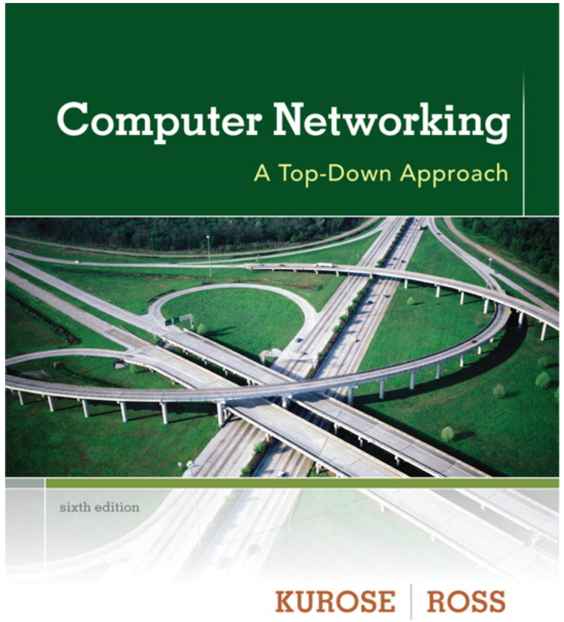
Chapter 4
Network Layer



Computer Networking

A Top-Down Approach

6th edition

Jim Kurose, Keith Ross

Addison-Wesley

March 2012

These slides are based on the slides made available by Kurose and Ross.

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*

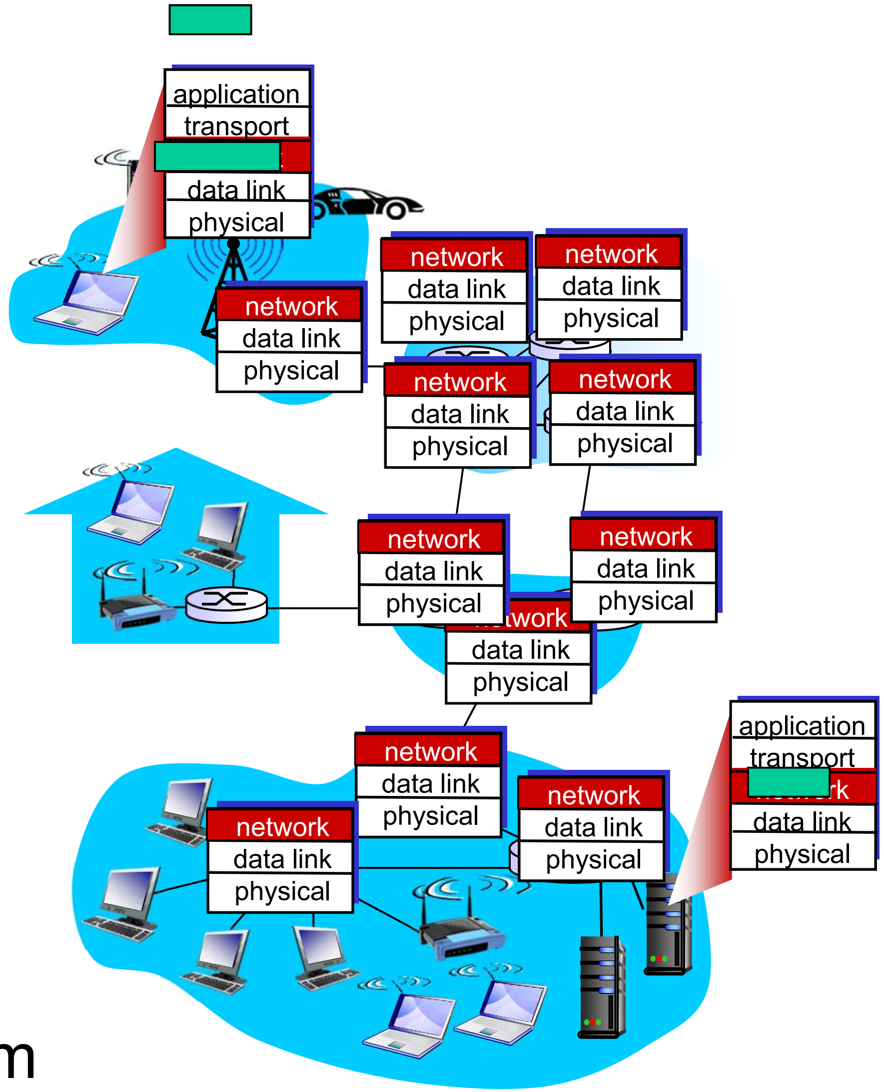
Network Layer Functions

- network layer protocol in *every* host *and* router

Consider transporting a segment from sender to receiver

- sending side: encapsulates segments into datagrams
- receiving side: delivers segments to transport layer

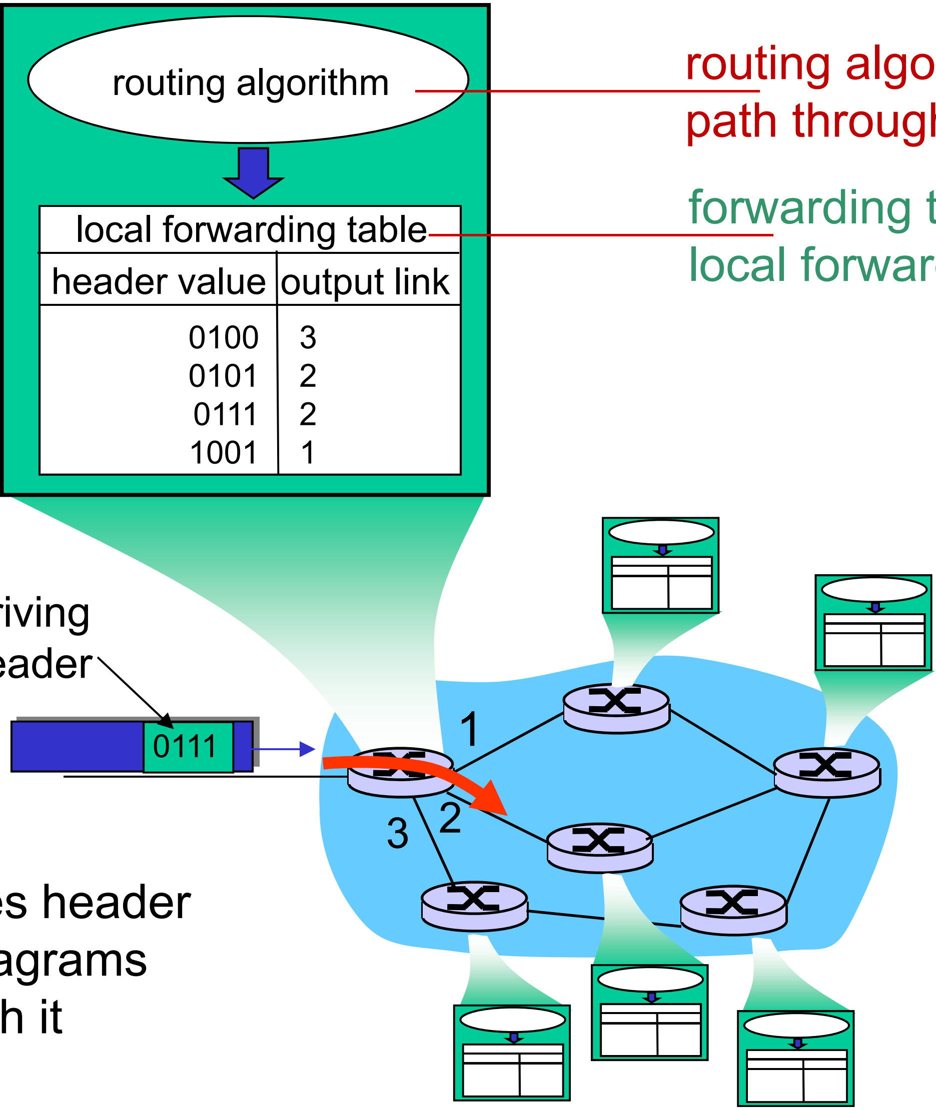
- Path Determination: sum of routes chosen by routers to deliver packets from source to destination.
- Forwarding: move packets from router's input to appropriate router's output



Routing and Forwarding

routing algorithm determines path through network

forwarding table determines local forwarding at this router



value in arriving packet's header

router examines header fields in all datagrams passing through it

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 Service Model

What *service model* can be considered for a network transporting packets from sender to receiver?

*example services for **individual** datagrams:*

- "best effort" delivery
- No constraints on delay or bandwidth

*example services for a **flow** of packets:*

- in-order delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet time-spacing

Connection-oriented & connectionless

- **Virtual Circuit-network provides link or network-layer connection-oriented service.**
- **Datagram-based network provides network-layer connectionless service.**
- **Analogous to the transport-layer services but:**
  - ***Service:*** host-to-host packet delivery
  - ***Implementation:*** every router in the network

Virtual Circuit: VC

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

- Performance-wise (but it is virtual circuit)
- Network actions along the source-to-destination path

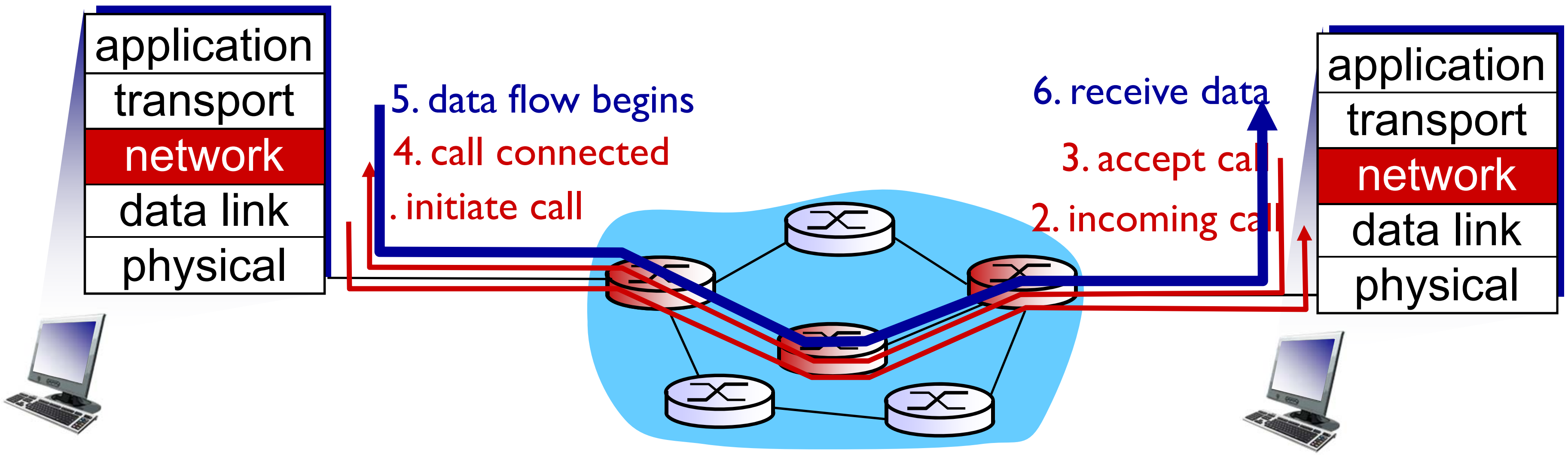
- Setup: for each connection before data packets can flow
- Each packet carries VC identifier (not destination address)
- Every router on the path maintains "state" for each passing connection.

Benefit: Link & router resources (bandwidth, buffers) may be allocated to VC

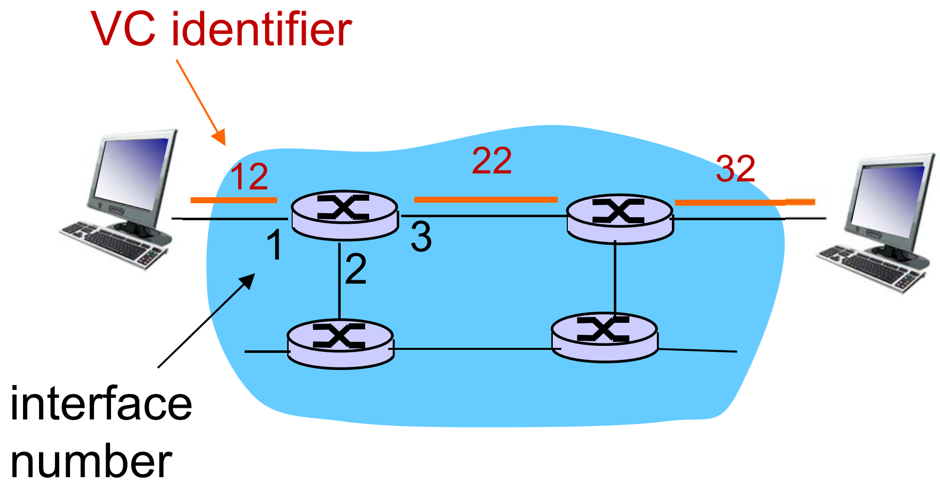
(dedicated resources = predictable service)

VC: Signaling Protocols

- used to setup, maintain and teardown VC
- used in ATM, Frame-Relay, X.25
- not used in today's Internet on network layer



VC: Forwarding Table



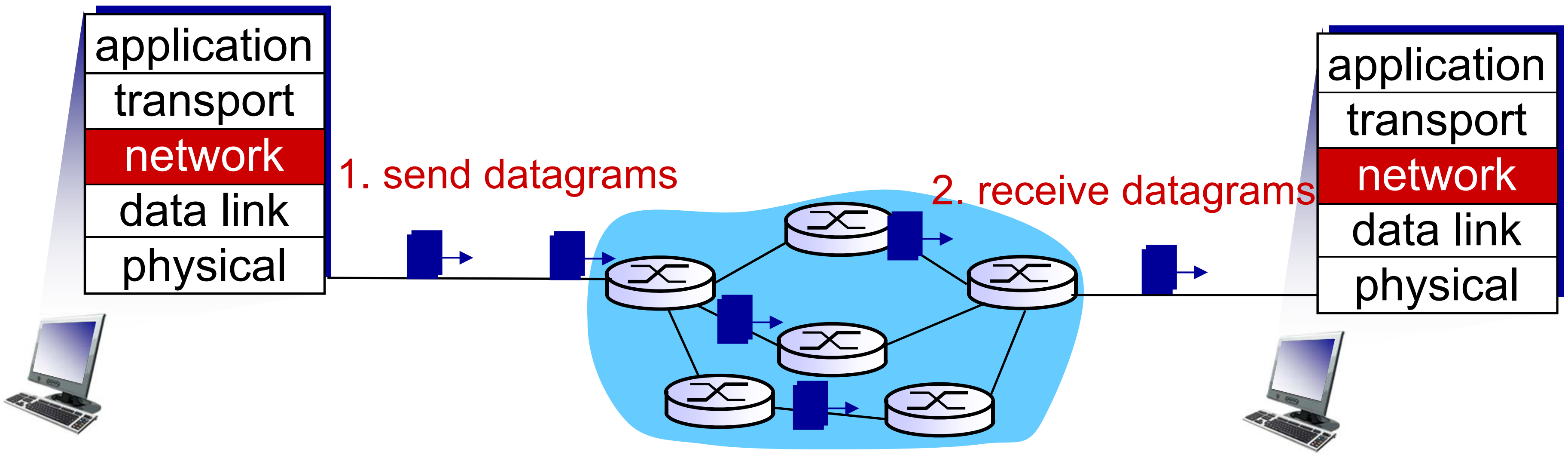
Forwarding table
in northwest router:

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

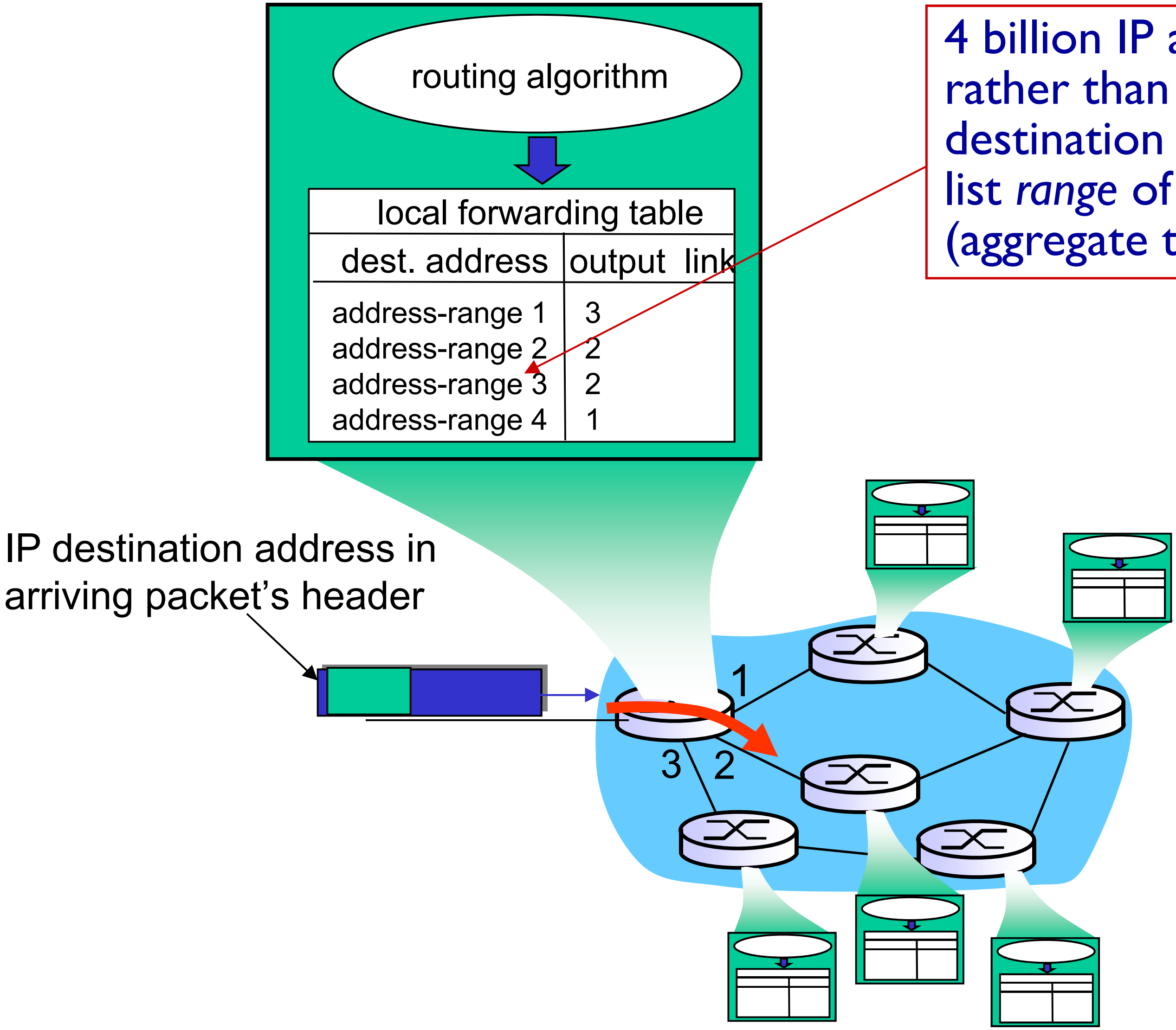
Routers maintain connection state information!

Datagram Networks (Internet)

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



Datagram: Forwarding Table



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

Datagram or VC network: why?

Internet (datagram)

- data exchange among computers
  - "elastic" service, no strict timing requirements.
- "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)

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

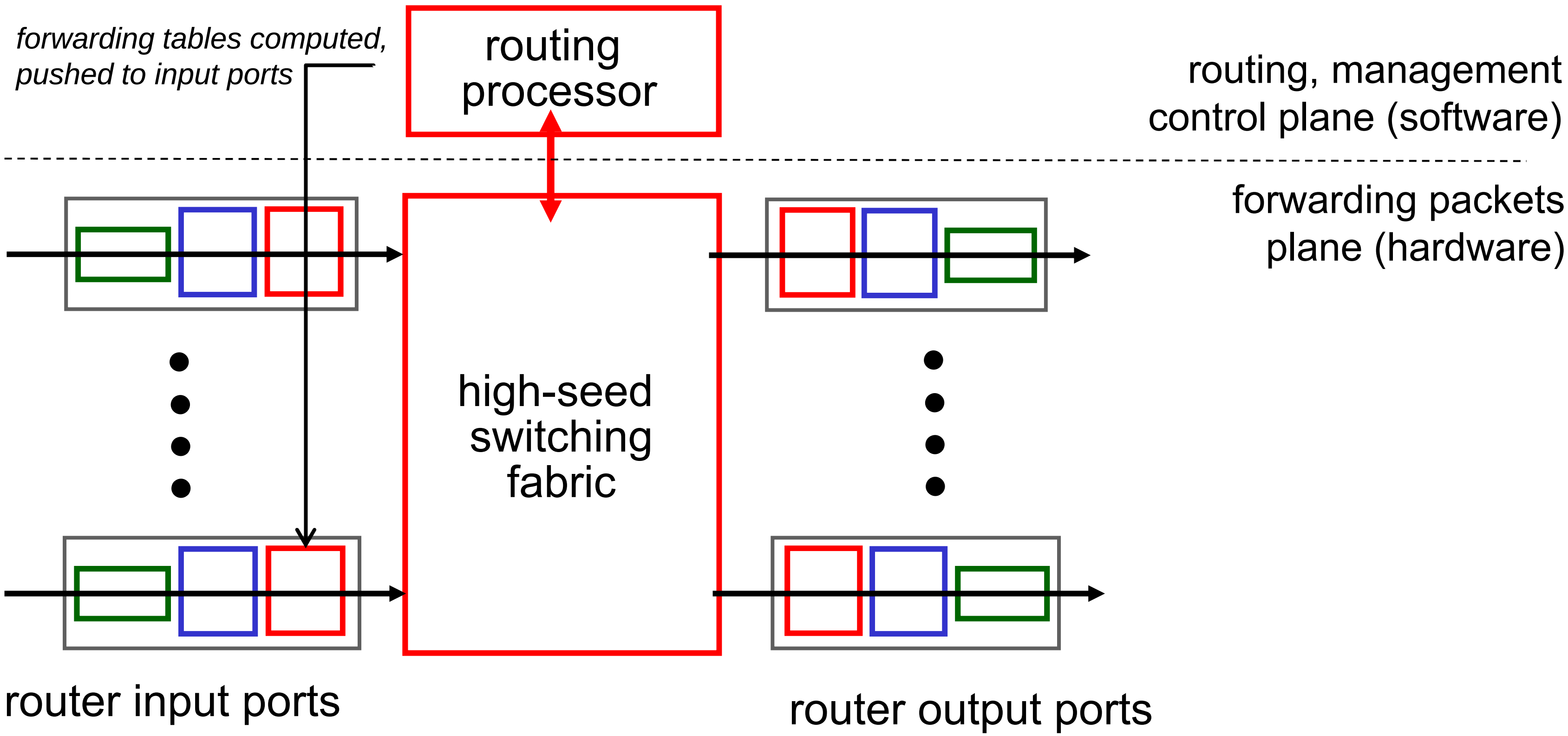
Chapter 4: Network Layer

- 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*

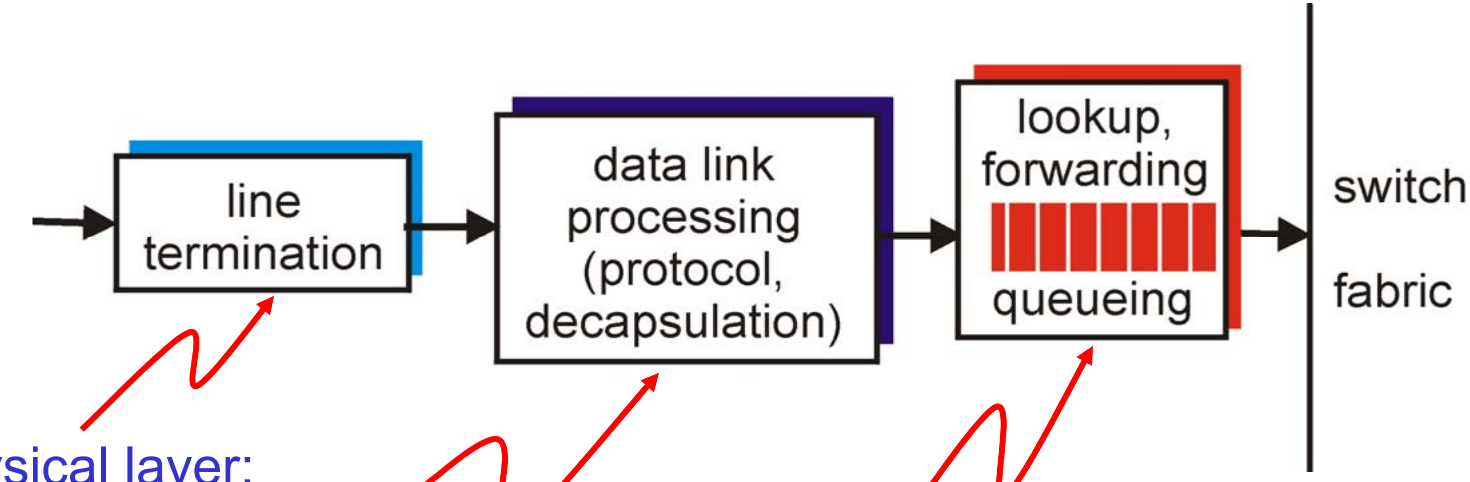
Router Architecture: Overview

Two key router functions:

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



Input Port Functions



Physical layer:
bit-level reception

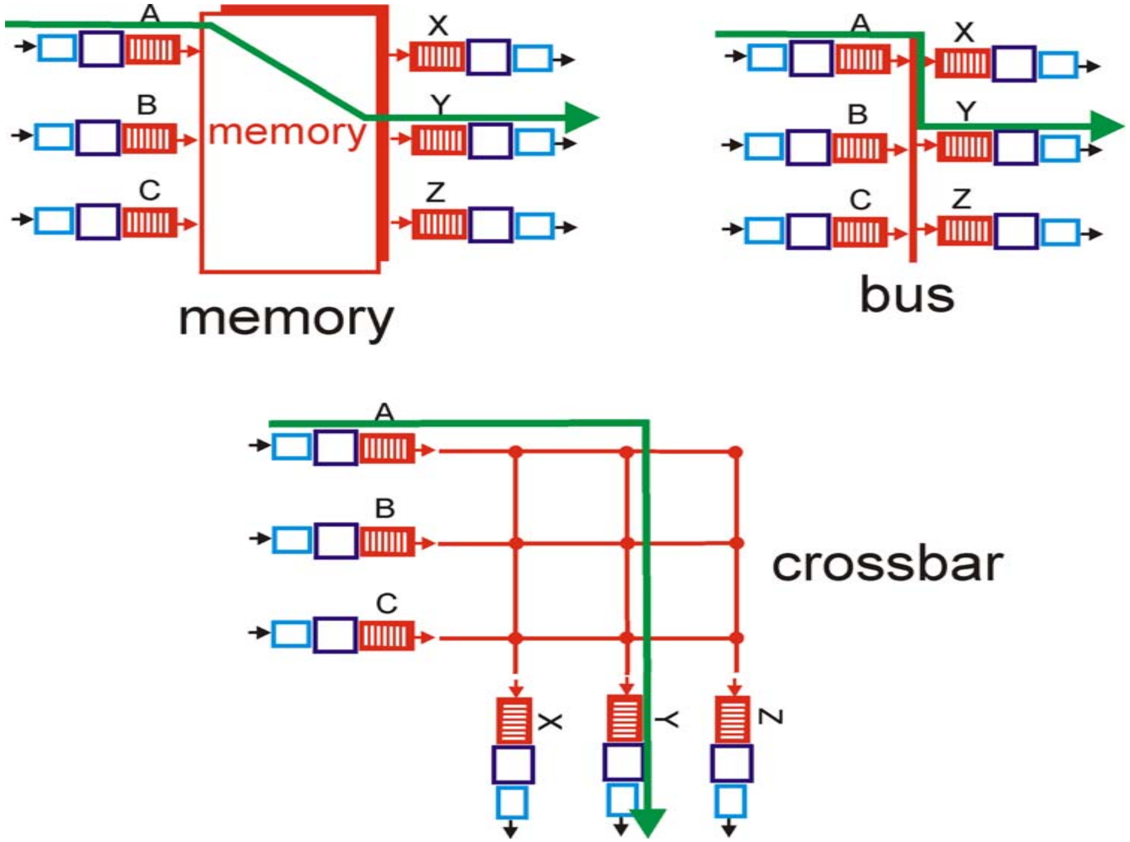
Data link layer:
e.g., Ethernet

Decentralized switching:

- given datagram destination address, 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

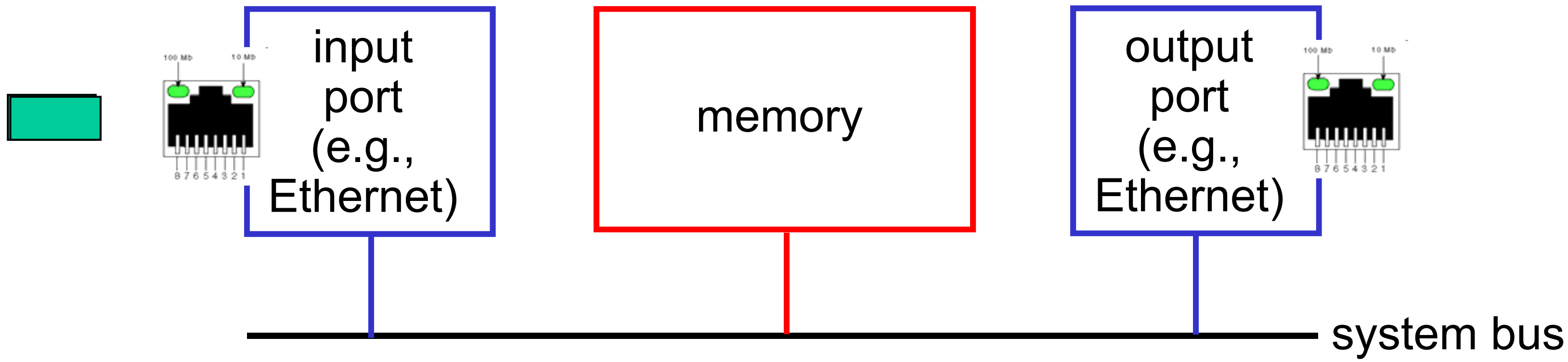
- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transferred from inputs to outputs
  - often measured as multiple of input/output line rates
  - N inputs: switching rate N times line rate is desirable



