The Network Layer

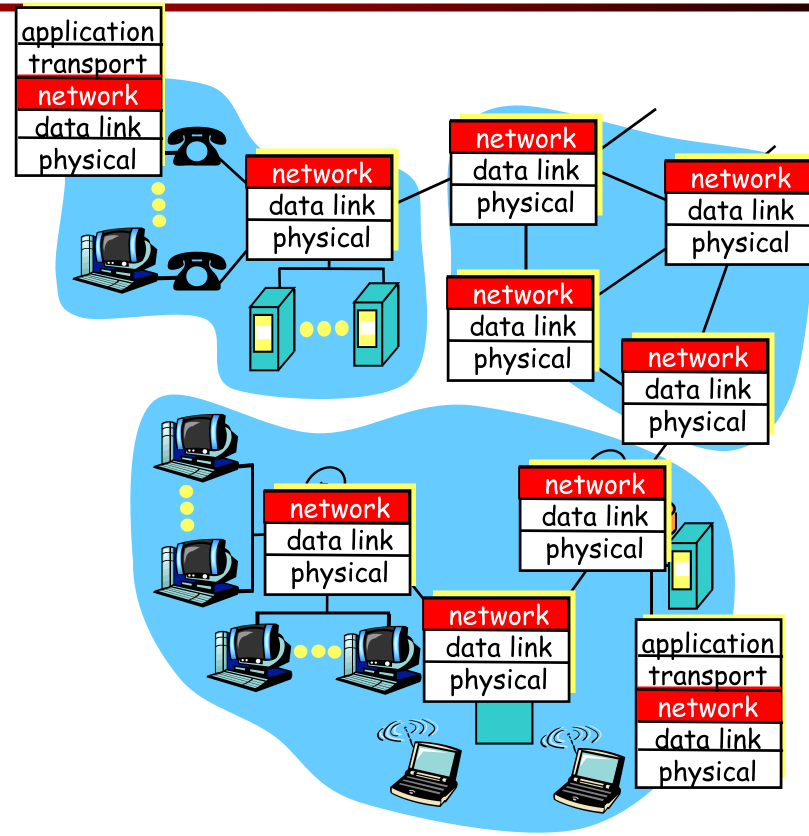
Slides were original prepared by Dr. Tatsuya Susa

Contents

Note that everything in this set of slides is about IPv4

Network layer

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



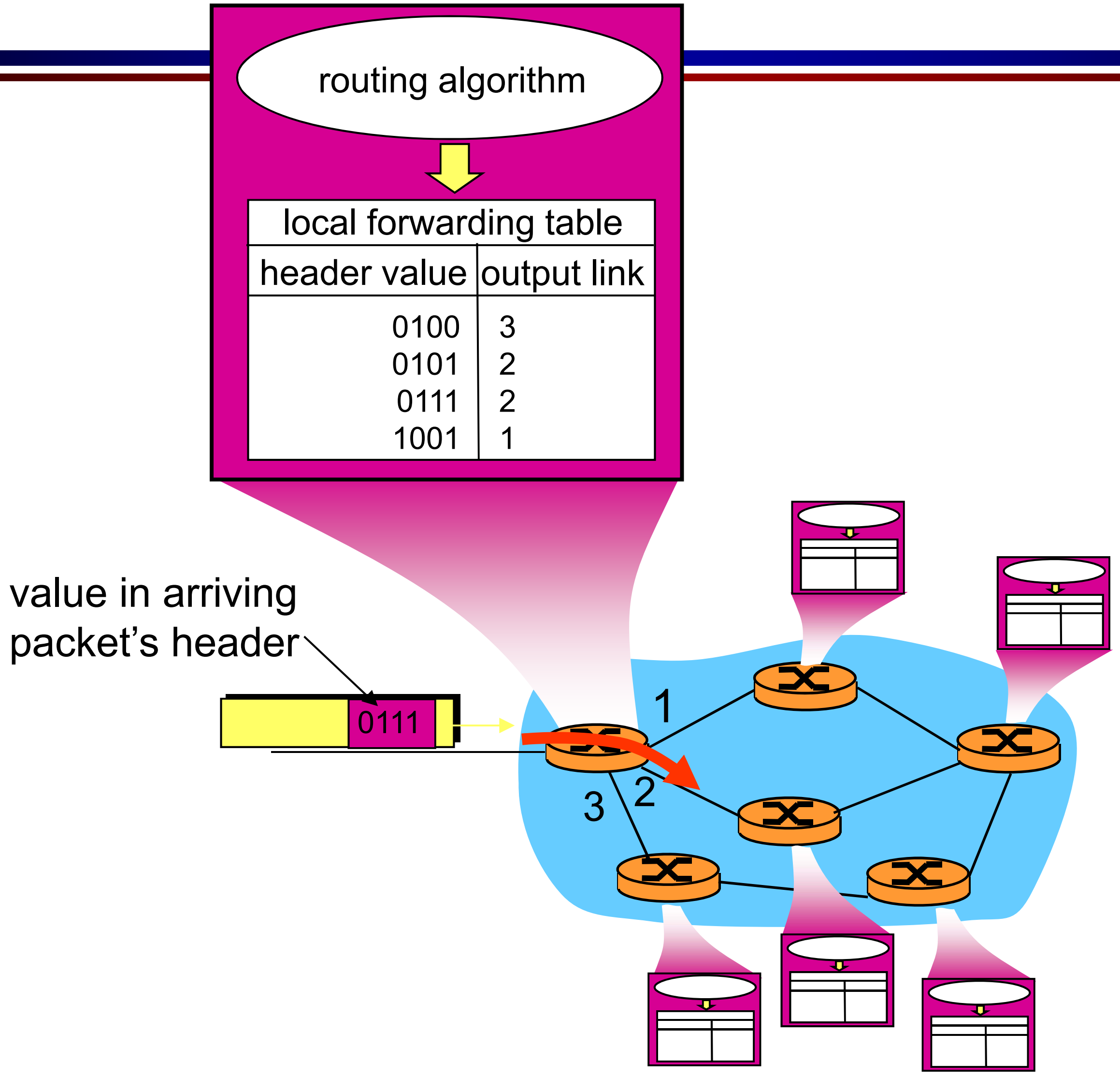
Two Key Network-Layer Functions

- *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



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

Contents

Note that everything in this set of slides is about IPv4

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

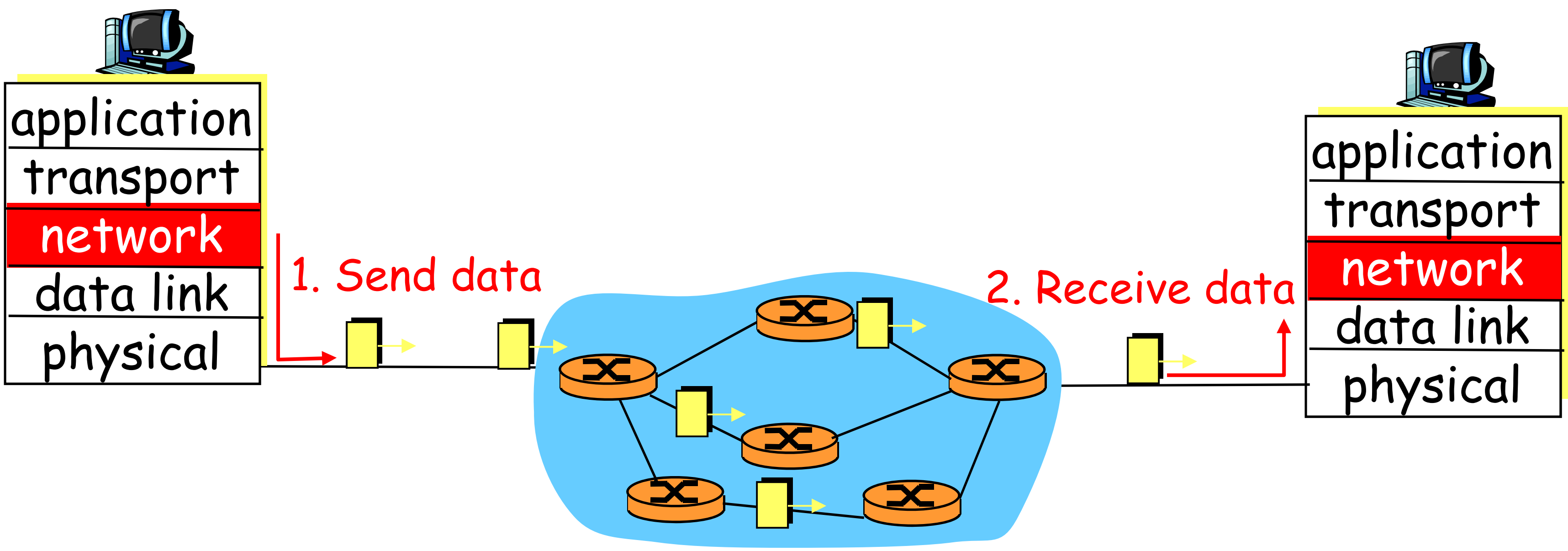
"source-to-dest path behaves much like telephone circuit"

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

- call setup for each call *before* data can flow
- Call teardown when finishes
- 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)

