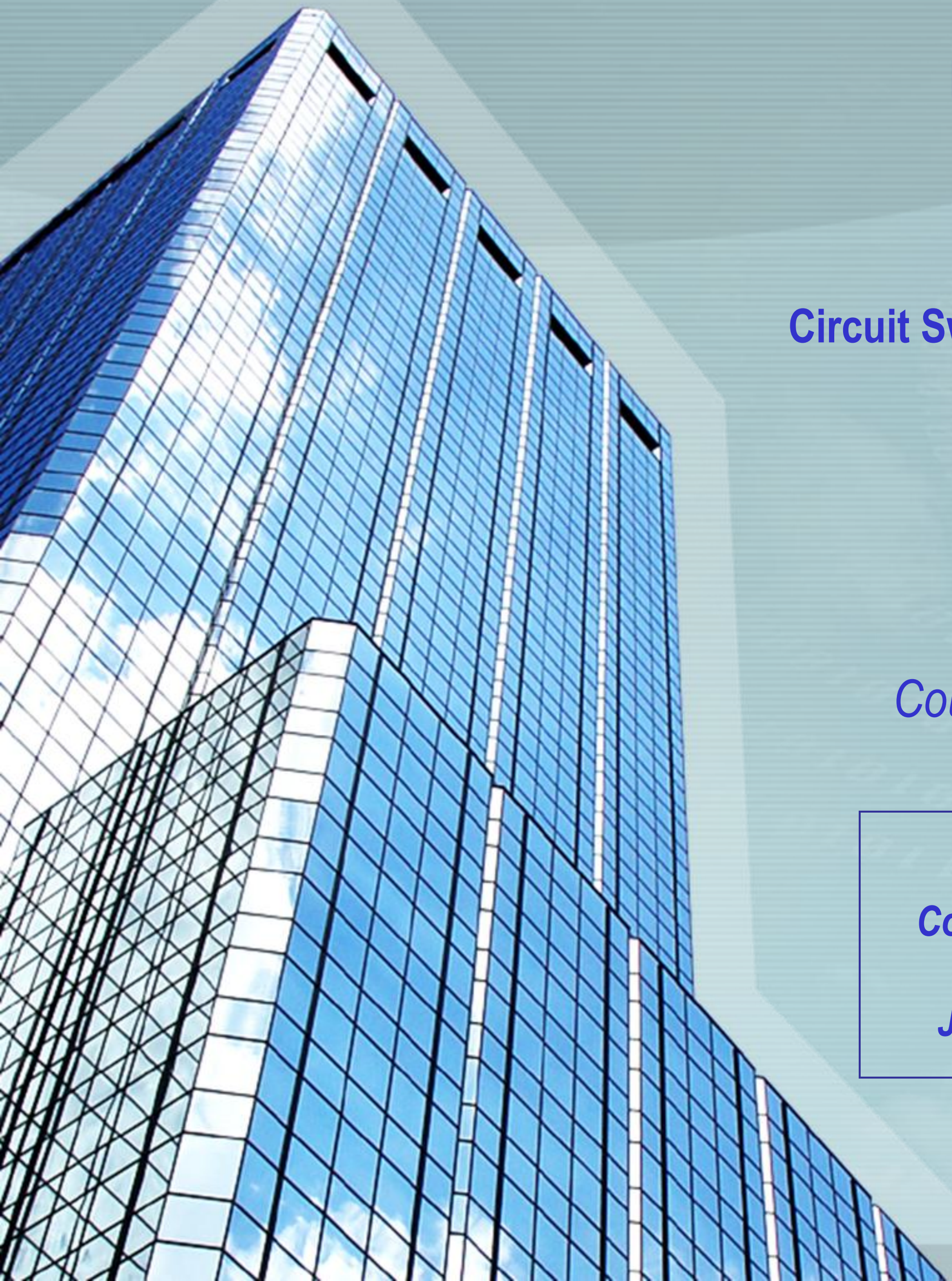


Data Communications & Networks

Session 7 – Main Theme
Networks: Part I
Circuit Switching, Packet Switching, The Network Layer

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

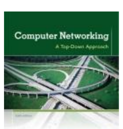
Agenda

1. Session Overview
2. Networks Part 1
3. Summary and Conclusion

What is the class about?

- Course description and syllabus:
  - » http://www.nyu.edu/classes/jcf/csci-ga.2262-001/
  - » http://cs.nyu.edu/courses/spring16/CSCI-GA.2262-001/index.html

- Textbooks:
  - » ***Computer Networking: A Top-Down Approach (6th Edition)***



James F. Kurose, Keith W. Ross
Addison Wesley
ISBN-10: 0132856204, ISBN-13: 978-0132856201, 6th Edition (02/24/12)

Course Overview

- Computer Networks and the Internet
- Application Layer
- Fundamental Data Structures: queues, ring buffers, finite state machines
- Data Encoding and Transmission
- Local Area Networks and Data Link Control
- Wireless Communications
- Packet Switching
- OSI and Internet Protocol Architecture
- Congestion Control and Flow Control Methods
- Internet Protocols (IP, ARP, UDP, TCP)
- Network (packet) Routing Algorithms (OSPF, Distance Vector)
- IP Multicast
- Sockets

Course Approach

- Introduction to Basic Networking Concepts (Network Stack)
- Origins of Naming, Addressing, and Routing (TCP, IP, DNS)
- Physical Communication Layer
- MAC Layer (Ethernet, Bridging)
- Routing Protocols (Link State, Distance Vector)
- Internet Routing (BGP, OSPF, Programmable Routers)
- TCP Basics (Reliable/Unreliable)
- Congestion Control
- QoS, Fair Queuing, and Queuing Theory
- Network Services – Multicast and Unicast
- Extensions to Internet Architecture (NATs, IPv6, Proxies)
- Network Hardware and Software (How to Build Networks, Routers)
- Overlay Networks and Services (How to Implement Network Services)
- Network Firewalls, Network Security, and Enterprise Networks

Networks Part 1 Session in Brief

- Understand principles behind network layer services:
  - Network layer service models
  - Forwarding versus routing
  - How a router works
- Instantiation, implementation in the Internet
- Conclusion

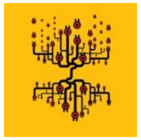
Icons / Metaphors

Information

Common Realization

Knowledge/Competency Pattern

Governance



Alignment

Solution Approach

Agenda

1 Session Overview

2 Networks Part 1

3 Summary and Conclusion

Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

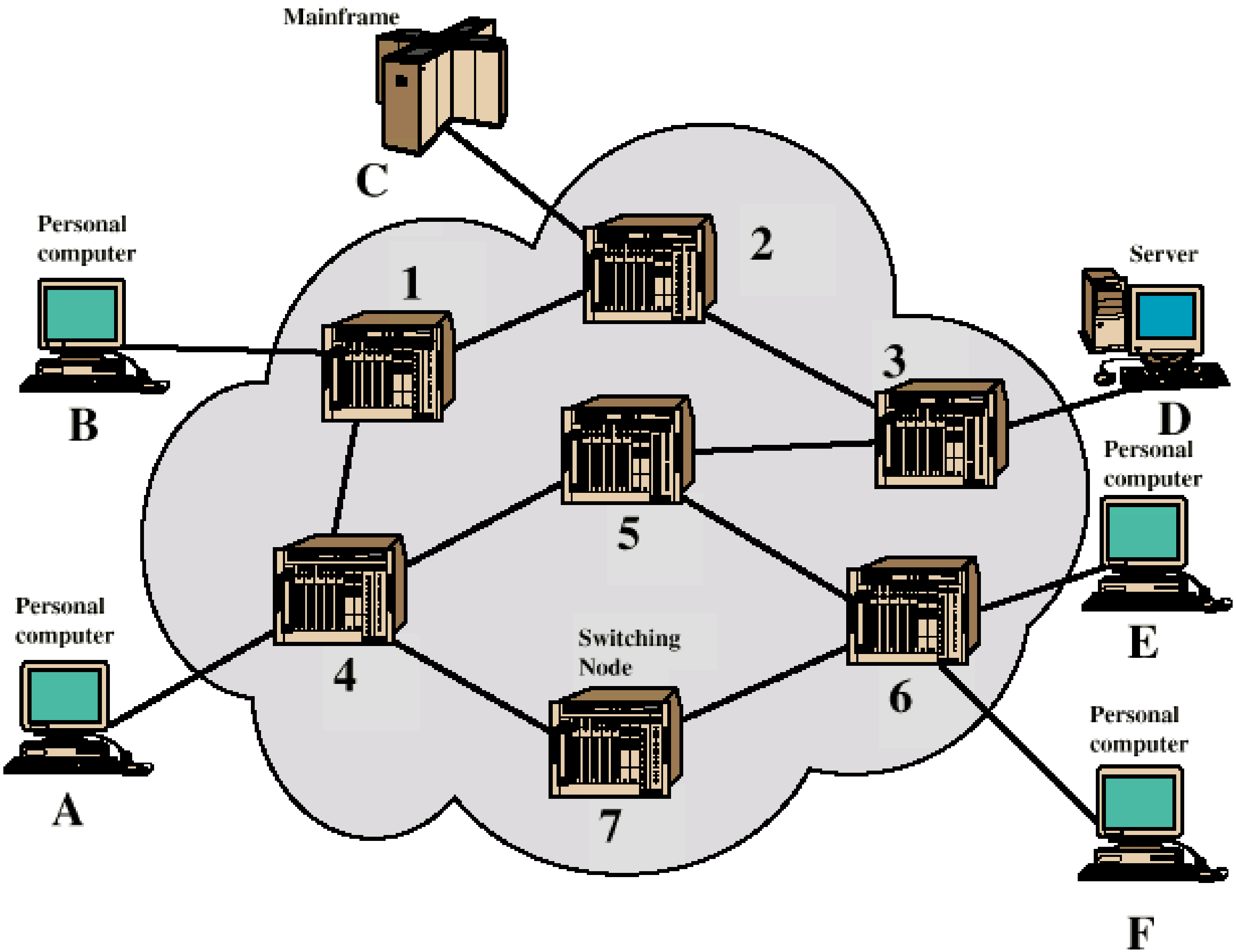
Switching Networks

- Long distance transmission is typically done over a network of switched nodes
- Nodes not concerned with content of data
- End devices are stations
  - Computer, terminal, phone, etc.
- A collection of nodes and connections is a communications network
- Data routed by being switched from node to node

Technology

- Two different switching technologies
  - Circuit switching
  - Packet switching

Simple Switched Network



Circuit Switching

- Dedicated communication path between two stations (during conversation)
- Three phases
  - Establish
  - Transfer
  - Disconnect
- Must have switching capacity and channel capacity to establish connection
- Must have intelligence to work out routing

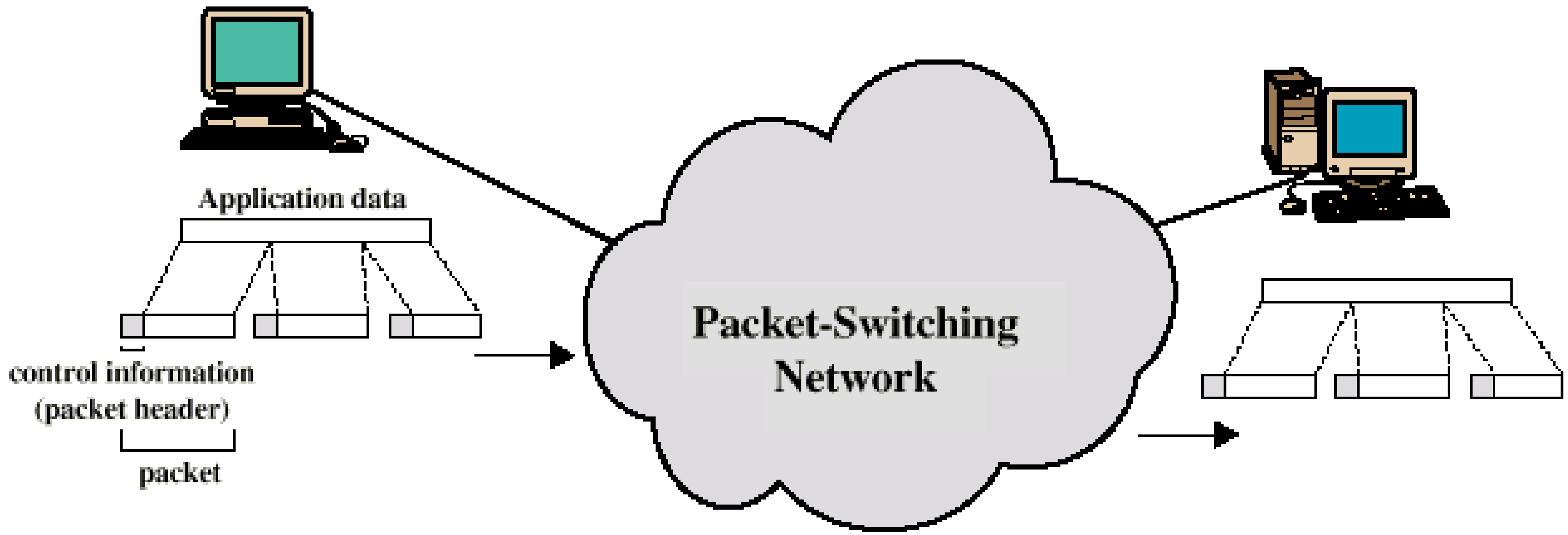
Circuit Switching - Issues

- Circuit switching is inefficient (designed for voice)
  - Resources dedicated to a particular call
  - Much of the time a data connection is idle
  - Data rate is fixed
    - Both ends must operate at the same rate
- Set up (connection) takes time
- Once connected, transfer is transparent

