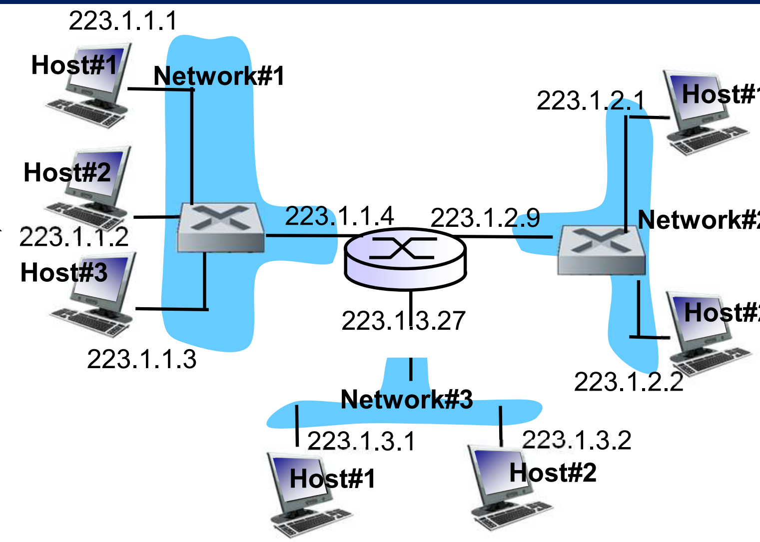
IP (V4) ADDRESSING: INTRODUCTION

- *IP (v4) 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 few (one/two?) interfaces (e.g., Ethernet, WiFi)



- *IP addresses are associated with each interface of L3 device*
  - *32-bits as 4-octets define {network ID, Host ID}*

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

- *Historical Class-based Addressing:*

Address Class	Address Range	First Octet	Subnet Mask	# IP Addresses in n/w	# Networks
Class A (0)	0. 0 – 127.255.255.255	0-127	255.0.0.0 (8)	2^{24}	128 (2^7)
Class B (10)	128.0 – 191.255.255.255	128-191	255.255.0.0 (16)	2^{16}	16384 (2^{14})
Class C (110)	192.0 – 223.255.255.255	192-223	255.255.255.0 (24)	2^8	2097152 (2^{21})
Class D (1110)	224.0 – 239.255.255.255	224-239	-	2^{28} (multicast groups)	-
Class E (1111)	240.0 – 255.255.255.255	240-255	-	2^{28} (reserved space)	-

IP ADDRESSING: CIDR

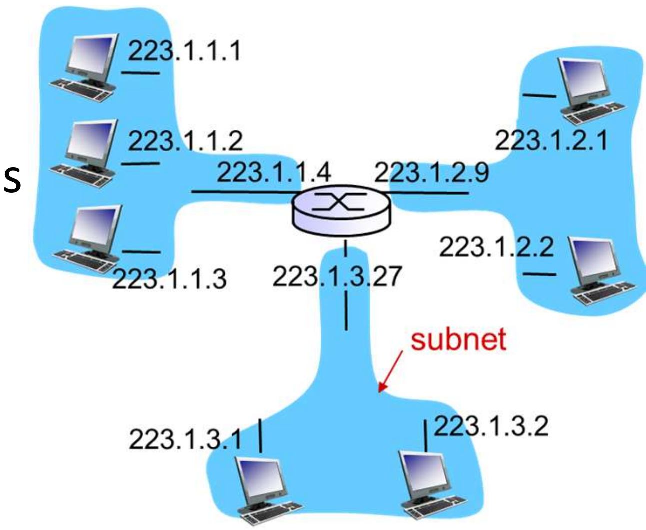
CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary length
- Format: a.b.c.d/x; where x is # bits in subnet portion of address

subnet part ← → host part

11001000 00010111 00010000 00000000

200.23.16.0/23



network consisting of 3 subnets

- **IP address:**
  - subnet part - high order bits
  - host part - low order bits

- ***What's a subnet ?***
  - Each isolated network
  - device interfaces with same subnet part of IP address
  - can physically reach each other *without intervening router*

IP ADDRESSES: HOW TO GET ONE?

Q: how does *network* get subnet part of IP addr?

A: gets allocated portion of its provider ISP's address space

ISP's block	11001000 00010111 00010000 00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000 00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010 00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100 00000000	200.23.20.0/23
...	....	....
Organization 7	11001000 00010111 00011110 00000000	200.23.30.0/23

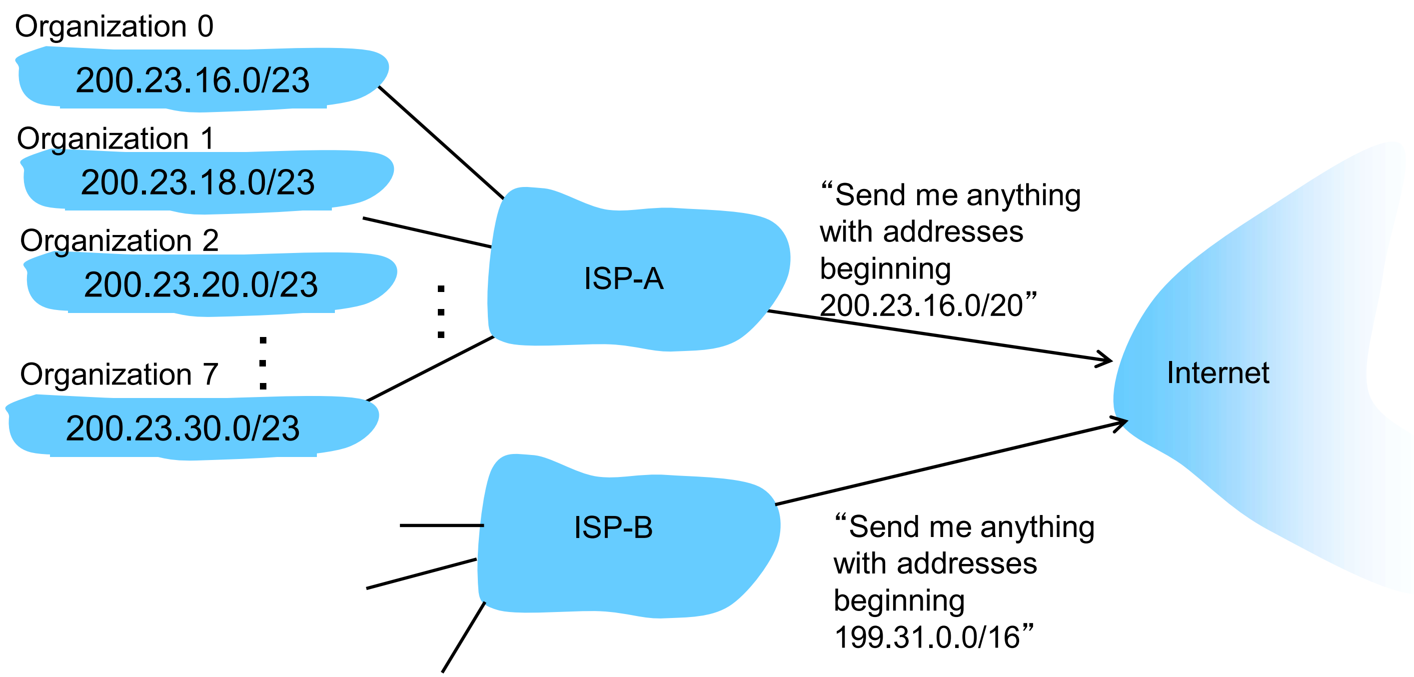
Q: how does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses (Public IPs)
- manages DNS
- assigns domain names, resolves disputes

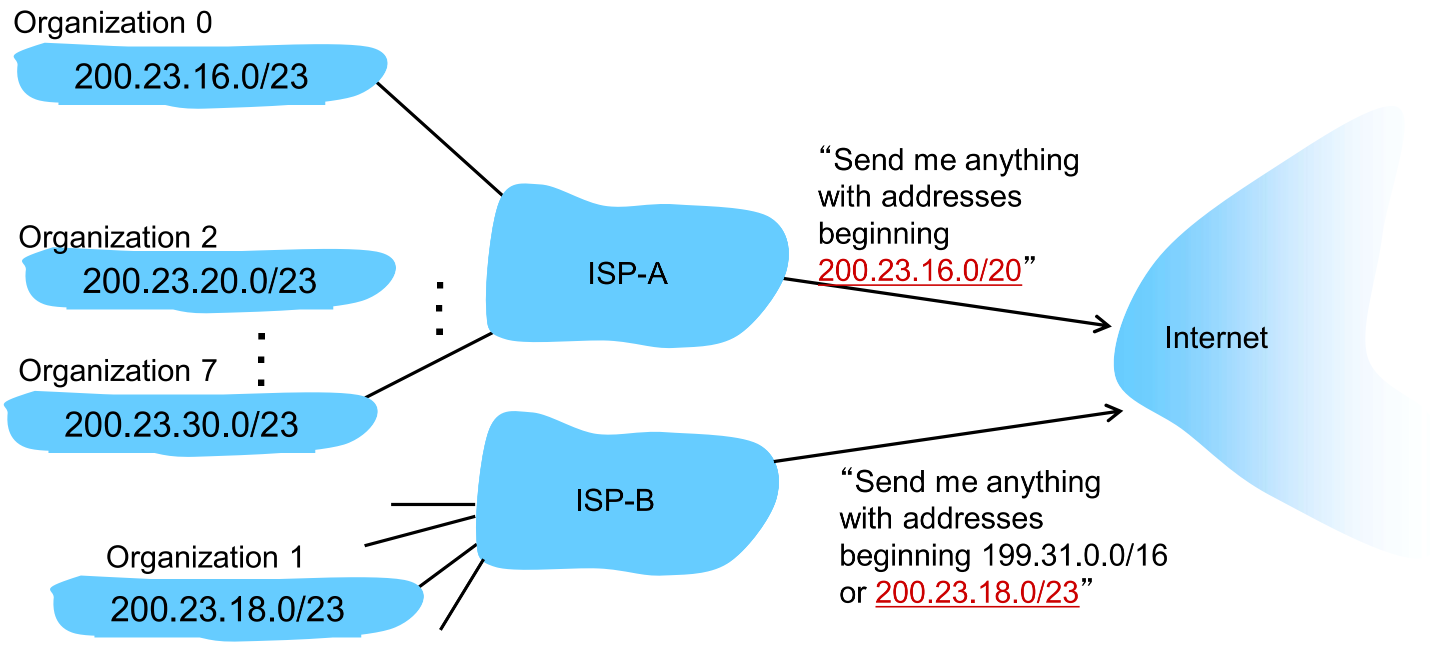
HIERARCHICAL ADDRESSING: ROUTE AGGREGATION

hierarchical addressing allows efficient advertisement of routing information:



HIERARCHICAL ADDRESSING: MORE SPECIFIC ROUTES

ISP-B has a more specific route to Organization 1

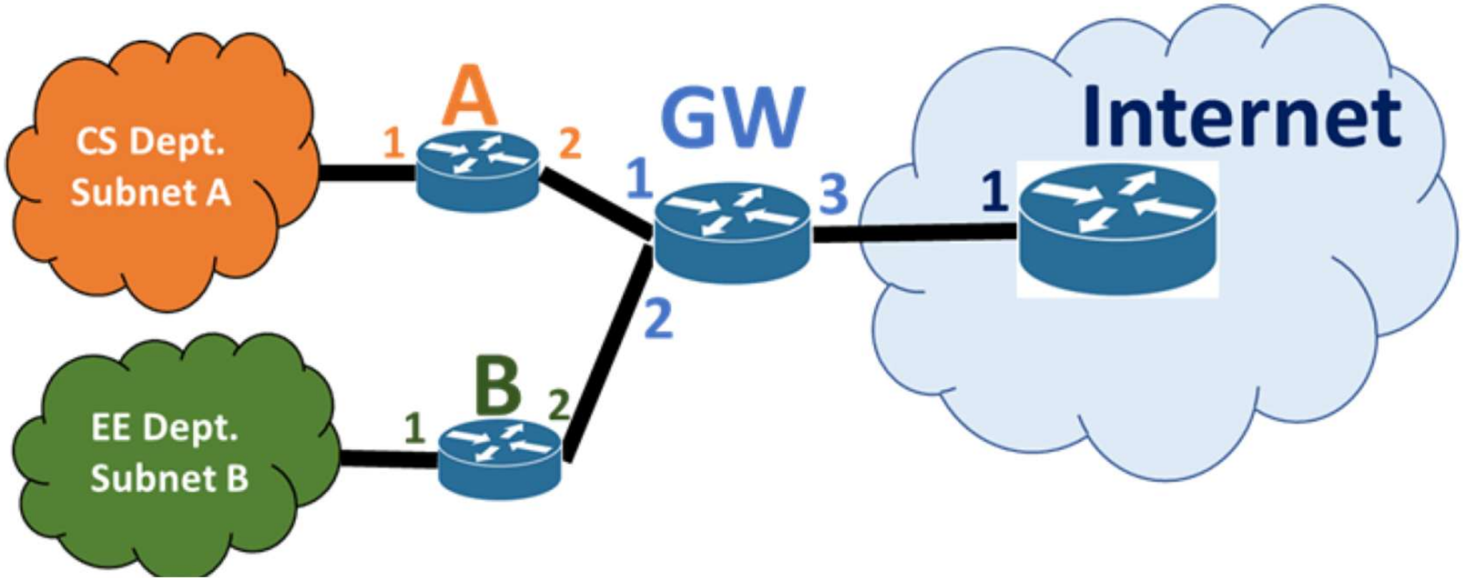


SAMPLE QUESTIONS

2. An Institute has the following chunk of CIDR-based IP addresses available with it: **128.160.128.0/20**. From this chunk, it needs to allocate half of the addresses to the CS department, say Subnet A, and a quarter to the EE department, say Subnet B. Suggest how it can allocate the addresses (CIDR block) accordingly. Precisely answer the following: Note: range of IP addresses means Start and End IPs. **(10 pts)**
    a. Current IP address range, total number of hosts the Institute can support **(2 pts)**
    b. CIDR block to be allocated for CS department (subet A) **(1 pts)**
    c. Start and End IP addresses for Subnet A **(2 pts)**
    d. CIDR block to be allocated for EE Department (subnet B) **(1 pts)**
    e. Start and End IP addresses for Subnet B **(2 pts)**
    f. Start and End IP addresses for the addresses retained by Institute **(2 pts)**

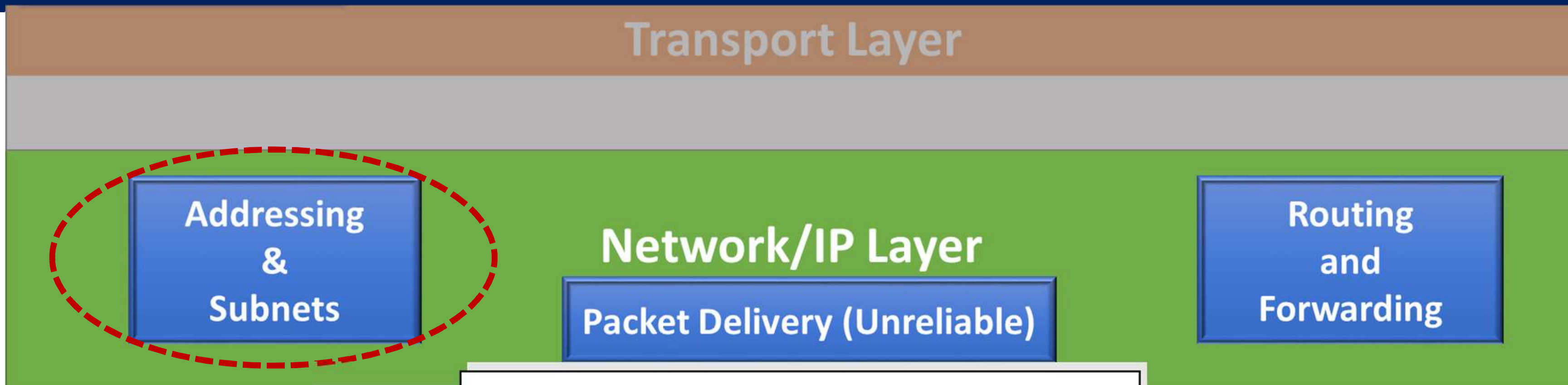
SAMPLE QUESTIONS

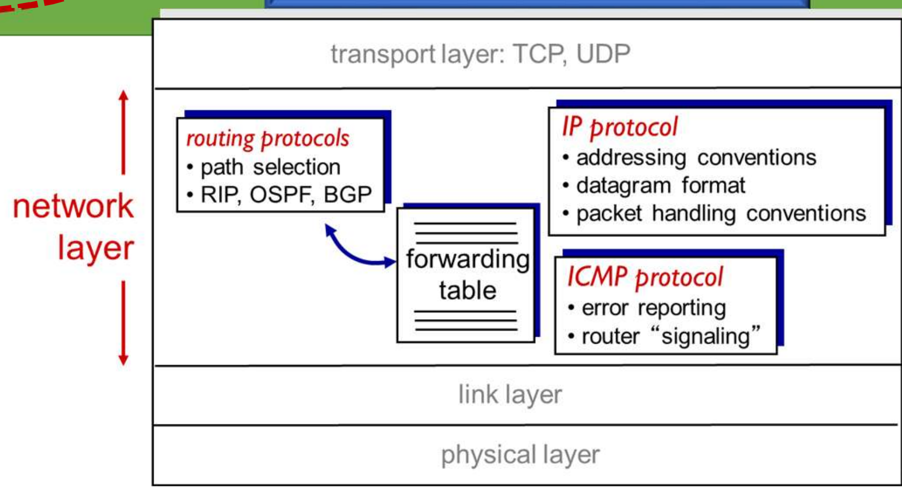
a. As a map to the above case, describe the routing information (routers forwarding tables) that needs to set at each of the routers in the below diagram: **(5 pts)** (Refer to the example table to fill the information):



Destination Address Range	Outgoing Interface

NETWORK LAYER SERVICES





Fundamentals: Forwarding vs. Routing
Networking Planes: Data and Control
Routers - Operation and types
Routing Protocols
IPv4 Addressing & Subnets
Classful Addressing vs CIDR

Reading Material: *RFC1122: by Robert Braden "Requirements for Internet Hosts — Communication Layers"*

Today's Focus:
IP Addressing – DHCP and NAT
IPv6 Protocol and Addressing

IP ADDRESSES: HOW TO GET ONE?

Q: How does a *host* get IP address?

- hard-coded by system admin in a file
  - Windows: control-panel->network->configuration->tcp/ip->properties
  - UNIX: /etc/rc.config
  - Linux: /etc/sysconfig/network-scripts/ifcfg-ethXX

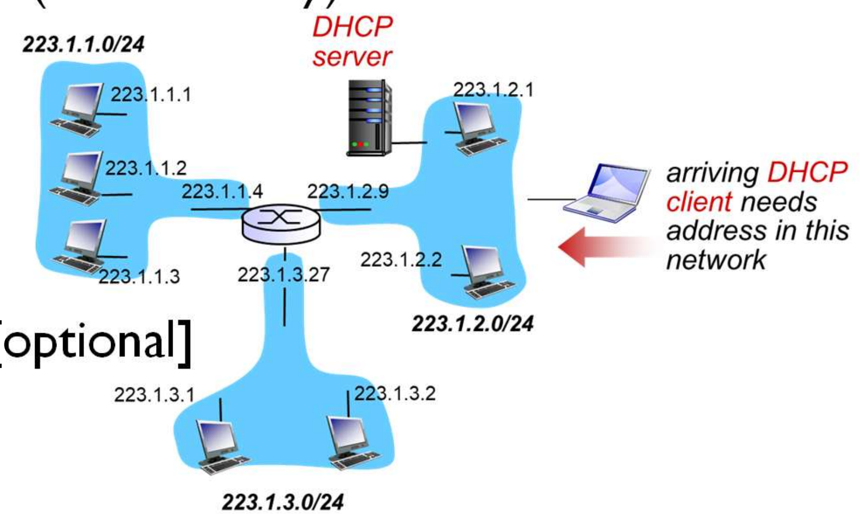
- DHCP: Dynamic Host Configuration Protocol: [RFC 2131]
  - dynamically get address from a server
  - "plug-and-play"

- DHCP can return more than just allocated IP address on subnet:
  - address of first-hop router for client
  - name and IP address of DNS sever
  - network mask (indicating network versus host portion of address)

DHCP: Dynamic Host Configuration Protocol

goal: allow host to *dynamically* obtain its IP address from network server when it joins network

