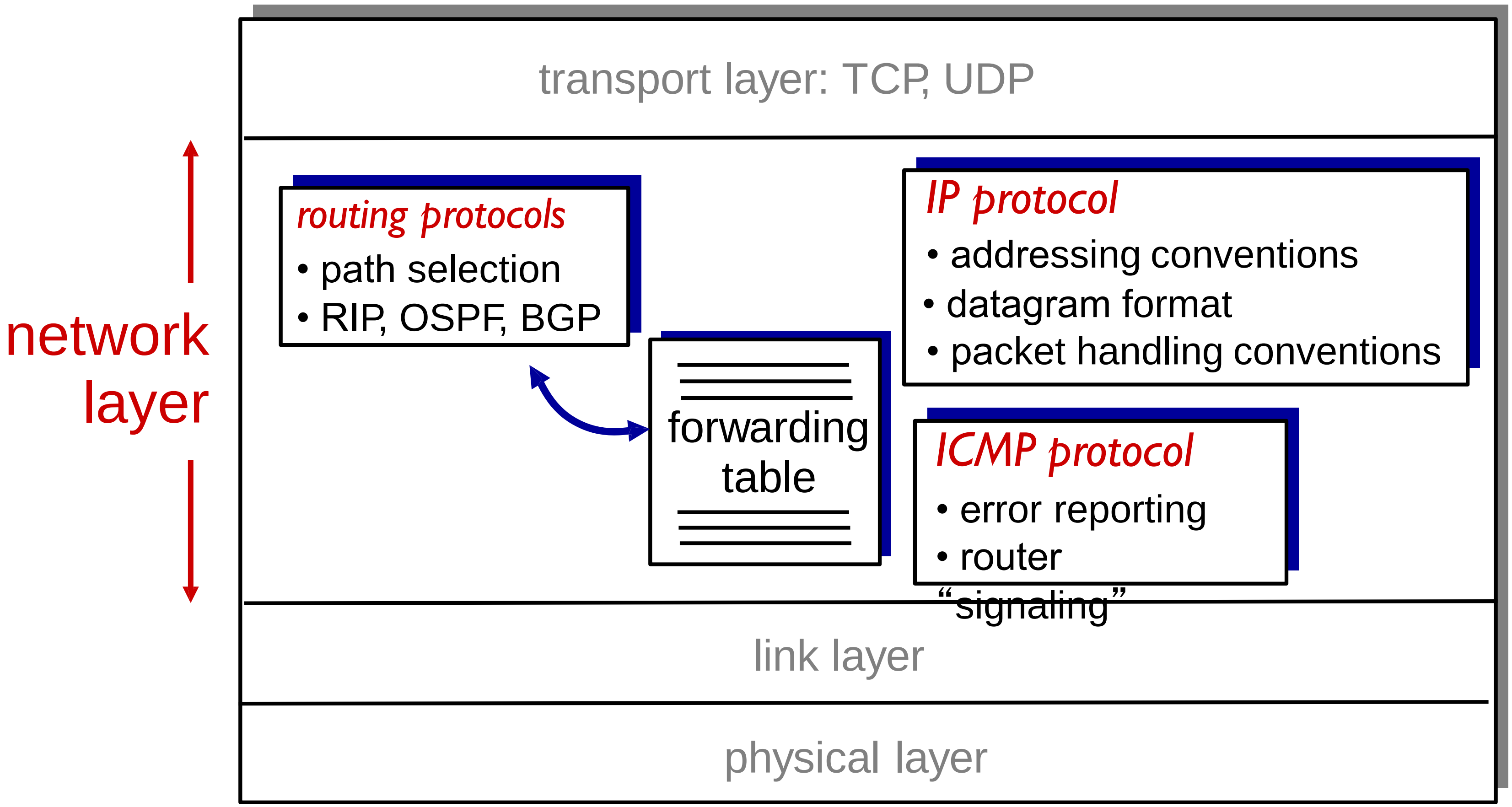
Packet-switching networks
Outline

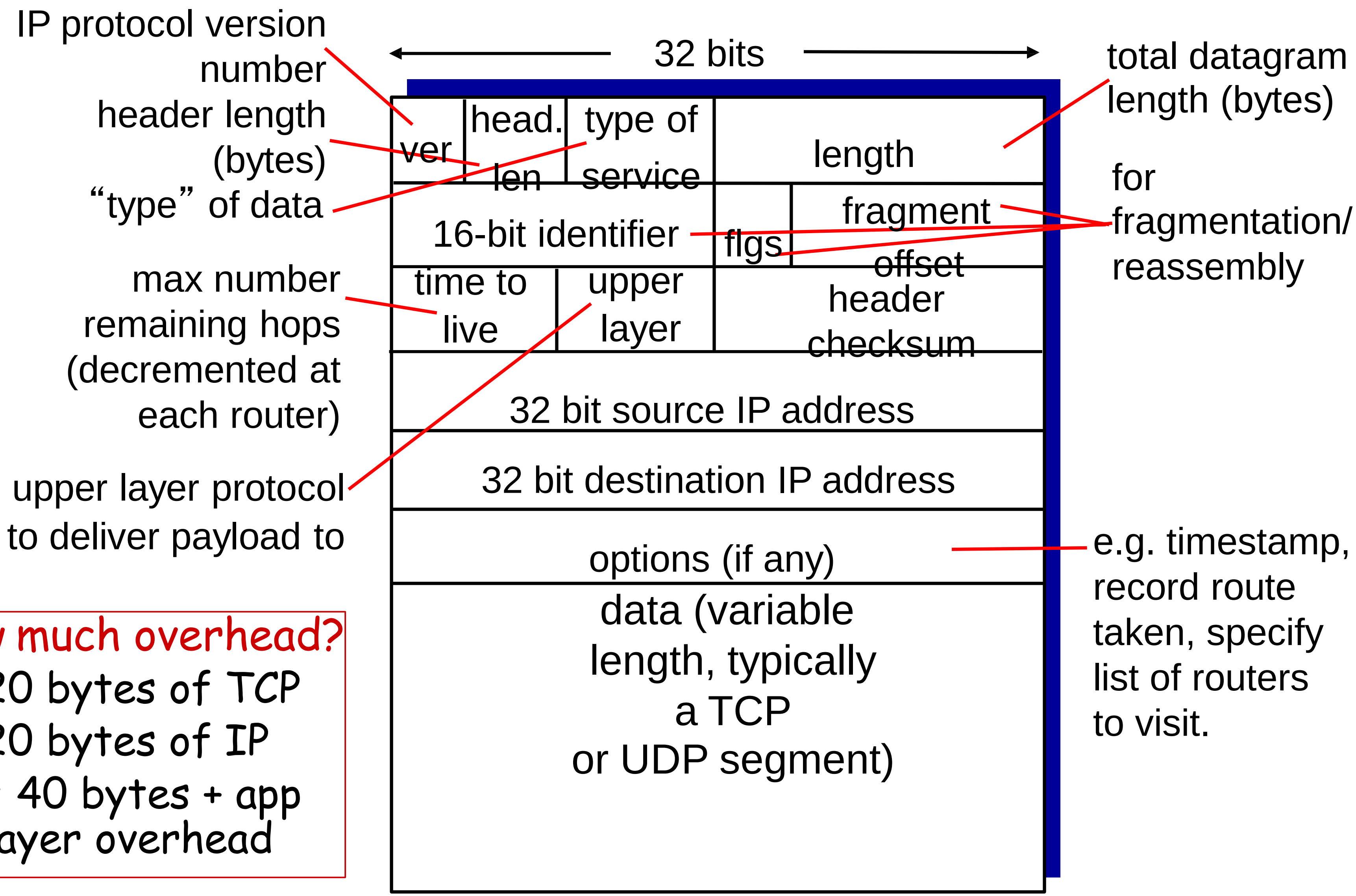
- Context/overview
- Basic approaches to operating a packet-network: datagrams and virtual circuits
- Network layer functions: Routing and forwarding
- Overview of Network layer: data plane and control plane
- Network layer: The Data Plane
  - What's inside a router
  - The Internet Protocol (IPv4, DHCP, NAT, IPv6)
  - Generalized Forward and SDN
- Network layer: The Control Plane
  - Overview of routing in packet networks
  - The SDN control plane
  - ICMP: The Internet Control Message Protocol

The Internet network layer

host, router network layer functions:



IP datagram format (IPv4)

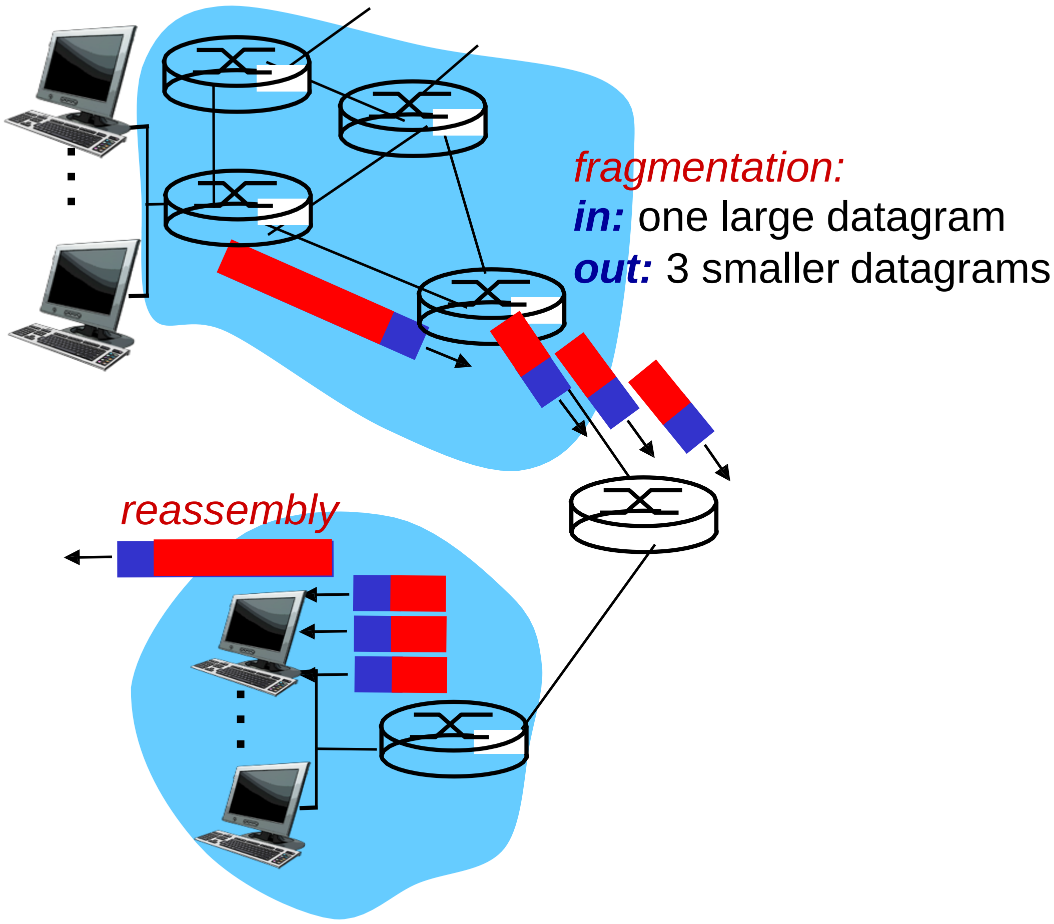


how much overhead?