Packet Switching – Basic Operation

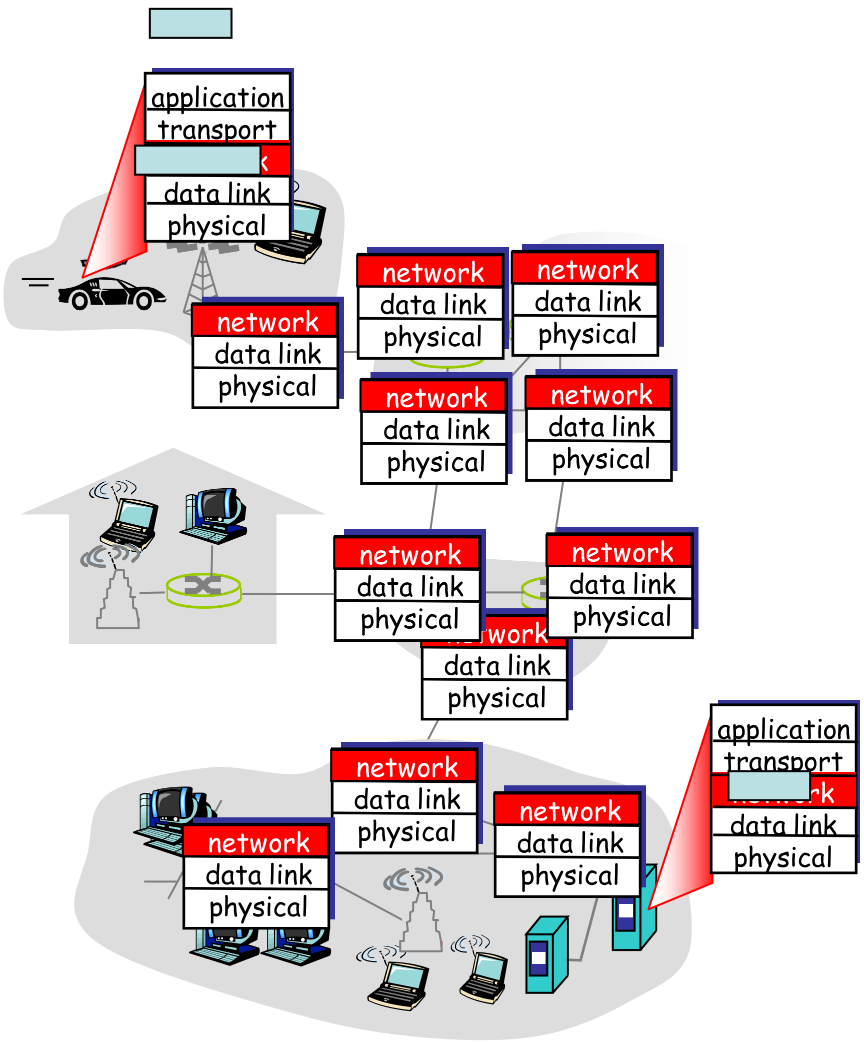
- Data transmitted in small packets
  - Typically 1000 octets
  - Longer messages split into series of packets
  - Each packet contains a portion of user data plus some control info
- Control info
  - Routing (addressing) info
- Packets are received, stored briefly (buffered) and passed on to the next node
  - Store and forward

Use of Packets



Network layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it



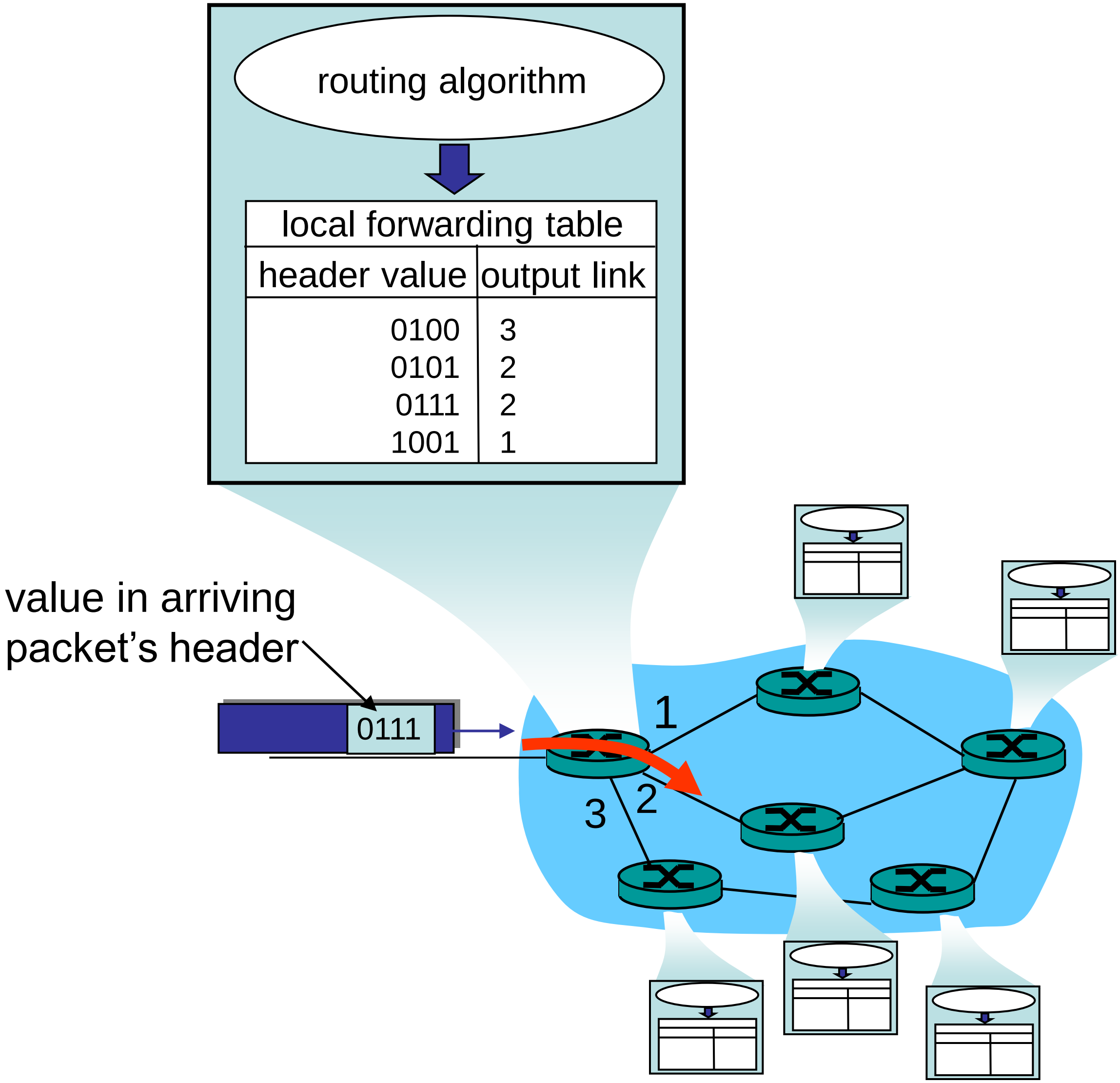
Two Key Network-Layer Functions

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

analogy:

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

Interplay between routing and forwarding



Connection setup

- 3rd important function in *some* network architectures:
  - ATM, frame relay, X.25
- before datagrams flow, two end hosts *and* intervening routers establish virtual connection
  - routers get involved
- network vs transport layer connection service:
  - network: between two hosts (may also involve intervening routers in case of VCs)
  - transport: between two processes

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

Network layer service models

Network Architecture	Service Model	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion feedback
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

Network layer connection and connection-less service

- datagram network provides network-layer connectionless service
- VC network provides network-layer connection service
- analogous to the transport-layer services, but:
  - » service: host-to-host
  - » no choice: network provides one or the other
  - » implementation: in network core

Virtual circuits

"source-to-dest path behaves much like telephone circuit"

- performance-wise
- network actions along source-to-dest path

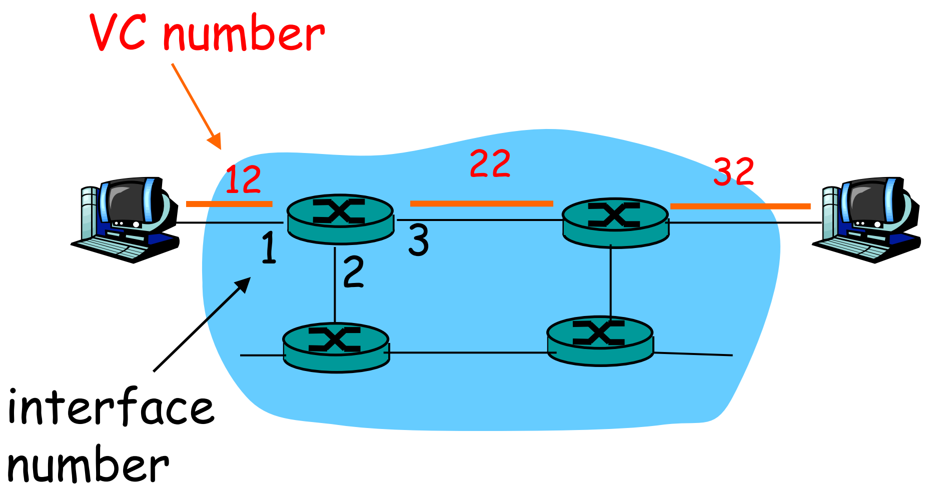
- call setup, teardown for each call *before* data can flow
- each packet carries VC identifier (not destination host address)
- *every* router on source-dest path maintains "state" for each passing connection
- link, router resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)

VC implementation

a VC consists of:

1. path from source to destination
2. VC numbers, one number for each link along path
3. entries in forwarding tables in routers along path

- packet belonging to VC carries VC number (rather than dest address)
- VC number can be changed on each link.
  - New VC number comes from forwarding table



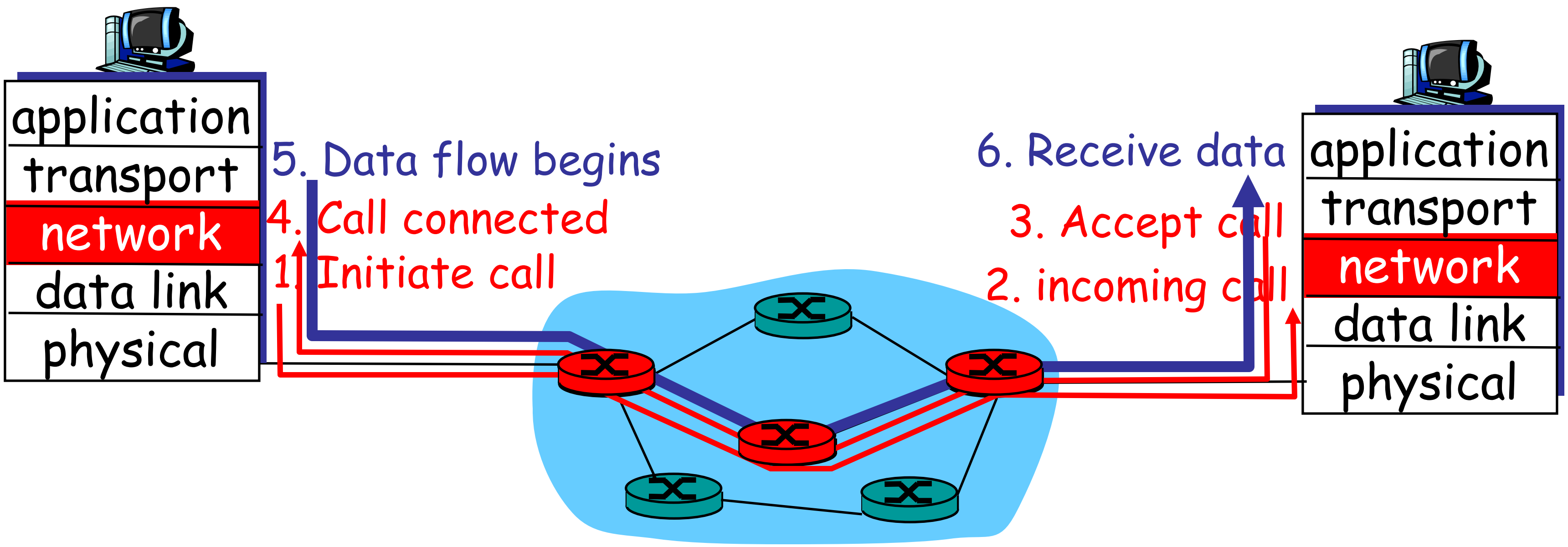
Forwarding table in northwest router:

Incoming interface	Incoming VC #	Outgoing interface	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
…	…	…	…

Routers maintain connection state information!

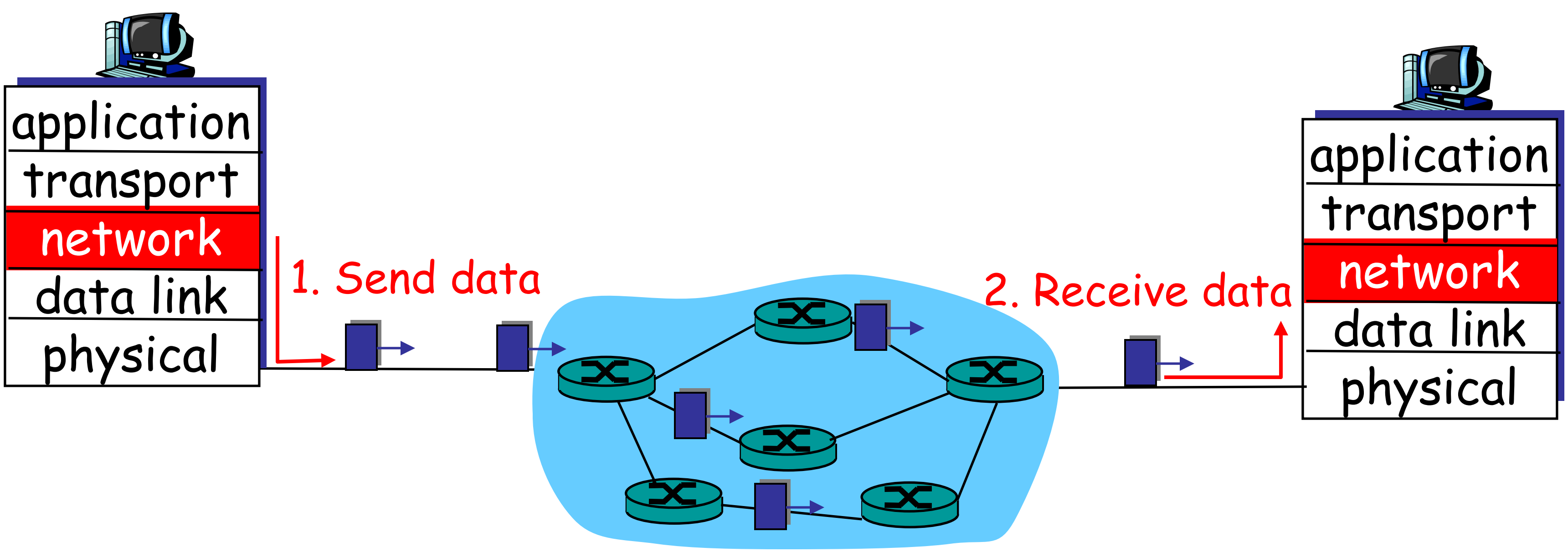
Virtual circuits: signaling protocols

- used to setup, maintain teardown VC
- used in ATM, frame-relay, X.25
- not used in today's Internet



Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of “connection”
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths



Forwarding table

4 billion possible entries

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

Longest prefix matching

Prefix Match	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?

Datagram or VC network: why?

