Internetworking: philosophy, addressing, forwarding, resolution, fragmentation

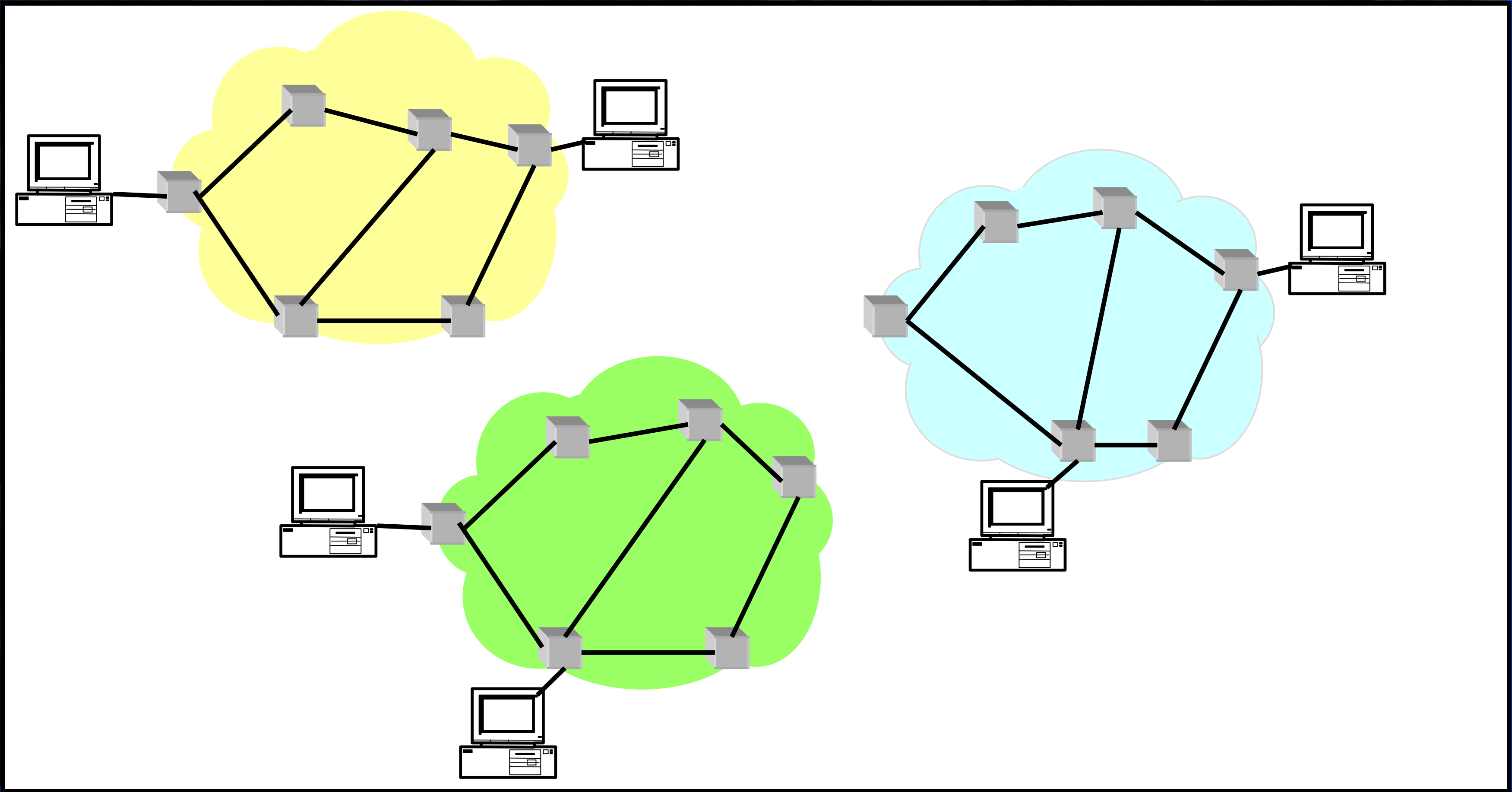
Shivkumar Kalyanaraman
Rensselaer Polytechnic Institute
shivkuma@ecse.rpi.edu
http://www.ecse.rpi.edu/Homepages/shivkuma

Based in part upon the slides of Prof. Raj Jain (OSU), J.Kurose (Umass), S. Keshav (Cornell), I.Stoica (UCB), S. Deering (Cisco)

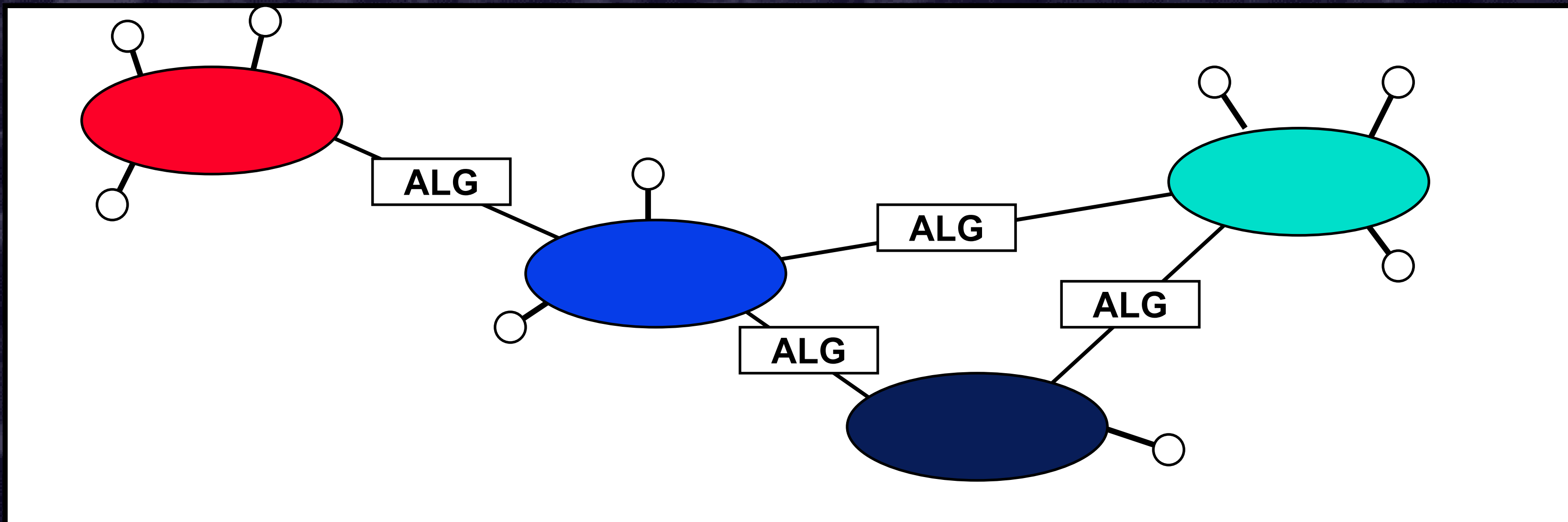
- Internetworking: heterogeneity & scale
- IP solution:
  - Provide new packet format and overlay it on subnets.
  - *Ideas:* Hierarchical address, address resolution, fragmentation/re-assembly, packet format design, forwarding algorithm etc
- Chapter 3,4,5,7 in Comer
- **Reading:** Clark: "The Design Philosophy of the DARPA Internet Protocols":
- **Reading:** Cerf, Kahn: "A Protocol for Packet Network Intercommunication"
- **Reading:** Mogul etal: "Fragmentation Considered Harmful"
- **Reading:** Addressing 101: Notes on Addressing: In PDF | In MS Word
- **Reading:** Notes for Protocol Design, E2e Principle, IP and Routing: In PDF
- Reference: RFC 791: Internet Protocol (IP) Spec.: In HTML

The Problem

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



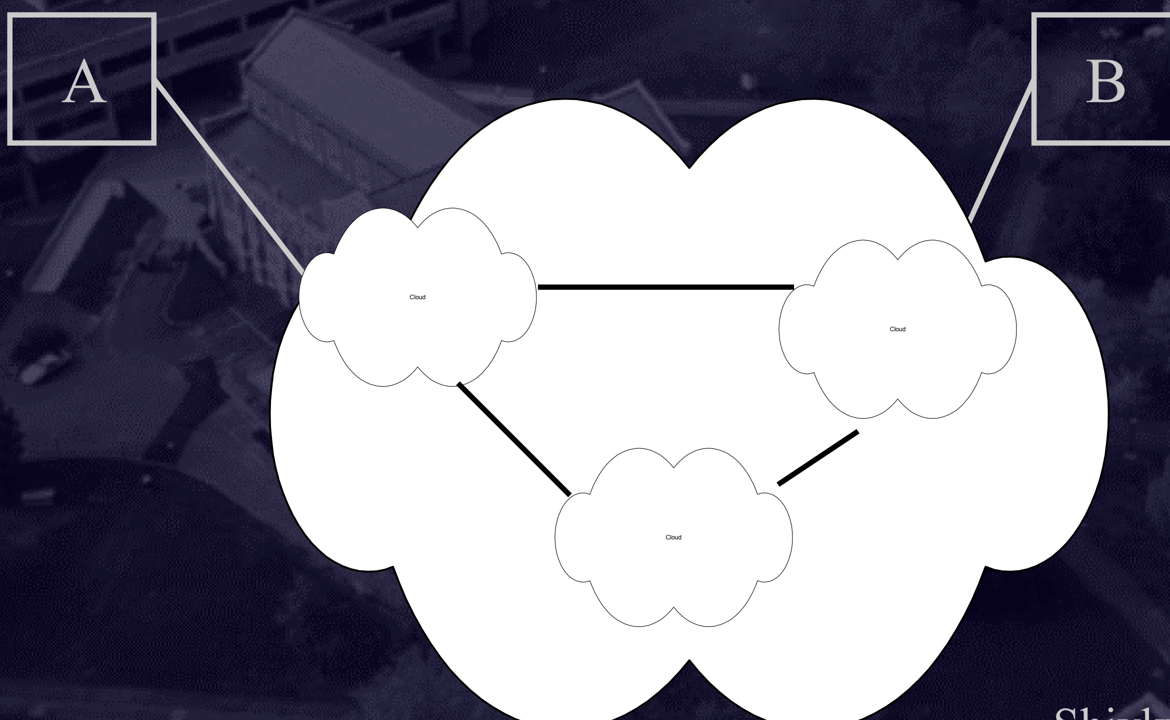
A Translation-based Solution



- application-layer gateways
  - inevitable loss of some semantics
  - 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

The Internetworking Problem

- Two nodes communicating across a "***network of networks***"…
  - How to <u>transport</u> packets through this heterogeneous mass ?



