ΕΠΛ 602:Foundations of Web Technologies

Communication: Networking and the Internet Protocols (IP)

Lecture Outline

Part I

- Main points of Lecture 2
- Networking and Network Protocols

Part II

- The Internet Protocols and IP

Main Points

System Models

Three different levels:

- **Physical Model**

Capture the hardware composition of a system in terms of the computer devices and their interconnecting networks

- **Architectural Model**

provide a high-level view of the distribution of functionality between components and the relationships between them

- **Fundamental Model**

vertical views representing some key aspects of distributed systems in an abstract way (e.g., failure or interaction models)

Architectural Models

- What are the entities that are communicating
- How they communicate (communication paradigms)
- What roles and responsibilities?
- How are they mapped on the physical distributed infrastructure (placement)

Architectural Models: what are the entities?

From a **system perspective**: **processes** coupled with appropriate interprocess communication paradigms

Nodes

Threads

From a **programming perspective**: **objects**

A computation consists of a number of interacting objects accessed via interfaces (IDL)

Components

+dependencies between objects (assumptions made about other components/interfaces – contract), non-functional properties such as security + deployment strategies)

Web services

distributed web applications that provide discrete functionality and expose it in a well-defined manner over standard Internet protocols to other web applications

Architectural Models: how they communicate?

- **Interprocess communication**

Relatively low-level support offered (e.g., message-passing primitives, direct access to the API offered by the Internet (*socket programming*) and support for multicast)

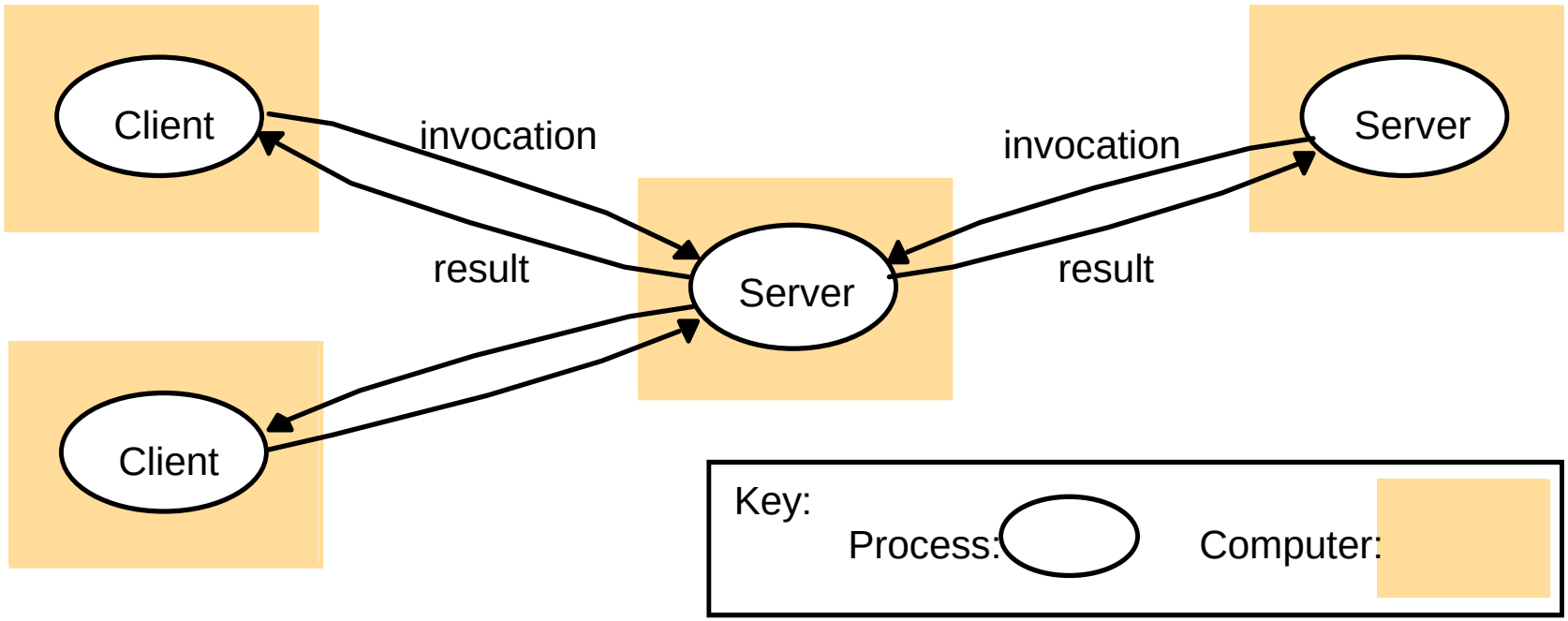
- **Remote invocation**
  - **request-reply protocols** (underlying primitive): a pairwise exchange of messages from server to client - 1st message: encoding of the operation to be executed by the server + arguments, 2nd message: encoded result (example: HTTP)
  - **remote procedure call (RPC):** servers offer a set of operations through a service interface and clients call these operations directly as if local
  - **remote method invocation (RMI):** an object can invoke a method in a remote object + object identity and pass objects as parameters

- **Indirect communication**

For example, group communication, pub/sub, message queues

Architectural Models: roles?

Client-server

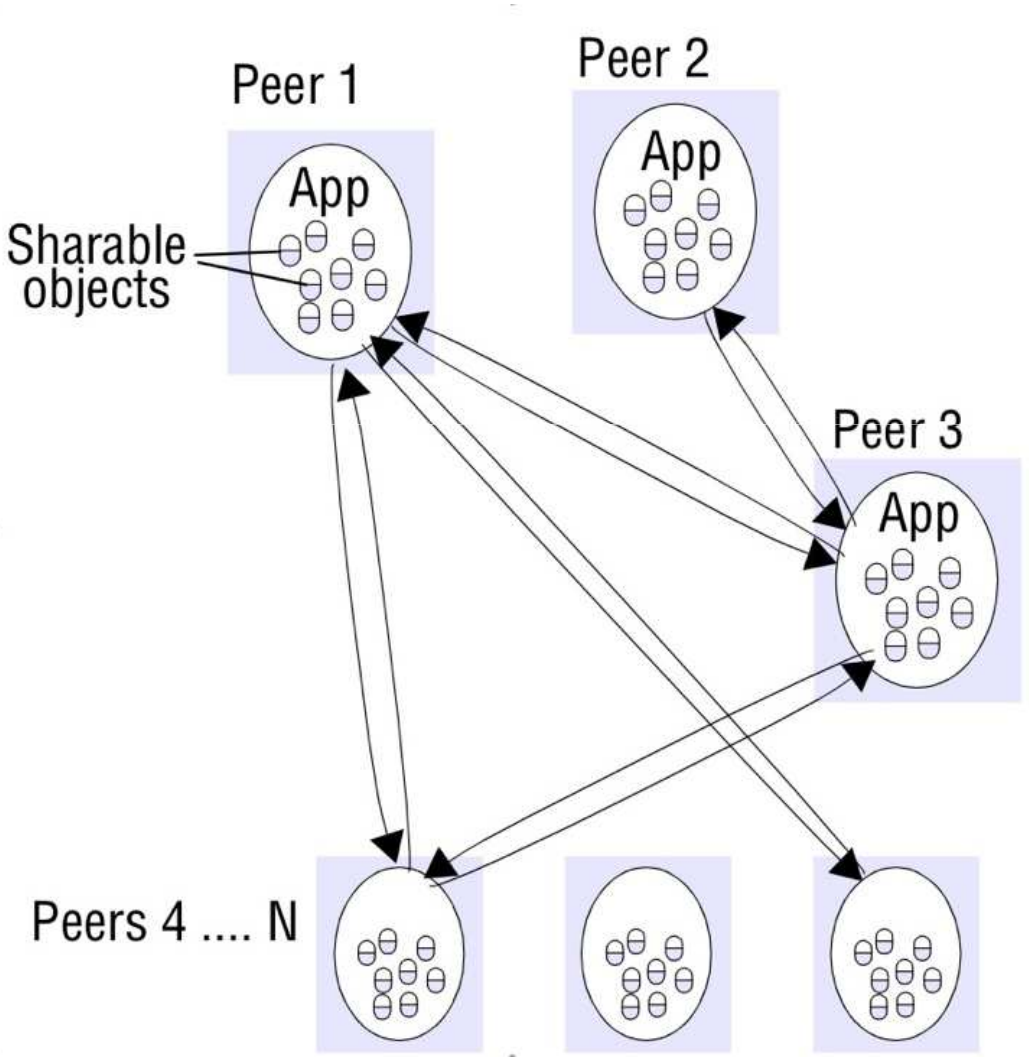


Architectural Models: roles?

Peer-to-peer

Address scalability,

Exploit resources (both data and hardware) at the edge of the network (users)



Architectural Models: placement

Where to place the entities (e.g., objects or services) in terms of machines or processes within machines

- Multiple servers: Partition vs Replication
- Caching
- Mobile code (e.g., applets)

Architectural Patterns

- Structures that have shown to work well
  - Layering
  - Tiering
  - Thin clients
  - Proxy
  - Broker

Middleware is a layer of software whose purpose is to mask heterogeneity and provide a convenient programming model to application programmers

Programming abstractions
+ infrastructure services

Fundamental Models

- Fundamental properties in processes and communication, shared among different architectures discussed previously

- Interaction
- Failures
- Security

Web 101

Client/Server Model Client = Web browser/Server = Web server

1. Markup language for formatting hypertext documents (**HTML**)
2. A uniform notation schema for addressing accessible resources over the Internet (**URL**)
3. A protocol for transporting messages over the network (**HTTP**)

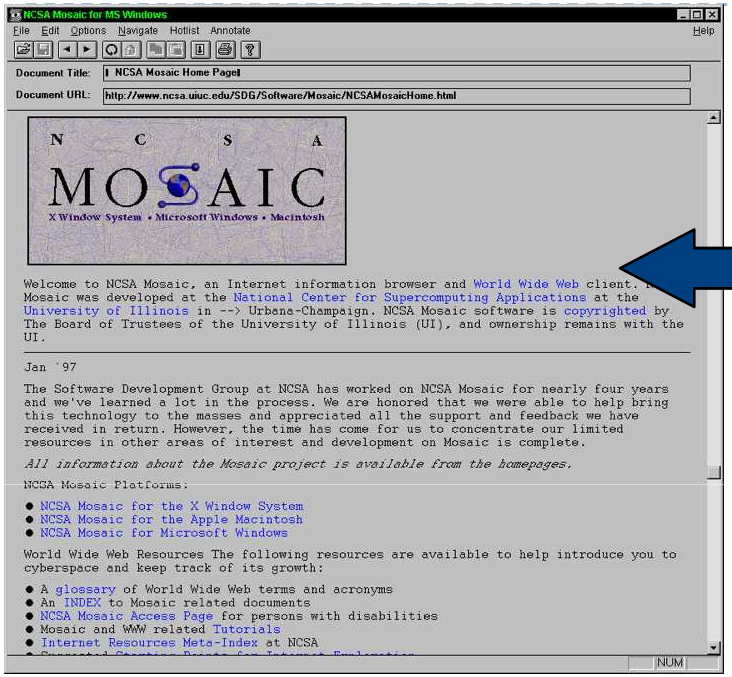
HTML and beyond

HTML a simple markup language to enable cross-referencing in documents through hyperlinks

Cascading Style Sheets (CSS) a mechanism for controlling the style for HTML rendering

XML is a meta-language for defining specialized mark-up languages
DTD, XML Schema
APIs for accessing XML parse trees, XSL, XQuery
XHTML: reformulation of HTML as an XML application
XHTML Mobile Profile

From Web pages to Web applications



The browser gradually became a “platform”

Web application: a client/server application that uses a web browser as its client program

Web browsers

1. Generate and submit requests
2. Accept responses from web servers
3. Render the result

Cookies: a mechanism for maintaining state in browsers across multiple HTTP requests

JavaScript: to support event-handlers – custom code that executes when a browser event occurs

AJAX: a set of programming techniques that enable browsers to communicate asynchronously with web servers
Common uses: code injection (a background request is sent to the server to fetch content – that causes discrete updates to the content displayed in the browser or the data stored on the server)

Web servers

1. Virtual hosting: if the web server is providing service for multiple domains, determine the target domain
2. Address mapping: whether the request is for static or dynamic content
3. Authentication

Delivery of dynamic content

CGI: the CGI mechanism assumes that when a request to execute a CGI script arrives at the server, a new “process” is spawned to execute a particular application program

Servlet API and JSP: Servlet are Java programs that have access to information in HTTP requests. JSP processors generate Java classes that extend the base class that implements the Servlet interface

Web services

Web services are distributed web applications that provide discrete functionality and expose it in a well-defined manner over standard Internet protocols to other web applications

Client -> web application

SOAP is an XML-based application layer protocol for constructing and processing web services requests and responses + Web Service Definition Language (WSDL), Universal Description, Discovery and Integration (UDDI) for registering and discovering web services

REST an architectural pattern as an alternative to SOAP
Items of interest on the web identified by their URL as resources, not static pages but calls to web applications
When accessed such resources return their representations that can be though of as the browser state

From Web from Web2.0

Incorporating applications that support user generated content, on-line communities and collaborative mechanisms for updating on-line content

Site visitors contribute information ranging from reviews and ratings to personal journals (blogs) to news.

Sophisticates web application technology incorporating *user authentication*, *access control* and *content management services*

The end-to-end argument

Question

- How to partition the functionality/roles between the various components in a client-server system

End-to-end argument

- "The function in question can be completely and correctly be implemented only with the knowledge and help of the application standing at the endpoints of the communication system".

- Therefore, providing that questioned function as a feature of the communication subsystem itself is not possible
  - Sometimes, an incomplete version of the function be provided by the communication subsystem may be useful as a performance enhancement.

Lecture Outline

- Main points of Lecture 2
- **Networking and Network Protocols**
- The Internet Protocols and IP

Networking and the Internet Protocols

Networks

Network: a set of devices connected through communication links

- *Why connect to a network?*
  - Share resources
  - help people communicate
- Sharing resources
  - from printers to supercomputer centers
- Helping people communicate
  - email, Web, active documents

Networks: components

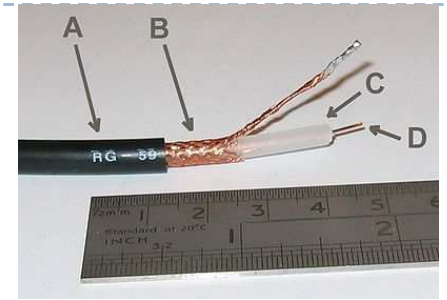
Communication Subsystem: hardware and software components that provide the communication facilities for the distributed system

- Transmission media (wire, fibre, wireless channels)
- Hardware devices (routers, switches, bridges, hubs, repeaters, network interfaces)
- Software (protocols stacks, communication handlers and drivers)

Subnets is a collection of nodes that may all be reached on the same physical network

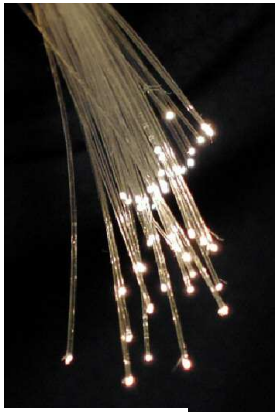
Internet – internetworking among subnets

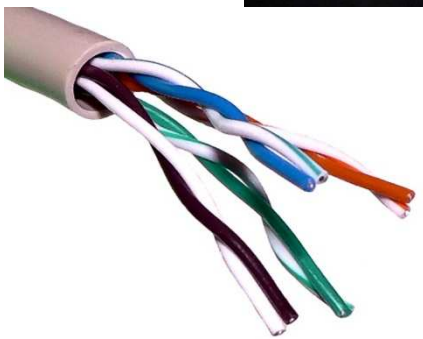
Networks: components



Transmission Media

- **Air**
  - Sound
- **Air, vacuum**
  - Radio, TV, microwave, cellular telephone
  - Satellite
- **Wire**
  - Copper: twisted pair, coaxial cable
- **Optical fiber**
  - LED





Networks: components

- Advantages of fiber
  - higher bandwidth
  - smaller and lighter
  - less prone to interference
  - less prone to eavesdropping
- advantages of copper
  - simple
  - cheap to interface to

Networks: components

Links

Fibers

Coaxial Cable

Interfaces

Ethernet card

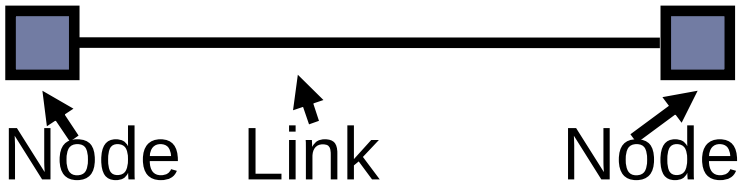
Wireless card

Switches/routers

Large router

Telephone switch

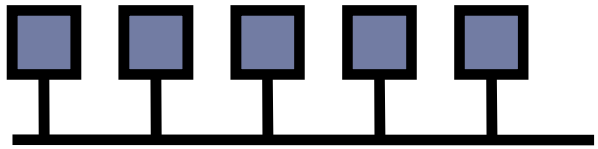
Simple Network



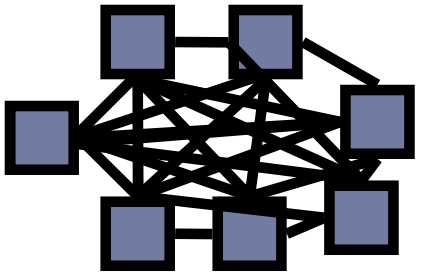
- Node: computer
  - End host: general-purpose computer, cell phone, PDA
  - Network node: switch or router
- Link: physical medium connecting nodes
  - Twisted pair: the wire that connects to telephones
  - Coaxial cable: the wire that connects to TV sets
  - Optical fiber: high-bandwidth long-distance links
  - Space: propagation of radio waves, microwaves, …

Connecting More Than Two Hosts

- Multi-access link: Ethernet, wireless
  - Single physical link, shared by multiple nodes
  - Limitations on distance and number of nodes
- Point-to-point links: fiber-optic cable
  - Only two nodes (separate link per pair of nodes)
  - Limitations on the number of adapters per node

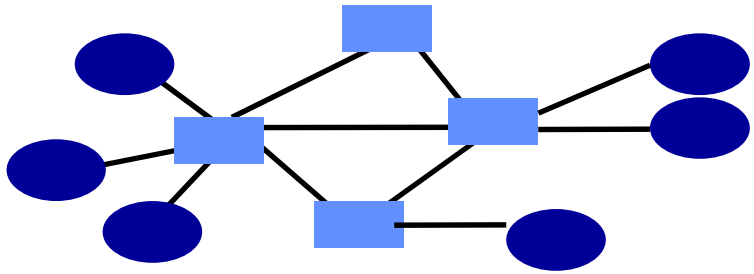


multi-access link



point-to-point links

Beyond Directly-Connected Networks



- Switched network
  - End hosts at the edge
  - Network nodes that switch traffic
  - Links between the nodes
- Multiplexing
  - Many end hosts communicate over the network
  - Traffic shares access to the same links

Networks: issues

Performance:

Latency (time between send and start to receive) (software overheads, routing delays, load arising from conflicting demands)

Data transfer rate (bits per second) [max] (determined by physical characteristics)

- Transmission time = latency + length / transfer rate

System bandwidth, throughput [actual]: total volume of traffic that can be transferred in a given amount of time

- Using *different channels* concurrently can make bandwidth > data transfer rate
- traffic *load* can make bandwidth < data transfer rate
- network speed < memory speed (about 1000 times)
- Access to local disk is usually faster than remote disk
  - Fast (expensive) remote disk + fast network can beat slow (cheap) local disks

Networks: issues

- scalability
- reliability
  - corruption is rare
  - mechanisms in higher-layers to recover errors [remember end-to-end argument]
  - errors are usually timing failures, the receiver doesn't have resources to handle the messages
- security
  - firewall on gateways (entry point to org's intranet): filter incoming and outgoing messages based on the org's policy
  - encryption is usually in higher-layers
- mobility--communication is more challenging: locating, routing,...
- quality of service--real-time services (multimedia data)
- multicasting--one-to-many communication

Types of networks

- Personal Area Networks (PANs)
- Local Area Networks (LANs)
- Wide Area Networks (WANs)
- Metropolitan Area Networks (MANs)

and wireless variants
WPANs, WLANs, WWANs

Types of networks: (W)PANs

Personal Area Networks (PANs)

The various digital devices of a user connected by a low-cost, low-energy network

WPANs

Bluetooth, infra-red for PDAs,

Types of networks: (W)LANs

Local Area Networks (LANs)

Floor/building wide
Relatively high speeds between computers connected by a single communication medium
Segment in a building - no routing (broadcast)
segments connected by switches or hubs

Ethernet (broadcast), from 10Mbps, 100Mbps, 1Gbps
high bandwidth, low latency

WLANs

Wireless: provide connectivity or remove the need for wired infrastructures within buildings
Several variants of IEEE 802.11 (WiFi)
10-100 Mbps up to 1.5 kms

Types of networks: (W)WANs

Wide Area Networks (WANs)

Lower speeds between nodes geographically distributed (world-wide, different organizations)
A set of communication circuits linking a set of dedicated computers – routers
Total latency depends on the route + load at the network segments 600Mbps, 1-10 Mbps for bulk transfers
1 - .5, lower bound propagation delay Europe – Australia (0.13 sec (terrestrial link) 0.20 sec (geostationary satellites)

WWAN

Mobile phones based on GSM
Wide area connection to the Internet
(3G) 2-14.4 Mbps (stationary) 348 kbps (while moving)

Types of networks: MANs

Metropolitan Area Networks (MANs)

Based on high-bandwidth copper and fiber cabling
City-wide, distances up to 50 kms
DSL uses ATM switches located in telephone exchanges to route digital data to the subscriber over twisted pairs of copper 1-10 Mbps
5.5 kms from the switch

Cable modem analogue signaling on cable television networks up to 1.5 Mbps
coaxial cable with greater ranges than DSL
VDSL 100Mbps

Types of networks: Internetworks

Intrernetworks linked together several networks

Openness

Interconnection through dedicated switching computers (**routers**) + general-purpose computers (**gateways**) + software layer

Virtual network

Example: Internet – TCP/IP

Types of networks

	Example	*Range*	*Bandwidth (Mbps)*	*Latency (ms)*
Wired:				
LAN	Ethernet	1–2 kms	10–10,000	1–10
WAN	IP routing	worldwide	0.010–600	100–500
MAN	ATM	2–50 kms	1–600	10
Internetwork	Internet	worldwide	0.5–600	100–500
Wireless:				
WPAN	Bluetooth (IEEE 802.15.1)	10–30m	0.5–2	5–20
WLAN	WiFi (IEEE 802.11)	0.15–1.5 km	11–108	5–20
WMAN	WiMAX (IEEE 802.16)	5–50 km	1.5–20	5–20
WWAN	3G phone	cell: 1–-5	348–14.4	100–500

Types of networks: errors

Very high reliability of the underlying transmission media

Most packet losses due to processing delays and buffer overflows

Out-of-order transmissions (e.g., in wide area networks where separate packets are individually routed)

Duplicate copies (retransmission)

Network Principles

In a nutshell

Before a **message** (a logical unit of information) is transmitted is subdivided into **packets** (transmission unit) of restricted length

Simplest form: sequence of bits of restricted length + addressing information (source, destination)

Packet switching (store-and-forward)
Just forward packets from the source to their destination
At each switching node a computer:
o Packets are queued in a buffer and transmitted when the link is available

Communication is asynchronous – messages arrive at their destination after a delay that varies

- o Why packets and packet switching?
- o How packets are routed?
- o Protocols
- o Internetworking