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

IP fragmentation, reassembly

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



IP fragmentation, reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes

	length	ID	fragflag	offset	
	=4000	=x	=0	=0	

one large datagram becomes several smaller datagrams

1480 bytes in data field

offset = 1480/8

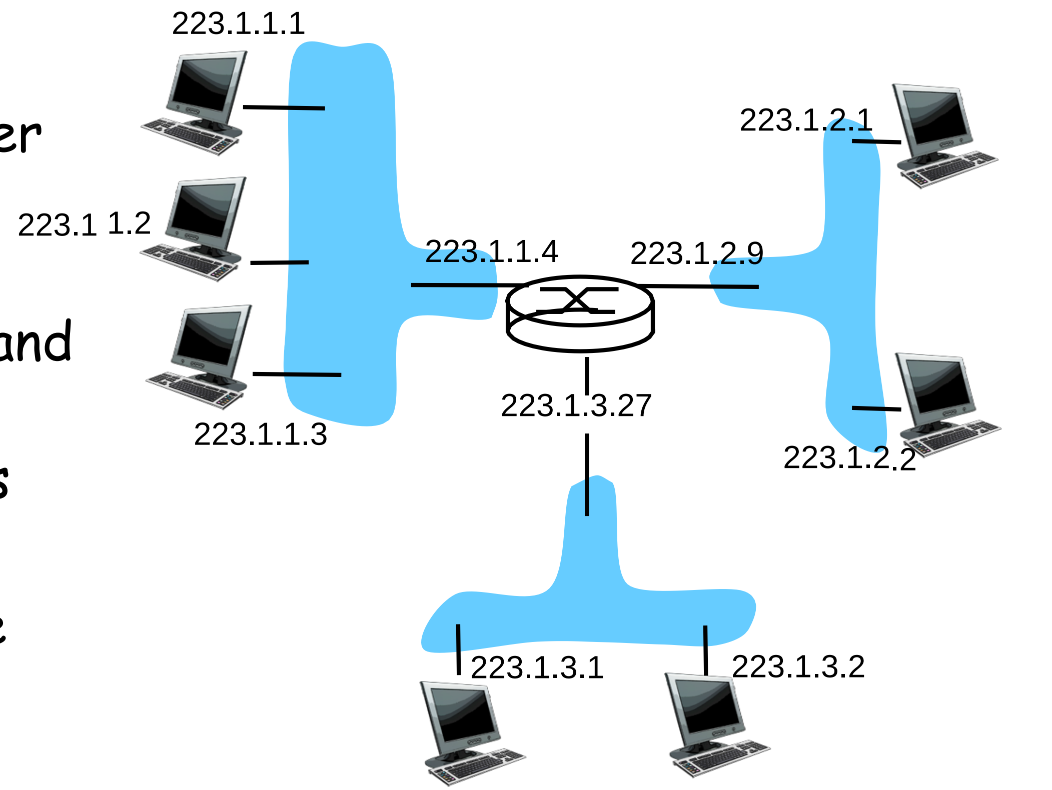
	length	ID	fragflag	offset	
	=1500	=x	=1	=0	

	length	ID	fragflag	offset	
	=1500	=x	=1	=185	

	length	ID	fragflag	offset	
	=1040	=x	=0	=370	

IP addressing: introduction

- IP address: 32-bit identifier for host, router interface
- interface: connection between host/router and physical link
  - router typically has multiple interfaces
  - host typically has one interface (e.g., wired Ethernet)
- IP addresses associated with each interface



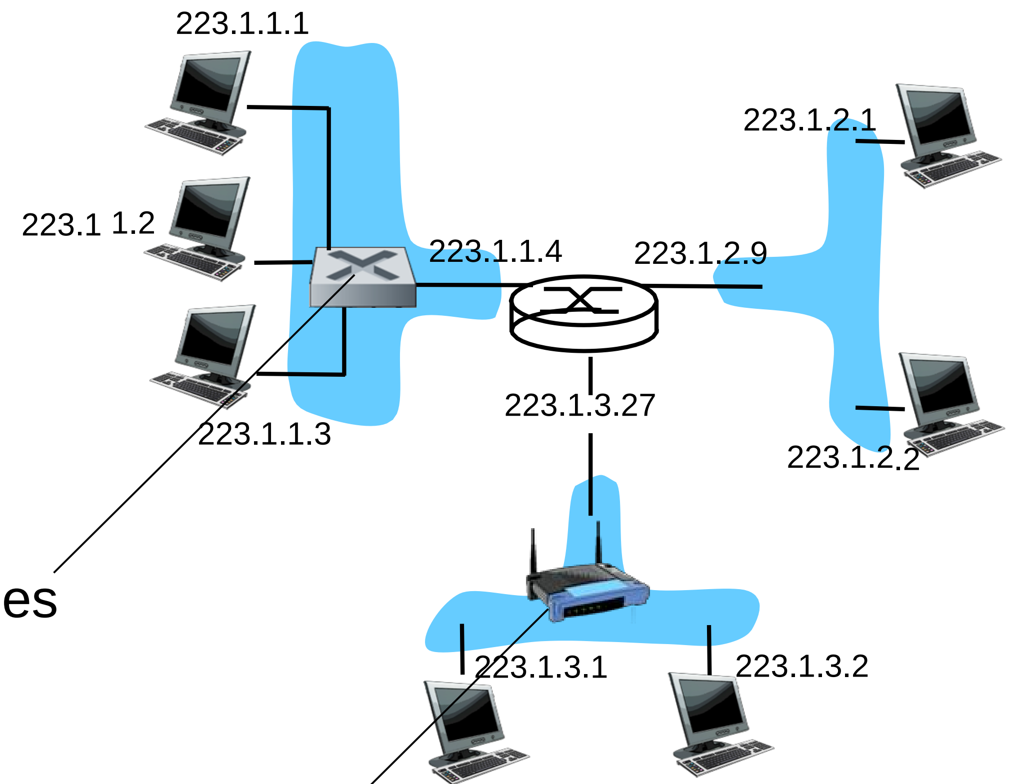
$$223.1.1.1 = \underbrace{11011111}_{223}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}\ \underbrace{00000001}_{1}$$

IP addressing: introduction

Q: how are interfaces actually connected?

A: more, later.

A: wired Ethernet interfaces connected by Ethernet switches

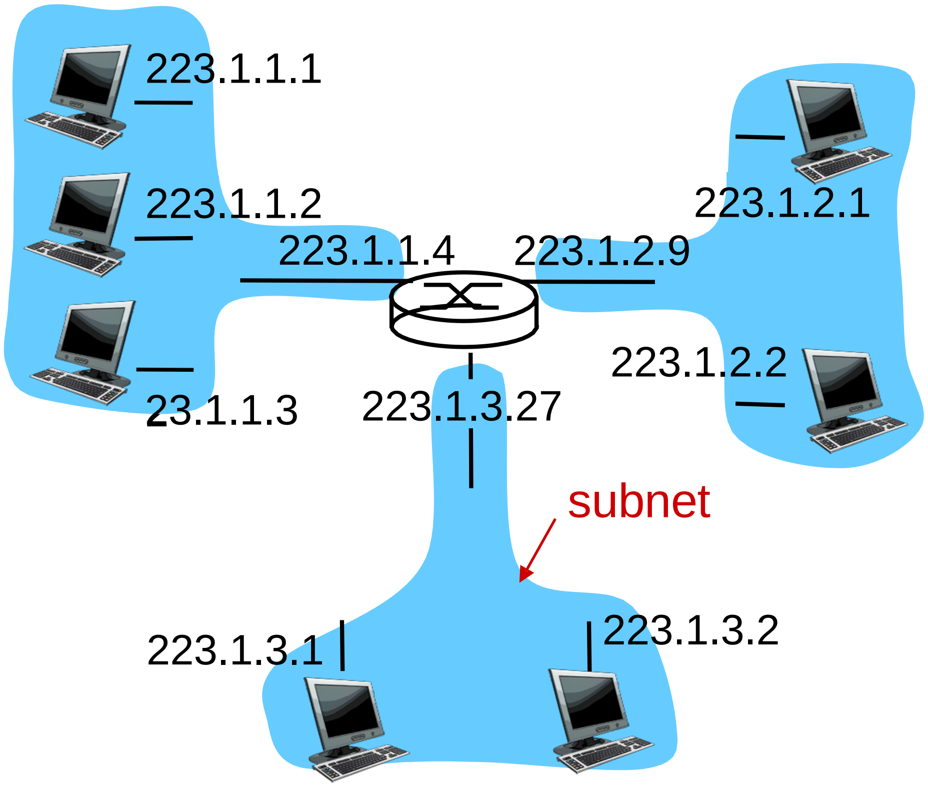


For now: don't need to worry about how one interface is connected to another (with no intervening router)

A: wireless WiFi interfaces connected by WiFi base station

Subnets

- IP address:
  - subnet part - high order bits
  - host part - low order bits
- what's a subnet ?
  - device interfaces with same subnet part of IP address
  - can physically reach each other without intervening router

