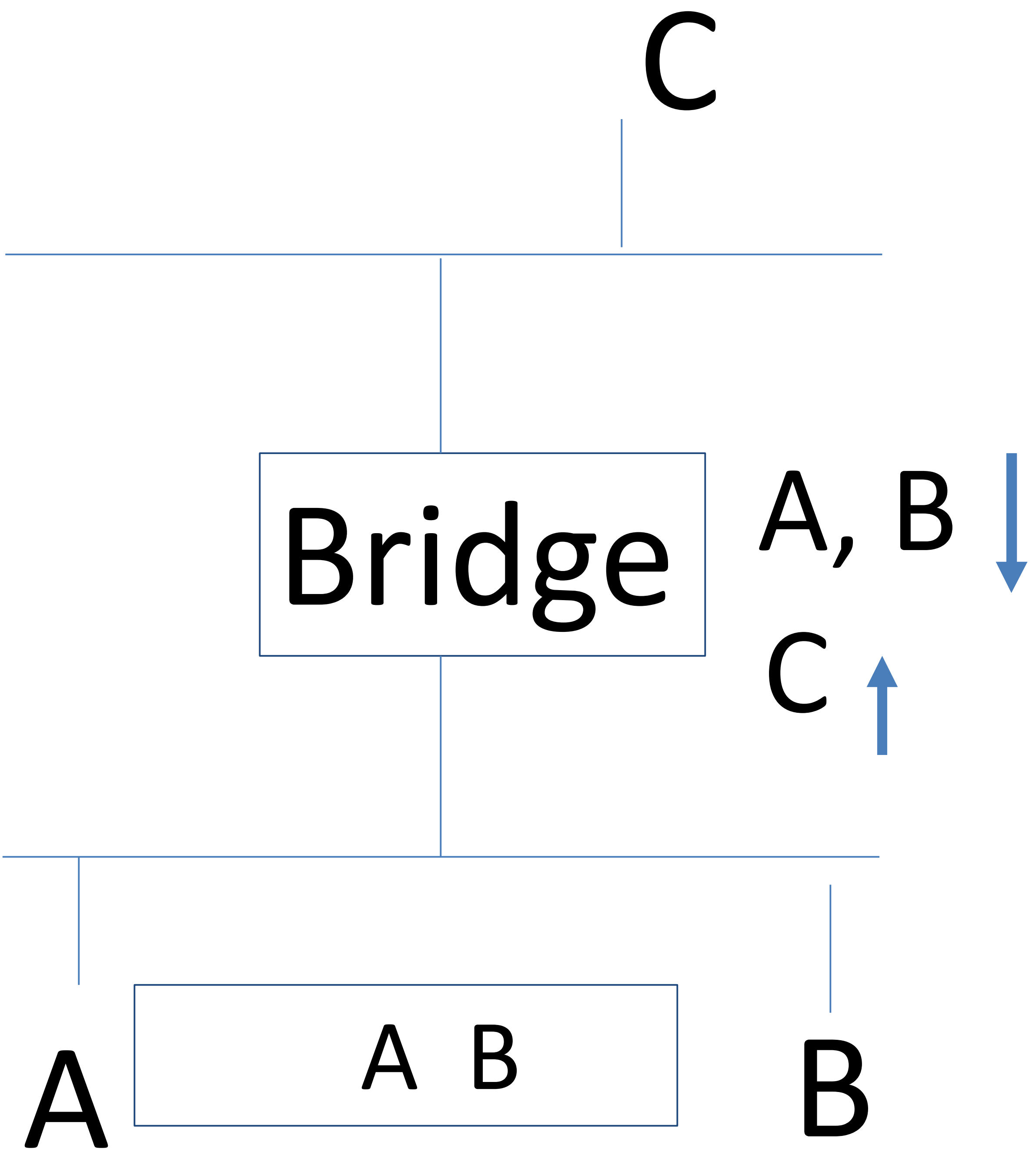
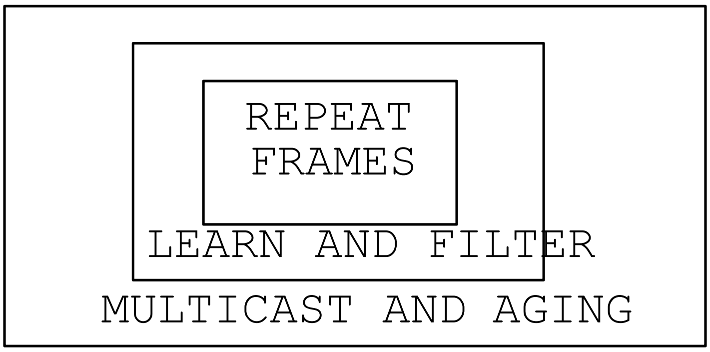
CS 118:Bridges versus Routers: To Switch or to Route

