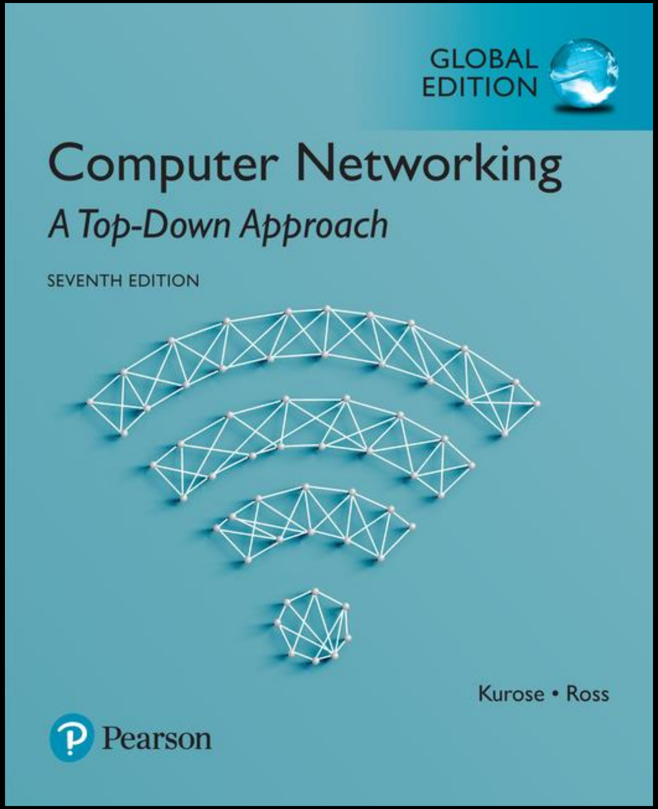
Computer Network

Network Layer I

School of Electric and Computer Engineering
Pusan National University, KOREA
Younghwan Yoo

Textbook



Computer Networking

A Top–Down Approach

7th edition

Jim Kurose, Keith Ross

Pearson

April 2016

Contents

Computer Network introduction

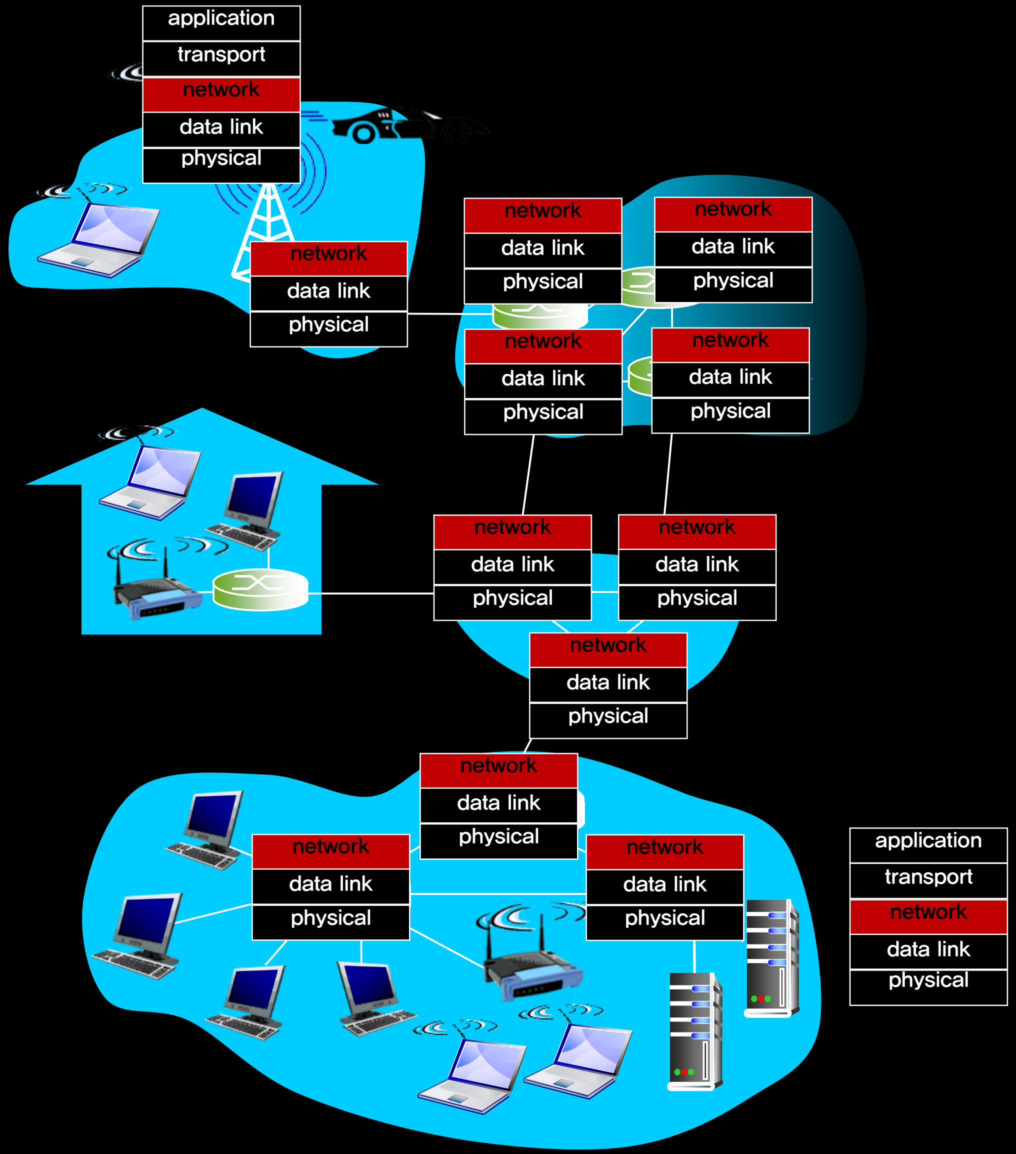
Contents

Computer Network introduction

01. Overview of Network Layer

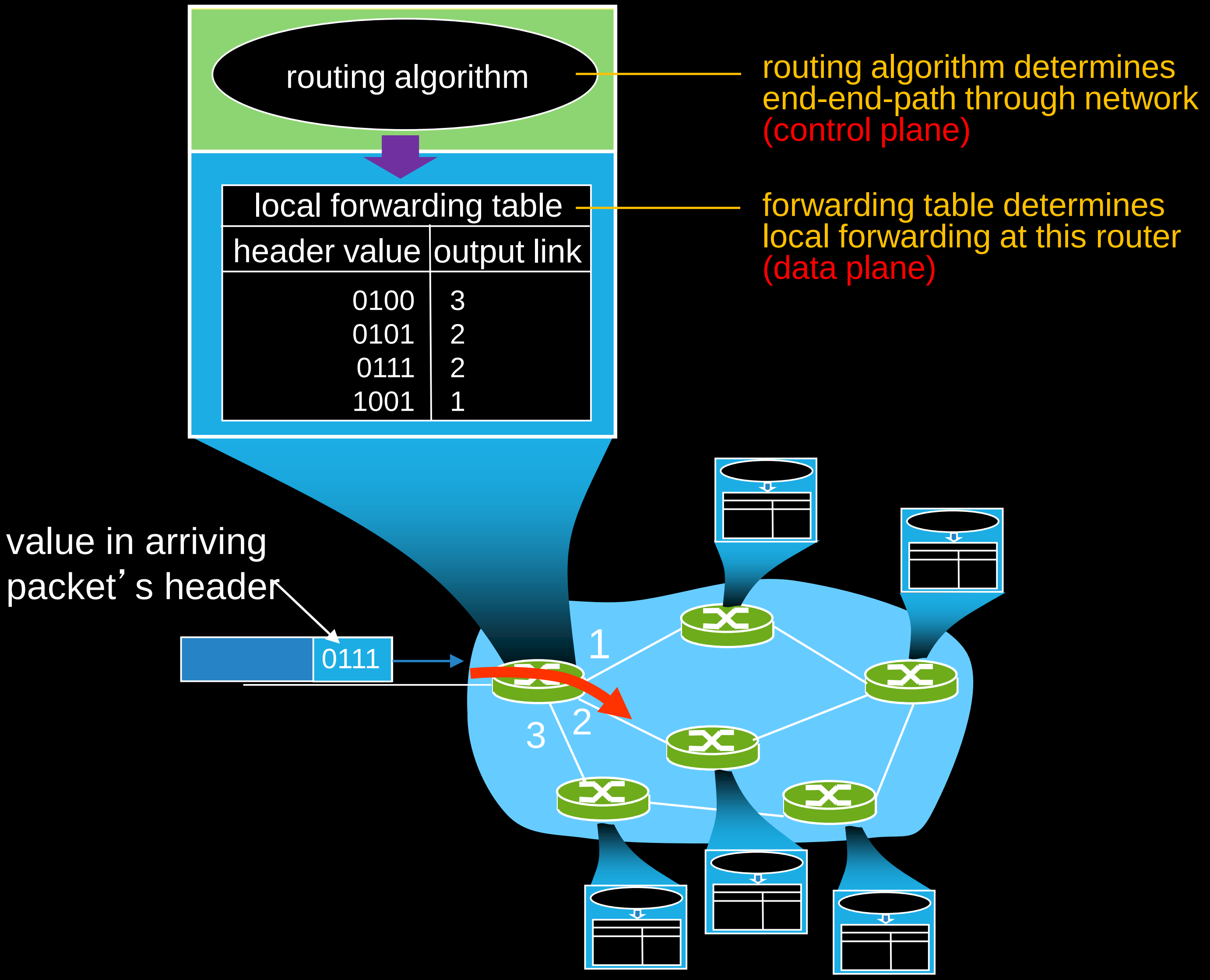
Network Layer

- Transport segment from sending to receiving host
- Sending side encapsulates segments into datagrams and sends them
- Receiving side receives datagrams and delivers segments to transport layer
- Network layer protocols in every host and router
- Router examines header fields in all IP datagrams passing through it



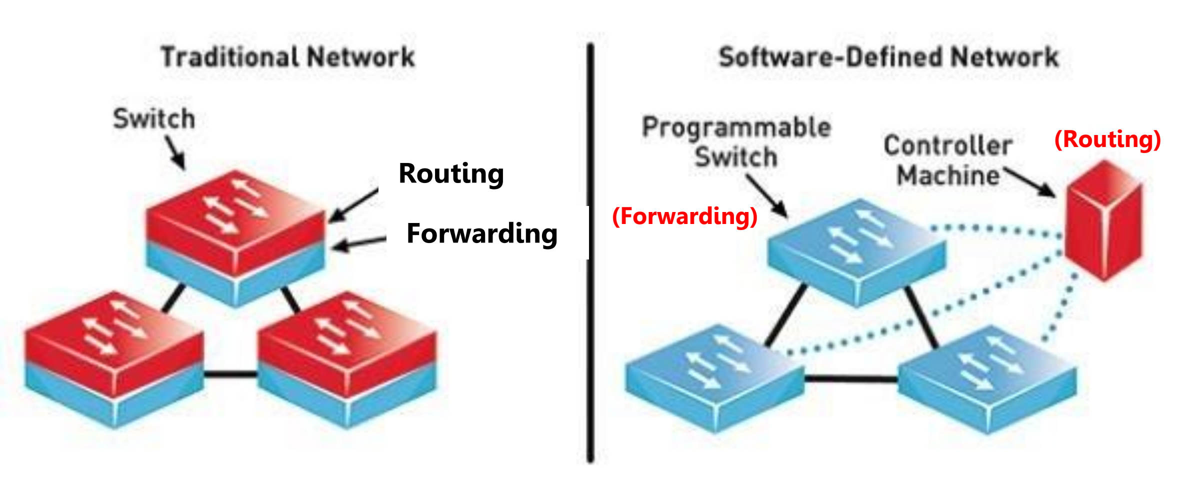
Two Key Functions in Network Layer

- Routing: determine route taken by packets from source to destination
  - routing algorithms
- Forwarding: move packets from router's input to appropriate router output
  - i.e., packet delivery to the next node



Traditional vs. SDN Network

- Traditional IP network
  - routing and forwarding at the same system
- Software–defined network (SDN)
  - routing and forwarding separated at different systems

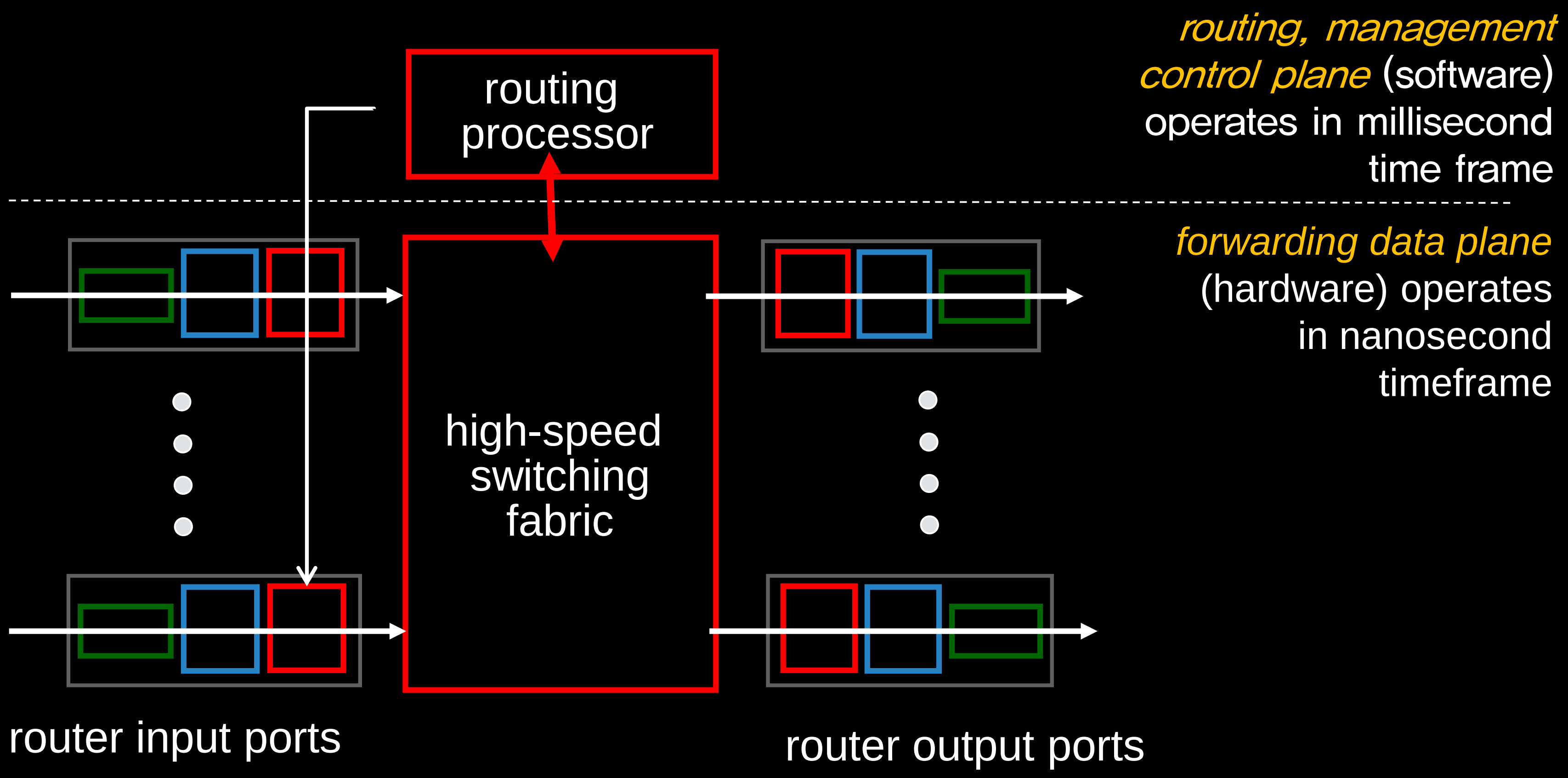


출처 – https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiDsaqCuP3bAhUZM94KHdm_Cp8QjRx6BAgBEAU&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FTraditional-Network-versus-SDN_fig1_319876305&psig=AOvVaw3Azn7yQLyweWfI2wFNRYhQ&ust=1530518464669003

