CSC4200/5200 – COMPUTER NETWORKING

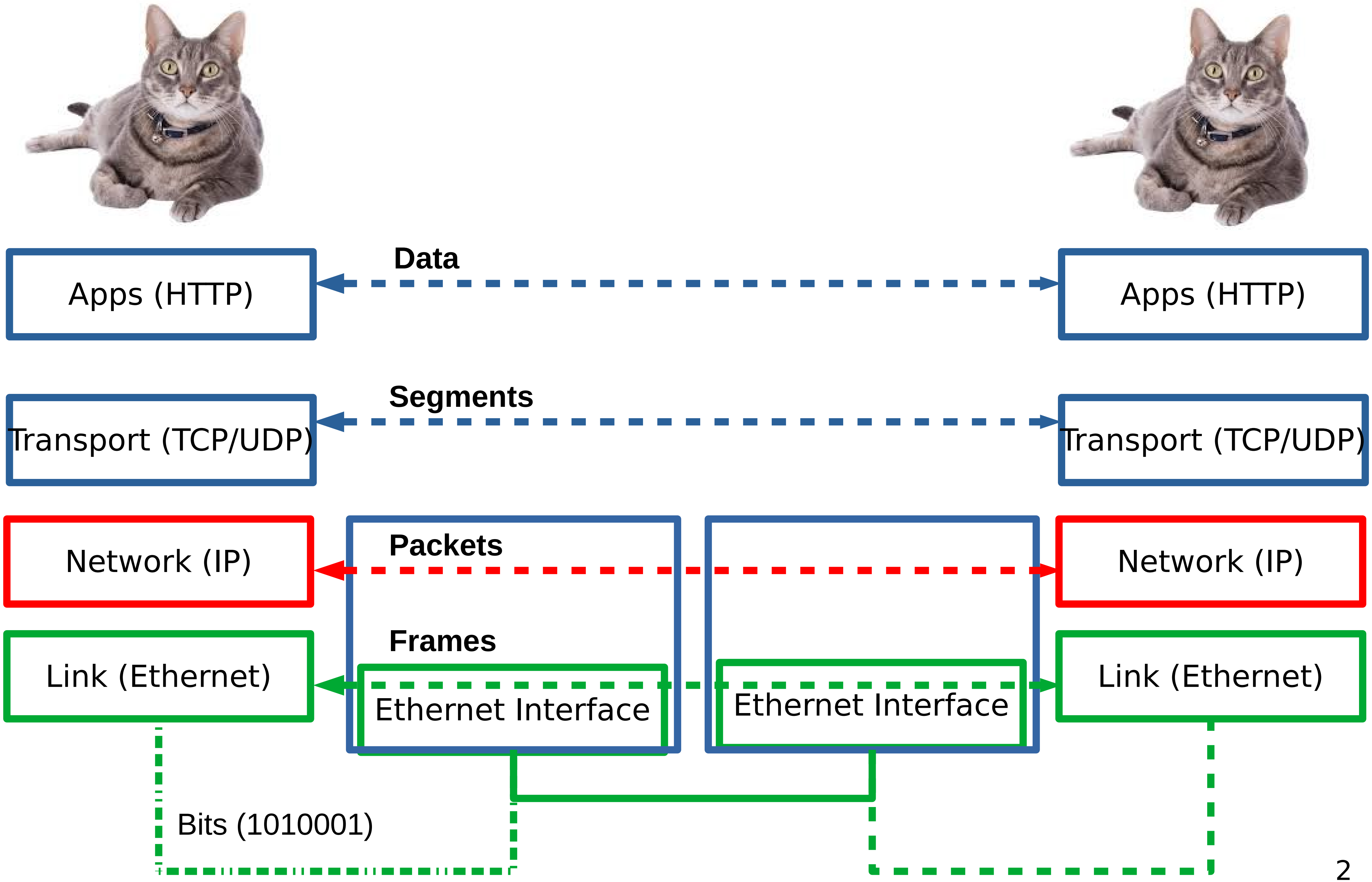
Instructor: Susmit Shannigrahi

INTERNET PROTOCOL (IP)

sshannigrahi@tntech.edu

GTA: dereddick42@students.tntech.edu

Tennessee TECH


Data
Apps (HTTP)
Apps (HTTP)
Segments
Transport (TCP/UDP)
Transport (TCP/UDP)
Packets
Network (IP)
Network (IP)
Frames
Link (Ethernet)
Ethernet Interface
Ethernet Interface
Link (Ethernet)
Bits (1010001)

CSC4200/5200 – COMPUTER NETWORKING

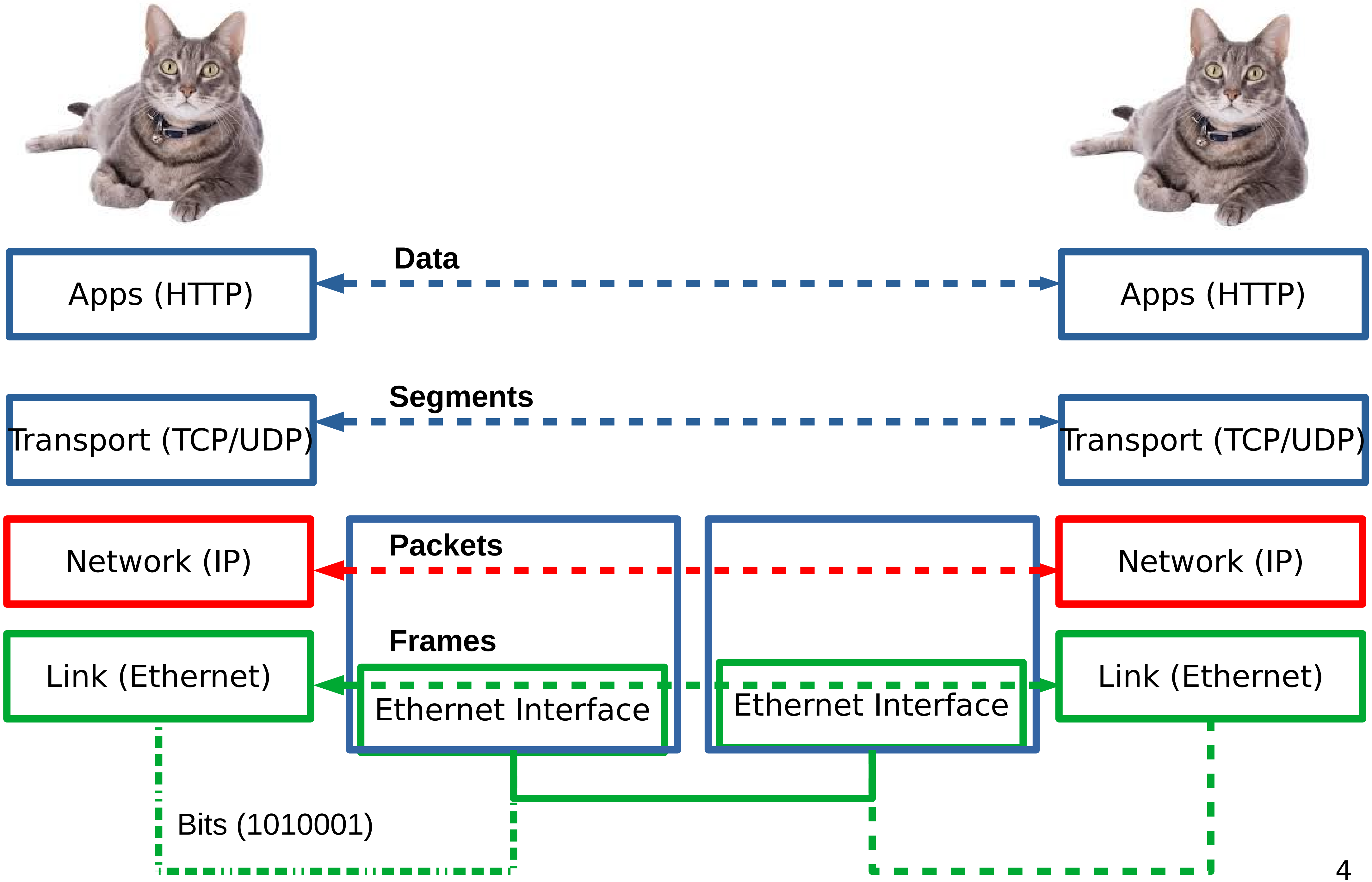
Instructor: Susmit Shannigrahi

INTERNETWORKING

sshannigrahi@tntech.edu

GTA: dereddick42@students.tntech.edu

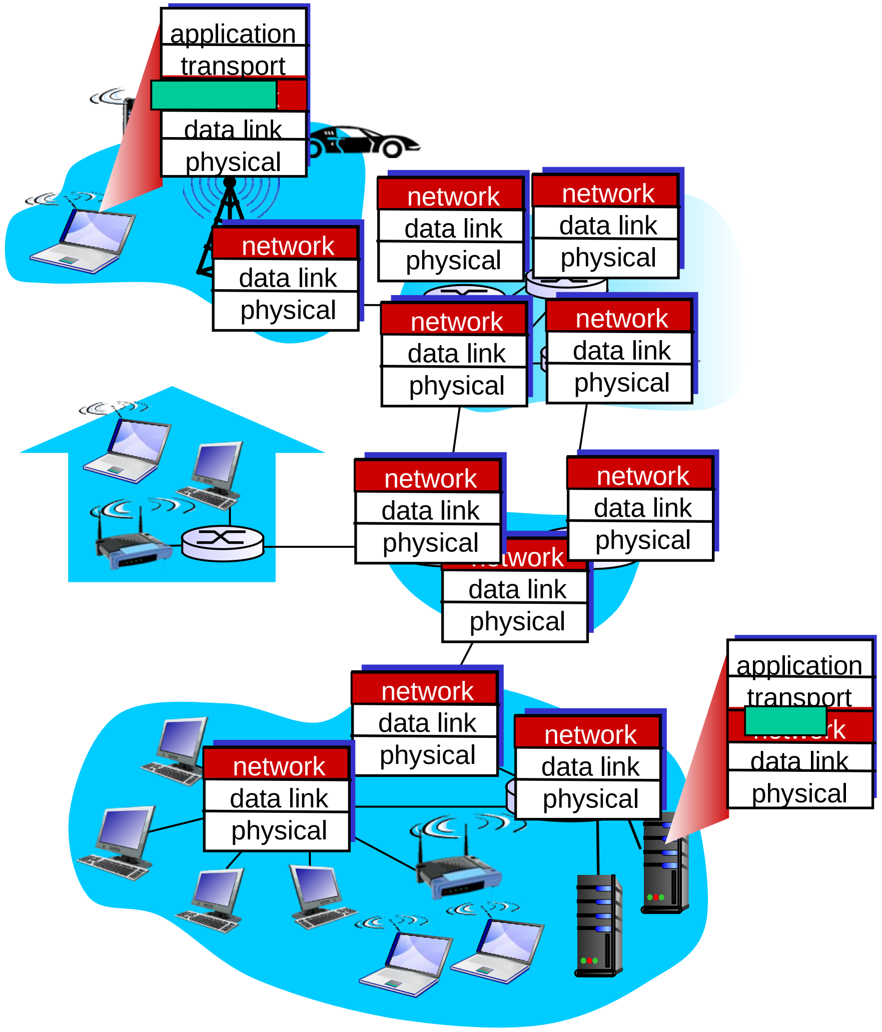
Tennessee TECH


Data
Apps (HTTP)
Apps (HTTP)
Segments
Transport (TCP/UDP)
Transport (TCP/UDP)
Packets
Network (IP)
Network (IP)
Frames
Link (Ethernet)
Ethernet Interface
Ethernet Interface
Link (Ethernet)
Bits (1010001)

So far…

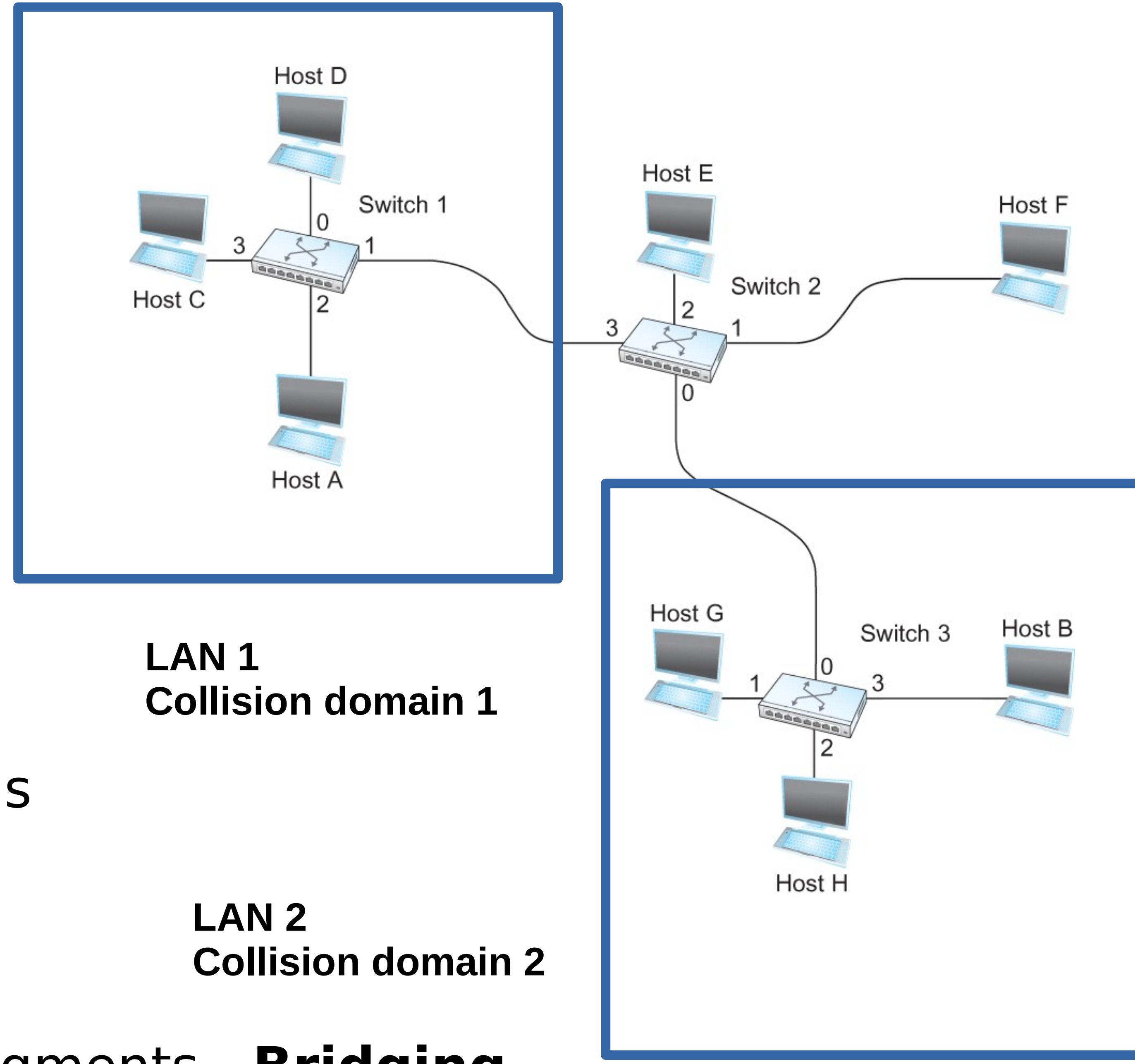
- we saw how to build a local network
- How do we interconnect different types of networks?