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

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

4 billion
possible entries

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

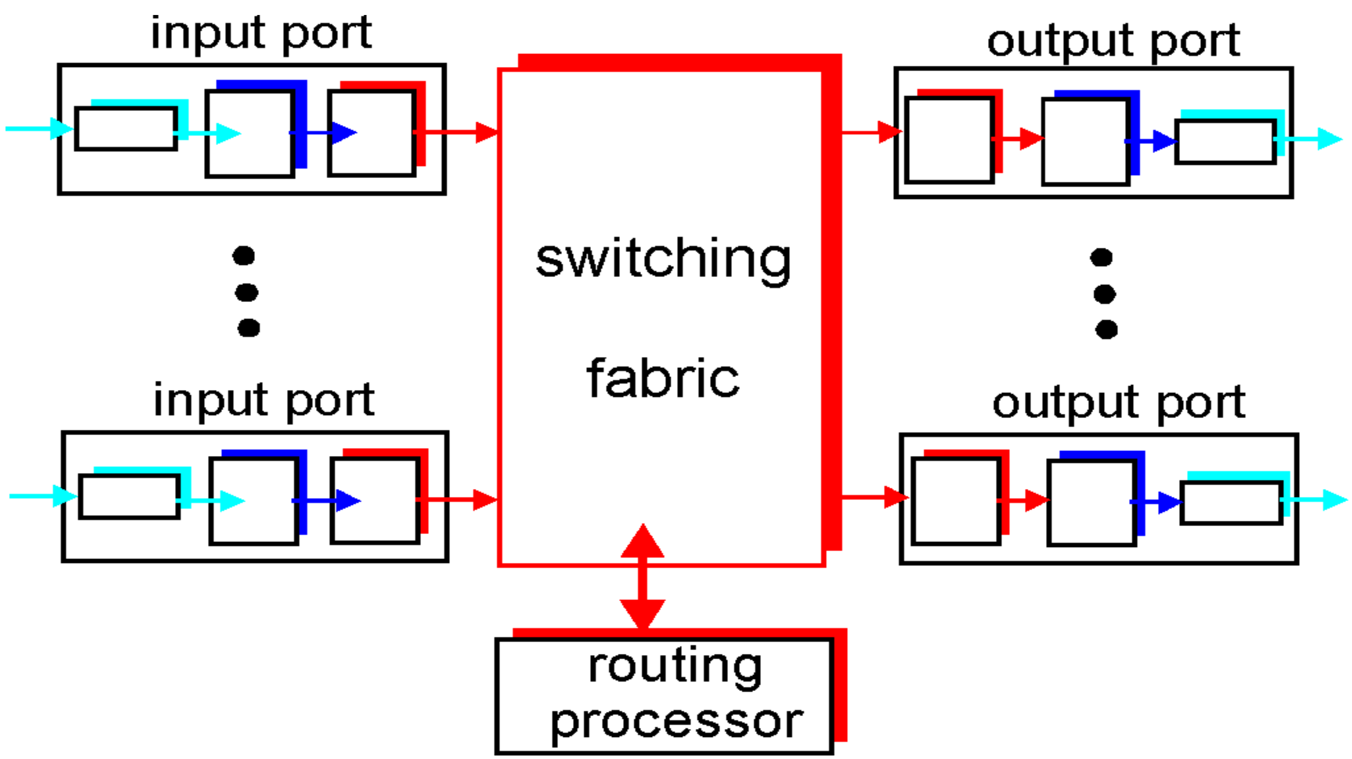
Contents

Note that everything in this set of slides is about IPv4

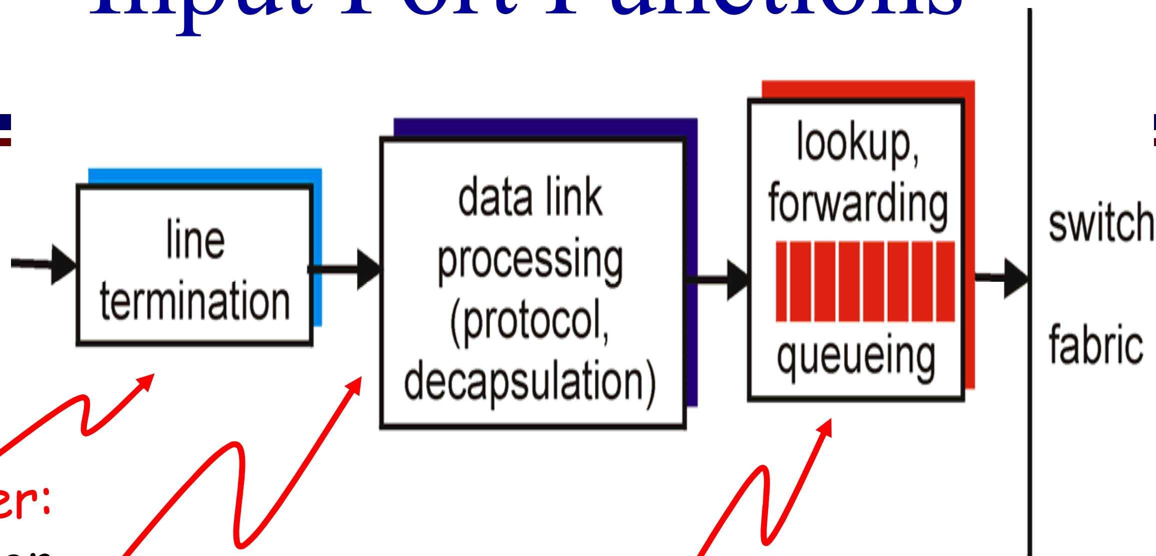
Router Architecture Overview

Two key router functions:

- run routing algorithms/protocol (RIP, OSPF, BGP)
- *forwarding* datagrams from incoming to outgoing link



Input Port Functions



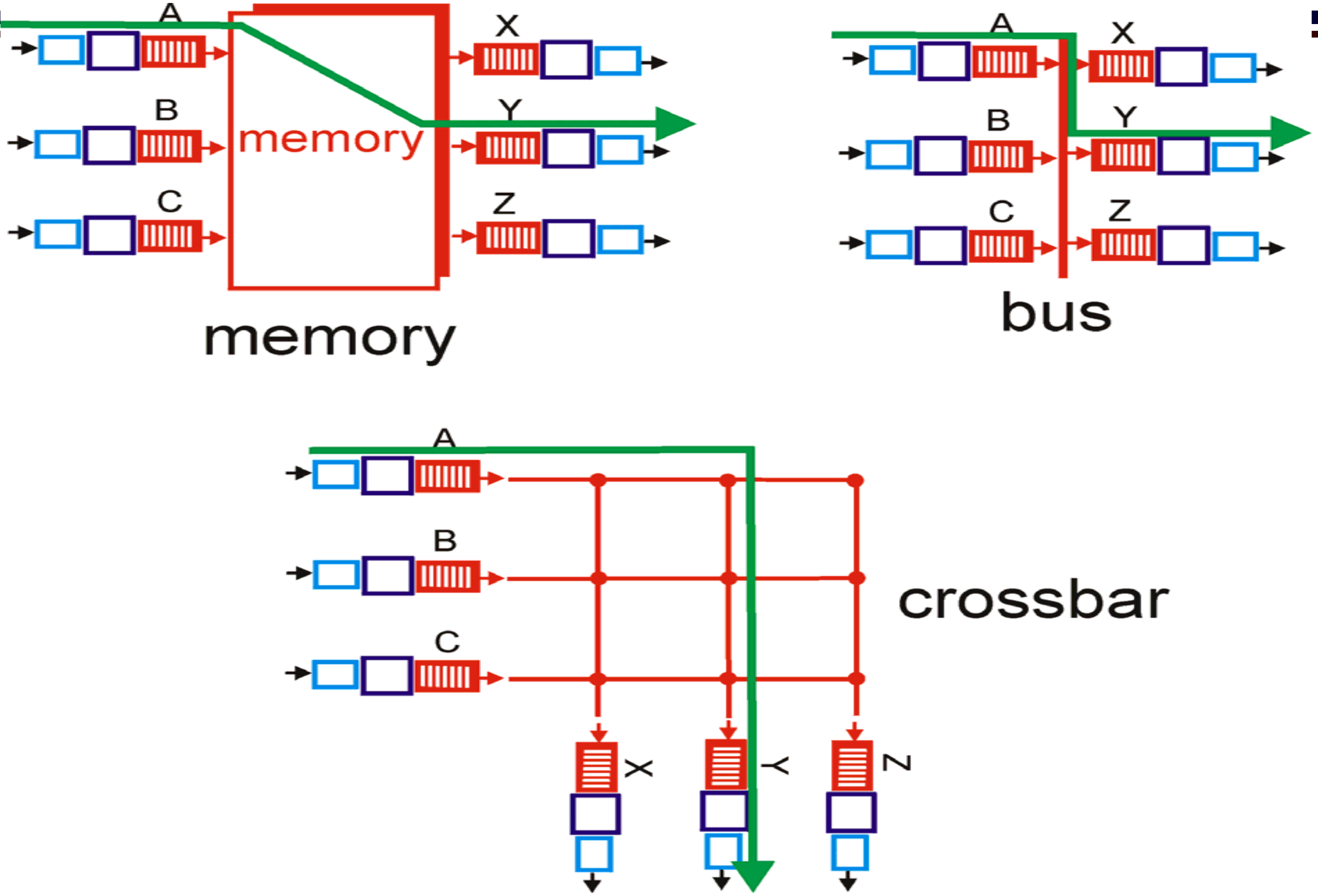
Physical layer:
bit-level reception

Data link layer:
e.g., Ethernet
see 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