Network Principles

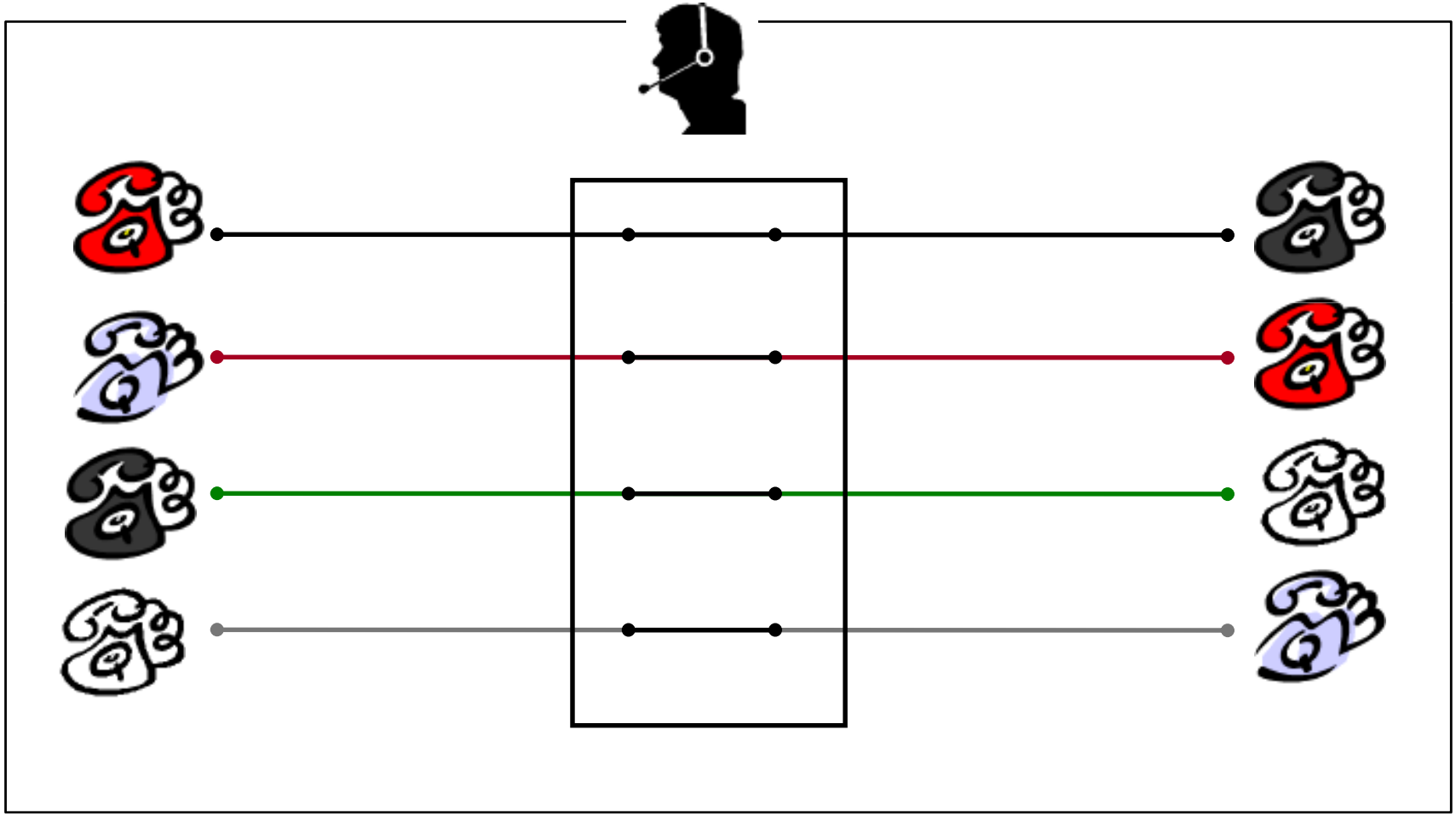
1. Packet Switching
2. Routing
3. Protocols
4. Internetworking

Switching

- Circuit switching
  - Plain old telephone system (POTS)
- Packet switching
  - Store-and-Forward
- Broadcast
  - No switching (Ethernet, wireless (cells), etc)
- Virtual circuit-frame relay
  - ATM networks
  - Small packets (frames), fast routing (on the fly)

Switching: circuit

Circuit Switching With Human Operator



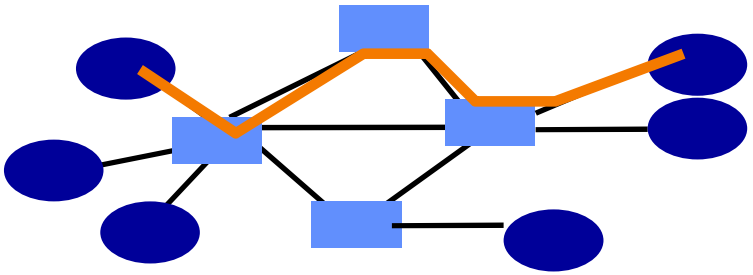
Switching: circuit

Example: telephony

- Resources reserved during call setup
- resources dedicated for duration of call
- conservative
  - guarantee high quality of service to all calls
  - resources dedicated even if call does not always need them
  - good for constant-bit-rate traffic

Switching: circuit

- Source establishes connection to destination
  - Node along the path stores connection info
  - Nodes may reserve resources for the connection
- Source sends data over the connection
  - No destination address, since nodes know path
- Source tears down connection when done



Switching: circuit

- Guaranteed bandwidth
  - Predictable communication performance
  - Not "best-effort" delivery with no real guarantees
- Simple abstraction
  - Reliable communication channel between hosts
  - No worries about lost or out-of-order packets
- Simple forwarding
  - Forwarding based on time slot or frequency
  - No need to inspect a packet header
- Low per-packet overhead
  - Forwarding based on time slot or frequency
  - No IP (and TCP/UDP) header on each packet

Switching: circuit

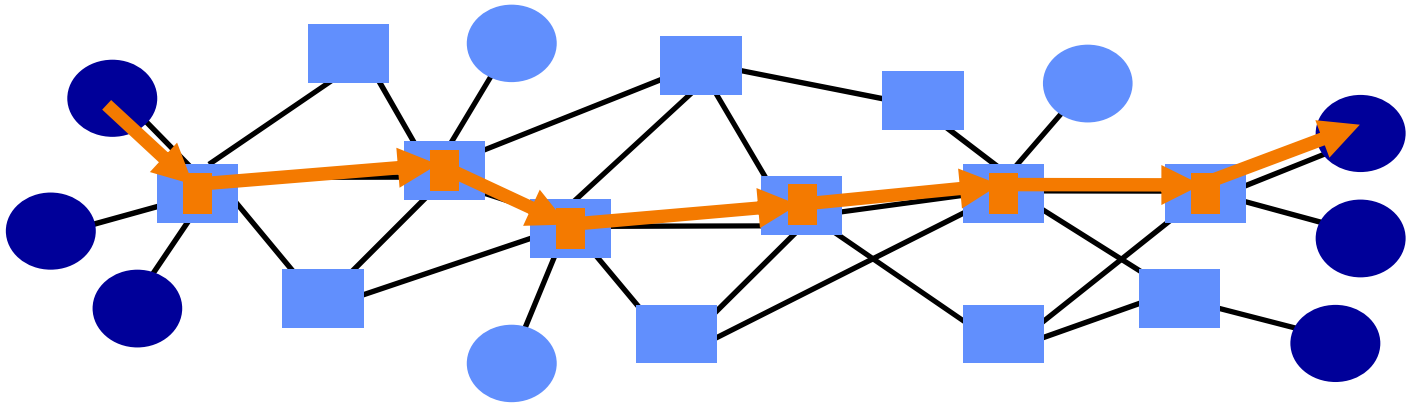
- Wasted bandwidth
  - Bursty traffic leads to idle connection during silent period
  - Unable to achieve gains from statistical multiplexing
- Blocked connections
  - Connection refused when resources are not sufficient
  - Unable to offer "okay" service to everybody
- Connection set-up delay
  - No communication until the connection is set up
  - Unable to avoid extra latency for small data transfers
- Network state
  - Network nodes must store per-connection information
  - Unable to avoid per-connection storage and state

Switching: packet

- Alternative to circuit switching
  - example: Internet
- entering data divided into packets
- Store-and-forward
- packets in network share resources
  - no performance guarantees
- queue packets if link contention
- statistical multiplexing of resources

Switching: packet

- Data traffic divided into packets
  - Each packet contains a header (with address)
- Packets travel separately through network
  - Packet forwarding based on the header
  - Network nodes may store packets temporarily
- Destination reconstructs the message



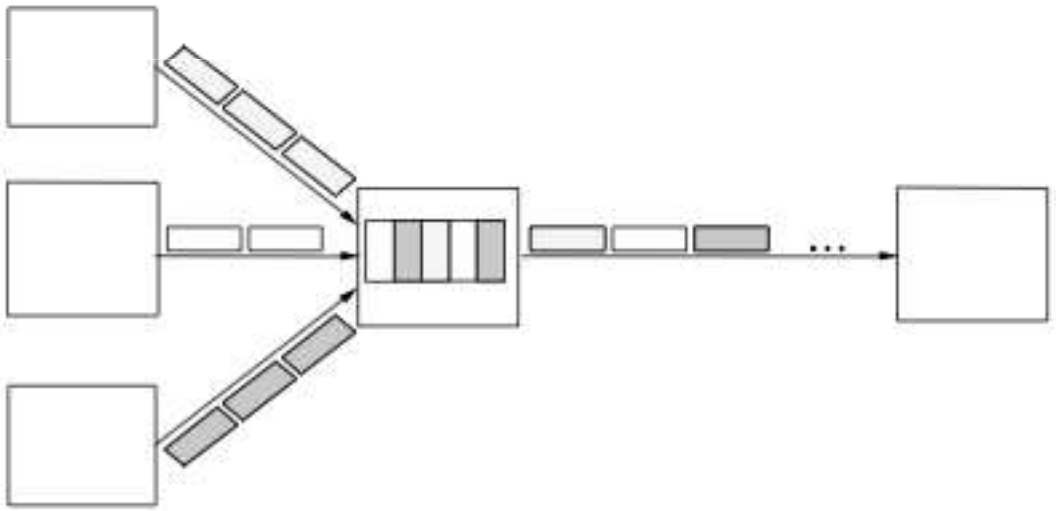
Switching: virtual circuit

- Cross between circuit switching and packet switching
  - set up path before data flows
  - resources along path are shared
  - Each packet now contains just a virtual circuit number
  - example: asynchronous transfer mode (ATM)
  - cheaper than circuit switching, better guarantees than packet switching
  - but complicated

Datagram Packet Delivery: IP, Ethernet and most wired and wireless local area network technologies

Multiplexing

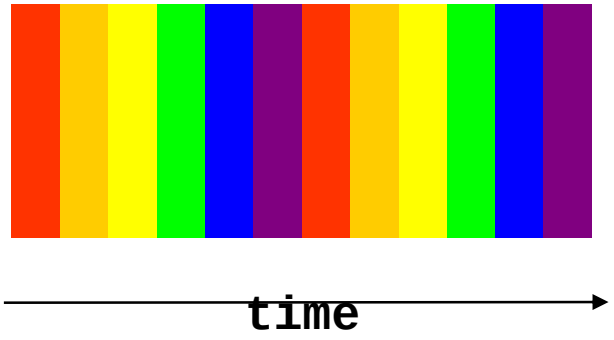
- Synchronous Time-Division Multiplexing (STDM)
- Frequency Division Multiplexing (FDM)

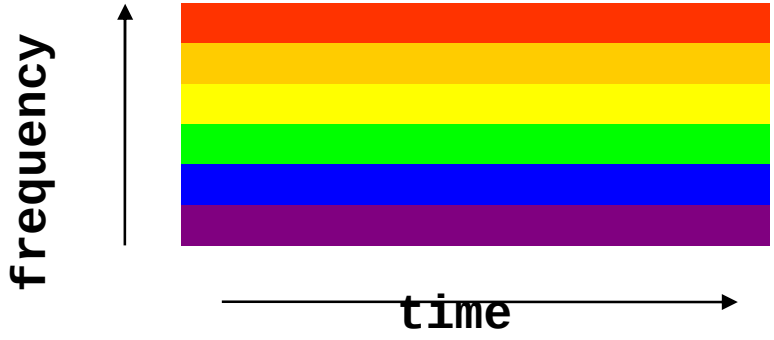


Multiplexing

- Time-division
  - Each circuit allocated certain time slots

- Frequency-division
  - Each circuit allocated certain frequencies



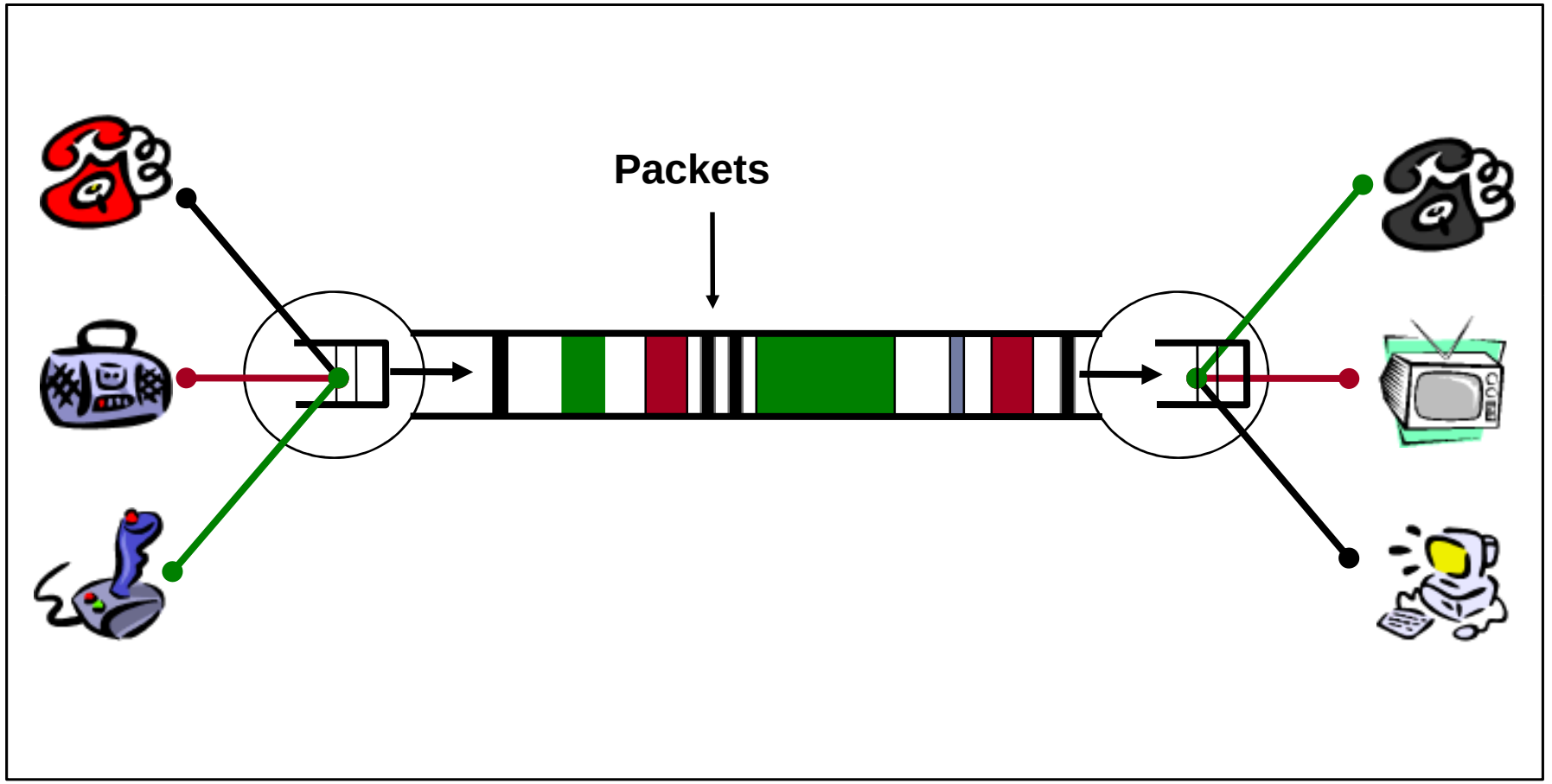


Multiplexing

Statistical Multiplexing

- on-demand time division
- Per packet
- Multiplexing packets from different sources
- Congestion: buffer full

Packet Switching: Statistical Multiplexing



Network Principles

1. Packet Switching
2. Routing
3. Protocols
4. Internetworking

Routing

Required in all networks but LAN

Routing Algorithm: decide which out-going link to forward the packet

- for circuit switching, the route is determined during the circuit setup time
- for packet switching, each packet is routed independently

Adaptive routing: the best route for communication between two points on the network is re-evaluated periodically

Routing algorithm:

- Determine the route for each packet
- Update its knowledge of the network (traffic monitoring and detection of configuration changes or failures)

Both distributed – based on local information In a hop-by-hop basis

Routing Table

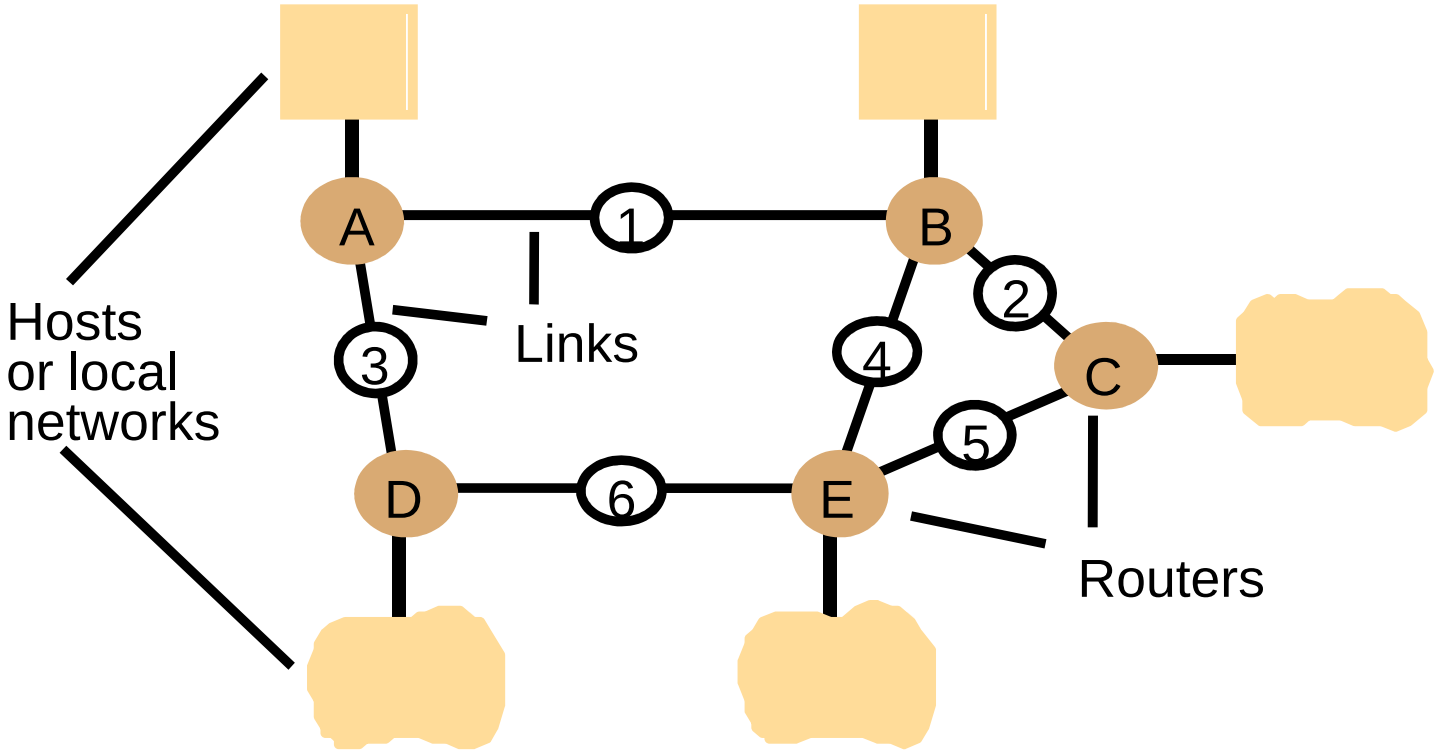
a record for each destination

fields: outgoing link, cost (e.g. hop count)

Path finding in graphs

Routing

- Router example



Routing

Routings from A

To	*Link*	*Cost*
A	local	0
B	1	1
C	1	2
D	3	1
E	1	2

Routings from B

To	*Link*	*Cost*
A	1	1
B	local	0
C	2	1
D	1	2
E	4	1

Routings from C

To	*Link*	*Cost*
A	2	2
B	2	1
C	local	0
D	5	2
E	5	1

Routings from D

To	*Link*	*Cost*
A	3	1
B	3	2
C	6	2
D	local	0
E	6	1

Routings from E

To	*Link*	*Cost*
A	4	2
B	4	1
C	5	1
D	6	1
E	local	0

Routing

- Router information protocol (RIP)
  - "Bellman-Ford distance vector" algorithm
  - Sender: send table summary periodically (every 30sec in the Internet) or changes to neighbors
  - Receiver: Consider *A* receives a table from *B*, *A* updates
    1. *A* -> *B* -> … -> *X*: *A* updates--*B* has more up-to-date (authoritative) info
    2. *A* -> not *B* -> … -> *X*: Does routing via *B* have a lower cost?
    3. *B* -> … -> *X*: *A* does not know *X*
    4. *[B* -> *A* -> … -> *X]*: *A* doesn't update--*A* has more up-to-date info
    5. Faulty link, cost is infinity

Routing

- *Tl* is the table local table; *Tr* is the received remote table

Send: Each *t* seconds or when *Tl* changes, send *Tl* on each non-faulty outgoing link.

Receive: Whenever a routing table *Tr* is received on link *n*:

```
for all rows Rr in Tr {
    if (Rr.link  != n) {  // destination not routed via the receiver
        Rr.cost = Rr.cost + 1;
        Rr.link = n;
        if (Rr.destination is not in Tl) add Rr to Tl;
         // add new destination to Tl
        else for all rows Rl in Tl {
            if (Rr.destination = Rl.destination and
                    (Rr.cost < Rl.cost or Rl.link = n)) Rl = Rr;
            // Rr.cost < Rl.cost : remote node has better route
            // Rl.link = n : remote node is more authoritative
        }
    }
}
```

Routing

RIP-1 (RFC 1058)

- More recent algorithms
  - more information, not just neighbors (e.g., actual bandwidth)
  - link-state algorithms, each node responsible for finding the optimum route
  - A database at each node that represent all or a substantial portion of the network
  - Optimum routes to the destinations in the database

Routing

- Congestion control
  - high traffic load, packets dropped due to limited resources
  - reducing transmission rate: "choke packets“ (request a reduction in transmission) from sender to receiver

Network Principles

1. Packet Switching
2. Routing
3. Protocols
4. Internetworking

What is protocol?

- "The forms of ceremony and etiquette observed by diplomats and heads of state."

  From Greek *protos*, first + *kollema*, sheets of papyrus glued together = table of contents.

- Proper behavior, required for independent beings to cooperate
- Examples:
  - Meeting someone on the street (varies by culture)
  - Attending class
  - Making a purchase
  - Having a conversation

What is protocol: in a restaurant

- Enter (reservation? waiting list, tip maître d')
- Checkroom (coats, umbrella, packages, ticket)
- Seating (choice of tables, follow maître d'?)
- Menu (sequence of items, prices, explanations)
- Order (sequence, choices, options, side dishes)
- Serving (sequence of service, placement of dishes)
- Eating (use of utensils)
- Bill, payment (check bill, credit card, tip, receipt)
- Leave (retrieve checked items)

What is protocol: setting a meeting

- You: When are you free to meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- You: Book me for 1.5 hours at 10:30am on Feb 8.
- Advisor: Yes.

What is protocol: setting a meeting

- Student #1: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #2: When can you meet for 1.5 hours during the next two weeks?
- Advisor: 10:30am on Feb 8 and 1:15pm on Feb 9.
- Student #1: Book me for 1.5 hours at 10:30am on Feb 8.
- Advisor: Yes.
- Student #2: Book me for 1.5 hours at 10:30am on Feb 8.
- Advisor: Uh… well… I can no longer can meet then. I'm free at 1:15pm on Feb 9.
- Student #2: Book me for 1.5 hours at 1:15pm on Feb 9.
- Advisor: Yes.

What is protocol: details

- How to identify yourself?
  - Name? Social security number?
- How to represent dates and time?
  - Time, day, month, year? In what time zone?
  - Number of seconds since Jan 1, 1970?
- What granularities of times to use?
  - Any possible start time and meeting duration?
  - Multiples of five minutes?
- How to represent the messages?
  - Strings? Record with name, start time, and duration?
- What do you do if you don't get a response?
  - Ask again? Reply again?