Why another layer?



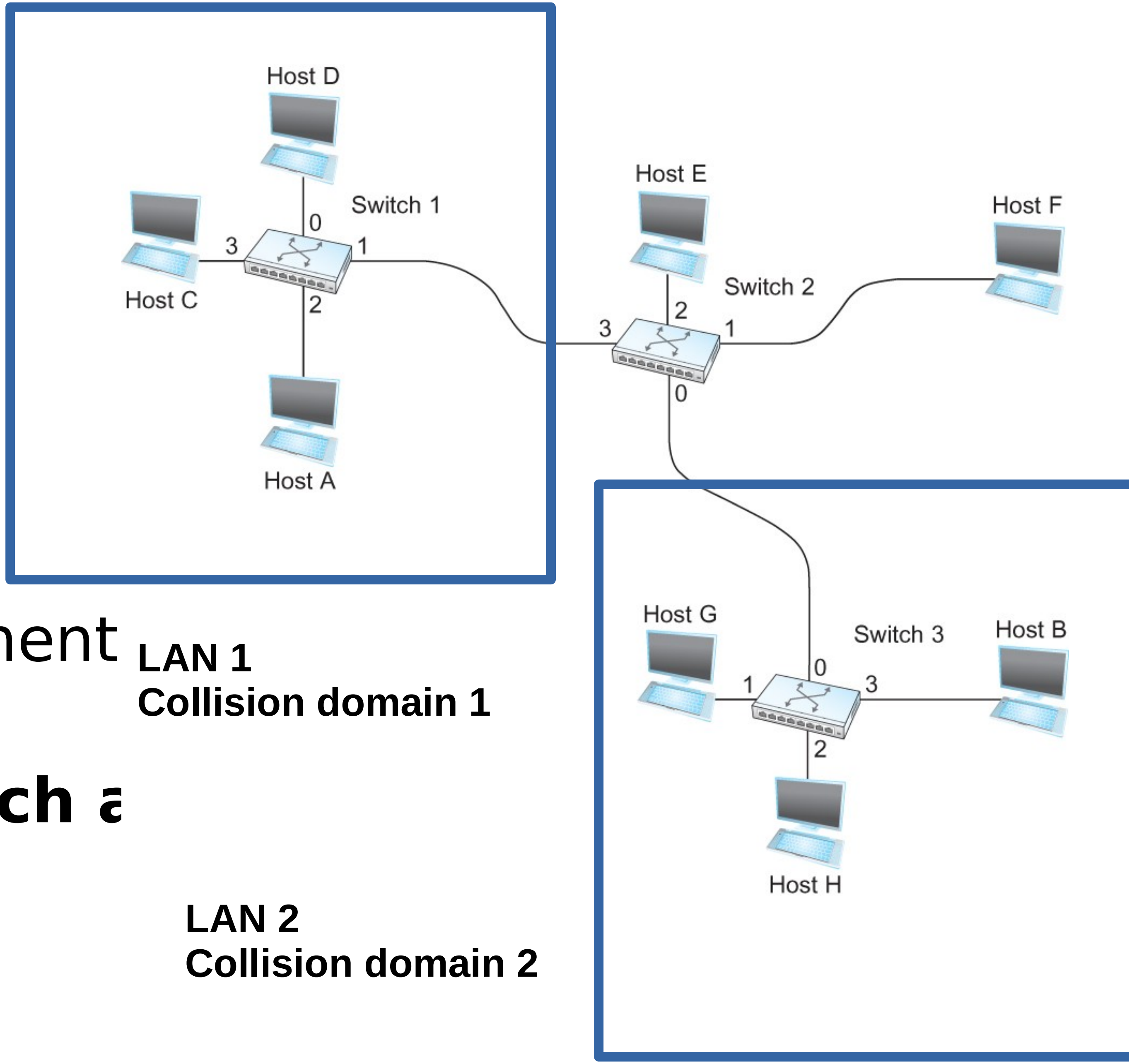
Switching

- Switch
  - A mechanism to interconnect links to form a large network
  - Forward **frames**
  - Separate the collision domains
  - Filter packets between LANs
  - Connects two or more LAN segments - **Bridging**



Switches are Self learning!

- No configuration needed
- Send frames to needed segment
- **How do they construct such a table?**



Switches are self learning!

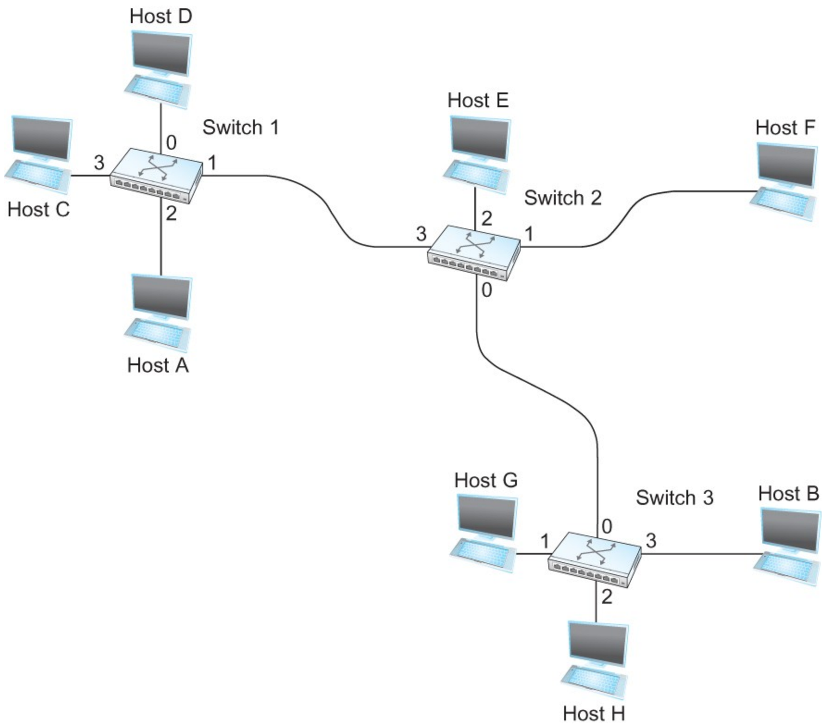
- Inspect the source MAC address
  - **What is a mac address?**

- Associate mac address and incoming interface

- Store this association for later use, (for some time)
  - aging-timer

Switching Table

- To decide how to forward a packet, a switch consults a *forwarding table*



Destination, Port

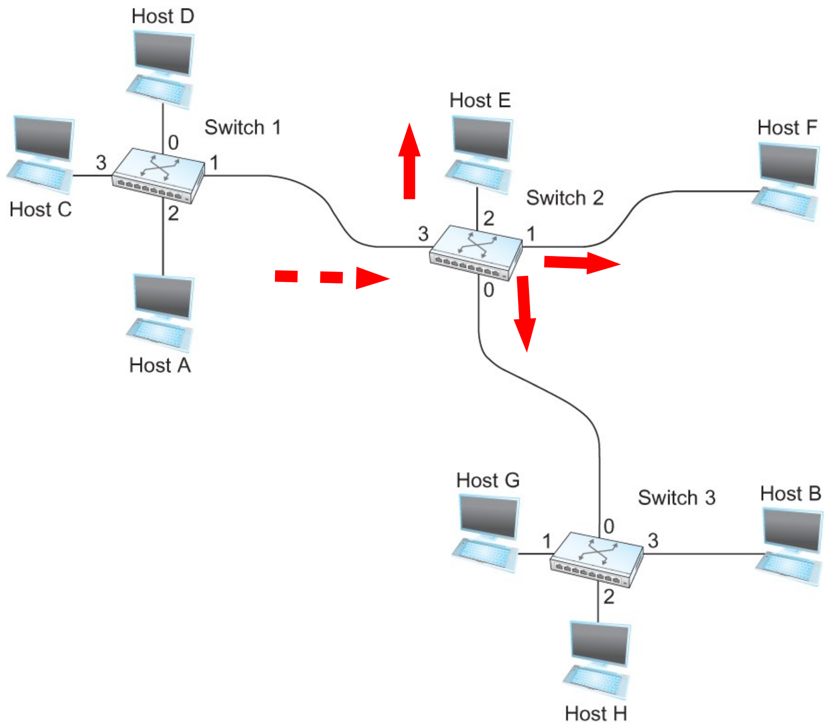
--

A 3
B 0
C 3
D 3
E 2
F 1
G 0
H 0

Forwarding Table for Switch 2

Switching Table

- Unknown destination → send out on all Interfaces **(flooding)**
  - **Skip the incoming interface**



Destination, Port

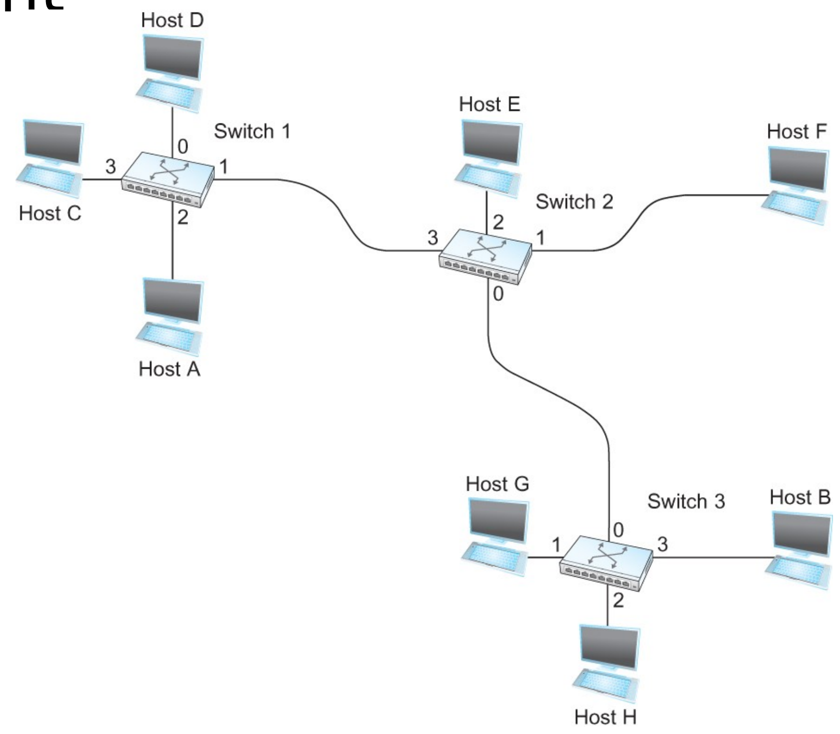
--

A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Forwarding Table for Switch 2

Switching Table   Algorithm

- Create the table first!
  - **For each packet**
    - If destination address in arriving segment
      - Drop
    - If destination is in another segment
      - Forward
    - If destination unknown
      - Flood!



Switching Table Algorithm

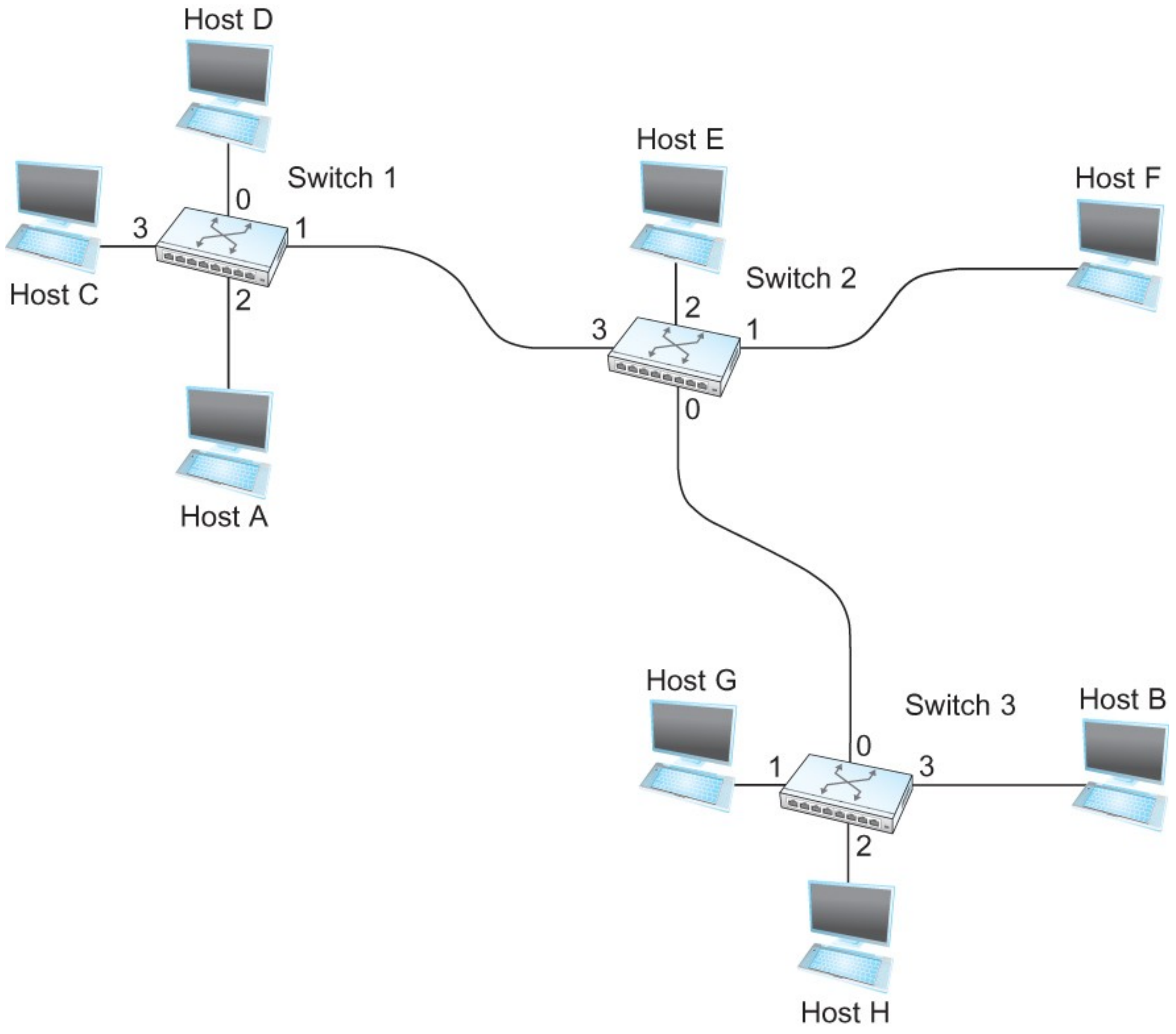
- **Send frame from C to F**

- Switch 1 →
  - Notes C is on Interface 3
  - Floods

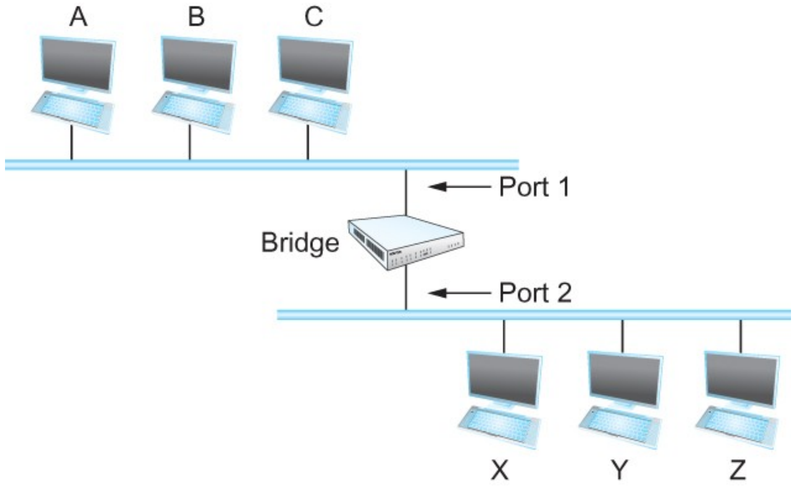
- Switch 2 →
  - Notes C is on Interface 3
  - Floods