Declared Goal

- "…both economic and technical considerations lead us to prefer that the <u>interface</u> be as <u>simple</u> and <u>reliable</u> as possible and <u>deal primarily</u> with passing data between networks using different packet switching strategies"

***V. G. Cerf and R. E. Kahn*, 1974**

The Challenge: Heterogeneity

- Share resources of different packet switching networks → interconnect existing networks
- … but, packet switching networks differ widely
  - different *services*
    - e.g., degree of reliability
  - different *interfaces*
    - e.g., length of the packet that can be transmitted, address format
  - different *protocols*
    - e.g., routing protocols

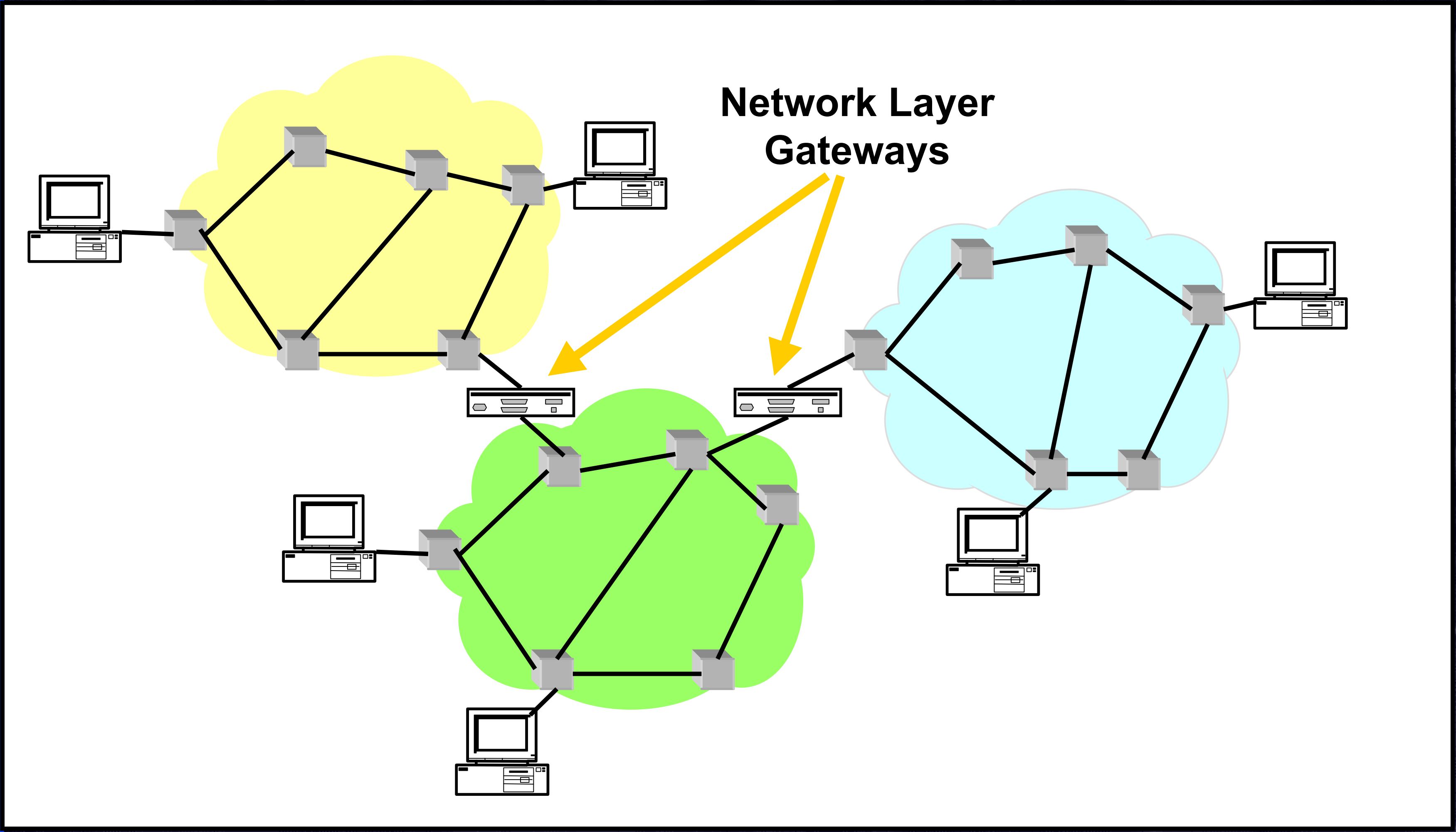
The Challenge: Scale

- Allow universal interconnection
  - Mantra: Connectivity is its own reward
- … but, core protocols had scalability issues
  - Routing algorithms were limited in the number of nodes/links they could handle and were unstable after a point
  - Universal addressing to go with routing
  - As large numbers of users are multiplexed on a shared system, a congestion control paradigm is necessary for stability
  - No universal, scalable naming system…

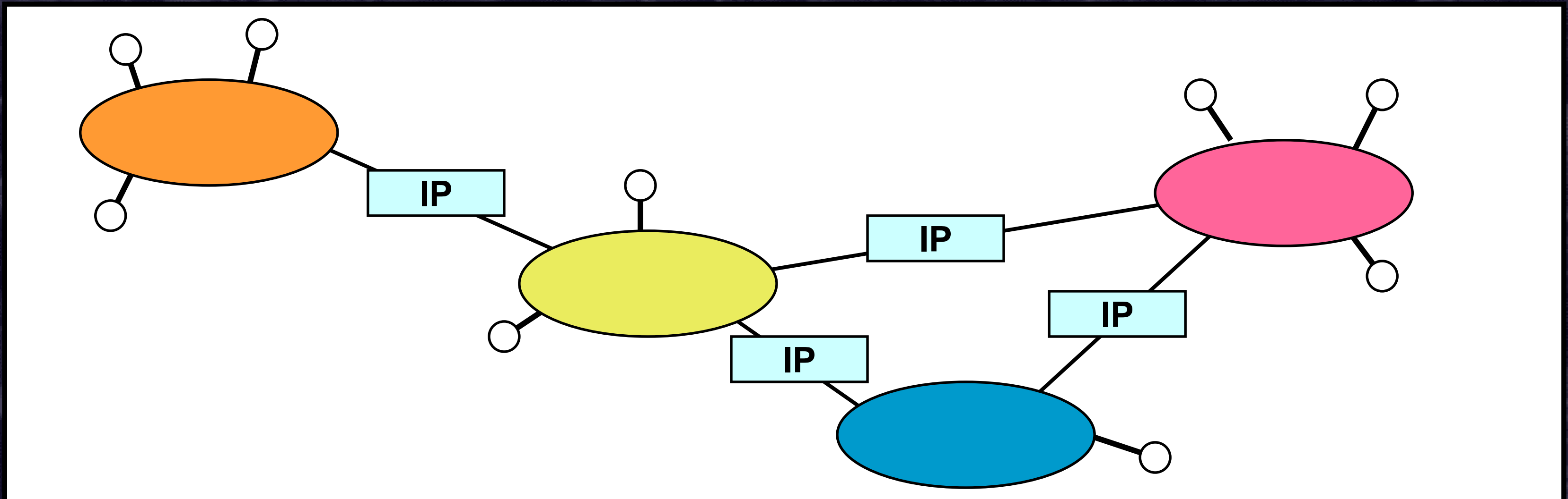
The Internetworking *Problem*

- Problems: *heterogeneity and scaling*
- *Heterogeneity*:
  - How to interconnect a large number of disparate *networks* ? (lower layers)
  - How to support a wide variety of *applications* ? (upper layers)
- *Scaling:*
  - How to support a large number of *end-nodes* and *applications* in this interconnected network ?