Protocols: Networks

A specification of:

1. The *sequence* of the messages to be exchanged
2. The *format* of the data in the messages

Implemented by a pair of software modules

Enables separate software components to be developed independently and implemented differently

Protocols: Layering

Arranged in layers

Each layer

- an interface to the layers above – provides a service
- extends the service provided by the layer below

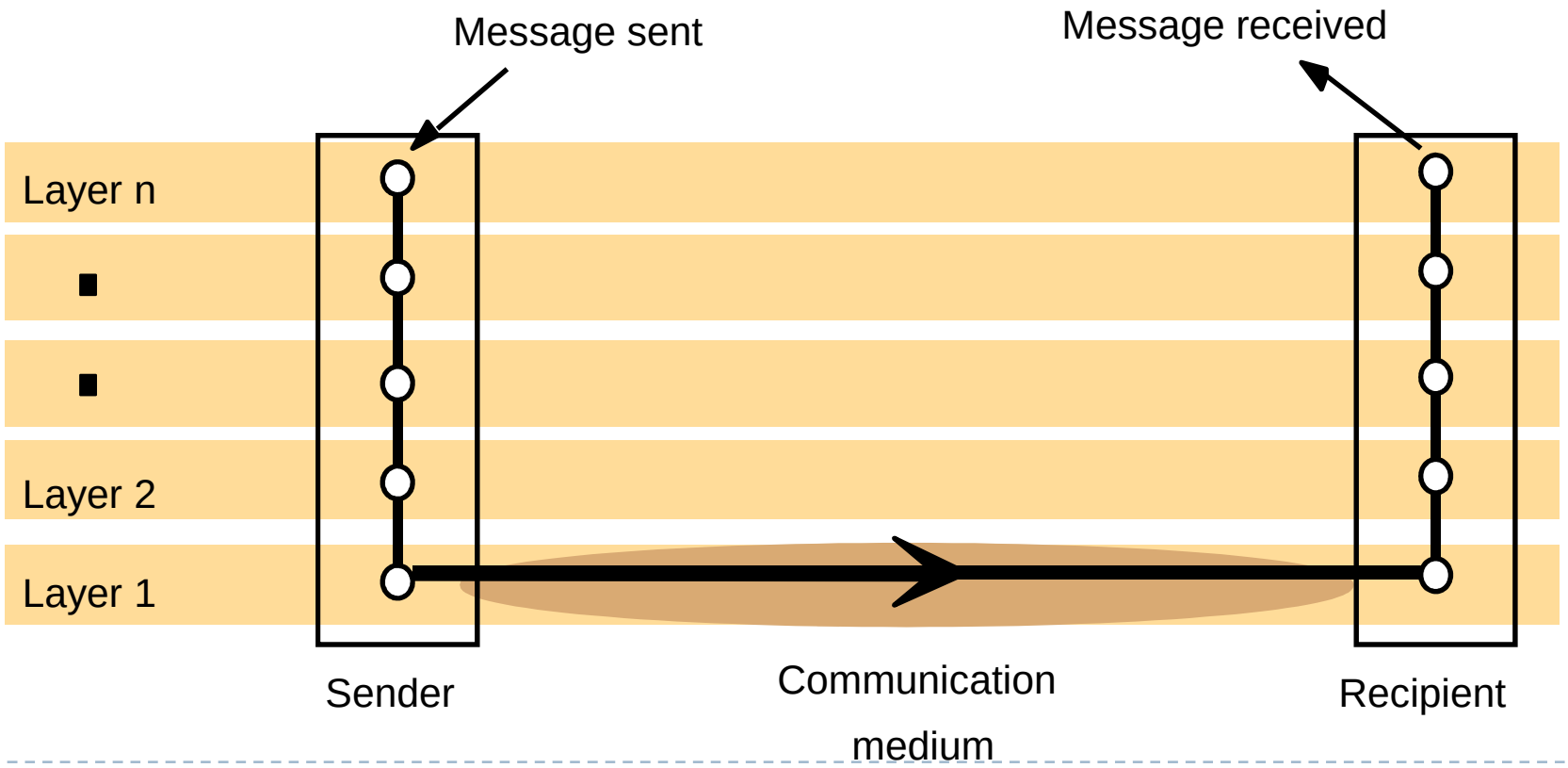
Each layer is represented by a software module in every computer connected to the network

At the bottom the physical layer

Protocols: Layers

Flow of data: each module appears to communicate directly with the module at the same level in the other computer, but data are transmitted directly between the module at each level

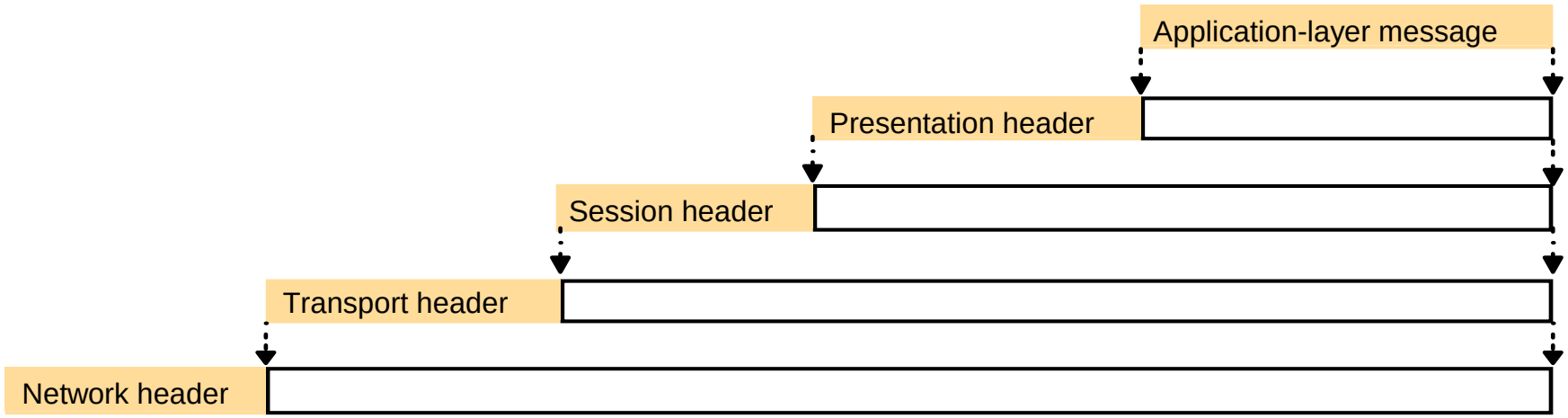
Each layer by local procedure calls to the layers above and below it



Protocols: Layers

Encapsulation

At the sending side, each layer accepts items at the specified format



Protocols: Layers

Protocol suite or Protocol stack: a complete set of protocols reflecting the layered structure

ISO OSI

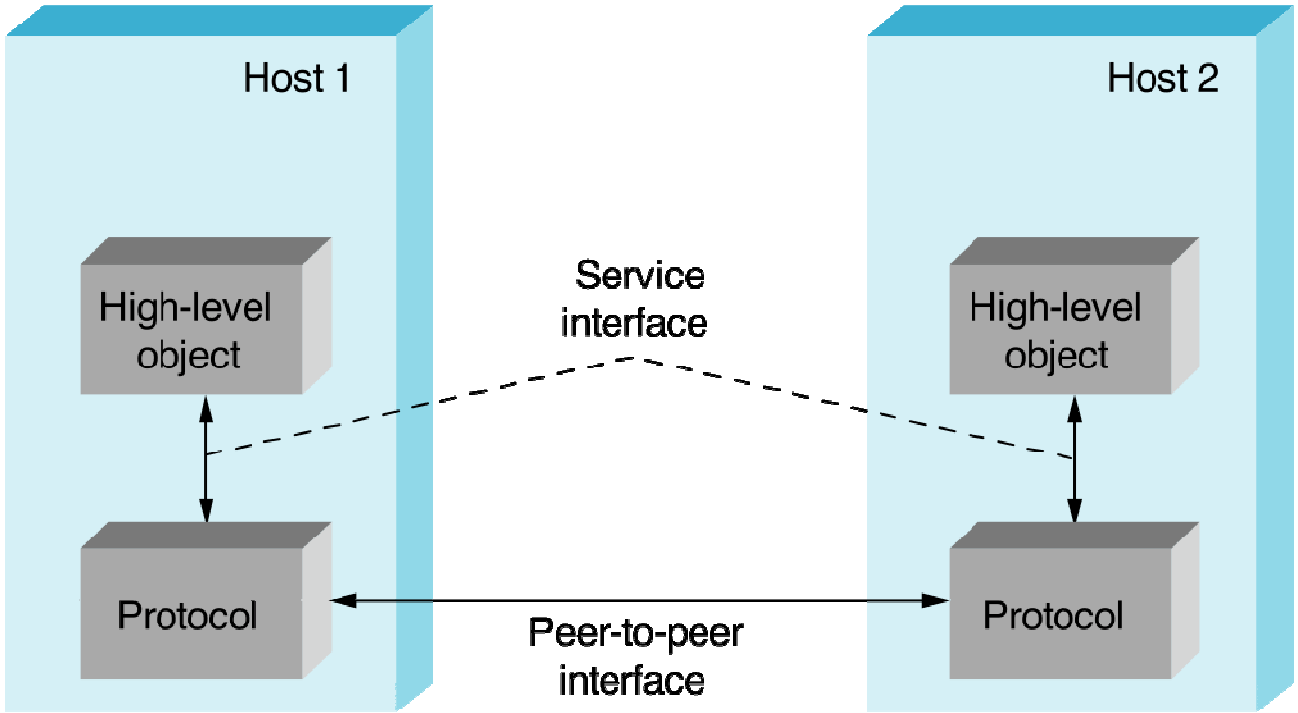
- Structure and function of data communications protocols described according to the architectural model developed by ISO and called the Open System Interconnect (OSI).

- contains *seven layers* that *define the functions* of data communication protocols.

- **A layer does not define a single protocol** - it defines a data communications function that may be performed by any number of protocols.
So, one layer may contain multiple protocols.

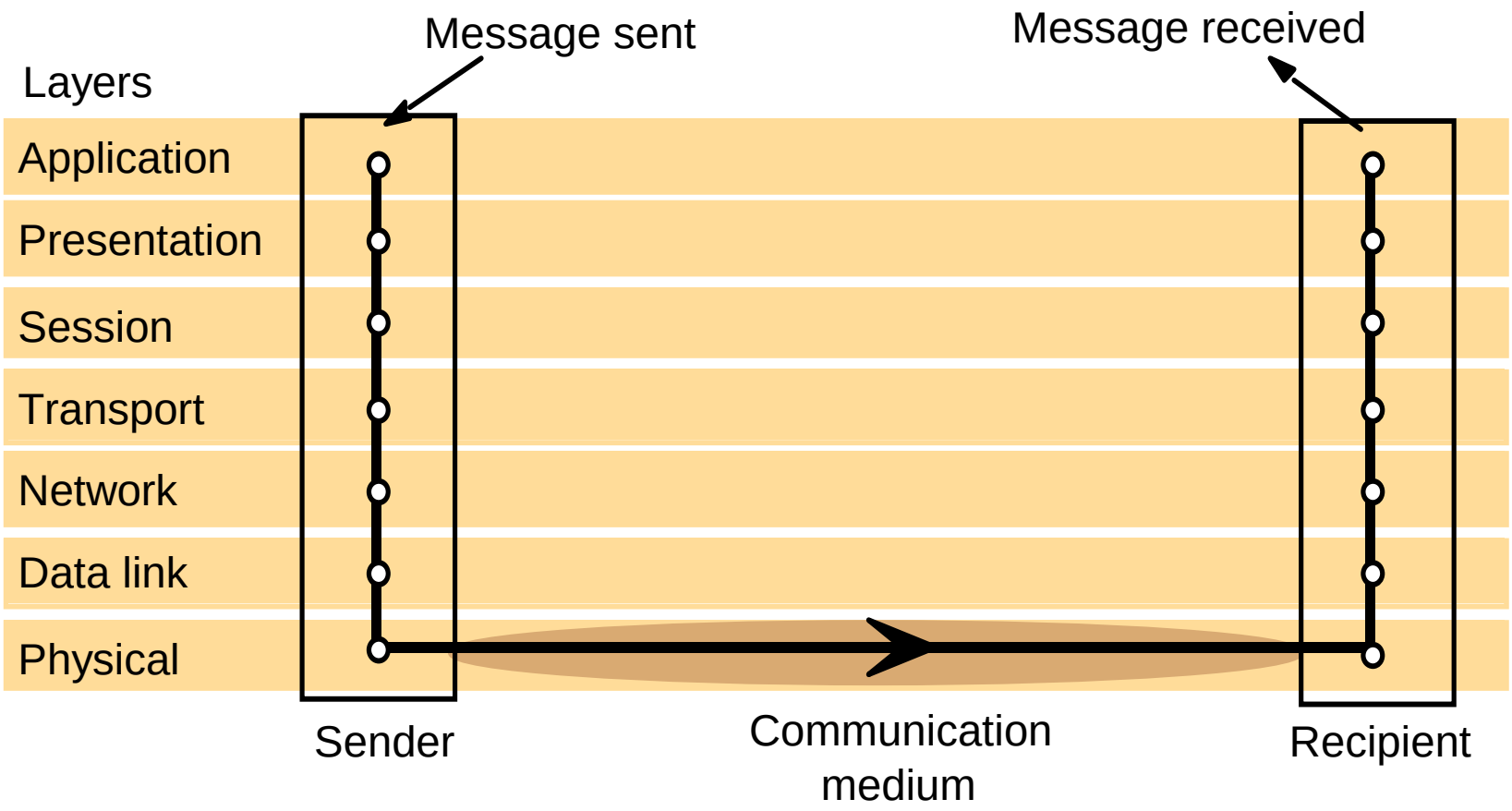
- Every protocol communicates with its peer. A **peer** is an implementation of the same protocols in the equivalent layer on a remote system.

Protocols: Layers



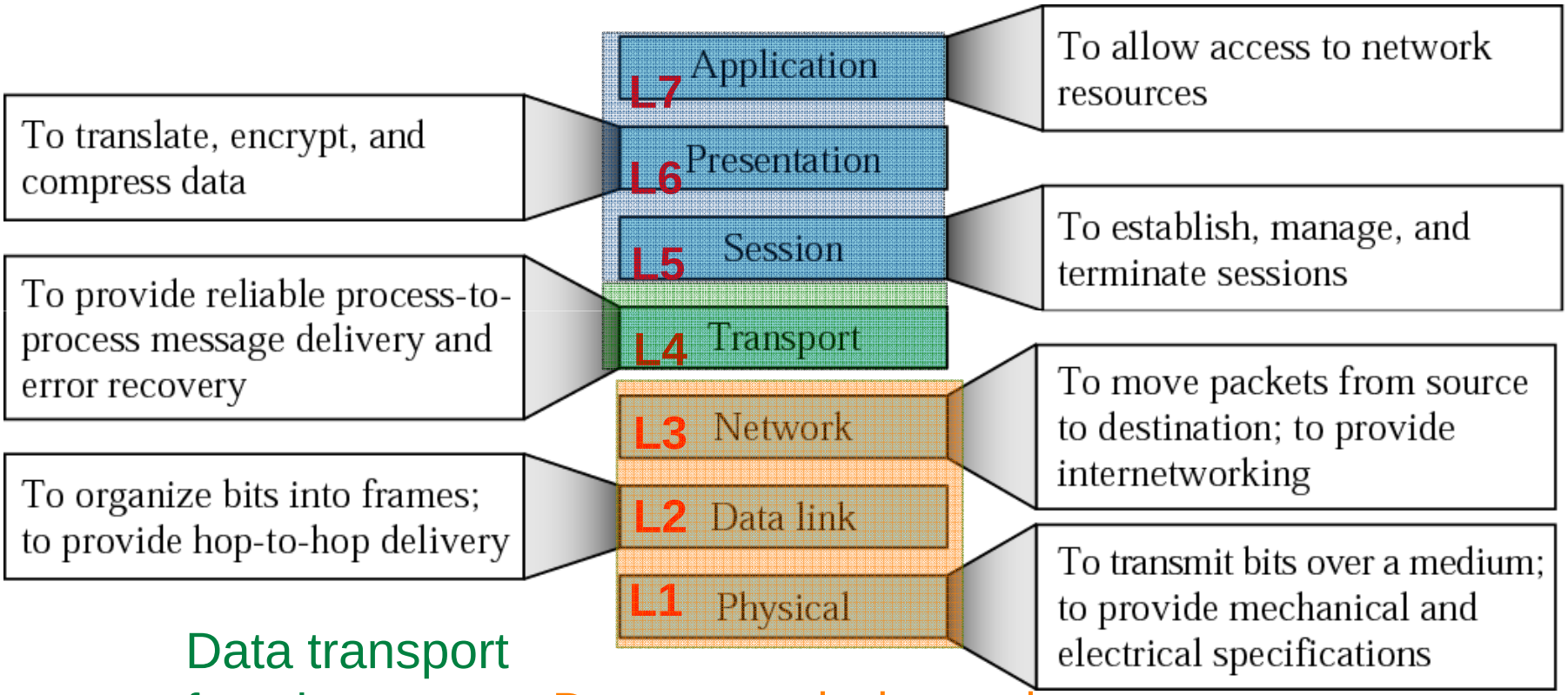
Protocols: OSI Layers

ISO Open Systems Interconnection (OSI) model



Protocols: OSI Layers

Application-related function



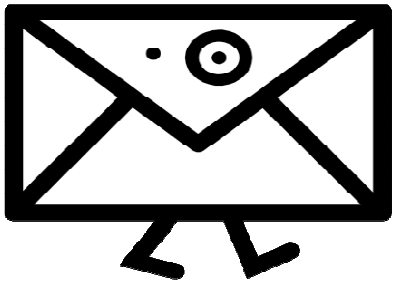
Data transport function

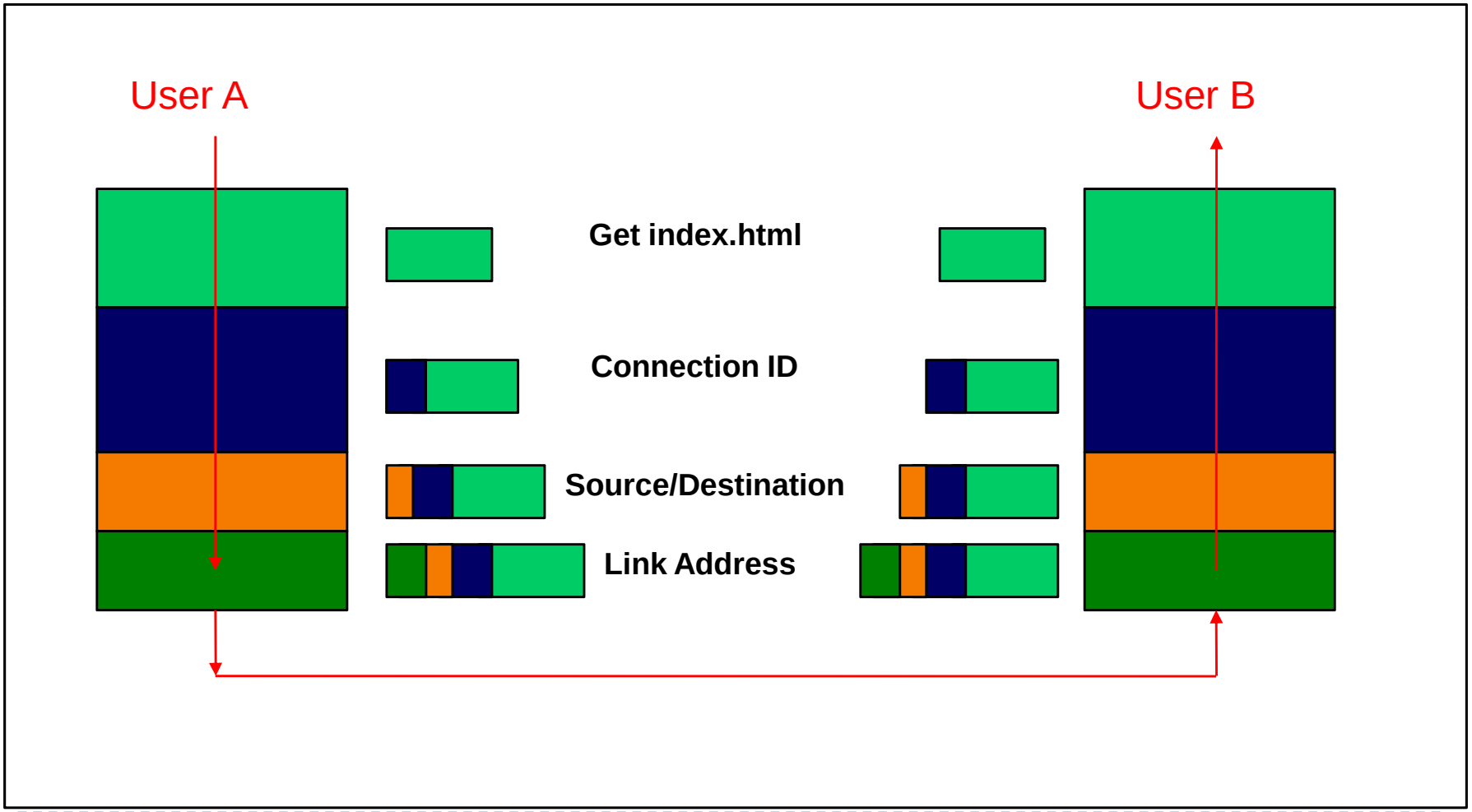
Data transmission and Routing functions

Protocols: OSI Layers (details)

Layer	Description	Examples
Application	Protocols that are designed to meet the communication requirements of specific applications, often defining the interface to a service.	**HTTP**, FTP , SMTP, CORBA IIOP
Presentation	Protocols at this level transmit data in a network representation that is independent of the representations used in individual computers, which may differ. Encryption is also performed in this layer, if required.	Secure Sockets (SSL),**CORBA** Data Rep.
Session	At this level *reliability and adaptation* are performed, such as detection of failures and automatic recovery.	
Transport	This is *the lowest level at which messages (rather than packets) are handled*. *Messages are addressed to communication ports attached to processes*, Protocols in this layer may be connection-oriented or connectionless.	**TCP**, UDP
Network	Transfers *data packets between computers in a specific network*. In a WAN or an internetwork this involves the generation of a route passing through routers. In a single LAN no routing is required.	**IP,** ATM virtual circuits
Data link	Responsible for transmission of *packets between nodes that are directly connected by a physical link*. In a WAN transmission is between pairs of routers or between routers and hosts. In a LAN it is between any pair of hosts.	Ethernet MAC, ATM cell transfer, PPP
Physical	The circuits and hardware that drive the network. It transmits sequences of binary data by analogue signalling, using amplitude or frequency modulation of electrical signals (on cable circuits), light signals (on fibre optic circuits) or other electromagnetic signals (on radio and microwave circuits).	Ethernet base-band signalling, ISDN

Layer Encapsulation



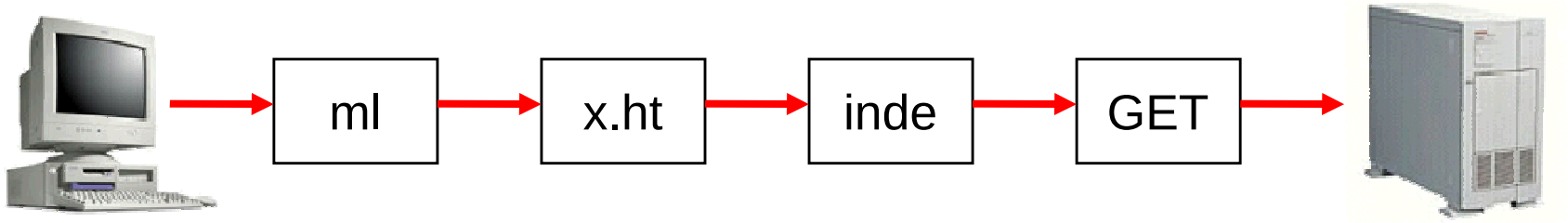


What if the Data Doesn't Fit?

