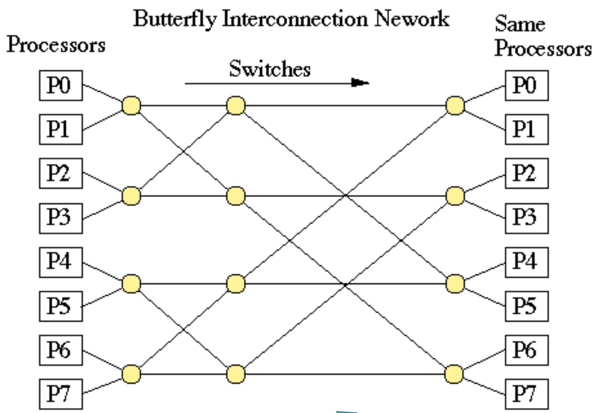
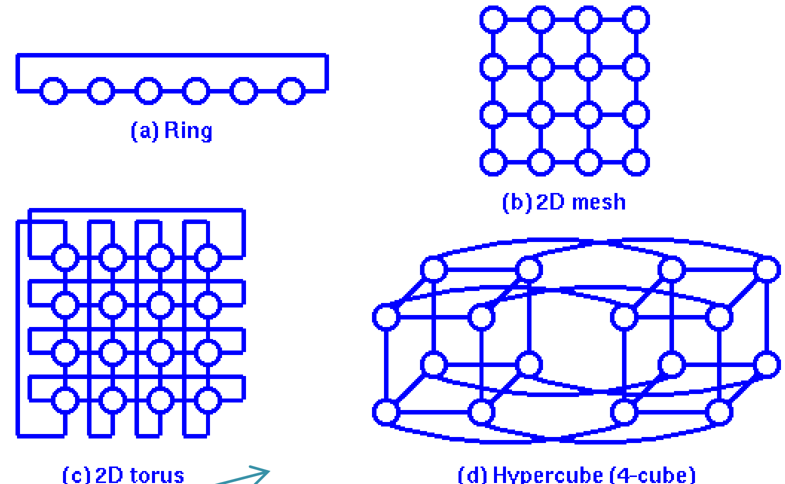
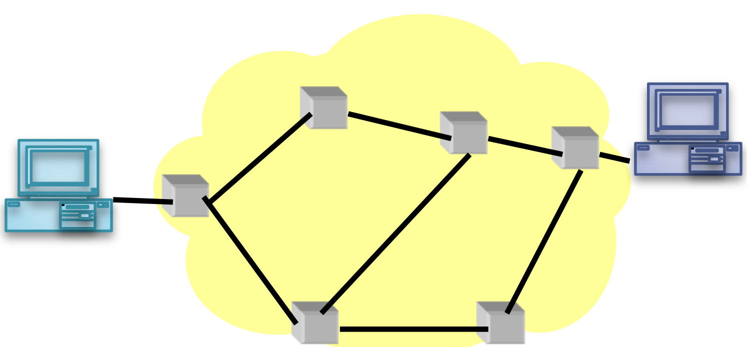
Simple Internetworking

Networks Everywhere!

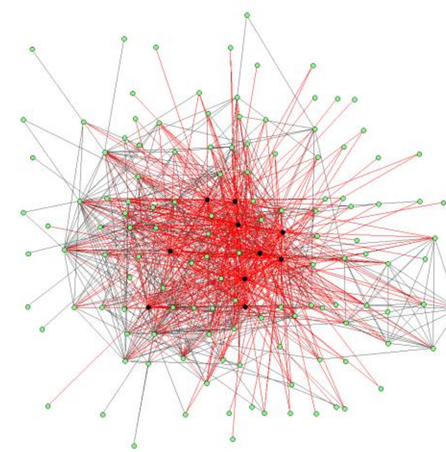




Processor networks – high performance computing



Computer networks



Biological networks

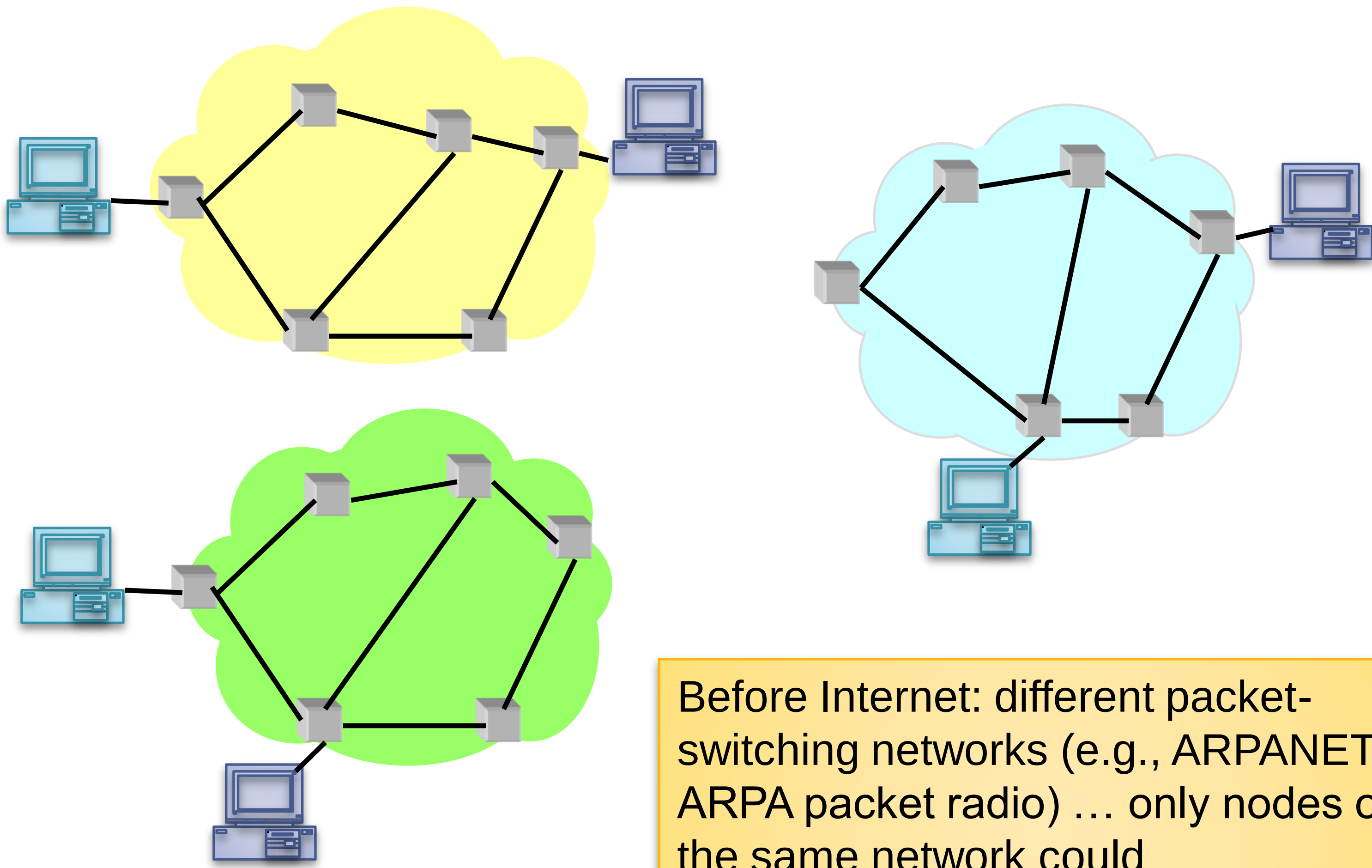
Network Fundamentals

- ***Addresses*** – most human-made networks explicitly assign addresses to nodes
- ***Forwarding*** – routing messages to a given target
- ***Routing*** – collecting state information for forwarding

Examples

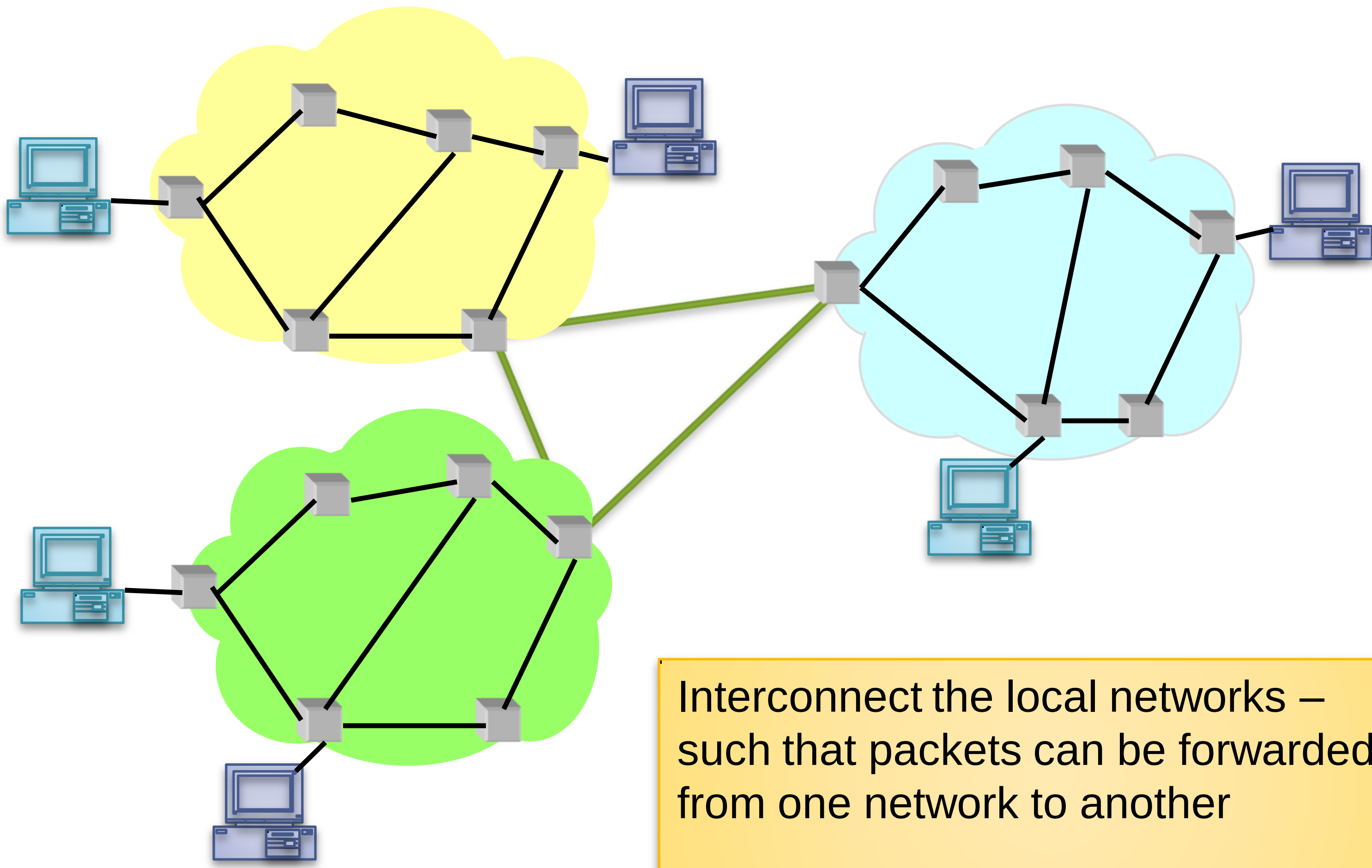
- Traffic/city networks – city names (addresses), road signs (forwarding), topological mapping (routing)
- Processor networks (in supercomputers) – processor numbers (addresses), switches (forwarding), topological mapping (done once assuming invariant topology)