Solution



The IP Solution …



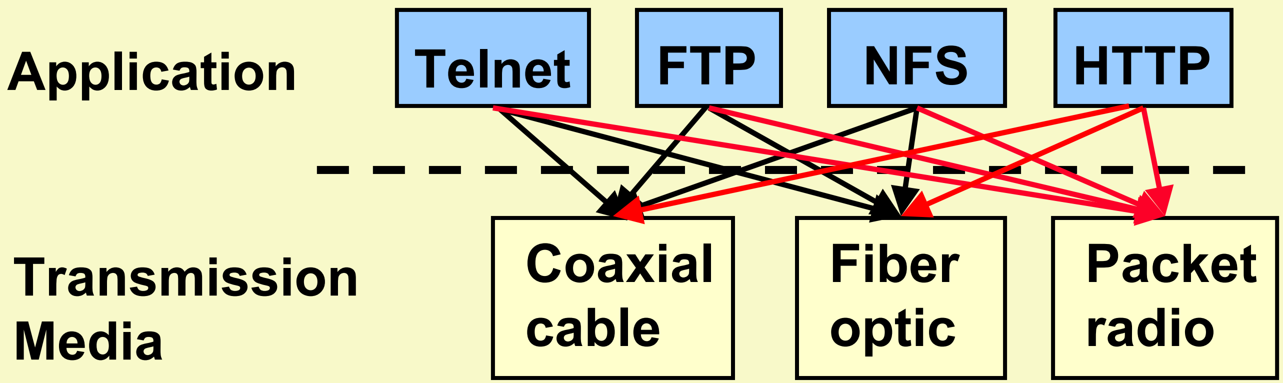
- internet-layer gateways & global addresses
- simple, application-independent, lowest denominator network service: best-effort datagrams
- stateless gateways could easily route around failures
- with application-specific knowledge out of gateways:
  - NSPs no longer had monopoly on new services
  - Internet: a platform for rapid, competitive innovation

Network-layer Overlay model

- Define a *new* protocol (IP) and map all applications/networks to IP
    - Require *only one mapping* (IP -> new protocol) when a new protocol/app is added
    - *Global address space* can be created for universal addressibility and scaling

Before IP

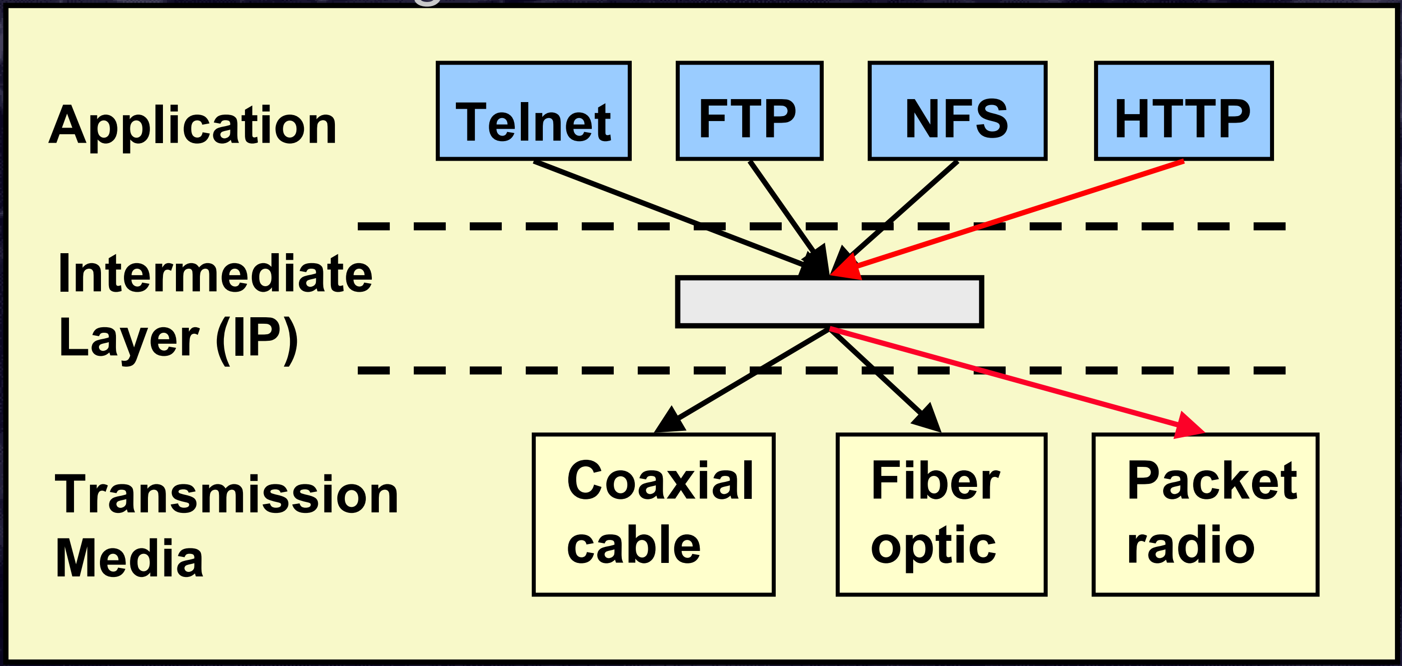
(FTP – File Transfer Protocol, NFS – Network File Transfer, HTTP – World Wide Web protocol)



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

IP

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



What About the Future ?

- Internet is running out of addresses
- Solutions
  - Classless Inter Domain Routing (CIDR)
  - Network Address Translator (NATs)
  - Dynamic Address Assignments
  - …
  - IPv6
- Why not variable-sized addresses?

Service to Apps

- Unbounded but finite length messages
  - byte streaming (What are the advantages?)
- Reliable and in-sequence delivery
- Full duplex
- Solution: Transmission Control Protocol (TCP)

Original TCP/IP (Cerf & Kahn)

- No separation between transport (TCP) and network (IP) layers
- One common header
  - use ports to multiplex multiple TCP connections on the same host

32	32	16	16	8n
Source/Port	Source/Port	Window	ACK	Text

- Byte-based sequence number (Why?)
- Flow control, but not congestion control

Today's TCP/IP

- Separate transport (TCP) and network (IP) layer (why?)
  - split the common header in: TCP and UDP headers
  - fragmentation reassembly done by IP
- Congestion control (later in class)

IP Datagram Format

0 4 8 16 32

Vers	H Len	TOS	Total Length

Identification	Flags	Fragment Offset

Time to live	Protocol	Header Checksum

Source IP Address

Destination IP Address

IP Options (if any)	Padding

Data

IP Datagram Format (Continued)

- First Word purpose: info, variable size header & packet.
  - Version (4 bits)
  - Internet header length (4 bits): *units of 32-bit words.* Min header is 5 words or 20 bytes.
  - Type of service (TOS: 8 bits): Reliability, precedence, delay, and throughput. Not widely supported
  - Total length (16 bits): header + data. *Units of bytes.* Total must be less than 64 kB.

IP Header (Continued)

- 2nd Word Purpose: fragmentation
  - Identifier (16 bits): Helps uniquely identify the datagram between any source, destination address
  - Flags (3 bits): More Flag (MF):more fragments
    Don't Fragment (DF)
    Reserved
  - Fragment offset (13 bits): *In units of 8 bytes*

IP Header (Continued)

- Third word purpose: demuxing, error/looping control, timeout.
  - Time to live (8 bits): Specified in router hops
  - Protocol (8 bits): Next level protocol to receive the data: for de-multiplexing.
  - Header checksum (16 bits): 1's complement sum of all 16-bit words in the header.
    - Change header => modify checksum using 1's complement arithmetic.

Recall: Signed Representations

Sign Magnitude	One's Complement	Two's Complement
000 = +0	000 = +0	000 = +0
001 = +1	001 = +1	001 = +1
010 = +2	010 = +2	010 = +2
011 = +3	011 = +3	011 = +3
100 = - 0	100 = - 3	100 = - 4
101 = - 1	101 = - 2	101 = - 3
110 = - 2	110 = - 1	110 = - 2
111 = - 3	111 = - 0	111 = - 1

One’s complement addition:

normal addition & increment of the total if there was a carry.

Eg: 110 (i.e. -1) + 111 (i.e. 0) = 101 +1 = 110 (i.e. –1)

Header Format (Continued)

- Source Address (32 bits): Original source. Does not change along the path
- Destination. Address (32 bits): Final destination. Does not change along the path.
- Options (variable length): Security, source route, record route, stream id (used for voice) for reserved resources, timestamp recording
- Padding (variable length): Makes header length a multiple of 4
- Payload Data (variable length): Data + header $\leq$ 65,535 bytes

TCP Header

0	4	10	16 31
Source port			Destination port
Sequence number			
Acknowledgement			
HdrLen		Flags	Advertised window
Checksum			Urgent pointer
Options (variable)			

- Sequence number, acknowledgement, and advertised window – used by sliding-window based flow control
- Flags (selected):
  - SYN, FIN – establishing/terminating a TCP connection
  - ACK – set when Acknowledgement field is valid
  - RESET – abort connection

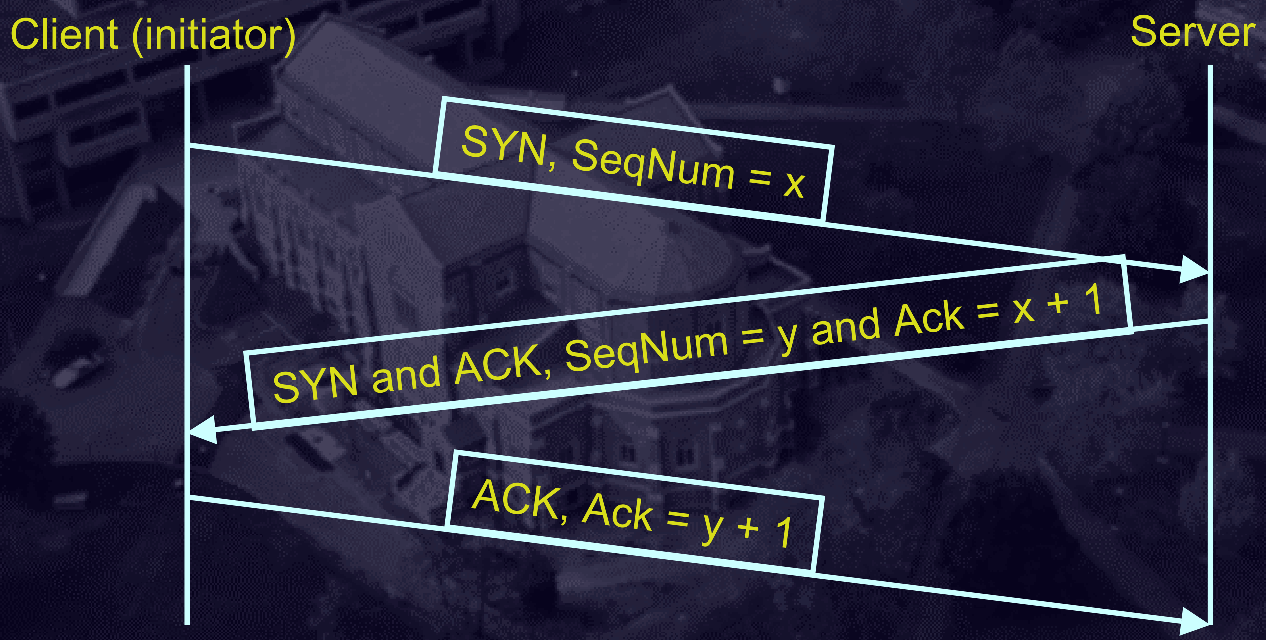
TCP Header (Cont)

- Checksum – 1's complement and is computed over
  - TCP header
  - TCP data
  - Pseudo-header (from IP header)
    - Note: breaks the layering!