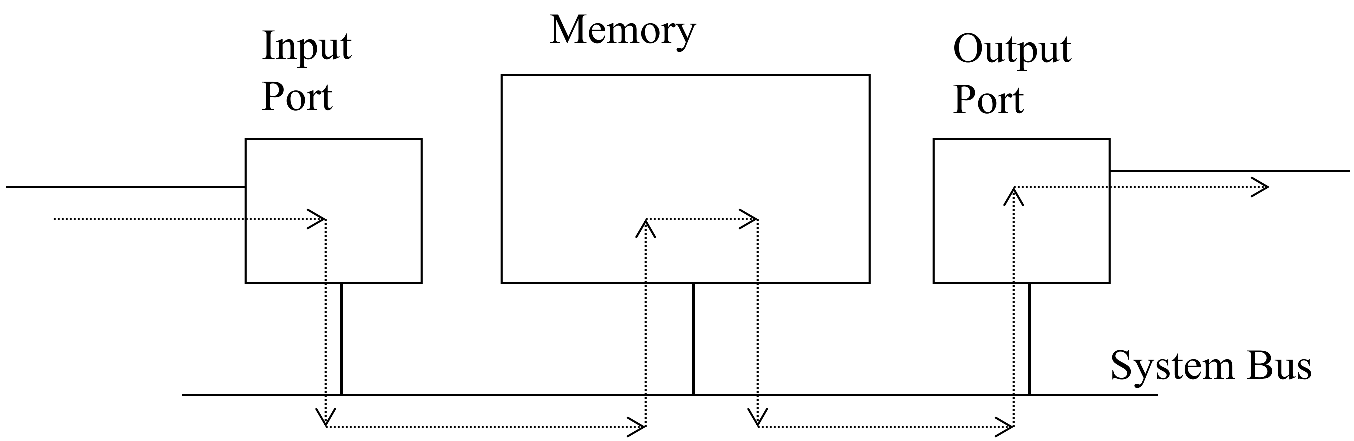
Three types of switching fabrics



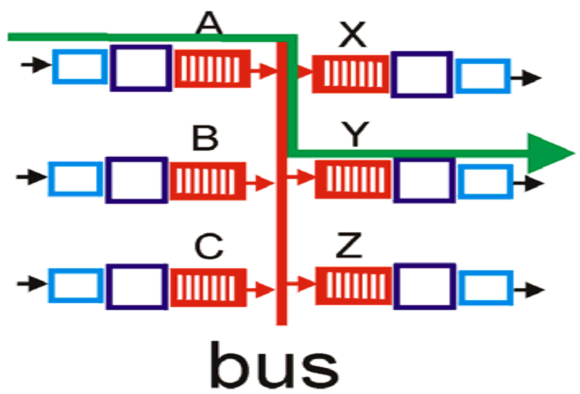
Switching Via Memory

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

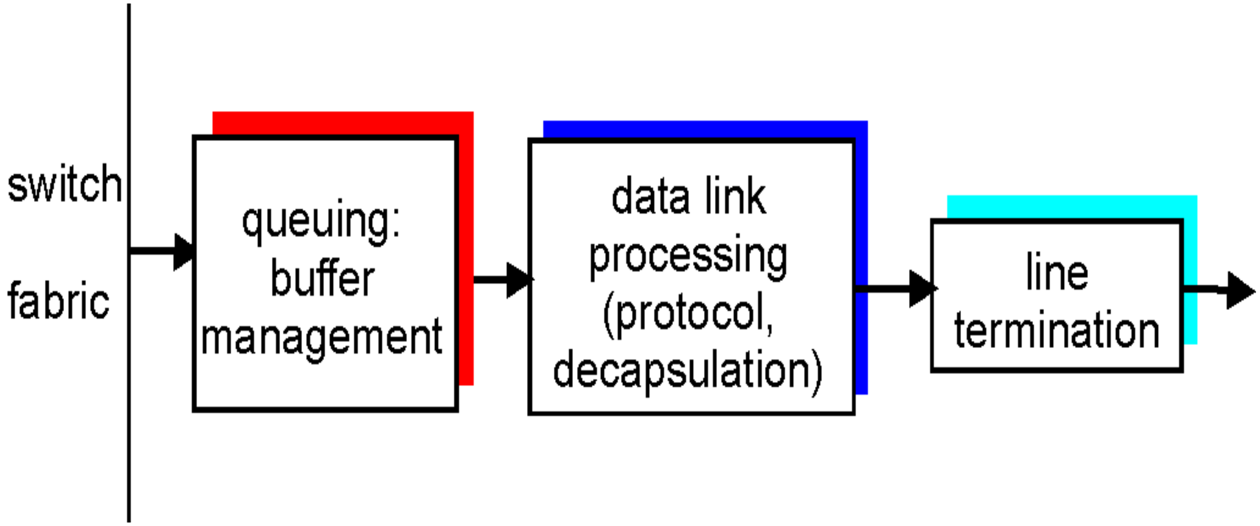


- datagram from input port memory to output port memory via a shared bus
- bus contention: switching speed limited by bus bandwidth
- 1 Gbps bus, Cisco 1900: sufficient speed for access and enterprise routers (not regional or backbone)

Switching Via An Interconnection Network

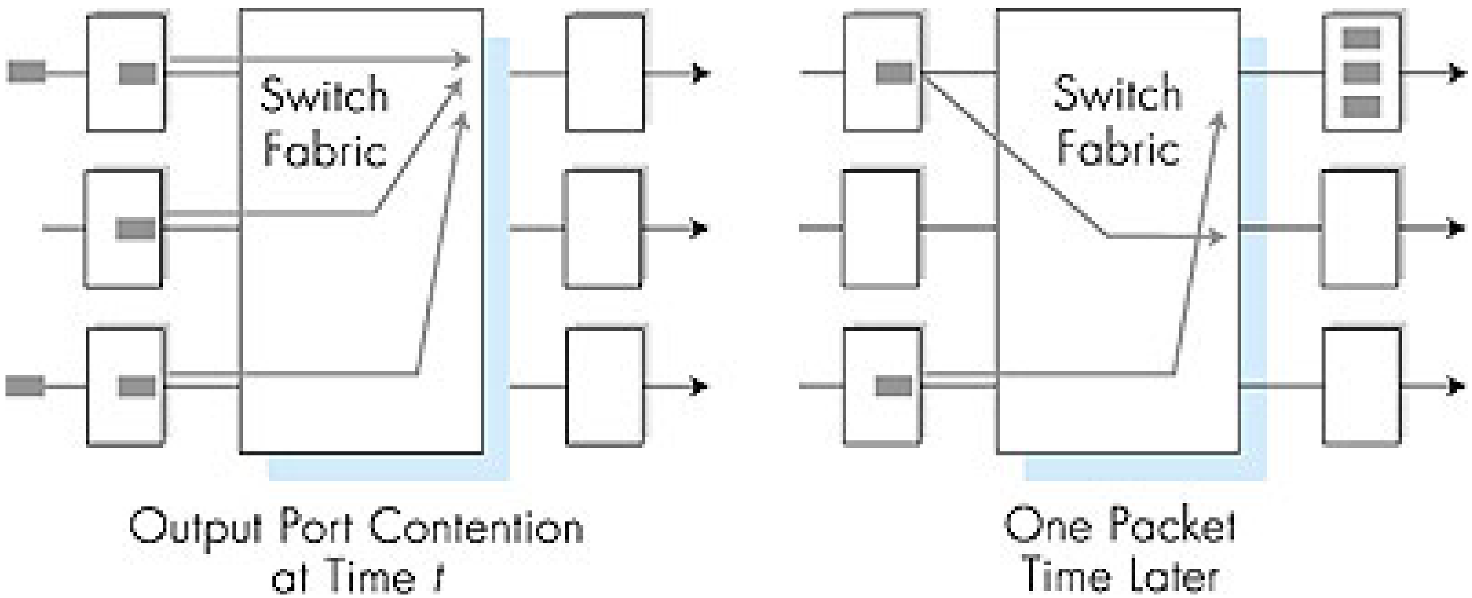
- overcome bus bandwidth limitations
- Similar to interconnection nets initially developed to connect processors in multiprocessor
- Cisco 12000: switches Gbps through the interconnection network

Output Ports



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

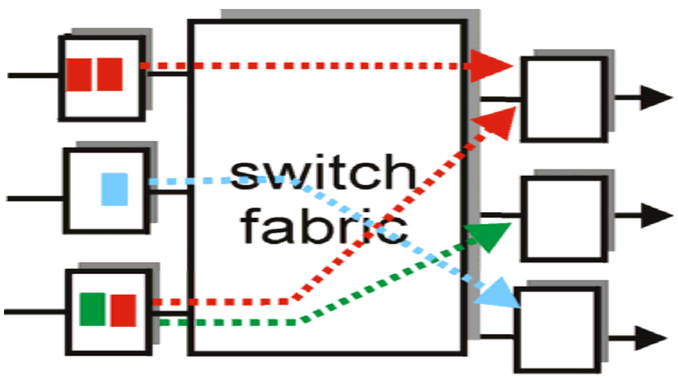
Output port queueing



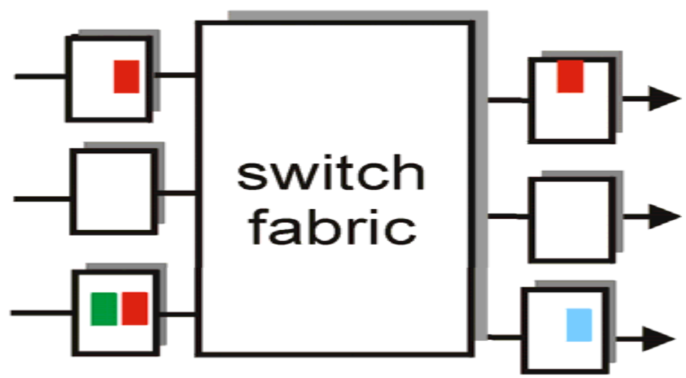
- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