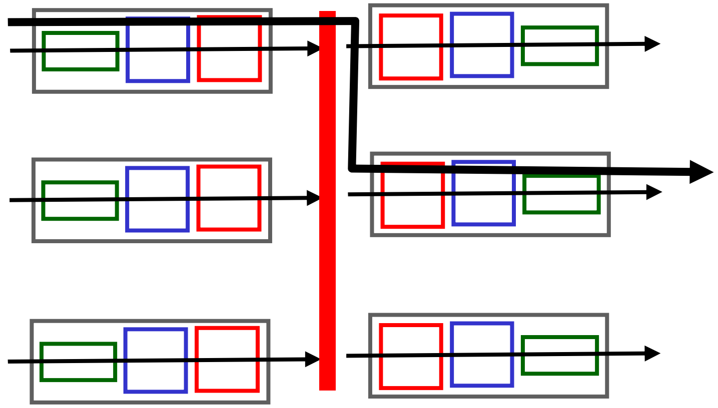
Switching via Memory

First generation routers:

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



Switching via Bus

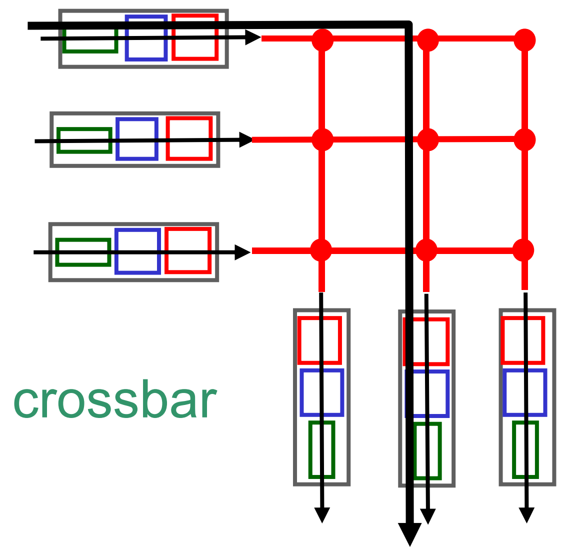


- datagram from input port memory

  to output port memory via a shared bus, one packet at a time
- bus contention: switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers

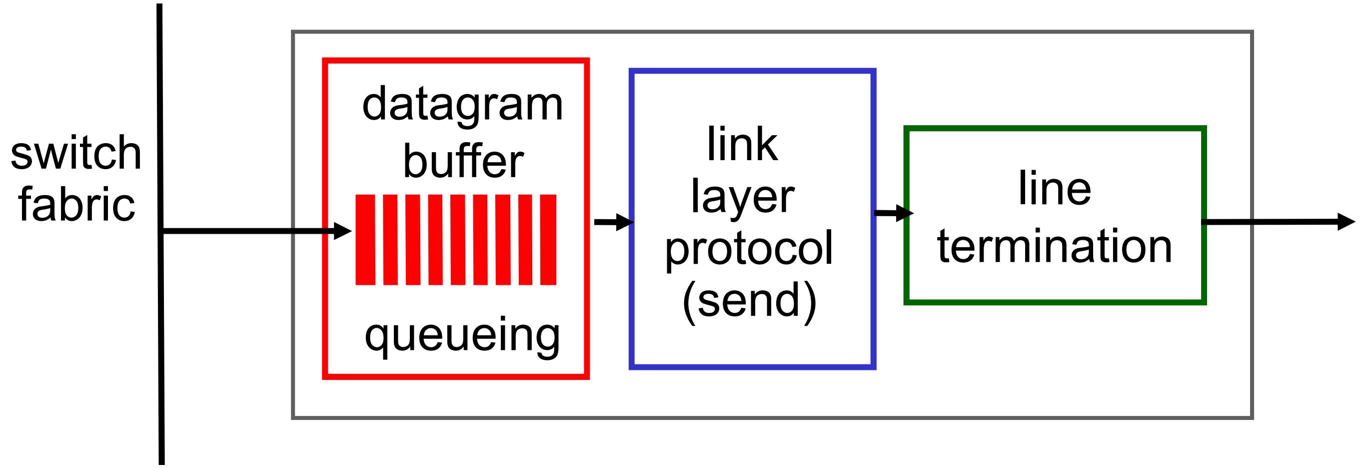
bus

Switching via Interconnection Network



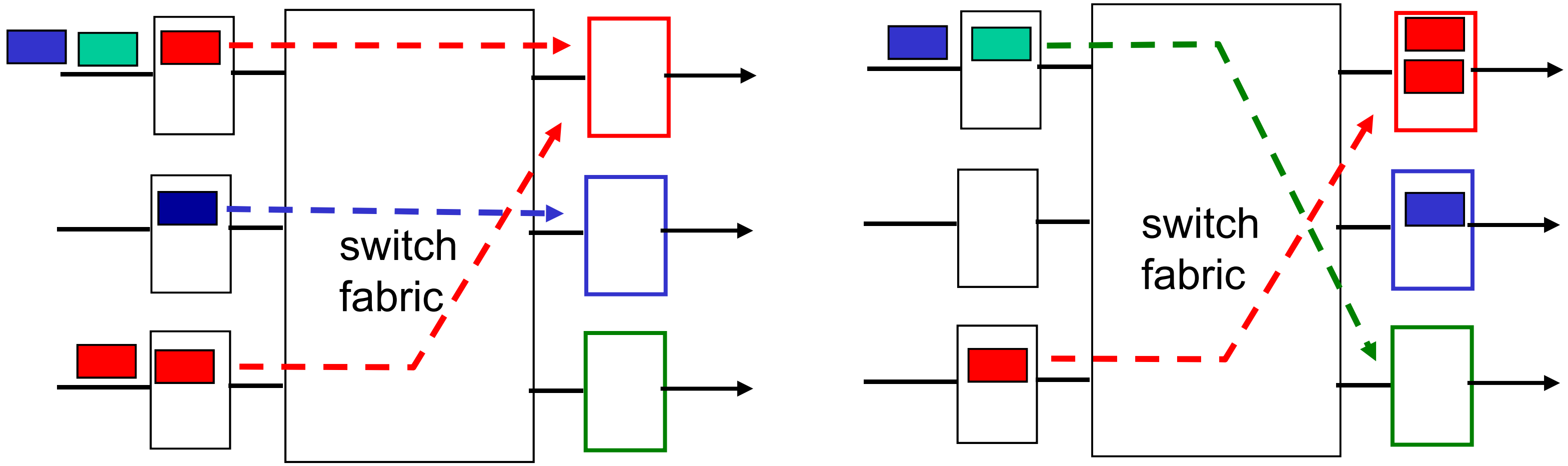
- overcome bus bandwidth limitations
- banyan networks, crossbar, other interconnection networks initially developed to connect processors in multiprocessor
- advanced design: fragmenting datagram into fixed length cells, tag and switch cells through the fabric.
- Cisco 12000: switches 60 Gbps through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

Output Ports



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

Output Port Queueing



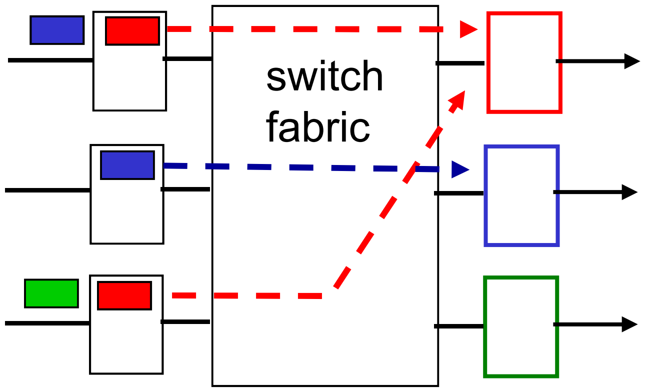
at *t,* packets more from input to output

one packet time later

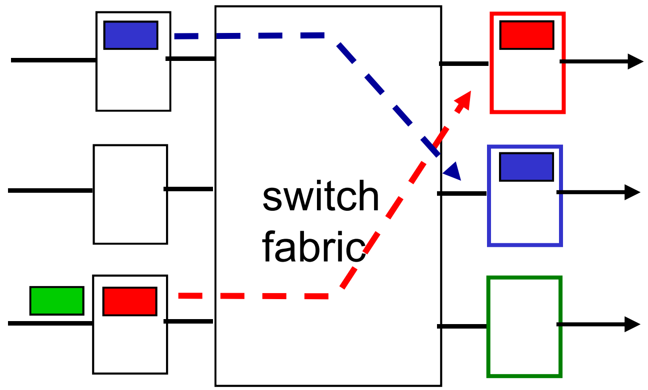
- buffering when arrival rate via switch exceeds output line speed
- *delay due to queueing and loss due to output port buffer overflow!*

Input Port Queuing

- fabric slower *(seldom!)* than input ports combined → queueing may occur at input port
  - *queueing delay and loss due to input buffer overflow!*
- Head-Of-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



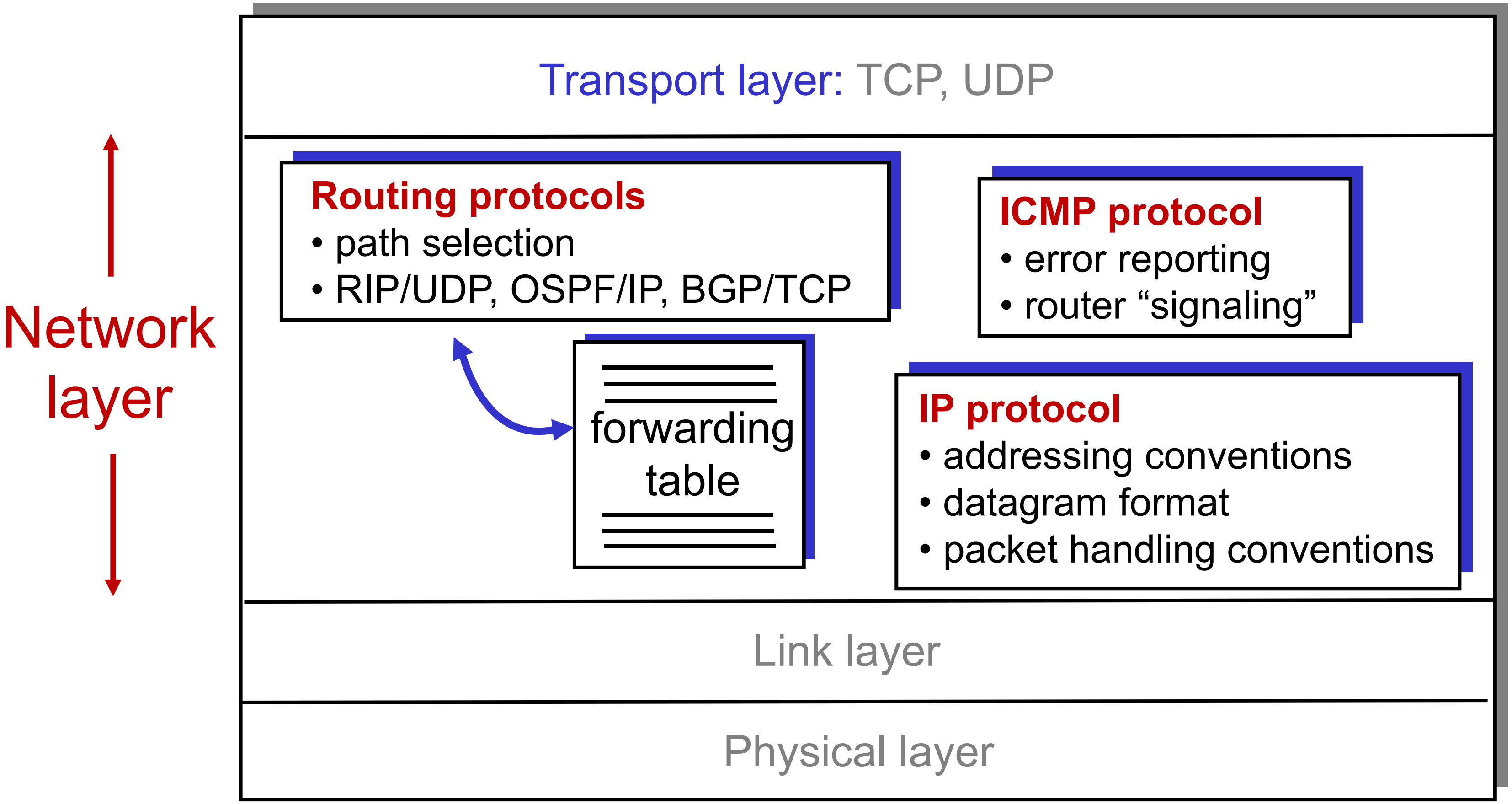
one packet time later: green packet experiences HOL blocking

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:



IP datagram format

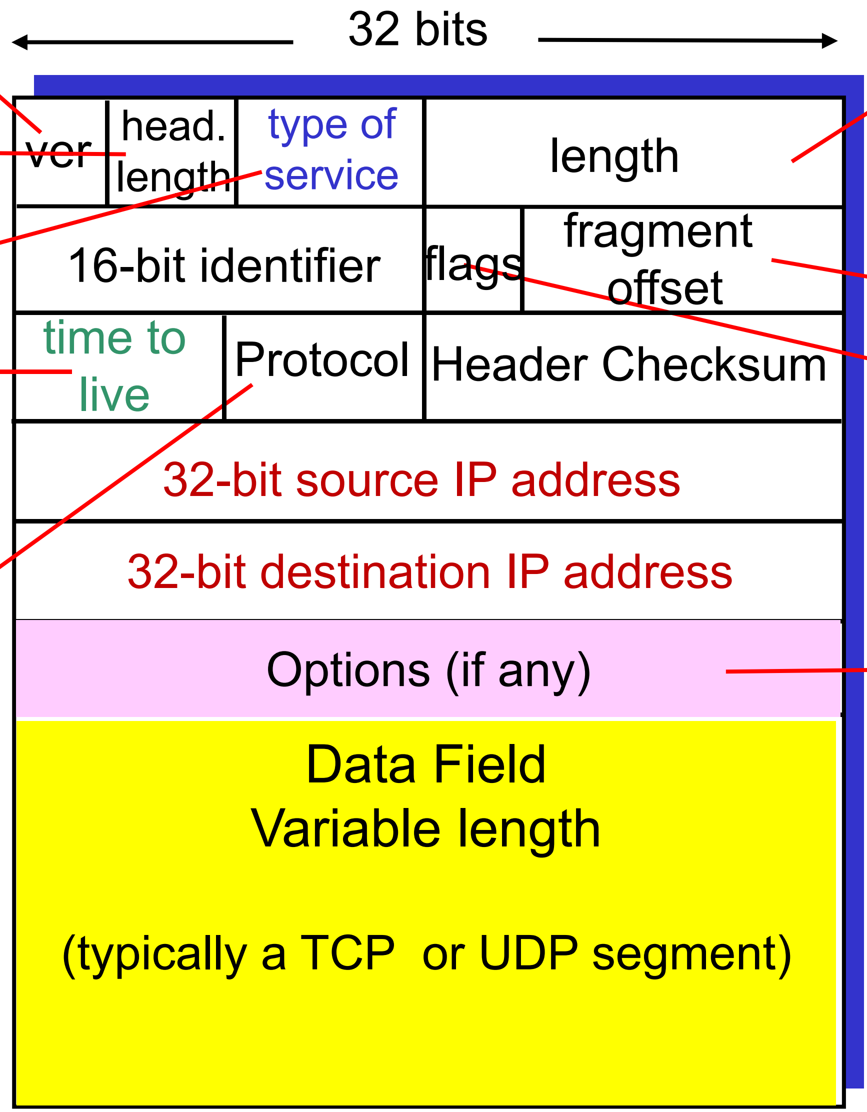
IP protocol version = 4

header length 32-bits blocks, 5 standard)

TOS (priority)

TTL: max number of remaining hops (decremented by one at each router)

Upper layer protocol to deliver payload to
6 for TCP
17 for UDP



total datagram length (bytes)

for fragmentation/ reassembly

Flags (3 bits):
Reserved (0)
DF= don't frag.
MF= more frag.

e.g. timestamp, security label, record route taken, specify list of routers to visit, etc

how much overhead?