The Internetworking Problem



Before Internet: different packet-switching networks (e.g., ARPANET, ARPA packet radio) … only nodes on the same network could communicate

The Internetworking Problem



Interconnect the local networks – such that packets can be forwarded from one network to another

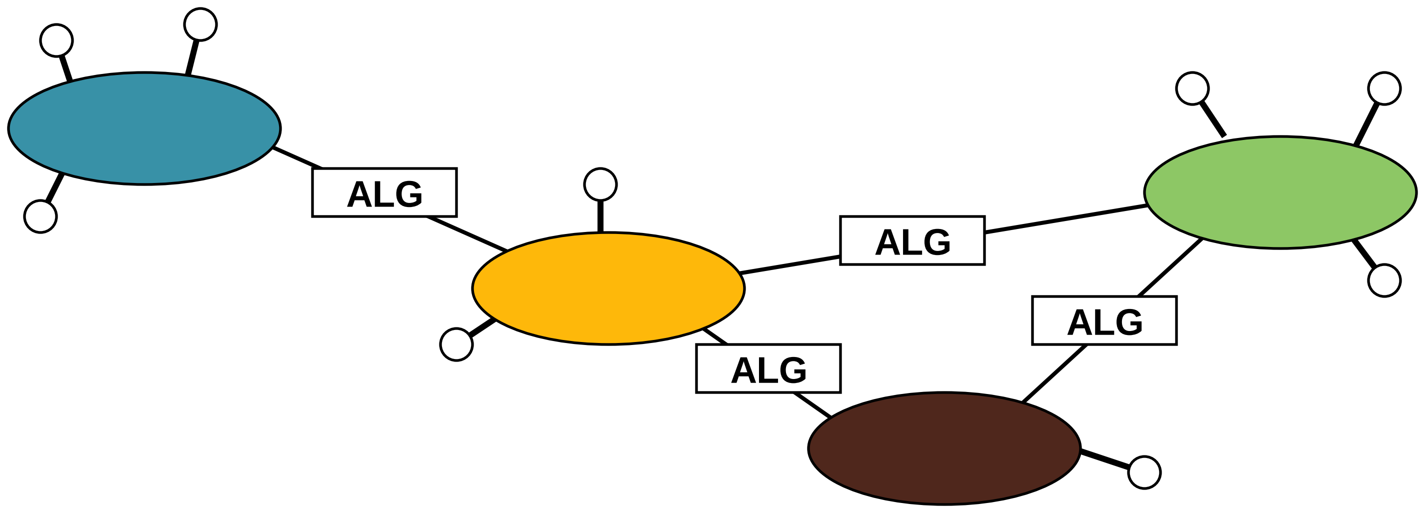
Internetworking Requirements

- ***Connectivity***
  - Need *access* to resources, data in different networks
  - Local networks cannot be changed dramatically – preserve *local autonomy*
- ***Cost***
  - Reuse existing functionality as much as possible
  - What can we *reuse?*

Reuse Local Network Functions

- What can we reuse?
- How do we reuse?
- Remember Internet is a network – it needs
  - Addresses
  - Forwarding functions
  - Routing functions

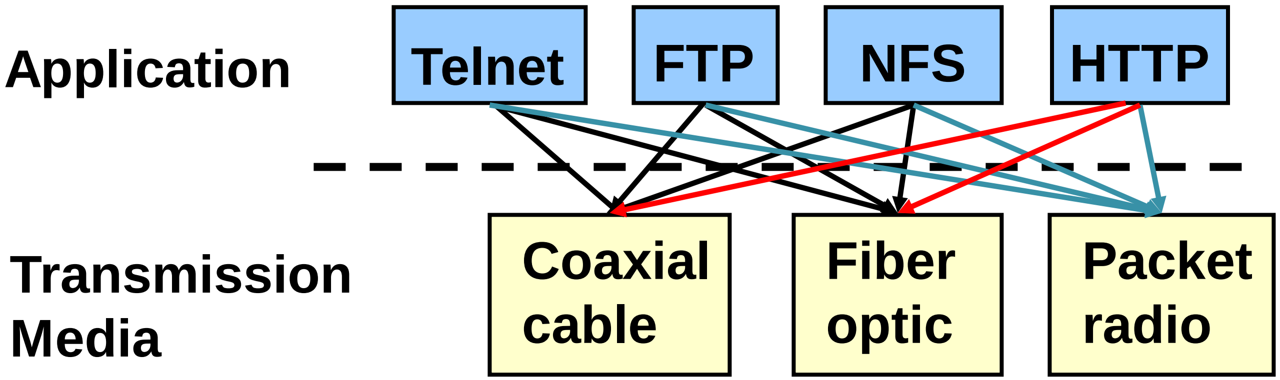
A Translation-based Solution



- application-layer gateways
  - difficult to deploy new internet-wide applications
  - hard to diagnose and remedy end-to-end problems
  - stateful gateways inhibited dynamic routing around failures
- no global addressability
  - ad-hoc, application-specific solutions

Native Solution: Directly on Network

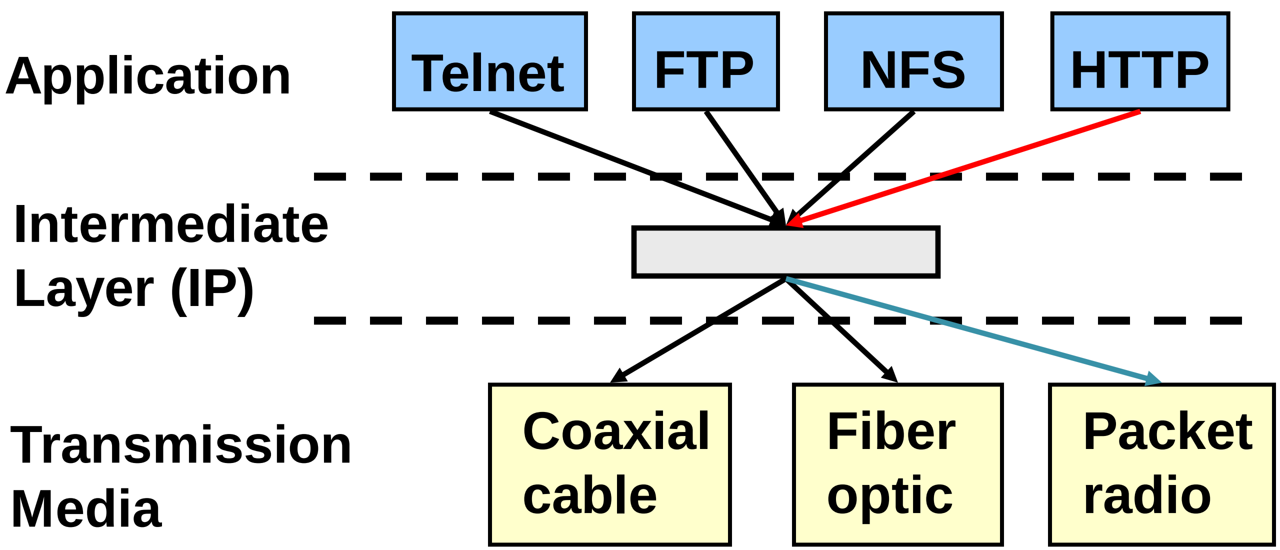
(FTP – File Transfer Protocol, NFS – Network File Transfer, HTTP – World Wide Web protoc



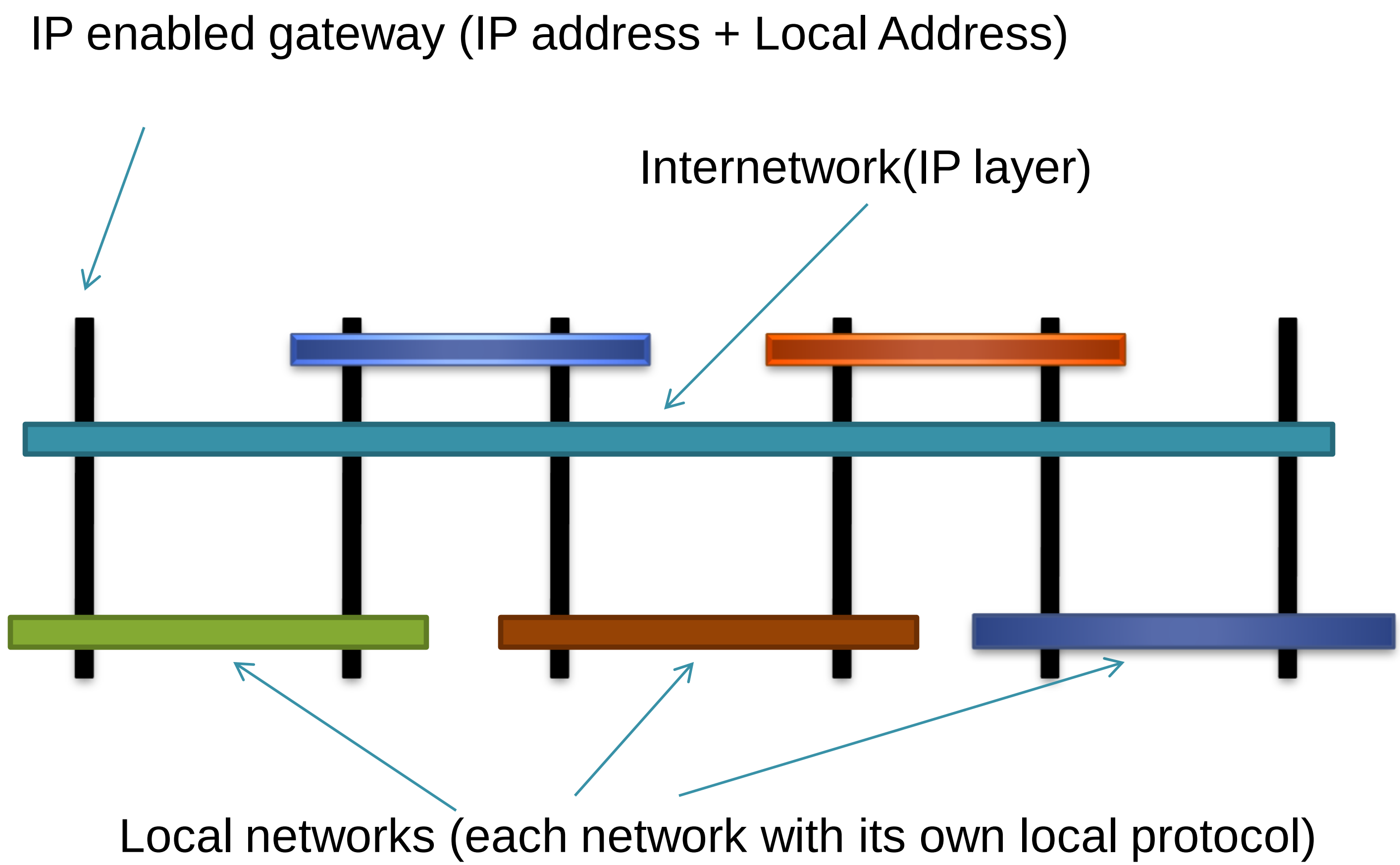
- No network level overlay: each new application has to be *re*-implemented for every network technology!