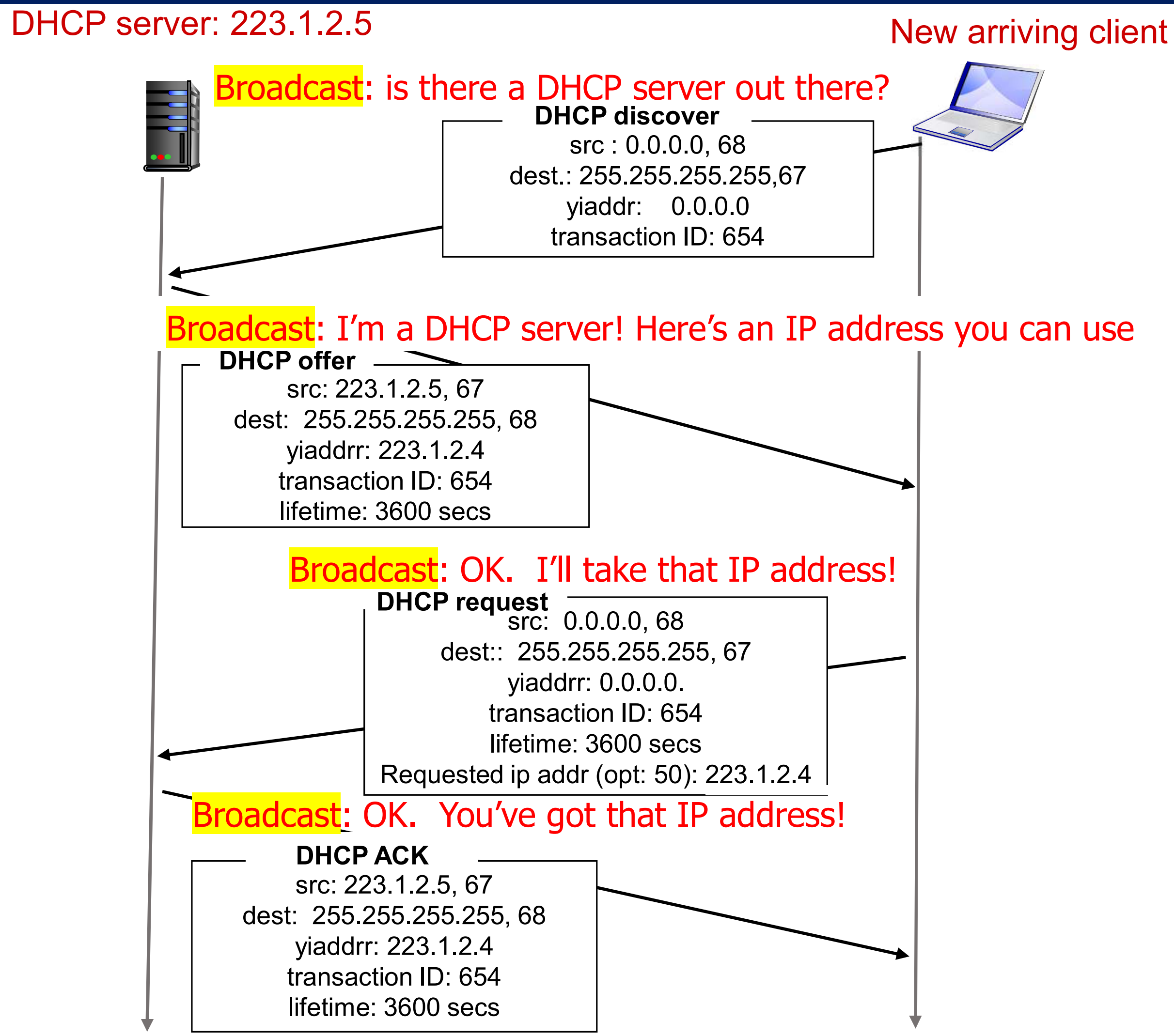
- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)



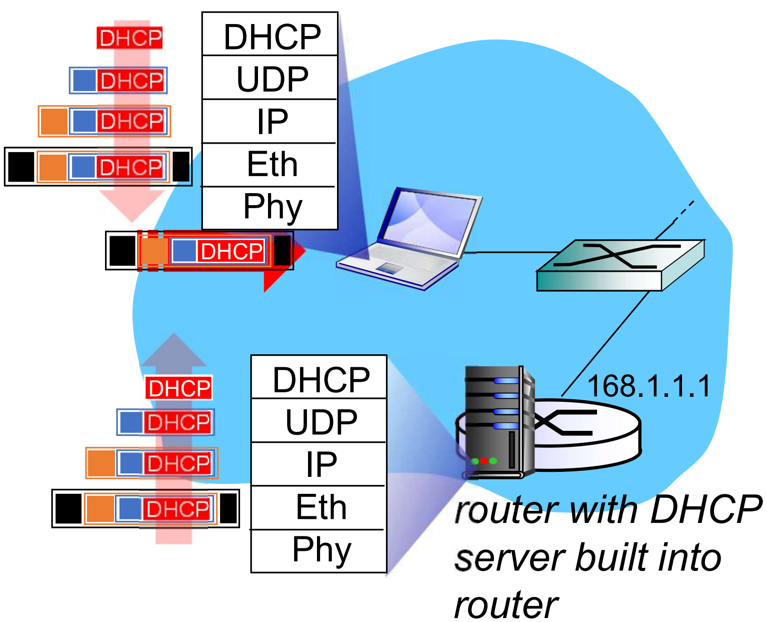
DHCP overview:

1. host broadcasts "DHCP discover" msg [optional]
2. DHCP server responds with "DHCP offer" msg [optional]
3. host requests IP address: "DHCP request" msg
4. DHCP server sends address: "DHCP ack" msg

DHCP CLIENT-SERVER SCENARIO



DHCP: EXAMPLE



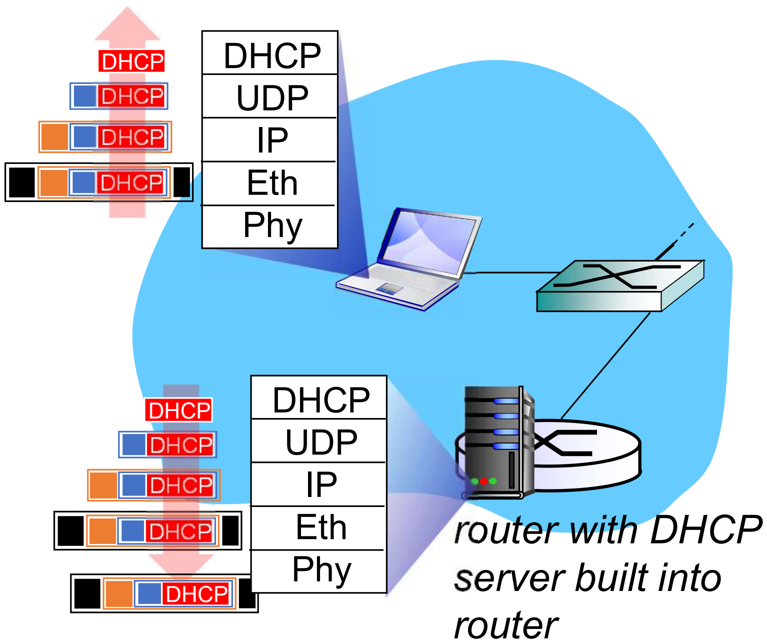
- connecting laptop needs its IP address, addr of first-hop router, addr of DNS server: use DHCP

- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.1 Ethernet

- Ethernet frame broadcast (dest: FFFFFFFFFFFF) on LAN, received at router running DHCP server

- Ethernet demuxed to IP demuxed, UDP demuxed to DHCP

DHCP: EXAMPLE

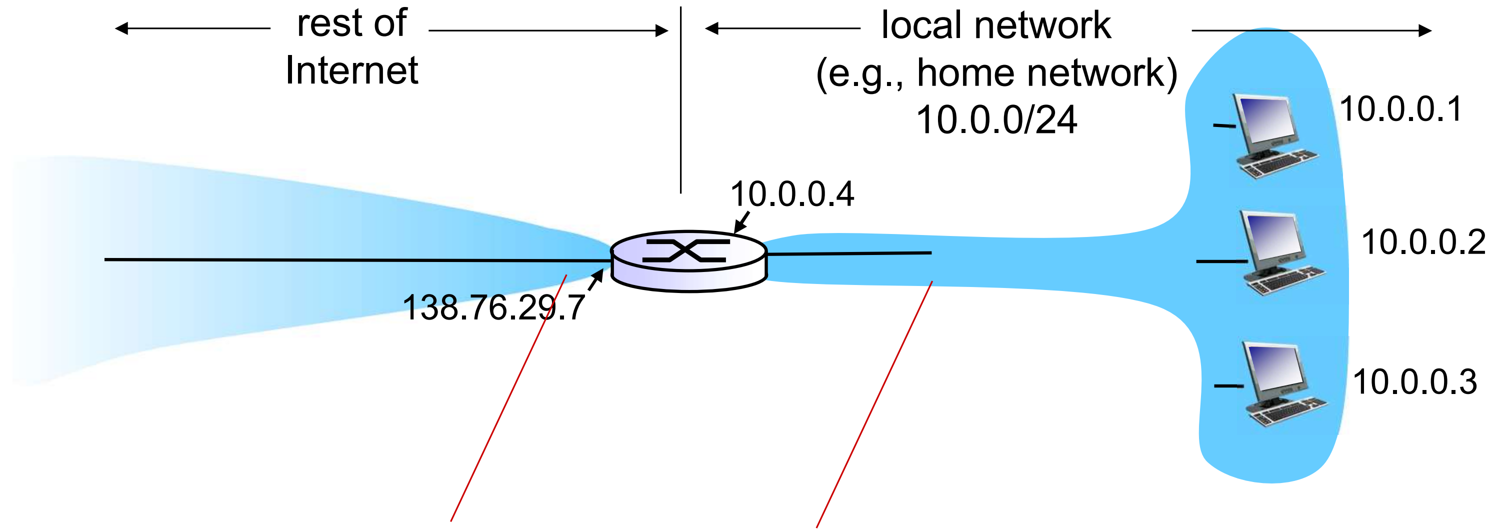


- DCP server formulates DHCP ACK (broadcast (dest: FFFFFFFFFFFF)) containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server

- 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 DNS server; IP address of its first-hop router

NAT(V4): NETWORK ADDRESS TRANSLATION



all datagrams *leaving* local network have *same* single source IP address i.e. NAT IP address: 138.76.29.7, mapped to different source port numbers.

datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT: NETWORK ADDRESS TRANSLATION

motivation:

local network uses just one IP 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)

NAT: NETWORK ADDRESS TRANSLATION

implementation: NAT router must:

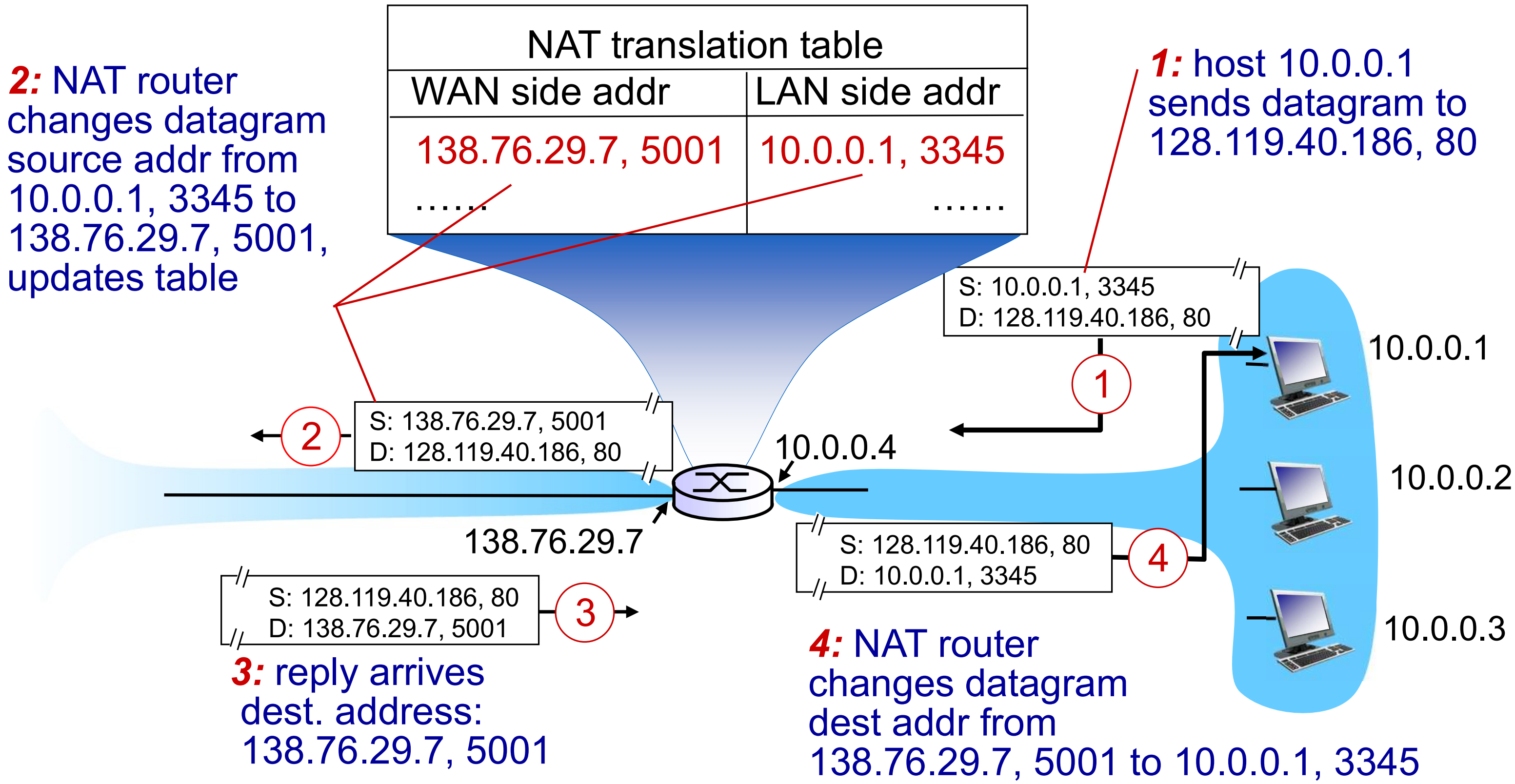
- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr

- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair

- *incoming datagrams: replace* (NAT IP address, new port #) in dest. fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: NETWORK ADDRESS TRANSLATION



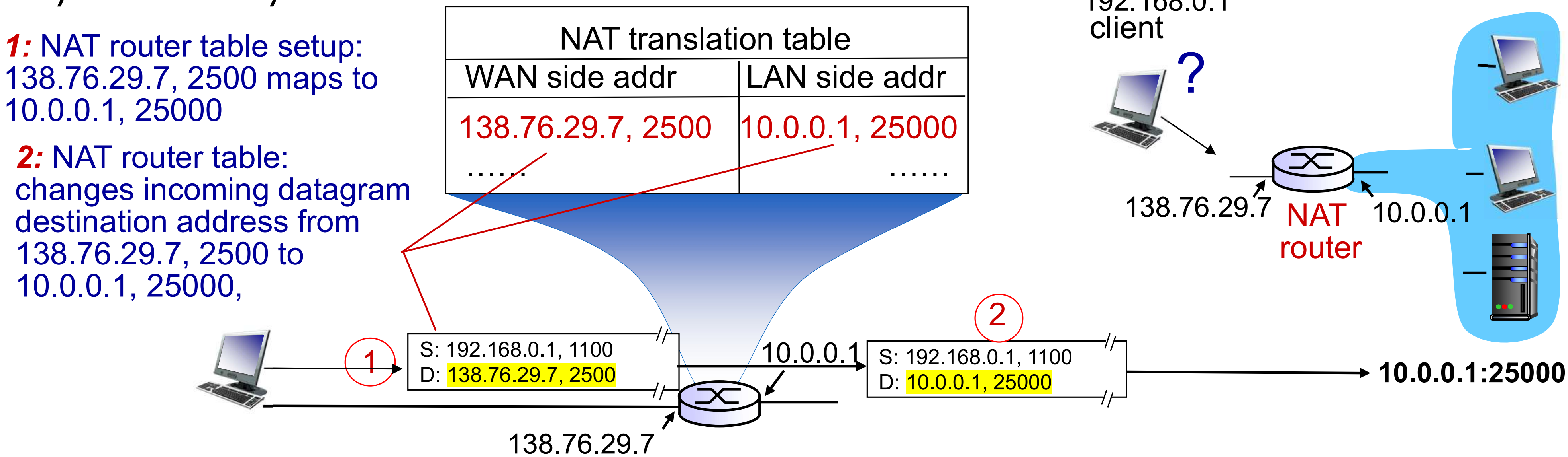
NAT: NETWORK ADDRESS TRANSLATION

- 16-bit port-number field:
  - 64,000 simultaneous connections with a single LAN-side address!

- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - address shortage should instead be solved by IPv6

NAT TRAVERSAL PROBLEM

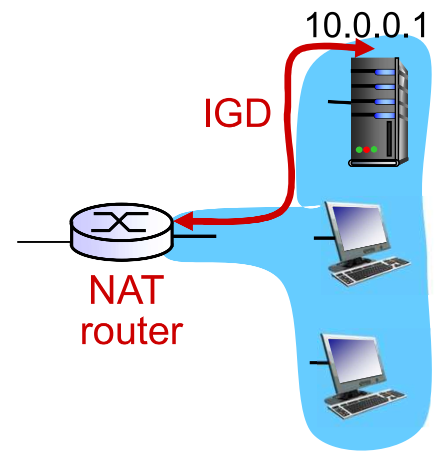
- client wants to connect to server with address 10.0.0.1:25000
  - server address 10.0.0.1 local to LAN (client can't use it as destination address)
  - only one externally visible NATed address: 138.76.29.7



- *solution 1:* Explicit Port Forwarding Rule Setup at the NAT routers
- statically configure NAT to forward incoming connection requests at a given port to a server.
  - e.g., (138.76.29.7, port 2500) always forwarded to (10.0.0.1 port 25000)

NAT TRAVERSAL PROBLEM

- *solution 2:* Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATed host to:
  - learn public IP address (138.76.29.7)
  - add/remove port mappings (with lease times)



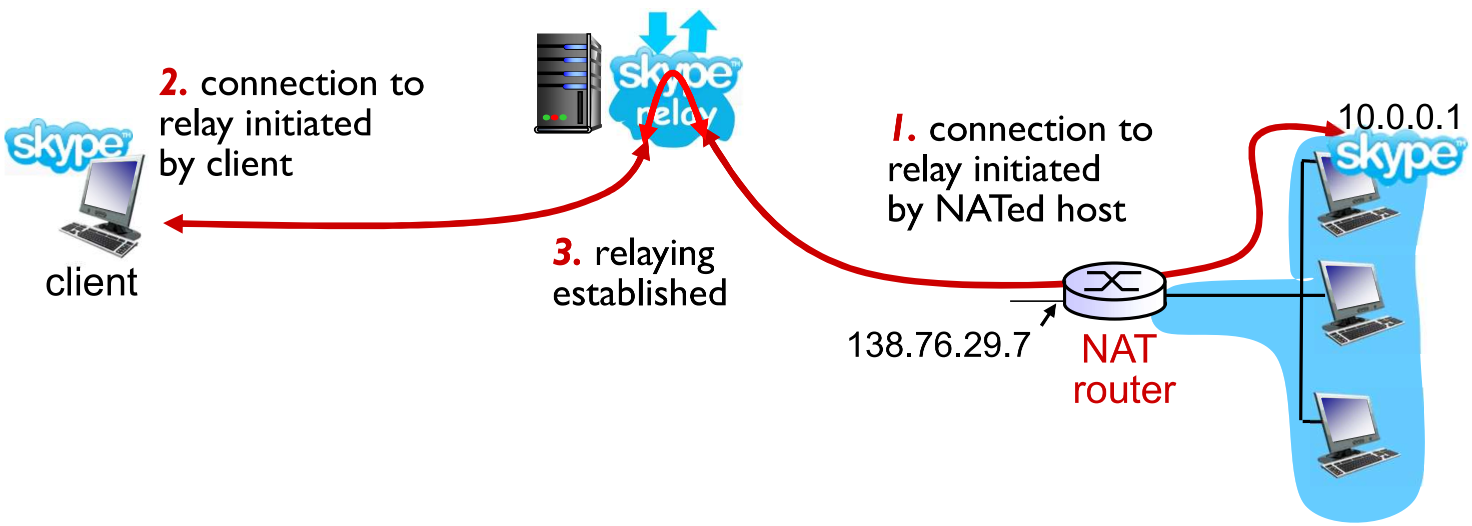
- App. in host requests mapping for specific public port #:
- (*pvt. IP addrs, pvt. port #*) ⇔ (*public IP addrs, public port #*).

- NAT accepts request and creates necessary mapping;
  i.e., *automate the static NAT port map configuration*

Session Traversal Utilities for NAT (STUN)
https://www.rfc-editor.org/rfc/rfc5389