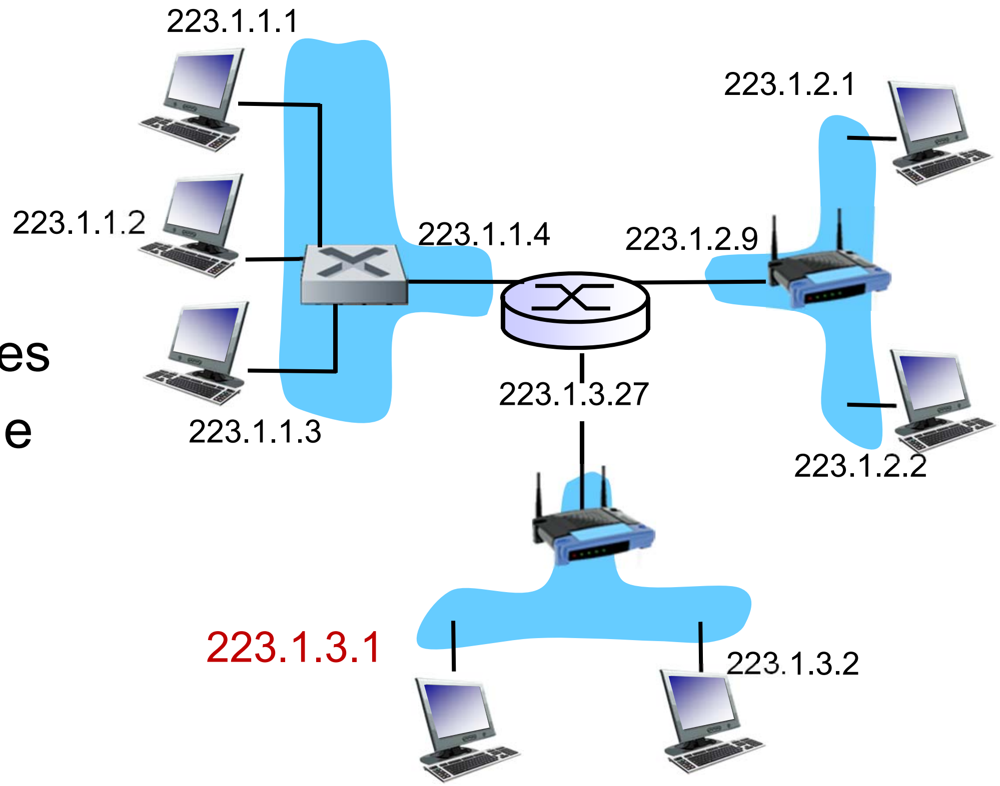
- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

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

- *interface:* connection between host/router and physical link
  - routers typically have multiple active interfaces
  - hosts typically have one active interface (either wired Ethernet or wireless 802.11)
  - IP address associated with each interface
- IP address: 32-bit identifier for host, router *interface*



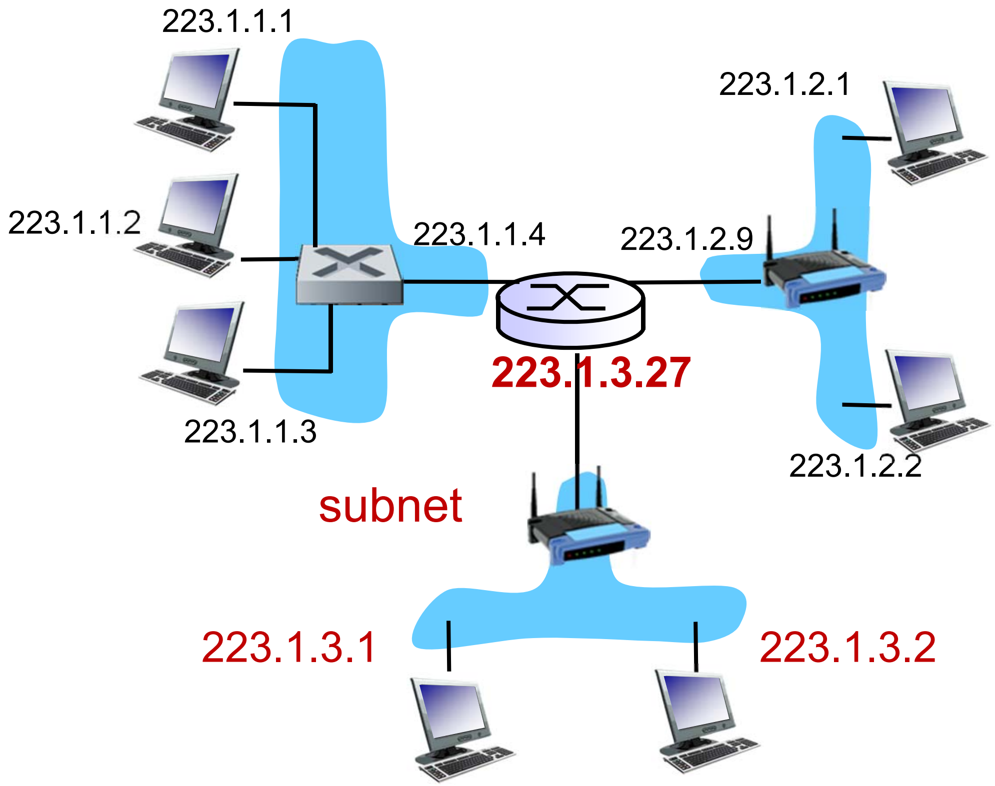
223.1.3.1 = 11011111 00000001 00000011 00000001

223 1 3 1

Dotted Decimal Notation

Subnets

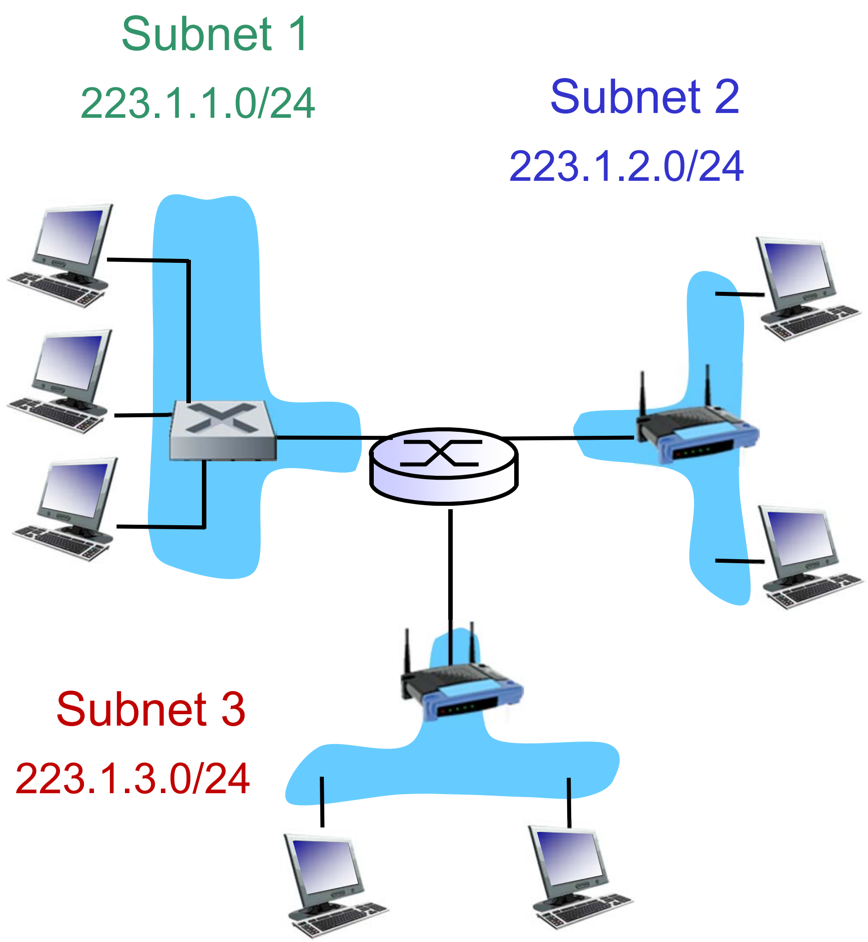
- IP address:
  - subnet part (high order bits)
  - host part (low order bits)
- *What's a subnet ?*
  - device interfaces with same subnet part of IP address
  - Contains hosts that can physically reach each other without intervening router
  - All other hosts are reached by sending datagrams to router interface that works as "default gateway"



network consisting of 3 subnets

Subnets

- How long should the network prefix be?
  - Depends on number of hosts on subnet
  - All hosts in subnet have same subnetwork part of the address.



Typical info given to a host:

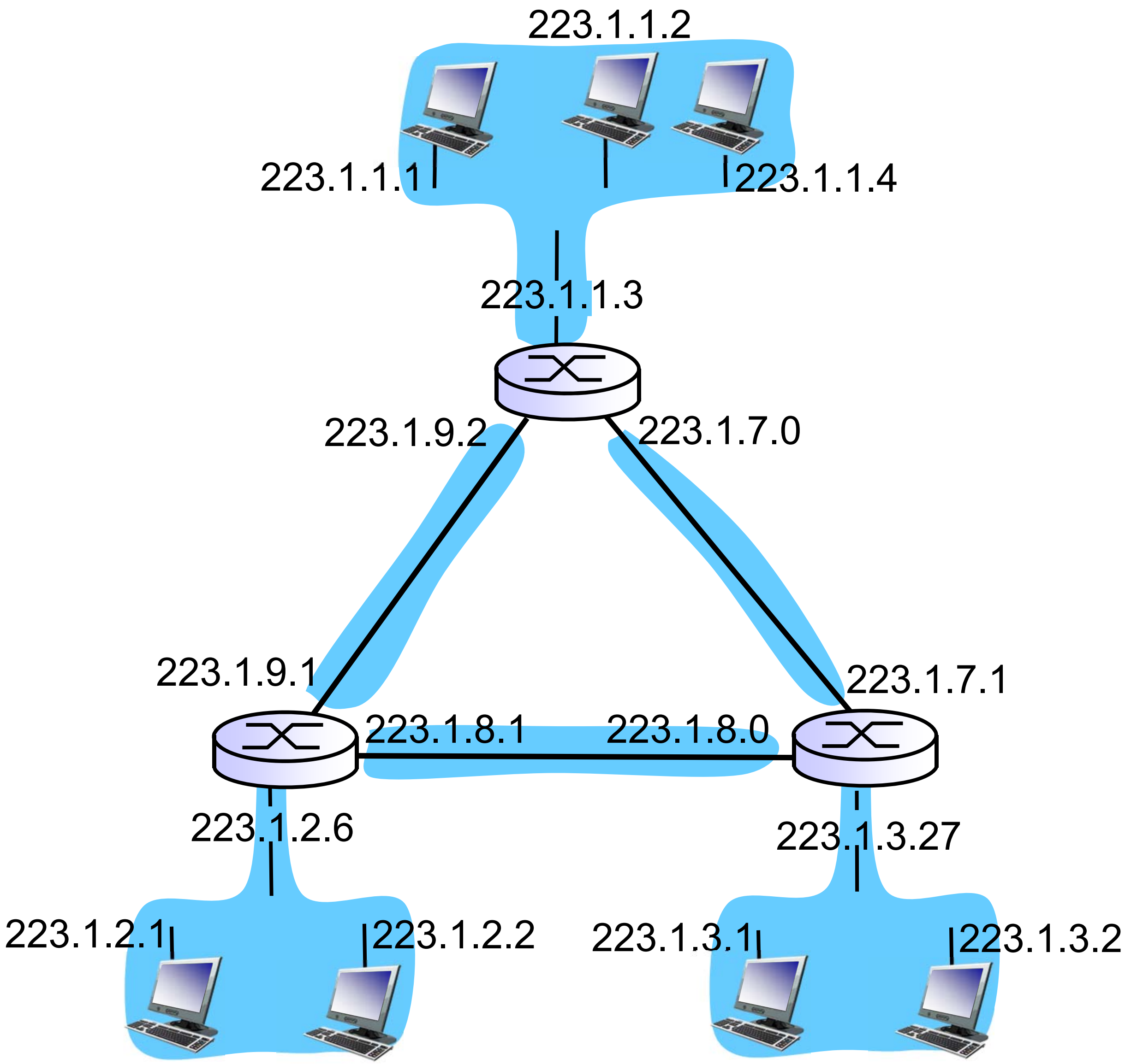
Your address is 223.1.3.1/24
Default route via 223.1.3.27

Subnet mask: /24

24 bits belong to the network
(called length of "CIDR" prefix)

Subnets

How many?



IP Addressing: CIDR

CIDR: Classless Inter-Domain 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

Subnets, masks, calculations

Example subnet: 192.168.5.0/24

	Binary form	Dot-decimal notation
IP address	11000000.10101000.00000101.10000010	192.168.5.130
Subnet mask	11111111.11111111.11111111.00000000 --------24 higher order bits set to 1------	255.255.255.0
Network prefix: *(bitwise AND of address, mask)*	11000000.10101000.00000101.00000000	192.168.5.0
Host part (similar calculation, with eg a "wild card" where the 32 – 24 lower order bits set to 1)	00000000.00000000.00000000.10000010	0.0.0.130

IP Addressing:

Q: How does an ISP get block of addresses?