IP as an Intermediary

- Key ideas:
  - Overlay: better than any↔any translation. Fewer, simpler mappings.
  - Network-layer: efficient implementation, global addressing



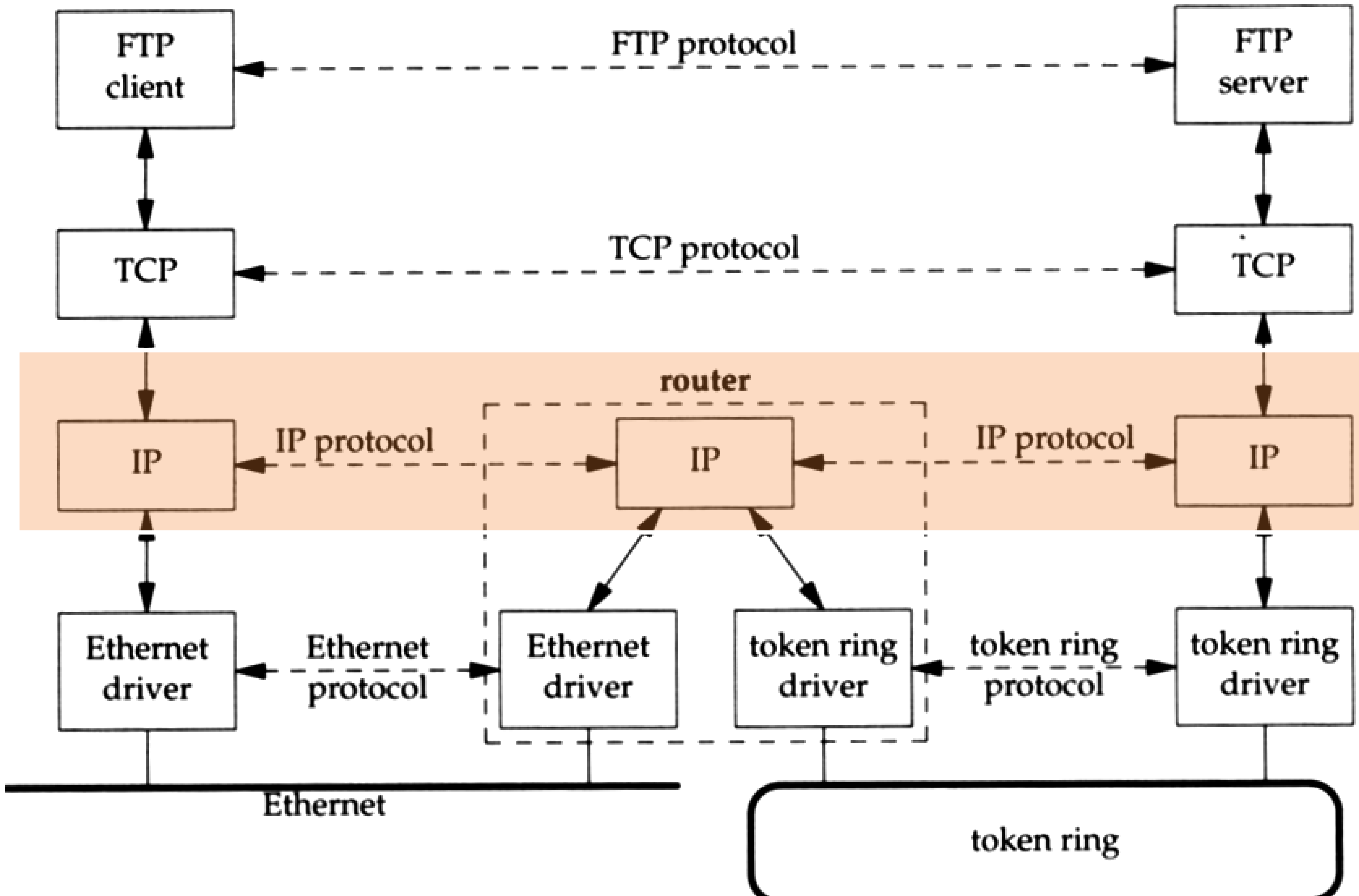
IP as an Intermediary



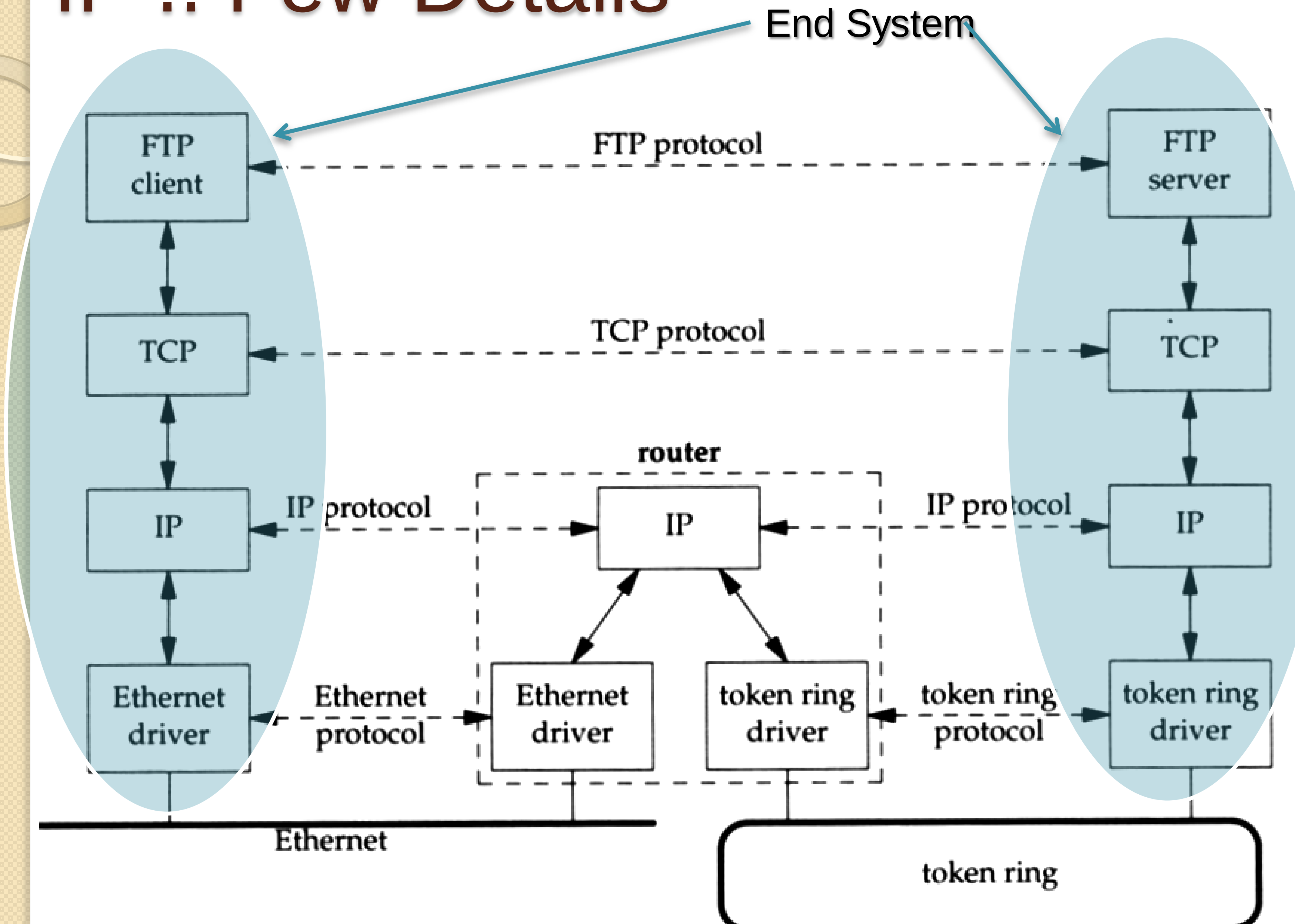
Internet Protocol: Few Details

- IP runs over everything
- Networks linked together by Internetwork programs -- each host supports them
- Example Internetwork shown next:
  - Consists of two networks – Ethernet and Token ring
  - Host connected to Ethernet can talk to one connected to the Token