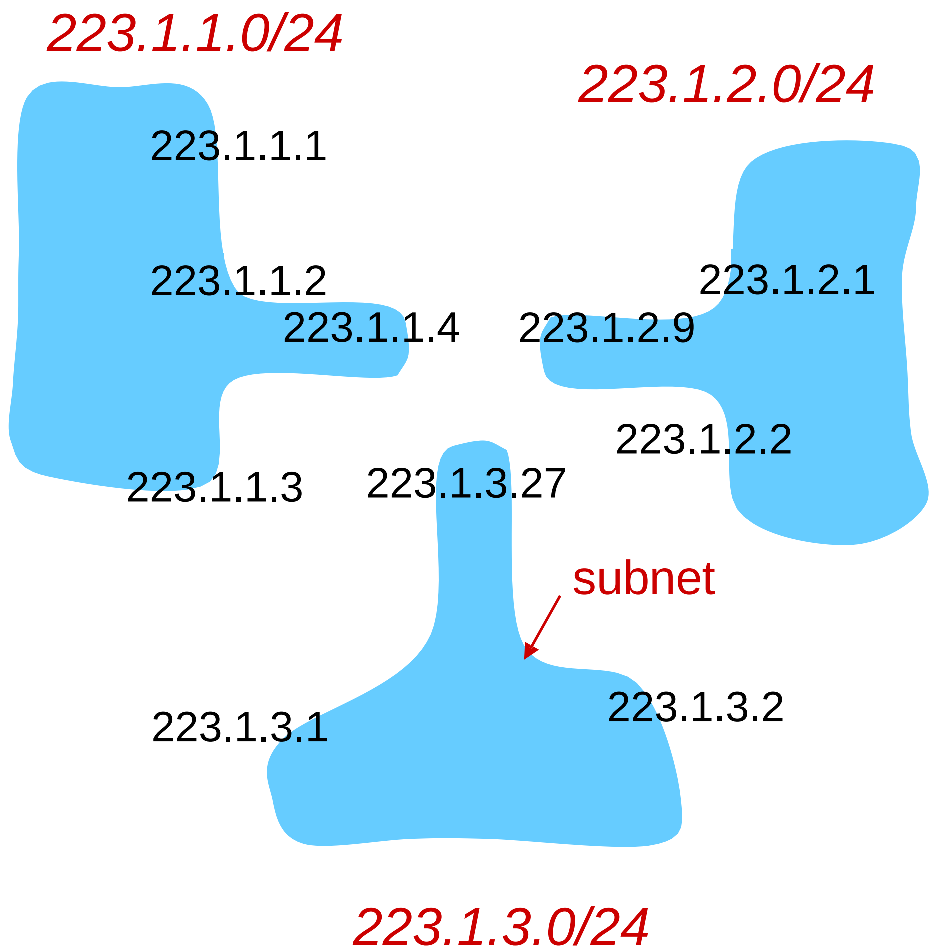


network consisting of 3 subnets

Subnets

recipe

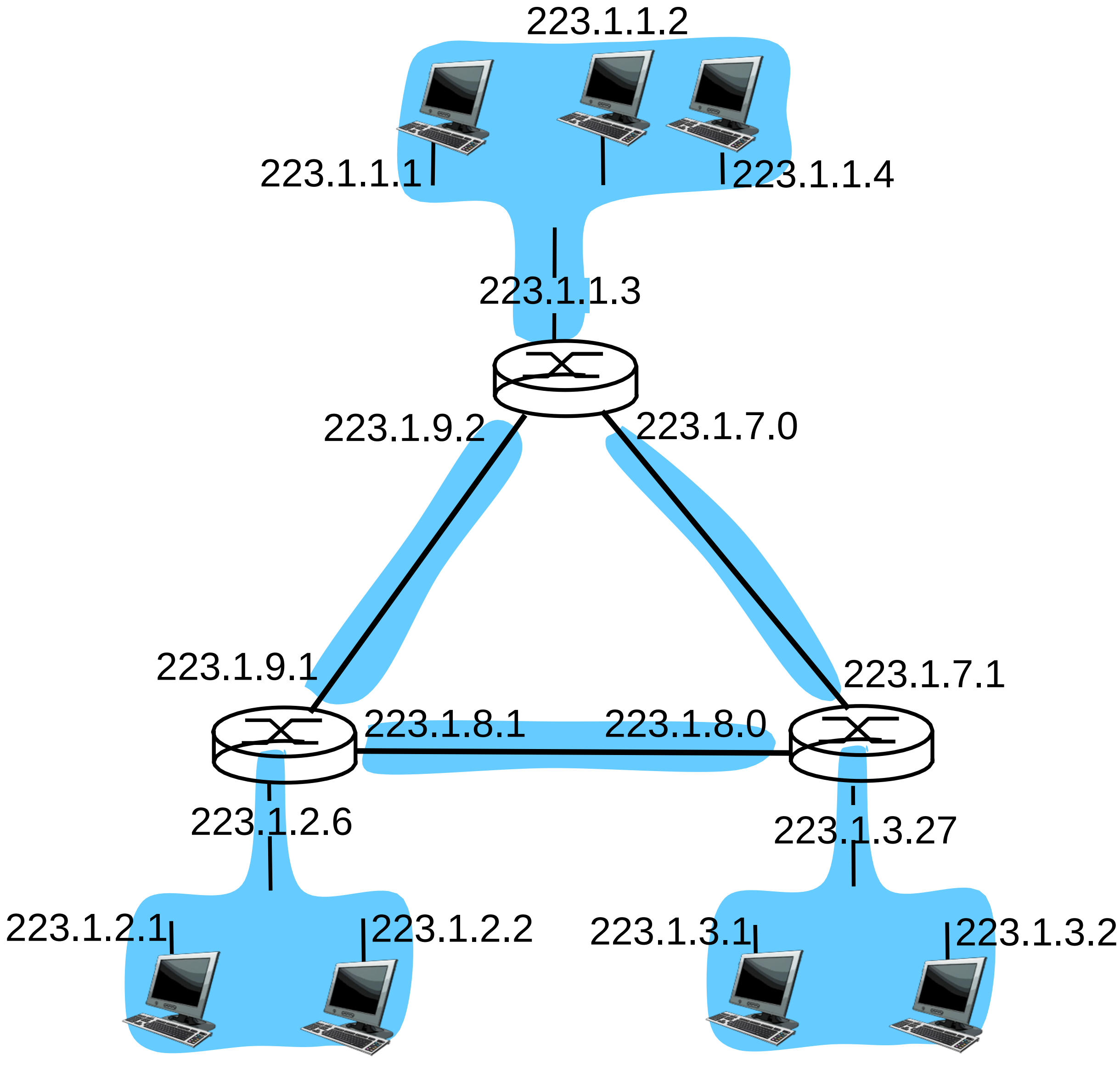
- to determine the subnets, detach each interface from its host or router, creating islands of isolated networks
- each isolated network is called a subnet



subnet mask: /24

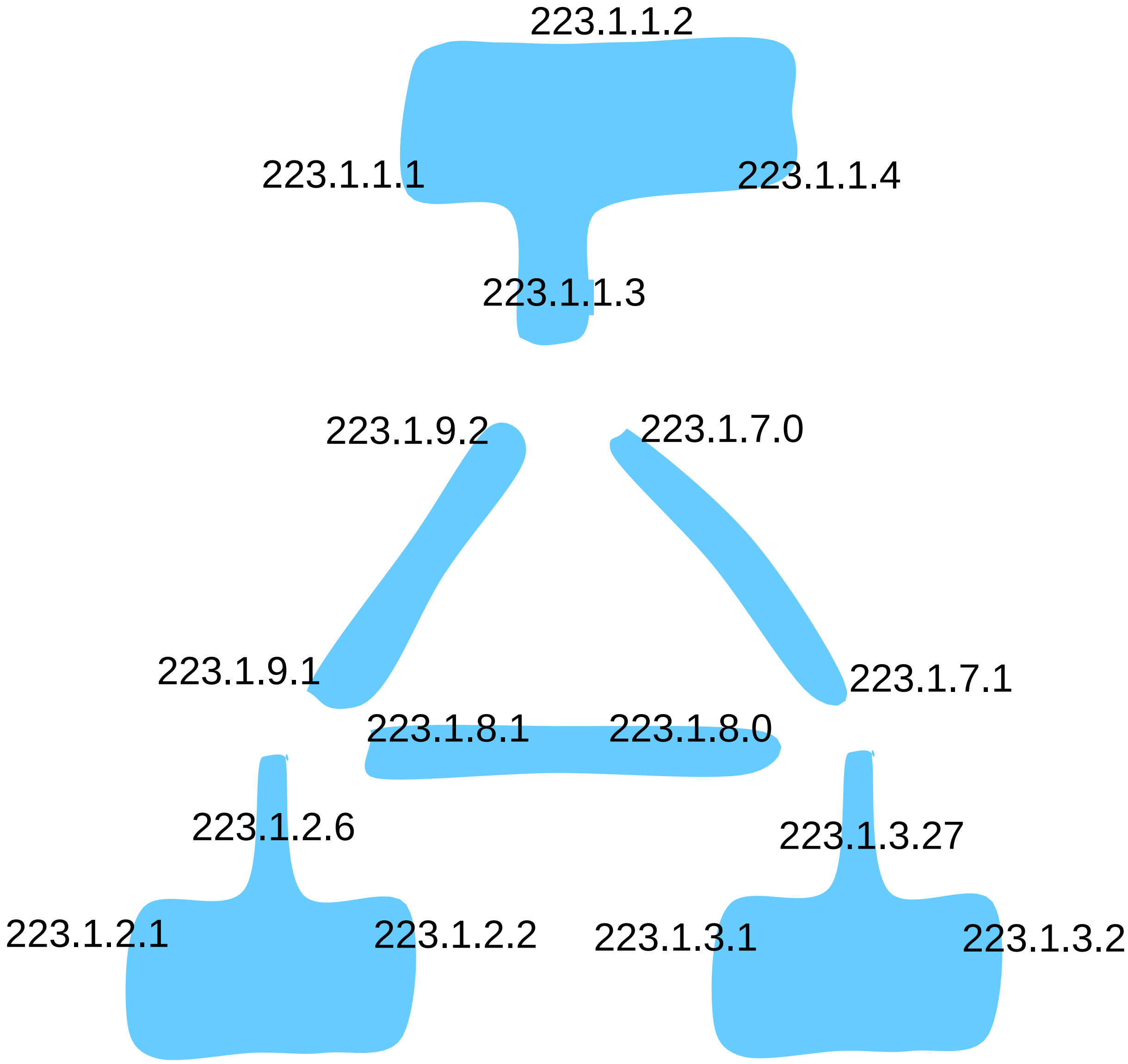
Subnets

how many?



