPv4: 12.134.203.142 ⟶ 4 dot-separated decimal no.s
- IPv6: 8 colon-separated hexadecimal no.s
- Most of the address space unused; many zeroes in address

colon hexadecimal notation

FDEC : 0000 : 0000 : 0000 : 0000 : BBFF : 0000 : FFFF

↓ compressed

FDEC : 0 : 0 : 0 : 0 : BBFF : 0 : FFFF

↓ zero compressed

FDEC : : BBFF : 0 : FFFF

all remaing bits zero; double-colon can be used only once

Note:

1) Address of just double colon —— :: is valid (128 bits 0)

2) Leading zeroes can be discarded

FDEC:0098:7600:0321:0ABD::

↓

FDEC:98:7600:321:ABD::

3) Only one double colon per IPv6 address allowed

FDEC:0:0:0:ABCD:0:0:0

↓ any one can be compressed

FDEC::ABCD:0:0:0

MIXED REPRESENTATION

- IPv4 addresses expressed as IPv6 address with leading zeroes
- /60 (prefix) —— CIDR address

Q: Show the unabbreviated colon hex notation for the following IPv6 addresses:

a. An address with 64 0s followed by 64 1s.

b. An address with 128 0s.

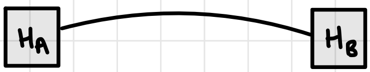
c. An address with 128 1s.

d. An address with 128 alternative 1s and 0s.

a. 0000 : 0000 : 0000 : 0000 : FFFF : FFFF : FFFF : FFFF
b. 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 0000
c. FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF : FFFF
d. AAAA : AAAA : AAAA : AAAA : AAAA : AAAA : AAAA : AAAA

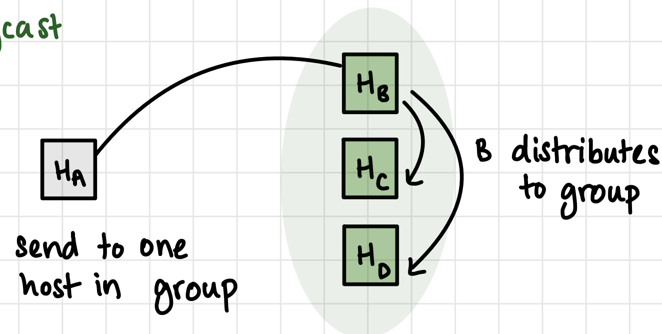
IPv6 Addresses

1. Unicast —— normal IP address - host/router

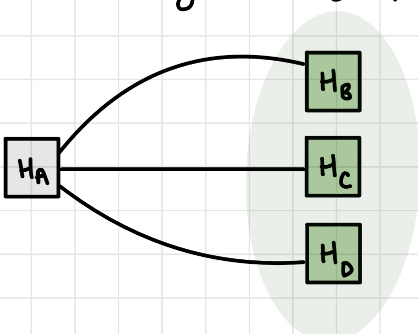


one-to-one

2. Anycast



3. Multicast
   - broadcasting is not there in IPv6 — wastage
   - only multicasting
   - broadcasting in very special cases



one-to-many

ADDRESS SPACE

1. Unicast

- First 3 bits — 001 ⇒ 0010 or 0011 first section bits (prefix 001) for Global Unicast
- In hexadecimal — 2 or 3

(a) Global unicast address

- global public IP, prefix : /3
- 0010 or 0011
- One-one communication between hosts in the internet
- Address block: 3 bits reserved, 125 available ; 2^{125} addresses in block

Global routing prefix	Subnet identifier	Interface identifier
← n →	← m →	← q →
48	16	64

- Only recommendation, not rule

Global Routing Prefix (GRP)

- 3 bits reserved, 45 free (2^{45} values)
- Each of the 2^{45} address blocks provided to ISPs
- One block for one organisation

Global routing prefix	Subnet identifier	Interface identifier
← n →	← m →	← q →
48	16	64

