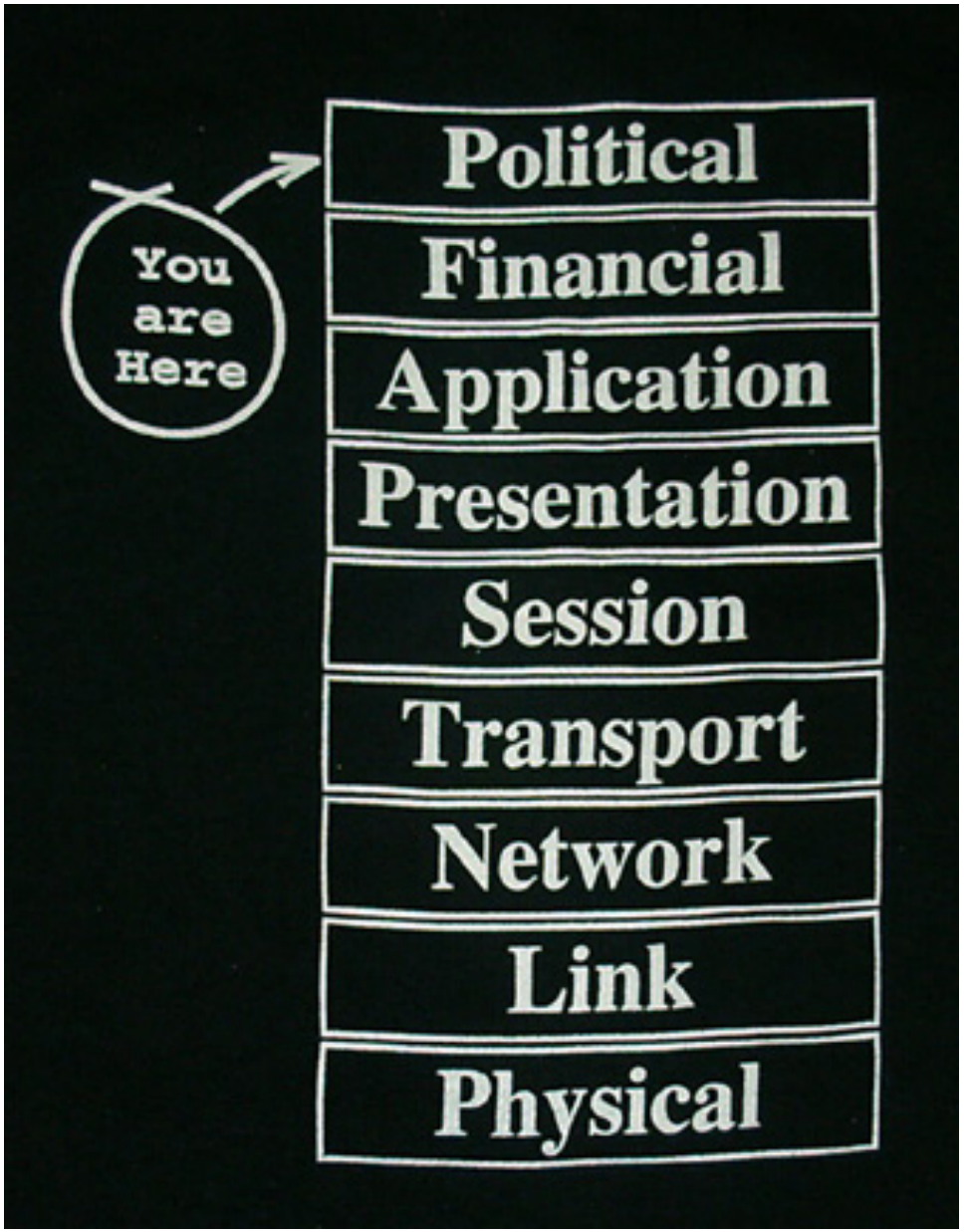
A quick description of Layers 1-4

1. Physical
2. Ethernet
3. IP
4. UDP, TCP, ICMP

Layer 1 = Physical

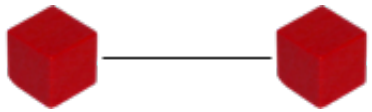
9+ Layer (cough) OSI Model



1

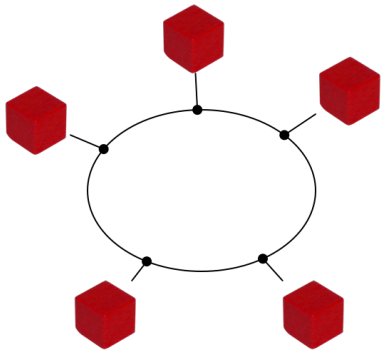
General Methods for Interconnecting Systems

