National University of Computer & Emerging Sciences

CS 3001 – COMPUTER NETWORKS

Lecture 15

Chapter 4

18th March, 2025

Nauman Moazzam Hayat

nauman.moazzam@lhr.nu.edu.pk

Office Hours: 11:30 am till 01:00 pm (Every Tuesday & Thursday)

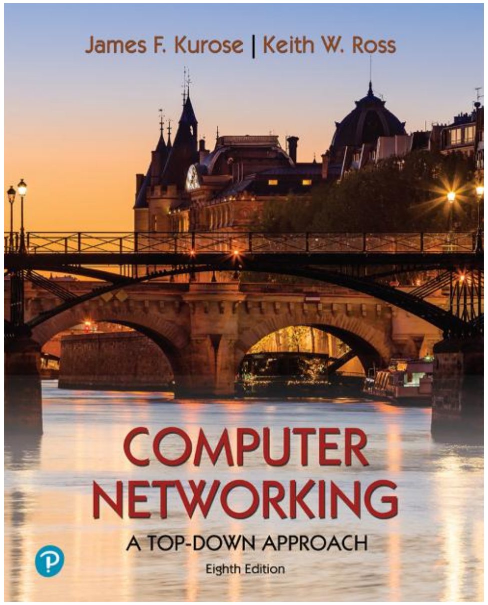
Chapter 4
Network Layer: Data Plane

A note on the use of these PowerPoint slides:
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR



Computer Networking: A Top-Down Approach
8th edition
Jim Kurose, Keith Ross
Pearson, 2020

Network layer: our goals

- understand principles behind network layer services, focusing on data plane:
  - network layer service models
  - forwarding versus routing
  - how a router works
  - addressing
  - generalized forwarding
  - Internet architecture

- instantiation, implementation in the Internet
  - IP protocol
  - NAT, middleboxes

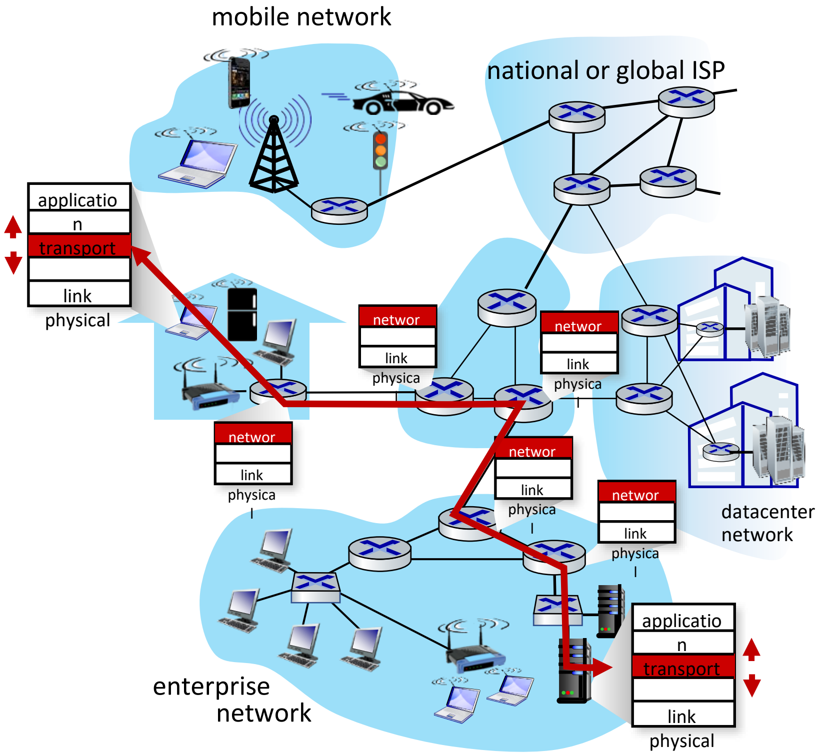
Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- ~~What’s inside a router~~
  - ~~input ports, switching, output ports~~
  - ~~buffer management, scheduling~~
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized Forwarding, SDN
  - Match+action
  - OpenFlow: match+action in action
- Middleboxes

