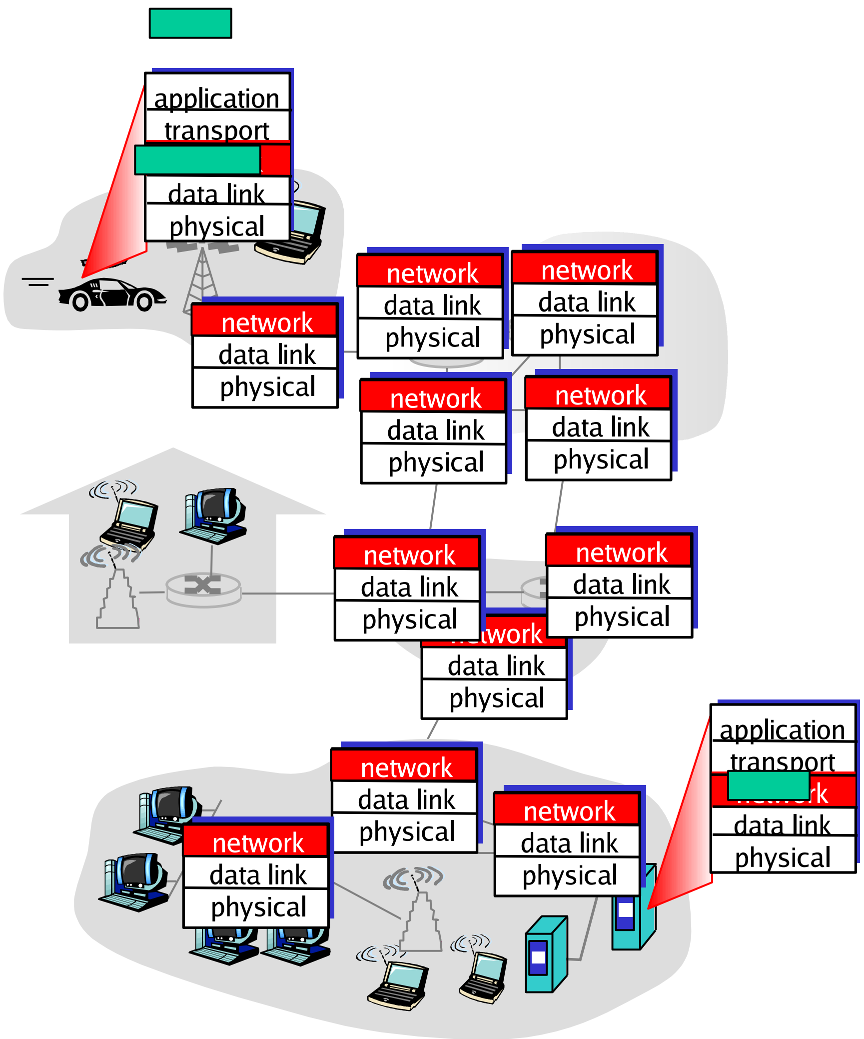
Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Network layer

- Delivers segments from sending to receiving host
- On sending side encapsulates segments into datagrams (or packets)
- On receiving side, delivers segments to Transport layer
- Network layer protocols implemented in *every* host, router
- Router examines header fields in all IP datagrams passing through it



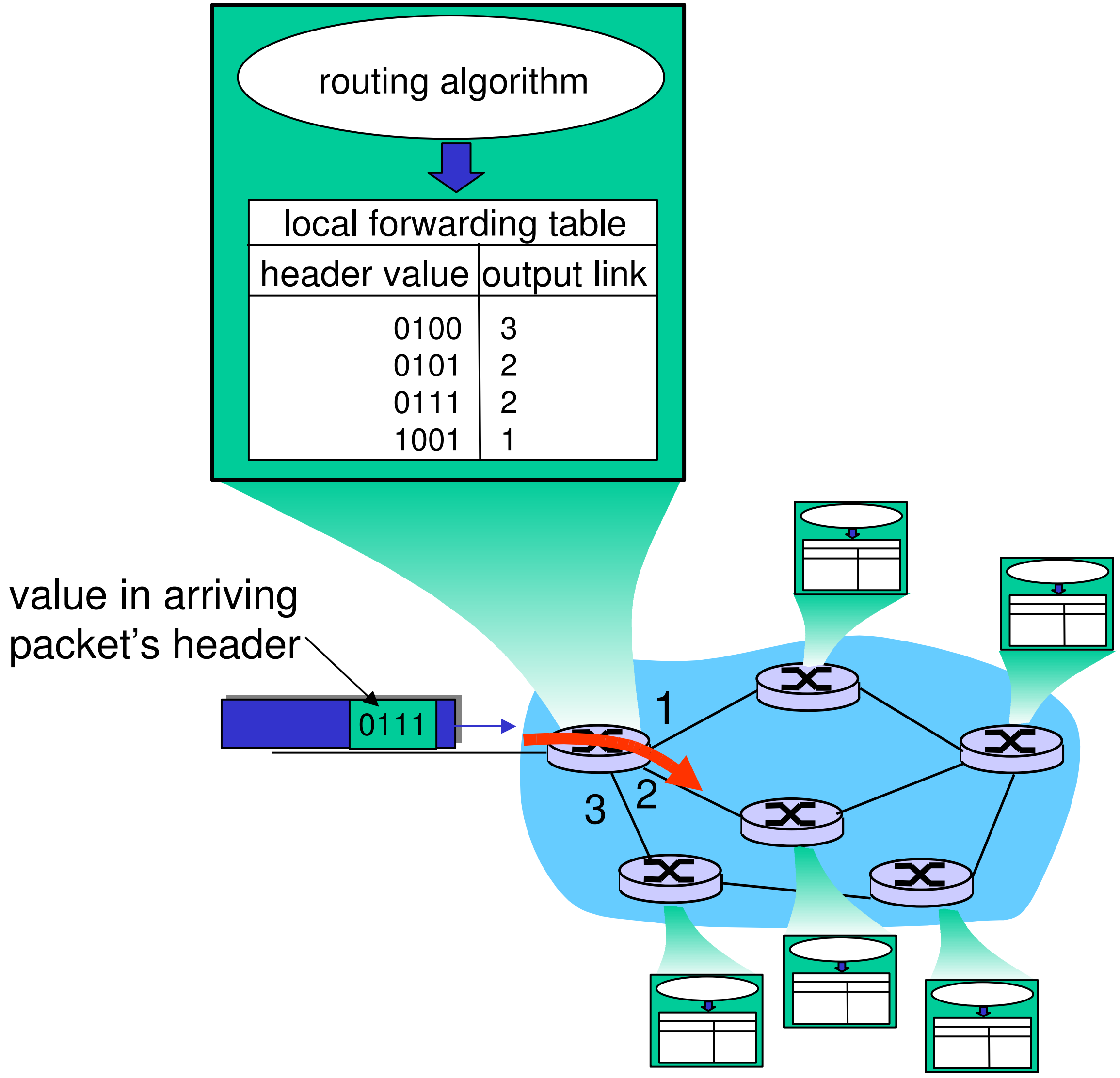
Two Key Network-Layer Functions

- *forwarding:* moving packets from a router's input to an appropriate router output
- *routing:* determine the path taken by packets from source to destination.

Analogy:

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

Interplay between routing and forwarding



Connection Setup

- 3rd important function in *some* network architectures:
  - ATM, frame relay, X.25
- Before datagrams can flow, two end hosts *and* intervening routers establish a 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

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Virtual Circuits

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

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

- Call Setup for (Connection) each call *before* data can flow
- Call Termination (Teardown) at the end of session
- Packets carry a 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)

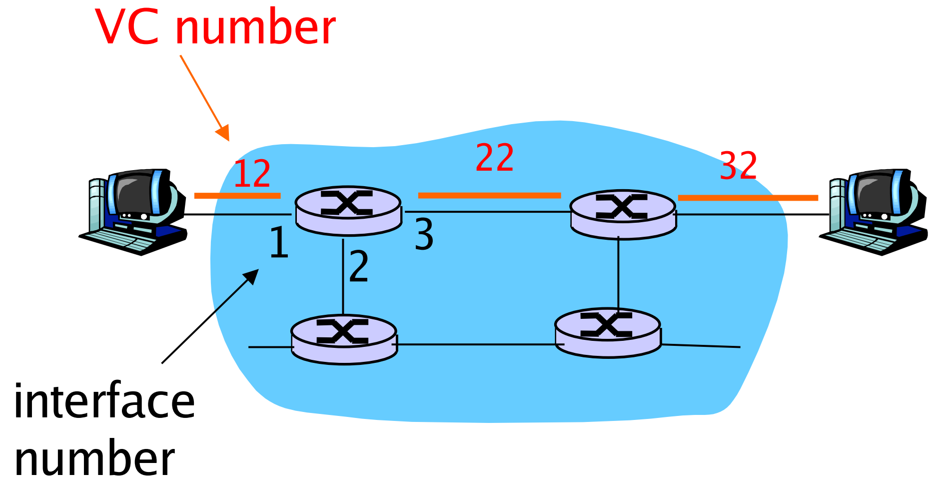
VC Implementation

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



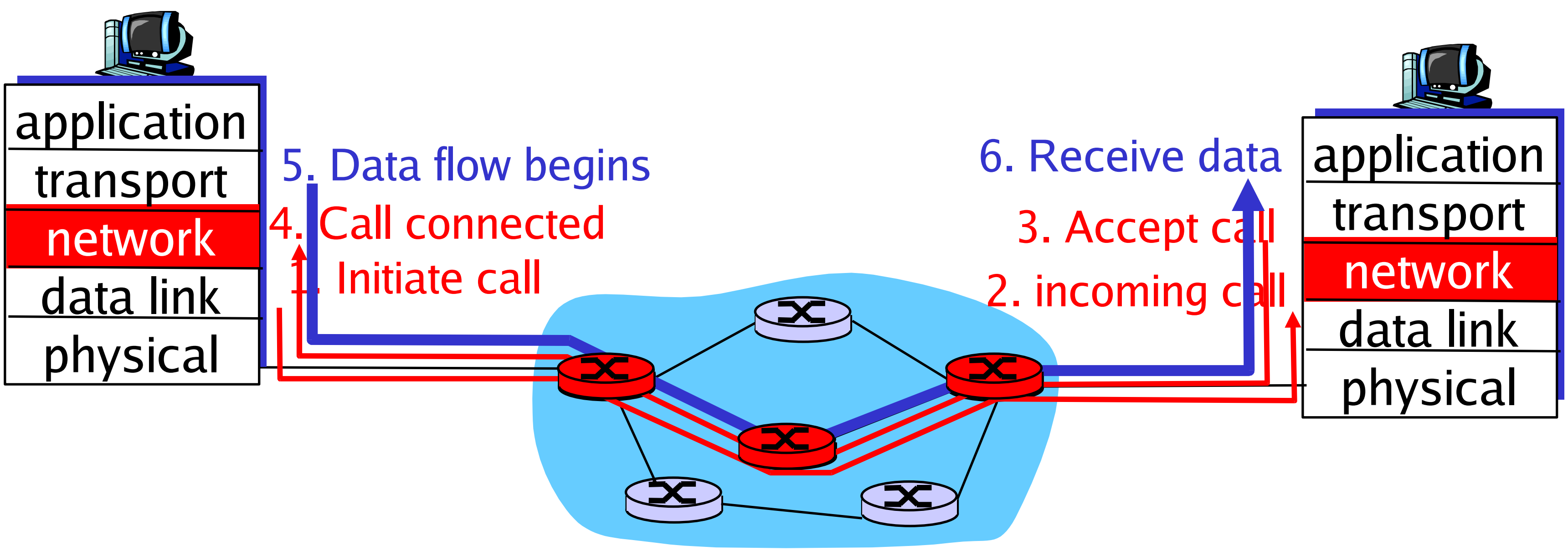
Forwarding table in northwest router:

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!