Internet (datagram)

- data exchange among computers
  - » "elastic" service, no strict timing req.
- "smart" end systems (computers)
  - » can adapt, perform control, error recovery
  - » simple inside network, complexity at "edge"
- many link types
  - » different characteristics
  - » uniform service difficult

ATM (VC)

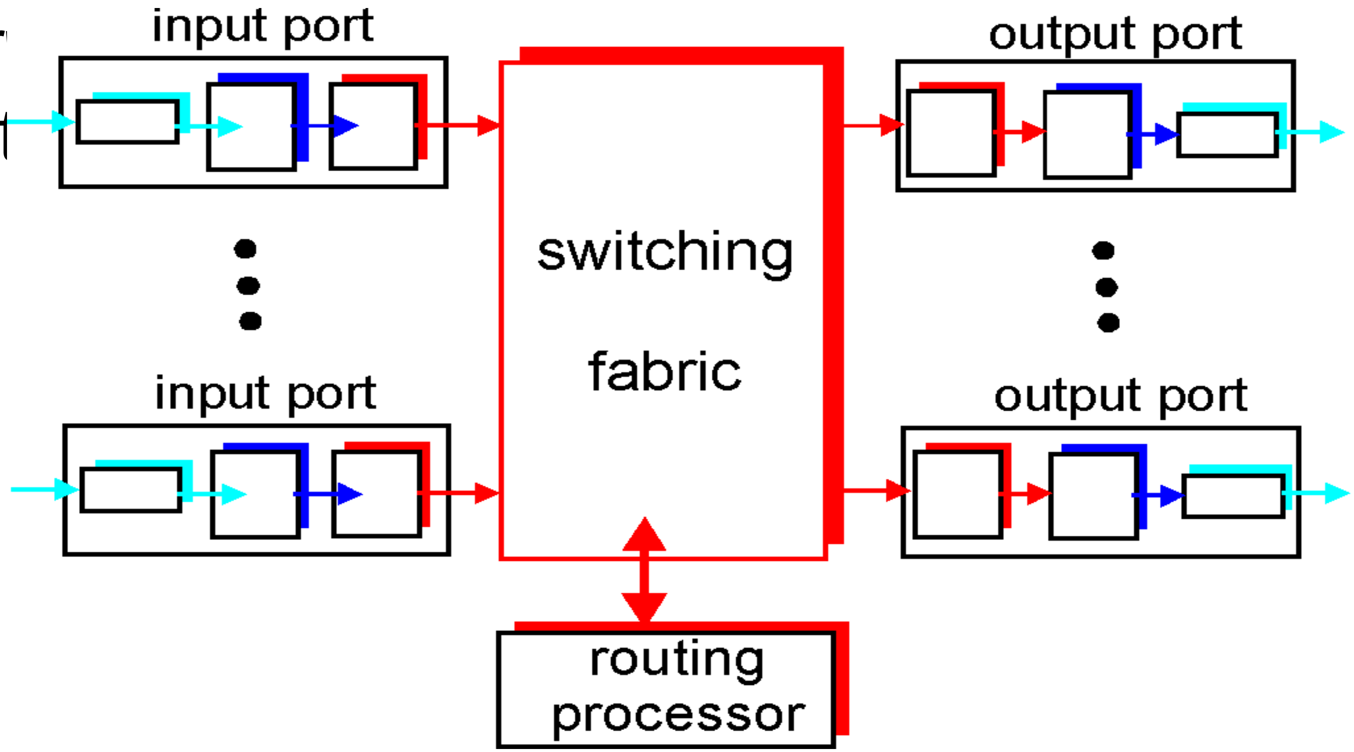
- evolved from telephony
- human conversation:
  - » strict timing, reliability requirements
  - » need for guaranteed service
- "dumb" end systems
  - » telephones
  - » complexity inside network

Networks Part 1 Agenda

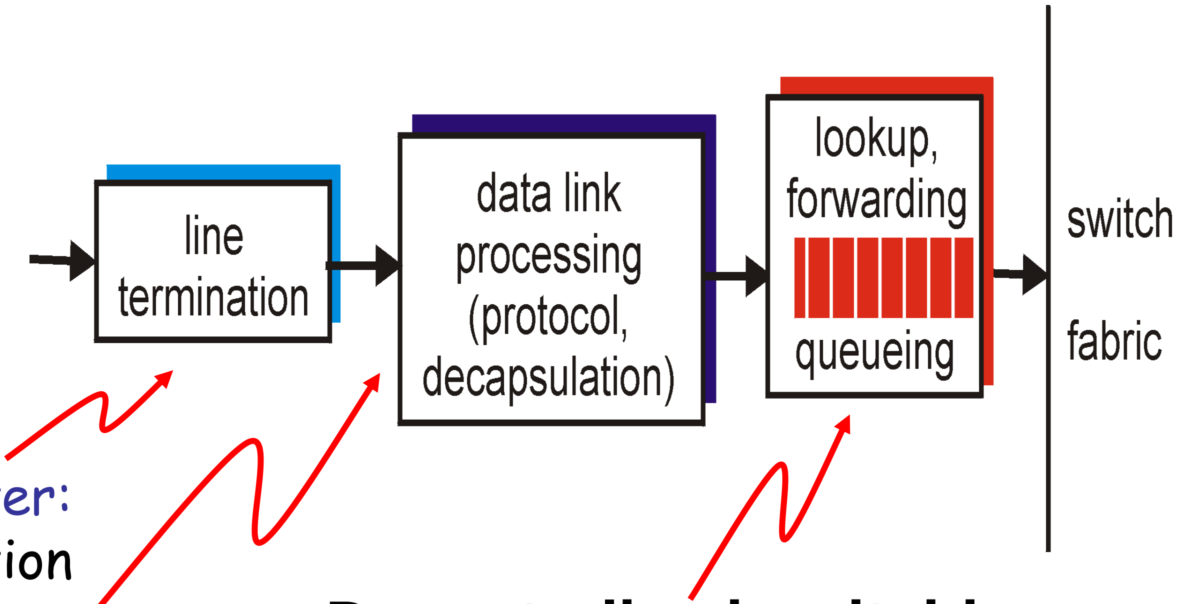
- Introduction
- Virtual circuit and datagram networks
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

Two key router functions:

- run routing algorithms/protocol (RIP, OSPF, BGP)
- *for*
  out



Input Port Functions



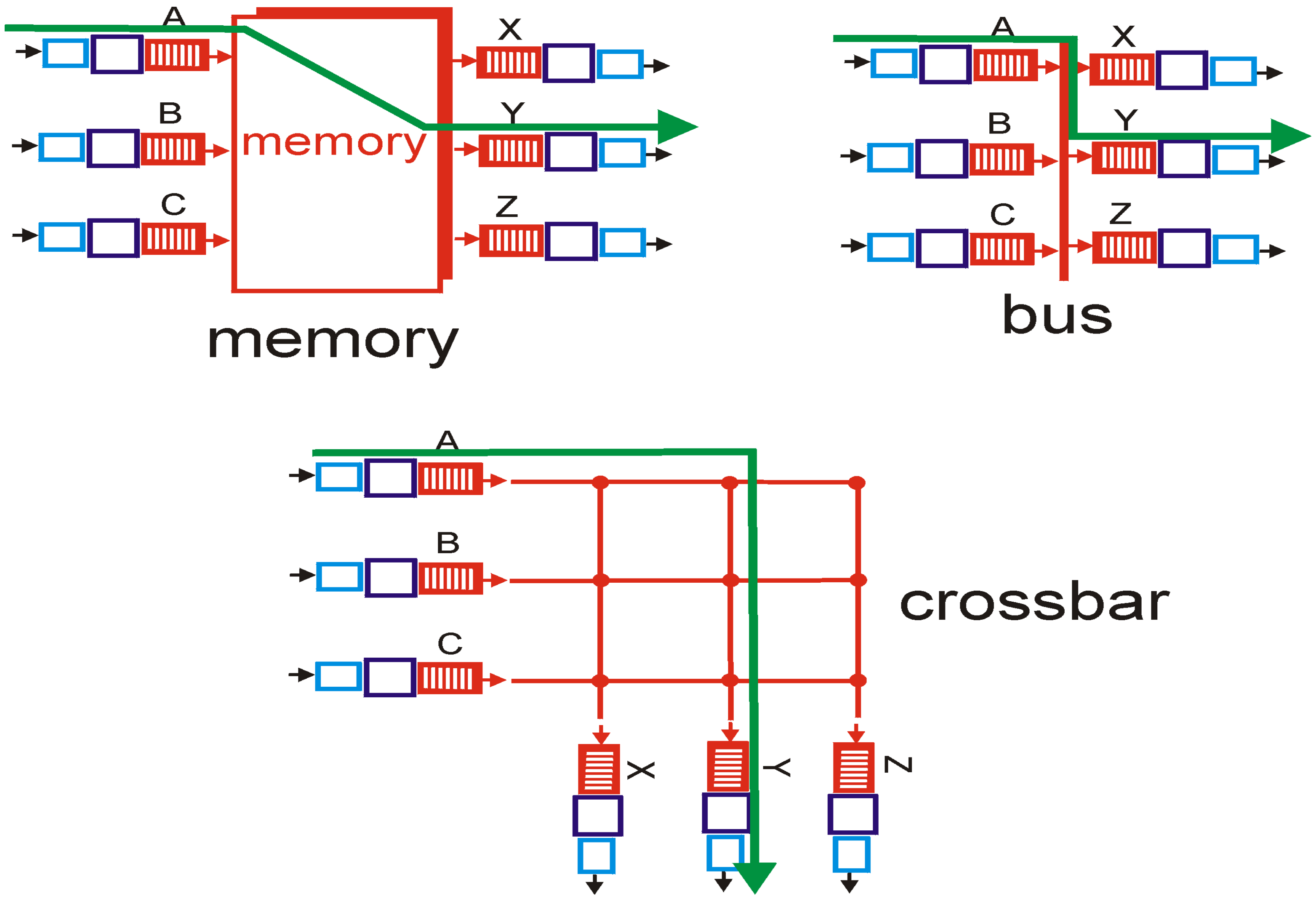
Physical layer:
bit-level reception

Data link layer:
e.g., Ethernet
(see Textbook Chapter 5)

Decentralized switching:

- given datagram dest., lookup output port using forwarding table in input port memory
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