Network-layer services and protocols

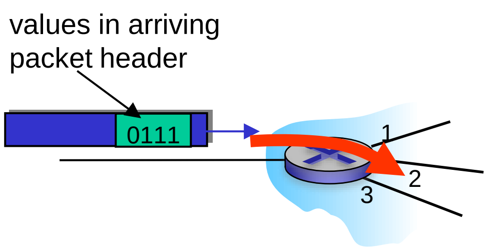
- transport segment from sending to receiving host
  - sender: encapsulates segments into datagrams, passes to link layer
  - receiver: delivers segments to transport layer protocol
- network layer protocols in *every Internet device*: hosts, routers
- routers:
  - examines header fields in all IP datagrams passing through it
  - moves datagrams from input ports to output ports to transfer datagrams along end-end path



Network layer: data plane, control plane

Data plane: (key function is forwarding)

- *local*, per-router function
- determines how datagram arriving on router input port is forwarded to router output port



Control plane: (key function is routing)

- *network-wide* logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers (the routing algorithm determines the contents of the routers' forwarding tables. A routing algorithm runs in each and every router and both forwarding and routing functions are contained within a router. The routing algorithm function in one router communicates with the routing algorithm function in other routers to compute the values for its own forwarding table.)
  - *software-defined networking (SDN)*: implemented in (remote) servers (a physically separate, remote controller computes and distributes the forwarding tables to be used by each and every router. The router device performs forwarding only, while the remote controller computes and distributes forwarding tables.)

Two key network-layer functions

network-layer functions:

- *forwarding:* move packets from a router's input link to appropriate router output link
- *routing:* determine route taken by packets from source to destination
  - *routing algorithms*

analogy: taking a trip

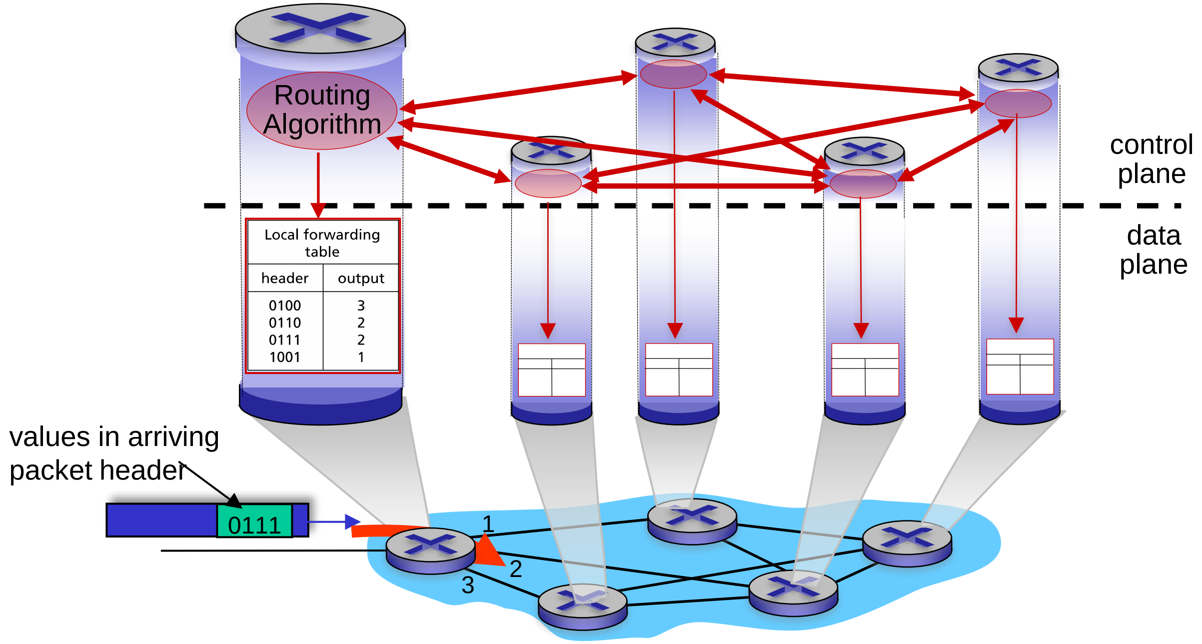
- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination

forwarding

routing

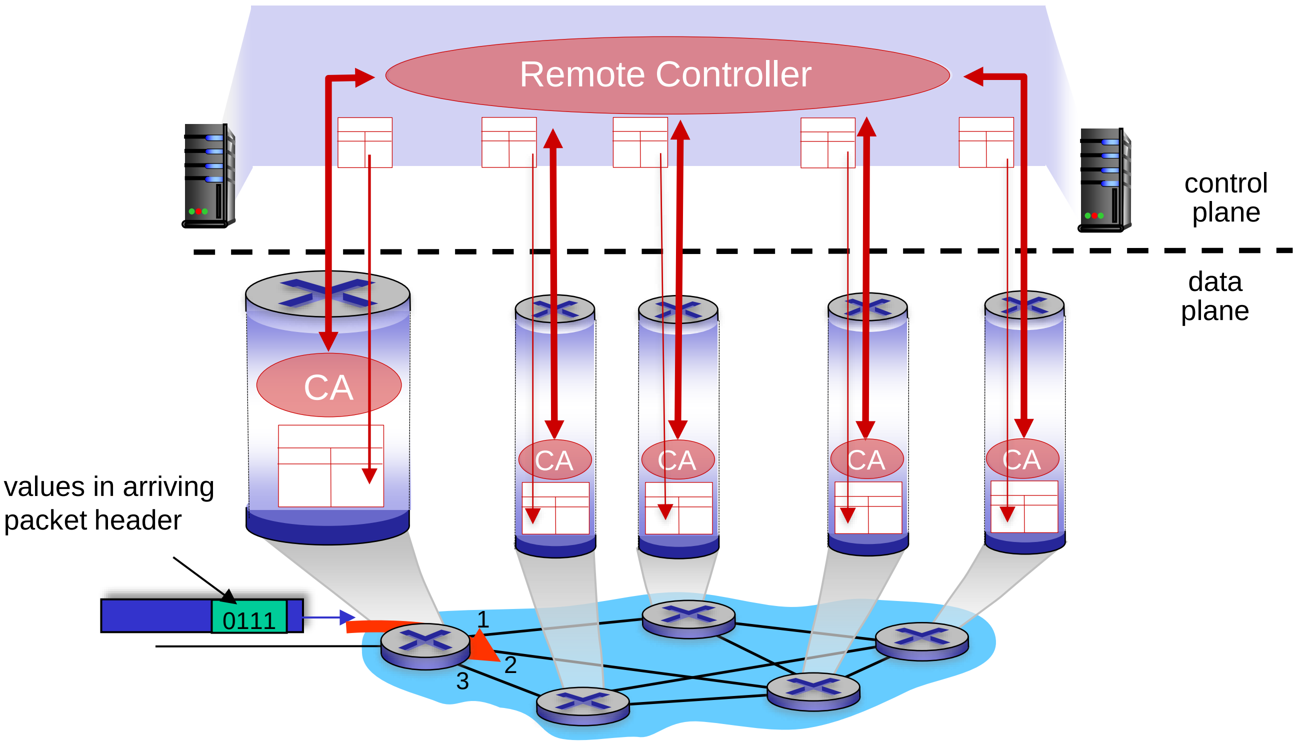
Per-router control plane

Individual routing algorithm components *in each and every router* interact in the control plane



Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



Network service model

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

example services for *individual* datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

example services for a *flow* of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing (jitter.)