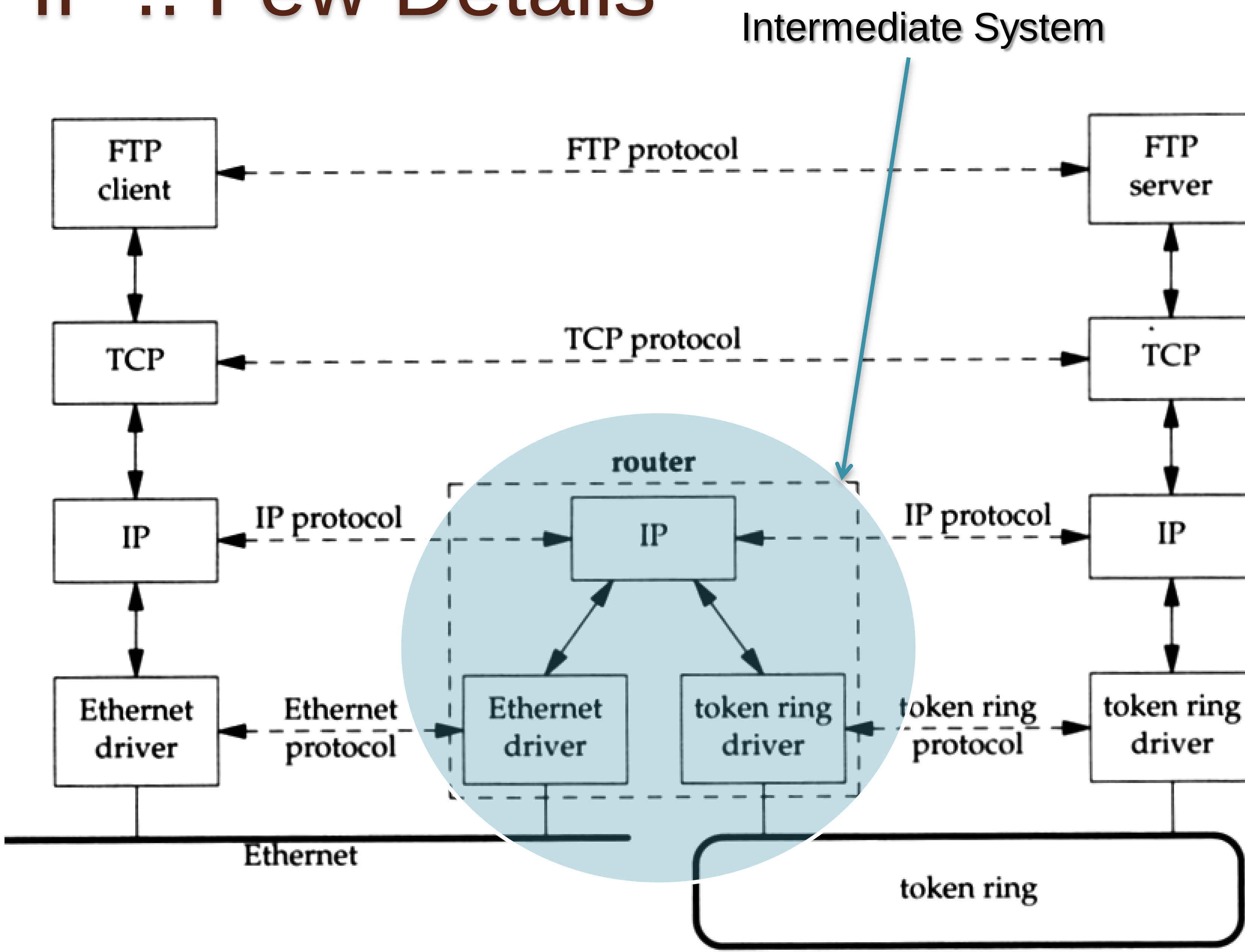
IP .. Few Details



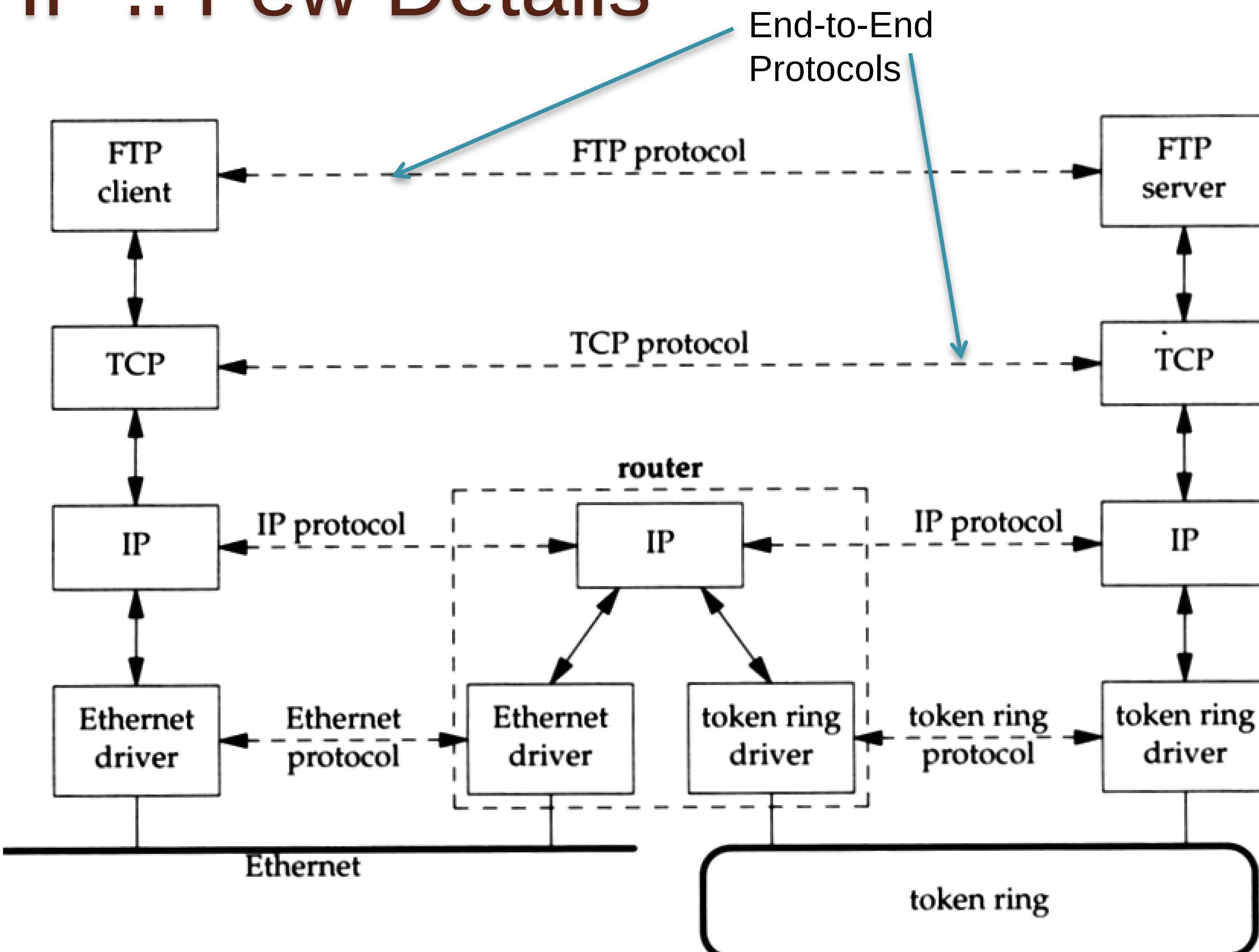
IP .. Few Details



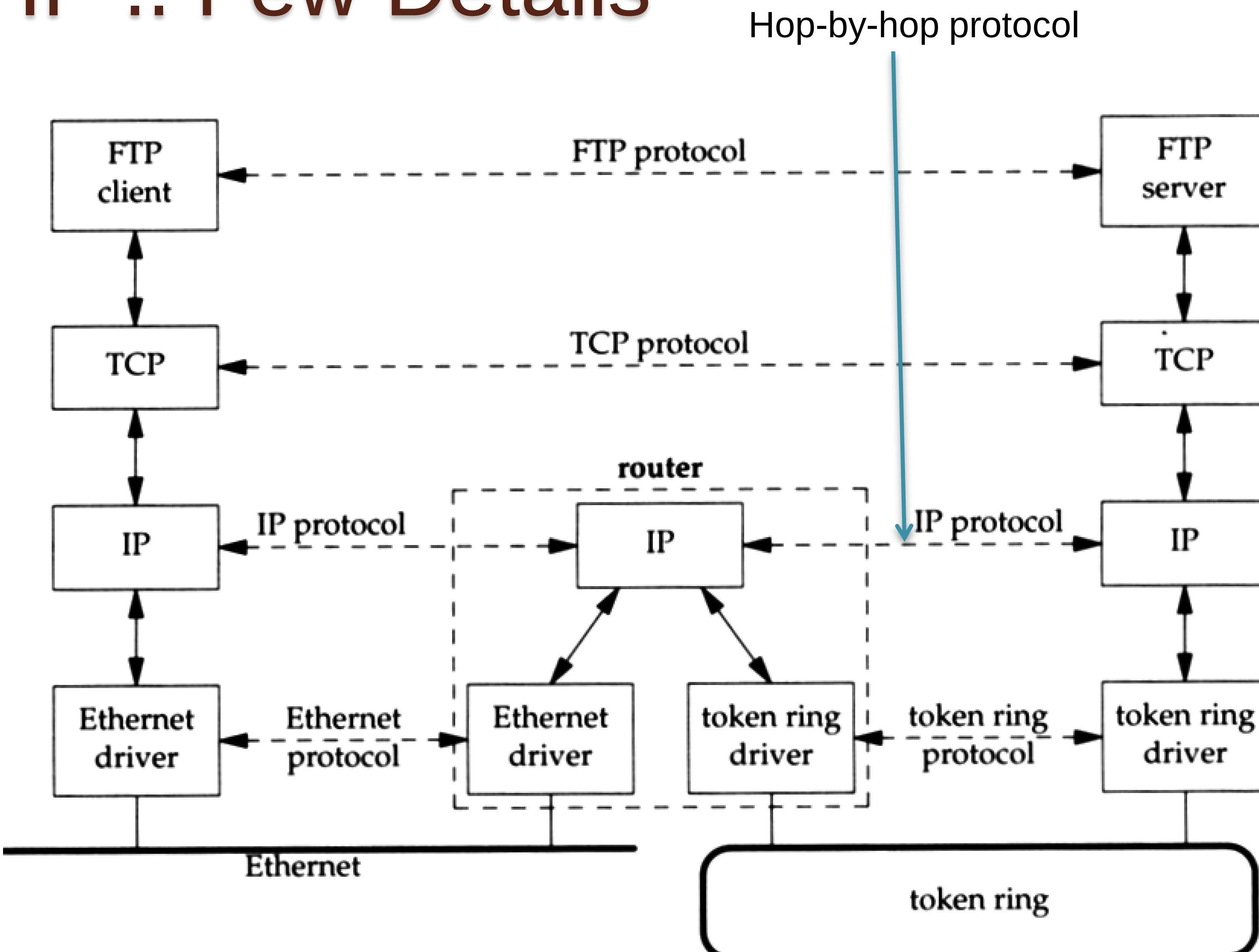
IP .. Few Details



IP .. Few Details

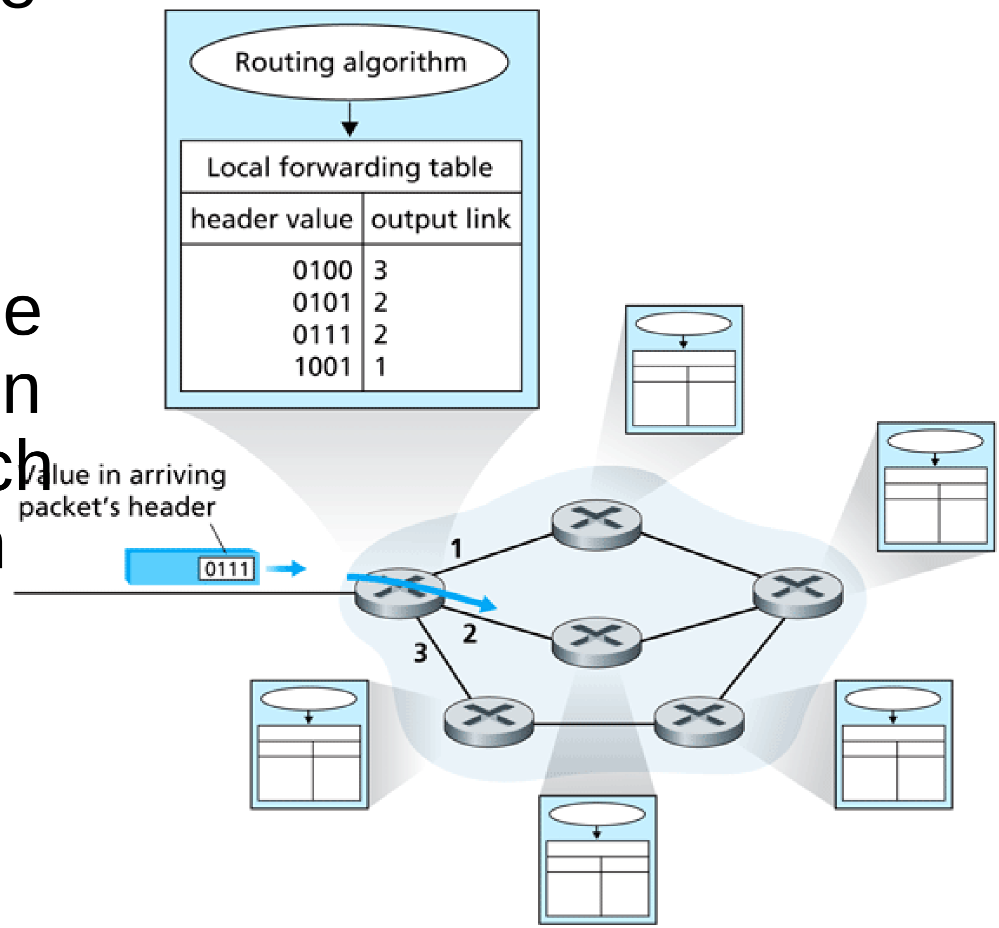


IP .. Few Details



Forwarding Vs Routing

- ***Forwarding:*** consult a local table to decide the best way of moving an incoming packet
- ***Routing:*** determine the path to be taken by a packet to reach a given destination