Subnets

6 subnets



IP addressing: CIDR

CIDR: Classless InterDomain Routing

- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address (= prefix or network-prefix)

subnet part

host part

11001000 00010111 00010000 00000000

200.23.16.0/23

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 a server
  - "plug-and-play"

DHCP: Dynamic Host Configuration Protocol

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

Can renew its lease on address in use

Allows reuse of addresses

Support for mobile users who want to join network

- DHCP uses UDP

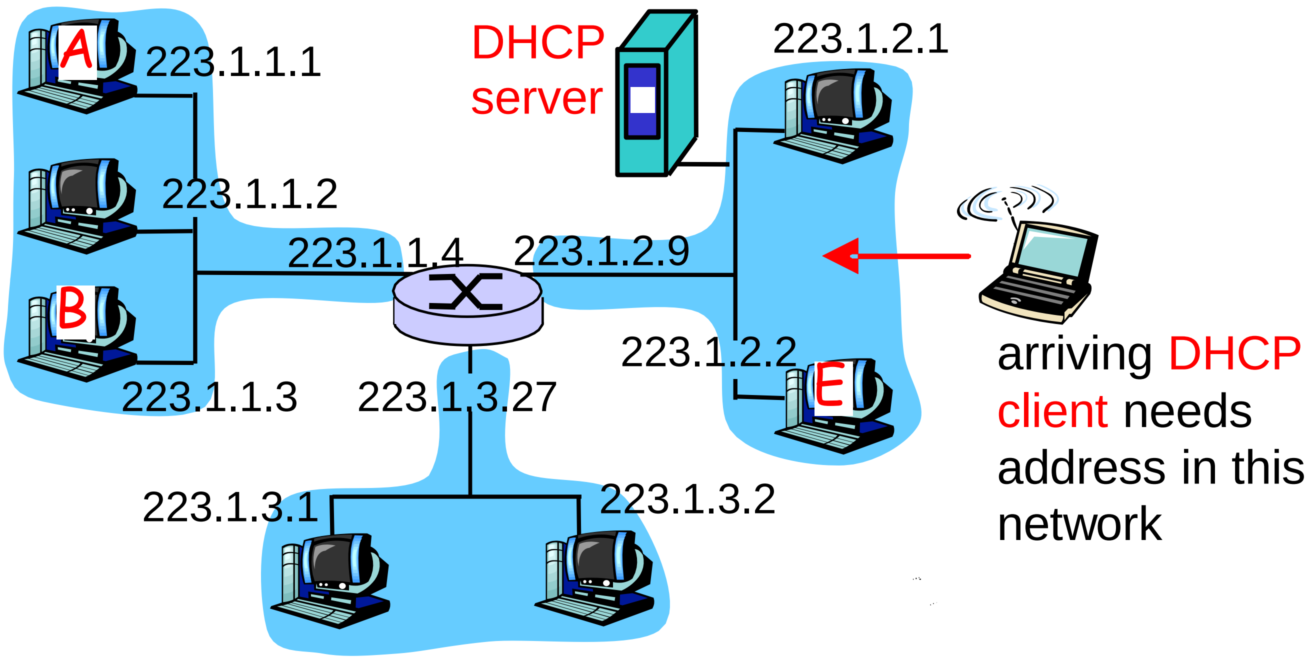
DHCP overview:

- host broadcasts "DHCP discover" msg
- DHCP server responds with "DHCP offer" msg

  (might get many offers from many servers)
- host requests IP address: "DHCP request" msg

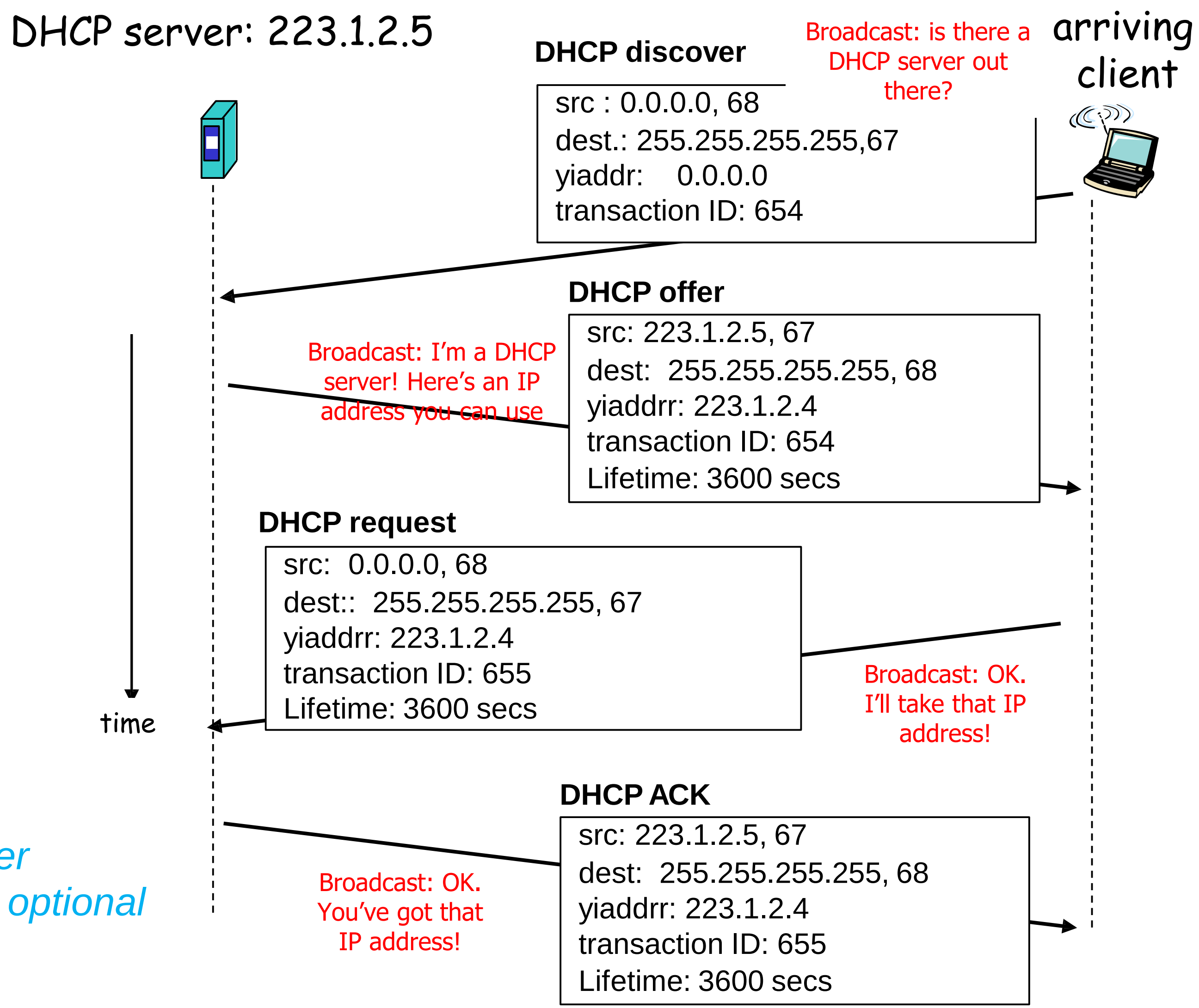
  (to a selected server)
- DHCP server sends address: "DHCP ack" msg

DHCP client-server scenario



A router may act as a relay agent

DHCP client-server scenario



Discover & offer messages are optional

DHCP is also seen in residential Internet access networks

DHCP: more than IP address

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

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

IP addresses: how to get one?

Q: How does network get subnet part of IP addr?

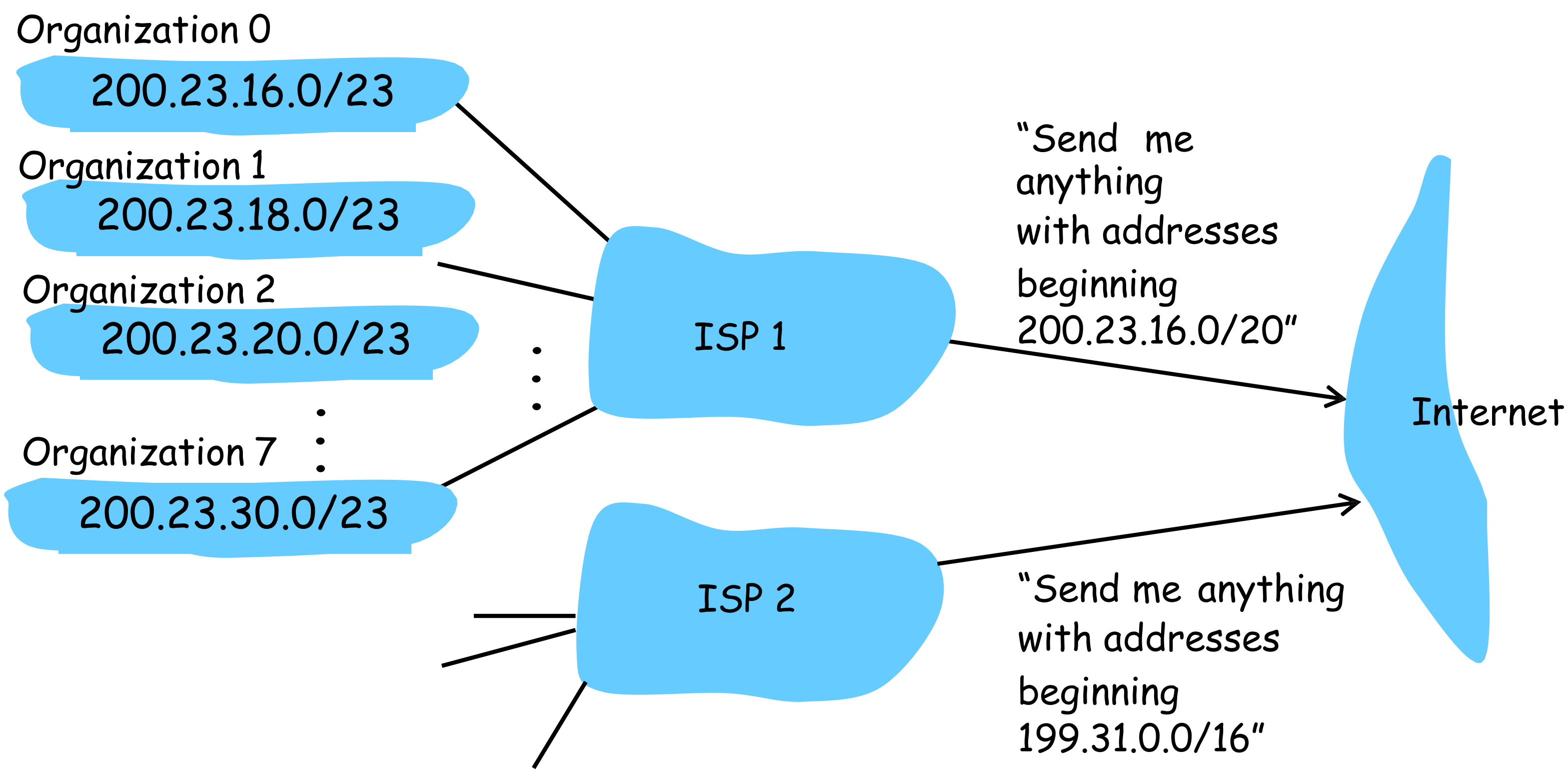
A: gets allocated portion of its provider ISP's address space

