MIEEC

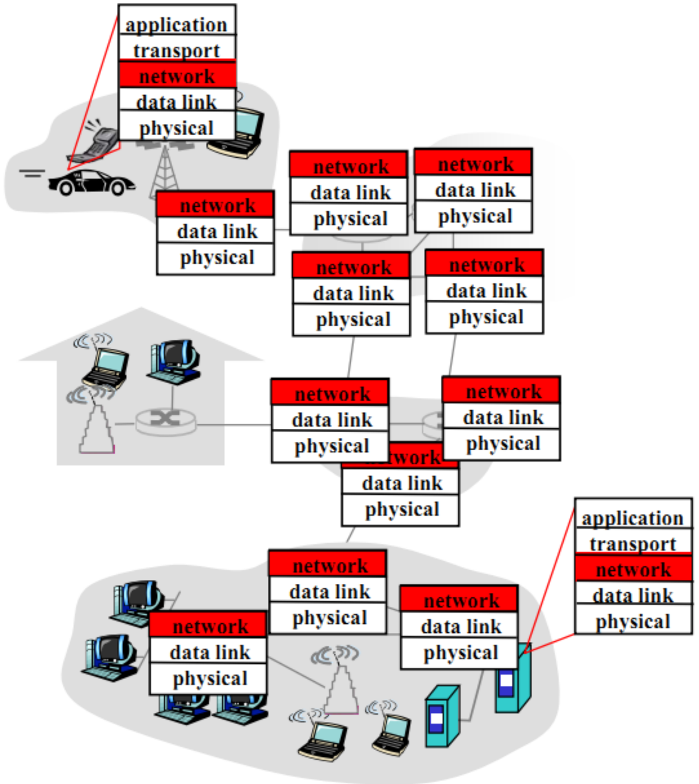
Computer Networks

Lecture note 6

Network layer

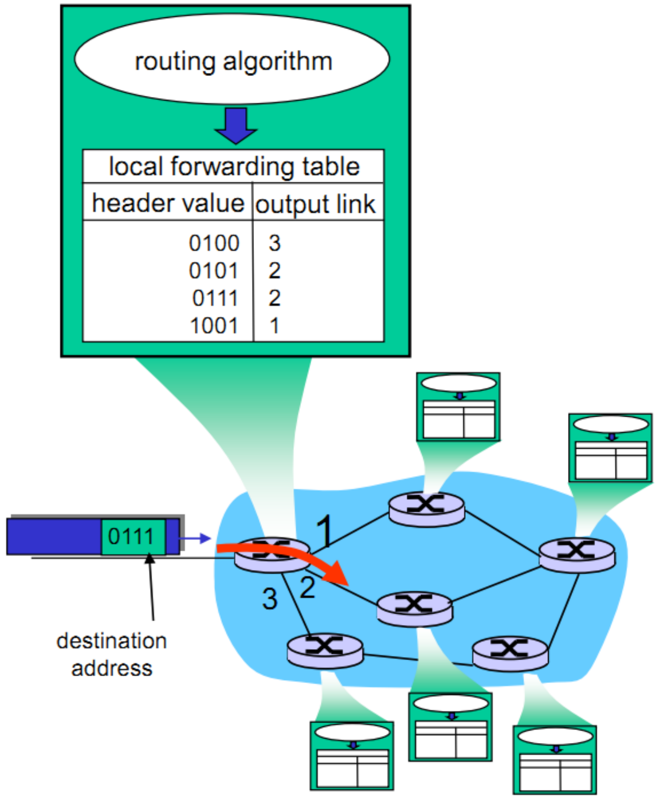
Overview

- Network layer
  - Transports packets
  - From sending to receiving host
  - Through path of routers
- Sender
  - Encapsulates transport data into packets
  - Sends packets
- Receiver
  - Receives packets
  - Delivers data to transport layer
- Router
  - Receives packets on input interface
  - Analyzes network layer header
  - Forwards packets to adequate output interface



Main functions

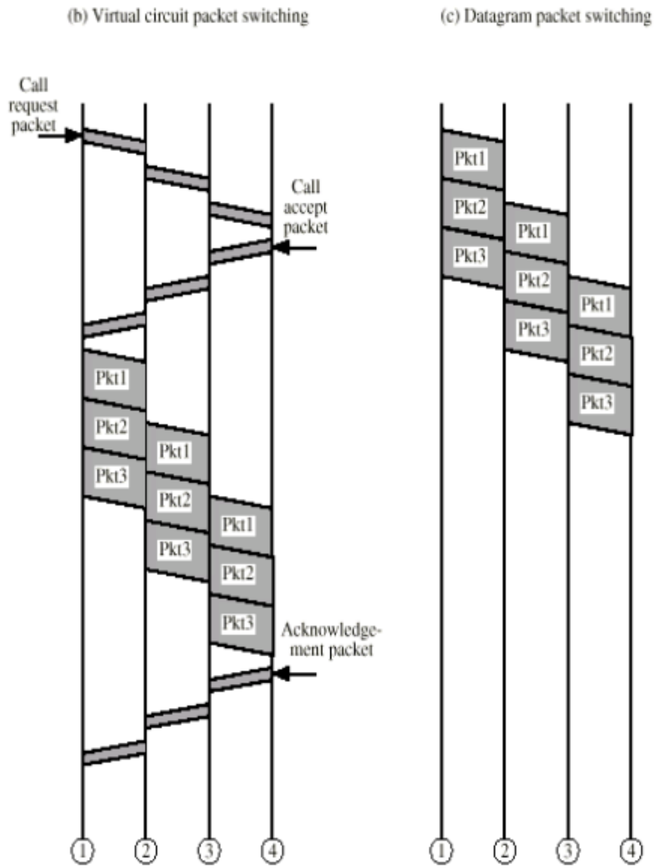
- Forwarding
  - packet
  - input interface => output interface
  - table lookup
- Routing
  - Determine route (i.e. path)
  - Taken by packets from source to destination
  - Algorithms, shortest path
  - Used to fill forwarding table



Virtual circuits and datagram networks

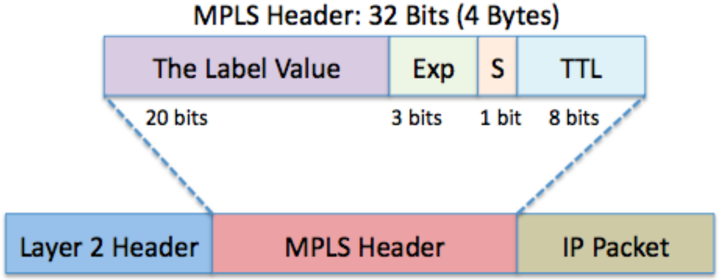
Connection and Connectionless network service

- Connection-less service
  - Datagram network
- Connection-oriented service
  - Virtual circuit network



Virtual Circuit

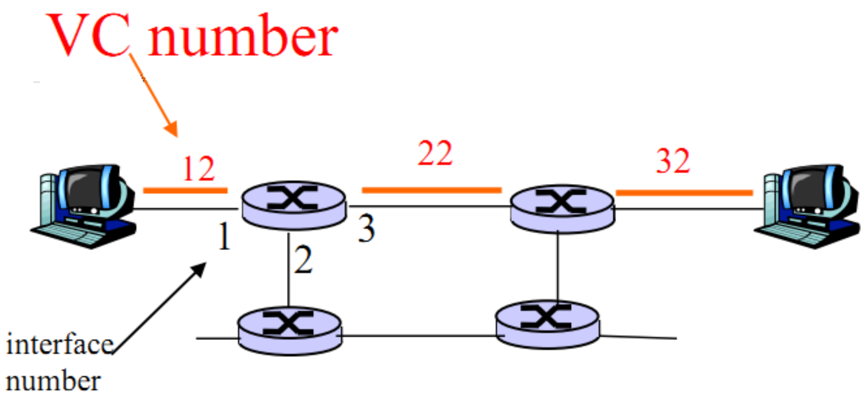
- Steps/phases
  1. Circuit establishment
  2. Data transfer
  3. Circuit termination



- Each VC has a pre-defined path
  - From source to destination
  - Sequence of VC identifiers, one for each link
  - Packet carries VC identifier (local, i.e. label)

- Router
  - Maintains state for each VC
  - Can allocate resources for each VC (bandwidth and buffers)

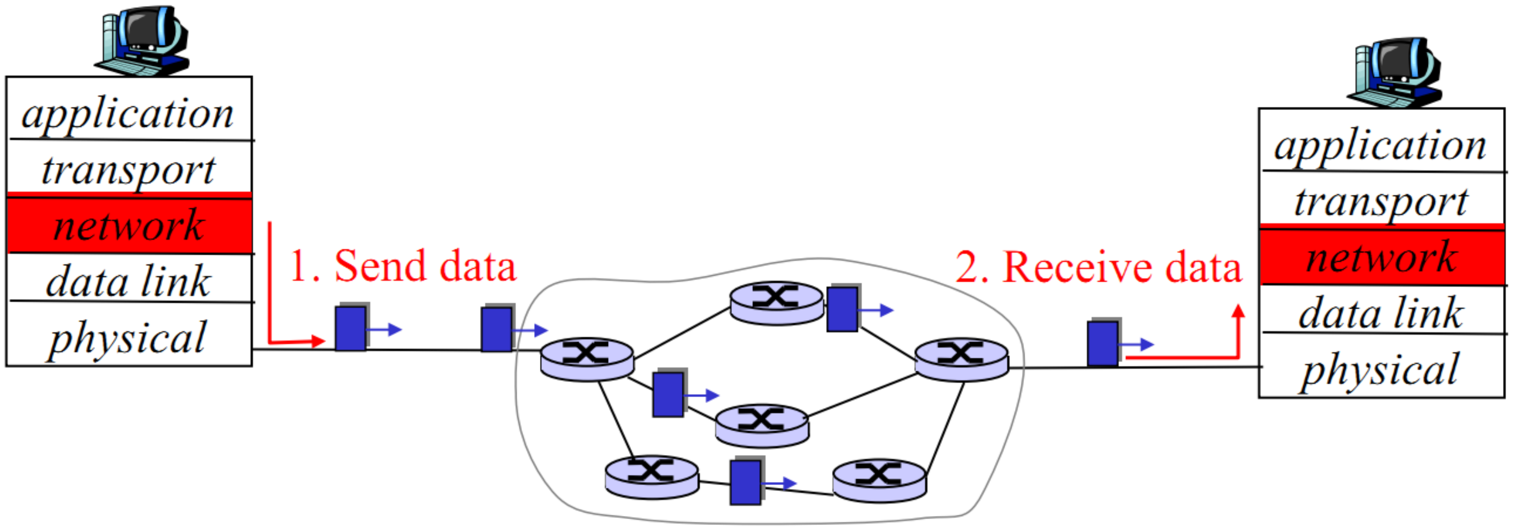
Forwarding table, VC



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

Datagram Networks

- No circuit establishment
  - No concept of “circuit”
- Forwarding lookup
  - uses destination host address
- Packets may follow different paths
  - even if they are generated by the same application



Forwarding table, Datagram

Destination Address Range	Output 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

2^{32} possibilities in IPv4

TO THINK

- 2^{32} possibilities in IPv4

- How do you reduce this number?

```
DA: 11001000  00010111  00010110  10100001

DA: 11001000  00010111  00011000  10101010
```

VC vs. Datagram Networks

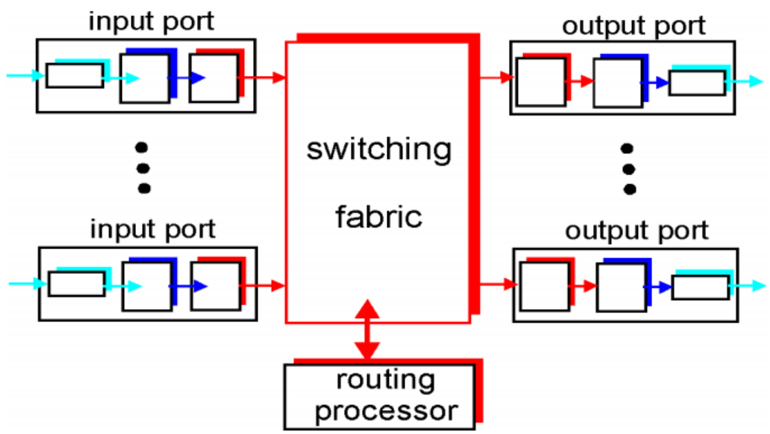
Issue	Datagram subnet	Virtual-circuit subnet
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Router architecture

Overview

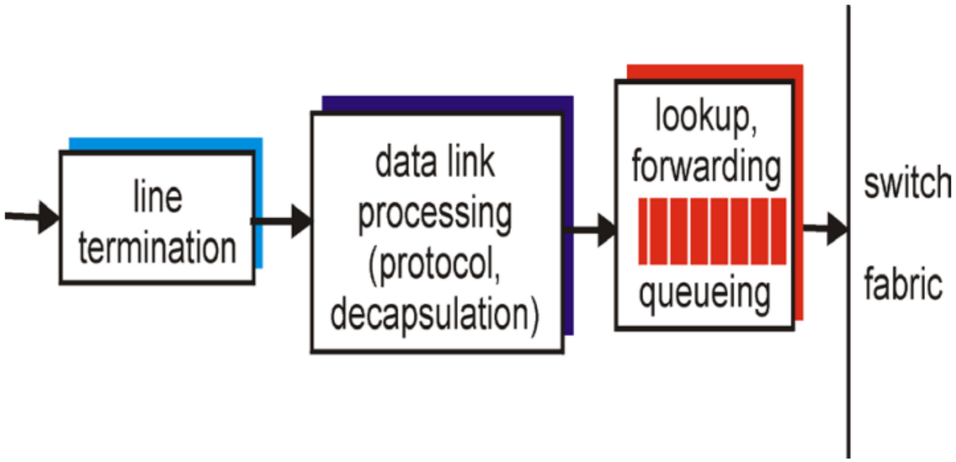
- Forwarding, DATA PLANE
  - Look up routing table
  - Send packets from input to output interface
- Routing, CONTROL PLANE
  - Routing protocols (RIP, OSPF, BGP)
  - => Another class

- Main components
  - Input port
  - Output port
  - Switching fabric
  - Routing processor

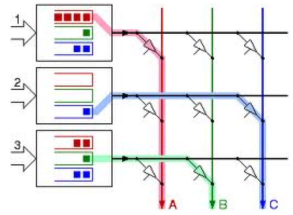


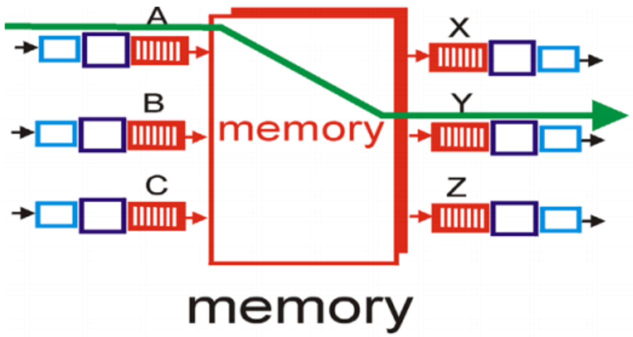
Input port

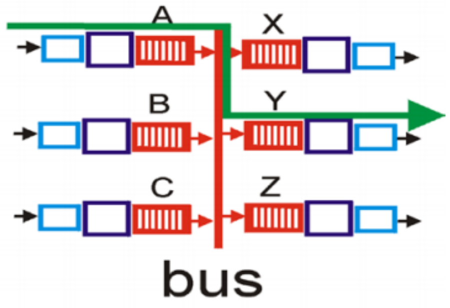
- Look up output port
  - Table in input port memory
- Send to switch fabric
- Goal: process at line speed
  - Queuing if packets arrive faster than forwarding rate

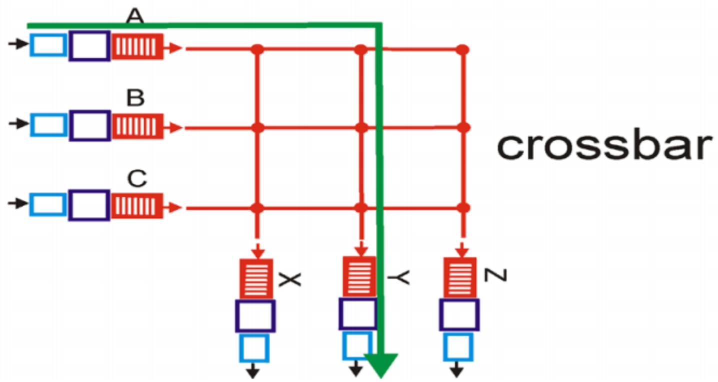


Switching fabrics



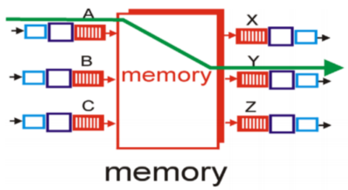


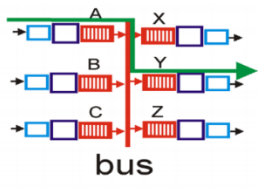


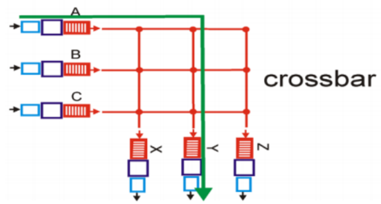


TO THINK

- Which switching fabric is better?



