The Internet Protocol (IP)

What problem are we trying to solve?

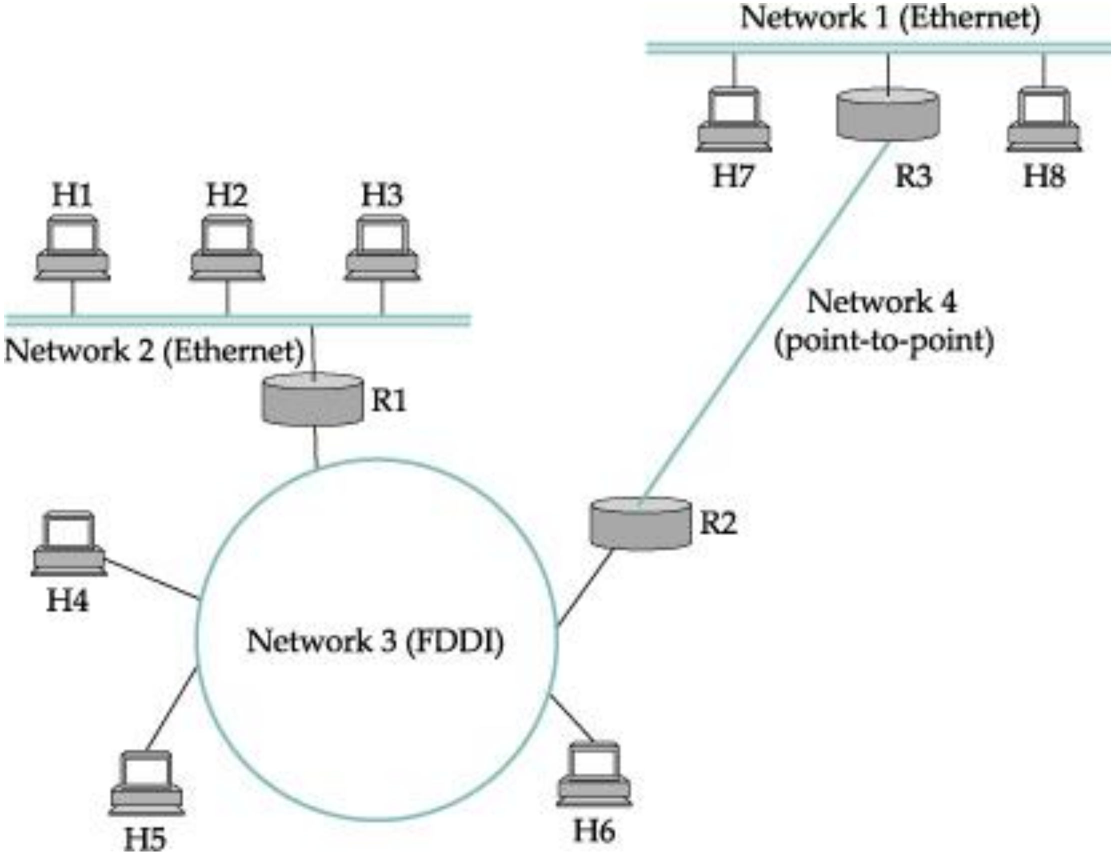
Since there are numerous DL technologies and protocols, an internetwork is going to need to pass data between subnetworks with different:

- protocols
- addressing schemes
- speeds
- …

How can we manage these problems efficiently in large internets?

Example

- Terms
  - Networks, internetwork
  - Router, gateways



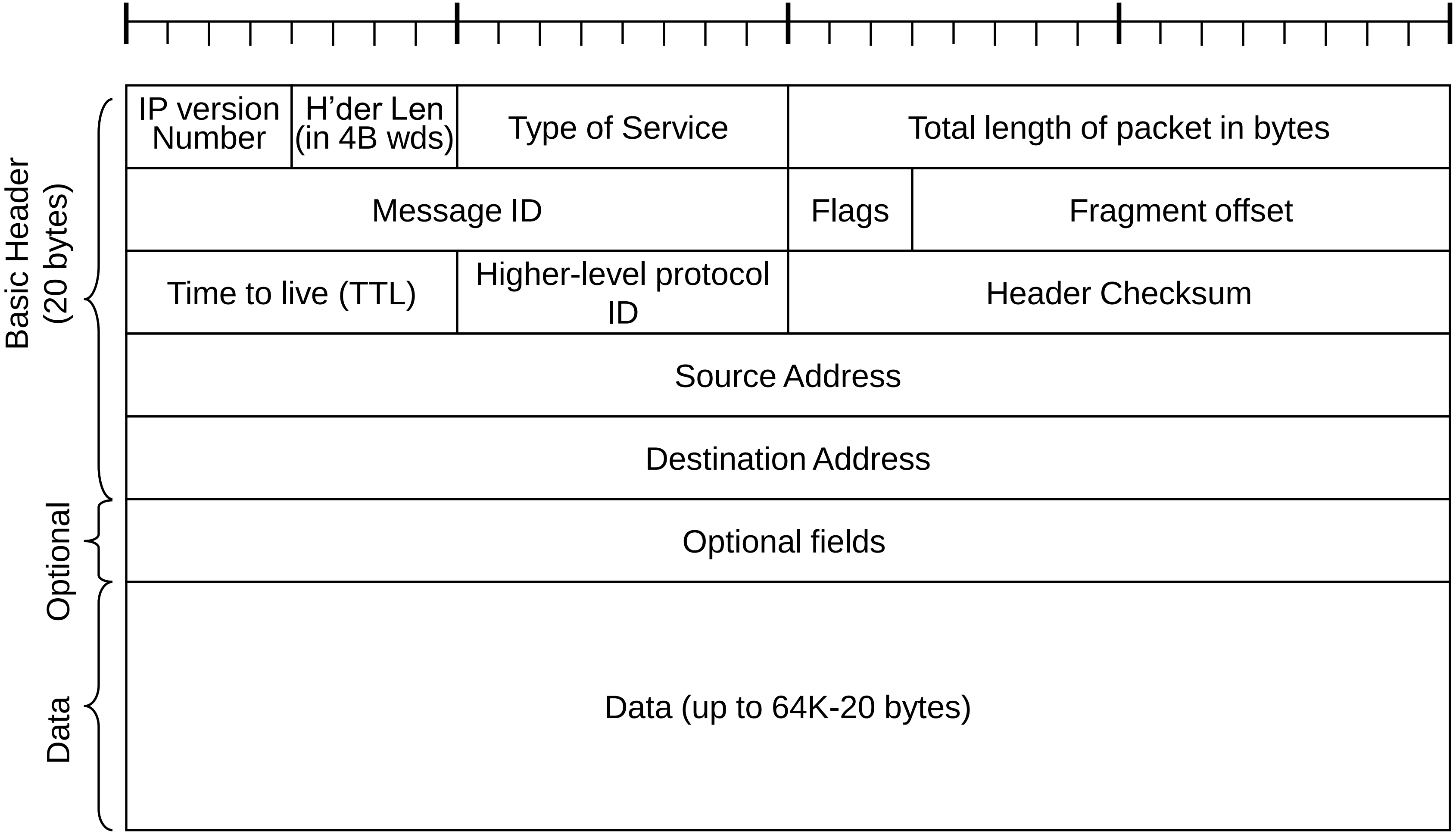
What is IP?

- Most widely applied internetworking protocol
- The L3 protocol of the Internet
- Addressing scheme
- Best-effort (“unreliable”), why?

- Two versions we care about:
  - IPv4 -- the version currently in use (mostly)
  - IPv6 -- the next version

IPv4

IPv4 packet format



Basic Header (20 bytes)

IP version Number | H'der Len (in 4B wds) | Type of Service | Total length of packet in bytes

Message ID | Flags | Fragment offset

Time to live (TTL) | Higher-level protocol ID | Header Checksum

Source Address

Destination Address

Optional

Optional fields

Data

Data (up to 64K-20 bytes)

Notes on some IPv4 header fields

Header Length: Measured in 32-bit wds. Minimum is 5.

Type of service: Options for how IP will treat the packet (will discuss when we get to QoS)

Message ID: Identifies this packet with a particular message between the source and destination. The combination of Source_address, Dest_address, Message_ID, Protocol, and Fragment_number identify this packet uniquely.

Flags: Only 2 of 3 bits defined. Used to support fragmentation (later chart).