Serial



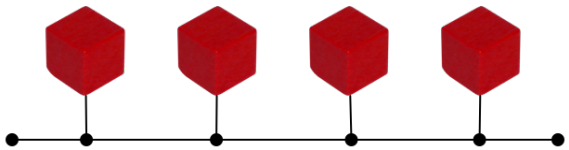
- 1:1 communications
- If you want to link multiple devices, each device needs to handle routing

Token



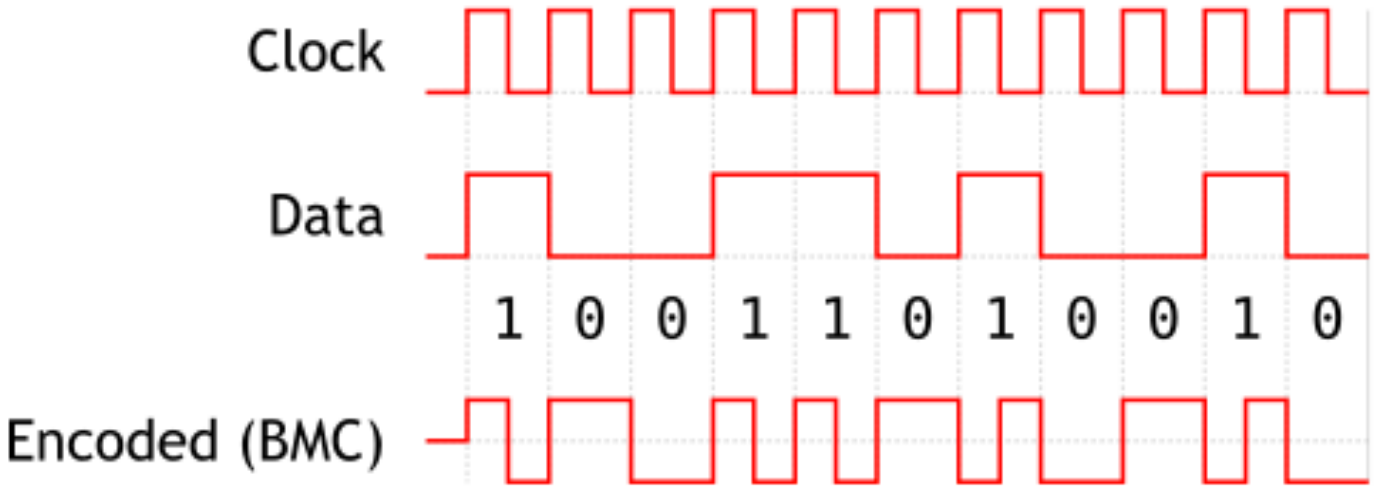
- Shared link, token is passed around to give rights to talk