IP Forwarding (Case I)

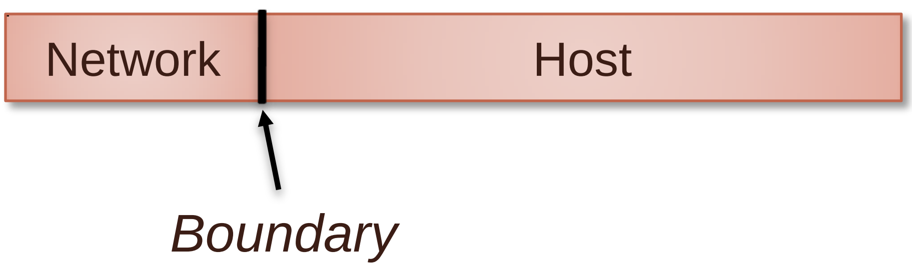
- *Source & Destination in same network (direct connectivity)*
  - Recognize that destination IP address is on same network.
  - Find the destination LAN address.
  - Send IP packet encapsulated in LAN frame directly to the destination LAN address.
    - Encapsulation => source/destination IP addresses don't change

IP Forwarding (Case II)

- B) *Source & Destination in different networks (indirect connectivity)*
  - Recognize that destination IP address is ***not*** on same network.
  - ***Look up*** destination IP address in a (L3 forwarding) table to find a match, called the next hop router IP address.
  - Send packet encapsulated in a LAN frame to the LAN address corresponding to the IP address of the next-hop router.

IP Addressing

- *How to find if destination is in the same network ?*
  - IP address = network ID + host ID.
    - *Source and destination network IDs match => same network (I.e. direct connectivity)*
  - Splitting address into multiple parts is called *hierarchical addressing*



Address Resolution

- *How to find the LAN address corresponding to an IP address ?*
  - *Address Resolution Problem.*
  - Solution: ARP

IP Forwarding: Example Scenario

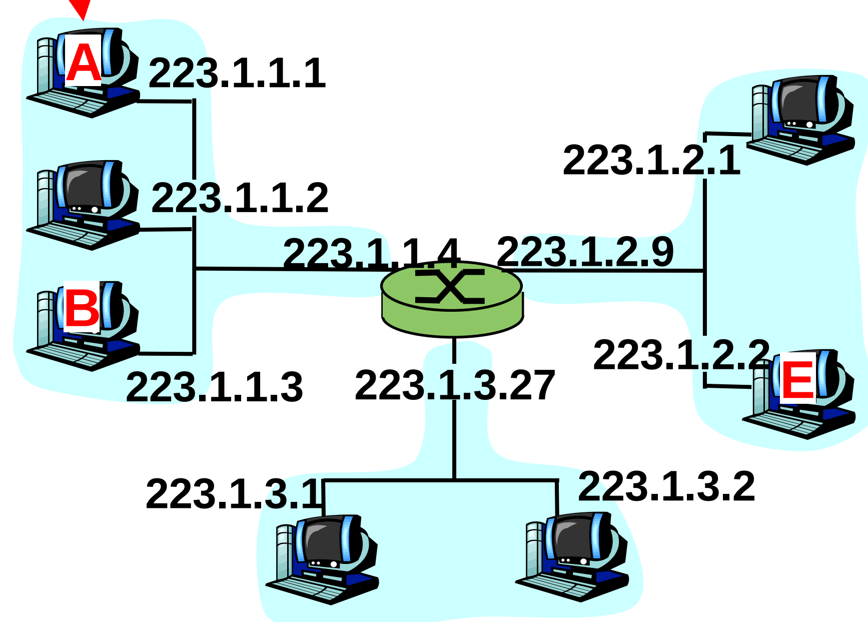
IP datagram:

misc fields	source IP addr	dest IP addr	data

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

routing table in A

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



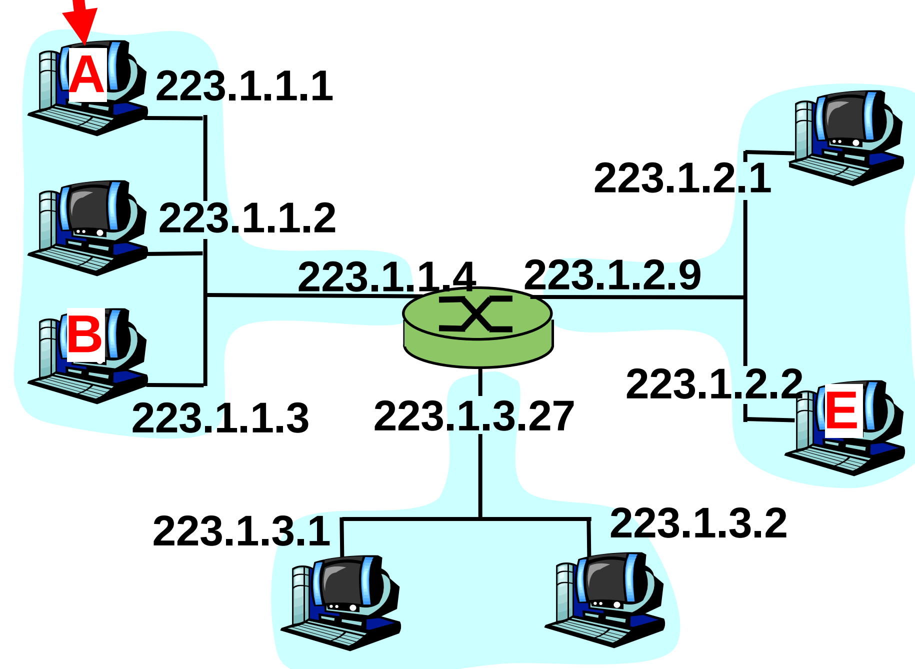
IP Forwarding (Direct)

misc fields	223.1.1.1	223.1.1.3	data

Dest. Net.	next router	Nhops
223.1.1		*1*
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

Starting at A, given IP datagram addressed to B:

- look up net. address of B
- find B is on same net. as A
- link layer will send datagram directly to B inside link-layer frame
  - B and A are directly connected



IP Forwarding (Indirect): Step 1

misc fields	223.1.1.1	223.1.2.2	data

Dest. Net.	next router	Nhops
223.1.1		1
223.1.2	*223.1.1.4*	*2*
223.1.3	223.1.1.4	2

Starting at A, dest. E:

look up network address of E

E on *different* network

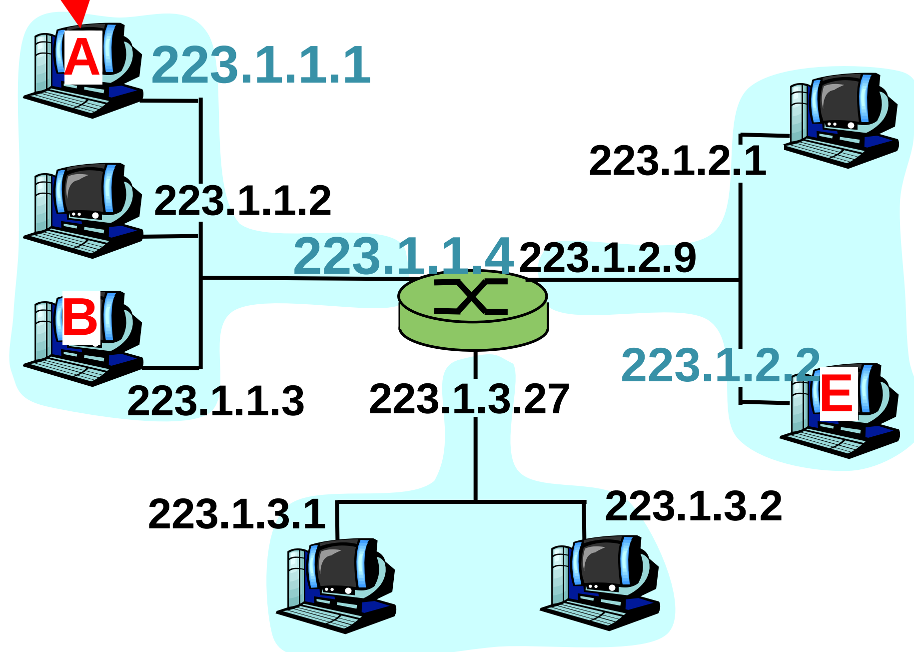
A, E not directly attached

routing table: next hop router to E is 223.1.1.4

link layer sends datagram to router 223.1.1.4 inside link-layer frame

datagram arrives at 223.1.1.4

continued…..



IP Forwarding (Indirect): Step 2

misc fields	223.1.1.1	223.1.2.2	data

Dest. network	next router	Nhops	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	*-*	*1*	*223.1.3.27*

Arriving at 223.1.1.4, destined for 223.1.2.2

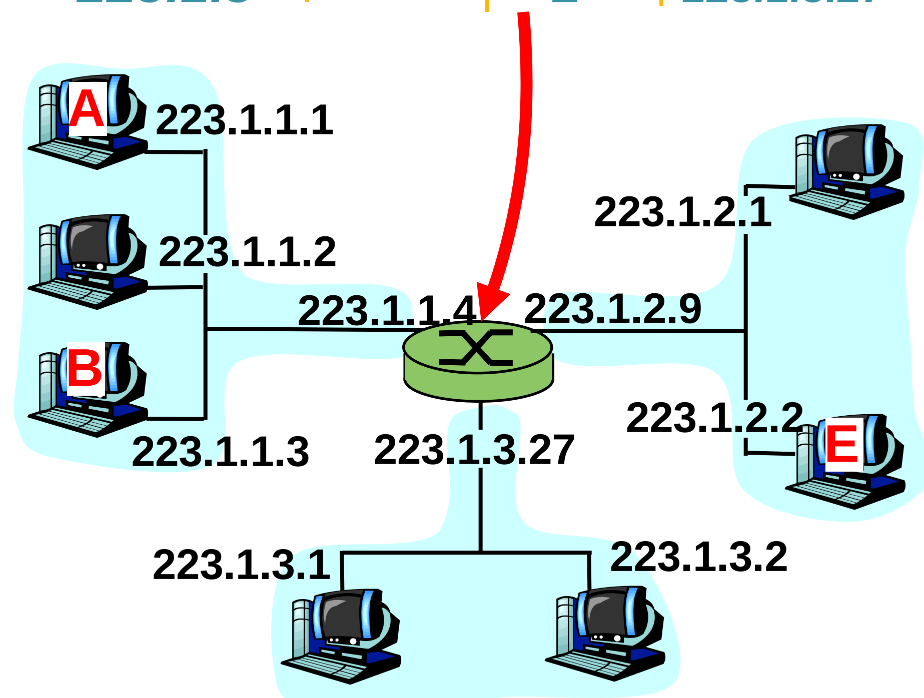
look up network address of E

E on *same* network as router's interface 223.1.2.9

router, E directly attached

link layer sends datagram to 223.1.2.2 inside link-layer frame via interface 223.1.2.9

datagram arrives at 223.1.2.2



IP Address Formats

Class A:	0	Network	Host	
	1	7	24	bits
Class B:	10	Network	Host	
	2	14	16	bits
Class C:	110	Network	Host	
	3	21	8	bits
Class D:	1110	Multicast Group addresses		
	4	28		bits