Problem: Packet size

- On Ethernet, max IP packet is 1500 bytes
- Typical Web page is 10 kbytes

Solution: Split the data across multiple packets



Protocols: Layers

Network-layer packets consist of a header and a data field

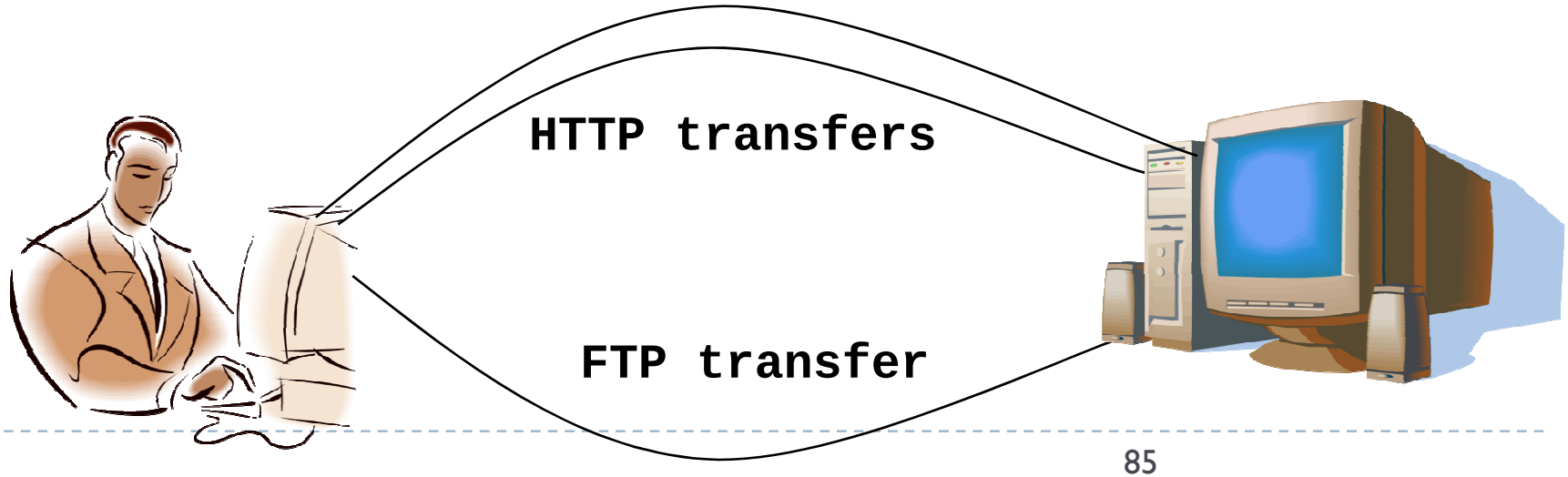
Maximum transfer unit (MTU)
MTU in IP 64 kbytes

Transport layer
Delivering messages to destinations with transport addresses
Transport address: network address + a port number
– ports are software-defined destination points at a host computer
Attached to processes

Well-known Internet services have allocated contact port numbers
HTTP (contact port: 80) and FTP (contact port: 21) – allocates a new port (with a private number)

Demultiplexing: Port Numbers

- Differentiate between multiple transfers
  - Knowing source and destination host is not enough
  - Need an id for *each transfer* between the hosts
- Specify a particular service running on a host
  - E.g., HTTP server running on port 80
  - E.g., FTP server running on port 21



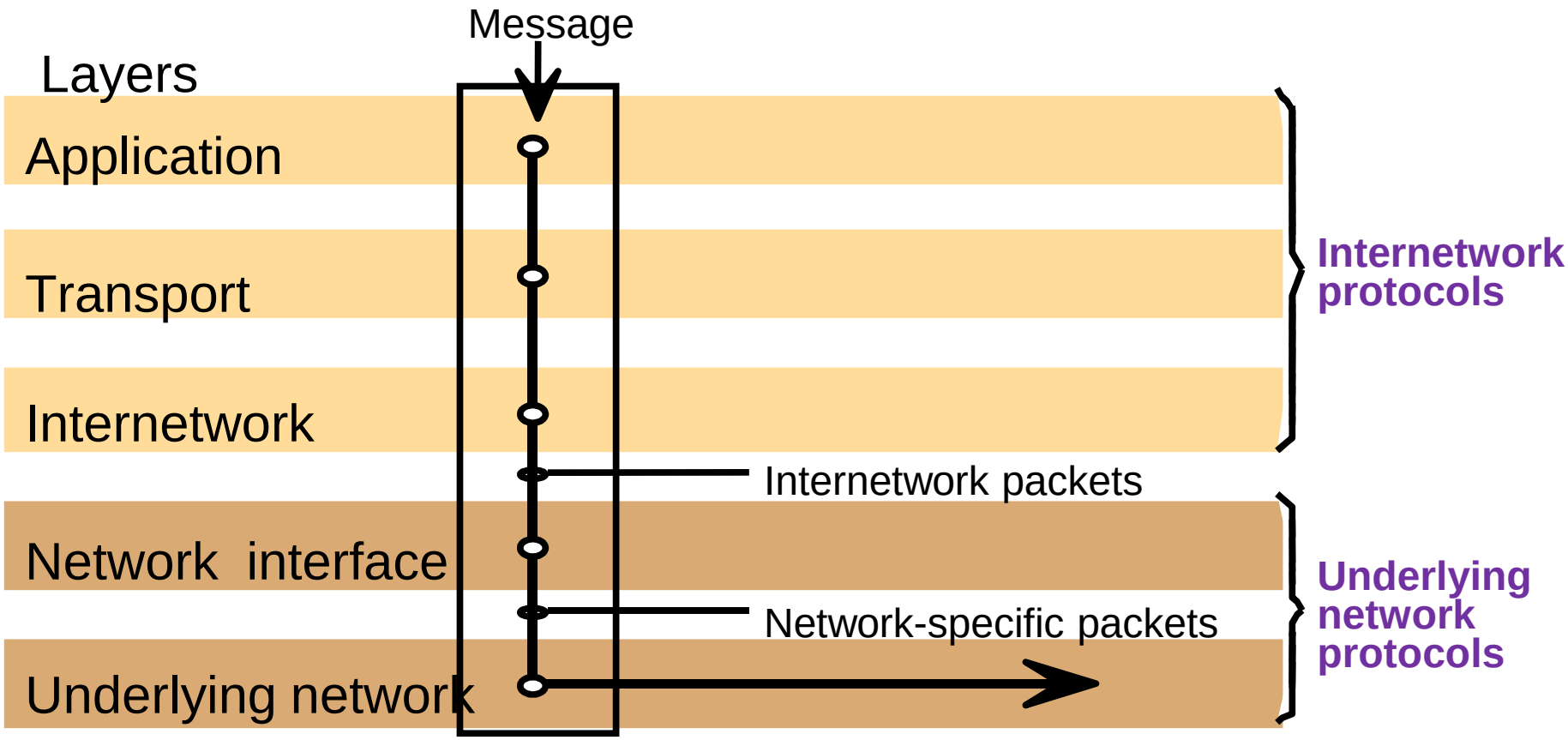
Protocols: Layers

Internet does not follow the OSI

Application, presentation and session layers are not clearly distinguished
Session layer is integrated wit the transport layer

Internetwork protocol suite:
- an application layer,
- a transport layer and
- an internetwork layer (virtual network – internetwork packet is the unit of data transmitted over an internetwork)

Protocols: Layers



Internet layers

Application = application + presentation
Transport = transport + session

Protocols: Layers

- Layer N may duplicate lower level functionality
  - E.g., error recovery to retransmit lost data
- Layers may need same information
  - E.g., timestamps, maximum transmission unit size
- Strict adherence to layering may hurt performance
  - E.g., hiding details about what is really going on
- Some layers are not always cleanly separated
  - Inter-layer dependencies for performance reasons
  - Some dependencies in standards (header checksums)
- Headers start to get really big
  - Sometimes more header bytes than actual content

Network Principles

1. Packet Switching
2. Routing
3. Protocols
4. Internetworking

Internetworking

Integrate many subnets

1. A unified *internetworking addressing scheme* that enables packets to be addressed to any host connected to any subnet (**IP addresses**)

2. A *protocol* defining the format of internetwork packets and giving rules according to which they are handled (**IP Protocol**)

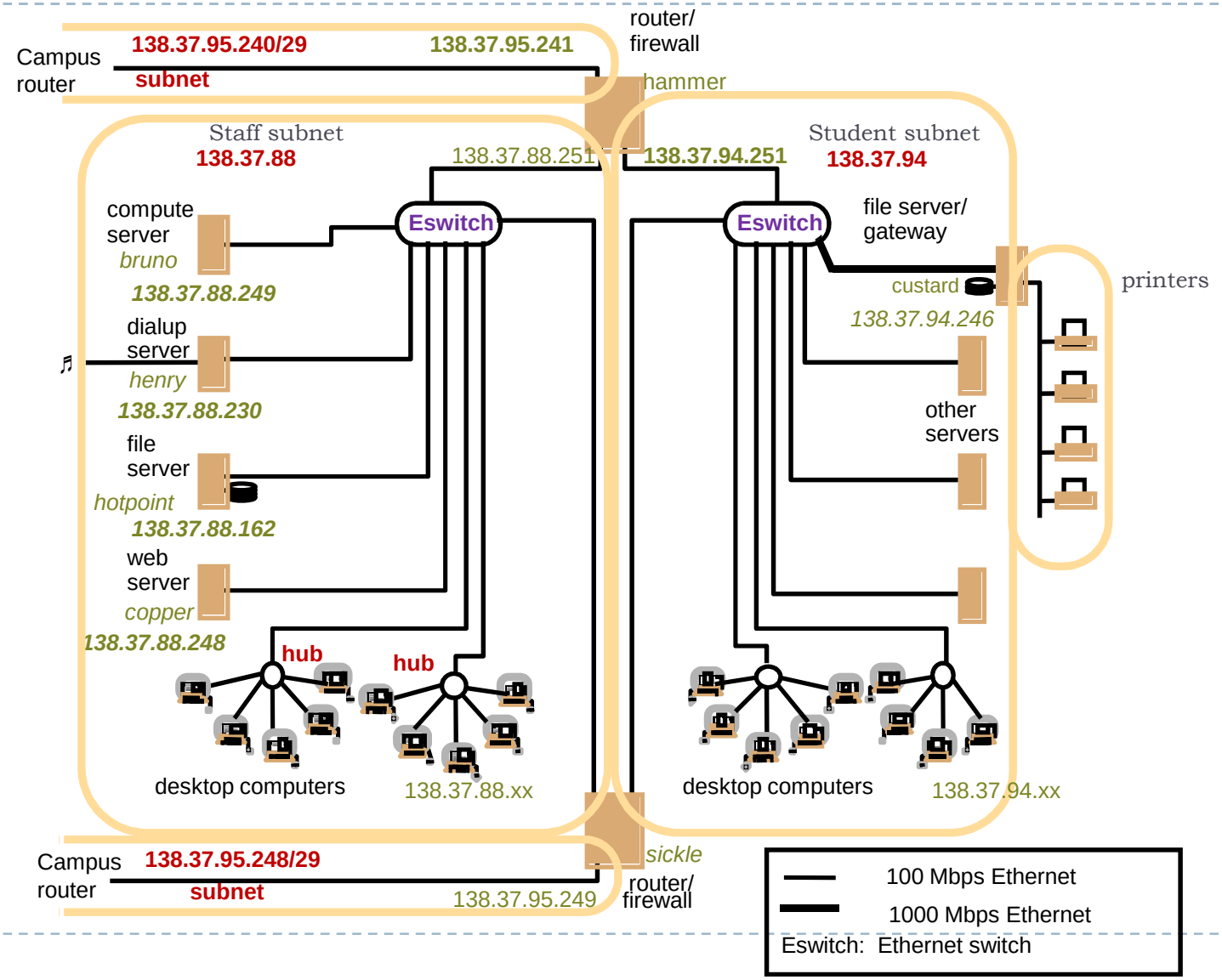
3. *Internetworking components* that route packets to their destinations (**Internet routers**)

Internetworking: Routers

5 subnets
IP addresses
3 share the IP address 138.37.95 (CIDR)

Routers (hammer, sickle) general purpose computers members of multipel subets
an IP for each

Internetworking: Routers



Internetworking: Routers

- Intranet
  - many elements in one administrative domain
- Internet
  - collection of interconnected networks, across administrative domains.
- Host
  - a computer on the net
- Router
  - a host that routes packets from one link to another at IP level. Often dedicated with no applications. Maintain routing tables. Linked to each other by direct connections or may be interconnected through subnets
- Switch
  - perform similar function to routers but *for local networks* only. They interconnect separate Ethernets, routing the incoming packets to the appropriate outgoing network.

Internetworking: Routers

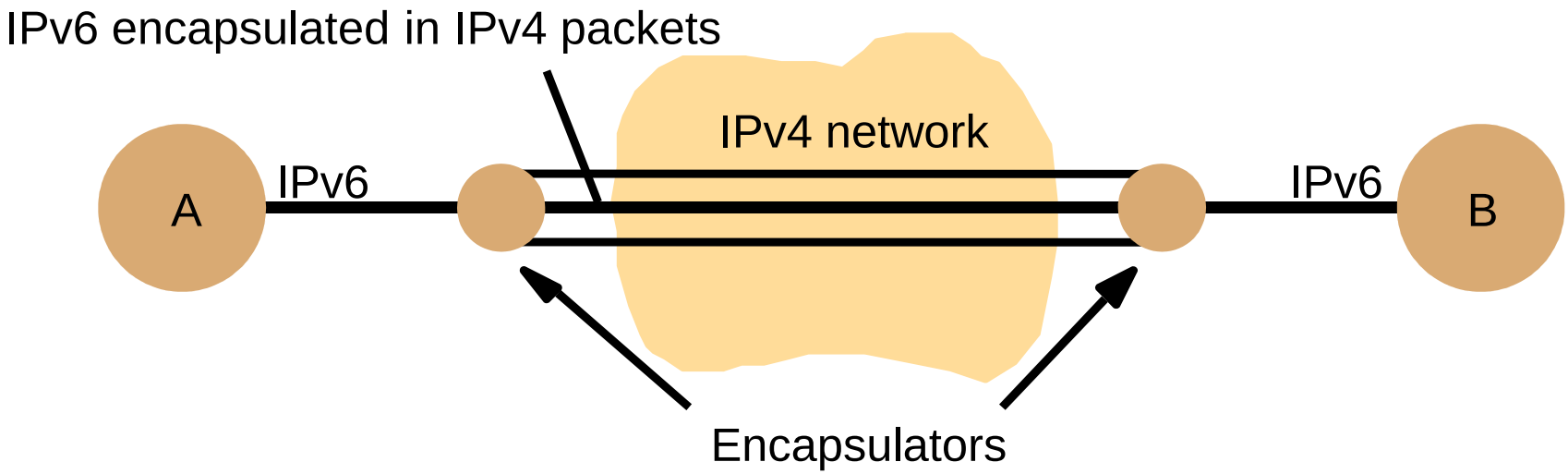
- Hubs:
  - connect several computers together in a network, extend segments of Ethernet, and allow various computers to communicate with each other. Have a number of sockets to which a host computer may be connected (to support many segments)
- Bridges:
  - links different LANs together in a WAN.
- Gateways:
  - connect different types of networks. Can be configured to enable security. Used by networks as their primary link to Internet.
- Repeaters:
  - signal amplifiers.
- Network Access Points (NAPs):
  - connect regional midlevel networks to each other.

Internetworking: Routers

- Network connecting devices
  - Hubs: extending a segment of LAN (broadcast)
  - Switches: switching traffic at data-link level (different segments of a LAN), making temporary hardware connections between two ports (or store and forward) [switches do not exchange info with each other]
  - Routers: routing traffic at IP level
  - Bridges: linking networks of different types, could be routers as well

Internetworking: Routers

- Tunneling
  - communicate through an "alien" protocol
  - “Hide” in the payload
  - IPv6 traffic using IPv4 protocols



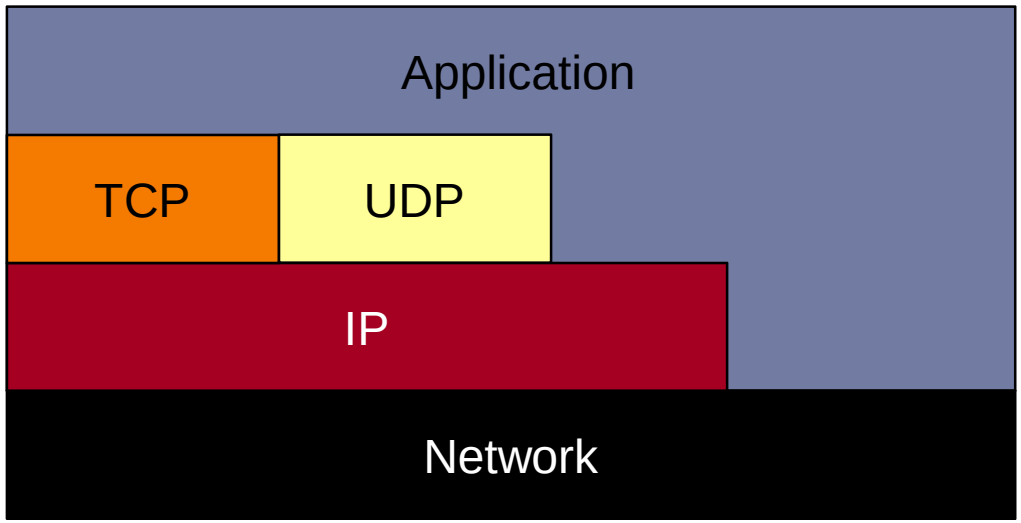
Lecture Outline

- Main points of Lecture 2
- Networking and Network Protocols
- PART II: The Internet Protocols and IP

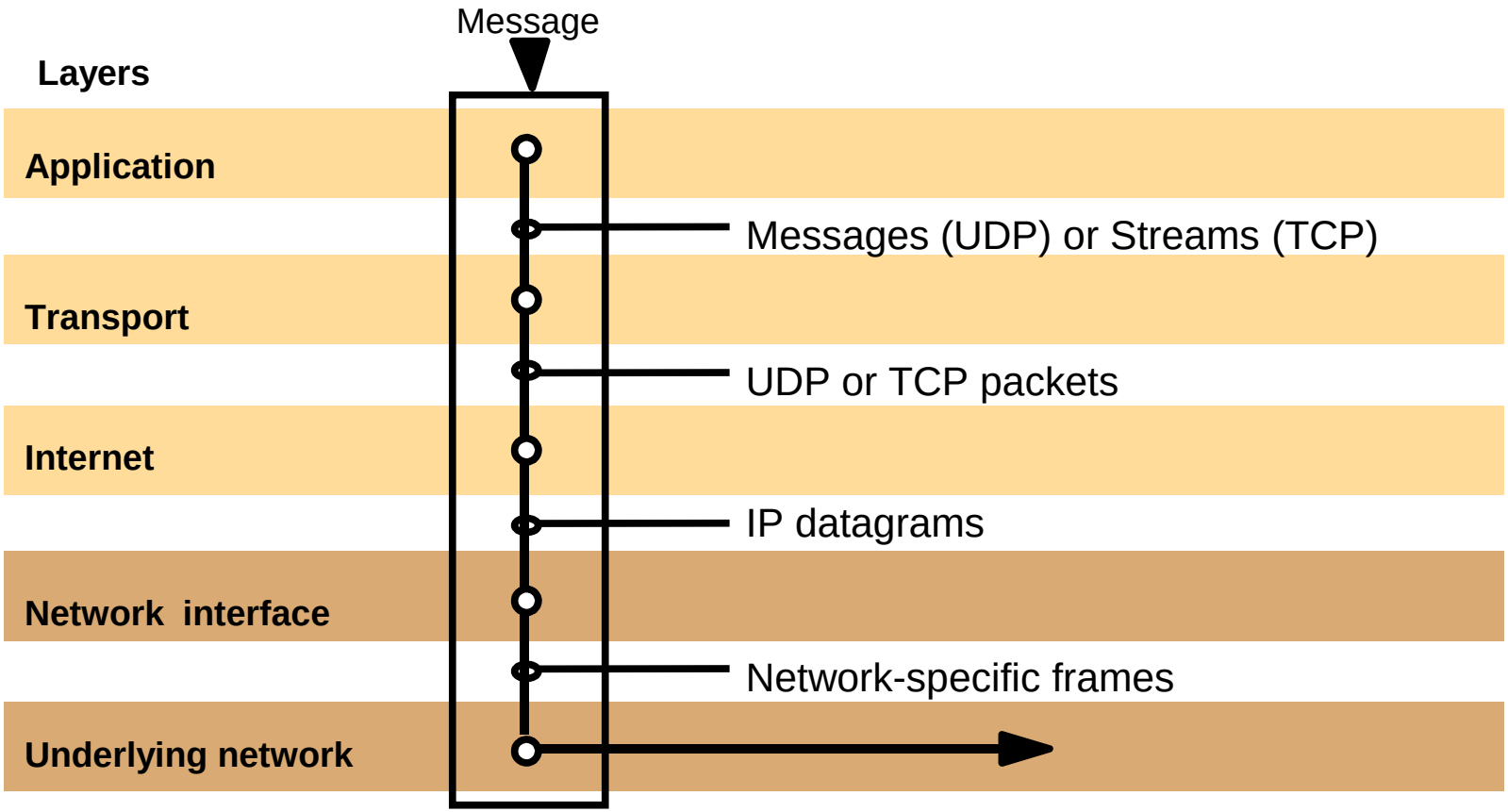
Internet Protocols

Internet Protocols

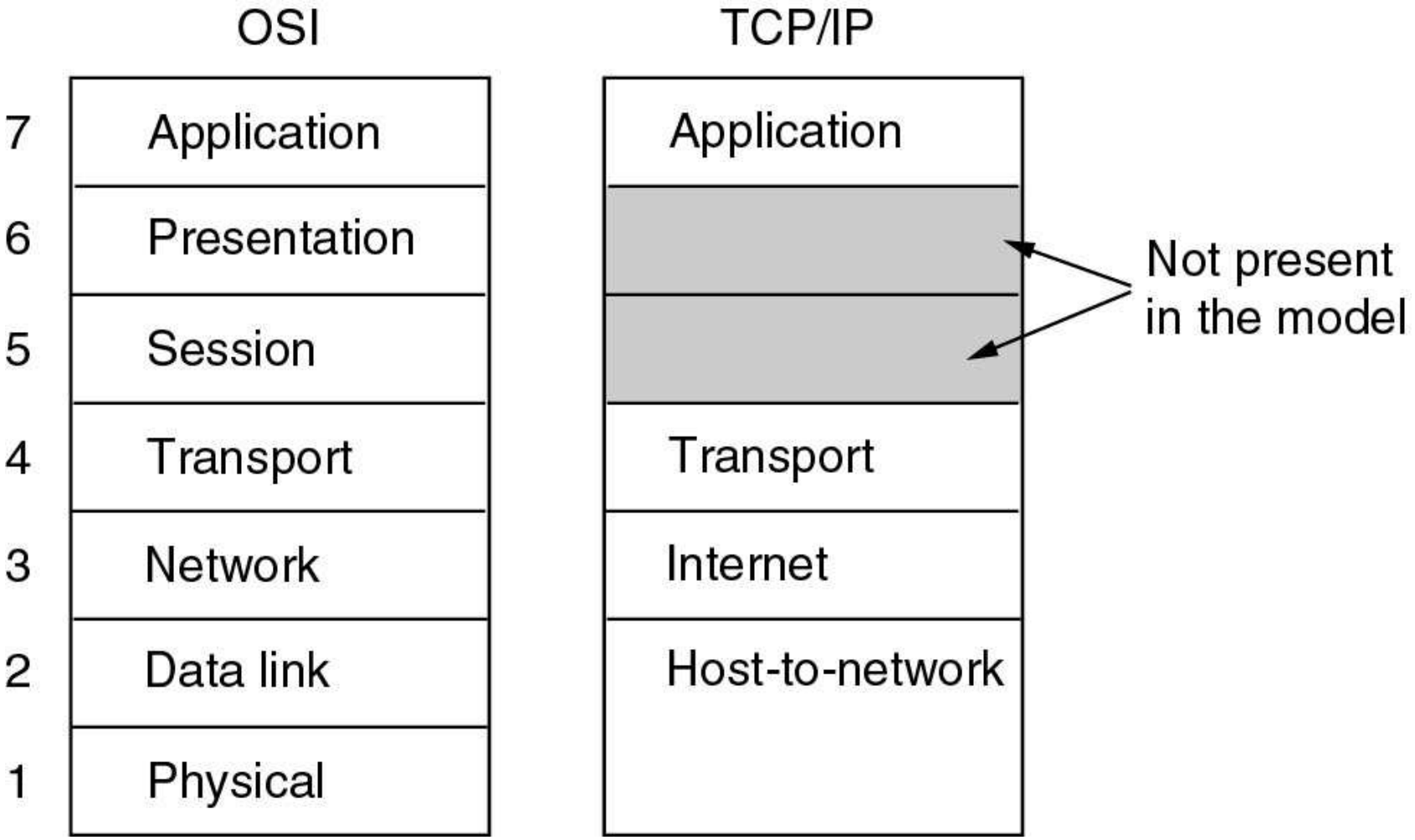
- IP (Internet Protocol)
  - "network" layer protocol
  - IP addresses
- TCP (Transmission Control Protocol)
  - transport layer
  - connection-oriented
- UDP (User Datagram Protocol)
  - transport layer
  - connection-less