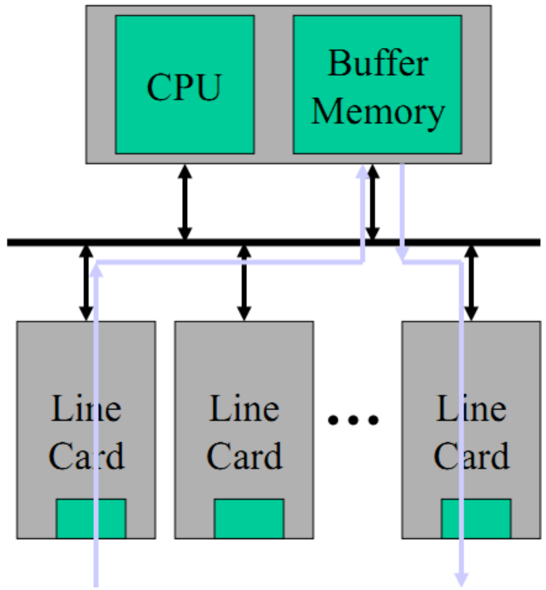




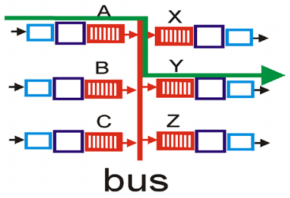
  - For what? Speed, cost?

Memory switch

- First generation router
- Switching controlled by CPU
  - Normal PC with PCI cards
- Each packet
  - goes through bus twice

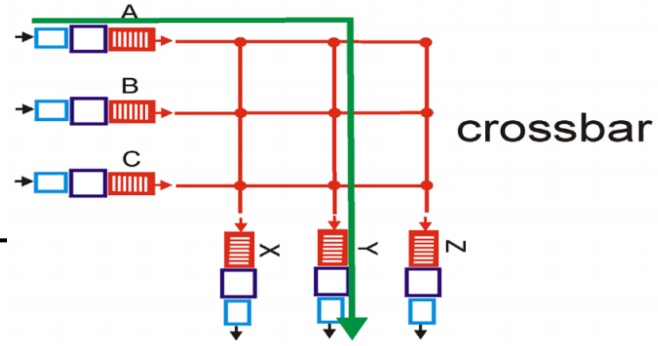


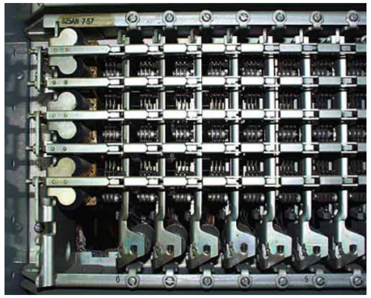
Shared bus switch

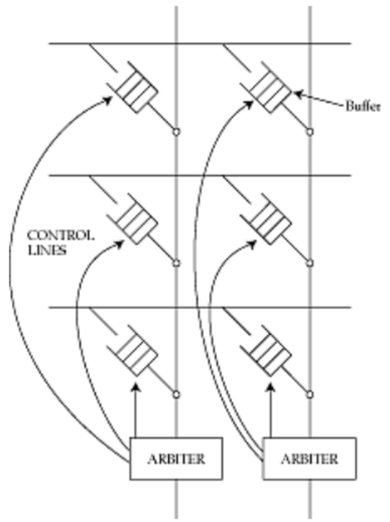


- Direct transfer from input to output
  - CPU not involved
- Bus contention
  - Total switching rate limited by bus bandwidth
- E.g. Cisco 5600
  - 32 Gbit/s
  - Enough for access and enterprise routers

Crossbar switch

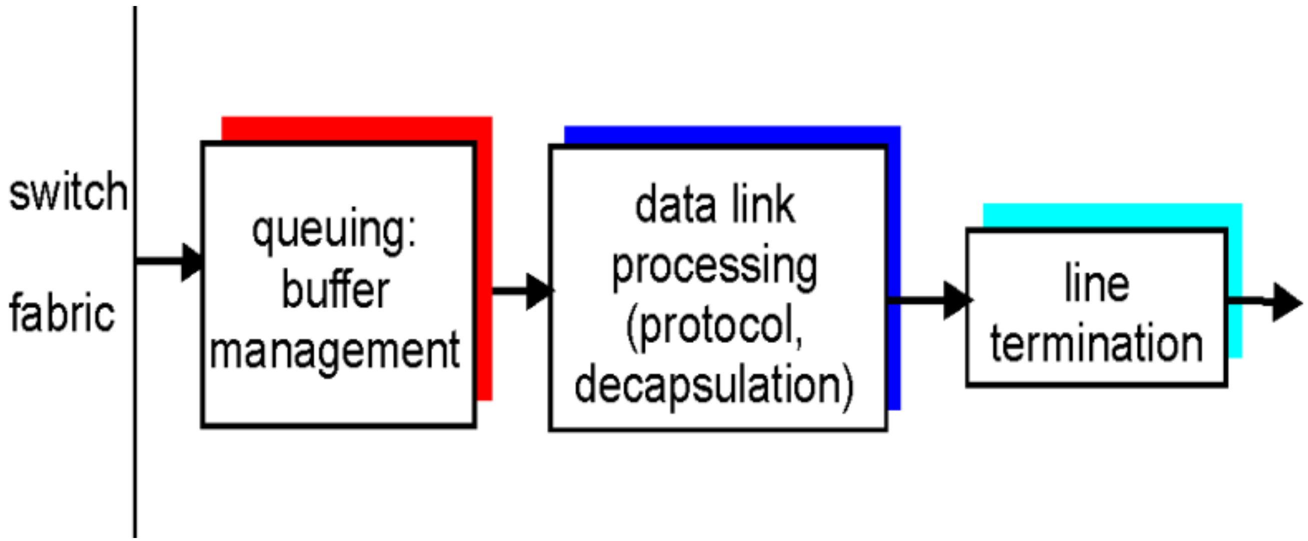






- Simultaneous forwarding
  - A -> Y, B -> X
- N^2 interconnections
  - May contain buffers
- Overcomes bus bandwidth limitations
- Cisco 12000
  - 60 Gbit/s

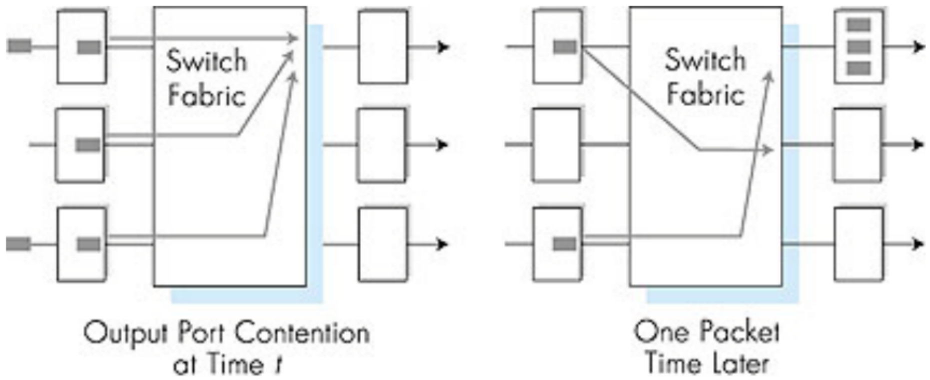
Output port



- Buffering
- Scheduling
  - Which packets go first, priority

Queuing - output port

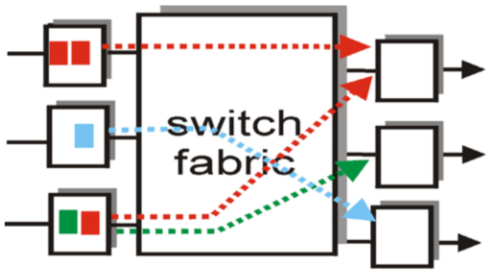
- Contention at the output port



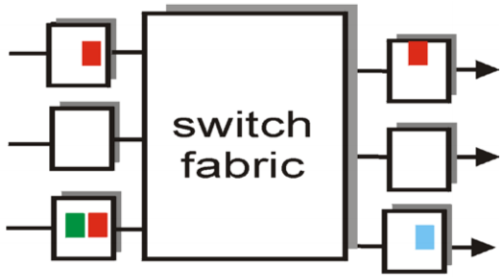
- Queuing when
  - Arrival rate at output port > line rate
    => queuing delay and loss if output buffer fills up

Queuing - input port

- Can happen when:
  - Total line rate > switching fabric rate
  - Head of line blocking
    - Packet at head of queue prevents packets behind it to be switched



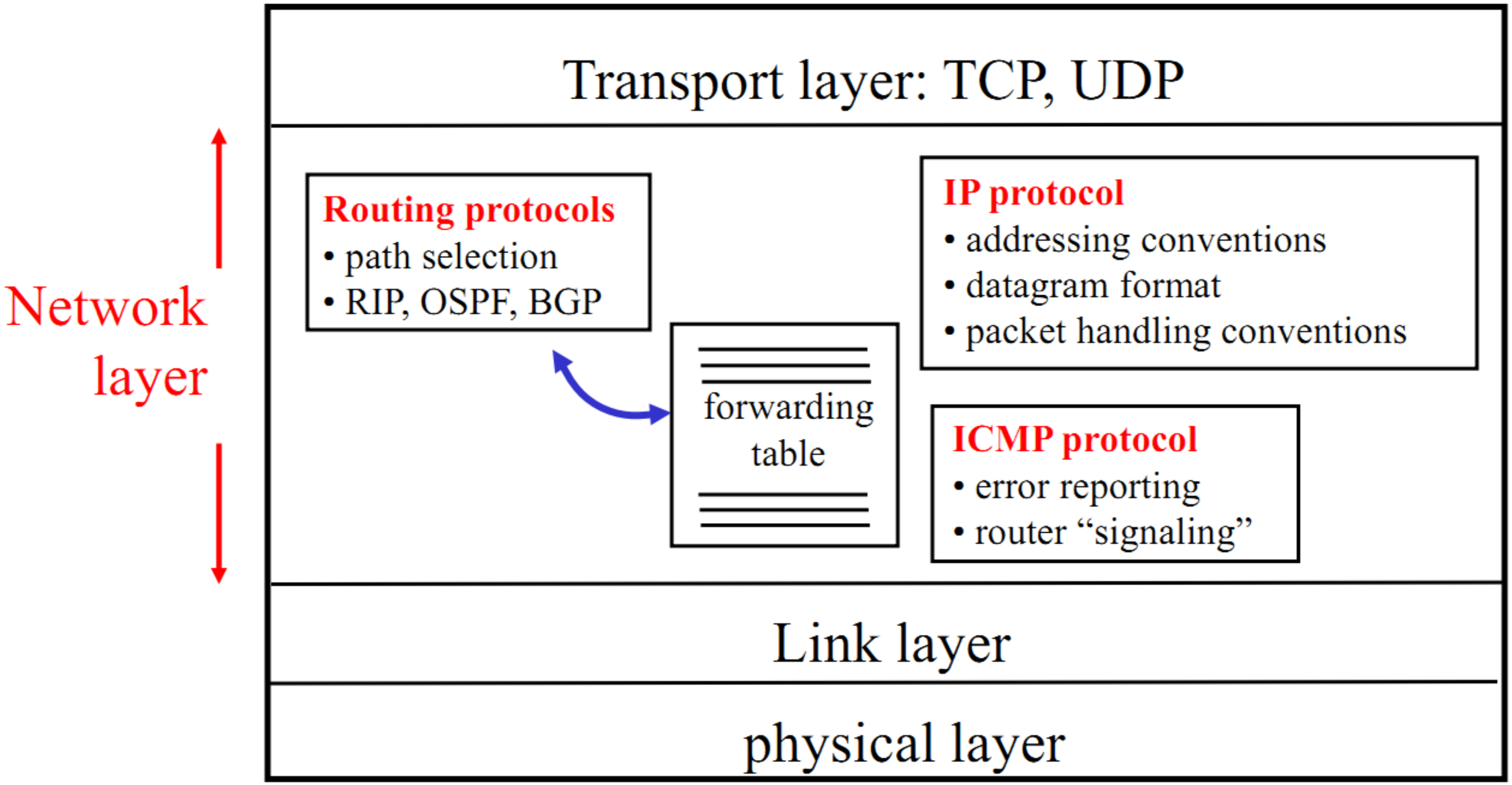
output port contention
at time t - only one red
packet can be transferred