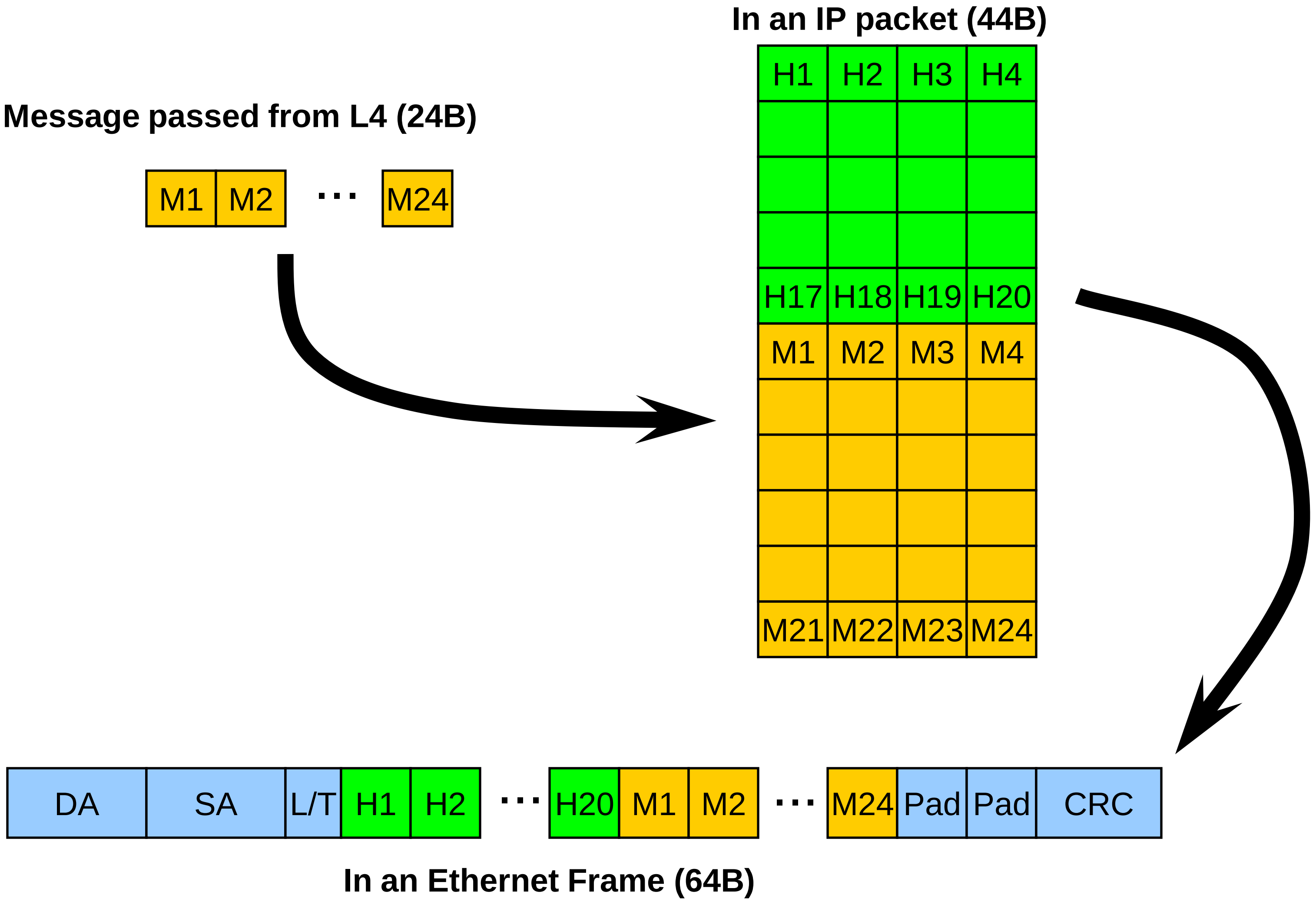
TTL: Used to ensure that packets will eventually die if not delivered. Originally intended to measure life in seconds; is processed as a hop count (every router decrements TTL until it reaches 0).

Protocol: Identifies the Transport-level protocol (usually TCP or UDP).

Options: Used by the sender to request network services (padded to be a multiple of 32 bits)

Data: The total packet length including header and options can be 64KB.

IP in the protocol stack



Message passed from L4 (24B)

M1 M2 ··· M24

In an IP packet (44B)

H1	H2	H3	H4
H17	H18	H19	H20
M1	M2	M3	M4
M21	M22	M23	M24

DA SA L/T H1 H2 ··· H20 M1 M2 ··· M24 Pad Pad CRC

In an Ethernet Frame (64B)

IP Fragmentation

IP Packet Fragmentation

- Assume we send an IP packet through a subnetwork in which the frame payload size is smaller than the packet size

- We could design to do either:

  (1) L2 Fragmentation: Divide the IP packet among frames when it enters the subnetwork, then recombine them when it leaves the subnetwork

  Problems:

  - May introduce high delay by repeatedly fragmenting and re-assembling the same packet in different subnetworks
  - Have to wait for all frames at the exit of each subnetwork

  (2) L3 Fragmentation: When entering the subnetwork, divide the packet into smaller IP-formatted packets. Re-assembly is doen at the receiver.

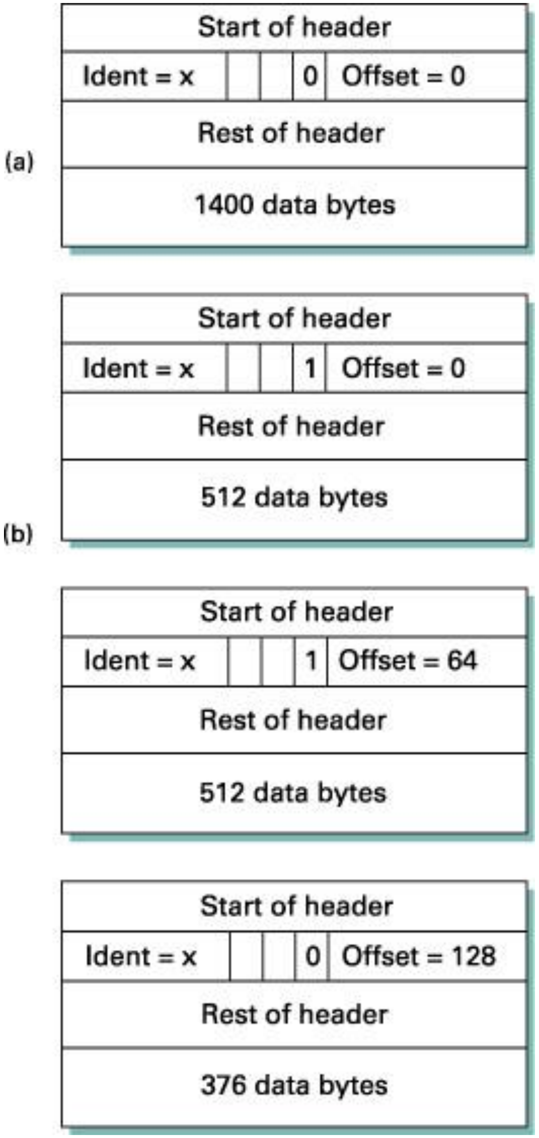
Header fields supporting fragmentation

- Source Address
- Destination Address
- Message ID

Uniquely identify the message that the fragment is part of

- Flags:
  - 0
  - 1
- Fragment Offset -- The offset (in bytes) of the data in this fragment packet referenced to the start of the data in the original packet

Header Fields



(a)

Start of header

Ident = x | | | 0 | Offset = 0

Rest of header

1400 data bytes

(b)