Source address		
Destination address		
0	Protocol (TCP)	TCP Segment length

TCP Connection Establishment

- Three-way handshake
  - Goal: agree on a set of parameters: the start sequence number for each side



IP Forwarding (I)

- *Source & Destination in same network (direct connectivity)*
  - Recognize that destination IP address is on same network. [1]
  - Find the destination LAN address. [2]
  - Send IP packet encapsulated in LAN frame directly to the destination LAN address.
    - Encapsulation => source/destination IP addresses don't change

IP Forwarding (II)

- B) *Source & Destination in different networks (indirect connectivity)*
  - Recognize that destination IP address is not on same network. [1]
  - 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. [2]

[1] Addressing

- *[1] 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*

[2] Address Resolution

- *[2]: How to find the LAN address corresponding to an IP address ?*
  - *Address Resolution Problem.*
  - Solution: ARP, RARP (later in this slide set)

IP Forwarding: Example Scenario

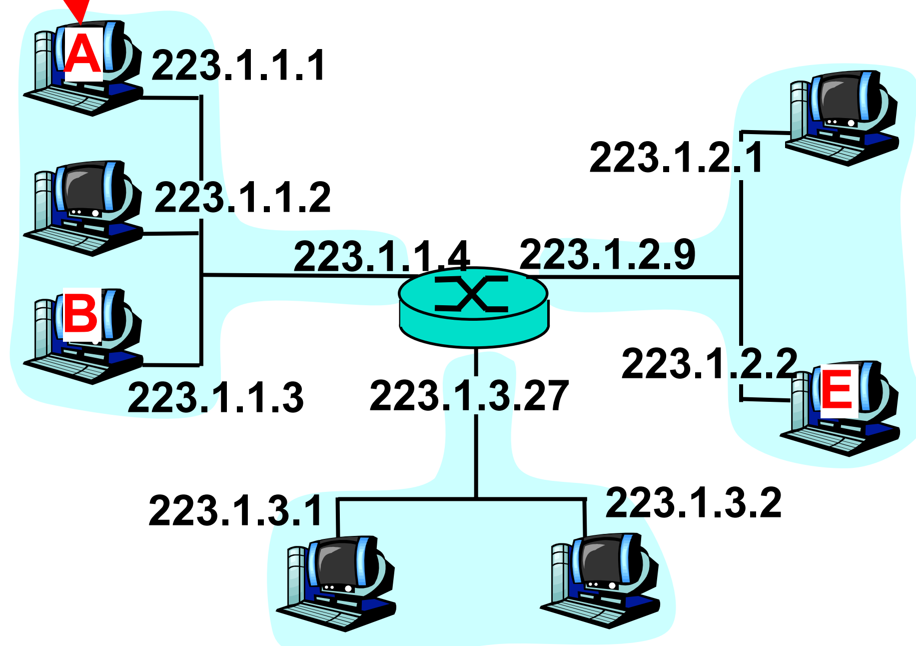
IP datagram:

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)

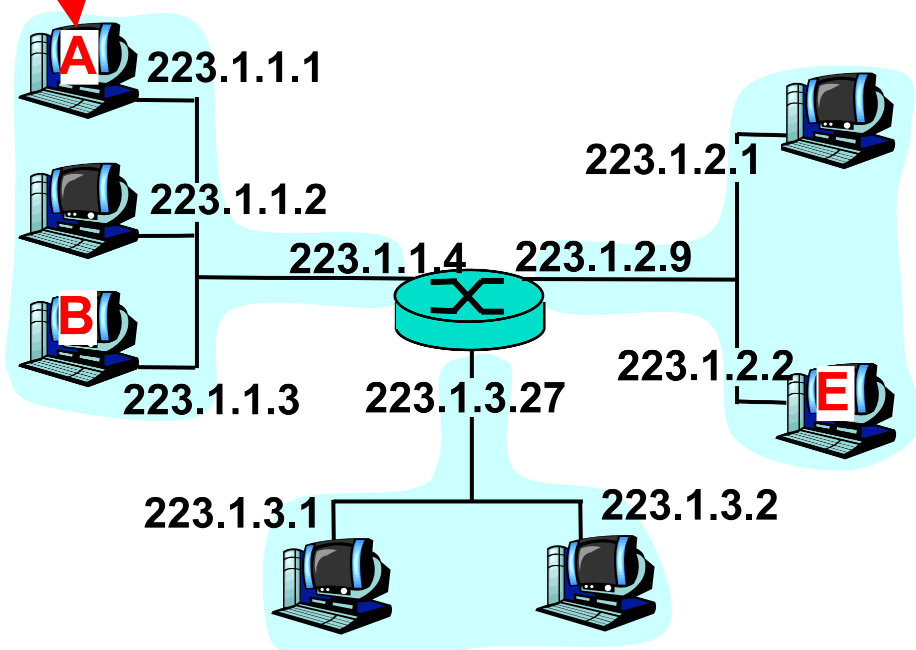
misc fields	223.1.1.1	223.1.1.3	data

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 <u>directly connected</u>



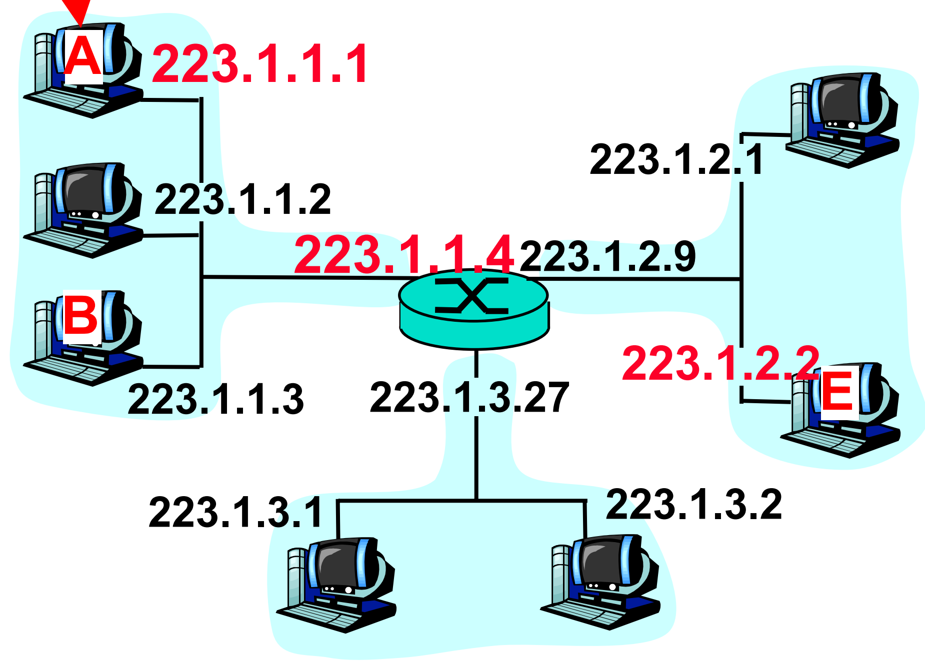
IP Forwarding (Indirect): Step 1

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…..