- Host F replies
  - Switch 2 notes F is on Interface 1
  - Sends back over Interface 3

- Switch 1 notes F is on Interface 1
  - Sends back over Interface 3
  - Host c receives frame



Bridges

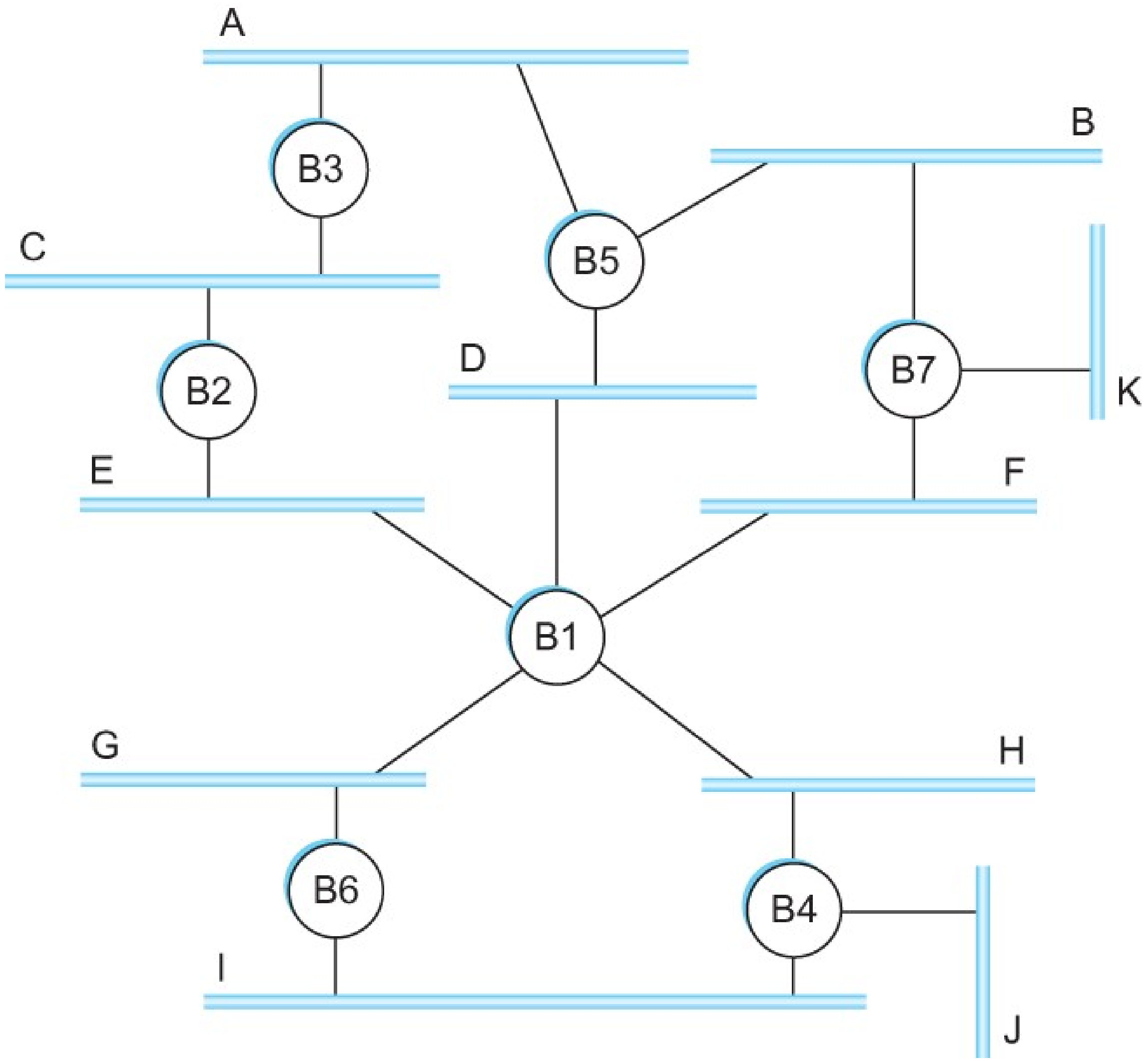


- Bridges and LAN Switches
  - Class of switches that is used to forward packets between shared-media LANs such as Ethernets
  - Known as LAN switches
  - Referred to as Bridges

- Suppose you have a pair of Ethernets that you want to interconnect
  - One approach is put a repeater in between them, physical limitations

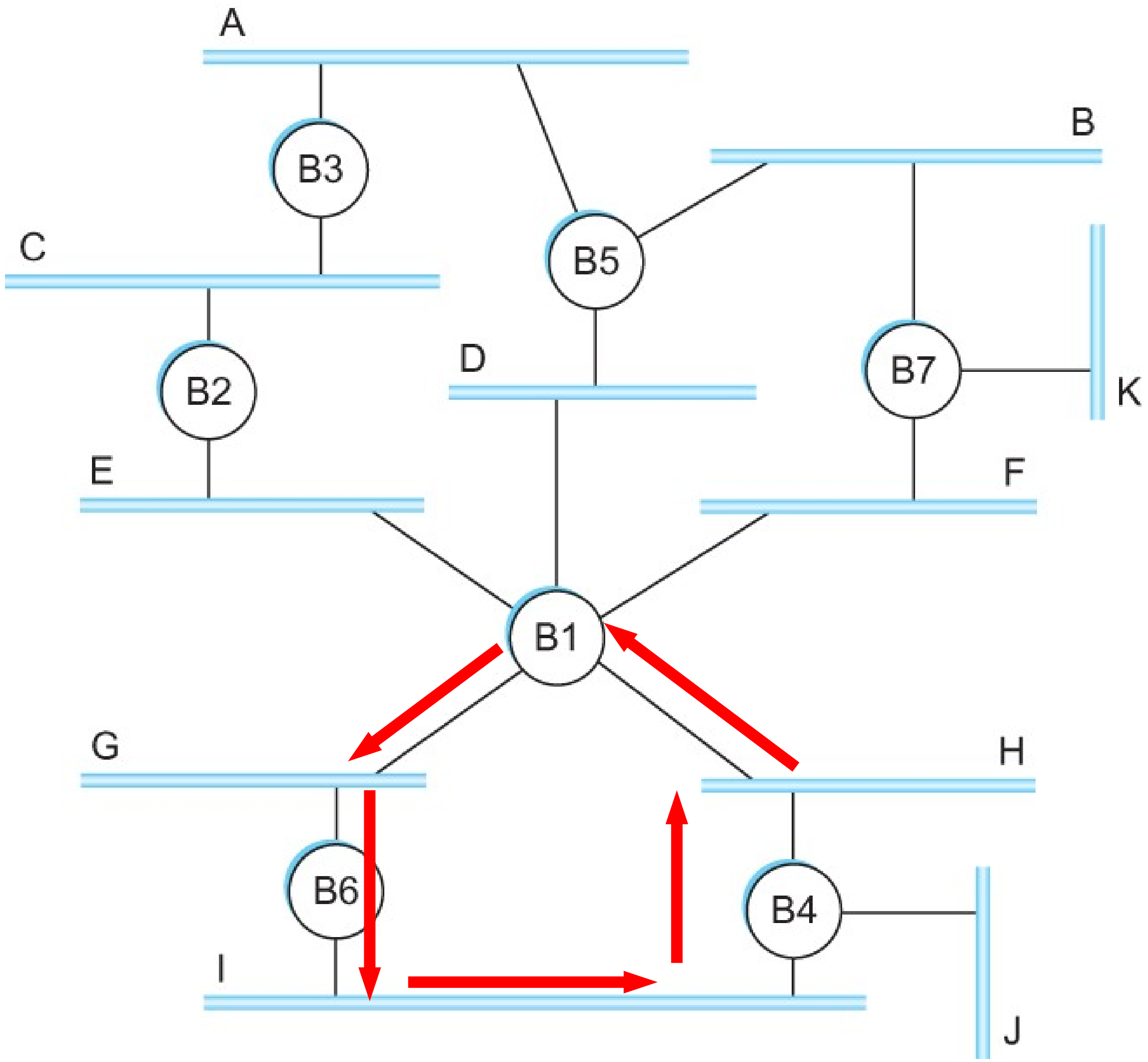
- An alternative would be to put a node between the two Ethernets and have the node forward frames from one Ethernet to the other
  - This node is called a **Bridge**
  - A collection of LANs connected by one or more bridges is usually said to form an **Extended LAN**

Flooding over bridges causes forwarding loops



Spot the loop
Why?

Loop

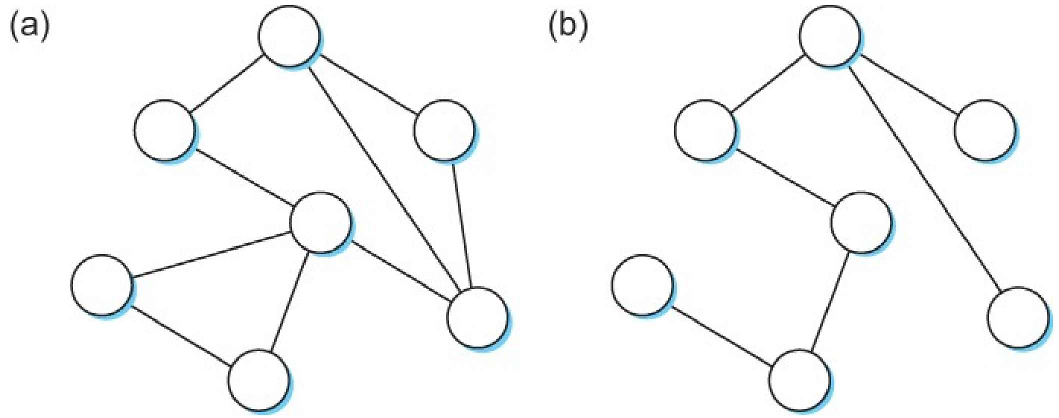


Spot the loop
Why?

Solution? Spanning Tree

Think of the extended LAN as being represented by a graph that possibly has loops (cycles)

- A spanning tree is a sub-graph of this graph that covers all the vertices but contains no cycles
- Spanning tree keeps all the vertices of the original graph but throws out some of the edges



Example of (a) a cyclic graph; (b) a corresponding spanning tree.

How do we create a spanning tree?

- Properties: No loops

- How?
  - Selectively flood

  - Distributed algorithm, no coordination!
  - Automatic reconciliation when failure occurs

How do we create a spanning tree?

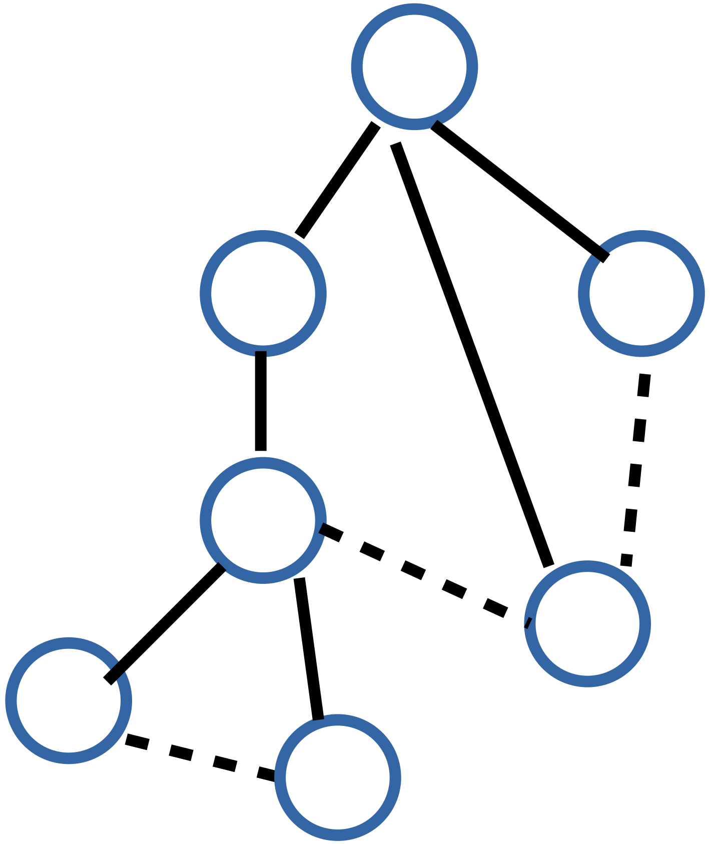
- Properties: No loops

- How?
  - Selectively flood

  - Distributed algorithm, no coordination!
  - Automatic reconciliation when failure occurs

- Switches elect a root
  - The switch with the smallest identifier
  - Each switch identifies if its interface is on the shortest path from the root
  - Exclude if not

- Send message (Y,d,X)
- From x, claims Y is the root, distance is d



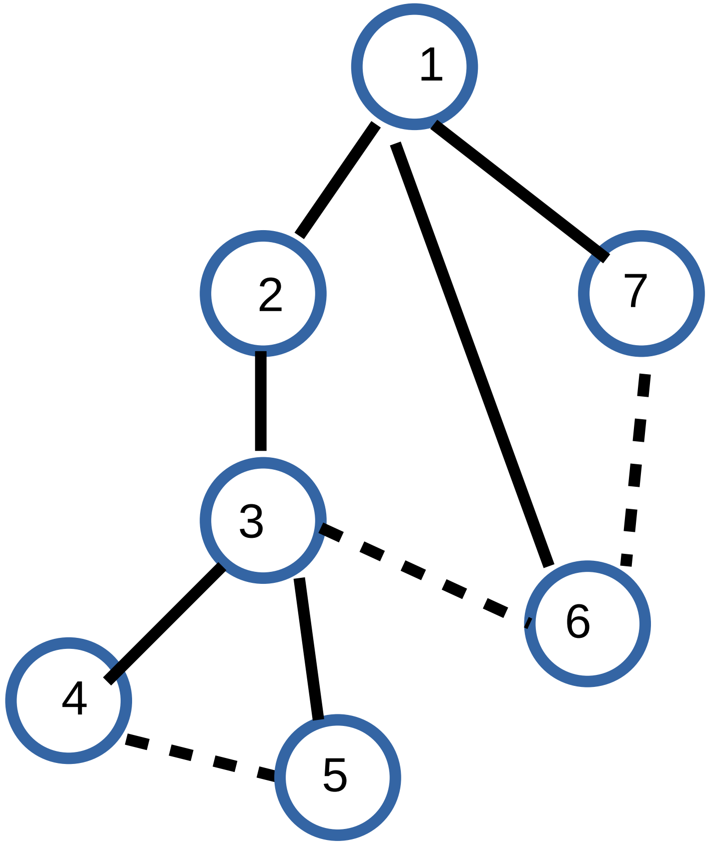
How do we create a spanning tree?