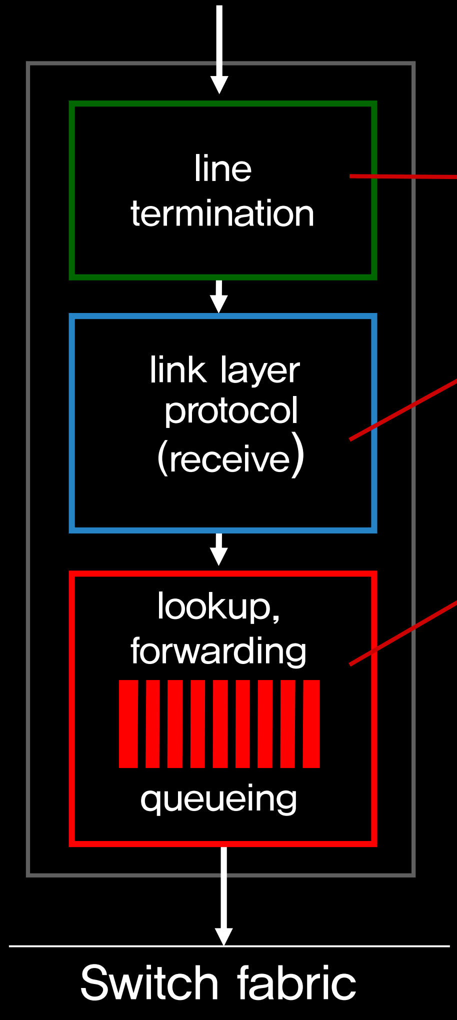
02. Inside of Router

Router Architecture Overview

- High–level view of generic router architecture



Input Port Functions



physical layer: bit-level reception

data link layer: e.g., Ethernet

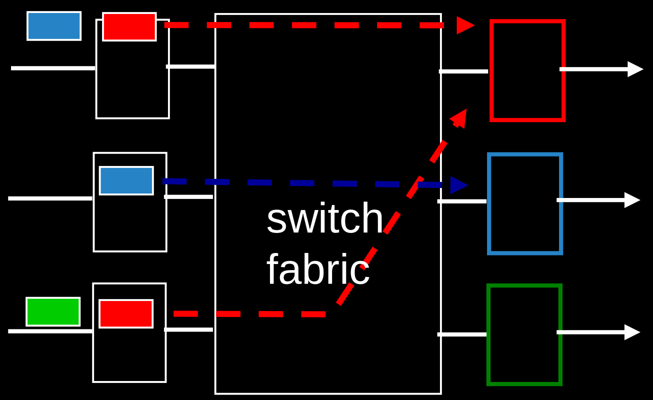
see Chapter 5

Decentralized switching:

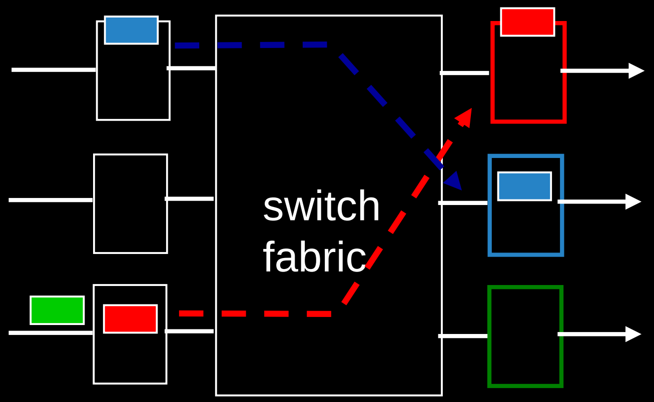
- using header field values, lookup output port using forwarding table in input port memory ("match plus action")
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Input Port Queuing

- Fabric slower than input ports combined -> queuing may occur at input queues
  - queuing delay and loss due to input buffer overflow!
- Head-of-the-Line (HOL) blocking: queued datagram at front of queue prevents others in queue from moving forward



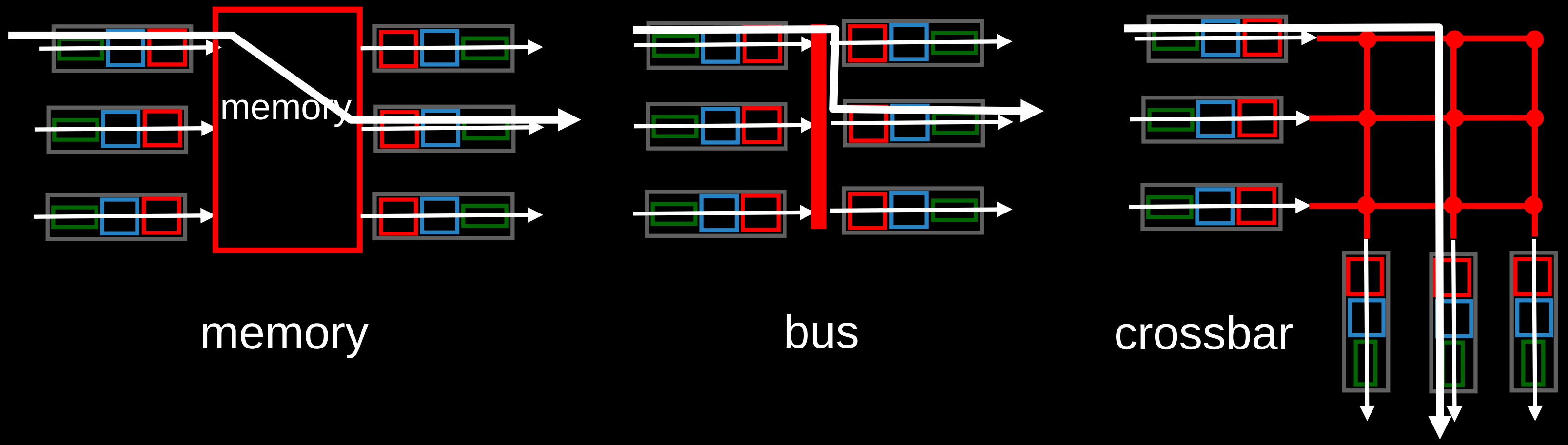
output port contention:
only one red datagram can be transferred.
lower red packet is blocked



one packet time later:
green packet experiences HOL blocking

Switching Fabrics

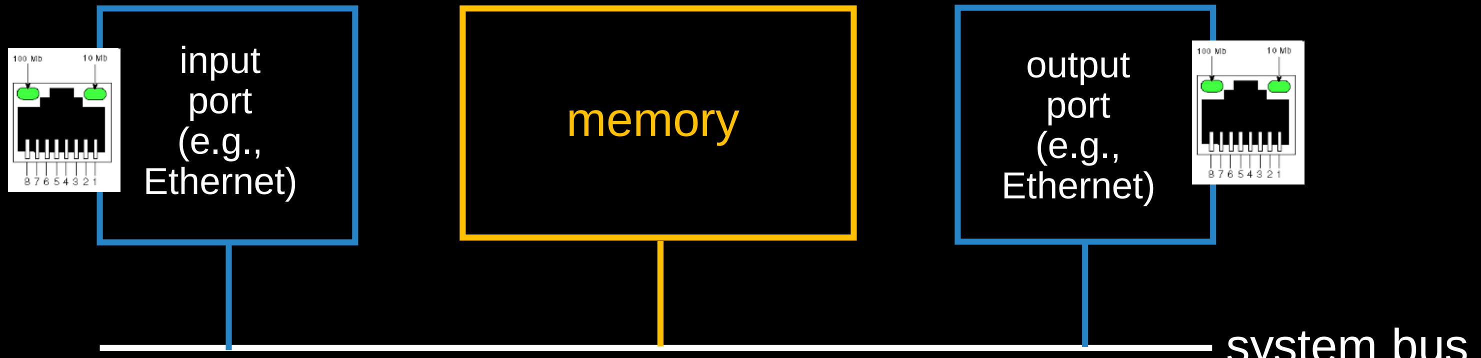
- Transfer packet from input buffer to appropriate output buffer
- Switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- Three types of switching fabrics



Switching via Memory

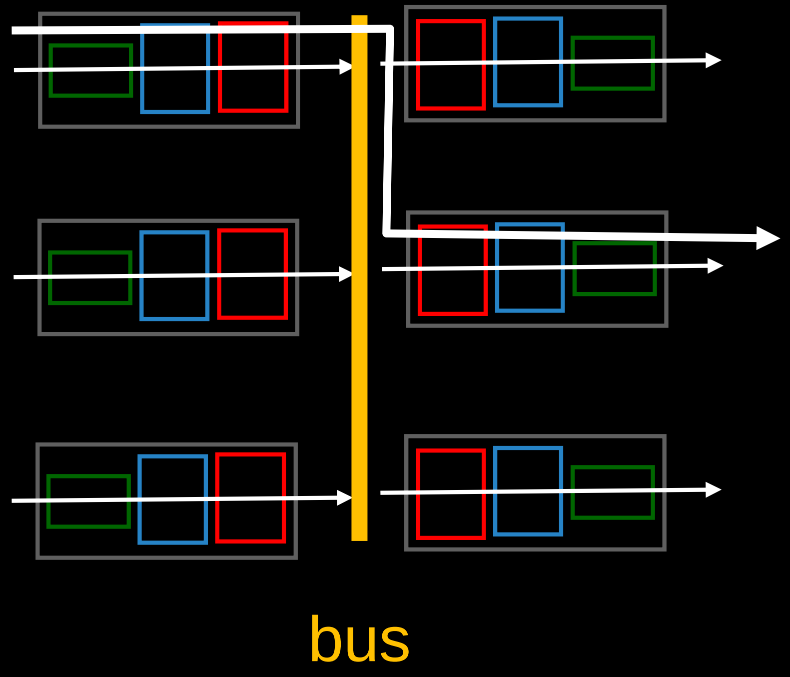
first generation routers:

- Traditional computers with switching under direct control of CPU
- Packet copied to system's memory
- Speed limited by memory bandwidth (2 bus crossings per datagram)



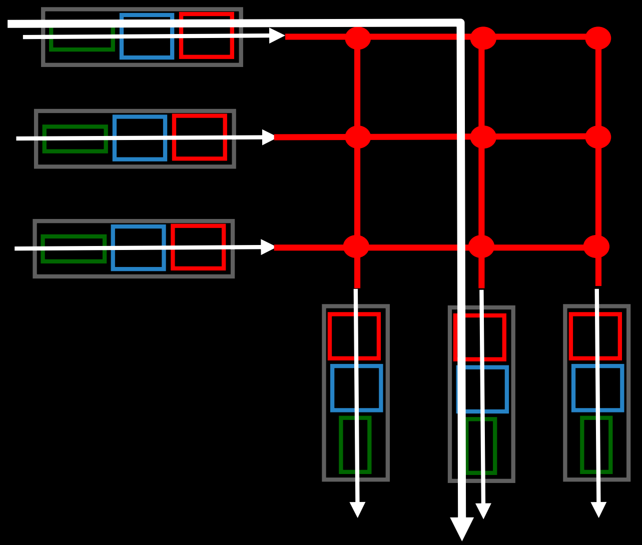
Switching via Bus

- Datagram from input port memory to output port memory via a shared bus
- Bus contention: switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers



- Banyan networks, crossbar, other interconnection nets initially developed to connect processors in multiprocessor
- Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through the interconnection network

<crossbar>



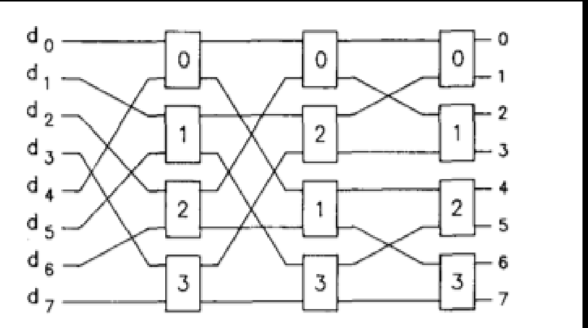
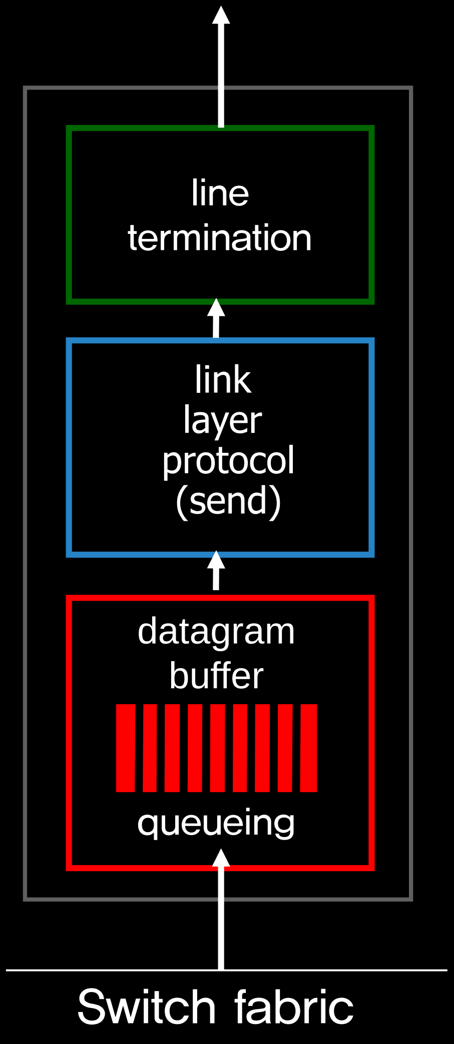


Fig. 2. A three-stage banyan network.

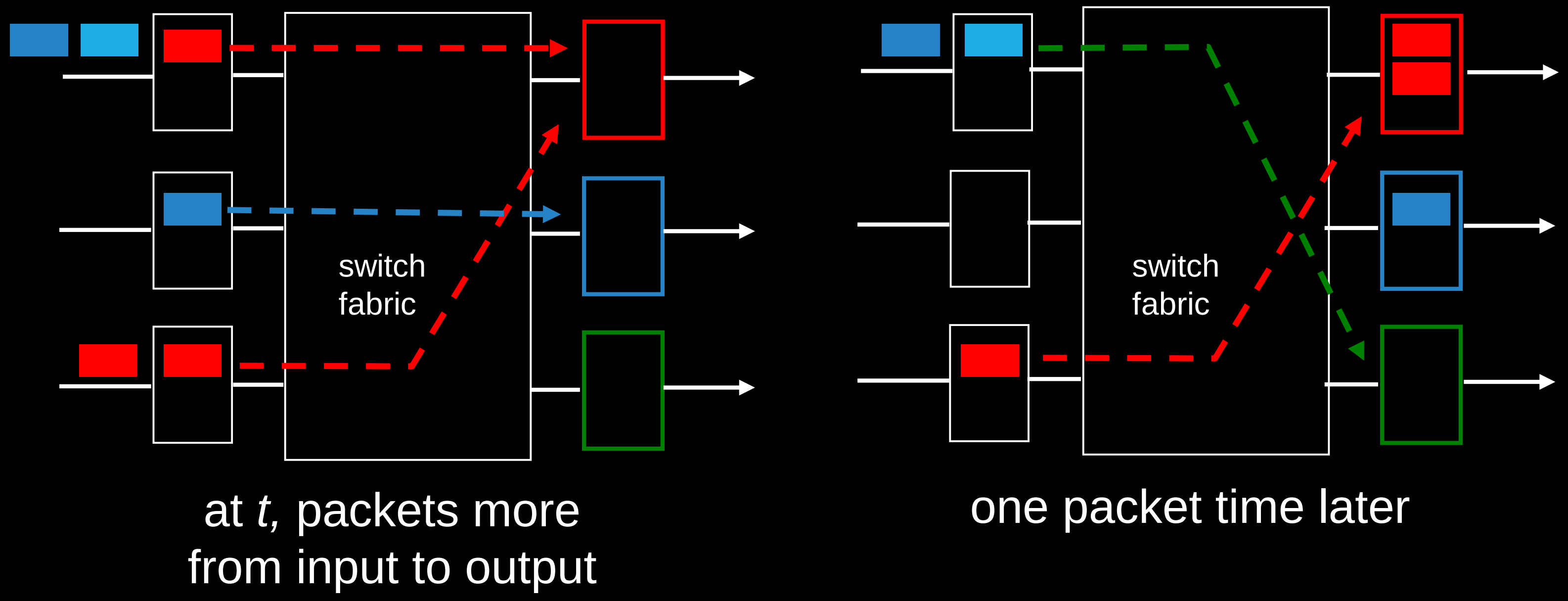
출처 – https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiKlMeL9lHcAhWVdt4KHWMDB68QjRx6BAgBEAU&url=https%3A%2F%2Fstackoverflow.com%2Fquestions%2F4751915%2Fhow-to-generate-a-banyan-network-for-n-inputs&psig=AOvVaw0LLA5cPGMgn-1ZsLTtjPte&ust=1530672078267513

Output Port Functions



- Buffering required when datagrams arrive from fabric faster than the transmission rate
- Scheduling discipline chooses among queued datagrams for transmission

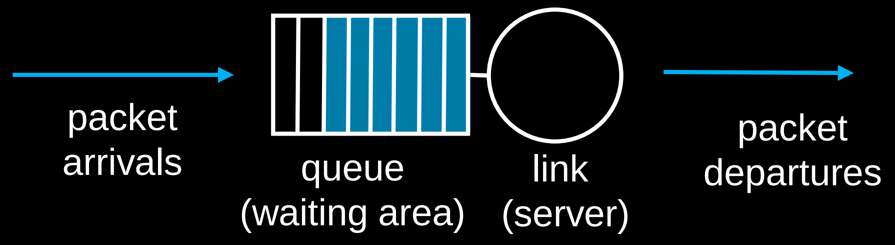
Output Port Queuing



- Buffering when arrival rate via switch exceeds output line speed
- Queuing (delay) and loss due to output port buffer overflow!

