F9: TCP/IP Architecture and Its Vulnerabilities

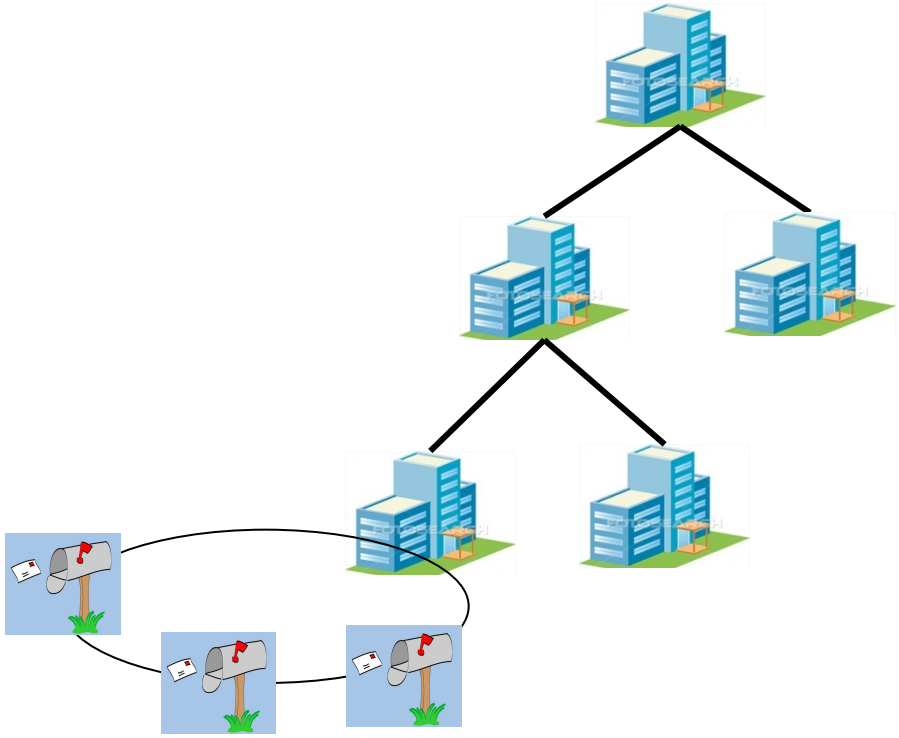
Many slides are from Jim Kurose and Keith Ross

outline

- Intro
- Appl
- TCP
- IP
- DNS
- ARP

Postal mail

- Mail address
- Mailbox at each house
- Local post office
- Hierarchical mail system
  - routing
- Some details
  - Contents invisible
  - Mail loss possible
  - Parcel size limited
  - Express delivery
  - Registered mail
  - sender address optional



From: Joe Biden
1600 Pennsylvania Avenue
Washington D.C., DC 20500

Via air mail

To: Rishi Sunak
10 Downing Street,
London, SW1A 2AA

Telephone network

- Telecommunication
- Aka PSTN or POTS
- Phone number
- Voice services
- Fixed phone or mobile phone
  - Dumb terminal & smart network
- signaling
- More expensive than VoIP
  - Better quality than VoIP
- VoLTE

Network vs service

Network classification

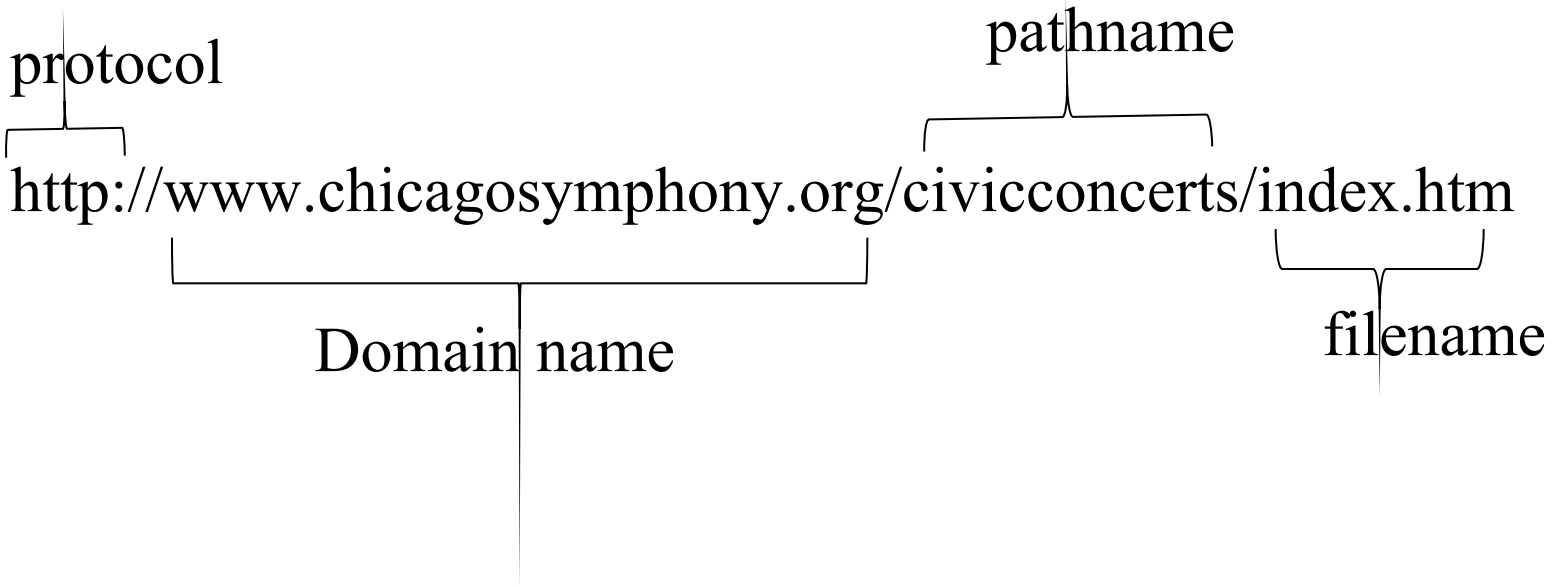
- Contents or traffic types
  - Telecommunications vs. data communications
- Range or size
  - BAN, PAN, LAN, MAN, WAN
  - Sizes vary depending on wired or wireless
- Medium
  - copper, optical fiber, air, water,…
  - electric signal, EM wave, light, acoustic sound
- Applications or services
  - IPTV, Sensor network, Walkie-Talkie
- Physicality or deployment
  - Satellite, CATV, …
- Infrastructure vs. ad hoc

Three elements of a network

Internet is... (technical)

- an electronic communications network that connects computer networks and organizational computer facilities around the world @merriam-webter
- a worldwide system of computer networks - a network of networks in which users at any one computer can, if they have permission, get information from any other computer @whatis
- A global computer network providing a variety of information and communication facilities, consisting of interconnected networks using standardized communication protocols @oxford

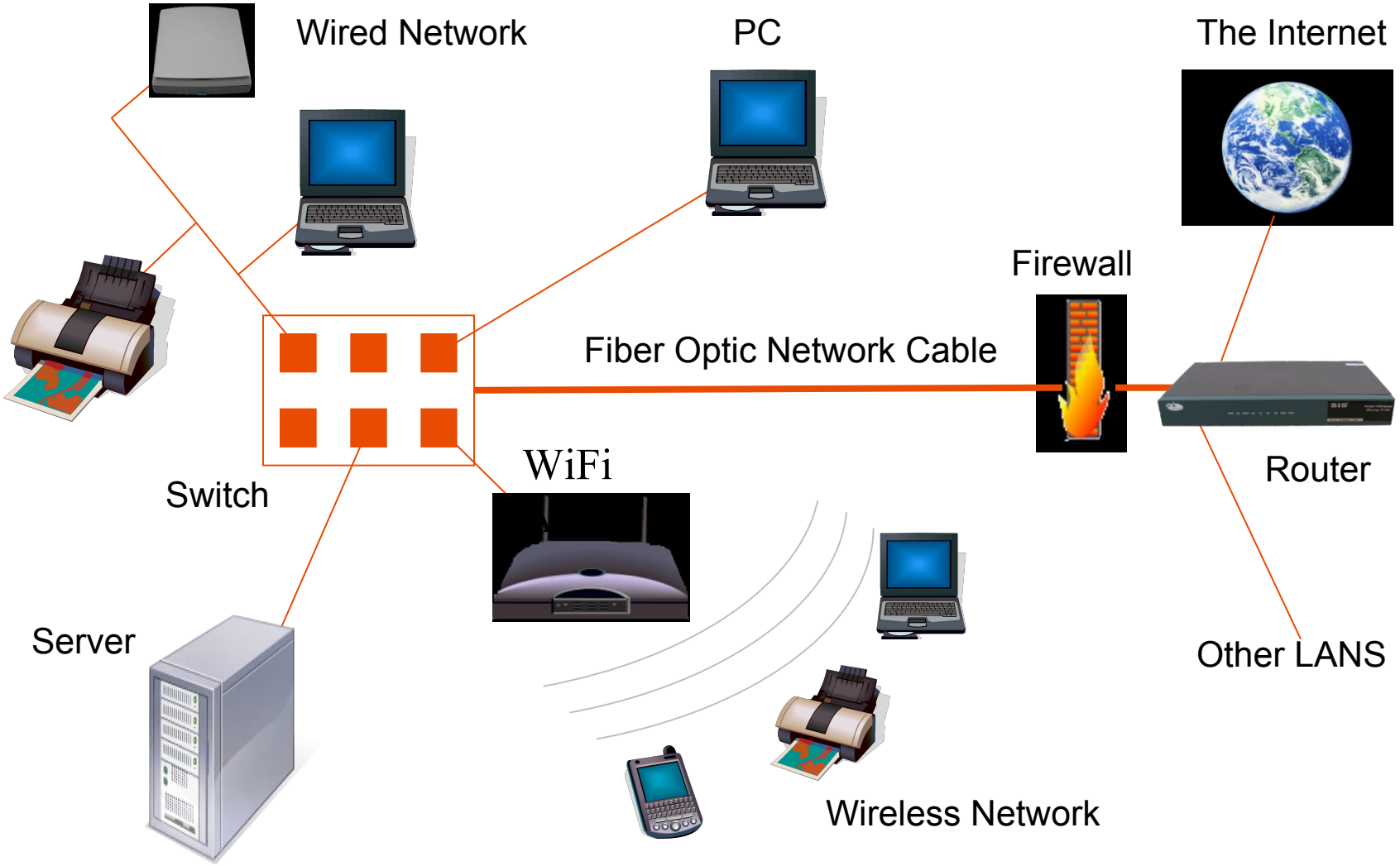
Address in Internet?



www.snu.ac.kr =>147.46.10.58

http: Hypertext Transfer Protocol

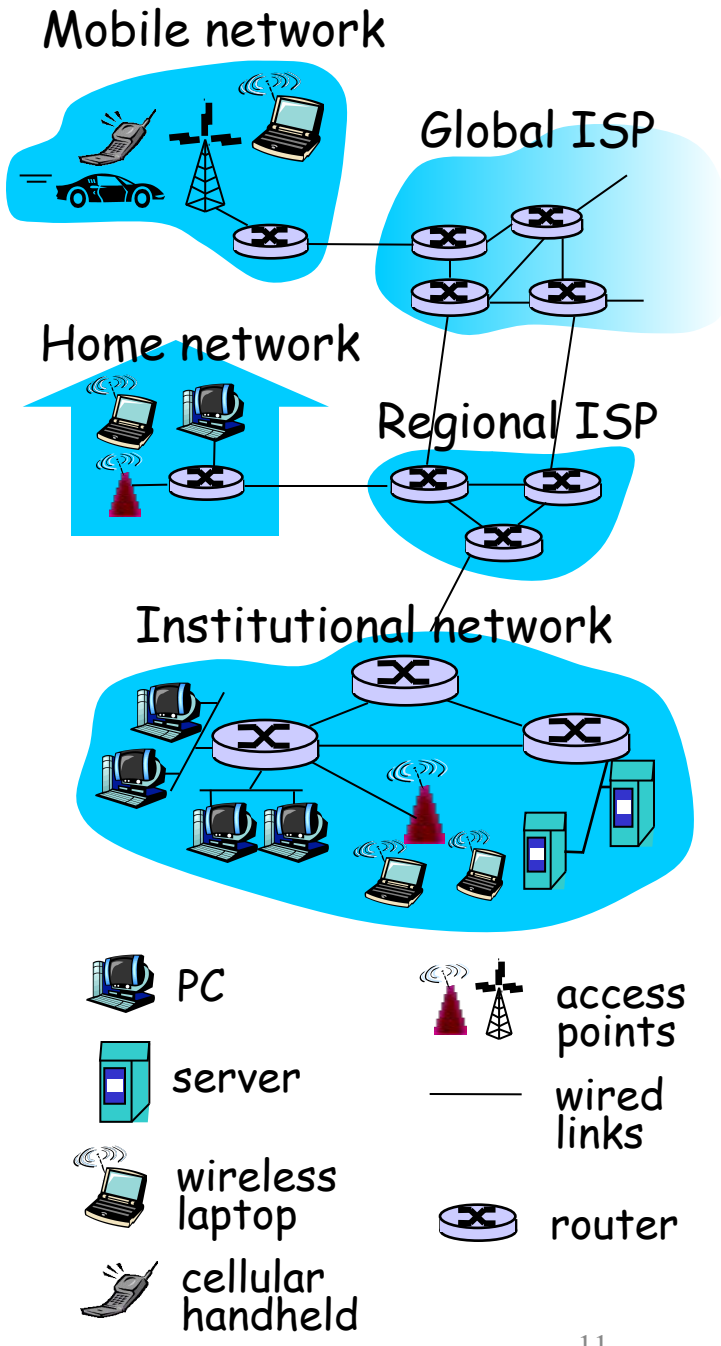
A SOHO network: illustration



Source: David Phenix

Internet: illustration

- millions of connected computing devices: *hosts = end systems*
  - running *network apps*
- *communication links*
  - fiber, copper, radio, satellite
  - transmission rate = *bandwidth*
- *routers:* forward packets (chunks of data)



*ISP: internet service provider

What's the Internet: network for service

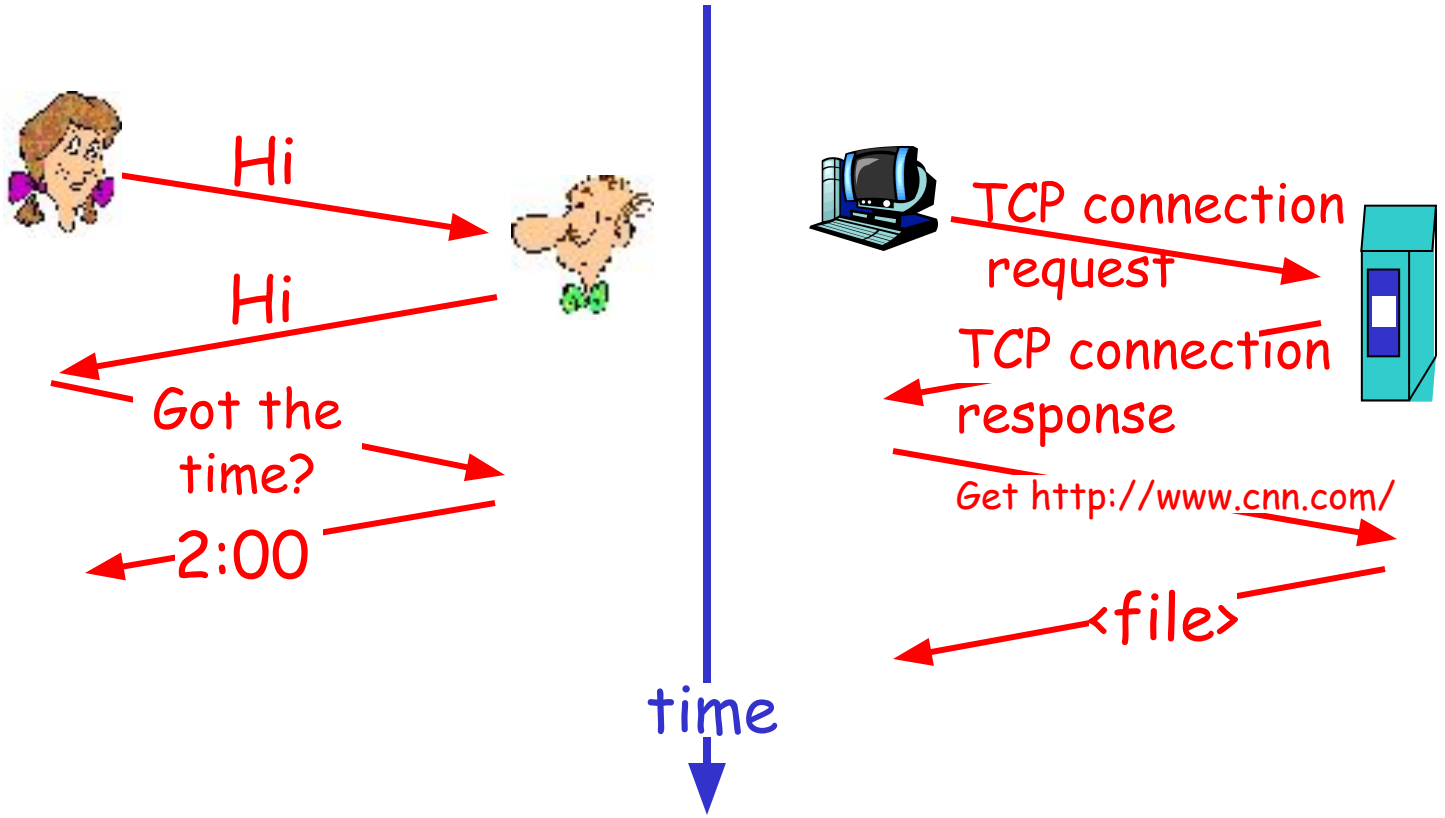
- *protocols* control sending, receiving of msgs
  - e.g., TCP, IP, HTTP, Skype, Ethernet
- *Internet:* "network of networks"
  - loosely hierarchical
  - public Internet versus private intranet
- Internet standards
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force

- communication *infrastructure* enables distributed applications:
  - Web, VoIP, email, games, e-commerce, file sharing
- communication services provided to apps:
  - reliable data delivery from source to destination
  - "best effort" (unreliable) data delivery

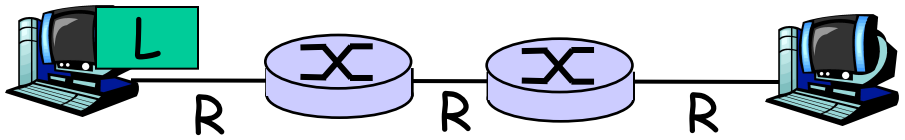
*msg: message, VoIP: voice over IP

What's a protocol?

a human protocol and a computer network protocol:



Packet-switching: store-and-forward



- takes L/R seconds to transmit (push out) packet of L bits on to link at R bps
- *store and forward:* entire packet must arrive at router before it can be transmitted on next link
- delay = 3L/R (assuming zero propagation delay)

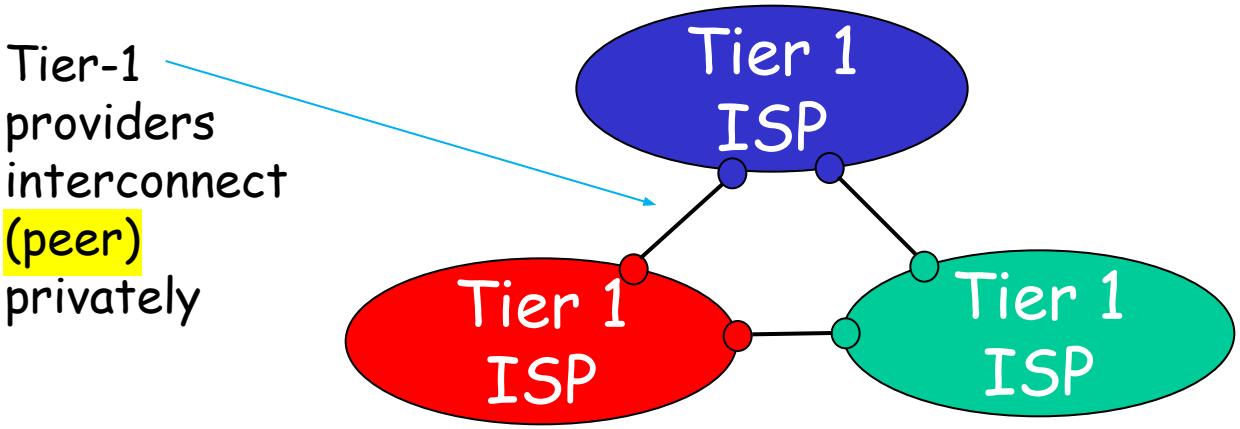
Example:

- L = 7.5 Mbits
- R = 1.5 Mbps
- transmission delay = 15 sec

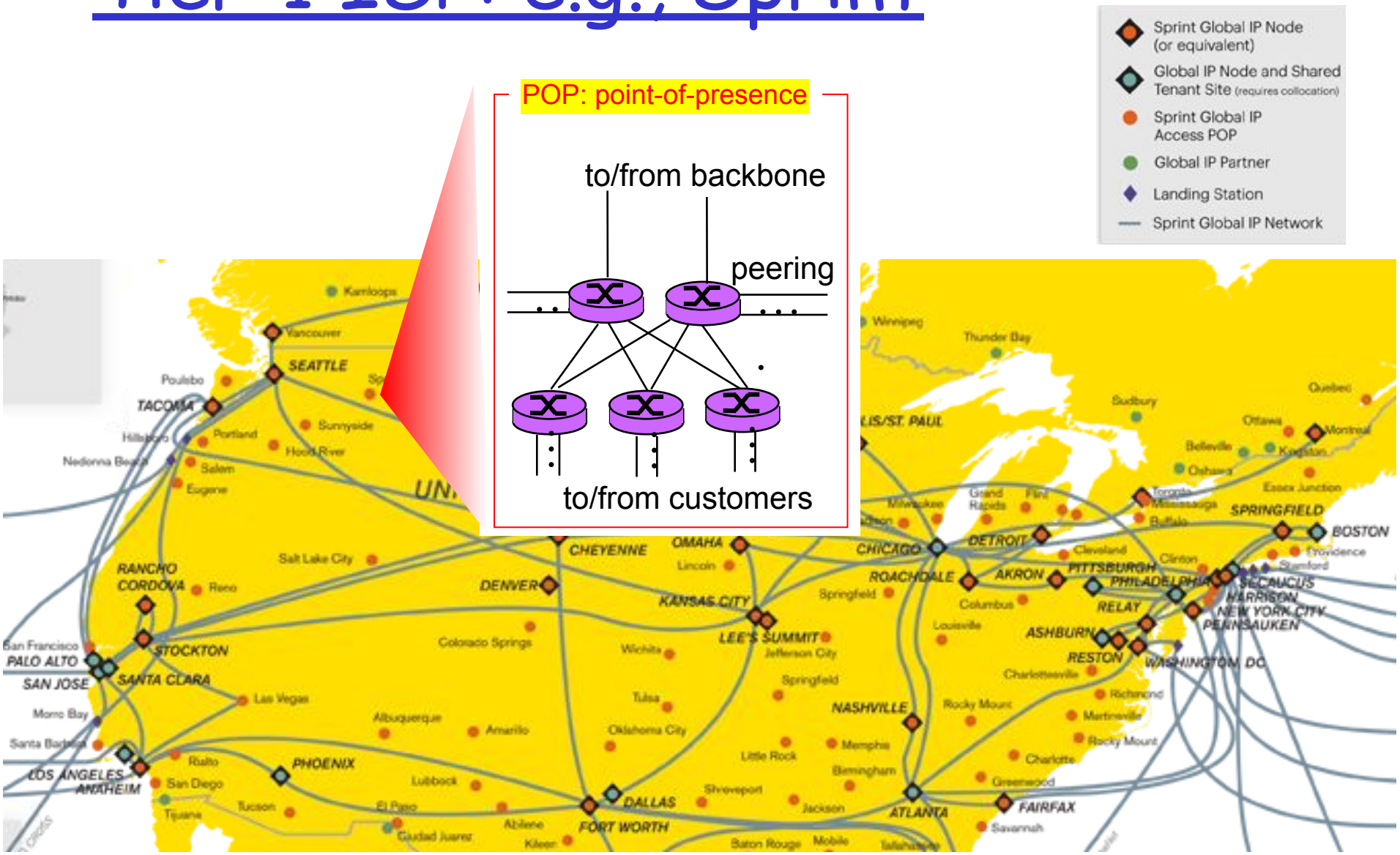
INTERNET STRUCTURE: HARD PART

Internet structure: network of networks

- roughly hierarchical
- at center: "tier-1" ISPs (e.g., Verizon, Sprint, AT&T), national/international coverage
  - treat each other as equals



Tier-1 ISP: e.g., Sprint

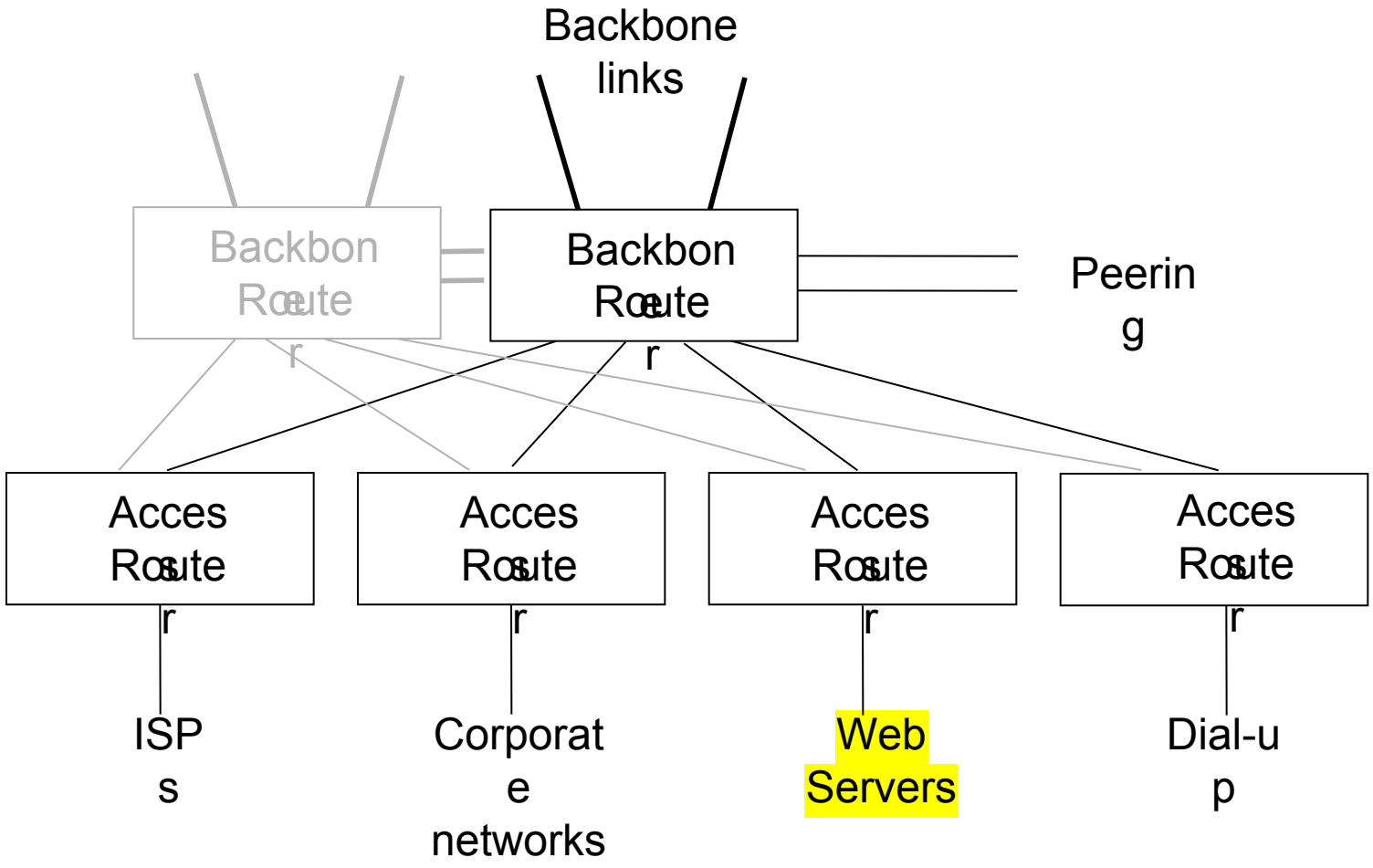


highway junction vs exit

- highway interchange?