Collision Detection



- Shared link
- Transmitting is done at random and collisions are managed

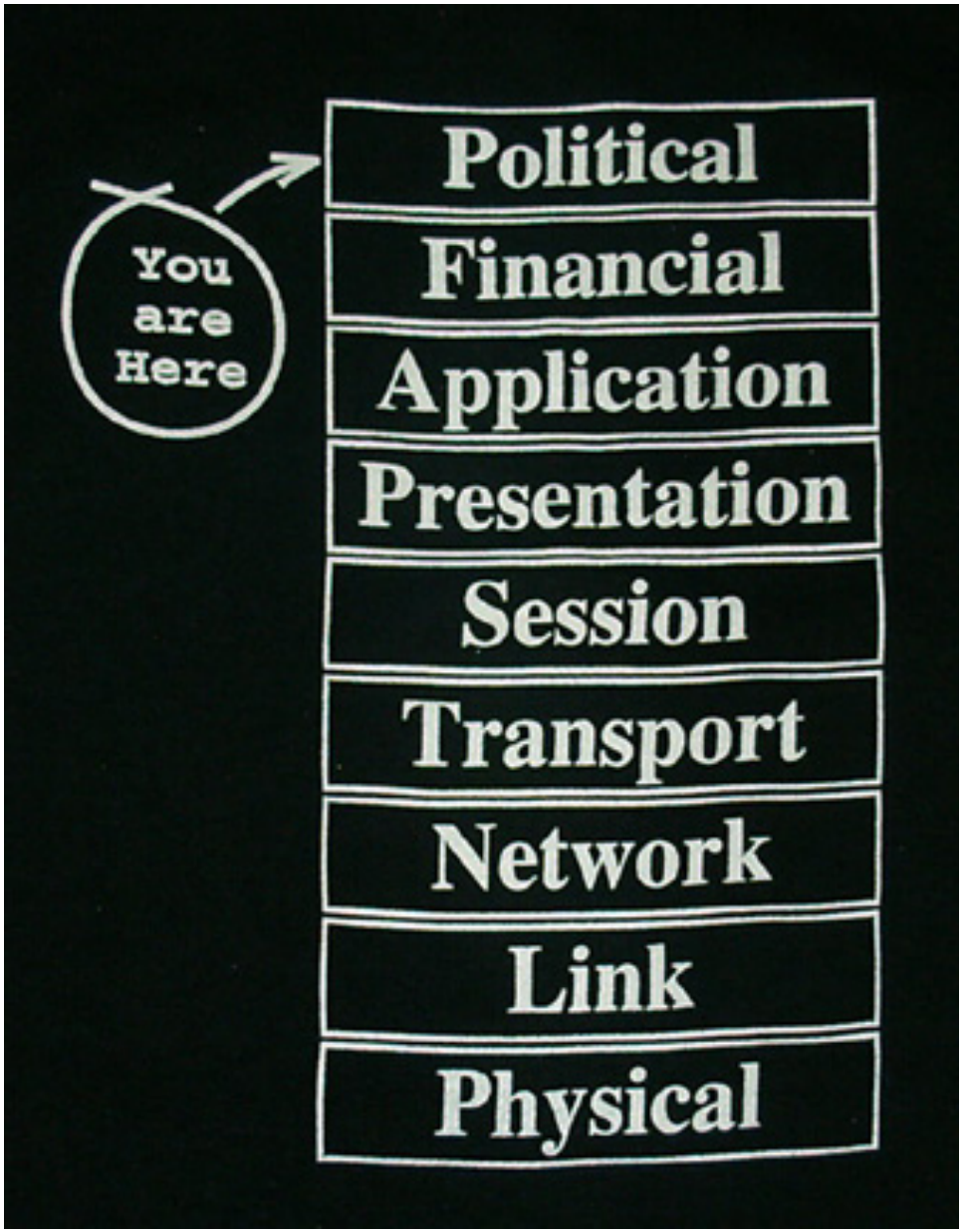
Manchester Encoding



- Some method for transmitting data over a wire.
- In this case, we use manchester encoding to send data with a clock signal over one line.