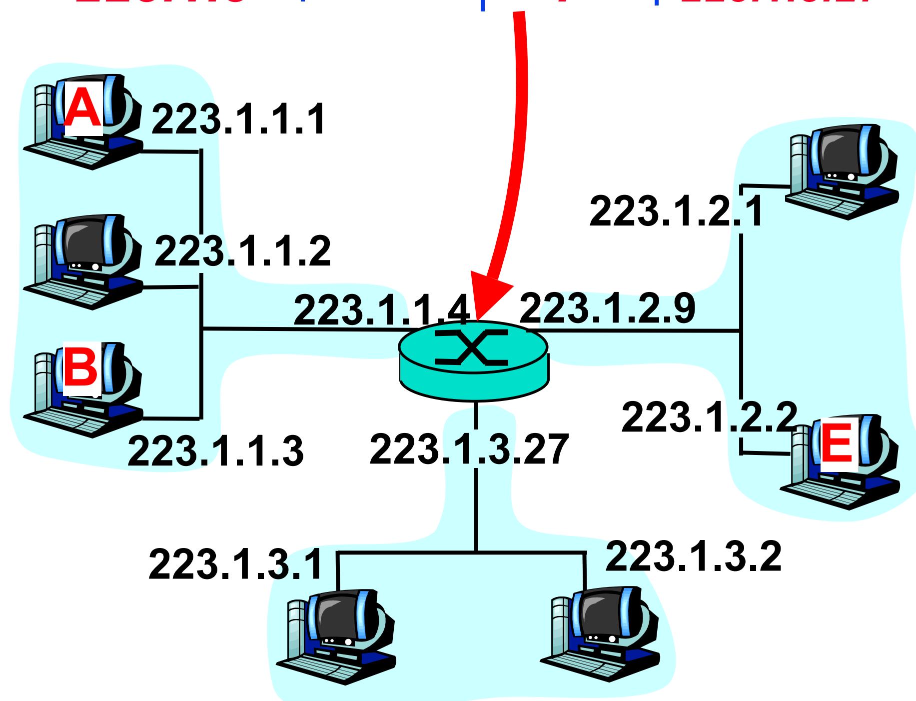
IP Forwarding (Indirect): Step 2

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*

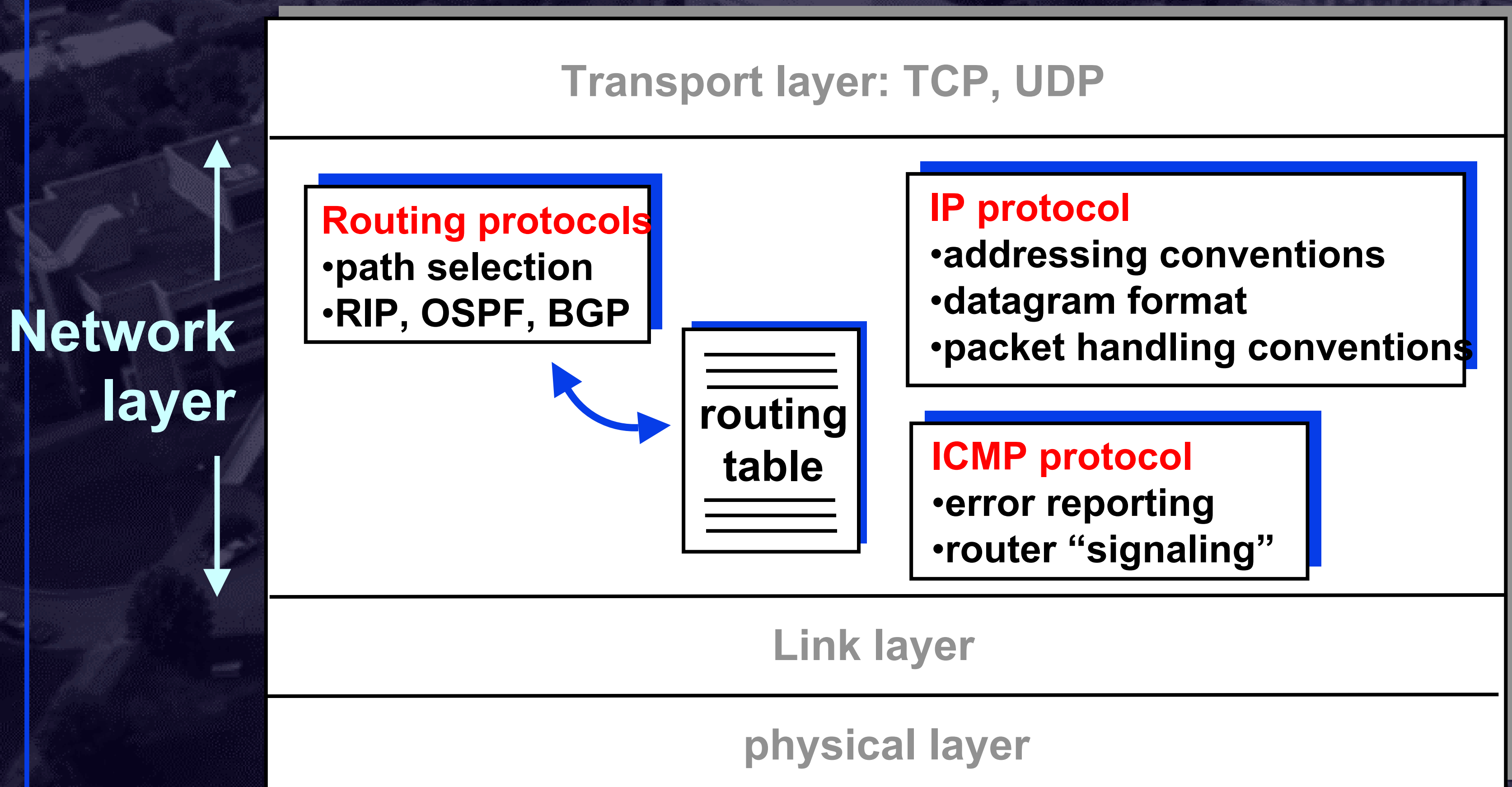
Arriving at 223.1.4, destined for 223.1.2.2

- 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



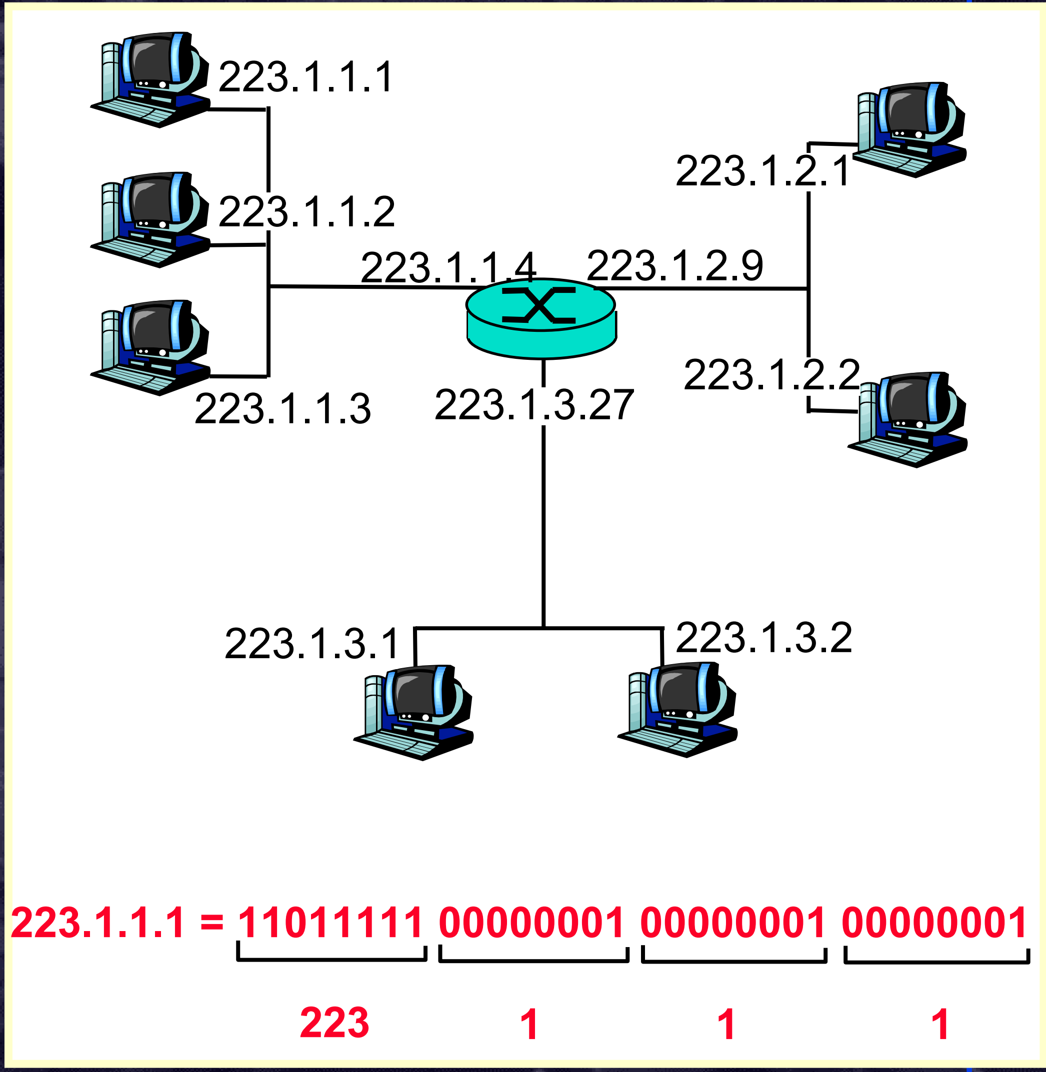
The Internet Network layer

Host, router network layer functions:



IP Addressing: introduction

- IP address: 32-bit identifier for host, router *interface*
- *Interface:* connection between host, router and physical link
  - router's typically have multiple interfaces
  - host may have multiple interfaces
  - IP addresses associated with interface, not host, router
- *Hosts in the same network have same network ID*



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
- Class E: Reserved.