Network-layer service model

Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no

Internet "best effort" service model

No guarantees on*:*

i. successful datagram delivery to destination
ii. timing or order of delivery
iii. bandwidth available to end-end flow

Network-layer service model

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 (Proposed service model extension)	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet (Proposed service model extension)	Diffserv (RFC 2475)	possible	possibly	possibly	no

Reflections on best-effort service:

- simplicity of mechanism has allowed Internet to be widely deployed adopted
- sufficient provisioning of bandwidth allows performance of real-time applications (e.g., interactive voice, video) to be “good enough” for “most of the time”
- replicated, application-layer distributed services (datacenters, content distribution networks) connecting close to clients’ networks, allow services to be provided from multiple locations
- congestion control of “elastic” services helps

It’s hard to argue with success of best-effort service model

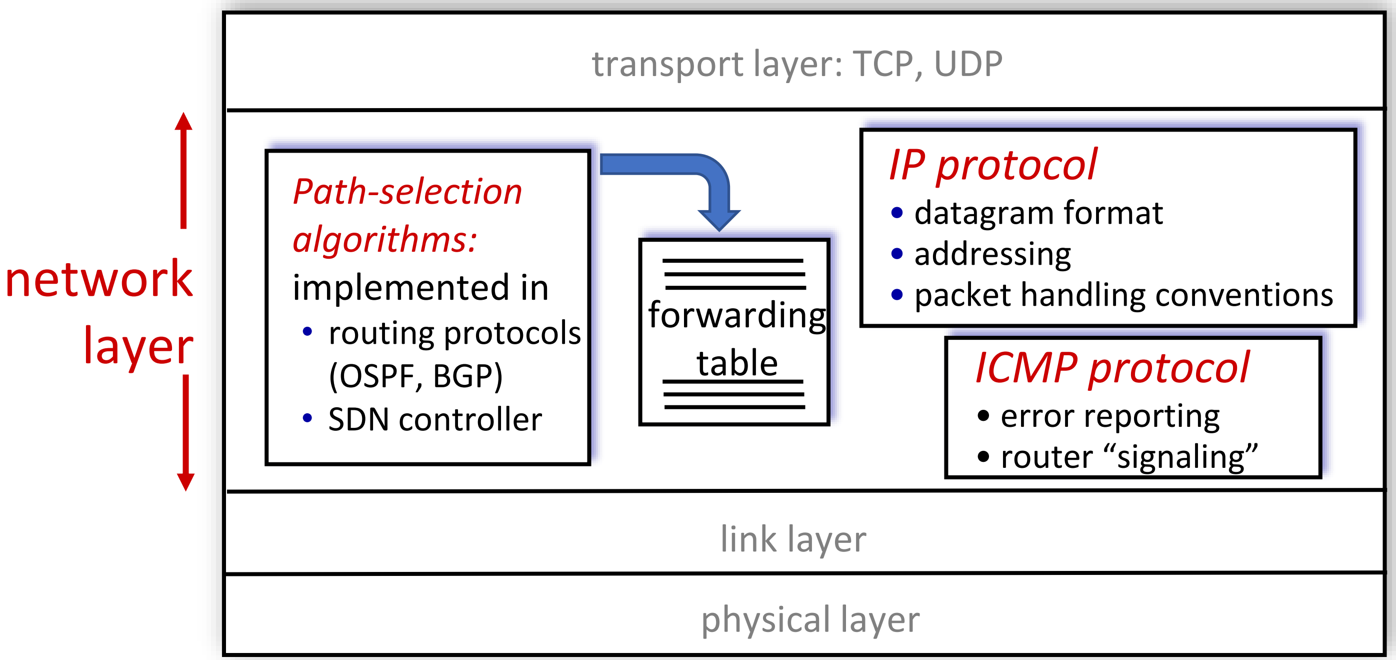
Network layer: “data plane” roadmap

- Network layer: overview
  - data plane
  - control plane
- What’s inside a router
  - input ports, switching, output ports
  - buffer management, scheduling
- IP: the Internet Protocol
  - datagram format
  - addressing
  - network address translation
  - IPv6

- Generalized Forwarding, SDN
  - match+action
  - OpenFlow: match+action in action
- Middleboxes

Network Layer: Internet

host, router network layer functions:



The Internet Network's Layer 3 major components are

1. IP Protocol
2. Routing Component
3. Reporting Errors / Responding to Requests (ICMP, IGMP etc.)

The Internet Protocol (IP)