NAT TRAVERSAL PROBLEM

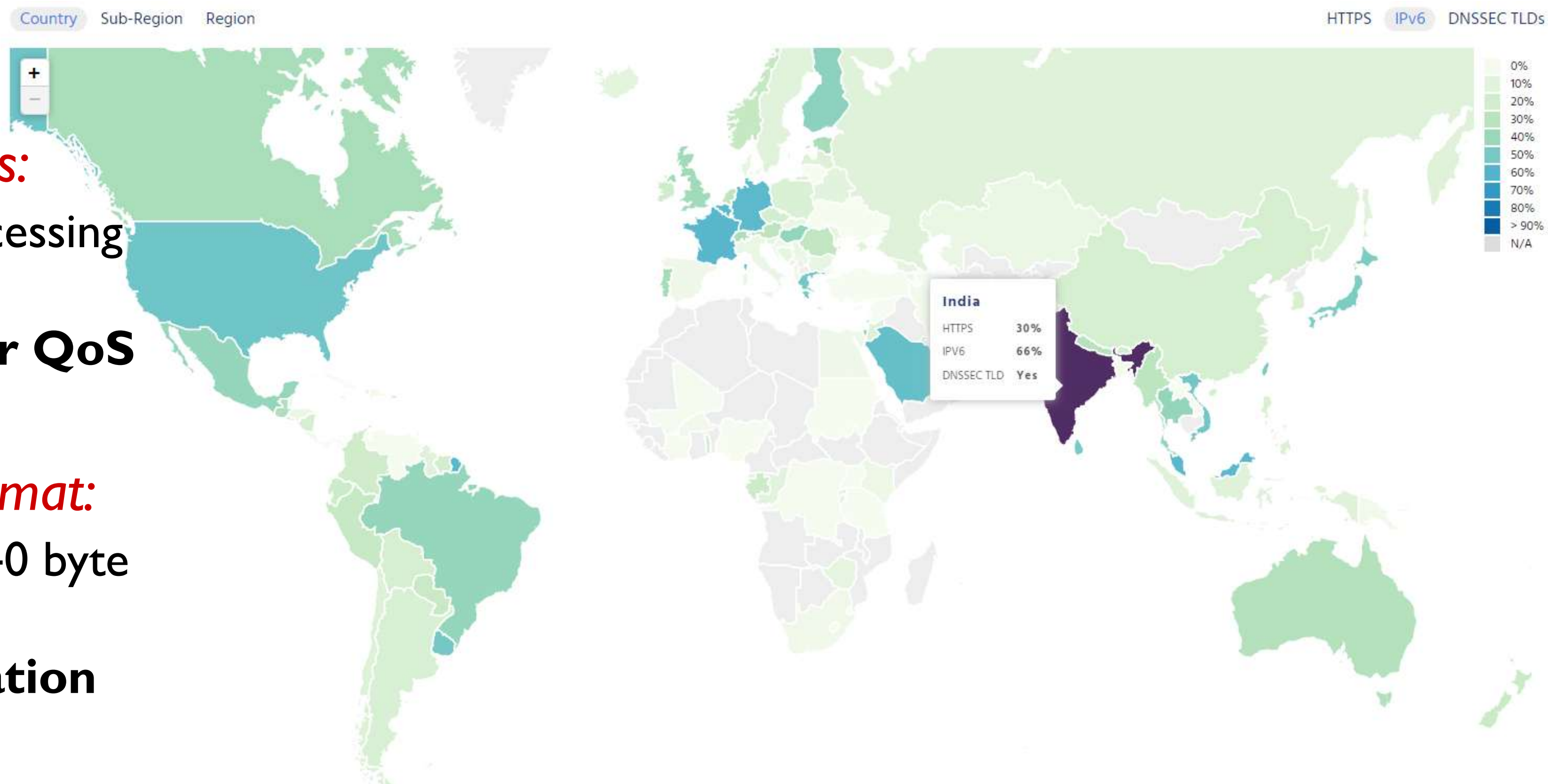
- *solution 3:* relaying (used in Skype)
  - NATed client establishes connection to relay
  - external client connects to relay
  - relay bridges packets between two connections



Traversal Using Relays around NAT (TURN)

IPv6: MOTIVATION

- *initial motivation:* 32-bit address space soon to be completely allocated.



- *other motivations:*
  - **speed up** processing /forwarding
  - facilitate **better QoS**

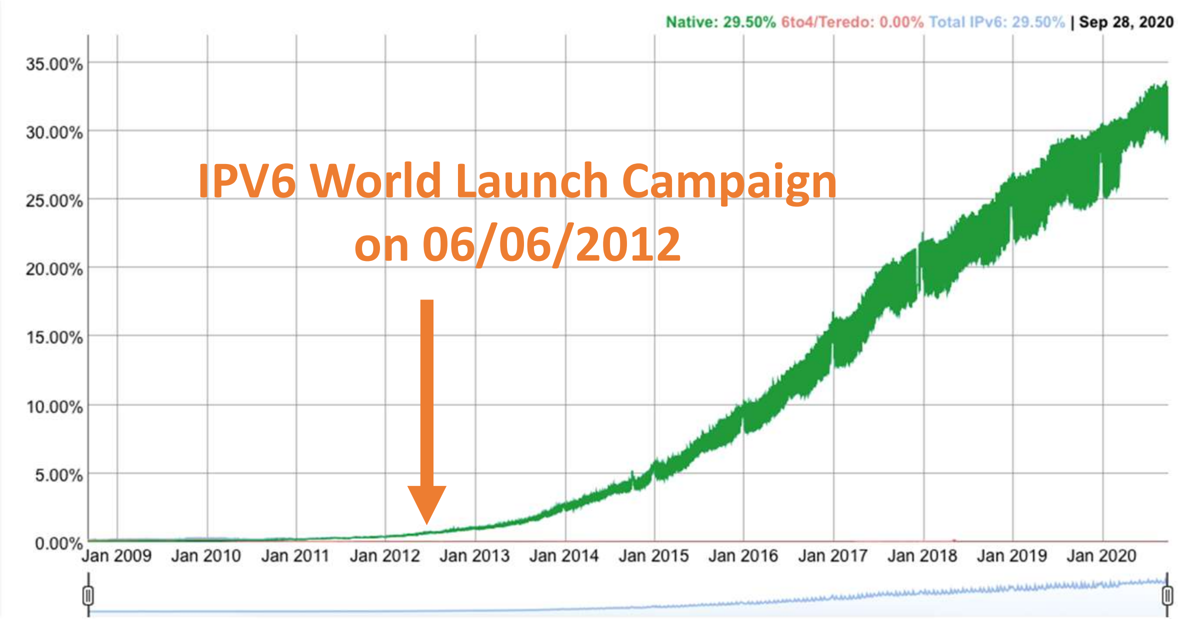
IPv6 datagram format:

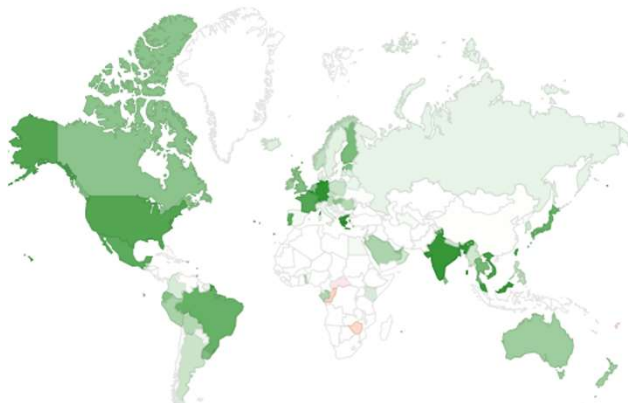
- **fixed-length** 40 byte header
- **no fragmentation** allowed.
- ***No checksum***

https://pulse.internetsociety.org/technologies

IPv6: ADOPTION

- 30~52% of of traffic is IPv6.
- *Long! time for deployment, use*
  - 24+ years and counting!
  - *Why?*
  - think of application-level changes in last 2 decades:
    - Facebook, streaming media, ~~Skype~~, Zoom, Hangout, Meet, Online Classrooms





India has reached an IPv6 adoption rate of around 60%; that is nearly double of US (September 2020 report).

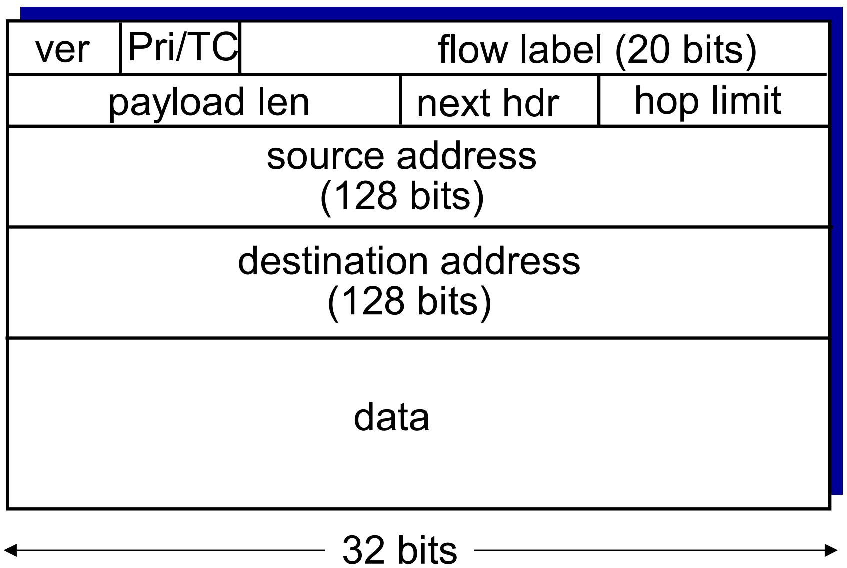
APNIC places India at more than 70% preferring IPv6.[154]

IPV6 DATAGRAM FORMAT

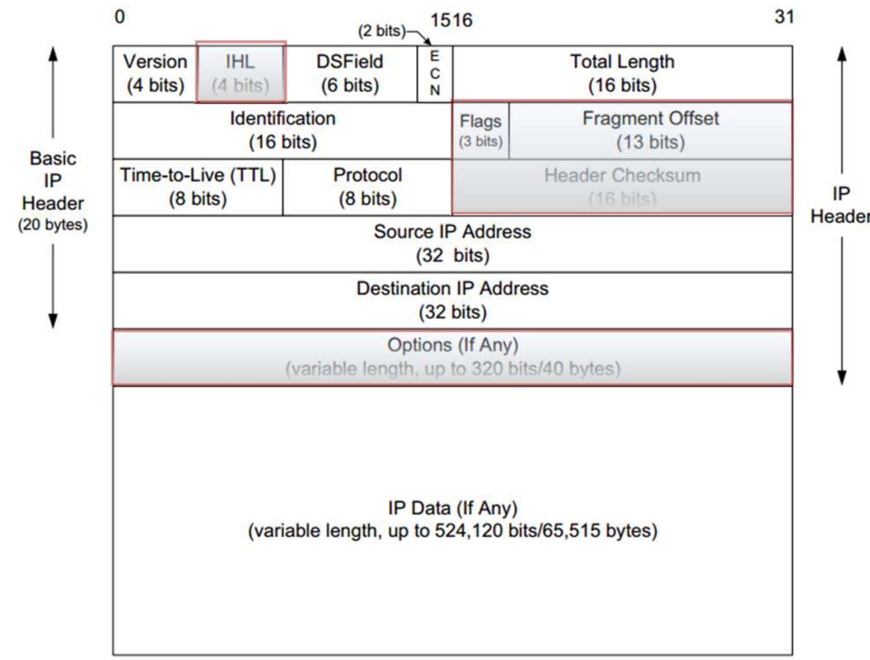
priority: identify priority among datagrams in flow

flow Label: identify datagrams in same "flow." (concept of "flow" not well defined).

next header: identify the following (upper layer) protocol for data



VS



checksum: removed entirely to reduce processing time at each hop

options: allowed, but outside of header, indicated by "Next Header" field

IPV6 ADDRESSING (RFC 4921)

- *Modes of Addressing*
- Unicast: Link-local vs Global
- Multicast (FF00::/8)
- ***Anycast***.

Address type	Binary prefix	IPv6 notation
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-Local unicast	1111111010	FE80::/10
Global Unicast	(everything else)	

```
|          n bits   n = 48   |    m bits    |          128-n-m bits          |
+--------------------------+ m = 16 -----+--------------------------------+
| global routing prefix    |  subnet ID   |          interface ID          |
+--------------------------+--------------+--------------------------------+
```

- *Interoperability/Compatibility with IPv4*
- IPv4-Compatible IPv6 Address (First 96 bits =0, Last 32 bits = IPv4) – outdated.
- IPv4-Mapped IPv6 Address. (Last 32 bits=IPv4, next higher 16 bits=FFFF)
- Well Known Prefix: 64.ff9b::/96 or NSP specific Prex64::/n [RFC 6052]
- *Programming Interface:* new version of socket interface
  - RFC 3493 : Basic Socket Interface Extension for Ipv6

CHALLENGES (CHANGES) IN ADOPTING IPV6

- *Adoption*
  - How to ensure every router in the world supports IPv6 from same date-time.
  - Backward-compatibility is key.