Subnet Identifier

- Per organisation, 2^{16} subnets
- No subnets, all zeroes

Global routing prefix	Subnet identifier	Interface identifier
← n →	← m →	← q →
48	16	64

Interface Identifier

- 2^{64} hosts per subnet

Global routing prefix	Subnet identifier	Interface identifier
← n →	← m →	← q →
48	16	64

(b) Unique Local Unicast

- private IP, not public (LAN)
- Prefix : /7 (1111 110 —— FC or FD)

(c) Link Local Address

- prefix : /10 (1111 1110 10 —— FE8 / FE9 / FEA / FEB)

2. Multicast

- FF00/8 or prefix (1111 1111 ---- ----)
- FF00::/8

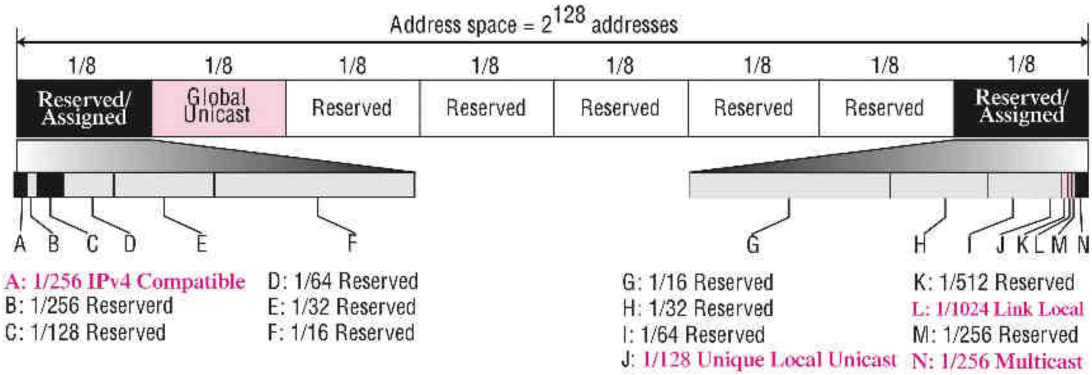
3. Anycast

- contact most reachable host in group

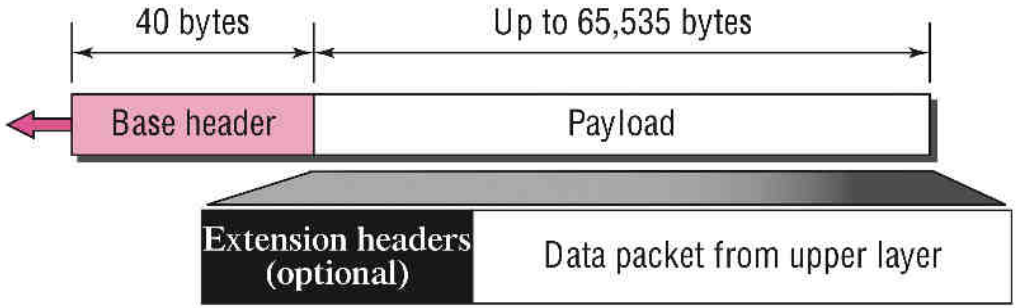
Table 26.1 *Prefixes for IPv6 Addresses*

	Block Prefix	*CIDR*	*Block Assignment*	*Fraction*
1	0000 0000	0000::/8	Reserved (IPv4 compatible)	1/256
	0000 0001	0100::/8	Reserved	1/256
	0000 001	0200::/7	Reserved	1/128
	0000 01	0400::/6	Reserved	1/64
	0000 1	0800::/5	Reserved	1/32
	0001	1000::/4	Reserved	1/16
2	**001**	**2000::/3**	**Global unicast**	**1/8**
3	010	4000::/3	Reserved	1/8
4	011	6000::/3	Reserved	1/8
5	100	8000::/3	Reserved	1/8
6	101	A000::/3	Reserved	1/8
7	110	C000::/3	Reserved	1/8
8	1110	E000::/4	Reserved	1/16
	1111 0	F000::/5	Reserved	1/32
	1111 10	F800::/6	Reserved	1/64
	1111 110	FC00::/7	Unique local unicast	1/128
	1111 1110 0	FE00::/9	Reserved	1/512
	1111 1110 10	FE80::/10	Link local addresses	1/1024
	1111 1110 11	FEC0::/10	Reserved	1/1024
	1111 1111	FF00::/8	Multicast addresses	1/256