A: **ICANN**: http://www.icann.org/

Internet **C**orporation for **A**ssigned **N**ames and **N**umbers

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

- These services were originally performed under U.S. Government contract by the Internet Assigned Numbers Authority (IANA) and other entities.
- The IANA now is part of ICANN.

IP Address Allocation:

- **ICANN is responsible for global coordination of the Internet Protocol addressing systems and other naming and numbering standards.**
- **Users are assigned IP addresses by Internet Service Providers (ISPs). ISPs obtain allocations of IP addresses from a Local Internet Registry (LIR) or National Internet Registry (NIR), or from their appropriate Regional Internet Registry (RIR).**

- **There are five RIRs :**

  AfriNIC, Africa
  APNIC, Asia Pacific
  ARIN, Canada, United States, Caribbean and North Atlantic Islands
  LACNIC, Latin America and parts of the Caribbean region
  RIPE NCC, Europe, Russia, Middle East, and Parts of Central Asia

(NIC Network Information Center)

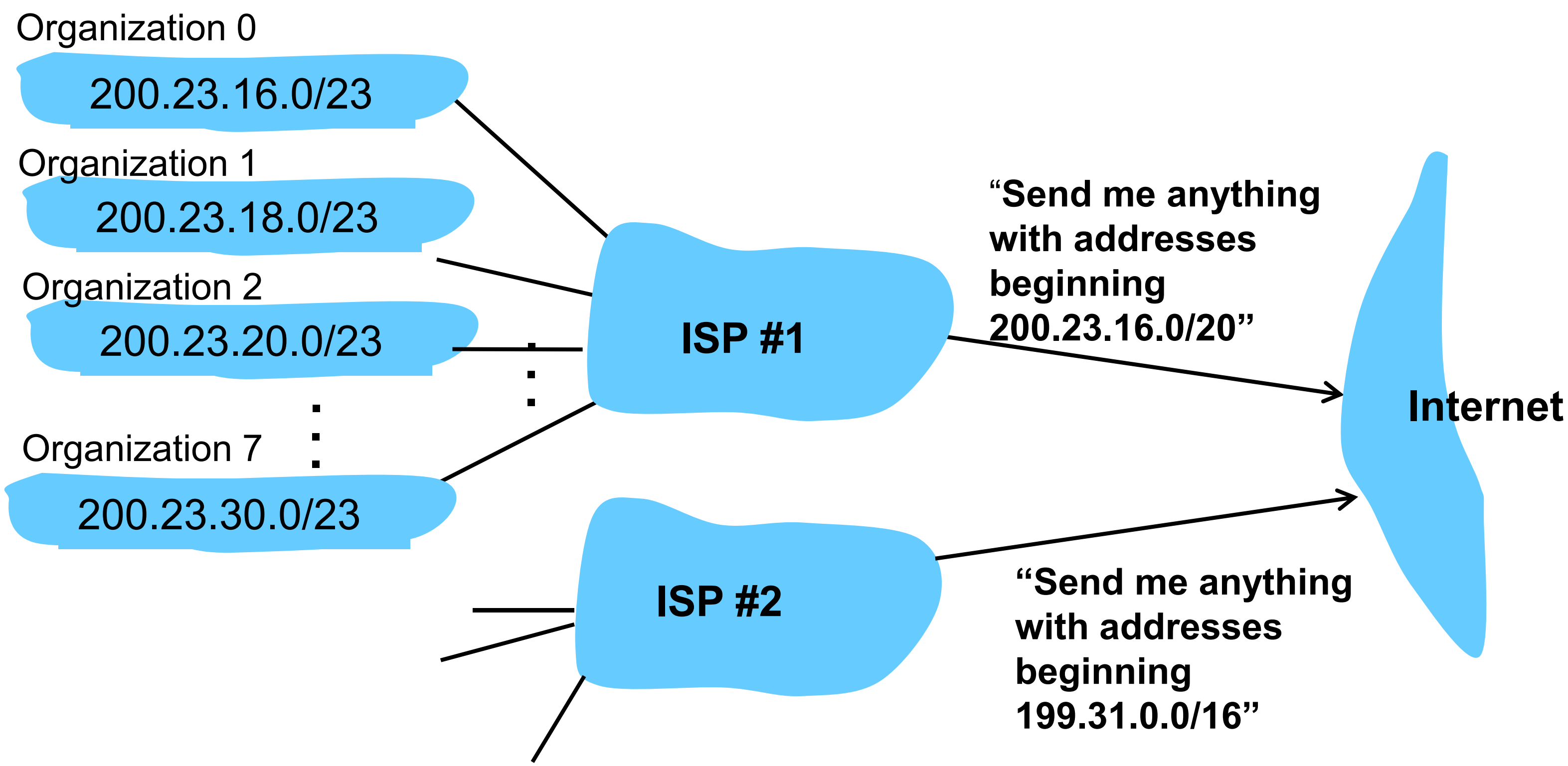
IP addresses: How to get one?

- **Network (subnet) addresses are allocated from a portion of its provider ISP's address space.**

	20 bits	3 bits	9 bits	
ISP's block	11001000 00010111 0001	000	0 00000000	200.23.16.0/20
Organization 0	11001000 00010111 0001	**000**	0 00000000	200.23.16.0/23
Organization 1	11001000 00010111 0001	**001**	0 00000000	200.23.18.0/23
Organization 2	11001000 00010111 0001	**010**	0 00000000	200.23.20.0/23
...	…..		….	….
Organization 7	11001000 00010111 0001	**111**	0 00000000	200.23.30.0/23

Hierarchical Addressing: Route Aggregation

- Hierarchical addressing allows efficient advertisement of routing information
- The “outside” does not need to know about subnets.



Classless Address: example

- An ISP has an address block 122.211.0.0/16
- A customer needs max. 6 host addresses,
- ISP can e.g. allocate: 122.211.**176.208**/29
  - 3 bits enough for host part
- subnet mask 255.255.**255.248**

	Dotted Decimal	Last 8 bits
Network	122.211.176.**208**	**11010**000
1st address	122.211.176.**209**	**11010**001
....	....	....
6th address	122.211.176.**214**	**11010**110
Broadcast	122.211.176.**215**	**11010**111

Reserved

CIDR Address Mask

CIDR Notation	Dotted Decimal	CIDR Notation	Dotted Decimal
/1	128.0.0.0	/17	255.255.128.0
/2	192.0.0.0	/18	255.255.192.0
/3	224.0.0.0	/19	255.255.224.0
/4	240.0.0.0	/20	255.255.240.0
/5	248.0.0.0	/21	255.255.248.0
/6	252.0.0.0	/22	255.255.252.0
/7	254.0.0.0	/23	255.255.254.0
/8	255.0.0.0	/24	255.255.255.0
/9	255.128.0.0	/25	255.255.255.128
/10	255.192.0.0	/26	255.255.255.192
/11	255.224.0.0	/27	255.255.255.224
/12	255.240.0.0	/28	255.255.255.240
/13	255.248.0.0	/29	255.255.255.248
/14	255.252.0.0	/30	255.255.255.252
/15	255.254.0.0	/31	255.255.255.254
/16	255.255.0.0	/32	255.255.255.255

Special IP Addresses

- Localhost and local loopback
  - 127.0.0.1 of the reserved 127.0.0.0 (127.0.0.0/8)
- Private IP-addresses
  - 10.0.0.0 – 10.255.255.255 (10.0.0.0/8)
  - 172.16.0.0 – 172.31.255.255 (172.16.0.0/12)
  - 192.168.0.0 – 192.168.255.255 (192.168.0.0/16)
- Link-local Addresses (stateless autoconfig)
  - 169.254.0.0 – 169.254.255.255 (169.254.0.0/16)

IP addresses: how to get one?

Q: How does *host* get IP address?

- **manually** hard-coded by system admin in a file
  - Windows:

    Control Panel → Network Connections → Local Area Connection → Properties → Internet Protocol (TCP/IP) → Properties
  - UNIX: **/etc/rc.config**
- **DHCP:** Dynamic Host Configuration Protocol (RFC 2131)

  dynamically gets address from a DHCP server

Dynamic Host Configuration Protocol

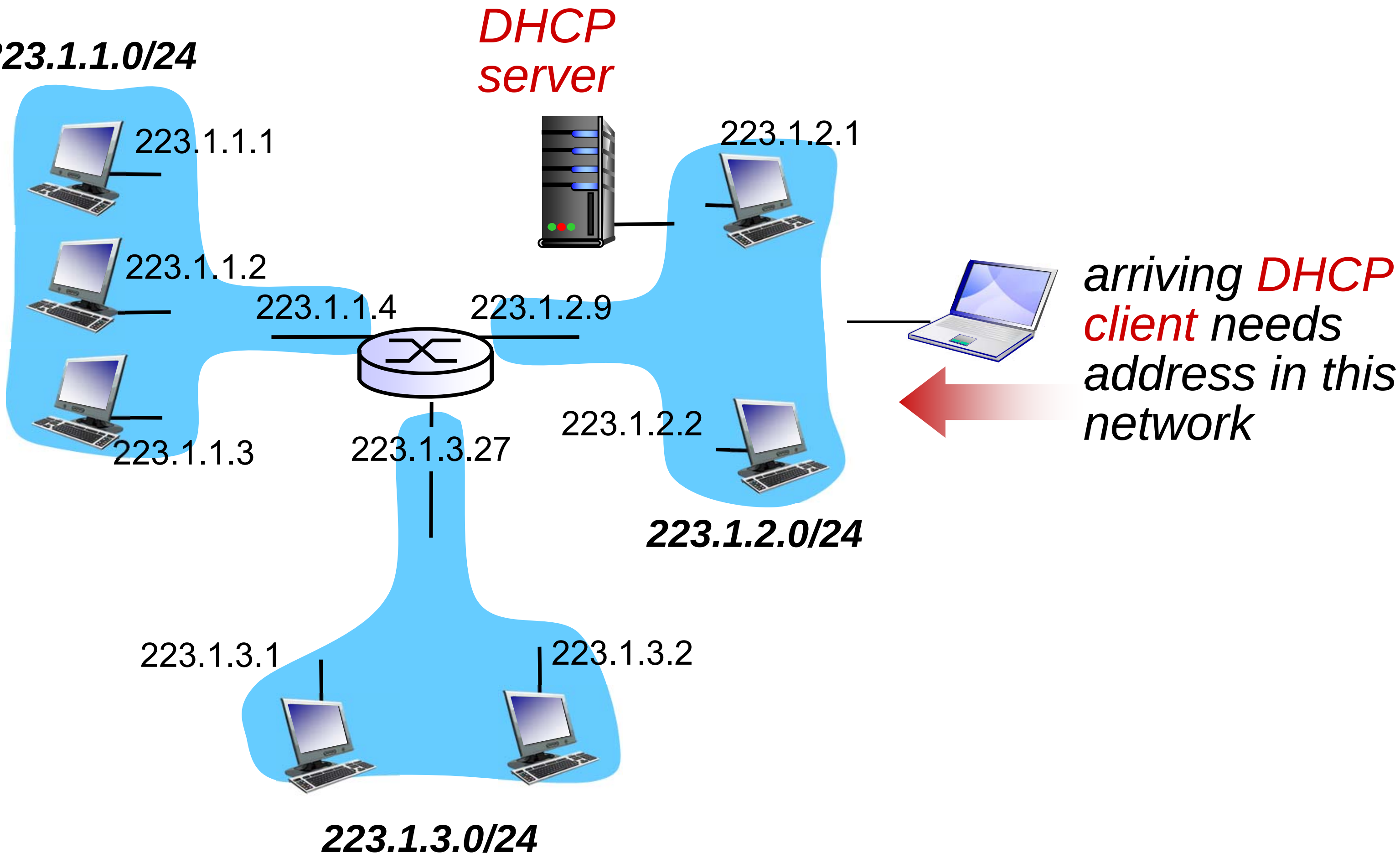
Goal: allows host to *dynamically* obtain its IP address from network server when it joins network.

- Host can renew its lease on address in use
- Allows reuse of addresses (only hold address while connected)
- Support for nomad users who want to join network (short time)

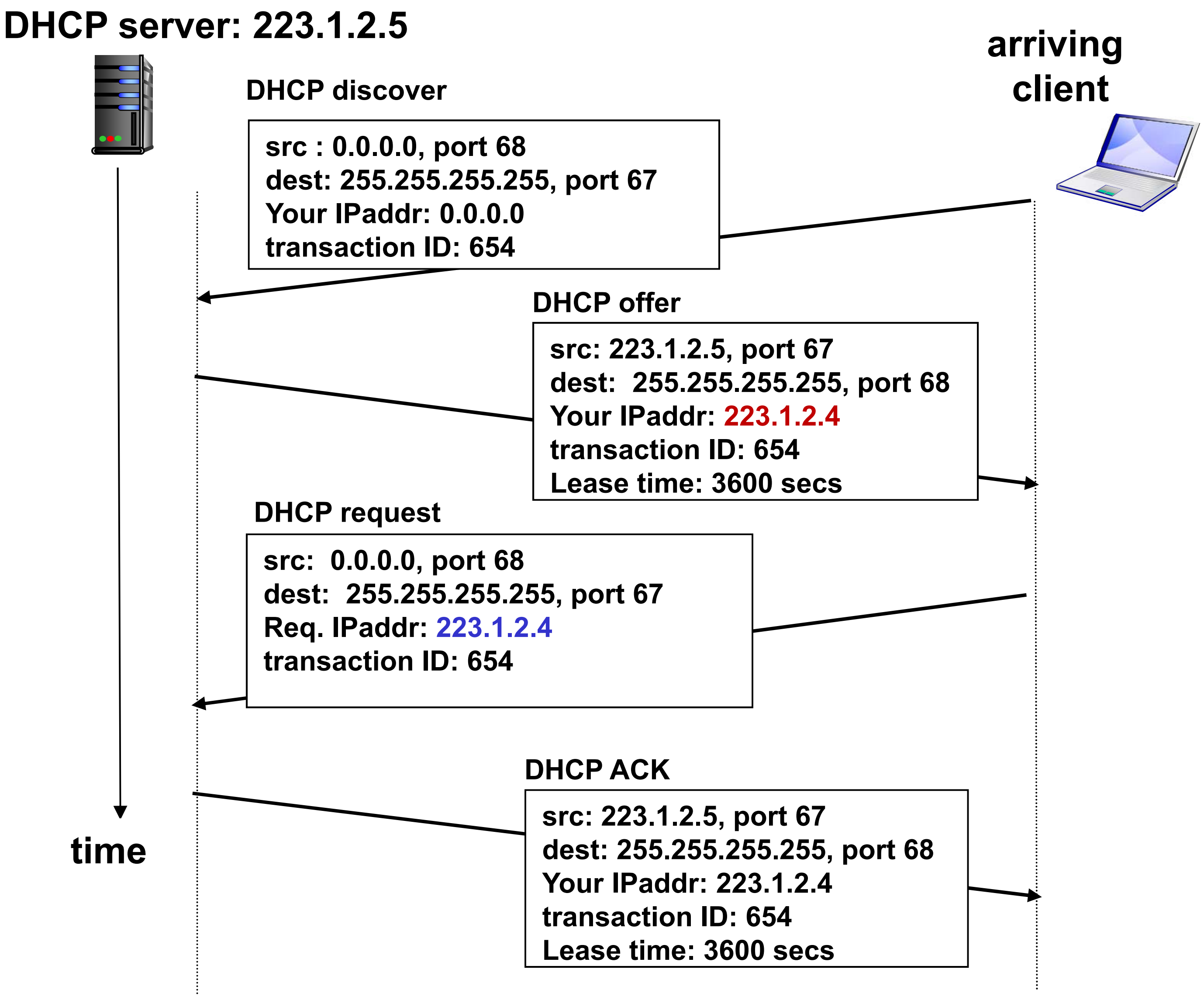
DHCP overview:

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

DHCP client-server scenario



DHCP client-server scenario

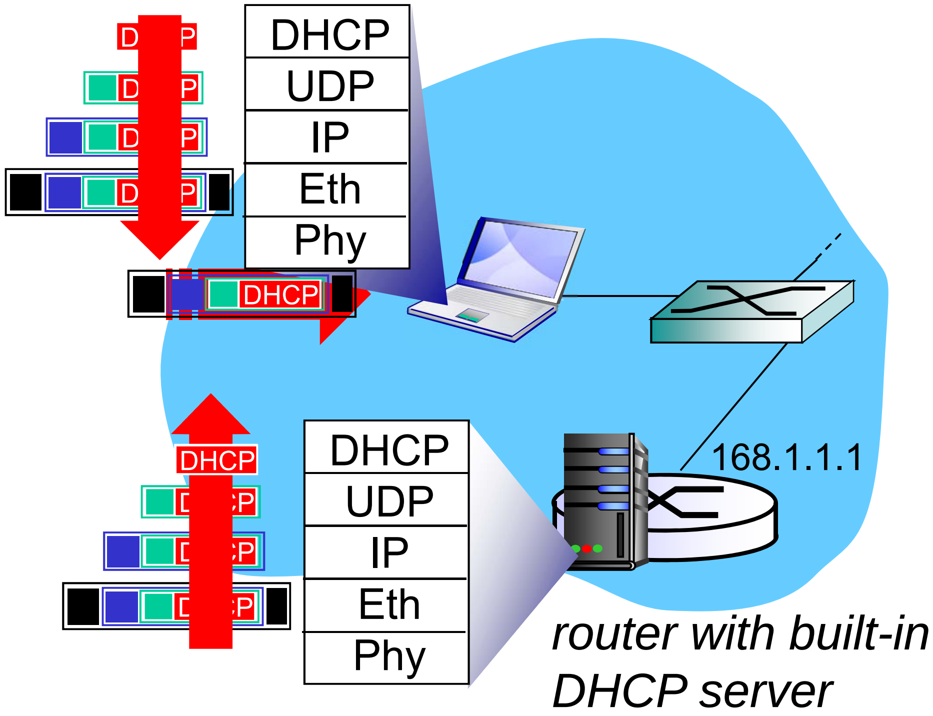


DHCP: more than an IP address

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

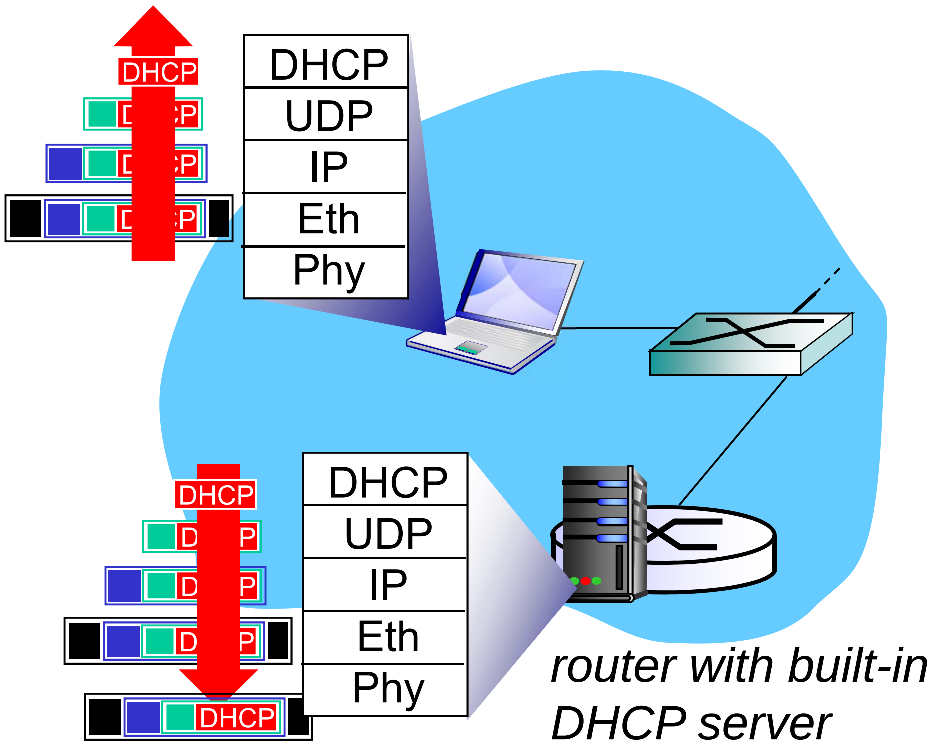
- address of first-hop router (default gateway)
- name and IP address of DNS sever
- network mask (indicating network portion of address)

DHCP: example



- Connecting laptop needs:
  - its IP address, subnetmask
  - address of first-hop router
  - address of DNS server
- DHCP request encapsulated in UDP, encapsulated in IP, encapsulated in 802.3 Ethernet MAC frame
- Ethernet frame broadcast (FFFFFFFFFFFF) on LAN, received at router running DHCP server

DHCP: example



- DHCP server formulates DHCP ACK containing client's IP address, IP address of first-hop router for client, IP address of DNS server

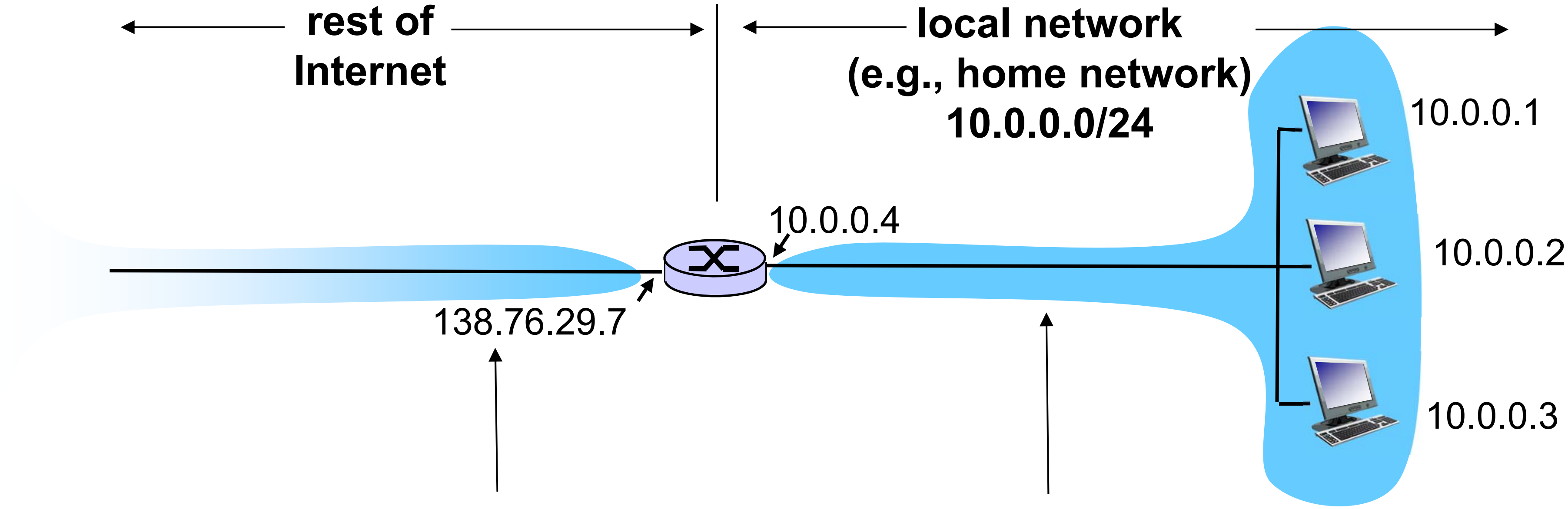
- encapsulation of DHCP server, frame forwarded to client

- client now knows its IP address, IP address of DNS server, IP address of its first-hop router

NAT: Network Address Translation

- **Router with NAT can translate network addresses**
  - Many internal (private) addresses translated to one (or few) external (global) addresses.
- **Gives freedom when configuring internal network**
  - fewer addresses needed from ISP or just one IP global 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
  - can hide internal structure (devices not visible by outside world, a security plus)
- **Internal network should use non-routable (private) addresses reserved for this purpose (RFC 1918)**
  - 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16

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 or destination (as usual)

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 address

- *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

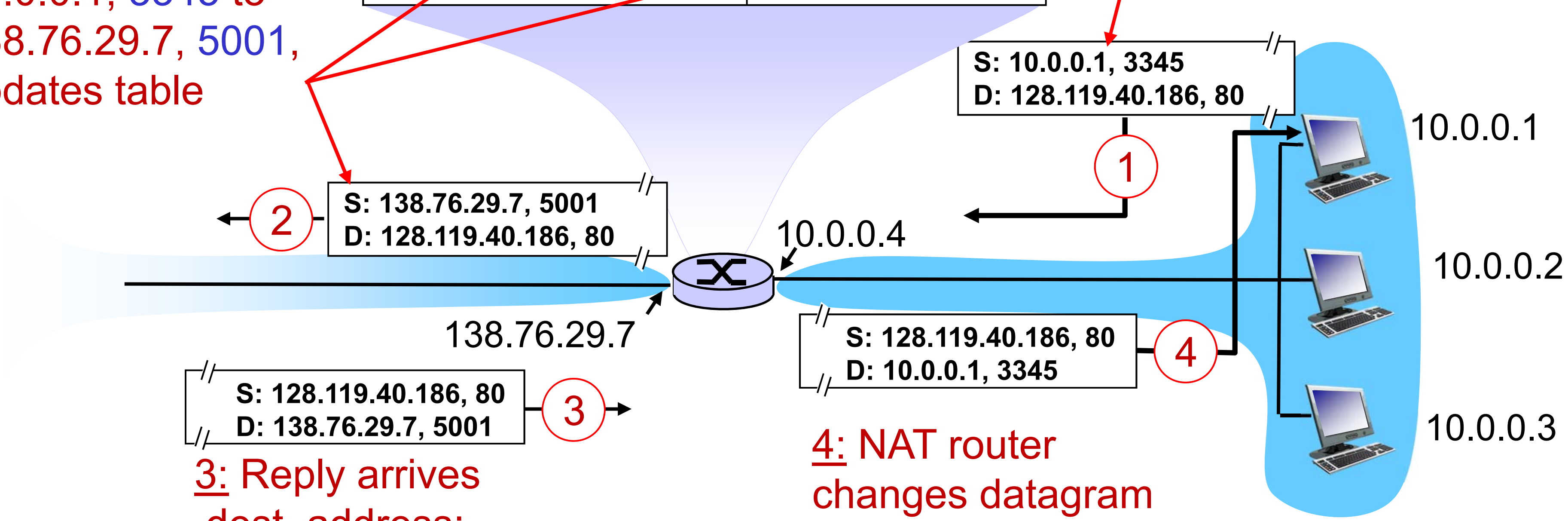
NAT: Network Address Translation

NAT translation table

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

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

1: host 10.0.0.1 sends datagram to 128.119.40.186, 80



3: Reply arrives dest. address: 138.76.29.7, 5001

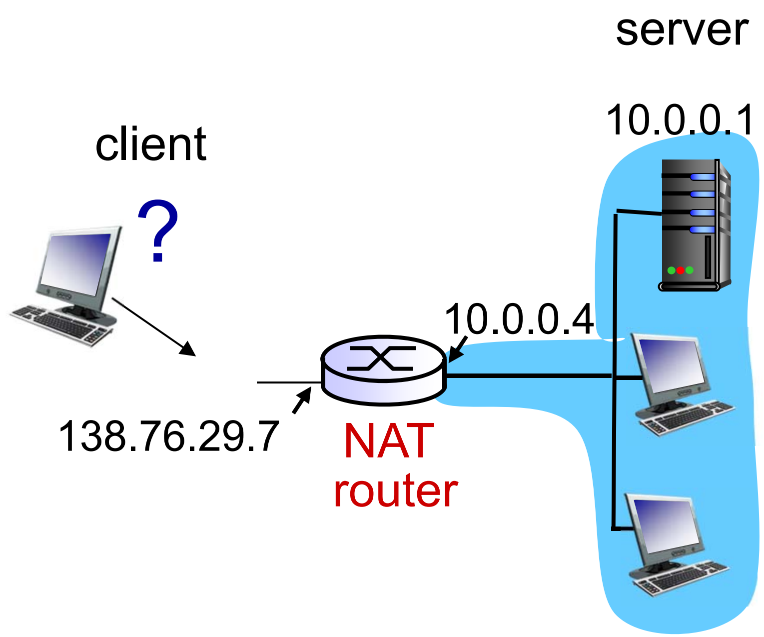
4: NAT router changes datagram dest addr from 138.76.29.7, 5001 to 10.0.0.1, 3345

NAT: Network Address Translation

- 16-bit port-number field:
  - 65,000 simultaneous connections with a single WAN-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 application designers, e.g., P2P applications
  - address shortage should instead be solved by IPv6 ….