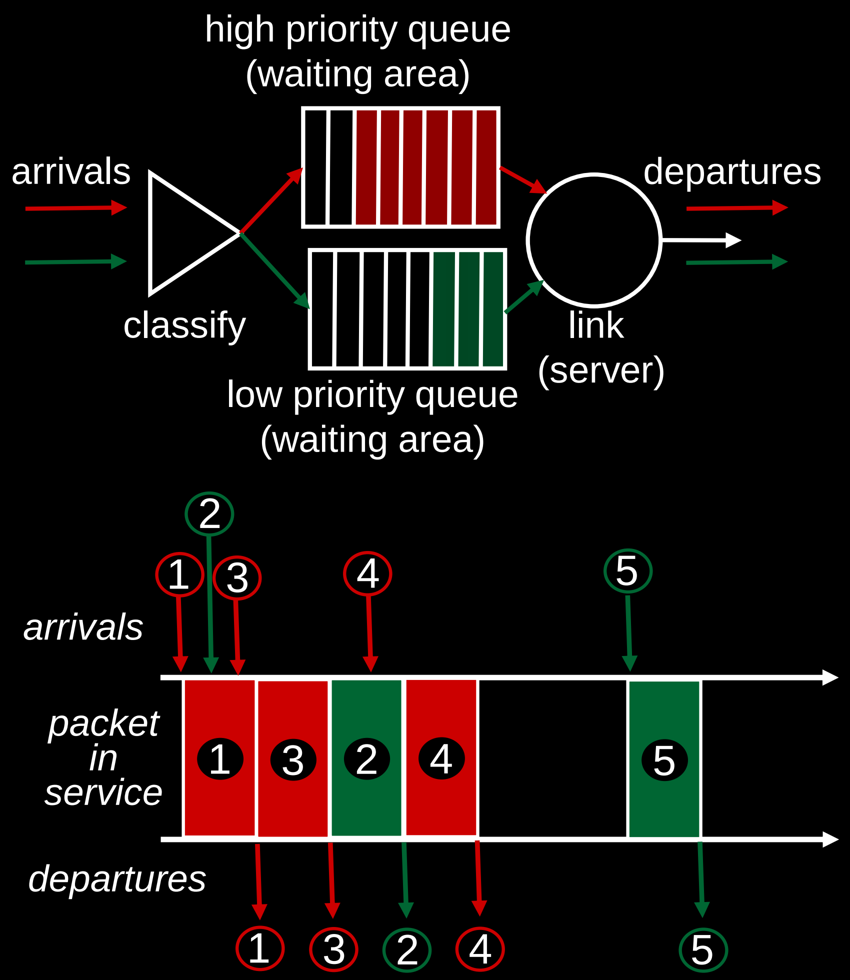
Scheduling Mechanisms

- Scheduling: choose next packet to send on link
- FIFO (first in first out) scheduling: send in order of arrival to queue
- Discard policy: if packet arrives to full queue: who to discard?
  - tail drop: drop arriving packet
  - priority: drop/remove on priority basis
  - random: drop/remove randomly



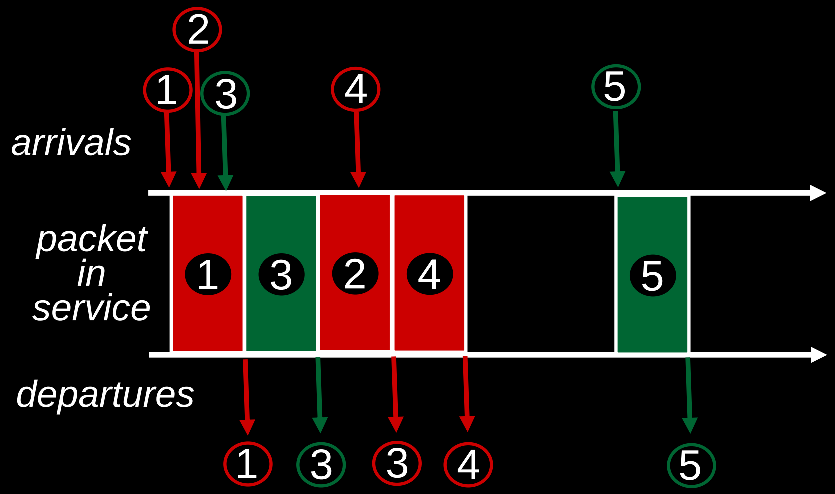
Priority Scheduling

- Send highest priority queued packet
- Multiple classes, with different priorities
- Class may depend on marking or other header info, e.g. IP source/dest, port numbers, etc.



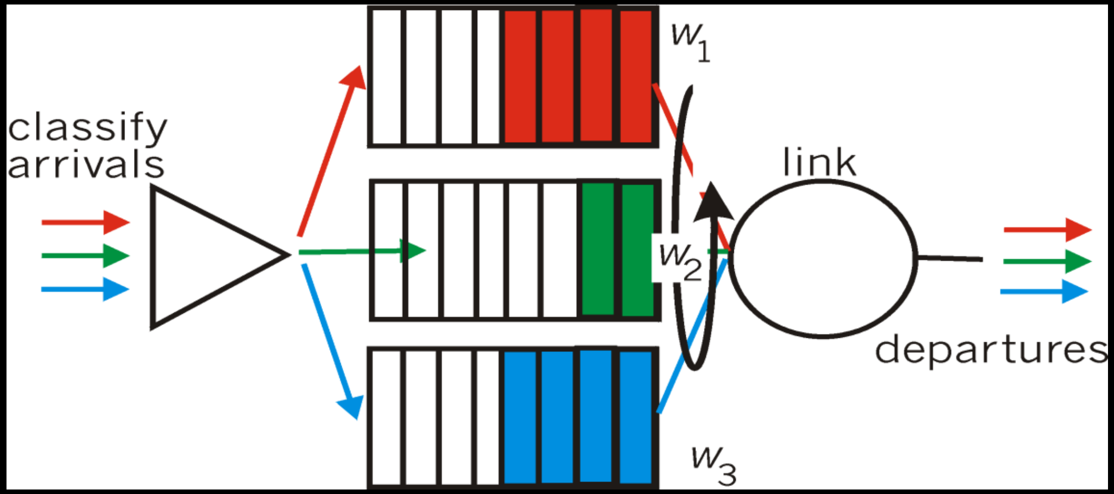
Round Robin (RR) Scheduling

- Multiple classes
- Cyclically scan class queues, sending one complete packet from each class (if available)



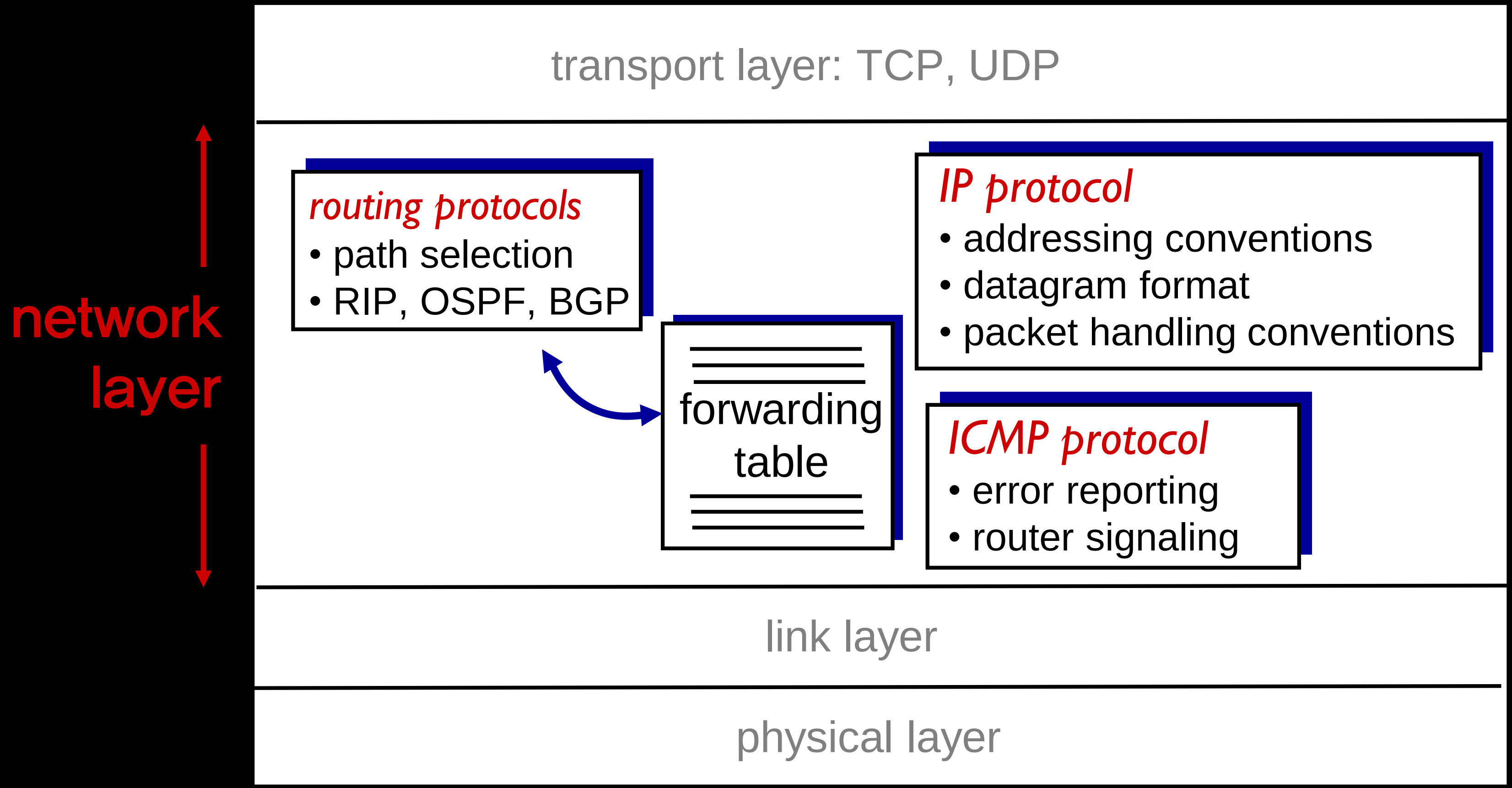
Weighted Fair Queuing (WFQ) Scheduling

- Generalized Round Robin
- Each class gets weighted amount of service in each cycle



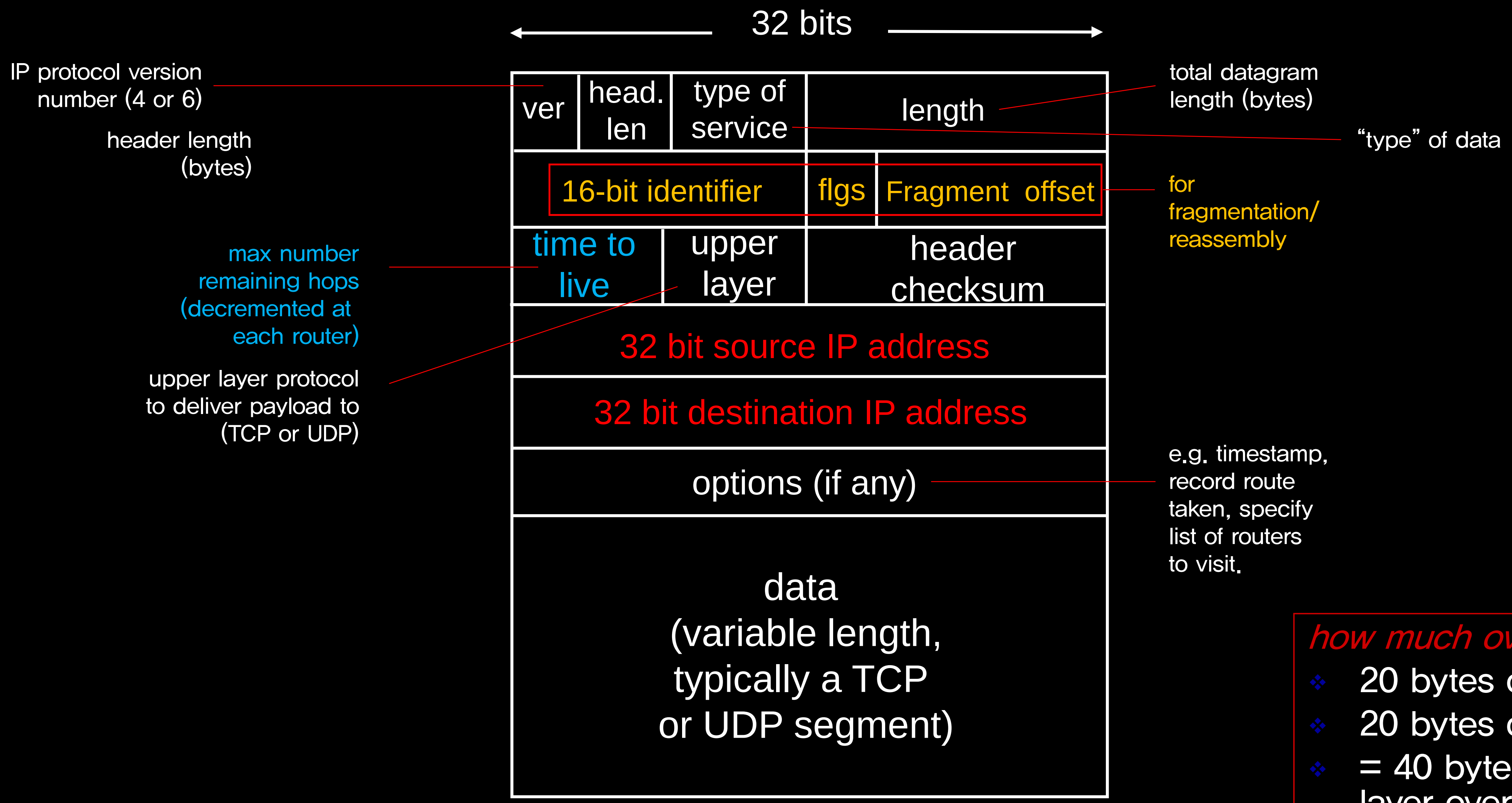
03. Internet Protocol Overview

- Host, router network layer functions:



- RIP (routing information protocol)
- OSPF (open shortest path first)
- BGP (border gateway protocol)
- ICMP (Internet control message protocol)

IP Datagram Format

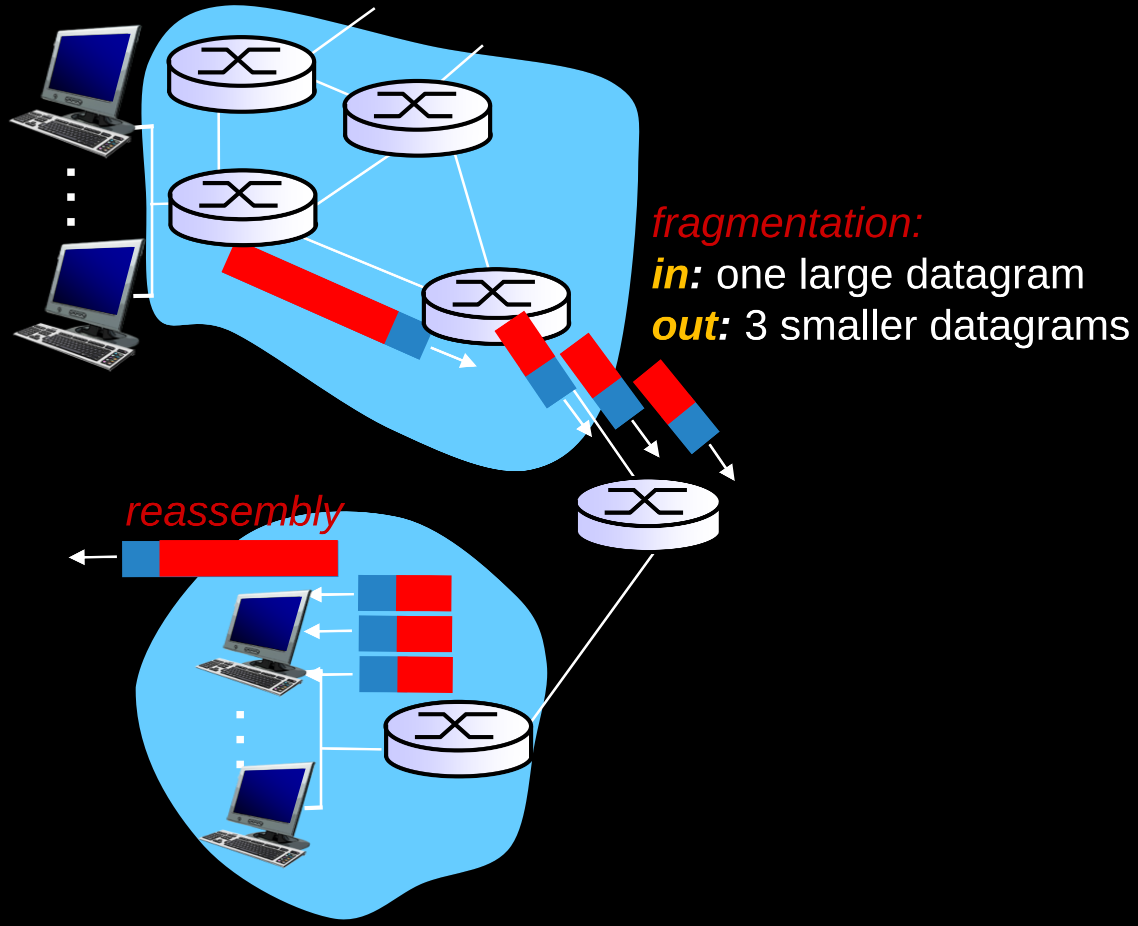


how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

IP Fragmentation & Reassembly

- Network links have MTU (max. transfer size)
  – largest possible link-level frame
    - different link types, different MTUs
- Large IP datagram divided ("fragmented") within network
    - one datagram becomes several datagrams
    - "reassembled" only at final destination
    - IP header bits used to identify, order related fragments