- *Connectionless* (no call set up at the network layer), *Unreliable designed to be used in a packet switched network like the Internet*
- No *state* about end-to-end connections
- *Best Effort Services* (no bandwidth, loss, error, in-order, timing guarantees)
- No *Congestion* indicators

IP relies on TCP for these services

IPv4 Datagram format

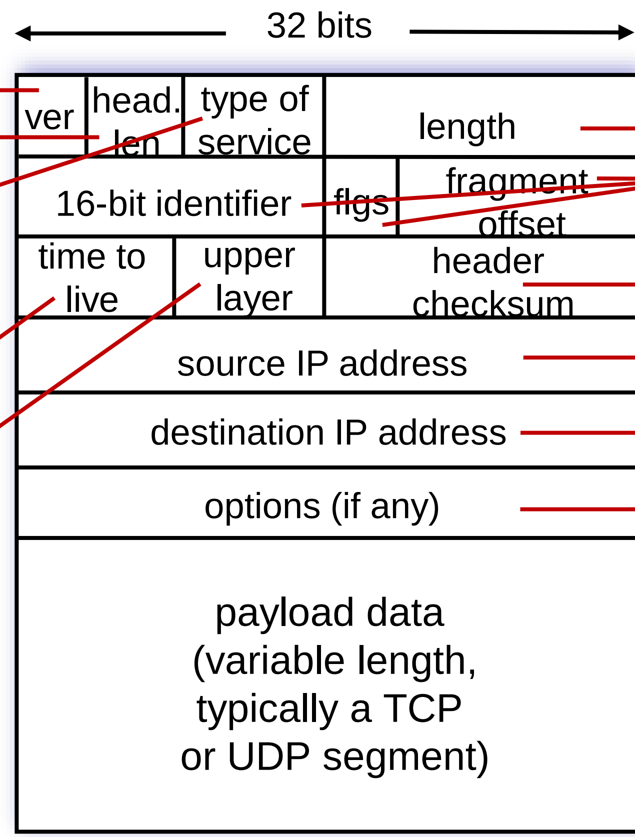
IP protocol version number

header length(bytes)

"type" of service:
- diffserv (0:5)
- ECN (6:7)

TTL: remaining max hops
(decremented at each router)

upper layer protocol (e.g., TCP or UDP)



total datagram length (bytes)