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

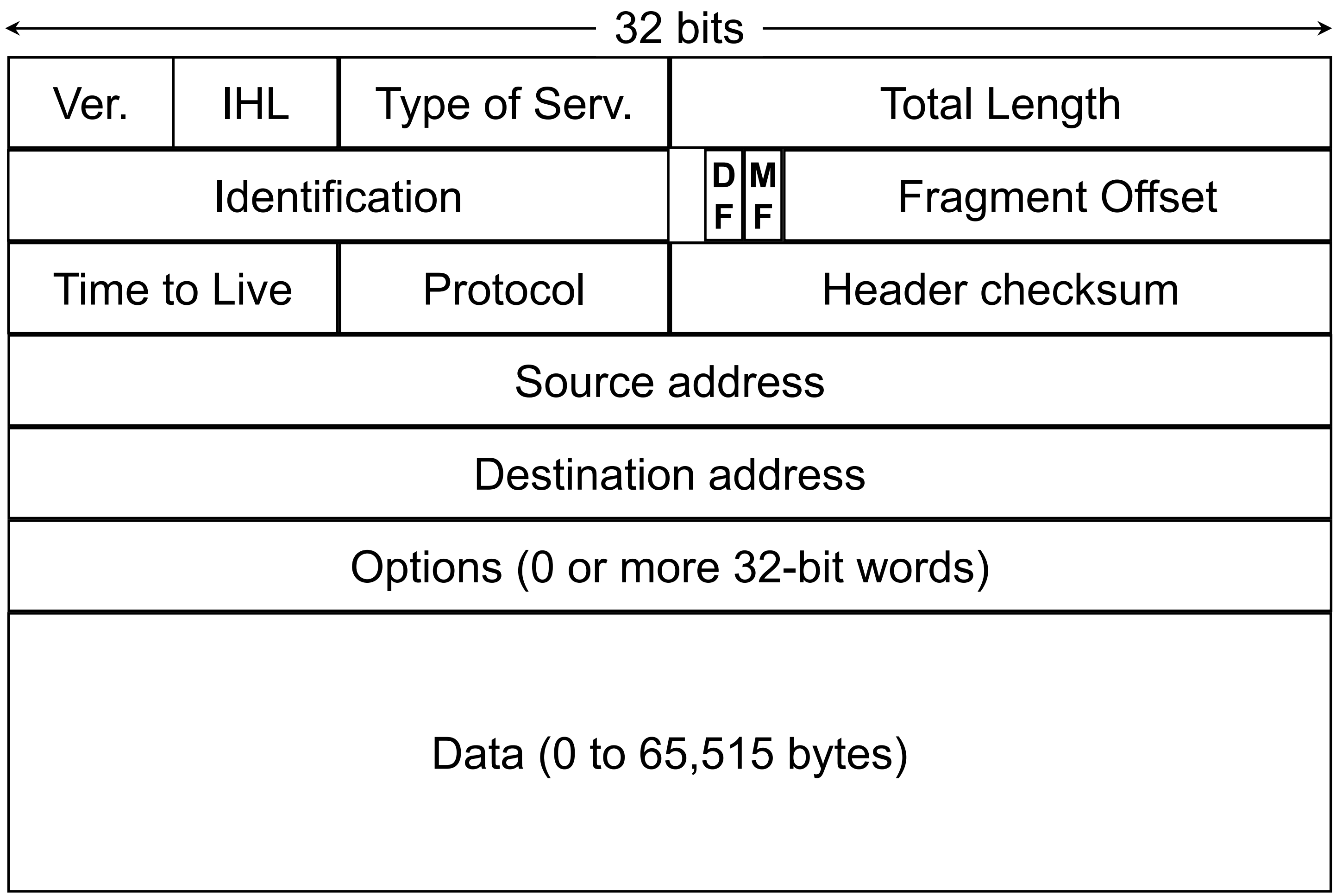
Contents

Note that everything in this set of slides is about IPv4

4. The Internet Protocol (IP)

- Provides delivery of packets from one host in the Internet to any other host in the Internet, even if the hosts are on different networks
- Internet packets are often called “datagrams” and may be up to 64 kilobytes in length (although they are typically much smaller)
- Internet IMPs are known as “routers” and they operate in a connectionless mode

4.1 IP Packet Format



IP Packet Fields

- Version
  - The IP version number (currently 4)
- IHL
  - IP Header Length in 32-bit words
- Type of Service
  - Contains priority information, rarely used
- Total Length
  - The total length of the datagram in bytes
  - Includes header

IP Packet Fields *(cont'd)*

- Identification
  - When an IP packet is segmented into multiple fragments, each fragment is given the same identification
  - This field is used to reassemble fragments
- DF
  - Don't Fragment
- MF
  - More Fragments
  - When a packet is fragmented, all fragments except the last one have this bit set
- Fragment offset
  - The fragment's position within the original packet

IP Packet Fields *(cont'd)*

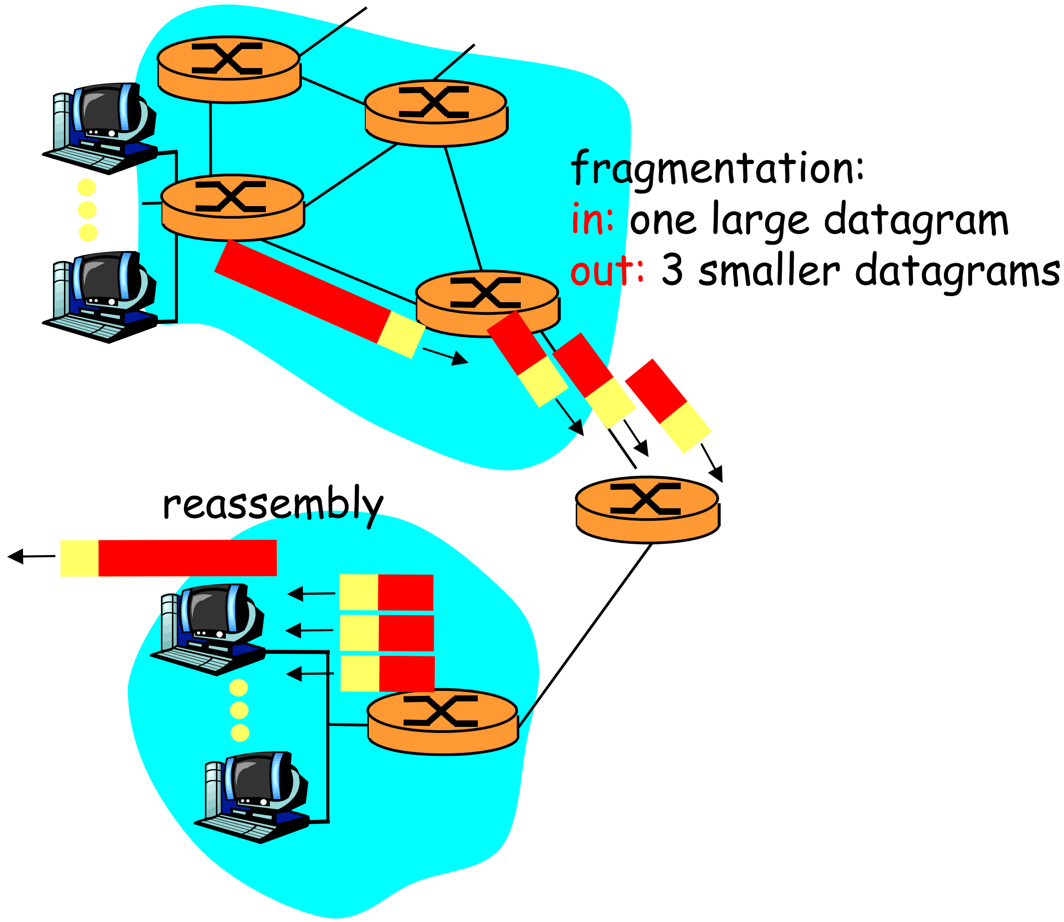
- Time to Live
  - Hop count, decremented each time the packet reaches a new router
  - When hop count = 0, packet is discarded
- Protocol
  - Identifies which transport layer protocol is being used for this packet
- Header Checksum
  - Verifies the contents of the IP header
  - Not polynomial-based

IP Packet Fields *(cont'd)*

- Source and Destination Addresses
  - Uniquely identify sender and receiver of the packet
- Options
  - Up to 40 bytes in length
  - Used to extend functionality of IP
  - Examples: source routing, security, record route