IP Fragmentation & Reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes

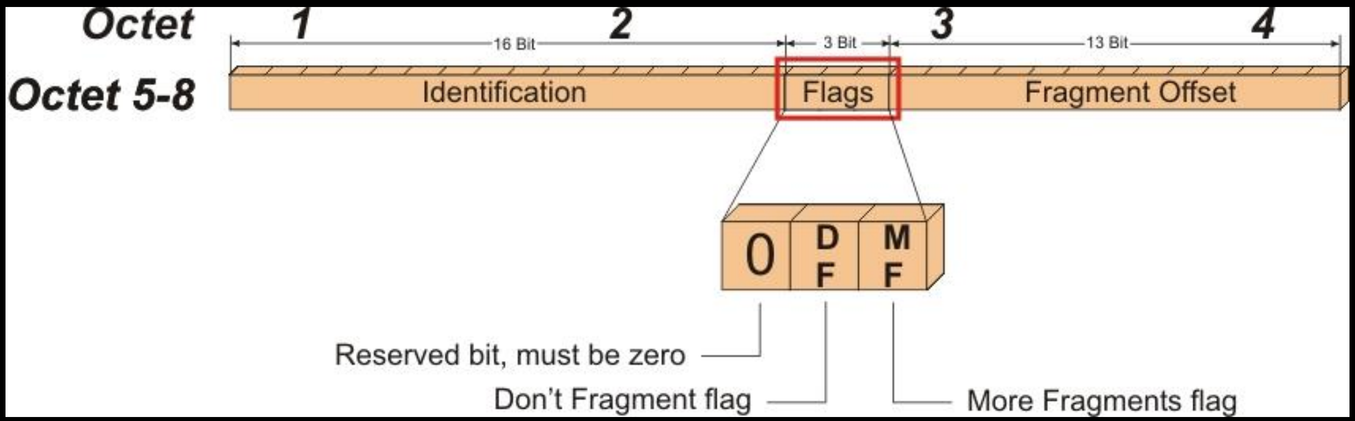
1480 bytes in data field

offset = 1480/8

length	ID	fragflag	offset
=4000	=x	=0	=0

one large datagram becomes several smaller datagrams

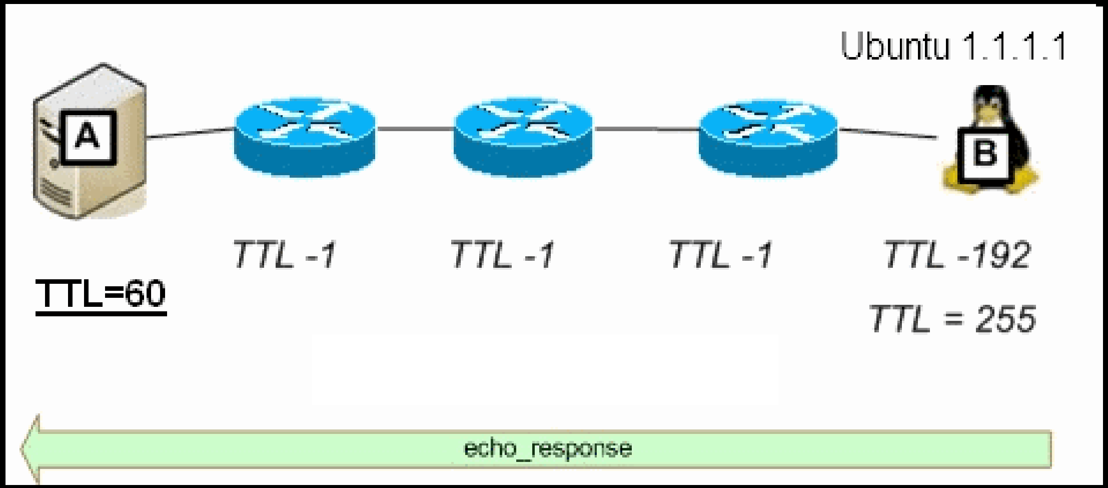
length	ID	fragflag	offset
=1500	=x	=1	=0
=1500	=x	=1	=185
=1040	=x	=0	=370



출처 – https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwi8oMXYyYLcAhXM62EKHYZyAmlQjRx6BAgBEAU&url=http%3A%2F%2Fwww.inacon.de%2Fph%2Fdata%2FIPv4%2FHeader_fields%2FIP-Header-Field-Flags_OS_RFC-791.htm&psig=AOvVaw2uflD8jt3ncstFh5tDEBqs&ust=1530695050103958

Time-to-Live (TTL)

- TTL field (8 bits)
  - included to ensure that datagrams do not circulate forever (due to, for example, a long-lived routing loop) in the network
  - decremented by one each time the datagram is processed by a router
  - if reaching 0, a router must drop that datagram

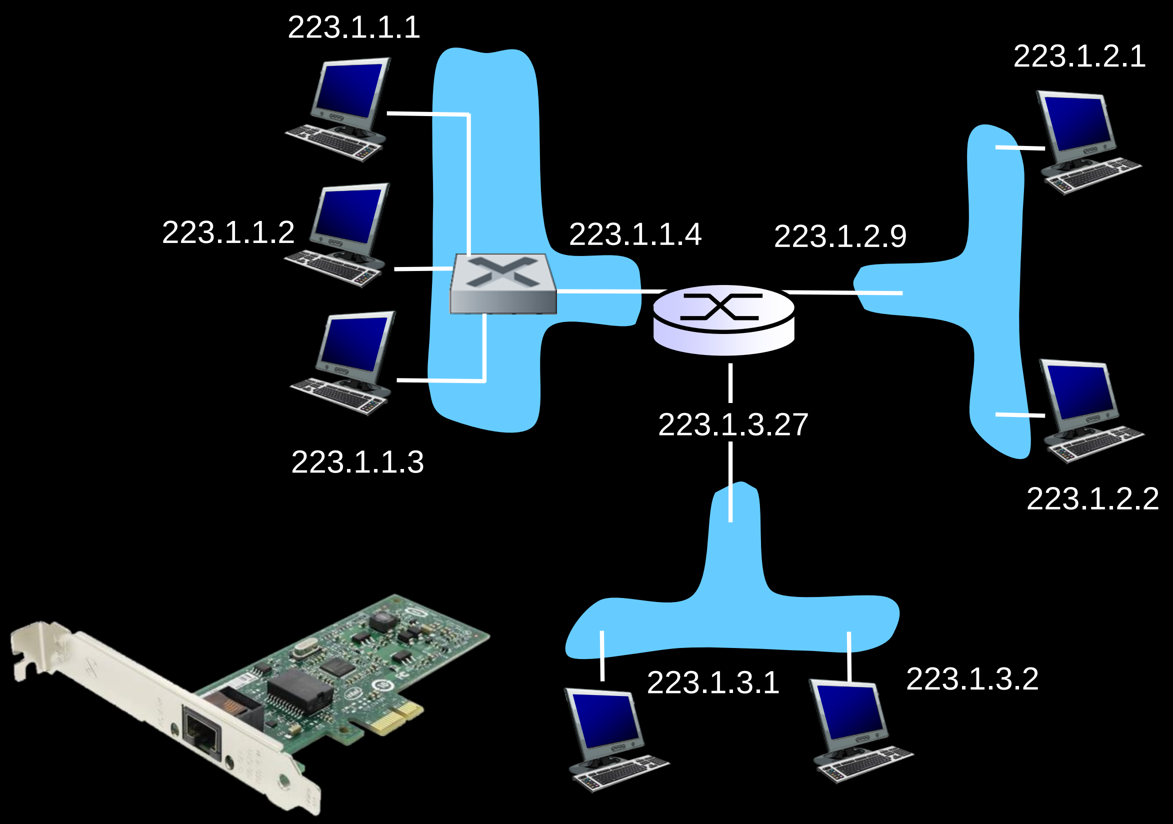


출처 – https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjTravfzYLcAhWNQN4KHcFRA44QjRx6BAgBEAU&url=https%3A%2F%2Fopenmaniak.com%2Fping.php&psig=AOvVaw1kKBGCdMZwSzgWsJQYYB-1&ust=1530695604173111

04. IP Addressing

Introduction to IP Address

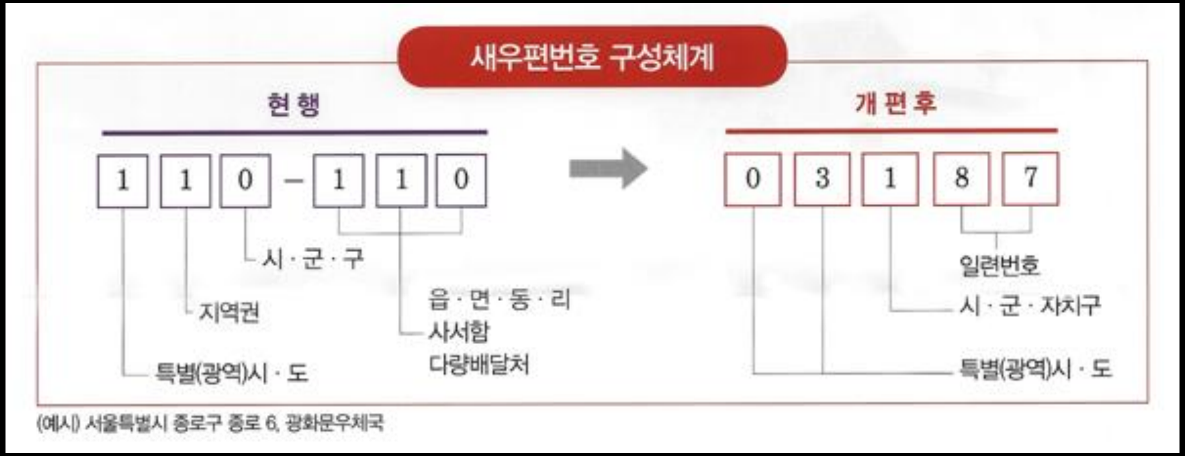
- IP address: 32-bit identifier for host, router interface
- Interface: connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- IP address associated with each interface



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

Hierarchical Addressing

- Korean postal code



출처 –
https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwidj-ru4lLcAhWCMN4KHT6ADjlQjRx6BAgBEAU&url=http%3A%2F%2Fwww.thedjnews.com%2Fnews%2FarticleView.html%3Fidxno%3D1493&psig=AOvVaw0Z4wtVpluH4N3PFxGbw2eQ&ust=1530701281466061

- Telephone number

+82-51-510-5555

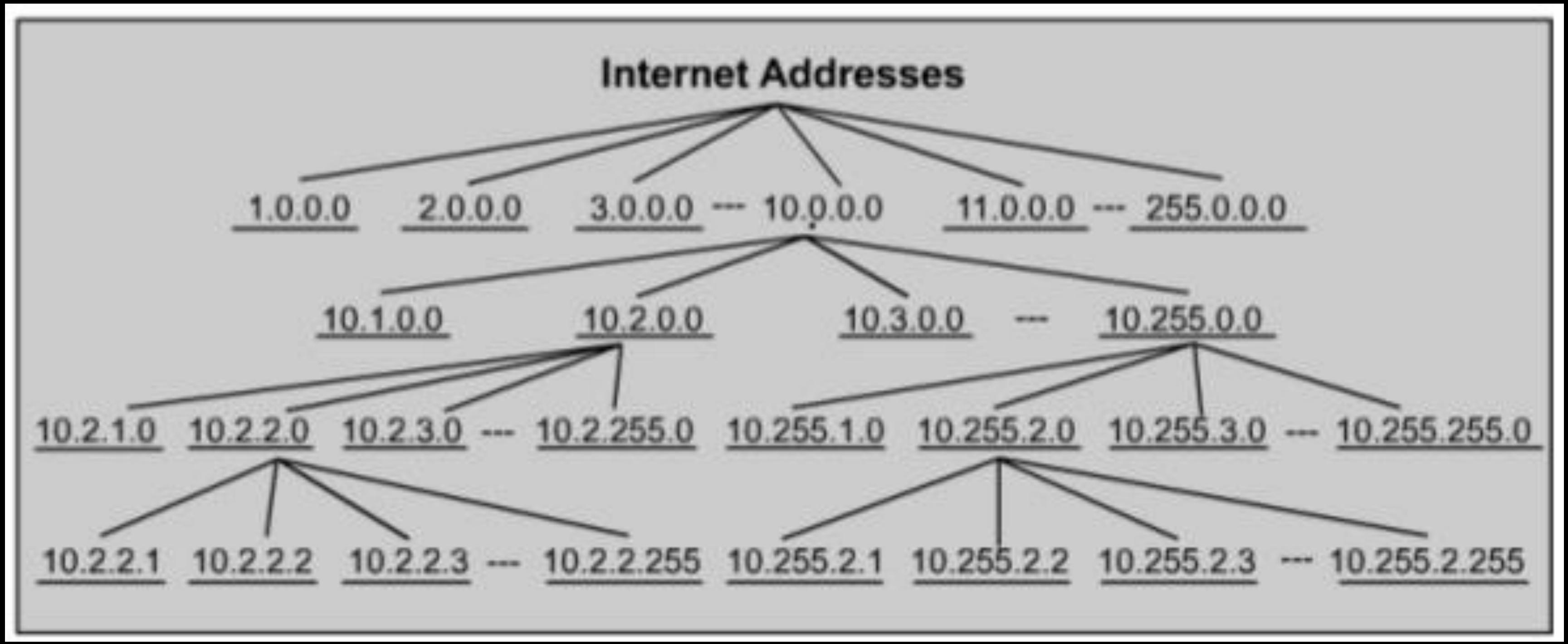
local exchange
area code
country code

- IP address

164.125.70.125

host id
Bldg. Comp. Eng.
Pusan National University

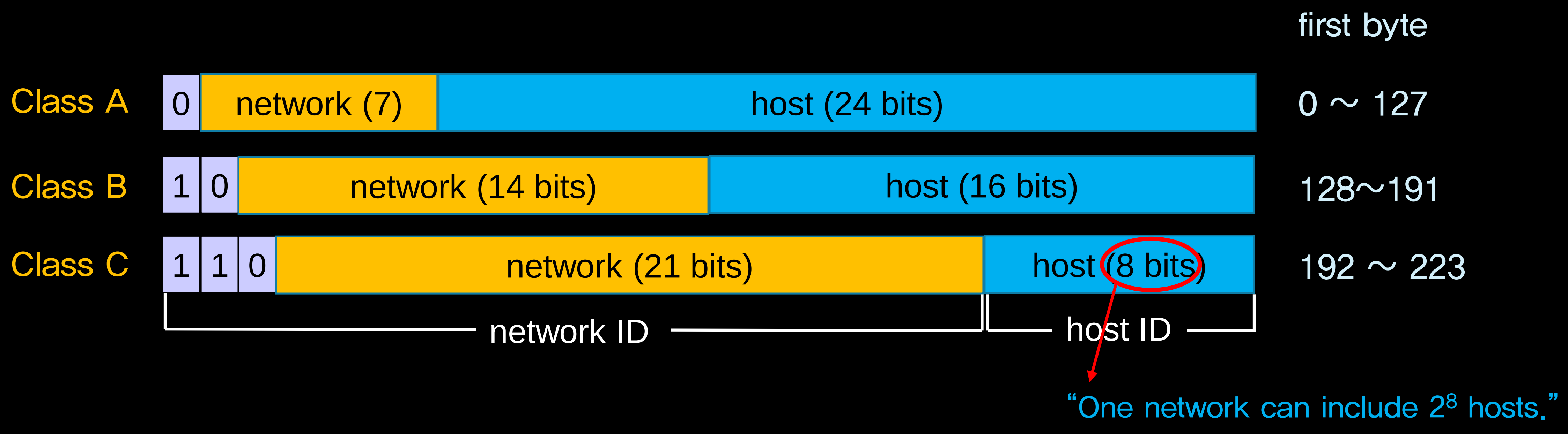
Example: Hierarchical IP Address



출처 –
https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj537vo44LcAhWVA4gKHWWiAPwQjRx6BAgBEAU&url=https%3A%2F%2Fwww.slideshare.net%2Fwelcometofacebook%2Fm06-35513859&psig=AOvVaw0kd7fRzS5vV-ls5QZC2_lB&ust=1530700263742351

