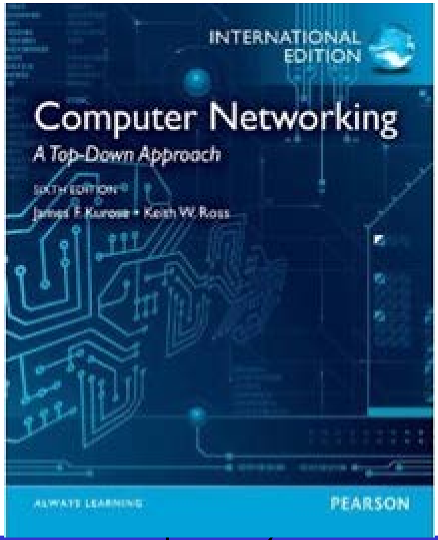
Chapter 4: Network Layer
Part A

Course on Computer Communication and Networks, CTH/GU

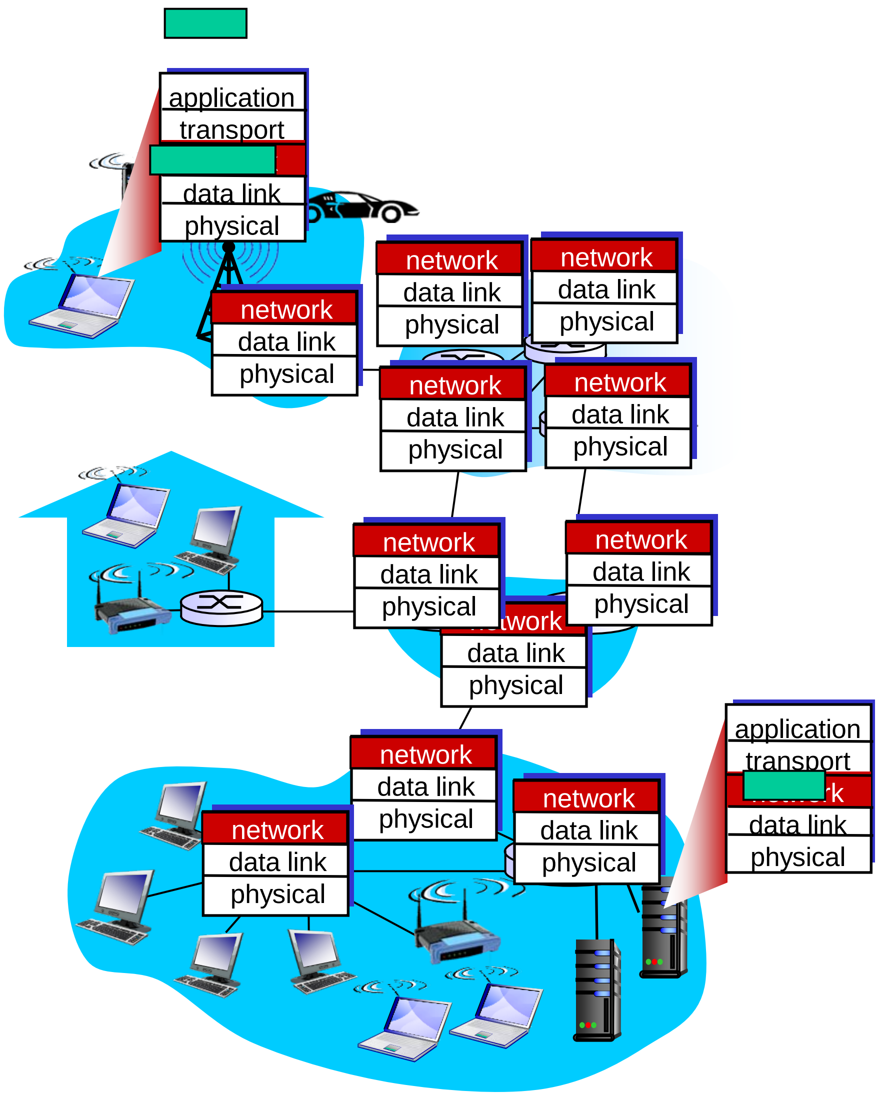
The slides are adaptation of the slides made available by the authors of the course's main textbook



Network layer

Consider transporting a segment from sender to receiver

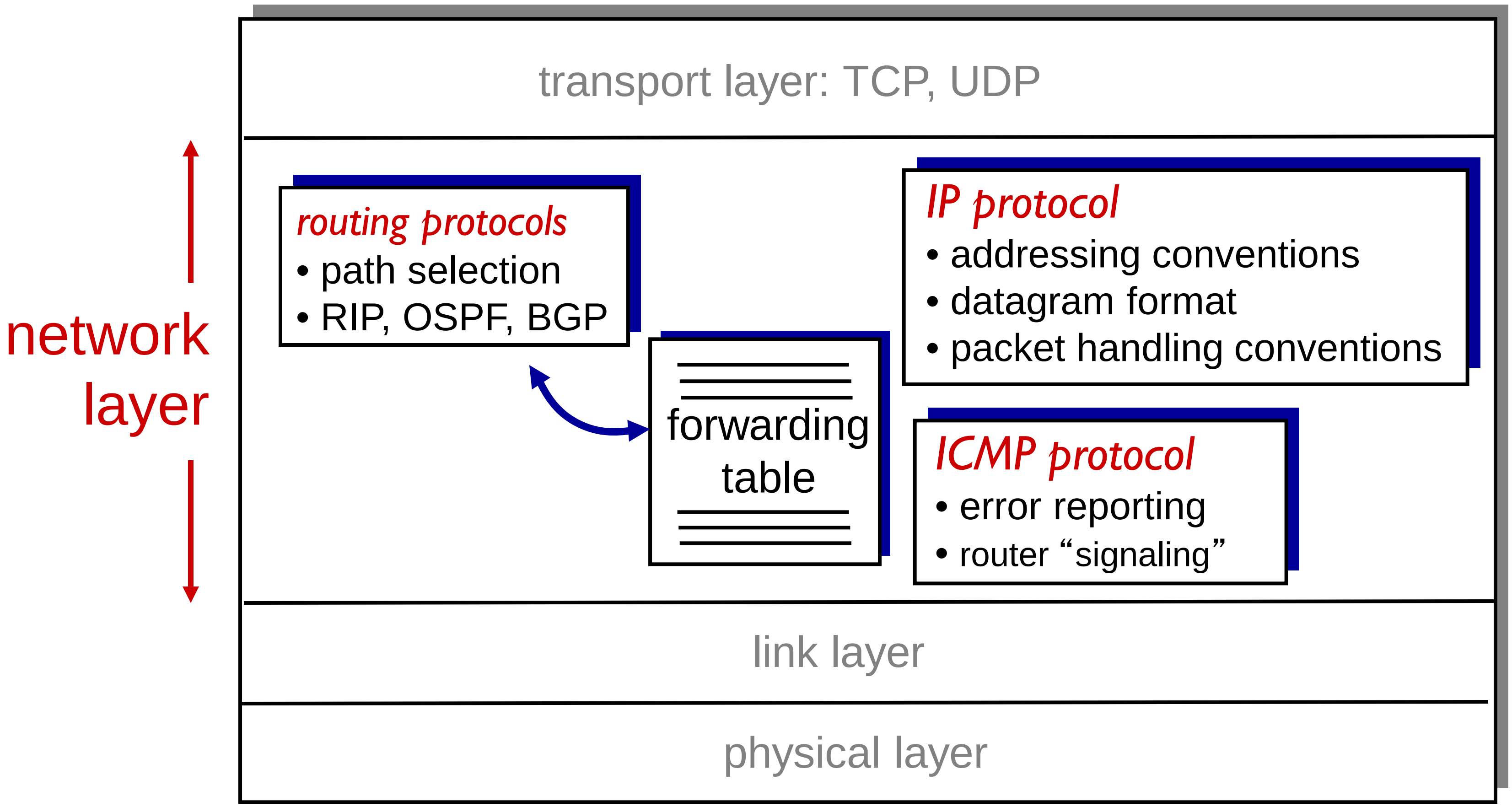
- sending side: encapsulates segments into datagrams
- receiving side: delivers segments to transport layer



- network layer protocols in *every* host, router
- router examines header fields in all datagrams passing through it

The Internet network layer

host, router network layer functions:



Roadmap

- Understand principles of network layer services:
  - **forwarding** versus **routing**
  - network layer **service models**
  - how a **router works**
- The **Internet Network layer**
- Routing

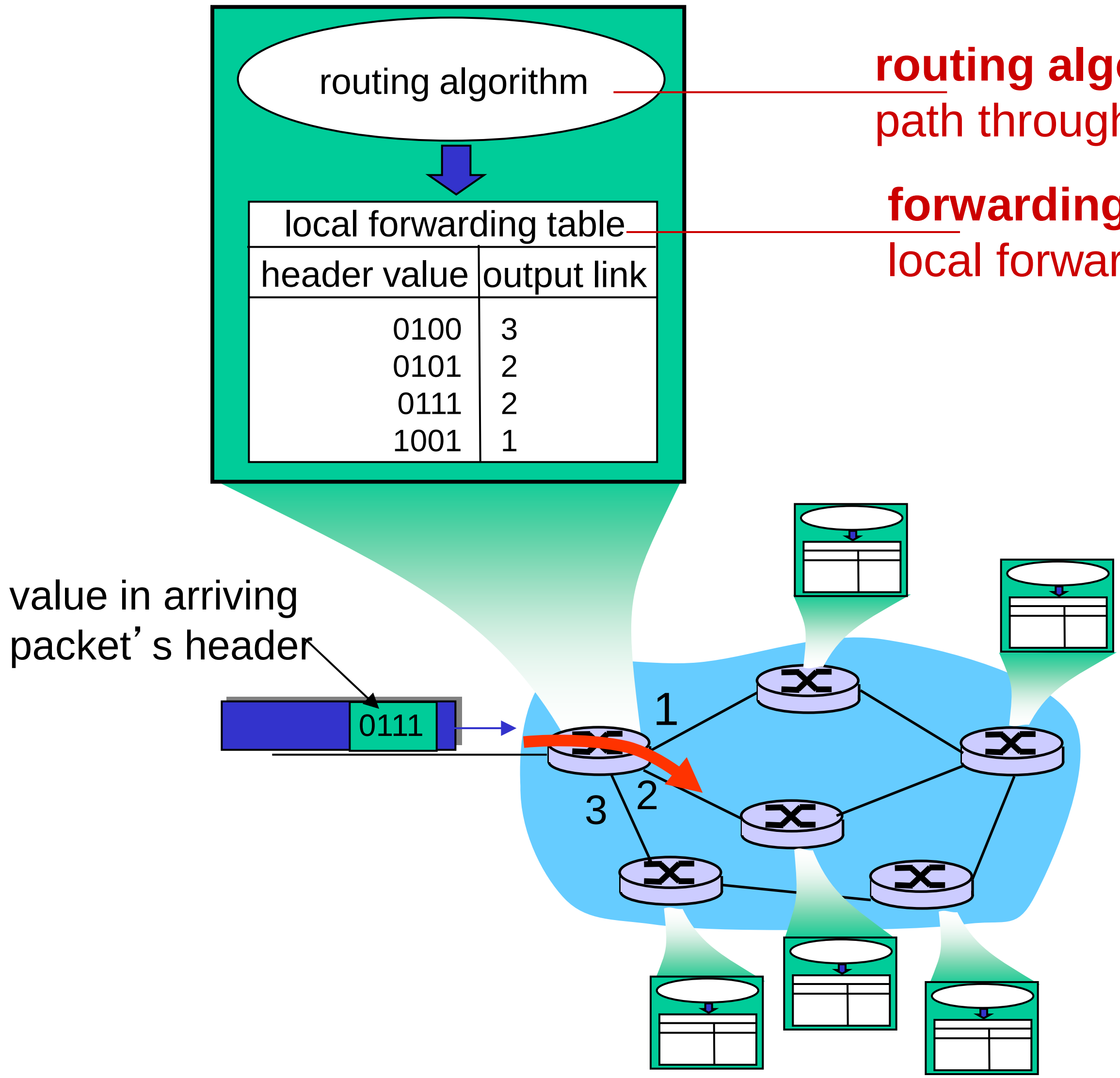
Two key network-layer functions

- *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



routing algorithm determines path through network

forwarding table determines local forwarding at this router

Network service model

Q: What *service model* can be considered for a "channel" transporting packets 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 packets:*