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

* Hierarchical Addressing

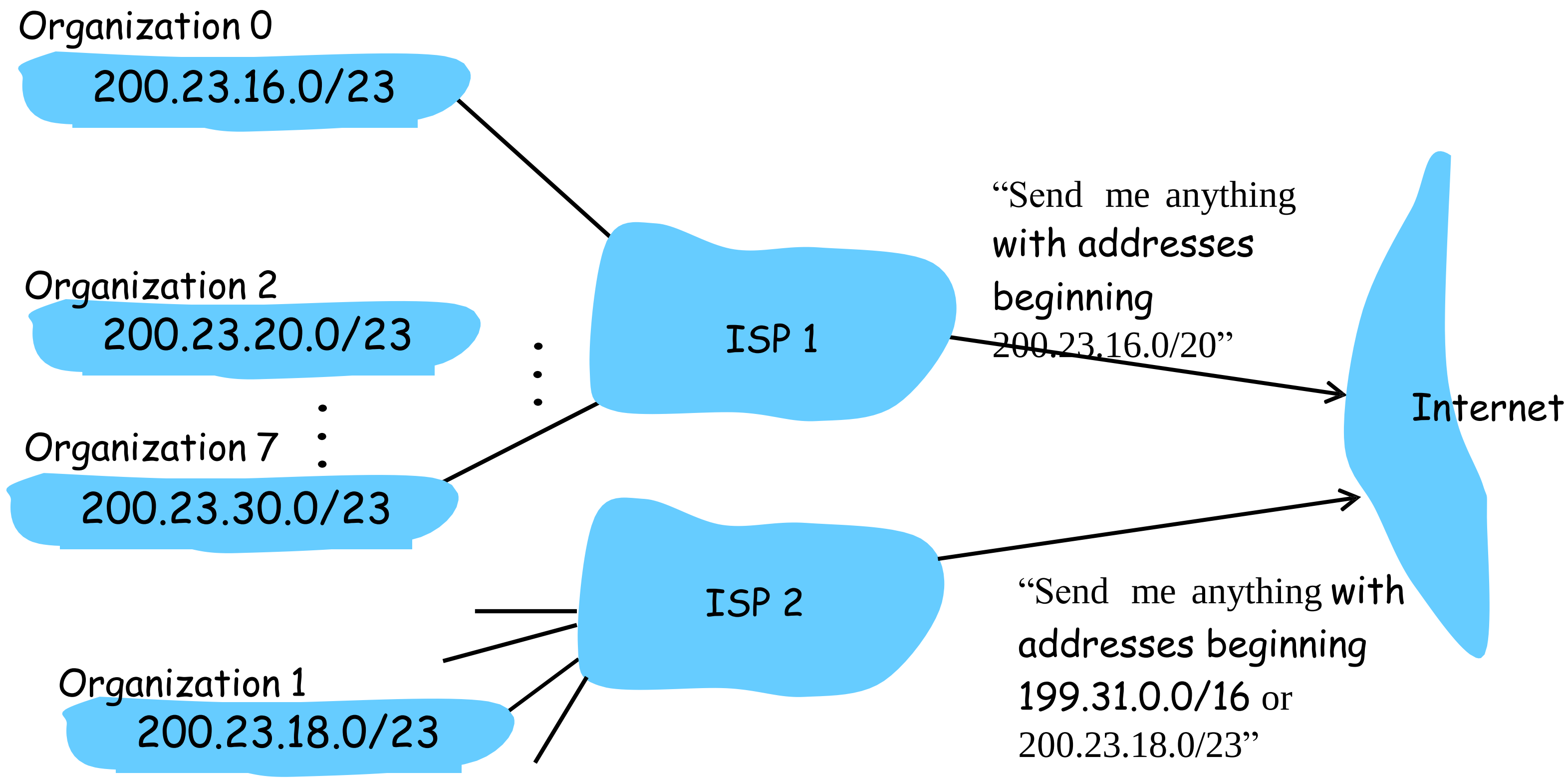
Hierarchical addressing: route aggregation

Hierarchical addressing allows efficient advertisement of routing information:



Hierarchical addressing: more specific routes

ISP-2 has a more specific route to Organization 1



IP addressing: the last word...

How does an ISP get block of addresses?

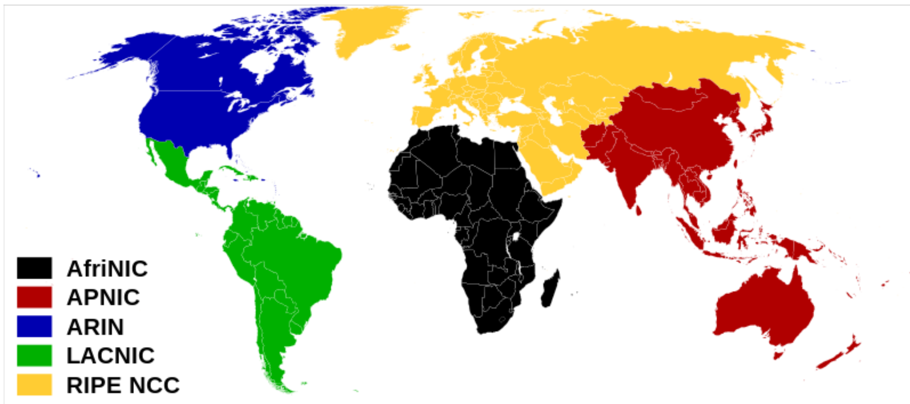
Is there a global authority that has ultimate responsibility for managing the IP address?

ICANN: Internet Corporation for Assigned Names and Numbers /IANA:Internet Assigned Numbers Authority

- allocates addresses
- manages DNS (manage DNS root servers)
- assigns domain names, resolves disputes

ICANN allocates addresses to Regional Internet Registries (RIR) such as: ARIN, RIPE, APNIC, LACNIC

RIRs delegate address to ISPs, National Internet Registries, and customers in their regions.



End-user organization can be assigned IP address space from one of the above

Special IP Addresses (IPv4)

- Reserved or (by convention) special addresses:

Loopback interfaces

- all addresses 127.0.0.1-127.255.255.255 (127/8) are reserved for loopback interfaces
- Most systems use 127.0.0.1 as loopback address
- loopback interface is associated with name "localhost"
- used to test network applications
- During loopback testing no packets ever leave a computer. The IP software forwards packets from one application to another

IP address of a network

- Host number is set to all zeros, e.g., 128.143.**0.0** (A network address should never appear as the destination address in a packet)

Broadcast address

- Host number is all ones, e.g., 128.143.**255.255** (broadcast on a specified network)
- 255.255.255.255 (broadcast on the local net) (see DHCP)
- Broadcast goes to all hosts on the network (subnet)
- Often ignored due to security concerns

Multicast address

- 224.0.0.0/4

'this computer' address

- 0.0.0.0 (see DHCP)

- Unicast: one-to-one
- Broadcast: one-to-all in the network
- Multicast: one-to-many (not all)

- Addresses for private networks or private addresses
  - The following address ranges are reserved for private networks (or experimental use). (see RFC 1918)