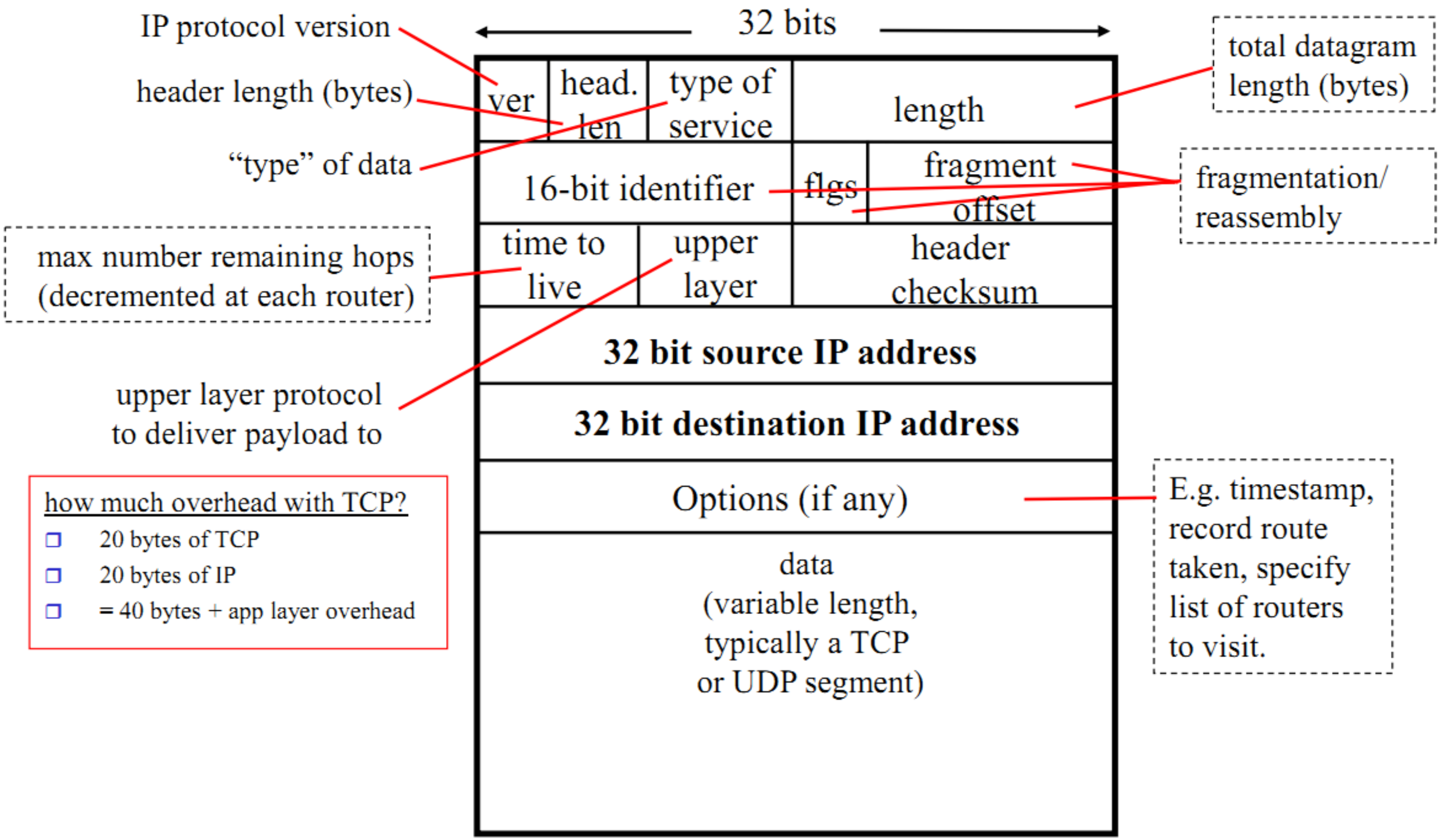
green packet
experiences HOL blocking

Internet Protocol

Host/Router functions



IP Datagram Format

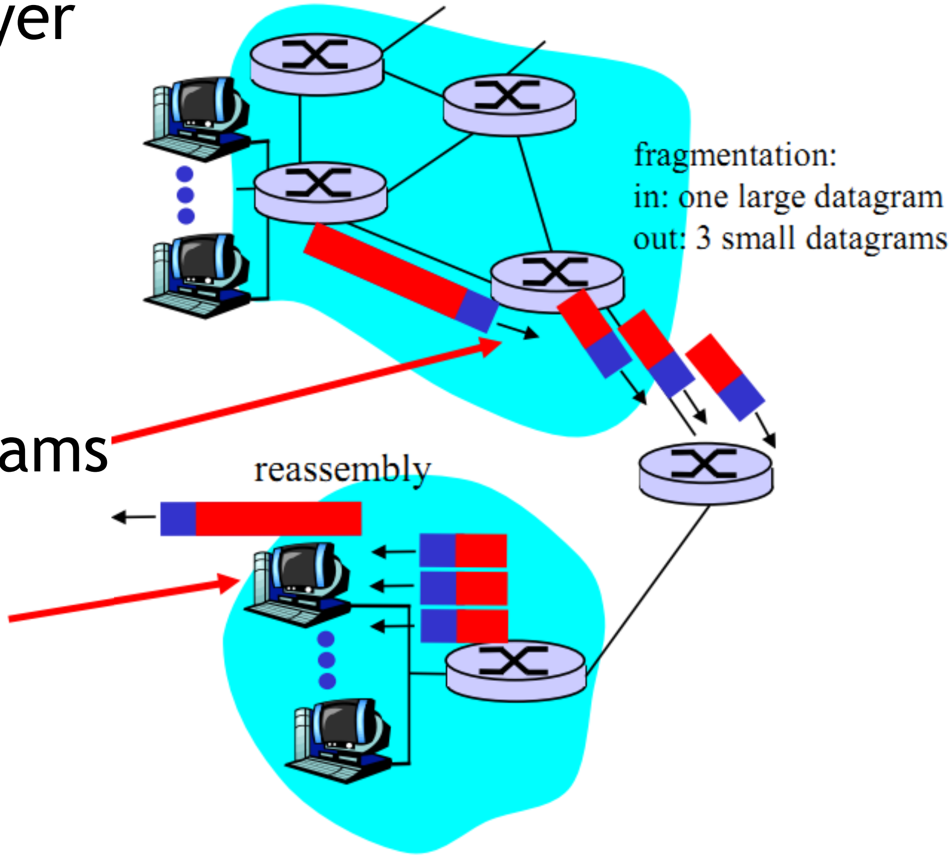


how much overhead with TCP?

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

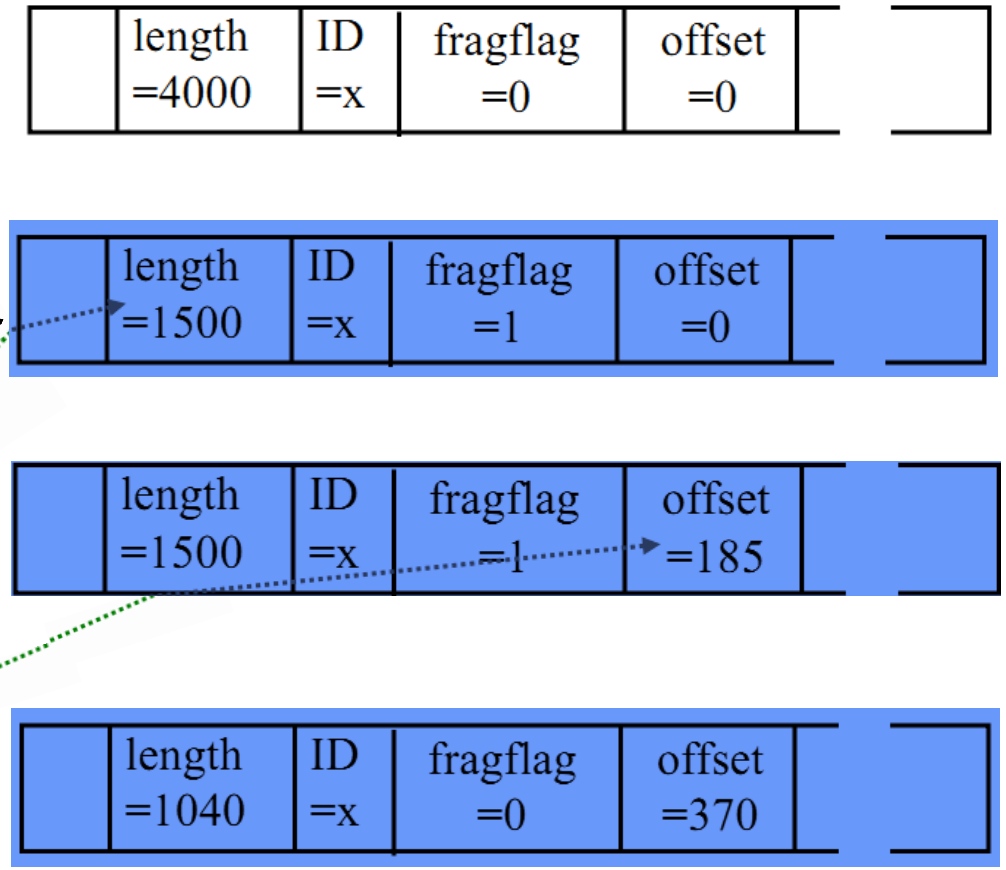
IP Fragmentation and reassembly

- MTU on network links
  - Max. packet size
  - Largest possible link-layer frame
  - Different MTUs for different link layers
- Large IP datagram is fragmented
  - 1 datagram => n datagrams
- "Reassembled" at final destination
  - IP header to identify, order fragments



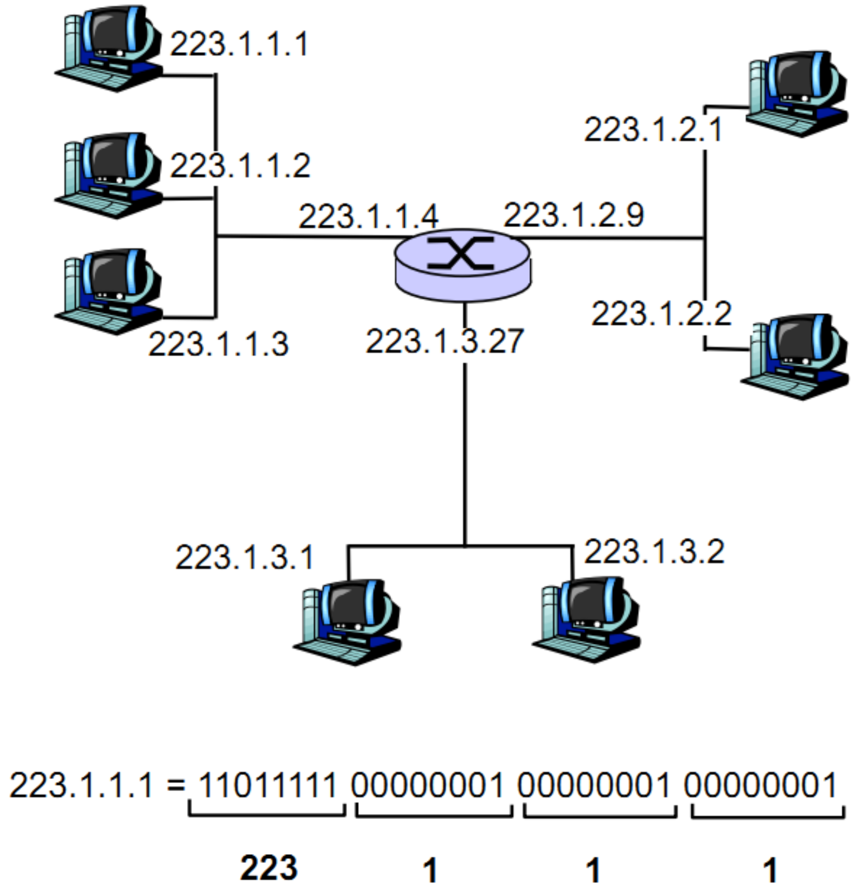
IP Fragmentation and reassembly

- Example
- 4000 byte datagram
  - 3980 payload
  - 20 bytes IP header
- 1500 byte MTU
  - 1480 byte in data field
  - Offset = 1480/8 = 185



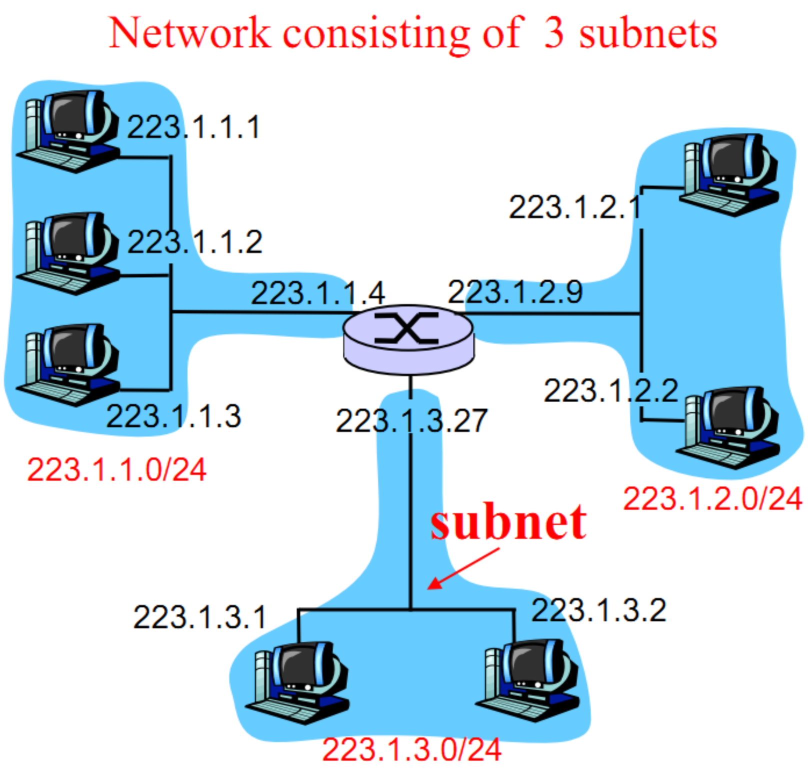
IP Addressing

- IP addresses
  - 32 bit identifier
  - Network interface
    - Of host or router
- Network interface
  - Connection between
    - Host/Router and physical link
  - Routers have multiple interfaces



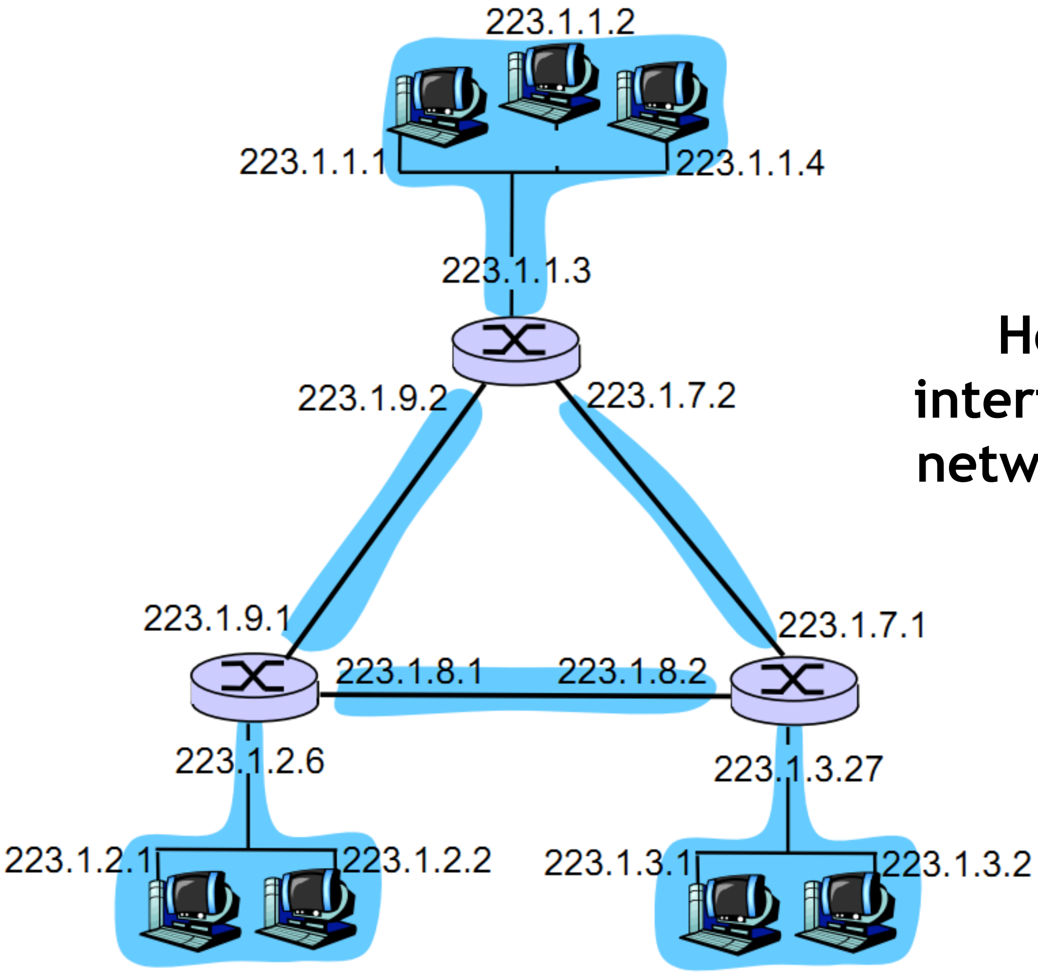
Subnets

- IP Address
  - Subnet part
    - higher order bits
  - Host part
    - Lower order bits



- Subnet
  - Set of With same network identifier
    - i.e. IP Address, subnet part
  - Can send packets directly
    - Without forwarding at the router
    - Use the same link-layer technology

6 Subnets



How many interfaces does a network support?