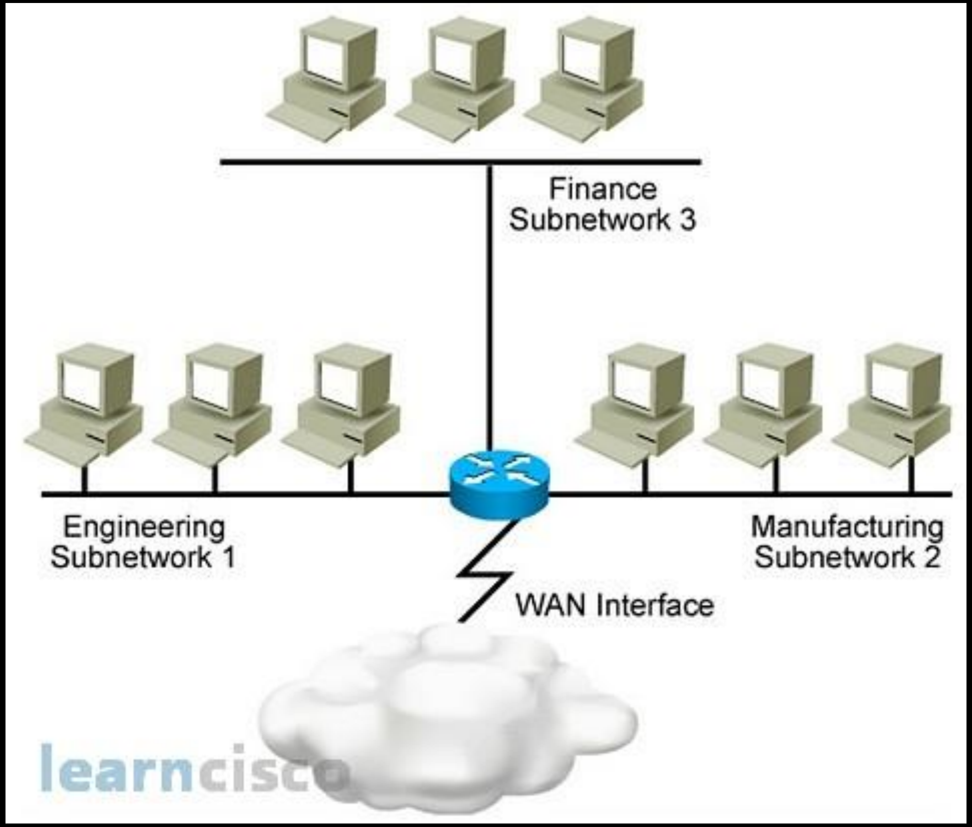
IP Address Class

- ICANN (Internet Corporation for Assigned Names and Numbers)
  - http://www.icann.org/
  - allocates addresses
  - manages DNS
  - assigns domain names, resolves disputes



Subnets

- Subnet: a logical subdivision of an IP network
- Why subnetworking?
  - Datagram forwarding performed by routers
  - Hosts in a same network can reach each other without intervening router
  - Too many hosts in a network increase maintenance overhead
  - "Divide and conquer"

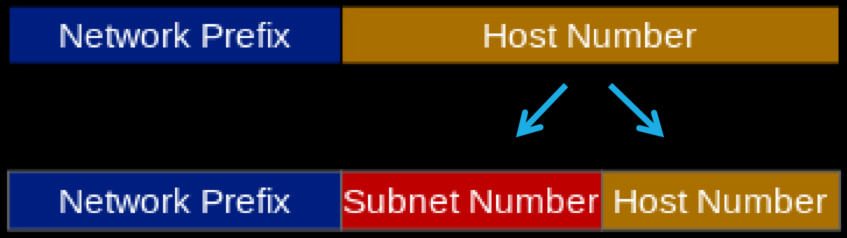


출처 – https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjWr4q3s4TcAhXGFogKHTNuC2gQjRx6BAgBEAU&url=http%3A%2F%2Fwww.learncisco.net%2Fcourses%2Ficnd-1%2Flan-connections%2Fnetwork-addressing-scheme.html&psig=AOvVaw24-bw_TTDQV85Eh-H39lpq&ust=1530757596051182

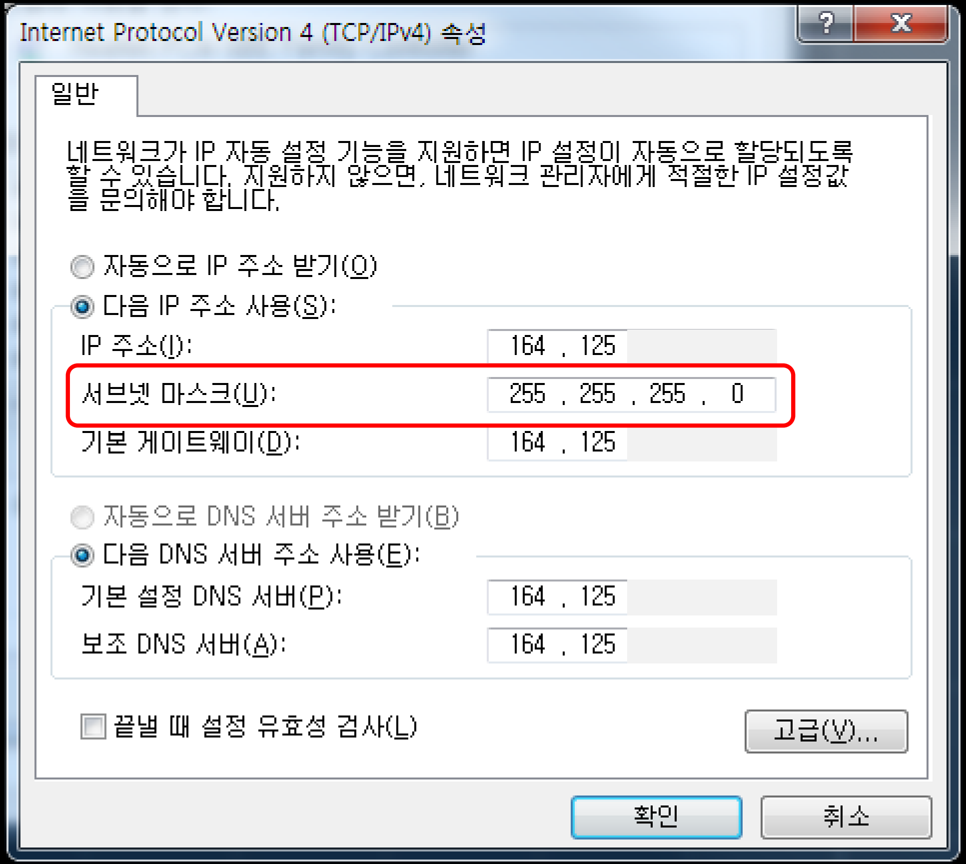
Subnet Mask

- Division of IP address
  - subnet part: high order bits of host id
  - host part: low order bits of host id

- How to decide the size of subnet number?
  - subnet mask: indicating the bits that will be used as the network number
  - e.g., 255.255.255.0 => 24 bits are used as the network number

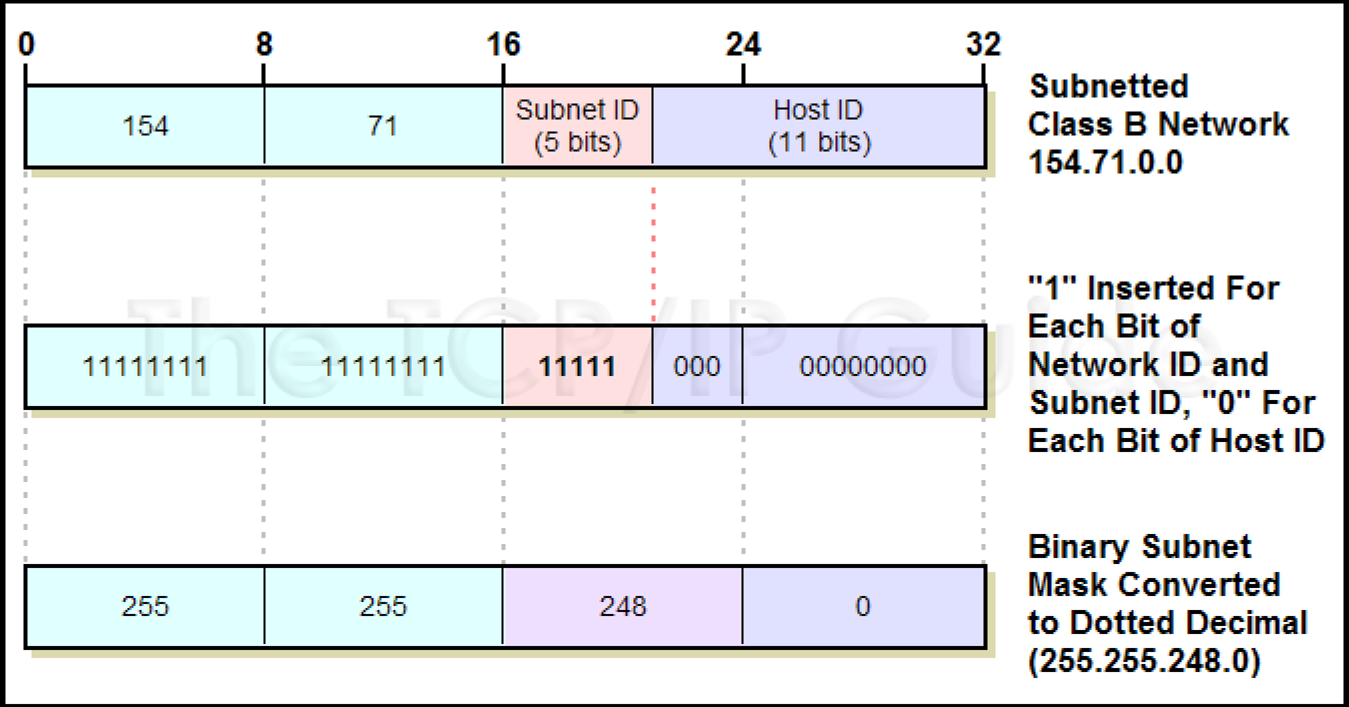


출처 – https://en.wikipedia.org/wiki/Subnetwork



Subnetworking Example

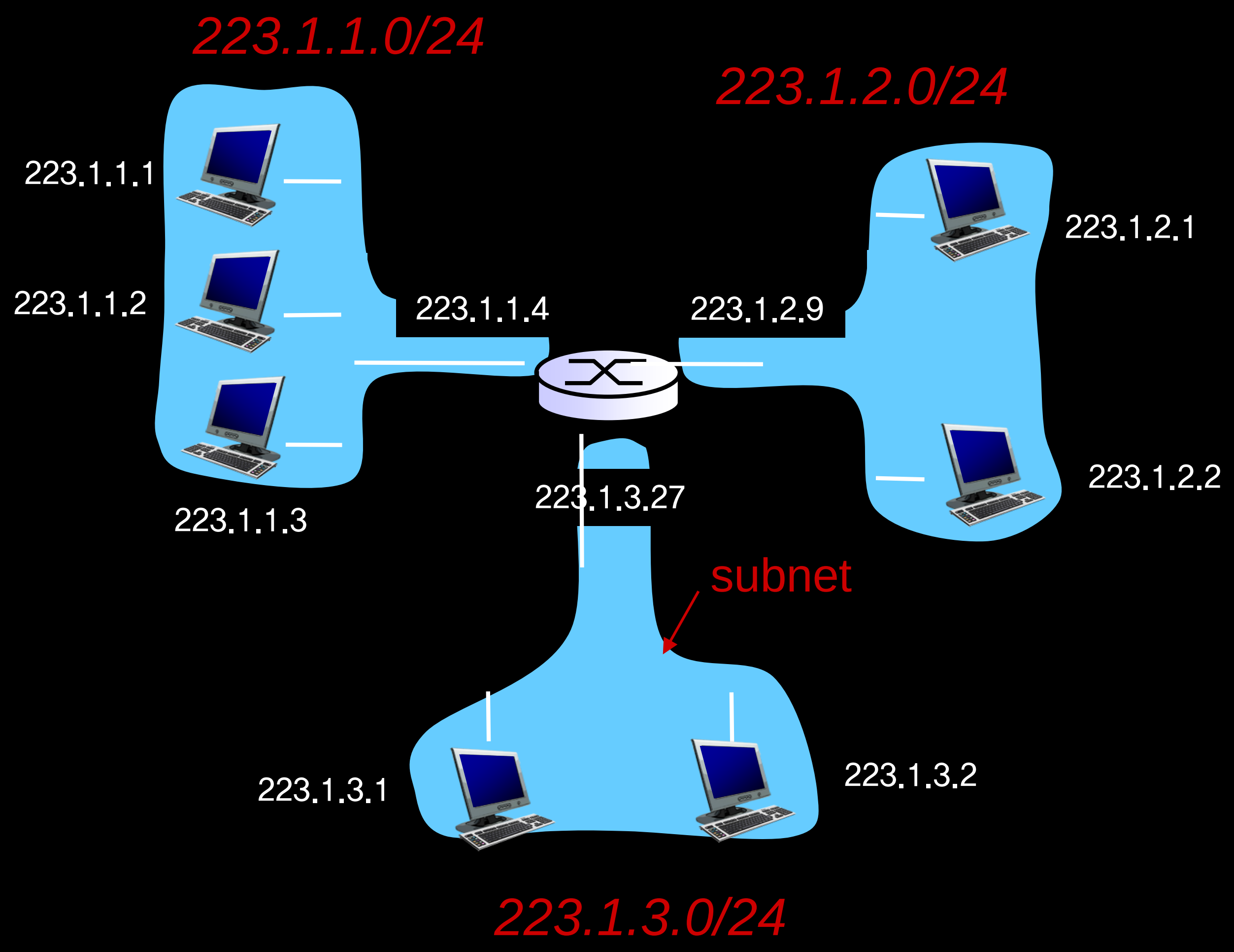
- e.g., subnet ID is 5 bits long
  - $2^5 = 32$ subnets can exist
  - each subnet can include 2^{11} hosts
  - subnet mask in
    - binary number
    - decimal number



출처 – https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwjut7mQroTcAhVCdt4KHcrgD1AQjRx6BAgBEAU&url=http%3A%2F%2Fwww.tcpipguide.com%2Ffree%2Ft_IPSubnetMasksNotationandSubnetCalculations-2.htm&psig=AOvVaw2fWClcQMJzlPoe3PYzjlZn&ust=1530756208997926

Subnetworking Example

- An IP network can be a subnet by itself
  - e.g., Class C network with subnet mask /24



subnet mask: /24

11111111 11111111 11111111 00000000

Classless Inter-Domain Routing (CIDR)

- A.k.a supernetting
- Address format: a.b.c.d/x, where x is # bits in network portion of address

subnet part ← → host part

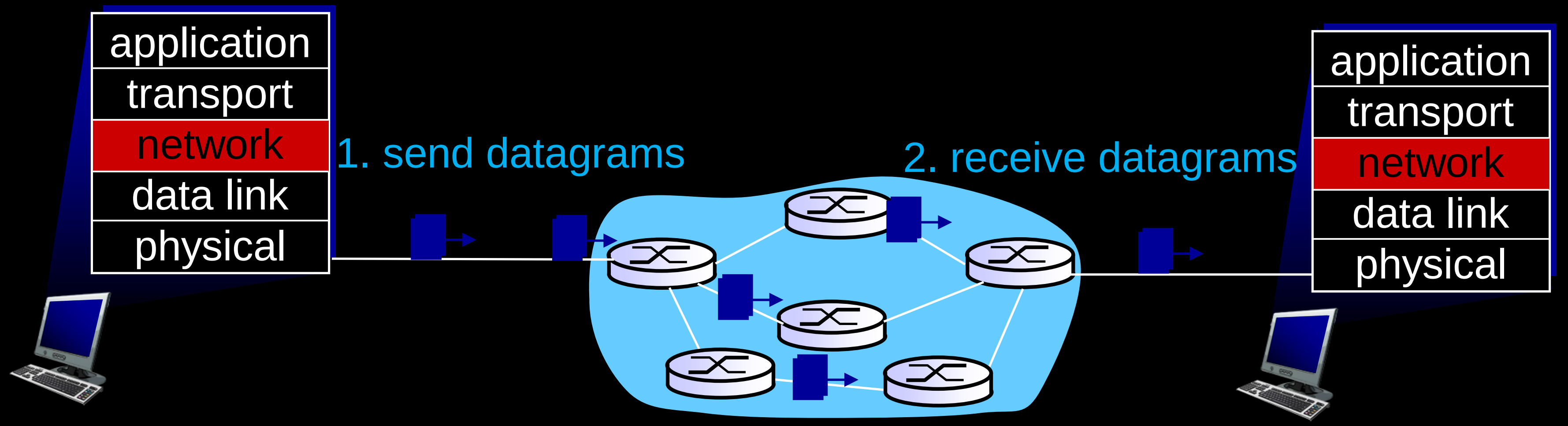
11000000	10101000	00000000	00000000	192.168.0.0/22
11000000	10101000	00000001	00000000	192.168.1.0/22
11000000	10101000	00000010	00000000	192.168.2.0/22
11000000	10101000	00000011	00000000	192.168.2.0/22