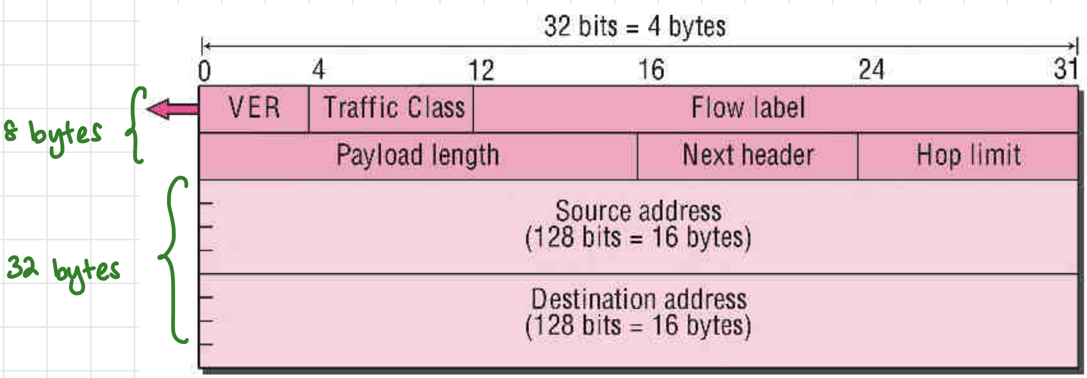
Address Space Allocation



IPv6 Datagram Format



Base Header



- **VER:** version of IP for router to process header (IPv4 and IPv6)

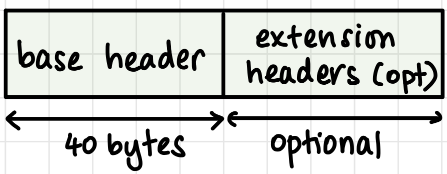
- **Traffic class:** like type of service field in IPv4 (way that data in datagram has to be handled — priority, speed etc)

- **Flow label:** not in IPv4 — labelling of flow of packets if has to be transferred over same link / specific link (eg: audio streaming)

  - Real Time Transport Protocol - RTP

  - Resourse Reservation Protocol - RSVP: reserve link — beyond capacity of IP protocol — works at IP layer with additional authority