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

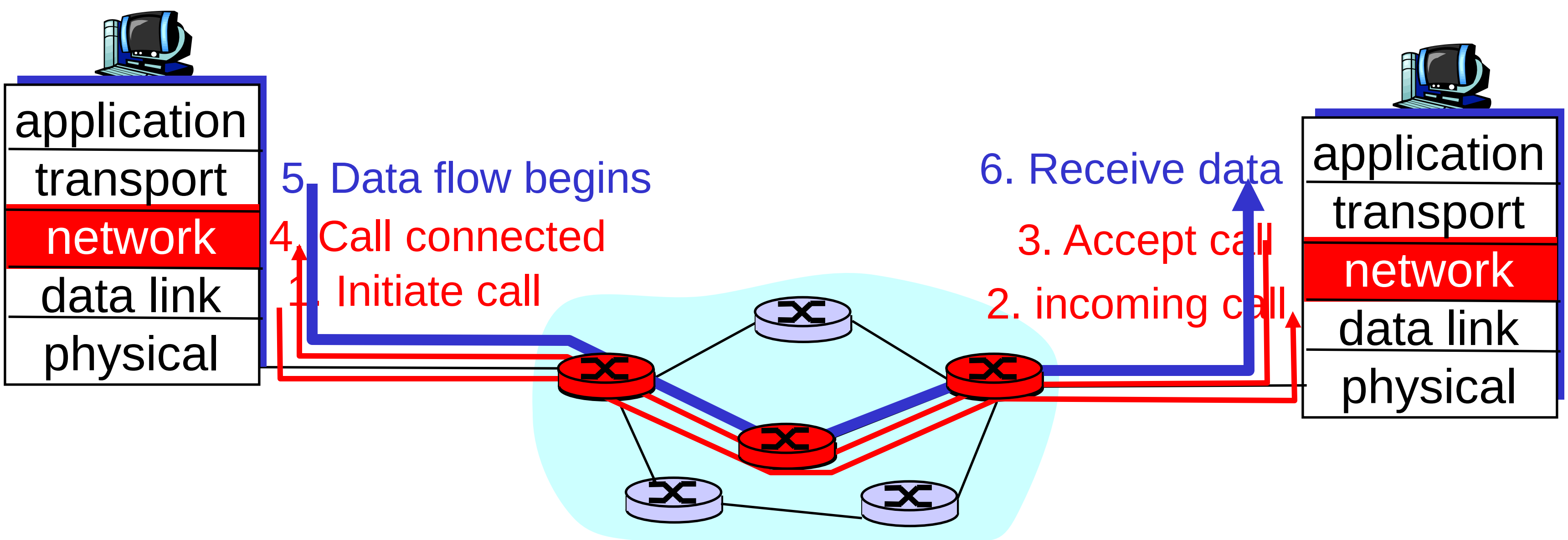
Connection, connection-less service

- *datagram* network provides network-layer *connectionless* service (Internet model)
- *virtual-circuit* network provides network-layer *connection* service (not in Internet)
- analogous to TCP/UDP connection-oriented / connectionless transport-layer services, **but**:
  - *service:* host-to-host
  - *no choice:* network provides one or the other
  - *implementation:* in network core

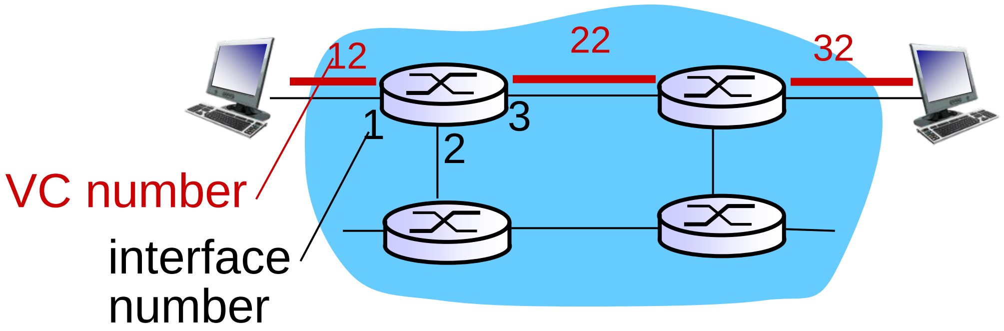
Virtual circuits:

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

- call setup, teardown for each call *before* data can flow
  - signaling protocols to setup, maintain, teardown VC (ATM, frame-relay, X.25; not in IP)
- each packet carries VC identifier (not destination host)
- *every* router maintains "state" for each passing connection
- resources (bandwidth, buffers) may be *allocated* to VC (dedicated resources = predictable service)



VC forwarding table



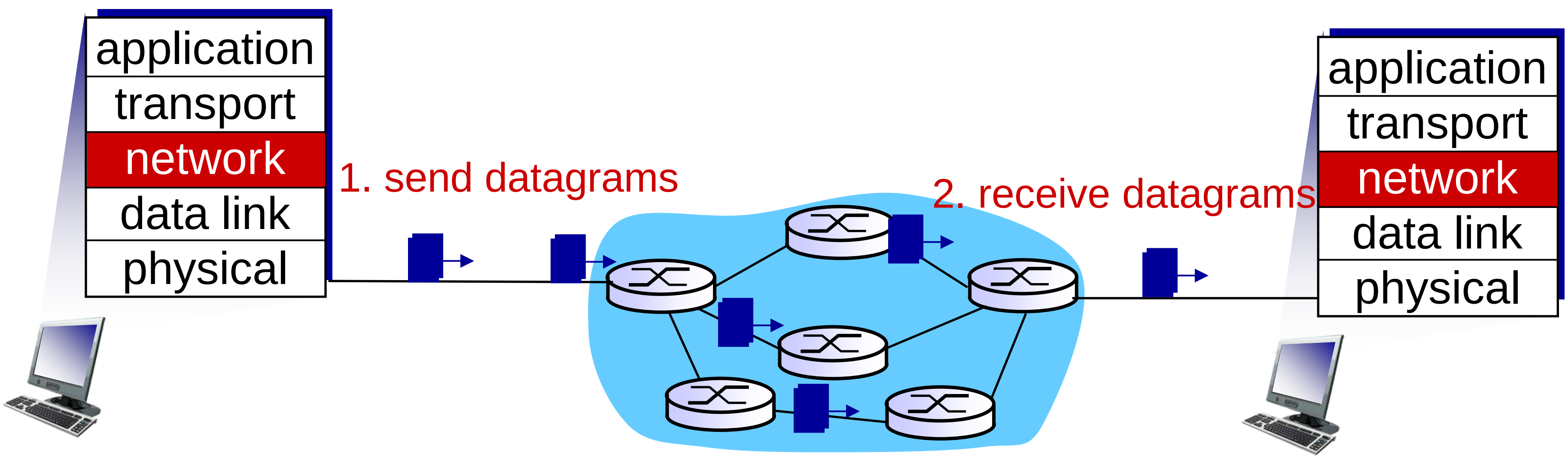
forwarding table in northwest router:

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

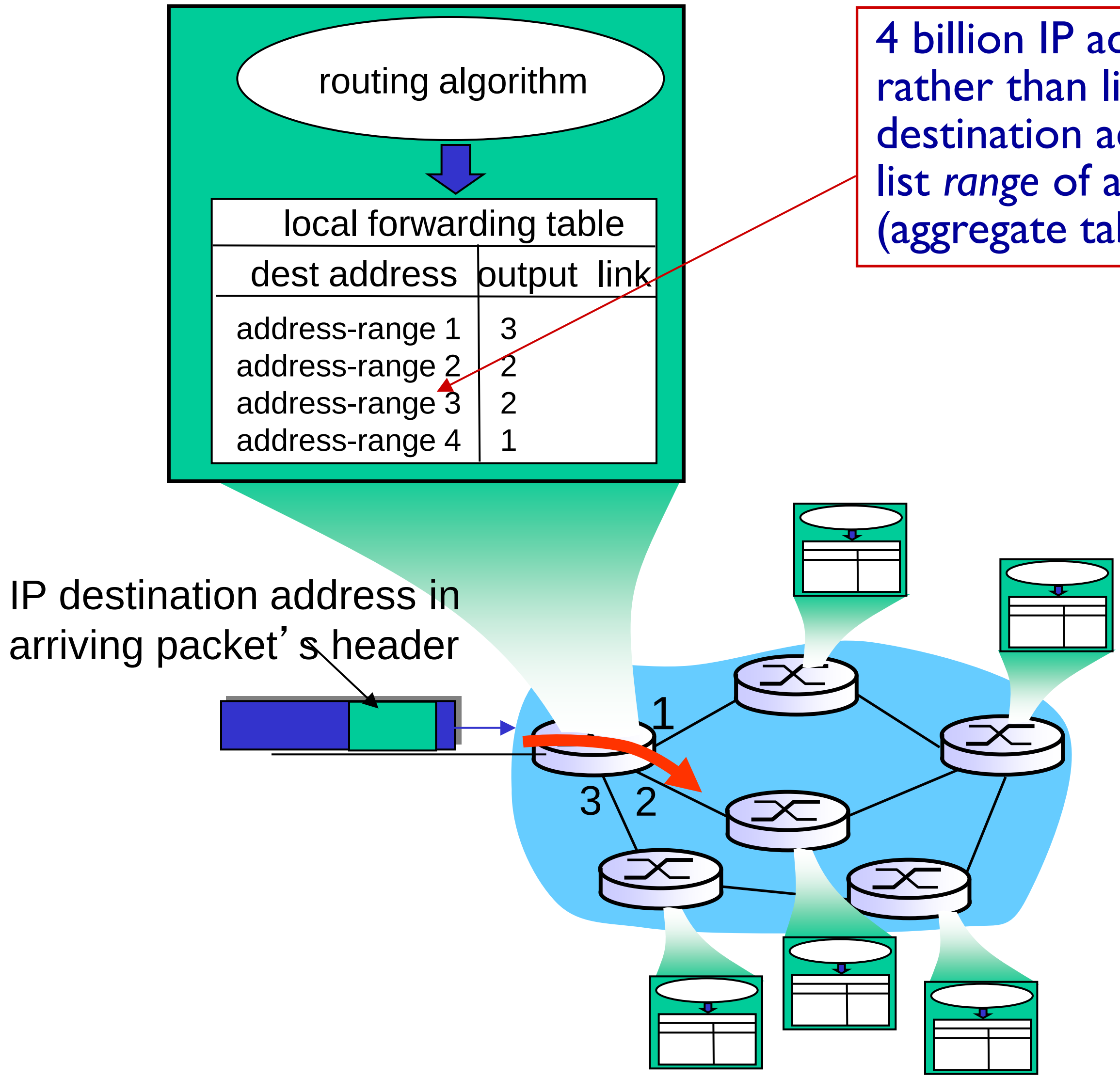
*VC routers maintain connection **state** information!*

Datagram networks (the Internet model)

- 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



Internet Datagram forwarding table



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

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?

Datagram or VC network: why?

Internet (datagram)

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

VC (eg ATM: a past's vision of the future's ww-network)

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

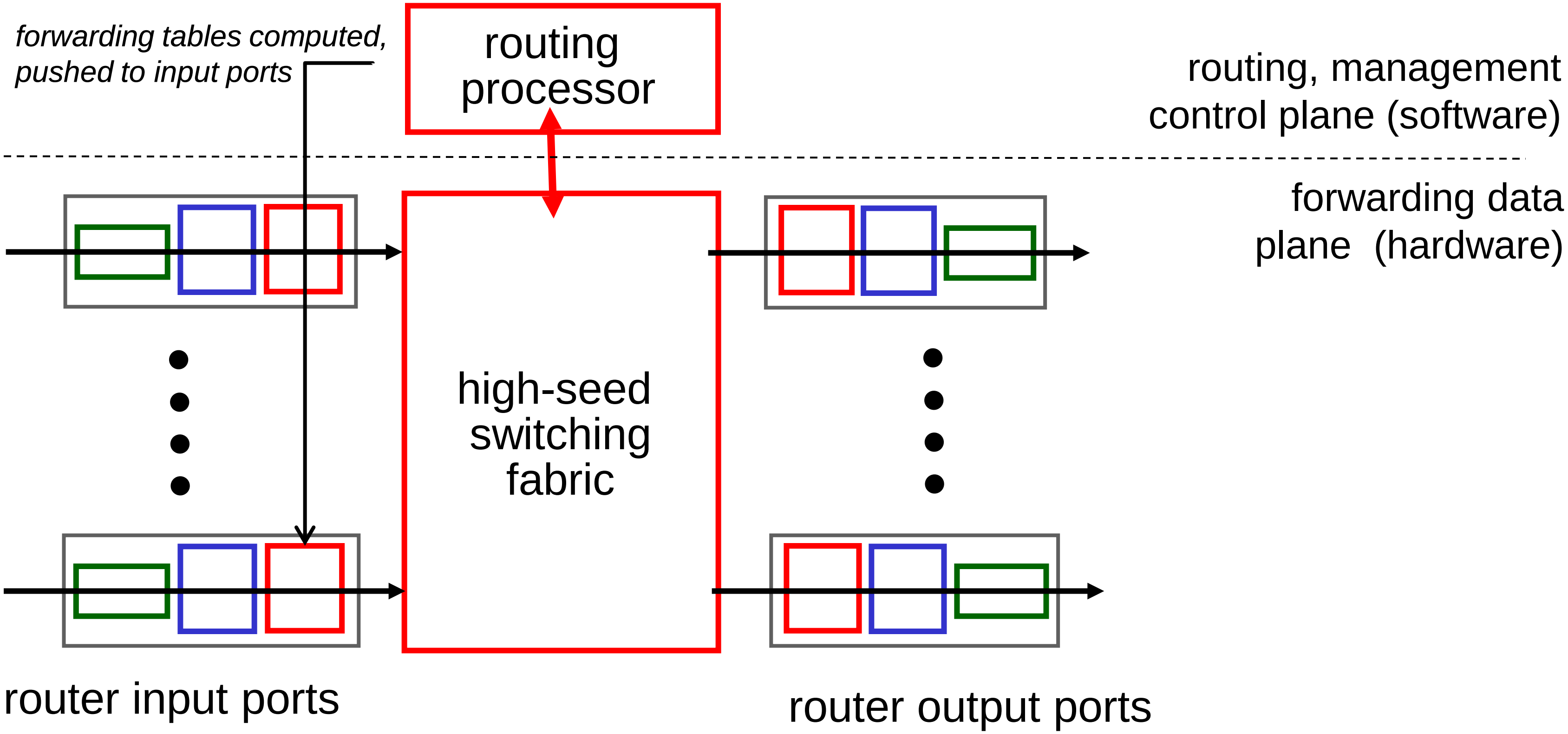
Roadmap

- Understand principles of network layer services:
  - **forwarding** versus **routing**
  - network layer **service models**
  - how a **router works**
- The **Internet Network layer**
- Routing