We can reference all these networks with a single route entry 192.168.0.0/22

05. Datagram Forwarding

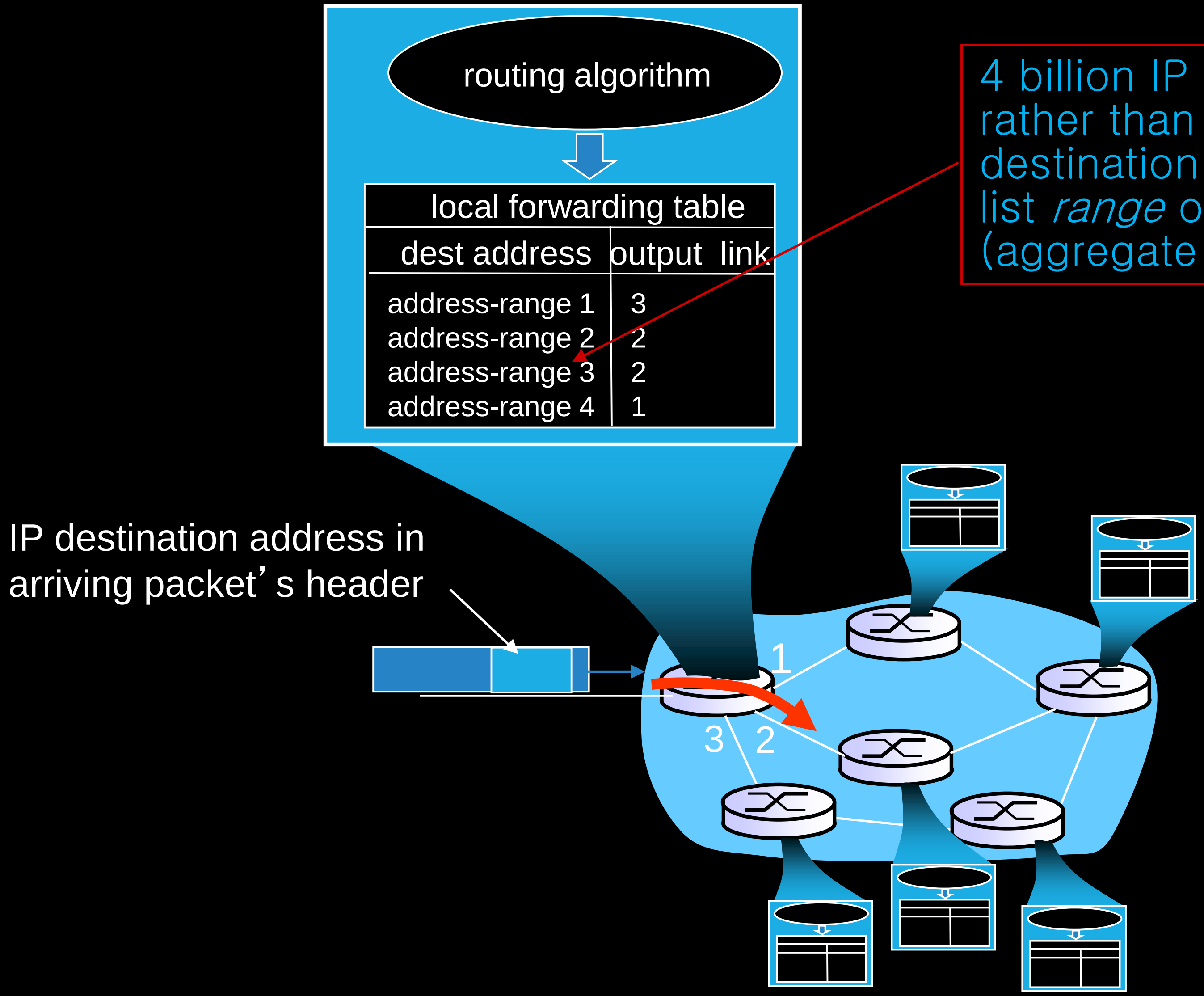
IP Network = Datagram Network

- No call setup at network layer
- Routers: no state about end-to-end connections
  - no network-level concept of "connection"



- Datagram forwarding
  - destination-based forwarding: forward based only on destination IP address (traditional)
  - generalized forwarding: forward based on any set of header field values (SDN)

Destination-based Forwarding Table



dest address	output link
address-range 1	3
address-range 2	2
address-range 3	2
address-range 4	1

4 billion IP addresses, so rather than list individual destination address list *range* of addresses (aggregate table entries)

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

Q: but what happens if ranges don't divide up so nicely?

Longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

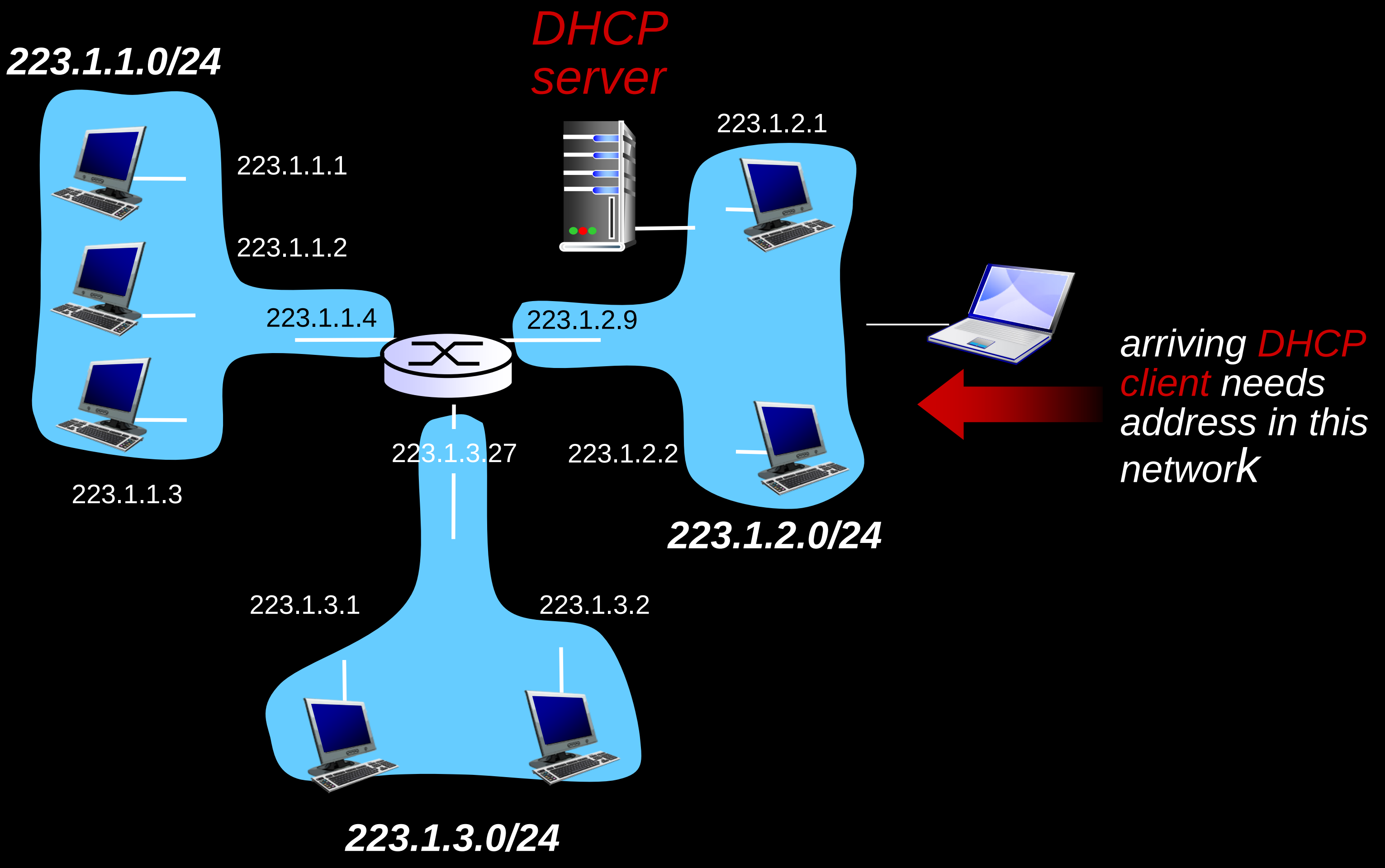
DA: 11001000 00010111 00010110 10100001 which interface?

DA: 11001000 00010111 00011000 10101010 which interface?

06. Dynamic Host Config. Protocol

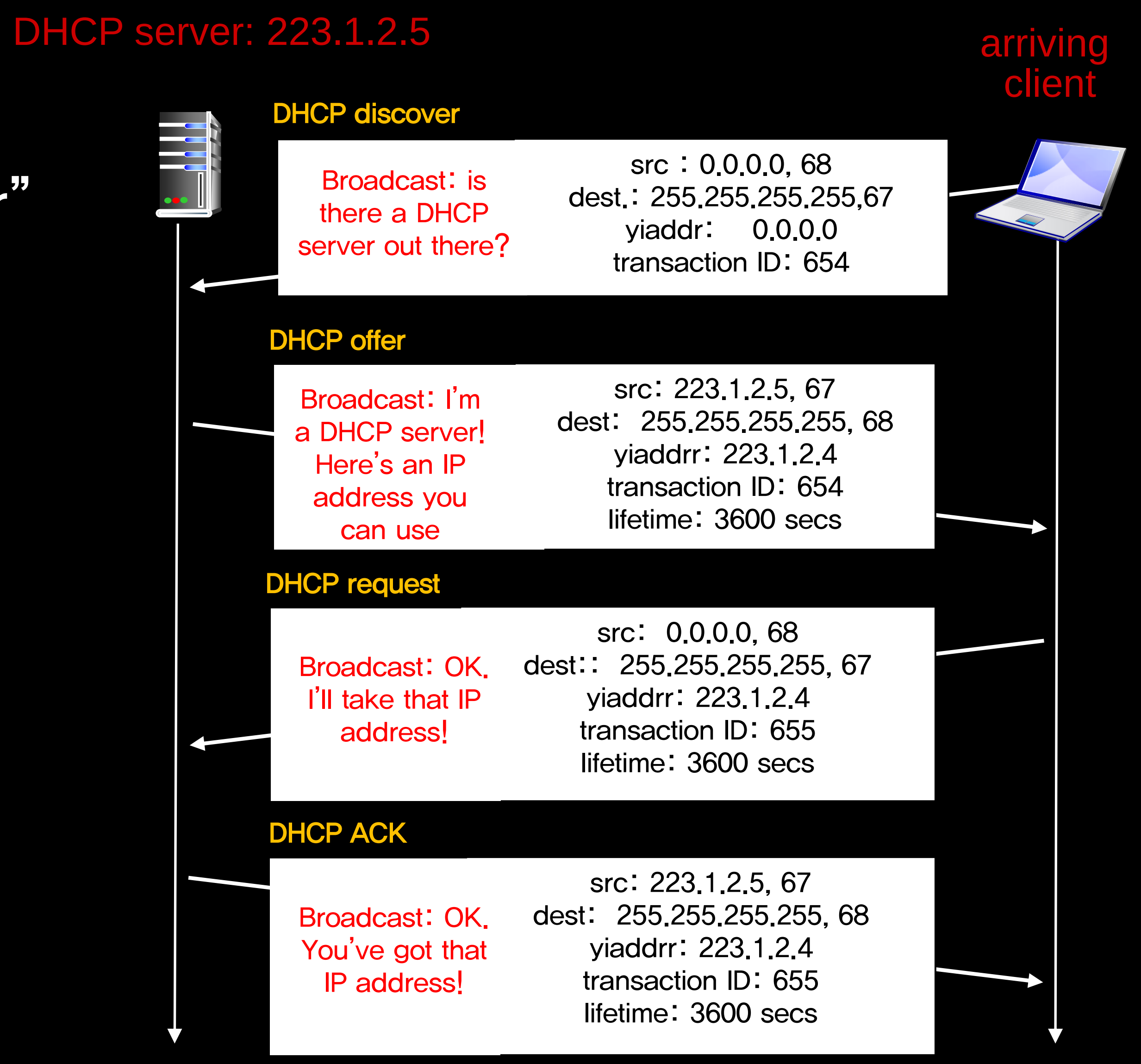
DHCP (Dynamic Host Configuration Protocol)

- Host dynamically obtains its IP address from network server when it joins network
  - can renew its lease on address in use
  - allows reuse of addresses (only hold address while connected/“on”)
  - support for mobile users who want to join network (more shortly)

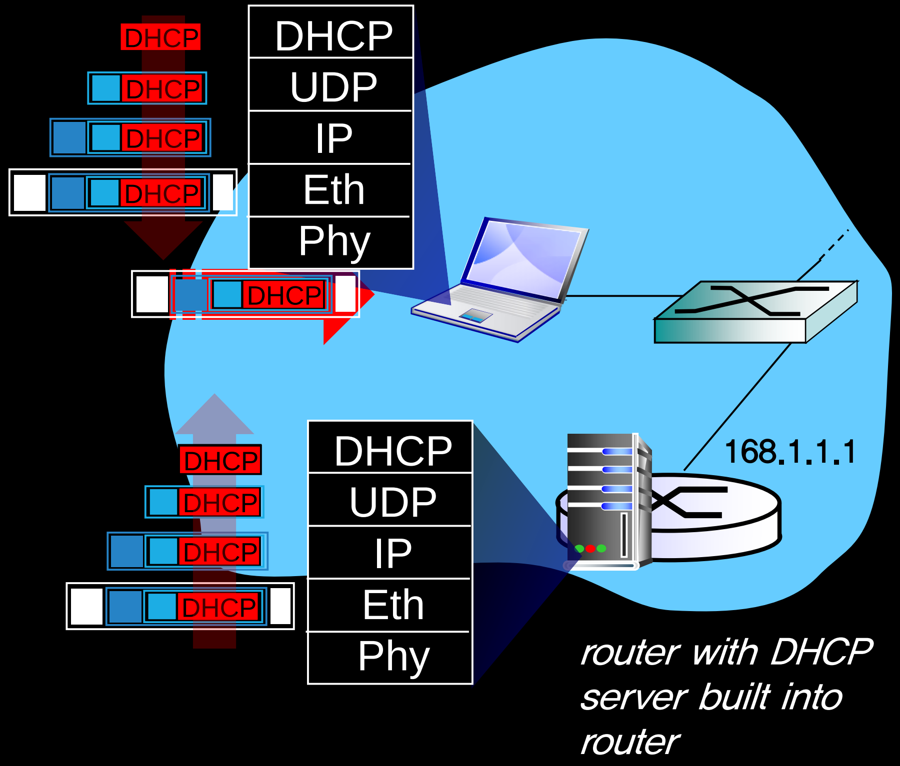


DHCP Overview

1) host broadcasts "DHCP discover"
2) DHCP server responds with "DHCP offer"
3) host requests IP address: "DHCP request"
4) DHCP server sends address: "DHCP ack"

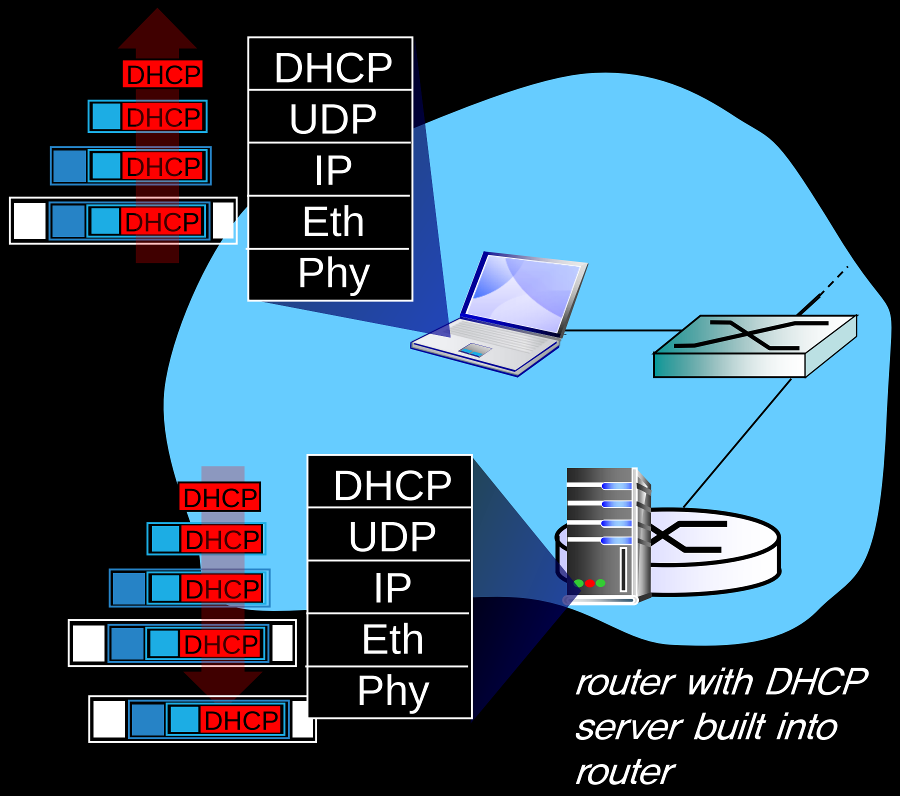


DHCP Example



- Connecting laptop needs its IP address, address of first-hop router, address of DNS server
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet
- Ethernet frame broadcast (dest MAC addr: FFFFFFFFFFFF) on LAN, received at router running DHCP server
- Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