- **Base header:** fixed at 40 bytes



- **Next header:** next header info

Table 27.1 *Next Header Codes*

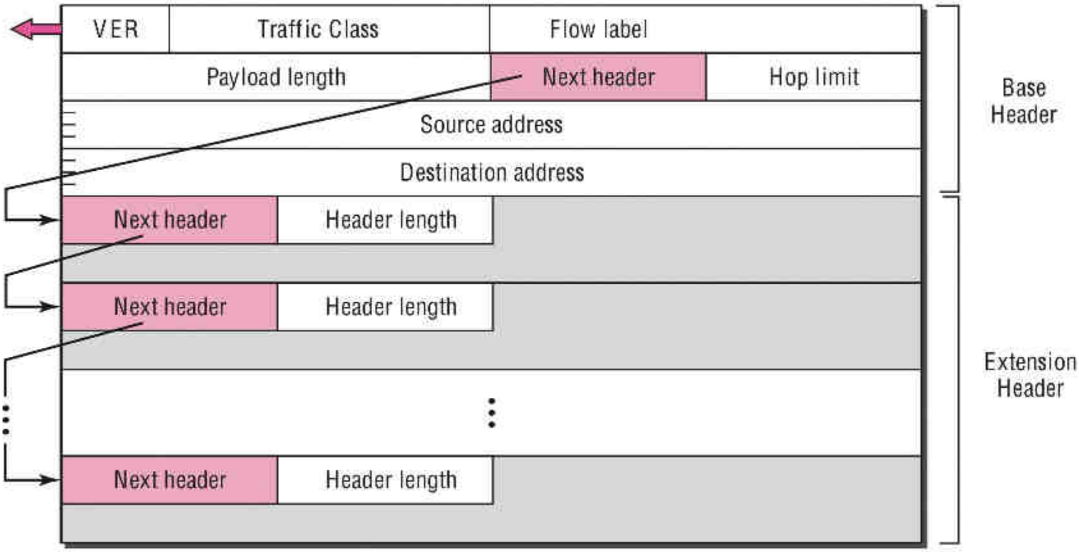
Code	Next Header	Code	Next Header
0	Hop-by-hop option	44	Fragmentation
2	ICMP	50	Encrypted security payload
6	TCP	51	Authentication
17	UDP	59	Null (No next header)
43	Source routing	60	Destination option

IPv4 also (6, 17)

payload contains no header (59)

- Hop limit : like TTL in IPv4 (8-bit hop limit)

Extended Header



Transition from IPv4 to IPv6

- All routers cannot be upgraded simultaneously
- No flag day — internet down day

Transition Strategies

1. Dual Stack
2. Tunnelling
3. Header Translation

1. Dual Stack

- If router capable of IPv6 communication, router sends IPv6 datagrams

- If only IPv4 enabled router encountered, IPv4 communication occurs

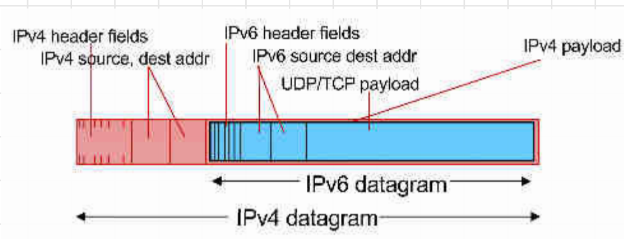
- All IPv6 enabled devices must be capable of IPv4 communication until complete transition occurs

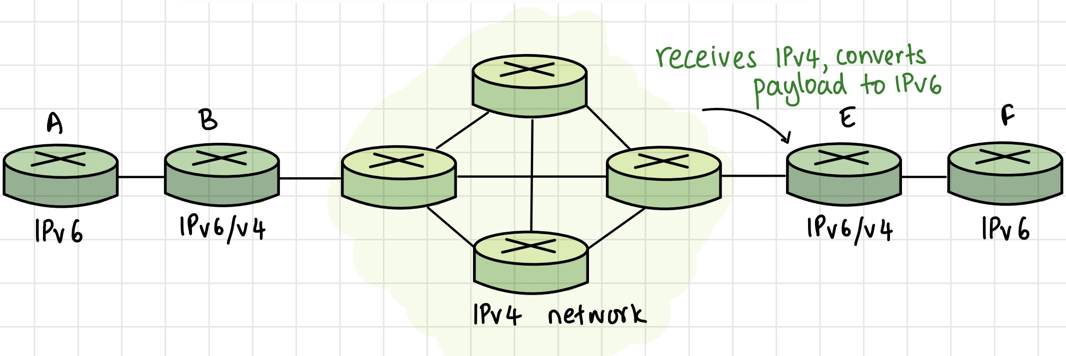
- If DNS server returns IPv4 address, IPv4 communication and if IPv6 address returned, IPv6 communication