Start of header

Ident = x | | | 1 | Offset = 0

Rest of header

512 data bytes

Start of header

Ident = x | | | 1 | Offset = 64

Rest of header

512 data bytes

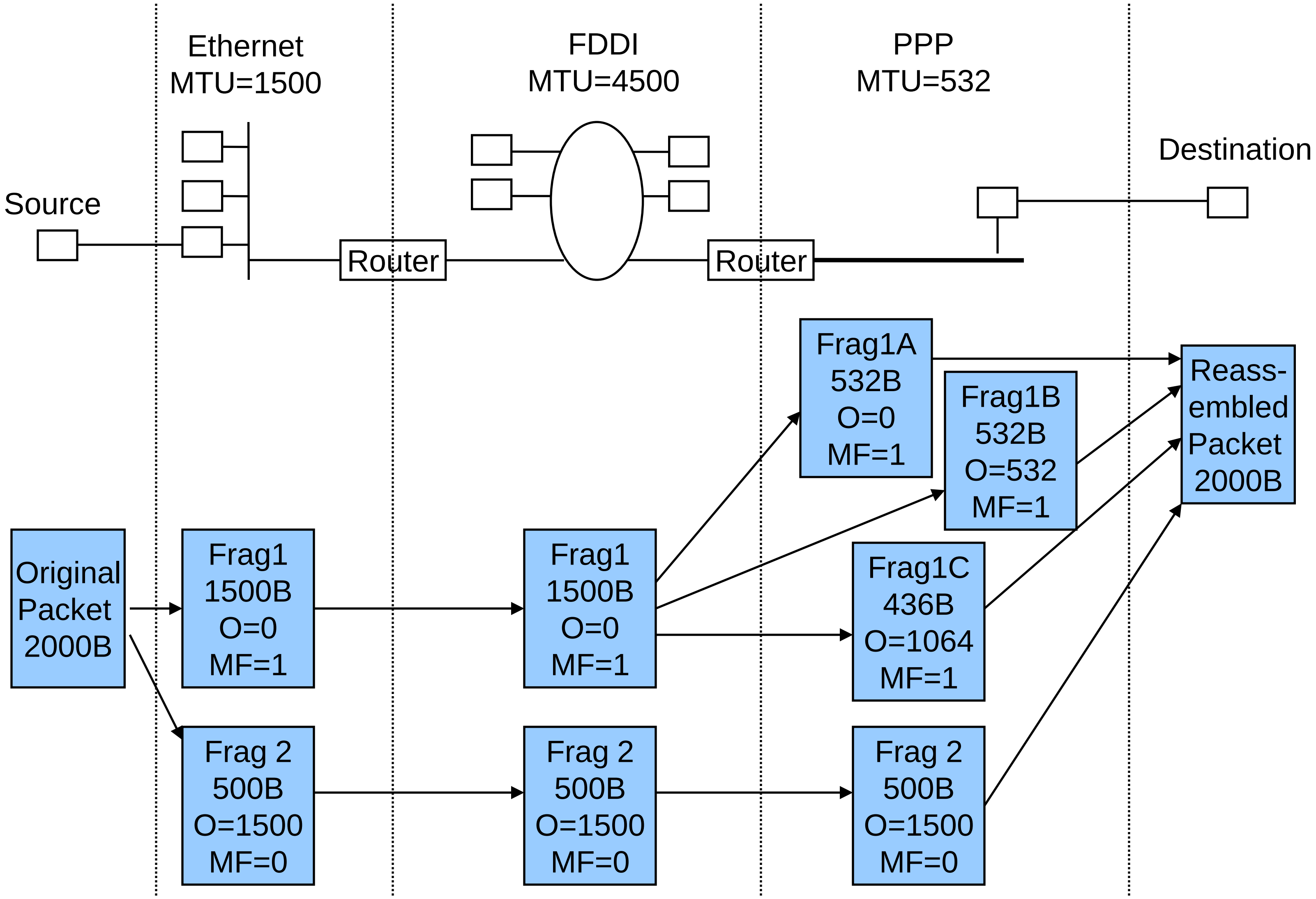
Start of header

Ident = x | | | 0 | Offset = 128

Rest of header

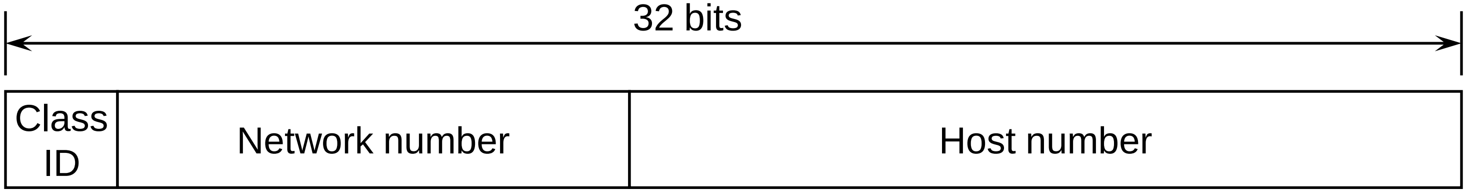
376 data bytes

Fragmentation Example



IPv4 addresses

General format:



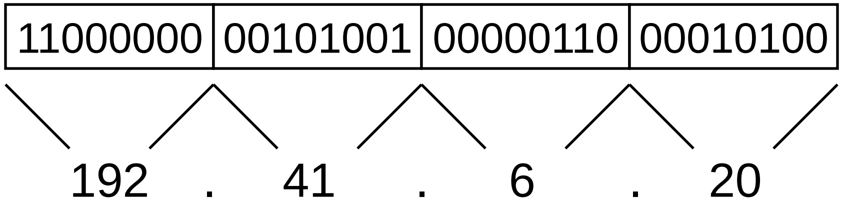
32 bits

Class ID	Network number	Host number

How they are usually written and talked about:

Dotted decimal notation: Express each byte as its equivalent in decimal.

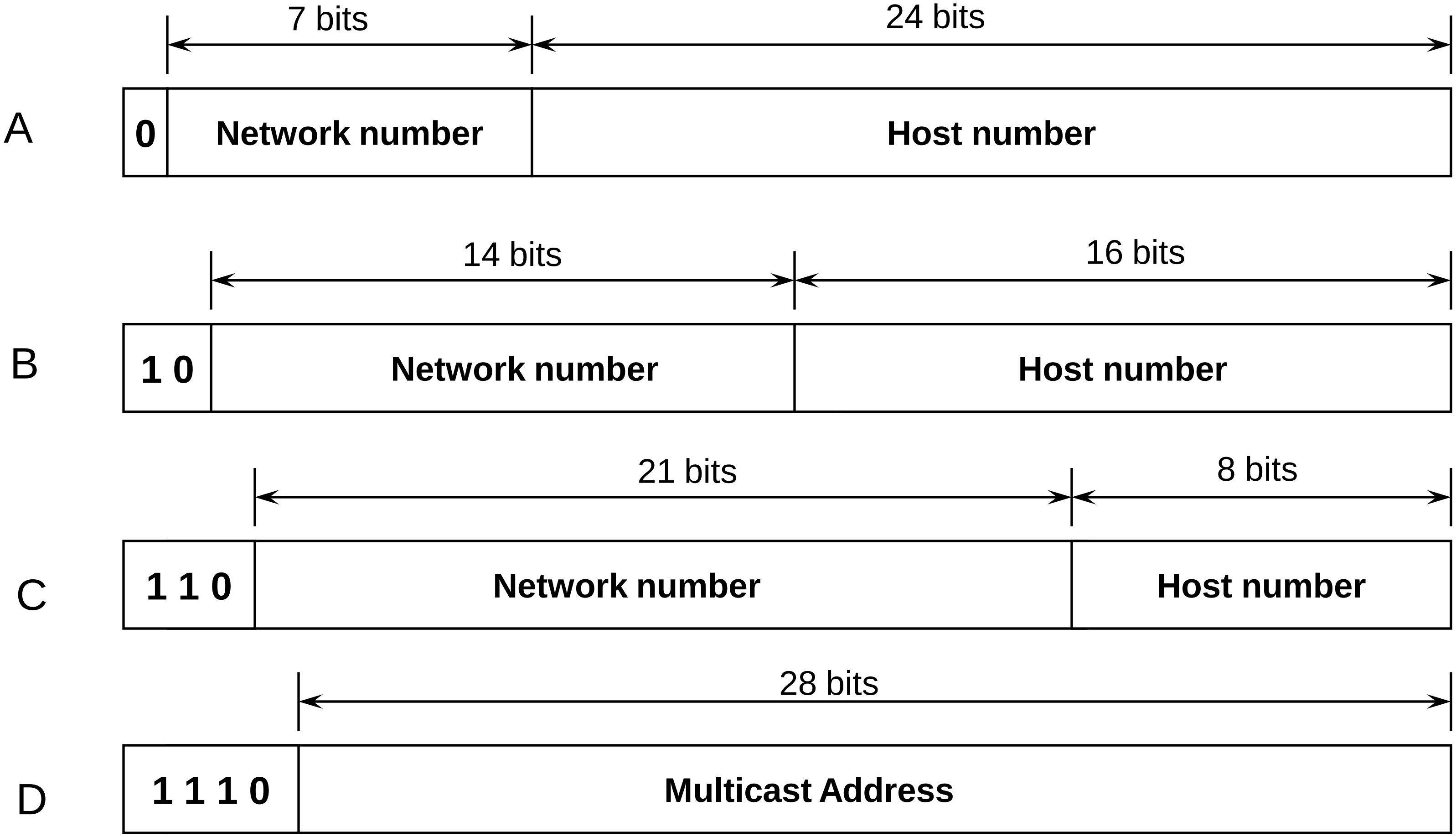
Example:



11000000	00101001	00000110	00010100
192	41	6	20

192 . 41 . 6 . 20

IPv4 address formats (“classful” addressing)



A

7 bits

24 bits

0 | Network number | Host number

B

14 bits

16 bits

1 0 | Network number | Host number

C

21 bits

8 bits

1 1 0 | Network number | Host number

D

28 bits

1 1 1 0 | Multicast Address

Note: Class E(“11110”) is reserved for future use.