George Varghese

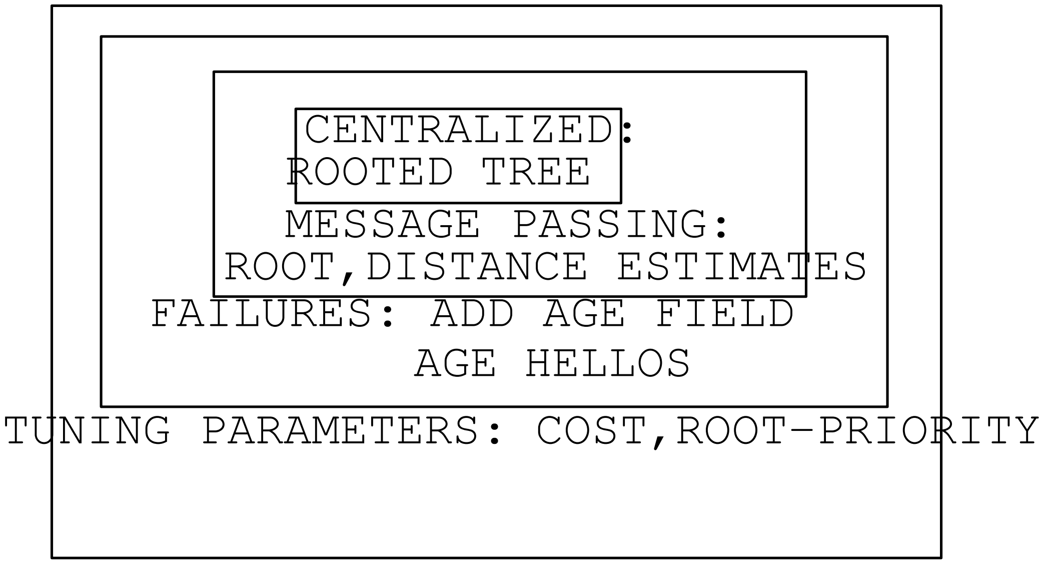


How to build database *automatically?*

BASIC BRIDGING IDEAS

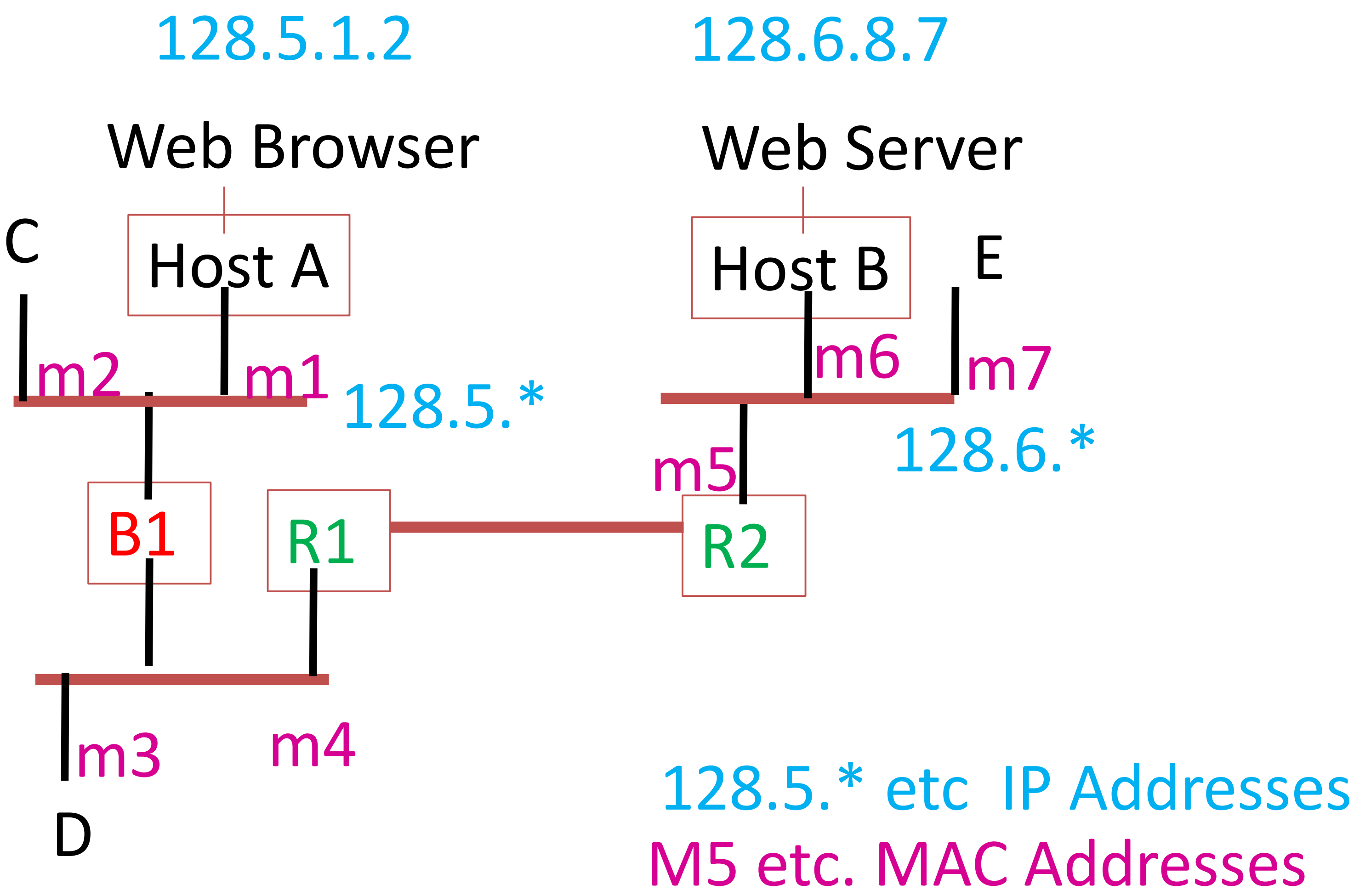


SPANNING TREE IDEAS

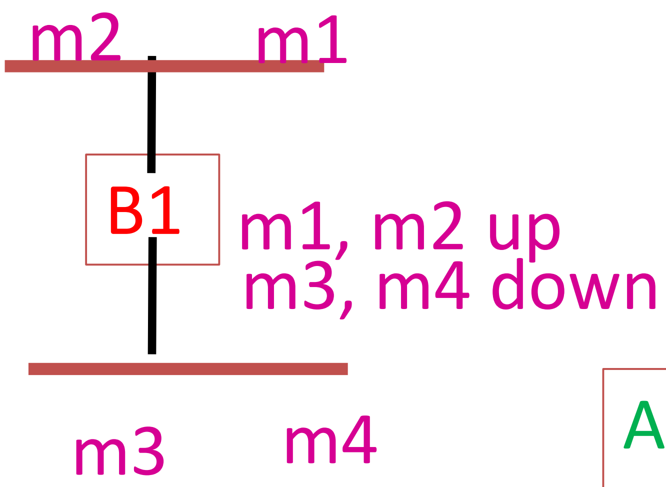


LETS START BY
REVIEWING BRIDGING

Understanding Bridge Router Topologies

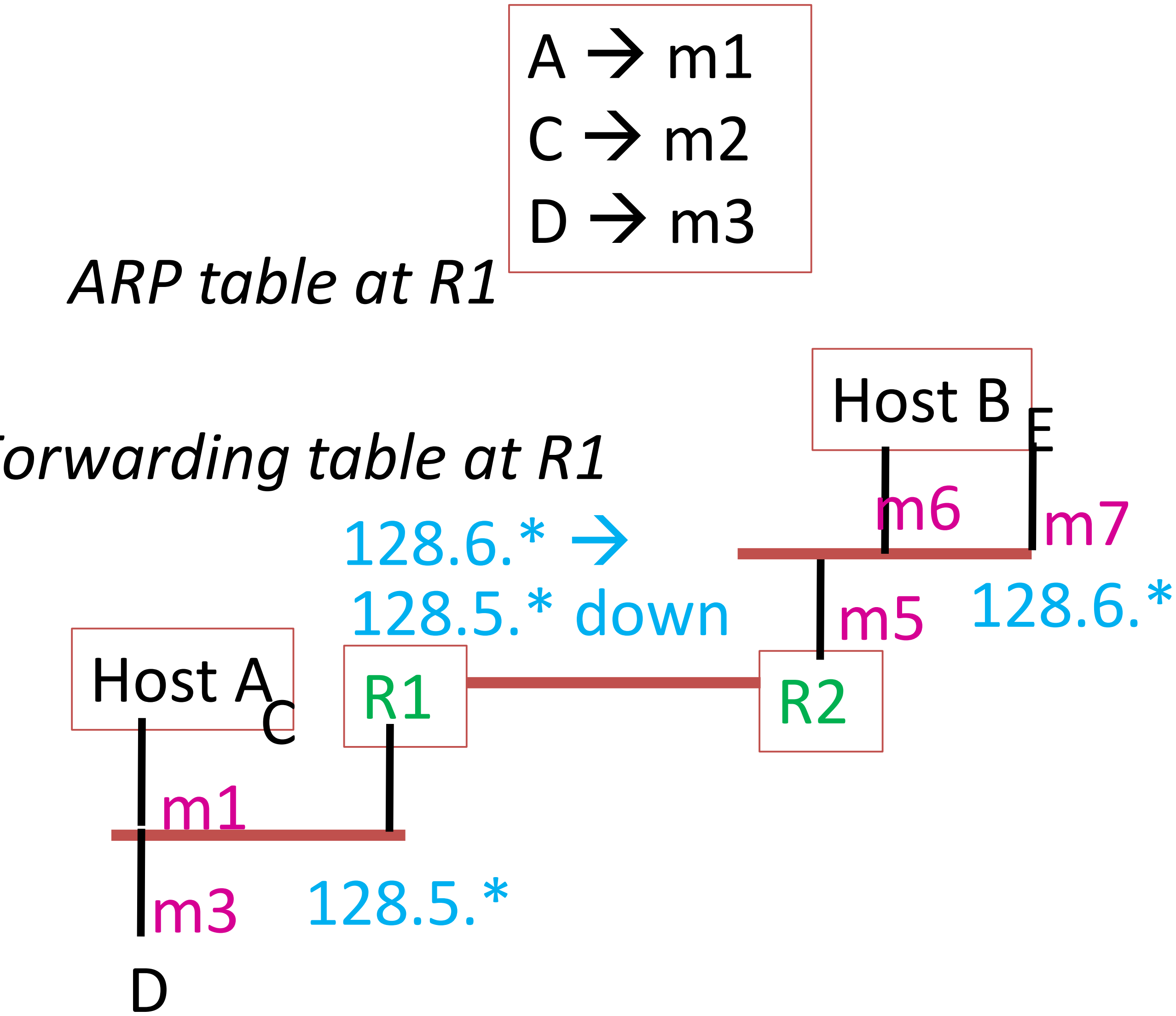


Bridge View of the World



All routers are just MAC addresses to Bridges. R1 is just m4

Router view



Naming: IP , MAC and Translation

BASIC QUESTION

- Original Data Link was single hop. But bridges make things multihop
- So why have routers? Good question.
- Main reason is that Data Link Headers were developed before bridges. So no support for Data Link relays in Data Link Header. The opposite is true for routing headers.

Data Link

D	S

Routing Header

RD	RS	Hop Count	Fragment Number	C.F.