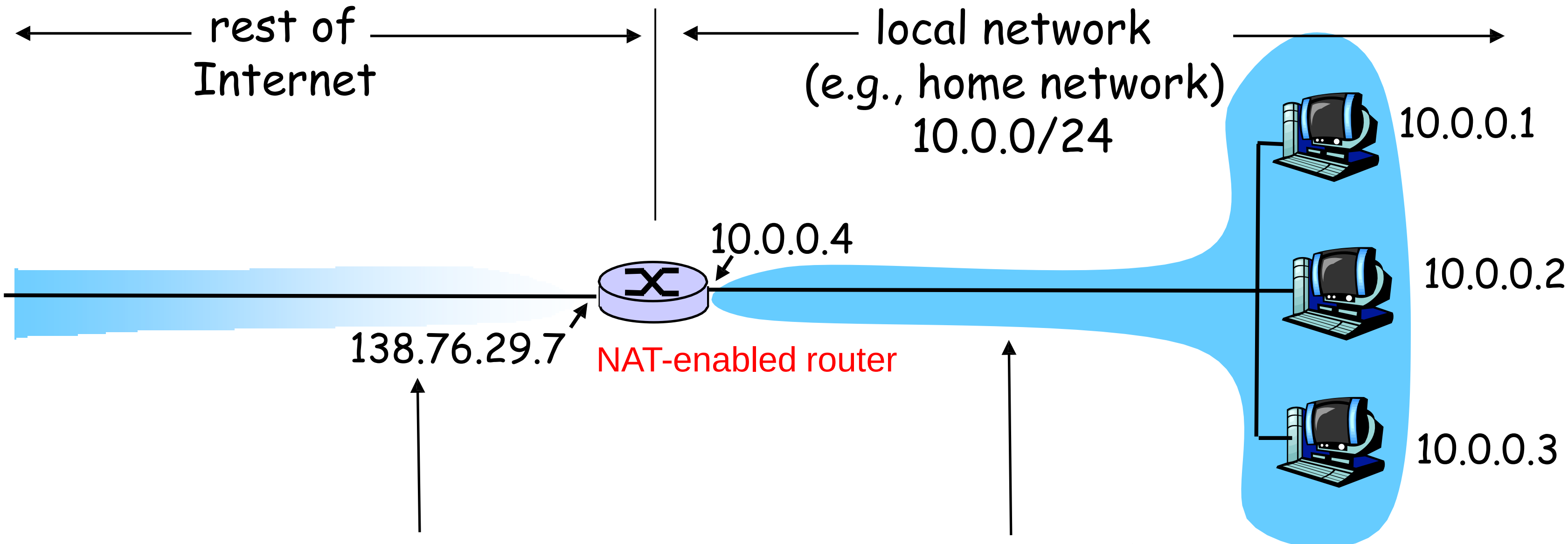
10.0.0.0	-	10.255.255.255	(10.0.0.0/8)
172.16.0.0	-	172.31.255.255	(172.16.0.0/12)
192.168.0.0	-	192.168.255.255	(192.168.0.0/16)

  - Private addresses only have meaning within a given network (no globally unique)
  - These addresses are characterized as private because they are not globally delegated (they are not allocated to any specific organization)
  - Within a private network, transmission can be done using this private addresses.
  - IP packets addressed with 'private addresses' cannot be transmitted through the public Internet (Packets should get dropped if they contain this destination address)
  - How is addressing handled when packets are sent to or received from the global Internet, where addresses are necessarily unique?

- Convention (but not a reserved address)

  Default gateway has host number set to '1', e.g., 192.0.1.**1**

NAT: Network Address Translation



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

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

(a.k.a., network address and port translation (NAPT), *port address translation* (PAT), *IP masquerading, NAT overload)*

NAT: Network Address Translation

- Motivation: local network uses just one IP address as far as outside world is concerned:
  - range of addresses not needed from ISP: just one IP address for all devices
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).

NAT: Network Address Translation

Implementation: NAT router must:

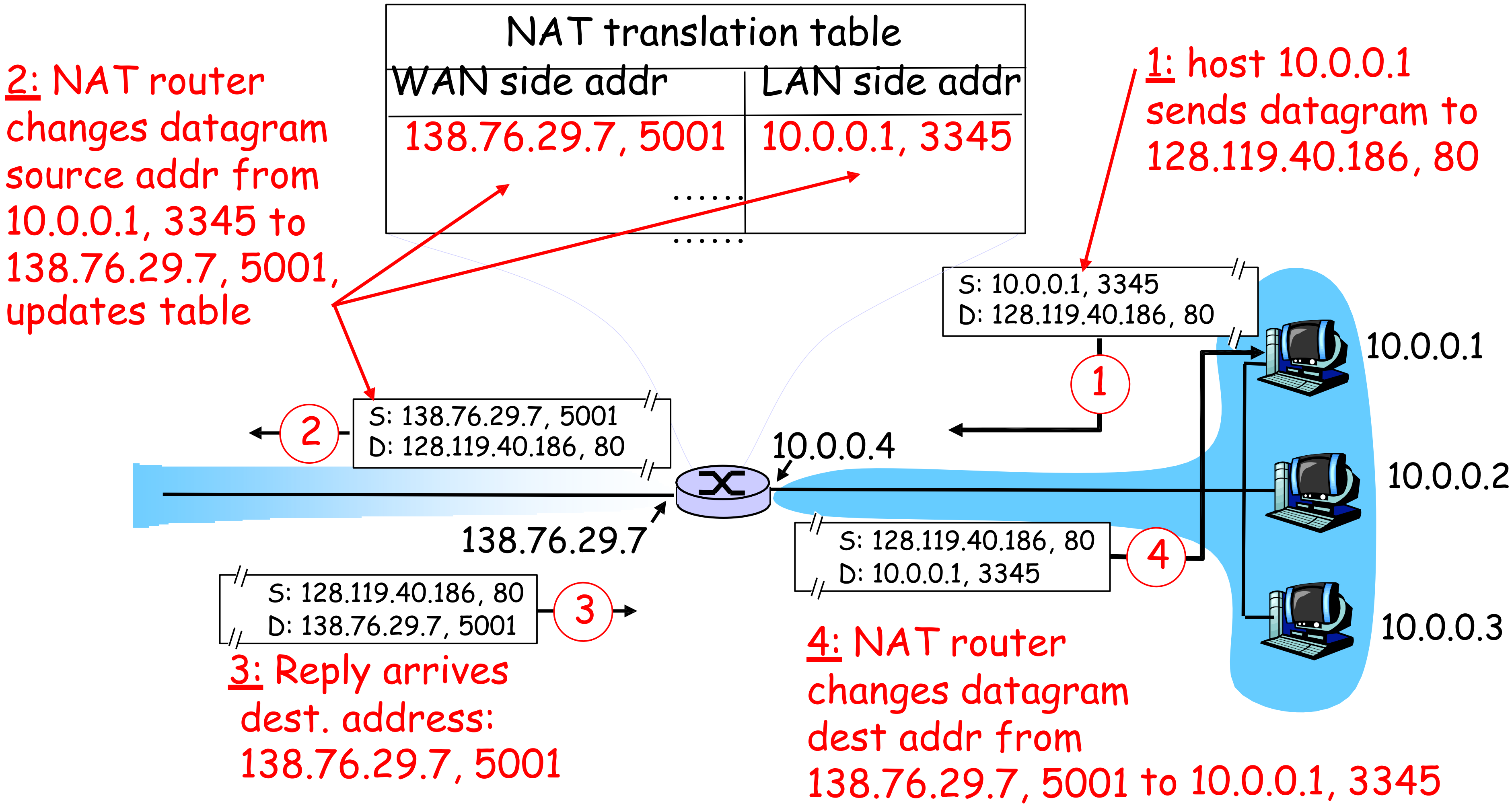
- outgoing datagrams: replace (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)

  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr.

- remember (in NAT translation table) every (source IP address, port #) to (NAT IP address, new port #) translation pair

- incoming datagrams: replace (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: Network Address Translation

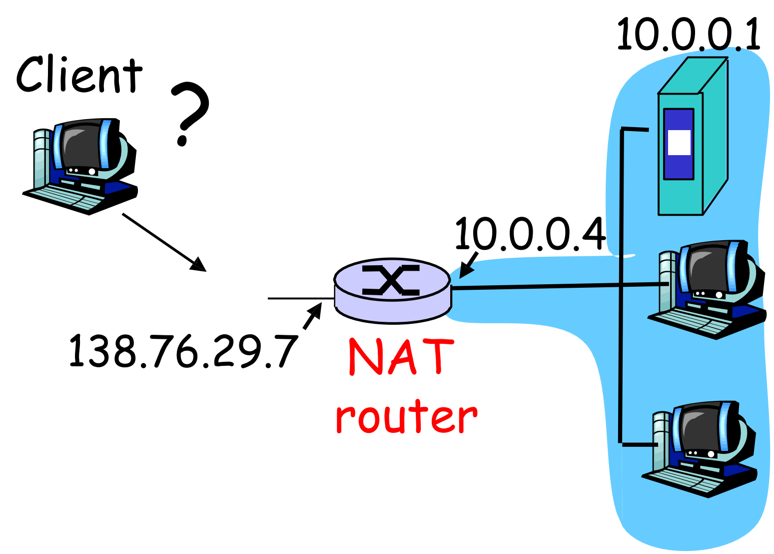


NAT: Network Address Translation

- 16-bit port-number field:
  - 60,000 simultaneous connections with a single LAN-side address!
- NAT is controversial:
  - routers should only process up to layer 3
  - violates end-to-end argument
    - NAT possibility must be taken into account by app designers, eg, P2P applications
  - address shortage should instead be solved by IPv6

NAT traversal problem

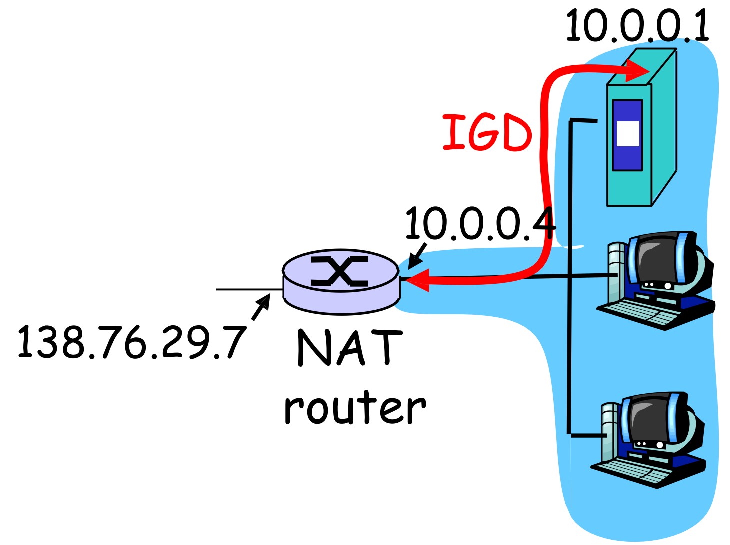
- client wants to connect to server with address 10.0.0.1
  - server address 10.0.0.1 local to LAN (client can"t use it as destination addr)
  - only one externally visible NATted address: 138.76.29.7
- solution 1: statically configure NAT to forward incoming connection requests at given port to server
  - e.g., (138.76.29.7, port 2500) always forwarded to 10.0.0.1 port 2500