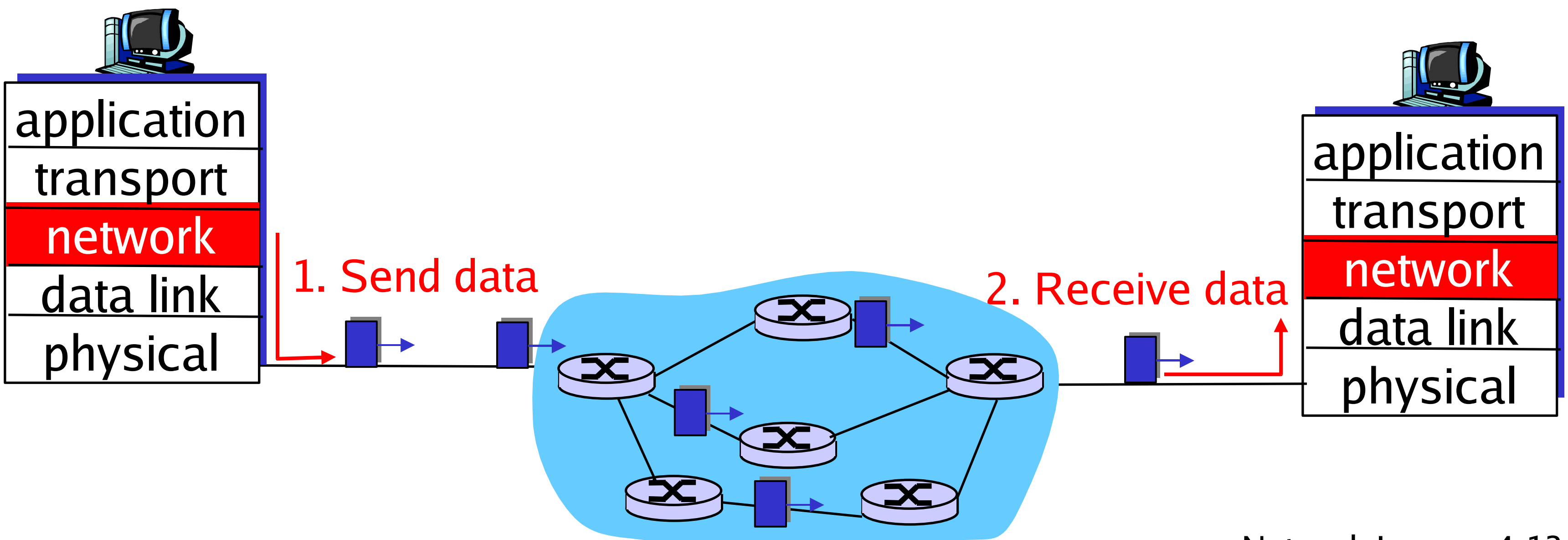
Virtual Circuits: Signaling Protocols

- 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?

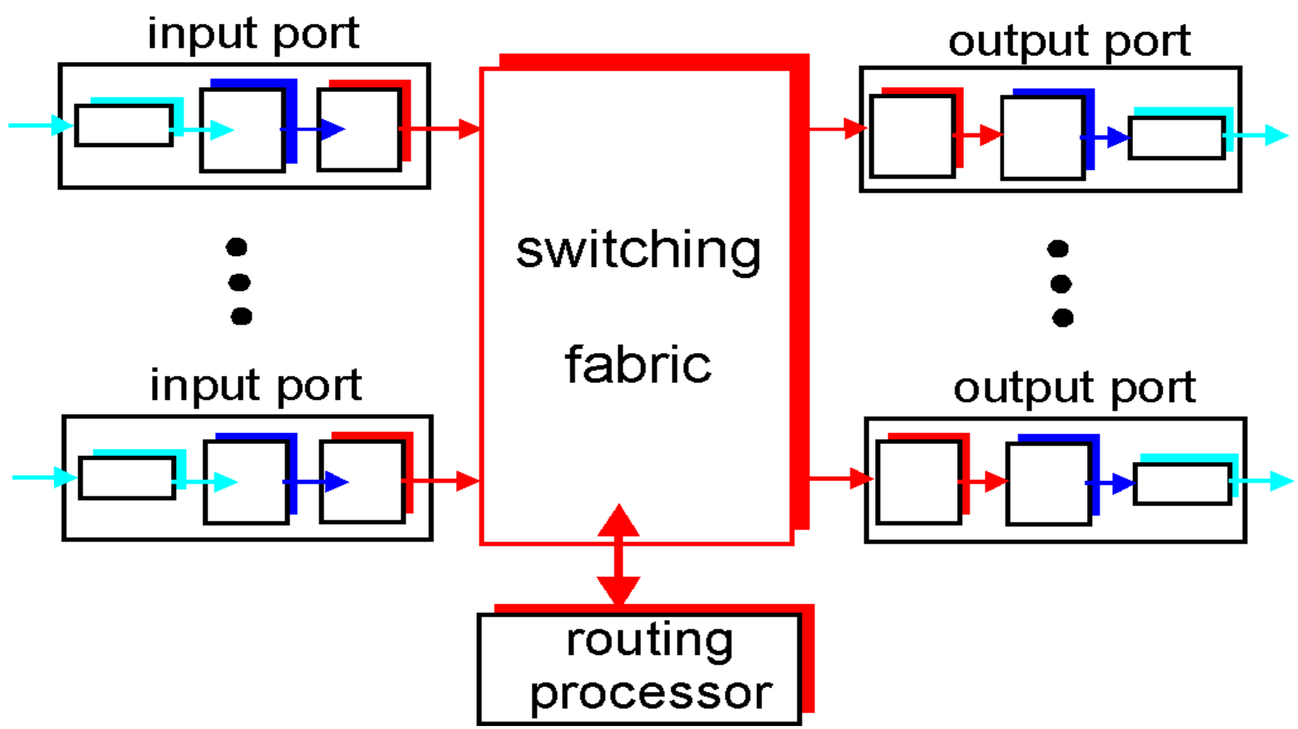
Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

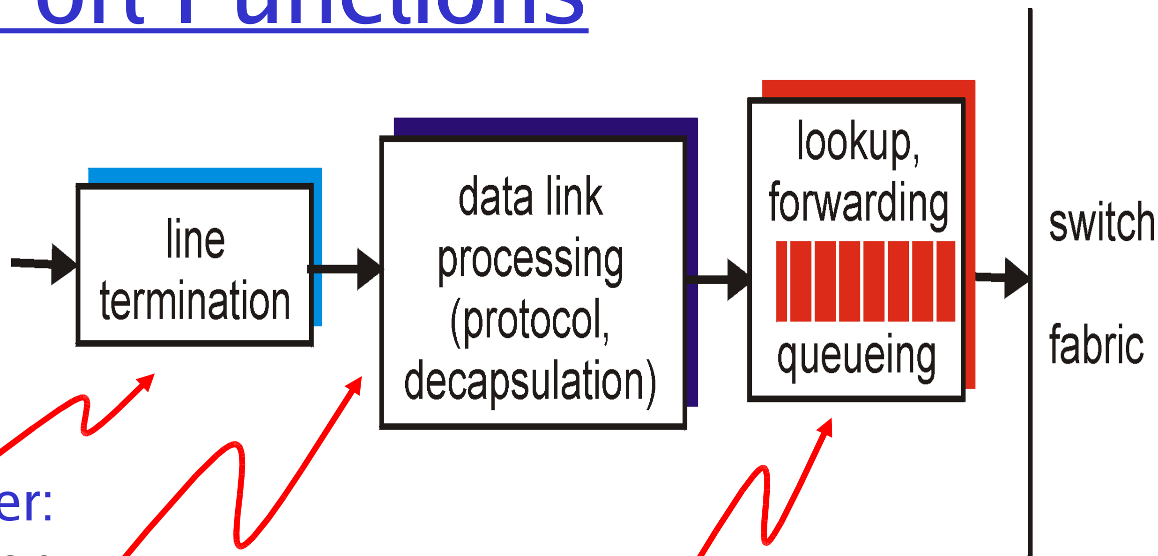
Router Architecture Overview

Two key router functions:

- implement routing algorithms/protocol (RIP, OSPF, BGP)
- *forward* datagrams from incoming to outgoing links



Input Port Functions



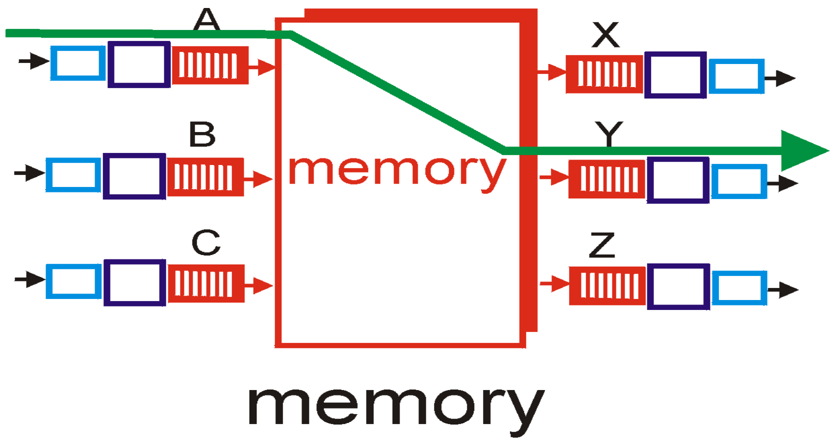
Physical layer:
bit-level reception

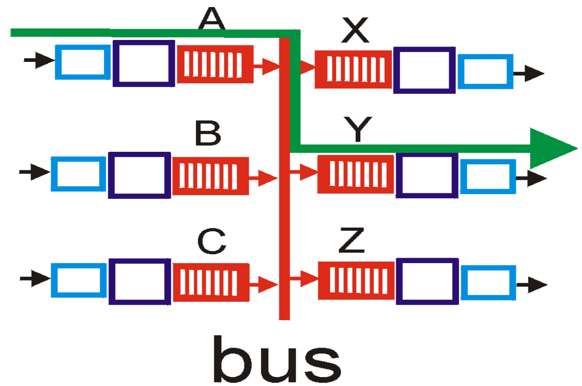
Data link layer:
e.g., Ethernet

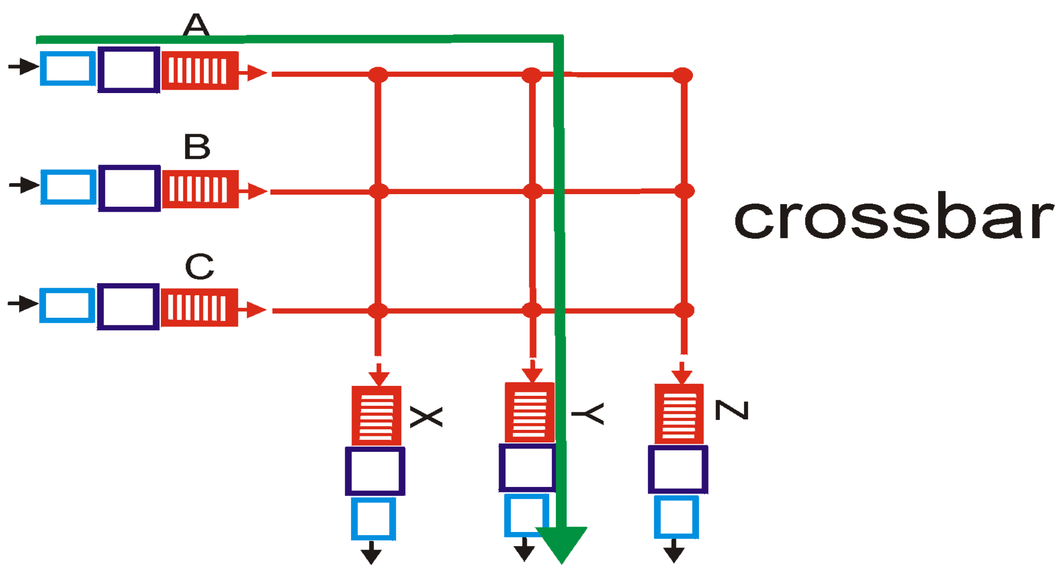
Decentralized switching:

- 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



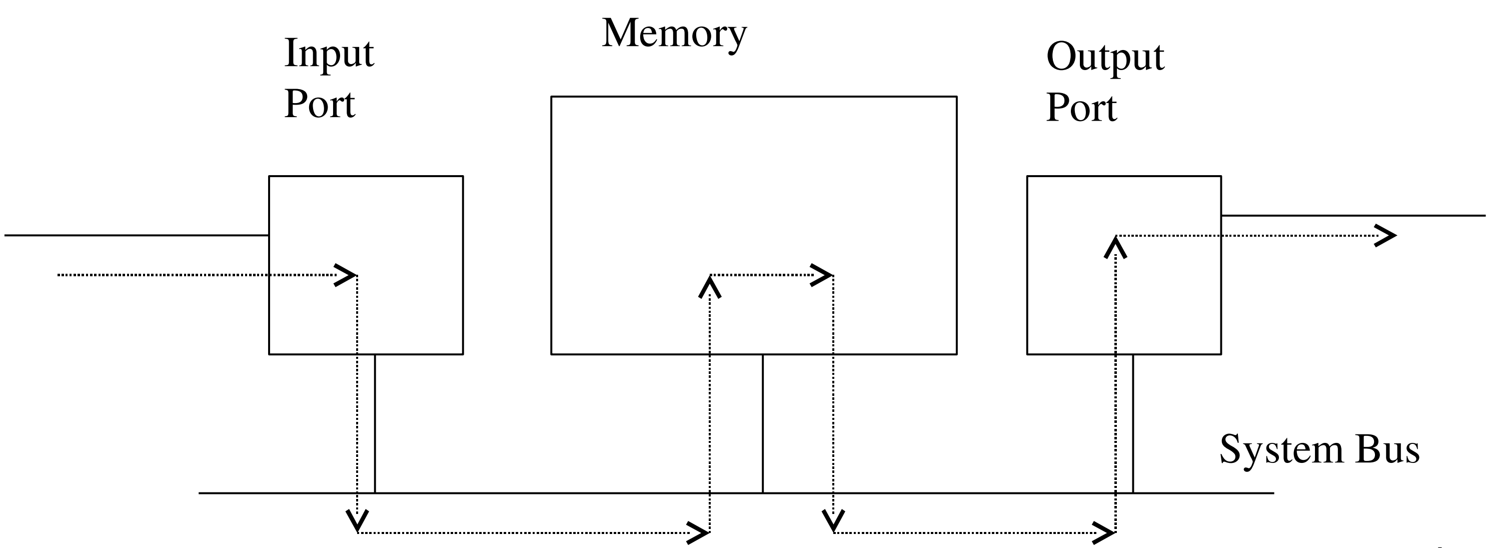




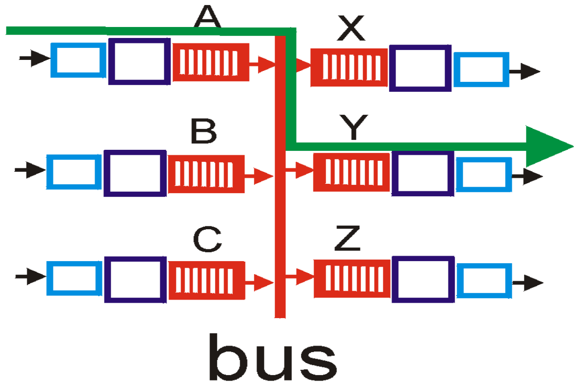
Switching Via Memory

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

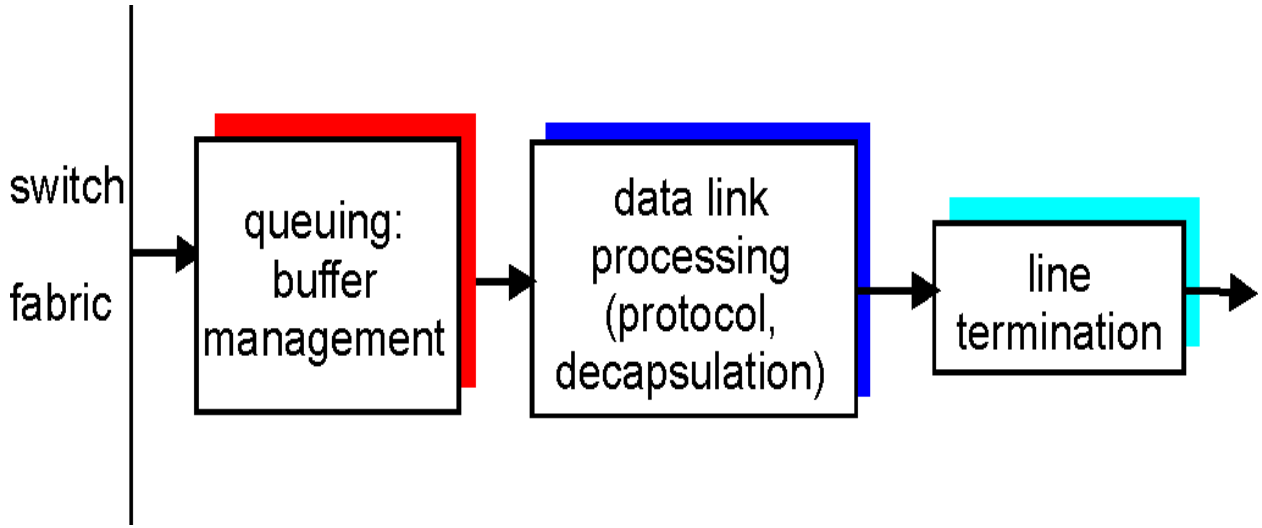


- datagram from input port memory to output port memory via a shared bus
- bus contention: switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers

Switching Via An Interconnection Network

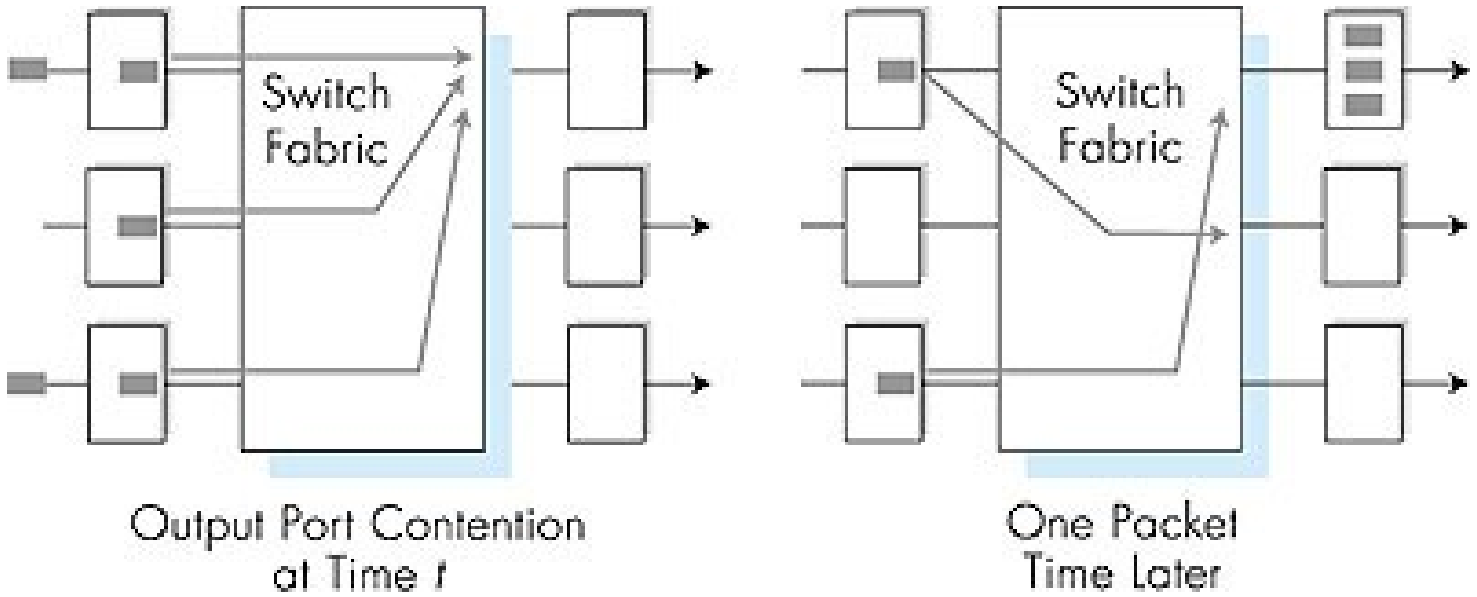
- overcome bus bandwidth limitations
- advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through the interconnection network

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



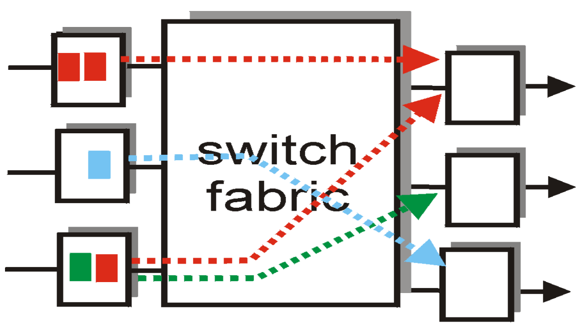
- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

How much buffering?