C.F.= Congestion Feedback

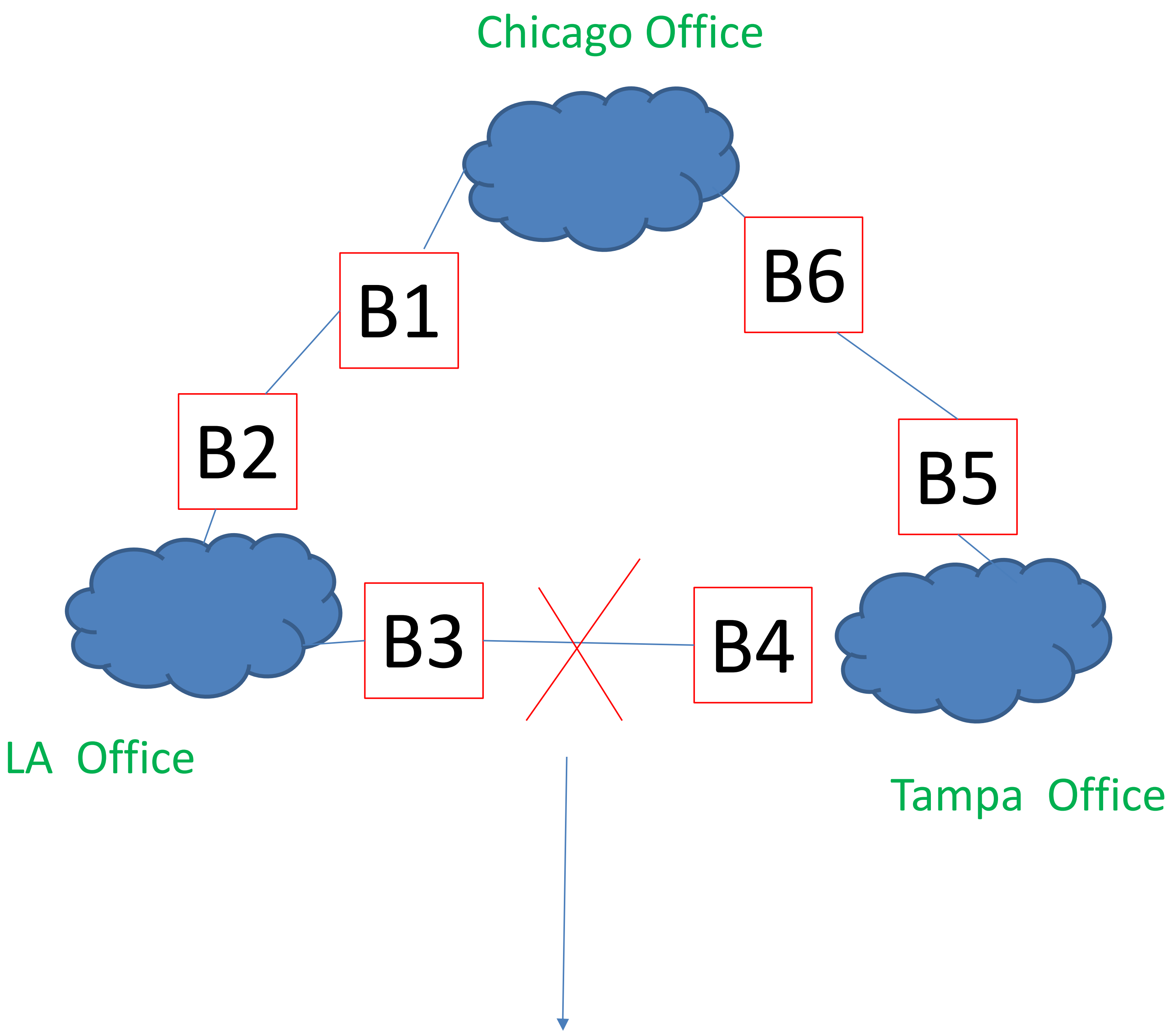
Why Bridges are Bad: Hetrogeneous Links

- Address Incompatibility (e.g., HDLC and Ethernet)
- Max Packet Size Incompatibility (e.g., FDDI and Ethernet)
- Bandwidth Incompatibility (e.g., FDDI and Ethernet)

Why Bridges are Bad: Large Networks

- 802 address is flat (like social-security numbers). Routing addresses hierarchical (like postal addresses). Bridges have to learn all addresses in an extended LAN. (more memory) Routers only learn addresses within each level of hierarchy.
- Spanning Tree inefficient. Does not route packets on shortest path between two points. Increased latency and smaller throughput. Also flooding wastes throughput.

WHY SPANNING TREES ARE BAD IN THE WIDE AREA



Spanning Tree algorithm has to break Loops. So can't use say this link

Why hierarchical addresses?