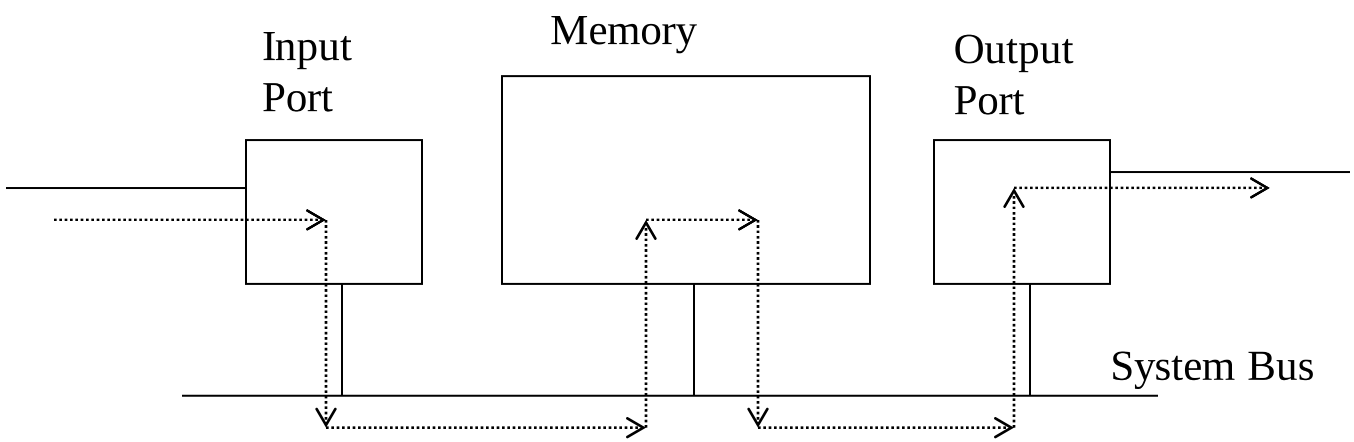
Three types of switching fabrics



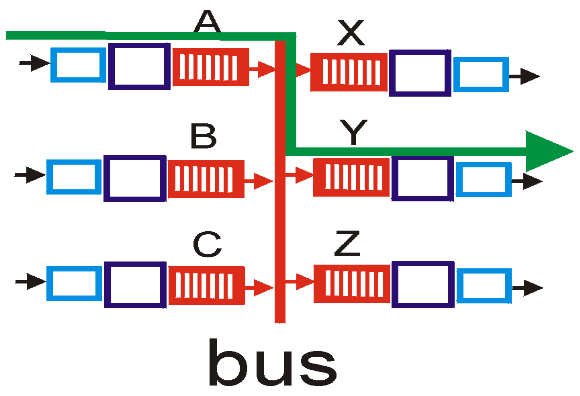
Switching Via Memory

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 a Bus



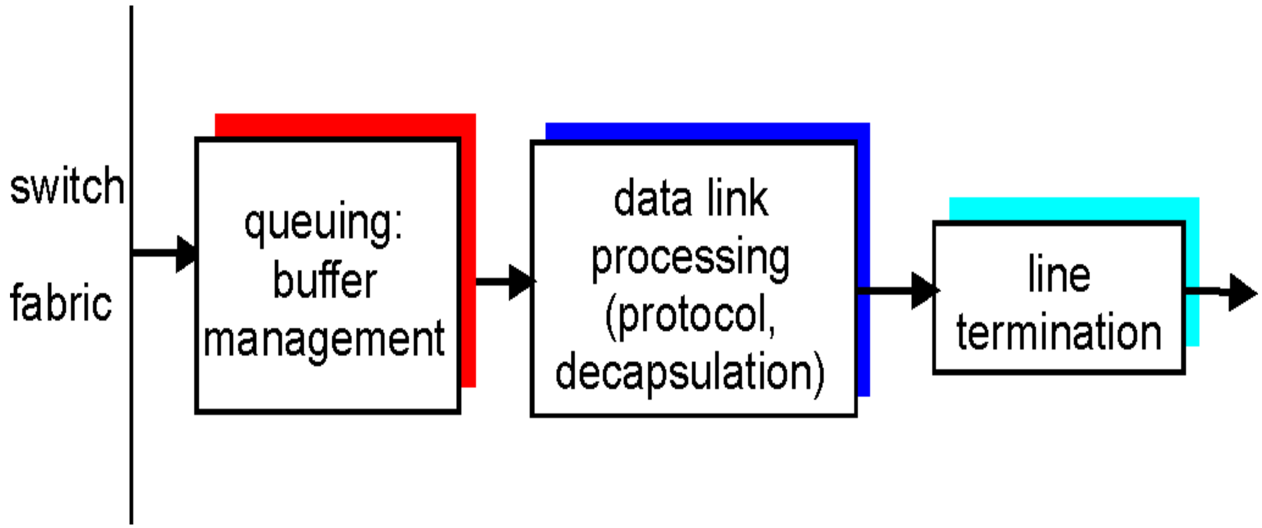
- datagram from input port memory

  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 An Interconnection Network

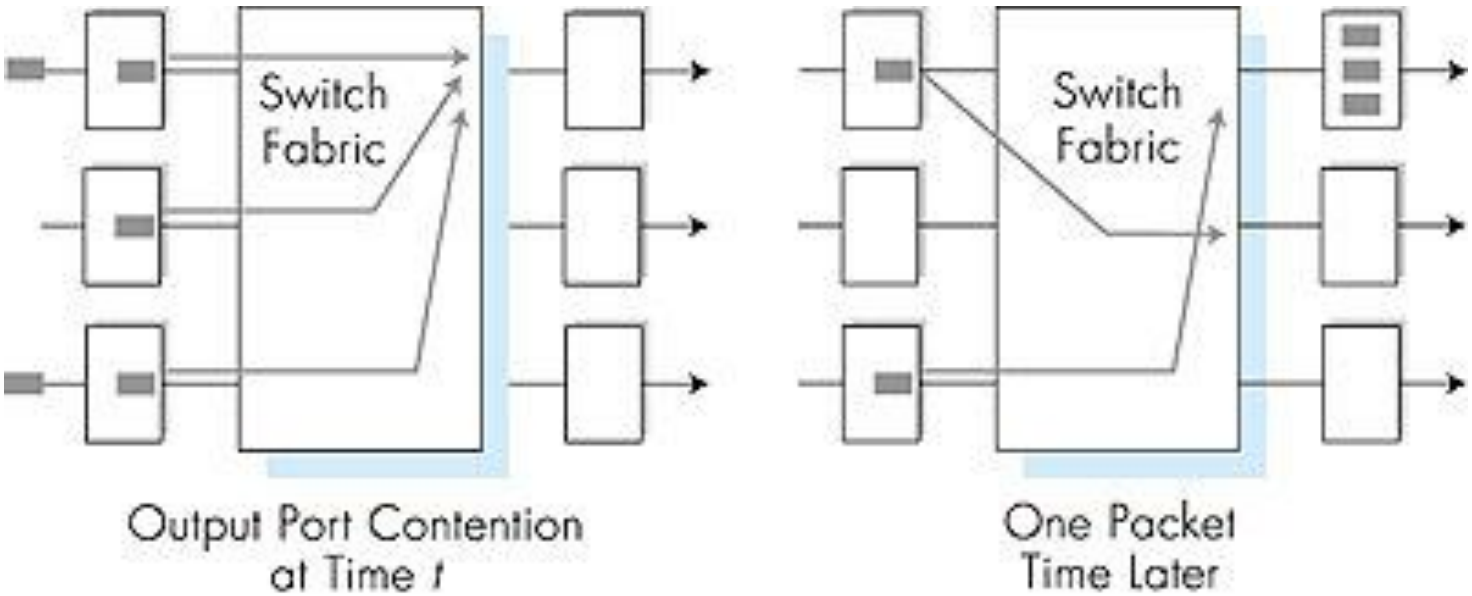
- overcome bus bandwidth limitations
- Banyan networks, other interconnection nets initially developed to connect processors in multiprocessor
- advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through the interconnection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

Output Ports



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

Output port queueing



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

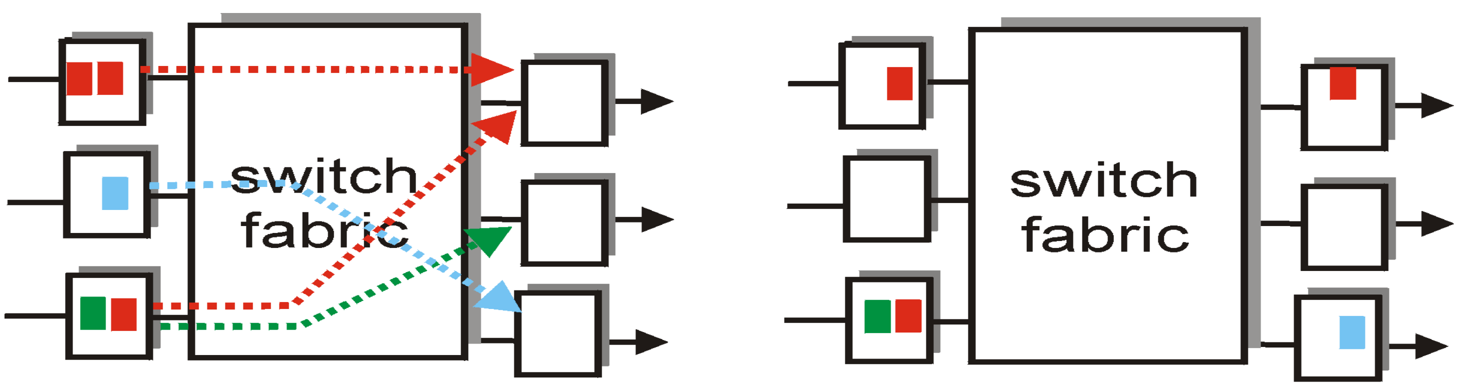
How much buffering?

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity C
  - » e.g., C = 10 Gps link: 2.5 Gbit buffer
- Recent recommendation: with *N* flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$

Input Port Queuing

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



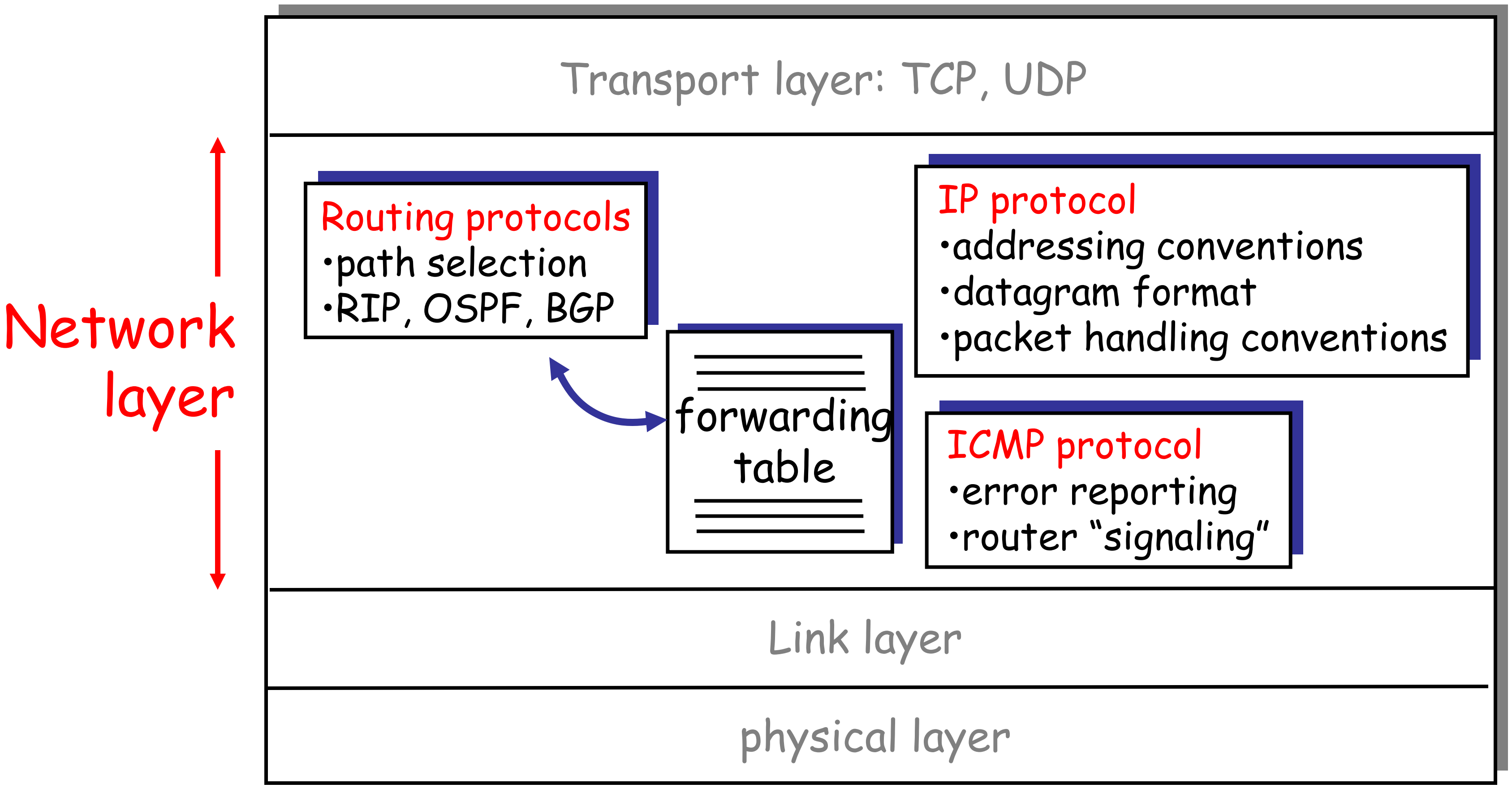
output port contention at time t - only one red packet can be transferred

green packet experiences HOL blocking

Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

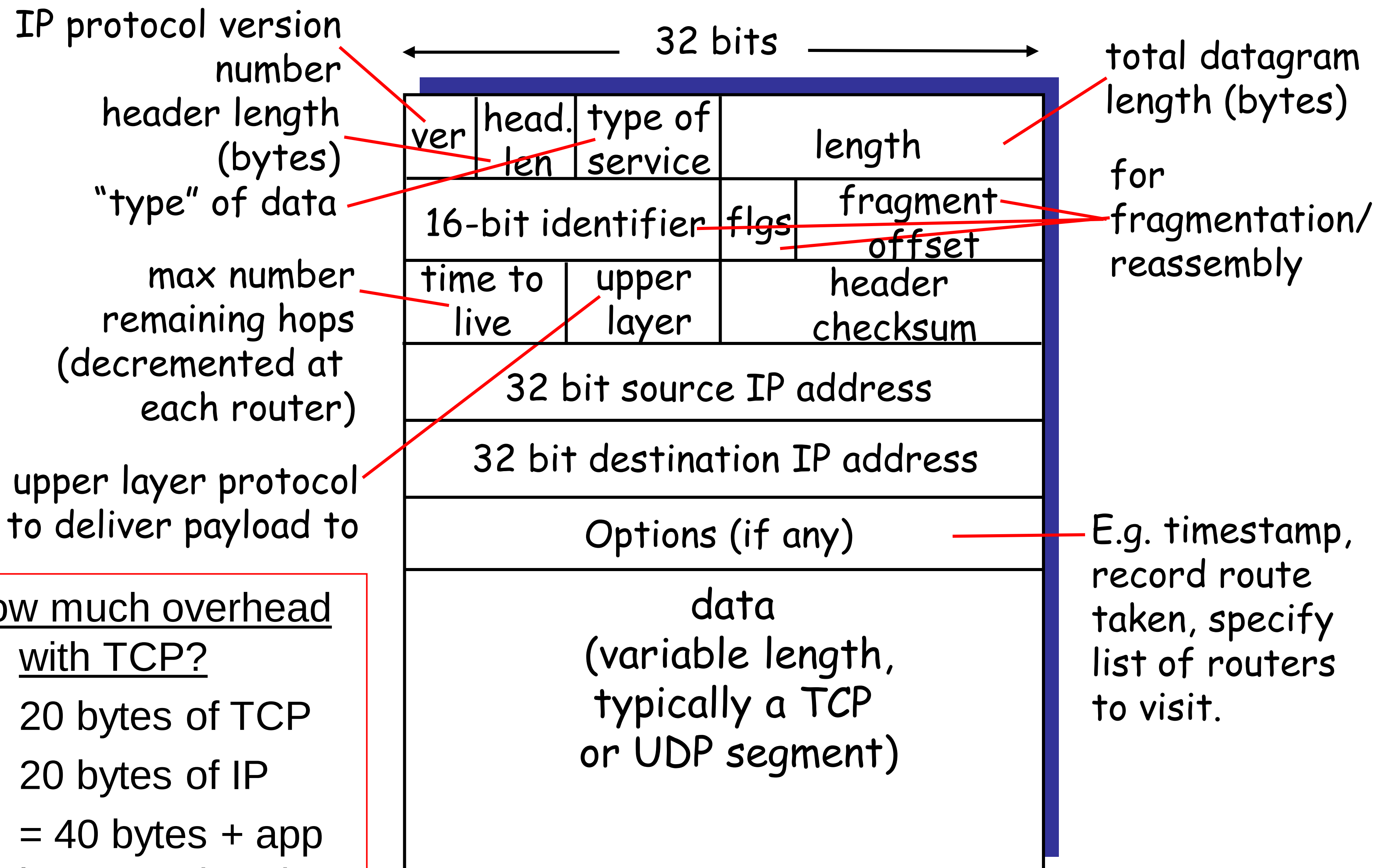
Host, router network layer functions:



Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

IP datagram format

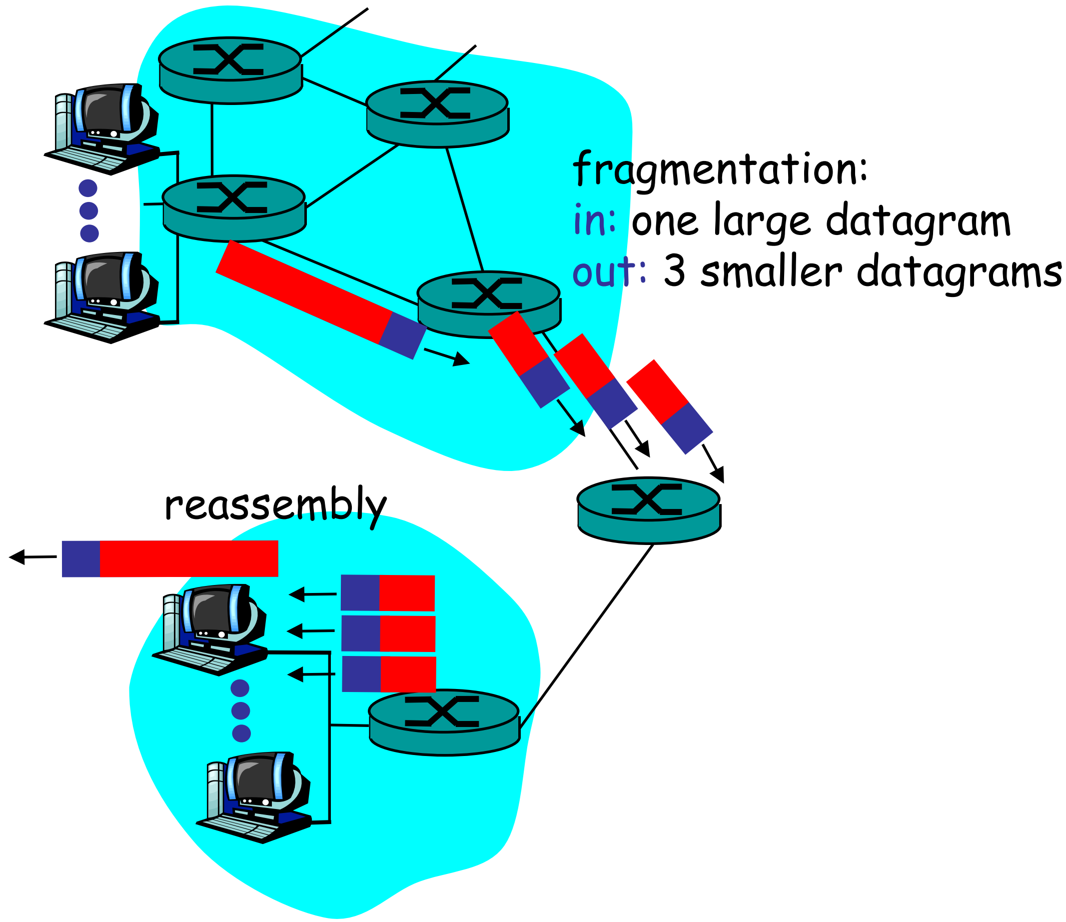


<u>how much overhead with TCP?</u>

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

IP Fragmentation & Reassembly

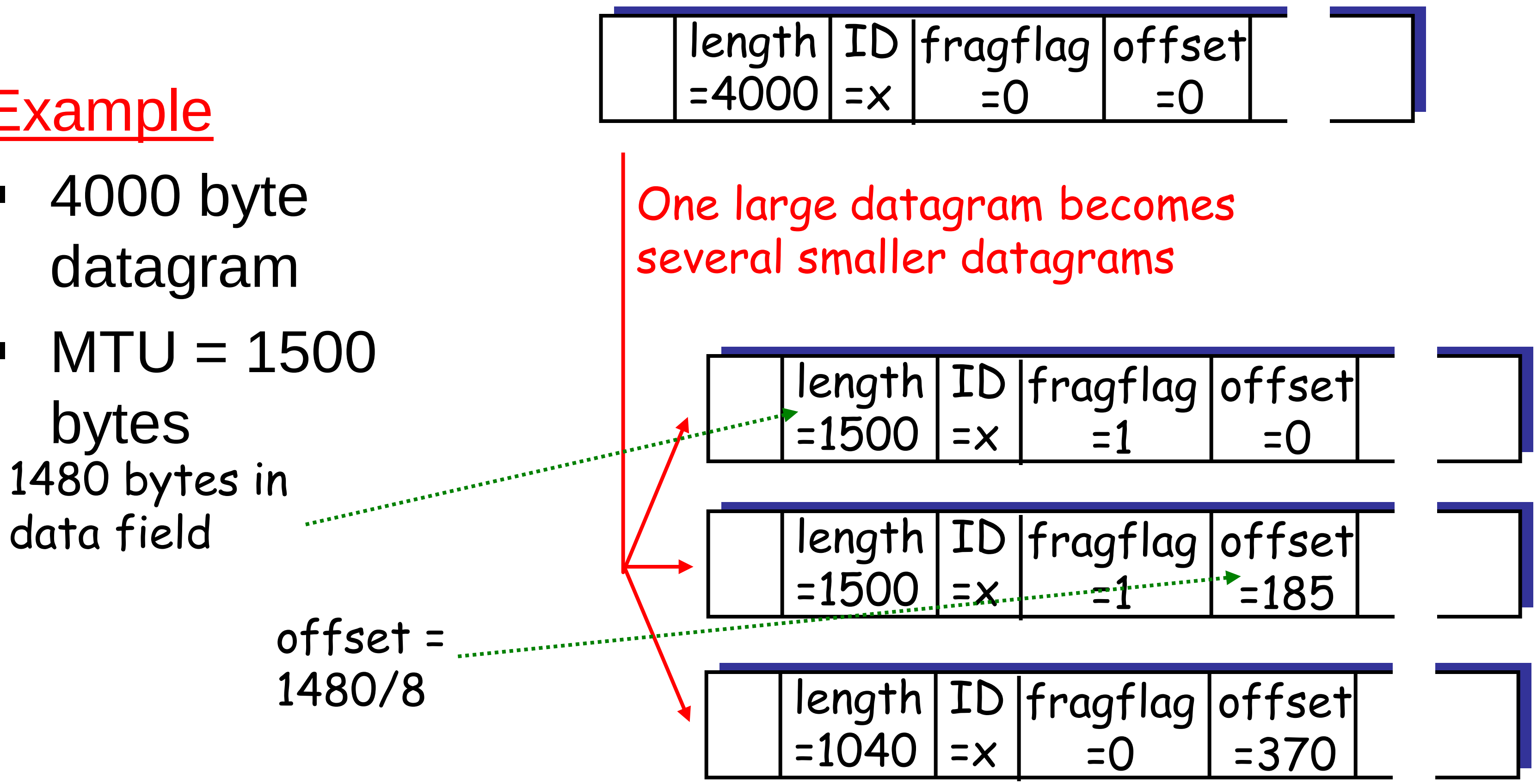
- network links have MTU (max.transfer size) - largest possible link-level frame.
  - » different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - » one datagram becomes several datagrams
  - » "reassembled" only at final destination
  - » IP header bits used to identify,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



IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes

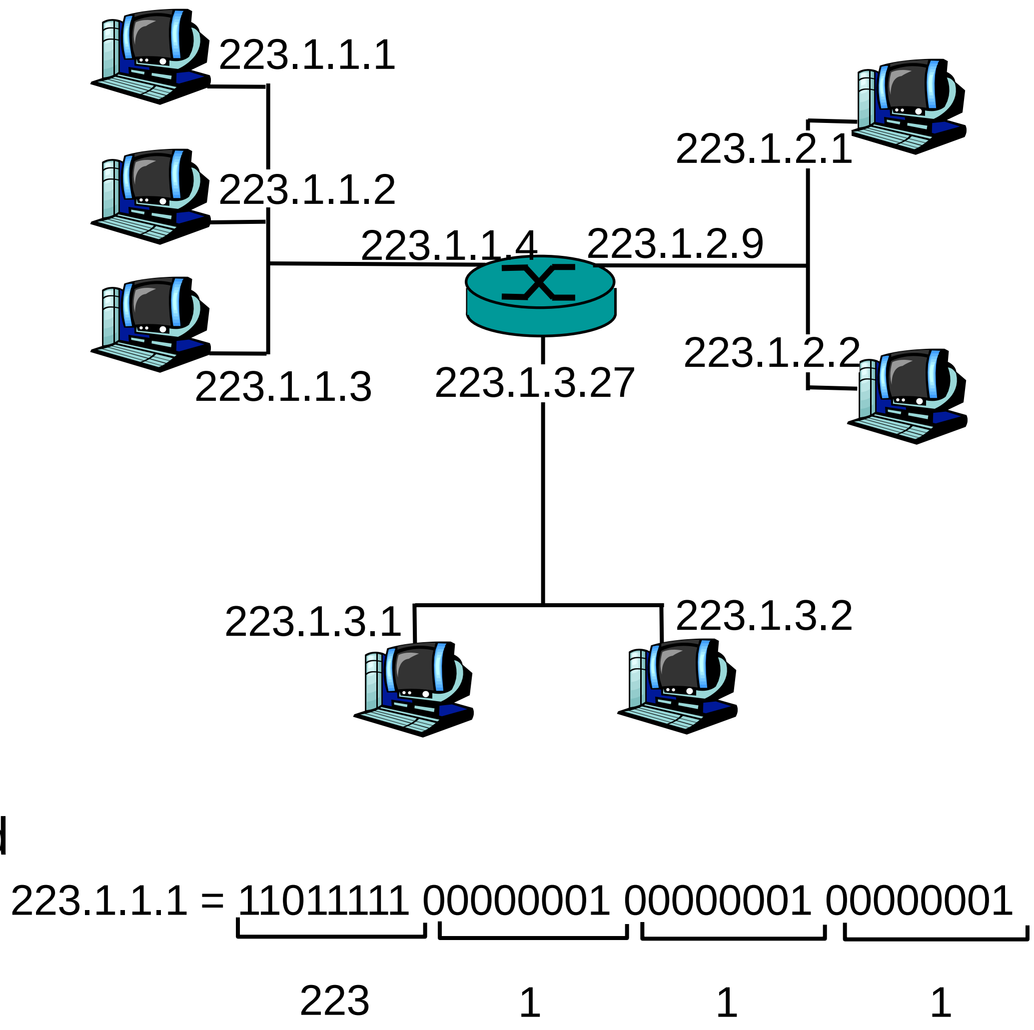


Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

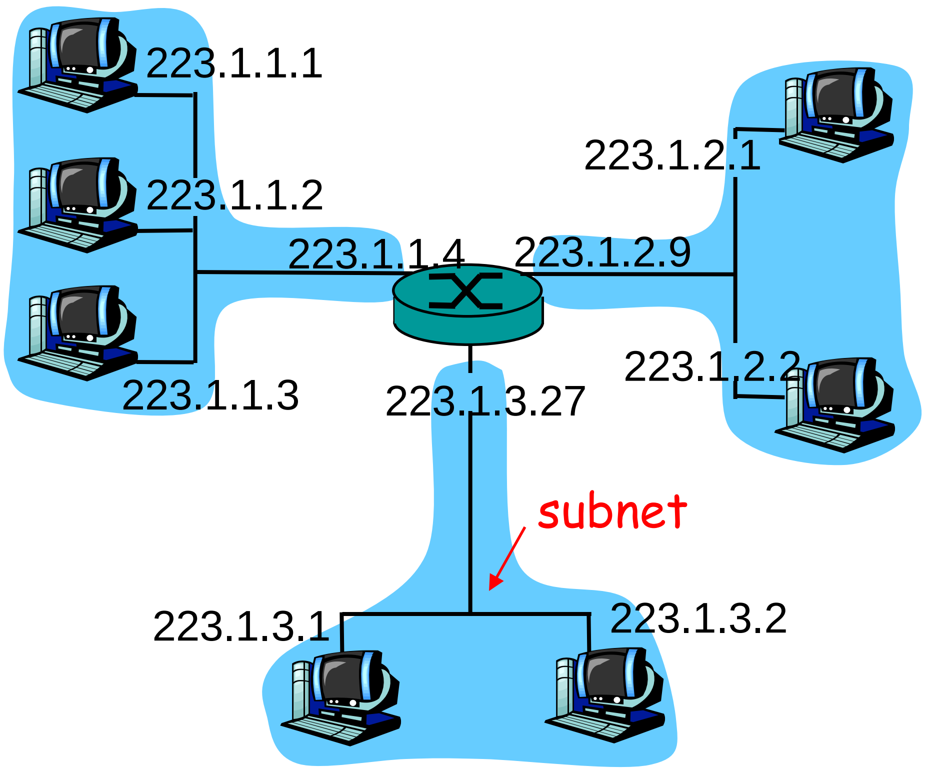
IP Addressing: introduction

- 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 interface
  - IP addresses associated with each interface



Subnets

- IP address:
  - » subnet part (high order bits)
  - » host part (low order bits)
- *What's a subnet ?*
  - » device interfaces with same subnet part of IP address
  - » can physically reach each other without intervening router