IP Addressing – CIDR

- Classless Inter-Domain Routing
  - Subnet portion has arbitrary length
    - What's the other approach?
  - Address format a.b.c.d/x
    - x is the number of bits of the subnet address

subnet part | host part

11001000 00010111 00010000 00000000

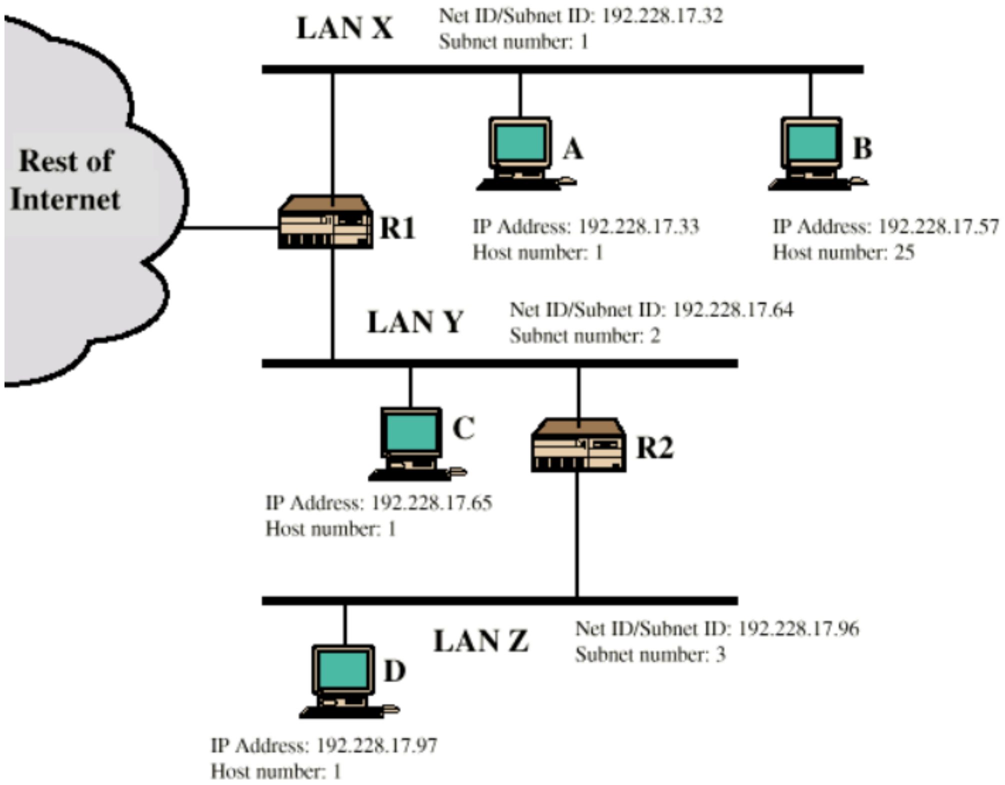
200.23.16.0/23

Special IP addresses

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		This host
0 0 . . . 0 0	Host	A host on this network
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		Broadcast on the local network
Network	1 1 1 1 . . . 1 1 1 1	Broadcast on a distant network
127	(Anything)	Loopback

Forming sub-networks

Network 192.228.17.0/24 is divided into 8 sub-networks, same size 27 bit network mask

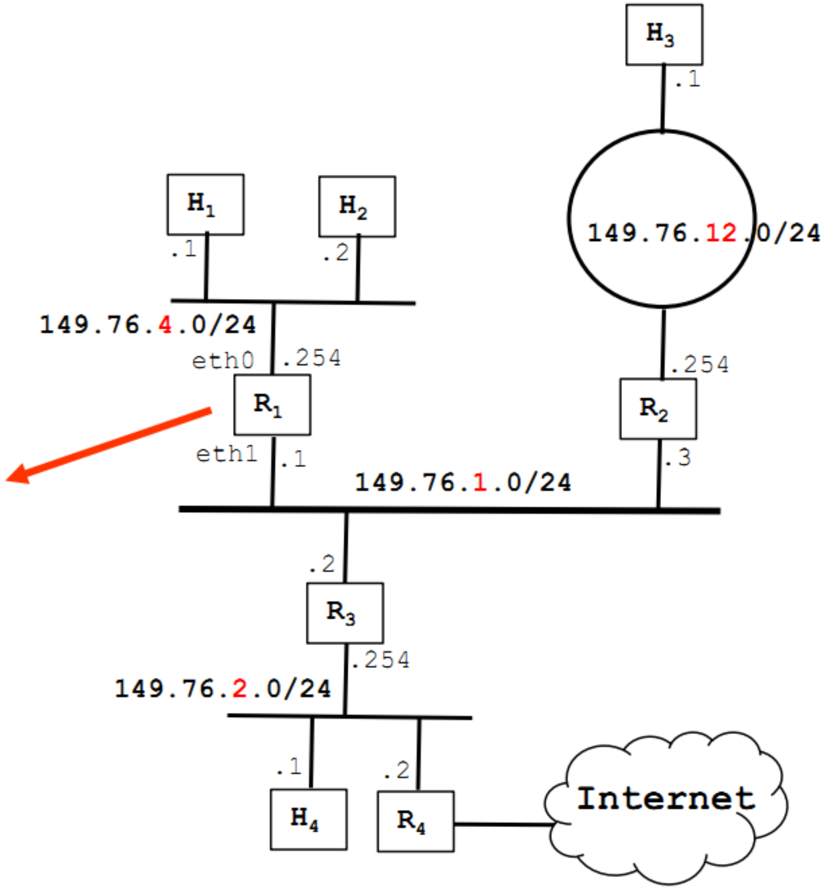


- Subnet mask
  - 27 bits
- Host mask
  - 32-27 = 5 bits
- 2^5 = 32
  - All 0's: subnet address
  - All 1's: broadcast address
  - 30 addresses for interfaces
    - Hosts and routers

- Example sub-networks
  - 192.228.17.0/27
  - 192.228.17.32/27
  - 192.228.17.64/27
- NOT 192.228.17.14/27
  - Why not?

Forwarding table

Destination	Gateway	Interface
149.76.1.0/24	-	eth1
149.76.2.0/24	149.76.1.2	eth1
149.76.4.0/24	-	eth0
149.76.12.0/24	149.76.1.3	eth1
0/0	149.76.1.2	eth1



IP Forwarding Function

- Forwarding table entries are pairs of:
  - {network address/mask, output port}
- When a datagram arrives with destination address A:
  - For each entry in forwarding table
    - val = A & mask* (e.g. mask=8, mask*=255.0.0.0)
    - if(val == networkAddress & mask*)
      - Add output port to set of candidates
  - Select port with longest mask

Longest prefix match

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

Examples. Which Interface?

DA: 11001000 00010111 00010110 10100001

DA: 11001000 00010111 00011000 10101010

IP Forwarding Function Example

- Example
- Forwarding table
  - <128.32.1.5/16,1>
  - <128.32.225.0/18,3>
  - <128.0.0.0/8,5>
- Destination A=128.32.195.1
- Set of candidate ports {1,3,5}
  - Selected port : 3 (largest mask, most specific)

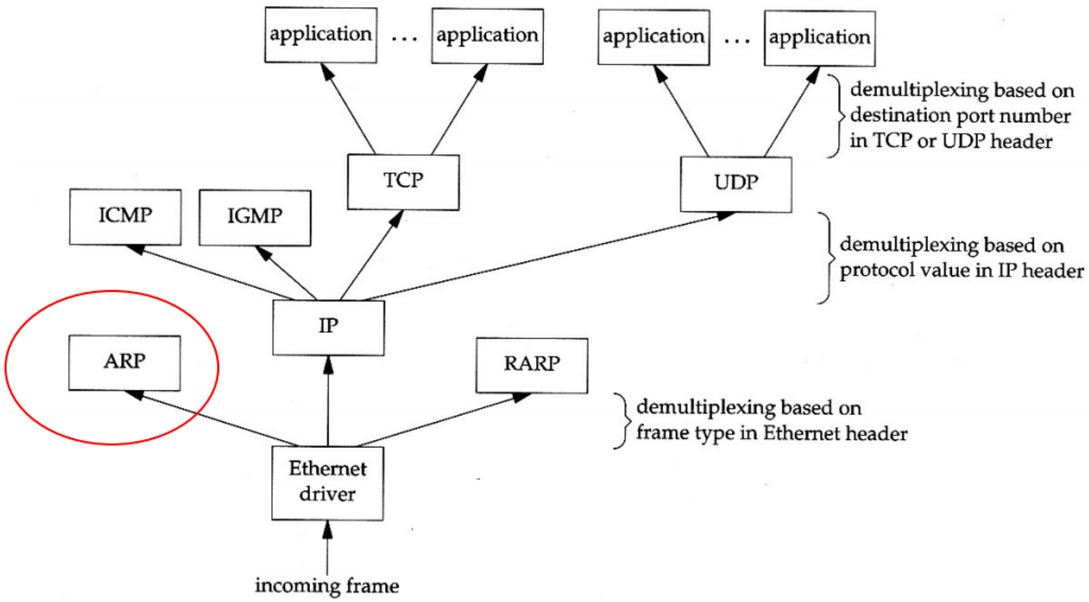
Address Resolution Protocol

Demultiplexing

- Ethernet header (type)
  - IP 0x0800
  - ARP 0x0806
  - RARP 0x8035
  - IPX 0x8037
  - IPv6 0x86DD
  - MPLS 0x8847

- IP header (protocol)
  - ICMP 1
  - IGMP 2
  - TCP 6
  - UDP 17

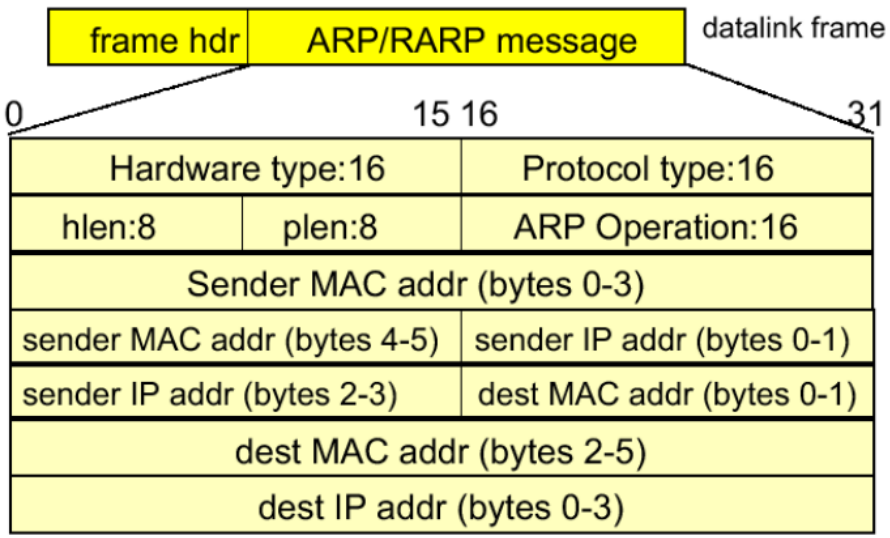
- TCP/UDP header (port)
  - FTP 21
  - Telnet 23
  - HTTP 80
  - SMTP 25



ARP - Address resolution protocol

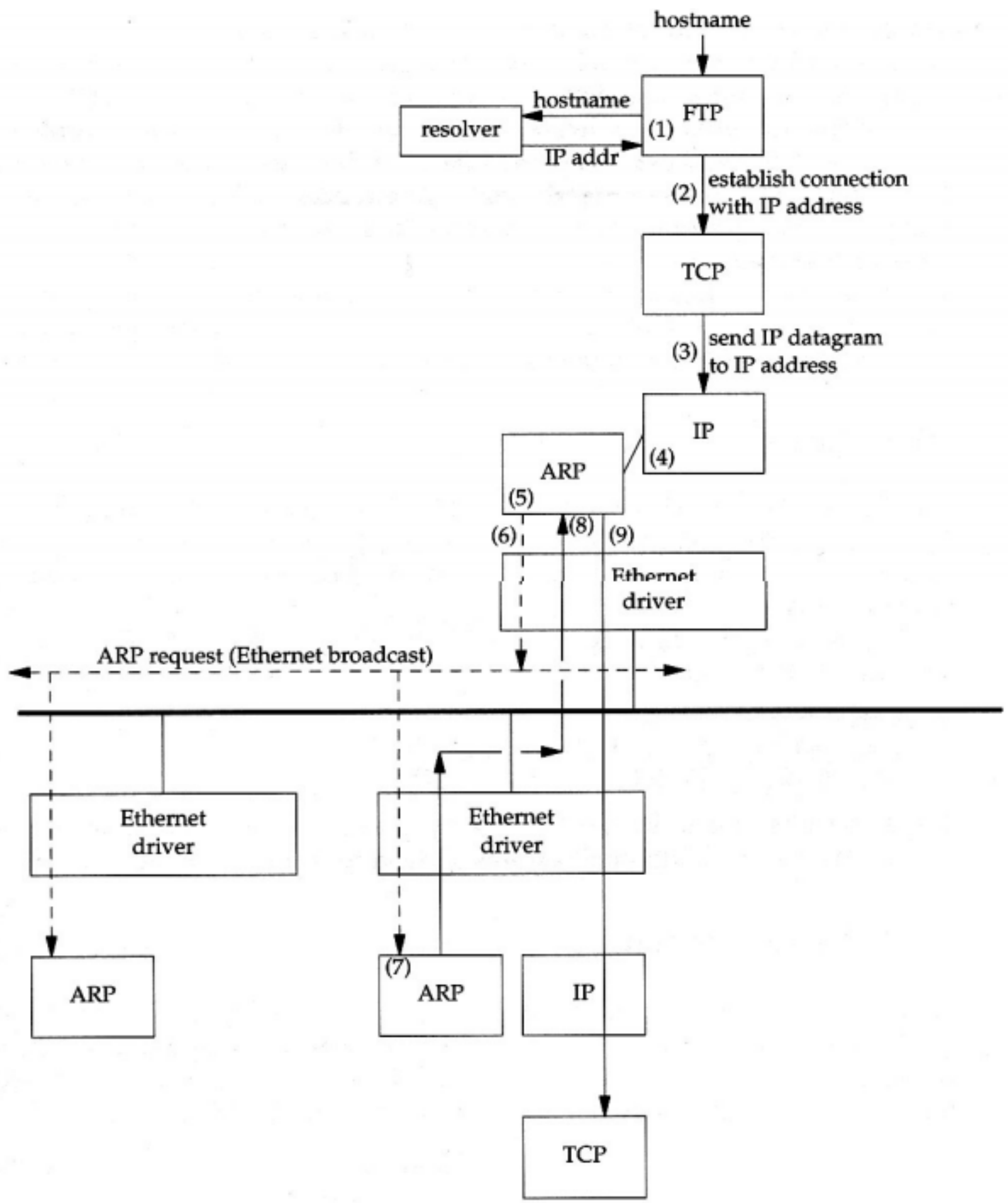
IP address ⇔

⇔ MAC address

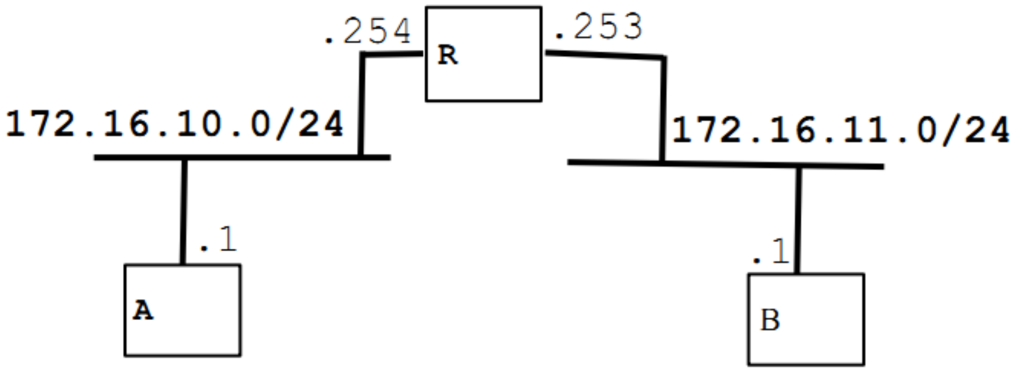


- **hardware type : Ethernet=1 ARCNET=7, localtalk=11**
- **protocol type : IP=0x800**
- **hlen : length of hardware address, Ethernet=6 bytes**
- **plen : length of protocol address, IP=4 bytes**
- **ARP operation : ARP request = 1, ARP reply = 2**
  RARP request = 3, RARP reply = 4

ARP Example


hostname
hostname
FTP
resolver
(1)
IP addr
(2)
establish connection
with IP address
TCP
(3)
send IP datagram
to IP address
IP
ARP
(4)
(5)
(6)
(8)
(9)
Ethernet
driver
ARP request (Ethernet broadcast)
Ethernet
driver
Ethernet
driver
ARP
(7)
ARP
IP
TCP

TO THINK



- Assume A sends IP packet to B via router R
- What are the observed MAC and IP source and destination addresses?
  - Routing table
- What's the role of ARP in this scenario?