Crucial for scaling :

- If we used bridges to connect the global Internet, each bridge would store billions of MAC addresses
- So MAC addresses are flat and unique worldwide but IP addresses are hierarchical
- Scalability requires hierarchy. Like telephone numbers and postal addresses but IP is weirder as we said in the Miniflip and we will study next

Why Bridges are Good

- Generality: Bridges allow stations with different routing protocols to use the same Extended LAN.
- Cost-Performance (since they do less, bridges cost less than routers with the same performance)
- Control Traffic: Smaller amount of routing control traffic (Spanning Tree traffic is small by comparison)

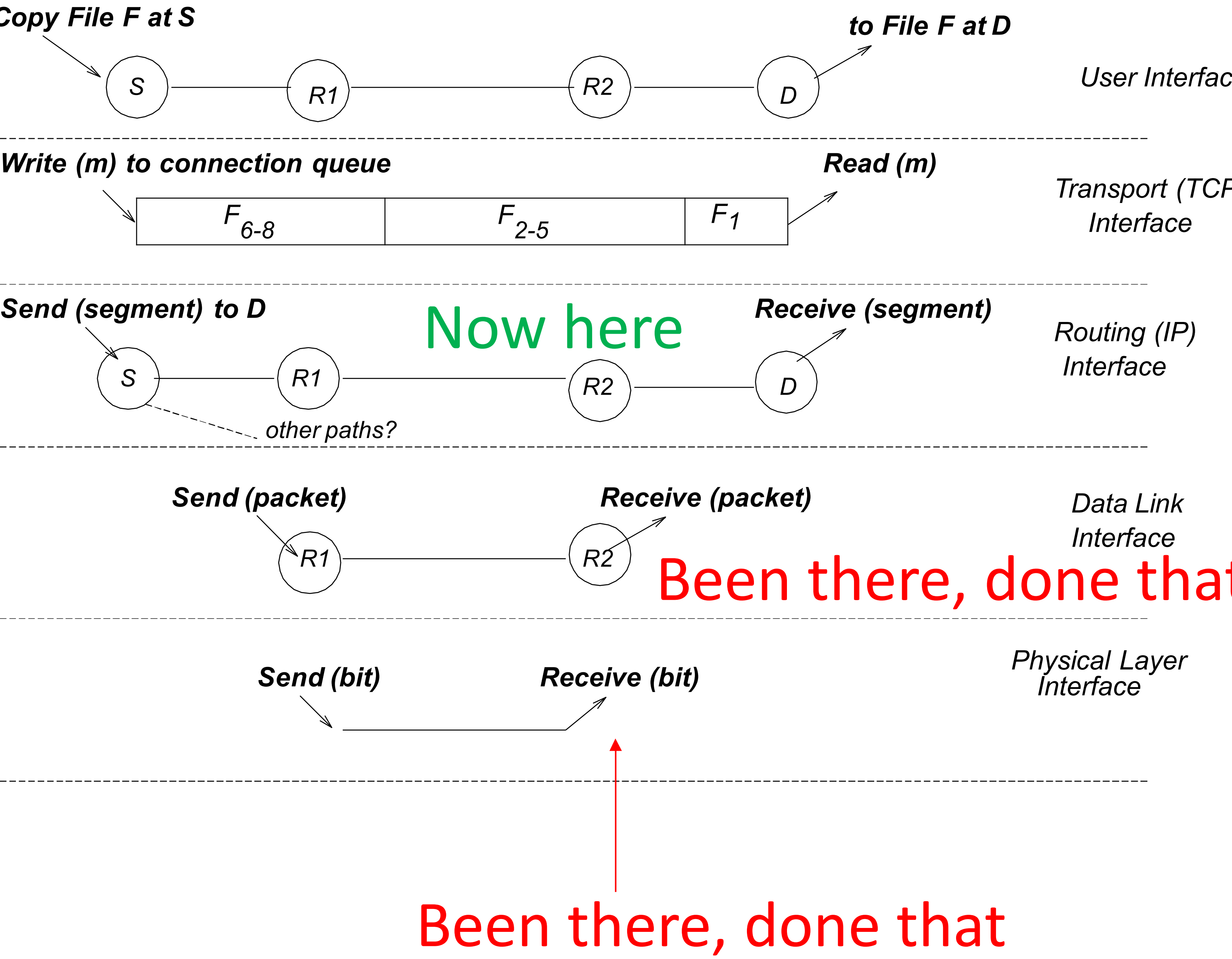
Conclusion

Each has a place:

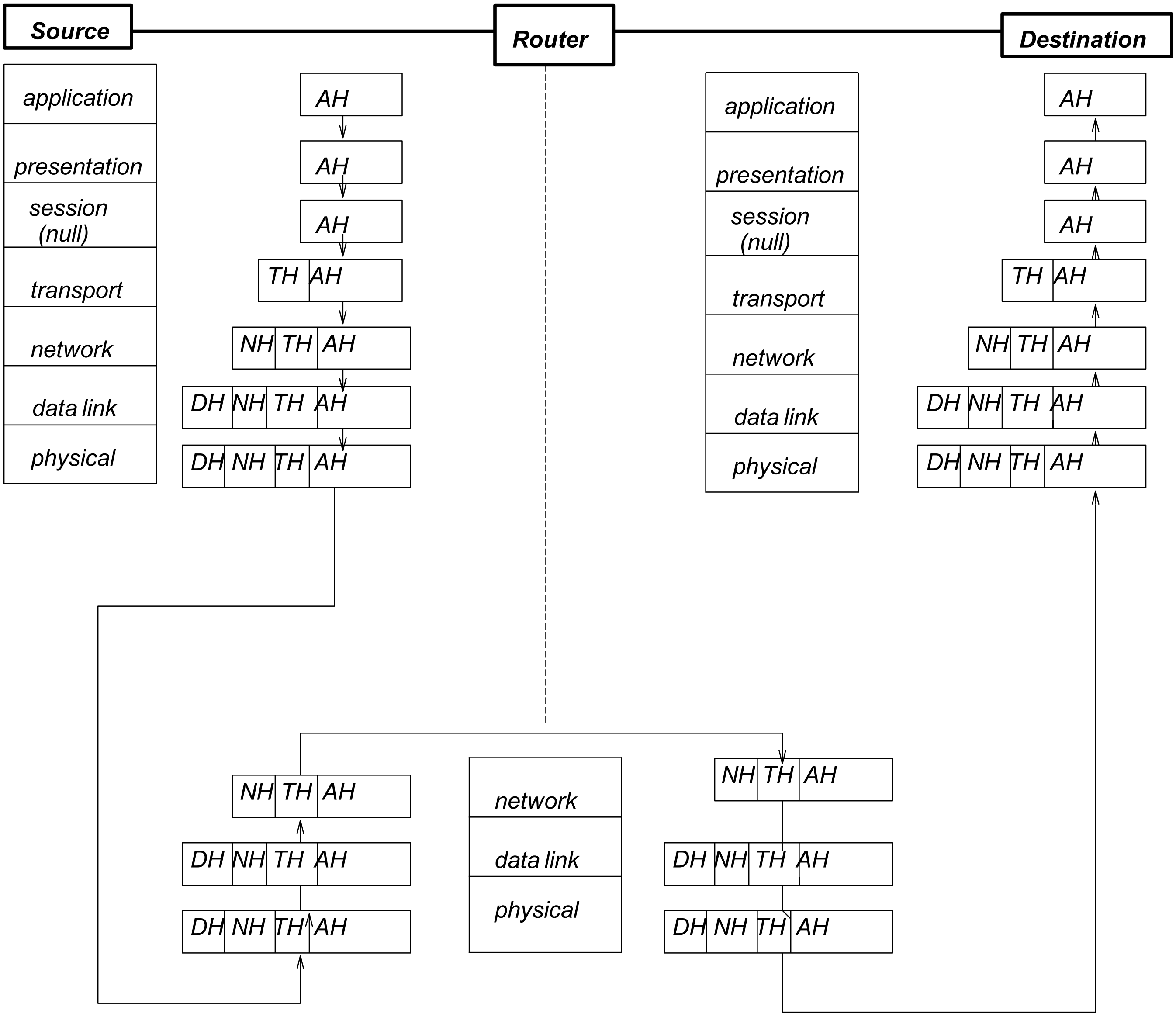
- Bridges can be used to connect a small number of compatible LANs to form an Extended LAN,
- Routers connect Extended LANs to form a Routing Network.
- Most routers today are multiport devices some ports can be bridges (switched ports) and some can be routers (router ports). Manager can configure