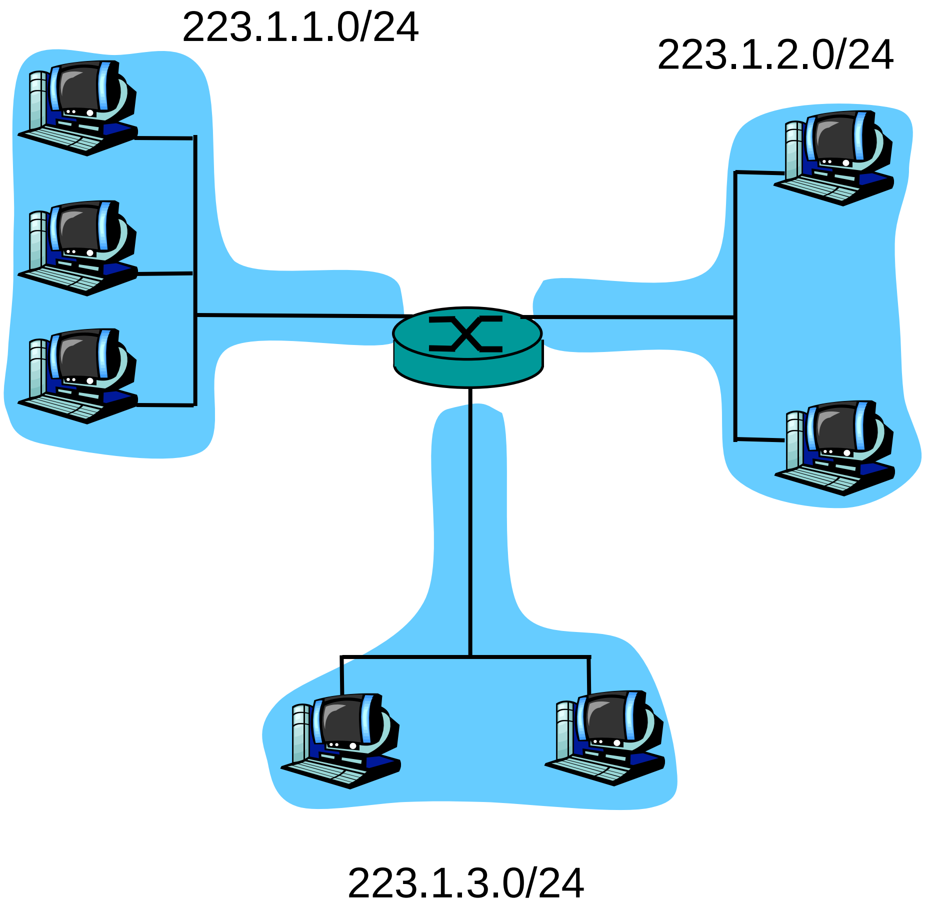


network consisting of 3 subnets

Subnets

<u>Recipe</u>

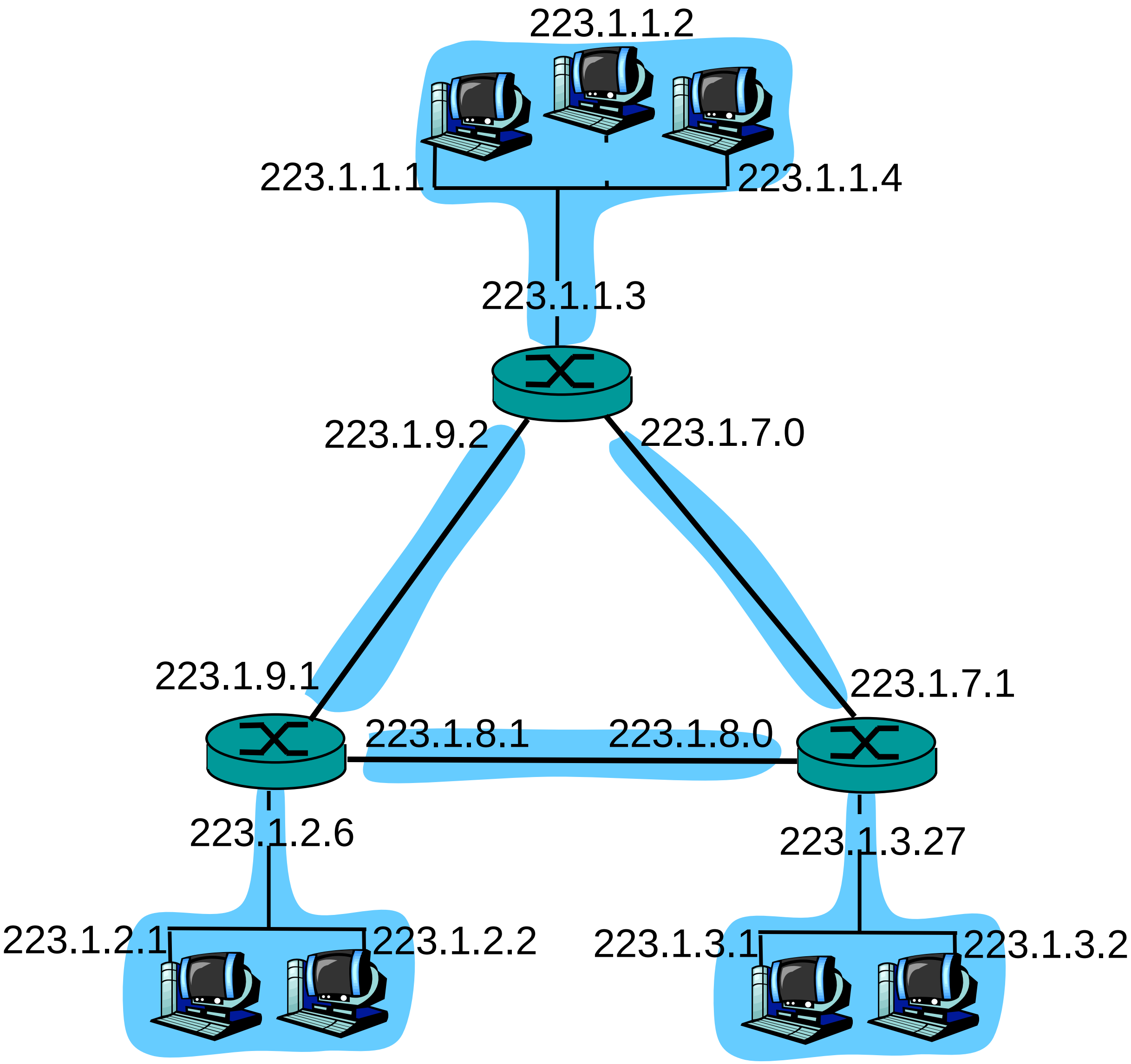
- To determine the subnets, detach each interface from its host or router, creating islands of isolated networks. Each isolated network is called a subnet.



Subnet mask: /24

Subnets

How many?



IP addressing: CIDR

CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary length
- address 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

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
- DHCP: Dynamic Host Configuration Protocol: dynamically get address from as server
  - » "plug-and-play"

DHCP: Dynamic Host Configuration Protocol

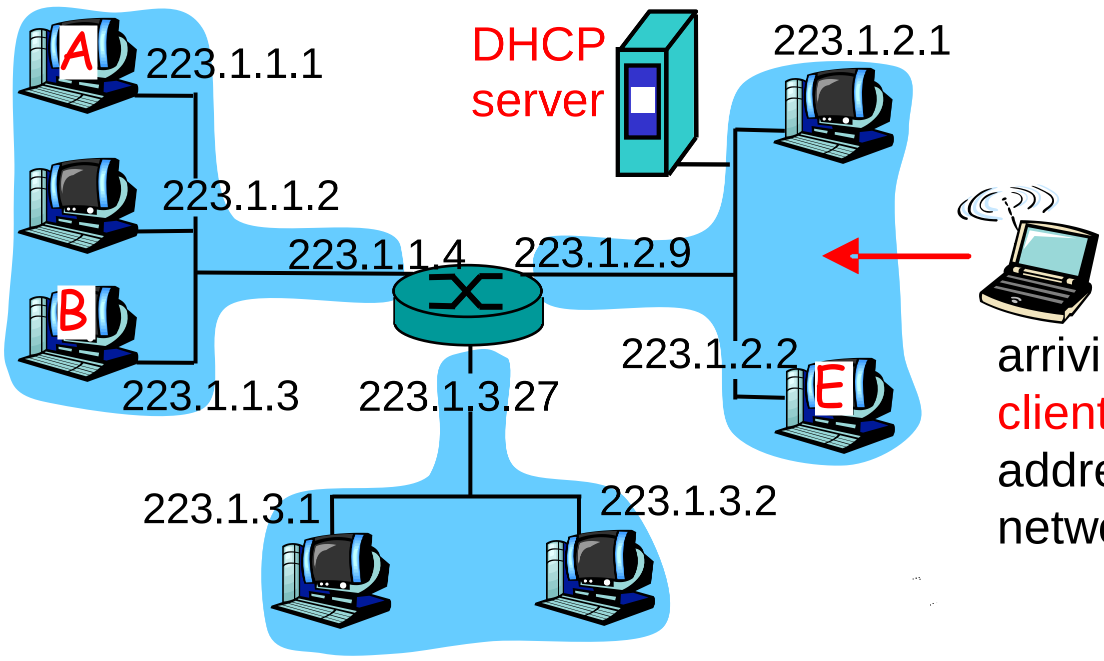
Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- Can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected an "on")
- Support for mobile users who want to join network (more shortly)

DHCP overview:

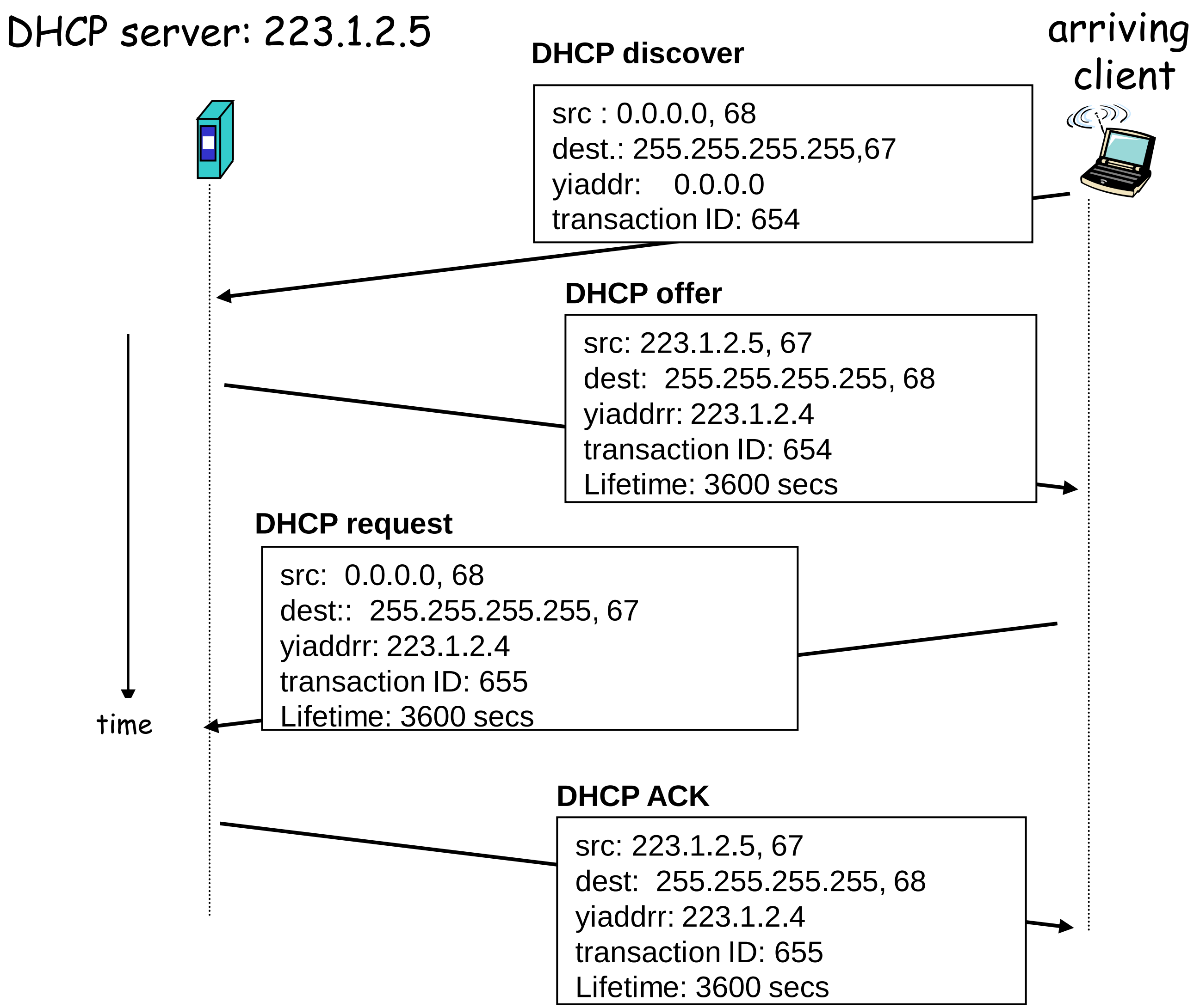
- » host broadcasts "DHCP discover" msg [optional]
- » DHCP server responds with "DHCP offer" msg [optional]
- » host requests IP address: "DHCP request" msg
- » DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



arriving DHCP client needs address in this network

DHCP client-server scenario

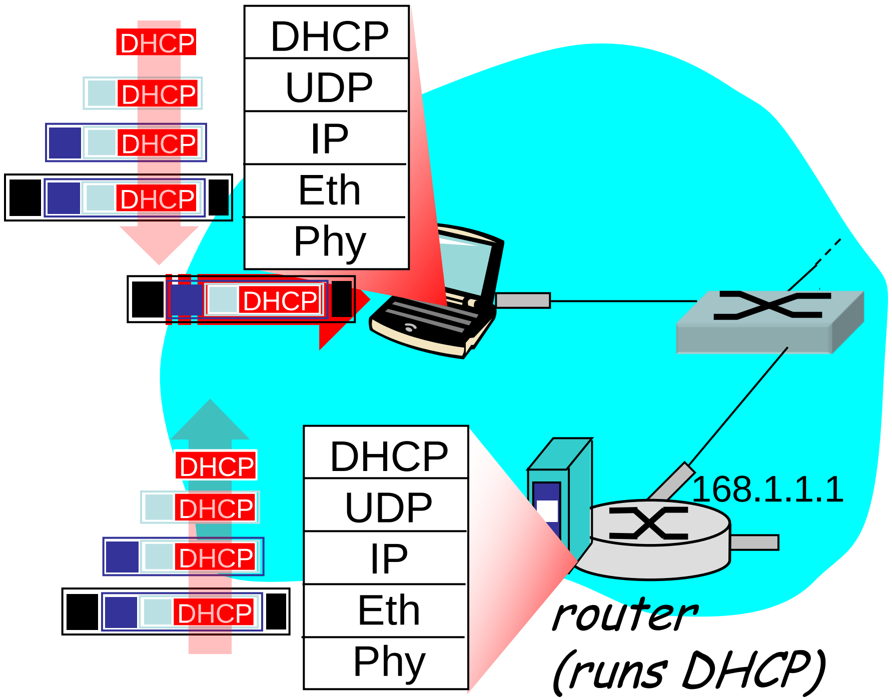


DHCP: more than IP address

DHCP can return more than just allocated IP address on subnet:

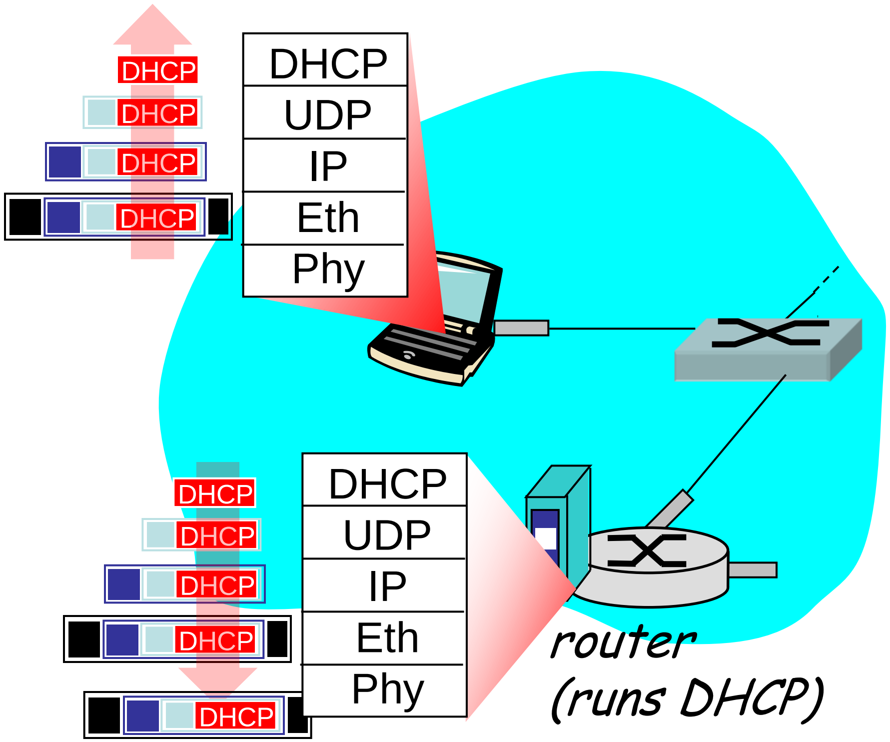
- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

DHCP: 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 demux'ed to IP demux'ed, UDP demux'ed to DHCP

DHCP: example



- 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, demux'ing 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

DHCP: wireshark output
(home LAN)

request

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) **DHCP Message Type = DHCP Request**
Option: (61) Client identifier
  Length: 7; Value: 010016D323688A;
  Hardware type: Ethernet
  Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
  Length: 11; Value: 010F03062C2E2F1F21F92B
  1 = Subnet Mask; 15 = Domain Name
  3 = Router; 6 = Domain Name Server
  44 = NetBIOS over TCP/IP Name Server
  ……

reply

Message type: **Boot Reply (2)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
  Length: 12; Value: 445747E2445749F244574092;
  IP Address: 68.87.71.226;
  IP Address: 68.87.73.242;
  IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."

IP addresses: how to get one?