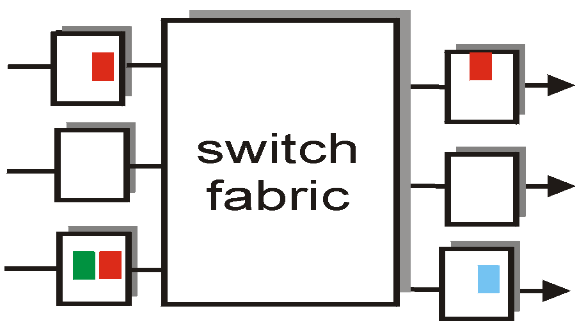
- RFC 3439 rule of thumb: average buffering equal to "typical" RTT (say 250 msec) times link capacity C
  - e.g., C = 10 Gps link: 2.5 Gbit buffer
- Recent recommendation: with *N* flows, buffering equal to $$\frac{RTT \cdot C}{\sqrt{N}}$$

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



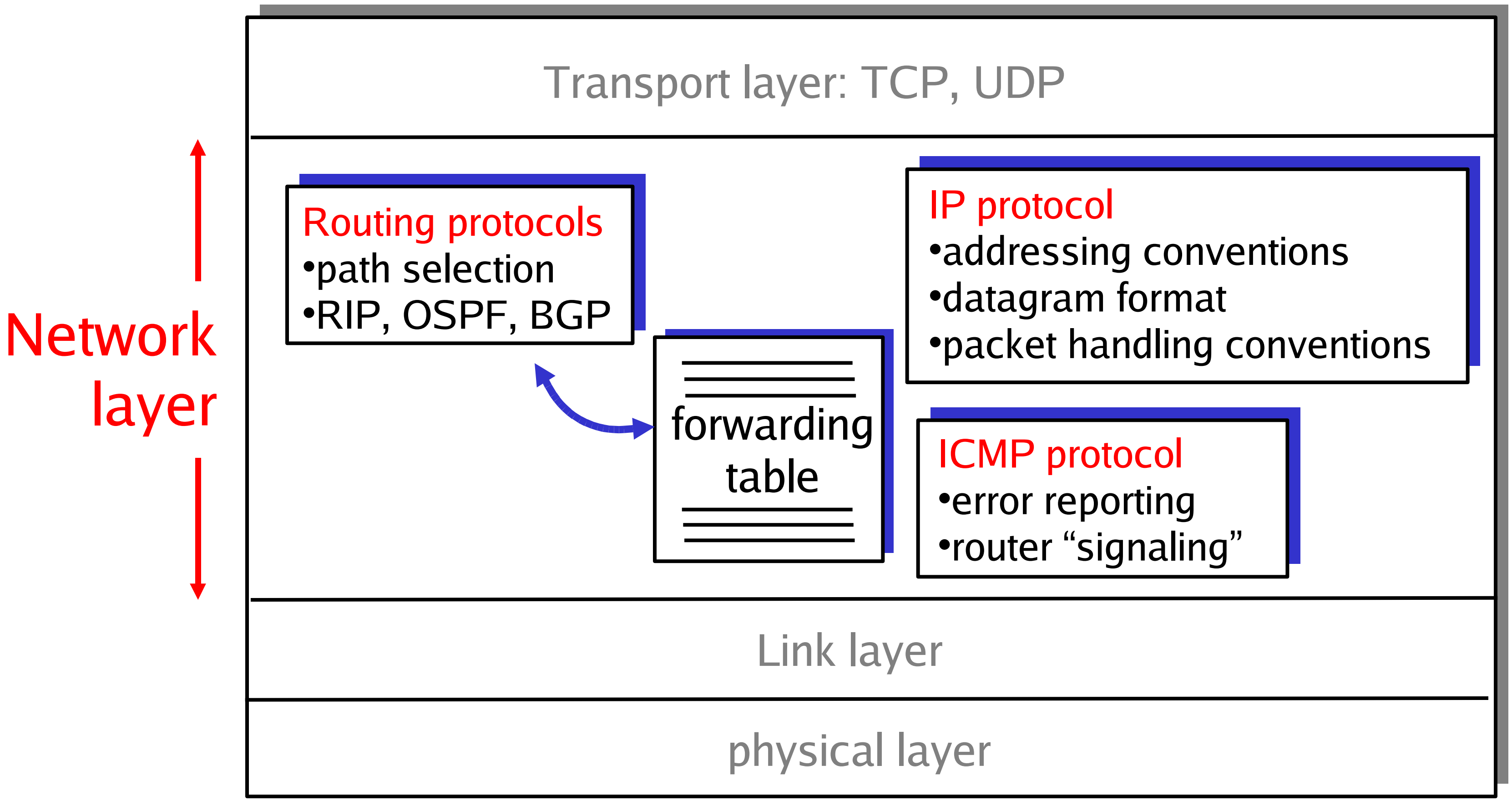
green packet experiences HOL blocking

Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

The Internet Network layer

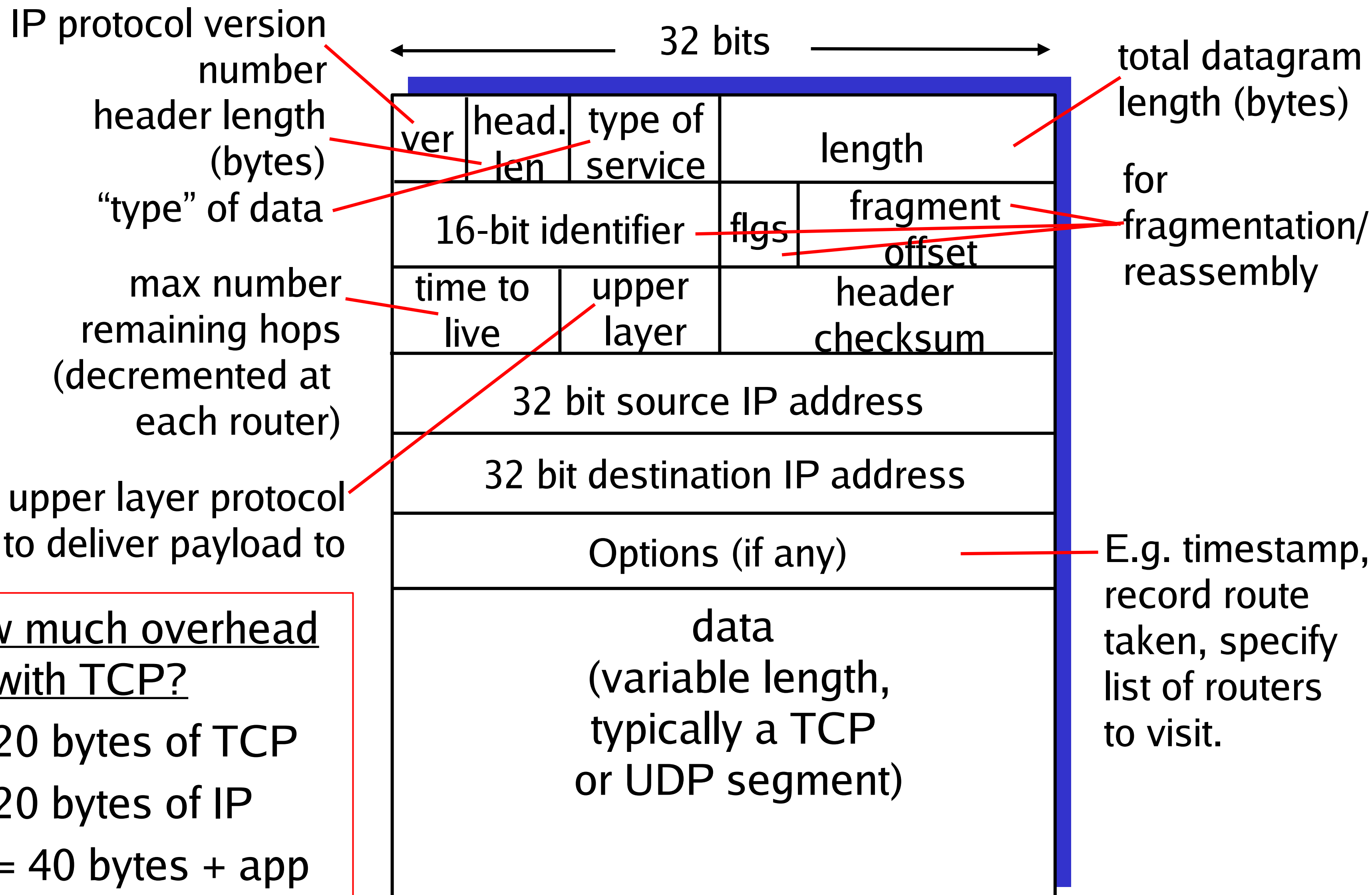
Host, router network layer functions:



Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

IP Datagram Format

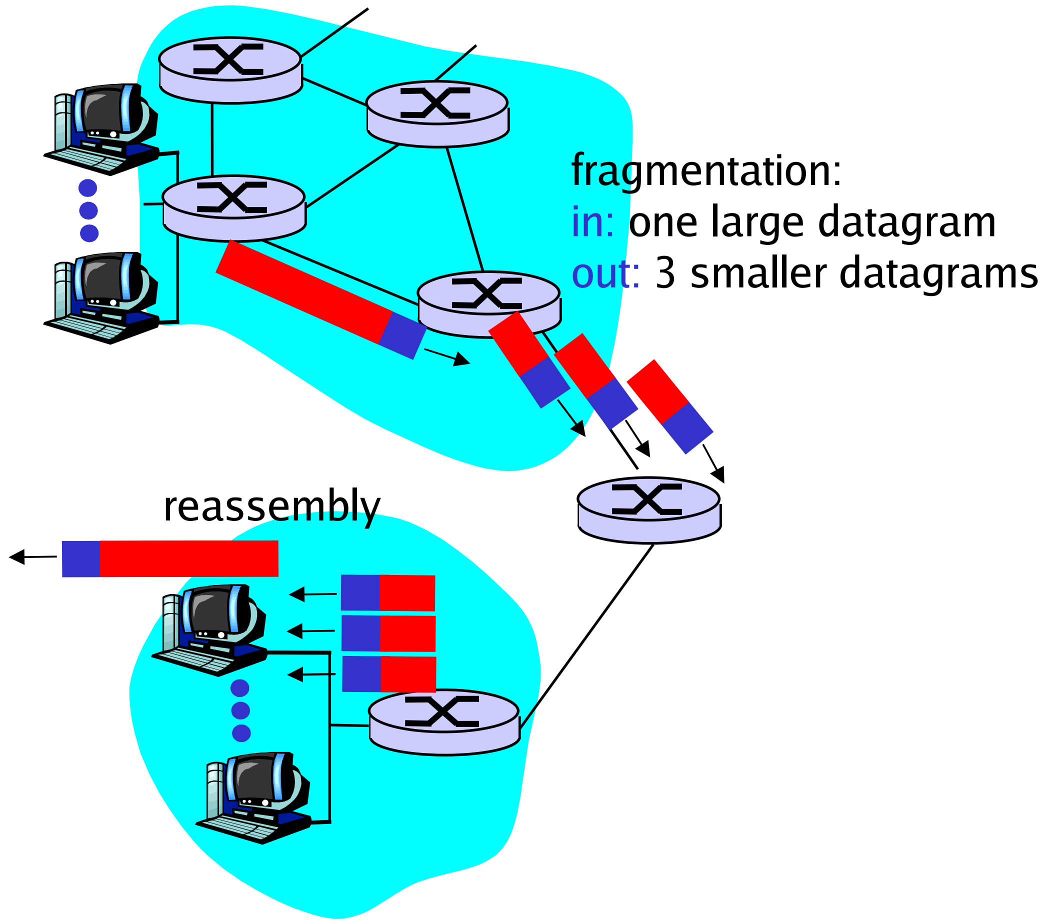


<u>how much overhead with TCP?</u>

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

IP Fragmentation & Reassembly

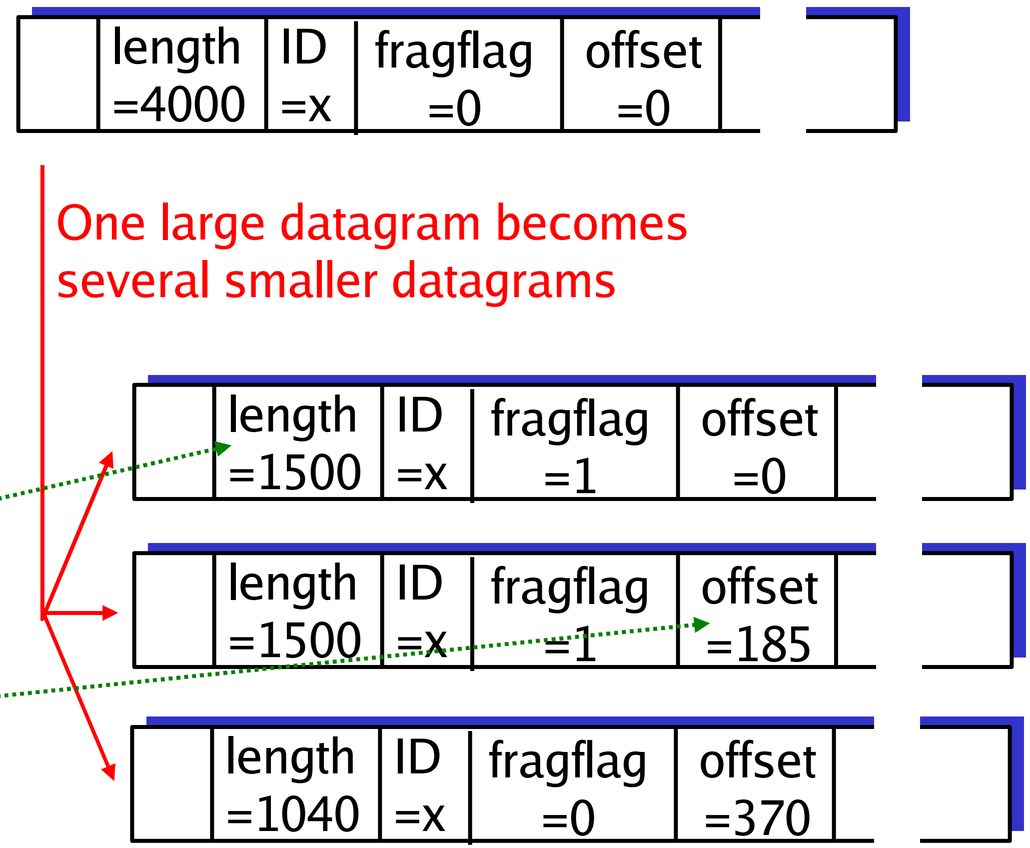
- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided (“fragmented”) within net
  - one datagram becomes several datagrams
  - “reassembled” only at final destination
  - IP header bits used to identify,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



IP Fragmentation and Reassembly

Example

- 4000 byte datagram
- MTU = 1500 bytes

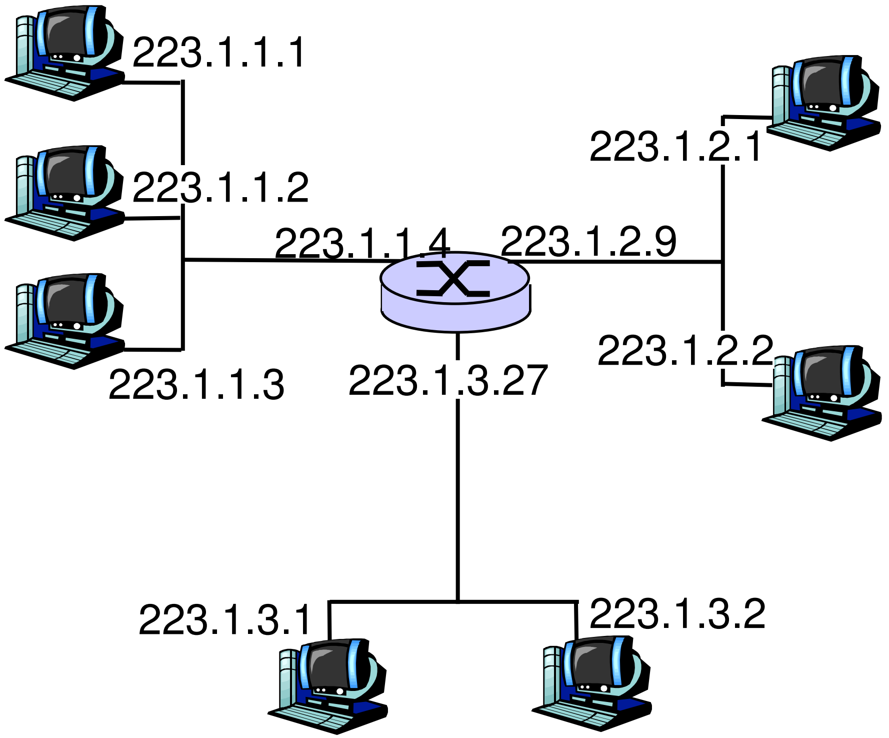


Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

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



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

Classful Addresses

Class A

	7 bits	**24 bits**
0	netid	hostid

- 126 networks with up to 16 million hosts

1.0.0.0 to 127.255.255.255

Class B

		14 bits	**16 bits**
1	0	netid	hostid

- 16,382 networks with up to 64,000 hosts

128.0.0.0 to 191.255.255.255

Class C

			22 bits	**8 bits**
1	1	0	netid	hostid

- 2 million networks with up to 254 hosts

192.0.0.0 to 223.255.255.255

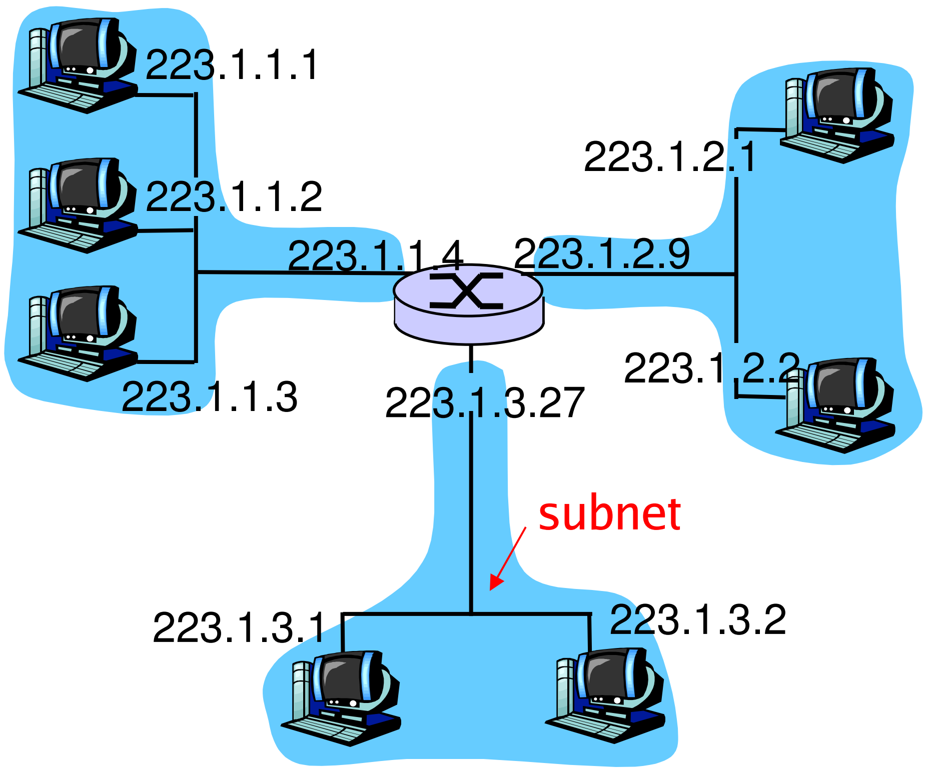
IP Address Classes

IP Address Class	First Octet Minimum	First Octet Maximum	Leading Bit Pattern	Number of Networks	Number of Hosts
Class A	1	126	0	126	16,777,214
Class B	128	191	10	16,384	65,534
Class C	192	223	110	2,097,152	254
Class D	224	239	1110		
Class E	240	247	11110		

- Class D addresses are reserved for multicast groups.
- Class E addresses are an experimental class of IP addresses.

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

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

Subnetting

- The Classic rules wastes large numbers of addresses, especially class A and B addresses
- **Classless Inter-Domain Routing** (**CIDR**) uses available IP addresses more efficiently.
- Default subnet mask
  - Class A = 255.0.0.0
  - Class B = 255.255.0.0
  - Class C = 255.255.255.0

Eg. Class B default subnet mask:

11111111 1111111 0000000 0000000

1s portion = network address

0s portion = host address

Assigning IP addresses:

Q: How does *host* get IP address?

- hard-coded by system admin in a file
  - Wintel: 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

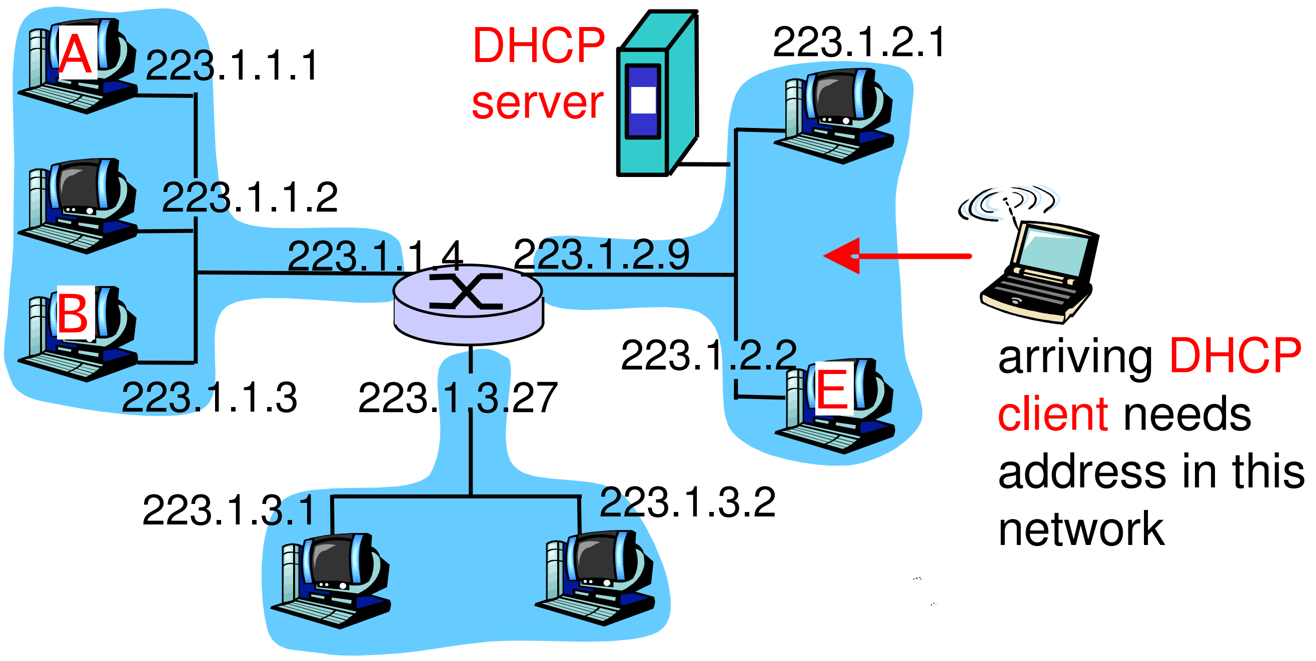
Goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- 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)