Internet Protocols

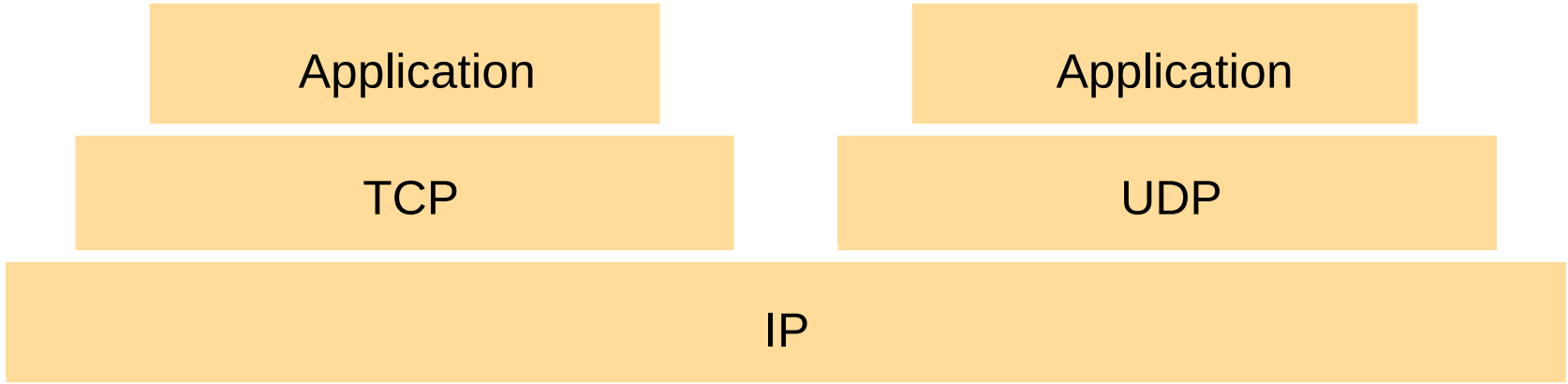


Internet Protocols

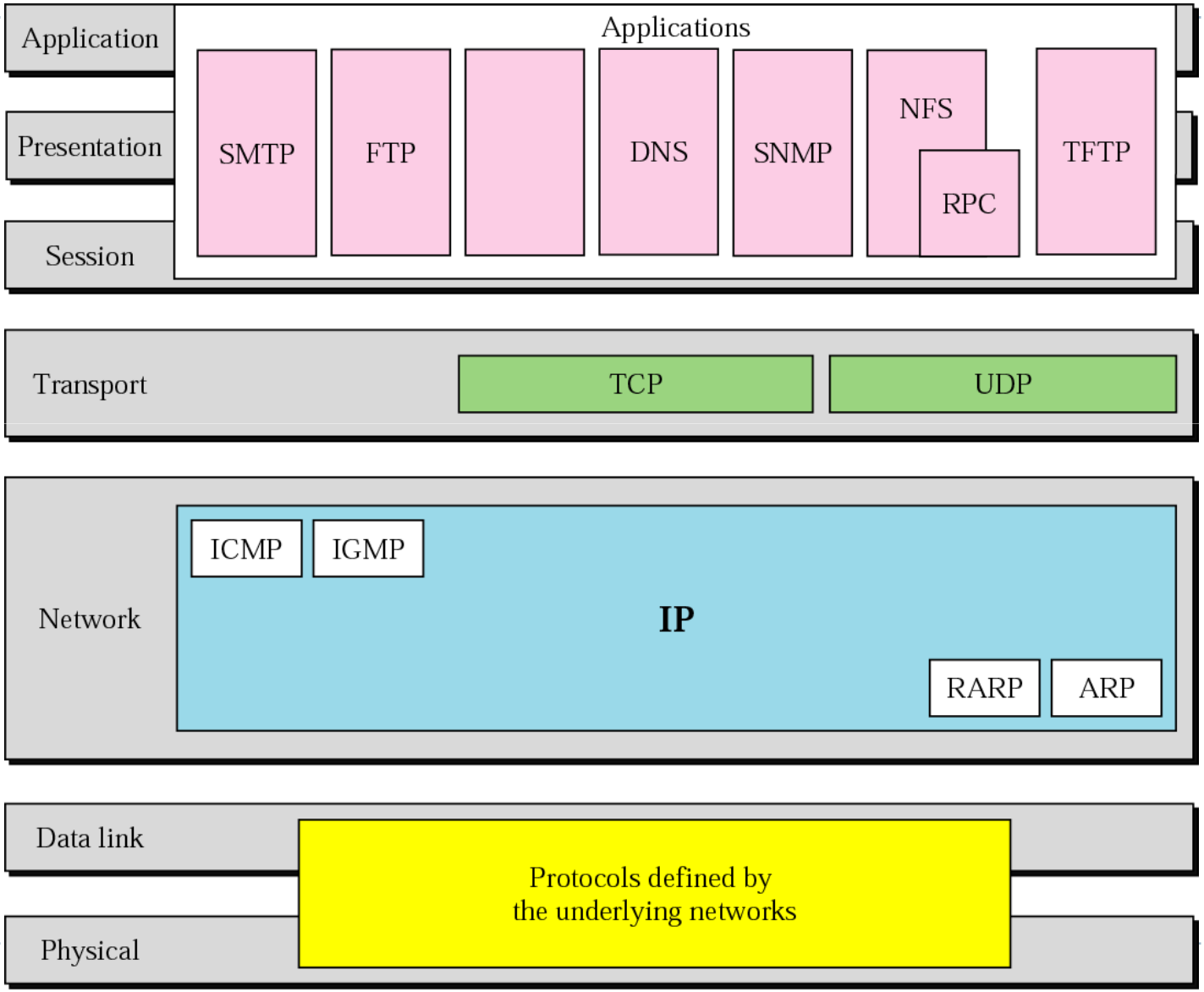


Internet Protocols

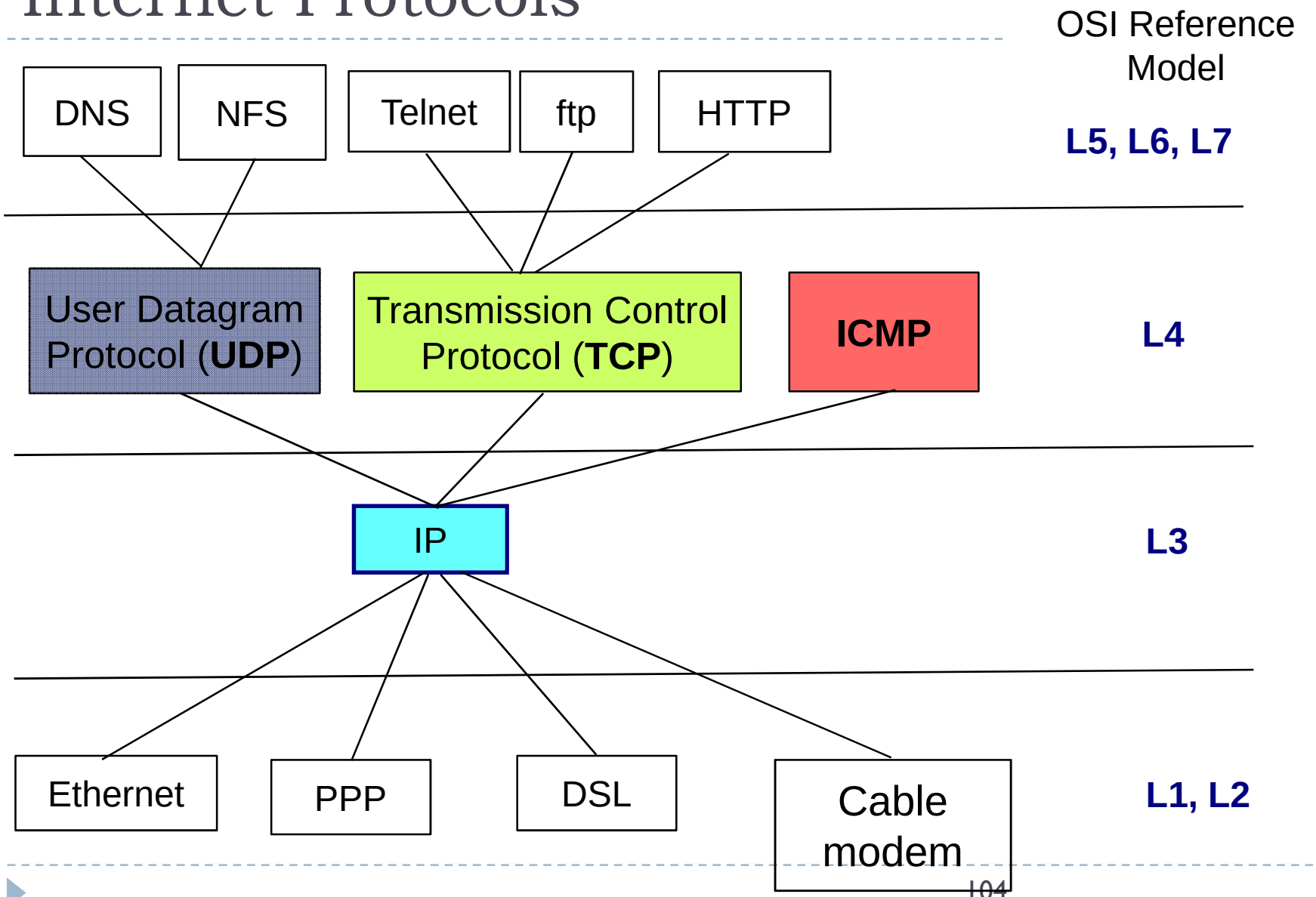
Transport protocols: TCP (reliable connection-oriented) UDP (datagram protocol that does not guarantee reliable delivery)
IP underlying “network” protocol



Internet Protocols



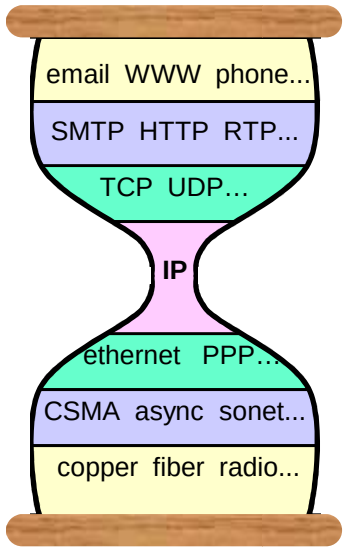
Internet Protocols



Internet Protocols

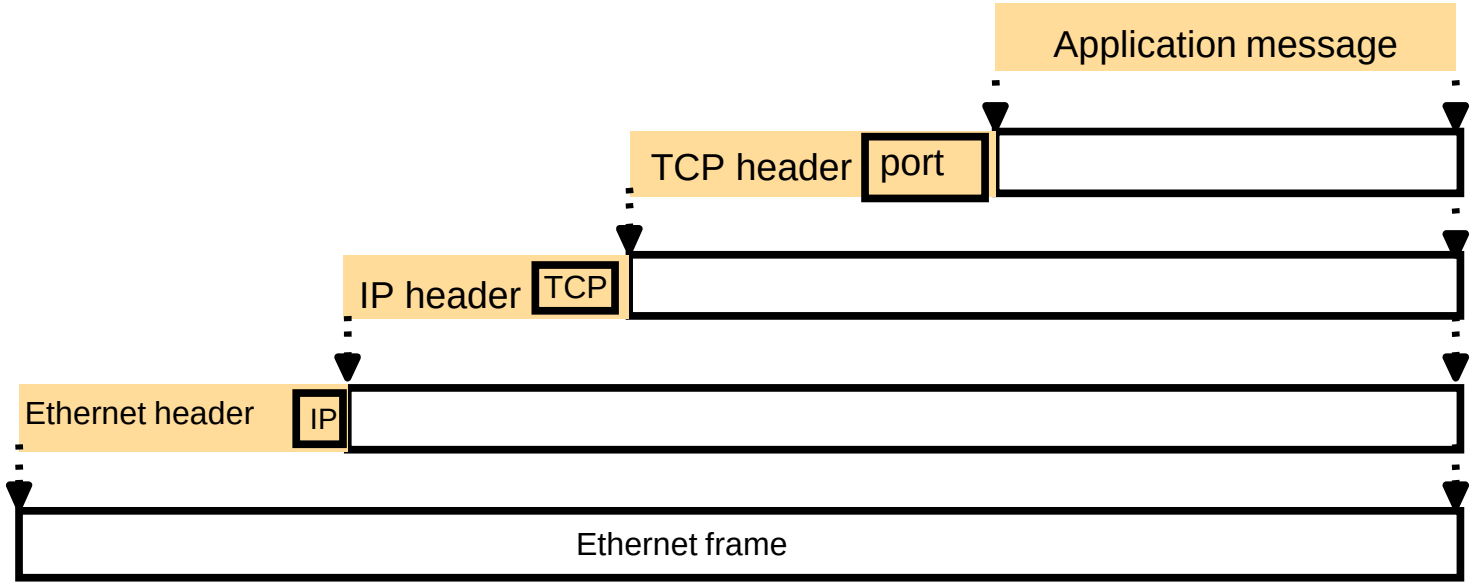
Why the Hourglass Architecture?

- Why an internet layer?
  - make a bigger network
  - global addressing
  - virtualize network to isolate end-to-end protocols from network details/changes
- Why a *single* internet protocol?
  - maximize interoperability
  - minimize number of service interfaces
- Why a *narrow* internet protocol?
  - assumes least common network functionality
    to maximize number of usable networks

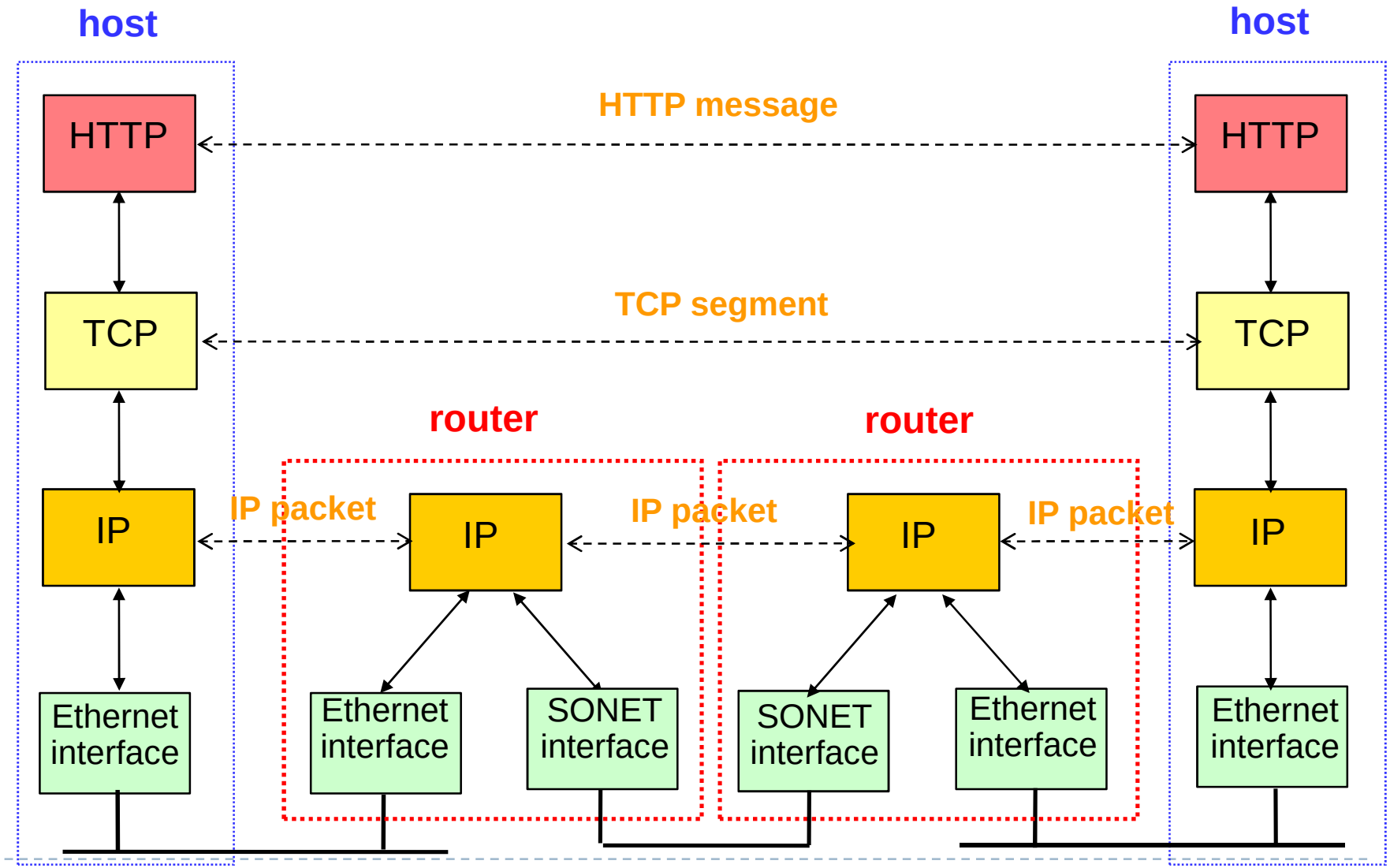


Source: Hoffman & Beaumont

Internet Protocols: encapsulation



IP Suite: End Hosts vs. Routers



Internet Protocols

- The popularity of TCP/IP is due to a number of important features:
  - Open protocol standards: freely available and developed independently from any computer hardware or OS.
  - Independence from specific physical network hardware.
  - A common addressing scheme.
  - Standardized high-level protocols.
- TCP/IP standards and protocols are published publicly as *Requests for Comments* (RFCs).

IP Overview

Internetworking

Integrate many subnets

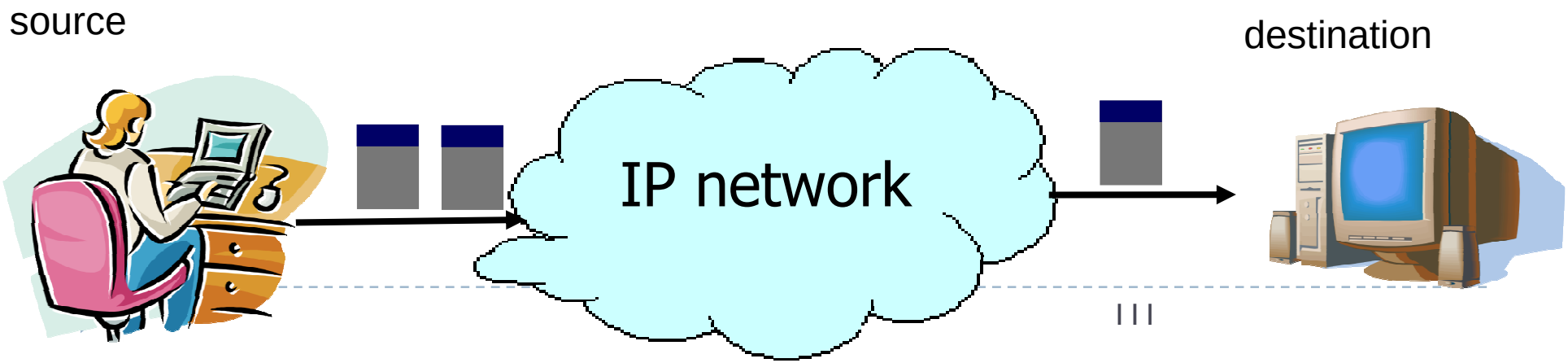
1. A unified *internetworking addressing scheme* that enables packets to be addressed to any host connected to any subnet (**IP addresses**)

2. A *protocol* defining the format of internetwork packets and giving rules according to which they are handled (**IP Protocol**)

3. *Internetworking components* that route packets to their destinations (**Internet routers**)

IP Service: Best-Effort Packet Delivery

- Packet switching
  - Divide messages into a sequence of packets
  - Headers with source and destination address
- Best-effort delivery
  - Packets may be lost
  - Packets may be corrupted
  - Packets may be delivered out of order



IP Service Model: Why Packets?

- Data traffic is bursty
  - Logging in to remote machines
  - Exchanging e-mail messages
- Don't want to waste reserved bandwidth
  - No traffic exchanged during idle periods
- Better to allow multiplexing
  - Different transfers share access to same links
- Packets can be delivered by almost anything
  - RFC 2549: IP over Avian Carriers (aka birds)
- … still, packet switching can be inefficient
  - Extra header bits on every packet

IP Service Model: Why Best-Effort?

- IP means never having to say you're sorry…
  - Don't need to reserve bandwidth and memory
  - Don't need to do error detection & correction
  - Don't need to remember from one packet to next
- Easier to survive failures
  - Transient disruptions are okay during fail-over

- … but, applications *do* want efficient, accurate transfer of data in order, in a timely fashion

IP Service: Best-Effort is Enough?

- No error detection or correction
  - Higher-level protocol can provide error checking
- Successive packets may not follow the same path
  - Not a problem as long as packets reach the destination
- Packets can be delivered out-of-order
  - Receiver can put packets back in order (if necessary)
- Packets may be lost or arbitrarily delayed
  - Sender can send the packets again (if desired)
- No network congestion control (beyond "drop")
  - Sender can slow down in response to loss or delay

History: Why IP Packets?