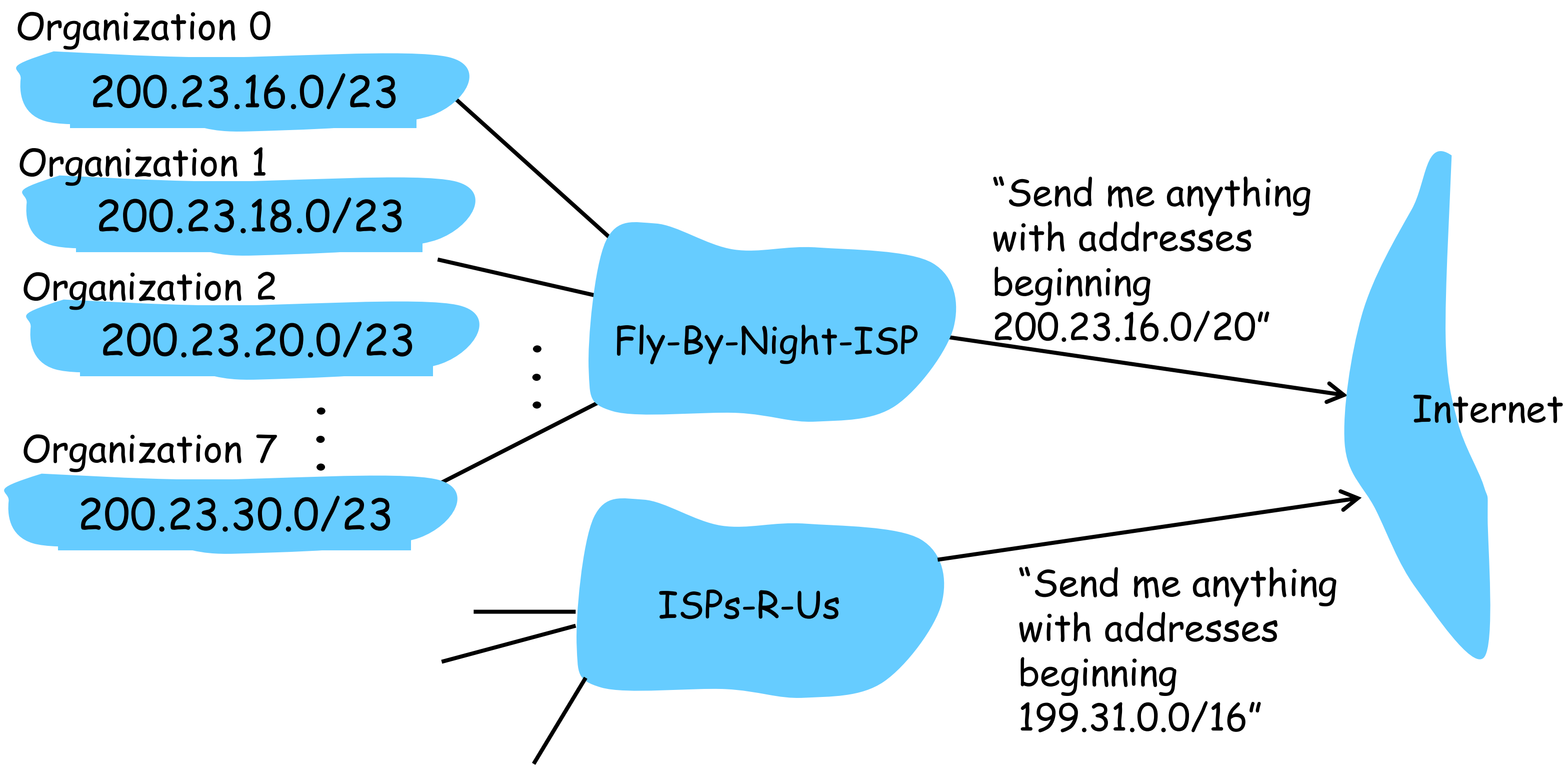
Q: How does *network* get subnet part of IP addr?

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

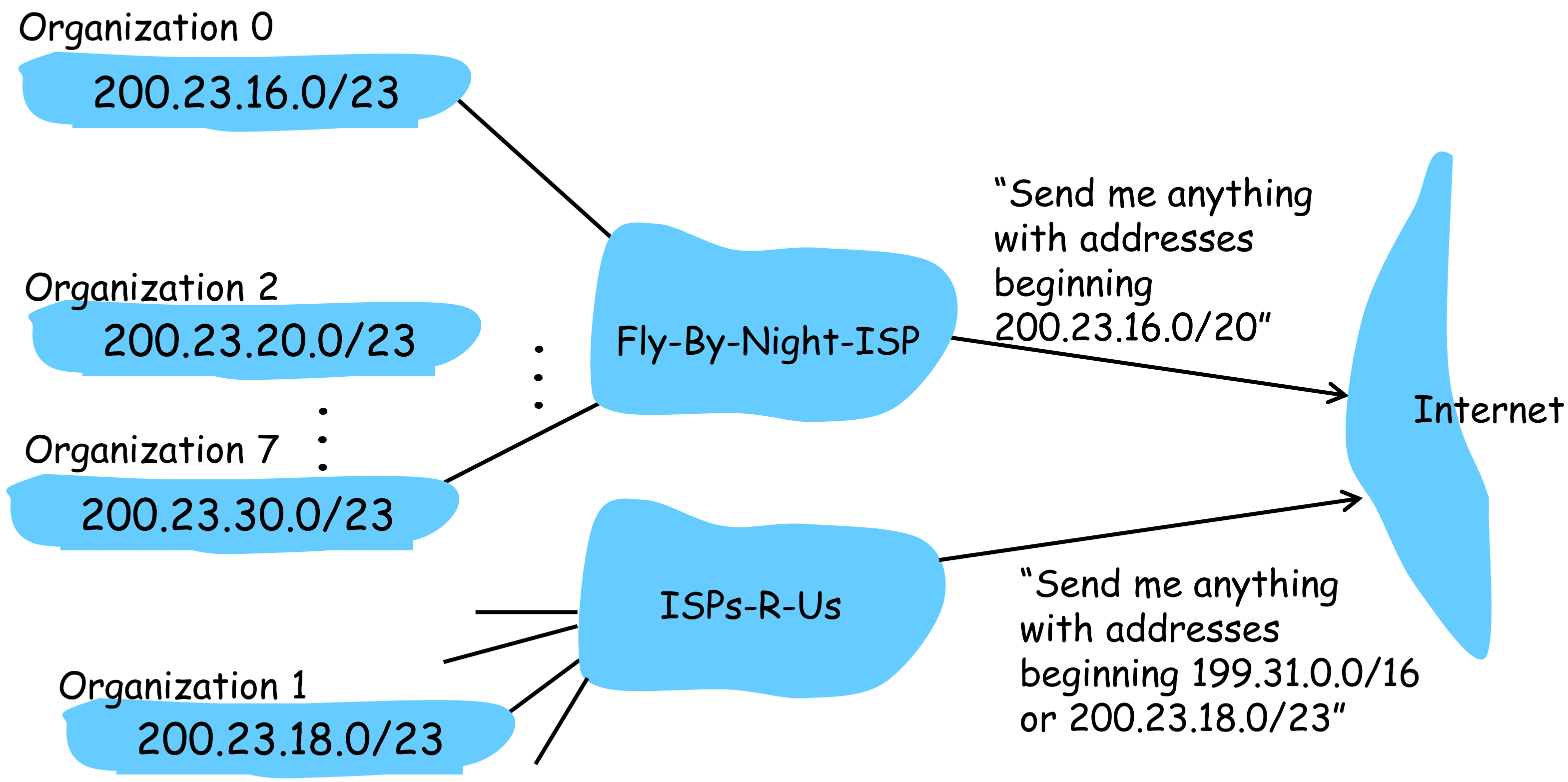
Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

ISPs-R-Us has a more specific route to Organization 1



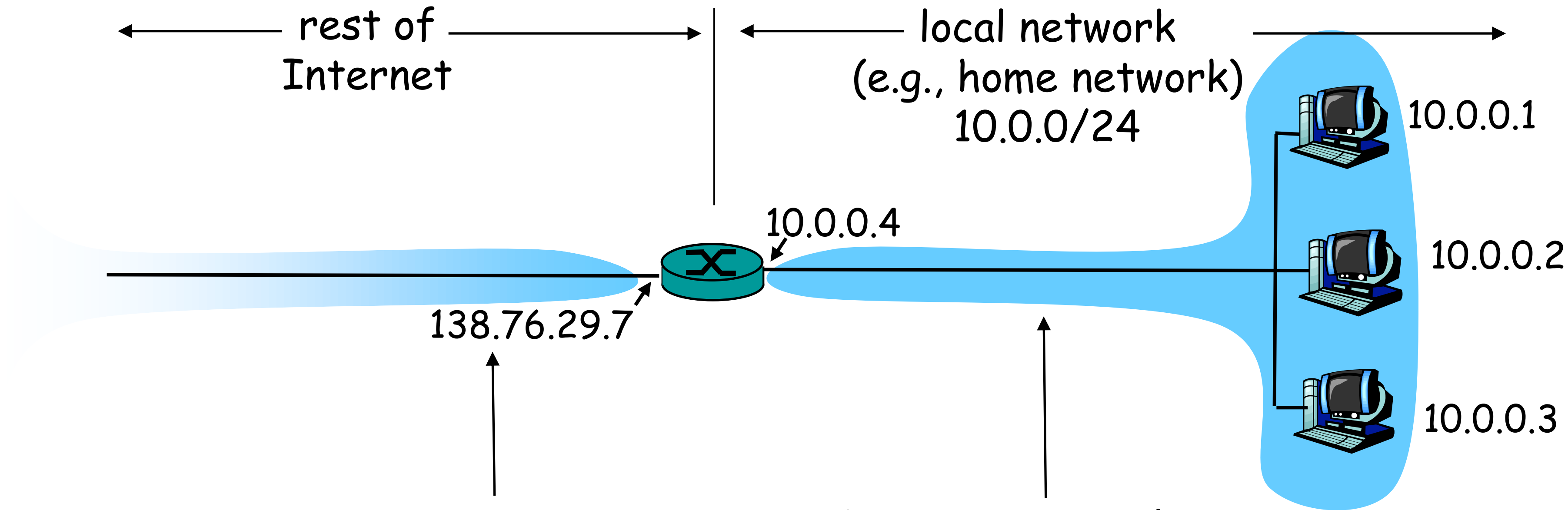
IP addressing: the last word...

Q: How does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

NAT: Network Address Translation



All datagrams *leaving* local network have *same* single source NAT IP address: 138.76.29.7, 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 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).

NAT: Network Address Translation

Implementation: NAT router must:

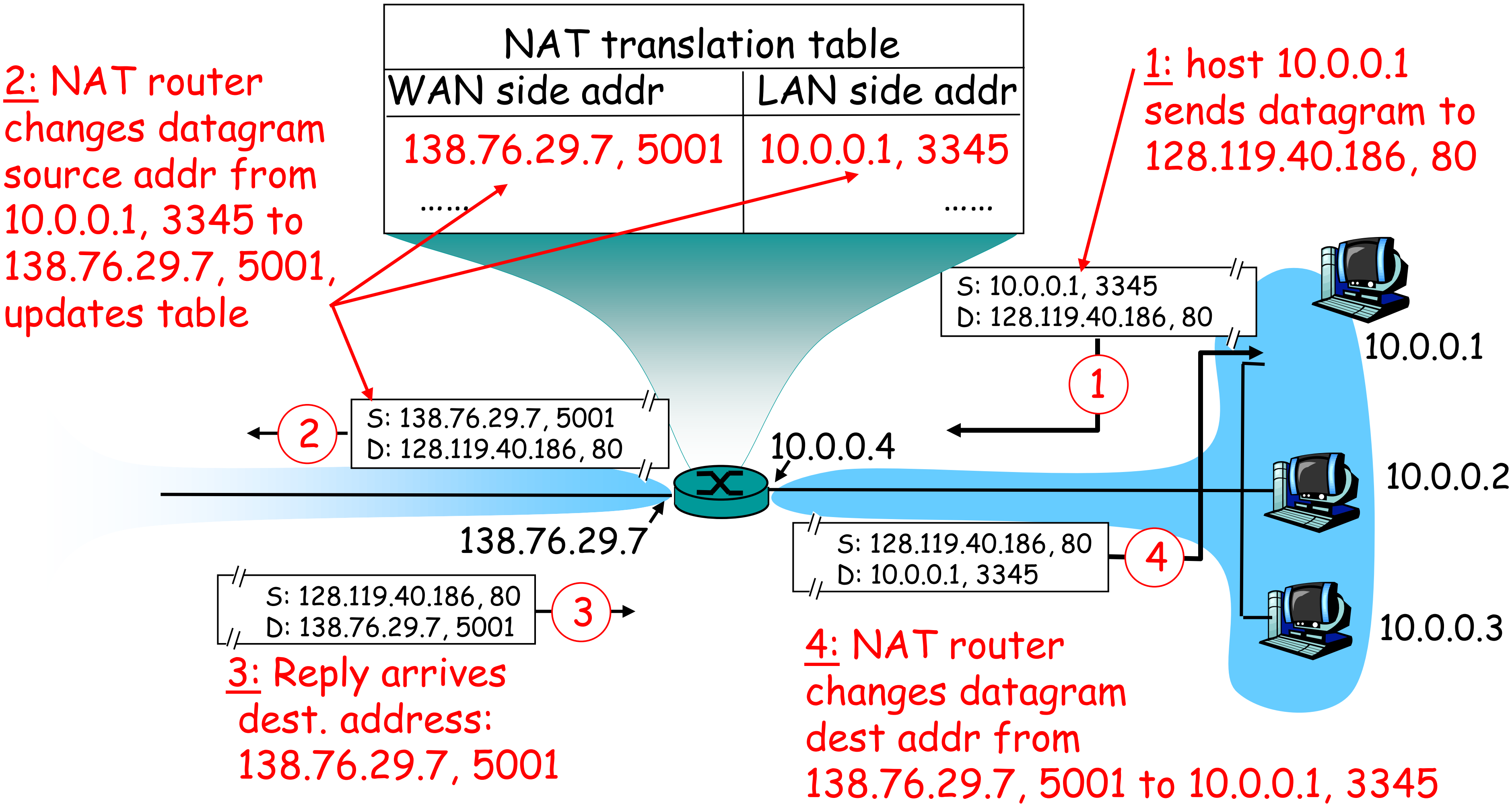
- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

  . . . 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

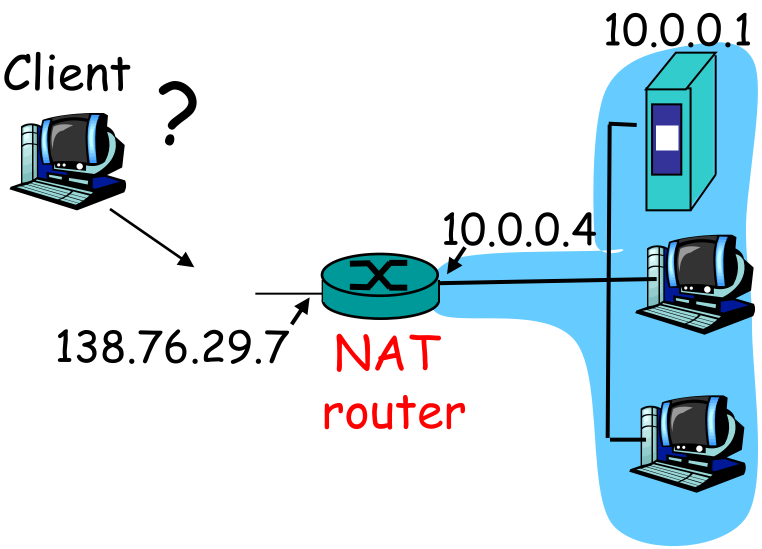


NAT: Network Address Translation

- 16-bit port-number field:
  - » 60,000 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, eg, P2P applications
  - » address shortage should instead be solved by IPv6