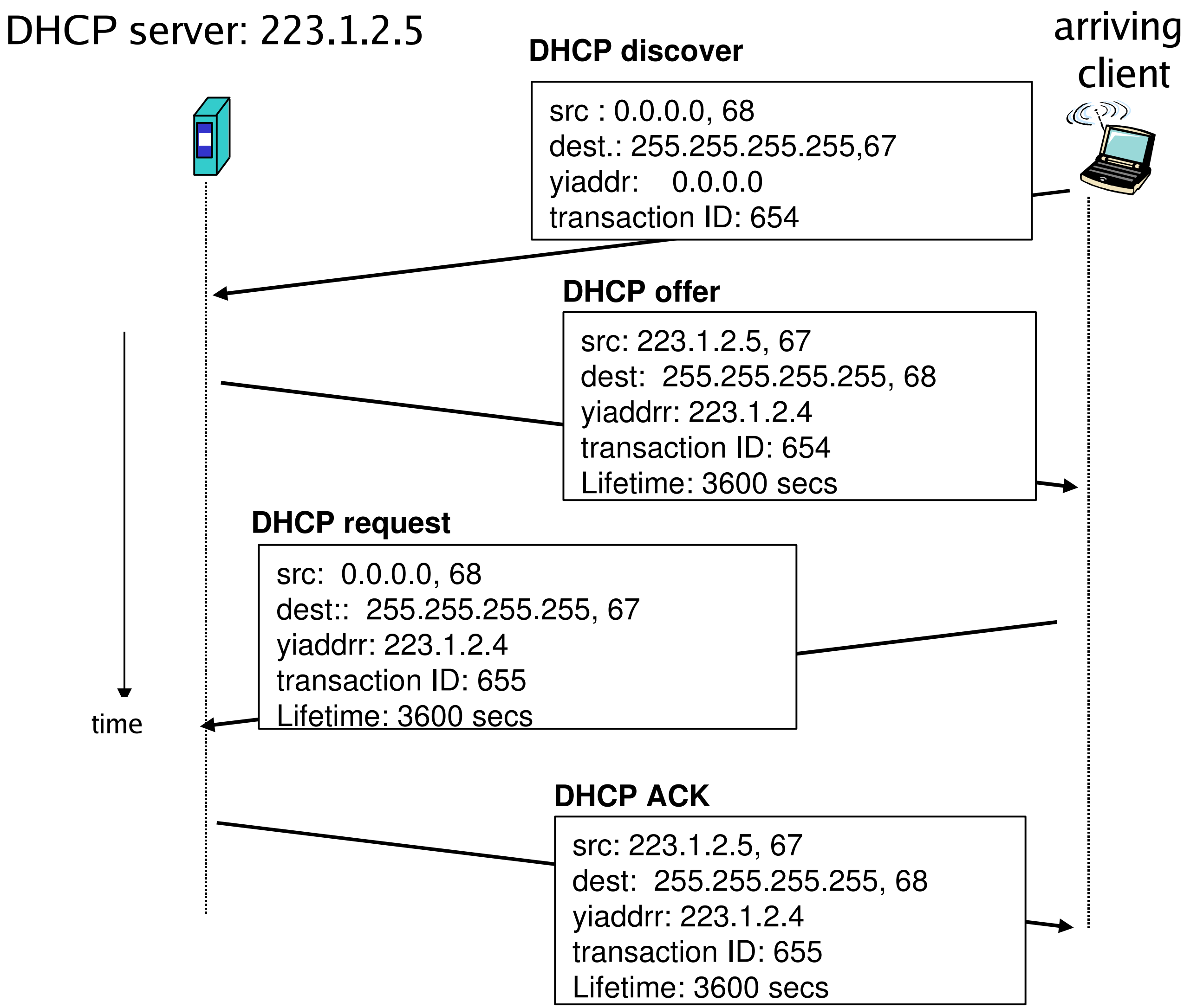
DHCP overview:

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

DHCP Client-Server scenario



DHCP Client-Server Scenario

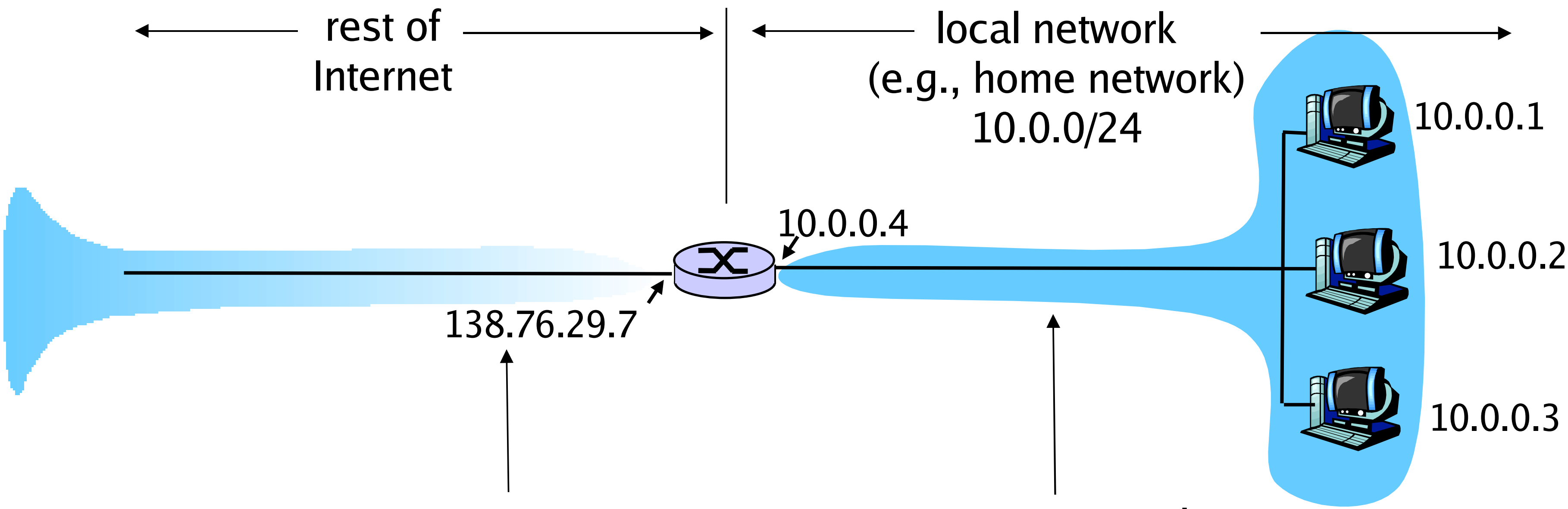


Private IP Address

Class	Address Range	Default Mask
A	10.xxx.xxx.xxx	255.0.0.0
B	172.016.xxx.xxx	255.255.0.0
C	192.168.xxx.xxx	255.255.255.0

- Address ranges:
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16
- Routers on the Internet will not forward packets coming from these addresses.

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

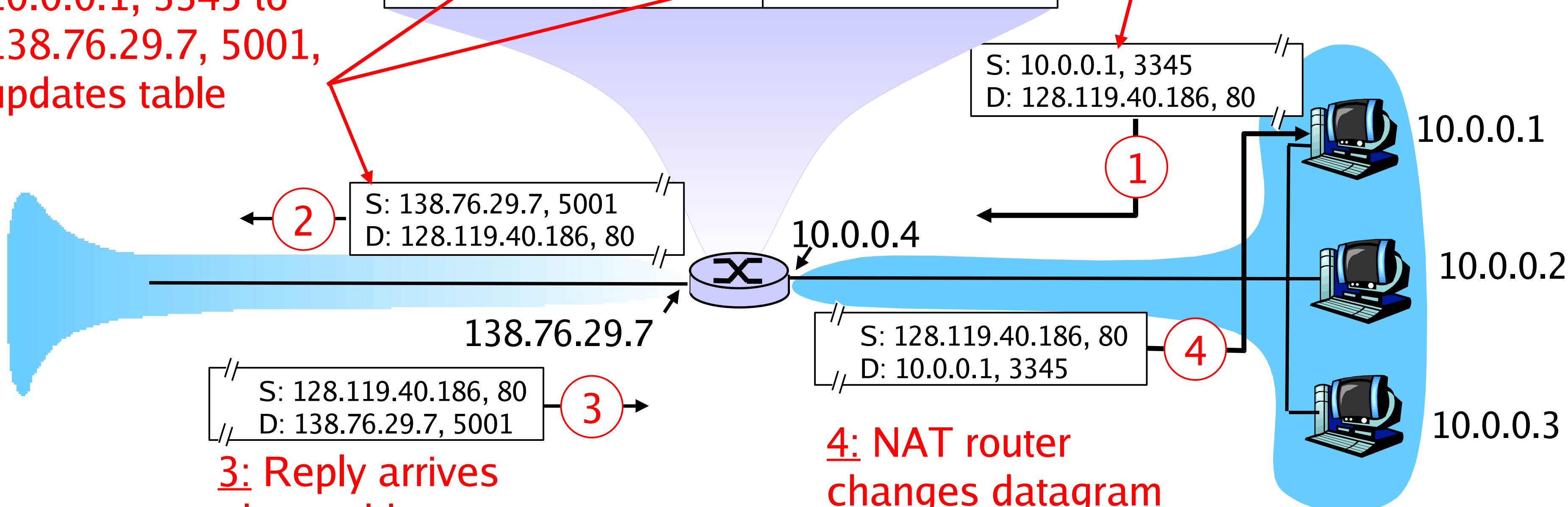
NAT: Network Address Translation

2: NAT router changes datagram source addr from 10.0.0.1, 3345 to 138.76.29.7, 5001, updates table

NAT translation table

WAN side addr	LAN side addr
138.76.29.7, 5001	10.0.0.1, 3345
…	…

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80



3: Reply arrives dest. address: 138.76.29.7, 5001

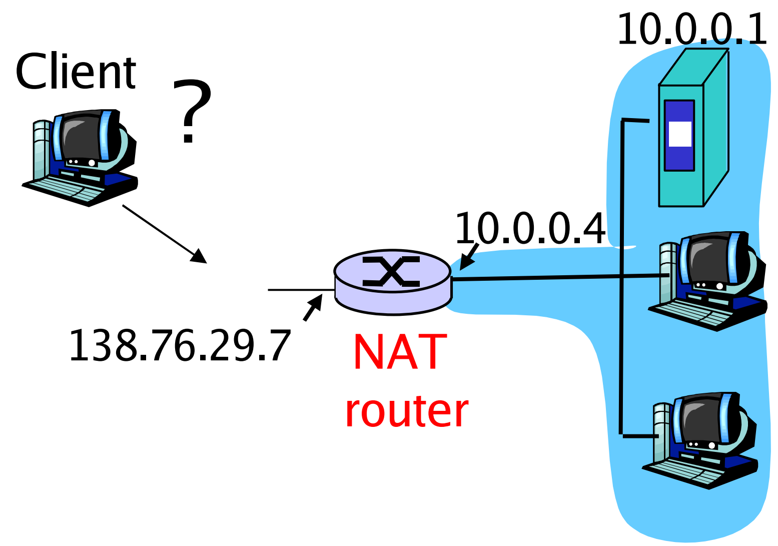
4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