- **Message (Y, d, X) - (to, distance, from)**

- 4 thinks it's the root

- Sends (4, 0, 4) to 3 and 5
- Receives (3,0,3) from 3
  - Sets it to as the root since 3 < 4

- Receives (3,1,5) from 5
  - Sees that this is a longer path to 3
  - 2 hops vs direct path (1 hop)
  - Removes 4-5 link from the tree



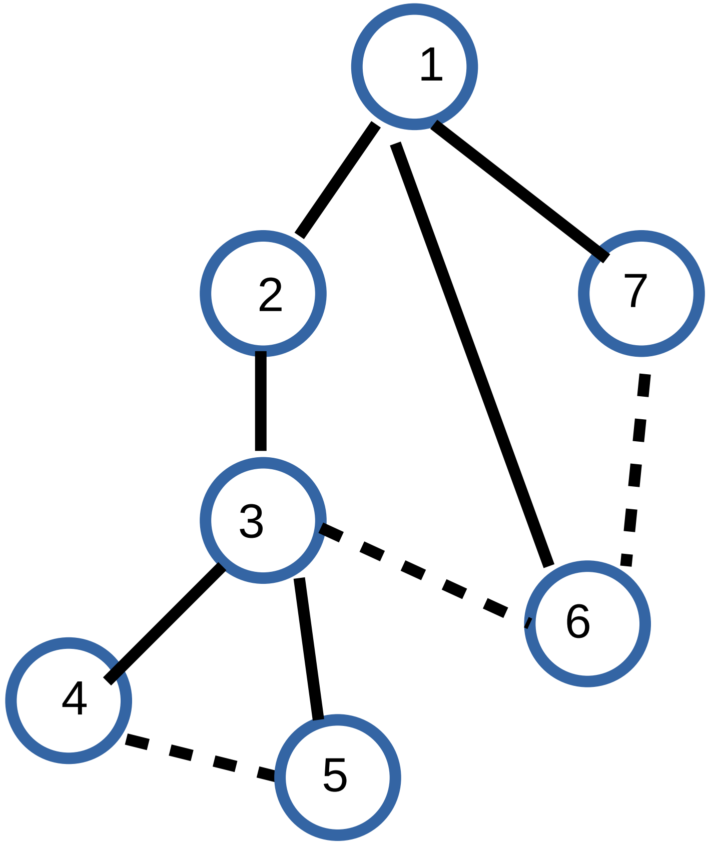
What does 4 do when it hears from 2?

- **Message (Y, d, X) - (to, distance, from)**

- 2 hears (1, 0, 1) from 1
- 2 sends (1, 1, 2) to 3
- 3 sends (1, 2, 3) to 5 and 4

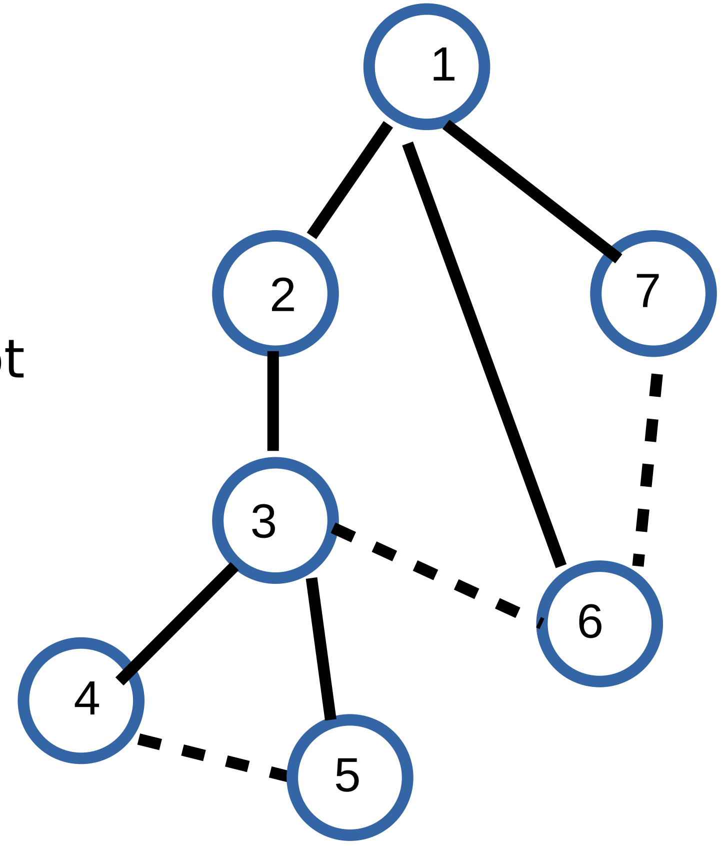
- 4 receives (1, 2, 3) from 3
- 4 receives (1, 3, 5) from 5

- Sets 1 as root (id=1 is < id=4)
- Prunes the 4-5 path since it is 4 hops compared to 3 hops via 3



Failure and Downsides

- Even after the system has stabilized, the root continues to send messages periodically
  - Other bridges continue to forward these messages
- When a bridge fails, the downstream bridges will not receive the configuration messages
  - After waiting a specified period of time, they will once again claim to be the root and the algorithm starts again
- No load balancing



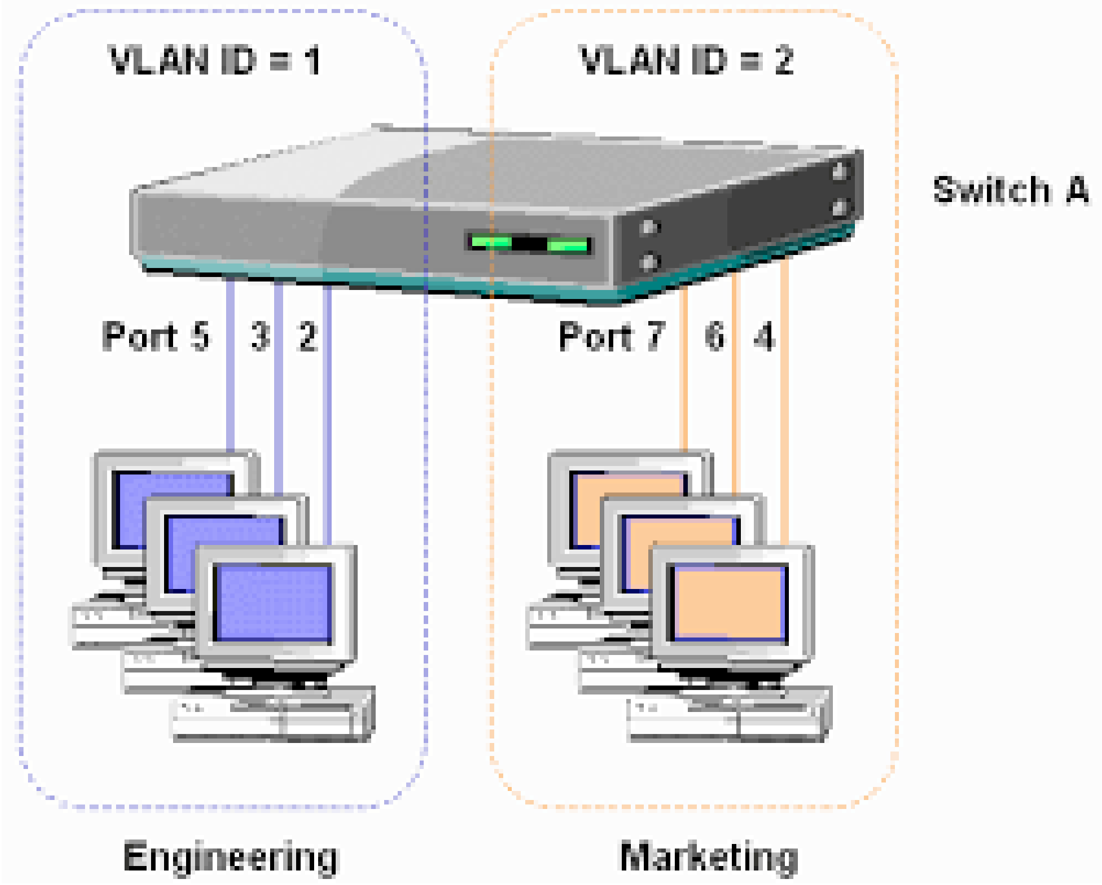
Virtual LAN (VLANs)

- LANs are on the same Ethernet segments
- Does not scale very well – too many wires
- How can we put multiple people in different locations on the same Ethernet segment (LAN)?
- How do we create multiple LANs over the same wire?

Why separate at all?

- LANs are on the same Ethernet segments! Security.
- Isolation – sensitive traffic vs normal traffic
- Containment of traffic – your for loop broke the internet
- How do we create multiple LANs over the same wire?

VLANs



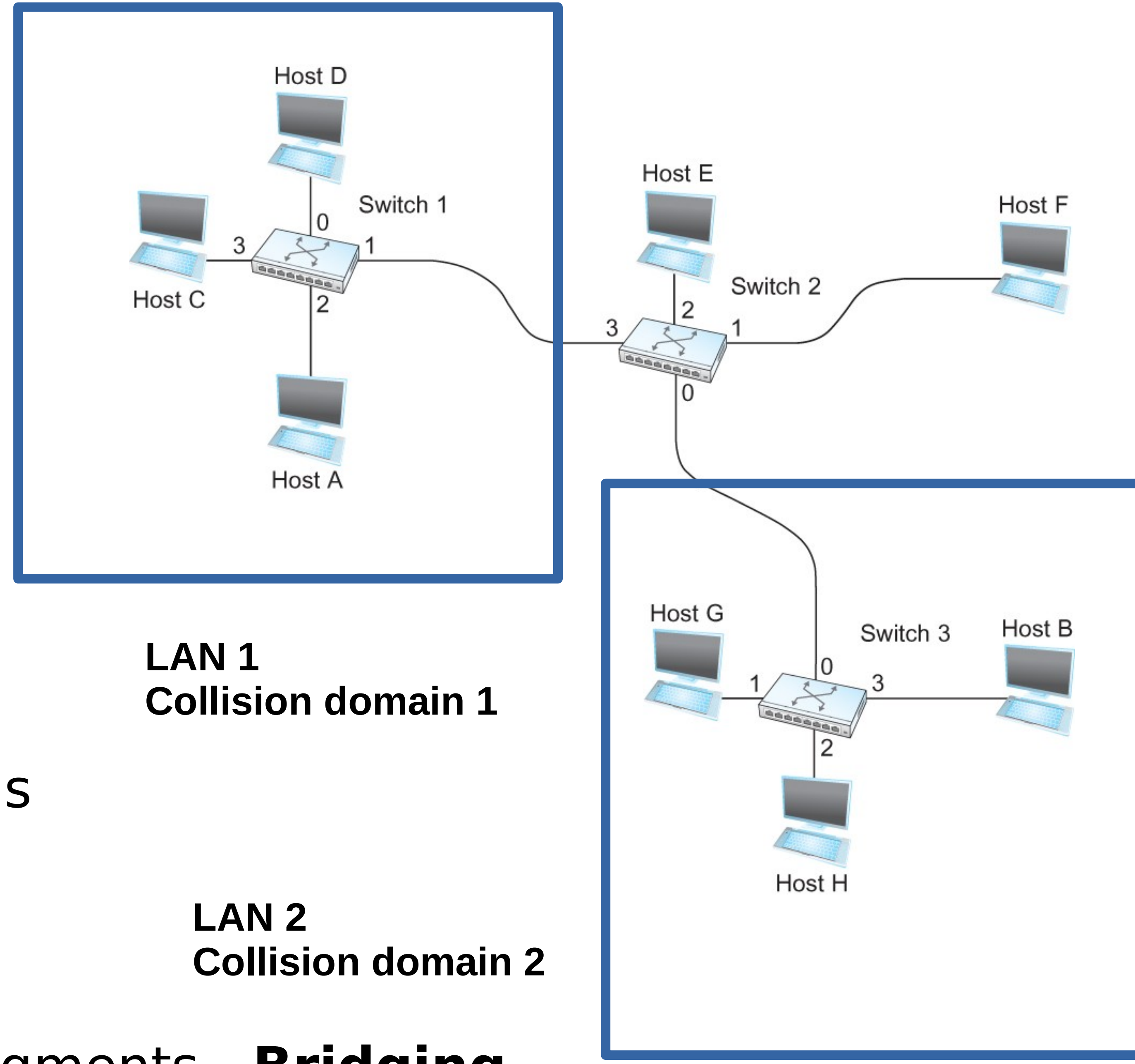
- Switches specify which VLAN is accessible over which interface
- Each interface can have a VLAN color
- Each Mac address can have a interface color
- Add VLAN tag to the Ethernet header

So far...

- We are forwarding packets between different LANs
- Spanning tree algorithm for preventing loops

Switching

- Switch
  - A mechanism to interconnect links to form a large network
  - Forward **frames**
  - Separate the collision domains
  - Filter packets between LANs
  - Connects two or more LAN segments - **Bridging**



How do we create a spanning tree?