fragmentation/ reassembly

header checksum

32-bit source IP address

32-bit destination IP address

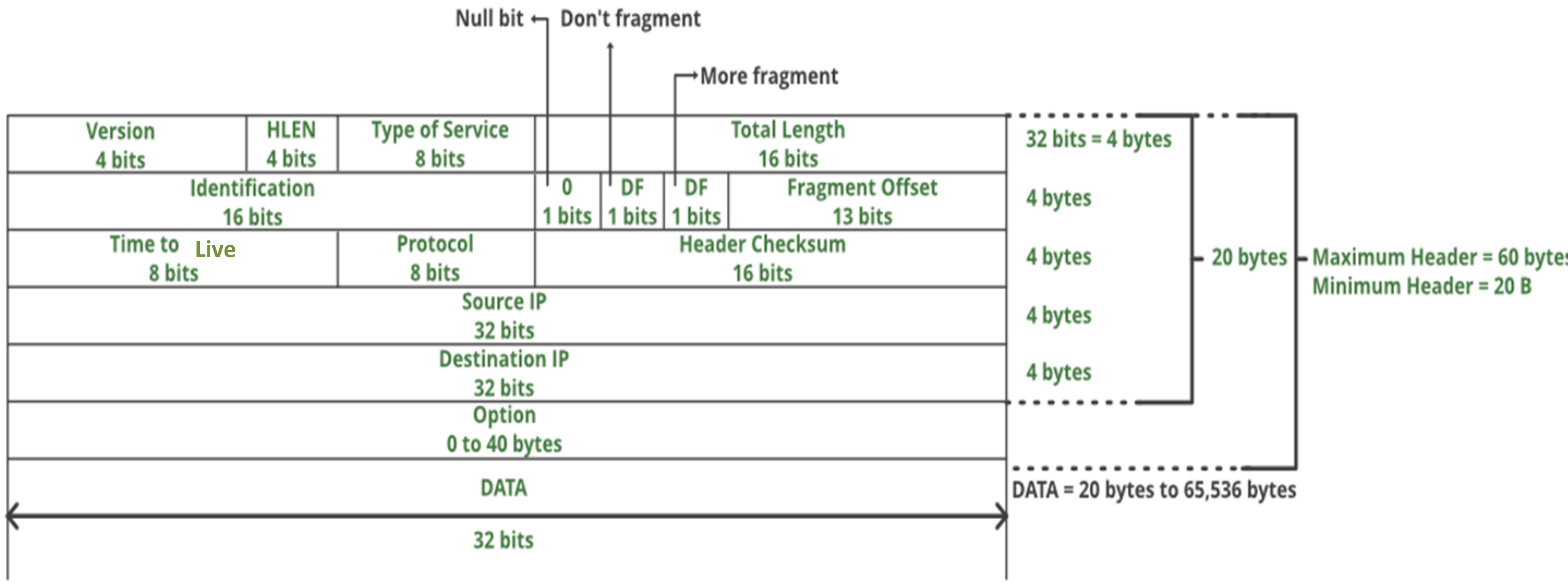
e.g., timestamp, record route taken

Maximum length: 64K bytes
Typically: 1500 bytes or less

overhead

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead for TCP+IP
- Typical IP datagram header is 20 bytes (if options not used)

IPv4 Header



- Ver field for IPv4 will always contain the decimal value 4 (i.e. 0100 in binary)
- Header length field is a 4 bit field that contains the length of the IP header in bytes, which always lies in the range of 20 bytes (min) to 60 bytes (max), but the range of these 4 bits can only be from 0000 (i.e. 0 in decimal) to 1111 (i.e. 15 in decimal), so to represent the header length, we use a scaling factor of 4. Thus

Actual Header length = (Header length field value x 4) bytes

IPv4 Header

Examples:

- If the header length field contains the value 0101 (i.e. 5 in decimal), then the Actual Header length = 5 x 4 = 20 bytes

- Similarly, if the header length field contains the value 1010 (i.e. 10 in decimal), then the Actual Header length = 10 x 4 = 40 bytes

Default TTL (IPv4) Values Or HOP Limit (IPv6)

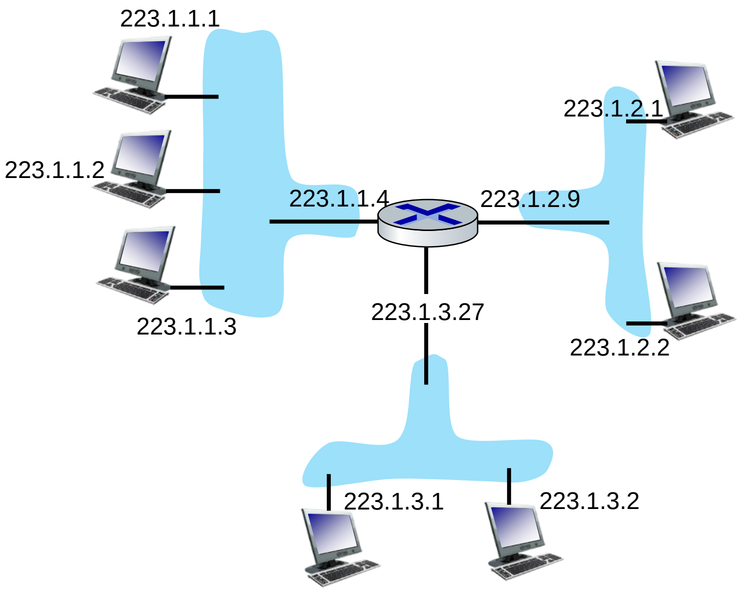
Default TTL and Hop Limit values vary between different operating systems, here are the defaults for a few:

- 64 – Linux/MAC OSX systems
- 128 – Windows systems

As you can see, the TTL or Hop Limit seen in packets from a host could, in part, be used to identify the operating system in use on that host.