2. Tunnelling

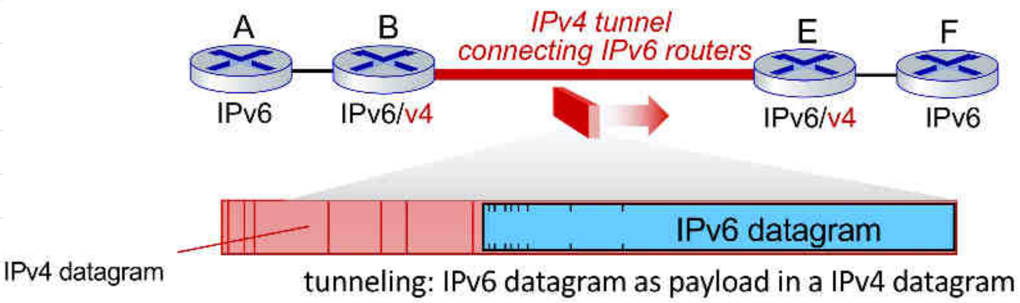
- IPv6 datagram carried as payload of IPv4 datagram

- Encapsulation takes time

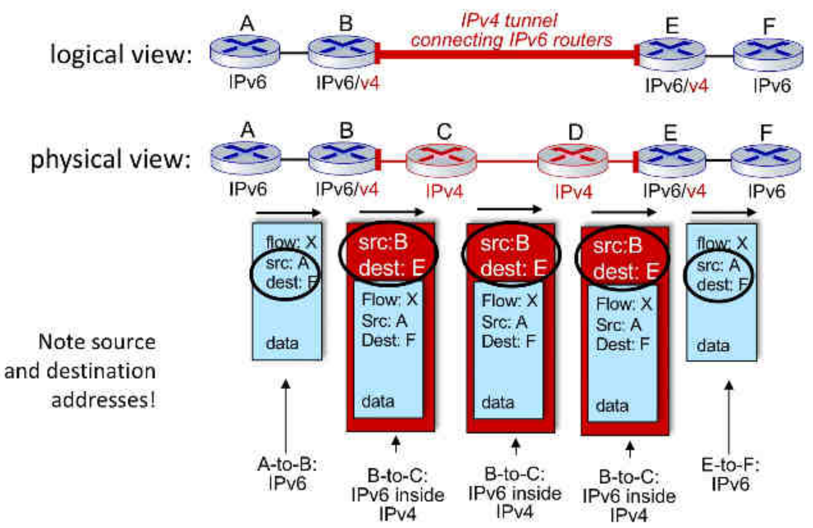




LOGICAL DIAGRAM



Tunnelling - Logical & Physical



Reserved & Assigned Addresses

Unspecified address in IPv6 is ::/128 – reserved and should not be used as destination

8 bits	120 bits
00000000	All 0's
prefix	suffix

Loopback address ::1/128 – never used as destination

8 bits	120 bits
00000000	000000 0001

Embedded IPv4 address — compatible

- CIDR notation — ::/96
- IPv6 → IPv6

96 bits	32 bits
All 0's	IPv4 address

Embedded IPv4 addresses - mapped addresses

- IPv6 → IPv4

80 bits	16 bits	32 bits
All 0's	All 1's	IPv4 address

Private Addressing in IPv6

- Interface identifier — MAC address used
- EVF64 (directly made interface address) or Ethernet MAC (48 bits used, fill 16 bits with specific bits)
- Embedded as whole or part of interface identifier (link level addressing)
- Site level & link level

Unique Local Unicast Block

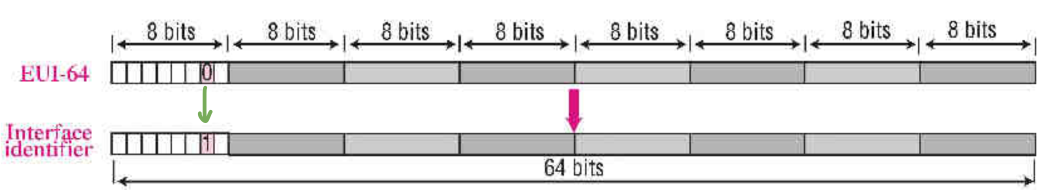
7 bits	0 or 1	40 bits	16 bits	64 bits
1111110		random number	subnet ID	interface ID