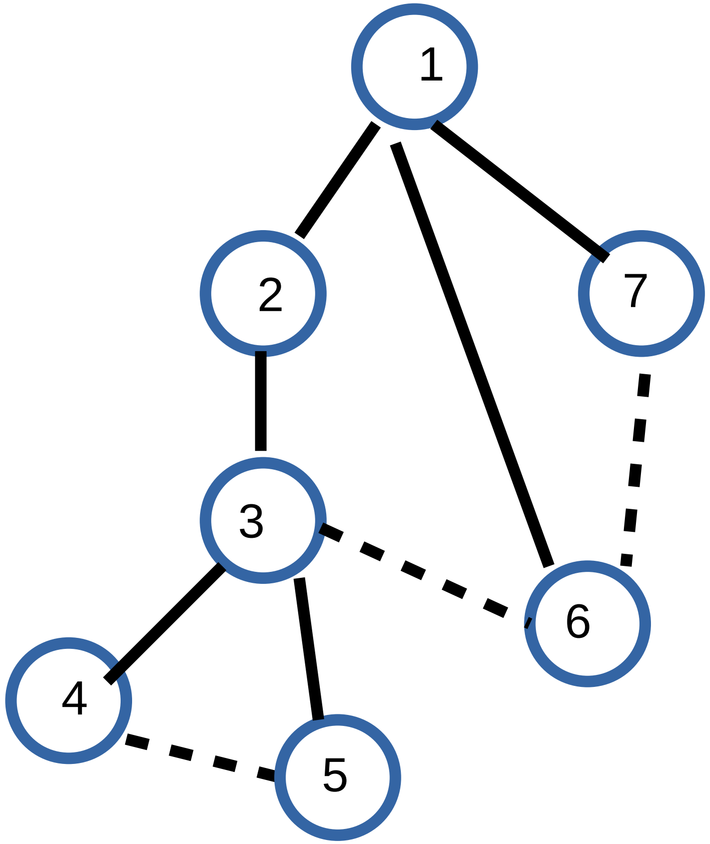
- **Message (Y, d, X) - (to, distance, from)**

- 4 thinks it's the root

- Sends (4, 0, 4) to 3 and 5
- Receives (3,0,3) from 3
  - Sets it to as the root since 3 < 4

- Receives (3,1,5) from 5
  - Sees that this is a longer path to 3
  - 2 hops vs direct path (1 hop)
  - Removes 4-5 link from the tree

- **Does not scale!**

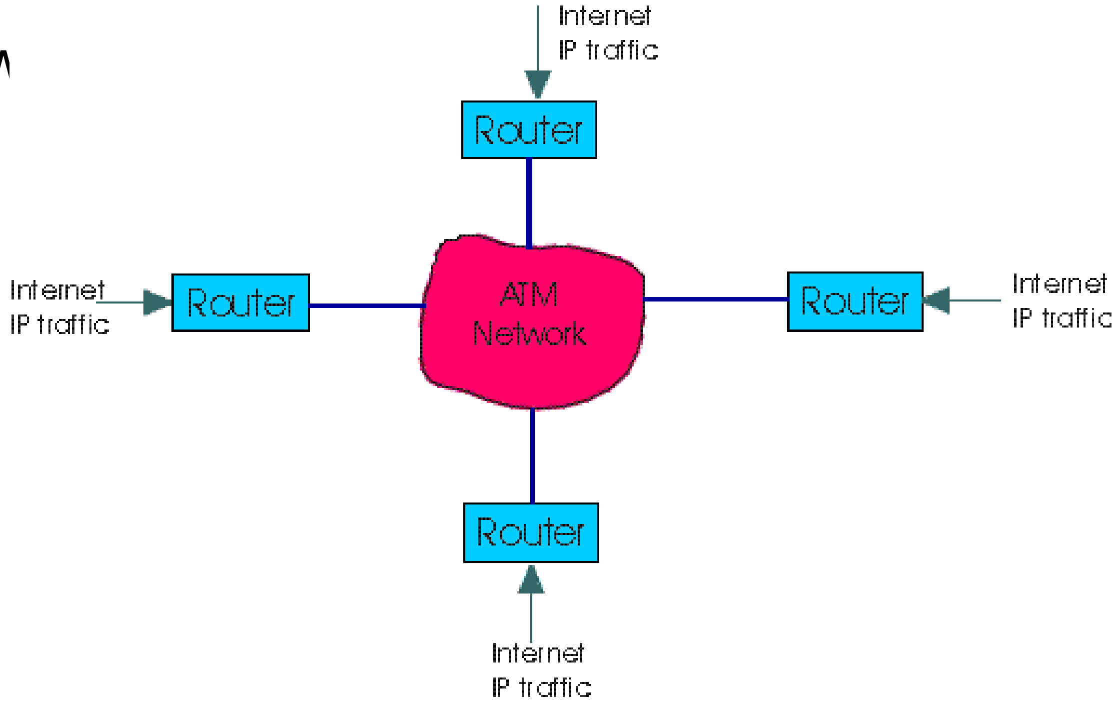


ATM (Carries Cells, not Money)

- ATM (Asynchronous Transfer Mode)
  - Connection-oriented packet-switched network
- Packets are called cells
- 5 byte header + 48 byte payload
- Fixed length packets are easier to switch in hardware
- **Why?**

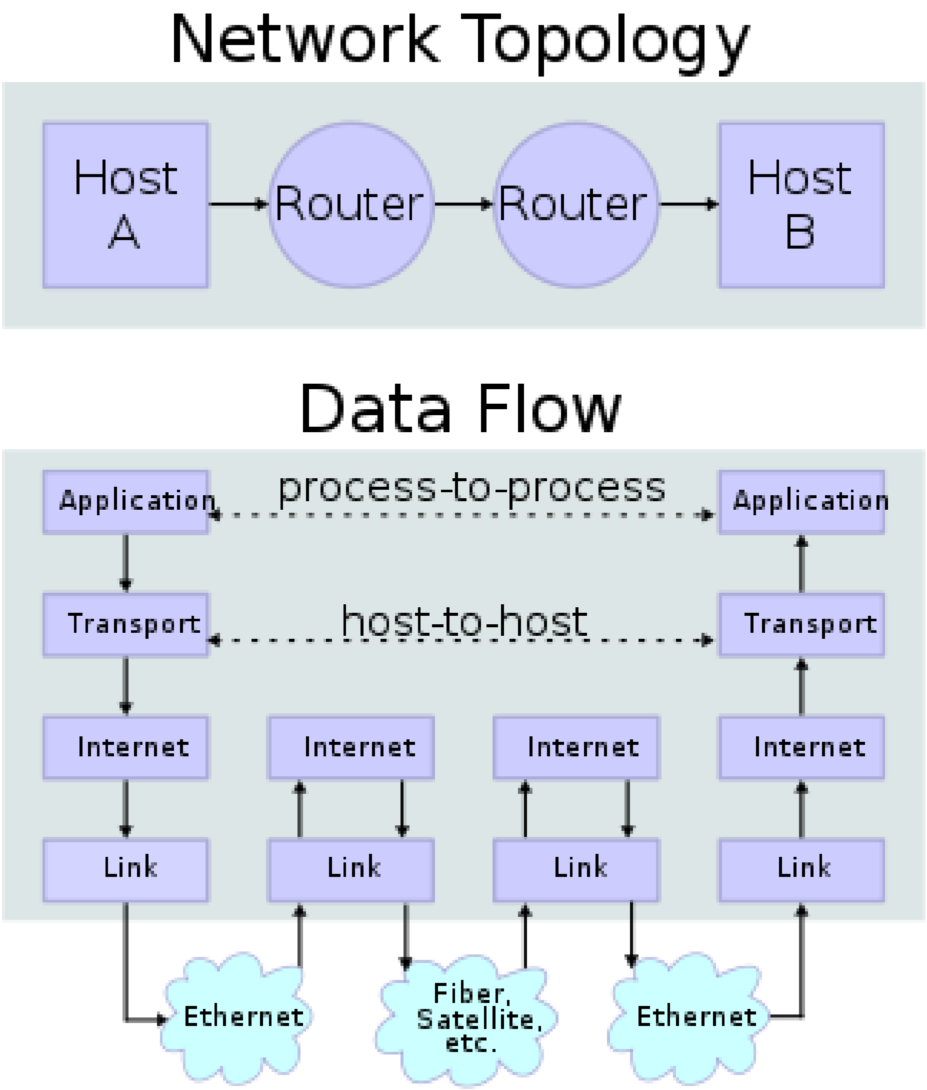
ATM (Carries Cells, not Money)

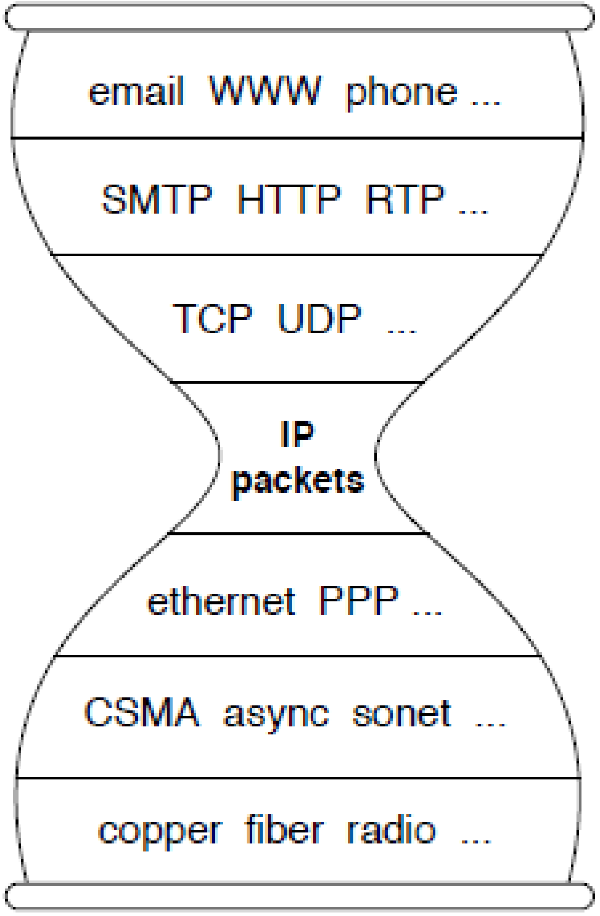
- ATM (Asynchronous Transfer Mode)
  - Connection-oriented packet-switched netw
  - Packets are called cells
  - 5 byte header + 48 byte payload
- Fixed length packets are easier to switch in hardware
  - Simpler to design
  - Enables parallelism
- Still used in long distance private links



kurose/ross

IP Suite – From the First Lecture

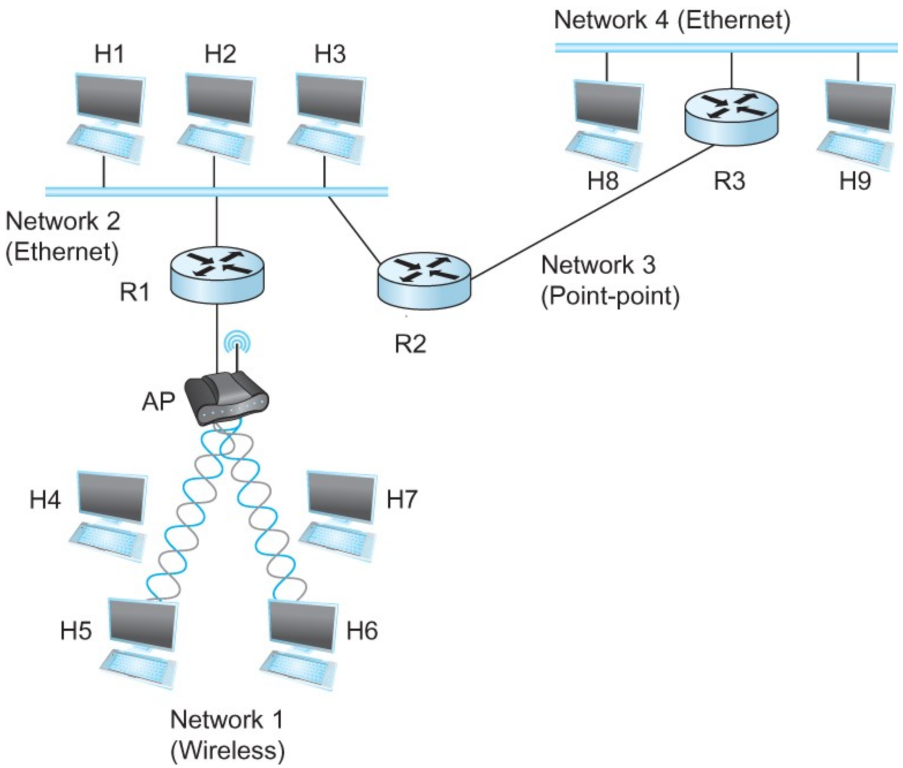




wikipedia

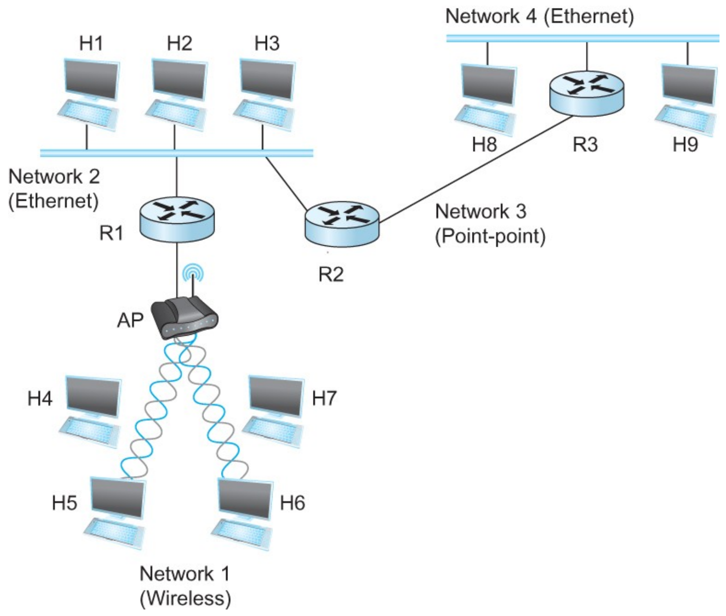
Internet Protocol (IP)

- What is an internetwork?
  - An arbitrary collection of networks interconnected to provide some sort of host-host to packet delivery service



But that’s what switches are for – No?

- Switches create networks, Routers connect different networks.
- Typically switches are at **Layer 2**, Routers are at **Layer 3**
- Switches forward **FRAMES,** Routers forward **PACKETS**



Apps (HTTP)

Transport (TCP/UDP)

Network (IP)

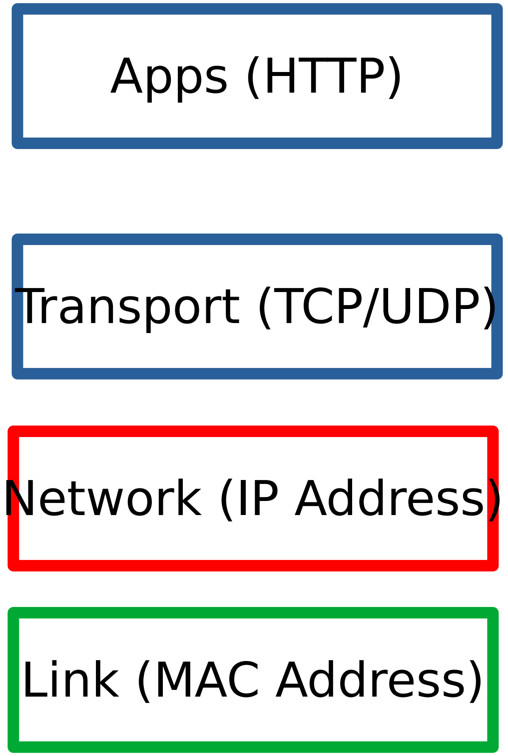
Link (Ethernet)

But that’s what switches are for – No?

- This room → Point-to-point link
- This room + next room → Switch
- This room + next room + foundation hall → Switches with VLAN
- This university + Internet → Router
- **Good for conceptualization - not always as simple**

Every device has a MAC – Why do we need another address?