- IP proposed in the early 1970s
  - Defense Advanced Research Project Agency (DARPA)
- Goal: connect existing networks
  - To develop an effective technique for multiplexed utilization of existing interconnected networks
  - E.g., connect packet radio networks to the ARPAnet
- Motivating applications
  - Remote login to server machines
  - Inherently bursty traffic with long silent periods
- Prior ARPAnet experience with packet switching
  - Previous DARPA project
  - Demonstrated store-and-forward packet switching

Other Main Driving Goals (In Order)

- Communication should continue despite failures
  - Survive equipment failure or physical attack
  - Traffic between two hosts continue on another path
- Support multiple types of communication services
  - Differing requirements for speed, latency, & reliability
  - Bidirectional reliable delivery vs. message service
- Accommodate a variety of networks
  - Both military and commercial facilities
  - Minimize assumptions about the underlying network

Other Driving Goals, Somewhat Met

- Permit distributed management of resources
  - Nodes managed by different institutions
  - … though this is still rather challenging
- Cost-effectiveness
  - Statistical multiplexing through packet switching
  - … though packet headers and retransmissions wasteful
- Ease of attaching new hosts
  - Standard implementations of end-host protocols
  - … though still need a fair amount of end-host software
- Accountability for use of resources
  - Monitoring functions in the nodes
  - … though this is still fairly limited and immature

IP Addressing

Internet Protocols: IP addressing

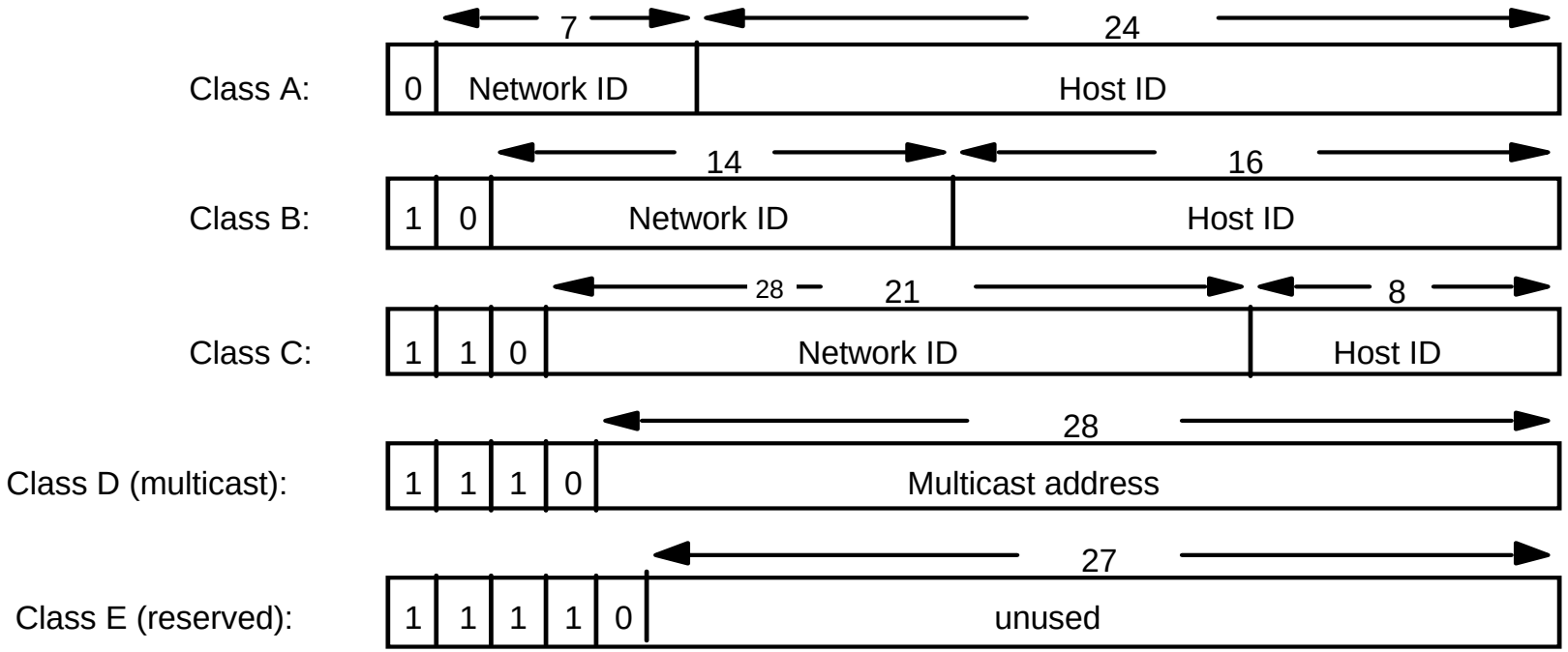
It must be

- Universal
- Efficient in its use of address space
  - 2^{32} or 4 billion
- Lead to a flexible and efficient routing scheme

IP version 4

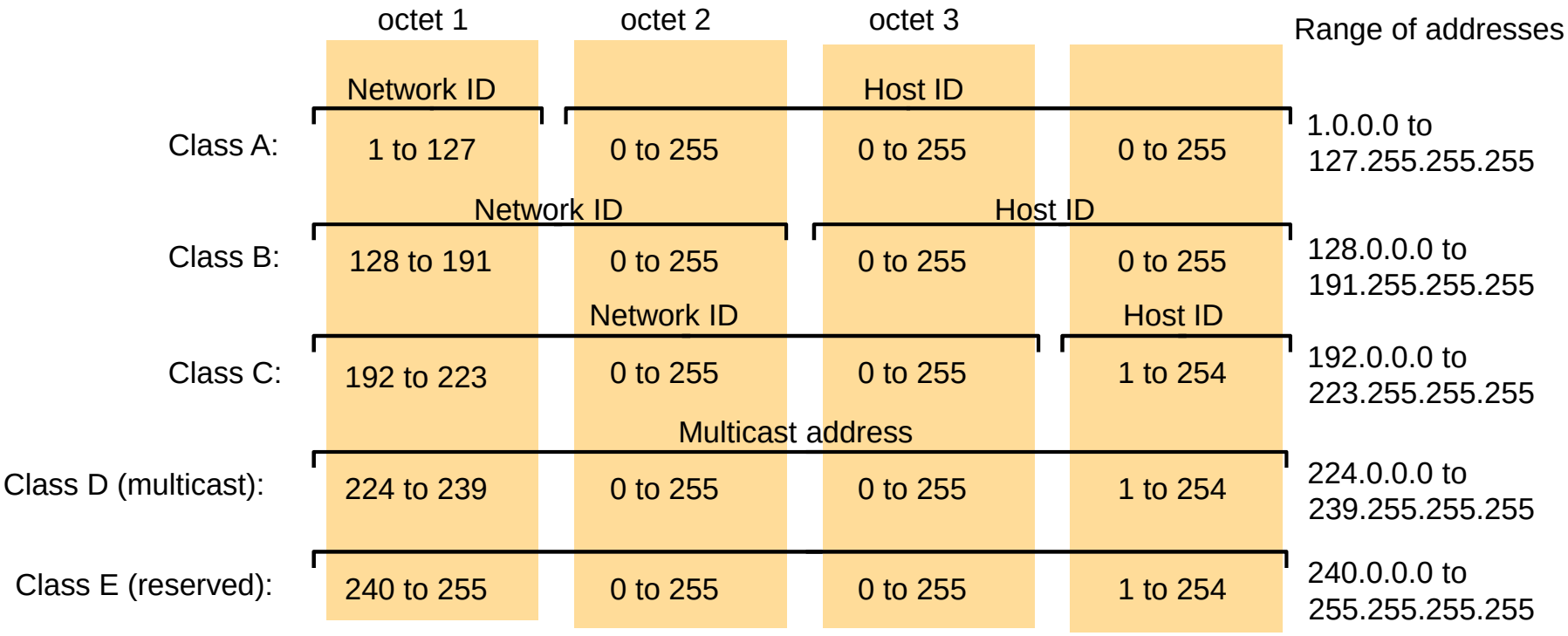
Internet Protocols: IP addressing

32 bits: Network Identifier (identifies one sub-network) + Host identifier (identifies the host connection to that network)



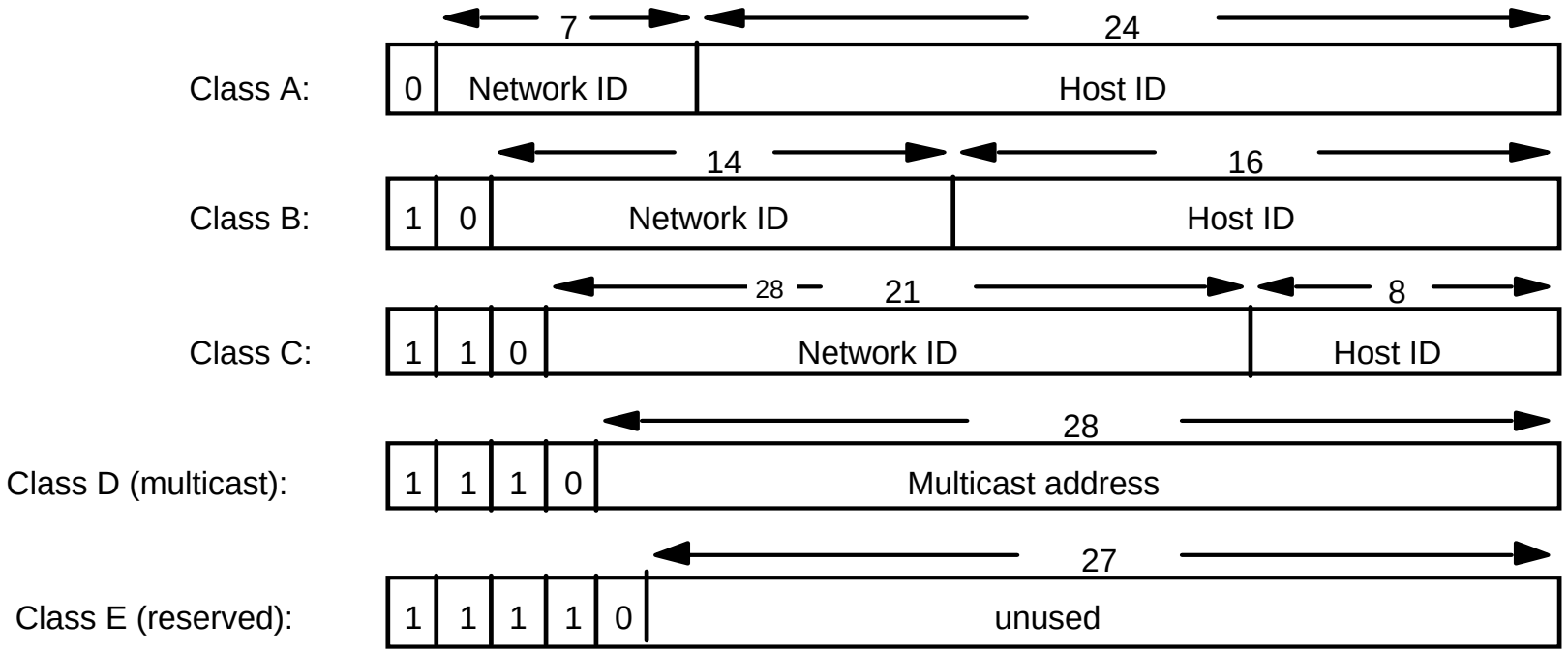
Internet Protocols: IP addressing

32 bits:
Written as sequence of four decimal bytes, one byte - octet



Internet Protocols: IP addressing

Class A 2^{24} hosts in each subnet, for very large networks
Class B 255 hosts
Class C fewer
Network identifiers assigner by IANA, host identifiers locally



Classful Addressing

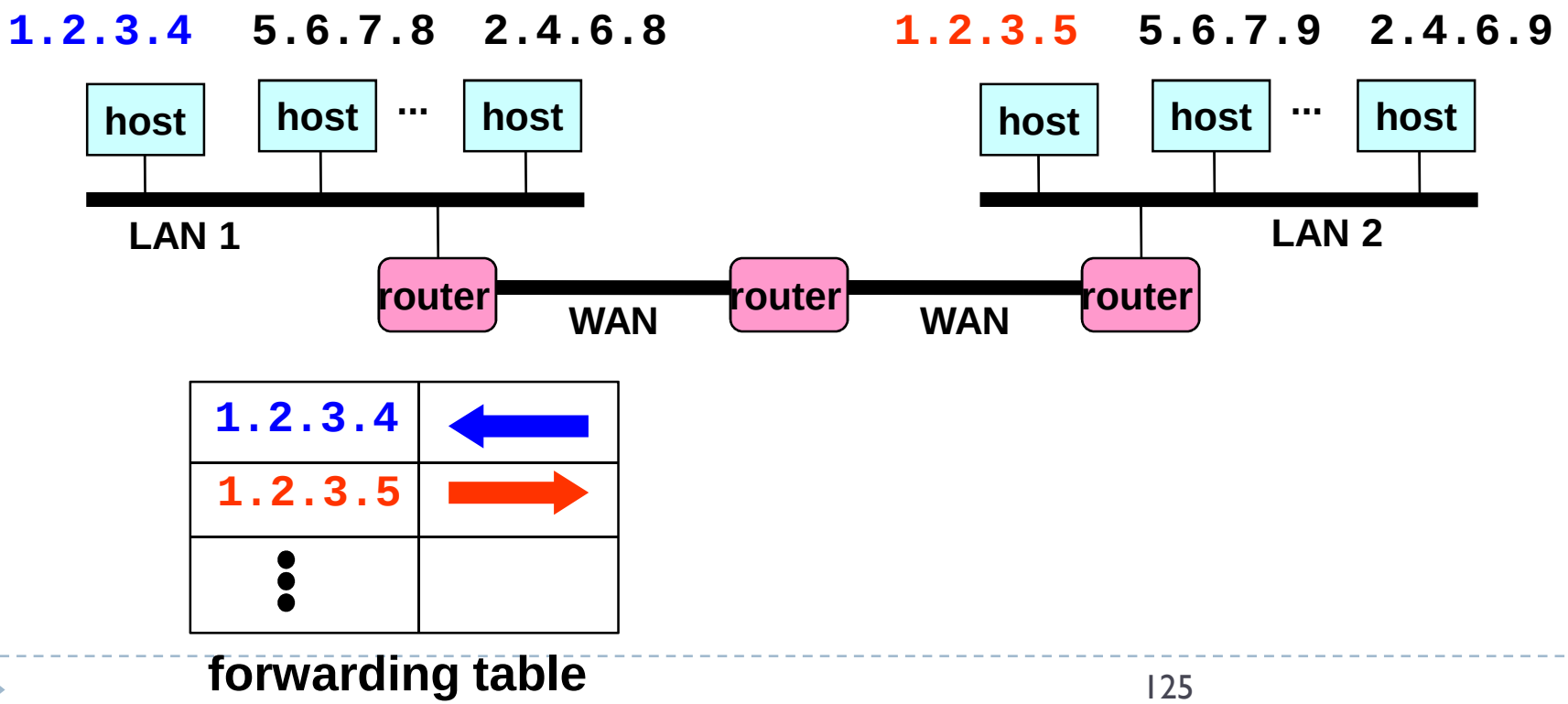
- In the older days, only fixed allocation sizes
  - Class A: 0*
    - Very large /8 blocks (e.g., MIT has 18.0.0.0/8)
  - Class B: 10*
    - Large /16 blocks (e.g,. Princeton has 128.112.0.0/16)
  - Class C: 110*
    - Small /24 blocks (e.g., AT&T Labs has 192.20.225.0/24)
  - Class D: 1110*
    - Multicast groups
  - Class E: 11110*
    - Reserved for future use

Internet Protocols: IP addressing

- Classless interdomain routing (CIDR)
  - shortage of Class B networks
  - Allocate a batch of contiguous Class C addresses to a subnet requiring more than 255 addresses (also, subdivide Class B)
  - Add a mask field (bit pattern) to indicate bits for network portion that is compared with the routing table entry
  - 138.73.59.32/22 [subnet: first 22 bits; host: 10 bits]

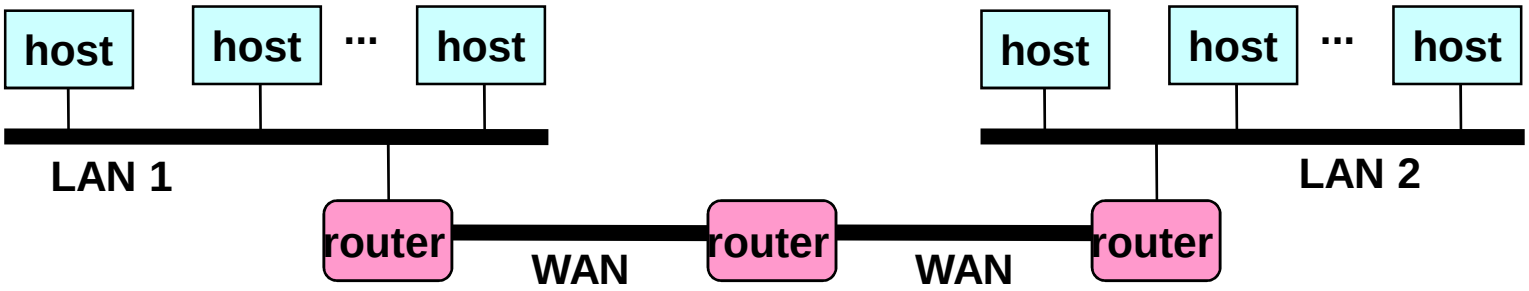
Scalability Challenge

- Suppose hosts had arbitrary addresses
  - Then every router would need a lot of information
  - …to know how to direct packets toward the host



Grouping Related Hosts

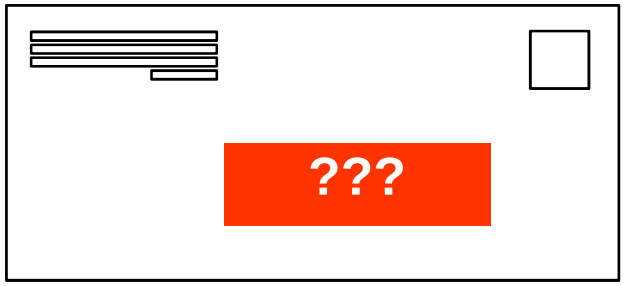
- The Internet is an "inter-network"
  - Used to connect *networks* together, not *hosts*
  - Needs a way to address a network (i.e., group of hosts)



LAN = Local Area Network
WAN = Wide Area Network

Hierarchical Addressing in U.S. Mail

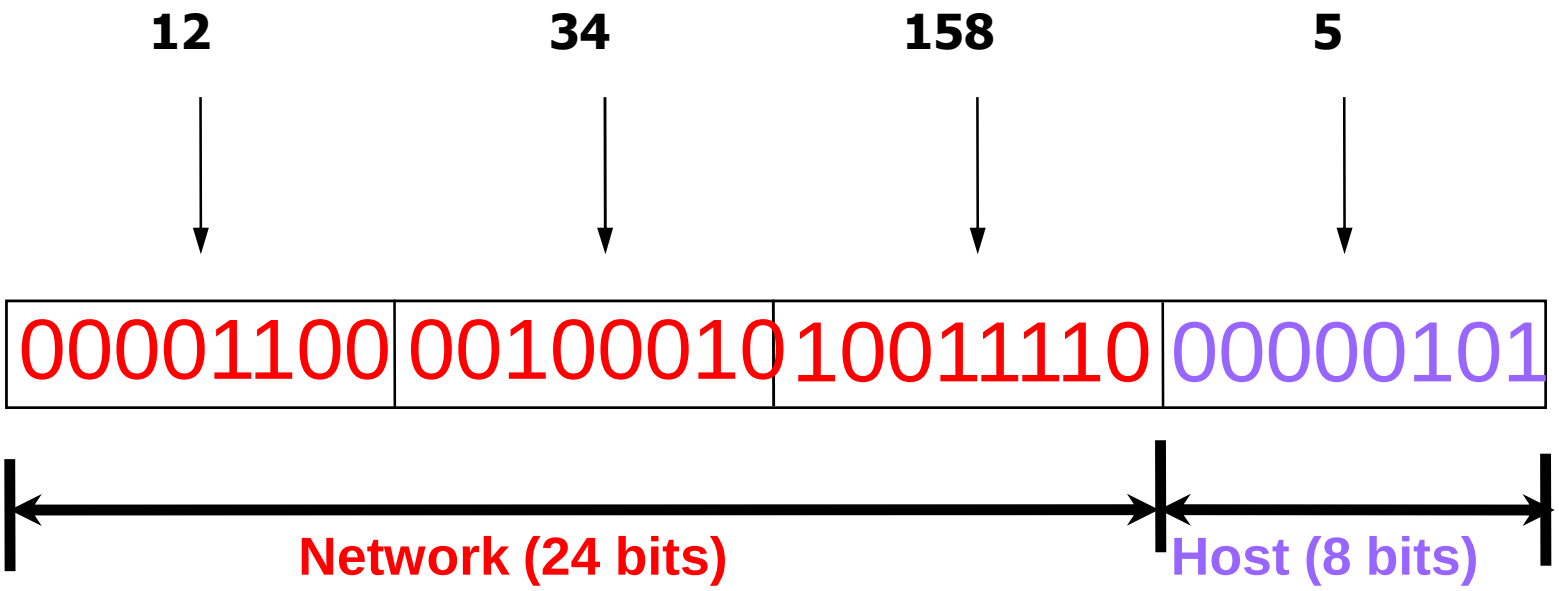
- Addressing in the U.S. mail
  - Zip code: 08540
  - Street: Olden Street
  - Building on street: 35
  - Room in building: 306
  - Name of occupant: Jennifer Rexford



- Forwarding the U.S. mail
  - Deliver letter to the post office in the zip code
  - Assign letter to mailman covering the street
  - Drop letter into mailbox for the building/room
  - Give letter to the appropriate person

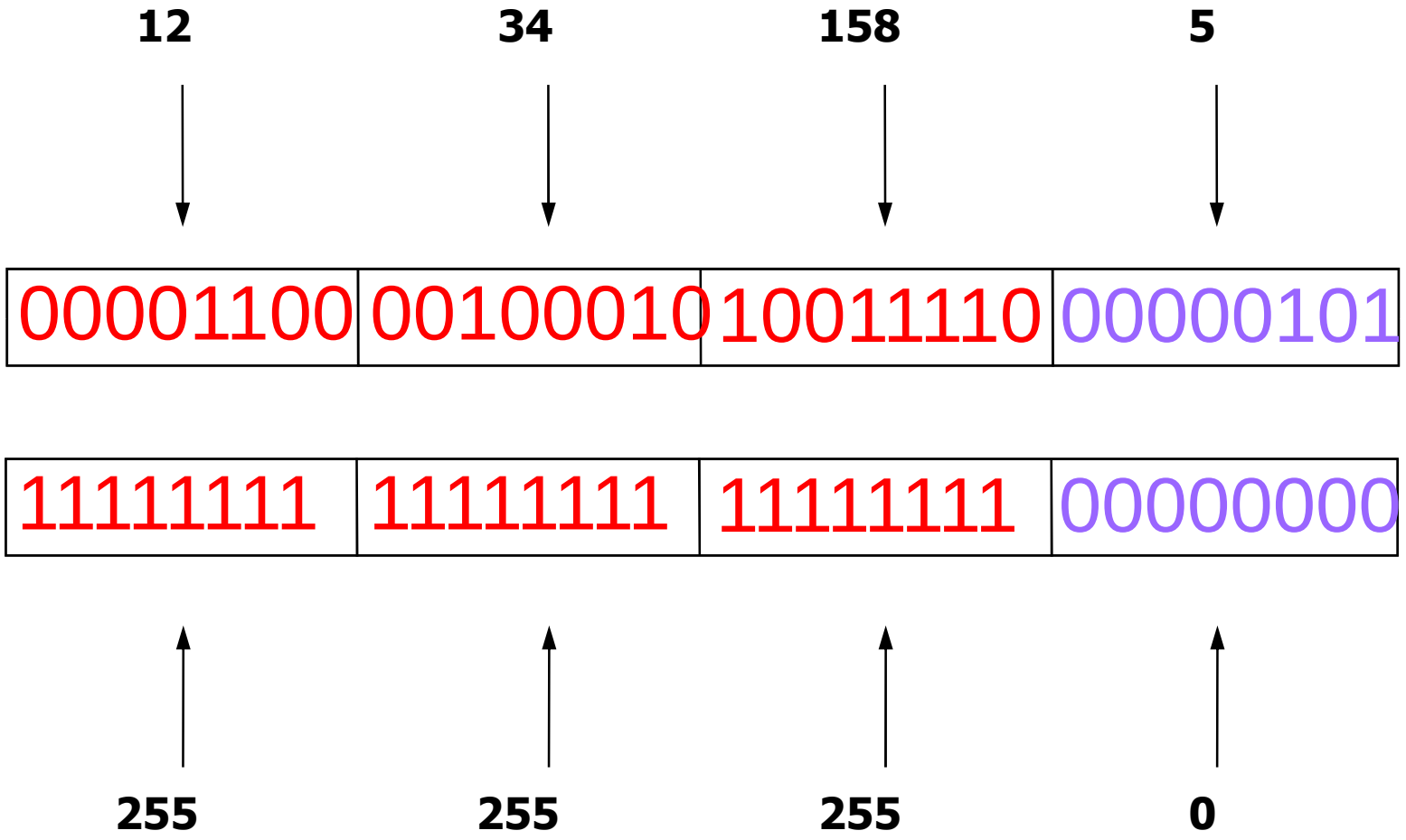
Hierarchical Addressing: IP Prefixes

- Divided into network & host portions (left and right)
- 12.34.158.0/24 is a 24-bit prefix with 2^8 addresses