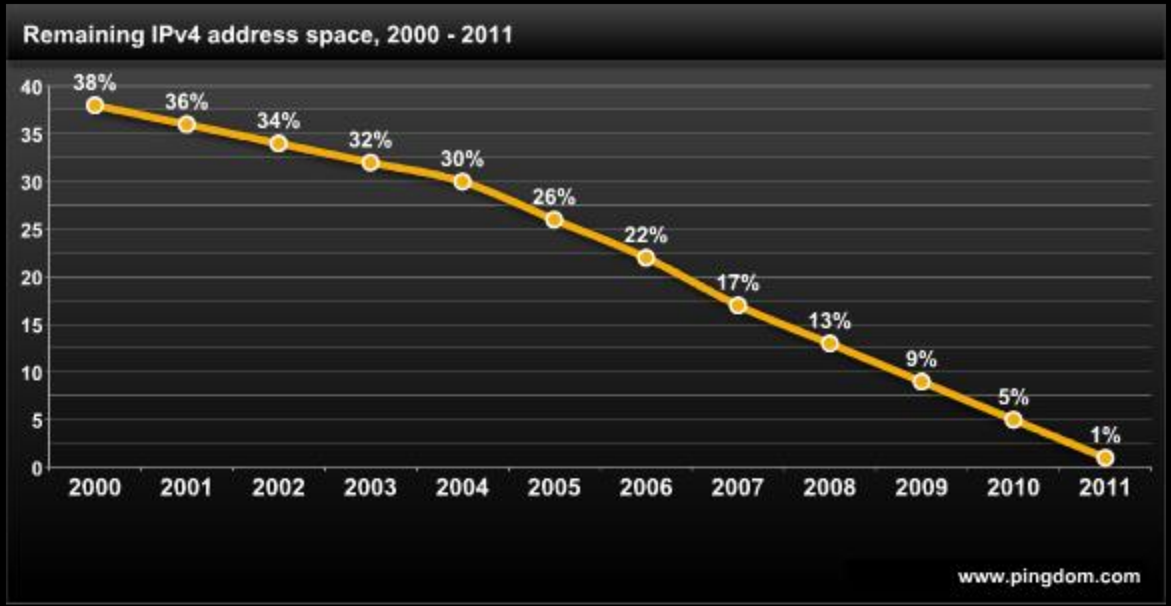
DHCP Example (cont'd)



- DCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- Encapsulation of DHCP server, frame forwarded to client, demuxing up to DHCP at client
- Client now knows its IP address, name and IP address of DSN server, IP address of its first-hop router

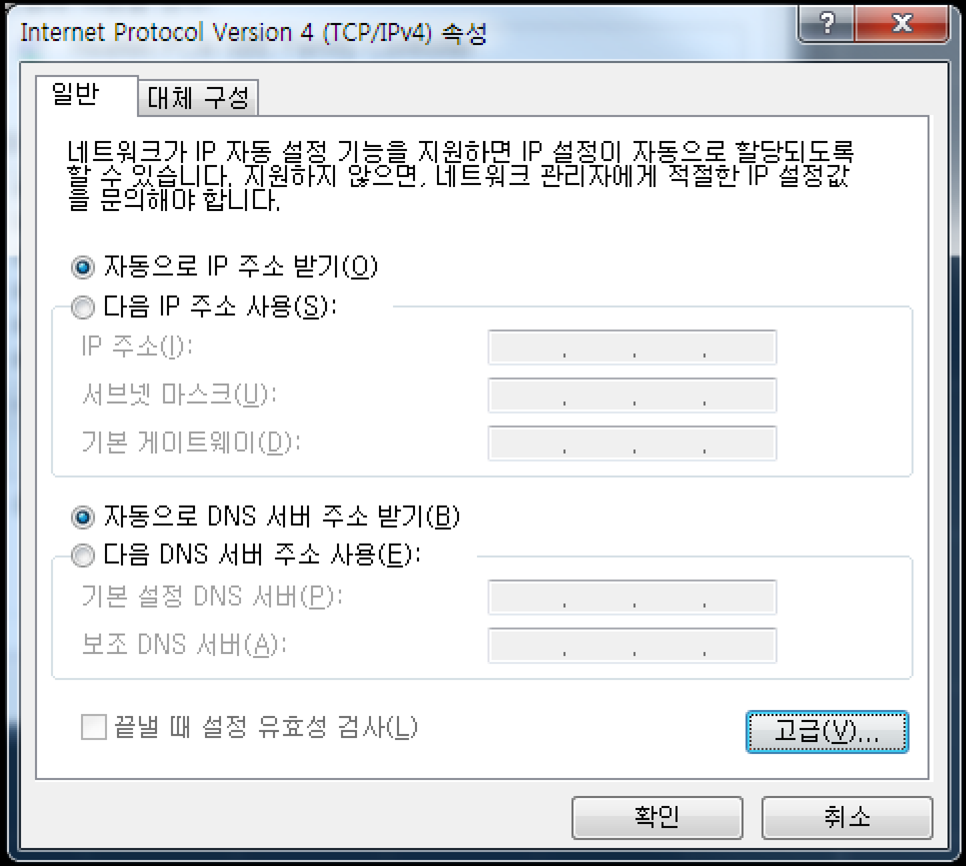
Advantages of DHCP

- Efficient use of IP addresses
  - one solution to the "IP address exhaustion" problem



출처 – https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiCprSz74TcAhWKFogKHTn6AUcQjRx6BAgBEAU&url=https%3A%2F%2Froyal.pingdom.com%2F2009%2F03%2F06%2Fa-crisis-in-the-making-only-4-of-the-internet-supports-ipv6%2F&psig=AOvVaw2mAkJPtS508ebKlOCFWFZp&ust=1530773146360152

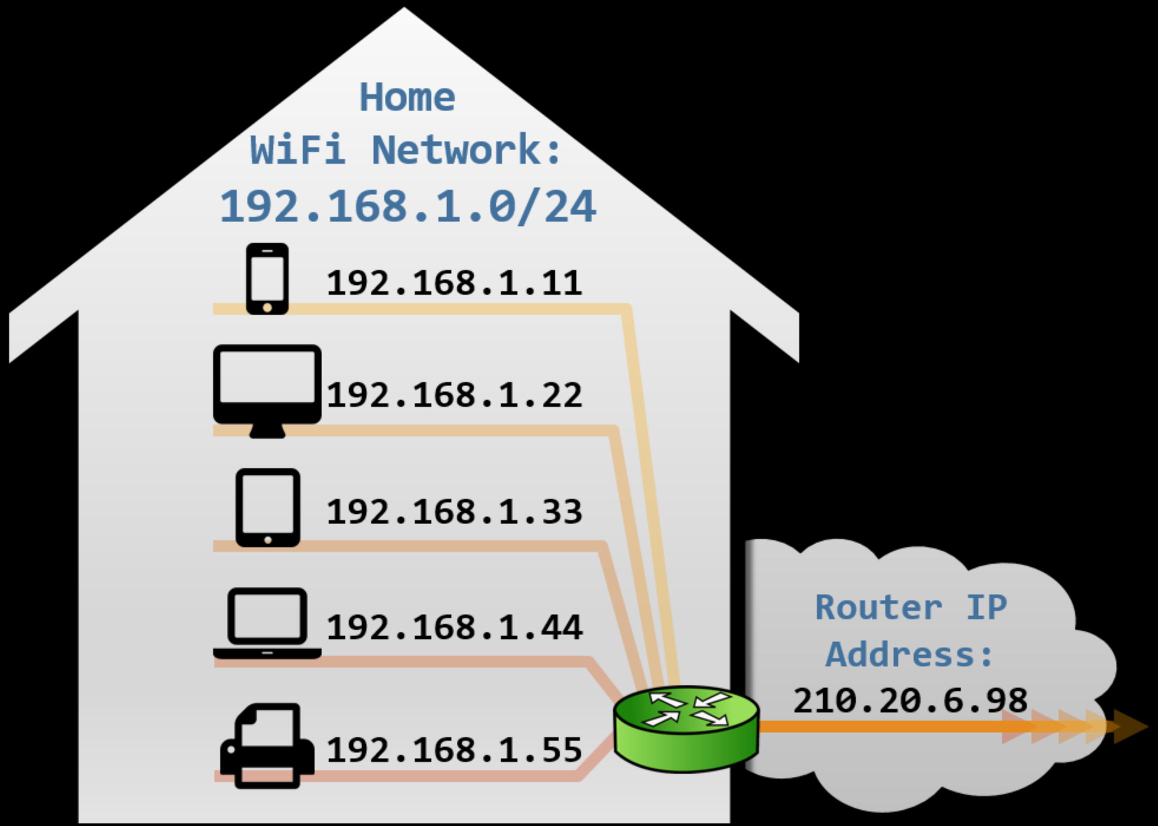
- Allow the "plug & play" of a computer system



07. Network Address Translation

NAT (Network Address Translation)

- Motivation: local network uses just one IP address as far as outside world is concerned:
  - range of addresses not needed from ISP: just one IP address for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus)



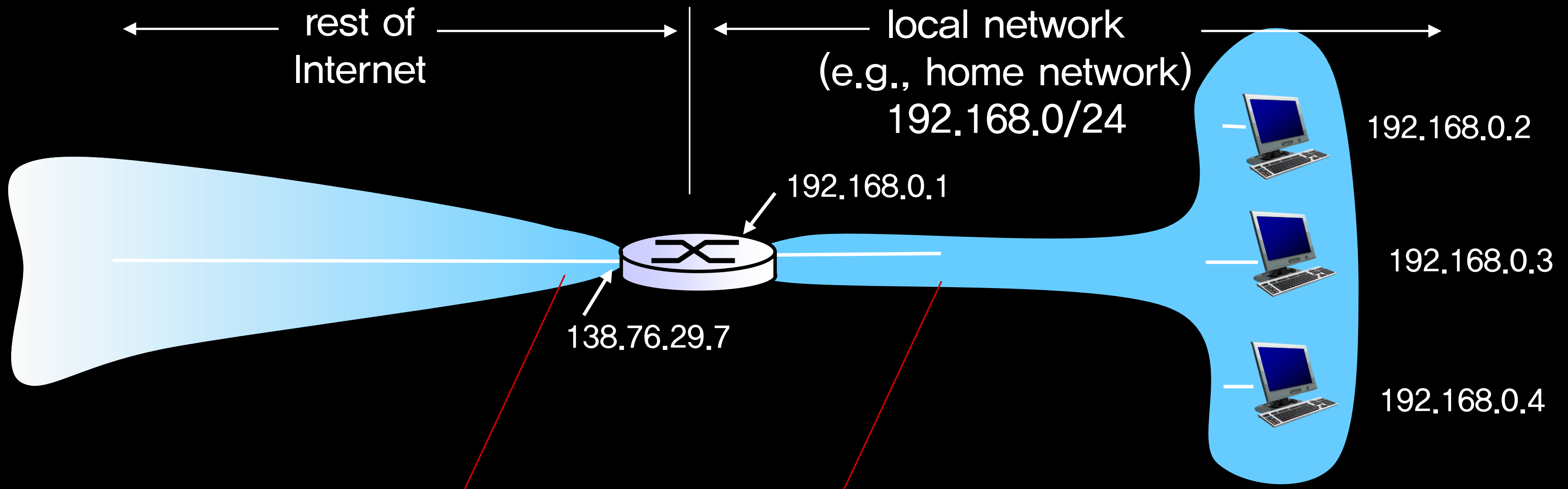
출처 – https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwii9YPw74TcAhWUZt4KHYuxABUQjRx6BAgBEAU&url=http%3A%2F%2Fwww.practicalnetworking.net%2Fseries%2Fnat%2Fwhy-nat%2F&psig=AOvVaw3Rck0oh7KTX3cP0a6Pv5w9&ust=1530773925619362

NAT Example

A class: 10.x.x.x
B class: 172.16.x.x ~ 172.31.x.x
C class: 192.168.0.x ~ 192.168.255.x

Public IP part

Private IP part



all datagrams *leaving* local network have the *same* single source NAT IP address: 138.76.29.7,different source port numbers

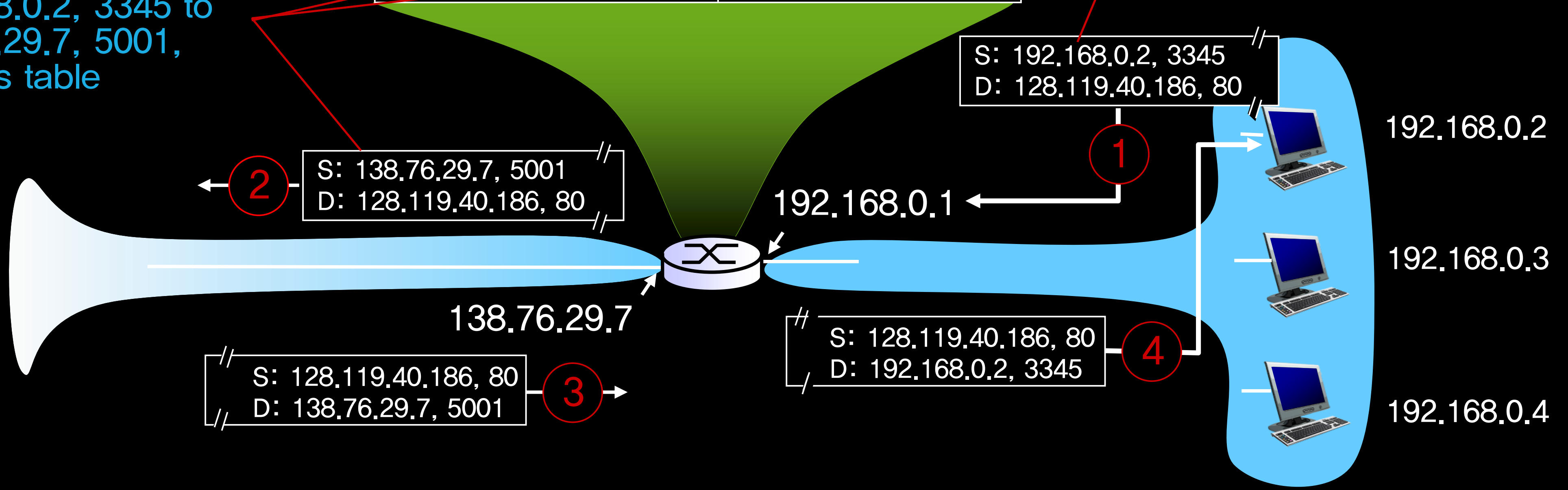
datagrams with source or destination in this network have 192.168.0/24 address for source, destination (as usual)

NAT Example (cont'd)

NAT translation table	
WAN side addr	LAN side addr
138.76.29.7, 5001	192.168.0.2, 3345
……	……

1: host 192.168.0.2 sends datagram to 128.119.40.186, 80

2: NAT router changes Datagram source addr from 192.168.0.2, 3345 to 138.76.29.7, 5001, updates table

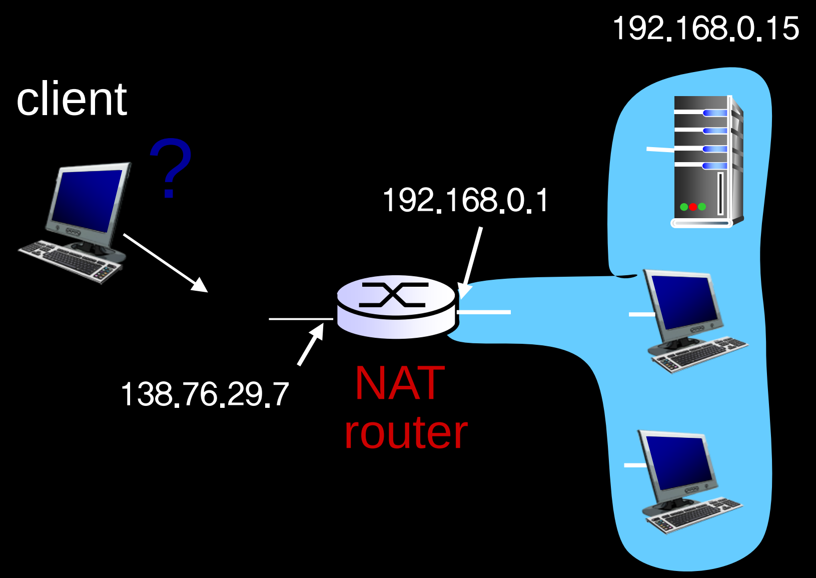


3: reply arrivesdest. address:138.76.29.7, 5001

4: NAT router changes datagram dest addr from138.76.29.7, 5001 to 192.168.0.2, 3345