CS 118: IP Addressing and Overview and Project Intro

George Varghese



RECALL THE IP ABSTRACTIONS:

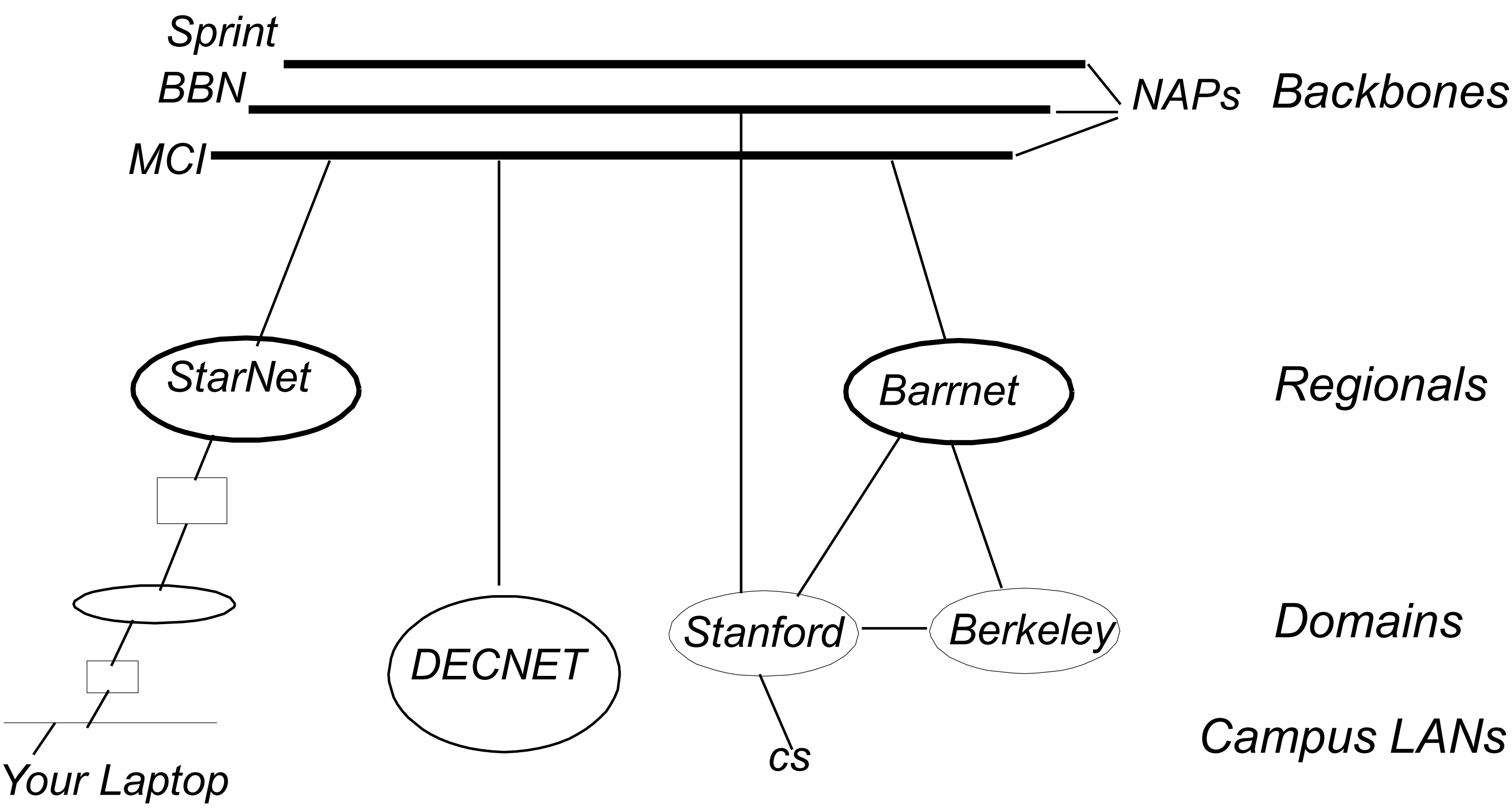


HEADERS: WATCH HOW THEY ARE ADDED AND REMOVED. STRICT LAYERING – THE DEFINITION

Life through the the Internet contains and endless set of unsorted pings. These prompts demand our attention, and present us with a perpetual onslaught of distractions and diversions . . . Our lives can easily be the story of how we respond to this endless set of prompts

From *Called*, by Mark Labberton, president of Fuller Seminary

TOPOLOGY



- Terminology: ISPs, POPs, Autonomous Systems, NAPs, Peering

Basic Internetworking in IP

- *Goal from start* Unlike DECNET, SNA etc. starts with a hierarchy of physical networks with network specific routing that IP does not care about, to create an Internetwork of physical networks
- *IP 's role* to route packets to the right physical network based on the network number. Offers a so-called datagram service with possible fragmentation and reassembly to deal with different maximum packet sizes
- *Error messages* companion protocol called ICMP for error messages (header checksum failed, maximum time exceeded, redirect etc.).

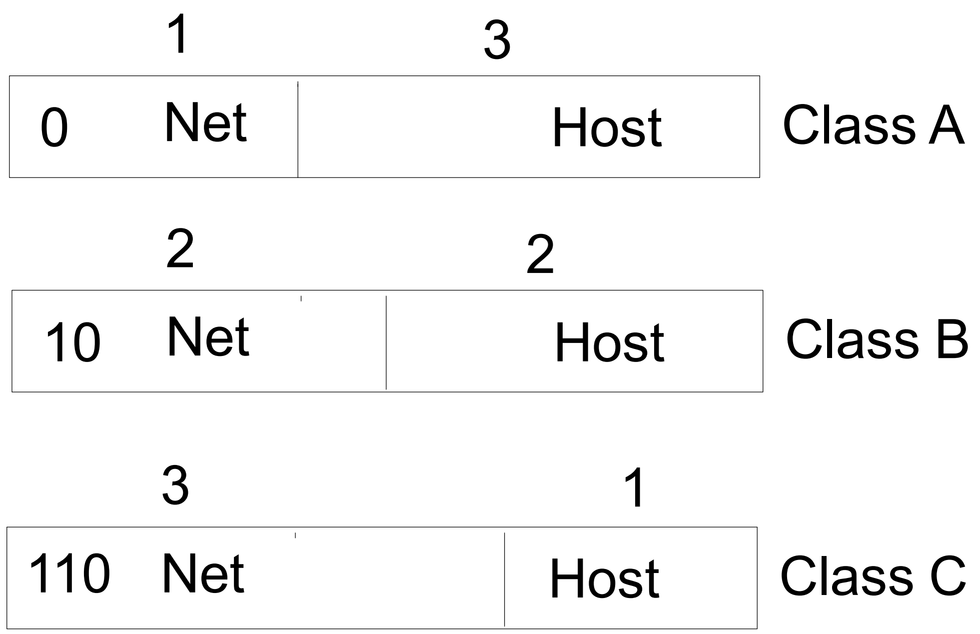
IP Evolution

- *ARPANET* started by linking government and university sites (including UCLA) in the 1970s
- *NSFNet* in 1983 ARPANET splits up into MILNET and ARPANET. In 1984 NSF establishes NSFNET to be backbone. Campuses attached to backbone via regional networks (NYSERNET etc.) Strict hierarchy breaks down because of direct connections between providers
- *Multiple Providers by lat 1980s* Internet becomes worldwide. From a research network to production quality. Multiple autonomous providers that need to work toi=gether