IPv4 addresses

Class	Format (when reading in dotted decimal)	Range of Unreserved Addresses	Approximate number of networks/hosts
A	N.H.H.H	1.0.0.0 to 126.255.255.255	126 / 16M
B	N.N.H.H	128.0.0.0 to 191.255.255.255	16K / 64K
C	N.N.N.H	192.0.0.0 to 223.255.255.255	2M / 256

Some special reserved addresses:

All zeroes:	This host
Network=0 w/ host #:	The indicated host on this network
All ones:	Broadcast on this network.
Network # w/ host=all ones:	Broadcast on the indicated network
Network=127	Loopback

Mapping IP addresses to L2 devices

IP addressing over MAC addresses

- IP addresses are “virtual” addresses assigned to a device. They do not relate to the device’s “real” address (its MAC address).

- When an IP packet arrives at its destination subnetwork, it needs to be delivered to the connected host having the specified IP address. But in most multidrop subnetworks (e.g, Ethernet), we need to know the MAC address -- the IP address does no good.

- This means that the subnetwork needs a system for translating IP addresses into MAC addresses.

The Address Resolution Protocol (ARP)

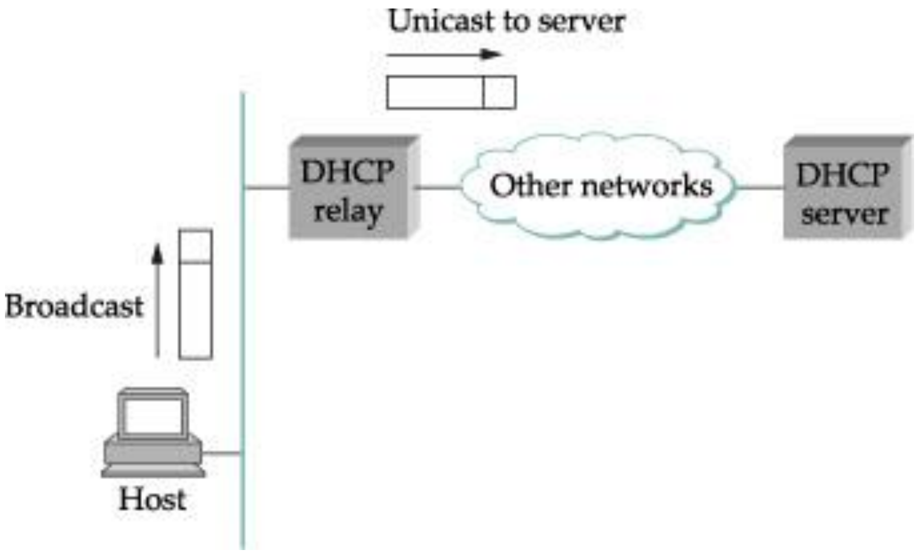
- Each host on the multidrop subnetwork maintains a table of the IP address and MAC address of each node on the subnetwork

- When a host wants to send a packet:
  - Check cache
  - No mapping, then invoke ARP
  - Broadcasting the target IP address, host’s IP address and MAC address
  - Each host checks its IP address
  - Match, send a response

Dynamic Host Configuration Protocol

DHCP

- DHCP server
  - A pool of addresses
- Discovery message
  - Broadcast



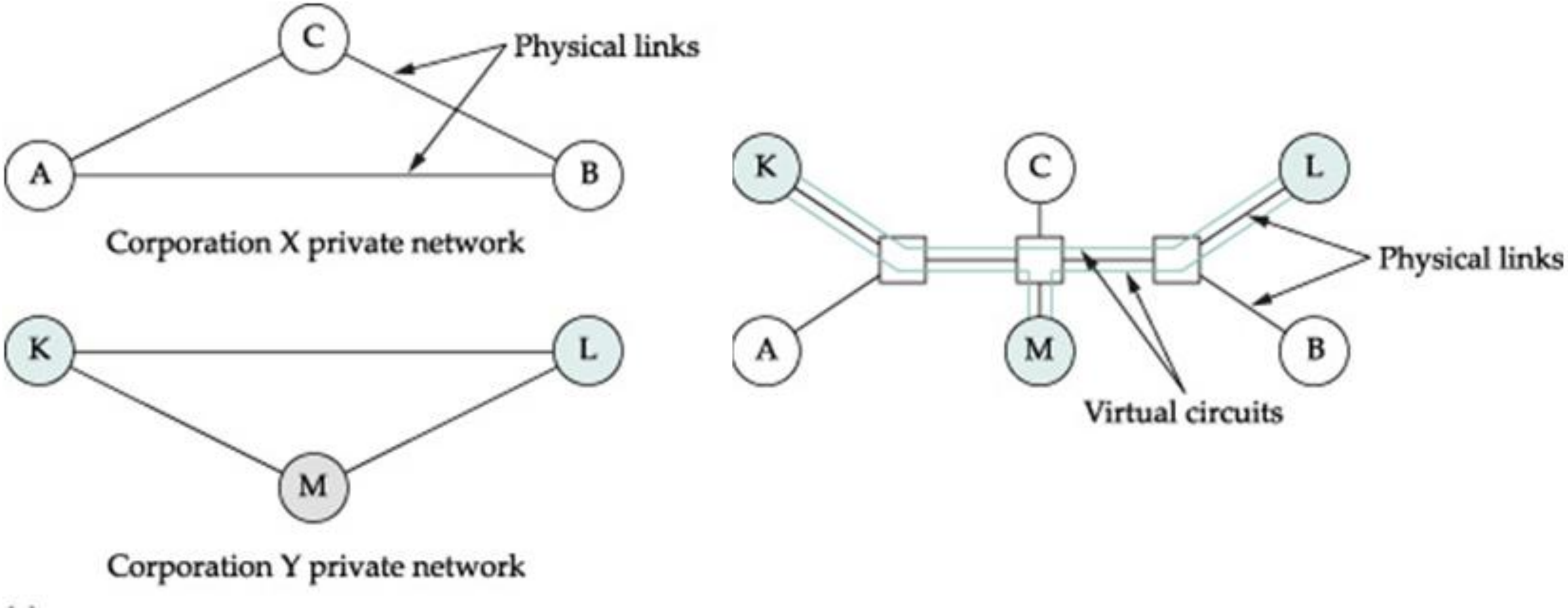
Internet Control Message Protocol

ICMP

- ICMP is actually an integral part of IP
- ICMP code
  - 0 = net unreachable
  - 1 = host unreachable
  - 2 = protocol unreachable
  - 3 = port unreachable
  - 4 = fragmentation needed and DF set
  - 5 = source route failed