PoP illustration

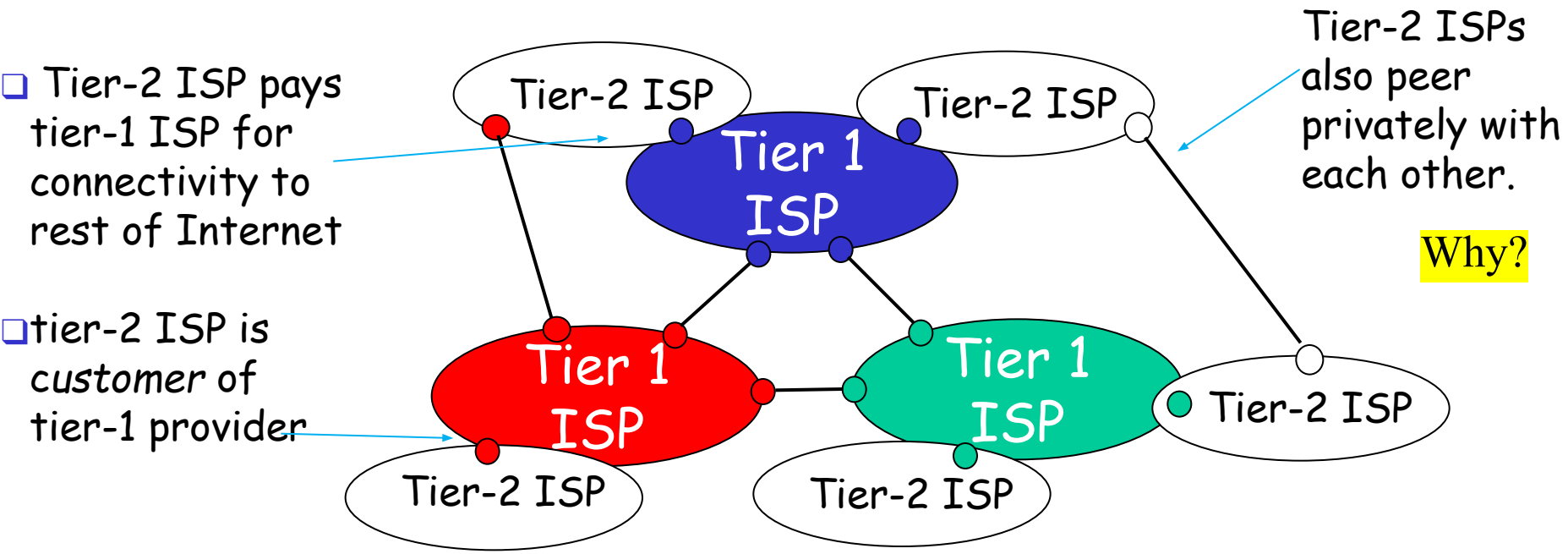
backbone router: highway junction
PoP: highway exit (interchange)



* Colocation center

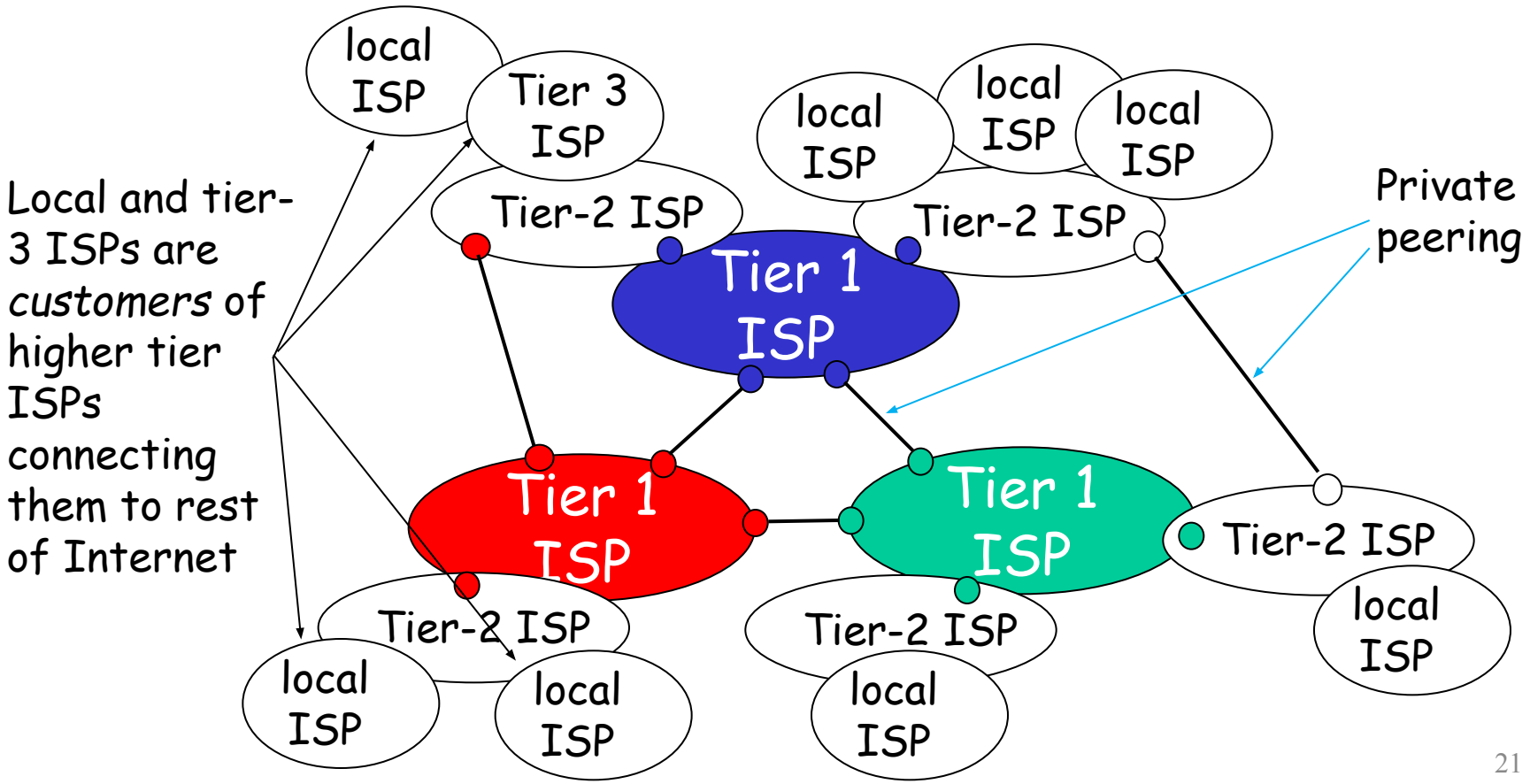
Internet structure: network of networks

- "Tier-2" ISPs: smaller (often regional) ISPs
  - Connect to one or more tier-1 ISPs, possibly other tier-2 ISPs



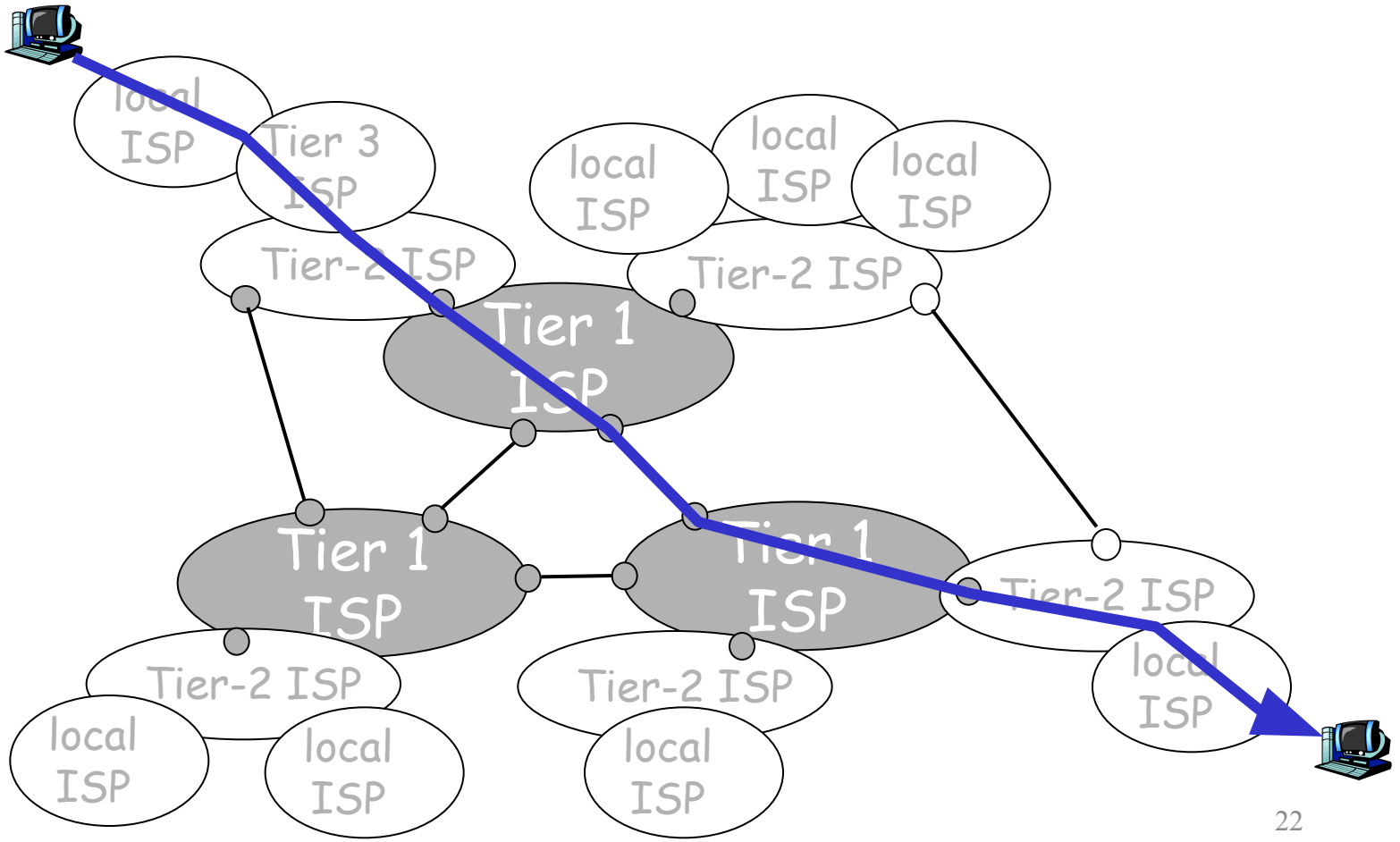
Internet structure: network of networks

- "Tier-3" ISPs and local ISPs
  - last hop ("access") network (closest to end systems)



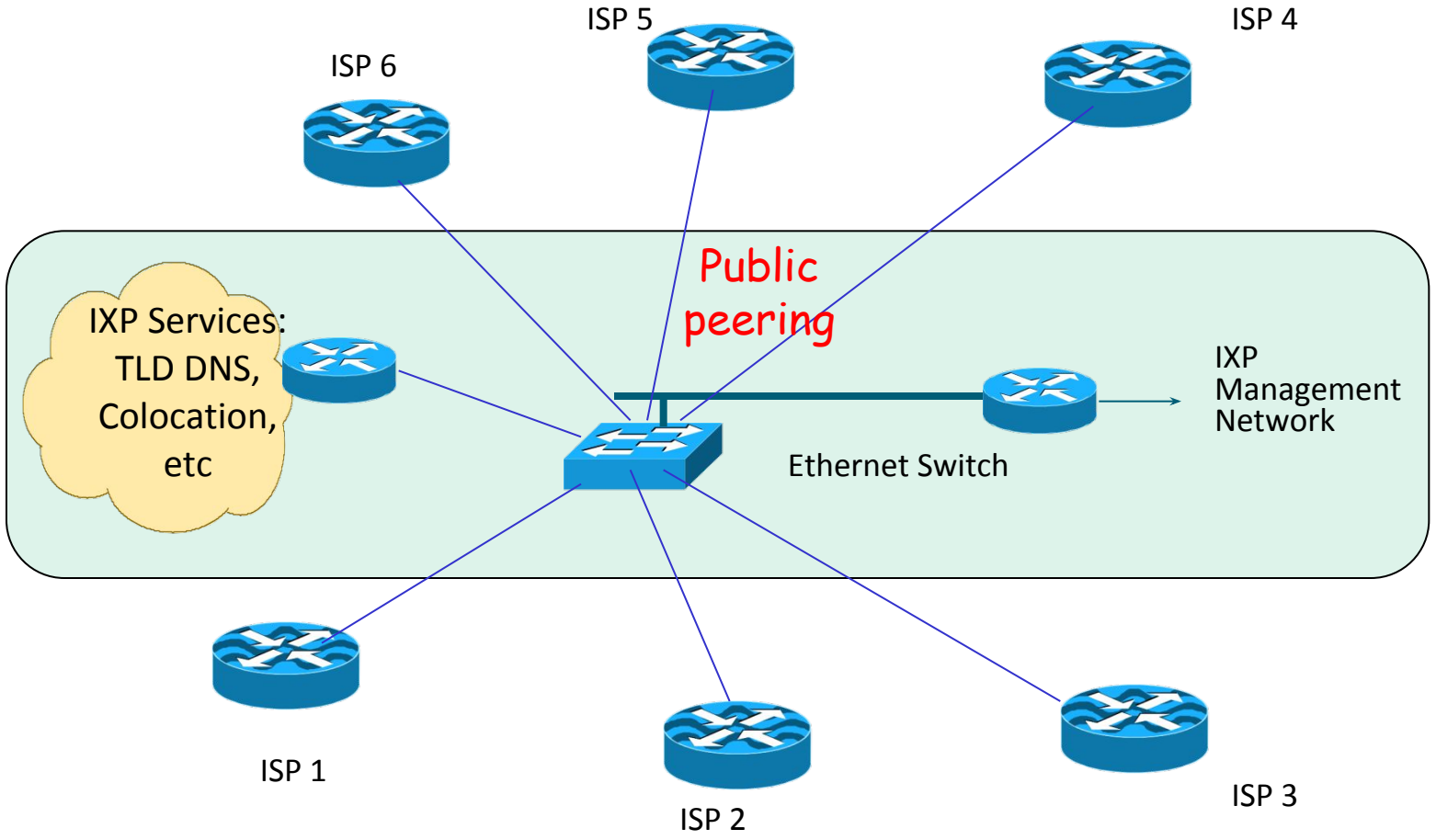
Internet structure: network of networks

- a packet passes through many networks!



Internet eXchange (Point)

- IX or IXP



Source: Michuki@Afrinic

INTERNET STRUCTURE: SOFT PART

Protocol "Layers"

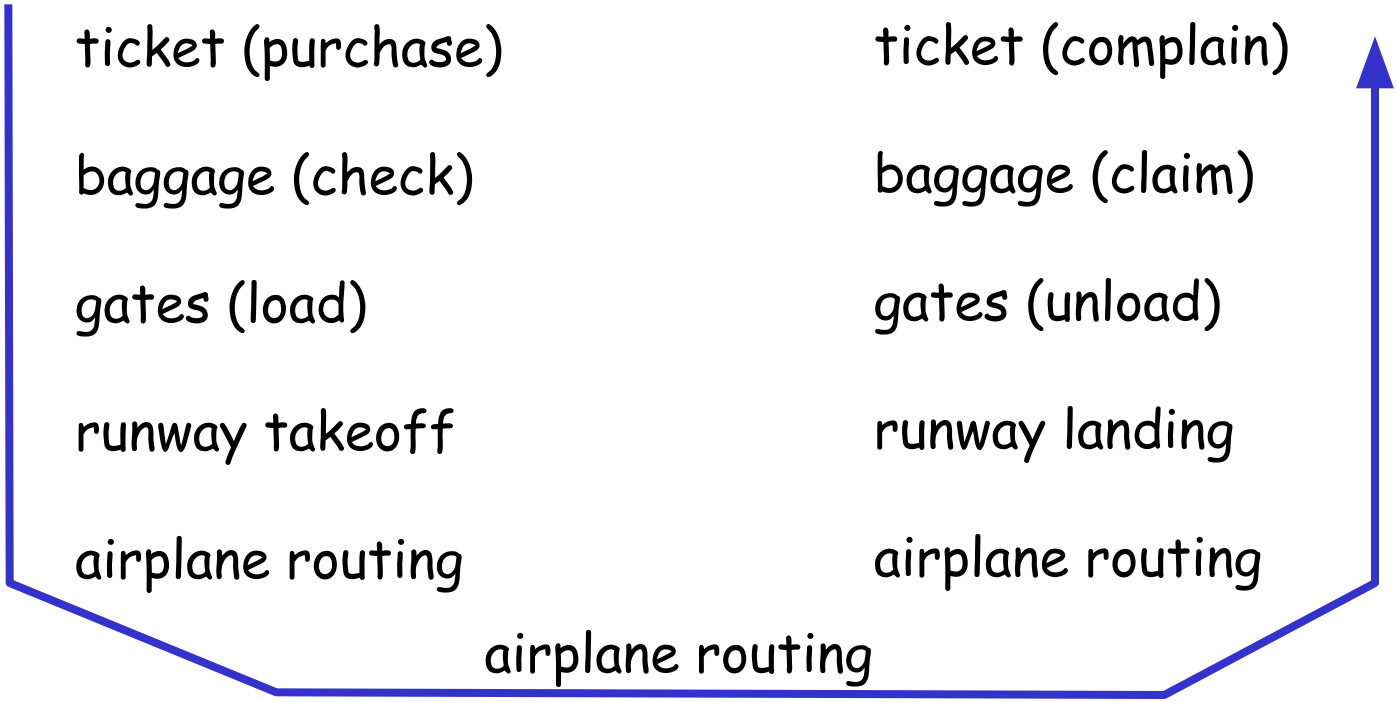
Networks are complex!

- many "pieces":
  - hosts
  - routers
  - links of various media
  - applications
  - protocols
  - hardware, software

Question:

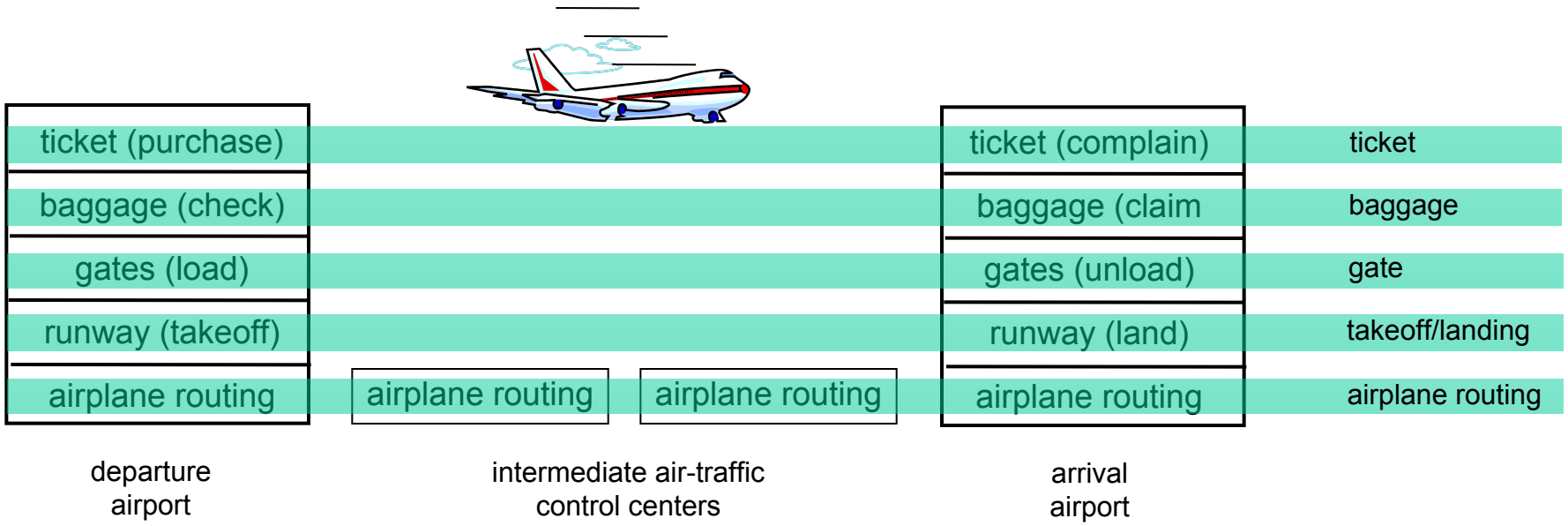
Is there any hope of *organizing* a structure of networking pieces?

Organization of air travel



- a series of steps

Layering of airline functionality



Layers: each layer implements a service

- via its own internal-layer actions
- relying on services provided by layer below

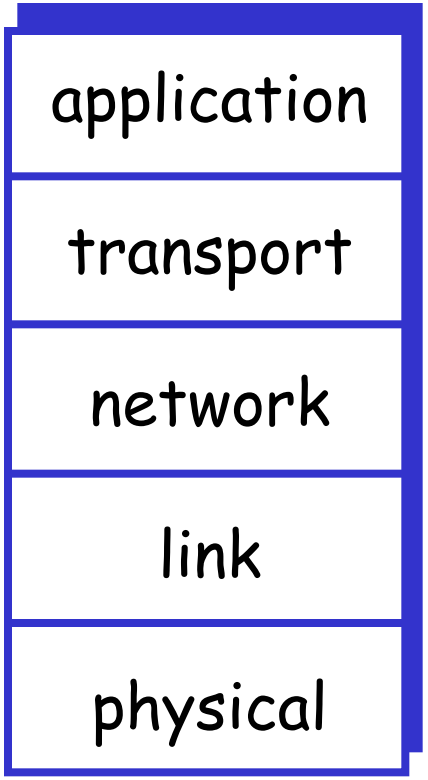
Why layering?

Dealing with complex systems:

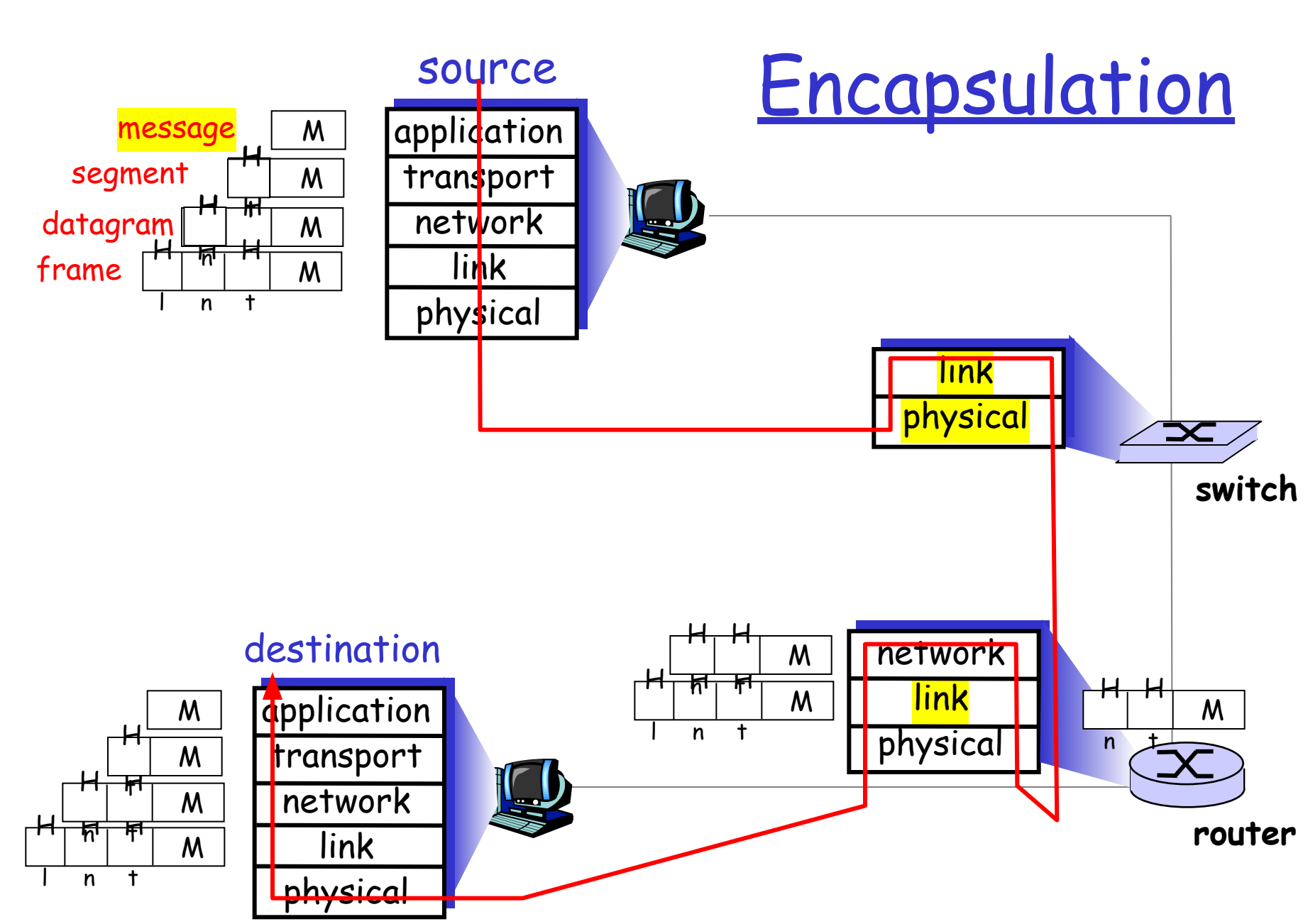
- explicit structure allows identification, relationship of complex system's pieces
  - layered reference model for discussion
- modularization eases maintenance, updating of system
  - change of implementation of layer's service transparent to rest of system: information hiding
  - e.g., change in gate procedure doesn't affect rest of system
- layering considered harmful?