IPv4 addressing: introduction

- **IP address:** 32-bit identifier associated with each host or router *interface* (thus 2^{32} i.e. approx. 4 billion globally unique IPv4 addresses possible)
- **interface:** connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
  - IP addresses associated with each interface (& not with host or router)

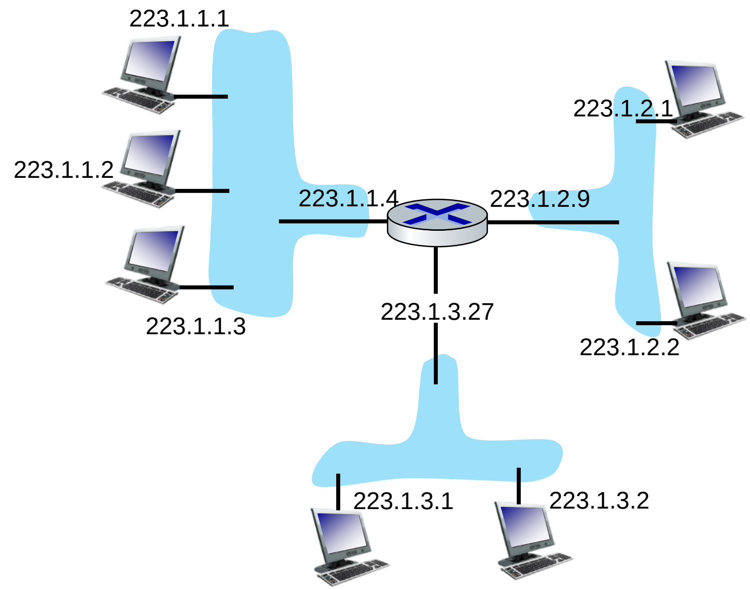


dotted-decimal IP address notation:

$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$

IP addressing: introduction

- IP address: 32-bit identifier associated with each host or router *interface*
- interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)



dotted-decimal IP address notation:

$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

IPv4 Addressing

- IPv4 address is a 32-bit address, implemented in software, is used to uniquely and globally identify a host or a router on the Internet
- A device can have more than one IP address if it is connected to more than one network (multi-homed)
- An IP address have two parts, the netid and the hostid. They have variable lengths depending on the class of the address
- All devices on the same network have the same netid

- Two types of IPv4 addressing schemes, i.e.
  - Classful IP Addressing
  - Classless IP Addressing