Names and Address

- *Names* when you send to a domain name like cs.Berkeley.edu , a resolver is your host translates the name to a 32-bit IP address. All messages carry IP destinations addresses
- *Domain Name Service* the translation is done using the so-called Domain Name Service or DNS which we will study later

ORIGINAL IP ADDRESSES: CLASSFUL



- Original Model: Small number of large networks (class A), moderate number of campus networks, large number of LANs
- Idea: Hierarchical address with a moverable boundary

Old IP Forwarding

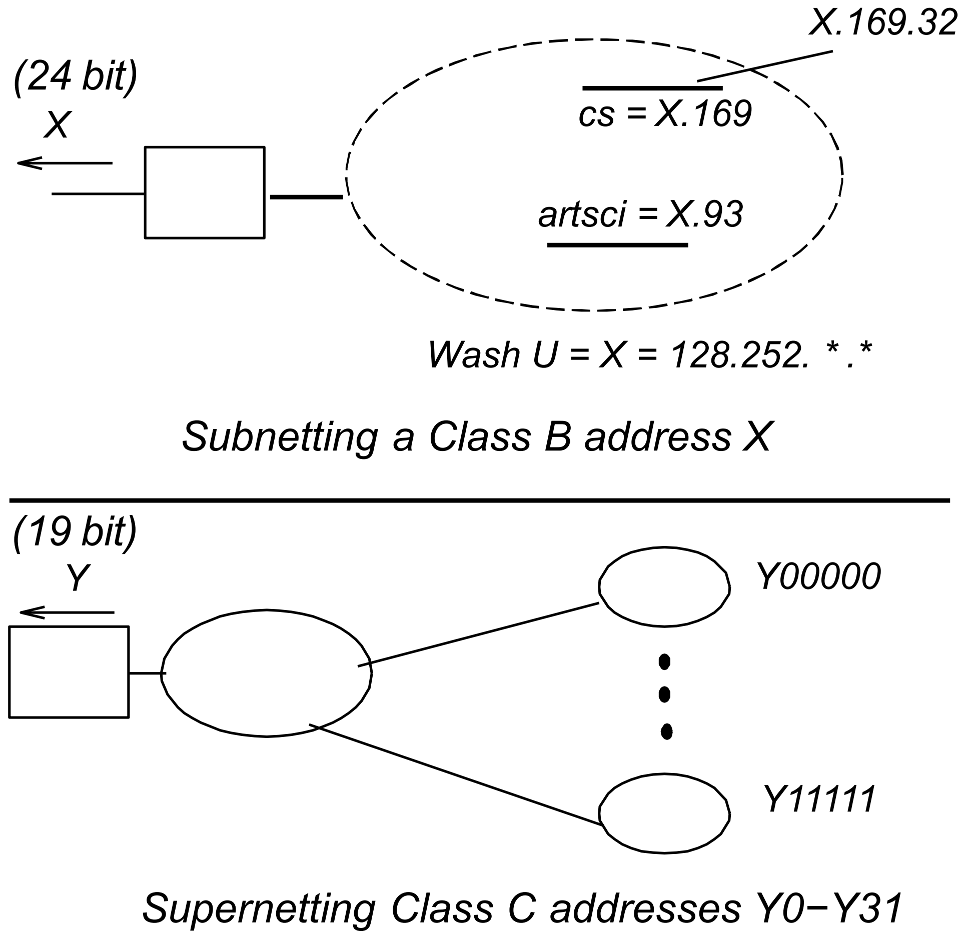
- *Find Destination* extract Network Number of destination address by parsing and checking for class A, class B etc,
- *Final hop reached?* If (Network Number of Dest = Network number of one of this router's local interfaces) deliver packet. Map to local address using ARP or some such network specific protocol
- *Lookup Router Table* Lookup Network Number in the corresponding routing table, If it exists, deliver packet to corresponding NextHop.
- *Lookup Router Table:* If no route entry exists, send yo default router. (This looks silly but is a great way to avoid keeping lots of table entries in stub organizations like UCLA).

Challenge-Response

One level of hierarchy good but IP quickly ran into two scaling challenges:

- *Inefficient address usage:* any organization that needed more than 255 addresses asked for a class B address (64,000) and they quickly ran out
- *Routing Table Growth:* the response to no more class B addresses was to assign multiple Class C addresses. But now every backbone router needed to know more addresses, more routing traffic, search times etc.
- *Response* changed IP forwarding to longest matching prefix. Why?

SUBNETTING AND SUPERNETTING



- Supernetting: Done recursively, leads to backbone routers only having hundreds of thousands of prefixes of lengths 8-32
- Temporary Measures: Often today new organizations are give 1 IP address and use NAT. Need the move to IPV6 (128 bits)