Virtual Private Network

VPN



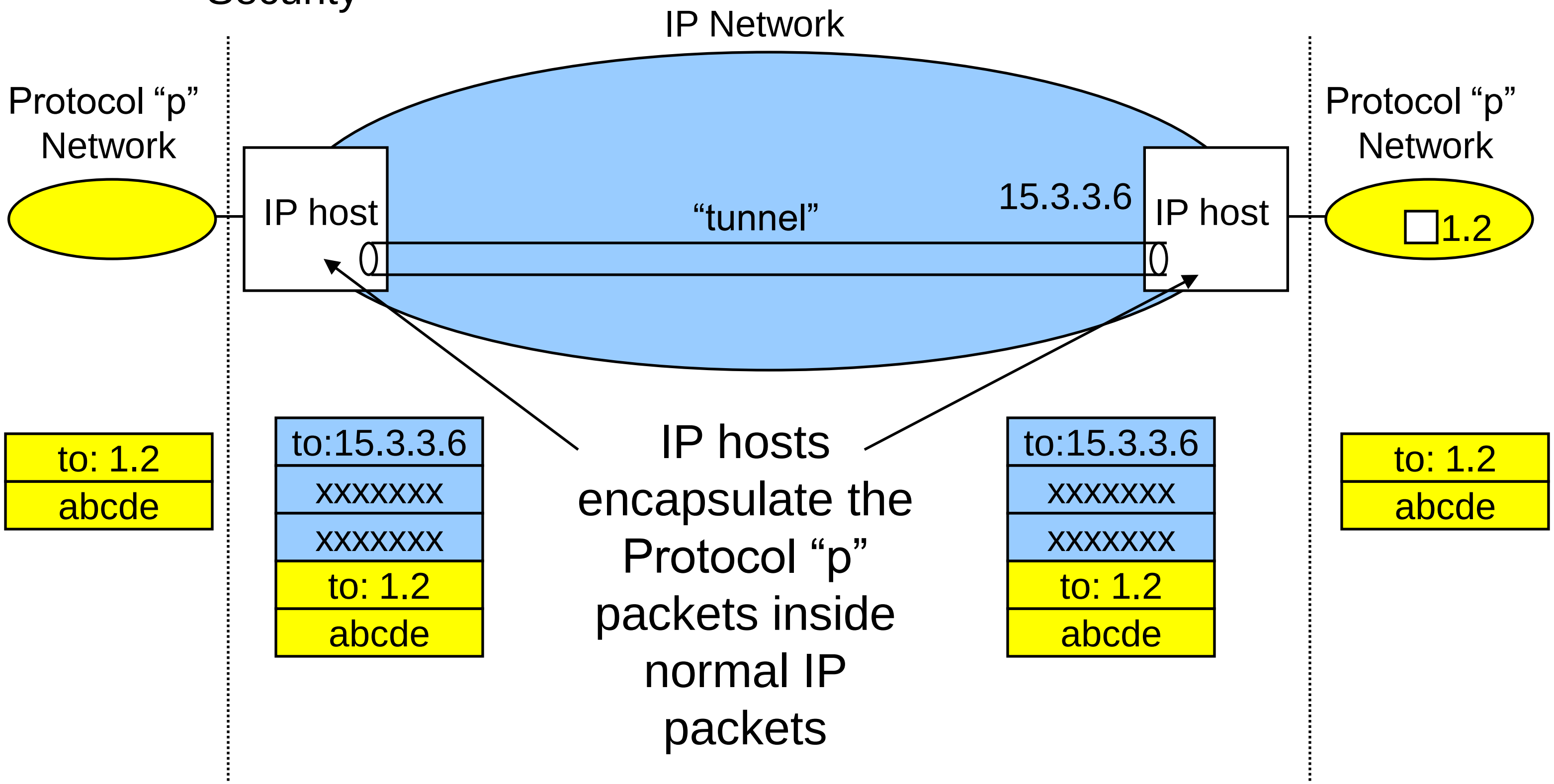
Corporation X private network

Corporation Y private network

IP tunneling

Sometimes, we want to set up a virtual point-to-point link across an IP internet

- Make a virtual “Direct Connection”
- Redirect traffic to other addresses
- Use non-IP protocols
- Security



to: 1.2
abcde

to:15.3.3.6
xxxxxxx
xxxxxxx
to: 1.2
abcde

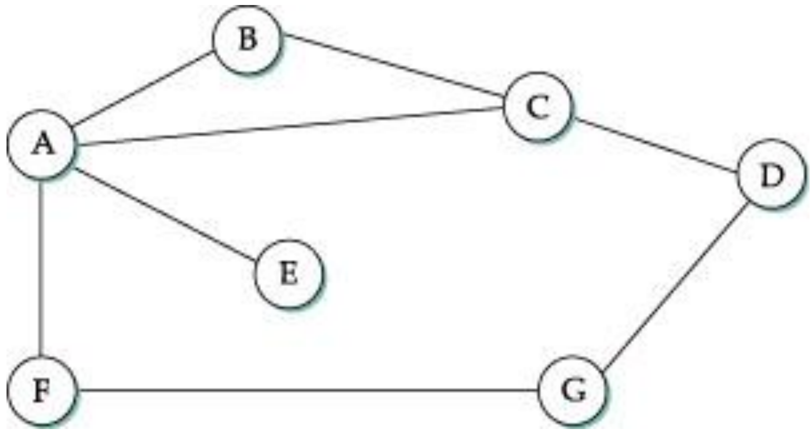
IP hosts encapsulate the Protocol “p” packets inside normal IP packets

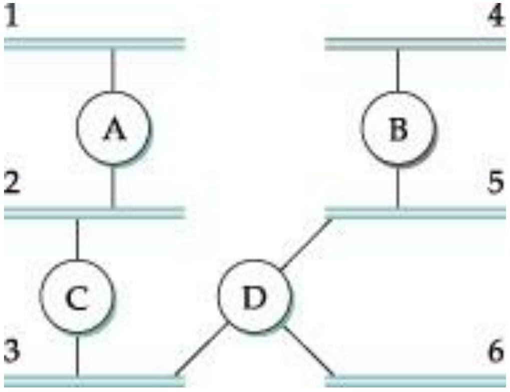
to:15.3.3.6
xxxxxxx
xxxxxxx
to: 1.2
abcde

to: 1.2
abcde

Routing

Distance Vector & Routing Information Protocol (RIP)





Step 1: Directly connected =1, otherwise = ∞

Step 2: Send message to direct neighbors its personal list of distances

Repeat Step 2, until convergence

Periodic update & triggered update

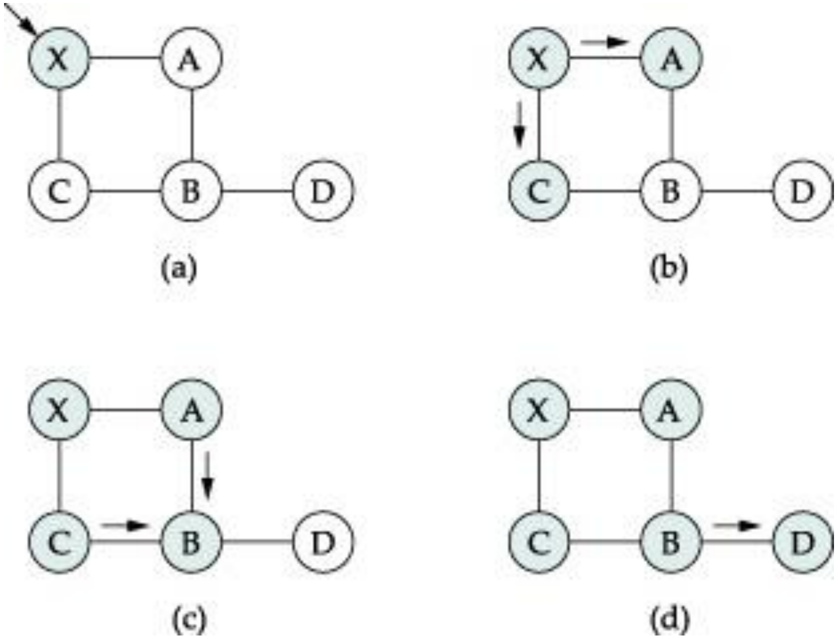
Count to infinity problem

Link State &
Open Shortest Path First Protocol (OSPF)

- Each node knows the state of the link to its neighbors and cost of each link
- Reliable dissemination of link-state information

OSPF (2)

- Reliable flooding
  - ID of the node that created the LSP
  - A list of directly connected neighbors and cost
  - Sequence number
  - TTL



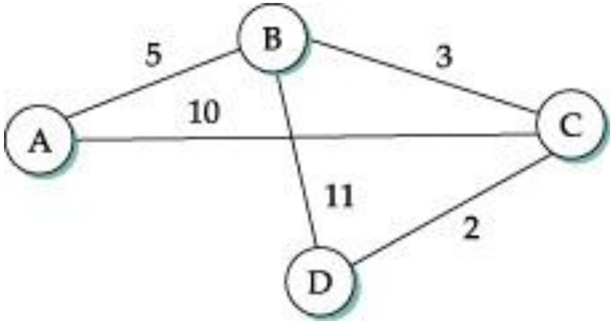
(a) (b) (c) (d)

OSPF (3)

- Check if the copy of LSP exists
- If yes, compare the sequence numbers
- Design goals
  - Reduce overhead (long timer)
  - Sequence numbers
  - TTL

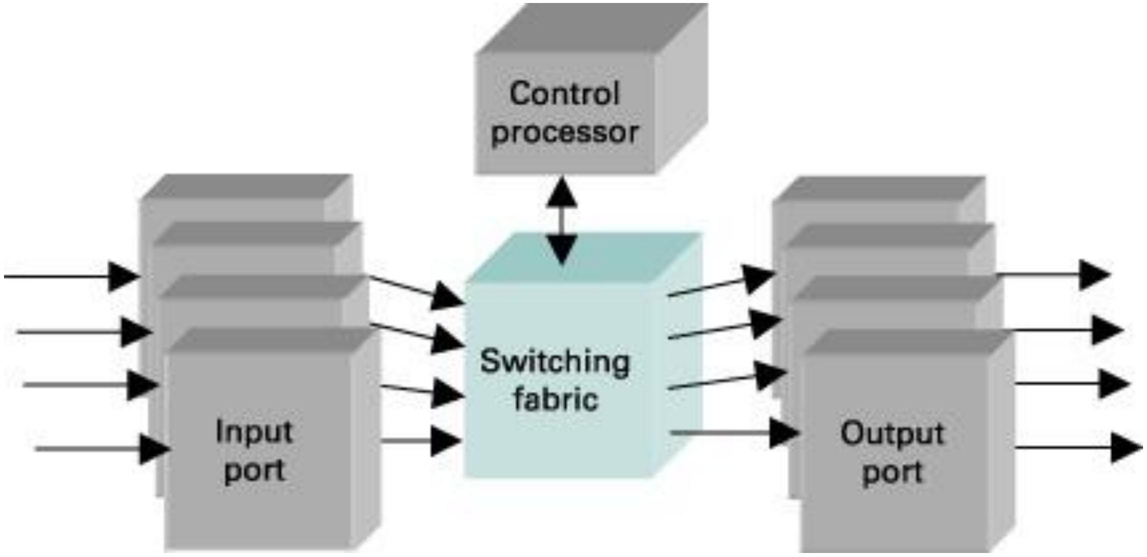
OSPF (3)