IP Fragmentation & Reassembly

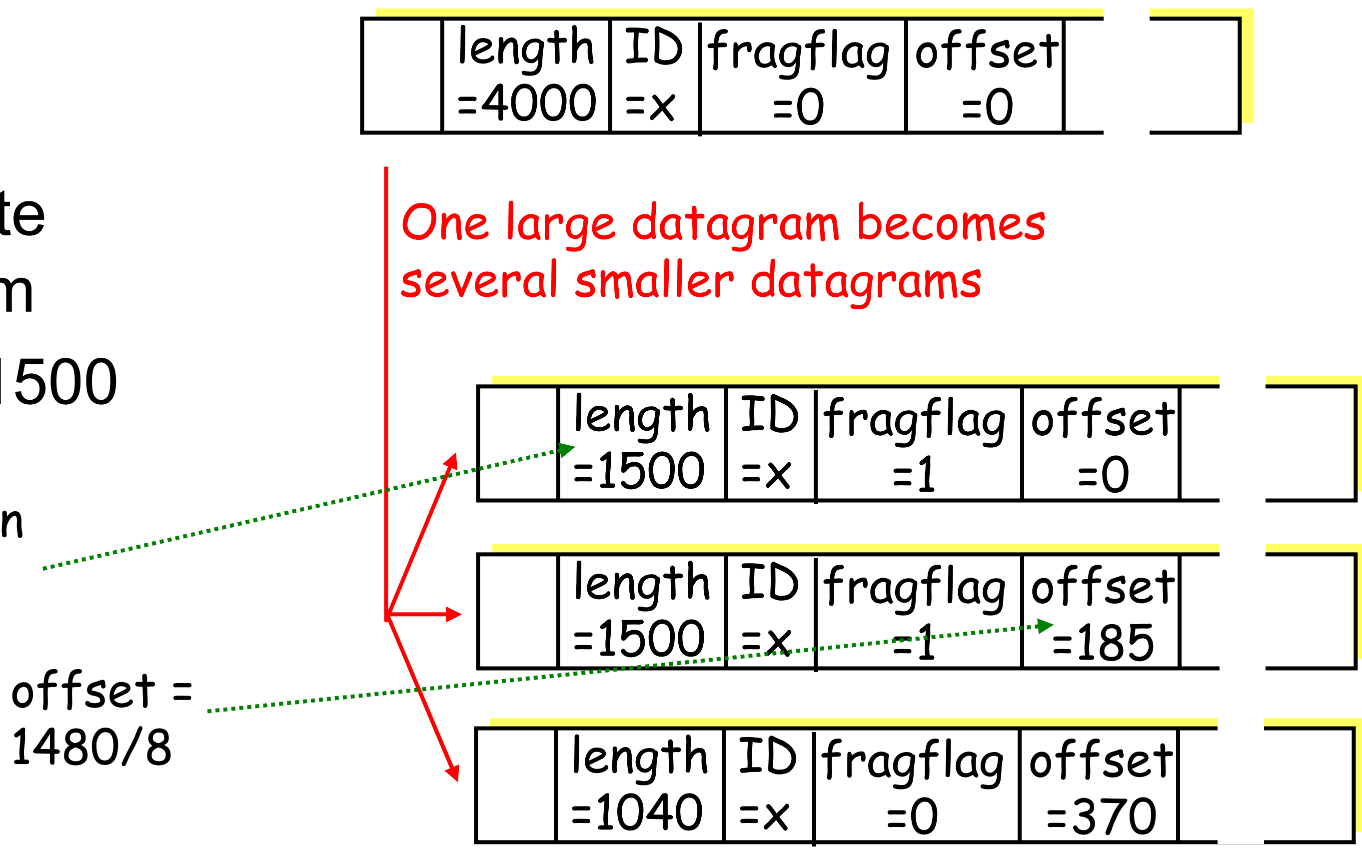
- network links have MTU (max.transfer size) - largest possible link-level frame.
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify,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



IP Fragmentation and Reassembly

Example

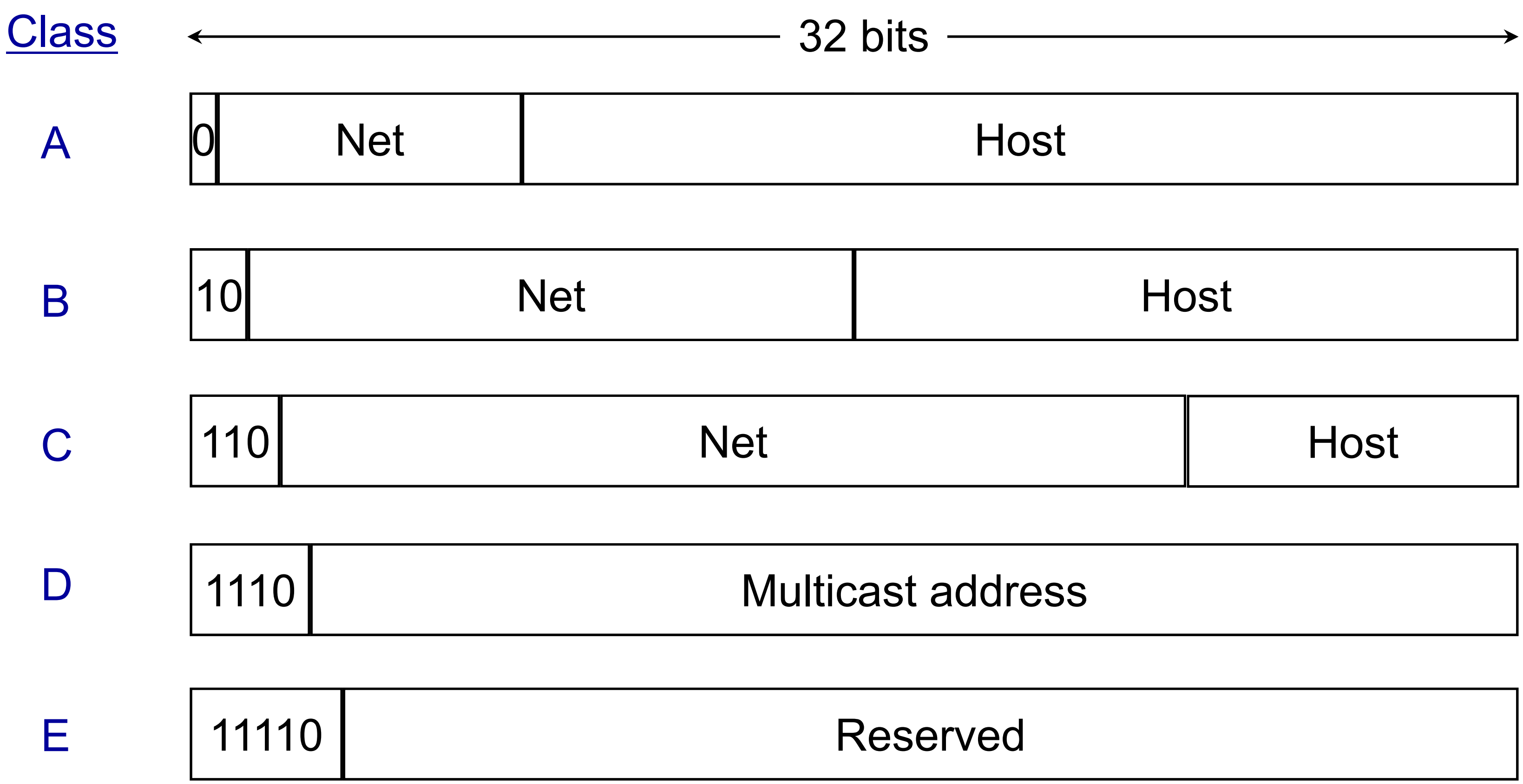
- 4000 byte datagram
- MTU = 1500 bytes



4.2 IP Addresses

- 32 bits long (4 bytes)
- Notation:
  - Each byte is written in decimal in MSB order, separated by decimals
  - Example: 128.195.1.80
  - 0.0.0.0 (lowest) to 255.255.255.255 (highest)
- Address Classes
  - Class A, B, C, D, E
  - Loopback
  - Broadcast

IP Address Classes



IP Address Classes

- Class A:
  - For very large organizations
  - around16 million hosts allowed (2 ^ 24 -2)
- Class B:
  - For large organizations
  - Around 65 thousand hosts allowed (2^16 – 2)
- Class C
  - For small organizations
  - Around 256 hosts allowed (2 ^ 8 -2)
- Class D
  - Multicast addresses
  - No network/host hierarchy

- Class E
  - reserved
- Loopback
  - 127.xx.yy.zz (127.anything) is reserved for loopback testing
  - packets sent to this address are not put out onto the wire; they are processed locally and treated as incoming packets.
- Broadcast
  - all 1s

IP Address Hierarchy

- Note that Class A, Class B, and Class C addresses only support two levels of hierarchy
- Each address contains a network and a host portion, meaning two levels of hierarchy
- However, the host portion can be further split into “subnets” by the address class owner
- This allows for more than 2 levels of hierarchy

Subnetting

Example: Class B address with 8-bit subnetting

	16 bits	8 bits	8 bits
	Network id	Subnet id	Host id
Example Address:	165.230	.24	.8

Class	32 bits		
B	10	Net	Host

Subnet Masks

Subnet masks allow hosts to determine if another IP address is on the same subnet or the same network

	16 bits	8 bits	8 bits
	Network id	Subnet id	Host id
Mask:	1111111111111111	11111111	00000000
	255.255	.255	.0

Subnet Masks *(cont'd)*

Assume IP addresses X and Y share subnet mask M.