Obtaining IP Addresses

TO THINK

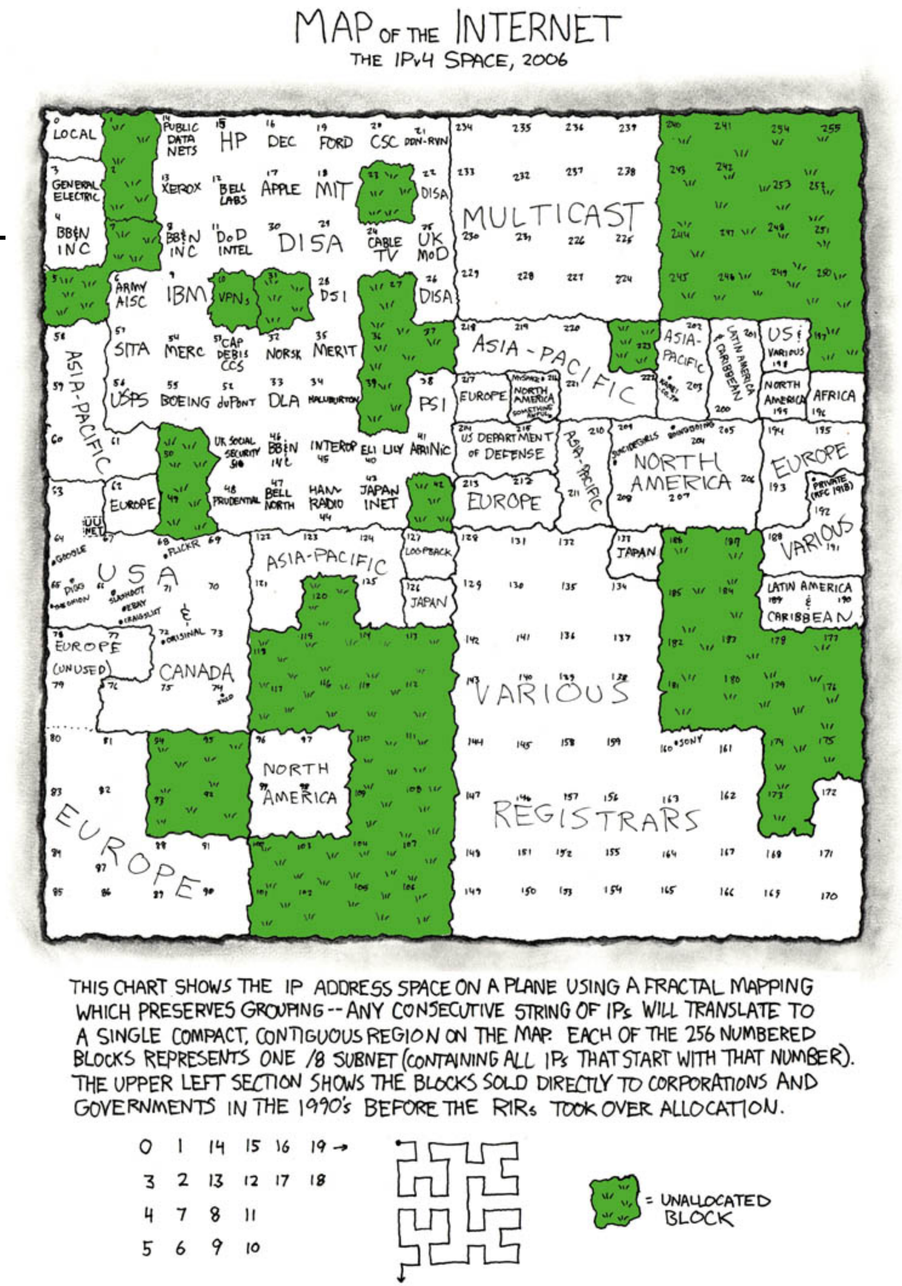
- How does a host get the subnet part of its address?

- How does a network get the subnet part?

Obtaining IP Addresses

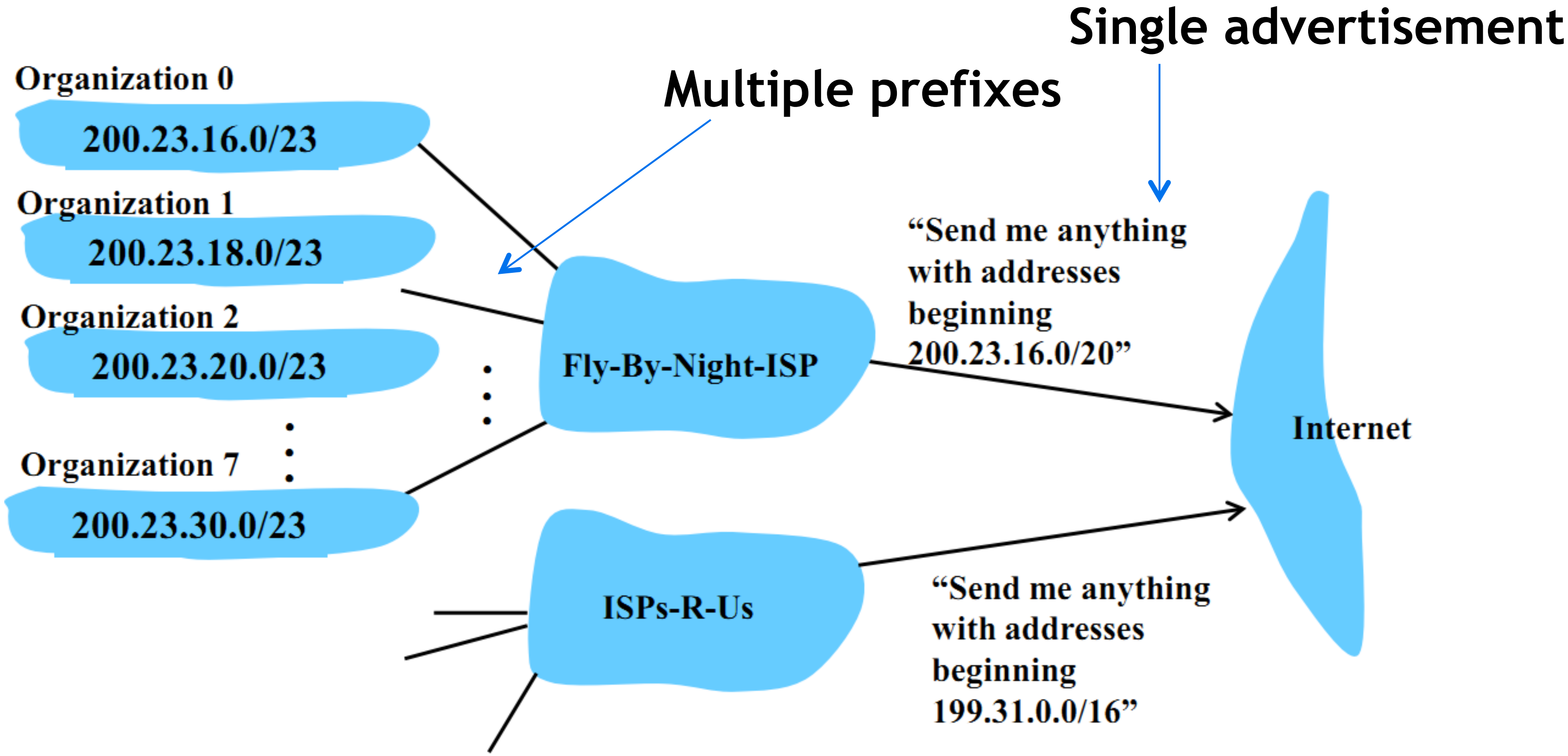
- Subnet part allocated by Internet provider
  - Typically a portion of the ISP's address space

- **ISP's block**
  - **200.23.16.0/20**
- Organization 0
  - 200.23.16.0/23
- Organization 1
  - 200.23.18.0/23
- Organization 2
  - 200.23.20.0/23
- Organization 7
  - 200.23.30.0/23



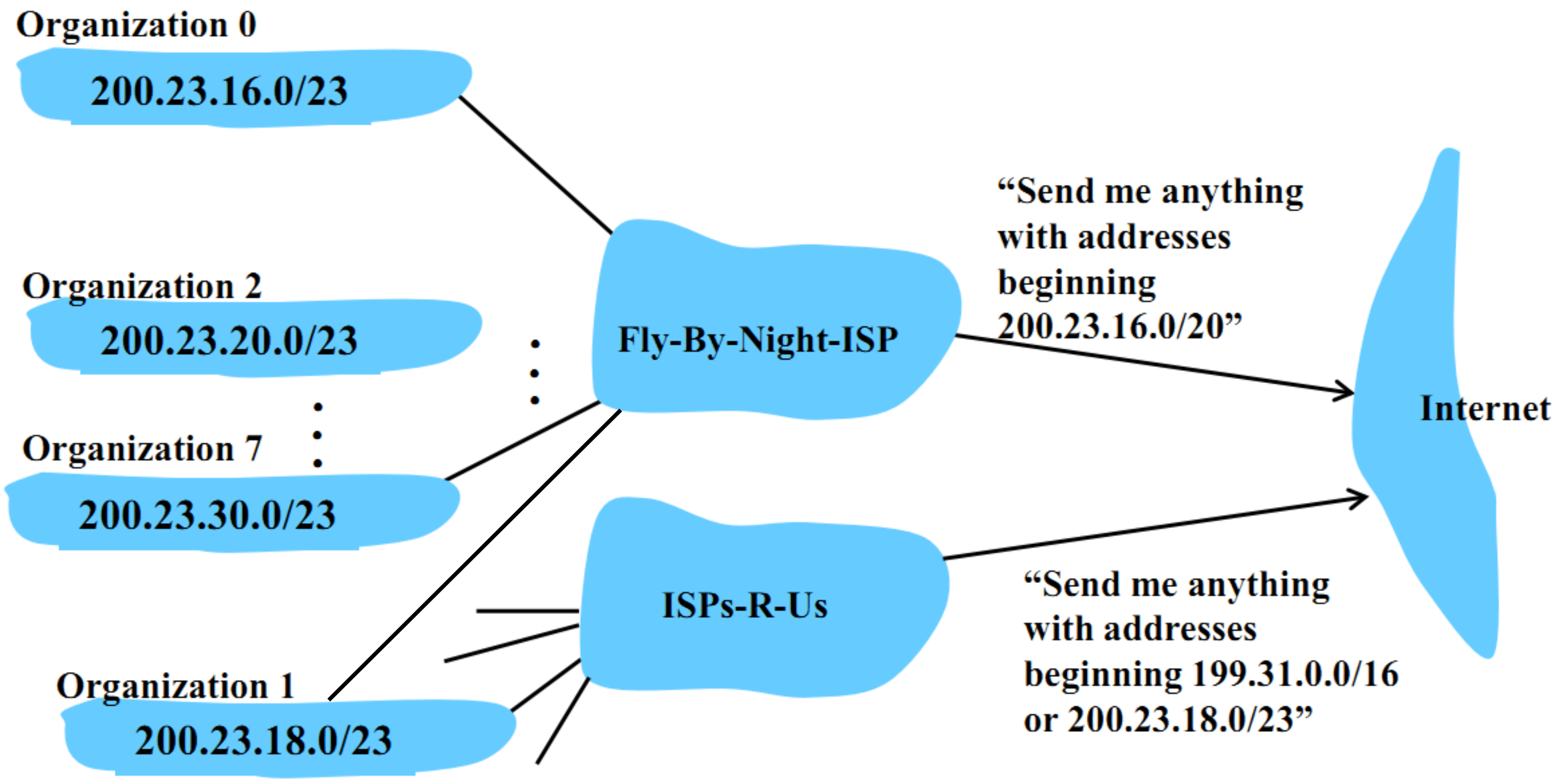
Hierarchical addressing and Route aggregation

- Efficient advertisement of routing information



Hierarchical addressing: More specific route

- ISPs-R-Us has a more specific route to Org1



ICANN

- Where does an ISP get its address block from?
- Internet Corporation for Assigned Names and Numbers
- Allocates addresses
- Manages DNS
- Assigns domain names
  - and provides dispute resolution services

Getting an IP Address: At the host

- How does a host get its IP address?