NAT: Traversal Problem

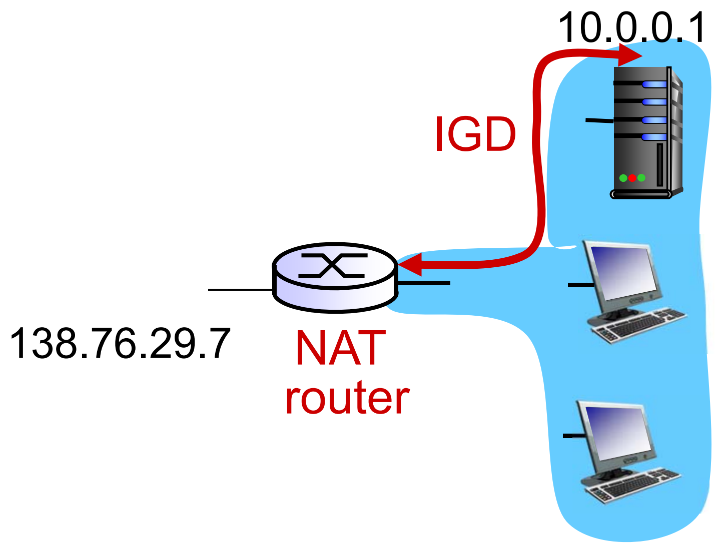
- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible NATed address: 138.76.29.7
- *solution1:* statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (123.76.29.7, port 2500) always forwarded to 10.0.0.1 port 2500



NAT: Traversal Problem

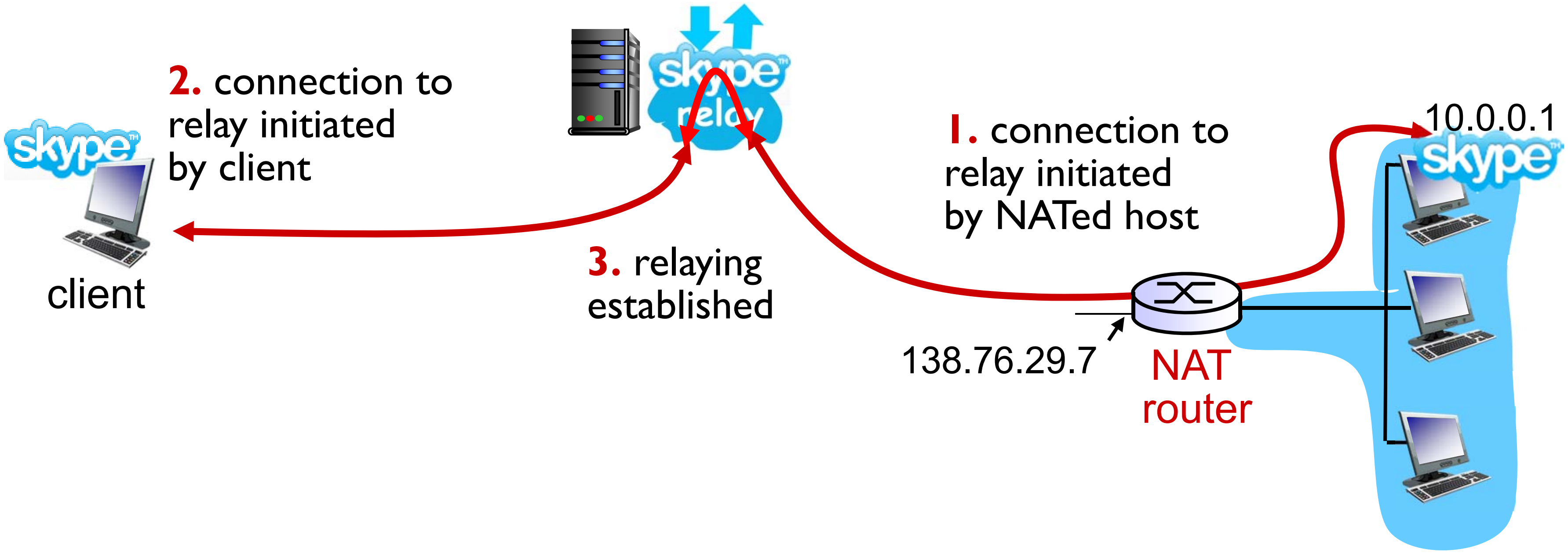
- *solution 2:* Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATed host to:
  - learn public IP address (138.76.29.7)
  - add/remove port mappings (with lease times)

i.e., automate static NAT port map configuration



NAT: Traversal Problem

- *solution 3:* relaying (used in p2p)
  - NATed host establishes connection to relay
  - external client connects to relay
  - relay bridges packets between two 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*

ICMP: Internet Control Message Protocol

- **Control and error messages** from network layer.
- All IP implementations must have ICMP support.
- ICMP messages carried in IP datagrams
- used by hosts & routers to communicate network-level control information and error reporting
  - Error reporting: e.g., unreachable network, host, ..
  - Example: (used by **ping** command)
    - Sends ICMP echo request
    - Receives ICMP echo reply
- Any ICMP error message may **never** generate a new one.

ICMP: message format

- ICMP message:
  - type field: 1 byte
  - code field: 1 byte
  - Checksum: 2 bytes
  - 0s, (ID + Seq. #) or other fields: 4 bytes
  - Optional data or when error reporting message always include header of IP datagram causing error plus first 8 bytes of its payload

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
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 destination
  - First has TTL =1
  - Second has TTL=2, etc.
  - Unlikely port number
- When datagram sent with TTL = n arrives to **n**:th router:
  - TTL becomes 0
  - Router discards datagram
  - Router sends to source an ICMP message “TTL expired” (type 11, code 0)
  - Message is carried in IP datagram with the router IP address as source

- When ICMP message arrives, source measures RTT
- Traceroute does this 3 times

Stop criteria

- UDP segment eventually arrives at destination host
- Destination returns ICMP message “destination port unreachable” (type 3, code 3)
- When source gets this ICMP 3 times, traceroute 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: motivation

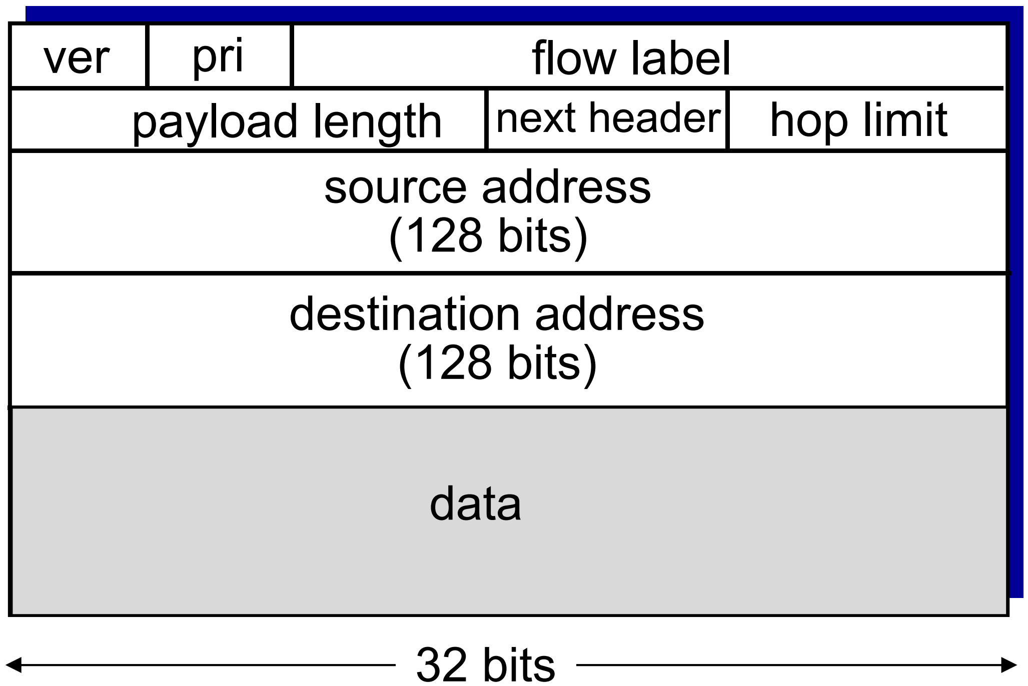
- *initial motivation:* 32-bit address space was about to be completely allocated.
- additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

IPv6 datagram format:

- fixed-length 40 byte header
- no fragmentation allowed
- *128-bit addresses* ($2^{128} = 10^{38}$ numbers)
- Standard subnet size: 64 bits

IPv6 datagram format

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



Other changes from IPv4

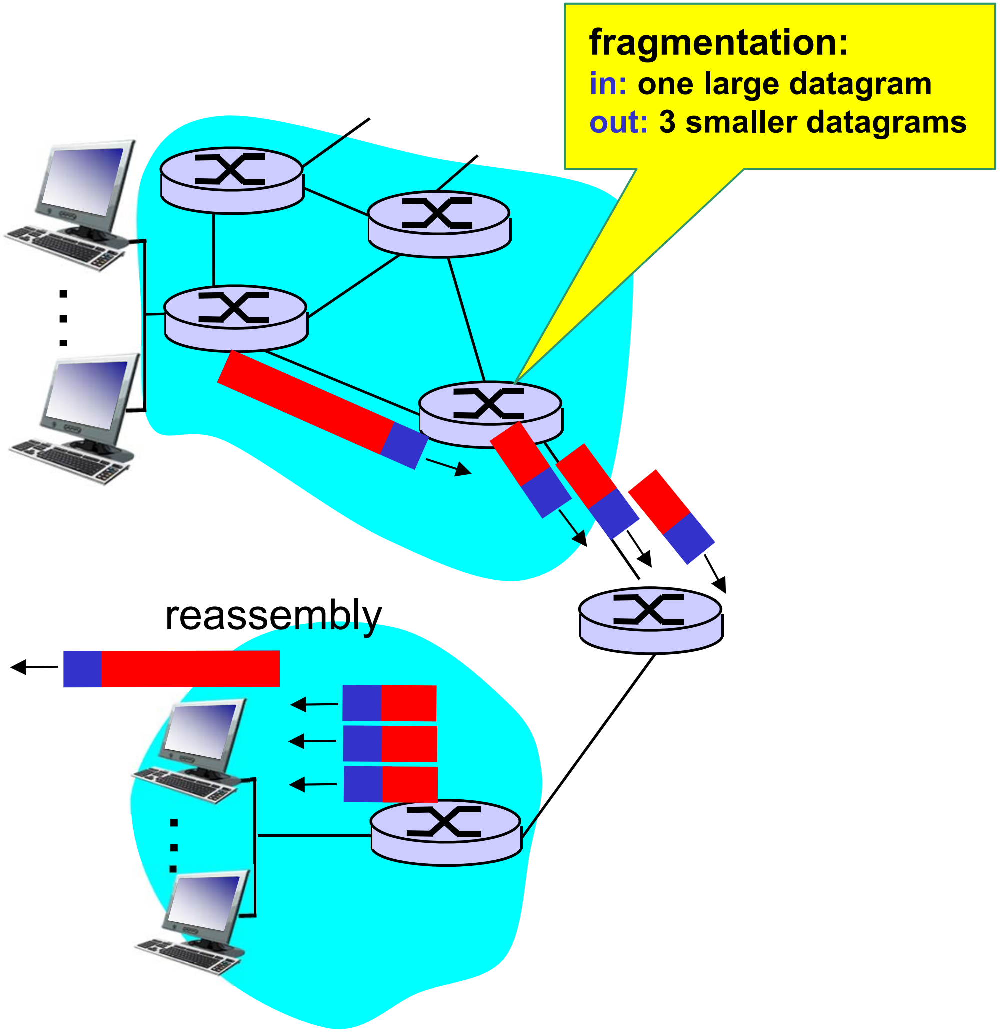
- *checksum*: removed entirely to reduce processing time at each hop
- *options:* allowed, but outside of header, indicated by “Next Header” field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. “Packet Too Big”
  - Neighbor and router discovery
  - multicast group management functions

More slides

- IPv4 Fragmentation
- Datagram Forwarding Table
- Getting a datagram from source to destination
- IPv6-IPv4 Tunneling

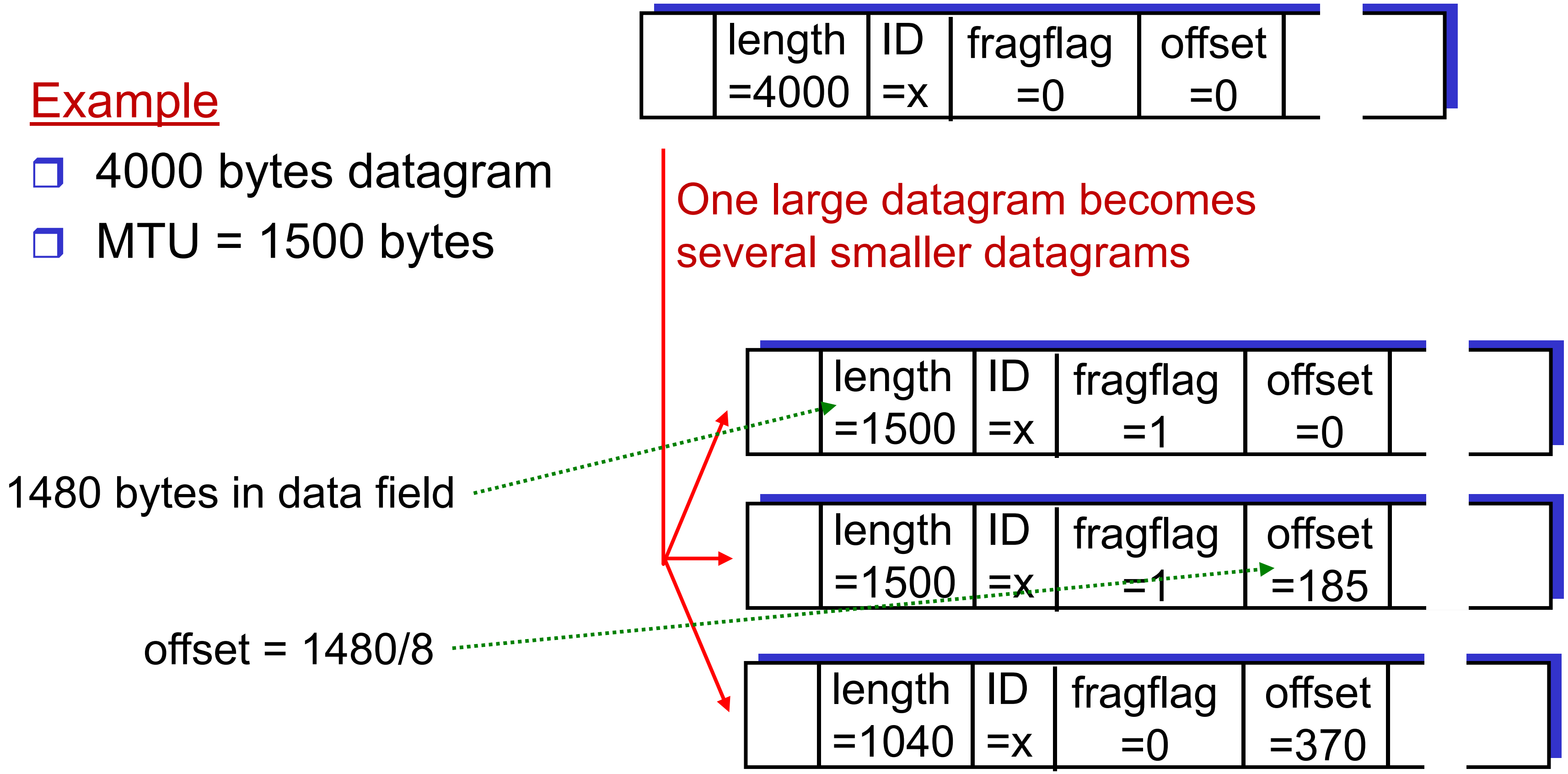
IP Fragmentation & Reassembly

- **MTU** (Maximum Transmission Unit) largest possible data amount carried by link-level frame.
  - different link types, different MTUs
- large IP datagrams will be divided ("fragmented") by host or router
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header fields used to identify, order related fragments
    - More Fragments **bit**
    - **Datagram ID**
    - **Fragment Offset (in 8-byte units)**



IP Fragmentation

Example
- 4000 bytes datagram
- MTU = 1500 bytes



Datagram forwarding table

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

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

Longest prefix matching

longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address (more on this coming soon)

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?