- *Interoperability*
  - Some devices/networks may still be IPv4 based.

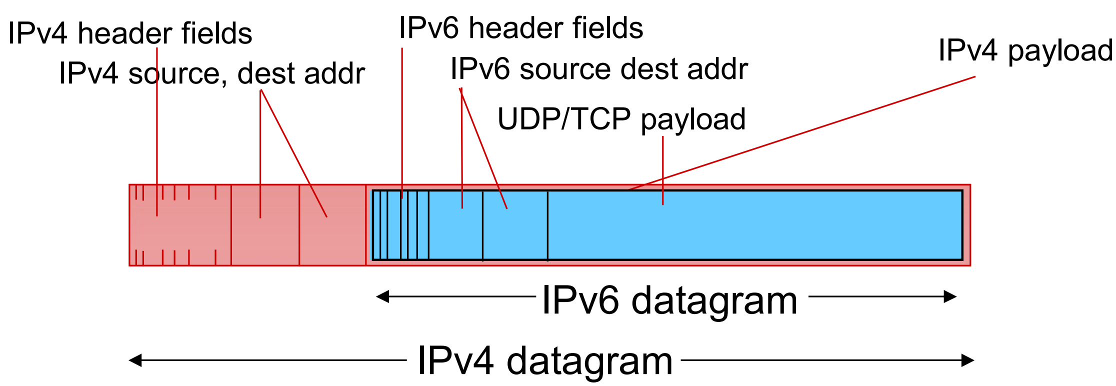
- *Routing – Addressing*
  - Routing algorithms need to support both v4 and v6
  - Routes may encompass multiple v4 and v6 hops.

- *Supporting Infrastructure:* new version of supporting protocols: ICMP/DHCP/NDP/DNS
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions
  - New DNS records for storing IPv6 addresses

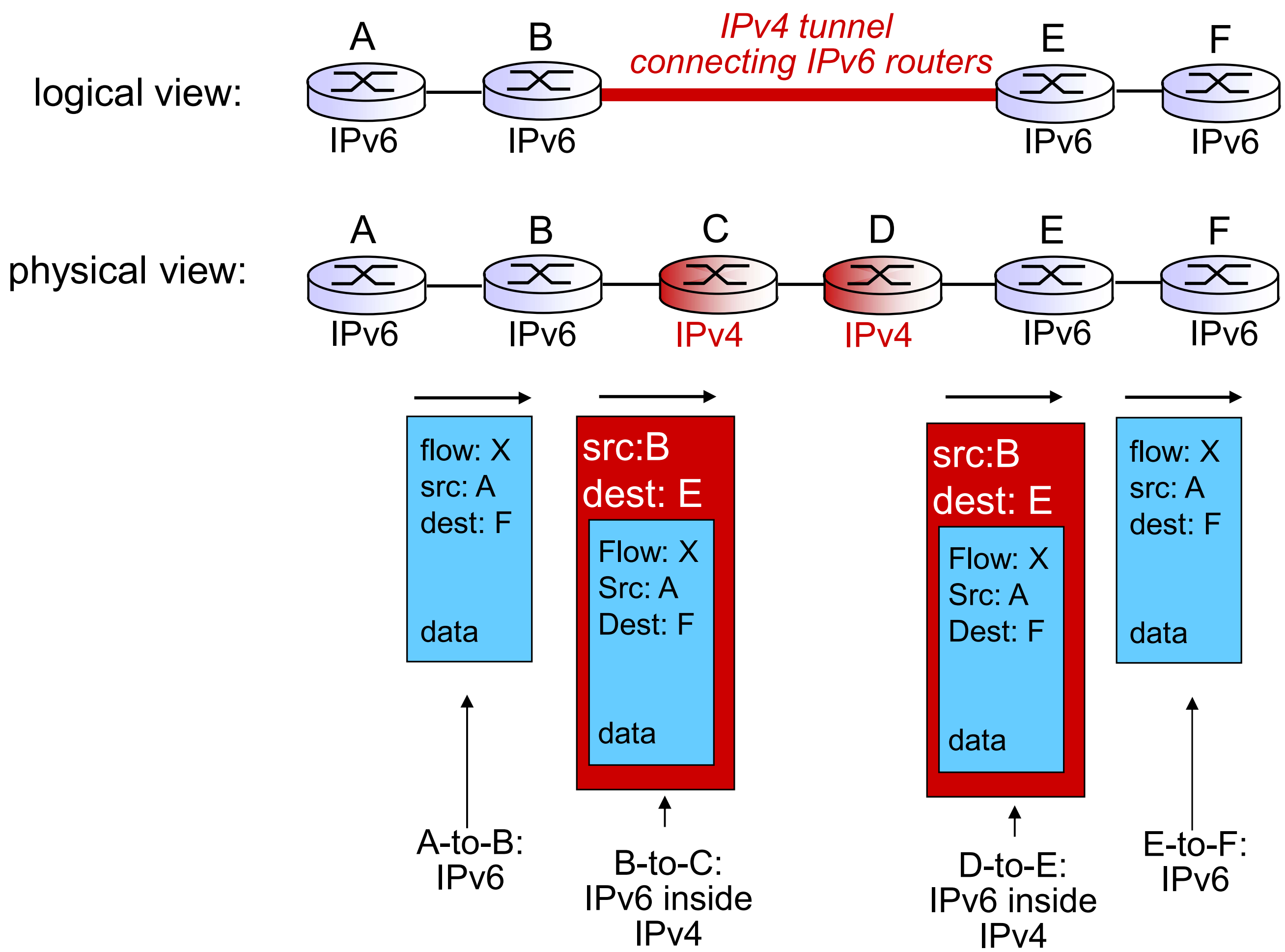
TRANSITION FROM IPV4 TO IPV6

- 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



IPv4 tunnel
connecting IPv6 routers
logical view:
A
B
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
D-to-E:
IPv6 inside
IPv4
E-to-F:
IPv6

ROUTERS – THE CORE NETWORKING DEVICES

Cisco 8100 Routers

A fixed platform with up to 12.8 Tbps of capacity and optimized to reduce rack space and power costs.

- Chassis Form: Fixed
- Bandwidth: 6.4 to 12.8 Tbps
- Available Ports:
  32 QSFP28 100GbE;
  32 QSFP56-DD 400GbE;
  64 QSFP28 100GbE
- Height: 1 RU, 2 RU

Source – Cisco Routers

Cisco 8200 Routers

A fixed platform with 10.8 Tbps of capacity for deployment in space and power constrained facilities.

- Chassis Form: Fixed
- Bandwidth: 10.8 Tbps
- Available Ports:
  12/24 QSFP56-DD 400GbE
- Height: 1 RU, 2 RU

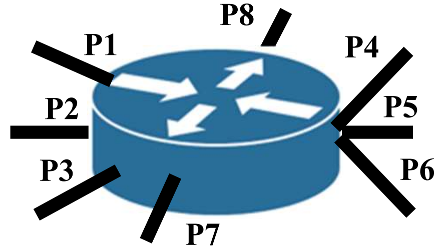
Cisco 8800 Routers

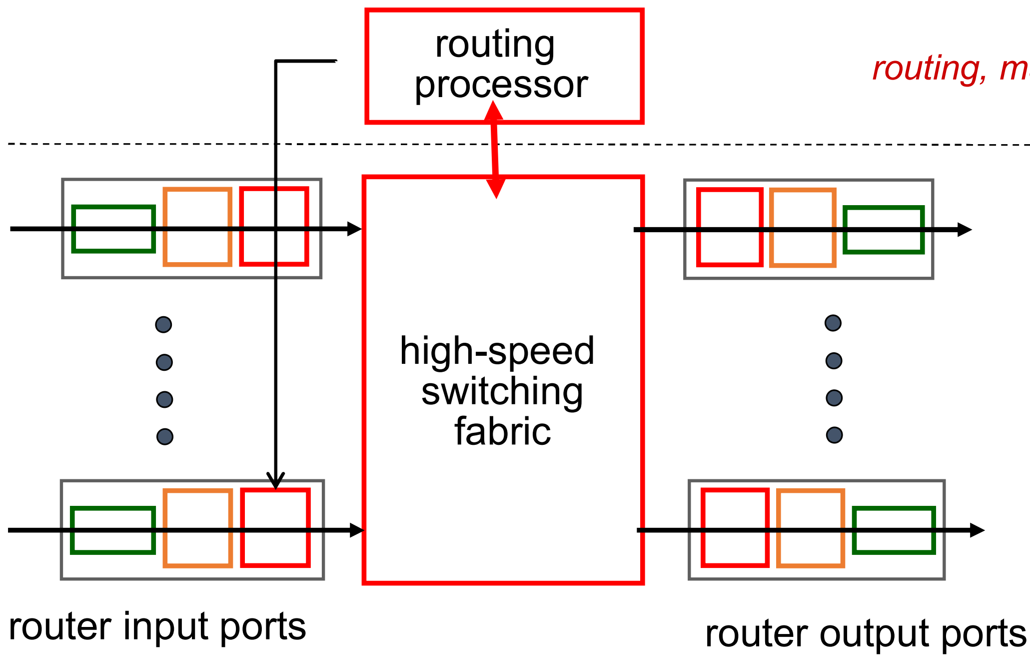
A high performance, high density modular platform with up to 259.2 Tbps of capacity that can consolidate the number of routers needed and reduce overall complexity.

- Chassis Form: Modular
- Bandwidth: Up to 259.2 Tbps
- Available Ports:
  36 QSFP56-DD 400GbE;
  48 QSFP28 100GbE with MACsec
- Height: 16 RU (8 slots), 21 RU (12 slots), 33 RU (18 slots)

Router architecture overview

- high-level view of generic router architecture:

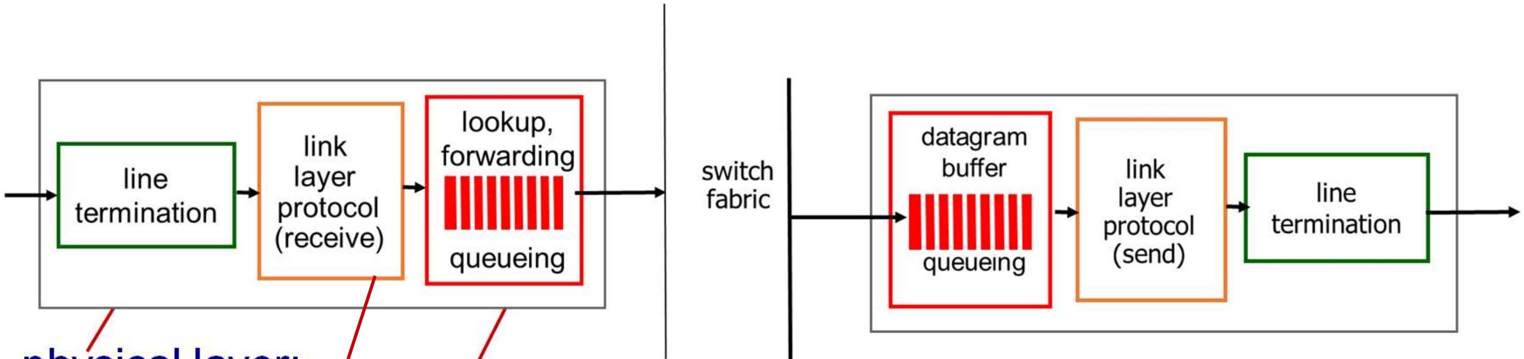




routing, management: control plane (**software**) operates in ***second*** time frame

forwarding data plane (**hardware**) operates in ***nanosecond*** timeframe

INPUT PORT VS OUTPUT PORT FUNCTIONS



physical layer:
bit-level reception

data link layer:
Data Framing

Network layer:
Packet switching

- *buffering* required when datagrams arrive from fabric at a rate slightly faster than the transmission rate

Datagram can be lost due to lack of buffers!