Internet protocol stack

- application: supporting network applications
  - FTP, SMTP, HTTP
- transport: process-process data transfer
  - TCP, UDP
- network: routing of datagrams from source to destination
  - IP, routing protocols
- link: data transfer between neighboring network elements
  - PPP, Ethernet
- physical: bits "on the wire"

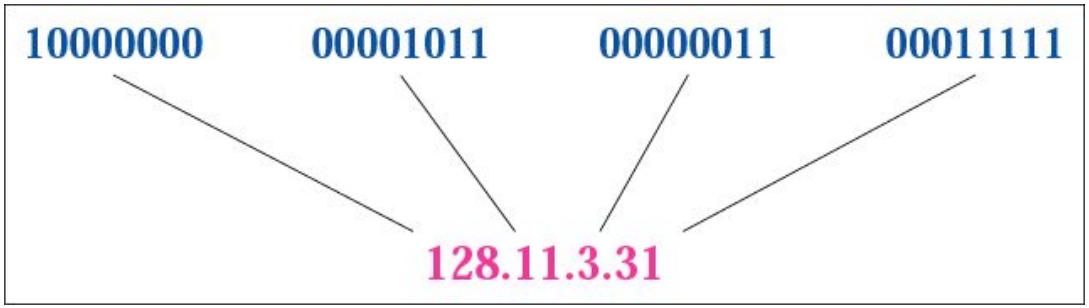


Encapsulation



Internet routing

- Internet's two-tiered topology in routing
  - Autonomous Systems, and connections between them
  - Routers, and the links between them in an AS
- AS-level topology
  - Autonomous System (AS) numbers
  - Business relationships between ASes
- IP address: 32 bits



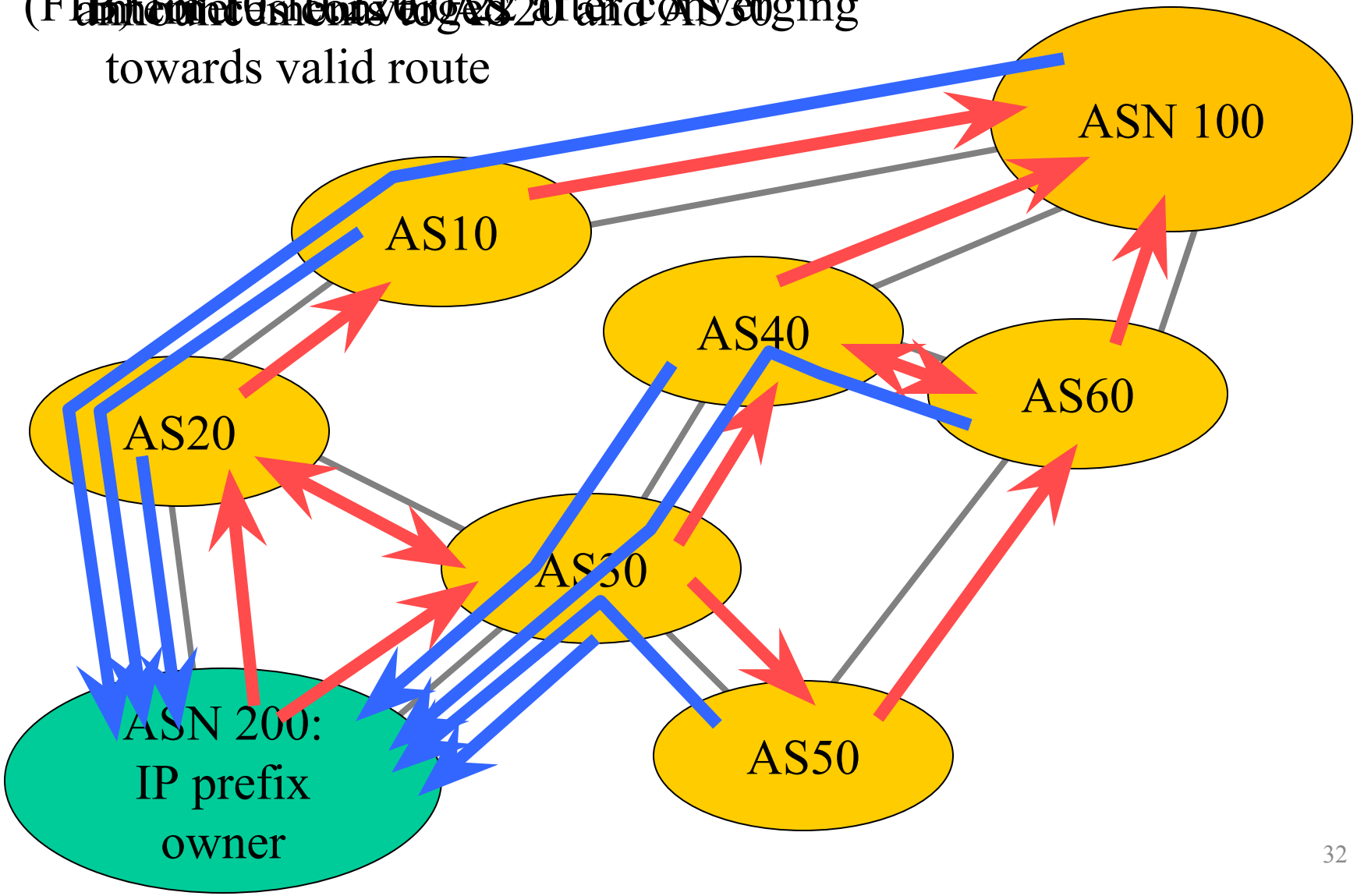
- SNU has an address block 147.46.0.0 – 147.46.255.255
  - SNU's IP prefix 147.46.0.0/16

BGP advertises IP prefixes

View of Forwarding Information Base (FIB) for 10.10.220.0/22 after convergence

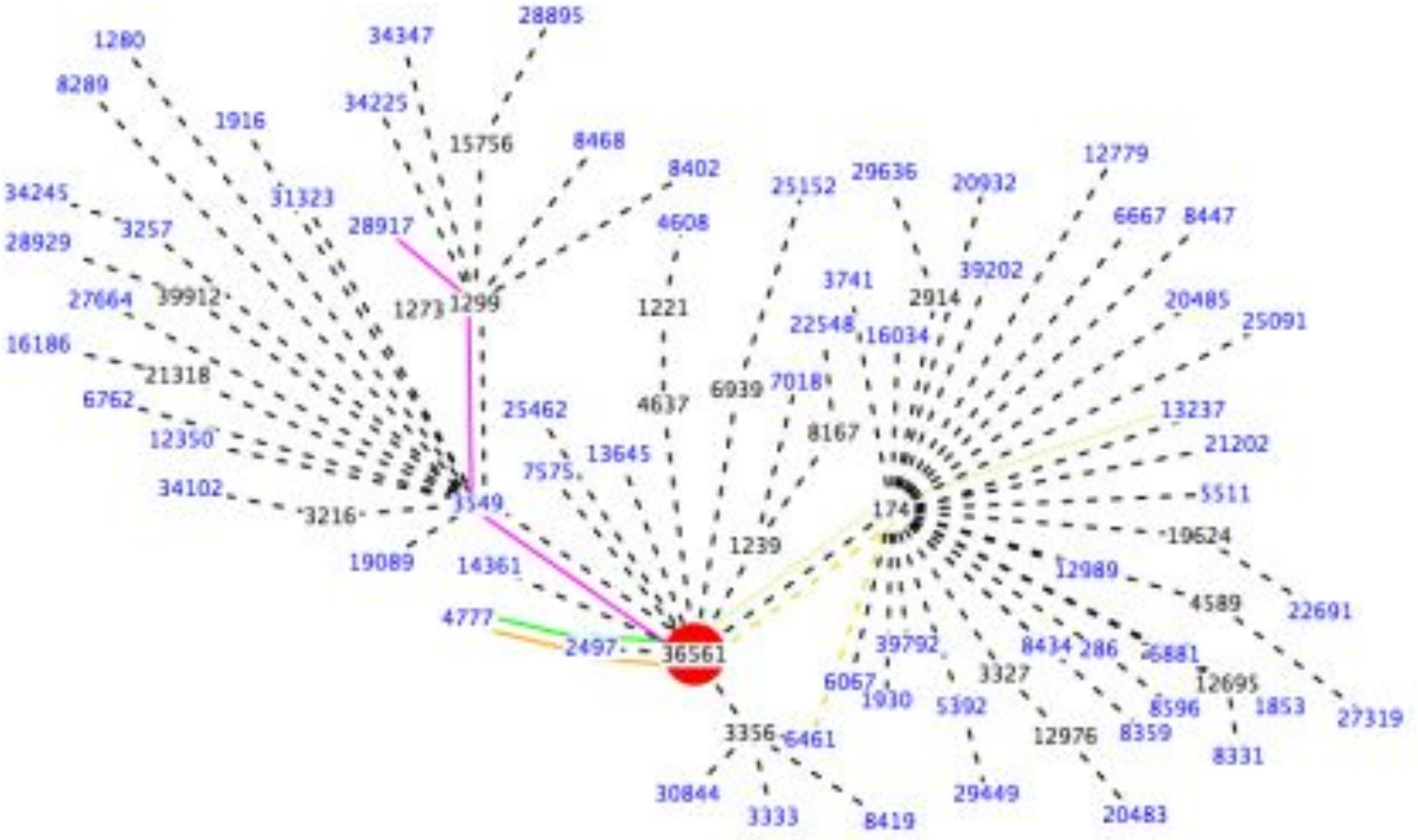
ASN 200 originates IP prefix 10.10.220.0/22, sends announcements to AS20 and AS30

Based on FIB, packets destined for 10.10.220.0/22 are forwarded towards AS20 and AS30 towards valid route



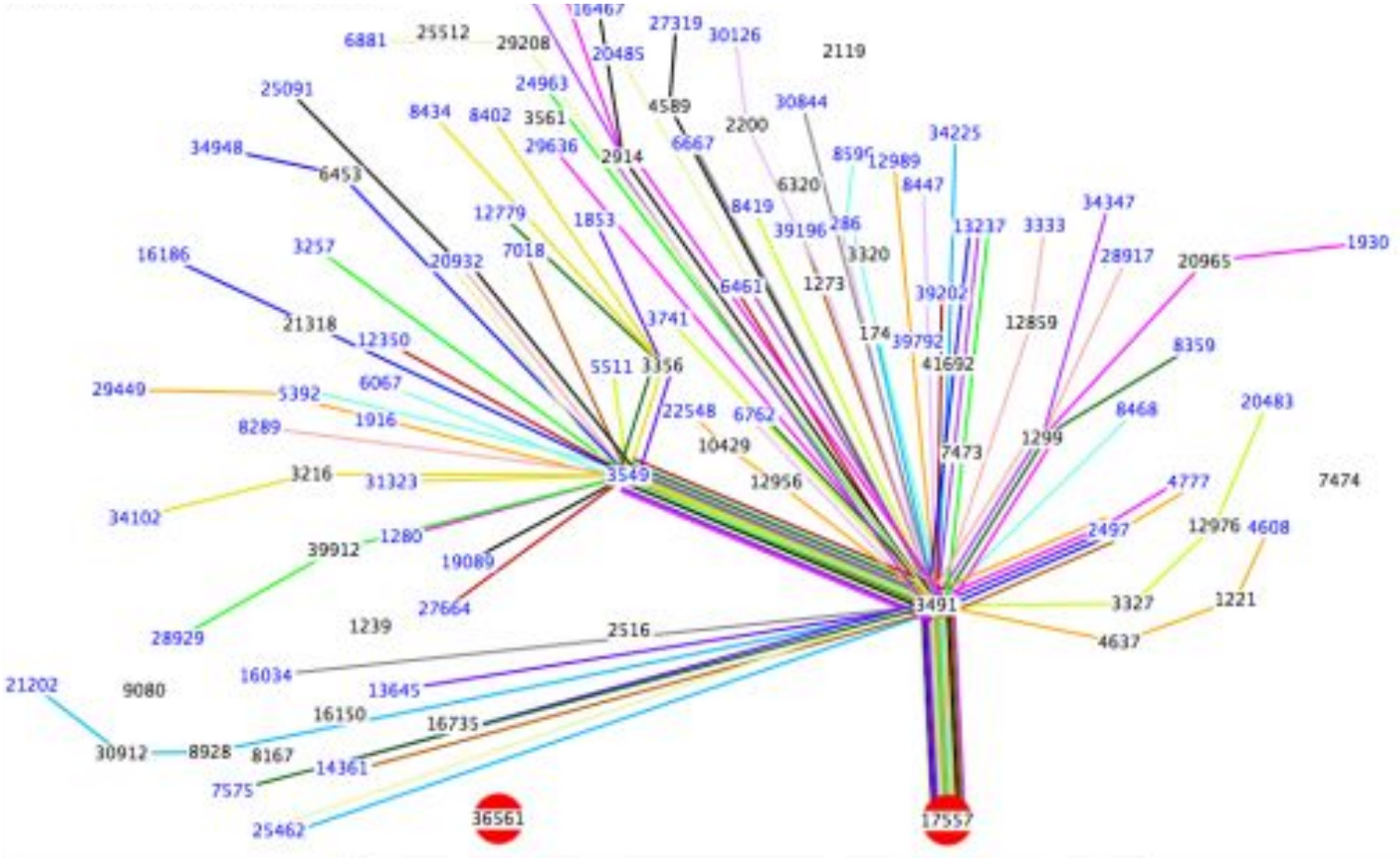
YouTube (Normally)

- AS36561 (YouTube) advertises



YouTube (February 24, 2008)

- Pakistan government wants to block YouTube
  - AS17557 (Pakistan Telecom) advertises 208.65.153.0/24
  - All YouTube traffic worldwide directed to AS17557



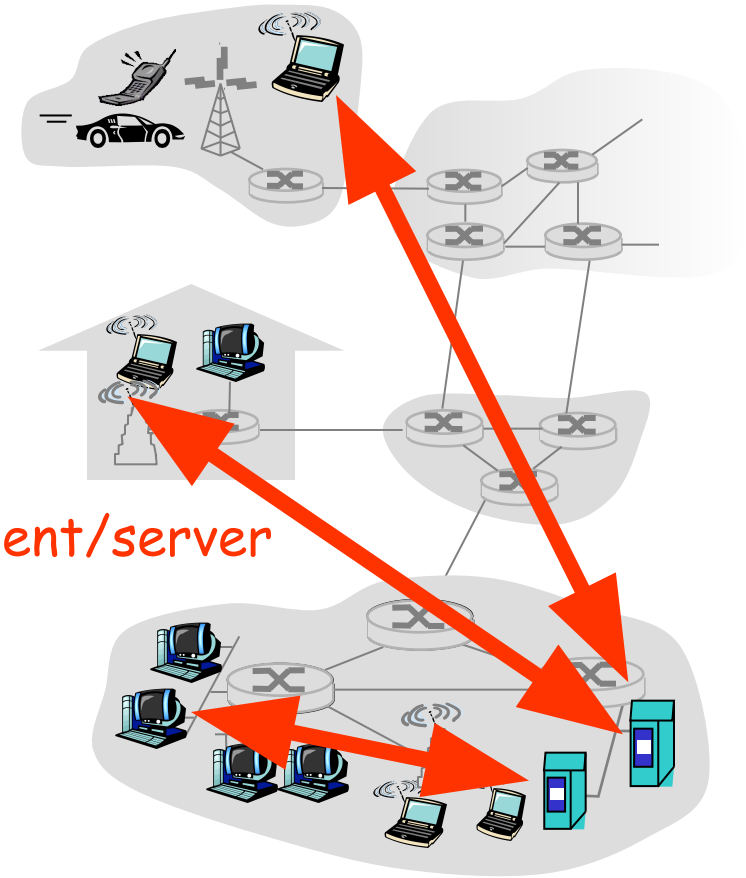
outline

- Intro
- Appl
- TCP
- IP
- DNS
- ARP

Some Internet apps

- e-mail
- web
- instant messaging
- remote login
- P2P file sharing
- multi-user network games
- streaming stored video clips
- voice over IP
- real-time video conferencing
- grid computing
- Tor
- Cybercurrency
- Blockchain

Client-server architecture



server:

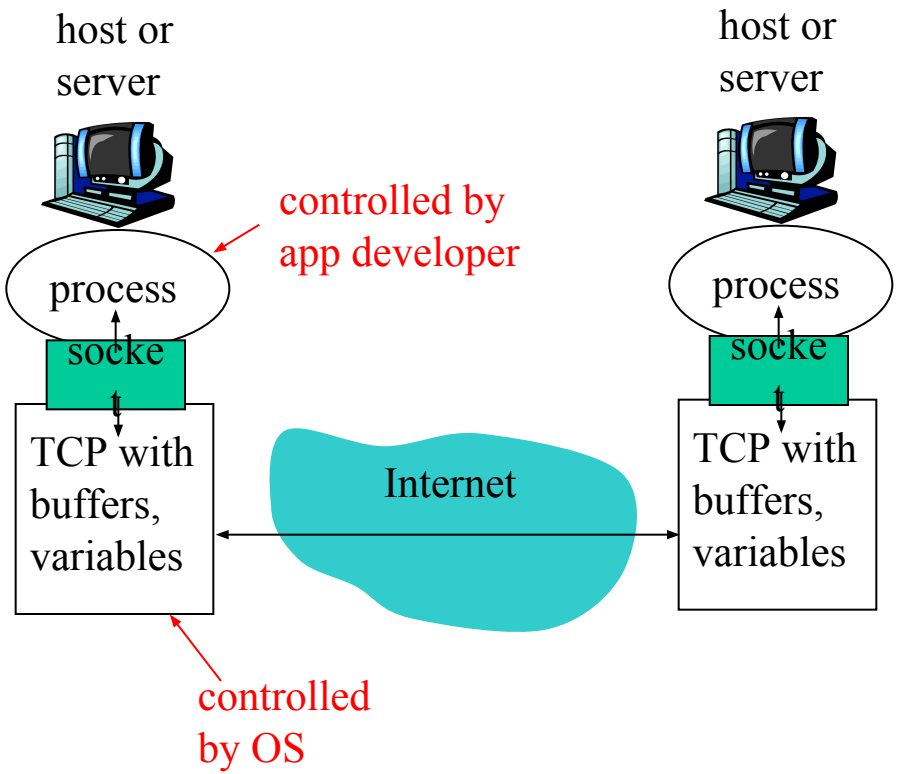
- always-on host
- permanent IP address
- server farms for scaling

clients:

- communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other

Sockets

- process sends/receives messages to/from its socket
- socket analogous to door
  - sending process shoves message out door
  - sending process relies on transport infrastructure on other side of door which brings message to socket at receiving process



r API: (1) choice of transport protocol; (2) ability to fix a few parameters

Addressing processes

- to receive messages, process must have *identifier*
- host device has unique 32-bit IP address
- Q: does IP address of host on which process runs suffice for identifying the process?
  - A: No, *many* processes can be running on same host
- *identifier* includes both IP address and port numbers associated with process on host.
- Example port numbers:
  - HTTP server: 80
  - Mail server: 25
- to send HTTP message to gaia.cs.umass.edu web server:
  - IP address: 128.119.245.12
  - Port number: 80

Internet transport protocols services

TCP service:

- *connection-oriented:* setup required between client and server processes
- *reliable transport* between sending and receiving process
- *flow control:* sender won't overwhelm receiver
- *congestion control:* throttle sender when network overloaded
- *does not provide:* timing, minimum throughput guarantees, security

UDP service:

- unreliable data transfer between sending and receiving process
- does not provide: connection setup, reliability, flow control, congestion control, timing, throughput guarantee, or security

Q: why bother? Why is there a UDP?

Internet apps: application, transport protocols

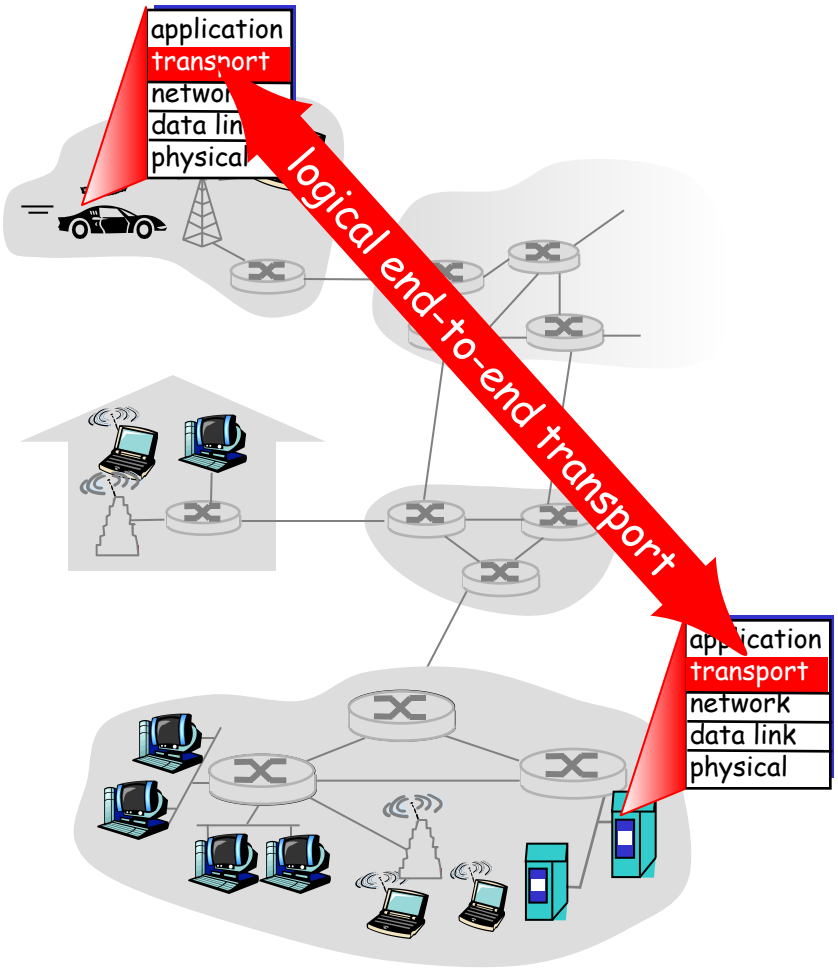
Application	Application layer protocol	Underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g. Youtube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g., Skype)	typically UDP

outline

- Intro
- Appl
- TCP
- IP
- DNS
- ARP

Transport services and protocols

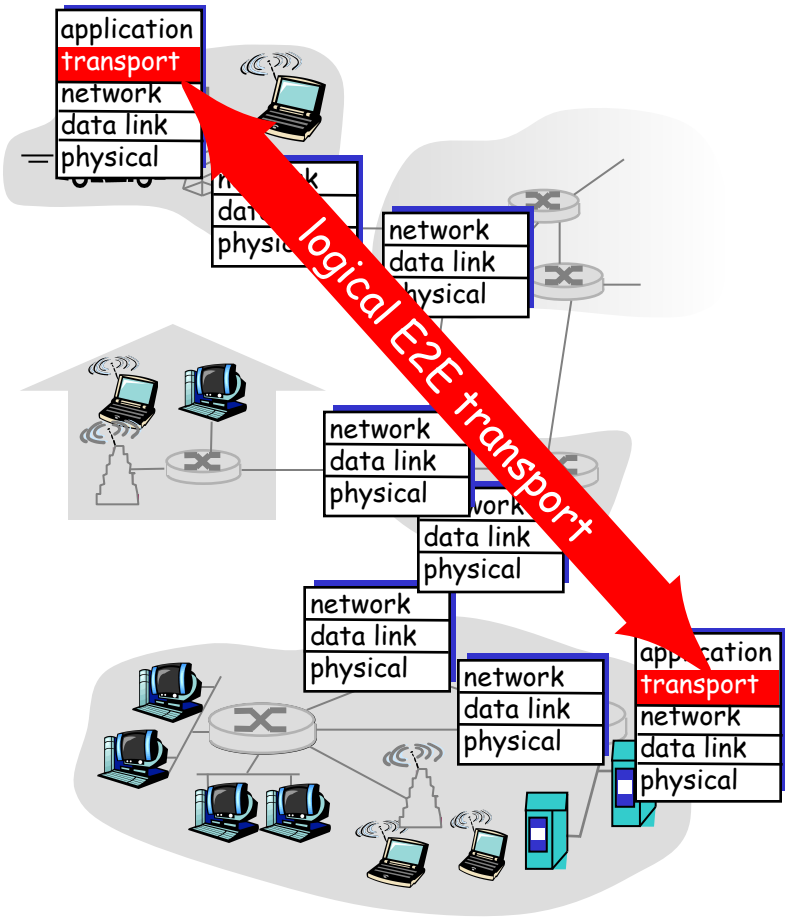
- provide *logical communication* between app processes running on different hosts
- transport protocols run in end systems
  - send side: breaks app messages into segments, passes to network layer
  - rcv side: reassembles segments into messages, passes to app layer
- more than one transport protocol available to apps
  - Internet: TCP and UDP



* rcv = recv = receive

Internet transport-layer protocols

- reliable, in-order delivery (TCP)
  - congestion control
  - flow control
  - connection setup
- unreliable, unordered delivery: UDP
  - no-frills extension of "best-effort" IP
- services not available:
  - delay guarantees
  - bandwidth guarantees



Multiplexing/demultiplexing

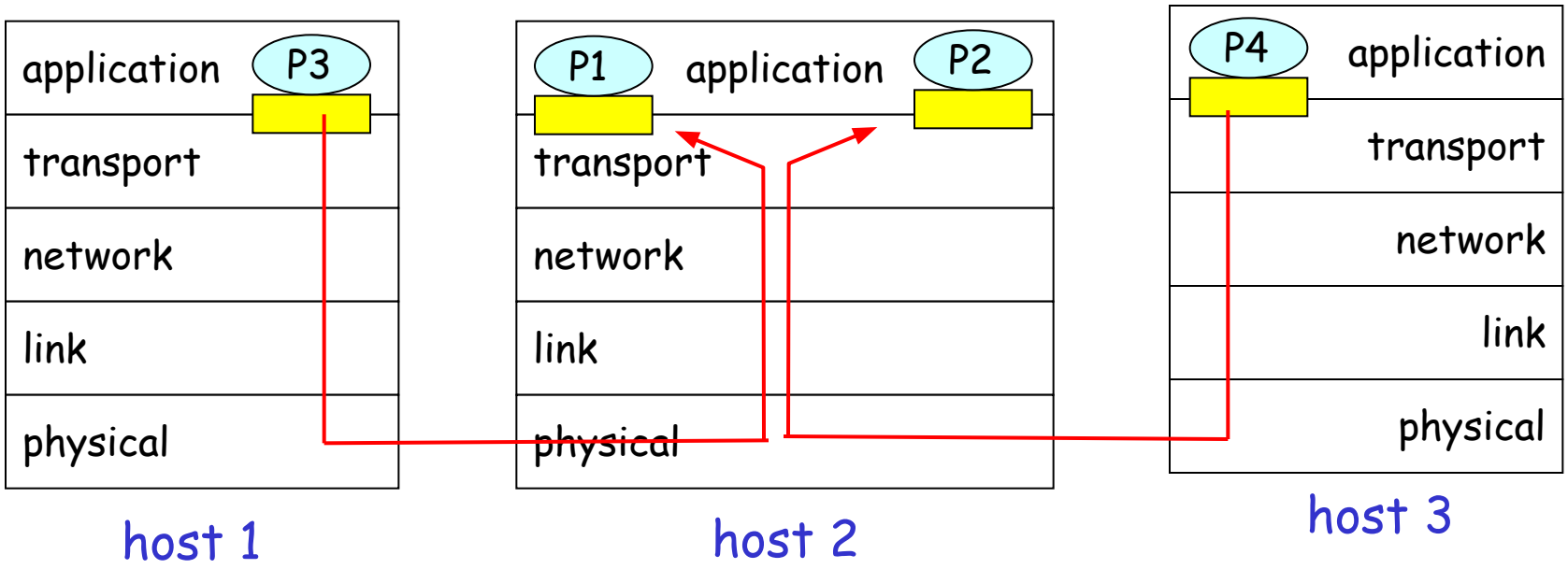
Demultiplexing at rcv host:

delivering received segments to correct socket

Multiplexing at send host:

gathering data from multiple sockets, enveloping data with header (later used for demultiplexing)

= socket = process



Connection-oriented demux

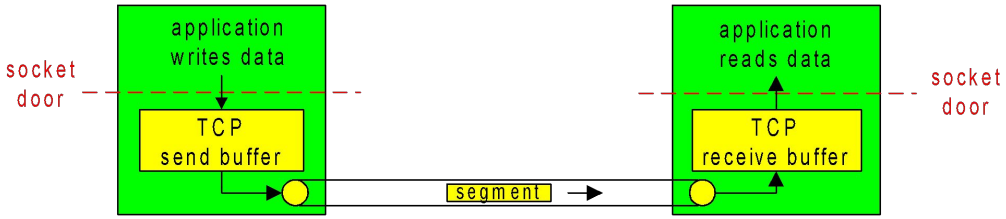
- TCP socket identified by 4-tuple:
  - source IP address
  - source port number
  - dest IP address
  - dest port number
- recv host uses all four values to direct segment to appropriate socket
- Server host may support many simultaneous TCP sockets:
  - each socket identified by its own 4-tuple
- Web servers have different sockets for each connecting client
  - non-persistent HTTP will have different socket for each request