IP Address and a 24-bit Subnet Mask

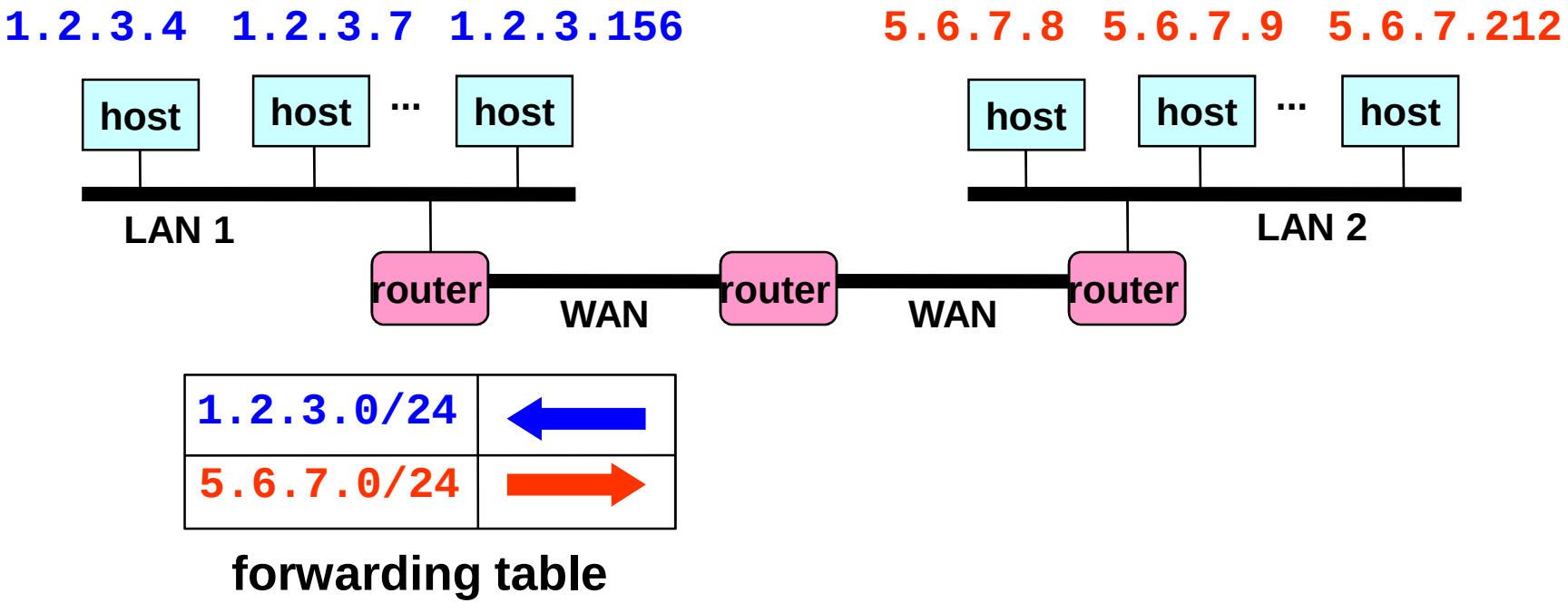
Address



Mask

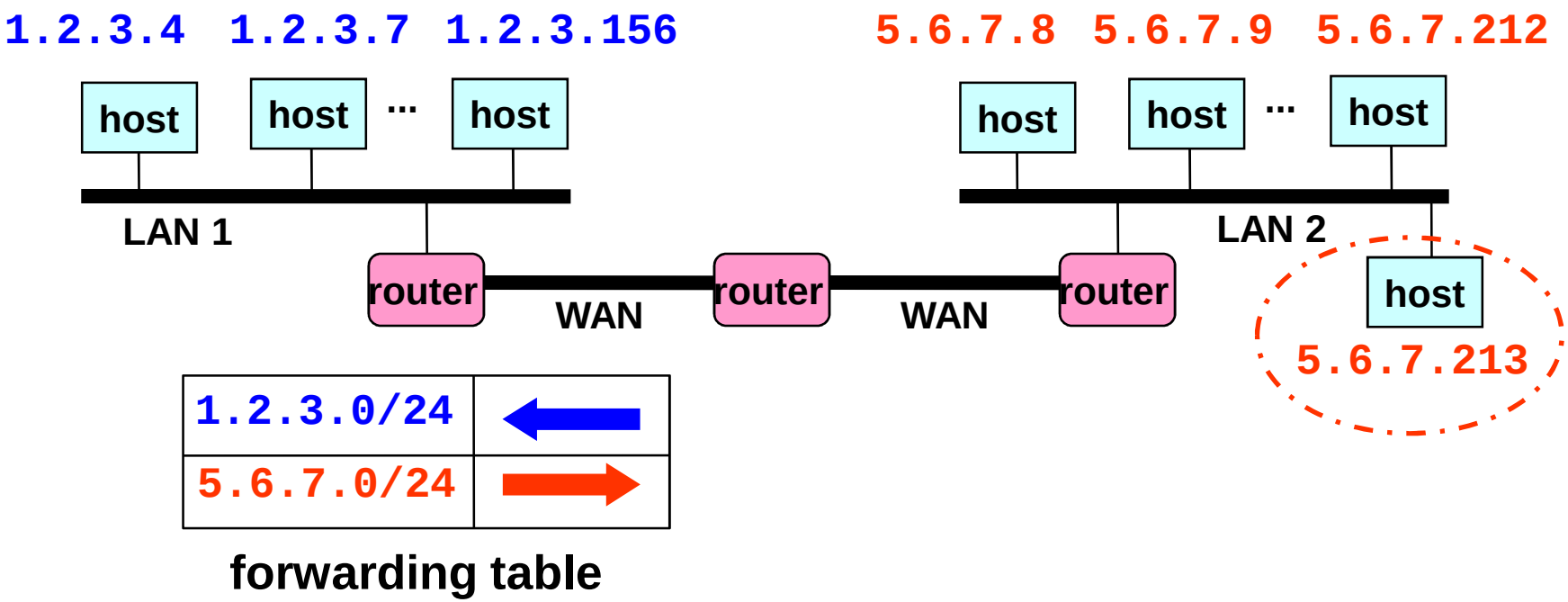
Scalability Improved

- Number related hosts from a common subnet
  - 1.2.3.0/24 on the left LAN
  - 5.6.7.0/24 on the right LAN



Easy to Add New Hosts

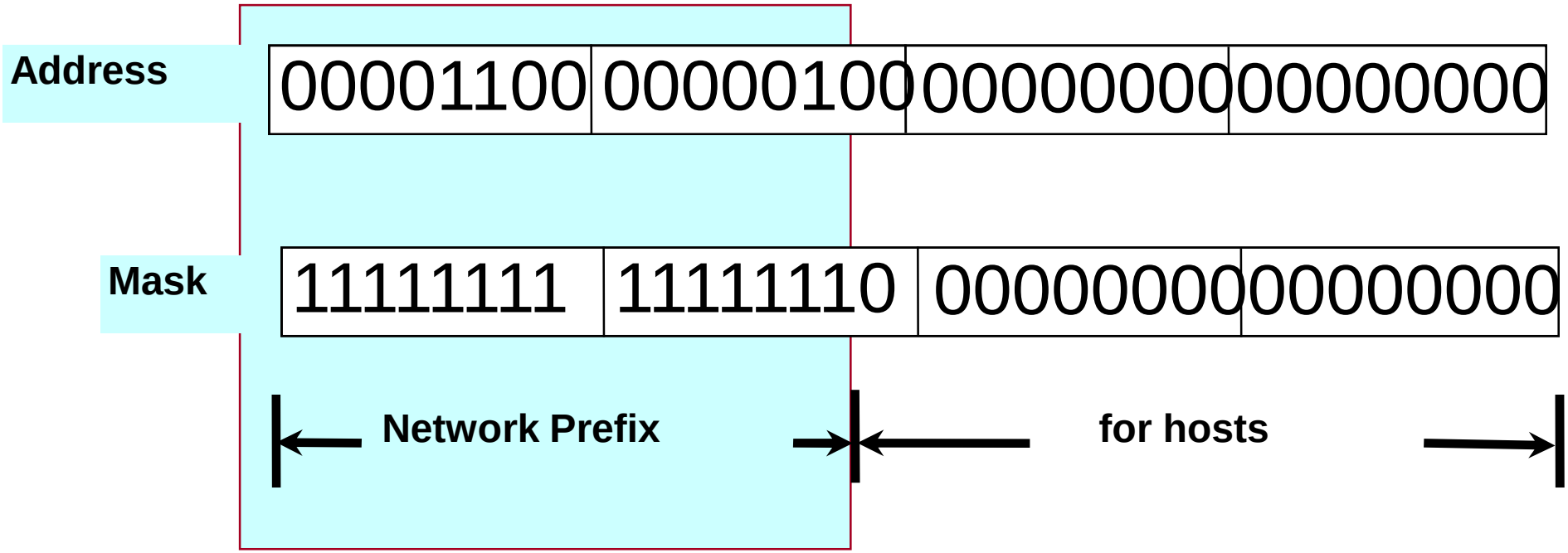
- No need to update the routers
  - E.g., adding a new host 5.6.7.213 on the right
  - Doesn't require adding a new forwarding entry



Classless Inter-Domain Routing (CIDR)

Use two 32-bit numbers to represent a network.
Network number = IP address + Mask

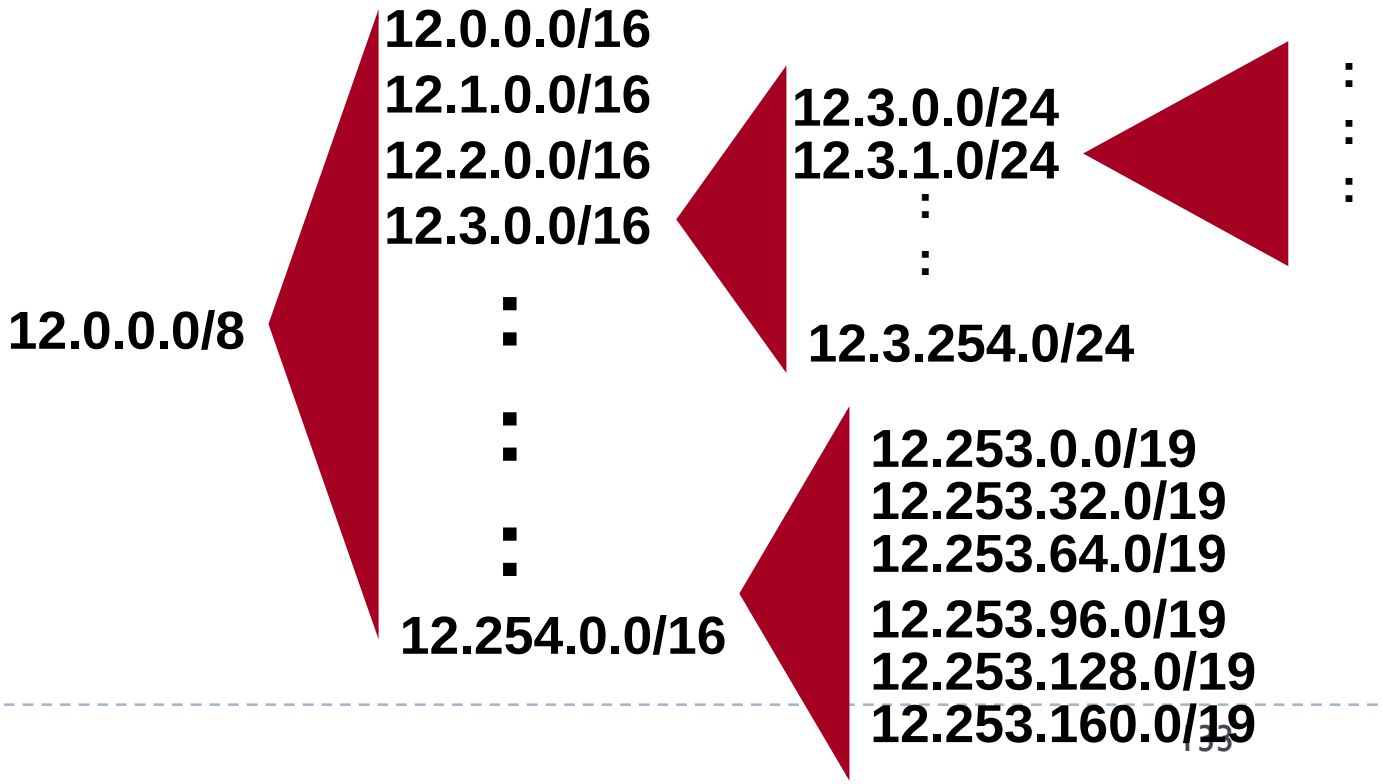
IP Address : 12.4.0.0 IP Mask: 255.254.0.0



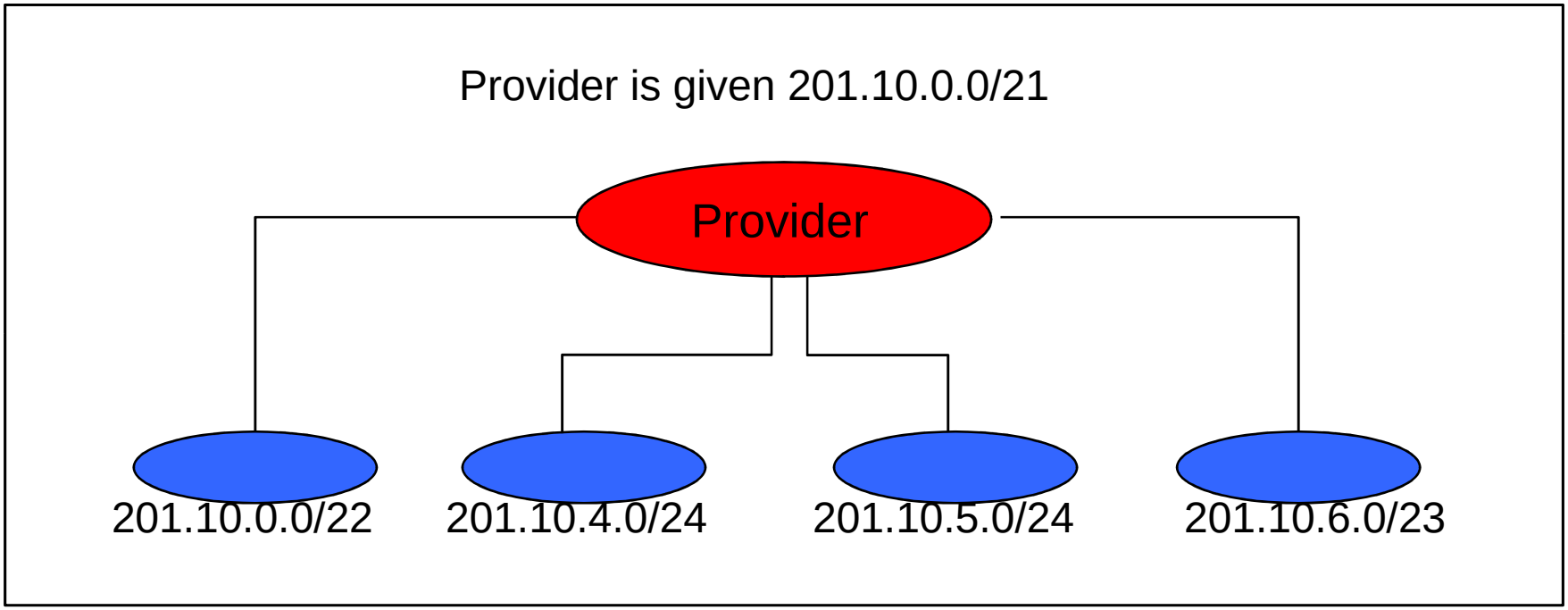
Written as 12.4.0.0/15

CIDR: Hierarchal Address Allocation

- Prefixes are key to Internet scalability
  - Address allocated in contiguous chunks (prefixes)
  - Routing protocols and packet forwarding based on prefixes
  - Today, routing tables contain ~150,000-200,000 prefixes

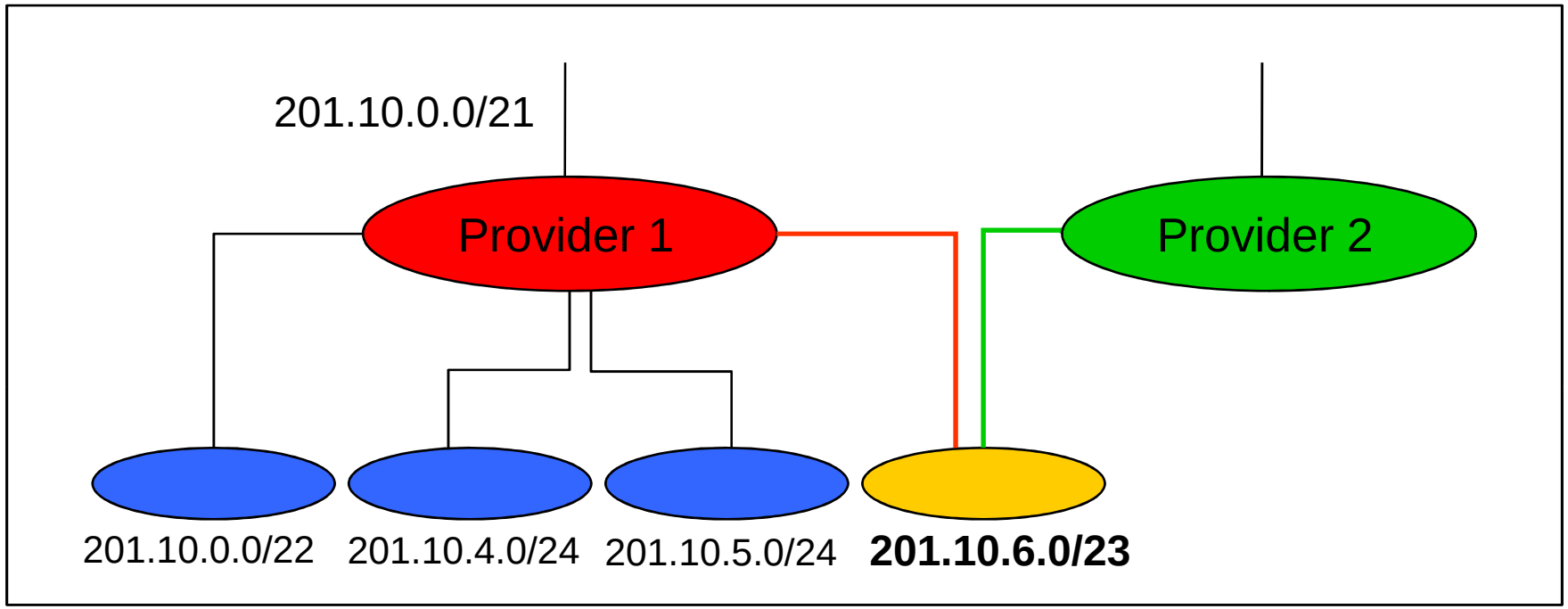


Scalability: Address Aggregation



Routers in the rest of the Internet just need to know how to reach 201.10.0.0/21. The provider can direct the IP packets to the appropriate customer.

But, Aggregation Not Always Possible



Multi-homed **customer with 201.10.6.0/23 has two providers. Other parts of the Internet need to know how to reach these destinations through *both* providers.**

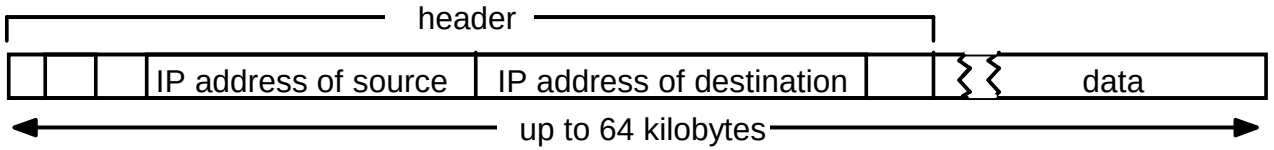
Obtaining a Block of Addresses

- Separation of control
  - Prefix: assigned to an institution
  - Addresses: assigned to nodes by the institution
- Who assigns prefixes?
  - Internet Corp. for Assigned Names and Numbers
    - Allocates large blocks to Regional Internet Registries
  - Regional Internet Registries (RIRs)
    - E.g., ARIN (American Registry for Internet Numbers)
    - Allocated to ISPs and large institutions in a region
  - Internet Service Providers (ISPs)
    - Allocate address blocks to their customers
    - Who may, in turn, allocate to their customers…

IP Protocol

Internet Protocols: The IP Protocol

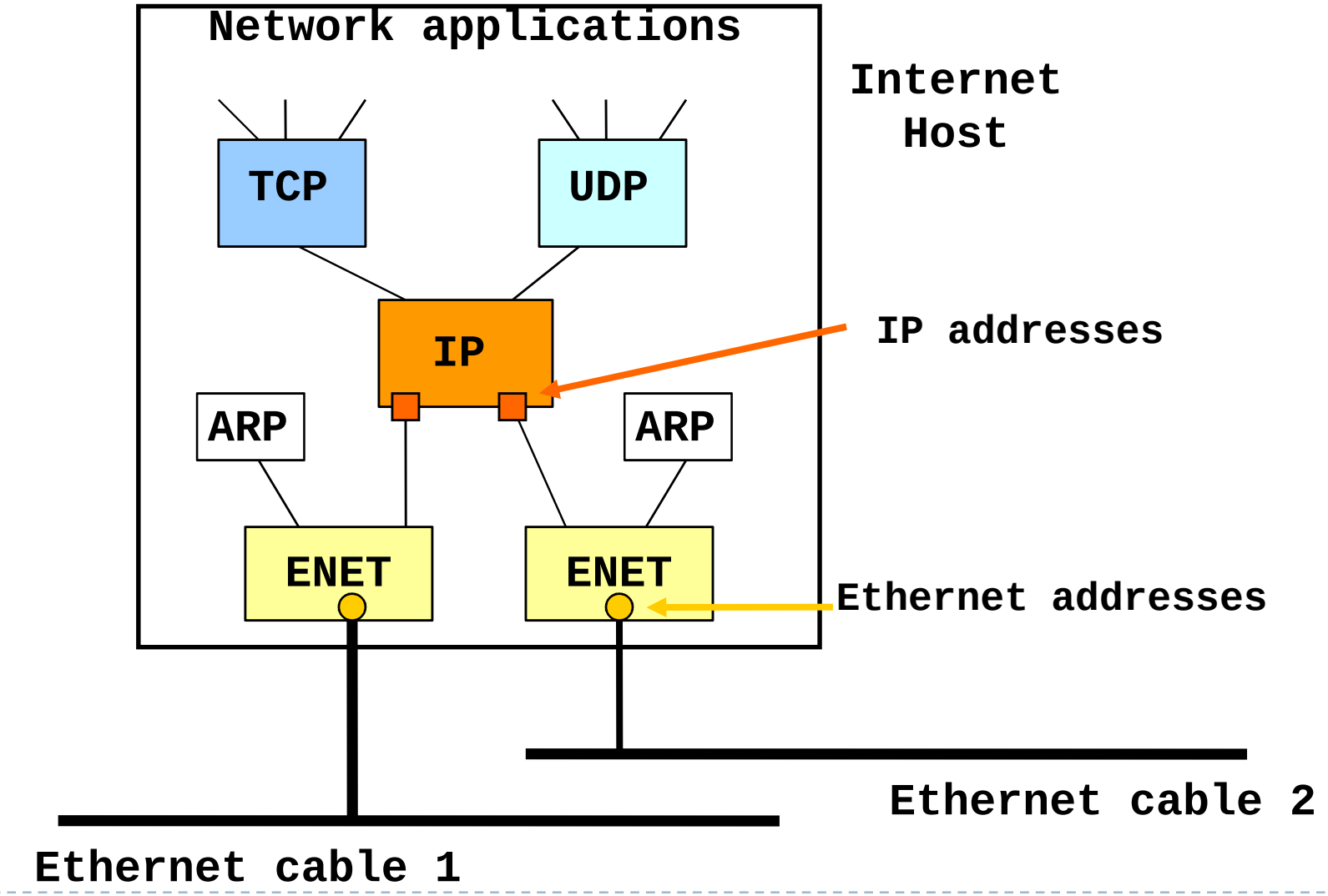
Transmits datagrams from one host to another, if necessary via intermediate routers



Provides only header checksum
Unreliable or best effort (packets may be lost, duplicated, delayed or delivered out of order)

If IP datagram is longer than MTU, it is broken into network packets
Also, inserts a "physical" network address attained through address resolution

IP Protocol


Network applications
Internet
Host
TCP
UDP
IP
IP addresses
ARP
ARP
ENET
ENET
Ethernet addresses
Ethernet cable 2
Ethernet cable 1

Internet Protocols: The IP Protocol

Address Resolution Protocol (APR)

converts *Internet addresses* to *network addresses* for a specific underlying network (physical addresses)
e.g., 32-bit Internet addresses to 48-bit Ethernet addresses

Network topology dependent

- If hosts corrected directly to Internet packet switches (no translation)
- Some LANs allow network addresses to be assigned dynamically to hosts chosen to match the id portion of the IP address
- For Ethernets:
  - Ethernet: cache; not in cache, broadcast IP addr, receive Ethernet addr

Address Resolution Protocol (ARP)

- Used to translate IP addresses to Ethernet addresses.
  - Translation done *only for outgoing IP packets*: this is when the IP header and the Ethernet header are created.
  - Translation is performed with a table look-up in an ARP Table; this is stored in memory and contains a row for each computer:

IP address	Ethernet address
223.1.2.1	08-00-39-00-2F-C3
223.1.2.3	08-00-5A-21-A7-22
223.1.2.4	08-00-10-99-AC-54

- The ARP table is necessary because the IP address and Ethernet address are selected independently:
  - the IP address is selected by the network manager based on the location of the computer on the internet.
  - the Ethernet address is selected by the manufacturer based on the Ethernet address space licensed by the manufacturer.
- Each host has separate ARP tables for each of its Ethernet interfaces.