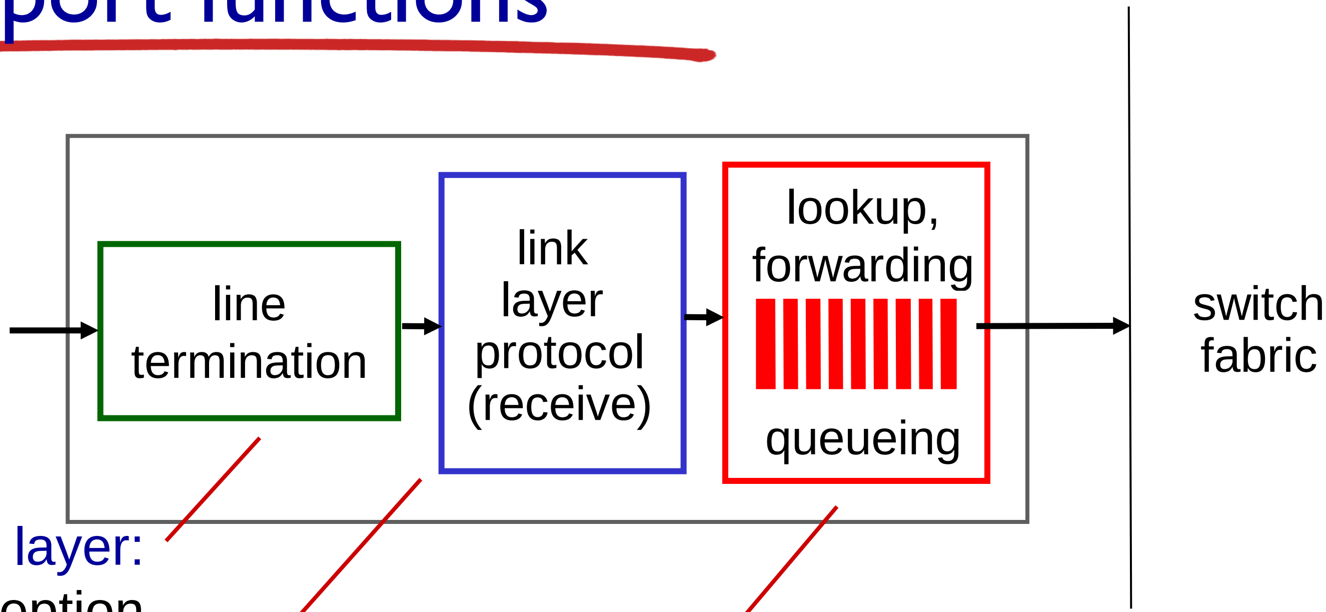
Router architecture overview

two key router functions:

- **run routing** algorithms/protocol (eg: RIP, OSPF, BGP; more on these next lecture)
- ***forwarding*** datagrams from incoming to outgoing link



Input port functions



physical layer:
bit-level reception

data link layer:
e.g., Ethernet
see chapter 5

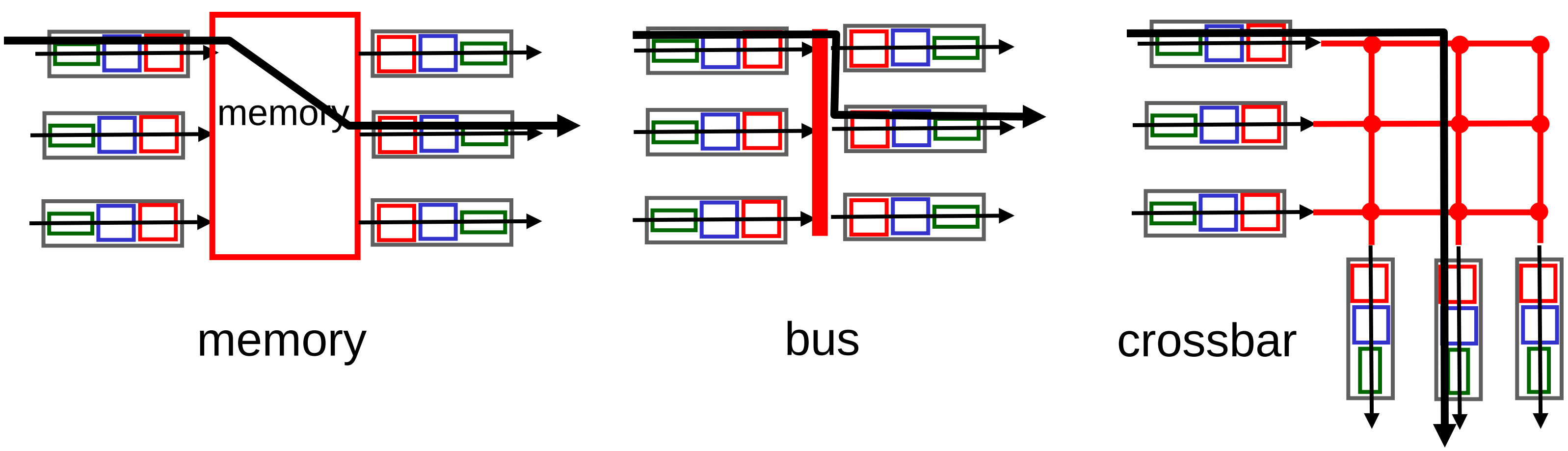
switching:

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

Switching fabrics

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

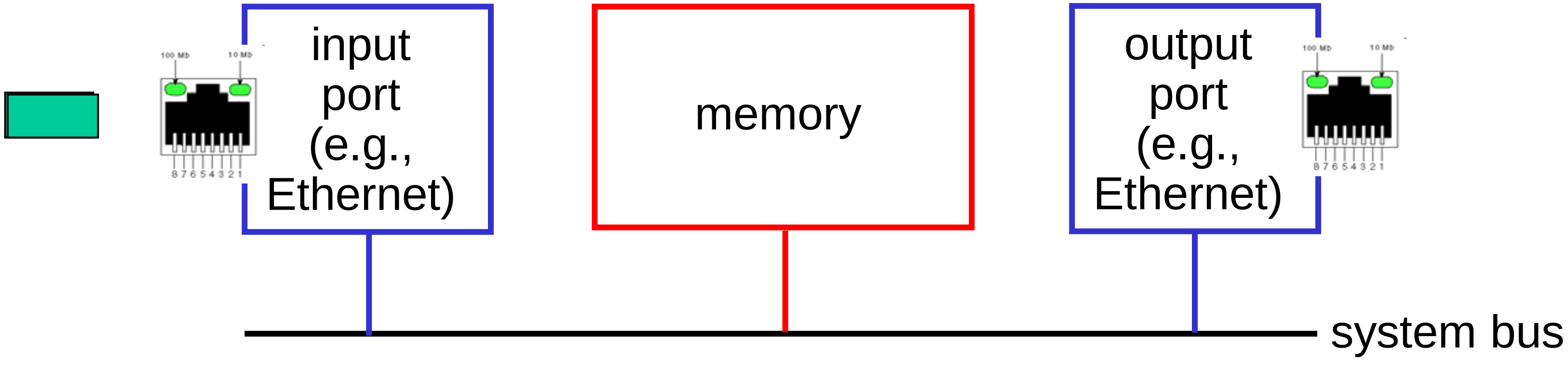
- three types of switching fabrics:



Switching via memory

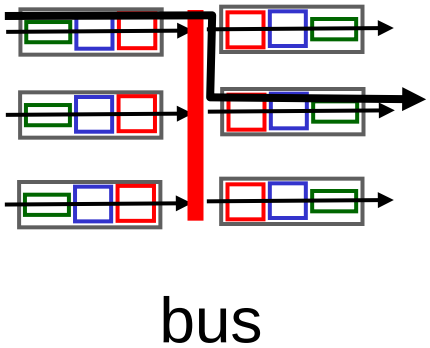
first generation routers:

- 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)



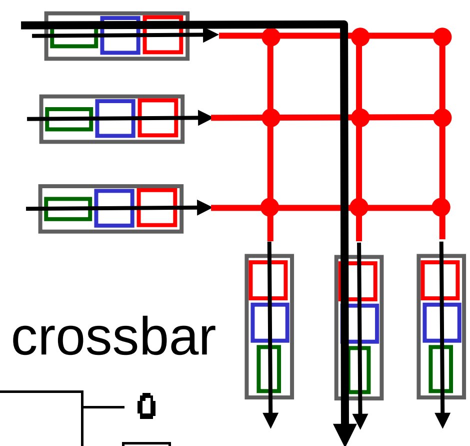
Switching via a bus

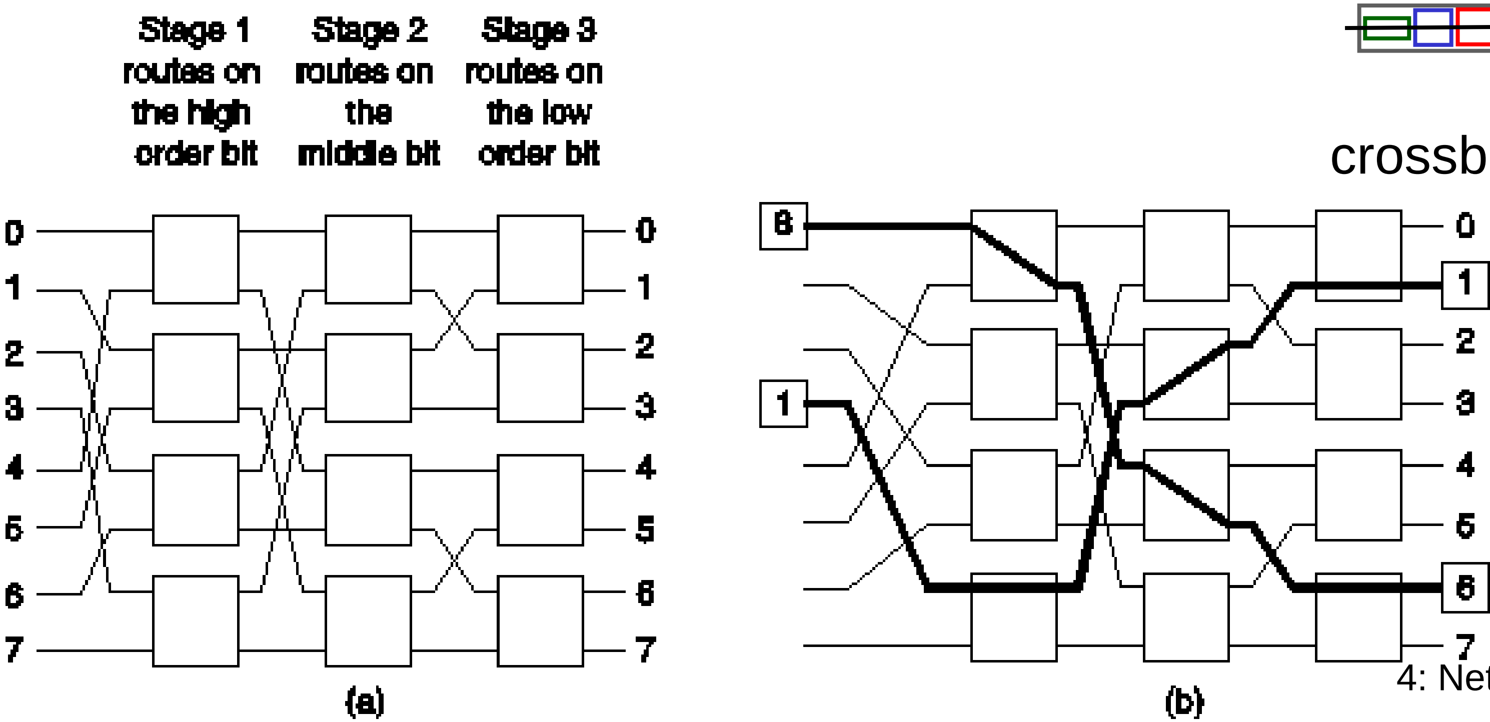
- datagram from input port memory to output port memory via a shared bus
- *bus contention:* switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access and enterprise routers



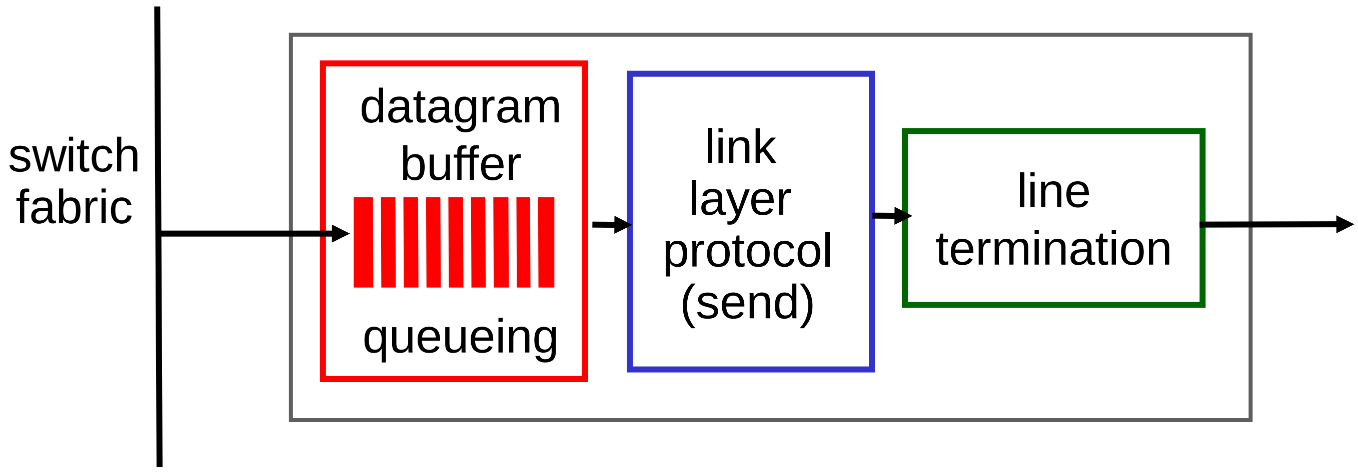
Switching Via An Interconnection Network

- Overcome bus bandwidth limitations
- Banyan networks, other interconnection nets (also used in processors-memory interconnects in multiprocessors)
  - Advanced design: fragmenting datagram into fixed length cells, switch cells through the fabric (ATM-network principle).
- Cisco 12000: switches at 60 Gbps