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

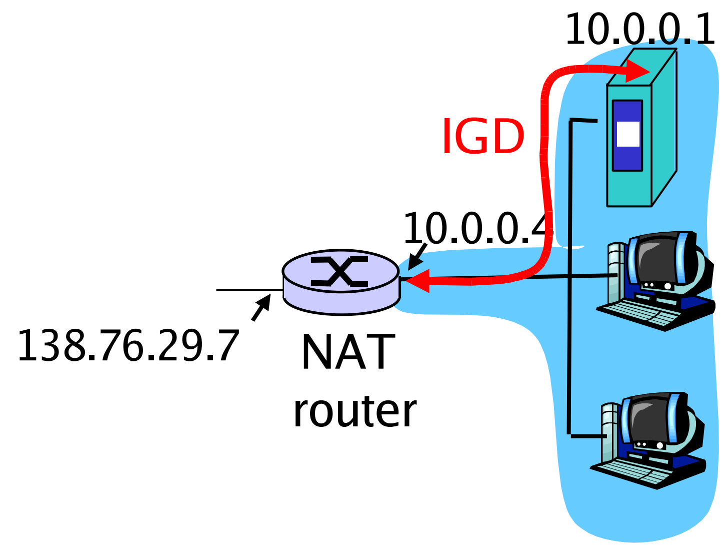
- client want to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can’t use it as destination addr)
  - only one externally visible NATted address: 138.76.29.7
- solution 1: statically configure NAT to forward incoming connection requests at given port to server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



NAT Traversal Problem

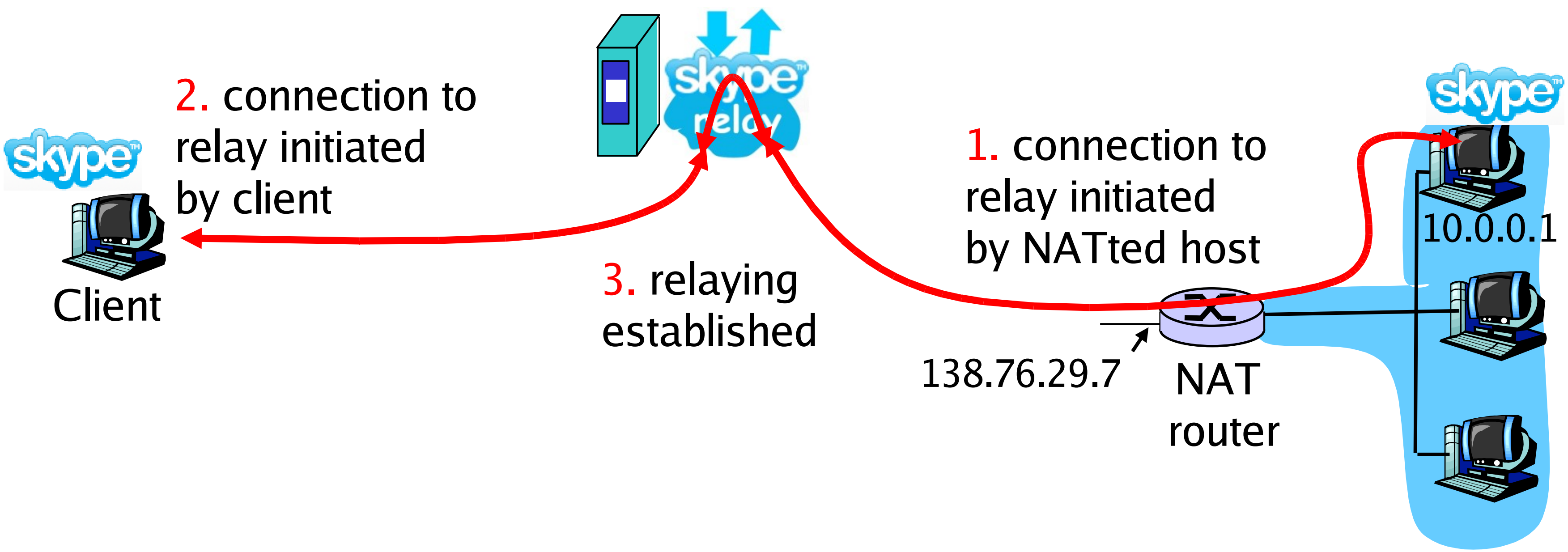
- solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATted host to:
  - learn public IP address (138.76.29.7)
  - enumerate existing port mappings
  - add/remove port mappings (with lease times)

i.e., automate static NAT port map configuration



NAT Traversal Problem

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



Chapter 4: Network Layer

- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

Internet Control Message Protocol (ICMP)

- Error reporting and diagnostic utility.
- Required part of any IP implementation.
- ICMP messages are used by routers, intermediary devices, or hosts.
- ICMP message: type, code plus first 8 bytes of IP datagram causing error

Type (8 bit)	Code (8 bit)	Checksum (16 bit)
[Unused] (32 bit)		
Internet Header + 64 bits of Original Data Datagram (32 bit)		

ICMP Header

Type

- Echo Request & Echo Reply
  - Used by "**ping**"
- Source Quench
  - Sent when the destination is unable to process traffic as fast as the source is sending it.
- Redirect Message
  - Generated by an intermediary device when a route being requested can be reached either locally or through a better path

Type	Description
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect Message
8	Echo Request
11	Time Exceeded
12	Bad IP Header Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

Destination Unreachable

Type 3 Code Value	Description
0	Network Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation needed and DF (Don't Fragment) set
5	Source route failed
6	Destination Network unknown
7	Destination Host unknown
8	Source Host isolated
9	Communication with Destination Network Administratively Prohibited
10	Communication with Destination Host Administratively Prohibited
11	Network Unreachable for Type Of Service
12	Host Unreachable for Type Of Service
13	Communication Administratively Prohibited by Filtering
14	Host Precedence Violation
15	Precedence Cutoff in Effect

ICMP Header (cont.)

Type

- Time Exceeded
  - Generated when a router or host **discards a packet due to a time-out.**
- Parameter Problem
  - Generated when an intermediary device or host **discards a datagram due to inability to process**.

  Eg. Corrupt header, missing options

Type	Description
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect Message
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

ICMP Header (cont.)

Type

- Timestamp Request & Timestamp Reply
  - Rudimentary method for **synchronizing the time** maintained on different devices.
- Information Request & Information Reply
  - **Obsolete** and no longer used.

Type	Description
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect Message
8	Echo Request
11	Time Exceeded
12	Parameter Problem
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

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

Stopping criterion

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

Chapter 4: Network Layer

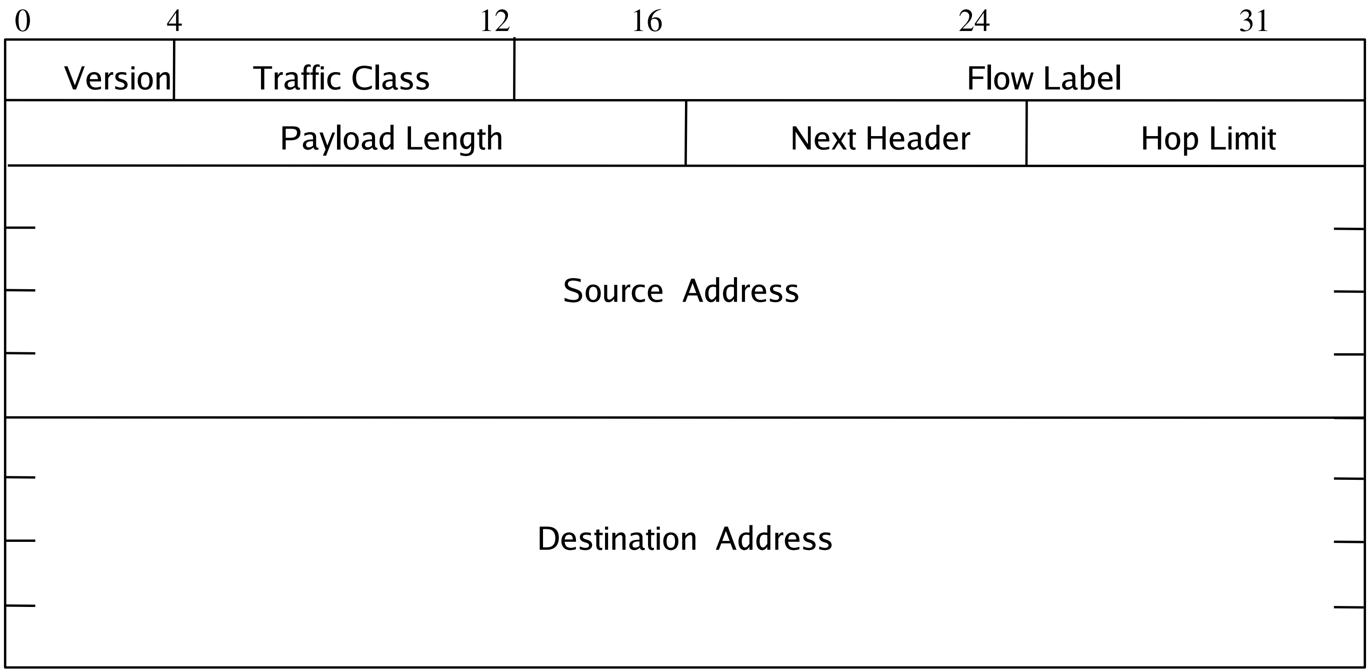
- 4. 1 Introduction
- 4.2 Virtual circuit and datagram networks
- 4.3 What's inside a router
- 4.4 IP: Internet Protocol
  - Datagram format
  - IPv4 addressing
  - ICMP
  - IPv6
- 4.5 Routing algorithms
  - Link state
  - Distance Vector
  - Hierarchical routing
- 4.6 Routing in the Internet
  - RIP
  - OSPF
  - BGP
- 4.7 Broadcast and multicast routing

IPv6

- Initial motivation: 32-bit address space soon to be completely allocated.
  - Length of the address field is extended from 32 bits to 128 bits.
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS
  - Options in Ipv6 are specified in optional **Extension Headers**.