- Route calculation, pp281
- Properties
  - Stabilize quickly
  - The amount of information stored can be large
- Authentication
- Additional hierarchy (*area*)
- Load balance (assign cost to links)



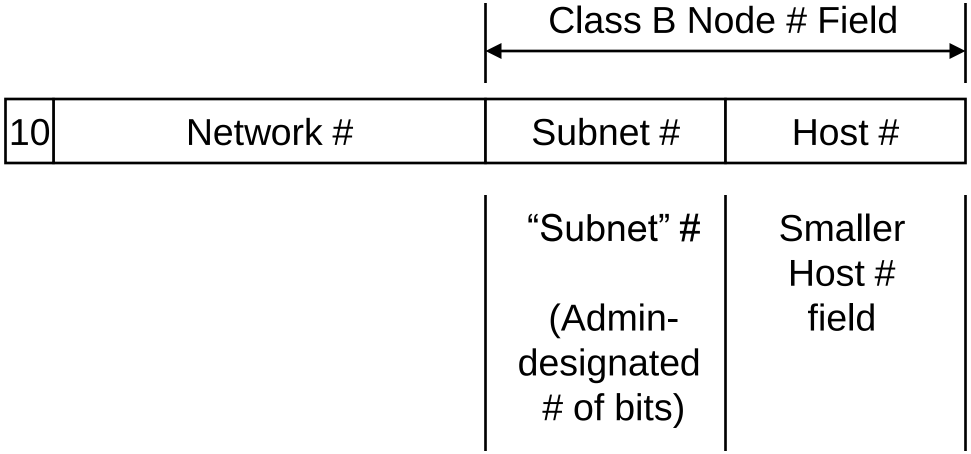
Router Implementation

- Handle variable-length packets
- Packets per second (pps)
- Centralized vs. distributed



Subnetting

- The idea:
  - Introduce a new level of hierarchy by using part of Host Number field as a “subnet” number
  - This lets us share a single Network # across several physical nets



Class B Node # Field

10 | Network # | Subnet # | Host #

“Subnet” #

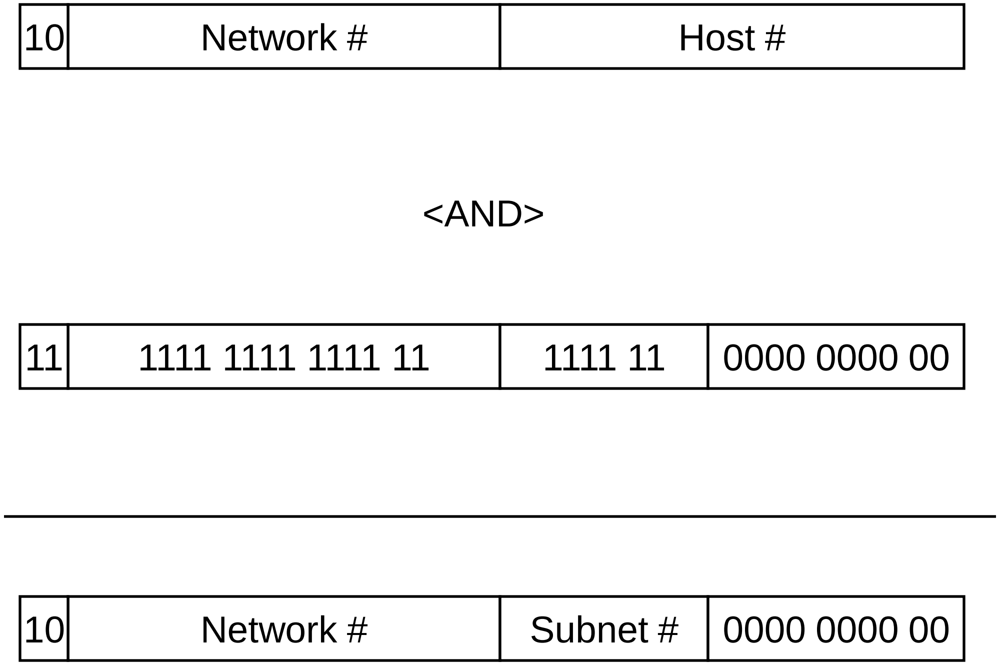
(Admin-designated # of bits)

Smaller Host # field

Addressing with Subnetting

Routed as normal
Class B address
outside of subnetted
area

Inside subnetted
area, subnet # is
determined using
"Subnet Mask"



10	Network #	Host #

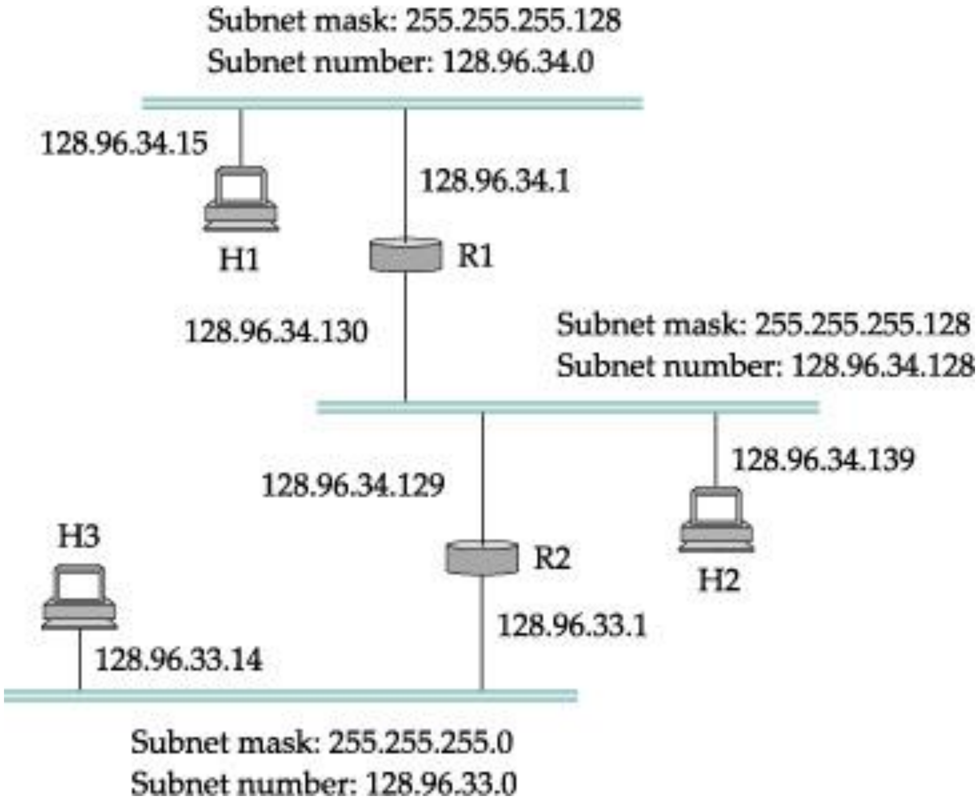
<AND>

11	1111 1111 1111 11	1111 11	0000 0000 00

10	Network #	Subnet #	0000 0000 00

Note: Host number is obtained by ANDing address with Complemented Subnet Mask

Example subnetting


Subnet mask: 255.255.255.128
Subnet number: 128.96.34.0
128.96.34.15
128.96.34.1
H1
R1
128.96.34.130
Subnet mask: 255.255.255.128
Subnet number: 128.96.34.128
128.96.34.139
128.96.34.129
H3
R2
H2
128.96.33.1
128.96.33.14
Subnet mask: 255.255.255.0
Subnet number: 128.96.33.0

Some notes on subnetting

- Subnets will usually be physically near to each other, since all their traffic will be routed to the same router

- Subnet masks are often described by the number of 1's (e.g, 128.96.*/24)

- Subnet masks don't necessarily have contiguous 1's, but anything else is confusing

Another approach to extending IPv4

- Subnetting subdivides Class B address spaces to form subnets that lie in between Class C and Class B in the hierarchy

- Another way we could achieve the same end is to combine contiguous Class C address spaces

Classless addressing (“Supernetting”)