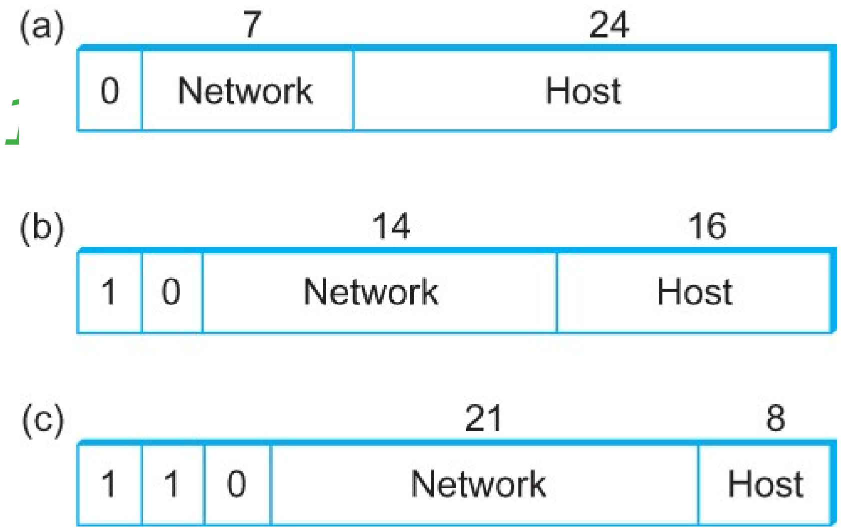
- Ethernet (MAC) addresses are flat
- Not the only link layer
- Not related to network topology
  - Remember – we are still connecting to hosts!
  - How do we go from: 52:54:00:86:38:14 to tntech?
  - **Other reasons?**



Global Address in IP – Each node has an unique address

- A 32 bit number in quad-dot notation
- Identifies an ***Interface***
  - ***A host might have several interfaces!!!***
- *129.82.138.254*

  10000001.01010010.10001010.111

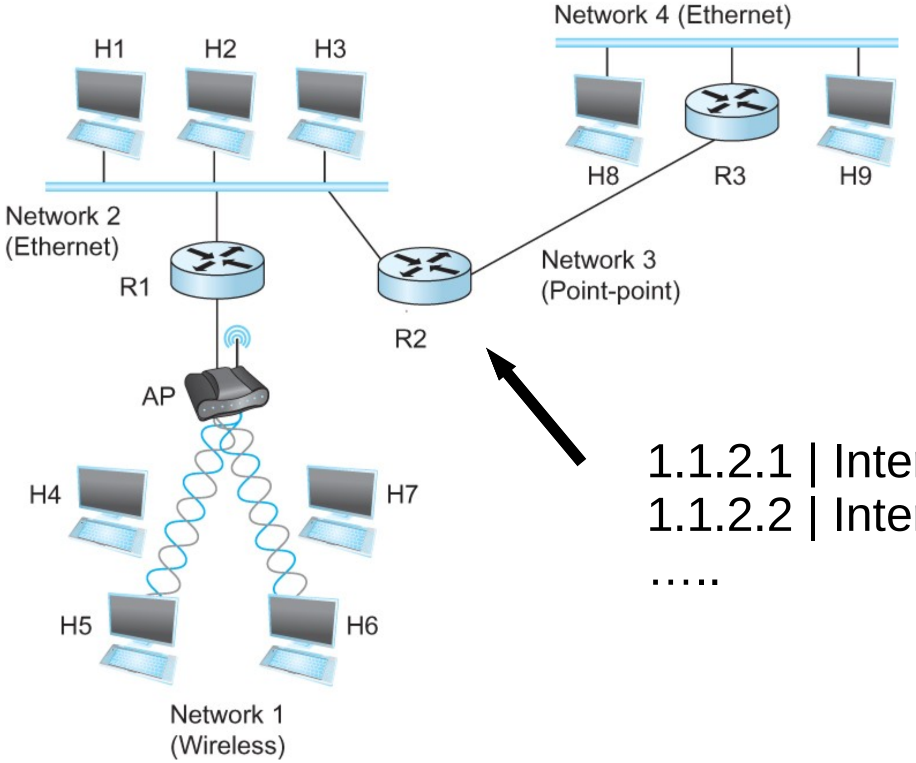


IP allows the network to scale!

- What if addresses were arbitrary?

5.1.6.7

1.1.2.2



1.1.2.1

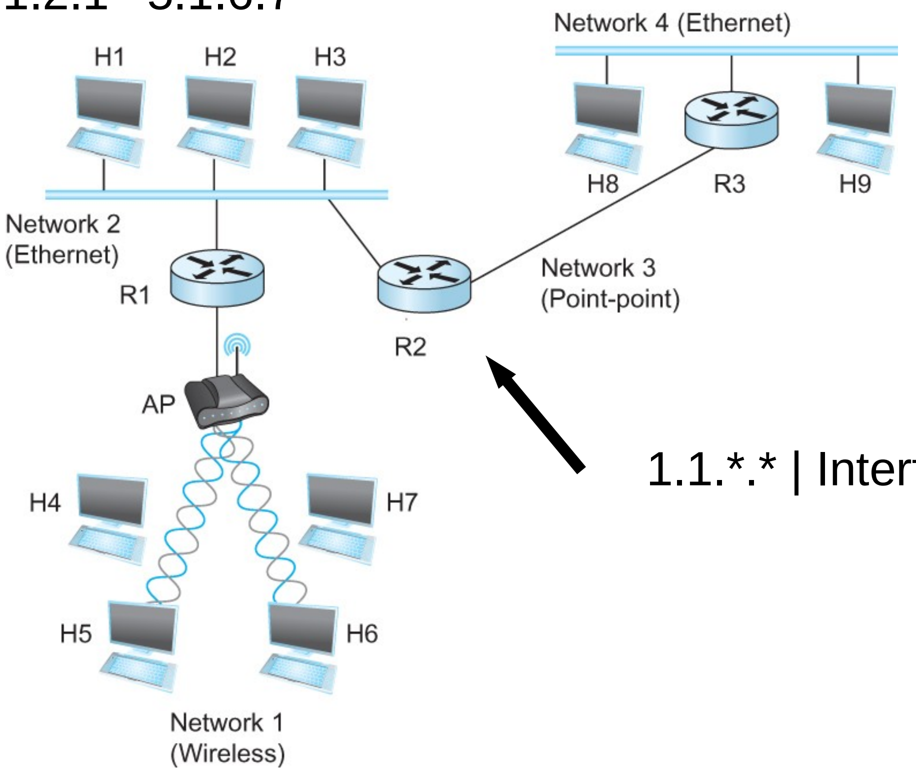
1.1.2.1 | Interface 3
1.1.2.2 | Interface 2
…..

Solution - Group hosts

- What if addresses were arbitrary?

1.1.2.1 ~~5.1.6.7~~

1.1.2.2



1.1.*.* | Interface 3

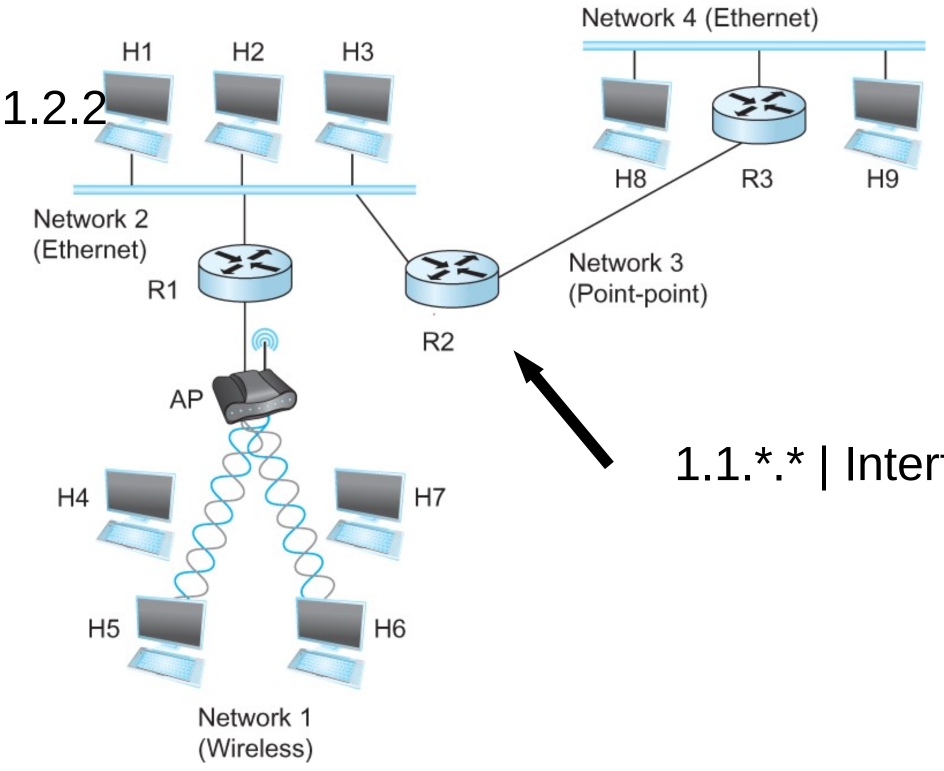
IP addresses are in Network + Host

- 1.1.2.1 →
  - 1.1 → Network part
  - 2.1 → host part
- Each octet can range from 1- 255

- Hierarchical address

1.1.2.1 ~~5.1.6.7~~

1.1.2.2



1.1.*.* | Interface 3

129.82.138.254

10000001.01010010.10001010.11111110

Network part (24 bits). Host part(8 bits)

How do we know host vs network → Subnetting

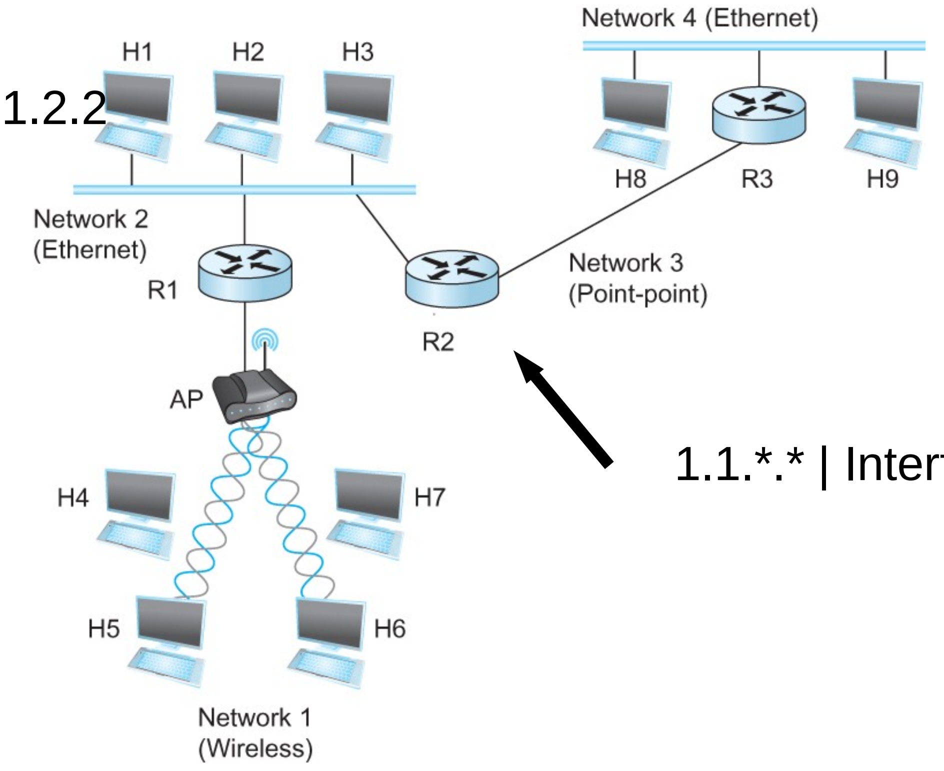
129.82.138.254 (Address)

10000001.01010010.10001010.11111110
11111111.11111111. 11111111.00000000

255.255.255.0 (Subnet mask)

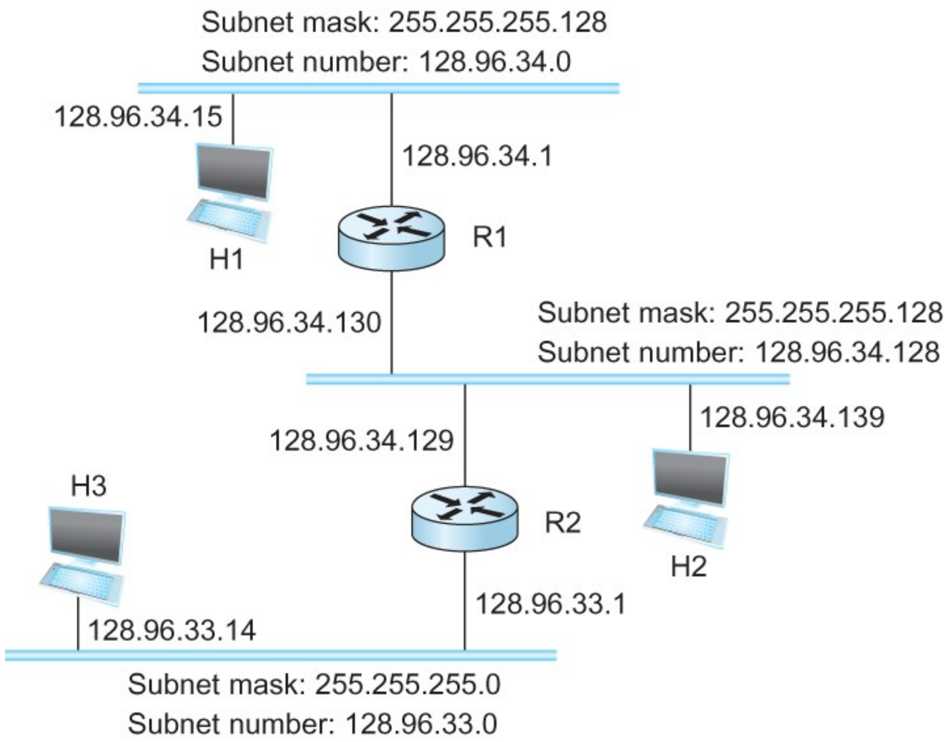
1.1.2.1 ~~5.1.6.7~~

1.1.2.2



1.1.*.* | Interface 3

Subnetting



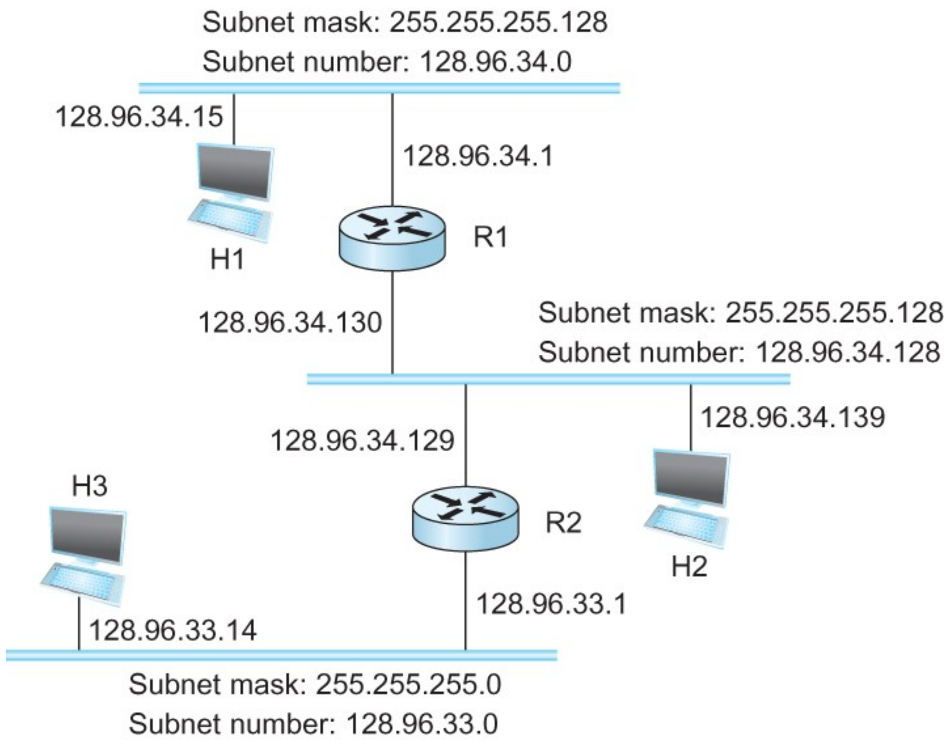
Forwarding Table at Router R1

SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

Subnetting

Three classes:

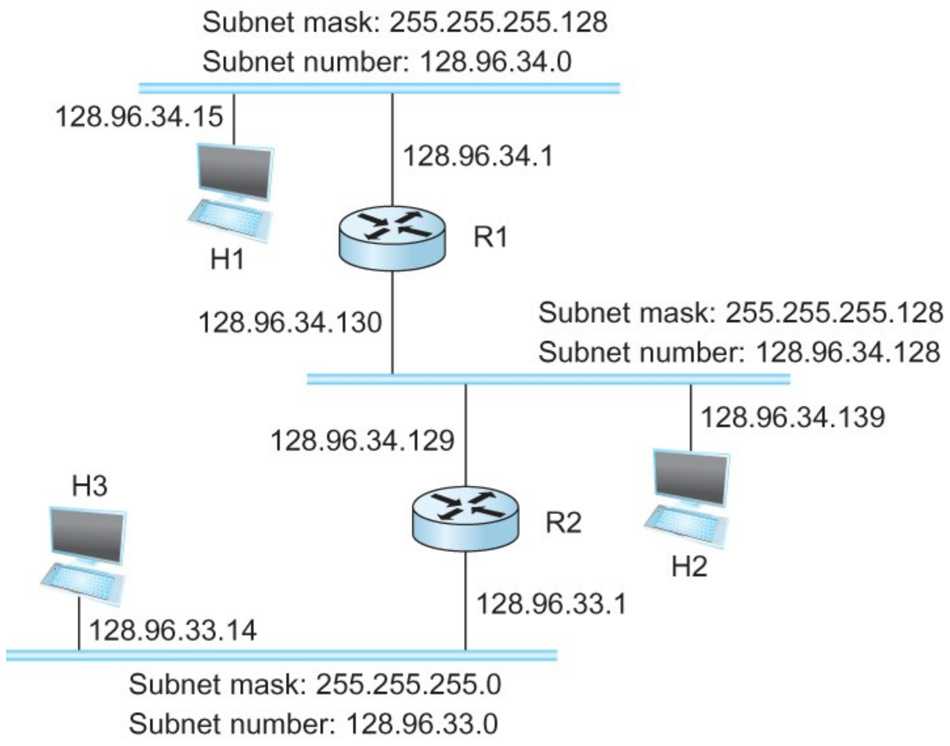
- Class A: 129.0.0.0/8
- Class B: 129.82.0.0/16
- Class C: 129.82.2.0/14



SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

Well, not really!

- CIDR: Classless Interdomain routing
- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address
  - 129.82.13.0/23
  - More flexible



SubnetNumber	SubnetMask	NextHop
128.96.34.0	255.255.255.128	Interface 0
128.96.34.128	255.255.255.128	Interface 1
128.96.33.0	255.255.255.0	R2

Now routers can operate on Network address!!!!

129.82.138.254 (Address)

10000001.01010010.10001010.11111110
11111111.11111111. 11111111.00000000

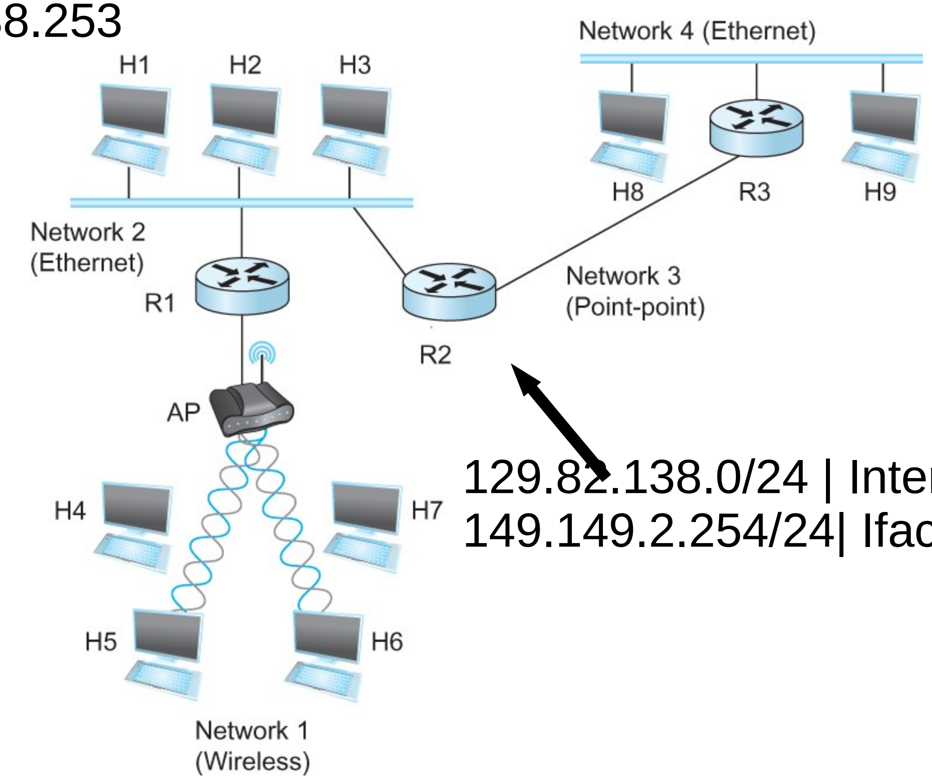
255.255.255.0 (Subnet mask)

129.82.138.254 + 255.255.255.0 → 129.82.138.0/24

149.149.2.254

129.82.138.252

129.82.138.253

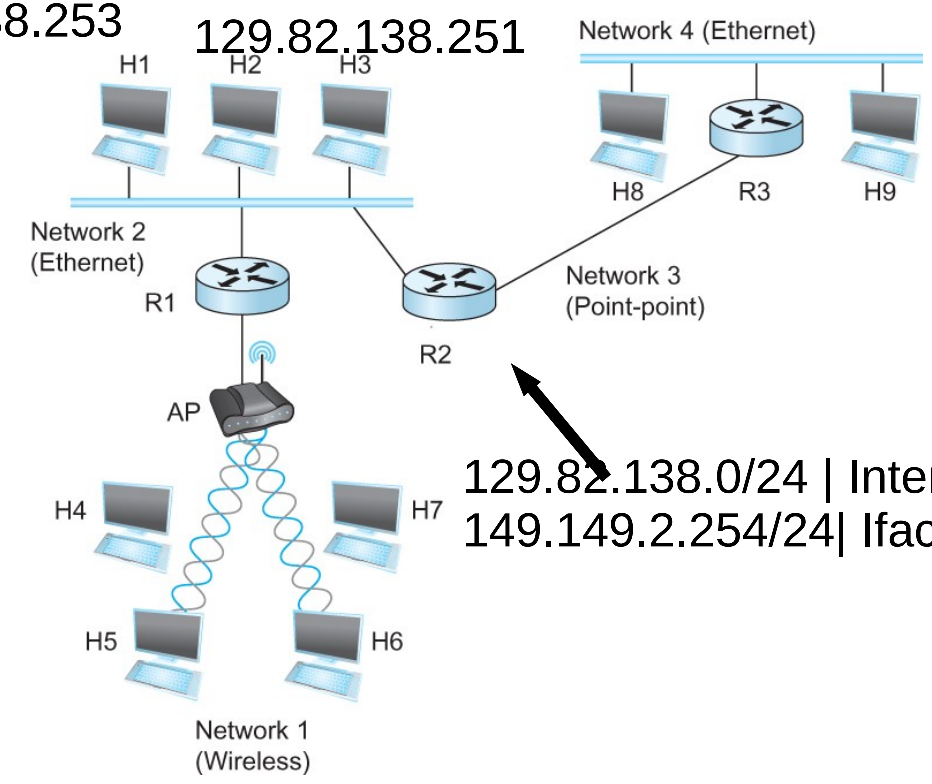


129.82.138.0/24 | Interface 2
149.149.2.254/24| Iface 3

Address management is localized

No coordination needed for adding 129.82.138.251

No routing update needs to go out



149.149.2.254

129.82.138.252

129.82.138.253

129.82.138.251

129.82.138.0/24 | Interface 2
149.149.2.254/24| Iface 3

Address management can be automated

ARP:

  Map IP address to MAC address

DHCP:

  Learn IP address, gateway, DNS

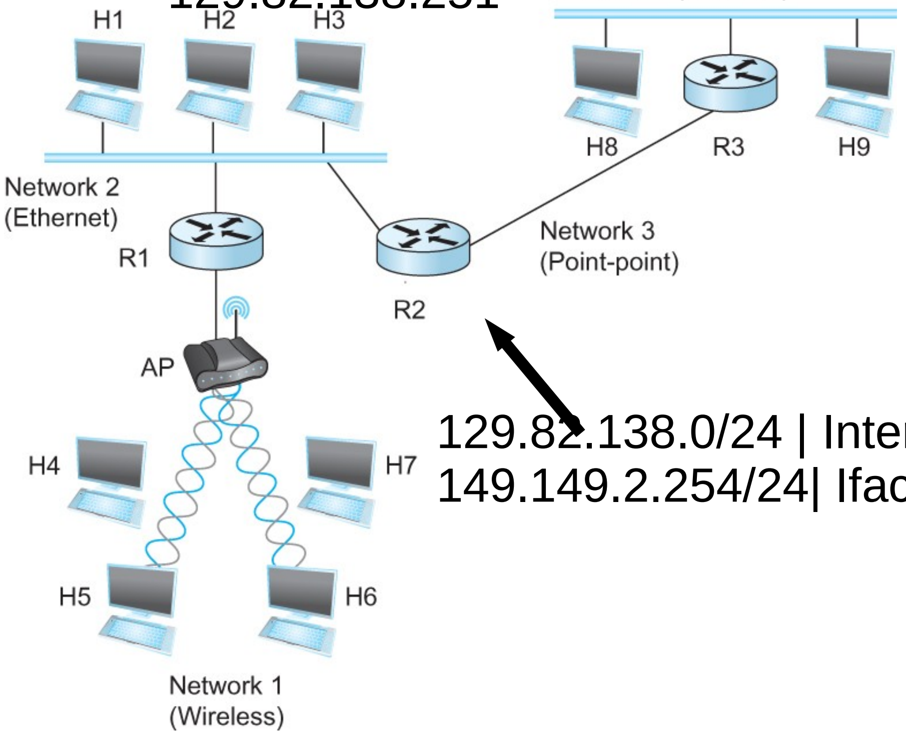
More on these later.

149.149.2.254

129.82.138.252

129.82.138.253

129.82.138.251



129.82.138.0/24 | Interface 2
149.149.2.254/24| Iface 3

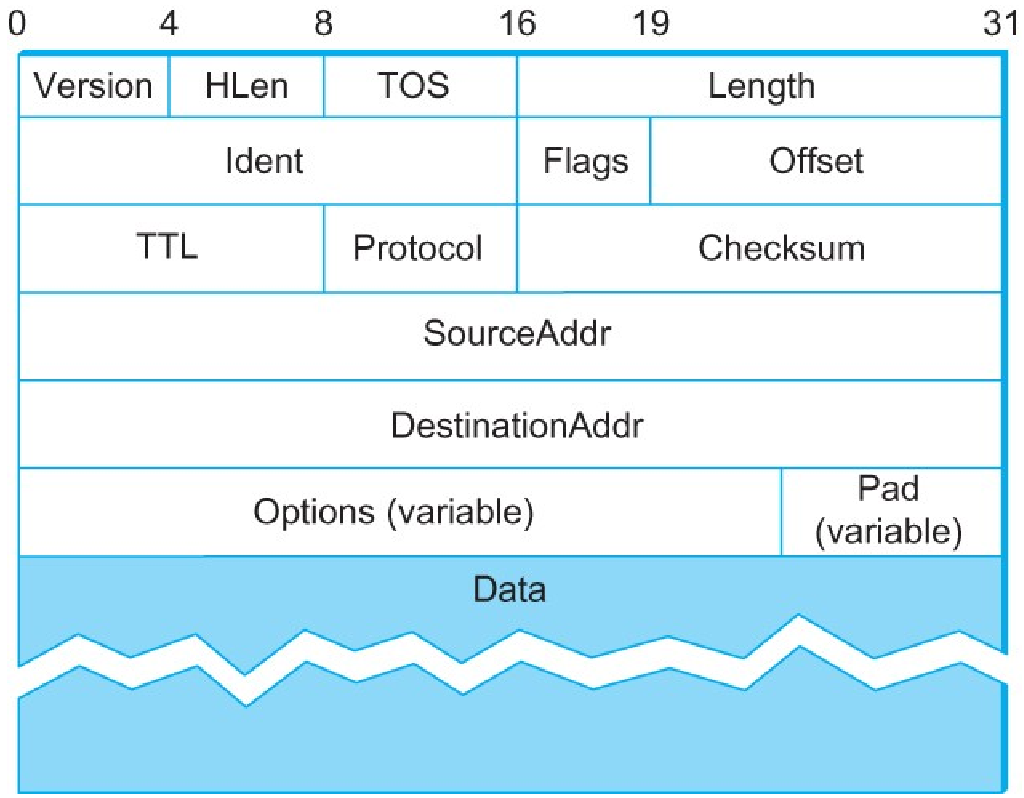
You have an address – Send data now. IP service model

- Packet Delivery Model
  - Connectionless model for data delivery

- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time

- Global Addressing Scheme
  - Provides a way to identify all hosts in the network