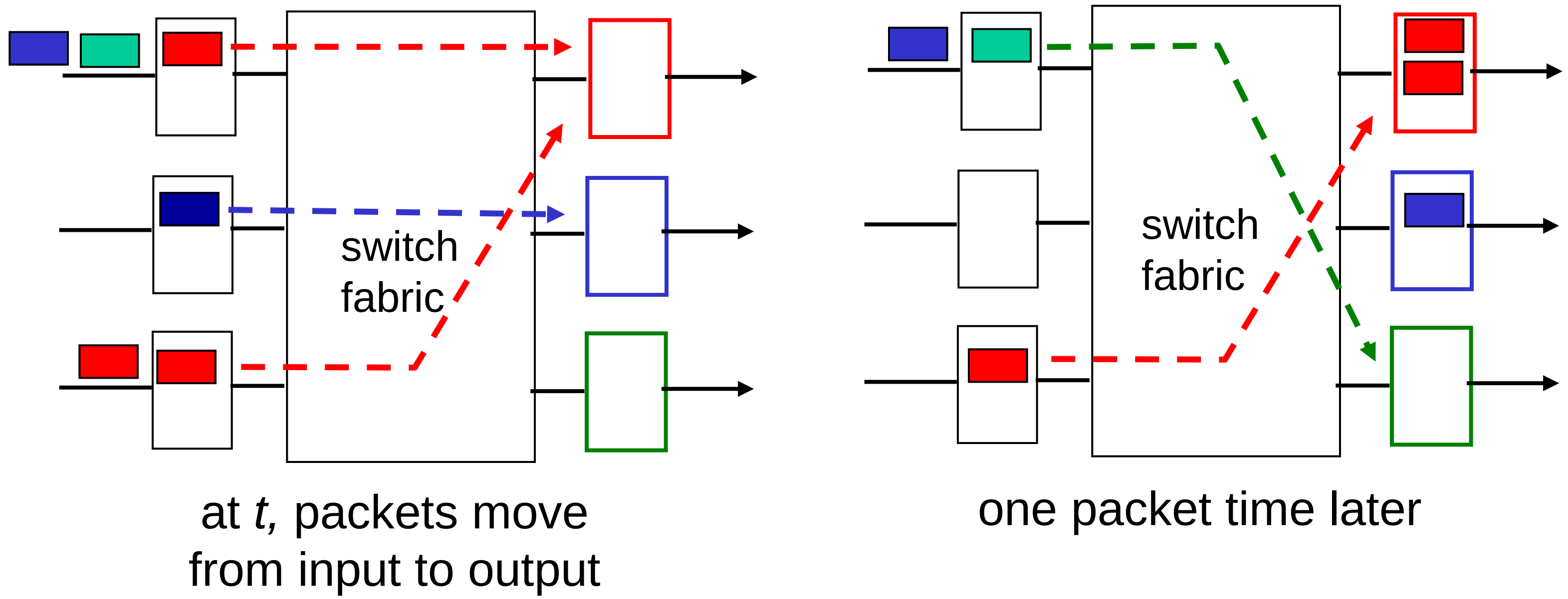


Output ports



- *buffering* required when datagrams arrive from fabric faster than the transmission rate
  - *queueing (delay) and loss due to output port buffer overflow!*
- *scheduling discipline* chooses among queued datagrams for transmission

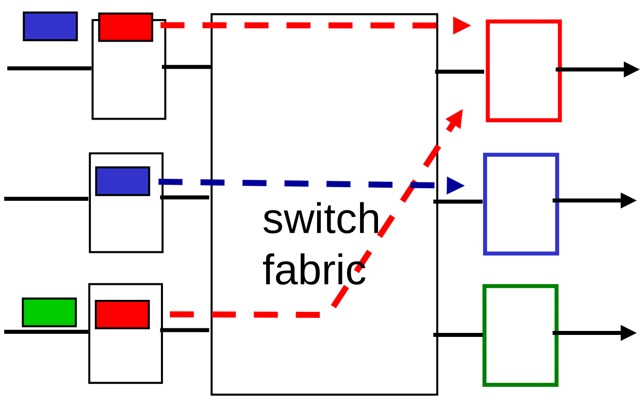
Output port queueing



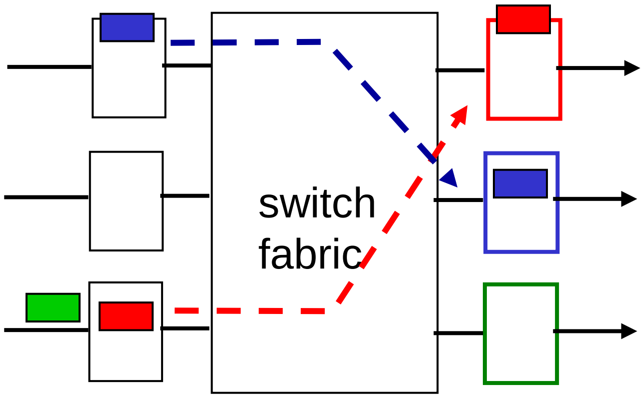
- buffering when arrival rate via switch exceeds output line speed
- *queueing (delay) and loss due to output port buffer overflow!*

Input port queuing

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



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



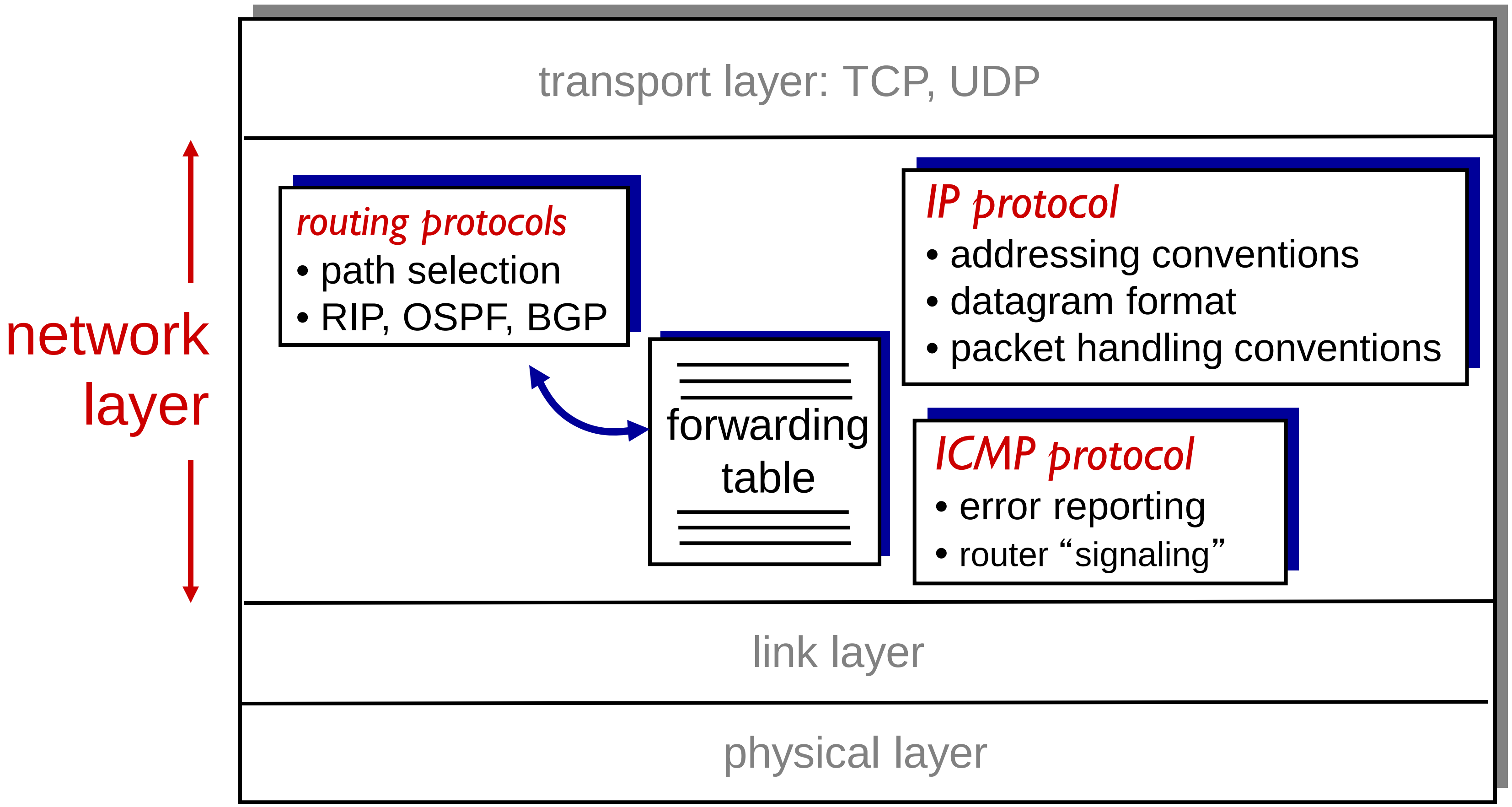
one packet time later:
green packet experiences HOL blocking

Roadmap

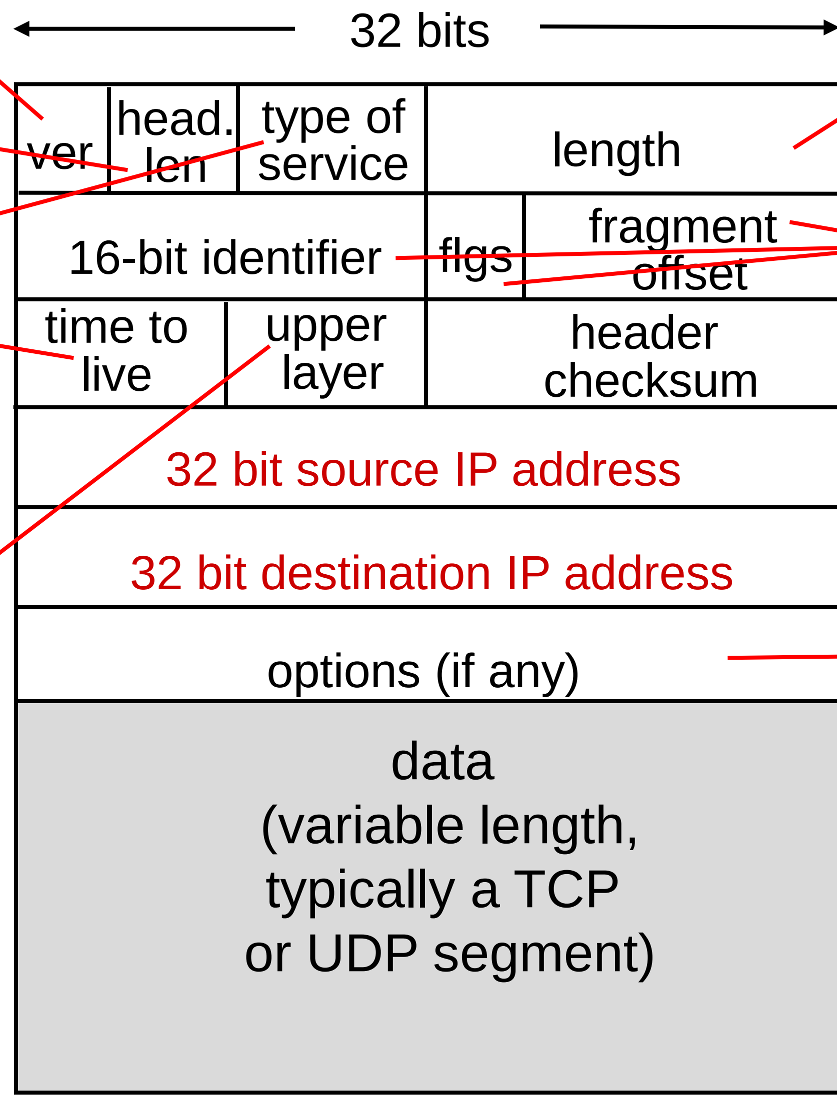
- Understand principles of network layer services:
  - **forwarding** versus **routing**
  - network layer **service models**
  - how a **router works**
- The **Internet Network layer**:
  - **IP, Addressing** and **delivery**
  - **Error** and **information reporting** with **ICMP**
  - **NAT**
  - **IPv6**
- Routing

The Internet network layer

host, router network layer functions:



IPv4 datagram format

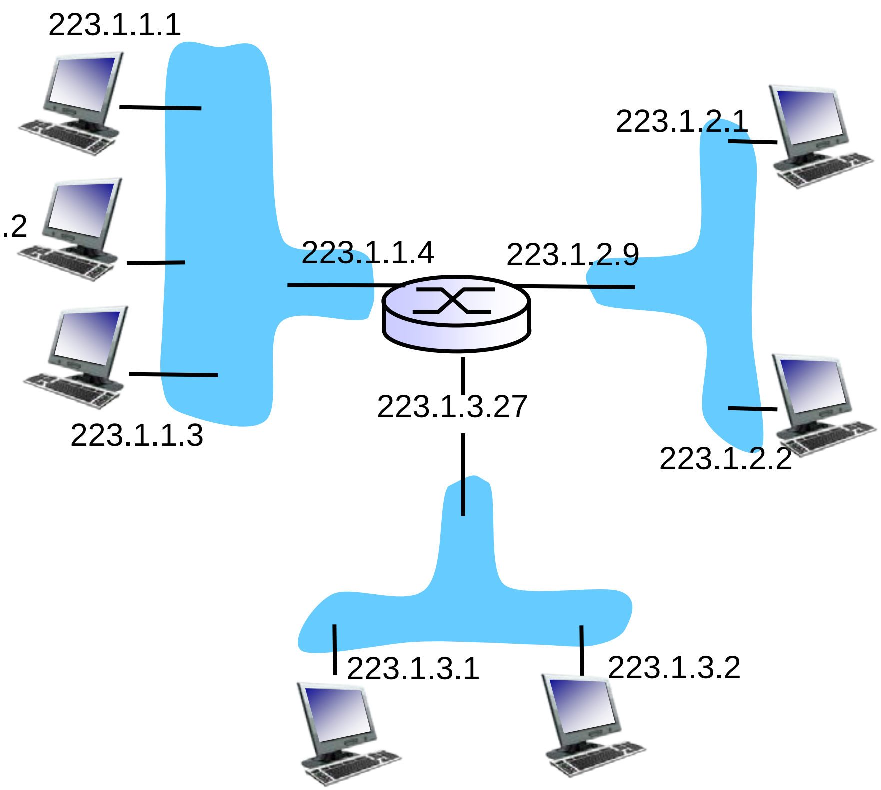


how much overhead?