NAT traversal problem

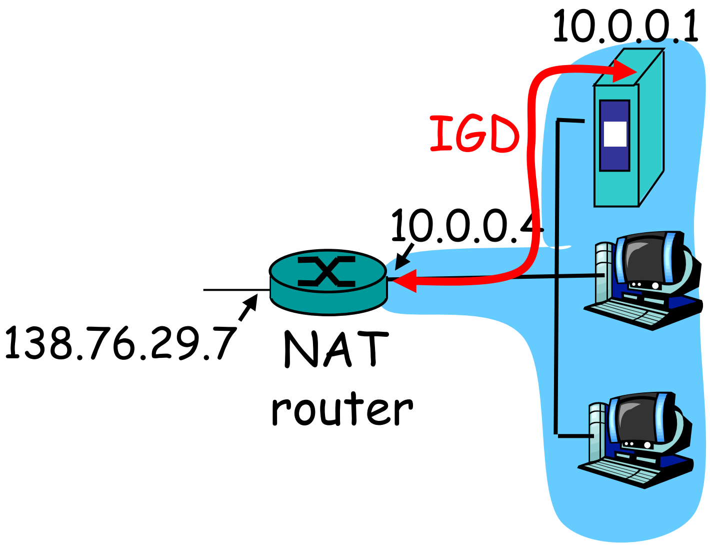
- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible NATted address: 138.76.29.7
- solution 1: statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (123.76.29.7, port 2500) always forwarded to 10.0.0.1 port 25000



NAT traversal problem

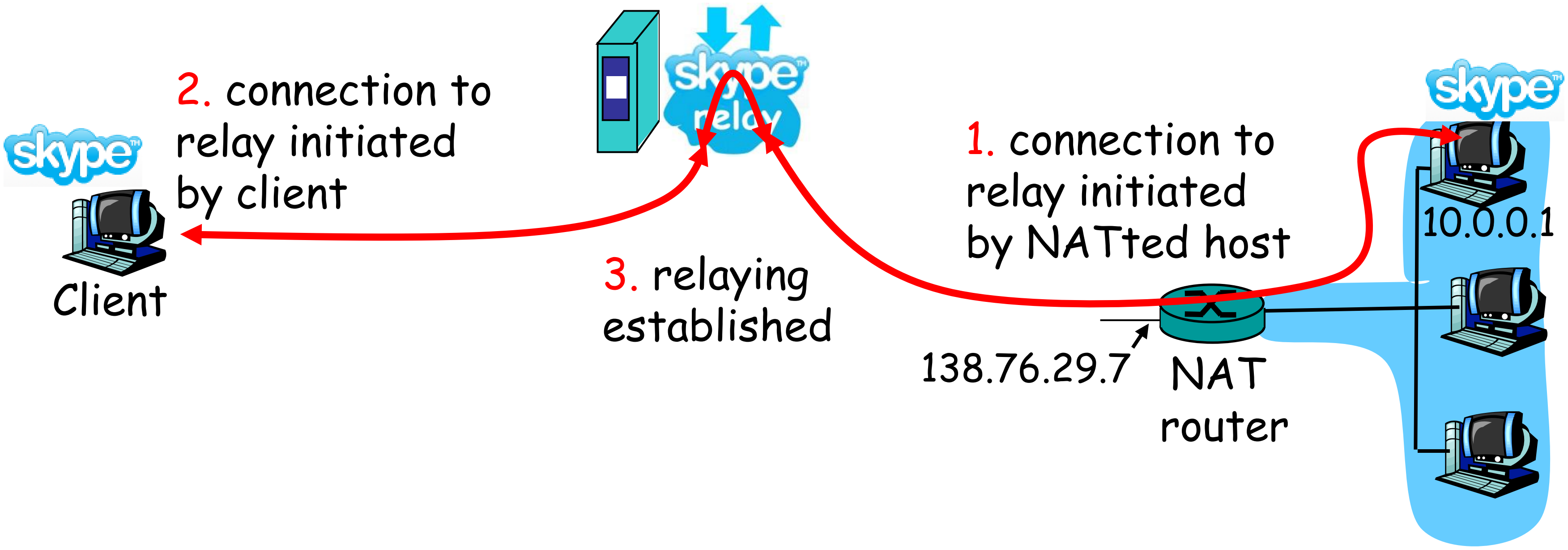
- solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATted host to:
  - learn public IP address (138.76.29.7)
  - add/remove port mappings (with lease times)

  i.e., automate static NAT port map configuration



NAT traversal problem

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



Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What's inside a router
- IP: Internet Protocol
    - Datagram format
    - IPv4 addressing
    - ICMP
    - IPv6

ICMP: Internet Control Message Protocol

- used by hosts & routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer "above" IP:
  - ICMP msgs carried in IP datagrams
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

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

Traceroute and ICMP

- Source sends series of UDP segments to dest
  - » First has TTL =1
  - » Second has TTL=2, etc.
  - » Unlikely port number
- When nth datagram arrives to nth router:
  - » Router discards datagram
  - » And sends to source an ICMP message (type 11, code 0)
  - » Message includes name of router& IP address
- When ICMP message arrives, source calculates RTT
- Traceroute does this 3 times

<u>Stopping criterion</u>

- UDP segment eventually arrives at destination host
- Destination returns ICMP “host unreachable” packet (type 3, code 3)
- When source gets this ICMP, stops.

Networks Part 1 Agenda

- Introduction
- Virtual circuit and datagram networks
- What's inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

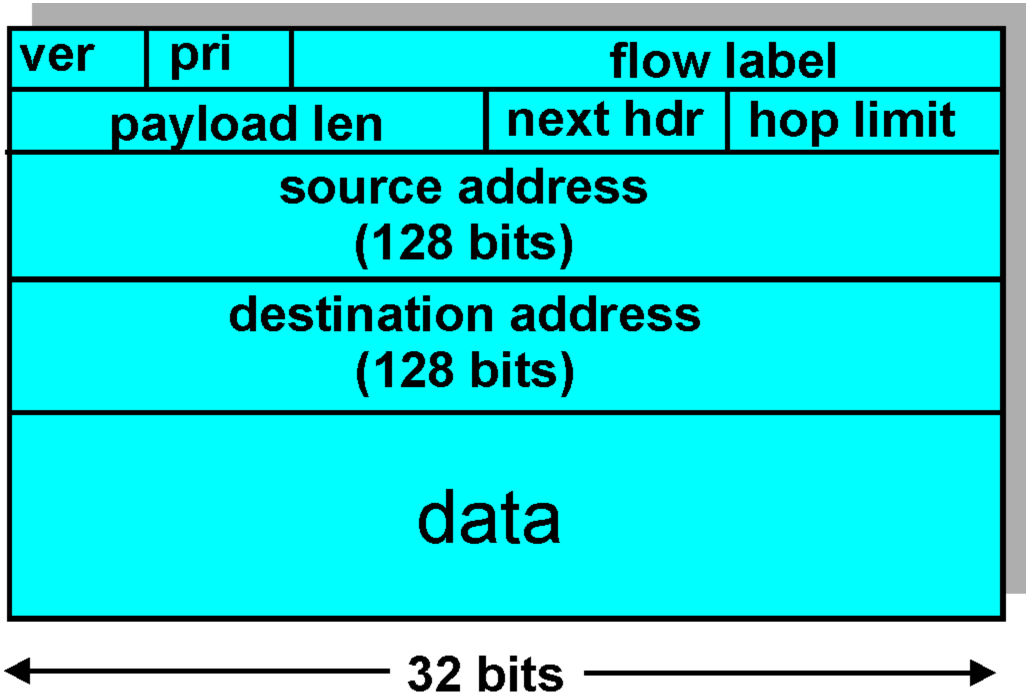
IPv6

- Initial motivation: 32-bit address space soon to be completely allocated.
- Additional motivation:
  - » header format helps speed processing/forwarding
  - » header changes to facilitate QoS

  IPv6 datagram format:
  - » fixed-length 40 byte header
  - » no fragmentation allowed

IPv6 Header (Cont)

Priority: identify priority among datagrams in flow
Flow Label: identify datagrams in same "flow."
(concept of "flow" not well defined).
Next header: identify upper layer protocol for data

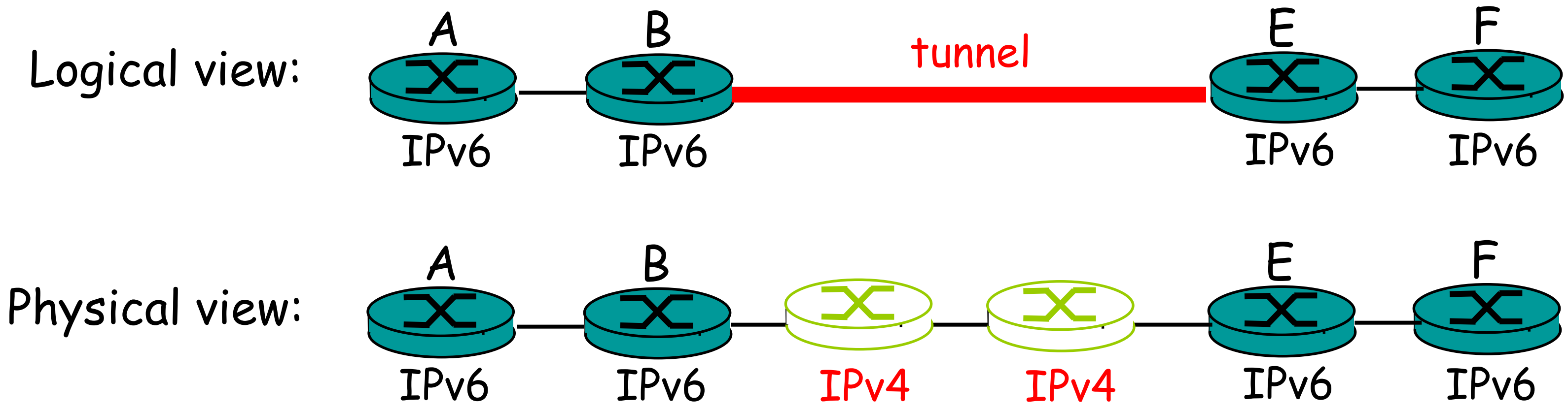


Other Changes from IPv4

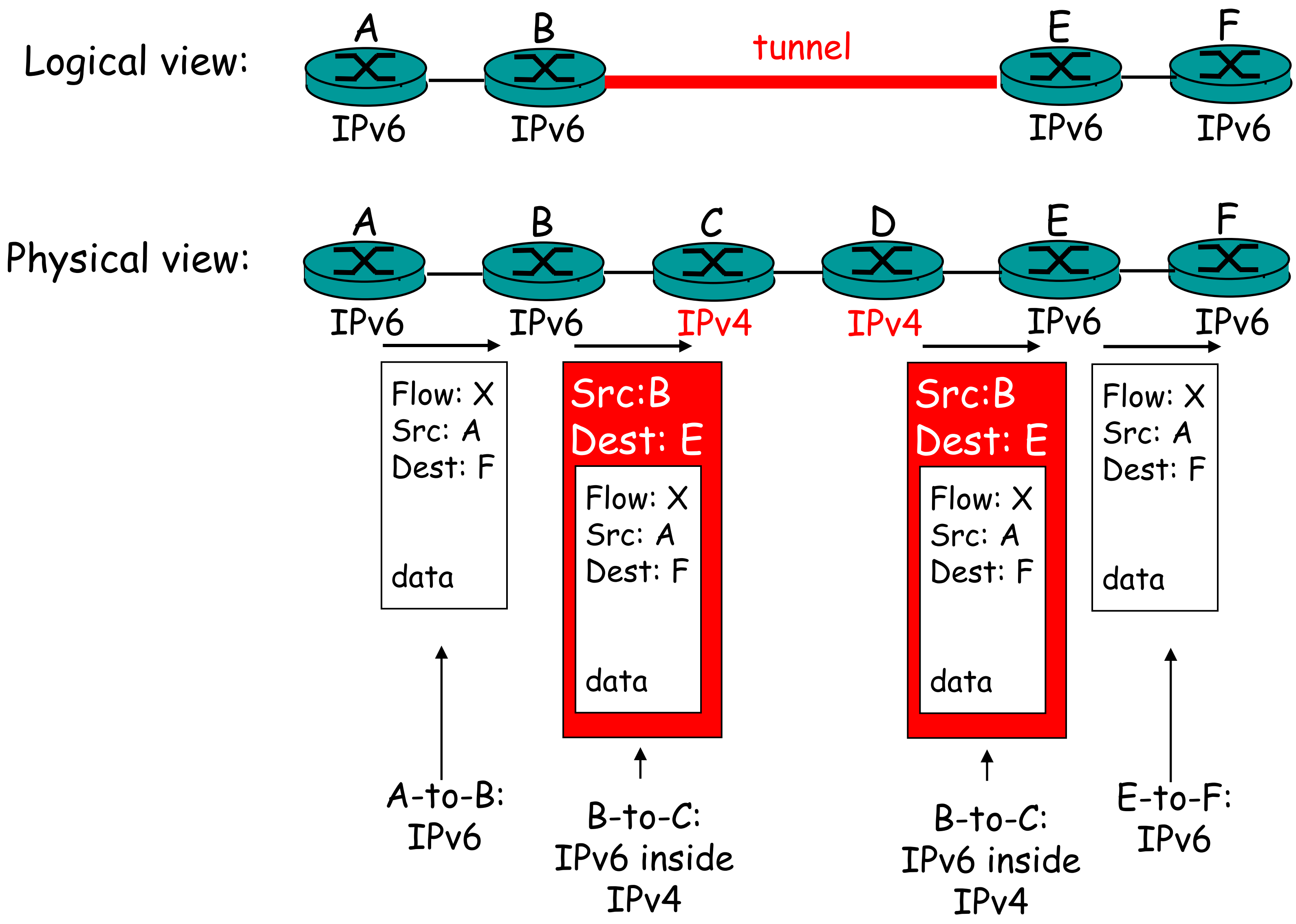
- *Checksum*: removed entirely to reduce processing time at each hop
- *Options:* allowed, but outside of header, indicated by “Next Header” field
- *ICMPv6:* new version of ICMP
  - » additional message types, e.g. “Packet Too Big”
  - » multicast group management functions

- Not all routers can be upgraded simultaneous
  - no “flag days”
  - How will the network operate with mixed IPv4 and IPv6 routers?
- *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

Tunneling (1/2)



Tunneling (2/2)



Agenda

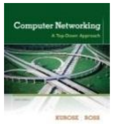
1. Session Overview
2. Networks Part 1
3. Summary and Conclusion

Summary

- Introduction
- Virtual circuit and datagram networks
- What’s inside a router
- IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6

Assignments & Readings

- Readings



  - Chapter 4
- Assignments #7

Next Session: Networks - Part II