IP Packet



Version (4): 4

Hlen (4): number of 32-bit words in header

TOS (8): type of service (not widely used)

Length (16): number of bytes in this datagram

Ident (16): used by fragmentation

Flags/Offset (16): used by fragmentation

TTL (8): number of hops this datagram has traveled

Protocol (8): demux key (TCP=6, UDP=17)

Checksum (16): of the header only

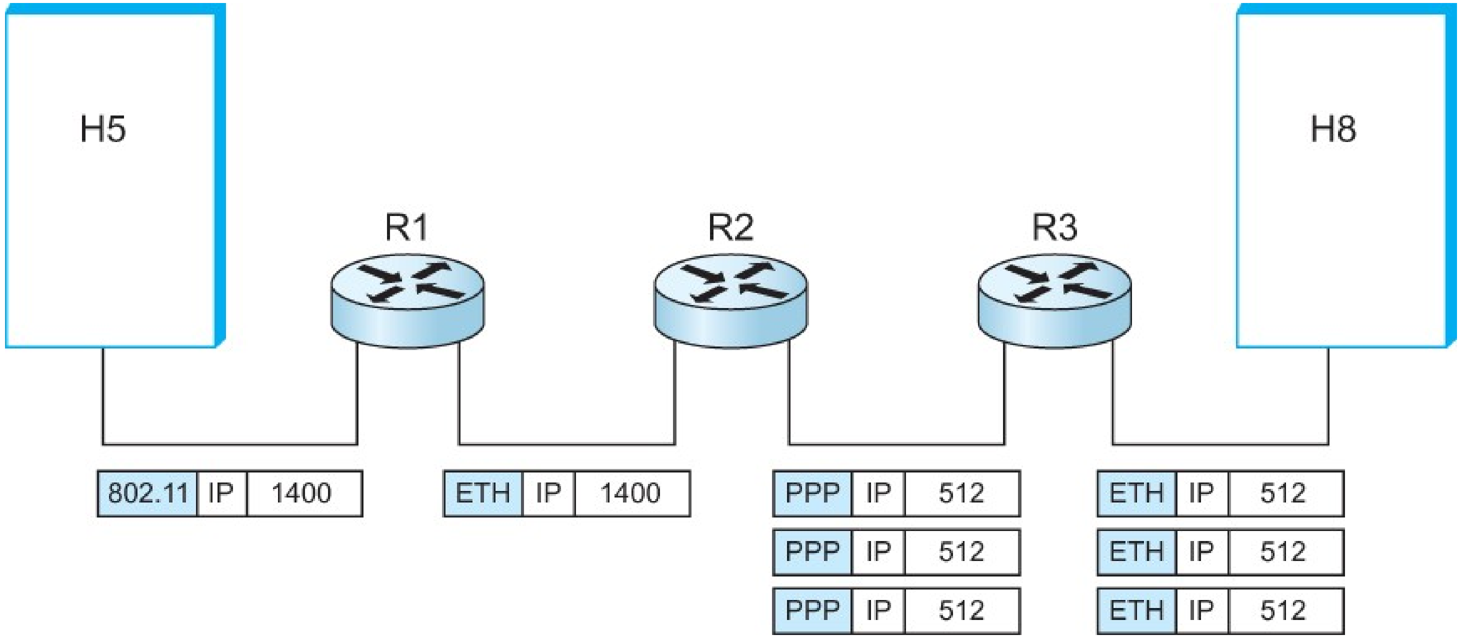
DestAddr & SrcAddr (32)

IP Fragmentation and Reassembly

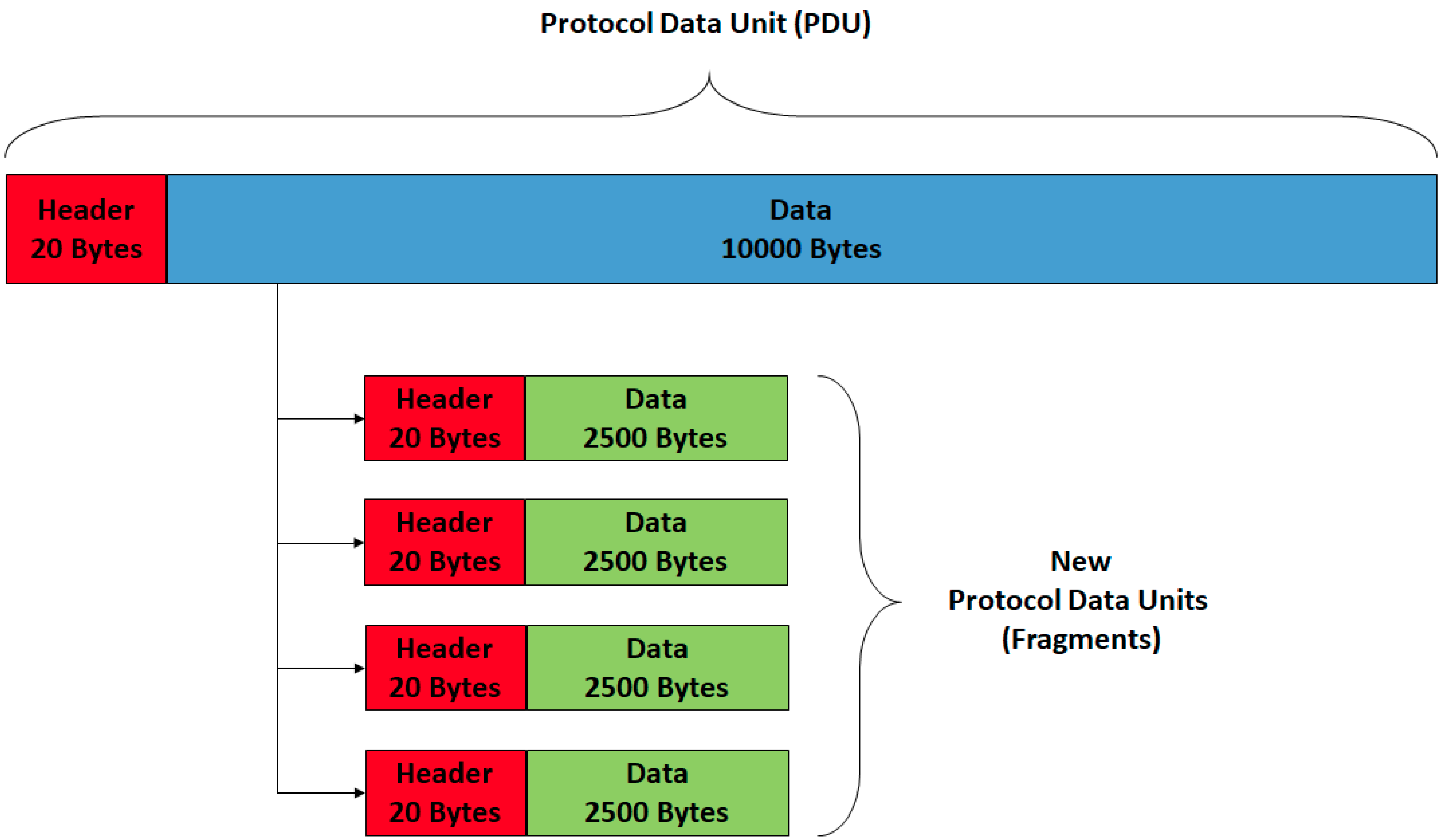
Underlying Layer 2 limitations

- Ethernet 1500
- PPP 512

- Break packets into smaller chunk and reassemble later

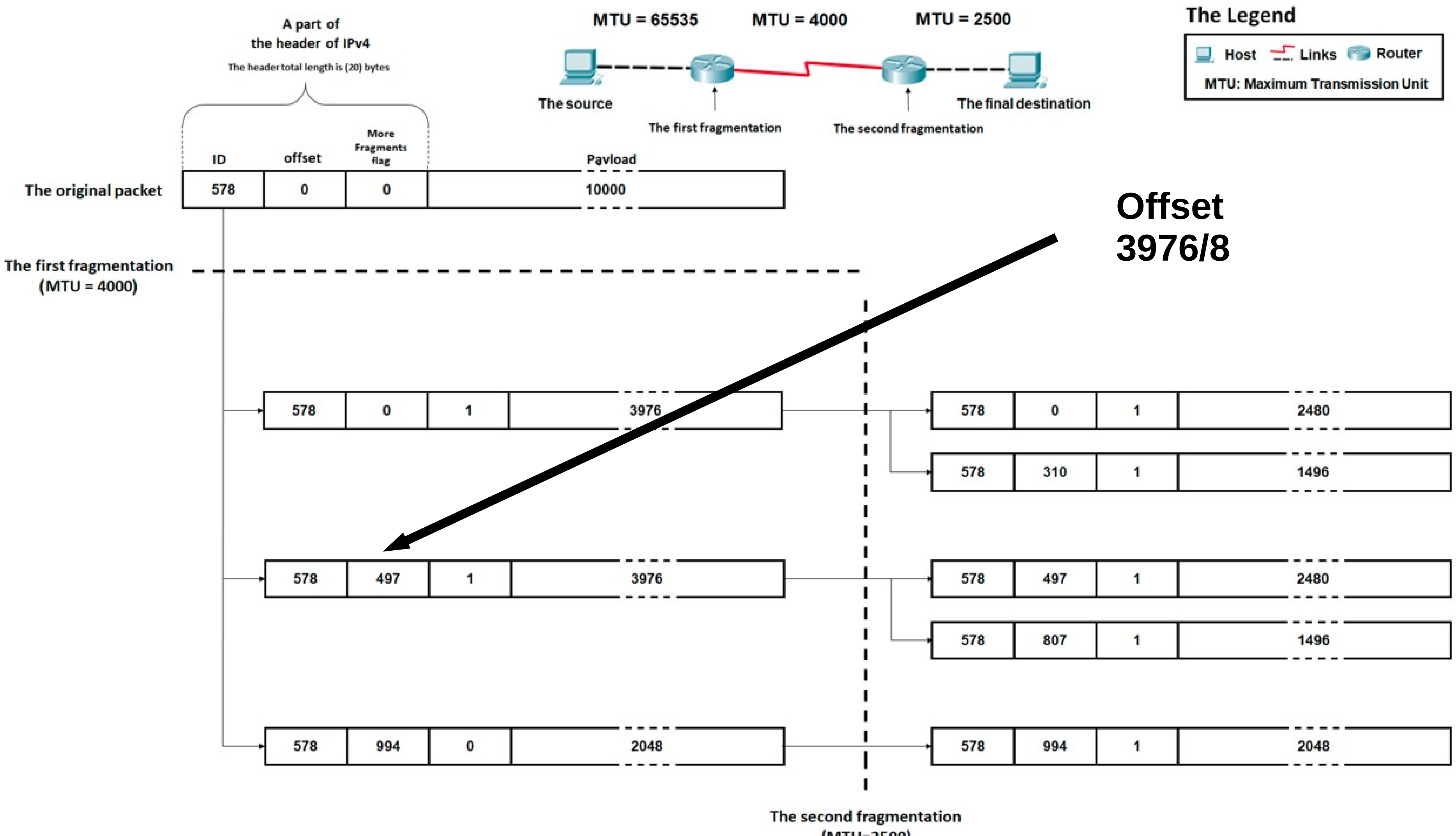


IP Fragmentation and Reassembly



wikipedia

IP Fragmentation and Reassembly



Offset
3976/8

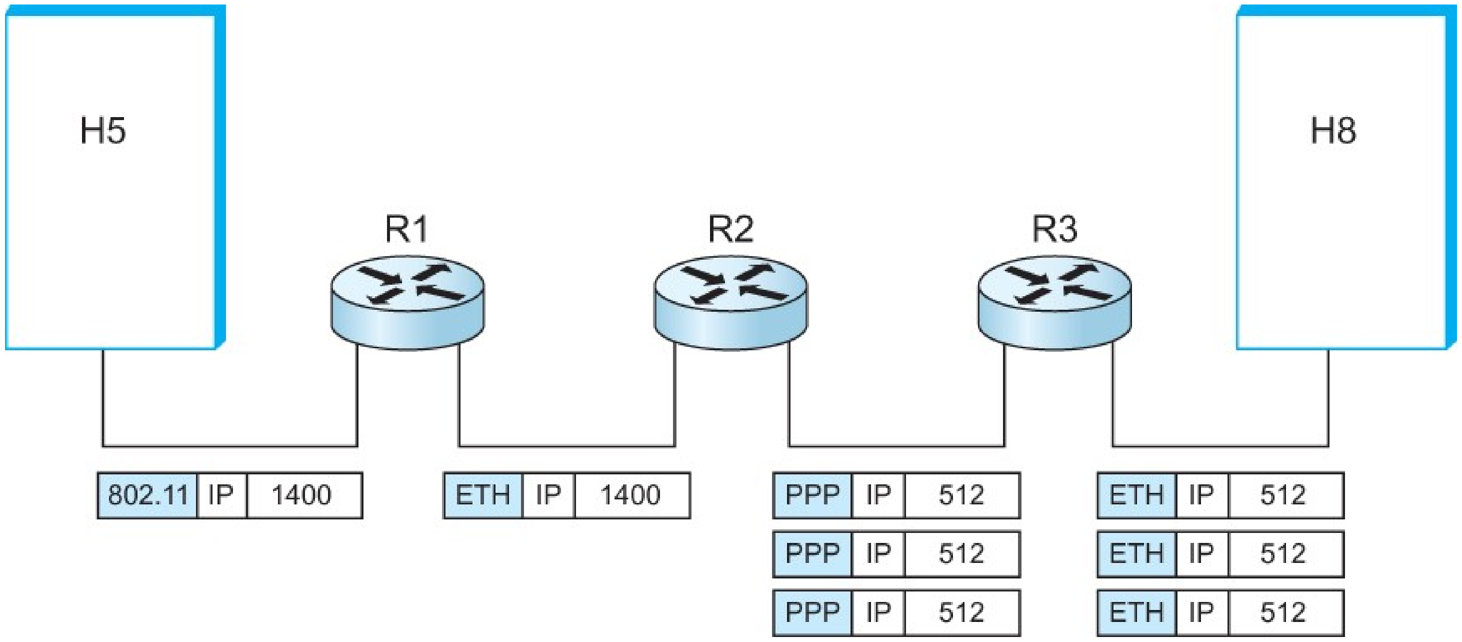
wikipedia

IP Fragmentation and Reassembly

Underlying Layer 2 limitations

- Ethernet 1500
- PPP 512

- Break packets into smaller chunk and reassemble later



Reading Assignments

Internetworking:

https://book.systemsapproach.org/internetworking/basic-ip.html#what-is-an-internetwork

Upto Global Addresses:

https://book.systemsapproach.org/internetworking/basic-ip.html#global-addresses

Reading Assignment

Switching Basics – Chapter 3.1

- https://book.systemsapproach.org/internetworking/switching.html#switching-basics
- *Up to (but not including) Virtual Circuit Switching*
- 20 minutes read

- Switched Ethernet, learning bridges, spanning tree algorithm, VLANs – Chapter 3.2
- https://book.systemsapproach.org/internetworking/ethernet.html#switched-ethernet
  - 30-40 minutes read