Getting a datagram from source to dest.

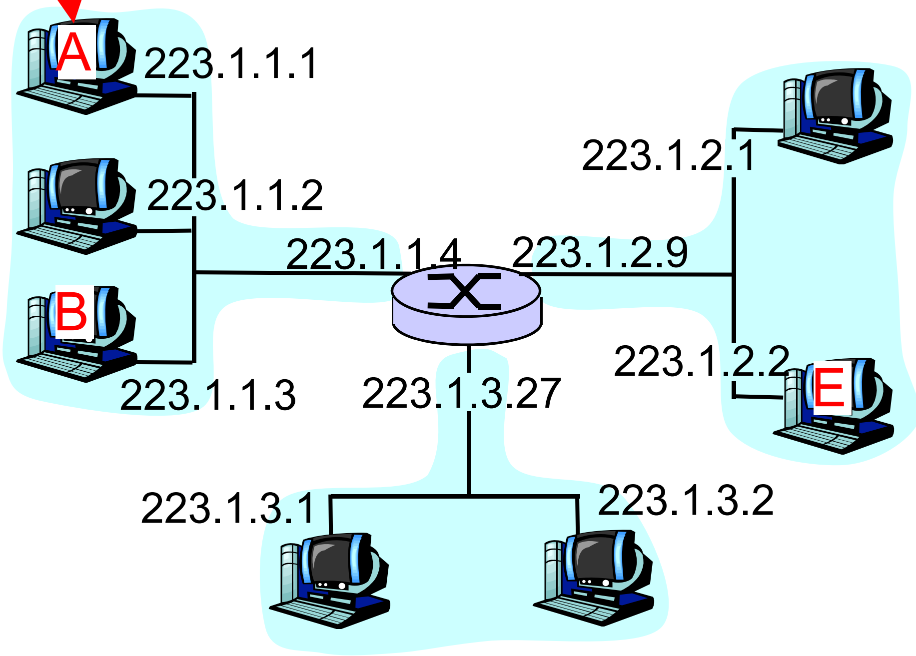
IP datagram:

misc fields	source IP addr	dest IP addr	data

- **Payload in datagram remains unchanged, as it travels source to destination**
- addr fields of interest here

forwarding table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



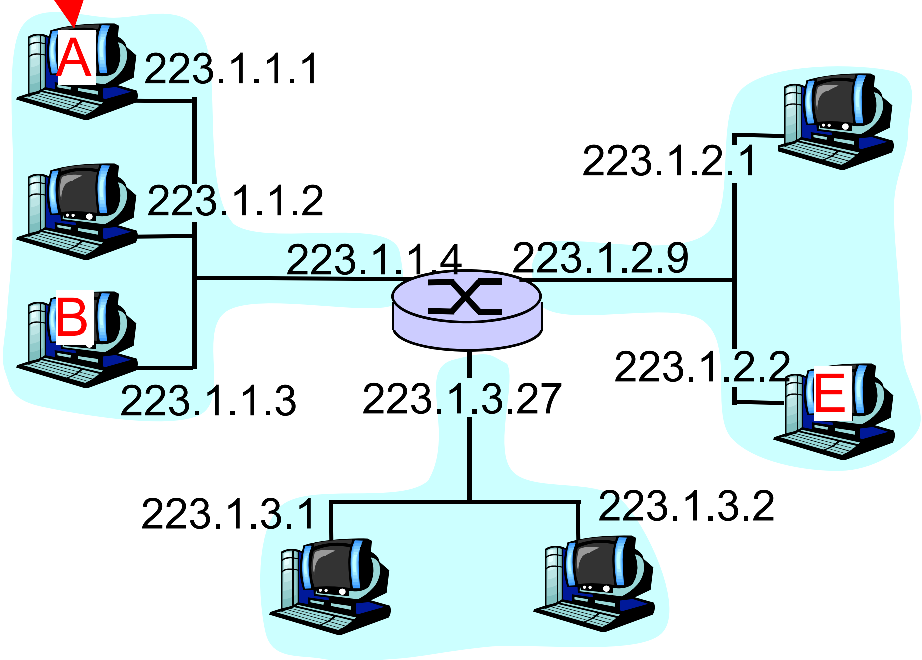
Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.1.3	data

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

Starting at A, given IP datagram addressed to B:

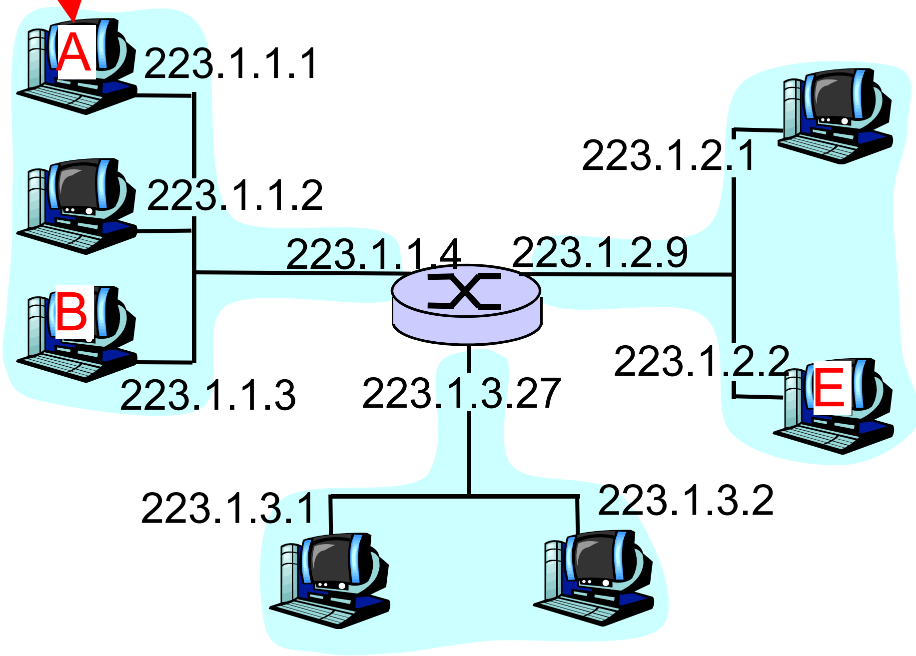
- look up net. address of B
- find B is on same net. as A (B and A are directly connected)
- link layer will send datagram directly to B (inside link-layer frame)



Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.2.3	data

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



Starting at A, dest. E:

- look up network address of E
- E on *different* network
- routing table: next hop router to E is 223.1.1.4
- link layer is asked to send datagram to router 223.1.1.4 (inside link-layer frame)
- datagram arrives at 223.1.1.4
- continued…..

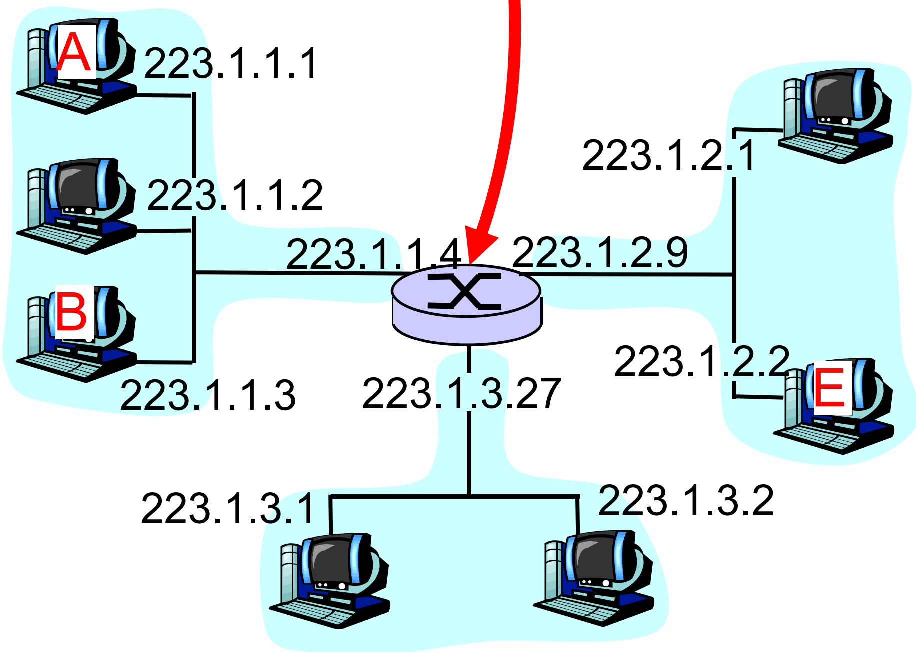
Getting a datagram from source to dest.

misc fields	223.1.1.1	223.1.2.3	data

Dest. network	next router	Nhops	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27

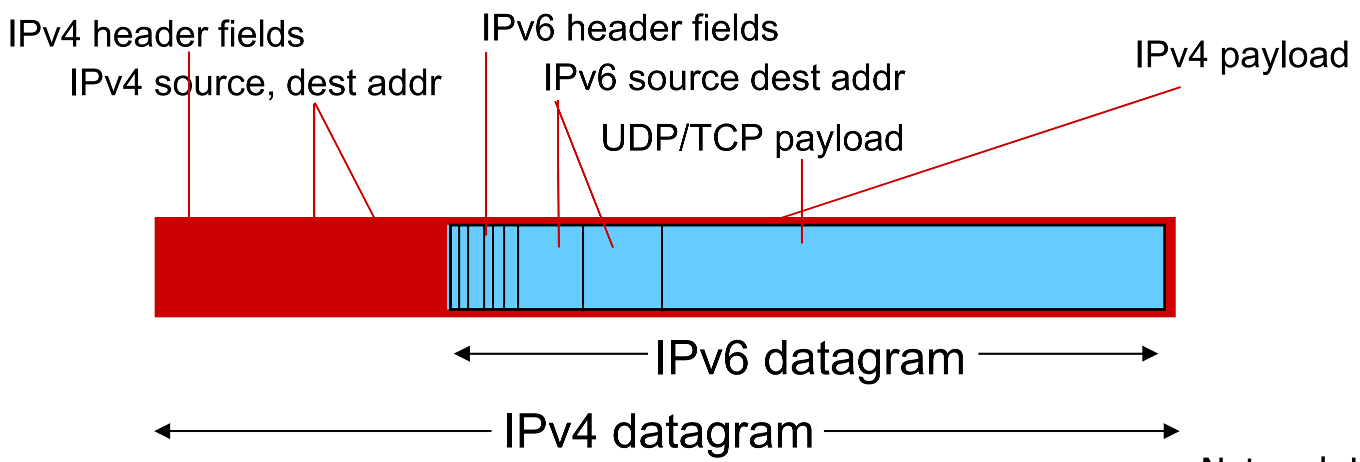
Arriving at 223.1.4, destined for 223.1.2.2

- look up network address of E
- E on *same* network as router's interface 223.1.2.9
  - router, E directly attached
- link layer sends datagram to 223.1.2.2 (inside link-layer frame) via interface 223.1.2.9
- datagram arrives at 223.1.2.2!!! (hooray!)

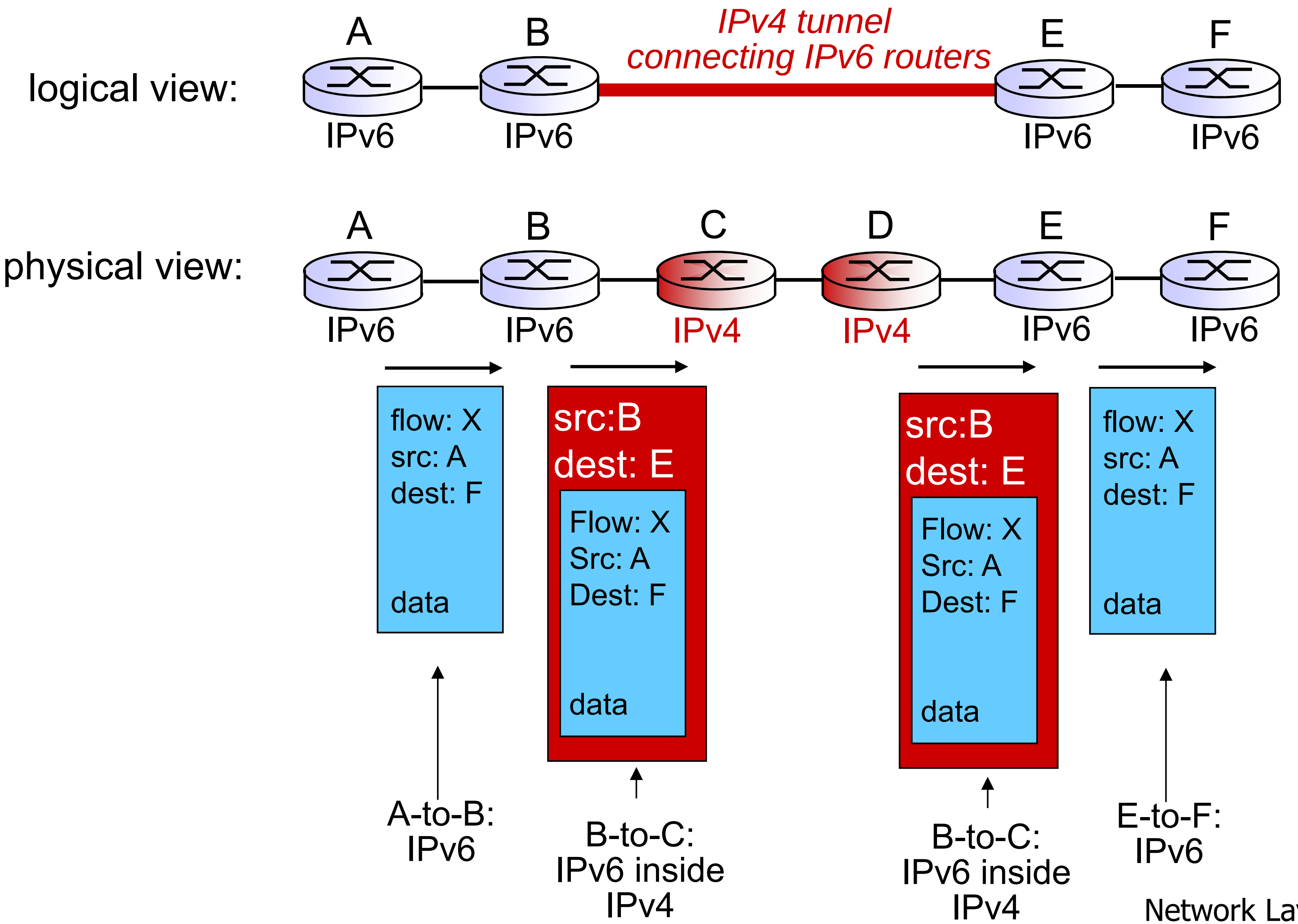


Transition from IPv4 to IPv6

- 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 (6in4 – static tunnel)



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*