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

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 or two interfaces (e.g., wired Ethernet and wireless 802.11)
- *IP addresses associated with each interface*



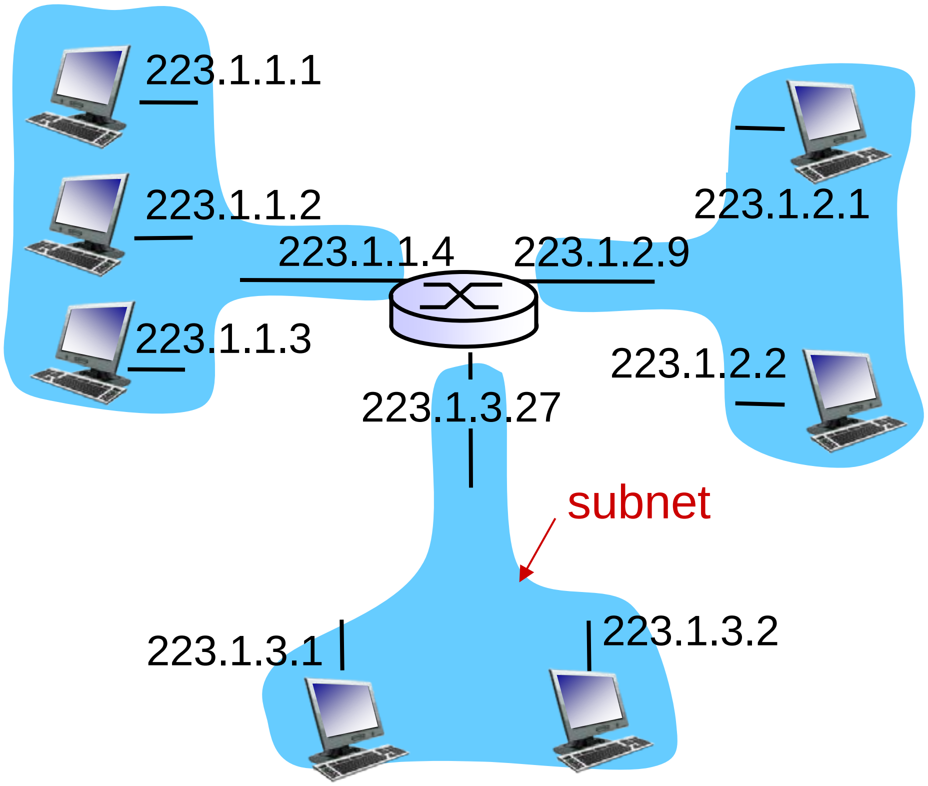
223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Dotted Decimal Notation

Subnets

- IP address:
  - subnet part - high order bits (variable number)
  - 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*

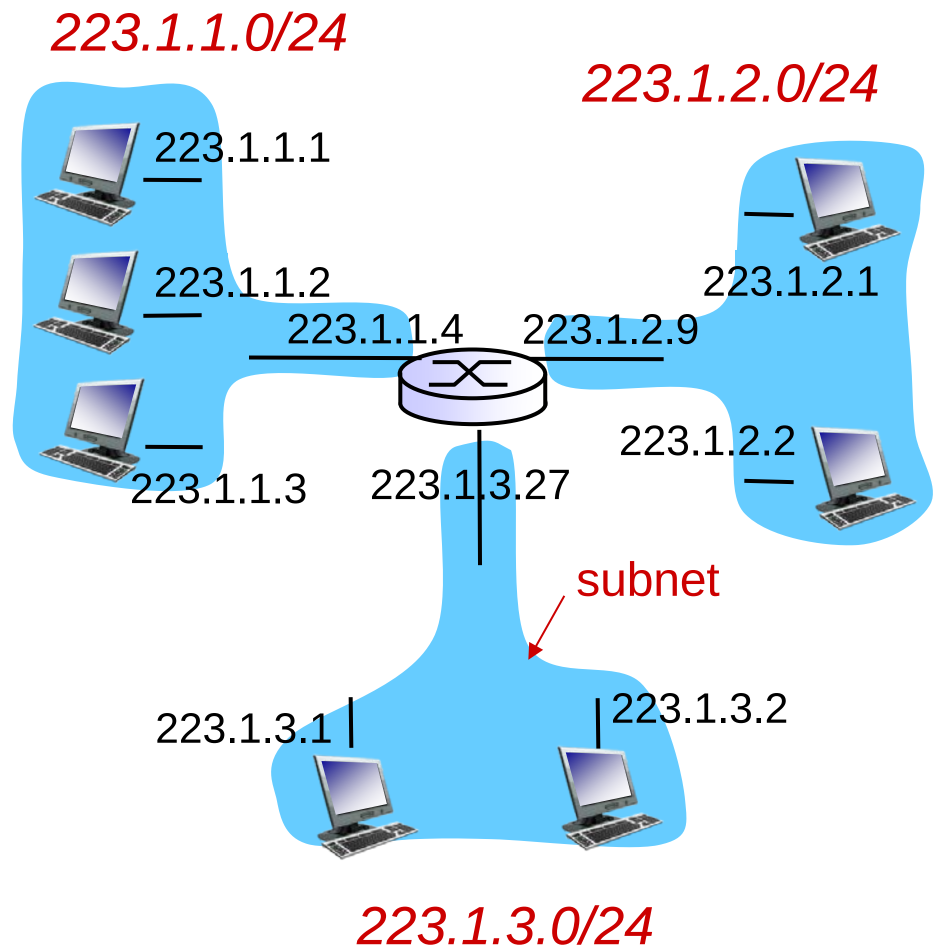


network consisting of 3 subnets

Subnets

recipe

- 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: eg /24
defines how to find the subnet part of the address …

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

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 first 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 "mask" where the 32 – 24 last bits set to 1)	00000000.00000000.00000000.10000010	0.0.0.130

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 (Network, Broadcast)

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 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"
- Display address information:
  - Windows: ipconfig /all
  - Linux: ip addr list

DHCP: Dynamic Host Configuration Protocol

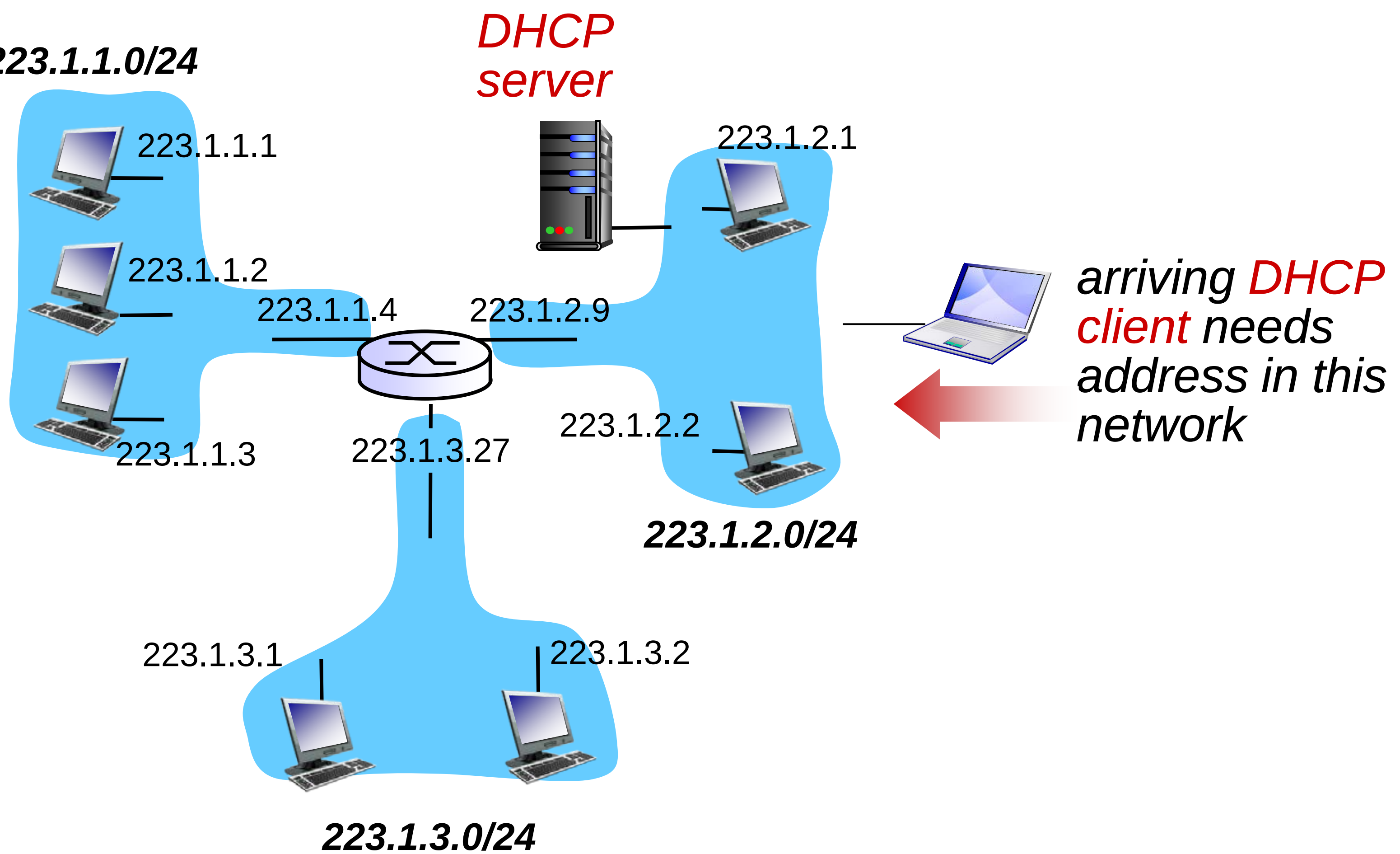
goal: allow host to *dynamically* obtain its IP address from network server when it joins network

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

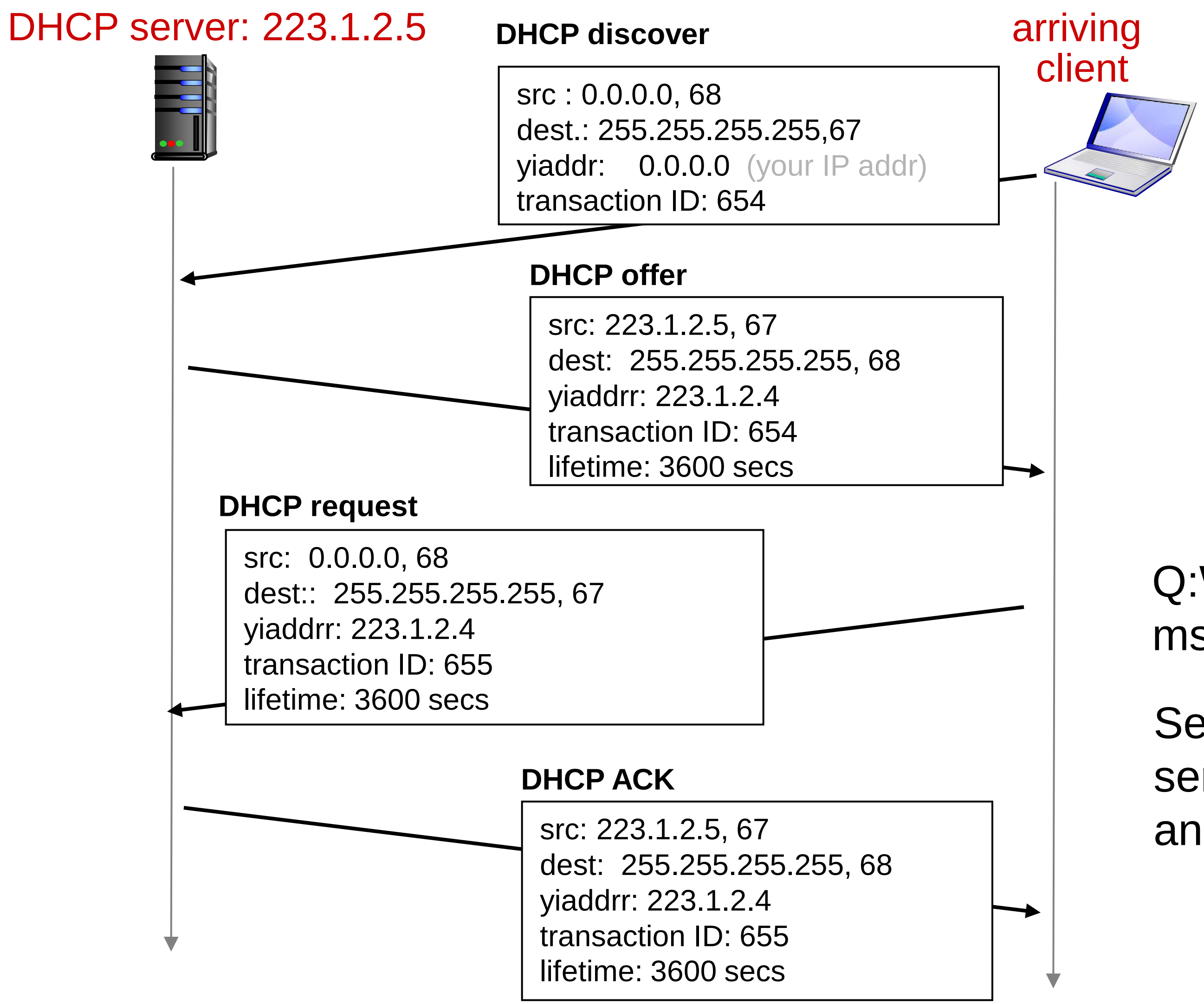
DHCP overview:

- 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



DHCP client-server scenario



Q:Why a request msg?