Internet Protocols: The IP Protocol

Address Resolution Protocol (APR)

IP spoofing: address can be stolen (not authenticated)

Denial of service: many ping requests to a large number of computers at several sites with the IP address of a target computer as source

Internet Protocols: The IP Protocol (NAT)

NAT-enabled routers maintain an address translation table (ATT)

- Router has a "global" IP address from ISP
- Each machine has a "local" IP address via DHCP

UDP or TCP packet from the internal network to a computer outside

Router

1. Receives the packet and saves the source IP address and port number to an available slot in the ATT
2. Replaces the source address in the packet with its own IP address and the source port with a virtual port number that indexes the table slot containing the sending computer's address information
3. Forward the modified packet

UDP or TCP packet from an external computer

Router

1. Uses the destination port number to access a slot in the ATT
2. Replaces the destination address and port with those stores in the slot and forwards the packet

Internet Protocols: The IP Protocol (NAT)

NAT-enabled routers maintain an address translation table (ATT)

Maintains (cache) the mapping – a timer (if not accessed expires)

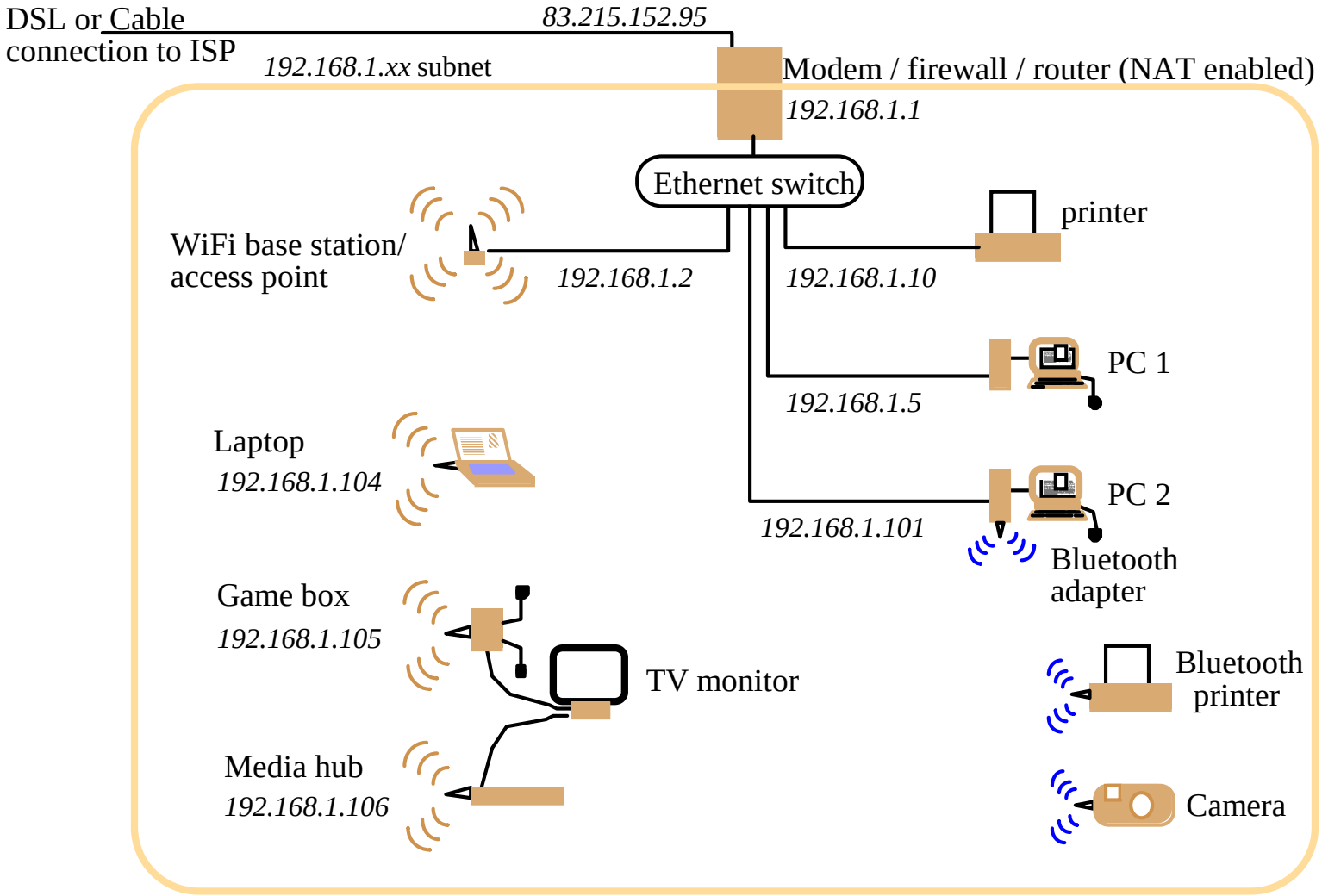
Ok, if as clients to external services, such as web services

As servers:
Manually configured to forward all of the incoming requests on a given port to one particular internal computer

Fixed internal addr and port #
Fixed entry in the table
All packets to the port on the router are forwarded to the internal addr and port # in the entry

Ok, if only one computer on any given port

Internet Protocols: IP Routing



Internet Protocols: The IP Protocol (NAT)

- Machine -> router
  - Router stores the local IP addr and source port #
  - Table entry indexed by a virtual port #
- Router -> outside
  - put the router IP addr and virtual port # in the packet
- Outside -> router
  - Reply to the router IP addr and virtual port #
- Router -> machine
  - Use the virtual port # to find table entry
  - Forward to the local IP address and port #

Internet Protocols: IP Routing

- RIP-1: discussed previously
- RIP-2: CIDR, better multicast routing, authentication of RIP packets
- link-state algorithms: e.g., open shortest path first (OSPF)
- Observed: average latency of IP packets peaks at 30-seconds intervals [RIP updates are processed before IP]
  - because 30-second RIP update intervals, locked steps
  - random interval between 15-45 seconds for RIP update

Internet Protocols: IP Routing

- large routing table size
  - all destinations!!

Solution 1:

Topological grouping of IP addresses

map ip to geographical location

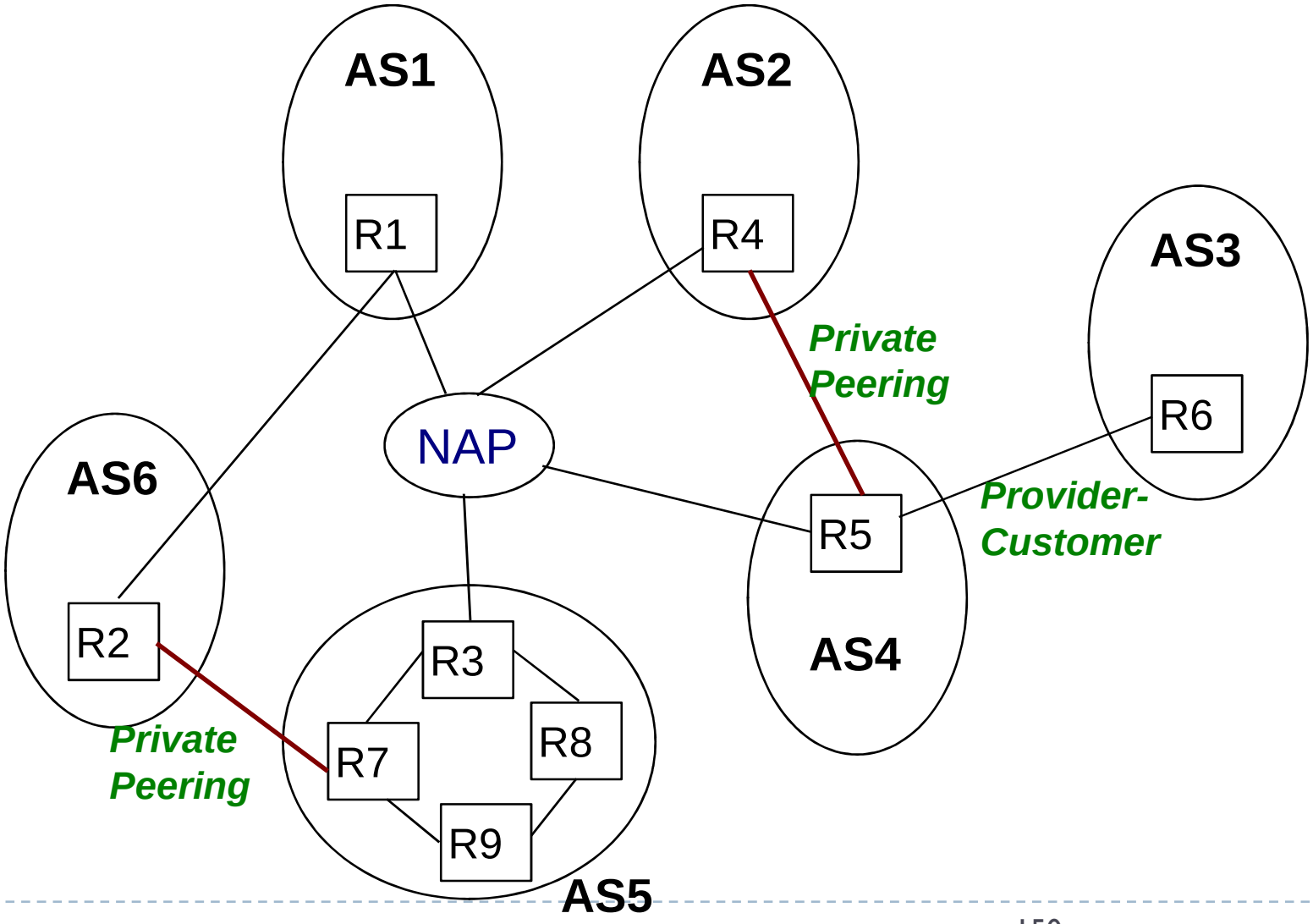
Solution 2:

- default route: store a subset, default to a single link for unlisted destinations
- (as long as key routers (those closest to the backbone links) have relatively complete table)

Internet Protocols: IP Routing

- The Internet is divided into regions under a single administrative control, each of which is called an Autonomous System and contains a group of network IDs.
- Routing inside an AS is completely hidden from the rest of the Internet.
- Routes between ASs are computed in terms of AS hops: lists of intermediary ASs from the source AS to the destination AS.
  - All outside networks that belong to the same AS share the same route, expressed as the list of intermediary ASs.
  - To route to arbitrary ASs on the Internet, a router need only know the next-hop router to every AS, rather than to individual destinations.

Autonomous Systems



Routing within ASs

- Under complete control of AS owner
- Small ASs have only one router, which does the internal and external routing (e.g., AS1, AS2, AS3, AS4, AS6)
- Larger ASs may have more than one routers:
  - some of them are *border routers* and deal with outgoing or incoming traffic to the AS (e.g., AS5)
  - for internal routing the Open Shortest Path First (OSPF) protocol is used

AS categories

- Transit AS:
  - an AS with connections to more than one AS that is also willing to carry datagrams that neither originate nor terminate on its own hosts
- Multihomed AS:
  - an AS that is connected to more than one AS and that does not accept datagrams not destined to itself
- Stud AS:
  - an AS that is connected to only one other AS

152

AS Connections

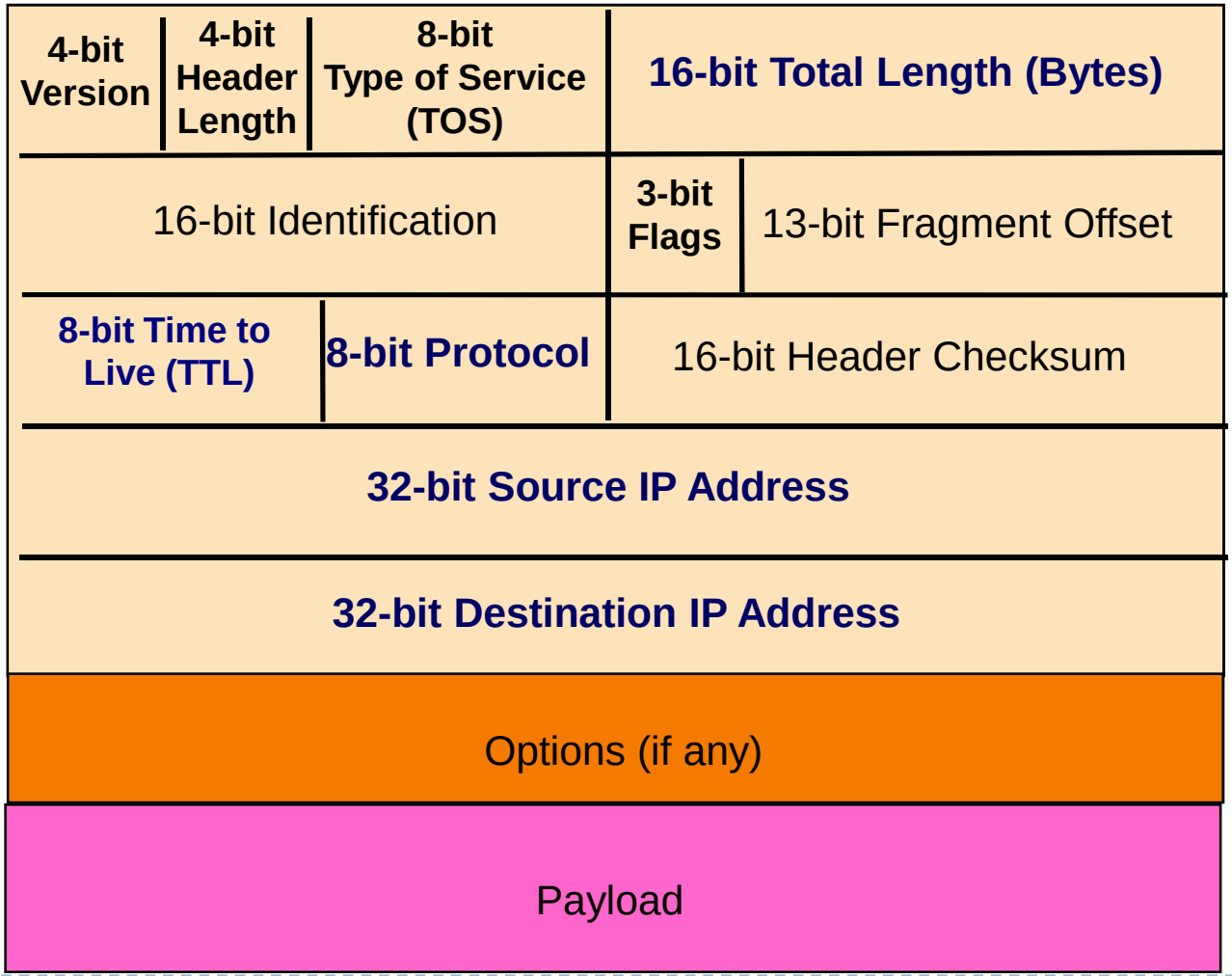
- Network Access Point (NAP): a physical network that connects routers from different ASs
- Private Peering Links: private physical networks connecting only the routers from two ASs
  - Peering agreements usually limit traffic over peering link to non-transit traffic
- Provider-customer relationship: similar to peering linkage but usually complies to different business model:
  - the provider delivers transit traffic to the customer

Routing between ASs

- Done according to the Border Gateway Protocol (BGP):
  - Each router advertises reachability information to neighbor routers for:
    - all networks within its AS and
    - for outside networks reachable via its AS (for transit ASs)
  - Reachability information includes a list of reachable networks and performance cost expressed as the number of hops to the destination
  - An AS can set routing policies that determines which reachability information is advertised to which routers

IP packet

IP Packet Structure



IP Packet Header Fields

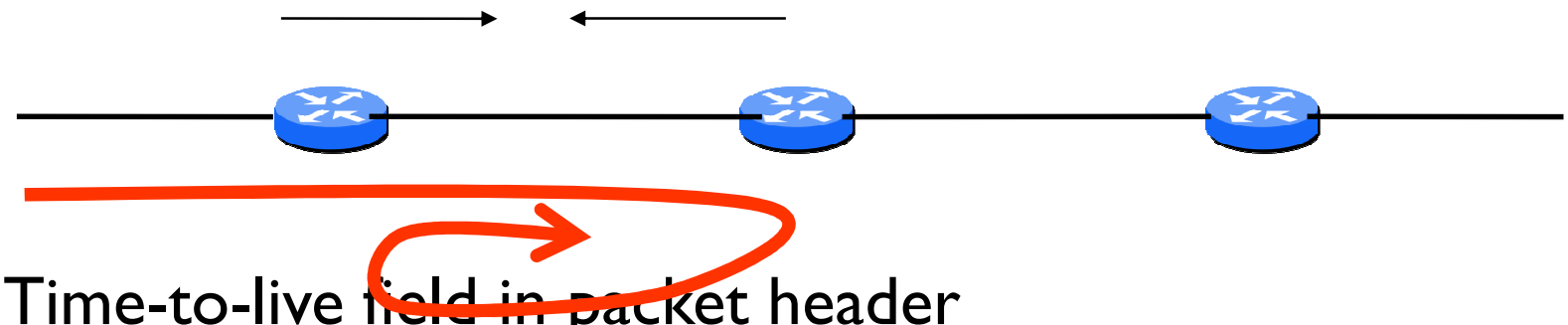
- Version number (4 bits)
  - Indicates the version of the IP protocol
  - Necessary to know what other fields to expect
  - Typically "4" (for IPv4), and sometimes "6" (for IPv6)
- Header length (4 bits)
  - Number of 32-bit words in the header
  - Typically "5" (for a 20-byte IPv4 header)
  - Can be more when "IP options" are used
- Type-of-Service (8 bits)
  - Allow packets to be treated differently based on needs
  - E.g., low delay for audio, high bandwidth for bulk transfer

IP Packet Header Fields (Continued)

- Total length (16 bits)
  - Number of bytes in the packet
  - Maximum size is 63,535 bytes (2^{16} -1)
  - … though underlying links may impose harder limits
- Fragmentation information (32 bits)
  - Packet identifier, flags, and fragment offset
  - Supports dividing a large IP packet into fragments
  - … in case a link cannot handle a large IP packet
- Time-To-Live (8 bits)
  - Used to identify packets stuck in forwarding loops
  - … and eventually discard them from the network

Time-to-Live (TTL) Field

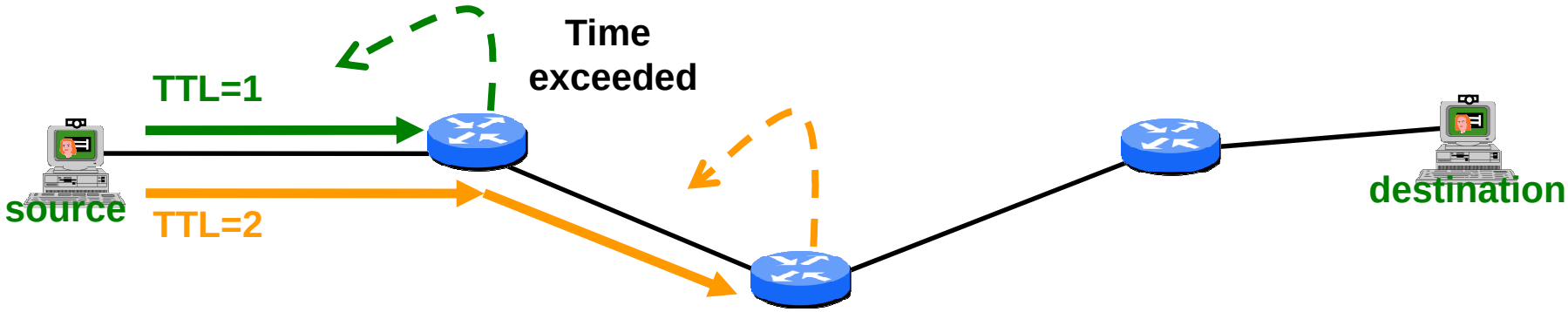
- Potential robustness problem
  - Forwarding loops can cause packets to cycle forever
  - Confusing if the packet arrives much later



- Time-to-live field in packet header
  - TTL field decremented by each router on the path
  - Packet is discarded when TTL field reaches 0…
  - …and "time exceeded" message is sent to the source

Application of TTL in Traceroute

- Time-To-Live field in IP packet header
  - Source sends a packet with a TTL of *n*
  - Each router along the path decrements the TTL
  - "TTL exceeded" sent when TTL reaches *0*
- Traceroute tool exploits this TTL behavior



Send packets with TTL=1, 2, … and record source of "time exceeded" message

Example Traceroute: Berkeley to CNN

Hop number, IP address, DNS name

1	169.229.62.1	inr-daedalus-0.CS.Berkeley.EDU
2	169.229.59.225	soda-cr-1-1-soda-br-6-2
3	128.32.255.169	vlan242.inr-202-doecev.Berkeley.EDU
4	128.32.0.249	gigE6-0-0.inr-666-doecev.Berkeley.EDU
5	128.32.0.66	qsv-juniper--ucb-gw.calren2.net
6	209.247.159.109	POS1-0.hsipaccess1.SanJose1.Level3.ne
7	*	?
8	64.159.1.46	?
9	209.247.9.170	pos8-0.hsa2.Atlanta2.Level3.net
10	66.185.138.33	pop2-atm-P0-2.atdn.net
11	*	?
12	66.185.136.17	pop1-atl-P4-0.atdn.net
13	64.236.16.52	www4.cnn.com

No response from router

No name resolution

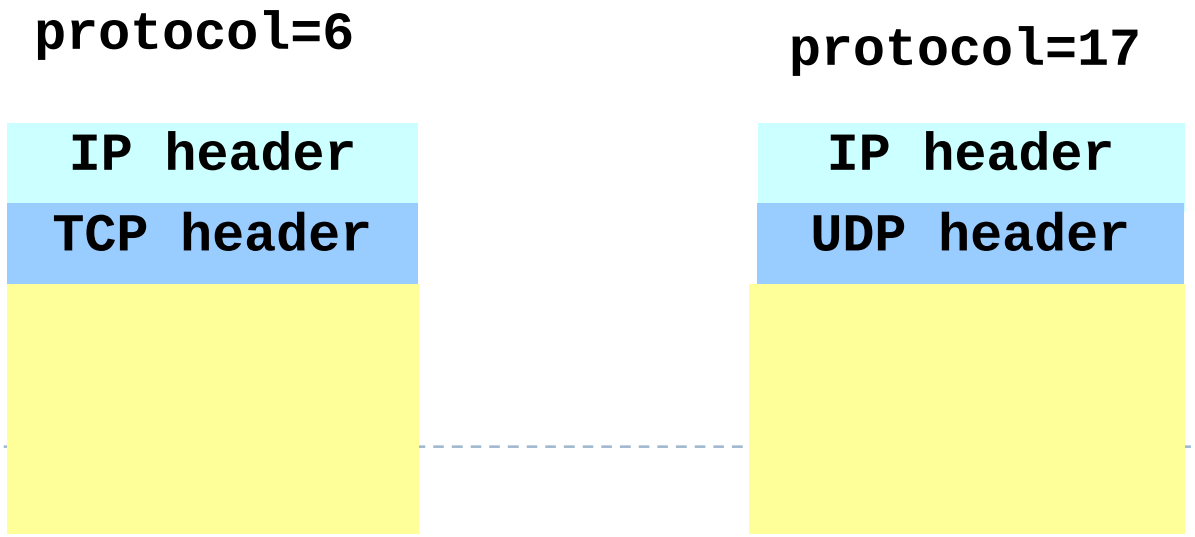
Try Running Traceroute Yourself

- On UNIX machine
  - Traceroute
  - E.g., "traceroute www.cnn.com" or "traceroute 12.1.1.1"
- On Windows machine
  - Tracert
  - E.g., "tracert www.cnn.com" or "tracert 12.1.1.1"

- Common uses of traceroute
  - Discover the topology of the Internet
  - Debug performance and reachability problems

IP Packet Header Fields (Continued)

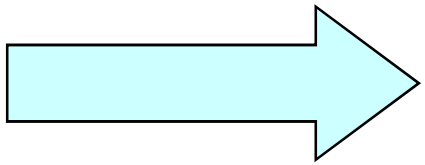
- Protocol (8 bits)
  - Identifies the higher-level protocol
    - E.g., "6" for the Transmission Control Protocol (TCP)
    - E.g., "17" for the User Datagram Protocol (UDP)
  - Important for demultiplexing at receiving host
    - Indicates what kind of header to expect next



IP Packet Header Fields (Continued)

- Checksum (16 bits)
  - Sum of all 16-bit words in the IP packet header
  - If any bits of the header are corrupted in transit
  - … the checksum won't match at receiving host
  - Receiving host discards corrupted packets
    - Sending host will retransmit the packet, if needed

```
  134          134
+ 212        + 216
-----        -----
= 346        = 350
```



Mismatch!

IP Packet Header (Continued)

- Two IP addresses
  - Source IP address (32 bits)
  - Destination IP address (32 bits)
- Destination address
  - Unique identifier for the receiving host
  - Allows each node to make forwarding decisions
- Source address
  - Unique identifier for the sending host
  - Recipient can decide whether to accept packet
  - Enables recipient to send a reply back to source

Questions?