1) Hard-coded in a file by sysadmin
   - Windows: ctrl-panel – network – config – tcp/ip – properties
   - Linux: /etc/rc.config, ifconfig
2) DHCP, Dynamic Host Configuration Protocol
   - dynamically gets its address from a remote server
   - Plug-and-play

DHCP

Dynamically obtain IP address from server

- when joining the network
- Reuse addresses

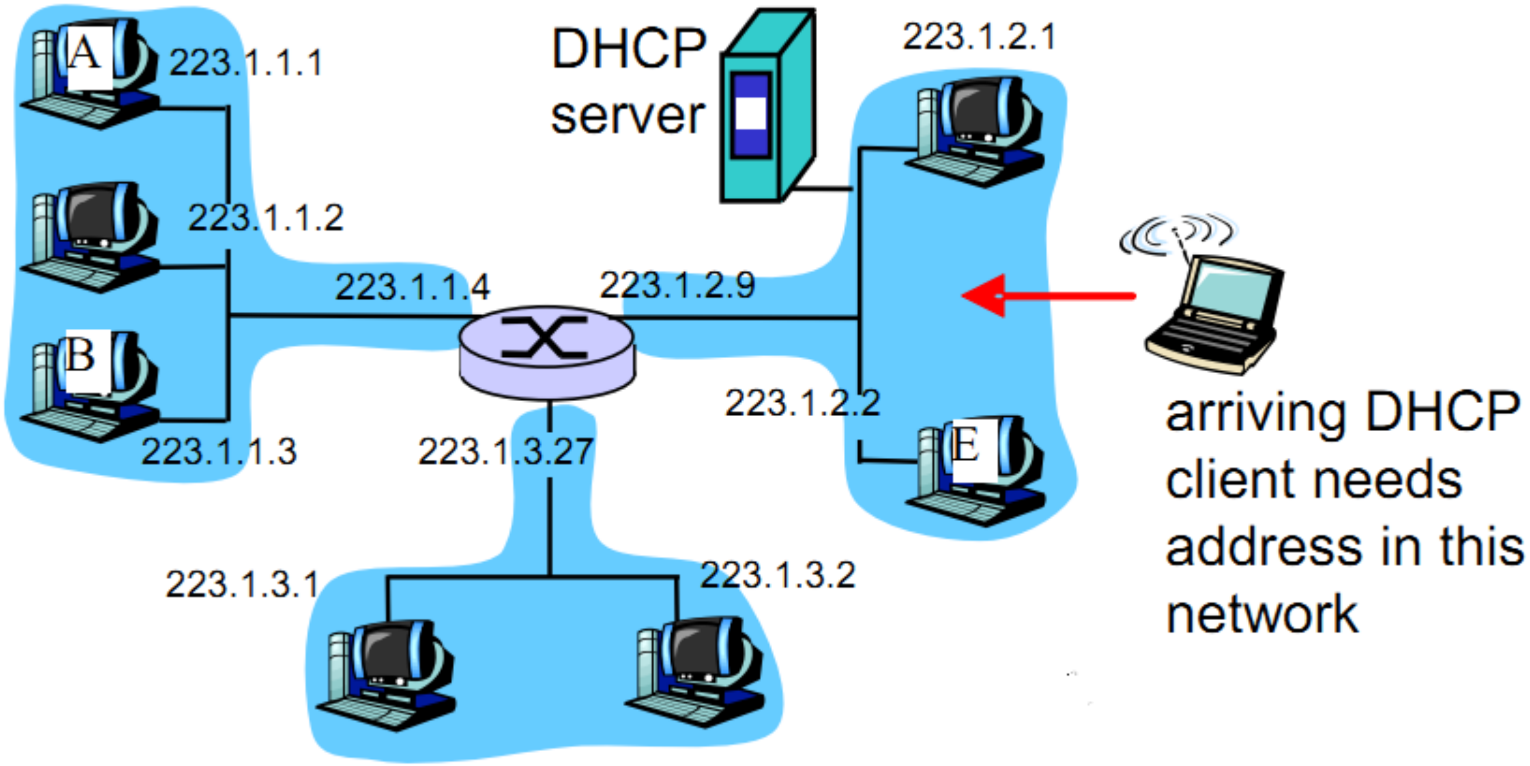
Host broadcasts “DHCP discover” message

DHCP server replies “DHCP offer”

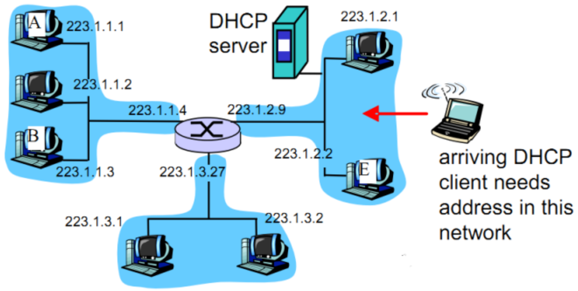
Host requests IP address “DHCP request”

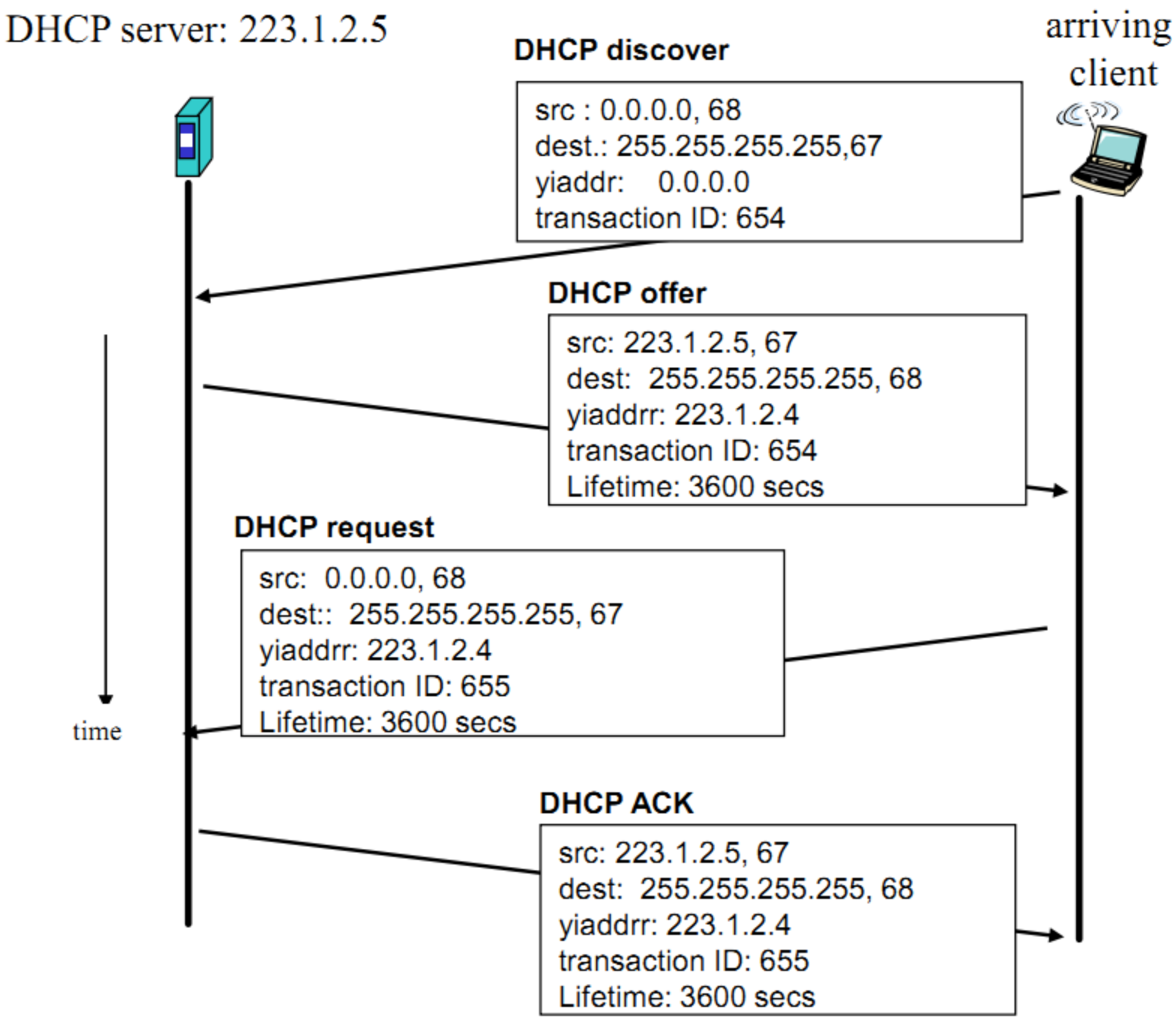
DHCP sends back address “DHCP ack”

DHCP Scenario



DHCP Scenario



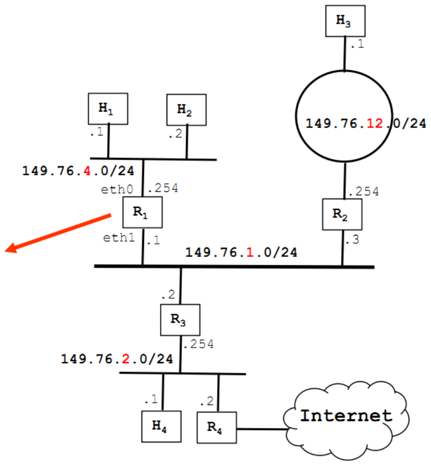


TO THINK

- Is the IP address enough for the newly arrived client?

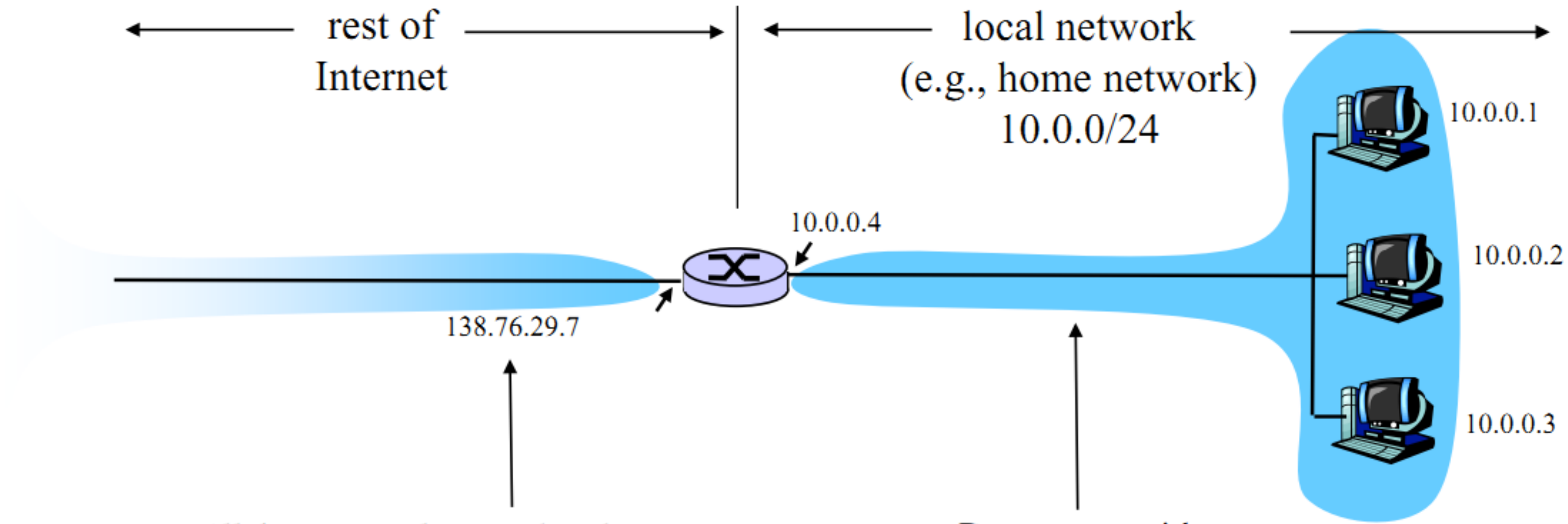
- What other information is relevant?

Destination	Gateway	Interface
149.76.1.0/24	-	eth1
149.76.2.0/24	149.76.1.2	eth1
149.76.4.0/24	-	eth0
149.76.12.0/24	149.76.1.3	eth1
0/0	149.76.1.2	eth1



Network Address Translation

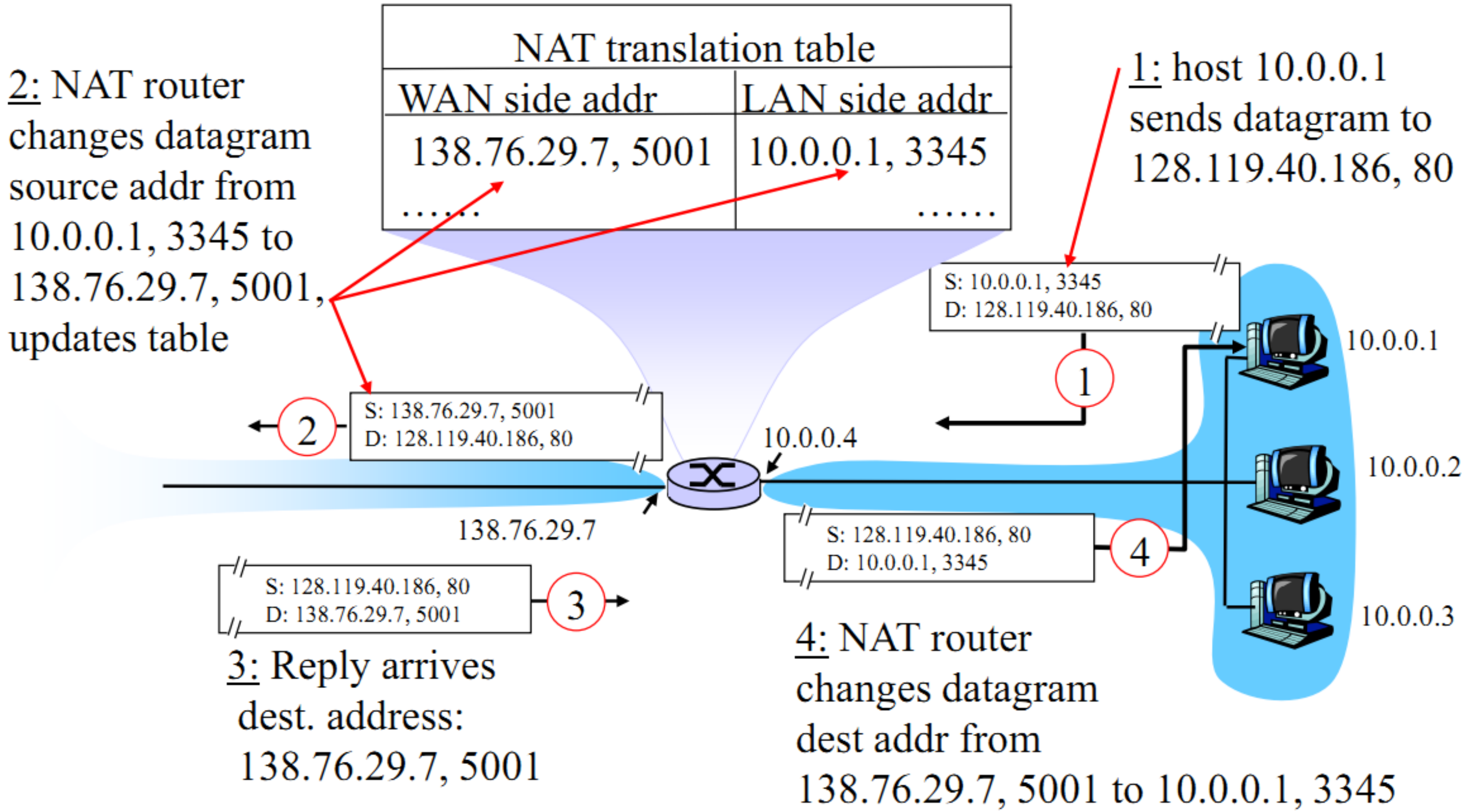
NAT



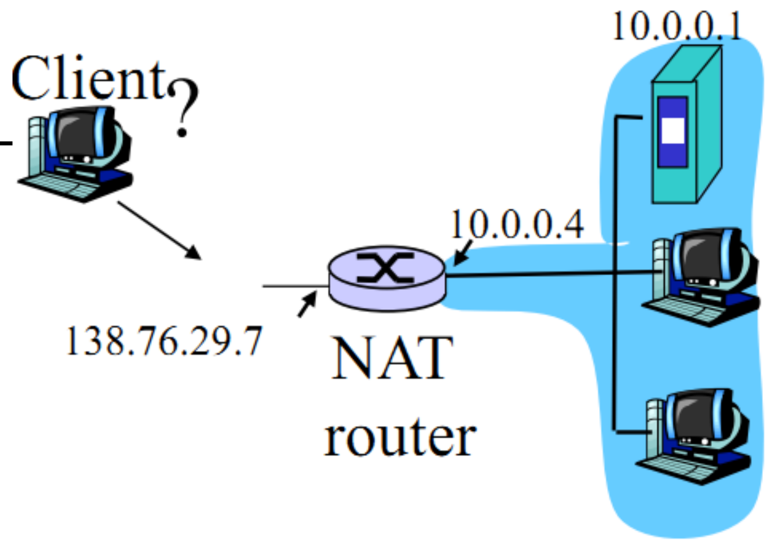
All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0/24 address for source, destination (as usual)