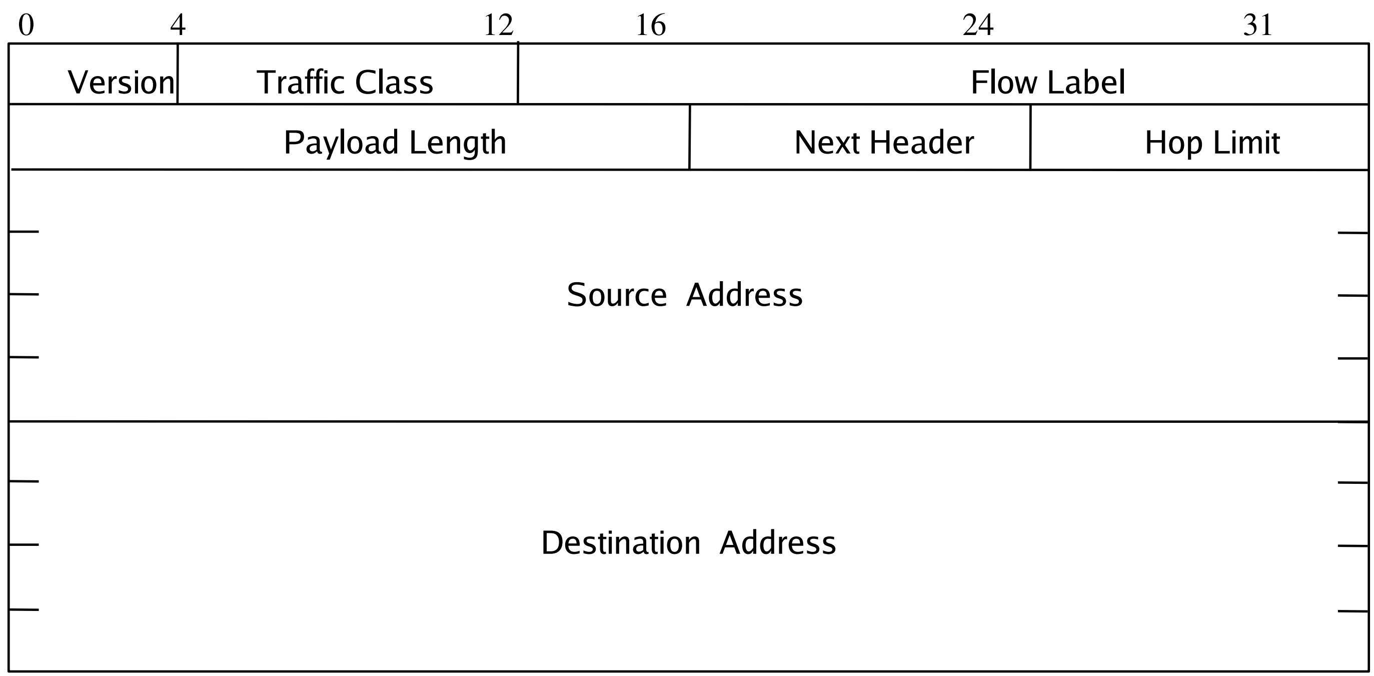
  IPv6 datagram format:
  - fixed-length 40 byte header
  - Routers will not fragment packets; fragmentation must be done at the source.

IPv6 Header Format



- Version field same size, same location
- Traffic class to support differentiated services
- Flow: sequence of packets from particular source to particular destination for which source requires special handling

IPv6 Header Format



- Payload length: length of data excluding header, up to 65535 B
- Next header: type of extension header that follows basic header
- Hop limit: # hops packet can travel before being dropped by a router

IPv6 Addressing

- Address Categories
  - Unicast: single network interface
  - Multicast: group of network interfaces, typically at different locations. Packet sent to all.
  - Anycast: group of network interfaces. Packet sent to only one interface in group, e.g. nearest.
- Hexadecimal notation
  - Groups of 16 bits represented by 4 hex digits
  - Separated by colons
    - 4BF5:AA12:0216:FEBC:BA5F:039A:BE9A:2176
  - Shortened forms:
    - 4BF5:0000:0000:0000:BA5F:039A:000A:2176
    - To 4BF5:0:0:0:BA5F:39A:A:2176
    - To 4BF5::BA5F:39A:A:2176
  - Mixed notation:
    - ::FFFF:128.155.12.198

Special Purpose Addresses

	n bits	m bits	o bits	p bits	(125-m-n-o-p) bits
010	Registry ID	Provider ID	Subscriber ID	Subnet ID	Interface ID

- *Provider-based Addresses*: 010 prefix
  - Assigned by providers to their customers
  - Hierarchical structure promotes aggregation
    - Registry ID: ARIN, RIPE, APNIC
    - ISP
    - Subscriber ID: subnet ID & interface ID
- *Local Addresses*: do not connect to global Internet
  - Link-local: for single link
  - Site-local: for single site
  - Designed to facilitate transition to connection to Internet

Special Purpose Addresses

- *Unspecified Address*: 0::0
  - Used by source station to learn own address
- *Loopback Address*: ::1
- *IPv4-compatible addresses*: 96 0's + IPv4
  - For tunneling by IPv6 routers connected to IPv4 networks
  - ::135.150.10.247
- *IP-mapped addresses*: 80 0's + 16 1's + IPv4
  - Denote IPv4 hosts & routers that do not support IPv6

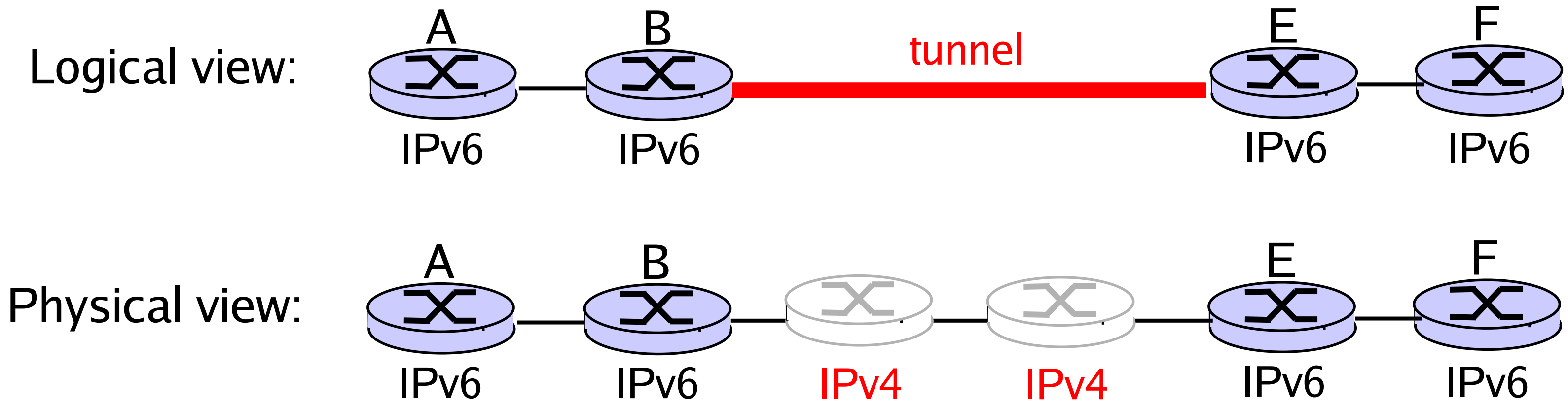
Other IPv6 Features

- **Flexible support for options:** more efficient and flexible options encoded in optional *extension headers*
- **Flow label capability:** “flow label” to identify a packet flow that requires a certain QoS
- **Security:** built-in authentication and confidentiality
- **Large packets:** supports payloads that are longer than 64 K bytes, called *jumbo* payloads.
- **Fragmentation at source only:** source should check the minimum MTU along the path
- **No checksum field:** removed to reduce packet processing time in a router

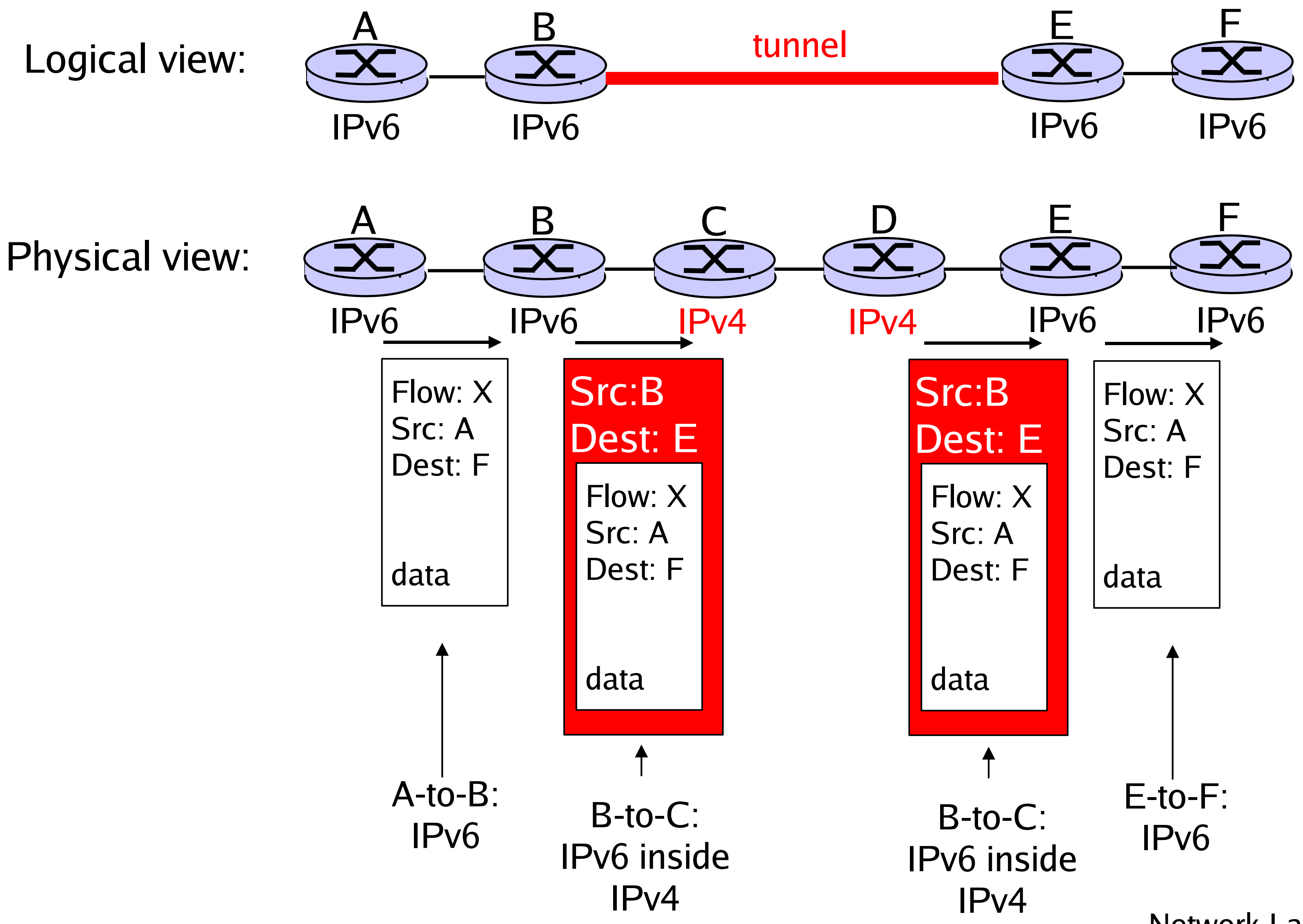
Migration from IPv4 to IPv6

- Gradual transition from IPv4 to IPv6
- Dual IP stacks: routers run IPv4 & IPv6
  - Type field used to direct packet to IP version
- IPv6 islands can tunnel across IPv4 networks
  - IPv6 carried as payload in IPv4 datagram among IPv4 routers
  - Tunnel endpoint at source host, intermediate router, or destination host

Tunneling


Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6

Tunneling



Logical view:
A
B
tunnel
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