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

Understanding Prefixes and Masks

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

RFC 1519: Classless Inter-Domain Routing (CIDR)

Pre-CIDR: Network ID ended on 8-, 16, 24- bit boundary

CIDR: Network ID can end at any bit boundary

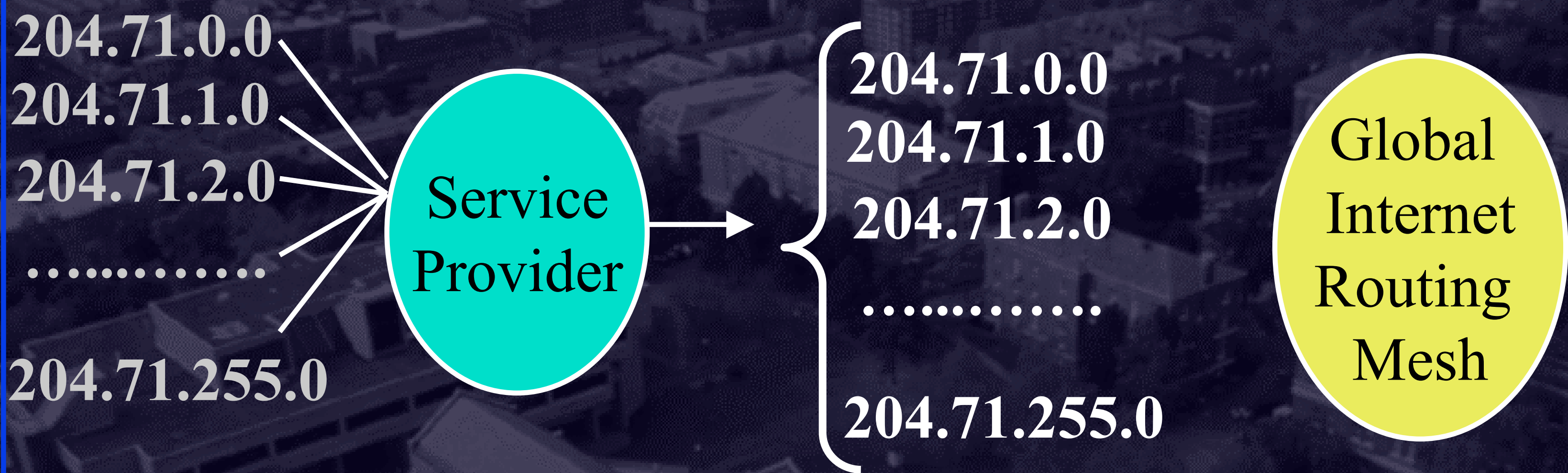
IP Address : 12.4.0.0 IP Mask: 255.254.0.0

Address	00001100	00000100	00000000	00000000
Mask	11111111	11111110	00000000	00000000

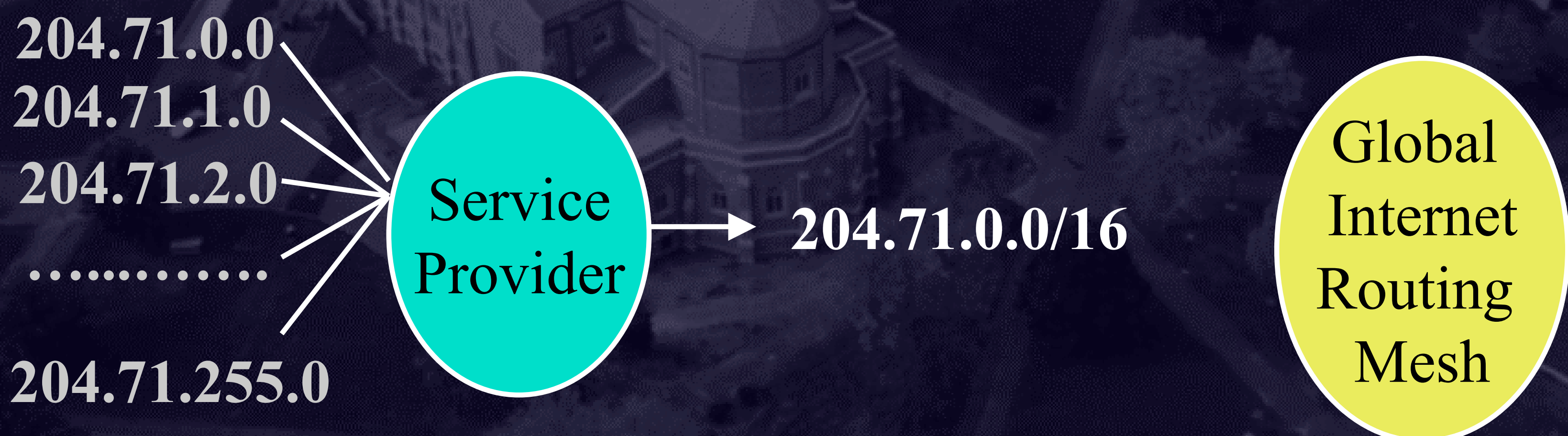
|← **Network Prefix** →|← **for hosts** →|

Usually written as 12.4.0.0/15, a.k.a "supernetting"

Inter-domain Routing Without CIDR



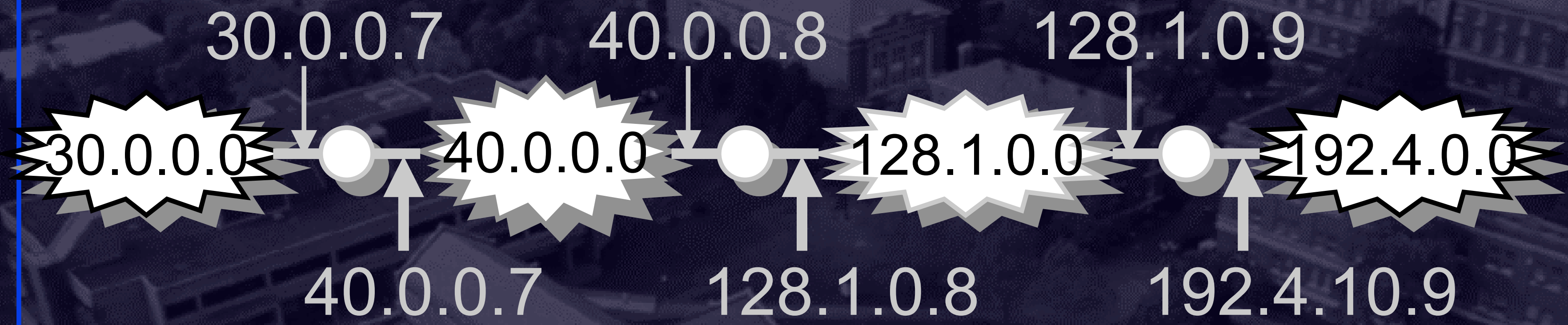
Inter-domain Routing With CIDR



Implication on Forwarding: Subnet

- Route table lookup:
  - IF ((Mask[i] & Destination Addr) = =
    Destination[i])
    Forward to NextHop[i]
- In theory, subnet mask can end on any bit.
- In practice, mask must have contiguous 1s followed by contiguous zeros. Routers do not support other types of masks.
- So, (Address, Mask) = (12.4.0.0, 255.254.0.0) may be written as 12.4.0.0/15

Route Table Lookup: Subnet Example

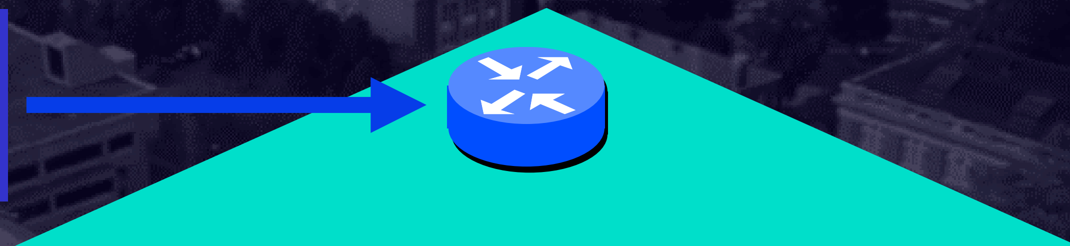


Destination	Mask	Next Hop
30.0.0.0	255.0.0.0	40.0.0.7
40.0.0.0	255.0.0.0	Deliver direct
128.1.0.0	255.255.0.0	Deliver direct
192.4.10.0	255.255.255.0	128.1.0.9

Implication on Forwarding: Supernetting (CIDR)

- Longest Prefix Match (Classless) Forwarding

Destination =12.5.9.16
----
payload



	Prefix	Next Hop	Interface
OK	0.0.0.0/0	10.14.11.33	ATM 5/0/9
better	12.0.0.0/8	10.14.22.19	ATM 5/0/8
even better	12.4.0.0/15	10.1.3.77	Ethernet 0/1/3
best!	12.5.8.0/23	attached	Serial 1/0/7

IP Forwarding Table

Variable Length Subnet Mask (VLSM)

- ***Basic subneting:*** refers to a fixed mask in addition to natural mask (i.e. class A, B etc).
  - I.e. only a single mask (eg:: 255.255.255.0) can be used for all networks covered by the natural mask.
- ***VLSM:*** Multiple different masks possible in a single class address space.
  - Eg: 255.255.255.0 and 255.255.254.0 could be used to subnet a single class B address space.
  - Allows more efficient use of address space.

Example: Address Block: 128.20.224.0/20.

Networks: 2 of size 1000 nodes each;

2 of size 500 nodes each;

3 of size 250 nodes each.

4 of size 50 nodes each. What are the allocations?

1000 nodes need 10 bits => 32 –10 =22 bit prefixes needed

128.20.1110 00 00. 0000 0000/22 = 128.20.224.0/22

128.20.1110 01 00. 0000 0000/22 = 128.20.228.0/22

500 nodes need 9 bits => 32 –9 =23 bit prefixes needed

128.20.1110100 0. 0000 0000/23 = 128.20.232.0/23

128.20.1110101 0. 0000 0000/23 = 128.20.234.0/23

250 nodes need 8 bits => 32 –8 =24 bit prefixes needed

128.20.11101100. 0000 0000/24 = 128.20.236.0/24

128.20.11101101. 0000 0000/24 = 128.20.237.0/24

128.20.11101110. 0000 0000/24 = 128.20.238.0/24

50 nodes need 6 bits => 32 –6 =26 bit prefixes needed

Addressing Summary

- Unique IP address per interface
- Classful (A,B,C) => address allocation not efficient
- Hierarchical => smaller routing tables
- Provision for broadcast, multicast, loopback addresses
- Subnet masks allow "subnets" within a "network" => improved address allocation efficiency
- Supernet (CIDR) allows variable sized network ID allocation
- VLSM allows further efficiency