Link Local Block

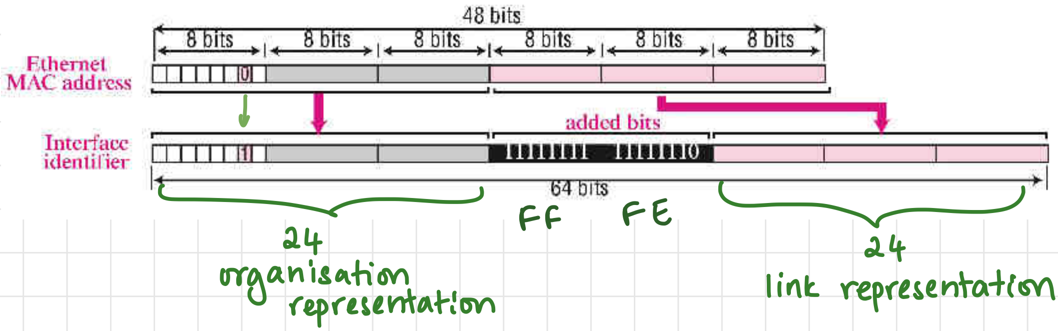
10 bits	38 bits	16 bits	64 bits
1111111010	All 0's	All 0's	interface ID

Global Unicast Address

mapping for EUI-64



Mapping for Ethernet MAC (48-bits)



Q: Find the interface identifier if the physical address in the EUI is $(F5-A9-23-EF-07-14-7A-D2)_{16}$ using the format we defined for ethernet addresses.

7^{th} bit from 0 → 1

1111 0101 ⟶ 1111 0111 ⟶ F7

F7A9:23EF:714:7AD2

Q: Find the interface identifier if the ethernet physical address is $(F5-A9-23-14-7A-D2)_{16}$ using the format we defined for ethernet addresses.

7^{th} bit from 0 → 1

1111 0101 ⟶ 1111 0111 ⟶ F7

After 3^{rd} byte, FFFE padding

F7A9:23FF:FE14:7AD2

Q: An organisation is assigned the block 2000:1456:2474/48. What is the CIDR notation for the blocks in the first and second subnets in this organisation?

Subnet identifier 1: 0000_{16}
Subnet identifier 2: 0001_{16}

Blocks: 2000:1456:2474:0000/64

2000:1456:2474:0001/64

Q: An organisation is assigned the block 2000:1456:2474/48. What is the IPv6 address of an interface in the third subnet if the IEEE physical address of the computer is $(F5\text{-}A9\text{-}23\text{-}14\text{-}7A\text{-}D2)_{16}$?