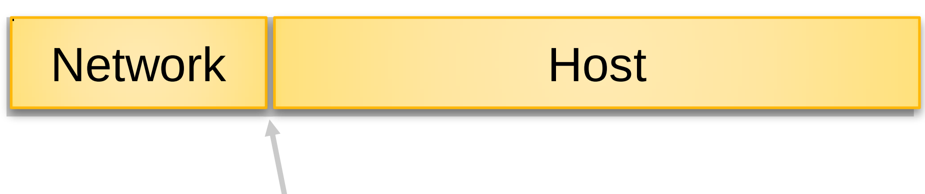
Dotted Decimal Notation

- Binary: 11000000 00000101 00110000 00000011
  Hex Colon: C0:05:30:03
  Dotted Decimal: 192.5.48.3

Class	Range
A	0 through 127
B	128 through 191
C	192 through 223
D	224 through 239
E	240 through 255

Subnet Addressing

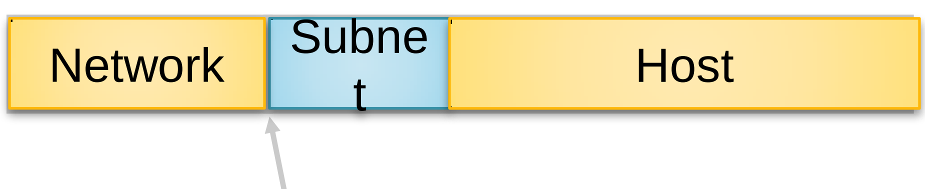
- *Classful* addressing inefficient: Everyone wants class B addresses
- Can we split class A, B addresses spaces and accommodate more networks ?
  - Need another level of hierarchy. Defined by "***subnet mask***", which in general specifies the sets of bits belonging to the network address and host address respectively



Boundary is flexible, and defined by subnet mask

Subnet Addressing

- *Classful* addressing inefficient: Everyone wants class B addresses
- Can we split class A, B addresses spaces and accommodate more networks ?
  - Need another level of hierarchy. Defined by "***subnet mask***", which in general specifies the sets of bits belonging to the network address and host address respectively



Boundary is flexible, and defined by subnet mask

Subnet Addressing...

- Example: Consider the address112.14.2.2 with a network mask 255.252.0.0.
- The address is structured as follows:
  - 0nnnnnnn.sssssshh.hhhhhhhh.hhhhhhhh
  - The given address is
  - 01110000.00001110.00000010. 00000010
  - in subnet with prefix 112.12.0.0.
  - The hosts in this subnet are numbered 112.12.0.1 to 112.15.255.254
  - 112.15.255.255 is the broadcast for the subnet
  - Each subnet has (2^{18} -2) hosts
  - CIDR prefix 112.12.0.0/14

Subnet Addressing...

- Subnet addressing happens when an ISP is issuing addresses to customers.
- We also have reverse problem – supernetting.
  - Aggregating networks together so that they can be specified by smaller number of prefixes
  - Subnetting creates more networks.
  - Therefore, supernetting is essential to reduce the workload of the core routers

More on IP prefixes

12.5.9.16 is covered by prefix 12.4.0.0/15

12.5.9.16	00001100	00000101	00001001	00010000
12.4.0.0/15	00001100	00000100	00000000	00000000
	11111111	11111110	00000000	00000000
12.7.9.16	00001100	00000111	00001001	00010000

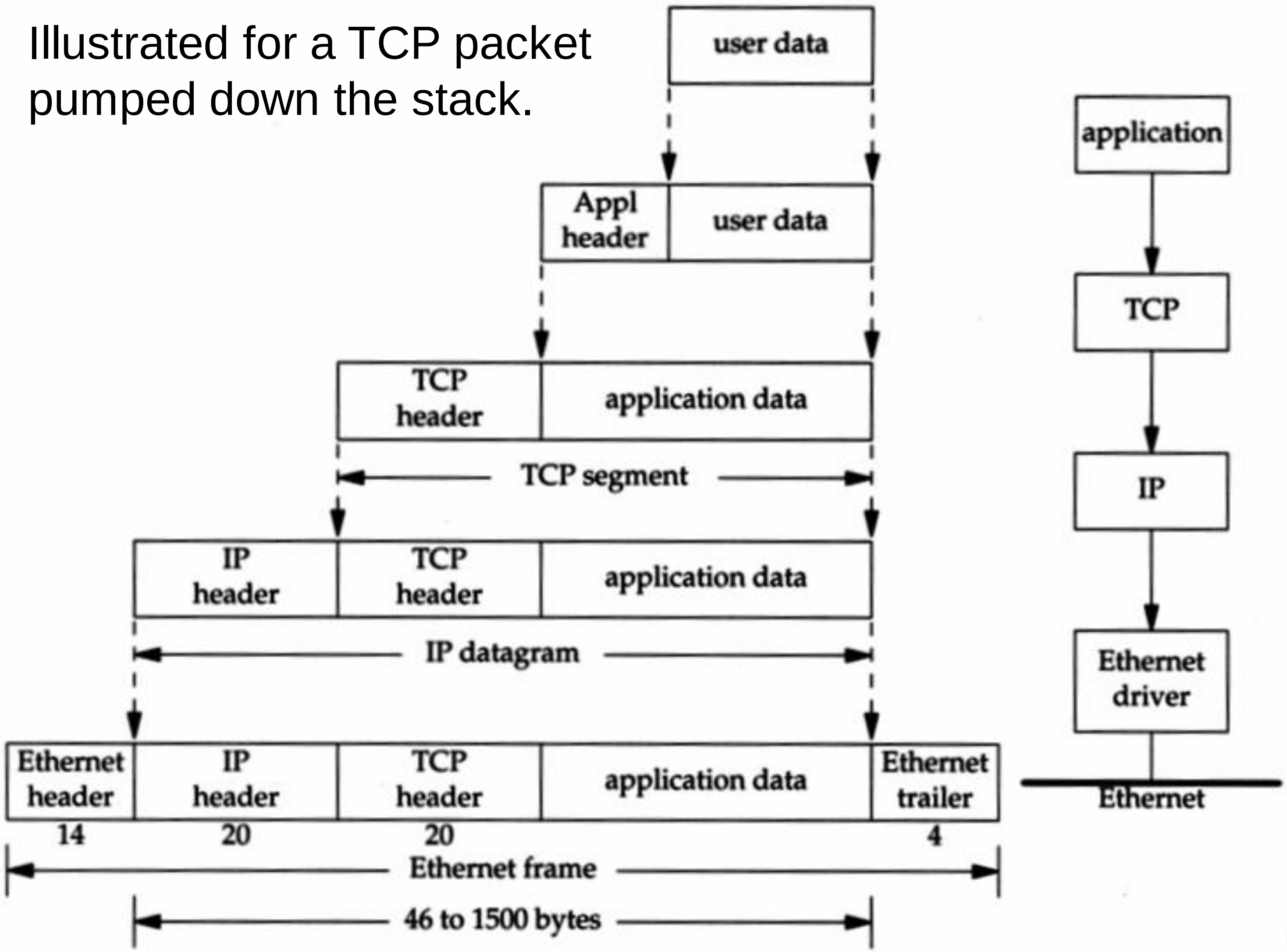
12.7.9.16 is not covered by prefix 12.4.0.0/15

IP Processing

- IP processing takes place at
  - Hosts – transmitting (initiating) and receiving IP packets
  - Routers – forwarding IP packets
- Host processing can involve
  - Reassembling packets, sending and receiving control messages (ICMP) related to IP packets
  - Deciding forwarding for its owns packets

IP Host Processing

Illustrated for a TCP packet pumped down the stack.



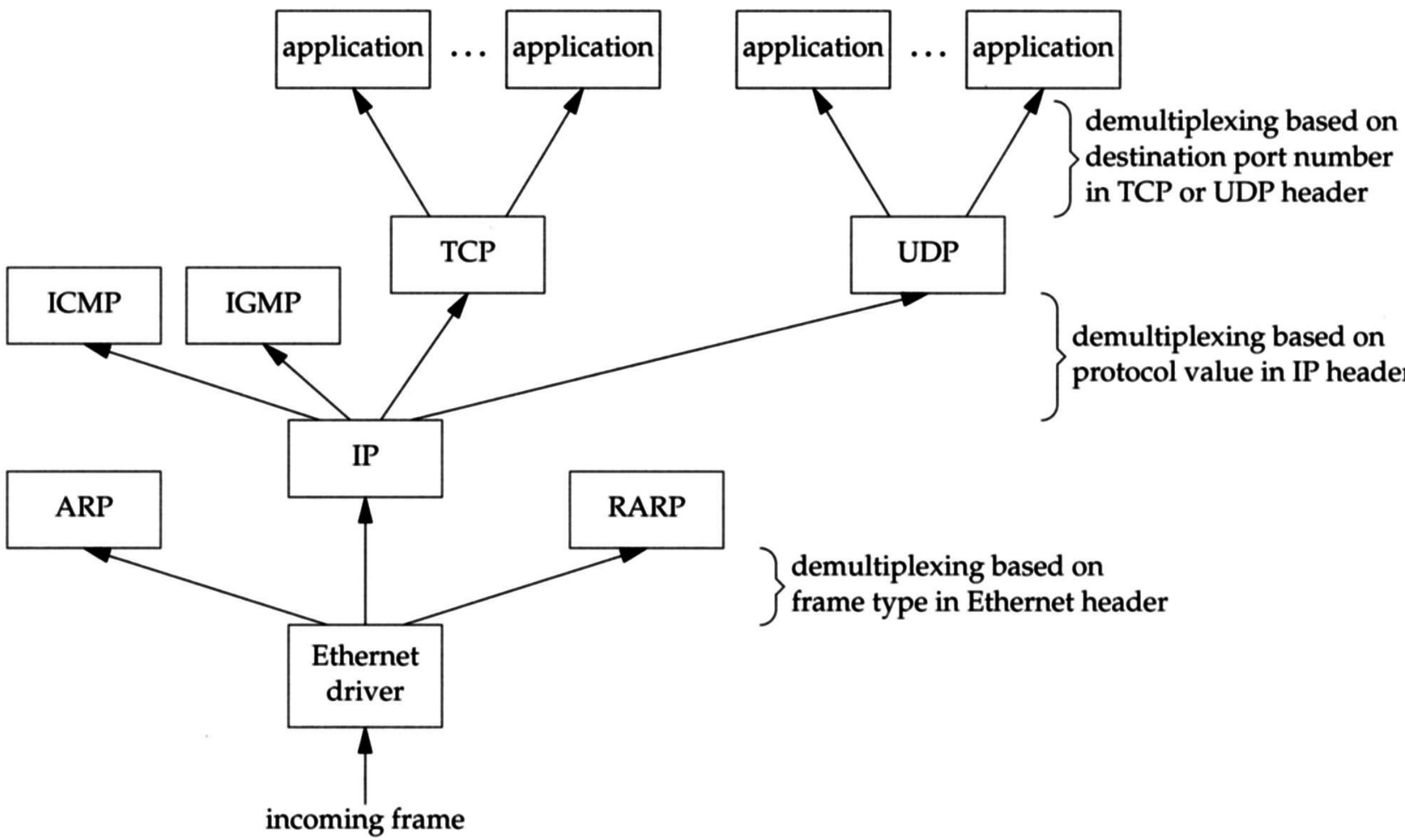
IP Host Processing …

- Nearly identical scheme true for other protocols
  - TCP, UDP, ICMP, and IGMP send data to IP
  - IP adds protocol field in the IP packet to indicate what type of data is in the data segment -- ICMP = 1, IGMP =2, TCP=6, UDP = 17
- Similarly, many different applications can be using TCP or UDP at any one time