- *scheduling discipline* chooses among queued datagrams for transmission

- decentralized switching*:*

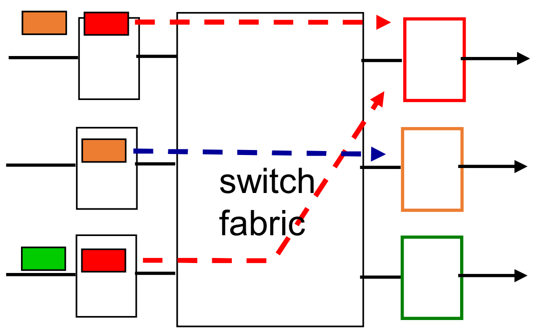
Use datagram header fields to lookup output port in the forwarding table *("**match plus action**")*

decentralized switching modes*:*

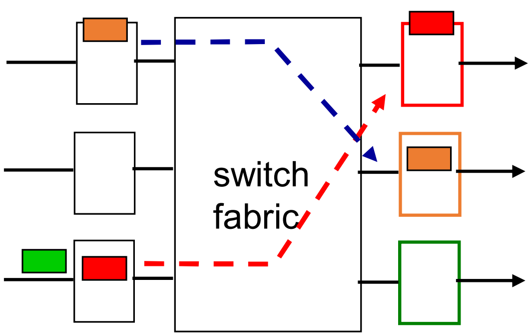
- *destination-based forwarding:* forward based only on destination IP address (traditional)
- *generalized forwarding:* forward based on any set of header field values.

INPUT PORT QUEUING

- Queuing: when the datagrams arrive faster than forwarding rate of switch fabric.
- fabric slower than input ports combined -> queueing may occur at input queues
  - *queueing 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

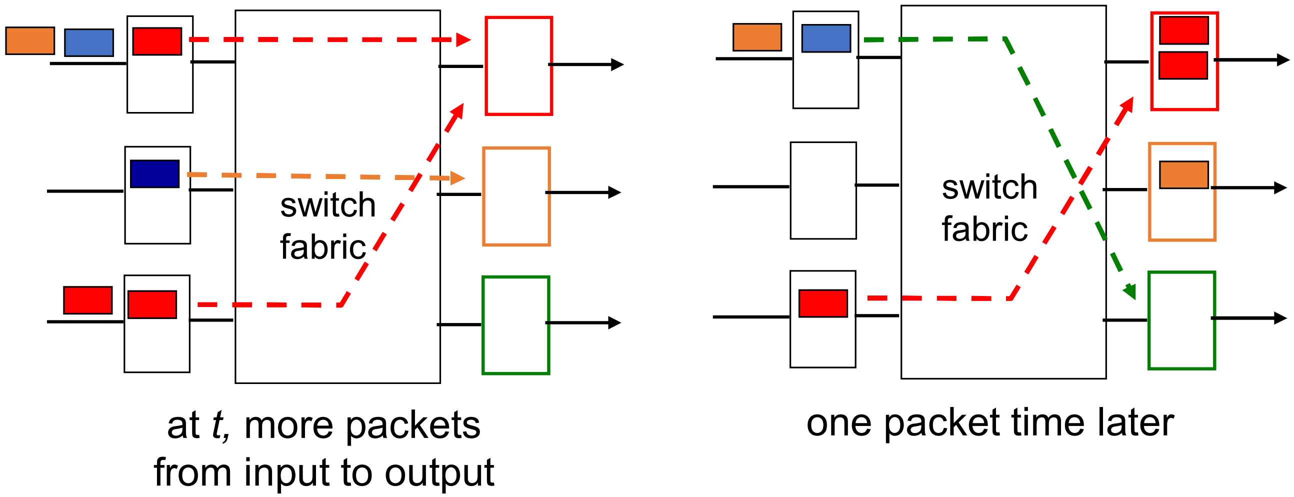


output port contention:
only one red datagram can be transferred.
***lower red packet** is blocked*



one packet time later:
green packet experiences
HOL blocking

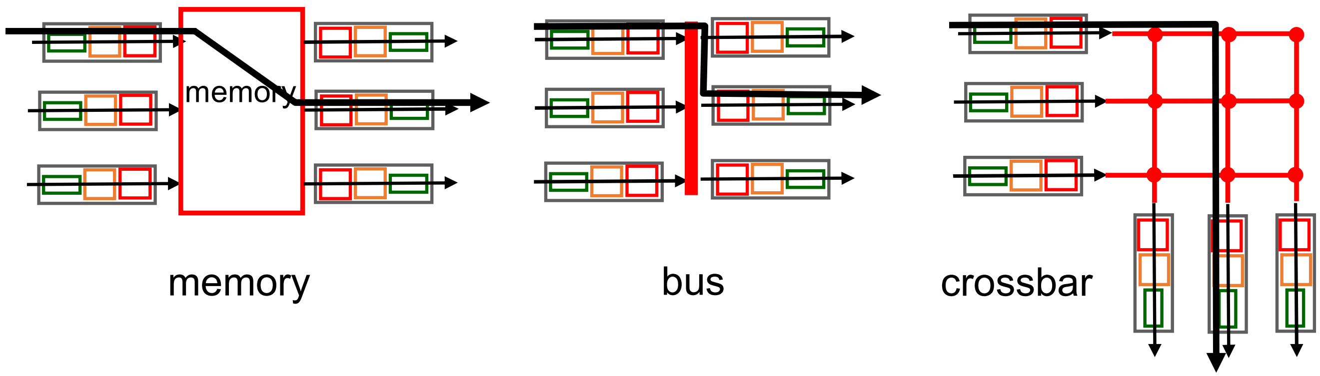
OUTPUT PORT QUEUEING



- buffering when arrival rate via switch fabric exceeds output line speed
- *queueing (delay) and loss can also occur due to output port buffer overflow!*
- RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 250 msec) times link capacity C.
  - e.g., C = 10 Gpbs link: 2.5 Gbit buffer
- recommendation (2004): with N flows, buffering equal to = $\frac{RTT \cdot C}{\sqrt{N}}$

SWITCHING FABRICS

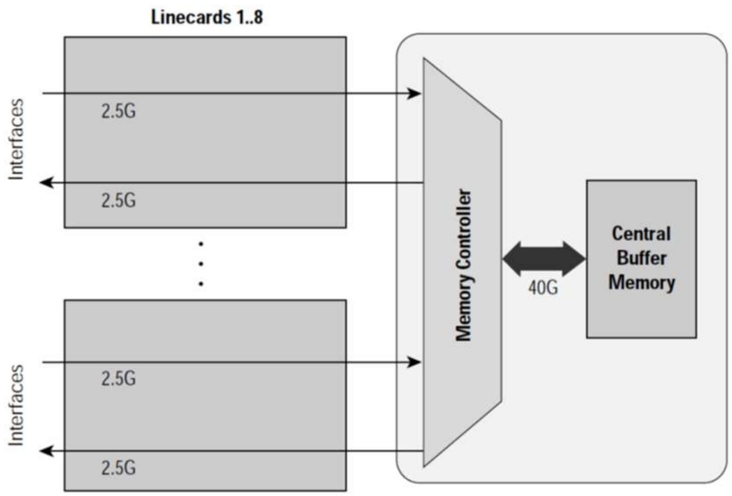
- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: desirable switching rate N times line rate
- three types of switching fabrics

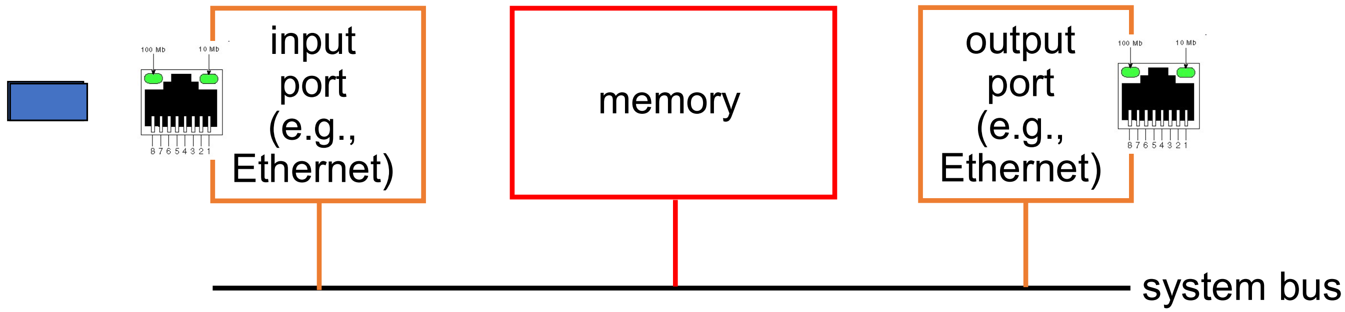


SWITCHING VIA MEMORY

first generation routers: Centralized shared Memory

- Similar to traditional computers:
  with switching under direct control of CPU
- Incoming packet copied to system's memory
- Processing speed is limited by memory bandwidth
  (need 2 bus crossings per datagram)