Are IP addresses X and Y on the same subnet?

1. Compute (X and M). (Boolean AND)
2. Compute (Y and M). (Boolean AND)
3. If (X and M) = (Y and M) then X and Y are on the same subnet.

Example: X and Y are class B addresses

X = 165.230.82.52
Y = 165.230.24.93
M = 255.255.255.0

Same network?
Same subnet?

- Note
  - 0 AND 0 = 0
  - 0 AND 1 = 1 AND 0 = 0
  - 1 AND 1 = 1
- Thus, computing (X and M) results in
  - Network ID = Network ID of X
  - Subnet ID = Subnet ID of Y
  - Host ID = 0

- Routing table

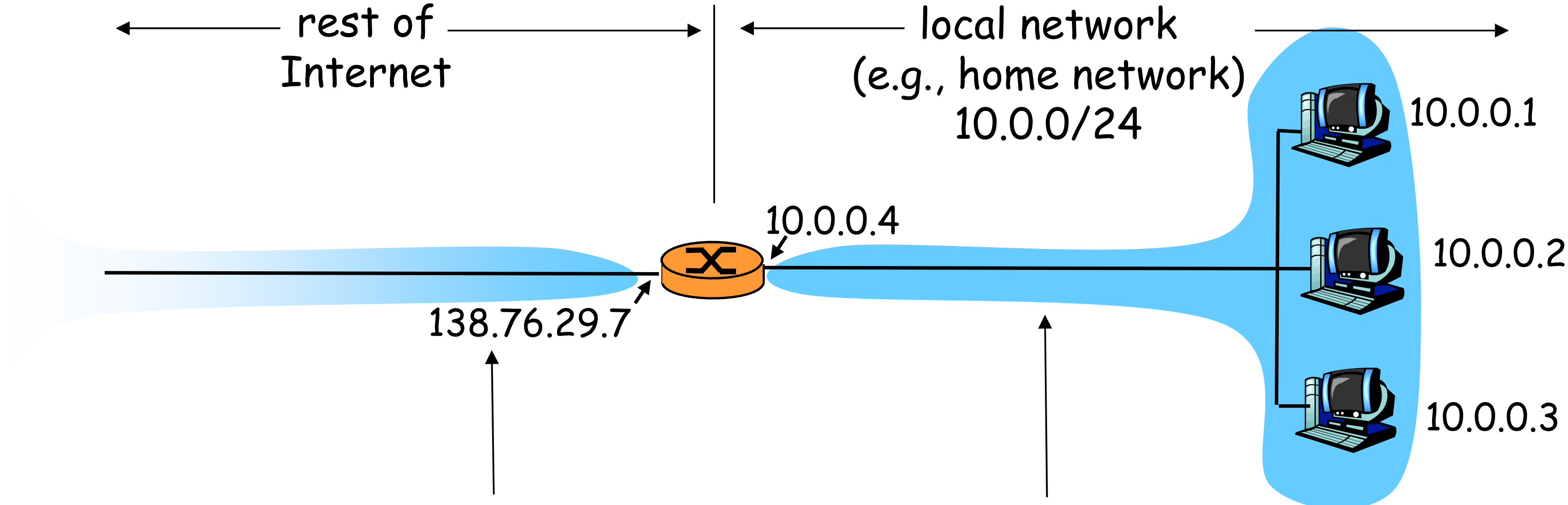
network ID	subnet ID	host ID
this network	this subnet	X
this network	this subnet	Y
this network	different subnet	0
this network	different subnet	0
different network	0	0

- Subnet mask helps quickly identifying which routing table entry to look up

IP Addressing

- How does an ISP get block of addresses?
  - 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

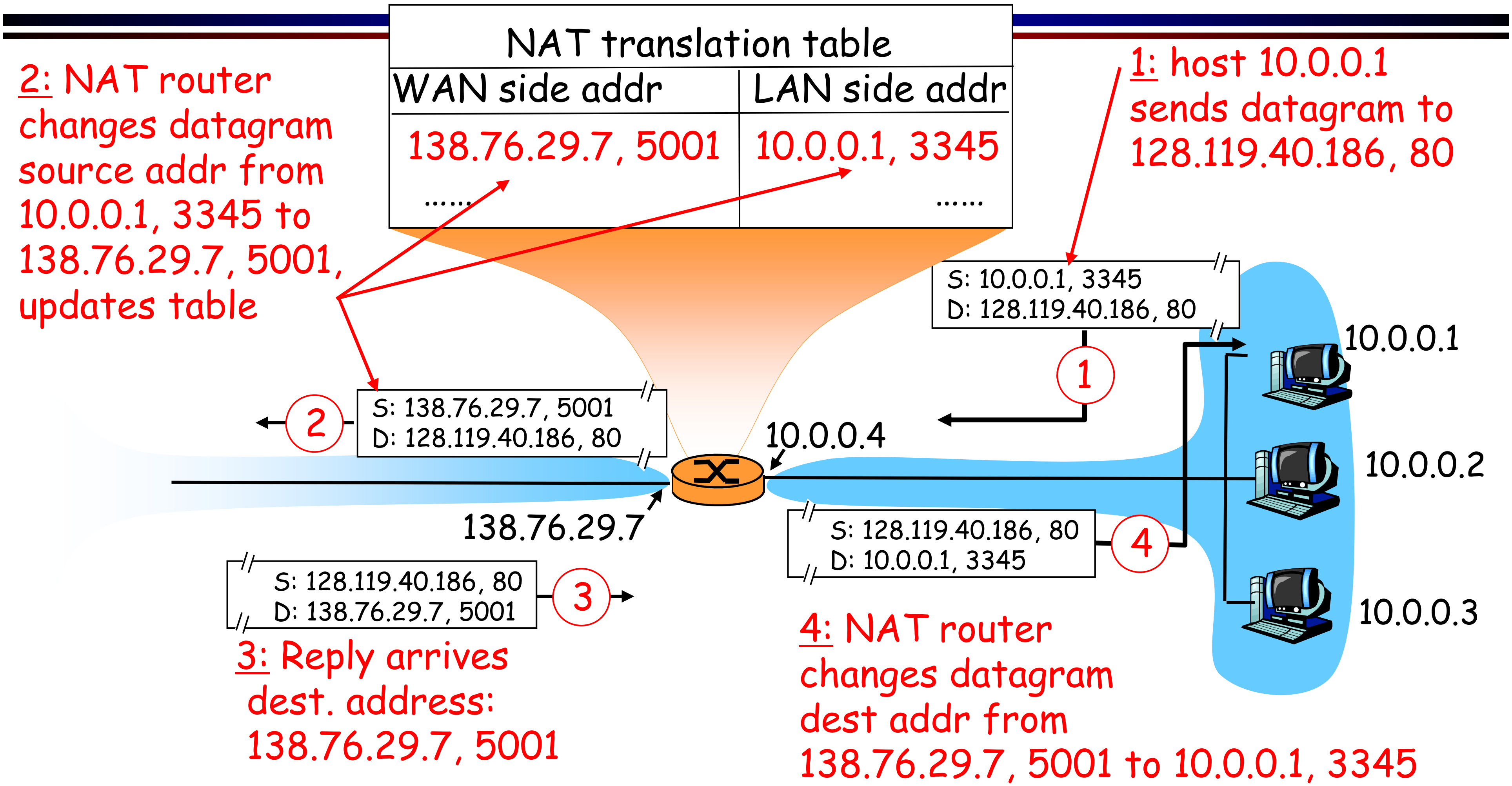
Implementation: NAT router must:

- *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

Contents

Note that everything in this set of slides is about IPv4

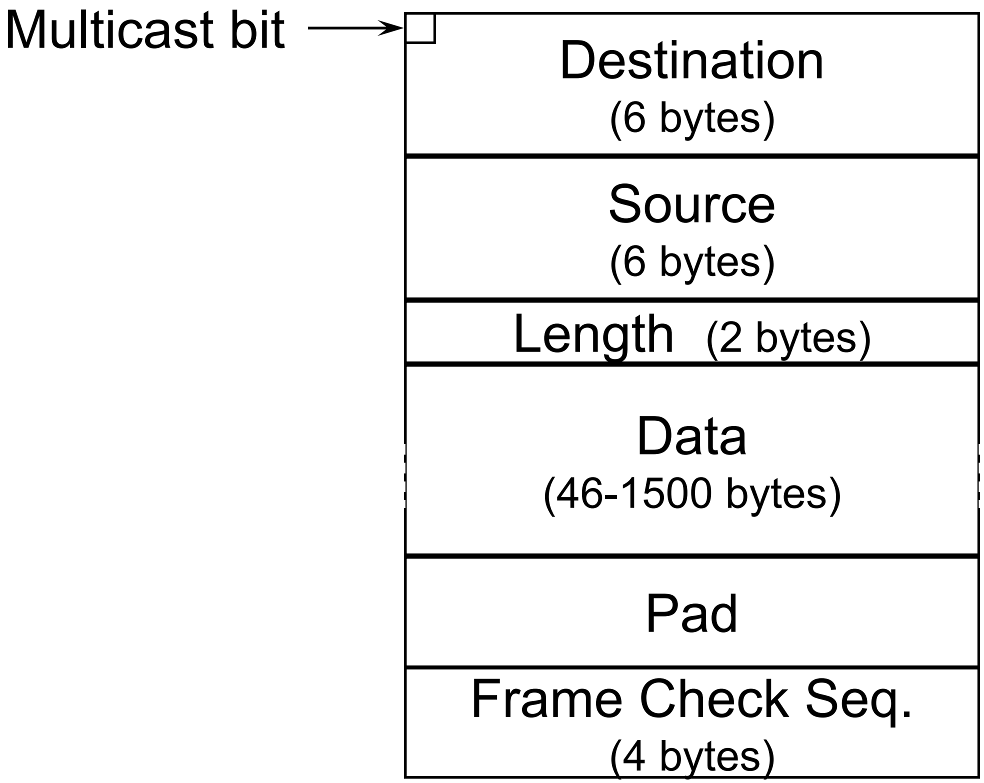
5. IP Support Protocols

- ARP
- RARP
- ICMP

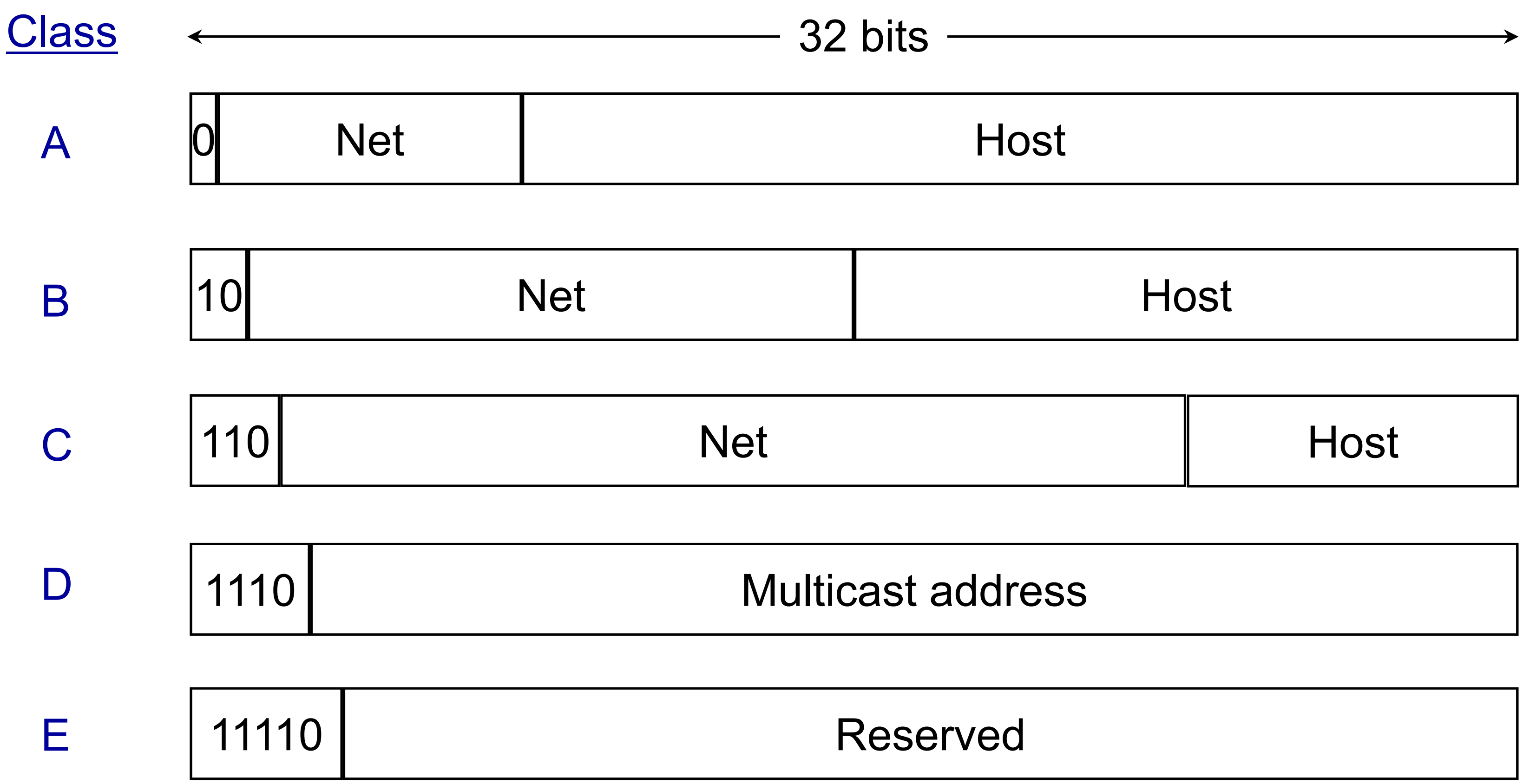
5.1 ARP

- Address Resolution Protocol
- Returns a MAC sublayer address when given an Internet (IP) address
- Commonly used in broadcast LANs so that two hosts can communicate using IP addresses instead of MAC sublayer addresses

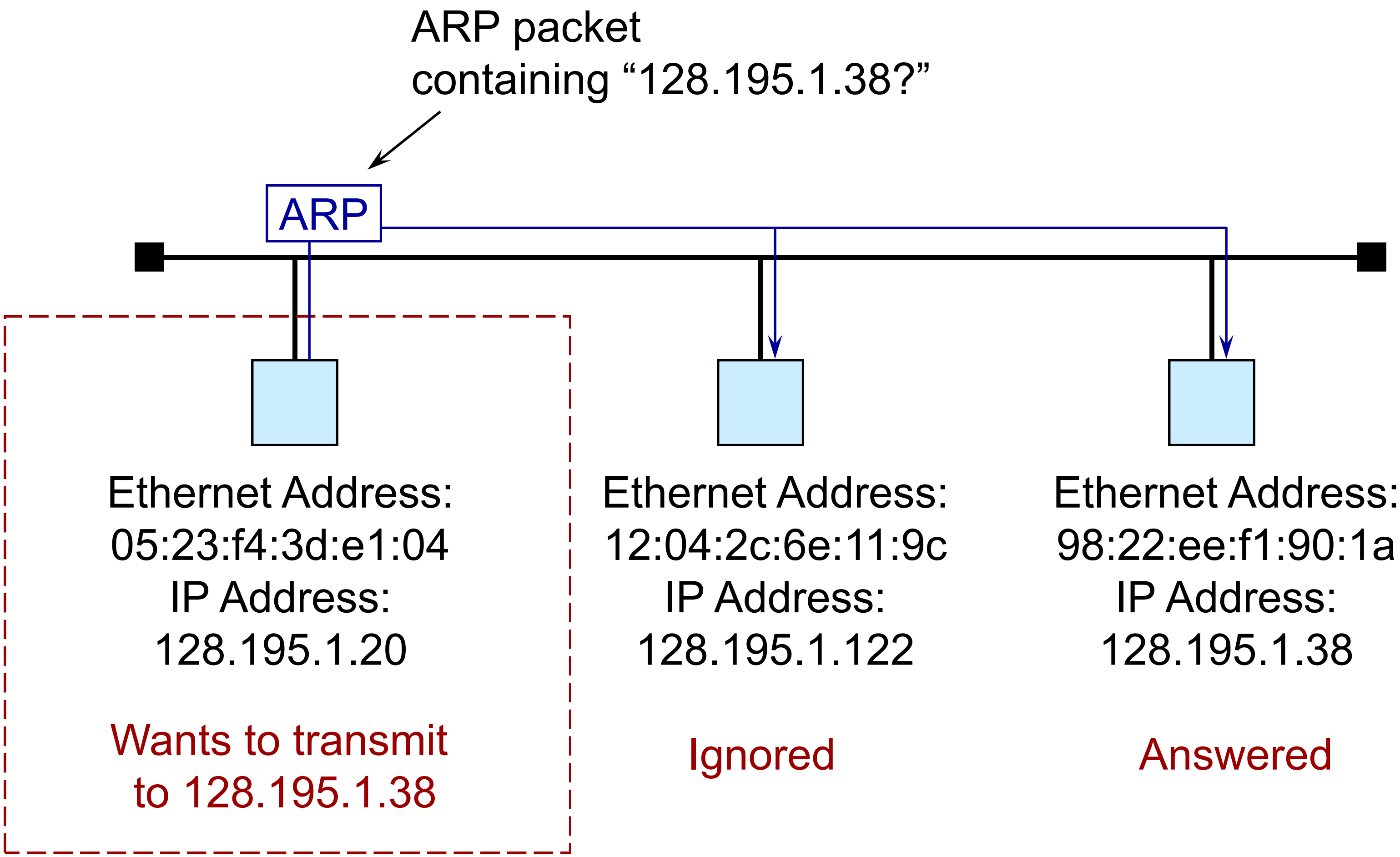
MAC Layer Ethernet Frame Format



IP Address Classes



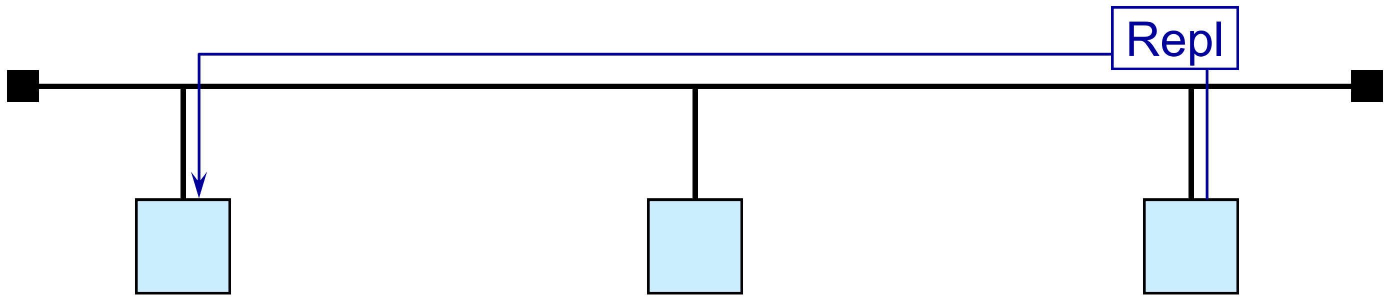
ARP *(cont'd)*



ARP packet
containing "128.195.1.38?"
ARP
Ethernet Address:
05:23:f4:3d:e1:04
IP Address:
128.195.1.20
Wants to transmit
to 128.195.1.38
Ethernet Address:
12:04:2c:6e:11:9c
IP Address:
128.195.1.122
Ignored
Ethernet Address:
98:22:ee:f1:90:1a
IP Address:
128.195.1.38
Answered