NAT


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
S: 10.0.0.1, 3345
D: 128.119.40.186, 80
1
10.0.0.1
10.0.0.2
10.0.0.3
2
S: 138.76.29.7, 5001
D: 128.119.40.186, 80
10.0.0.4
138.76.29.7
S: 128.119.40.186, 80
D: 10.0.0.1, 3345
4
S: 128.119.40.186, 80
D: 138.76.29.7, 5001
3
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

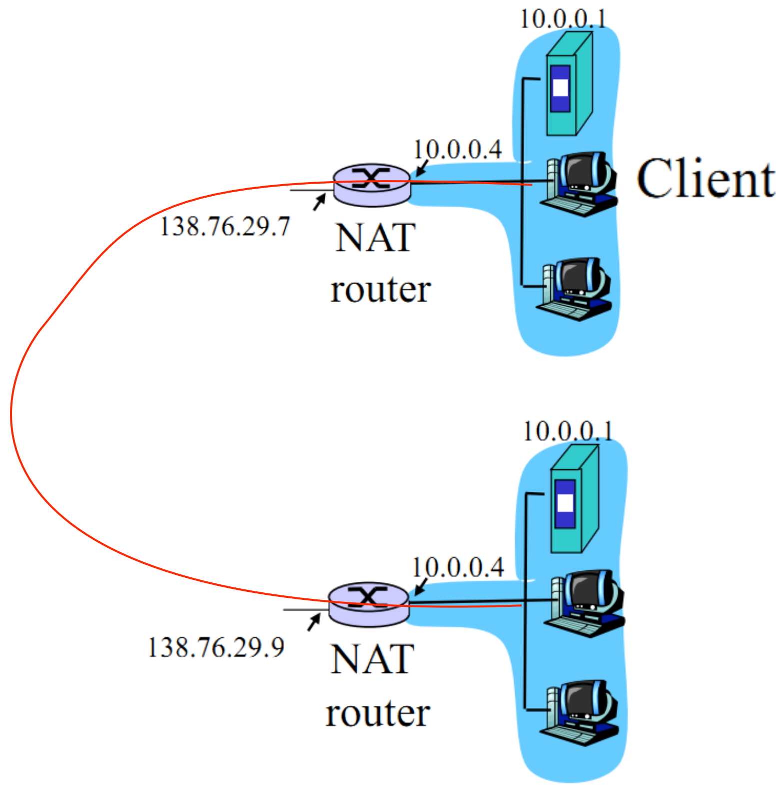
NAT traversal



- Client needs to contact server at 10.0.0.1
  - 10.0.0.1 is private, not announced on the internet
  - One externally visible IP address: 138.76.29.7
- Possible solution: port forwarding
  - Configure NAT statically
  - E.g. {138,76,29,7:2500} always forwarded to 10.0.0.1 on port 2500

TO THINK

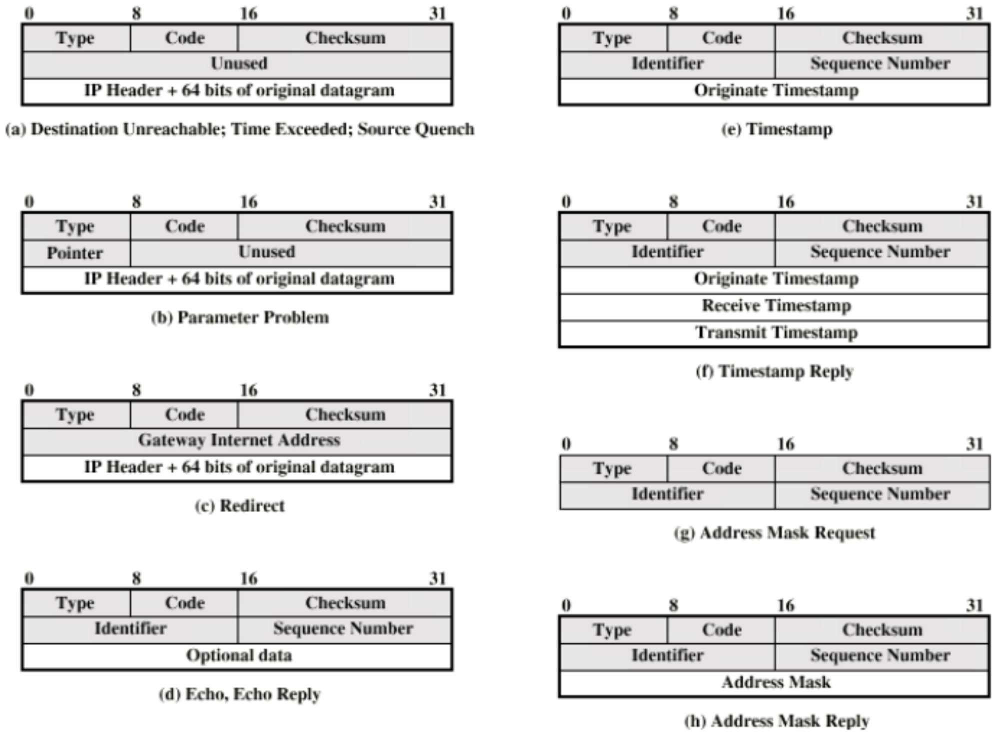
- How do two hosts communicate behind different NATs?
  - How does Skype work?



Internet Control Message Protocol

ICMP

- Layer 3 error or control messages
  - Sent by hosts/routers to other hosts/routers
- Carried in IP packets



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)
5		Redirect
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Echo request and reply

Do I (X) have IP connectivity to host Y?
What's the RTT delay between X and Y?

```
sun% ping gemini
PING gemini: 56 data bytes
64 bytes from gemini (140.252.1.11): icmp_seq=0. time=373. ms
64 bytes from gemini (140.252.1.11): icmp_seq=1. time=360. ms
64 bytes from gemini (140.252.1.11): icmp_seq=2. time=340. ms
64 bytes from gemini (140.252.1.11): icmp_seq=3. time-320. ms
64 bytes from gemini (140.252.1.11): icmp_seq=4. time=330. ms
64 bytes from gemini (140.252.1.11): icmp seq=5. time=310. ms
64 bytes from gemini (140.252.1.11): icmp seq=6. time==290. ms
64 bytes from gemini (140.252.1.11): icmp_seq-7. time=300. ms
64 bytes from gemini (140.252.1.11): icmp_seq=8. time=280. ms
64 bytes from gemini (140.252.1.11): icmp_seq=9. time=290. ms
64 bytes from gemini (140.252.1.11): icmp_seq=10. time=300. ms
64 bytes from gemini (140.252.1.11): icmp_seq=11. time=280. ms
--gemini PING Statistics--
12 packets transmitted, 12 packets received, 0% packet loss
round-trip (ms) min/avg/max = 280/314/373
```

Traceroute

How is the RTT distributed across the path between X and Y?
Which link/host is responsible for the large RTT in the path between X and Y?

(1) Source sends UDP packets to destination

- 1st with TTL=1, 2nd TTL=2, etc.
- Send to likely unused port number

(2) When datagram arrives to router n in the path

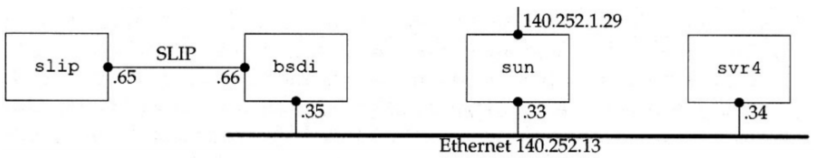
- TTL expired
- Router sends ICMP TTL expired message to source
- Message includes router name and IP address

(3) Source computes RTT upon "ICMP TTL expired" arrival

- 3 times
- Stops when
  - Contacts final destination
  - Router returns "destination unreachable"
  - Source stops sending UDP packets

```
svr4% traceroute slip
traceroute to slip (140.252.13.65), 30 hops max. 40 byte packets
1 bsdi (140.252.13.35) 20 ms 10 ms 10 ms
2 slip (140.252.13.65) 120 ms 120 ms 120 ms
```

```
slip% traceroute svr4
traceroute to svr4 (140.252.13.34), 30 hops max, 40 byte packets
1 bsdi (140.252.13.66) 110 ms 110 ms 110 ms
2 svr4 (140.252.13.34) 110 ms 120 ms 110 ms
```



ICMP Redirect

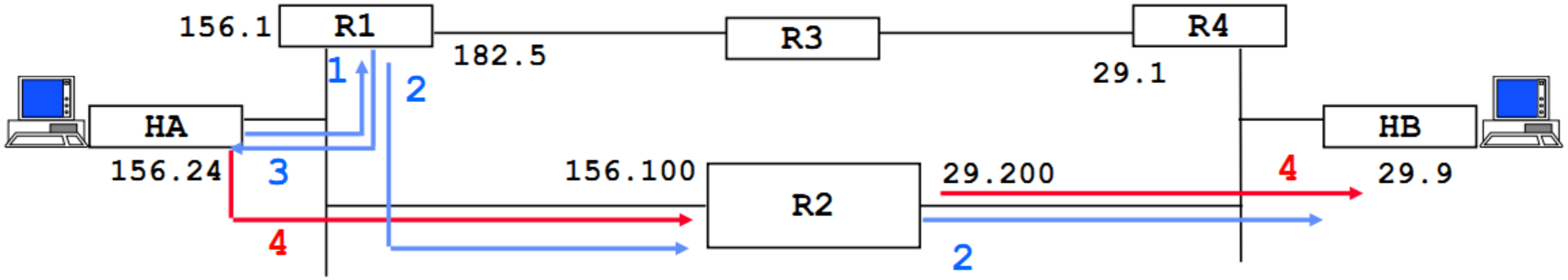
Is a host routing table misconfigured? Can it be optimized?
How can routers tell hosts to update their routing table?

- Hosts are typically not involved in routing
  - Routers can have extensive knowledge of routes
  - Hosts can learn from ICMP redirects
- Problem
  - R1 receives message from host A with destination B
  - R1 finds that next hop R2 is on-link with A
- Solution
  - R1 sends ICMP redirect to A saying next hop is R2
  - A updates routing table

ICMP Redirect Format

```
/                                                               /
|                  IP datagram header  (prot = ICMP)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type=5      |     code      |           checksum            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Router IP address that should be preferred       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        IP header plus 8 bytes of original datagram data       |
/                                                               /
```

ICMP Redirect example



	dest IP addr	srce IP addr	prot	data part
1:	193.154.29.9	193.154.156.24	udp	xxxxxxx
2:	193.154.29.9	193.154.156.24	udp	xxxxxxx
3:	193.154.156.24	193.154.156.1	icmp	type=redir code=host cksum 193.154.156.100 xxxxxxx (28 bytes of 1)
4:	193.154.29.9	193.154.156.24	udp	....

ICMP Redirect example

After 4

```
HA$ netstat -nr
Routing Table:
  Destination            Gateway              Flags  Interface
-------------------- -------------------- ----- ---------
127.0.0.1            127.0.0.1             UH      lo0
193.154.29.9         193.154.156.100       UGH     eth0
193.154.156.0        193.154.156.24        U       eth0
224.0.0.0            193.154.156.24        U       eth0
default              193.154.156.1         UG      eth0
```

Flags:
- **U -** *route Up*
- **G -** *route to a Gateway (next hop router)*
- **H -** *route to a Host*

IPv6

IPv4 Address Space

- 2^32 = 4 294 967 296
  - ~4.3 billion (*milhares de milhão*)
- Reuters – Oct 31 2011

A crowded world's population hits 7 billion

Recommend 339 recommendations. Sign Up to see what your friends recommend.

Tweet 40

Share 14

Share this

+1 0

Email

Print

Related News

Population boom heralds big global

By Pawan Kumar

Mon Oct 31, 2011 10:28am EDT

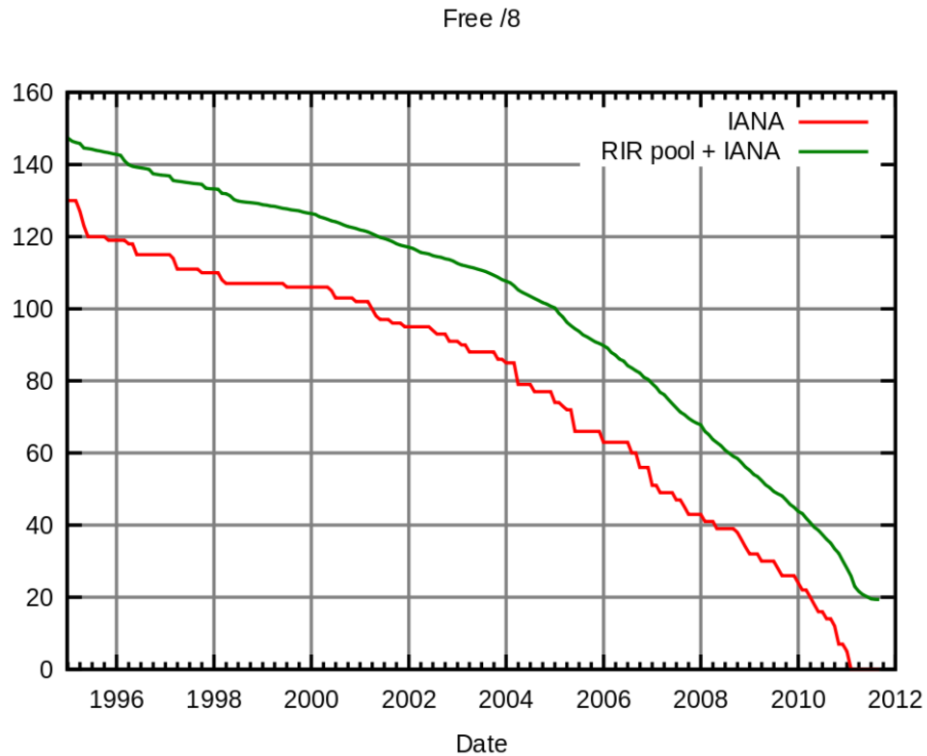
IPv4 and Address Space

- IPv4 – 1970's, RFC 1981
- 1985 – 1/16 of total address space used
- 1990 – 1/8 of total address space used

- 1992 – publicly recognized problem
- 1994 – IETF launches discussions on IPv6
- 1995 – 1/4 of total address space used
- 2000 – 1/2 of total address space used
- 2002 – 2/3 of total address space used

February 3, 2011

Press conference:
Free Pool of IPv4 Address Space Depleted



Mitigation efforts

- NAT
  - Rather than assigning a public IP address for each PC at home, assign single IP address
  - Reduce demand
- Reclamation of unused space
  - Historical assignments of large blocks of addresses (universities, corporations)
- ISP-level NAT
  - Don't provide public addresses to customers
  - Do NAT from IPv6
- IPv4 market
  - Buy/sell address space
- Permanent solution
  - IPv6

IPv6 - some features

- 128 bit addresses
  - 2^128 = 3.40E38
  - 1E29 addresses per person on Earth (in 2011)
- Native flow label
  - Better QoS support
- Peer authentication, data encryption
  - Intrinsic/native security
- Intrinsic auto-configuration

IPv6 Address Representation

- 8x16 bit, hexadecimal, separated by ':'
  - 47CD : 1234 : 3200 : 0000 : 0000 : 4325 : B792 : 0428
- Compressed format
  - FF01:0:0:0:0:0:0:43 => FF01::43
- Compatibility with IPv4
  - 0:0:0:0:0:0:13.1.68.3   or   ::13.1.68.3
- Loopback address ::1
- Same notation for network prefix
  - FEDC:BA98:7600::/40 => 40 bit network mask