Classless
Inter-
Domain
Routing

CIDR example

- You have a network with 16*254 hosts.
- To conserve Class B space, assign 16 contiguous Class C networks (e.g, 192.4.6.* -> 192.4.21.*)
- Some number of the high-order bits will be the same (for the example, all addresses start with 1100 0000 0000 0100 000 – 19 bits are the same)
- We can think of this as a new type of network with a 19-bit network number

High-order 19 bits	Low-order 13 bits
1100 0000 0000 0100 000	Node # (or subnetting)

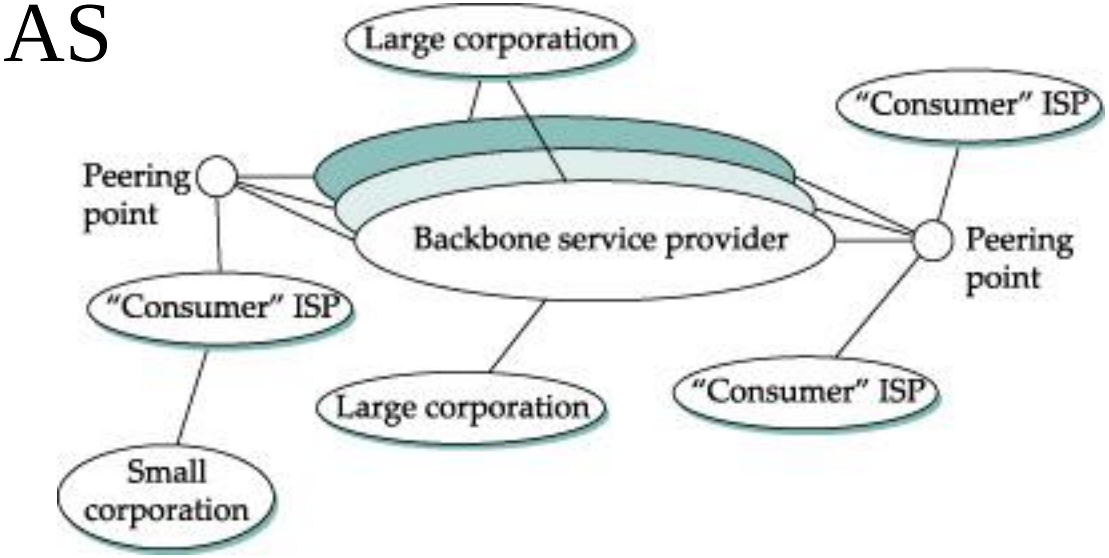
- Anywhere from 4 to 30 bits could be used as determined by the number of Class C’s that are combined

A complication with CIDR

- Since network numbers do not occupy fixed fields, backbone routers must be able to interpret the CIDR encoding
- This can be confusing since we may have the same high-order bits for two different CIDR nets:
  - 171.69/16 and 171.69.10/24 can be two different networks
- For routing, use the “longest match” principle – choose the network that matches the most high-order bits of the IP address:
  - 171.69.10.5 -> 171.69.10/24
  - 171.69.20.5 -> 171.69/16

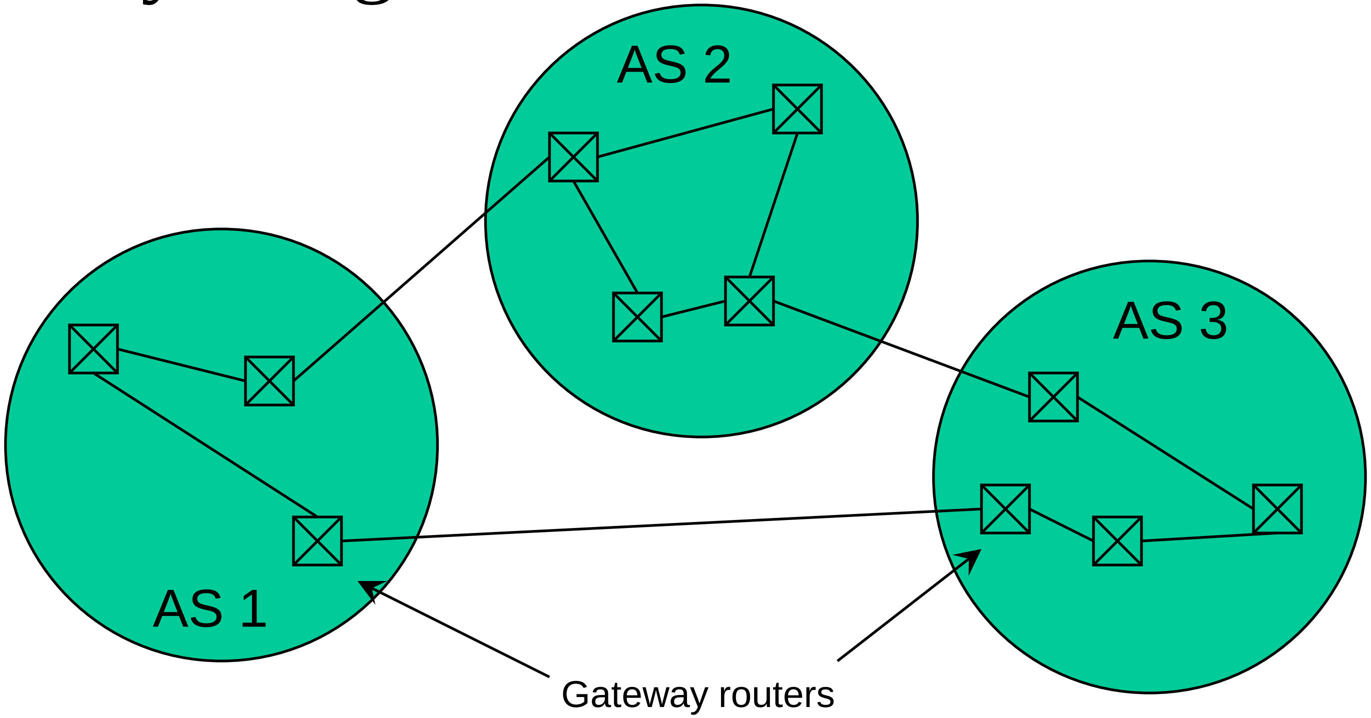
Internet Structure

- Sub AS
- Multihomed AS
- Transit AS



Autonomous Systems (AS)

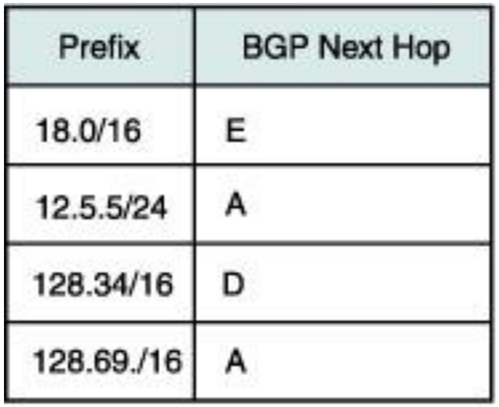
- aka “Routing Domains”
- Large networks are divided into AS’s, usually along administrative boundaries



Border Gateway Protocol (BGP)

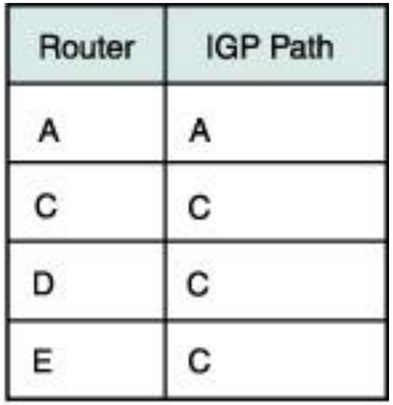
- Challenges
  - Scalability
  - Reachability (impossible to calculate path cost)
  - Trust
- Use “BGP speaker” to exchange reachability information
  - Avoid loops
  - Withdrawn route
- Border gateways

Inter-domain and Intra-domain Routing



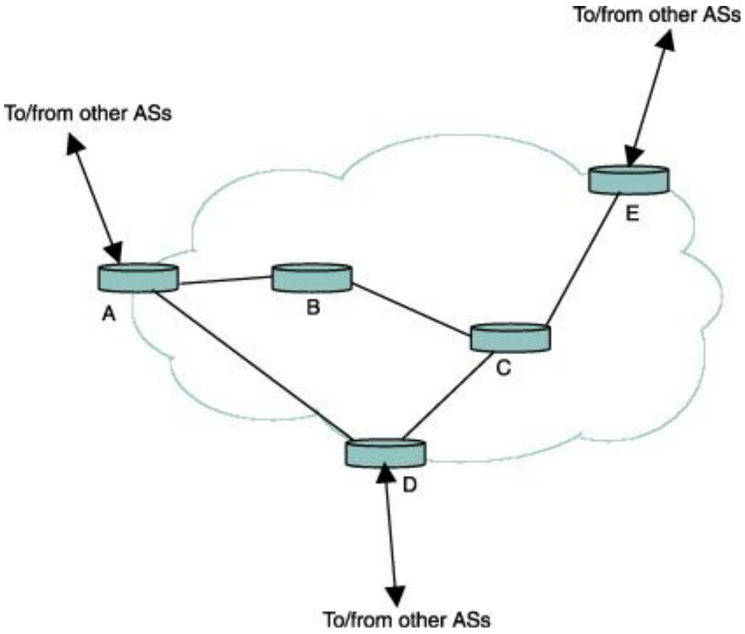
Prefix	BGP Next Hop
18.0/16	E
12.5.5/24	A
128.34/16	D
128.69./16	A