NAT traversal problem

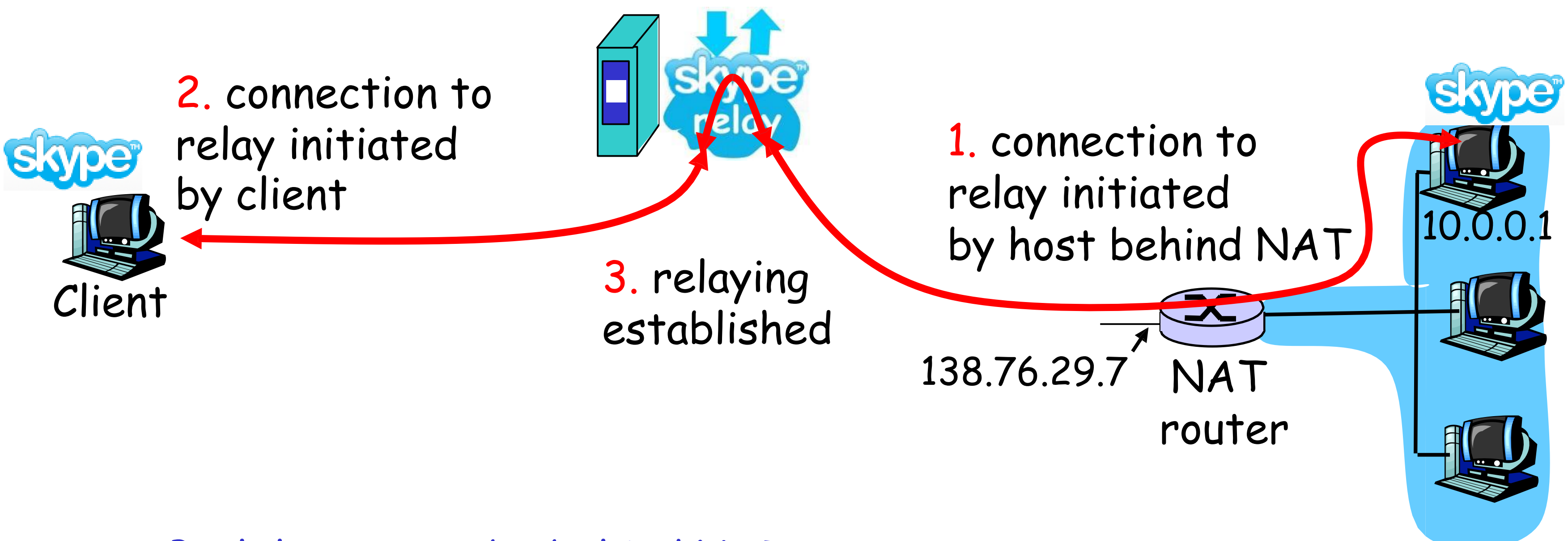
- solution 2: Universal Plug and Play (UPnP) Internet Gateway Device (IGD) Protocol. Allows NATted host to:
  - learn public IP address (138.76.29.7)
  - add/remove port mappings (with lease times)

  i.e., automate static NAT port map configuration



NAT traversal problem

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



Both hosts may be behind NATs

Packet-switching networks
Outline

- Context/overview
- Basic approaches to operating a packet-network: datagrams and virtual circuits
- Network layer functions: Routing and forwarding
- Overview of Network layer: data plane and control plane
- Network layer: The Data Plane
  - What's inside a router
  - The Internet Protocol (IPv4, DHCP, NAT, IPv6)
  - Generalized Forward and SDN
- Network layer: The Control Plane
  - Overview of routing in packet networks
  - The SDN control plane
  - ICMP: The Internet Control Message Protocol

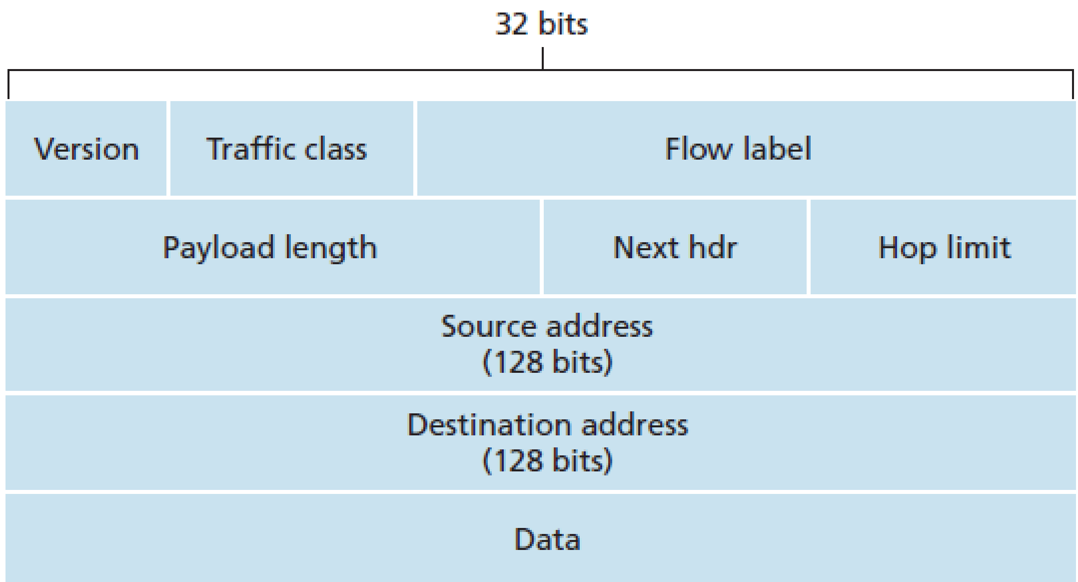
IPv6

RFC 2460

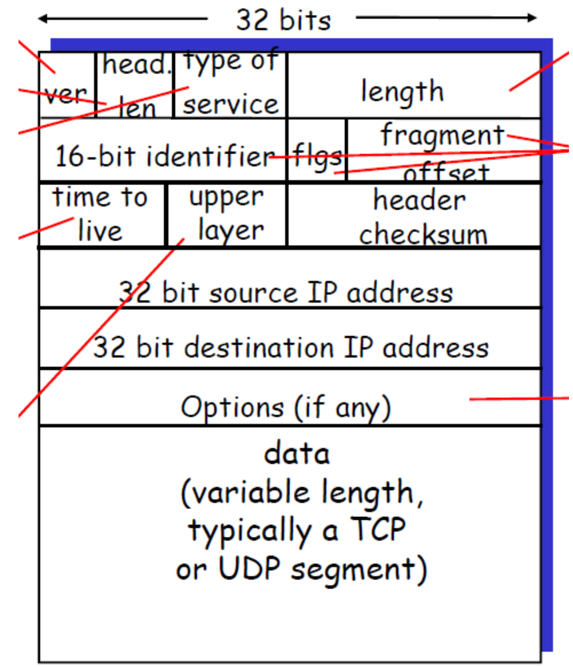
- Initial motivation: 32-bit address space soon to be completely allocated (see below).
- Additional motivation:
  - header format to help speed processing/forwarding
  - header changes to facilitate QoS
  - to facilitate many other features:
    - improved addressing
    - autoconfiguration (or stateless autoconfiguration i.e., without requiring a server such as DHCP server)
    - advanced routing capabilities (source-directed routing, anycast)
    - allow large packets
    - security
    - mobility, …
- IPv6: a good choice for IoT (Internet of Things)

- 3 February 2011, IANA allocated out the last remaining pool of unassigned IPv4 addresses to a regional registry. On 15 April 2011, the APNIC pool reached the last /8 of available IPv4 addresses

IPv6 Header



IPv6



IPv4

- Traffic class: identify priority of datagrams within flow or in different apps
- Flow Label: identify datagrams in same "flow"
- *Next header:* identify upper layer protocol for data. The options field (IPv4) is one of the possible next headers pointed to from within the IPv6 header.

Changes from IPv4

expanded addressing
- 128 bits IPv6 addresses

faster processing/ forwarding
- no fragmentation allowed
- no checksum
- Options: allowed, but outside of the 'base header' as 'extension headers', indicated by "Next Header" field (removal of 'options' results in a fixed-length 40 byte IPv6 header)

ICMPv6: additional message types, e.g. "Packet Too Big"

IPv6 Base and extension headers: examples

Base Header NEXT=TCP	TCP Segment

(a)

Base Header NEXT=ROUTE	Route Header NEXT=TCP	TCP Segment

(b)

Base Header NEXT=ROUTE	Route Header NEXT=AUTH	Auth Header NEXT=TCP	TCP Segment

(c)

IPv6 Address notation

- Each IPv6 address occupies 16 bytes (128 bits)

- IPv6 addresses are expressed in the colon hexadecimal notation
  654E:223F:0:FF26:89A1:5FFD:2980:96A

- Zero compression: A string of repeated zeroes can be replaced by a pair of colons
  FF05:0:0:0:0:0:0:B3 is written as FF05::B3

- Zero compression can be applied only once in a address

- IPv6 Addresses with Embedded IPv4 Addresses
  - IPv4-mapped IPv6 address (e.g., of a host whose IPv4 address 128.96.33.81)
    ::FFFF:128.96.33.81
- Slash notation is also used
  12AB::CD30:0:0:0:0/60

There are three address types in IPv6

- **Unicast** addresses: One-to-one. Destination address specifies a single computer. The packet must be routed to the destination via a shortest path.

- **Anycast** addresses: One-to-anyone of a set. Destination address specifies a set of computers, possibly at different locations. All share a single address. Packet must be routed to **any one in the set** along a shortest path.

- **Multicast** addresses: One-to-many of a set. Destination address specifies a set of computers, possibly at multiple Locations. One copy of the packet should be sent to **selected members of the group.**
  (another type: **Broadcast**: One-to-all in a set. Unlike in IPv4 there is no mechanism in IPv6 to broadcast)

Unicast and anycast address format

bits	48 (or more)	16 (or fewer)	64
field	*routing prefix*	*subnet id*	*interface identifier*

The *network prefix (64 bits)*

(automatically generated from the interface's MAC address using the modified EUI-64 format or obtained from a DHCPv6 server or automatically established randomly, or assigned manually)

Link-local address format

bits	10	54	64
field	*prefix*	*zeroes*	*interface identifier*

contains the binary value 1111111010 (FE80::/10). The 54 zeroes that follow make the total network prefix the same for all link-local addresses (fe80::/64 link-local address prefix), rendering them non-routable.