Forwarding Summary

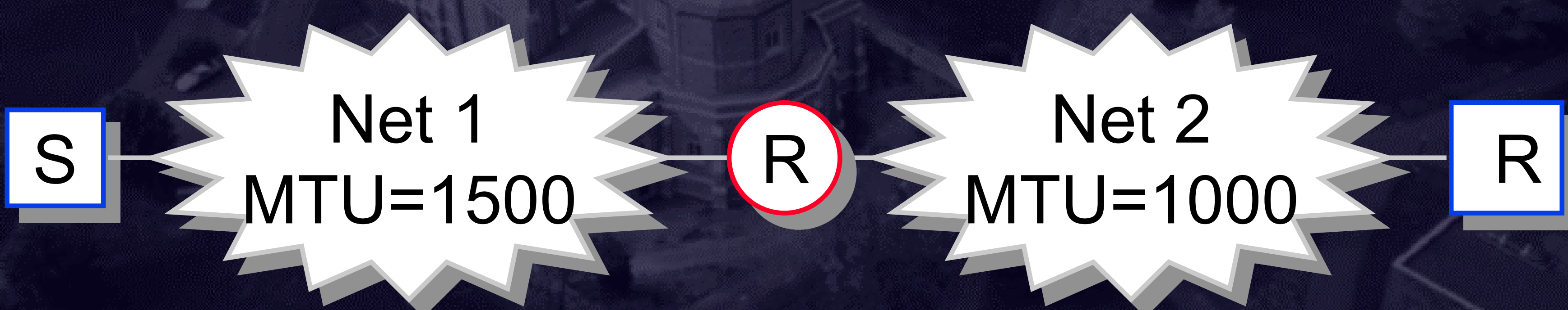
- <u>Forwarding:</u>
  - Simple *"next-hop"* forwarding.
  - Last hop forwards directly to destination
  - *Best-effort delivery :* No error reporting. Delay, out-of-order, corruption, and loss possible => problem of higher layers!
  - Forwarding vs routing: tables setup by separate algorithm (s)

What IP does NOT provide

- End-to-end data reliability & flow control (done by TCP or application layer protocols)
- Sequencing of packets (like TCP)
- Error detection in payload (TCP, UDP or other transport layers)
- Error reporting (ICMP)
- Setting up route tables (RIP, OSPF, BGP etc)
- Connection setup (it is connectionless)
- Address/Name resolution (ARP, RARP, DNS)
- Configuration (BOOTP, DHCP)
- Multicast (IGMP, MBONE)

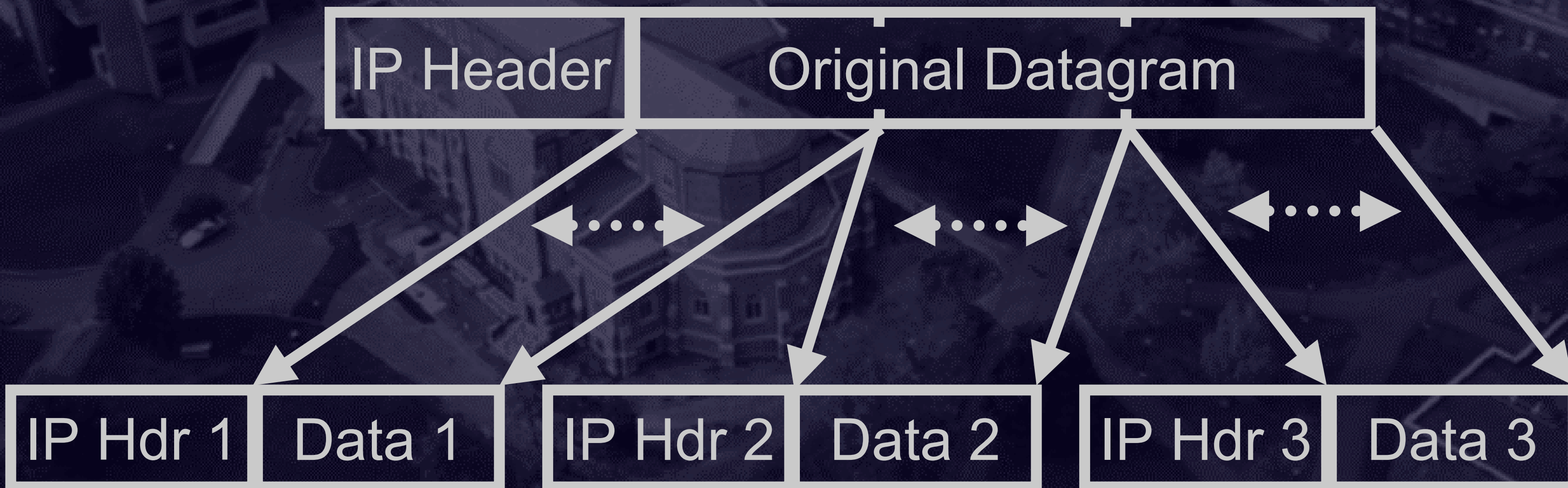
Maximum Transmission Unit

- Each subnet has a *maximum frame size*
  Ethernet: 1518 bytes
  FDDI: 4500 bytes
  Token Ring: 2 to 4 kB
- Transmission Unit = IP datagram (data + header)
- Each subnet has a maximum IP datagram length (header + payload) = MTU



Fragmentation

- Datagrams larger than MTU are fragmented
- Original header is copied to each fragment and then modified (fragment flag, fragment offset, length,...)
- Some option fields are copied (see RFC 791)



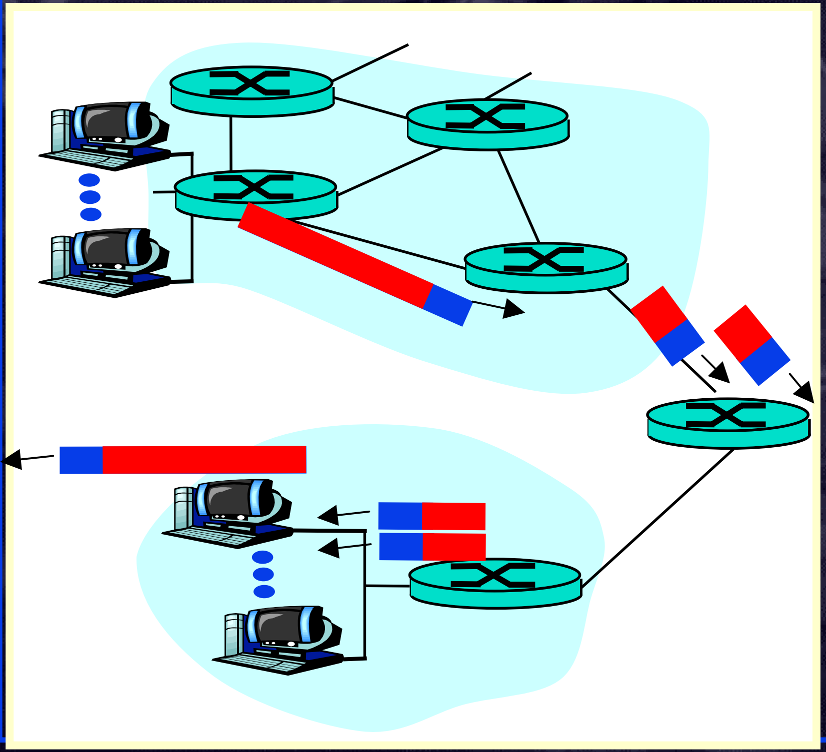
Fragmentation Example

MTU = 1500B — MTU = 280B

IHL = 5, ID = 111, More = 0
Offset = 0W, Len = 472B

IHL=5, ID = 111, More = 1
Offset = 0W, Len = 276B

IHL=5, ID = 111, More = 0
Offset = 32W, Len = 216B



Fragmentation Example (Continued)

- Payload size 452 bytes needs to be transmitted
- across a Ethernet (MTU=1500B) and a SLIP line (MTU=280B)
- Length = 472B, Header = 20B => Payload = 452B
- Fragments need to be multiple of 8-bytes.
  - Nearest multiple to 260 (280 -20B) is 256B
  - First fragment length = 256B + 20B = 276B.
  - Second fragment length = (452B- 256B) + 20B = 216B

Reassembly

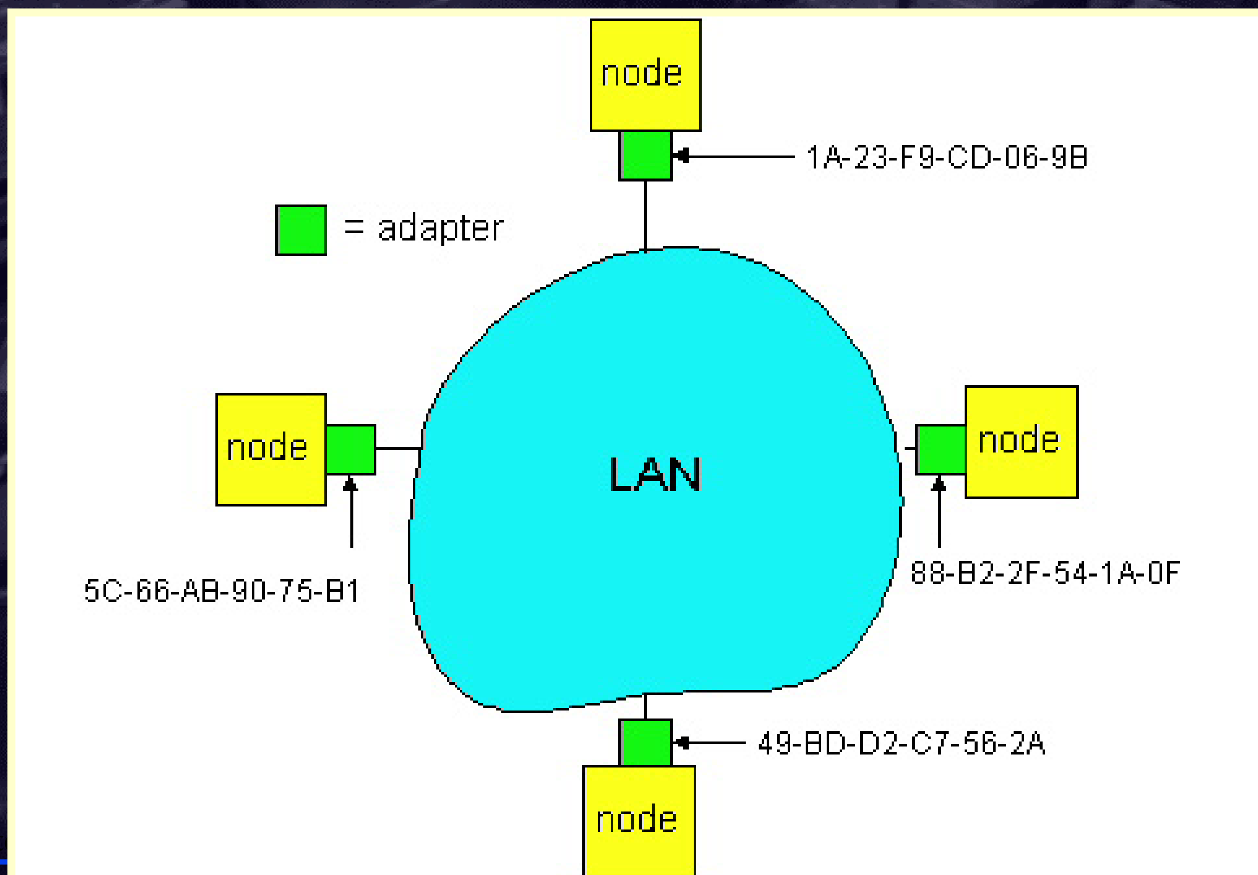
- Reassembly only at the final destination
- Partial datagrams are discarded after a timeout
- Fragments can be further fragmented along the path. Subfragments have a format similar to fragments.
- Minimum MTU along a path ⇒ Path MTU