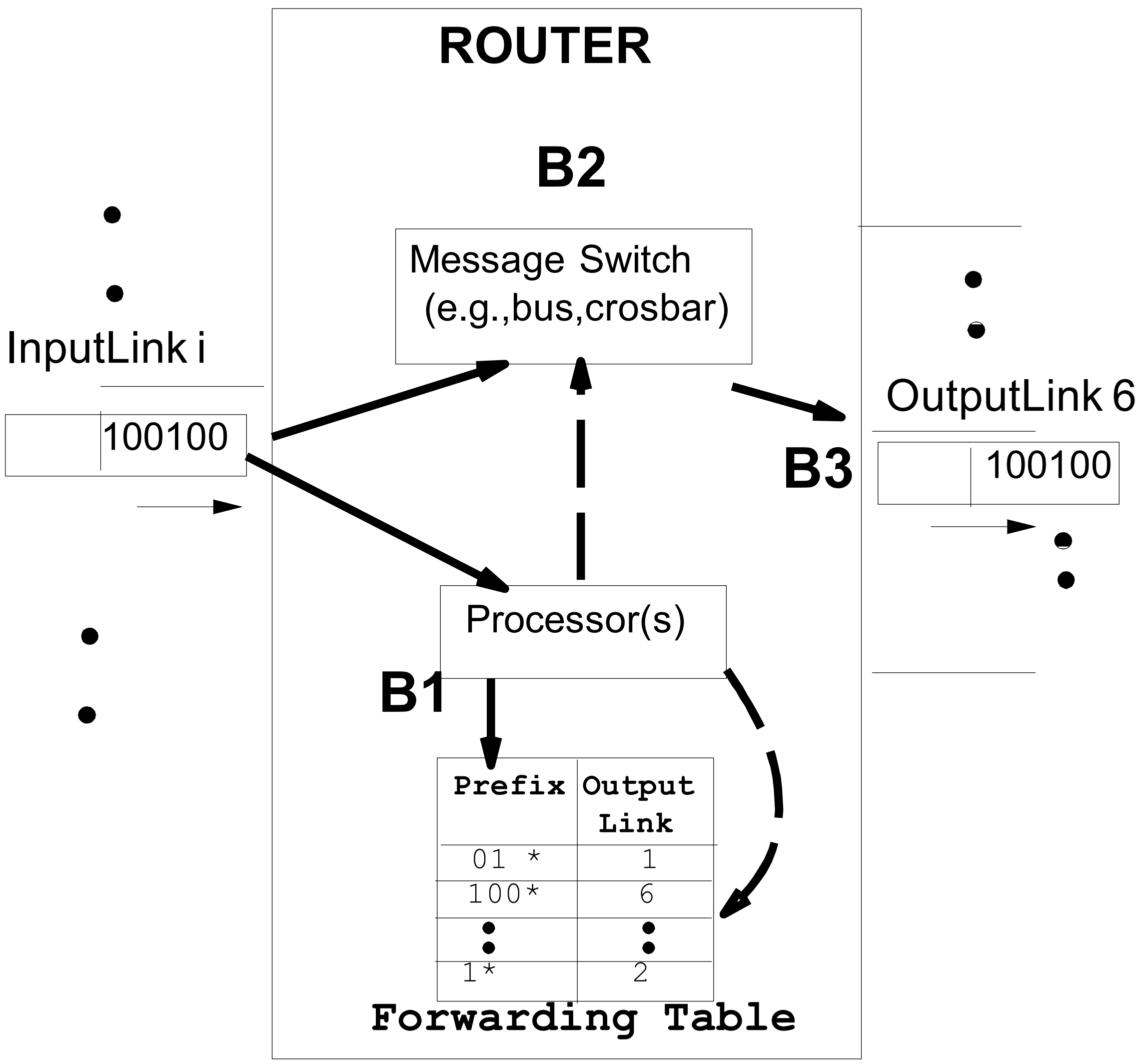
IP evolved to meet challenges

- *Challenge 1* Interconnecting diverse networks → Net Numbers (Class A, 1 byte)
- *Challenge 2:* Ethernets led to an explosion of networks → Hack to add Class B, Class C
- *Challenge 3* Class B addresses ran out → Give consecutive class C and use Longest Prefix Match
- *Challenge 4*: Even Class C's started running out → NAT and concurrent move to IP v6

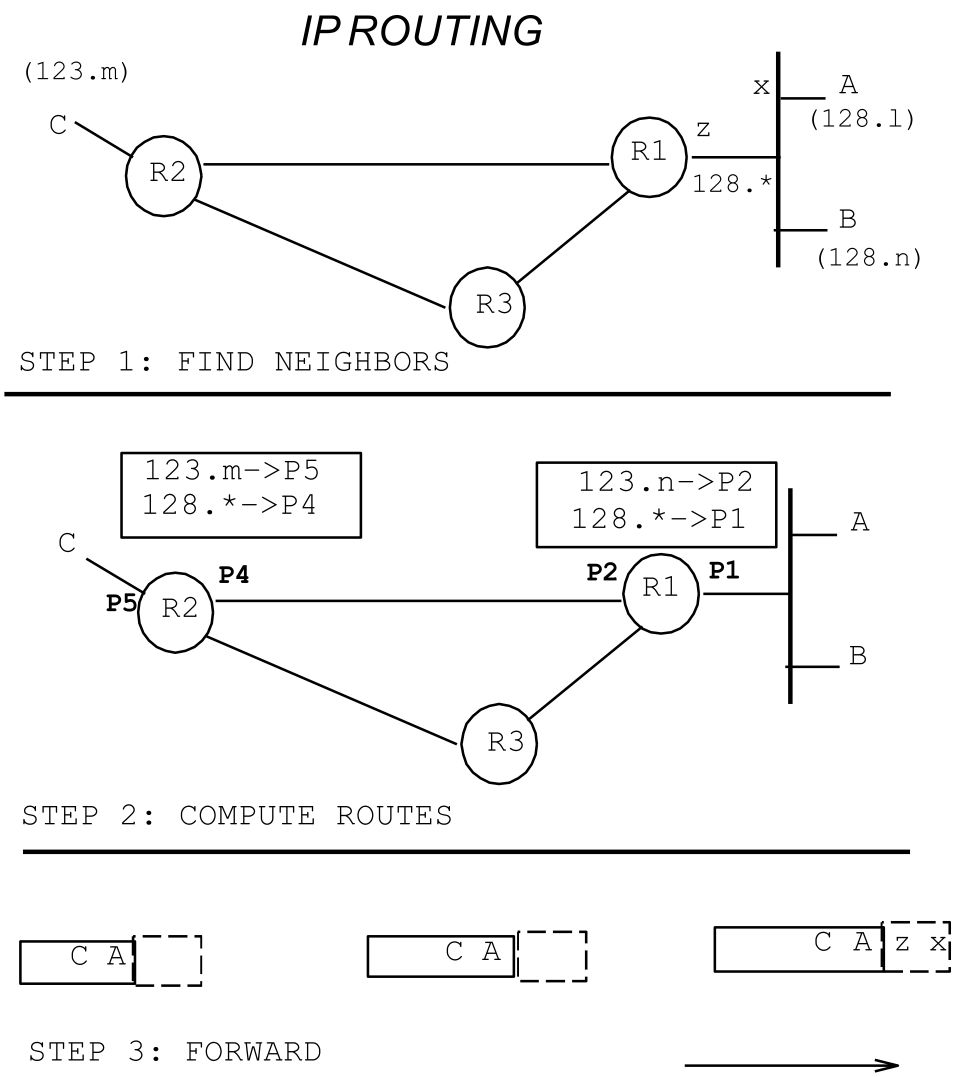
New IP Forwarding

- *Lookup* find longest matching prefix P of destination IP address in packet in forwarding table
- *Default or Local?* If P is nil forward on default route. If the next hop associated with P is a local interfaces, deliver packet. Map to local address using ARP or some such network specific protocol
- *Send on its way* if not, send packet to NextHop route associated with P
- Backbone routers in default-free zone have to have many hundred thousand prefixes to reach everyone. Enterprise routers have 1000s because of heavy use of default routes.