Stateless address autoconfiguration: On system startup, a node automatically creates a link-local address on each IPv6-enabled interface. It does so independently and without any prior configuration by stateless address autoconfiguration (SLAAC), using a component of the 'Neighbor Discovery Protocol'. This address is selected with the prefix fe80::/64. The lower 64 bits of these addresses are populated with a 64-bit interface identifier in modified EUI-64 format.

Multicast address format

bits	8	4	4	112
field	*prefix*	*flg*	*sc*	*group ID*

binary value 11111111 (ff00::/8)

Prefix (IPv6)	**Explanation**	**IPv4 Equivalent**
::/128	**Unspecified** This address may only be used as a source address by an initialising host before it has learned its own address.	0.0.0.0
::1/128	**Loopback** This address is used when a host talks to itself over IPv6. This often happens when one program sends data to another.	127.0.0.1
::ffff/96 Example: ::ffff:192.0.2.47	**IPv4-Mapped** These addresses are used to embed IPv4 addresses in an IPv6 address. One use for this is in a dual stack transition scenario where IPv4 addresses can be mapped into an IPv6 address. See RFC 4038 for more details.	
fc00::/7 Example: fdf8:f53b:82e4::53	**Unique Local Addresses (ULAs)** These addresses are reserved for local use in home and enterprise environments and are not public address space. The block is split into two halves, the upper half (fd00::/8) is used for "probabilistically unique" addresses in which a 40-bit pseudorandom number is used to obtain a /48 allocation. This means that there is only a small chance that two sites that wish to merge or communicate with each other will have conflicting addresses. No allocation method for the lower half of the block (fc00::/8) is currently defined	Private address space 10.0.0.0/8 172.16.0.0/12 192.168.0.0/16

fe80::/10 Example: fe80::200:5aee:feaa:20a2	**Link-Local Addresses** These addresses are used on a single link or a non-routed common access network, such as an Ethernet LAN. They do not need to be unique outside of that link. Link-local addresses may appear as the source or destination of an IPv6 packet. Routers must not forward IPv6 packets if the source or destination contains a link-local address. Addresses in the link-local prefix are only valid and unique on a single link. Within this prefix only one subnet is allocated (54 zero bits), yielding an effective format of fe80::/64. The least significant 64 bits are usually chosen as the interface hardware address constructed in modified EUI-64 format.	169.254.0.0/16
2000::/3	**Global Unicast** Only one eighth of the total address space is currently allocated for use on the Internet in order to provide efficient route aggregation, thereby reducing the size of the Internet routing tables; the rest of the IPv6 address space is reserved for future use or for special purposes. The address space is assigned to the RIRs in large blocks of /23 up to /12. The RIRs assign smaller blocks to local Internet registries that distributes them to users. These are typically in sizes from /19 to /32. The addresses are typically distributed in /48 to /56 sized blocks to the end users.	No equivalent single block

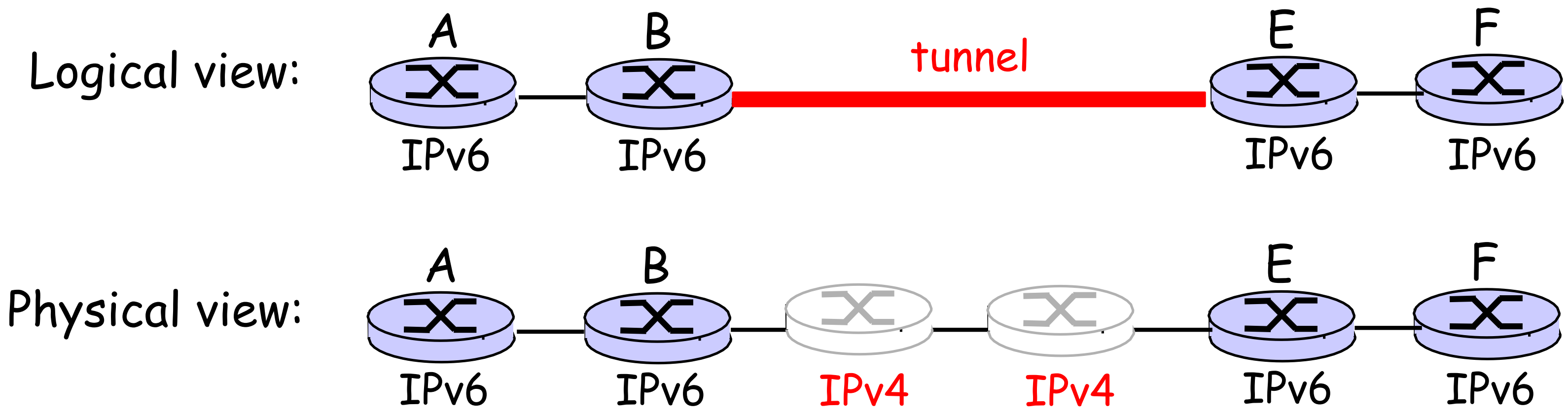
ff00::/8 Example: ff01:0:0:0:0:0:0:2	**Multicast** These addresses are used to identify multicast groups. They should only be used as destination addresses, never as source addresses.	224.0.0.0/4
There are special purpose addresses: 2001:0000::/32 2001:0002::/48 2001:0010::/28 2002::/16 2001:db8::/32	 Teredo Benchmarking Orchid 6to4 Documentation	
	Reserved anycast addresses The lowest address within each subnet prefix (the interface identifier set to all zeroes) is reserved as the "subnet-router" anycast address. Applications may use this address when talking to any one of the available routers, as packets sent to this address are delivered to just one router. The 128 highest addresses within each /64 subnet prefix are reserved to be used as anycast addresses.	

::/0	**Default route** The default route address covering all addresses (unicast, multicast and others).	0.0.0.0/0

Transition From IPv4 To IPv6

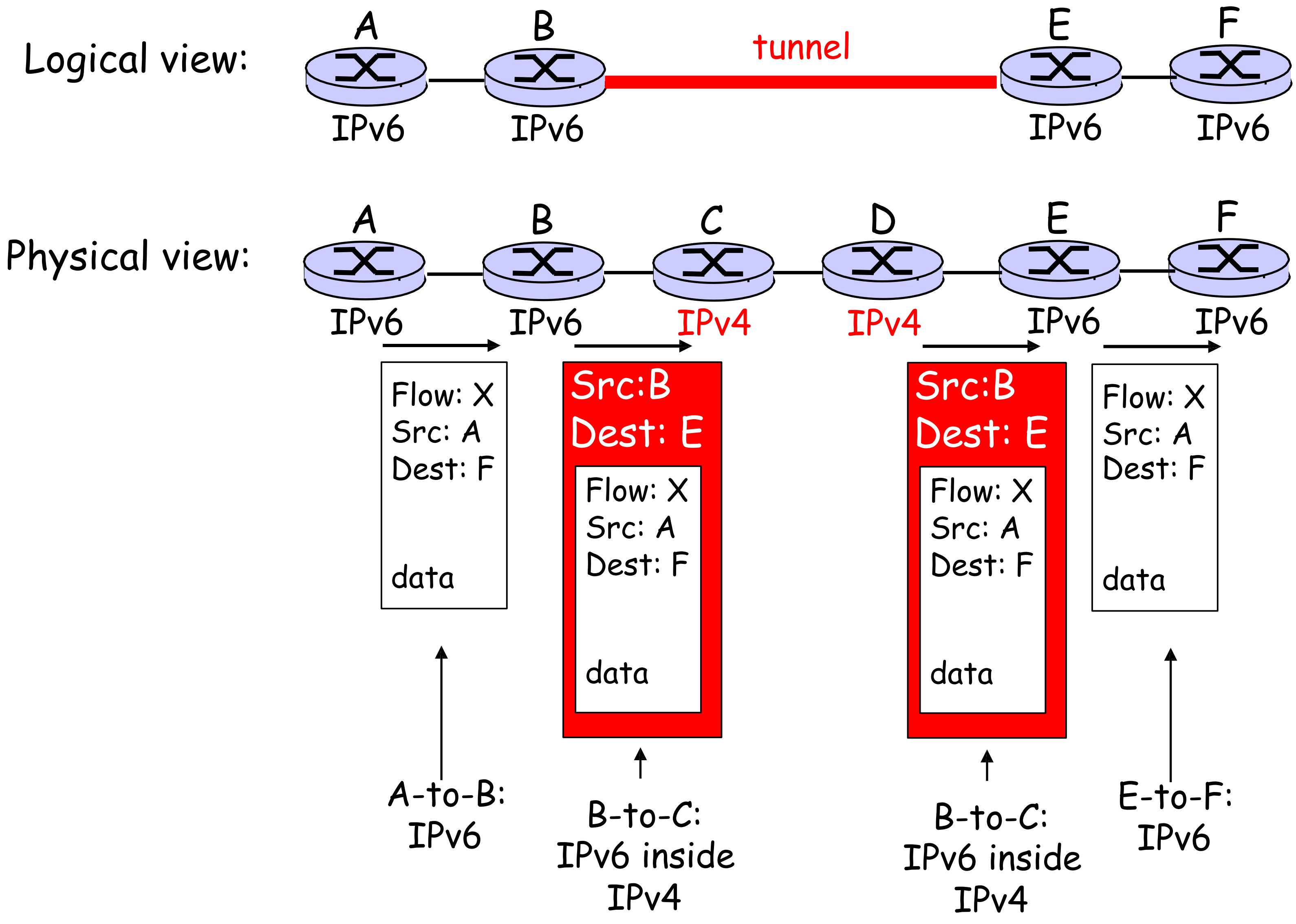
- Not all routers can be upgraded simultaneous
  - no "flag day"
  - How will the network operate with mixed IPv4 and IPv6 routers?
- *Tunneling:* IPv6 carried as payload in IPv4 datagram among IPv4 routers

Tunneling



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

Tunneling


Logical view:
A
B
tunnel
E
F
IPv6
IPv6
IPv6
IPv6
Physical view:
A
B
C
D
E
F
IPv6
IPv6
IPv4
IPv4
IPv6
IPv6
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Src:B
Dest: E
Flow: X
Src: A
Dest: F
data
Flow: X
Src: A
Dest: F
data
A-to-B: IPv6
B-to-C: IPv6 inside IPv4
B-to-C: IPv6 inside IPv4
E-to-F: IPv6