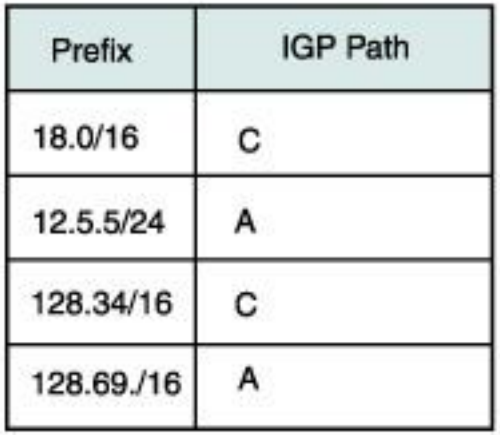
BGP Table for the AS



Router	IGP Path
A	A
C	C
D	C
E	C

IGP Table for Router B





Prefix	IGP Path
18.0/16	C
12.5.5/24	A
128.34/16	C
128.69./16	A

Combined Table for Router B

IPv6

IPv6 (“IPng”)

- Intended to be a long-term fix
- Goals:
  1. Extend the address space
  2. Some additional feature (QoS support, security support, autoconfiguration, support mobile host)
  3. Improved performance
  4. Transition

- Characteristics:
  - 128-bit addresses (3.4 x 10^{38} hosts max) (1,500 per square foot)
  - Classless addressing
  - Less complicated packet format than v4 (7 vs 13 header fields)

IPv6 addresses

- Address prefix, pp320

- No longer using dotted decimal. Changed to hex format:
  - Ex: 8000:0000:0000:0000:0123:4567:89AB:CDEF

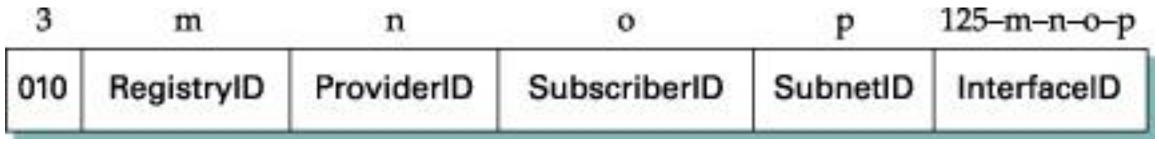
- Simplifications:
  - Omit leading zeros in a group
  - Replace zero groups with pair of colons when not ambiguous
  - Ex: 8000::123:4567:89AB:CDEF

IPv4 to IPv6 Transition

- Dual-stack
- Tunneling
- Extending IPv4 addresses
  - IPv4 addresses written as pair of colons followed by dotted decimal
    ::FFFF:192.31.20.46

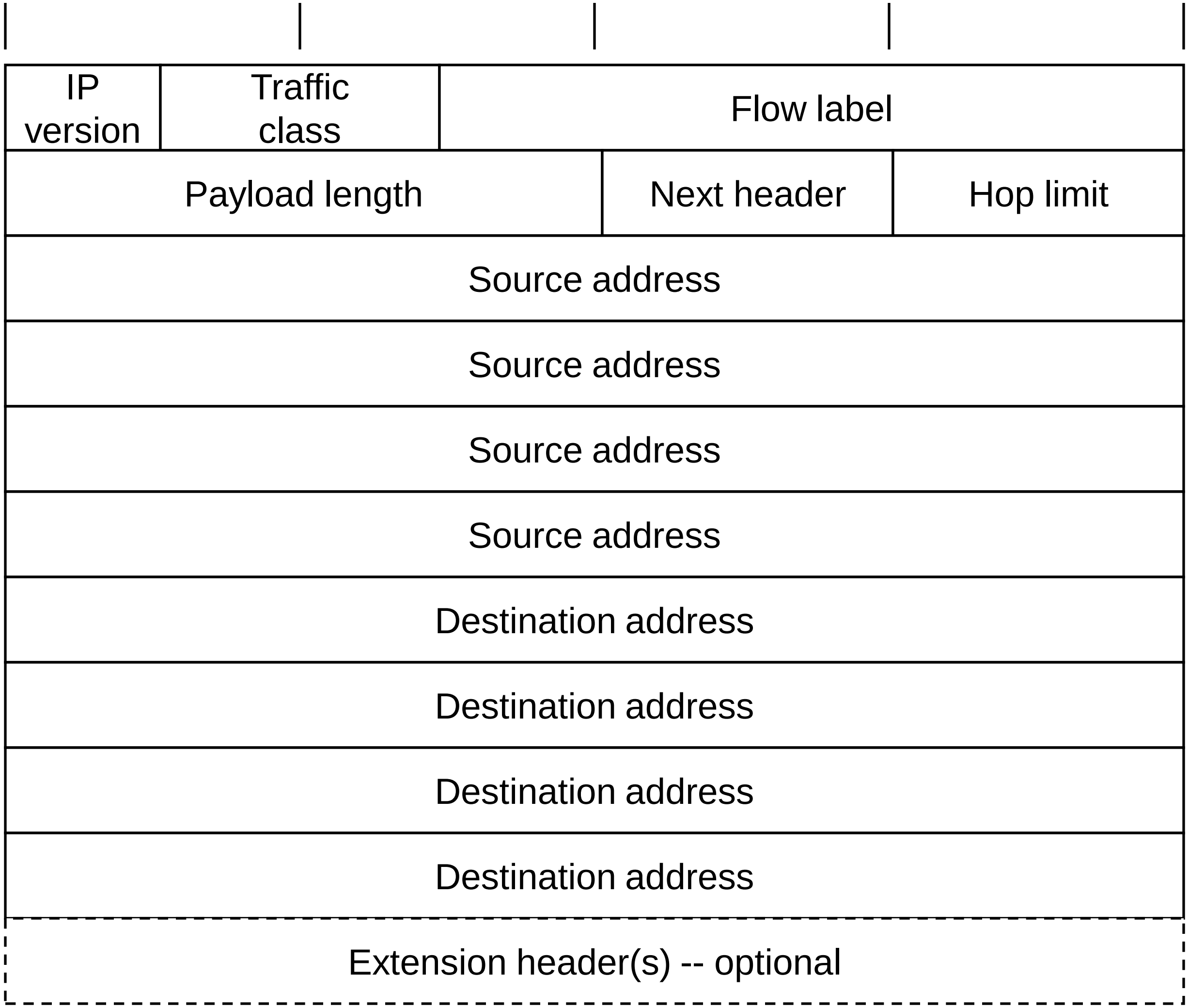
Unicast Address

- Simplify routing
  - Hierarchical structure
  - Using prefix
    - Change provider
    - Multiple providers
  - Location-based



3	m	n	o	p	125–m–n–o–p
010	RegistryID	ProviderID	SubscriberID	SubnetID	InterfaceID

IPv6 packet header format



IP version | Traffic class | Flow label

Payload length | Next header | Hop limit

Source address

Source address

Source address

Source address

Destination address

Destination address

Destination address

Destination address

Extension header(s) -- optional

Changes in header fields

- New
  - Traffic Class – ID’s special delivery req’ts (e.g, real time delivery)
  - Flow label – supports circuit-oriented channels
  - Payload length -- # data bytes (not including header)
  - Next header – ID’s extension header type, if any or Transport protocol (TCP or UDP)
  - Extension headers – various optional header fields to modify basic format (e.g, over-length payloads, authentication,…)
- Gone
  - IHL – basic and extension headers are now fixed-length
  - Protocol – function replaced by Next Header
  - Fragmentation support fields – fragmentation handled differently
  - Checksum – to improve performance (redundant with L2, L4)

Autoconfiguration

- Stateless autoconfiguration
  - Correct prefix
    - Router
  - Interface ID
    - Link local unicast + 0s + link-level address

The state of IPv6

- Hasn’t caught on as fast as expected
  - Huge cost of changing hardware
  - Success of IPv4 extension measures

- Many IPv6 “islands” now in operation on the Internet
  - Communicate by tunneling through IPv4

- Eventually, islands will merge, “take over”