* demux: demultiplex, mux: multiplex

TCP: Overview

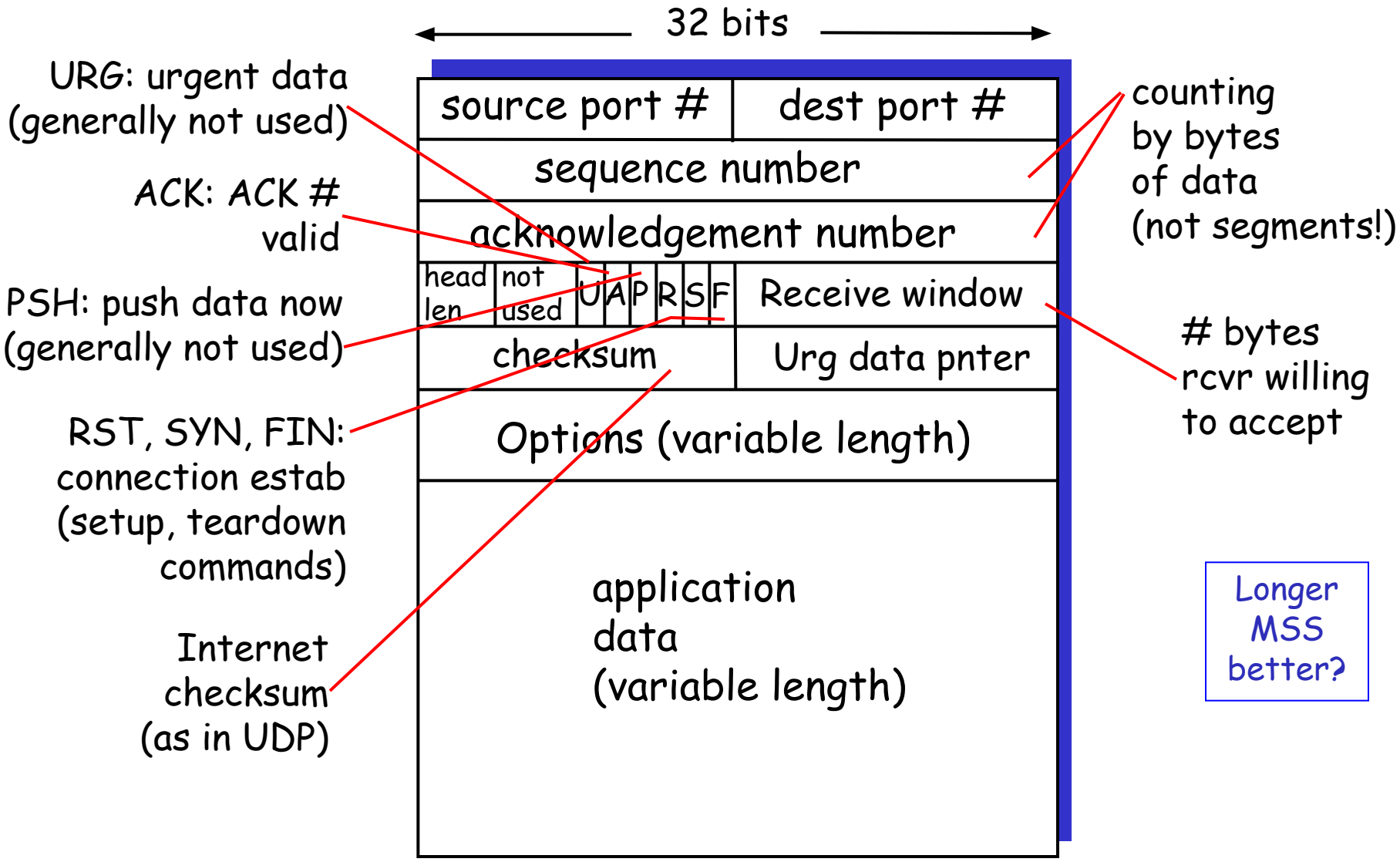
RFCs: 793, 1122, 1323, 2018, 2581

- point-to-point:
  - one sender, one receiver
- reliable, in-order *byte steam:*
  - no "message boundaries"
- pipelined:
  - TCP congestion and flow control set window size
- *send & receive buffers*



- full duplex data:
  - bi-directional data flow in same connection
  - MSS: maximum segment size
- connection-oriented:
  - handshaking (exchange of control msgs) initiates sender, receiver state before data exchange
- flow controlled:
  - sender will not overwhelm receiver's memory

TCP segment structure



* rcvr = receiver

TCP seq. #'s and ACKs

Seq. #'s:

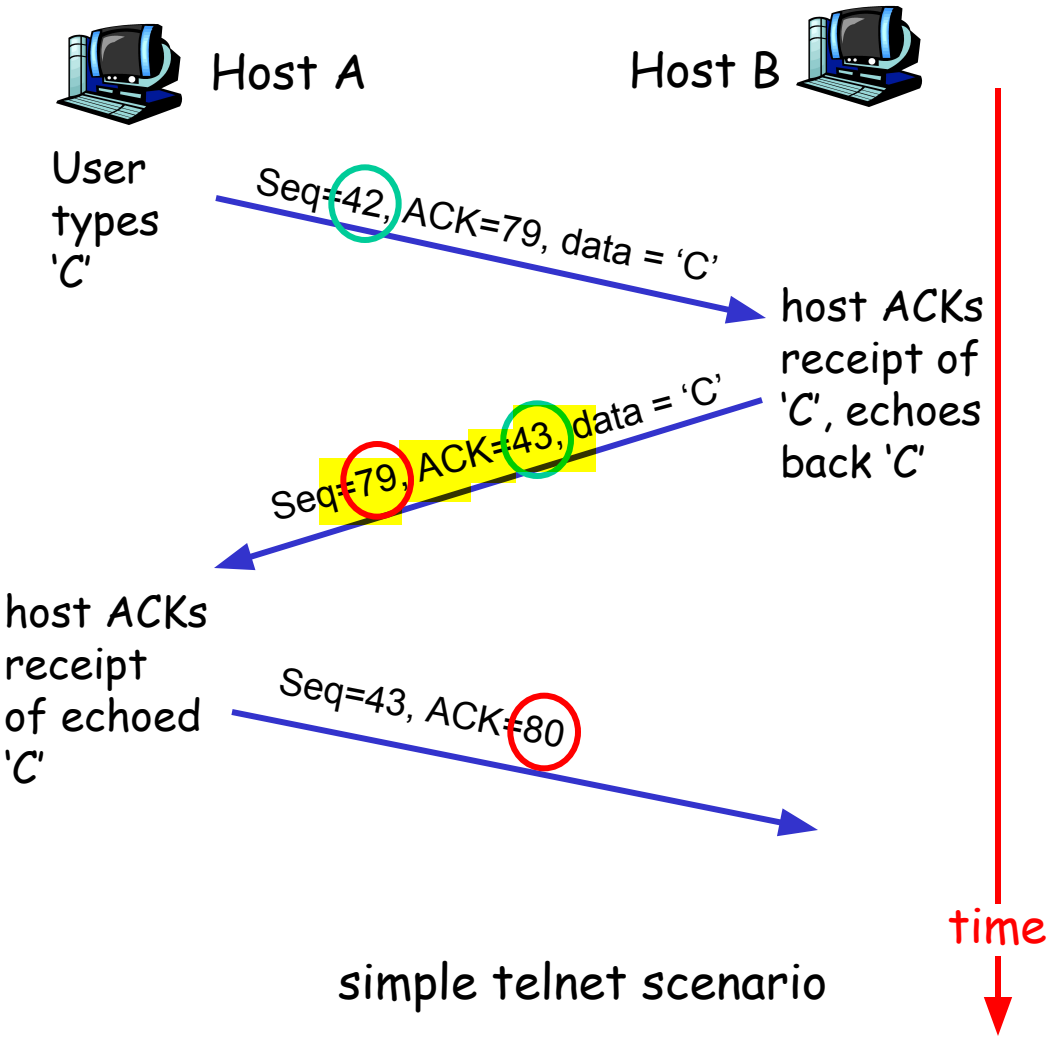
- byte stream "number" of first byte in segment's data

ACKs:

- seq # of next byte expected from other side
- cumulative ACK

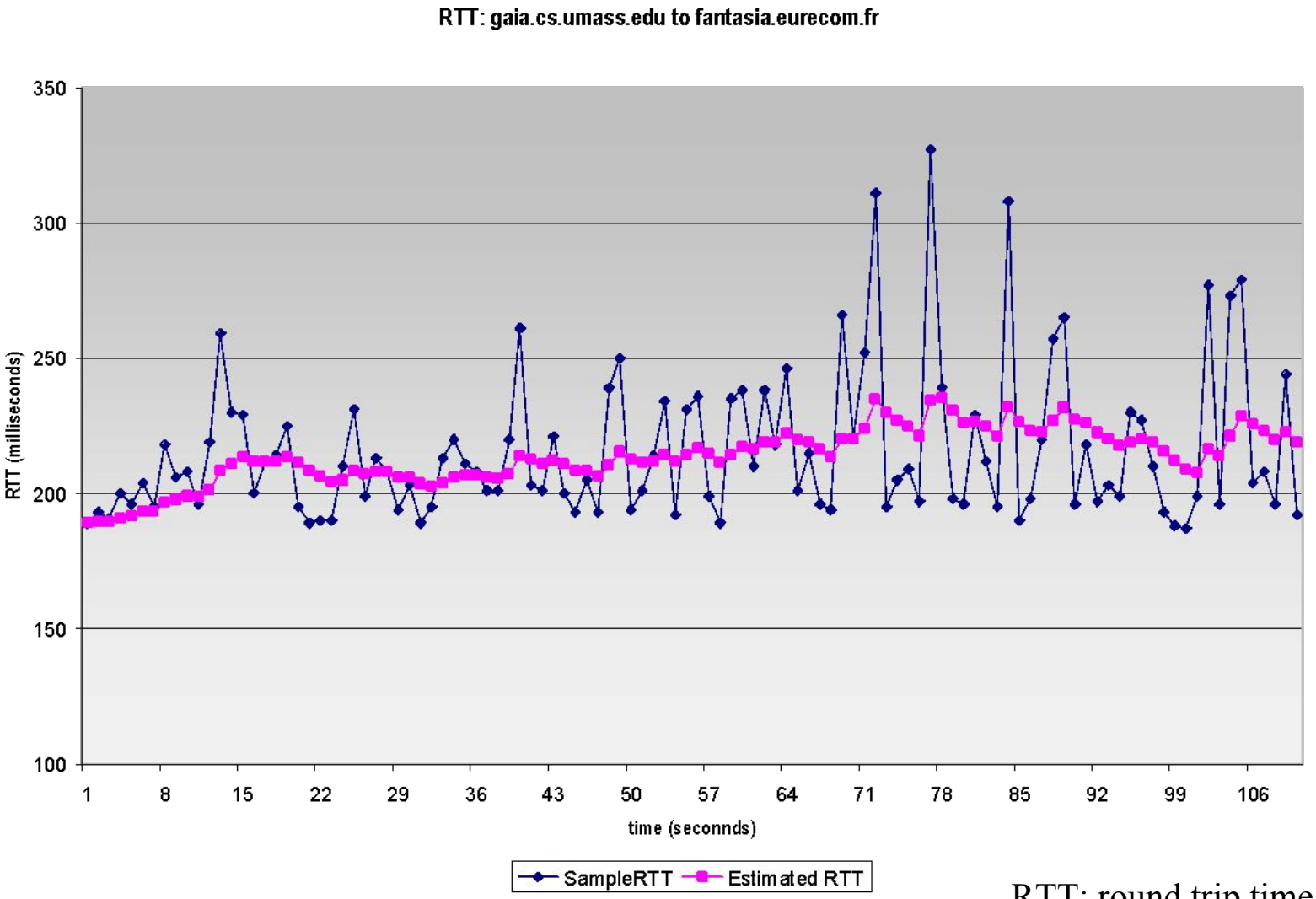
Q: how receiver handles out-of-order segments

- A: TCP spec doesn't say, - up to implementor



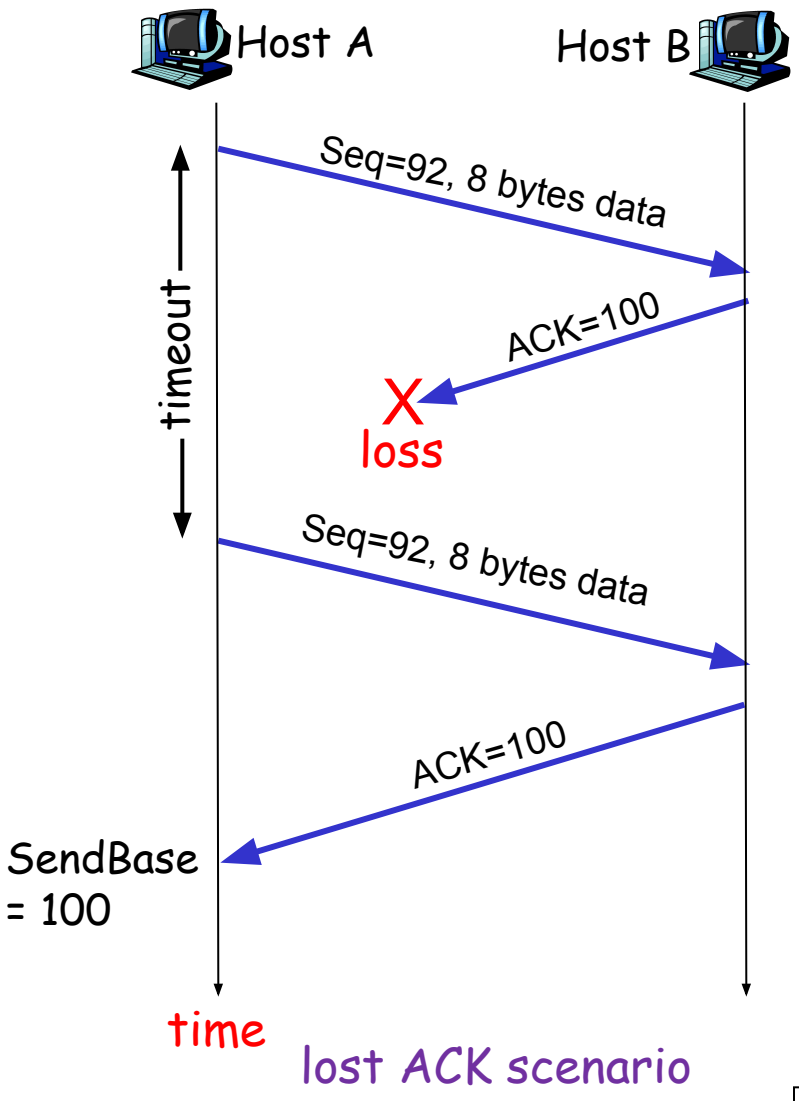
simple telnet scenario

Example RTT estimation:

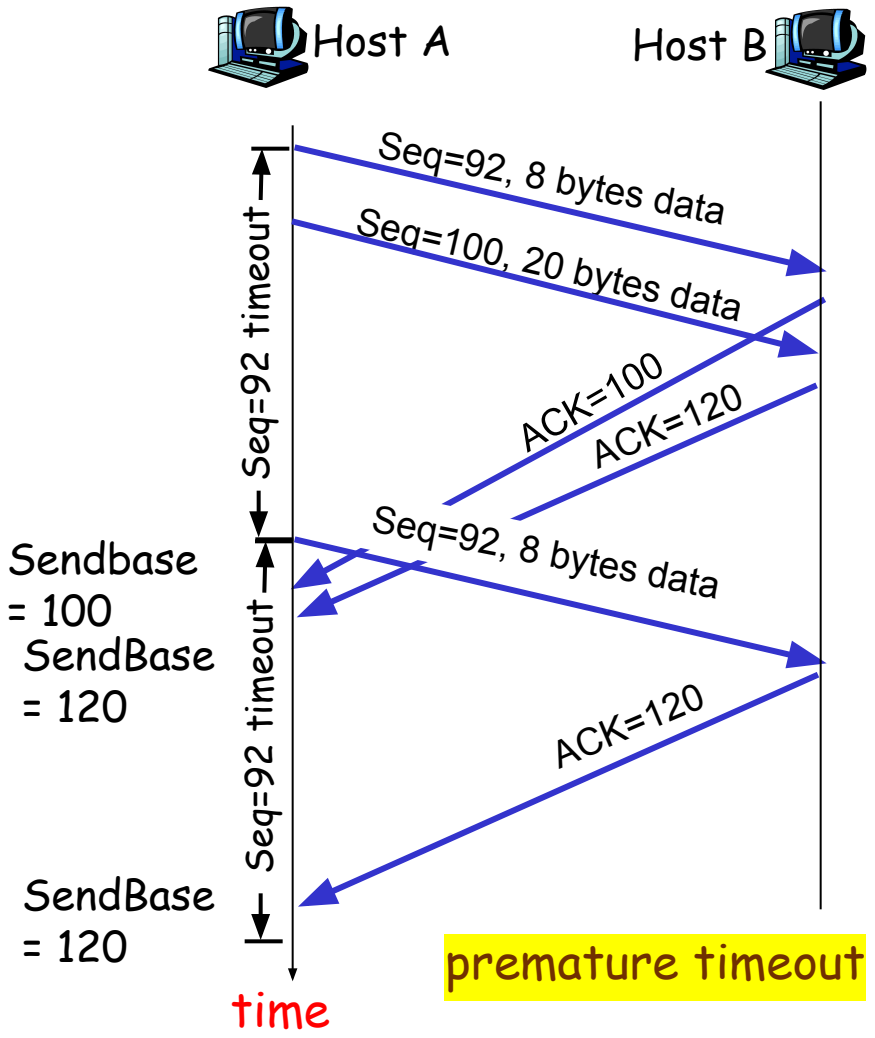


RTT: round trip time

TCP: retransmission scenarios



lost ACK scenario



premature timeout

*SendBase: Seq# of oldest unACK'd byte

Fast Retransmit

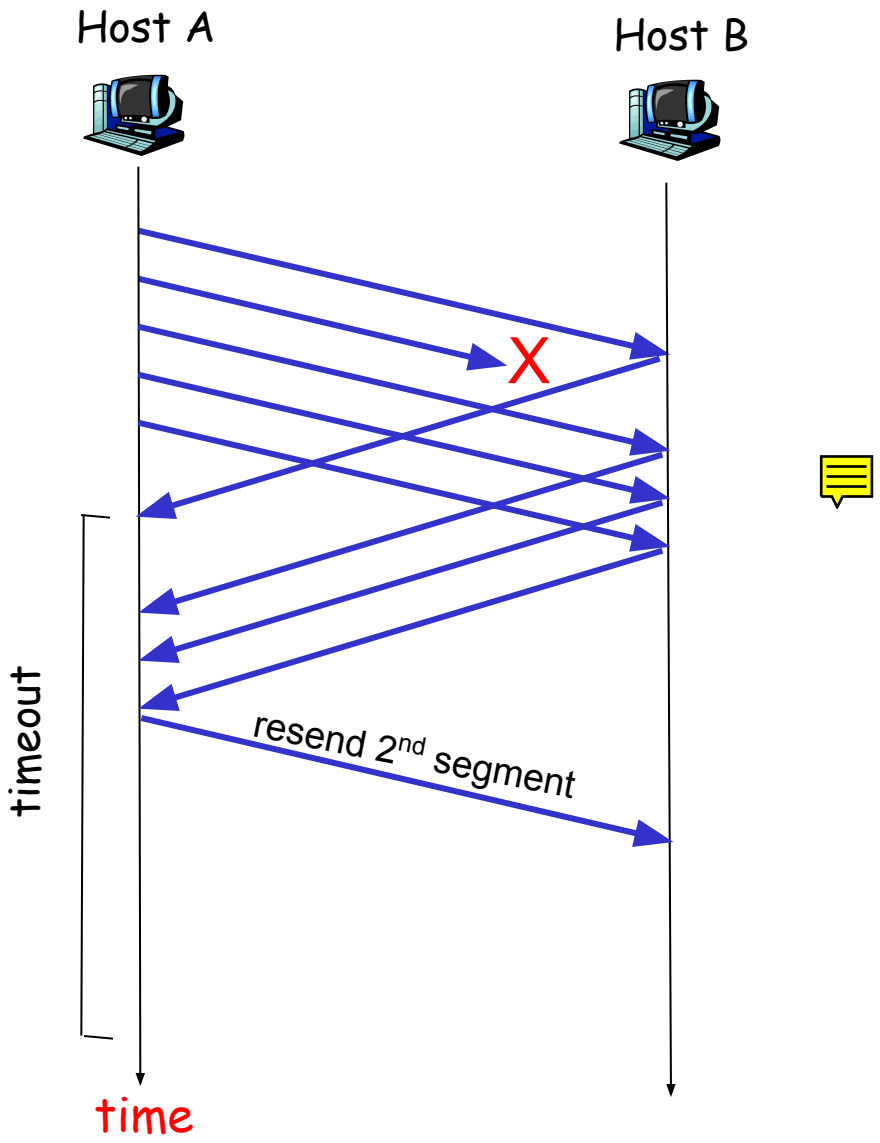


Figure 3.37 Resending a segment after triple duplicate ACK

TCP Connection Management

Recall: TCP sender, receiver establish "connection" before exchanging data segments

- initialize TCP variables:
  - seq. #s
  - buffers, flow control info (e.g. **RcvWindow**)
- *client:* connection initiator

```
Socket clientSocket = new
 Socket("hostname","port
 number");
```

- *server:* contacted by client

```
Socket connectionSocket =
 welcomeSocket.accept();
```

Three way handshake:

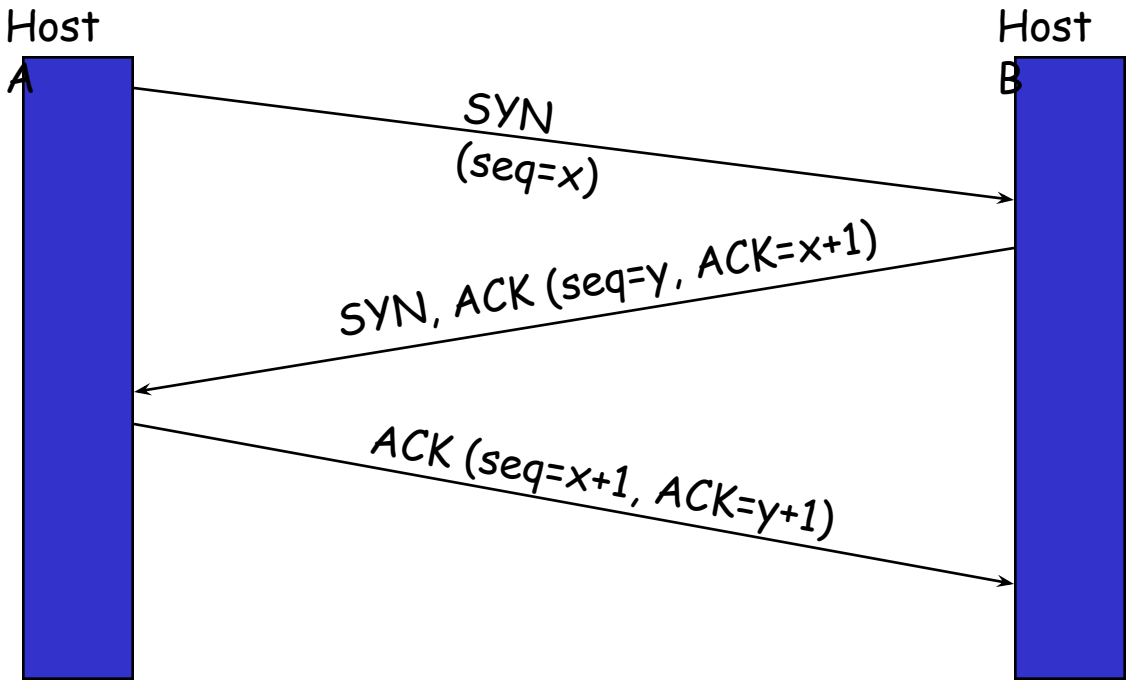
Step 1: client host sends TCP SYN segment to server
- specifies initial seq #
- no data

Step 2: server host receives SYN, replies with SYNACK segment
- server allocates buffers
- specifies server initial seq. #

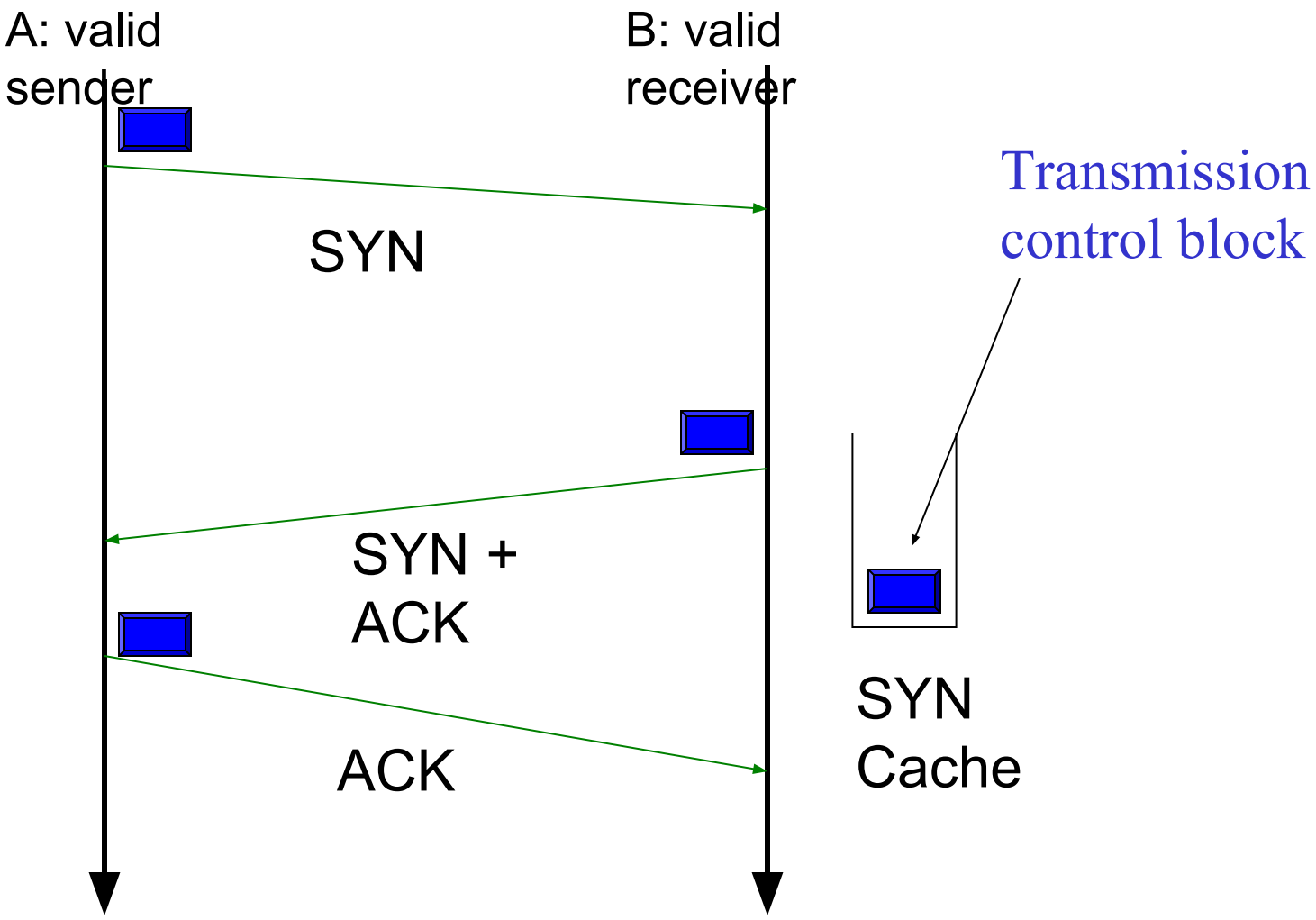
Step 3: client receives SYNACK, replies with ACK segment, which may contain data

TCP Connection Establishment

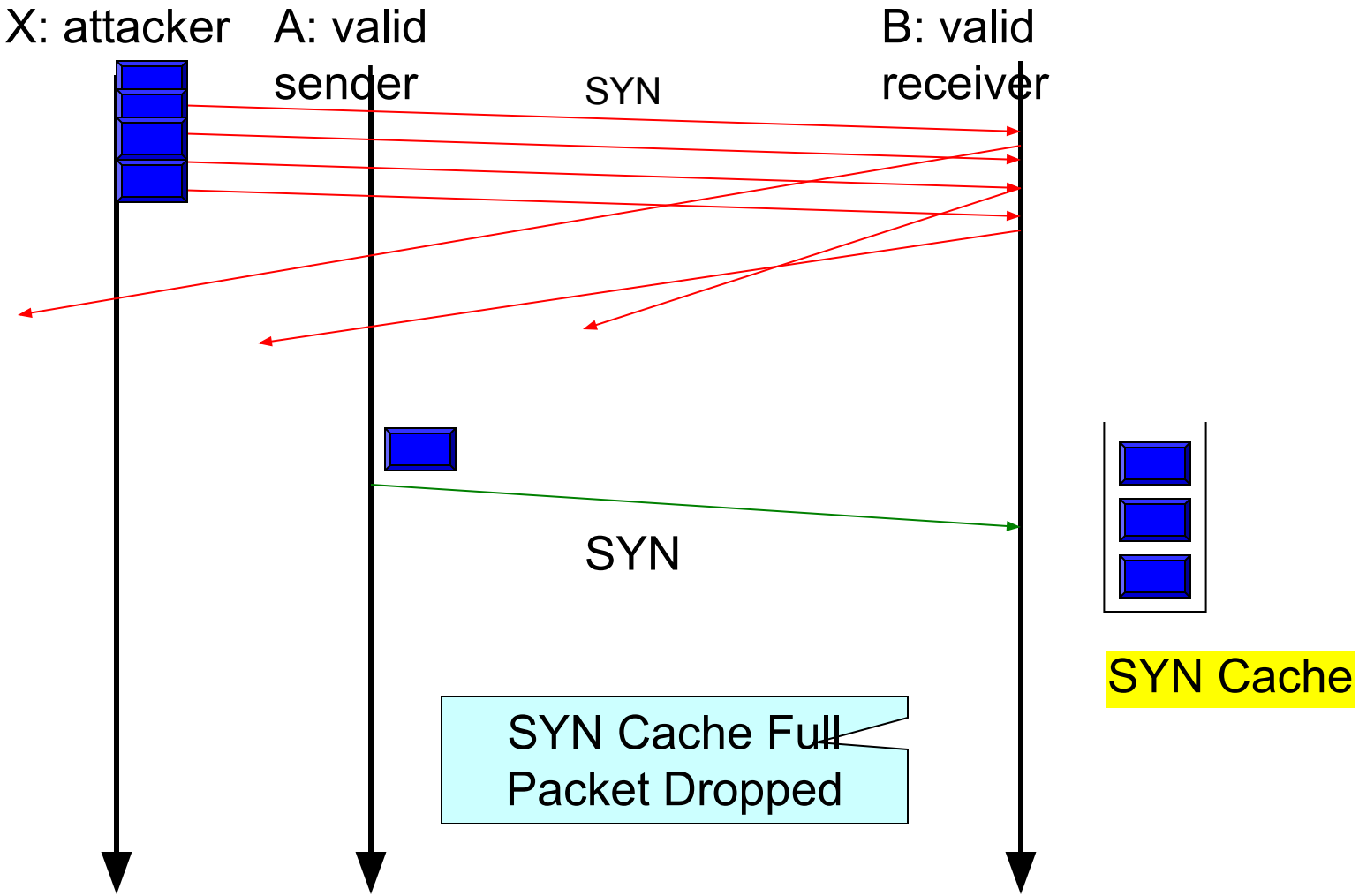
- Three-way Handshake



Denial of Service (DoS) attack



DOS attack!