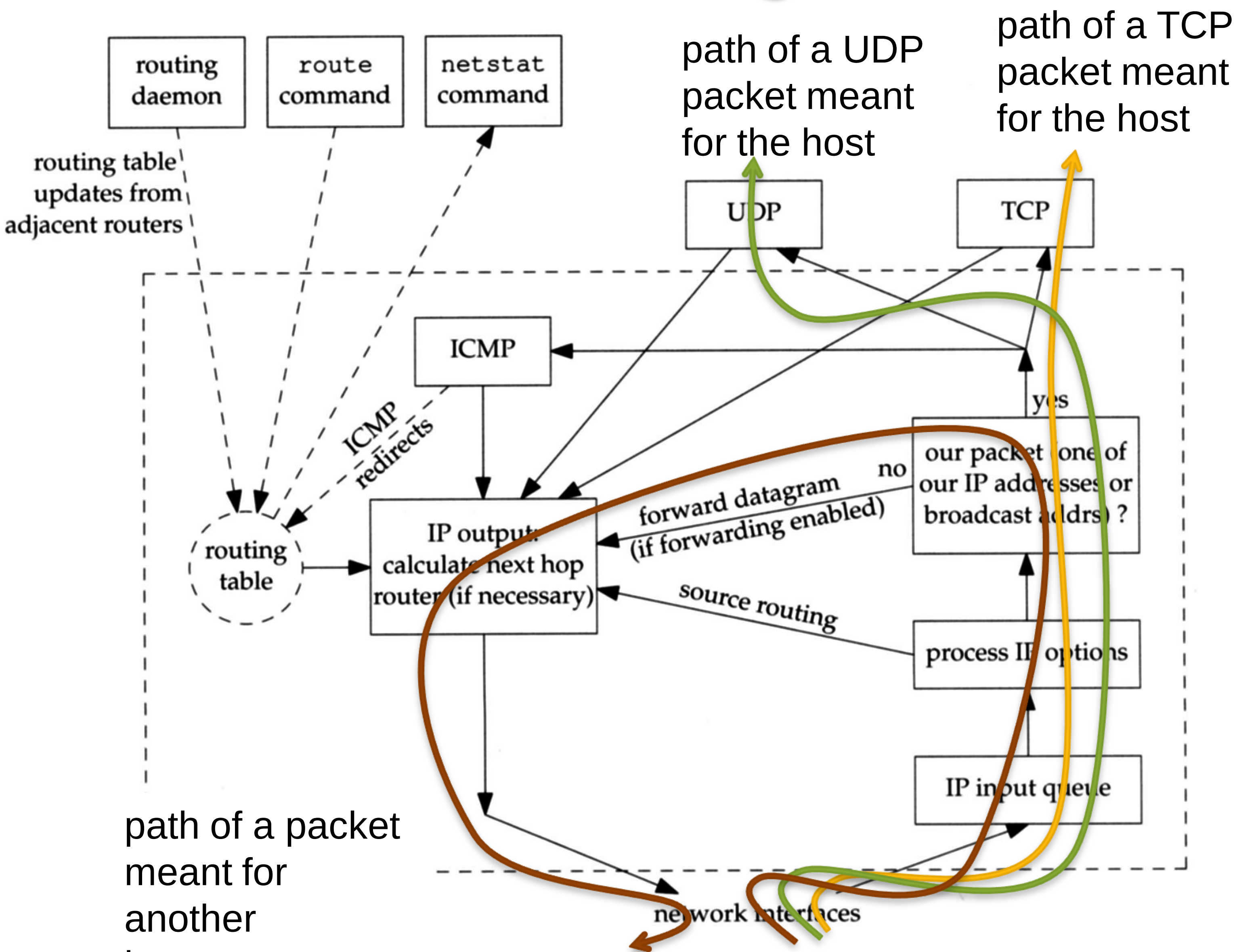
IP Host Processing…

- Network interface sends/receives frames
  - Frame type on Ethernet header is used to indicate the protocol
  - Protocol could be on behalf of IP, ARP, RARP -- a 16-bit frame type field in the Ethernet header is used to indicate the protocol
- Headers are removed as packet goes up
- Each protocol box uses the protocol field to demultiplex the packet among the upper layer receivers

IP Host Processing…

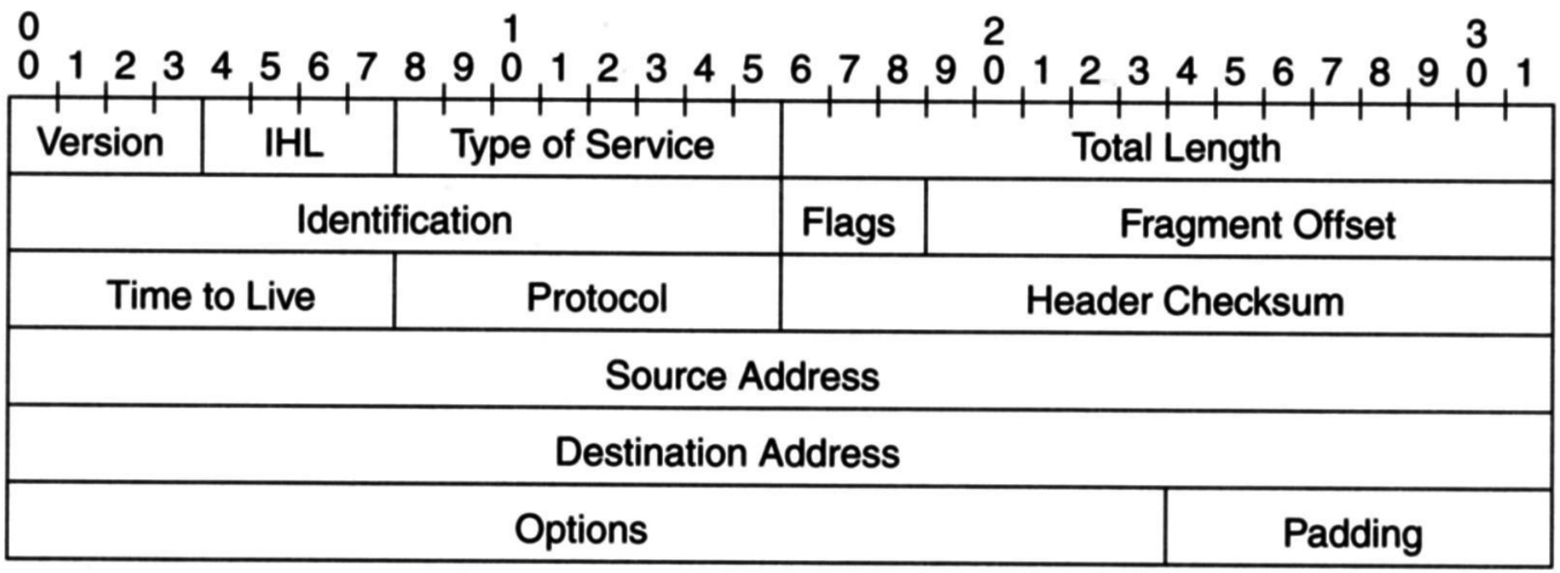


IP Host Processing…



Internet Protocol: Details

- IP header carries
  - Internet source and destination addresses
  - Specify parameters essential for routing (e.g., TTL, type of service, options)



IP Details…

- Header has fixed fields present in every packets and several options
- Header fields are aligned at 32-bit boundary
- **Version:** 4 = IPv4
- **IHL:** Internet header length -- in 32-bit words -- the length is 5 (when no options) and varies to 15 -- 40 bytes are allowed for IP options
- **Type of service:** defines packet's precedence and desired type of routing

IP Details…

- **Total length:** number of bytes contained in the packet -- including the IP header
- **TTL (time-to-live):** sets the upper limit on the number of routers through which a datagram can pass
  - each router decrements this by 1 as the packet passes
  - when it reaches 0, the packet is thrown away and an ICMP message is sent to the source

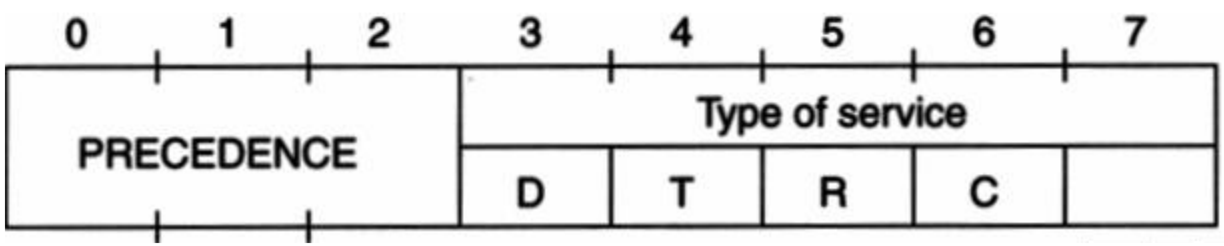
Internet Protocol

- **Identification, Flags, Fragment offset:** used b fragmentation and reassembly
- **Protocol:** use at the destination for demultiplexing the packet

Decimal	Keyword	Protocol
0		Reserved
1	ICMP	Internet Control Message
2	IGMP	Internet Group Management
3	GGP	Gateway-to-Gateway
4	IP	IP in IP (encapsulation)
5	ST	Stream
6	TCP	Transmission Control
8	EGP	Exterior Gateway Protocol
17	UDP	User Datagram
29	ISO-TP4	ISO Transport Protocol Class 4
38	IDPR-CMTP	IDPR Control Messager Transport Protocol
80	ISO-IP	ISO Internet Protocol (CLNP)
88	IGRP	IGRP
89	OSPF	Open Shortest Path First
255		Reserved

Internet Protocol

- Routing protocols normally use the “best” route -- there are several definitions of “best” -- cheapest, fastest, most reliable
- Type of service specifies the requirement of the application to the routing protocol



Internet Protocol

- Precedence indicator does not affect routing but queuing
  - several packets waiting for transmission on the same channel
  - highest precedence should in theory be transmitted first
- The “differentiated services” effort is refining this traditional interpretation of type-of-service