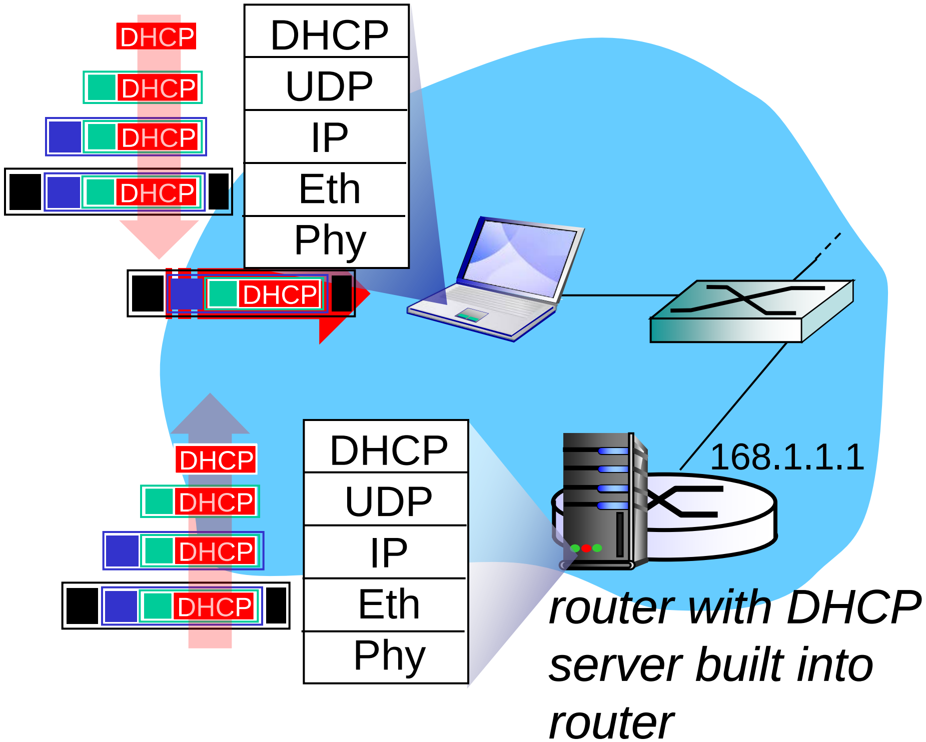
Several DHCP servers may answer and offer addresses

DHCP: more than an IP address

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

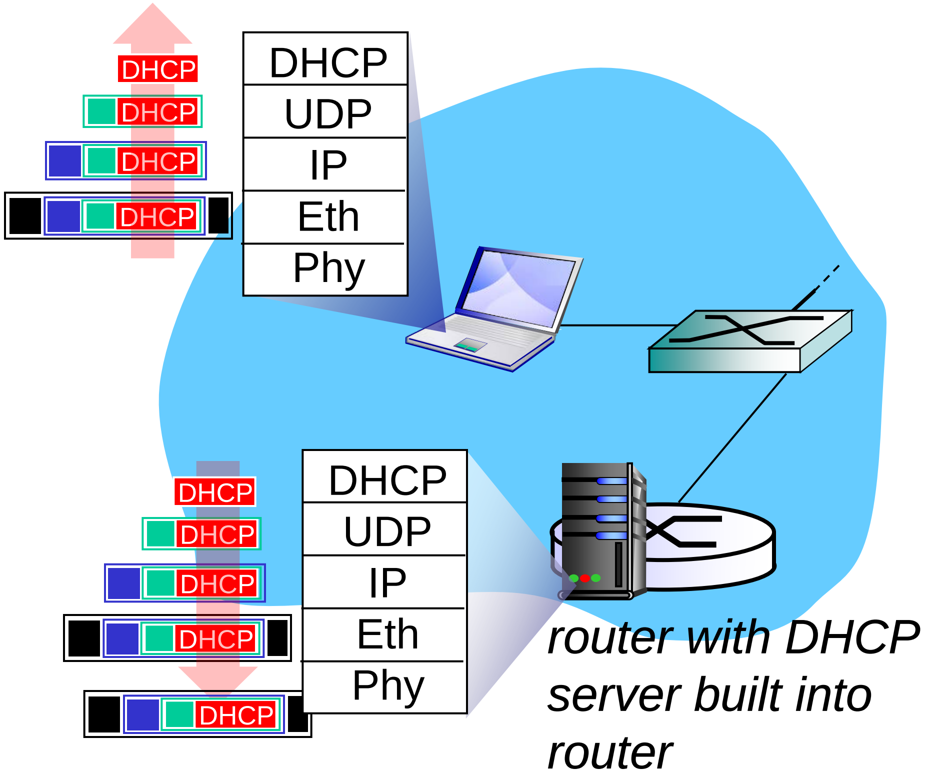
- address of first-hop router for client
- name and IP address of DNS sever
- network mask (indicating network versus host portion of address)

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 demuxed to IP demuxed, UDP demuxed 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, demuxing up to DHCP at client
- client now knows its IP address, name and IP address of DNS server, IP address of its first-hop router

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 an organisation get a network of IP addresses?

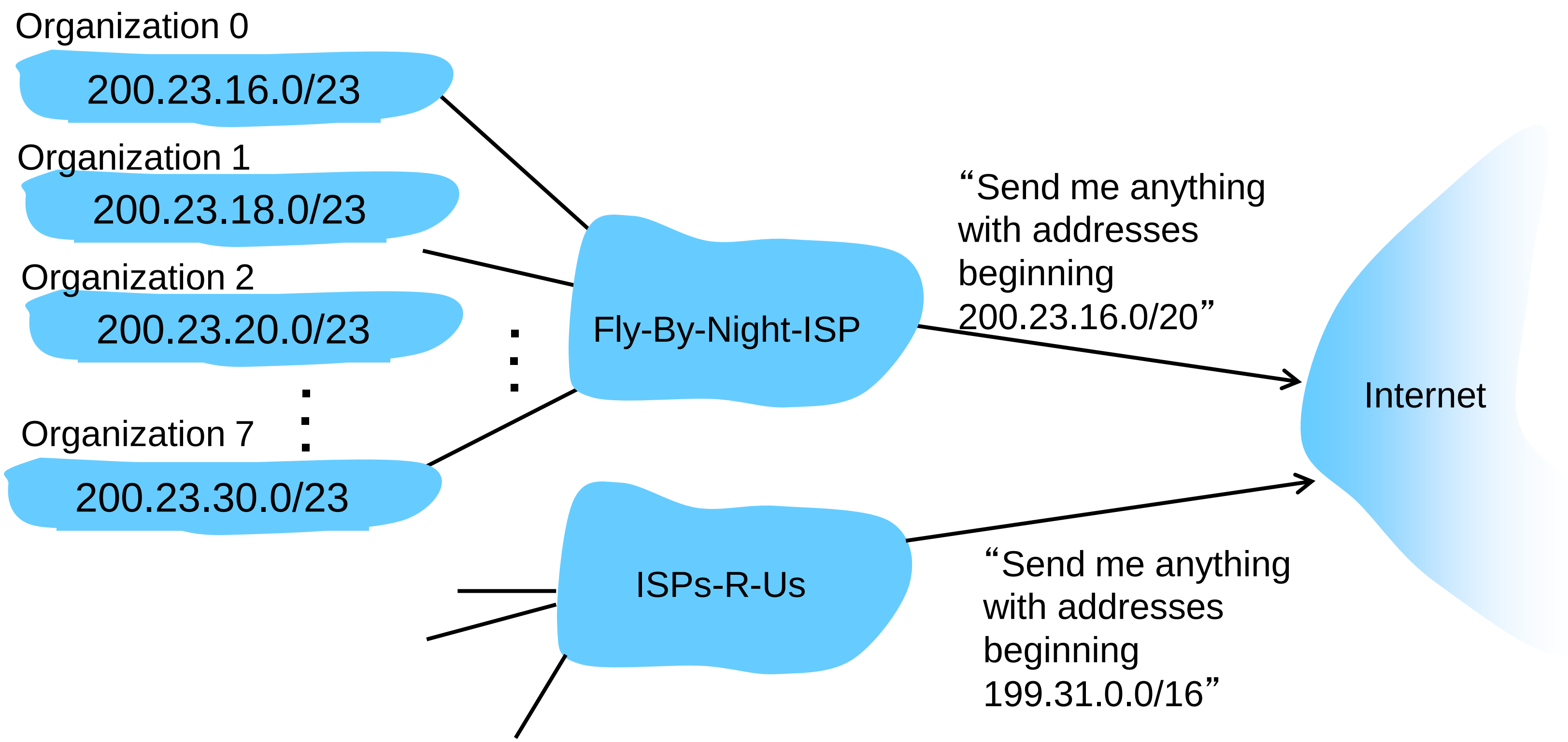
A: gets allocated as a portion (*subnet*) of its ISP's address space; eg:

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**

3 bits, 8 networks

Hierarchical addressing: route aggregation

hierarchical addressing allows efficient advertisement of routing information:



IP addressing: the last word...

Q: how does an ISP get block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

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

Getting a datagram from source to dest.

Getting a datagram from source to dest.

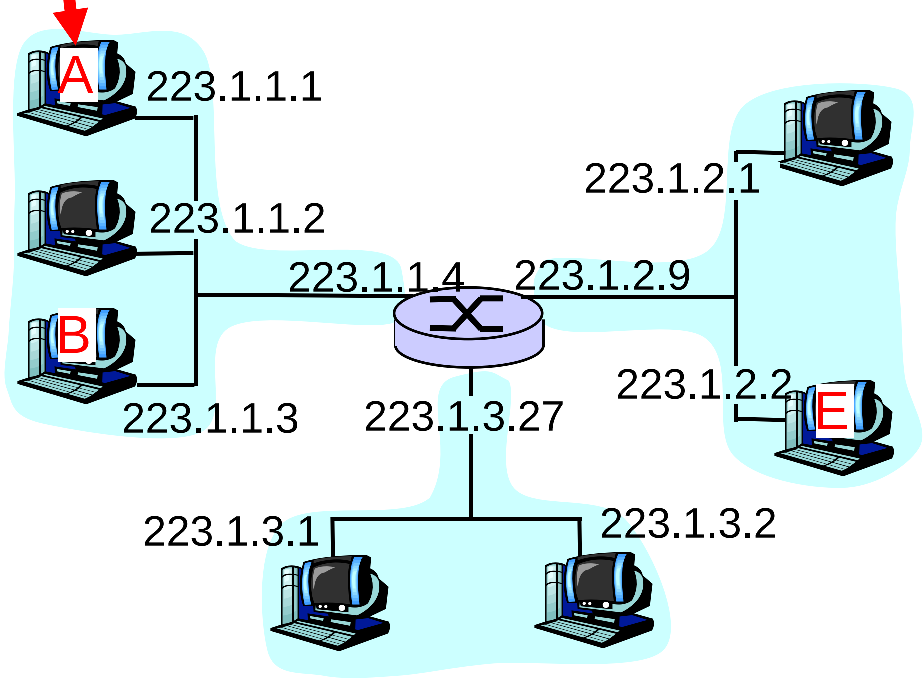
forwarding table in A

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

IP datagram:

misc fields	source IP addr	dest IP addr	data

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



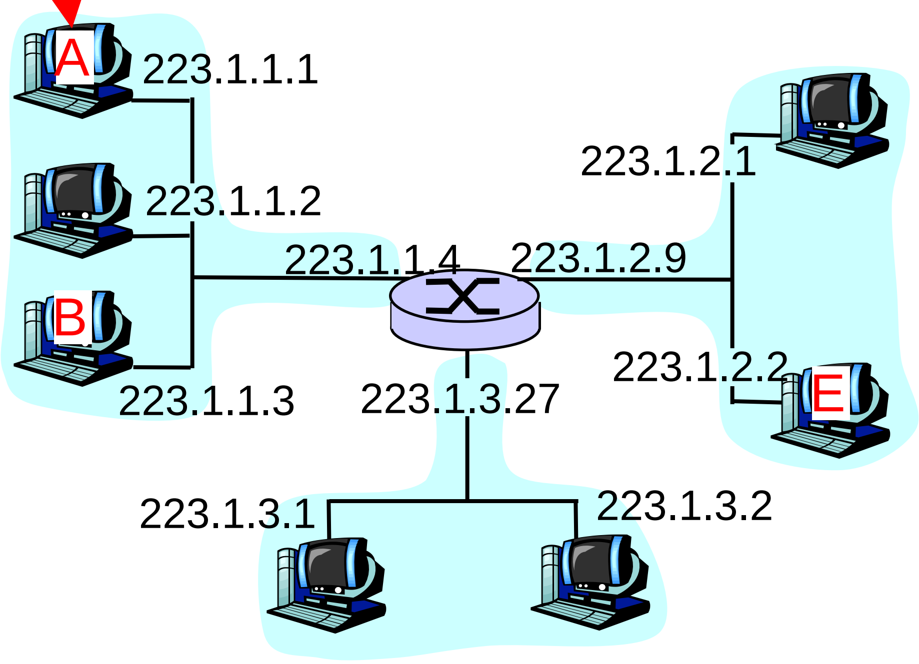
Getting a datagram from source to dest.