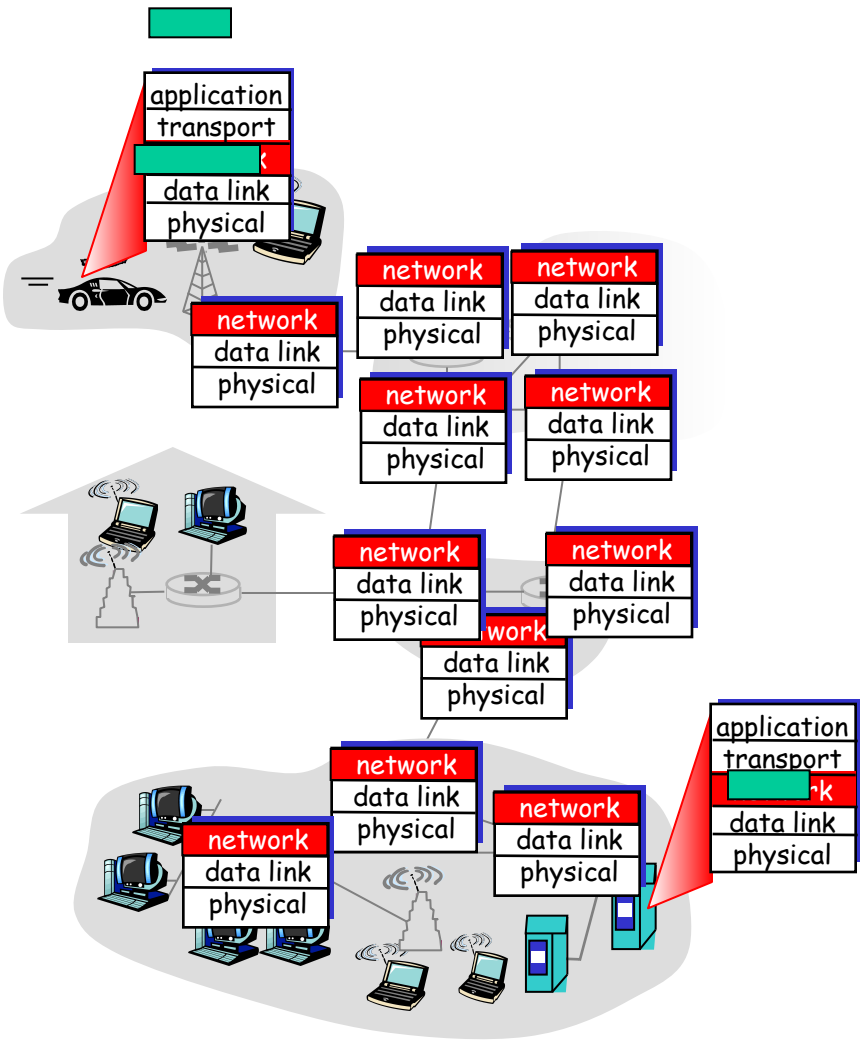

X: attacker
A: valid sender
B: valid receiver
SYN
SYN
SYN Cache
SYN Cache Full
Packet Dropped

outline

- Intro
- Appl
- TCP
- IP
- DNS
- ARP

Network layer

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



* message, segment, packet, datagram, frame

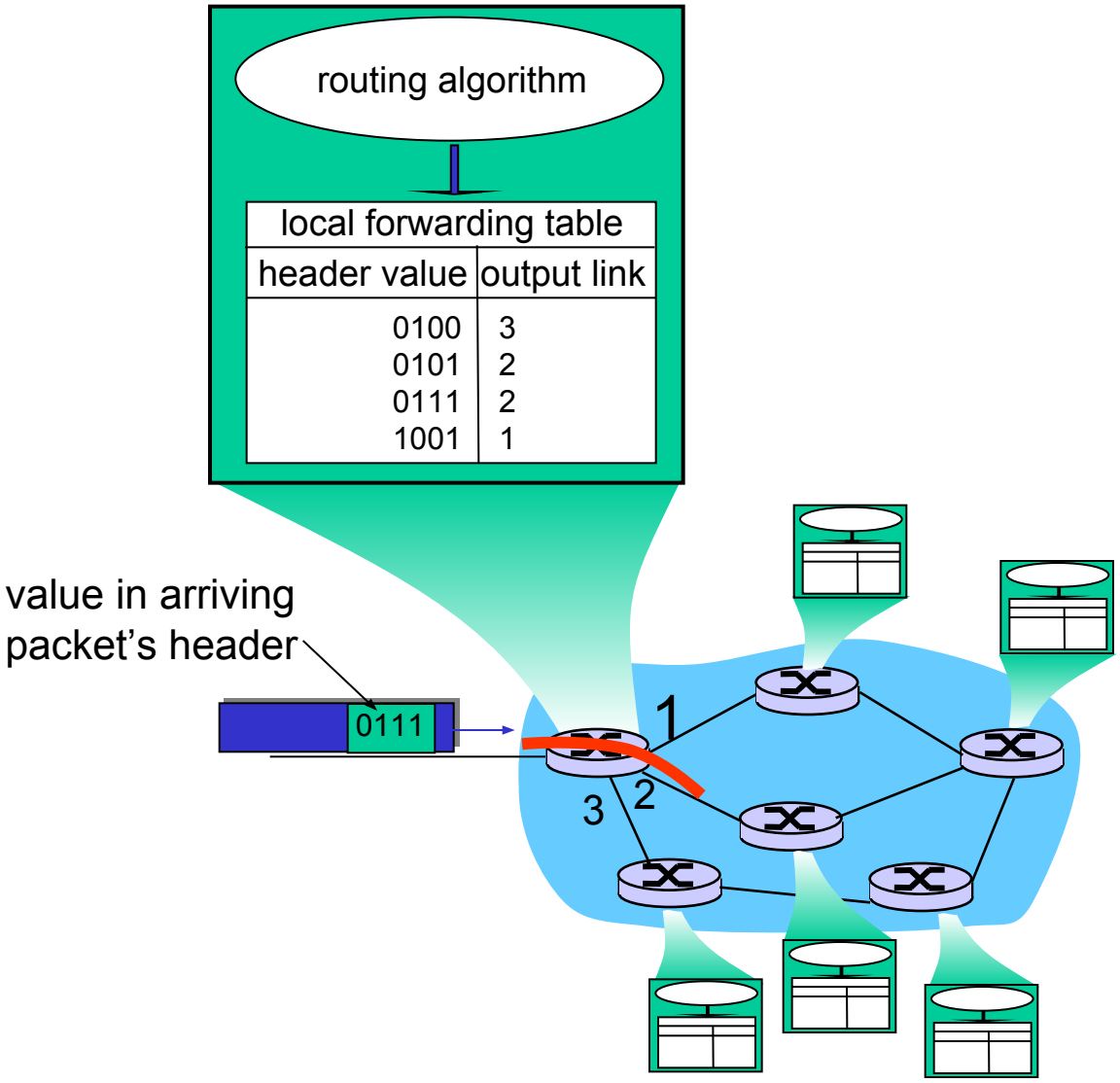
Two Key Network-Layer Functions

- *forwarding:* move packets from router's input to appropriate router output
- *routing:* determine route taken by packets from source to dest.
  - *routing algorithms*

analogy:

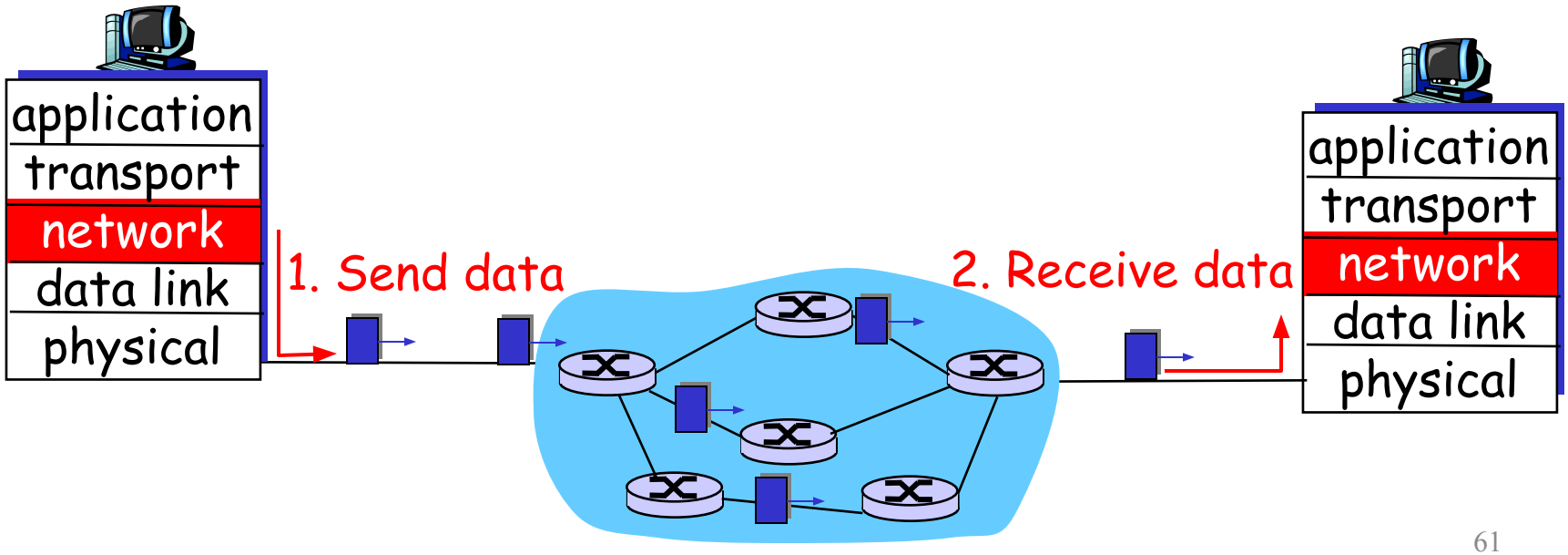
- routing: process of planning trip from source to dest
- forwarding: process of getting through single interchange

Interplay between routing and forwarding



Datagram networks

- no call setup at network layer
- routers: no state about end-to-end connections
  - no network-level concept of "connection"
- packets forwarded using destination host address
  - packets between same source-dest pair may take different paths

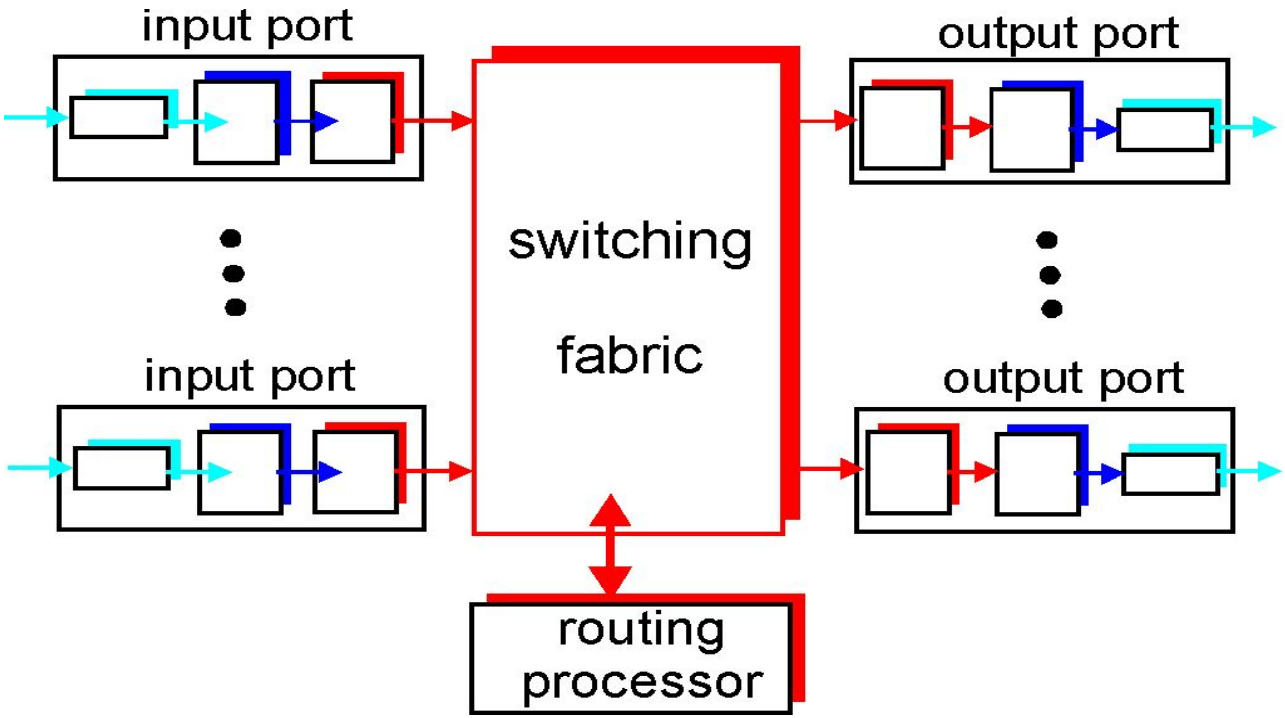


Routers process headers

Router Architecture Overview

Two key router functions:

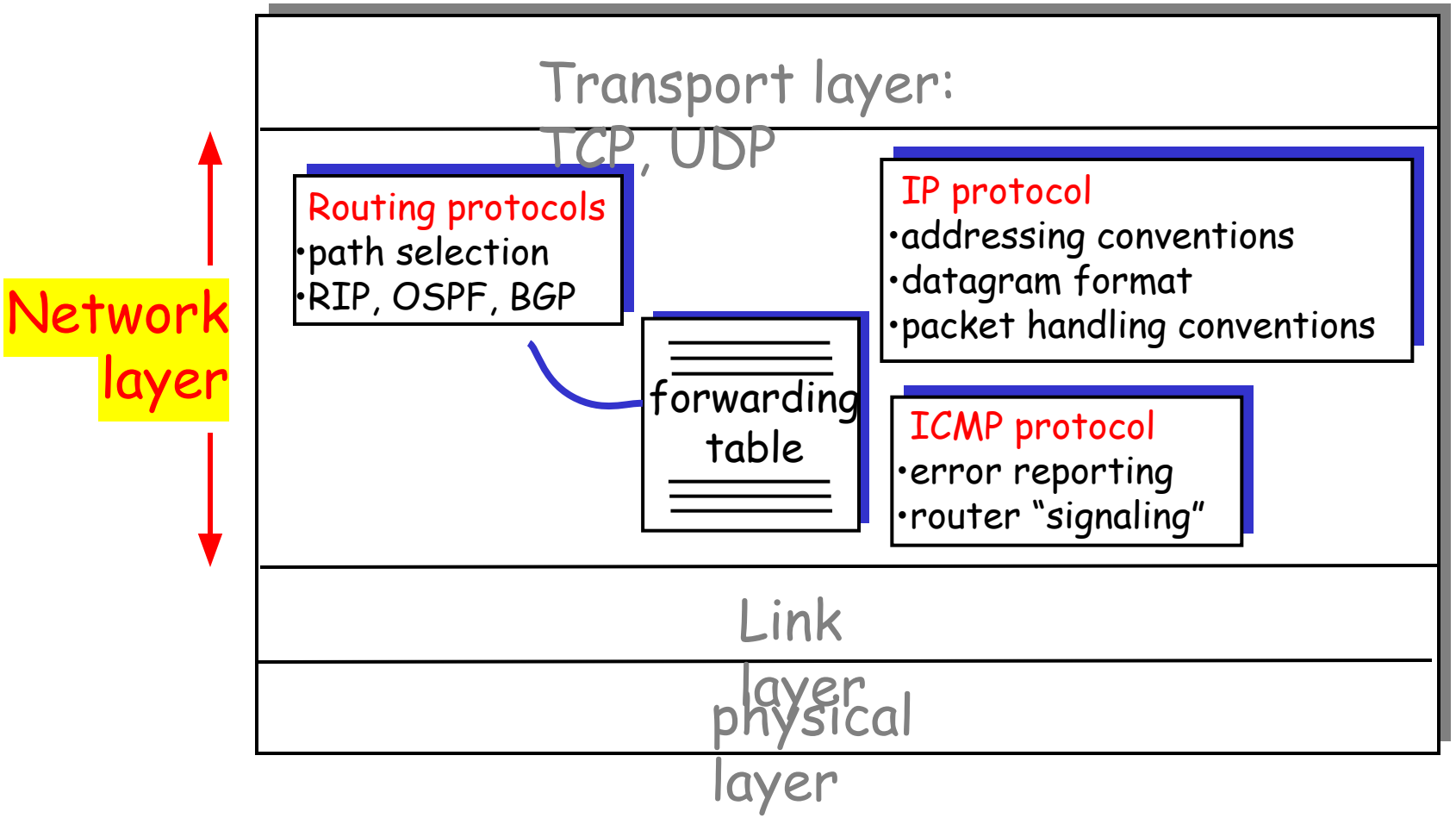
- run routing algorithms/protocol (RIP, OSPF, BGP)
- *forwarding* datagrams from incoming to outgoing link



- Queueing at input or output port -> RTT variation
- memory is limited!

The Internet (or IP) layer

Host, router network layer functions:

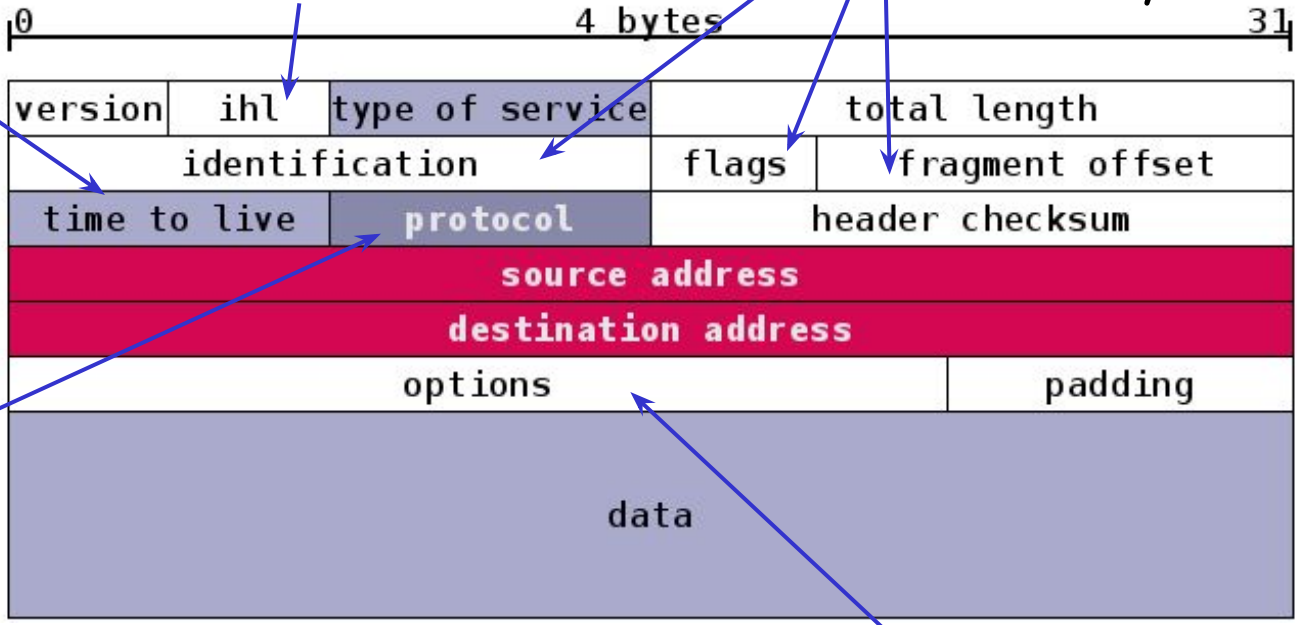


IP datagram format

max number remaining hops (decremented at each router)

IP header length

for fragmentation/ reassembly



upper layer protocol to deliver payload to

E.g. timestamp, record route taken, specify list of routers to visit.

how much overhead with TCP?

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

Network (IP) prefix and host identifier

- The network prefix identifies a network and the host identifier specifies a host (actually, interface on the network)

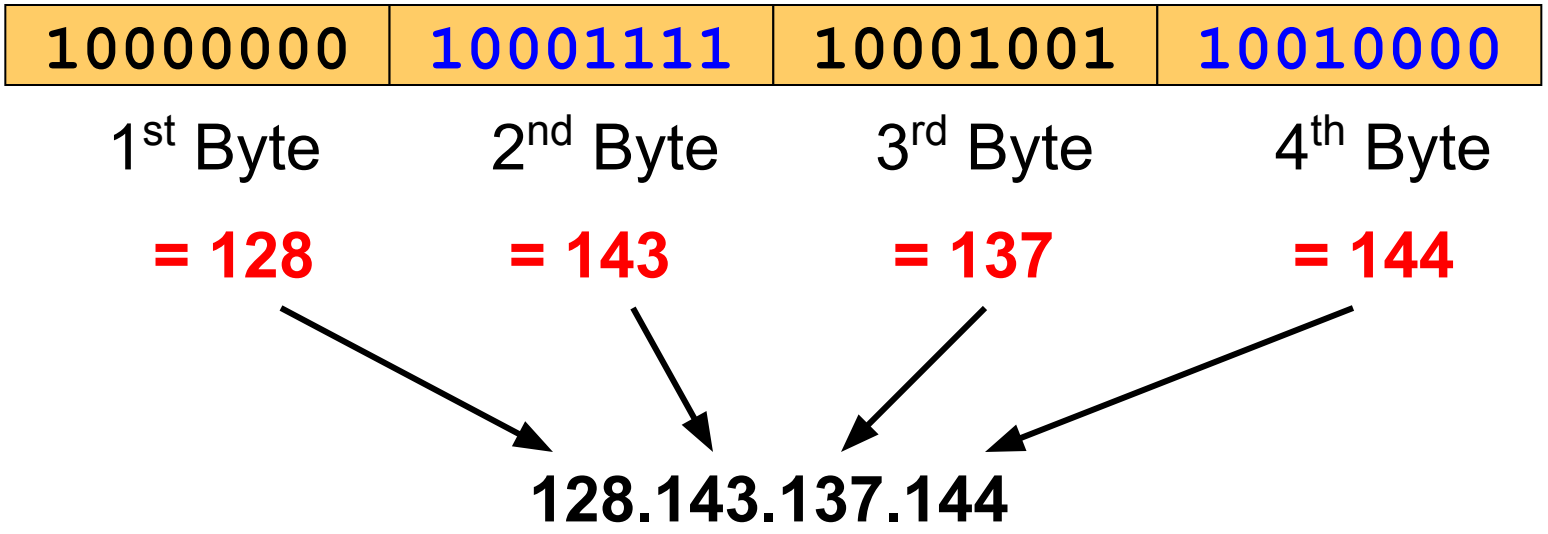
- How do we know how long the network prefix is?
  - Before 1993: The network prefix is implicitly defined: classful addressing

  or

  - After 1993: The network prefix is indicated by a netmask: classless inter-domain routing (CIDR)