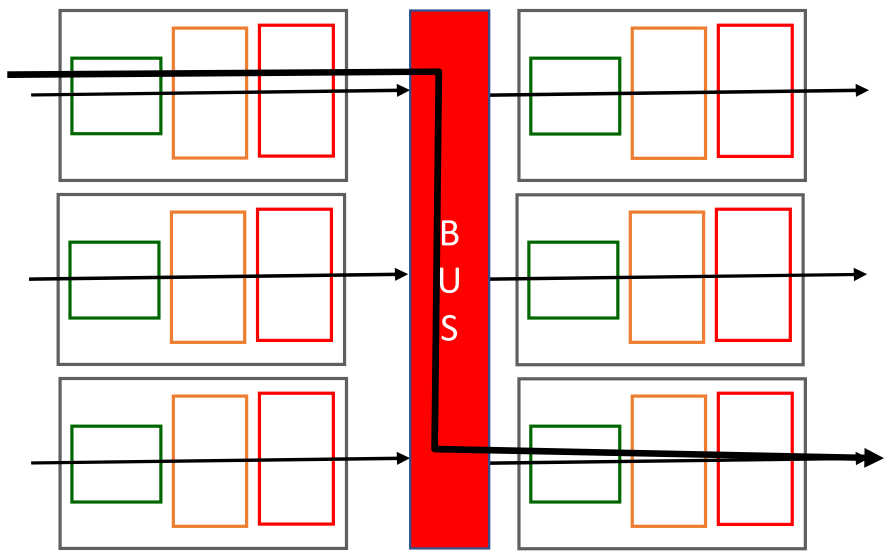




SWITCHING VIA A BUS

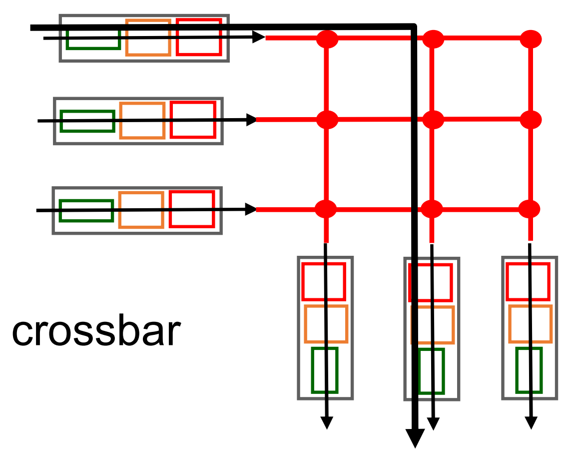
- datagram from input port memory to output port memory via a shared bus

- *bus contention:* switching speed limited by bus bandwidth
  - 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers

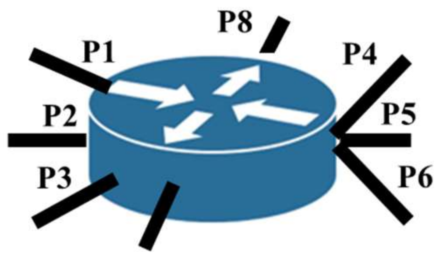


Switching via interconnection network

- overcome bus bandwidth limitations
- 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



Typical Router Architecture

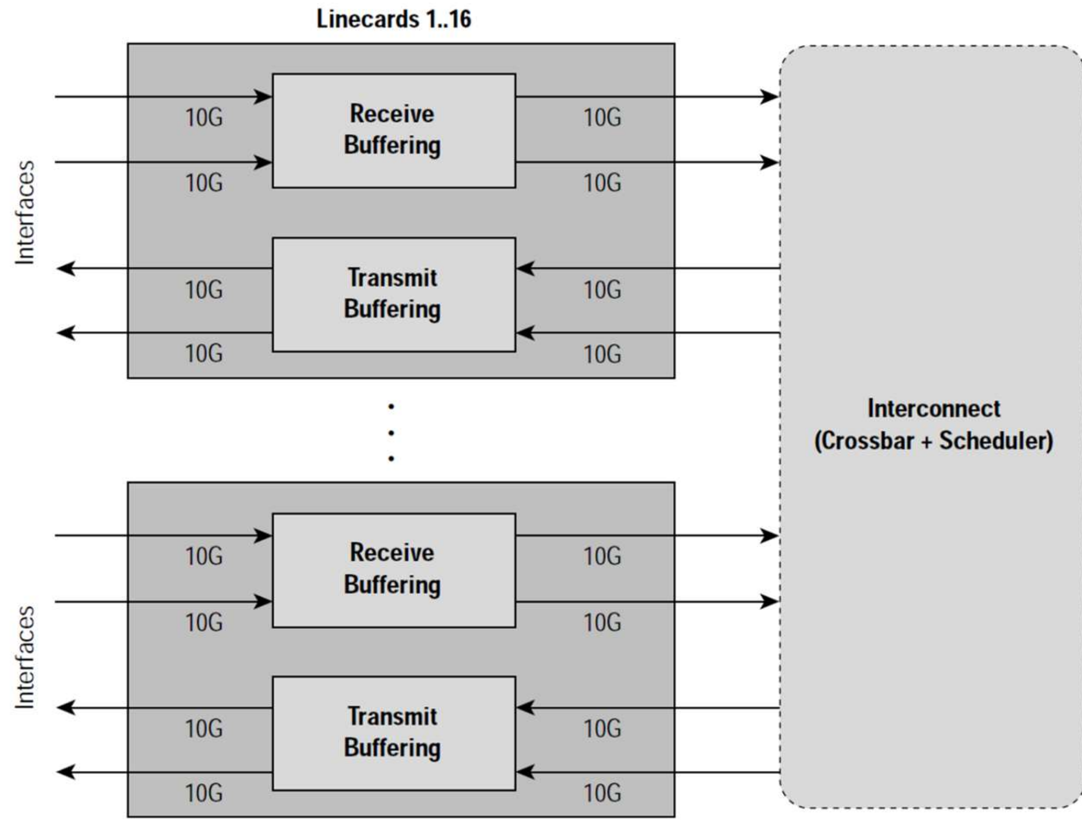


Router Jobs:

- Compute Best Path/Route → Role of a Router Processor (Control plane)

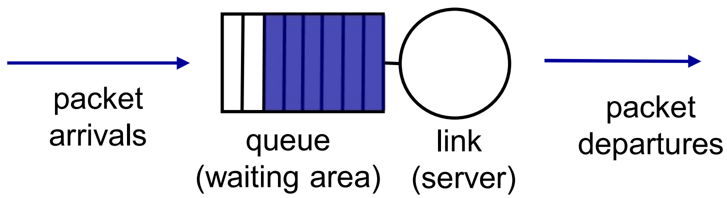
  Account for network policies & constraints

- Packet Forwarding → Role of a Forwarding Engine (Data Plane)
- Buffering and Scheduling → Role of a Scheduler (Data Plane)



SCHEDULING MECHANISMS

- *scheduling:* choose next packet to send on link
  - *discard policy: if packet arrives to full queue: who to discard?*
    - *tail drop:* drop arriving packet
    - *random:* drop/remove randomly
    - *priority:* drop/remove on priority basis

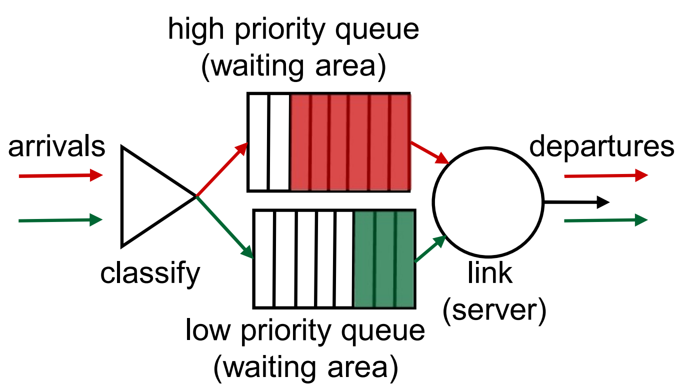


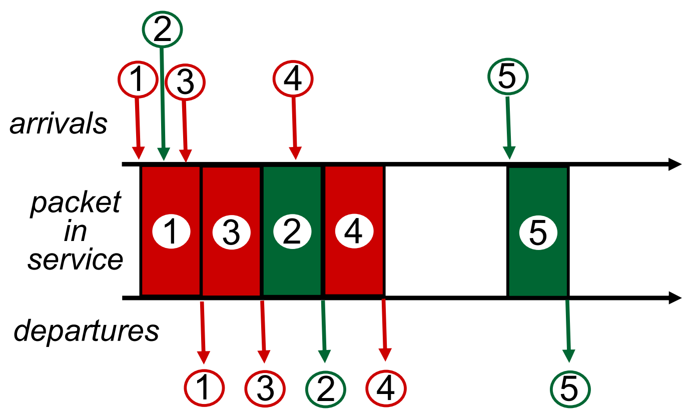
- *FIFO (first in first out) scheduling aka FCFS:* send in order of arrival to queue;

SCHEDULING POLICIES: PRIORITY

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.

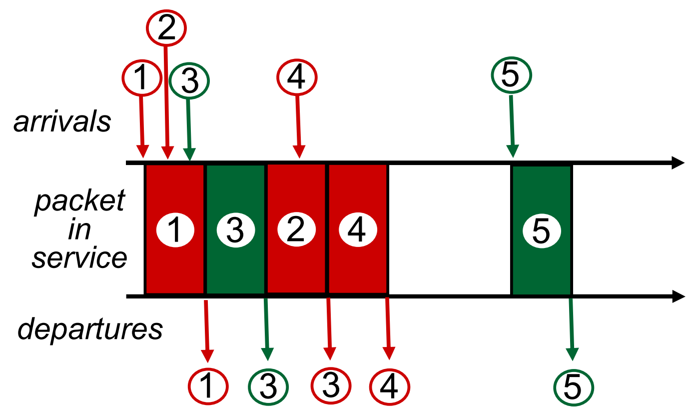




SCHEDULING POLICIES: STILL MORE

Round Robin (RR) scheduling:

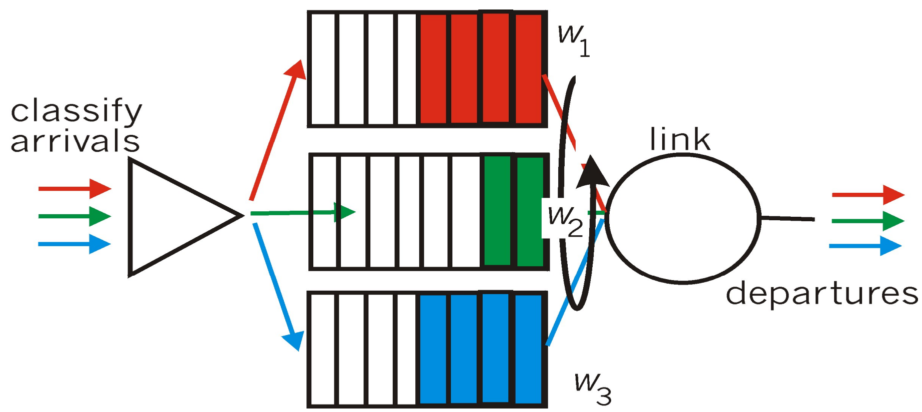
- multiple classes
- cyclically scan class queues, sending one complete packet from each class (if available)



SCHEDULING POLICIES: STILL MORE

Weighted Fair Queuing (WFQ):

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



DESTINATION-BASED FORWARDING

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through (200.23.16.0 - 200.23.23.255) 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through (200.23.24.0 - 200.23.24.255) 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through (200.23.25.0 - 200.23.31.255) 11001000 00010111 00011111 11111111	2
otherwise	3

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

LONGEST PREFIX MATCHING

lookup, forwarding

queueing

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?

- longest prefix matching: often performed using TCAMs
  - *content addressable:* retrieve address in one clock cycle, regardless of table size.
  - Cisco Catalyst: can up ~1M routing table entries in TCAM