Internet Protocol

<u>Fragmentation and Reassembly:</u>

- Internetworking programs are expected to relay packets between heterogeneous networks
- Network technology implies a maximum packet size
- Incoming packet fragmented:

		IP header fields					Data field
Incoming packet:		Id=X,	L=4020,	DF=0,	MF=0,	offset=0	A---AB---BC---
Fragment	1:	Id=X,	L=1520,	DF=0,	MF=1,	offset=0	A---A
Fragment	2:	Id=X,	L=1520,	DF=0,	MF=1,	offset=1500	B---B
Fragment	3:	Id=X,	L=1020,	DF=0,	MF=0,	offset=3000	C---C

Internet Protocol

- Incoming fragment fragmented:

		IP header fields					Data field
Incoming Fragment	2:	Id=X,	L=1520,	DF=0,	MF=1,	offset=1500	B- - -B
Fragment	2a:	Id=X,	L= 520,	DF=0,	MF=1,	offset=1500	B- -
Fragment	2b:	Id=X,	L= 520,	DF=0,	MF=1,	offset=2000	- - -
Fragment	2c:	Id=X,	L= 520,	DF=0,	MF=1,	offset=2500	- -B

- Identification + source address uniquely identifies the packet fragment for the destination
- Receiver assembles all fragments with same ID according to the “offset”

Internet Protocol

- Host can’t reuse an identifier if there is a risk of fragmentation and new fragments mixing with old fragments
- Wait for the expiration of the fragments -- TTL
- With a packet size of 4k, this translates to about 17Mbps transfer rate -- clearly not adequate
- Correct way is to discovery the path MTU and use it as the max packet size -- no fragmentation

Internet Protocol

Path MTU discovery:

- sets the DF (don't fragment bit) in the IP header to discover if any router on the current path needs to fragment
- ICMP error message is returned by a router asked to forward an IP with DF set when MTU is less than datagram size
- this error message is used to decrease the datagram size until no error -- TCP path MTU discovery

Internet Protocol

<u>IP Options:</u>

- IP options field is used to carry specific functions
  - request specific routing for some packets, e.g., loose source routing, strict source routing
  - in source routing, the send specifies the route by specifying intermediate routers
- options are rarely used now.

Internet Protocol

<u>Options and header processing:</u>

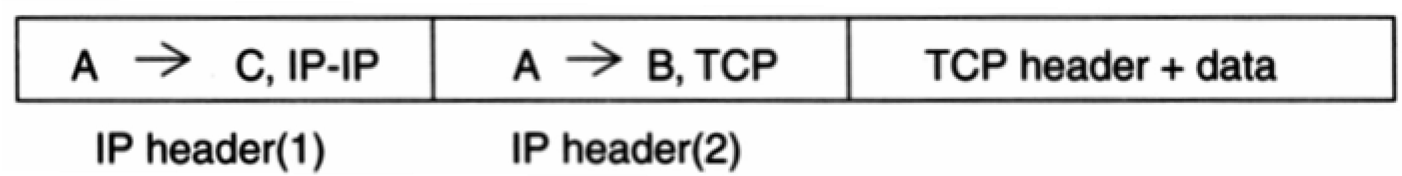
- IP options are used rarely because of the processing costs
- naive implementation of IP routing will perform
  - verify version field, checksum, compatibility checks, and parse any IP options
  - look for next hop for destination address considering type of service, interface and so on.
  - takes hundreds of instructions

Internet Protocol

- To speed up router processing
  - optimize the most commonly used case
  - without the IP options -- the header has five 32-bit words -- makes verification faster
  - frequently used routes can be cached -- to achieve Gigabit-per-second routing!
- Packet with options create problems -- they are processed with low priority than “normal” packets

Internet Protocol

- Because of the performance penalty for options -- alternatives are used for source routing
- One technique is called "encapsulation"
- Instead of specifying a loose source routing "from A to B through C" -- encapsulate a packet "from A to B" in another packet "from A to C"



- When C receives it, protocol type "IP in IP" denotes encapsulation -- unwraps the packet and forwards

Internet Protocol

- Cost at router C will be comparable, i.e., processing options and unwrapping
- Cost at intermediate routers A to C, and C to B are less with the encapsulation -- processed using the optimized methods -- no option processing

Internet Control Message Protocol

- IP is straightforward and simple –no feedback for diagnosing error conditions
- Internet Control Message Protocol (ICMP) does this feedback
  - layered on top of IP -- protocol type 1
  - all routers and hosts are expected to “speak” this protocol

ICMP Details…

- Most ICMP packets are diagnostic info. sent back when a router destroys a packet -- e.g., destination unreachable, TTL expired
- ICMP also defines a echo function used for testing connectivity
- ICMP does not make IP datagram service reliable

ICMP Details…

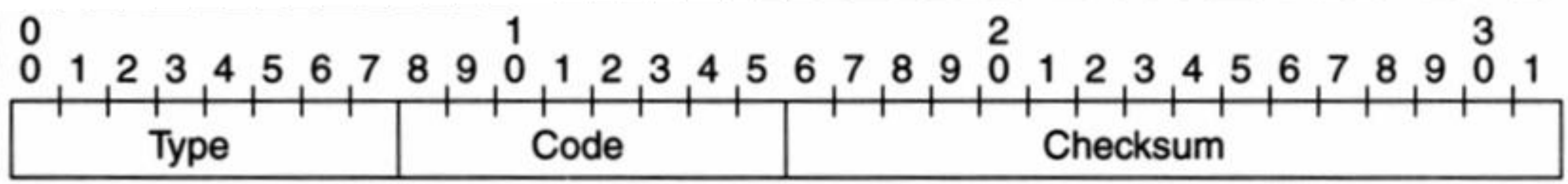
- Just provides feedback about network problems
- ICMP is carried on IP datagrams -- these packets themselves could be victim of errors
- No ICMP error is triggered by an ICMP message

ICMP Details…

- All ICMP messages start with a common 32-bit ICMP header

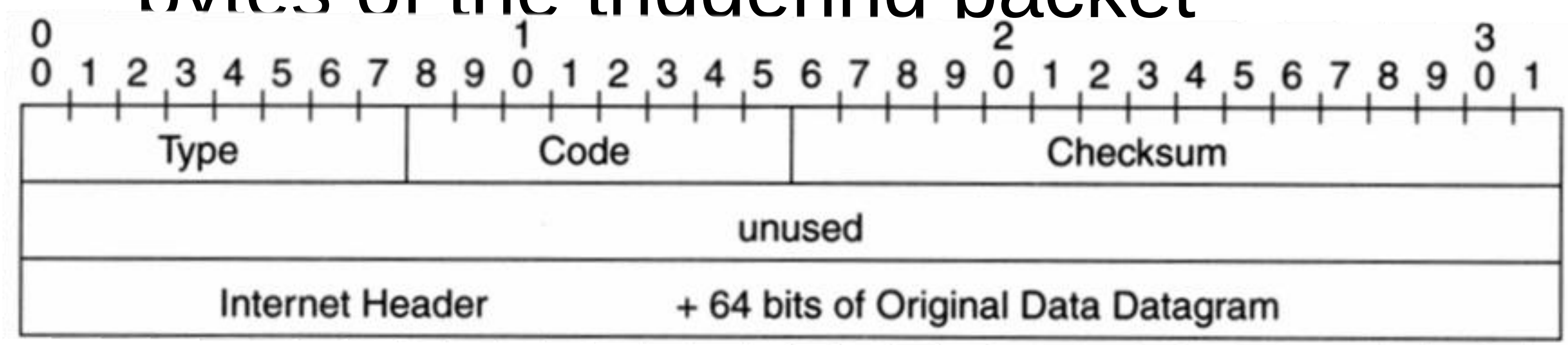
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect
8	Echo
9	Router Advertisement
10	Router Solicitation
11	Time Exceeded
12	Parameter Problem
13	Timestamp
14	Timestamp Reply
15	Information Request
16	Information Reply

— ICMP message types —



ICMP Details…

- Reporting “operational” problems such as time exceeded, destination unreachable, source quench is the most common use
- These packets have the same format -- includes the entire header of and 8 bytes of the triggering packet



ICMP Details…

- Destination unreachable messages are sent when a router cannot forward a packet

  0 = net unreachable
  1 = host unreachable
  2 = protocol unreachable
  3 = port unreachable
  4 = fragmentation needed and DF set
  5 = source route failed

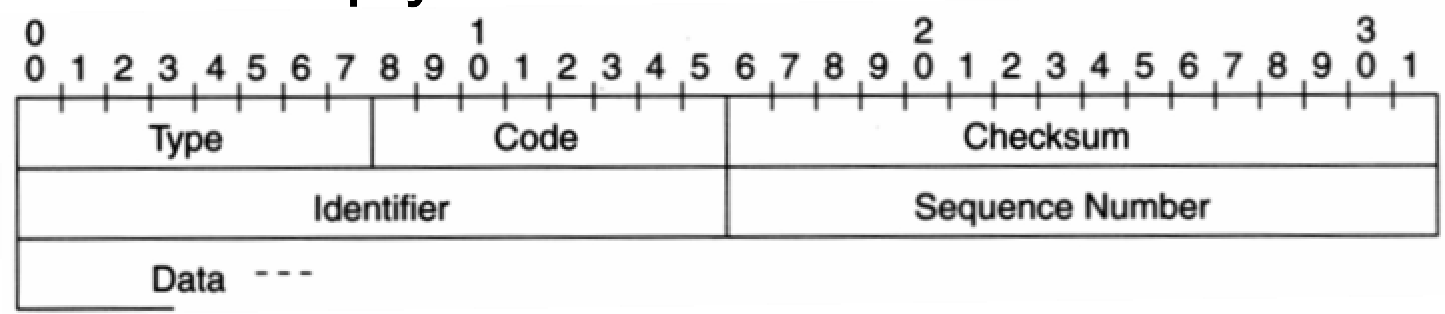
- Time exceeded message sent when a packet is destroyed because TTL expired

ICMP Details…

- Source quench messages are sent by a router that detects congestion -- source is supposed to reduce sending rate when it receives this message
- Parameter problem message is sent by a router that find an error in the encoding of the IP header

ICMP Ping

- When a router/host receives an ICMP message of type echo, it responds by an “echo reply”



- Reply is derived from request by swapping the IP header’s source and destination address -- replacing ECHO by ECHO-REPLY and computing new checksums

ICMP Traceroute

- Traceroute tries to discover intermediate routers
- Send packet with TTL = 1; first router decrements TTL to 0, destroy the packet and send back a “TTL expired” ICMP message
- Source address of the ICMP identifies the first router
- Next message is sent with TTL+1 for second router
- Packet sent for an unused UDP port an ICMP port unreachable -- message is sent back

ICMP Router Discovery

- To send a packet – a host needs nexthop
  - Test whether packet destination in current subnet
  - If not, forward packet to a router so that packet can reach the destination
- When there are multiple routers connected to the local network -- host should select the one nearest to the destination

ICMP Router Discovery …

- How to discovery local routers?
  - Read from a config file – static solution
  - Dynamic solution – zero admin overhead – discovery procedure
- Router discovery using special ICMP messages
- Routers send “router advertisements” at regular intervals
- Hosts trigger this by sending “router solicitations”

ICMP Router Discovery …

- Router advertisements contain a list of routers with a preference notation
- Hosts select router with highest preference