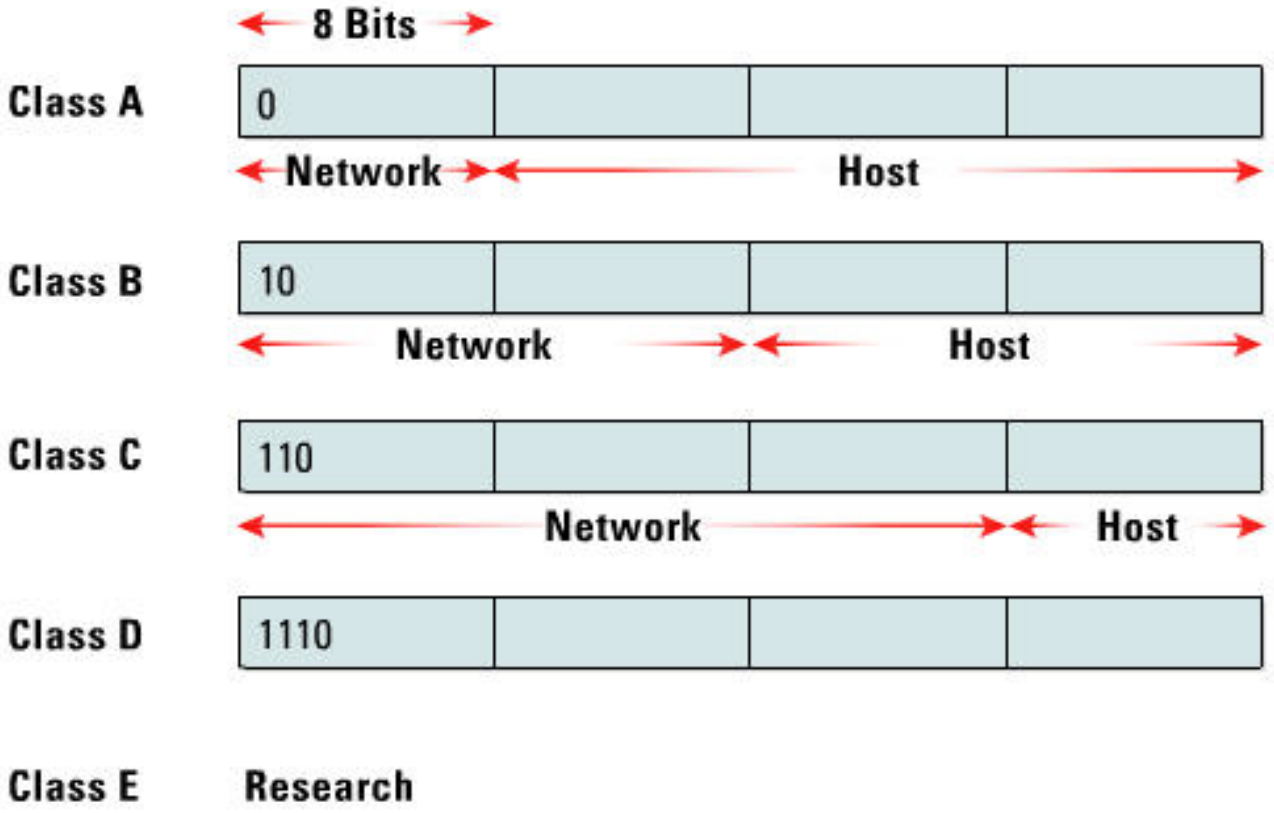
Dotted Decimal Notation

- IP addresses are written in a so-called *dotted decimal notation*
- Each byte is identified by a decimal number in the range [0..255]:
- Example:



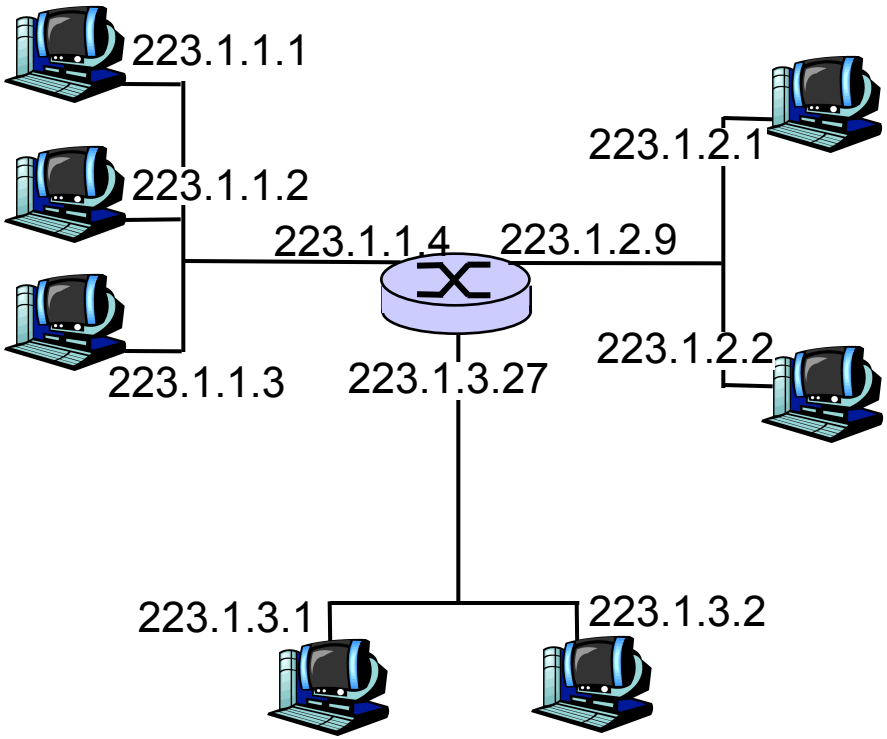
Classful addressing

- Classes A, B, C: three main address blocks
- Class D: multicast address
- Class E: reserved



IP Addressing: An Illustration

- 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 interface
  - IP addresses associated with each interface

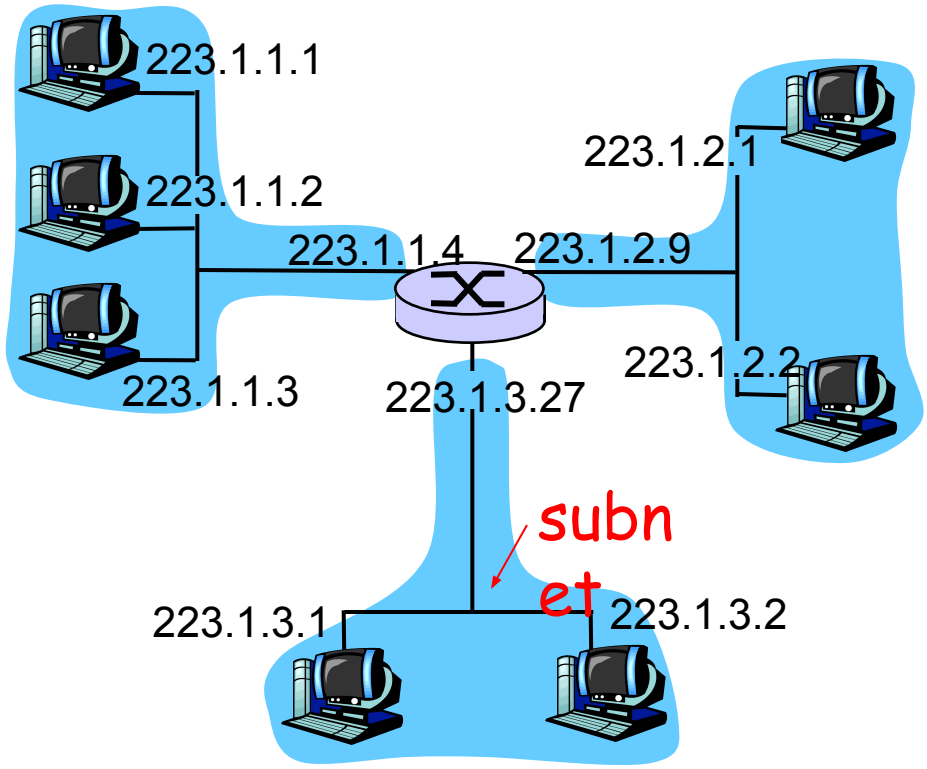


223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

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

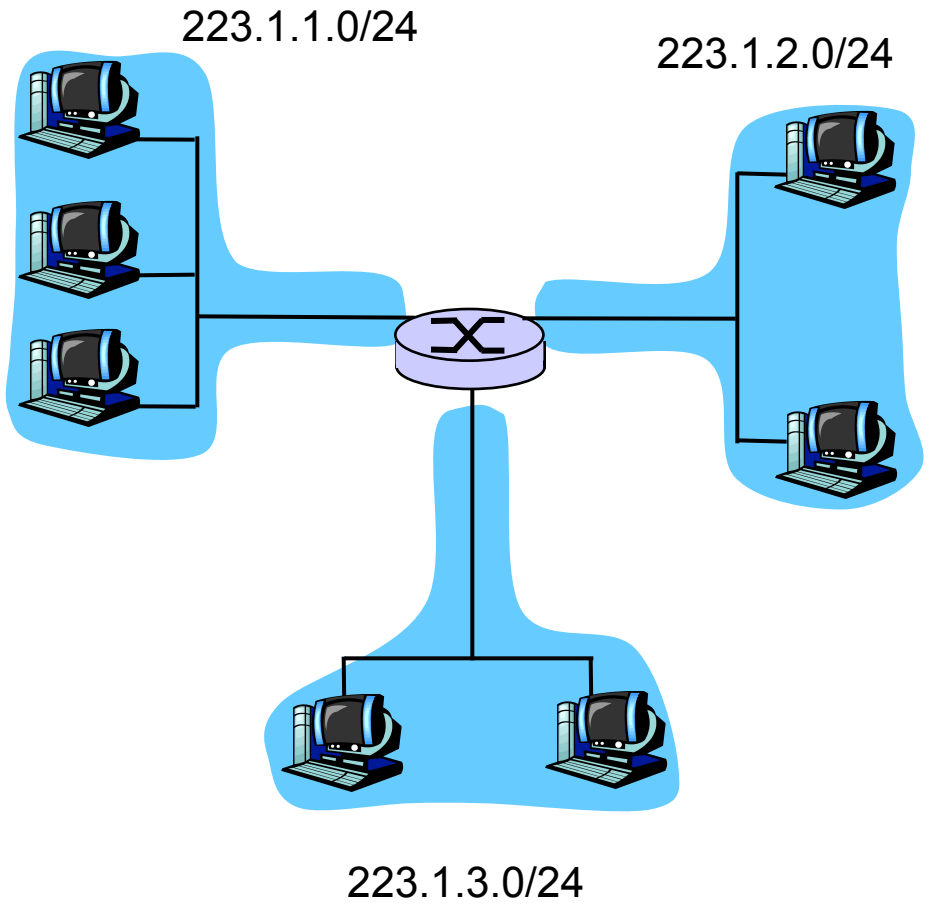


network consisting of 3 subnets

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.

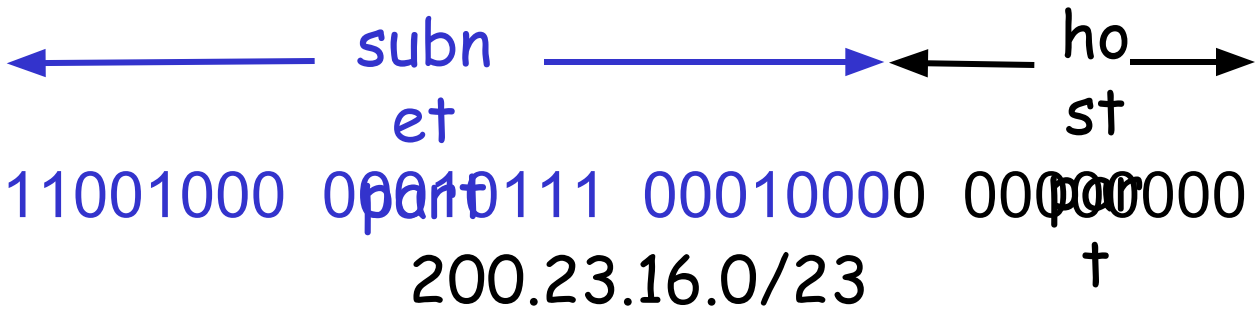


Subnet mask: /24

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

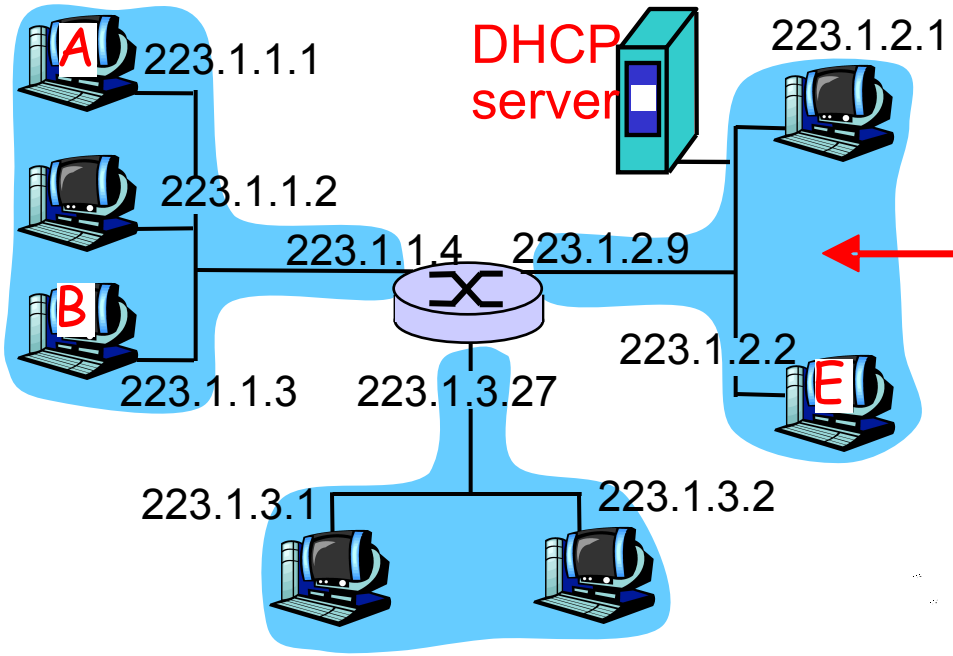


IP addresses: how to get one?

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 server
  - "plug-and-play"

DHCP client-server scenario



arriving DHCP client needs address in this network

IP addresses: how to get one?

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

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

ISP's block 200.23.16.0/20	11001000 00010111 00010000 00000000
Organization 0 200.23.16.0/23	11001000 00010111 00010000 00000000
Organization 1 200.23.18.0/23	11001000 00010111 00010010 00000000
Organization 2 200.23.20.0/23	11001000 00010111 00010100 00000000
...	

IP addressing: how to get?

Q: How does an ISP get a block of addresses?

A: ICANN: Internet Corporation for Assigned Names and Numbers

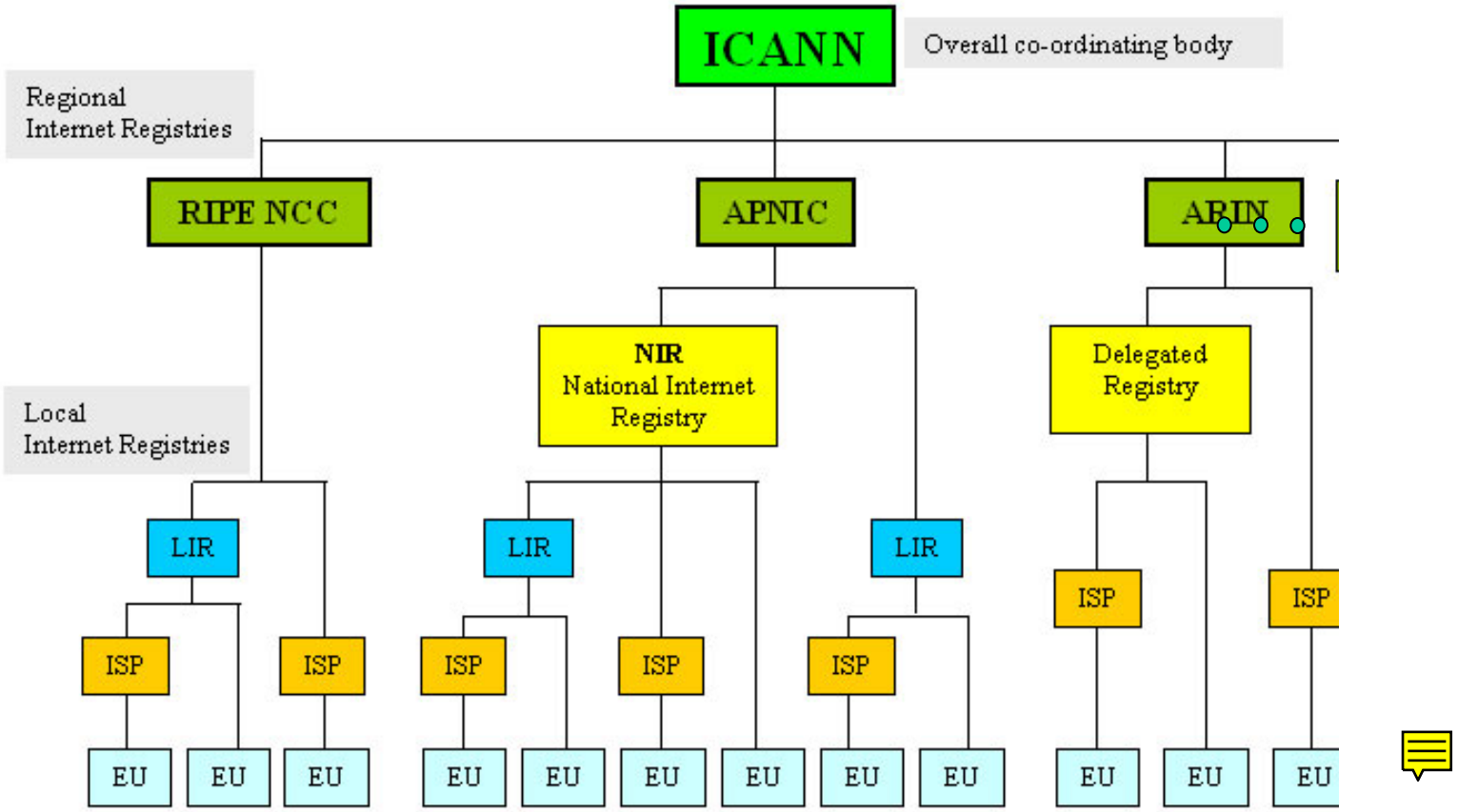
- allocates addresses
- manages DNS
- assigns domain names, resolves disputes

Q: in Korea?

A: KISA: Korea Internet and Security Agency

Internet organization

- Hierarchical management of number resources

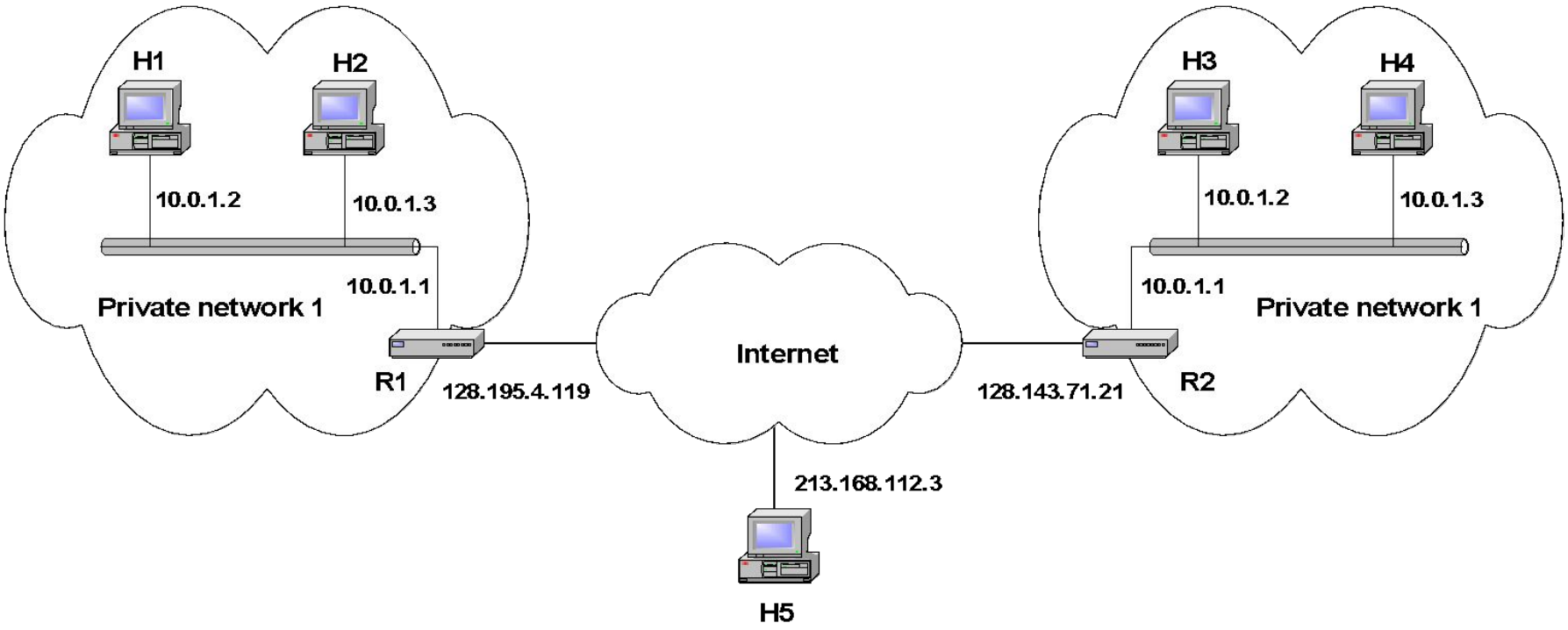


Source: APNIC

Private Network

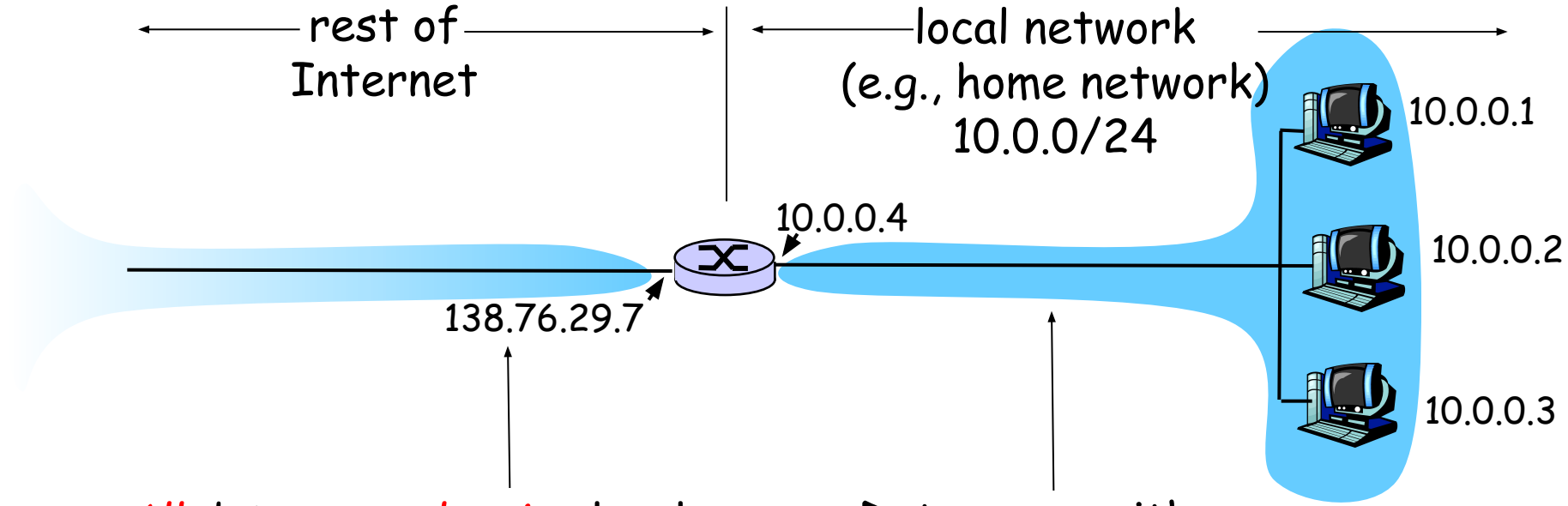
- *Private IP* network is an IP network that is not directly connected to the Internet
- IP addresses in a private network can be assigned arbitrarily.
  - Not registered and not guaranteed to be globally unique
- Generally, private networks use addresses from the following experimental address ranges (*non-routable addresses*):
  - 10.0.0.0 – 10.255.255.255
  - 172.16.0.0 – 172.31.255.255
  - 192.168.0.0 – 192.168.255.255

Private Addresses



How can a host with a private address can communicate with another host outside?

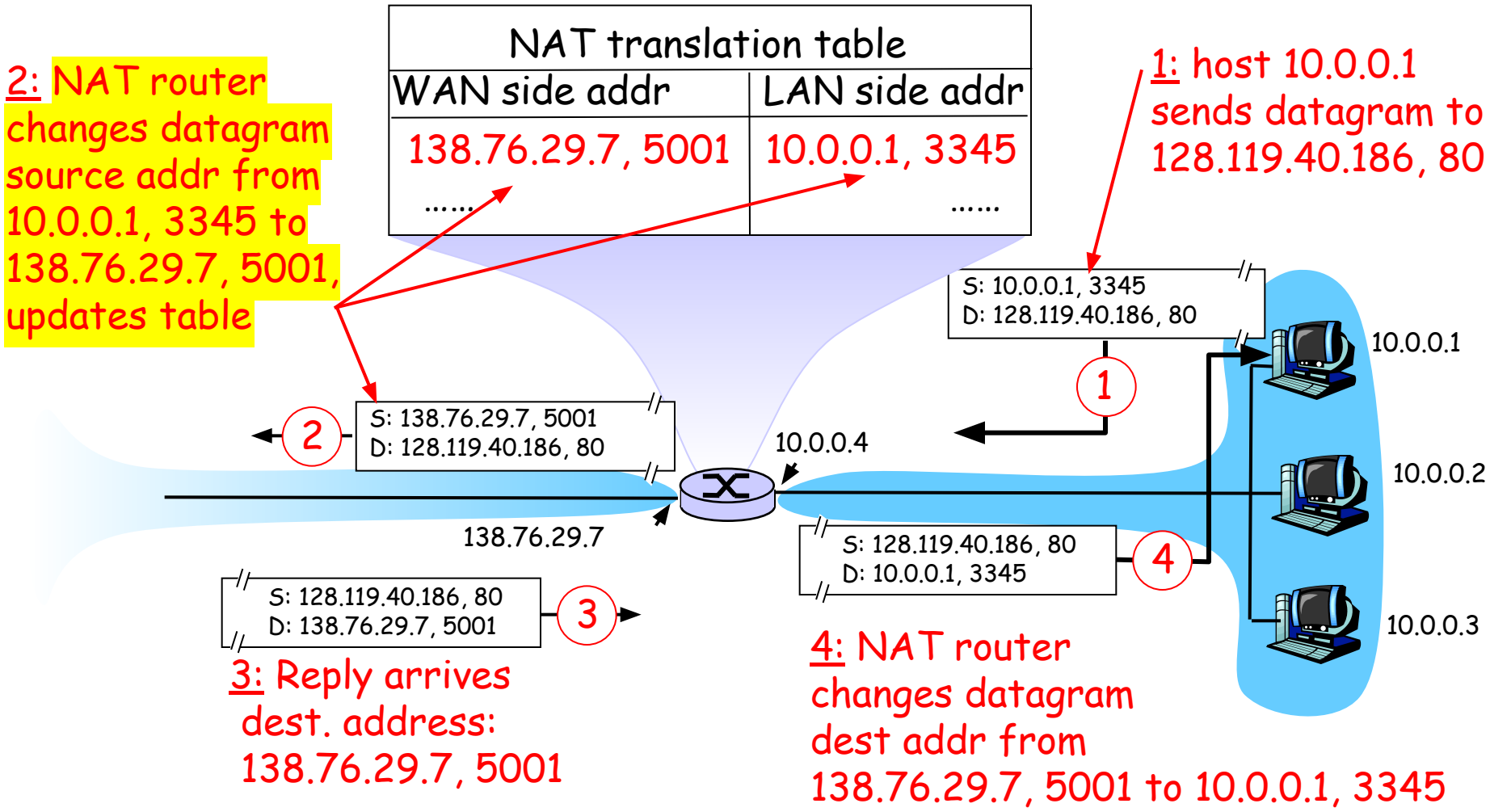
NAT: Network Address Translation



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

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

NAT: Network Address Translation



Aka network address and port translation (NAPT)