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)



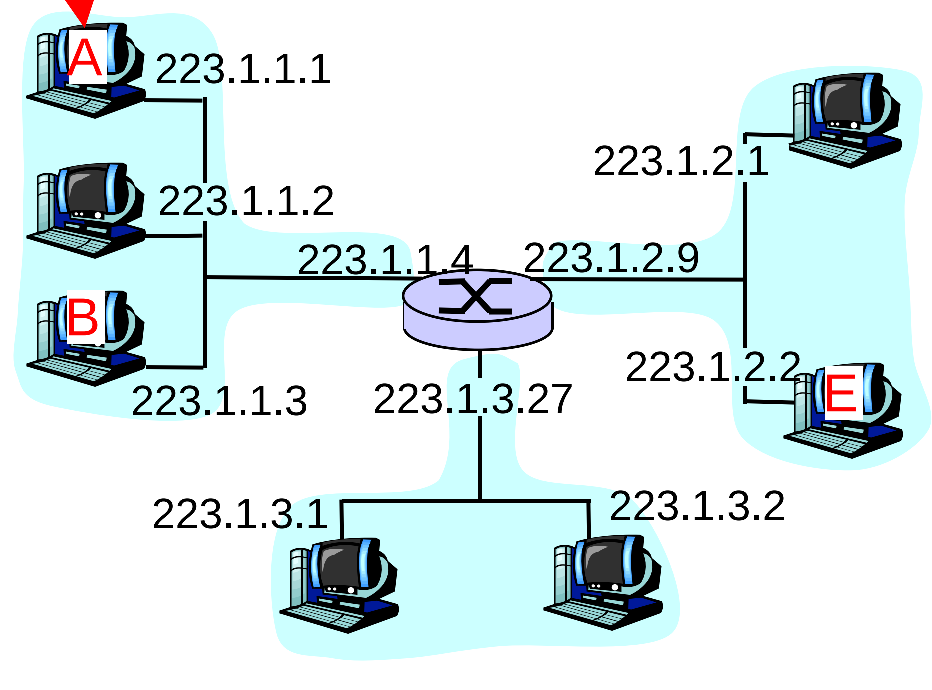
Getting a datagram from source to dest.

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…..



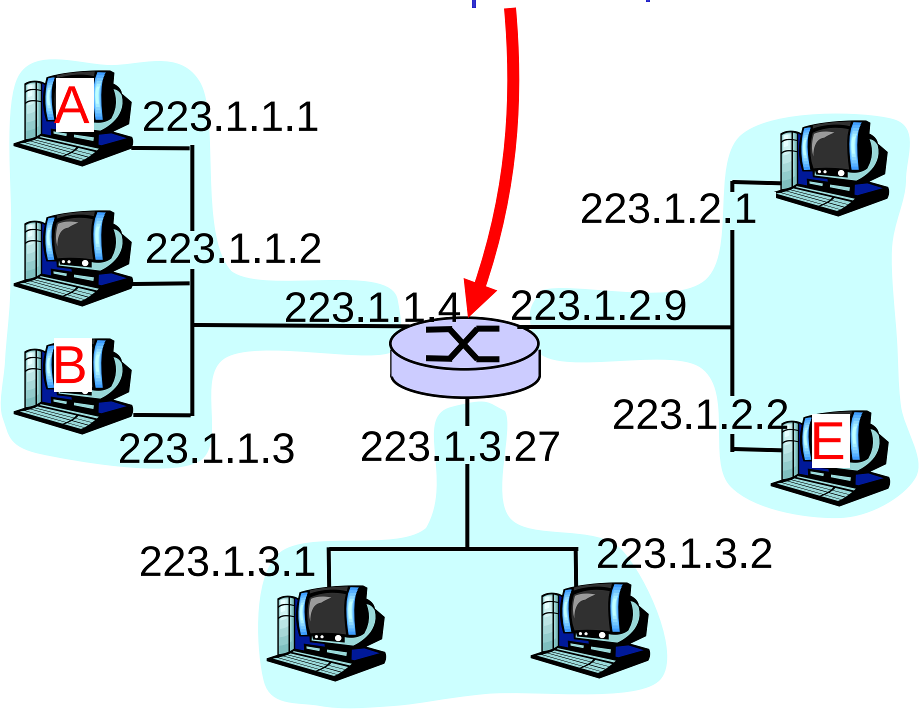
Getting a datagram from source to dest.

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!)



Roadmap

- Understand principles of network layer services
- The **Internet Network layer**:
  - **IP, Addressing** and **delivery**
  - **Error** and **information reporting** with **ICMP**
  - **NAT**
  - **IPv6**
- Routing

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

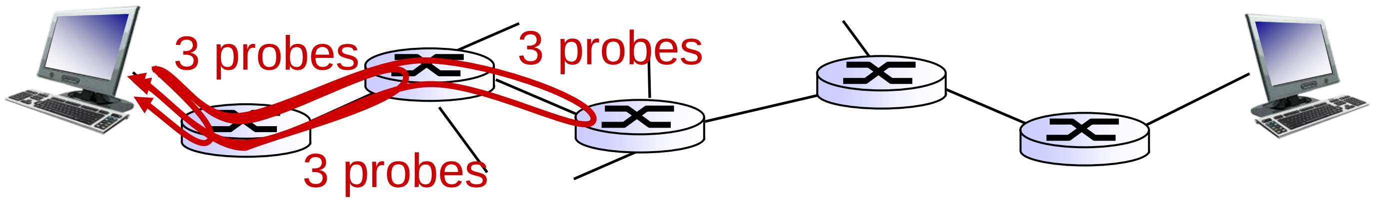
Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
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 set has TTL =1
  - second set has TTL=2, etc.
  - unlikely port number
- when *n*th set of datagrams arrives to nth router:
  - router discards datagrams
  - and sends source ICMP messages (type 11, code 0)
  - ICMP messages includes name of router & IP address
- when ICMP messages arrive, source records RTTs

stopping criteria:

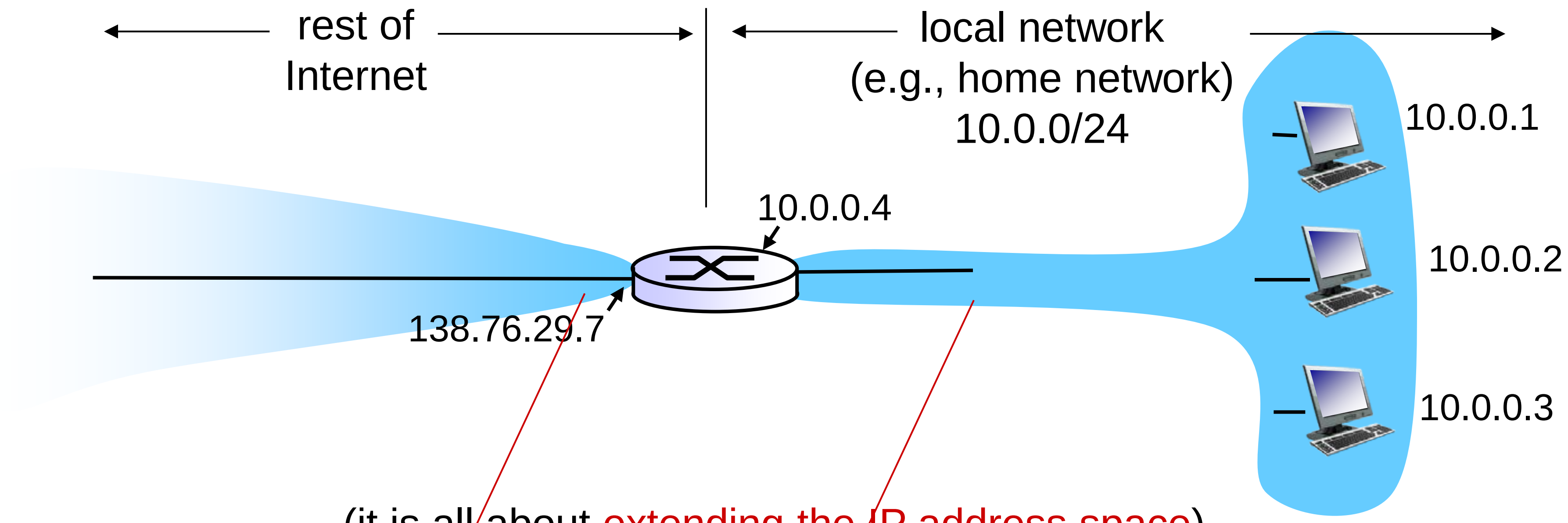
- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops



Roadmap

- Understand principles of network layer services
- The **Internet Network layer**:
  - **IP, Addressing** and **delivery**
  - **Error** and **information reporting** with **ICMP**
  - **NAT**
  - **IPv6**
- Routing

NAT: network address translation



(it is all about extending the IP address space)

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:

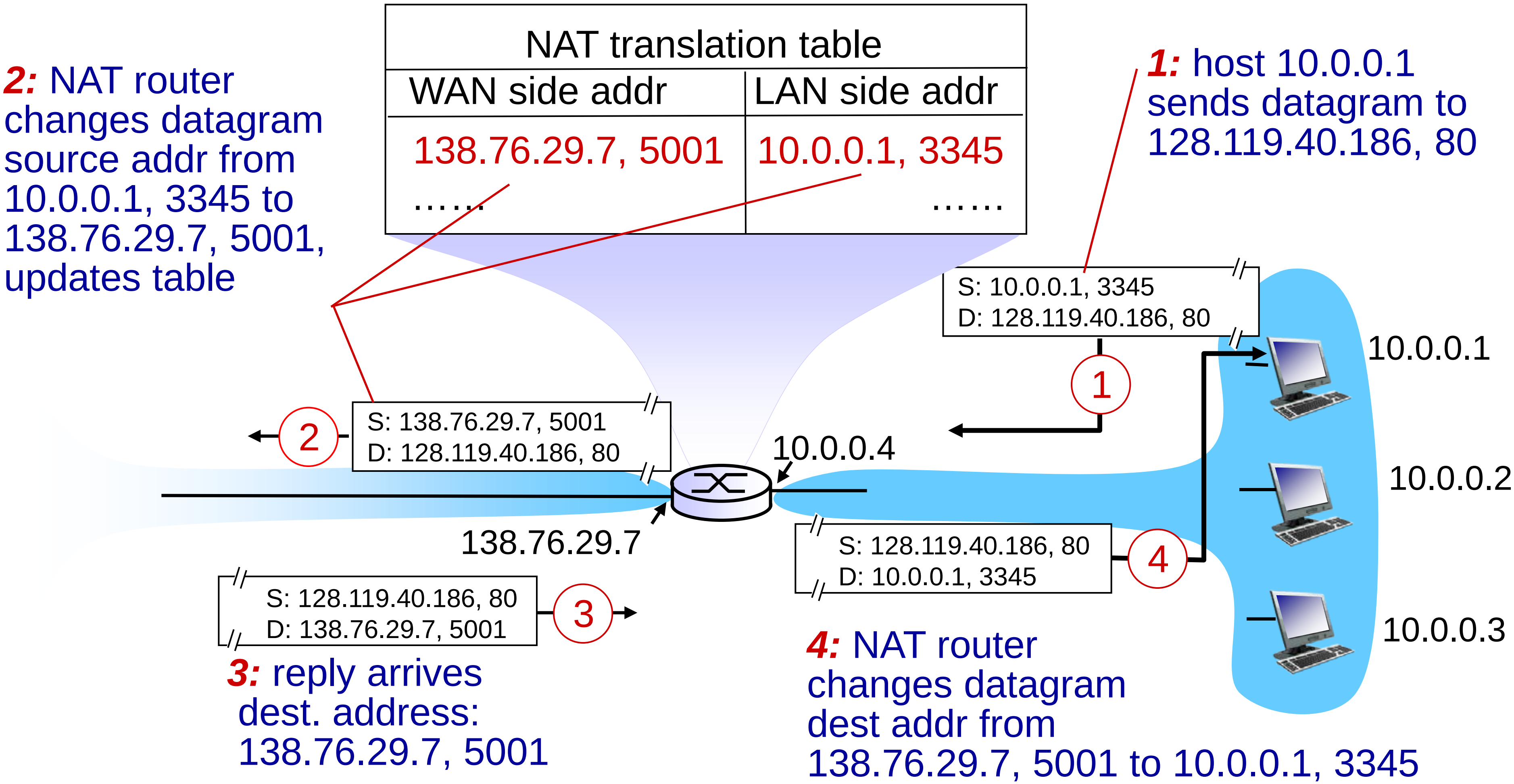
outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

. . . 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

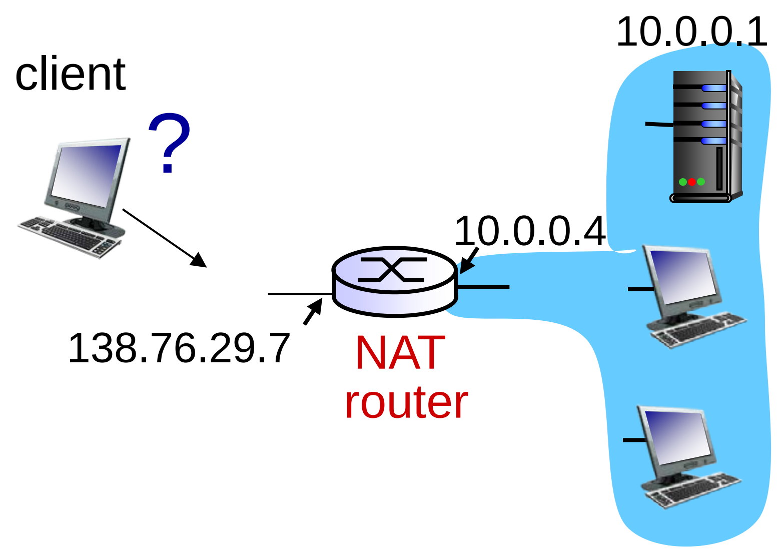


NAT: network address translation

- 16-bit port-number field:
  - 64k simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, e.g., P2P applications
  - address shortage should instead be solved by IPv6