Address allocation

Allocation	Prefix (binary)	Fraction of Address Space
Unassigned	0000 0000	1/256
Unassigned	0000 0001	1/256
Reserved for NSAP Allocation	0000 001	1/128
Unassigned	0000 01	1/64
Unassigned	0000 1	1/32
Unassigned	0001	1/16
Global Unicast	001	1/8
Unassigned	010	1/8
Unassigned	011	1/8
Unassigned	100	1/8
Unassigned	101	1/8
Unassigned	110	1/8
Unassigned	1110	1/16
Unassigned	1111 0	1/32
Unassigned	1111 10	1/64
Unassigned	1111 110	1/128
Unassigned	1111 1110 0	1/512
Link-Local Unicast Addresses	1111 1110 10	1/1024
Site-Local Unicast Addresses	1111 1110 11	1/1024
Multicast Addresses	1111 1111	1/256

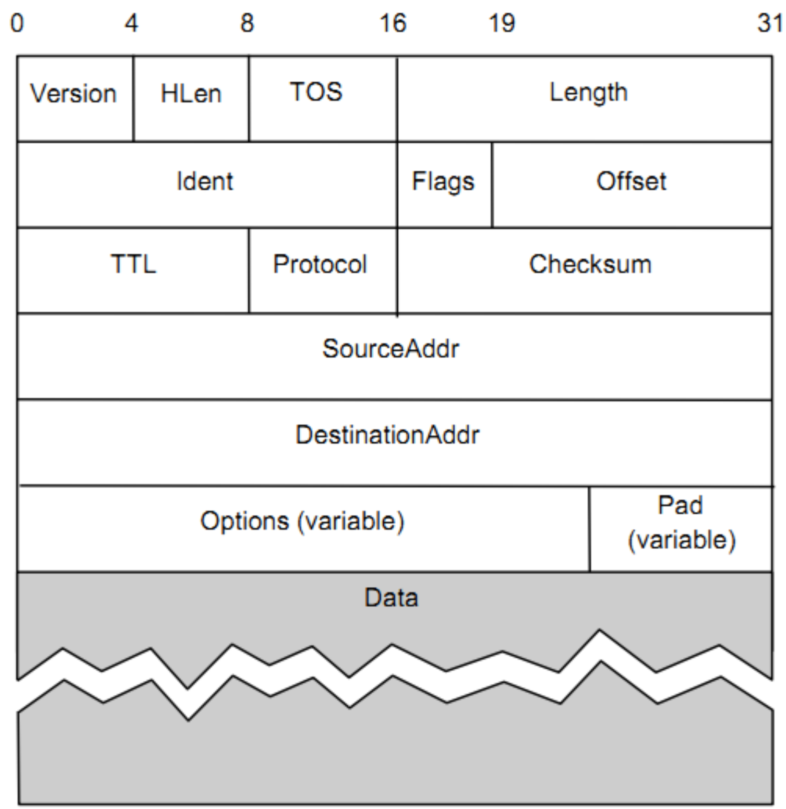
Address types

- Link-local
  - Use for communication in same LAN/link
  - Built from MAC address
  - Routers don't forward packets with link-local destination addresses
    - Why not?
- Global unicast
  - Global addresses
  - Network prefix + interface identifier
  - Structured prefixes
    - e.g. all LANs must be /64
    - Fewer entries in the routing tables
- Anycast
  - Group address: each packet is received by any but only one member of the group
  - Nearest replicated server, content delivery network
- Multicast
  - Group address: each packet is received by all the members of the group
  - Video multicast

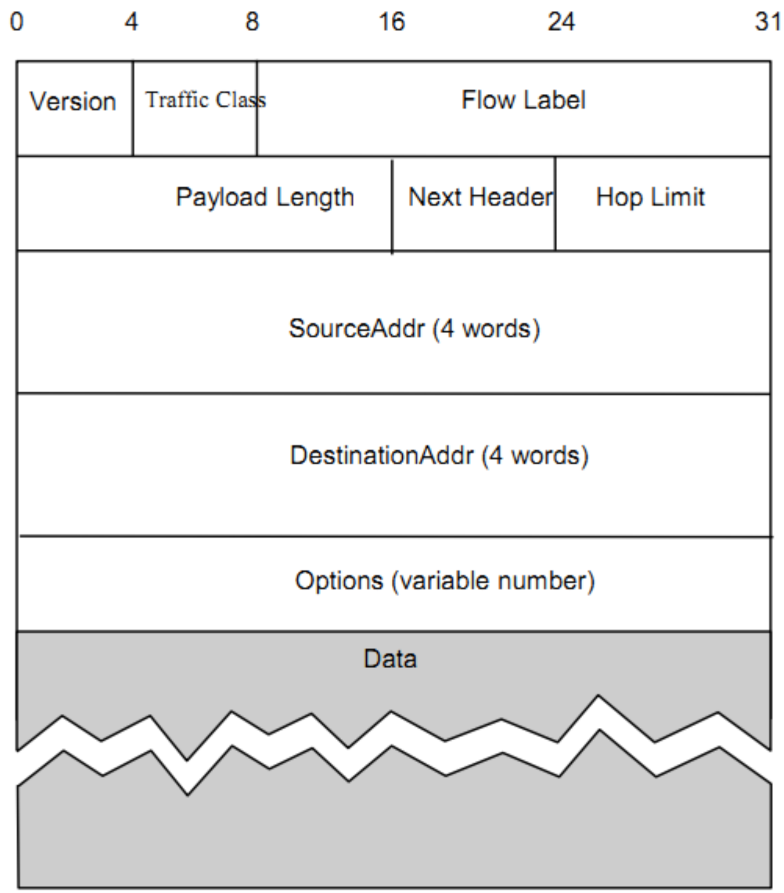
Address Formats

```
|          n bits         |  m bits  |       128-n-m bits         |   Global Unicast Address
+-------------------------+----------+----------------------------+   (2000::/3)
|001 global rout prefix   | subnet ID|       interface ID         |
+-------------------------+----------+----------------------------+

|   10     |
|  bits    |        54 bits          |         64 bits            |   Link-Local Unicast address
+----------+-------------------------+----------------------------+   (fe80::/10)
|1111111010|           0             |       interface ID         |
+----------+-------------------------+----------------------------+

|   10     |
|  bits    |        54 bits          |         64 bits            |   Site-Local Unicast address
+----------+-------------------------+----------------------------+   (fec0::/10) (not used)
|1111111011|      subnet ID          |       interface ID         |
+----------+-------------------------+----------------------------+

|                     n bits                     |   128-n bits   |   Anycast address
+------------------------------------------------+----------------+
|                 subnet prefix                  | 00000000000000 |
+------------------------------------------------+----------------+

|   8    |  4 |  4 |                 112 bits                     |
+------ -+----+----+----------------------------------------------+   Multicast address
|11111111|flgs|scop|                 group ID                     |   Scope - link, site, global, ...
+--------+----+----+----------------------------------------------+          (ff::/8)
```

IPv6 Header (vs. IPv4)



IPv4

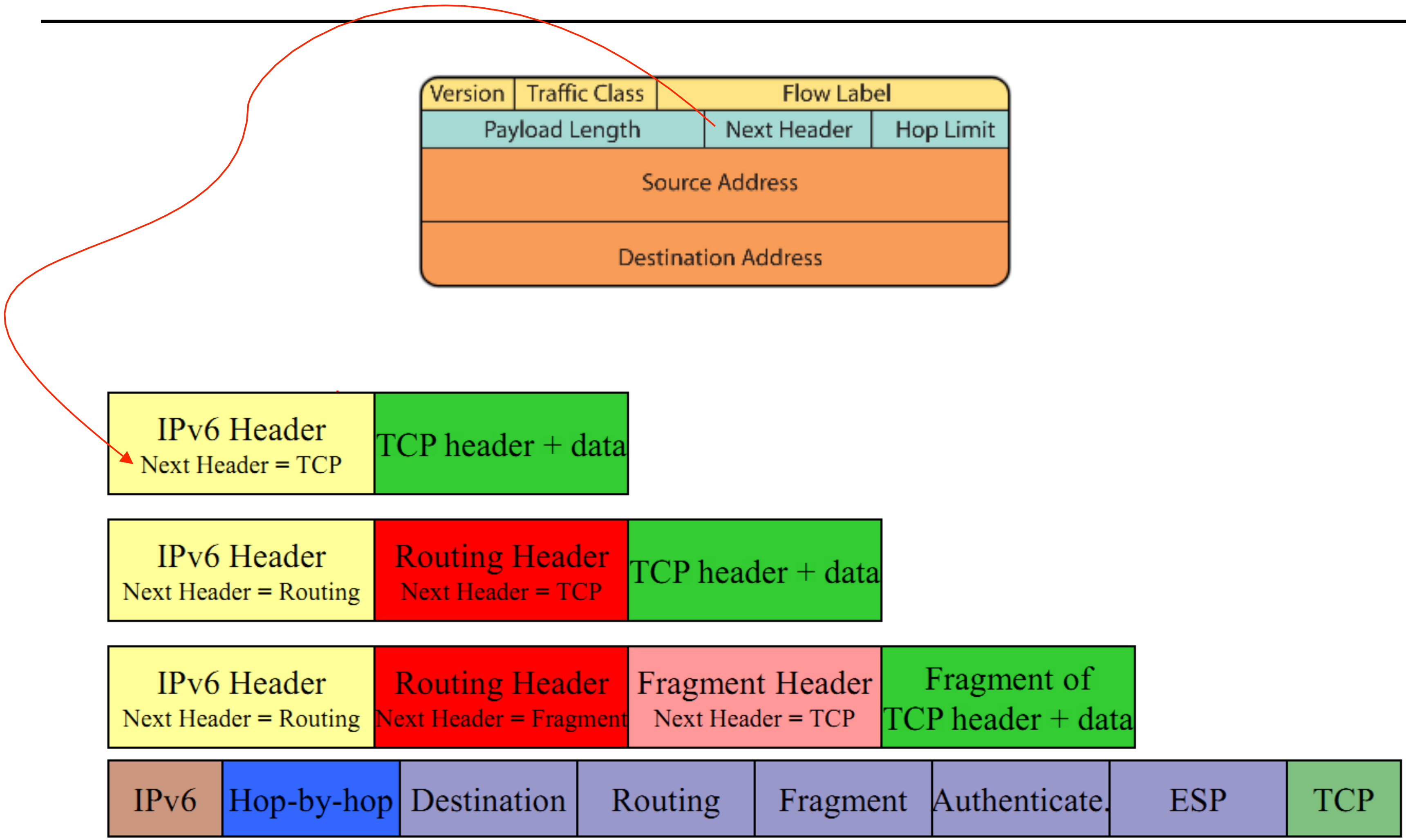


IPv6

IPv6 Header

- Flow label
  - Identifies to which flow this packet belongs
  - QoS, resource reservation
  - Packets with the same flow receive similar service
- Payload length
  - Header not included
- Hop limit (=v4 TTL)
- Next header
  - Allows for extensions
  - Points to location of the next header in the packet
- Options
  - I.e. extensions (next slide)

Extension headers



Extension headers

- Routing
  - List of nodes to be visited by the packet
- Fragmentation
  - MTU
- Authentication
  - Authentication signature of the packet header
- ESP
  - Data encryption options
- Destination
  - Information for the destination node
- Hop-by-hop
  - Additional information to be inspected by all routers in the path

Lab Network Example

```
quadro
                                                       porta

banc_3                                                 banc_6
pc3---[HUB]---pc2----+            +----pc2---[HUB]---pc3
2000:0:0:3::/64      |            |      2000:0:0:6::/64
                     |            |
banc_2               |            |                banc_5
pc3---[HUB]---pc2--[HUB]-+    +-[HUB]--pc2---[HUB]---pc3
2000:0:0:2::/64      |   |    |   |      2000:0:0:5::/64
                     |   |    |   |
banc_1               |   |    |   |                banc_4
pc3---[HUB]---pc2----+   |    |   +----pc2---[HUB]---pc3
2000:0:0:1::/64          |    |          2000:0:0:4::/64
                         |    |
          2000:0:0:e::/64|    |2000:0:0:d::/64
                         |    |
                       [routerv6]
```

2000:0:0:1::1

2000:0:0:1::aa

2000:0:0:e::1

Configuration example, Linux

```
tux13:~# /sbin/ifconfig eth0 inet6 add 2000:0:0:1::1/64
tux13:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:C0:DF:08:D5:99
          inet addr:172.16.1.13  Bcast:172.16.1.255  Mask:255.255.255.0
          inet6 addr: 2000:0:0:1::1/64 Scope:Global
          inet6 addr: fe80::2c0:dfff:fe08:d599/10 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:81403 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2429 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:4981344 (4.7 MiB)  TX bytes:260692 (254.5 KiB)
          Interrupt:5
tux13:~# /sbin/route -A inet6 add 2000::/3  gw 2000:0:0:1::aa
tux13:~# route -A inet6
Kernel IPv6 routing table
Destination                    NextHop        Flags Metric Ref Use Iface
::1/128                        ::             U     0      0   0    lo
2000:0:0:1::1/128              ::             U     0      0   0    lo
2000:0:0:1::/64                ::             UA    256    0   0    eth0
2000::/3                       2000:0:0:1::aa UG    1      0   0    eth0
fe80::2c0:dfff:fe08:d599/128   ::             U     0      0   0    lo
fe80::/10                      ::             UA    256    0   0    eth0
ff00::/8                       ::             UA    256    0   0    eth0
::/0                           ::             UDA   256    0   0    eth0
```

IPv6 link-local from IEEE EUI-64

```
Method to create a IEEE EUI-64 identifier from an IEEE 48bit MAC identifier.
This is to insert two octets, with hexadecimal values of 0xFF and 0xFE,
in the middle of the 48 bit MAC (between the company_id and vendor supplied id).
For example, the 48 bit IEEE MAC with global scope:

 |0              1|1              3|3              4|
 |0              5|6              1|2              7|
 +----------------+----------------+----------------+
 |cccccc0gcccccccc|ccccccccmmmmmmmm|mmmmmmmmmmmmmmmm|
 +----------------+----------------+----------------+   00:C0:DF:08:D5:99

where "c" are the bits of the assigned company_id, "0" is the value of the
universal/local bit to indicate global scope, "g" is individual/group bit,
and "m" are the bits of the manufacturer-selected extension identifier.
The interface identifier would be of the form:

 |0              1|1              3|3              4|4              6|
 |0              5|6              1|2              7|8              3|
 +----------------+----------------+----------------+----------------+
 |cccccc1gcccccccc|cccccccc11111111|11111110mmmmmmmm|mmmmmmmmmmmmmmmm|
 +----------------+----------------+----------------+----------------+

                                                   fe80::2c0:dfff:fe08:d599
```

Neighbor Discovery Protocol

- IPv6 node uses ND to:
  - Find other nodes in same link/LAN
  - Find the MAC address of a node
    - Replaces ARP
  - Find routers in its network
  - Maintain information about neighboring nodes
- Similar functions in IPv4
  - ARP
  - ICMP Route discovery
    - ICMP type 10, finds routers with ICMP requests
  - ICMP Redirect

ND Messages

- ICMP over IP
  - Use link-local addresses

- Neighbor solicitation
  - Sent by host to obtain MAC address of neighbor or verify its presence
- Neighbor advertisement
  - To reply to the solicitation message
- Router advertisement
  - Information about network prefix
  - Periodic or under request
  - Sent by router on link-local multicast
- Router solicitation
  - Request by host for a router advertisement message
- Redirect
  - Router informs host of best route to destination

HOMEWORK

- Review slides

- Read "The Network Layer" from:
  - Kurose – Chap. 4
  - Or Tanenbaum – Chap. 5

- Do your Moodle homework