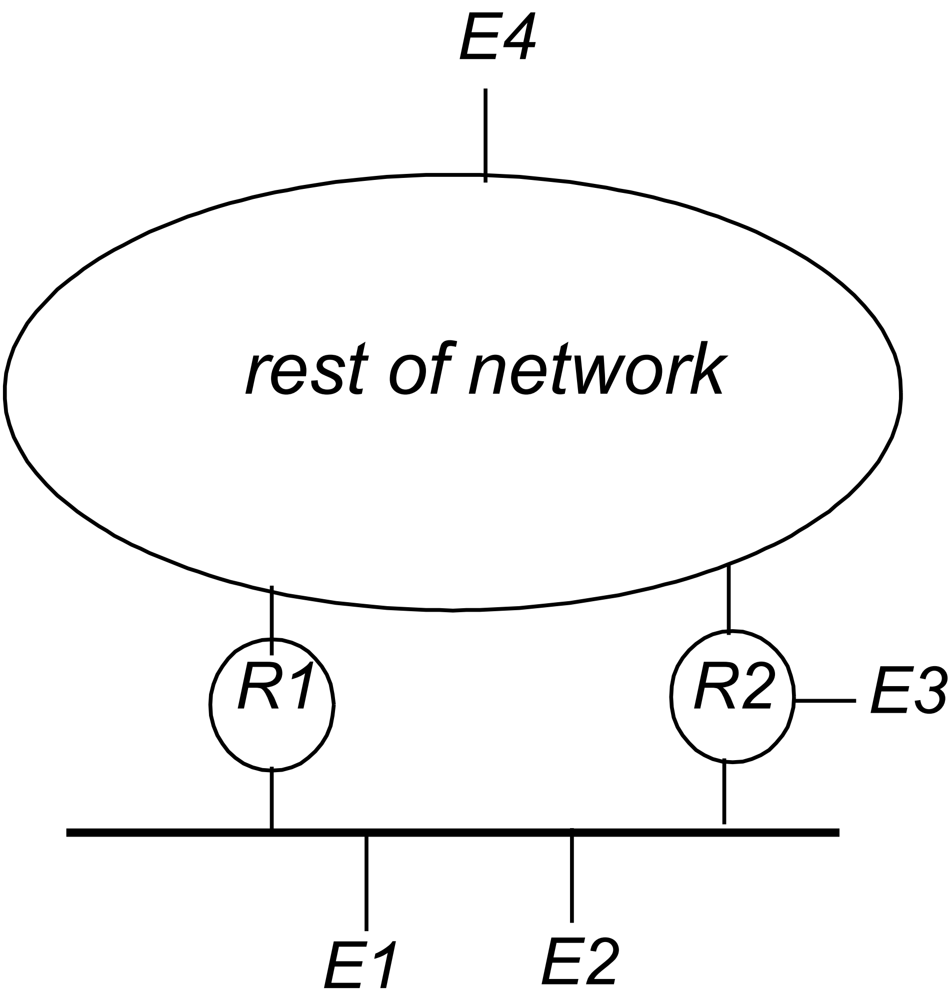
ROUTER MODEL



FORWARDING AND ROUTING



FOUR PROBLEMS ENDNODES MUST SOLVE



- P1: Routers need Data Link Addresses of endnodes
- P2: Endnodes need DL address of 1 router
- P3: E1 and E2 should be able to communicate without a router
- P4: E1 to E3 traffic should go through R2

IP Solutions to End-node Problems

- *P1:* ARP for MAC address of destination
- *P2:* a service called called DHCP gives you the IP address of one router (autoconfiguration)
- *P3:* two endnodes know they are on same subnet by comparing masks. Then ARP
- *P4:* send to router and router sends redirect if packet returns on interface it entered router. (Ignore this code in project),

Packet Format

Ethernet Frame

Destination address	Source address	Type	Payload

Type: ARP or IPv4

ARP packet

Opcode	Src MAC address	Src IP address	Dst MAC address	Dst IP address	Payload

Opcode: ARP request or ARP reply

IPv4 packet

Version	…	TTL	Checksum	Src IP address	Dst IP address	Payload

ICMP packet

Type	Code	Checksum	Identifier	Seq Num	Payload

IPv4 header also contains header length, total length, ID, flags, fragment offset, and protocol fields.

Forwarding Pseudocode: ARP

- 1. Find input network interface: findIfaceByName. Drop packet if interface is unknown

2. Read ethernet header and check the eth_type field. Ignore all but ARP and IPv4 types

3. If eth_type is ARP:

   a. If ARP Request packet:

   - Prepare and send ARP response packet

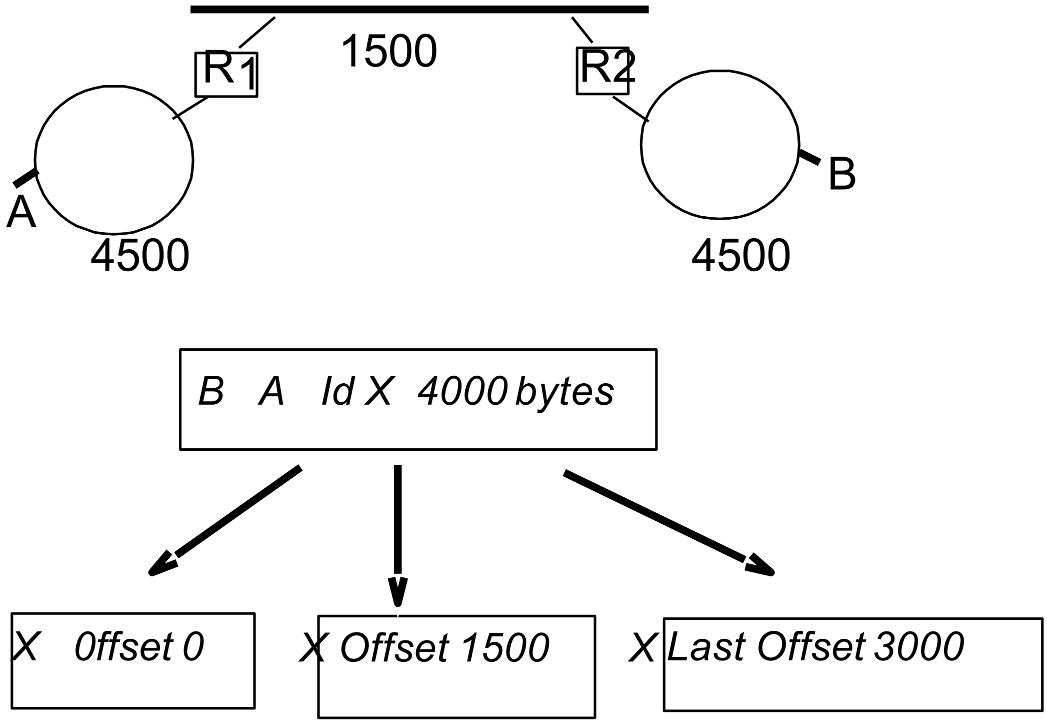
   b. If ARP Response packet:

   - record IP-MAC mapping information in ARP cache
   - send out all enqueued packets for ARP entry

Forwarding Code: IPv4

- 4. If eth_type is IPv4:
  - verify checksum, length, discard invalid packets
  - if packet is to router
    → If ICMP packet then handle Ping
- 5. Use the Longest Prefix Match algorithm to find a next-hop IP address in the routing table
- 6. Lookup ARP cache for MAC address mapped to the next hop destination IP address
  -- If valid entry found: forward packet
  - Else: queue received packet and send ARP request to discover the IP-MAC mapping.

ASIDE ON FRAGMENTATION AND REASSEMBLY



- Path MTU instead: Modern end-nodes find the right size (1500) instead of asking the routers to fragment. Fields ignored by most routers and in your project.