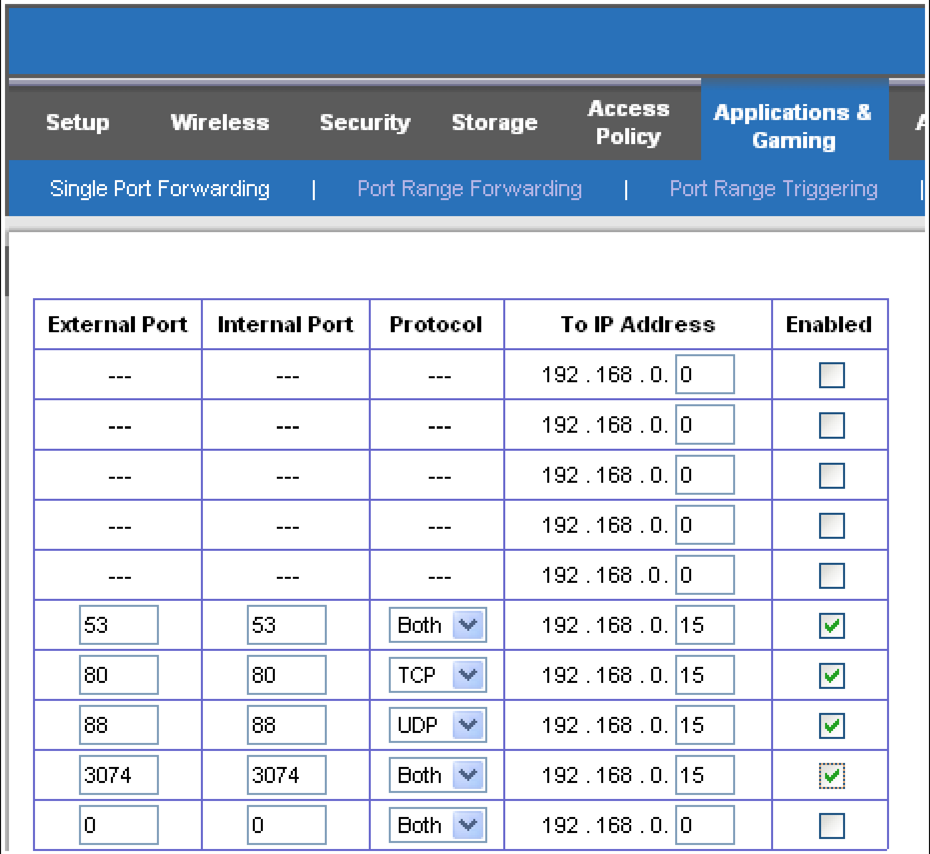
NAT Address Traversal Problem

- Client wants to connect to server at address 192.168.0.15
- However, client cannot use it as destination address because it is a private address
- Only one externally visible NATed address: 138.76.29.7



Port Forwarding

- Statically configure NAT to forward incoming connection requests at given port to server
- e.g., “138.76.29.7, port 80” always forwarded to “192.168.0.15, port 80”

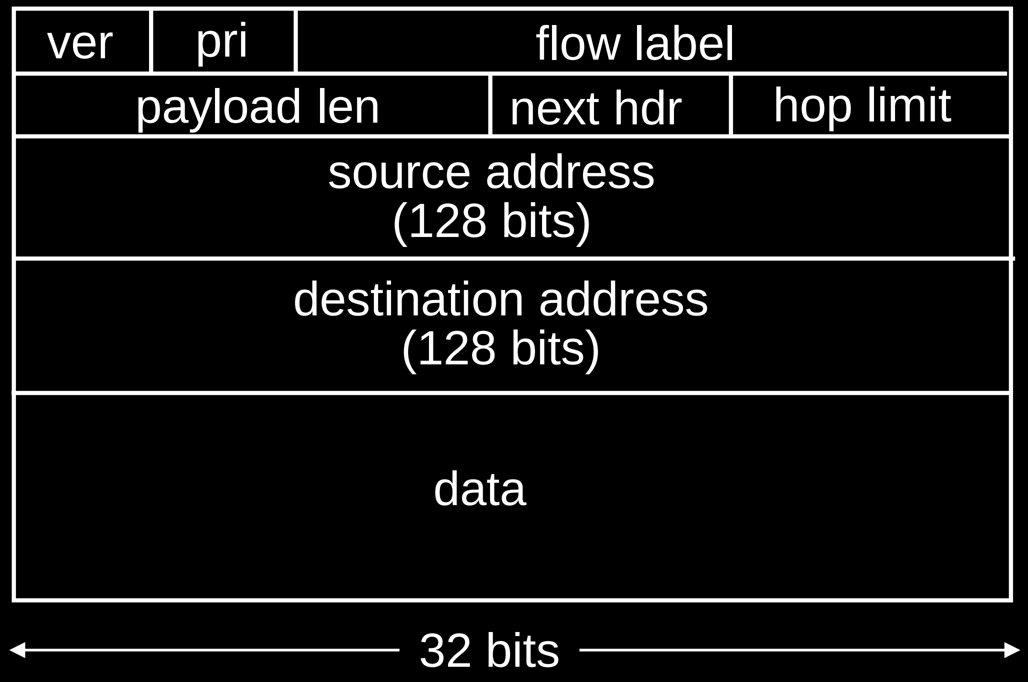


출처 –
https://www.google.co.kr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj638Pw84TcAhXl7GEKHSSoDiMQjRx6BAgBEAU&url=http%3A%2F%2Fwww.linksys.com%2Fus%2Fsupport-article%3FarticleNum%3D136711&psig=AOvVaw09SlD6bZzholqucUCF2OVm&ust=1530775024822632

08. IPv6

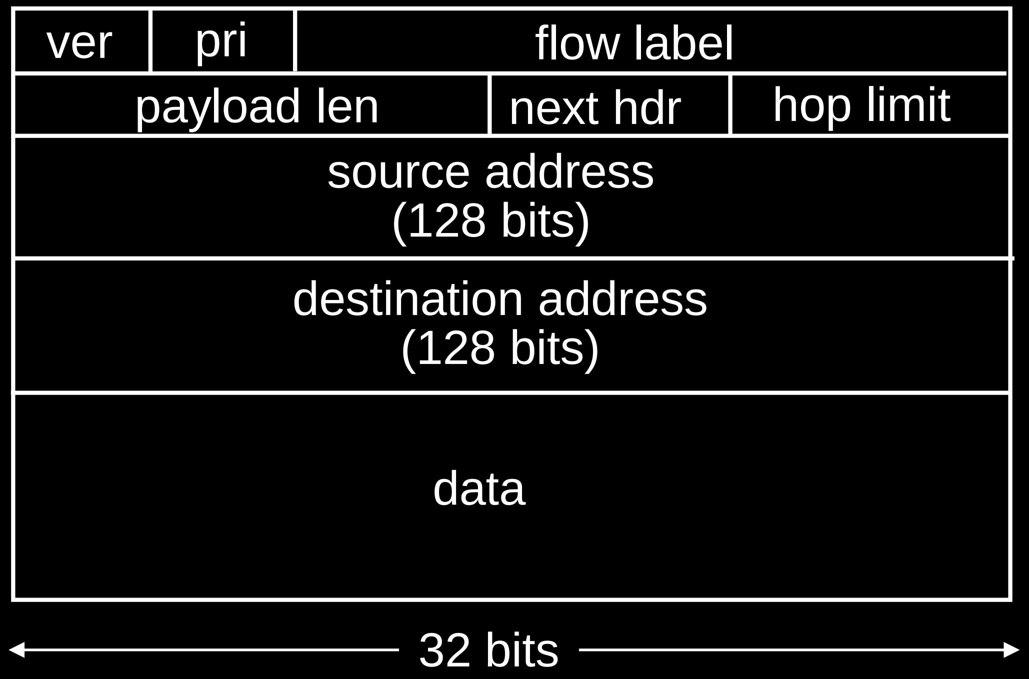
IPv6 Motivation

- Initial motivation
  - In the early 1990s, the IETF began an effort to develop a successor to the IPv4 protocol to respond to the depletion of IP addresses
  - In Feb. 2011, IANA allocated out the last remaining pool of unassigned IPv4 addresses to a regional registry
- Additional motivation
  - header format changes to speed up processing/forwarding
  - header changes to facilitate QoS
- IPv6 datagram format
  - fixed-length 40 byte header
  - no fragmentation allowed



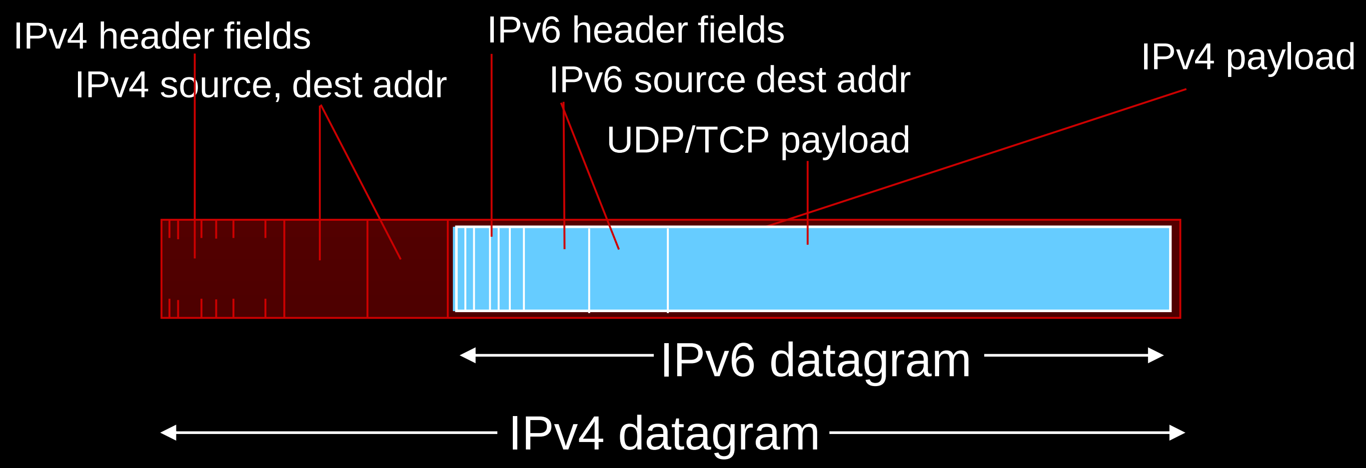
IPv6 Datagram Format

- priority: identify priority among datagrams in flow
- flow Label: identify datagrams in same “flow”
- next header: identify upper layer protocol for data
- hop limit: same as TTL in IPv4
- options: allowed, but outside of header, indicated by “Next Header” field
- Checksum removed entirely to reduce processing time at each hop

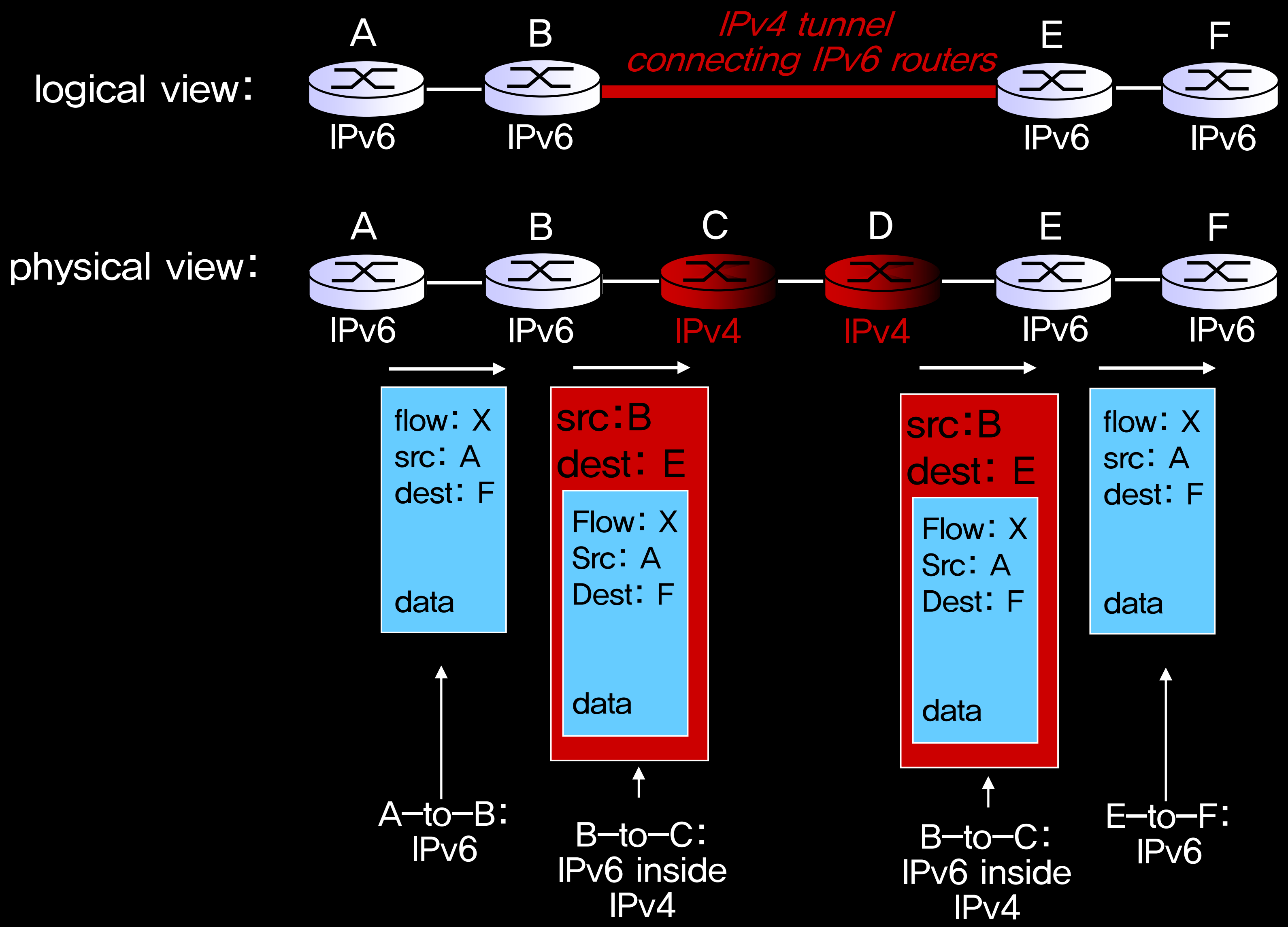


Migration from IPv4 to IPv6

- Not all routers can be upgraded simultaneously
  - no "flag days"
  - how will network operate with mixed IPv4 and IPv6 routers?
- Tunneling: IPv6 datagram carried as payload in IPv4 datagram among IPv4 routers



Tunneling



logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6
IPv6
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
A-to-B:
IPv6
B-to-C:
IPv6 inside
IPv4
B-to-C:
IPv6 inside
IPv4
E-to-F:
IPv6

IPv6 Adoption

- Long (long!) time for deployment!
  - Google: 8% of clients access services via IPv6
  - NIST: 1/3 of all US government domains are IPv6 capable
  - in Korea: IPv6 implemented since 2008, but just less than 0.1% of the 3.4 million domains have been IPv6-enabled

- In the future,
  - In preparation for the IoT era, major international sites (such as Google, Facebook, Amazon, LinkedIn, YouTube, and Netflix) have completed and are servicing IPv6
  - To facilitate the use of IPv6, KISA (Korea Internet & Security Agency) supports a total budget of 1 billion won, mainly for mobile service providers with high adoption

Summary

Summary (1/2)

01

Overview of Network layer
- two key functions of network layer: routing and forwarding
- architecture of traditional network & software-define network

02

Inside of Router
- components: input port, output port, switching fabric
- queuing and scheduling

03

Internet Protocol Overview
- IPv4 datagram format
- fragmentation and reassembly

04

IP Addressing
- hierarchical IP addressing
- subnet and CIDR

Summary (2/2)

05 **Datagram Forwarding**
- destination–based forwarding
- longest prefix matching

06 **Dynamic Host Configuration Protocol**
- dynamic and automatic allocation of IP address to host
- advantages of DHCP

07 **Network Address Translation**
- translation between public and private address
- network address traversal

08 **IPv6**
- resolution for the IP address exhaustion
- header changes that help speed processing/forwarding