Further notes on Fragmentation

- Performance: single fragment lost => entire packet useless. Waste of resources all along the way. Ref: Kent & Mogul, 1987
- Don't Fragment (DF) bit set => datagram discarded if need to fragment. ICMP message generated: may specify MTU (default = 0)
- Used to determine Path MTU (in TCP & UDP)
- The transport and application layer headers do not appear in all fragments. Problem if you need to peep into those headers.

Resolution Problems and Solutions

- *Indirection* through addressing/naming => requires address/name *resolution*
- Problem is to *map* destination layer N address to its layer N-1 address to allow packet transmission in layer N-1.



Resolution Problems and Solutions (Continued)

- ***1. Direct mapping:*** Make the physical addresses equal to the host ID part.
  - Mapping is easy.
  - Only possible if admin has power to choose both IP and physical address.
  - Ethernet addresses come preassigned (so do part of IP addresses!).
  - Ethernet addresses are 48 bits vs IP addresses which are 32-bits.

ARP techniques (Continued)

- **2: Table Lookup:**
  Searching or indexing to get MAC addresses
  - Similar to lookup in /etc/hosts for names
  - Problem: change Ethernet card => change table

IP Address	MAC Address
197.15.3.1	0A:4B:00:00:07:08
197.15.3.2	0B:4B:00:00:07:00
197.15.3.3	0A:5B:00:01:01:03

ARP techniques (Continued)

- 3. **Dynamic Binding: ARP**
  - The host *broadcasts* a request: "What is the MAC address of 127.123.115.08?"
  - The host whose IP address is 127.123.115.08 replies back: "The MAC address for 127.123.115.08 is 8A-5F-3C-23-45-56_{16}"
  - ARP responses cached; LRU + Entry Timeout
- All three methods are allowed in TCP/IP networks.

ARP Message Format

0	8	16	24	32
H/W Address Type		Protocol Address Type		
H/W Adr Len	Prot Adr Len	Operation		
Sender's h/w address (6 bytes)				
		Sender's Prot Address (4 bytes)		
		Target h/w address (6 bytes)		
Target Protocol Address (4 bytes)				

- Type: ARP handles many layer 3 and layer 2s
- Protocol Address type: 0x0800 = IP
- Operation: 1= Request, 2=Response
- ARP messages are sent directly to MAC layer

Back to Goals (Clark'88)

0 **Connect existing networks**
  - initially ARPANET and ARPA packet radio network

1. Survivability
   - ensure communication service even in the presence of network and router failures
2. Support multiple types of services
3. Must accommodate a variety of networks
4. Allow distributed management
5. Allow host attachment with a low level of effort
6. Be cost effective
7. Allow resource accountability

1. Survivability

- Continue to operate even in the presence of network failures (e.g., link and router failures)
  - as long as the network is not partitioned, two endpoint should be able to communicate…moreover, any other failure (excepting network partition) should be transparent to endpoints
- Decision: maintain state only at end-points (*fate-sharing*)
  - eliminate the problem of handling state inconsistency and performing state restoration when router fails
- Internet: stateless network architecture

2. Types of Services

- Add UDP to TCP to better support other types of applications
  - e.g., "real-time" applications
- This was arguably the main reasons for separating TCP and IP
- Provide datagram abstraction: lower common denominator on which other services can be built
  - service differentiation was considered (remember ToS?), but this has never happened on the large scale (Why?)

3. Variety of Networks

- Very successful (why?)
  - because the minimalist service; it requires from underlying network only to deliver a packet with a "reasonable" probability of success
- …does not require:
  - reliability
  - in-order delivery
- The mantra: IP over everything
  - Then: ARPANET, X.25, DARPA satellite network..
  - Now: ATM, SONET, WDM…

Other Goals

- Allow distributed management
  - Remember that IP interconnects networks
    - each network can be managed by a different organization
    - different organizations need to interact only at the boundaries
    - … but this model complicates routing
- Cost effective
  - sources of inefficiency
    - header overhead
    - retransmissions
    - Routing
  - …but routers relatively simple to implement (especially software side)

Other Goals (Cont)

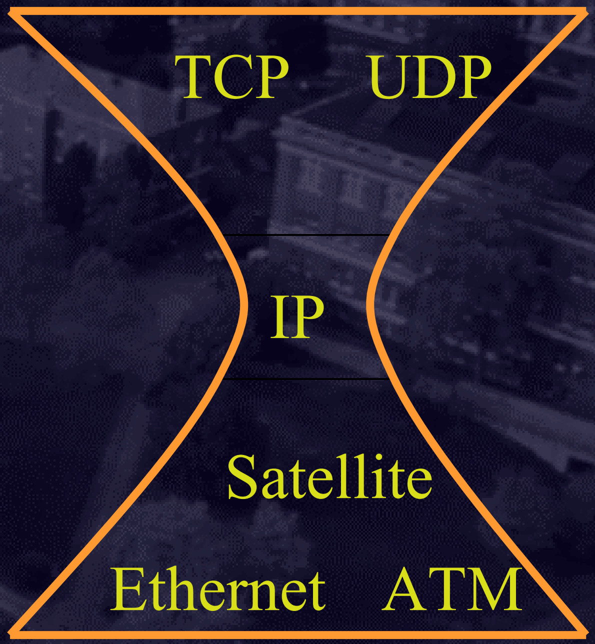
- Low cost of attaching a new host
  - not a strong point → higher than other architecture because the intelligence is in hosts (e.g., telephone vs. computer)
  - bad implementations or malicious users can produce considerably harm (remember fate-sharing?)
- Accountability
  - very little so far

What About the Future?

- Datagram not the best abstraction for:
  - resource management,accountability, QoS
- A new abstraction: flow?
- Routers require to maintain per-flow state (what is the main problem with this raised by Clark?)
  - state management
- Proposed Solution
  - soft-state: end-hosts responsible to maintain the state
  - Problem: increase in control-traffic to maintain state, unless efficiently piggybacked

Summary: Internet Architecture

- Packet-switched datagram network
- IP is the glue (network layer overlay)
- Hourglass architecture
  - all hosts and routers run IP
- Stateless architecture
  - no per flow state inside network



Summary: Minimalist Approach

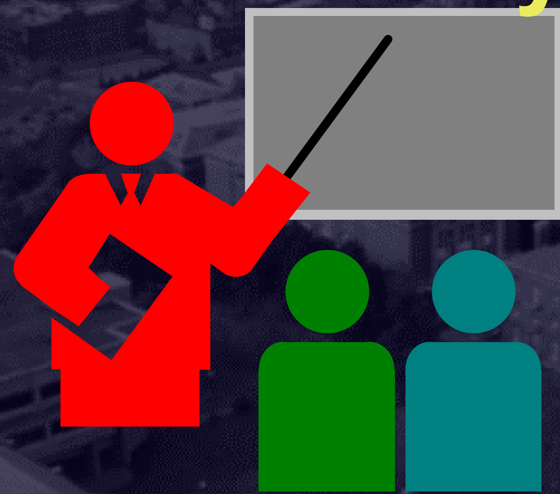
- **Dumb network**
  - IP provide minimal functionalities to support connectivity
  - addressing, forwarding, routing
- **Smart end system**
  - transport layer or application performs more sophisticated functionalities
  - flow control, error control, congestion control
- Advantages
  - accommodate heterogeneous technologies (Ethernet, modem, satellite, wireless)
  - support diverse applications (telnet, ftp, Web, X windows)
  - decentralized network administration

Connect Existing Networks

- Existing networks: ARPANET and ARPA packet radio
- Decision: packet switching
  - Existing networks already were using this technology
- Packet switching -> store and forward router architecture

- Internet: a packet switched communication network consisting of different networks connected by store-and-forward routers

Summary



- Internetworking Problem
- IP header: supports connectionless delivery, variable length pkts/headers/options, fragmentation/reassembly,
- Fragmentation/Reassembly, Path MTU discovery.
- ARP, RARP: address mapping
- Internet architectural principles