NAT traversal problem

- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible 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 25000
- *Solution 2*: automate the above through a protocol (universal plug-and-play)
- *Solution 3*: through a proxy/relay (will discuss in connection to p2p applications)



Roadmap

- Understand principles of network layer services
- The **Internet Network layer**:
  - **IP, Addressing** and **delivery**
  - **Error** and **information reporting** with **ICMP**
  - **NAT**
  - **IPv6**
- Routing

IPv6: motivation

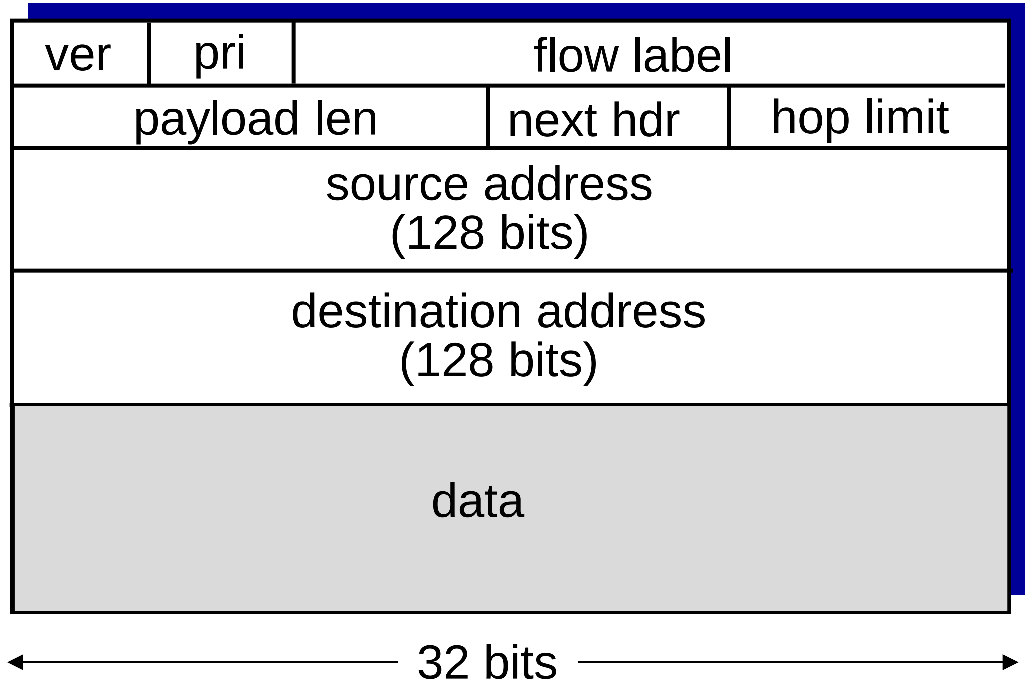
- *initial motivation:* 32-bit address space soon to be completely allocated.
- additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

IPv6 datagram format:

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

IPv6 datagram format

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

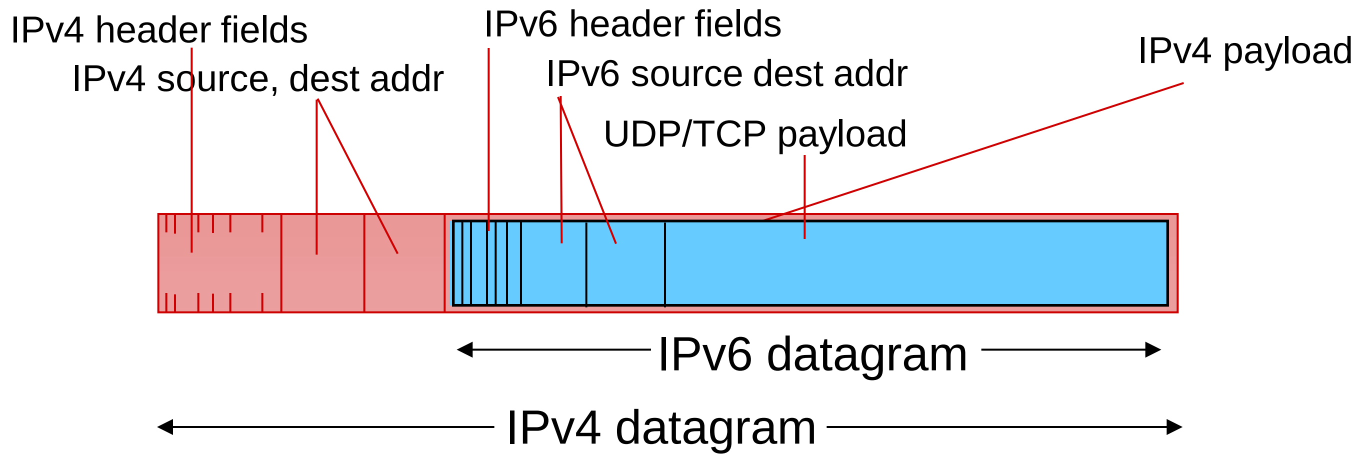


Other changes from IPv4

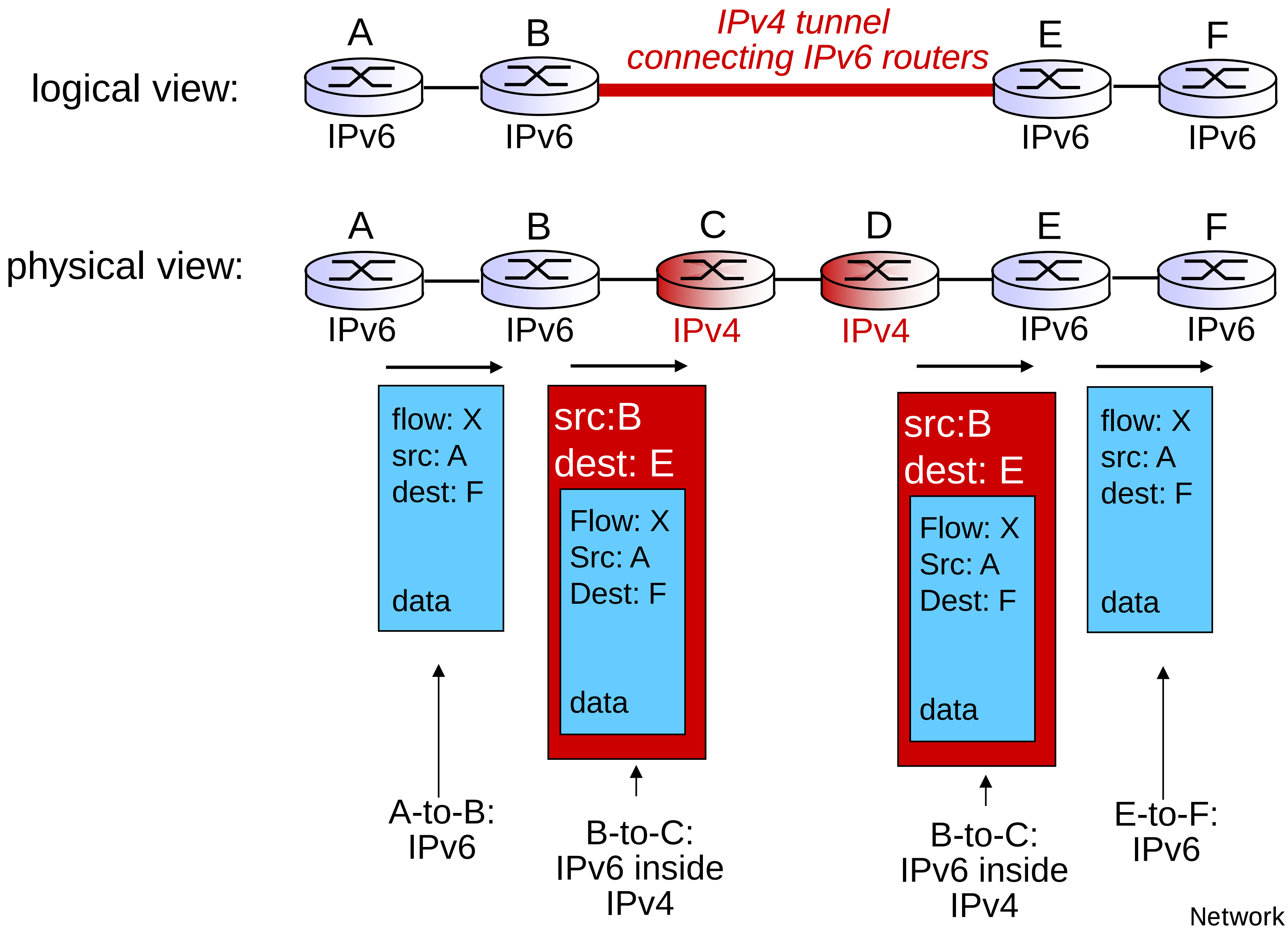
- *checksum:* removed entirely to reduce processing time at each hop
- *options:* allowed, but outside of header, indicated by "Next Header" field
- *ICMPv6:* new version of ICMP
  - additional message types, e.g. "Packet Too Big"
  - multicast group management functions

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)



Roadmap

- Understand principles of network layer services
- The **Internet Network layer**:
  - **IP, Addressing** and **delivery**
  - **Error** and **information reporting** with **ICMP**
  - **NAT**
  - **IPv6**
- Routing

Reading instructions

- Main textbook: careful: 4.1-4.6, quick/optional: 4.7
- Further, optional study: cf embedded, no-ppt-show slides

Review questions for this part

- network layer **service models**
  - Contrast virtual circuit and datagram routing (simplicity, cost, purposes, what service types they may enable)
- **forwarding** versus **routing**
  - Explain the interplay between routing and forwarding
- how a **router works**
  - What is inside a router? How/where do queueing delays happen inside a router? Where/why can packets be dropped at a router?
- What is subnet? What is subnet masking?
  - Train/exercise masking calculations
- Explain how to get an IP packet from source to destination
- Explain how NAT works.

Some complementary material /video-links

- How does BGP choose its routes
  http://www.youtube.com/watch?v=RGe0qt9Wz4U&feature=plcp

- IP addresses and subnets
  http://www.youtube.com/watch?v=ZTJIkjgyuZE&list=PLE9F3F05C381ED8E8&feature=plcp

Some taste of layer 2: no worries if not all details fall in place, need the lecture also to grasp them. The lecture will be held next week

- Hubs, switches, routers
  http://www.youtube.com/watch?v=reXS_e3fTAk&feature=related
-
- What is a broadcast + MAC address
  http://www.youtube.com/watch?v=BmZNcjLtmwo&feature=plcp

- Broadcast domains:
  http://www.youtube.com/watch?v=EhJO1TCQX5I&feature=plcp

Extra slides

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

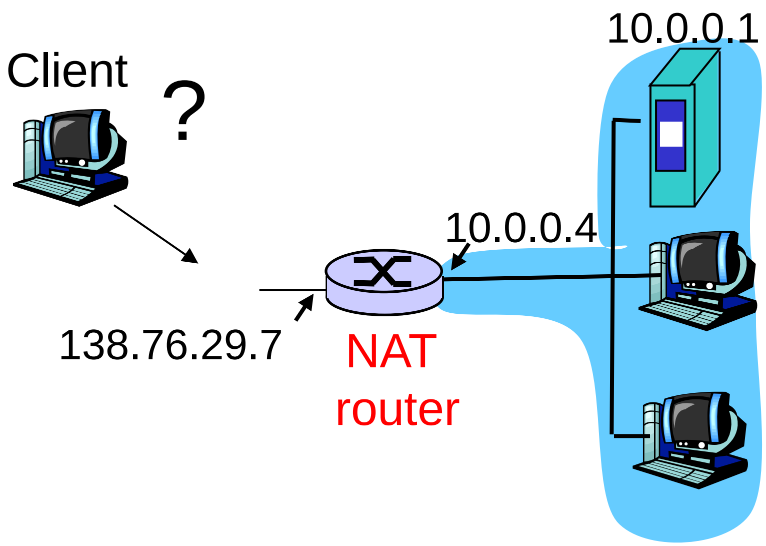
Network layer service models:

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

- Internet model being extented: Intserv, Diffserv
  - (will study these later on)

NAT traversal problem

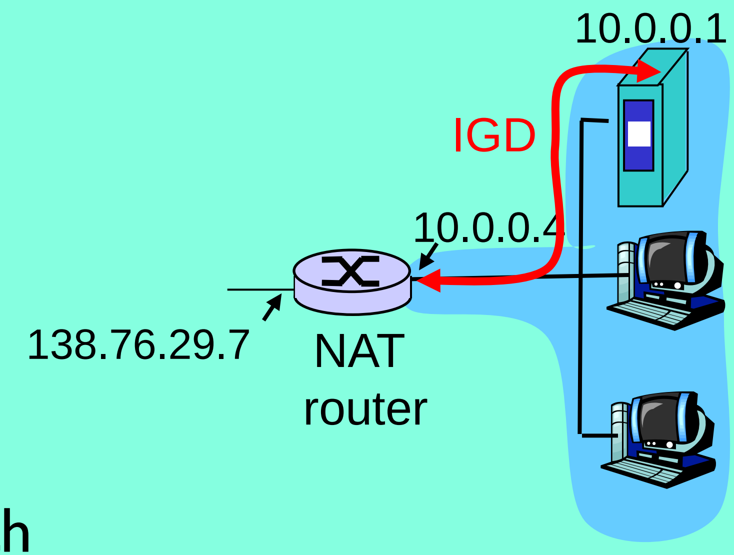
- client want to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can't use it as destination addr)
  - only one externally visible NATted address: 138.76.29.7
- solution 1 (manual): statically configure NAT to forward incoming connection requests at given port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



NAT traversal problem

- solution 2 (protocol) : 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 (application): relaying (used in Skype)
  - NATed server establishes connection to relay
  - External client connects to relay
  - relay bridges packets between two connections