ARP *(cont'd)*

ARP response packet
containing "98:22:ee:f1:90:1a"



Ethernet Address: 05:23:f4:3d:e1:04 IP Address: 128.195.1.20	Ethernet Address: 12:04:2c:6e:11:9c IP Address: 128.195.1.122	Ethernet Address: 98:22:ee:f1:90:1a IP Address: 128.195.1.38

5.2 RARP

- Reverse Address Resolution Protocol
- RARP performs the inverse action of ARP
- RARP returns an IP address for a given MAC sublayer address
- Operationally, RARP is the same as ARP

5.3 ICMP

- Internet Control Message Protocol
- Handles special Internet control functions
- Responsibilities:
  - Reporting unreachable destinations
  - Reporting IP packet header problems
  - Reporting routing problems
  - Reporting echoes (pings)

ICMP

- Protocol for error detection and reporting
  - tightly coupled with IP, unreliable
- ICMP messages delivered in IP packets
- ICMP functions:
  - Announce network errors
  - Announce network congestion
  - Assist trouble shooting
  - Announce timeouts

ICMP MSG

IP header Source, Destination Address, TTL, ...
ICMP MSG Message type, Code, Checksum, Data

Message type examples (Figure 6.3 in Stevens book):

0 (8) echo request (reply)
3 destination unreachable
4 source quench
11 time exceeded

type	*code*	Description	Query	Error
0	0	echo reply (Ping reply, Chapter 7)	•	
3		destination unreachable:		
	0	network unreachable (Section 9.3)		•
	1	host unreachable (Section 9.3)		•
	2	protocol unreachable		•
	3	port unreachable (Section 6.5)		•
	4	fragmentation needed but don't-fragment bit set (Section 11.6)		•
	5	source route failed (Section 8.5)		•
	6	destination network unknown		•
	7	destination host unknown		•
	8	source host isolated (obsolete)		•
	9	destination network administratively prohibited		•
	10	destination host administratively prohibited		•
	11	network unreachable for TOS (Section 9.3)		•
	12	host unreachable for TOS (Section 9.3)		•
	13	communication administratively prohibited by filtering		•
	14	host precedence violation		•
	15	precedence cutoff in effect		•
4	0	source quench (elementary flow control, Section 11.11)		•
5		redirect (Section 9.5):		
	0	redirect for network		•
	1	redirect for host		•
	2	redirect for type-of-service and network		•
	3	redirect for type-of-service and host		•
8	0	echo request (Ping request, Chapter 7)	•	
9	0	router advertisement (Section 9.6)	•	
10	0	router solicitation (Section 9.6)	•	
11		time exceeded:		
	0	time-to-live equals 0 during transit (Traceroute, Chapter 8)		•
	1	time-to-live equals 0 during reassembly (Section 11.5)		•
12		parameter problem:		
	0	IP header bad (catchall error)		•
	1	required option missing		•
13	0	timestamp request (Section 6.4)	•	
14	0	timestamp reply (Section 6.4)	•	
15	0	information request (obsolete)	•	
16	0	information reply (obsolete)	•	
17	0	address mask request (Section 6.3)	•	
18	0	address mask reply (Section 6.3)	•	

Figure 6.3 ICMP message types.

Specific uses of ICMP

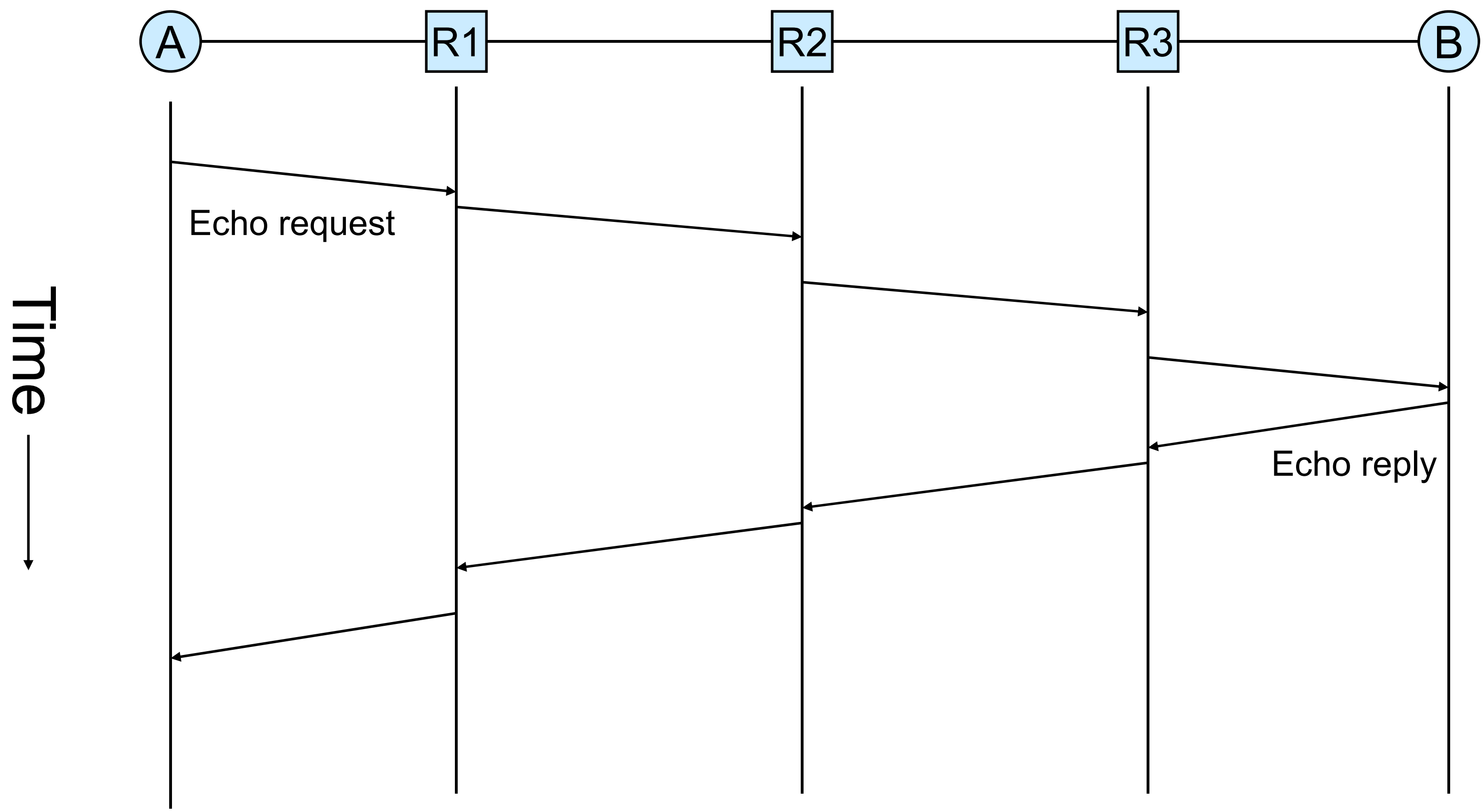
- Echo request/reply
  - Can be used to check if a host is alive
- Address mask request/reply
  - Learn the subnet mask
- Destination unreachable
  - Invalid address and/or port
- Source quench
  - choke packet

- TTL expired
  - Routing loops, or too far away

Ping

- Uses ICMP echo request/reply
- Source sends ICMP echo request message to the destination address
  - Echo request packet contains sequence number and timestamp
- Destination replies with an ICMP echo reply message containing the data in the original echo request message
- Source can calculate round trip time (RTT) of packets
- If no echo reply comes back then the destination is unreachable

Ping *(cont'd)*



Traceroute

- Traceroute records the route that packets take
- A clever use of the TTL field
- When a router receives a packet, it decrements TTL
- If TTL=0, it sends an ICMP time exceeded message back to the sender
- To determine the route, progressively increase TTL
  - Every time an ICMP time exceeded message is received, record the sender's (router's) address
  - Repeat until the destination host is reached or an error message occurs

Traceroute *(cont'd)*

Te = Time exceeded
Pu = Port unreachable