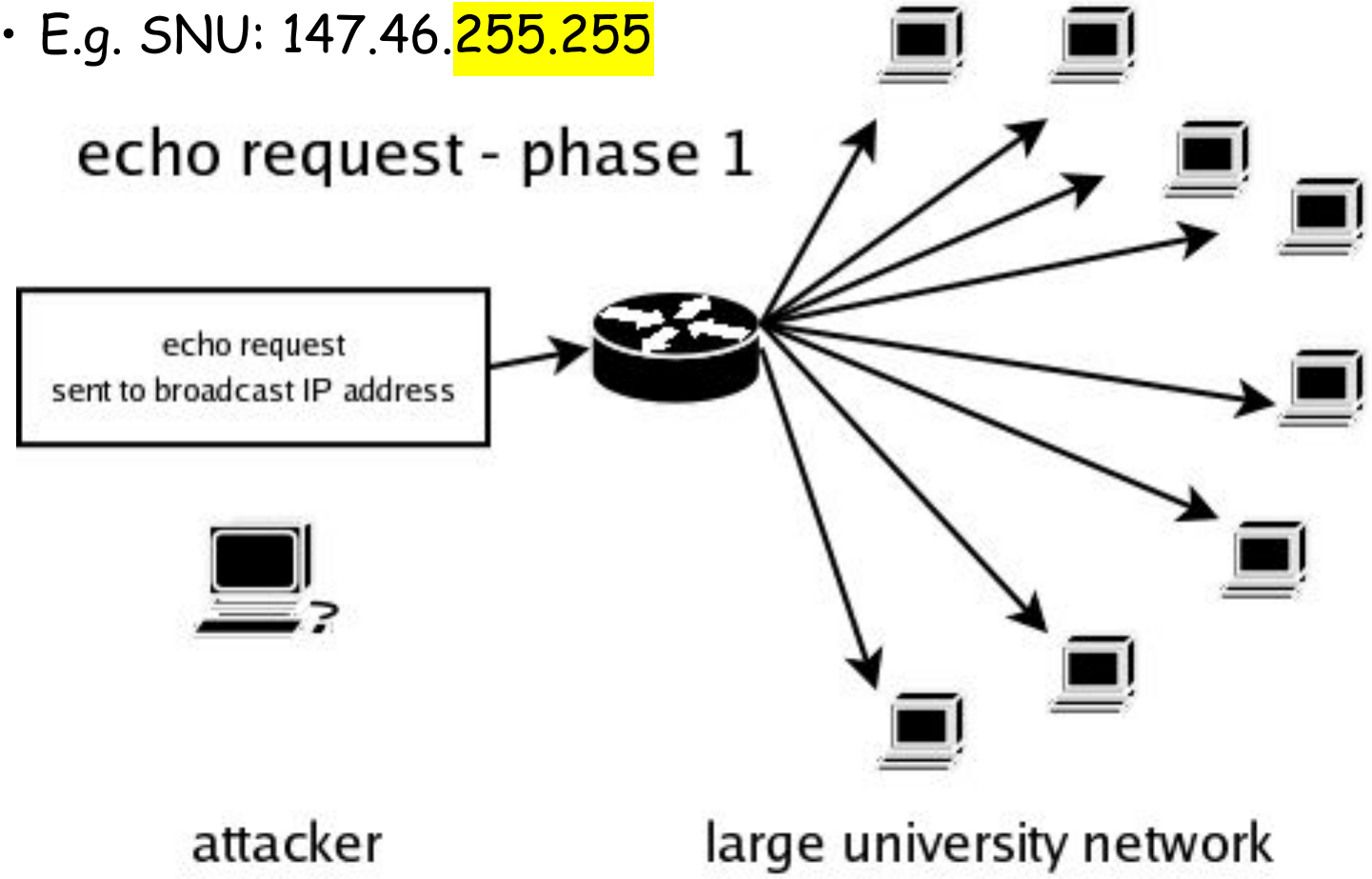
ICMP: Internet Control Message Protocol

- 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

Ping flooding attack

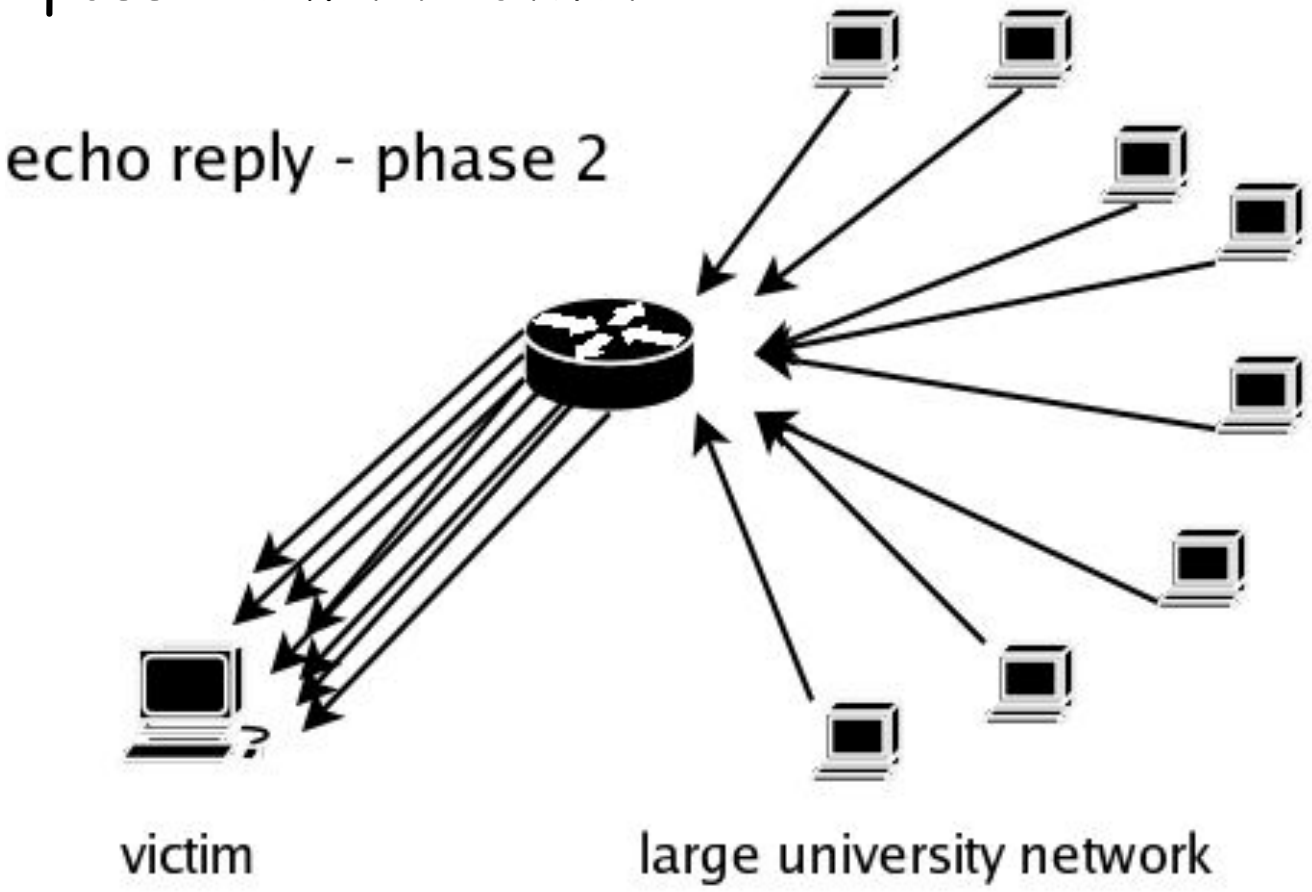
- IP broadcast address
  - Host id part is all 1s
    - E.g. SNU: 147.46.255.255



Source: tomicki.net

Ping flooding attack

- smurf attack, reflection attack
- Now routers forward no broadcast address
- Still possible with botnet



Source: tomicki.net

IPv6

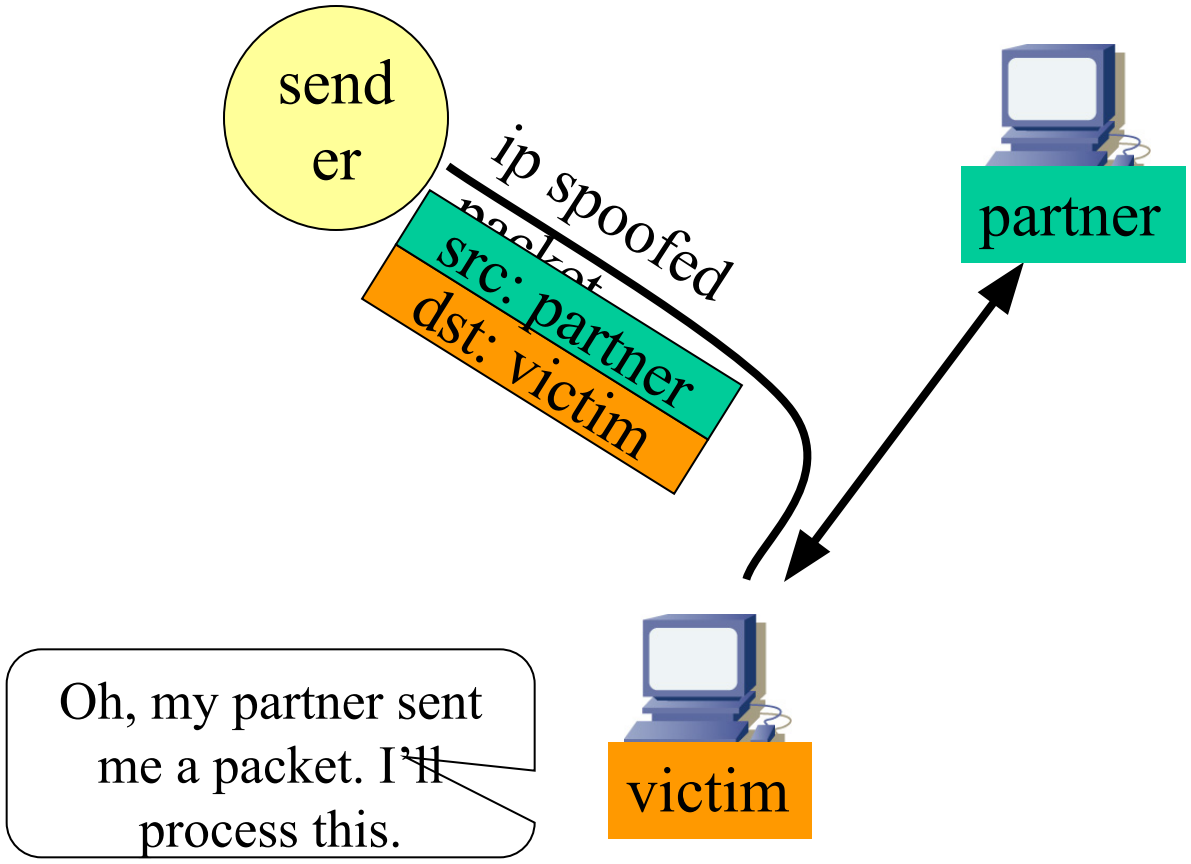
- Initial motivation: 32-bit address space has been completely allocated.
  - 128 bit address
- Additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

IPv6 datagram format:

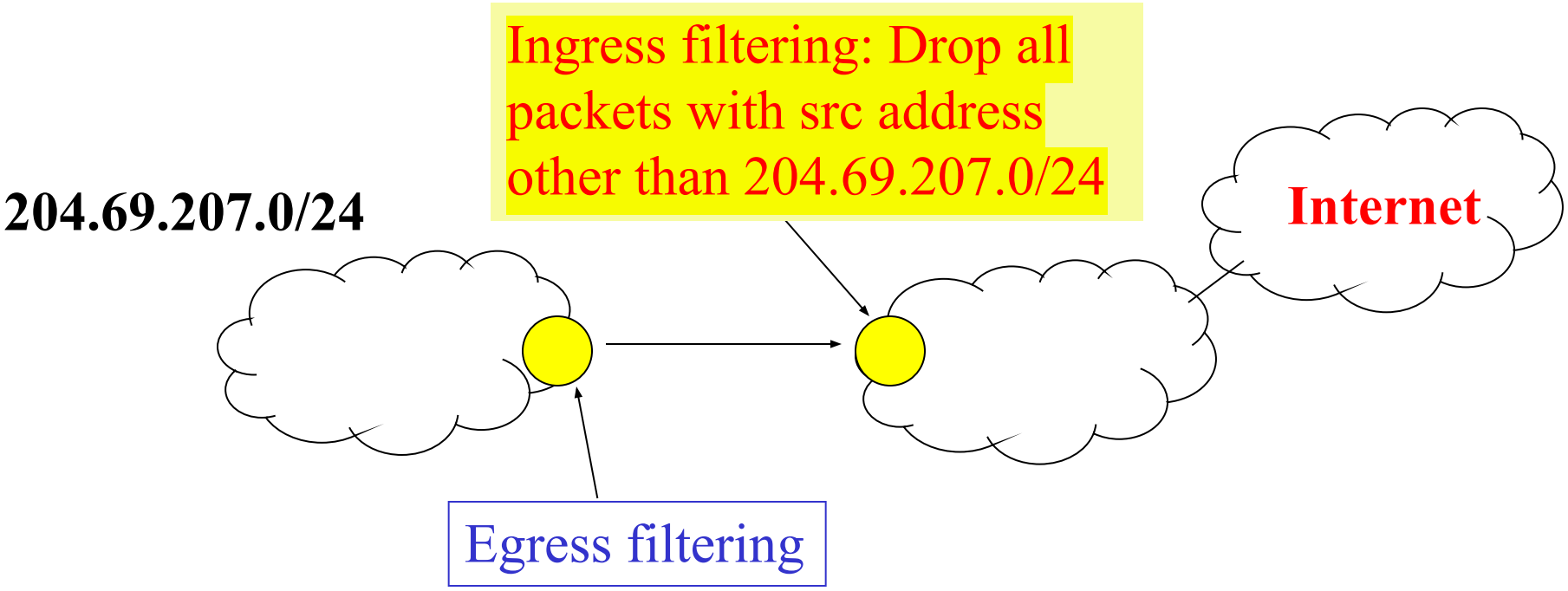
  - fixed-length 40 byte header
  - no fragmentation allowed

IP Spoofing Attacks:

impersonation

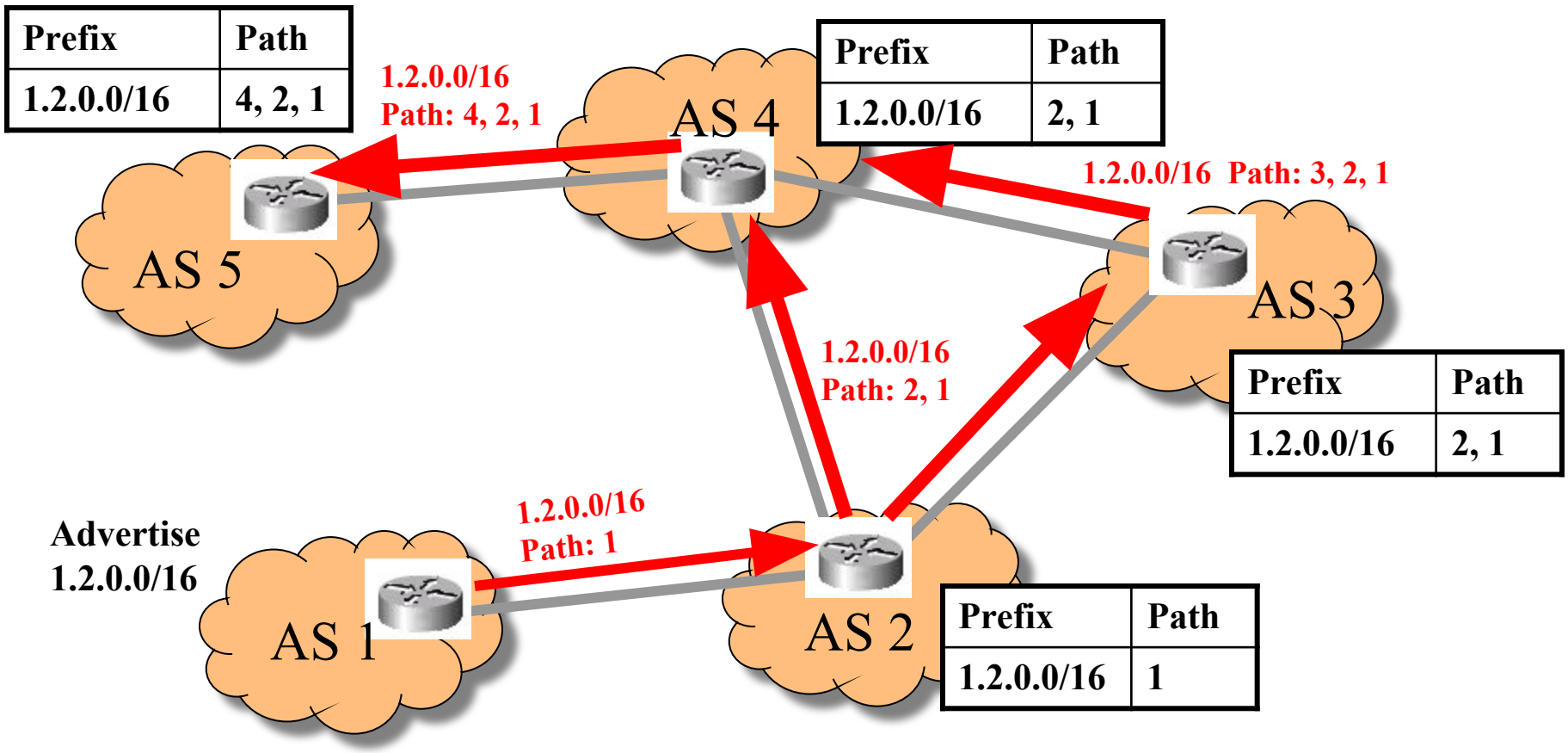


Ingress Filtering

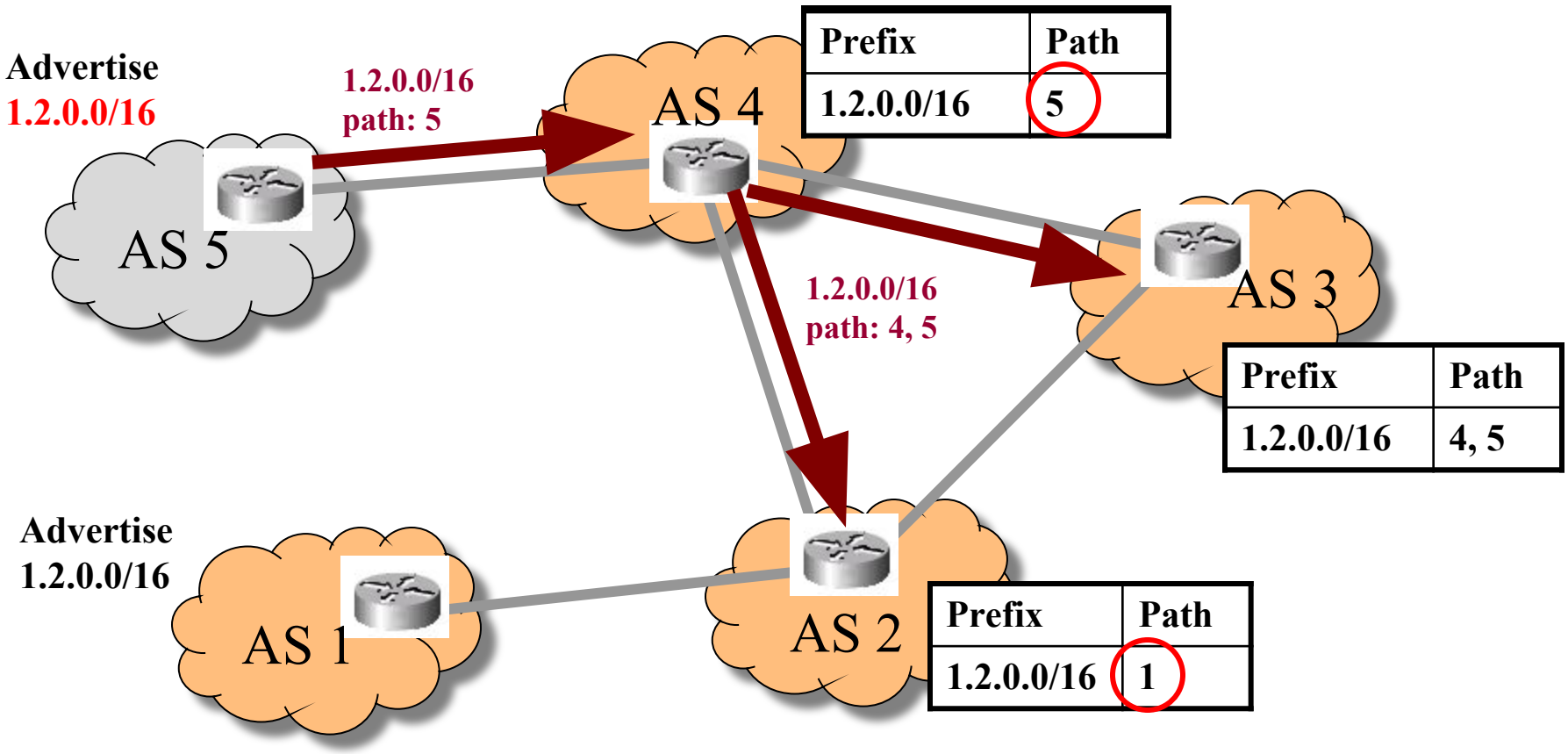


- Routers install filters to drop packets from networks that are not downstream
- Feasible at edges
- Difficult to configure closer to network "core"

IP prefix announcement (BGP)



IP prefix hijacking



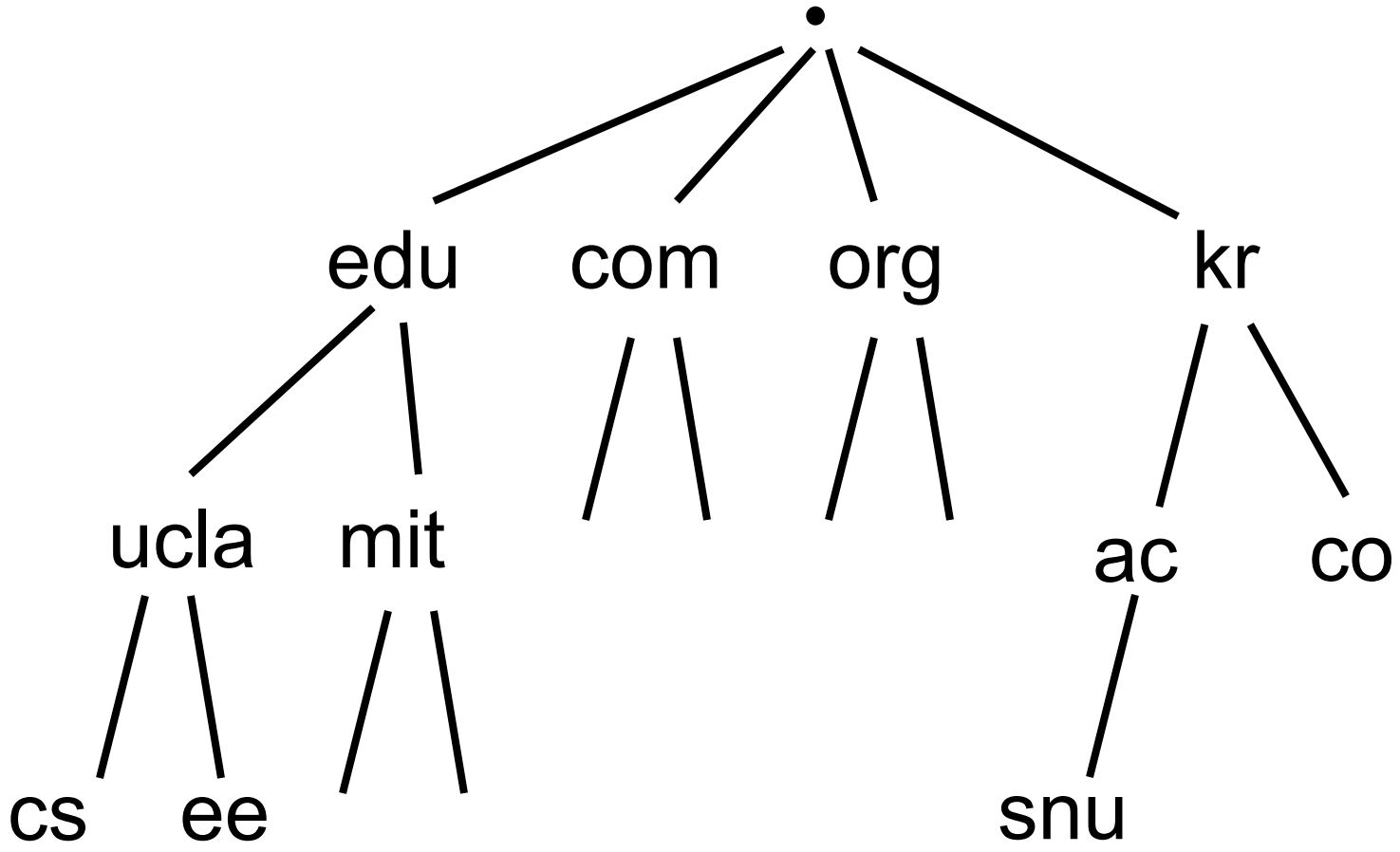
outline

- Intro
- Appl
- TCP
- IP
- DNS
- ARP

The Domain Name System

- The DNS is usually used to translate a host name into an IP address
  - E.g. www.yahoo.com 87.248.113.14
- Domain names comprise a hierarchy so that names are unique, yet easy to remember

DNS Hierarchy



Top level domains (TLDs)

- Generic TLD (gTLD)
  - edu, gov, com, net, org, mil, int,"arpa"
- Country code TLD (ccTLD)
  - 2 letter domain name in ISO 3166
    - .kr .uk .ca .au, …
- New top level domains include:

  .aero .biz .coop .info .name .pro .jobs .mobi

Address and Routing Parameter Area

DNS Organization

- Distributed Database
  - The organization that owns a domain name is responsible for running a DNS server (or nameserver) that can provide the mapping between hostnames within the domain to IP addresses.
  - E.g. some machine run by mit is responsible for everything within the mit.edu domain.

Concept: DNS Names

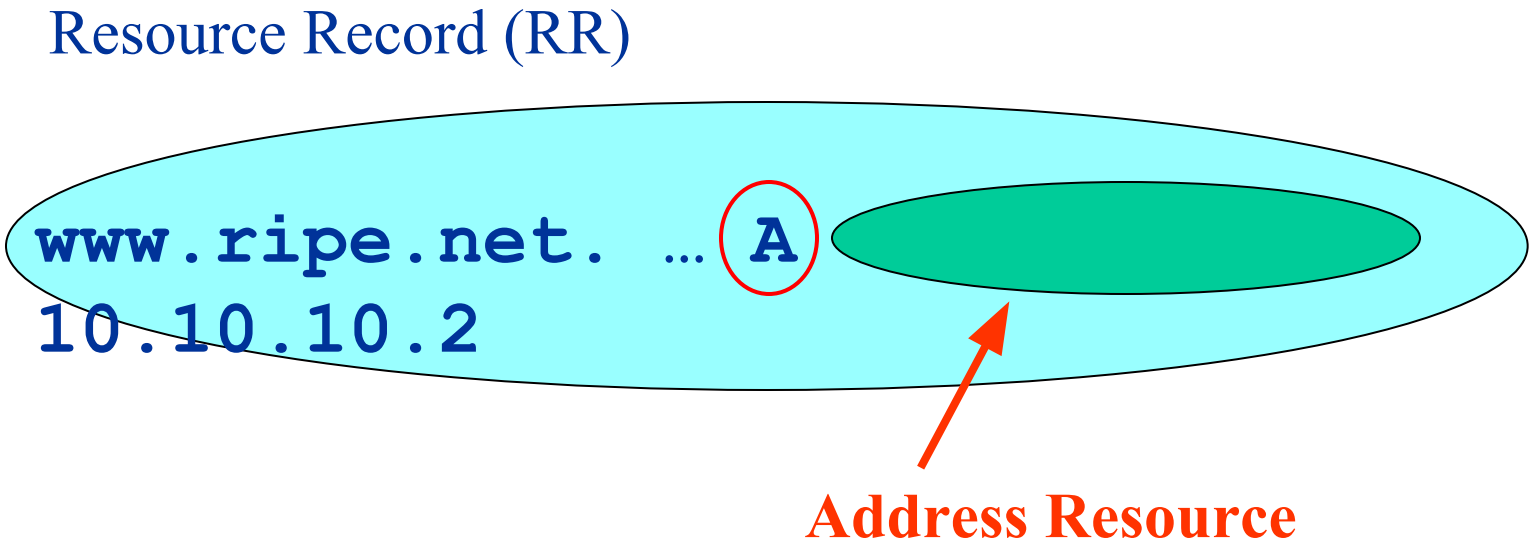
Fully Qualified Domain Name (FQDN)

WWW.RIPE.NET.