Classful IPv4 Addressing

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

a. Binary notation

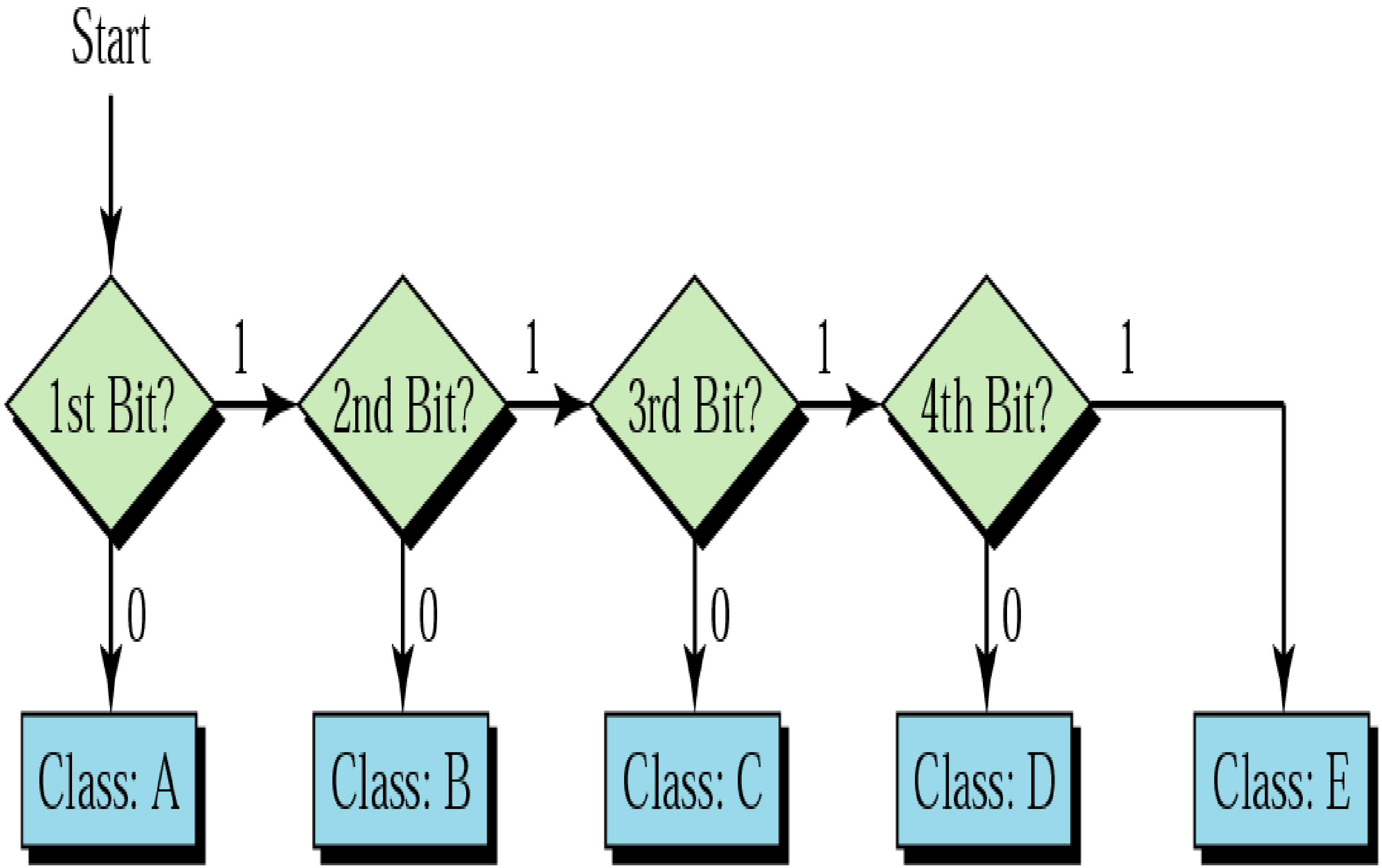
	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191			
Class C	192–223			
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

⇒ Where = net ID & = host ID - *Big Big Waste, thus being replaced by Classless IP Addressing*

Class	Max. Number of Networks (Blocks)	Max. # of nodes in the network (Block Size)	Application
A	$2^7 = 128$	$2^{24} = 16{,}777{,}216$	Unicast
B	$2^{14} = 16{,}384$	$2^{16} = 65{,}536$	Unicast
C	$2^{21} = 2{,}097{,}152$	$2^8 = 256$	Unicast
D	1	$2^{28} = 268{,}435{,}456$	Multicast
E	1	$2^{28} = 268{,}435{,}456$	Reserved

Finding the Class



Assignment # 3 (Chapter – 3) (Already announced)

- *3rd Assignment will be uploaded on Google Classroom on Thursday, 13th March, 2025, in the Stream - Announcement Section*

- *Due Date: ~~Thursday, 20th March~~ Tuesday, 25th March, 2025 (Handwritten solutions to be submitted during the lecture; deadline extended due to LAB midterms next week)*

- *Please read all the instructions carefully in the uploaded Assignment document, follow & submit accordingly*

Quiz # 3 (Chapter – 3) (Already announced)

- *On: ~~Thursday, 20th March, 2025~~ , Tuesday, 25th March, 2025(During the lecture; deadline extended due to LAB midterms next week)*

- *Quiz to be taken during <u>own section class only</u>*