Third subnet — 2000:1456:2474:0002/64

Interface identifier — F7A9:23FF:FE14:7AD2

2000:1456:2474:0002:F7A9:23FF:FE14:7AD2

AUTO CONFIGURATION

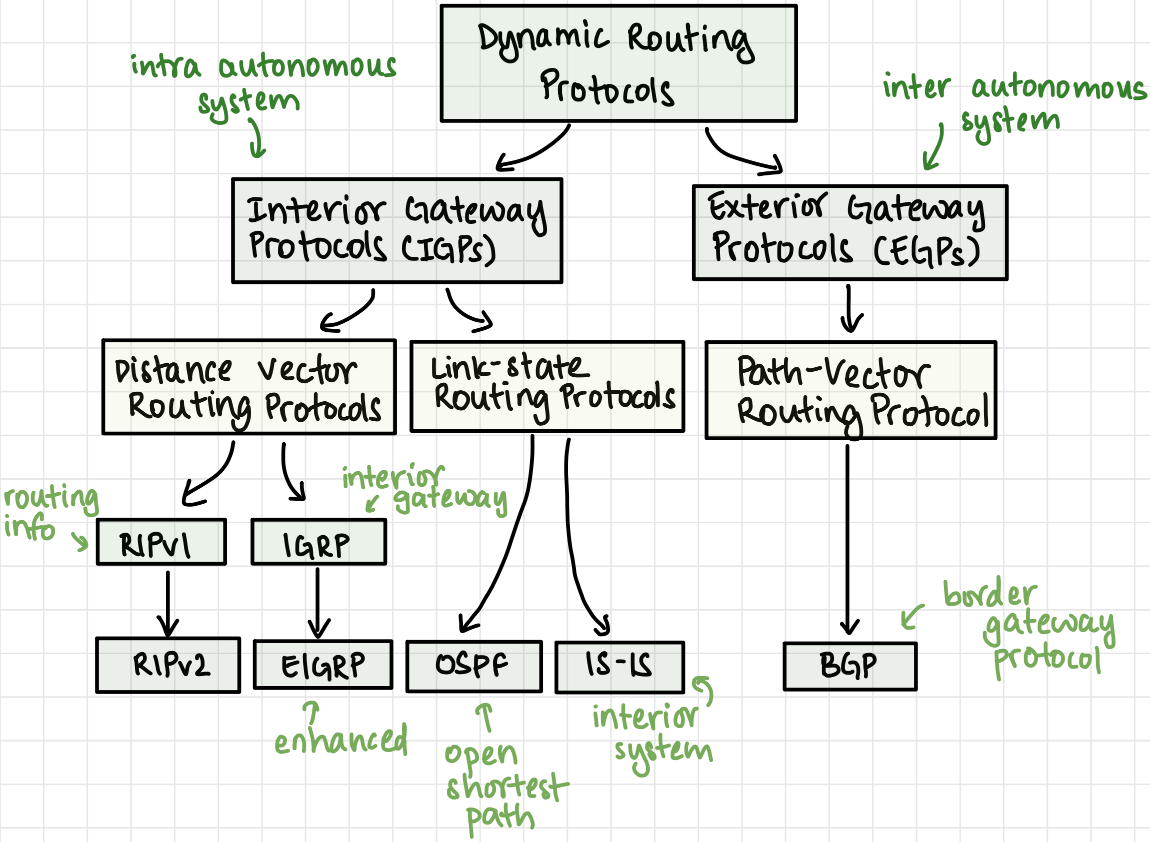
- DHCP can be used to allocate IPv6 address to host
- Host can also configure itself - auto configuration
- Host creates link local address for itself
  10-bit prefix (1111 1110 10) + 54 zeroes + 64-bit interface ID
- Check if link local address is unique
- Sends neighbour solicitation message and waits for neighbour advertisement message
- If it fails, DHCP is used to configure address (as link local address is not unique)
- Otherwise, address used locally; global — host sends router solicitation message and if router present, returns router advertisement message with global unicast prefix & subnet ID

Renumbering

- change ISP — prefix must change
- Router must advertise new prefix
- Neighbours know old prefix, router table has old prefix
- Router allowed to use both prefixes for a while (problems with implementation)
- Solution: updated DNS and ICMP protocols

ROUTING ALGORITHMS

- Routing table gets too big for all IP addresses and dynamic links to manually update
- Dynamic routing algorithms update routing table entries



Distance Vector vs Link State

	Distance Vector	Link State
Primary principle	Send entire routing table to its neighbours	Only provides link state information
Learning about network	Learn about network only from neighbours	Learn about network from all routers
Building the routing table	Based on inputs from only neighbours	Based on complete database collected from all routers
Advertisement of updates	Sends periodic updates every 30-90 seconds – **Broadcasts updates**	Use triggered updates, only when there is a change – **Multicasts updates**
Routing loops	Vulnerable	Less prone to routing loops
Convergence (stabilization)	Slow	Fast
Resources	Less CPU power and memory	More CPU power and memory required Cost
Cost		More than distance vector
Scalability		More scalable than distance vector
Examples	RIP, IGRP	OSPF, IS-IS