- labels separated by dots
- DNS provides a mapping from FQDNs to resources of several types
- Domain names are used as a key when looking up the resource records (RRs)

Concept: Resource Records

- The DNS maps names into data using Resource Records.



DNS RRs

DNS: distributed db storing resource records (RRs)

RR format: ***(name, value, type, ttl, class=IN)***

- *Type=A*
  - ***name*** *is hostname*
  - ***value*** *is IP address*
  - *AAAA type for IPv6*
- *Type=NS*
  - ***name*** *is domain (eg., foo.com)*
  - ***value*** *is hostname of authoritative name server for this domain*
- *Type=CNAME*
  - ***name*** *is alias name for some "canonical" (the real) name, e.g., www.ibm.com is really servereast.backup2.ibm.com*
  - ***value*** *is canonical name*
- *Type=MX*
  - ***value*** *is name of mailserver associated with* ***name***

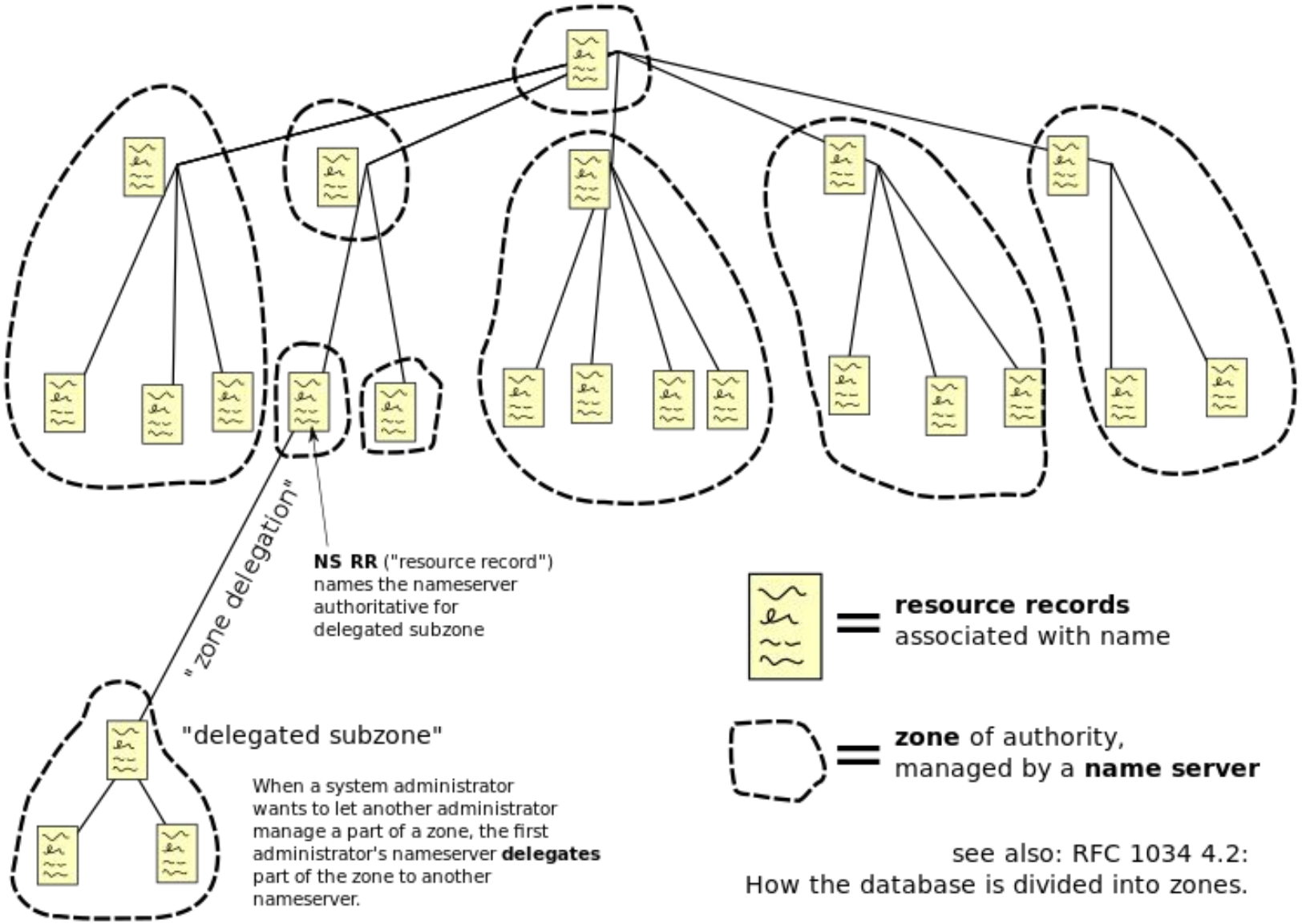
DNS: recursive server

- Recursive servers do the actual lookups; they ask questions to the DNS on behalf of the clients.
- Answers are obtained from authoritative servers
  - then answers forwarded to the clients
- Answers are stored for future reference in the cache
  - If the target domain name is in cache
    - It replies directly
    - Non-authoritative answer

Domain Name Space

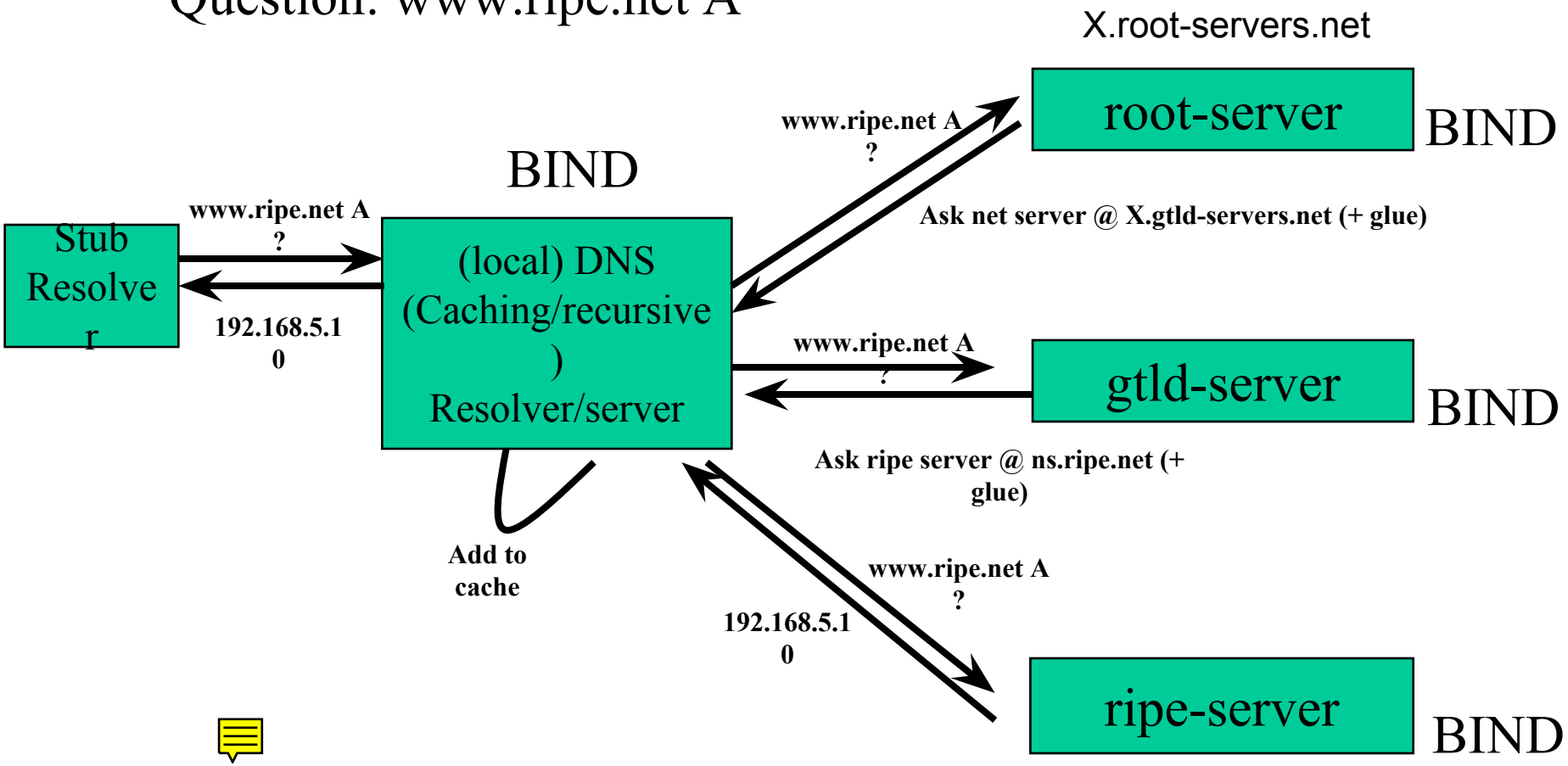
- Domain: A node in the DNS tree
- DNS Zones
  - A zone is a group of nodes in the tree, authoritatively served by an *authoritative nameserver*.
  - Each zone may be sub-divided, or delegated
- Authoritative servers
  - Answer queries about their zones
  - Provide mapping for leaf nodes or downward delegation
- Hierarchical service
  - *Root name servers* for top-level domains
  - *Authoritative name servers* for subdomains

DNS zone illustration



Resolving process & Cache

Question: www.ripe.net A

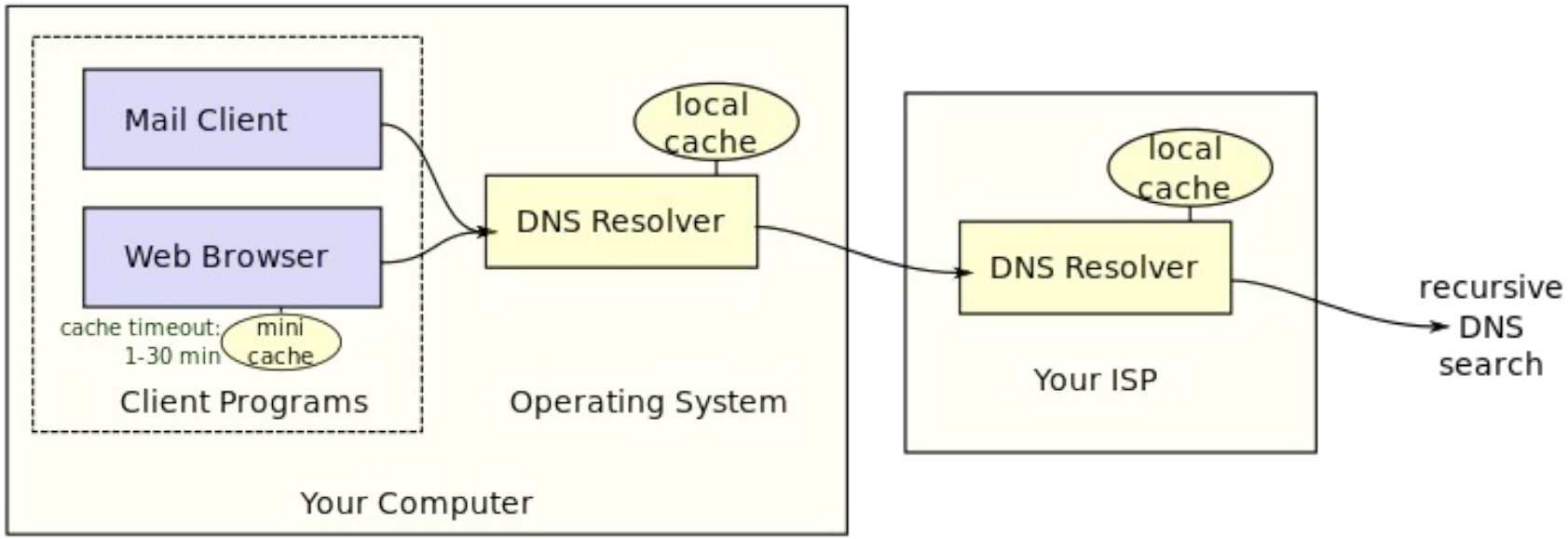


Glue record: IP address of the nameserver

X: [a..m]

DNS Resolver: Stub Resolver

- Stub resolver
  - Not interact with the zone hierarchy
  - Pose basic queries to recursive servers
  - May cache answers
  - PC, client applications



Caching

- DNS responses are cached
  - Quick response for repeated translations
  - Useful for finding servers as well as addresses
    - NS records for domains
- Negative results are also cached
  - Save time for nonexistent sites, e.g. misspelling
- Cached data periodically times out
  - Each record has a TTL field

Alias and multiple IP addresses

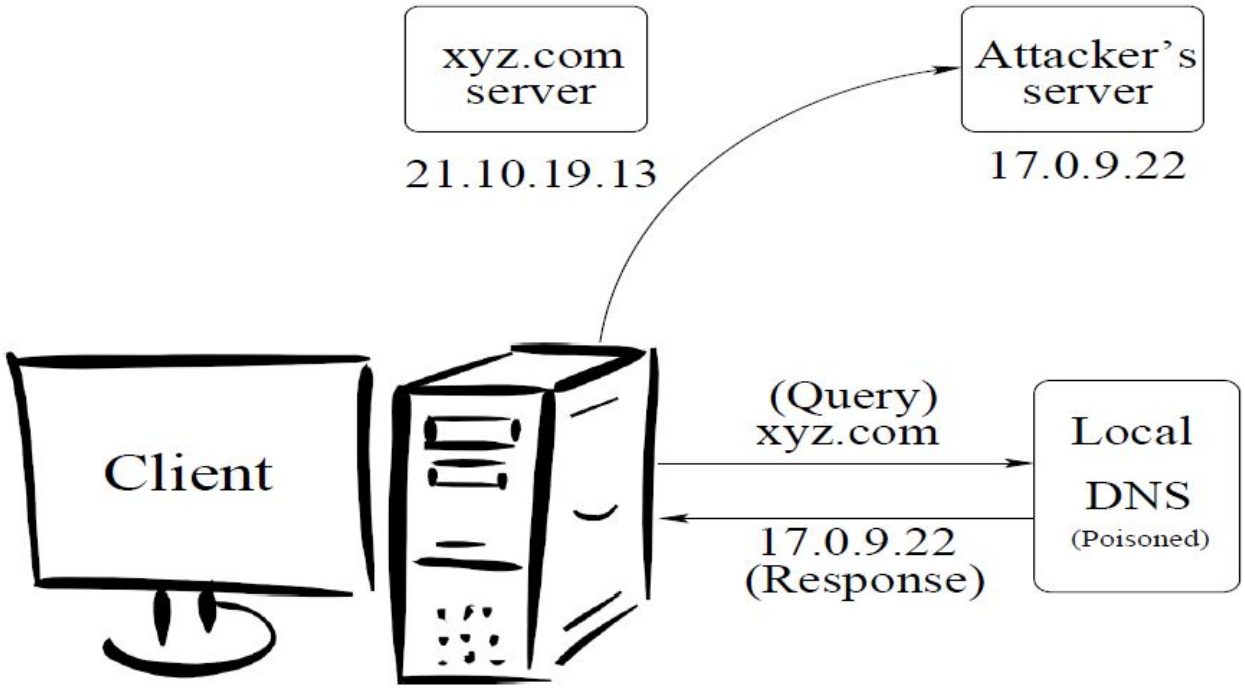
- Alias
  - Multiple domain names for a single host
  - foo.example.com. CNAME bar.example.com.
    bar.example.com. A 192.0.2.23
- Multiple IP addresses for a single domain name
  - Load balancing
  - E.g. www.cnn.com
    - 157.166.255.25
    - 157.166.226.26
    - …

Inherent DNS Vulnerabilities

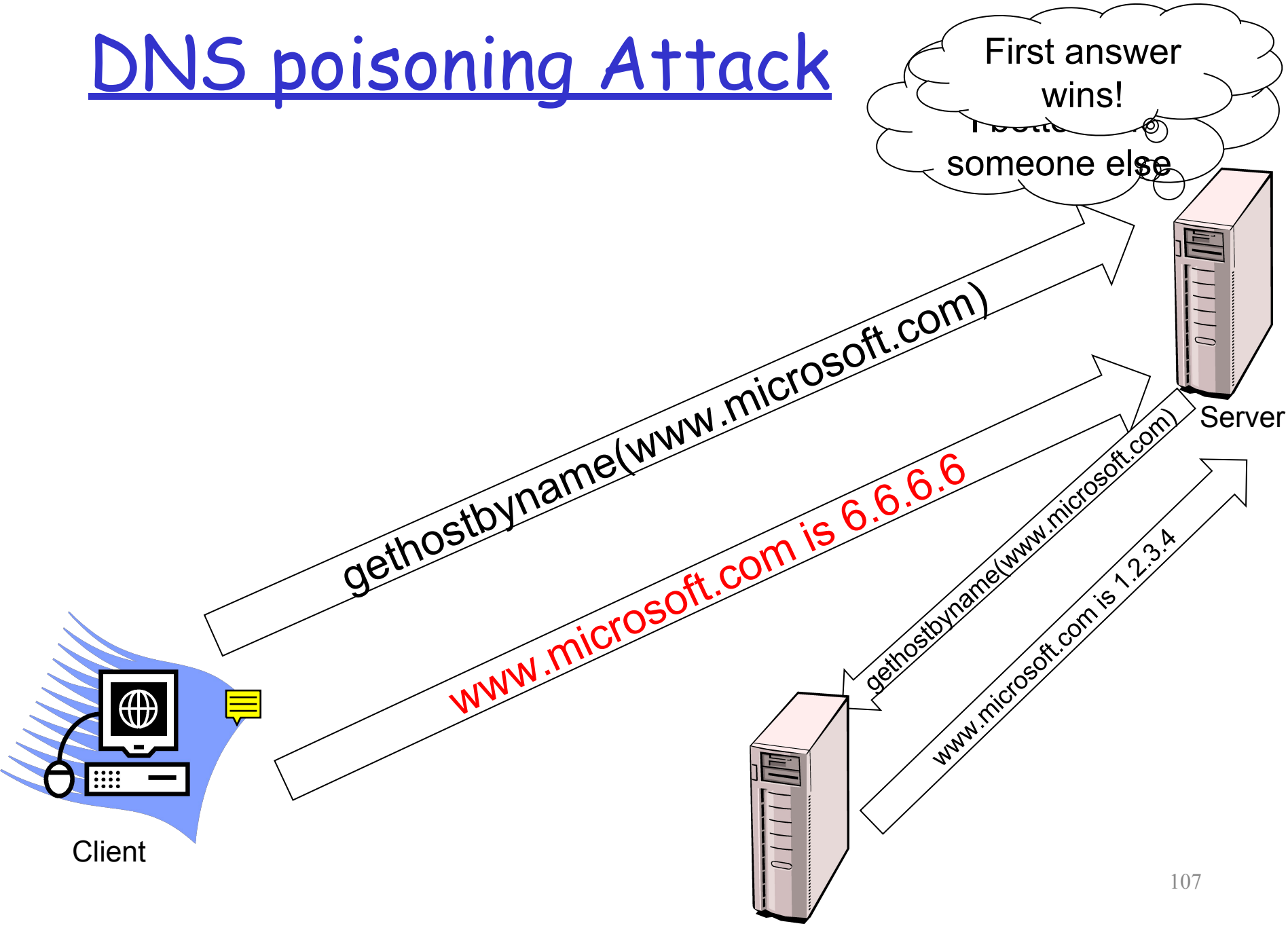
- Users/hosts typically trust the host-address mapping provided by DNS
  - What bad things can happen with wrong DNS info?
- DNS resolvers trust responses received after sending out queries
  - How to attack?
- Responses can include DNS information unrelated to the query
- Obvious problems
  - No authentication for DNS responses

Pharming

- Exploit DNS poisoning attack
  - Change IP addresses to redirect URLs to fraudulent sites
  - Potentially more dangerous than phishing attacks
  - No email solicitation is required
- DNS poisoning attacks have occurred



DNS poisoning Attack



outline

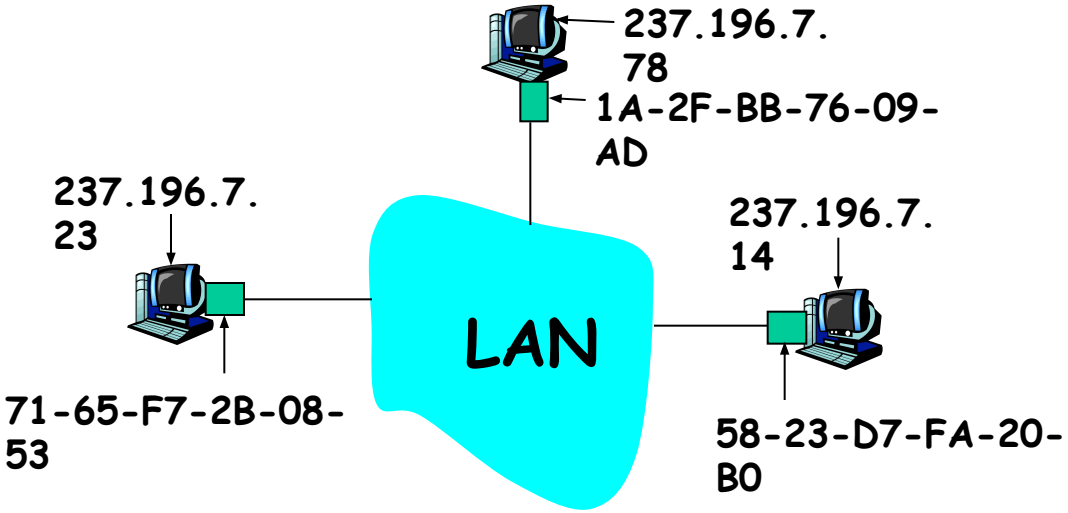
- Intro
- Appl
- TCP
- IP
- DNS
- ARP

Media access control (MAC) address

- 32-bit IP address:
  - *network-layer* address
  - used to get datagram to destination IP subnet
- MAC (or LAN or physical) address:
  - link layer address
  - used to get datagram from one interface to another physically-connected interface (same network)
  - 48 bit MAC address (for most LANs) burned in the adapter ROM
  - Some Network Interface Cards (NICs) can change their MAC

ARP: Address Resolution Protocol

Question: how to determine MAC address of host B when knowing B's IP address?



- Each IP node (Host, Router) on LAN has ARP table
- ARP Table: IP/MAC address mappings for some LAN nodes

<IP address; MAC address; TTL>

  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

ARP

- ARP works by broadcasting requests and caching responses for future use
- The protocol begins with a computer broadcasting a message of the form

  who has <IP address1> tell <IP address2>
- When the machine with <IP address1> or an ARP server receives this message, its broadcasts the response

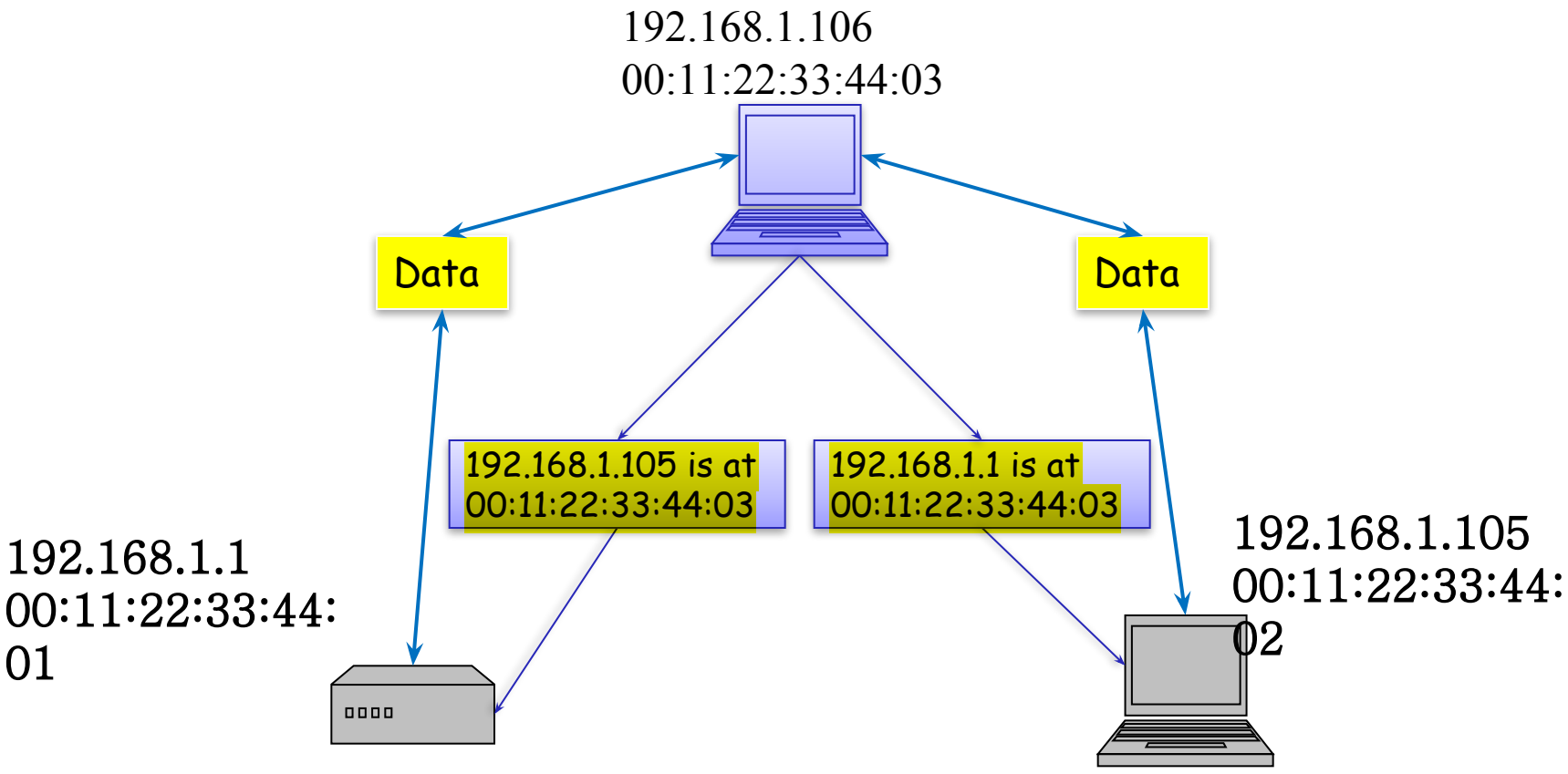
  <IP address1> is <MAC address>
- The requestor's IP address <IP address2> is contained in the link header
- The Linux and Windows command arp - a displays the ARP table

Internet Address	Physical Address	Type
128.148.31.1	00-00-0c-07-ac-00	dynamic
128.148.31.15	00-0c-76-b2-d7-1d	dynamic
128.148.31.71	00-0c-76-b2-d0-d2	dynamic
128.148.31.75	00-0c-76-b2-d7-1d	dynamic
128.148.31.102	00-22-0c-a3-e4-00	dynamic
128.148.31.137	00-1d-92-b6-f1-a9	dynamic

ARP Poisoning (ARP Spoofing)

- According to the standard, almost all ARP implementations are stateless
- An arp cache updates every time that it receives an arp reply… even if it did not send any arp request!
- It is possible to "poison" an arp cache by sending gratuitous arp replies

Poisoned ARP Caches: MitM attack



Poisoned ARP Cache	
192.168.1.**105**	00:11:22:33:44:**03**

Poisoned ARP Cache	
192.168.1.**1**	00:11:22:33:44:**03**