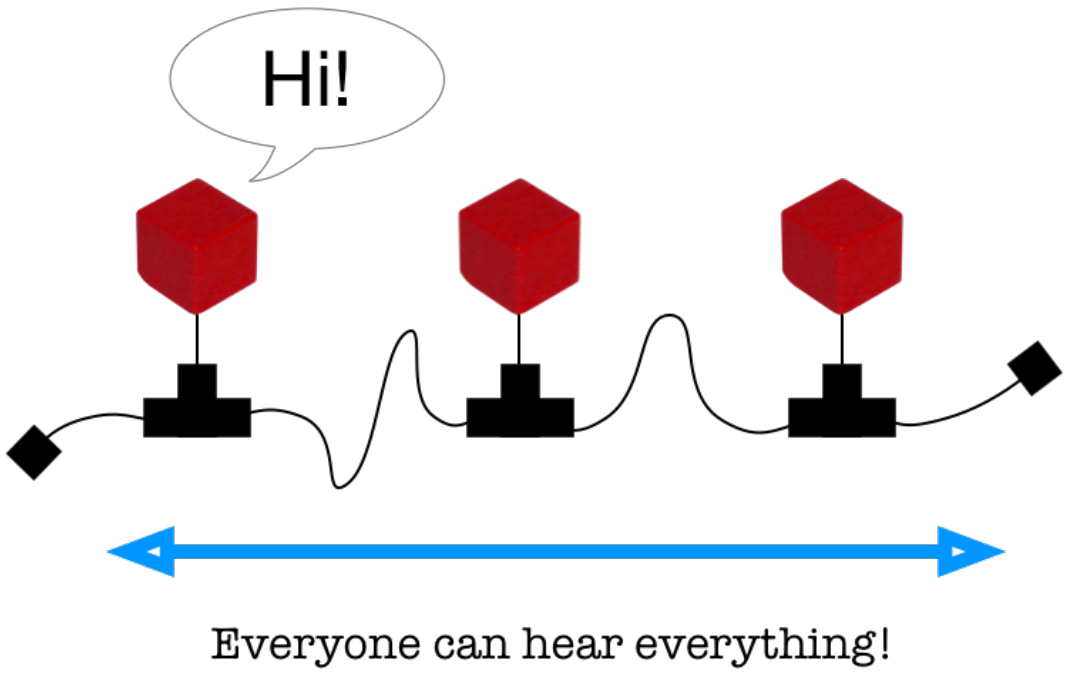
Layer 2 = Ethernet

9+ Layer (cough) OSI Model



2

Olden Days

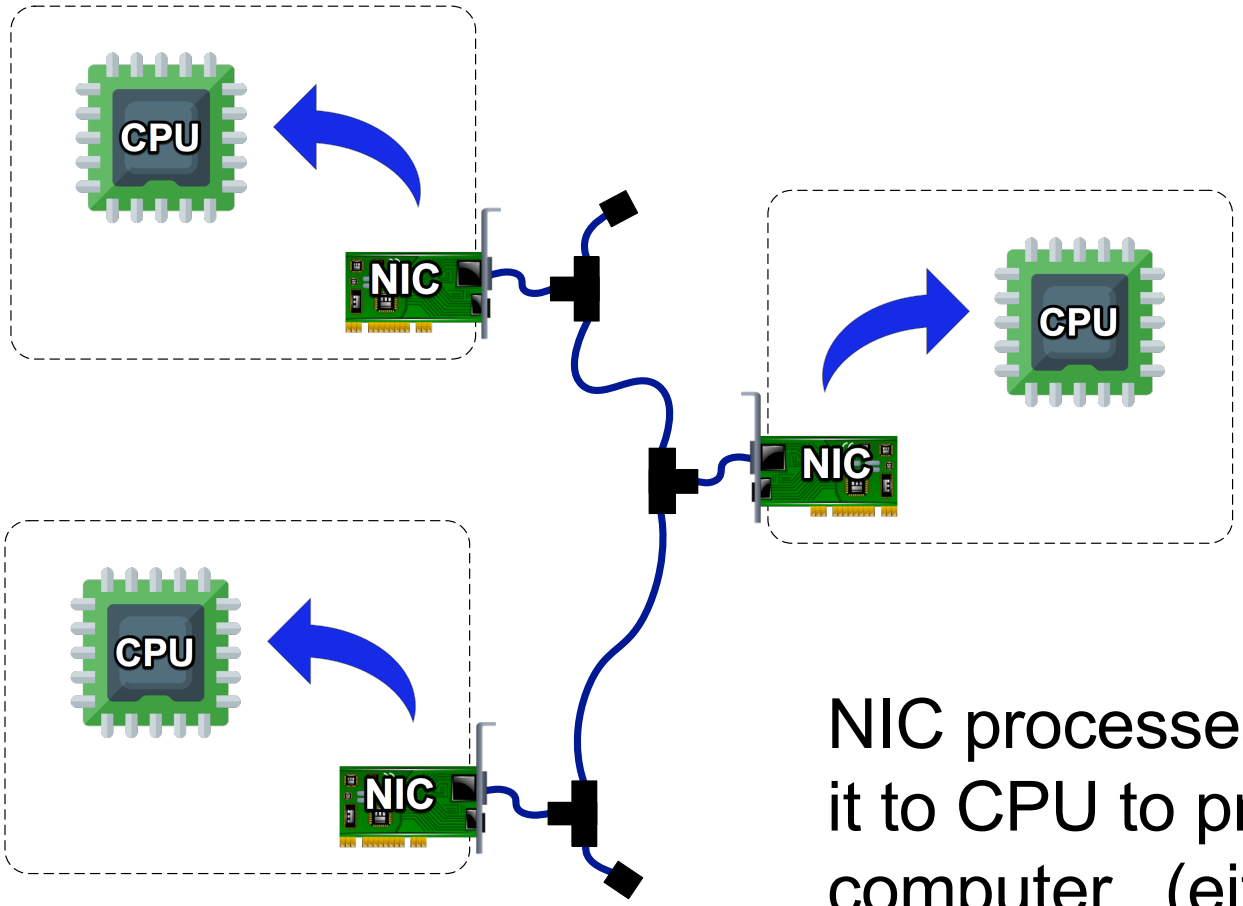


- 10 Base 2 (thinnet – cheapnet)
- Everyone sees all the traffic
- Need terminators at end to prevent echo's (bounceback).

Broadcast Technology

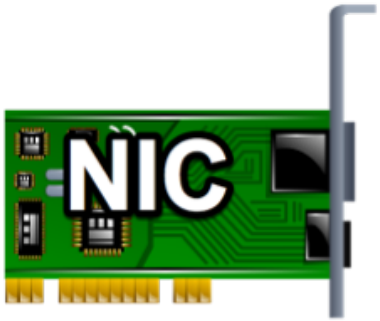
- If you want to say something, stick your head out and scream
- If you hear someone talk when you talk, you “backoff” and tray talking again in some random time later
- To reduce echo, you can put pillows at the end of the hallways. (pillows = terminators)
- But if the pillows go away, your echo will prevent anyone from being able to talk.

CPU doesn't get data unless its for that system



NIC processes traffic & only sends it to CPU to process if it's for that computer. (either a broadcast, or the MAC of the machine)

The MAC Address



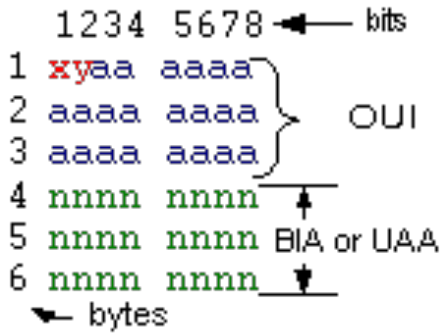
AA:BB:CC:DD:EE:FF

OUI

Organizationally
Unique
Identifier

- Every network card has its own, built in “address”.
- This address is assigned by the factory (you don’t do this)*
- There are two parts to the MAC address; the OUI, and the Manufacture assigned address.
- (*) There are two kinds of OUI address’s; Universal and Local. Universal are given out via IEEE, Local are randomly made up (*cheap and free* + *mess*).

IEEE Standards Association
IEEE OUI and Company_id Assignments:
http://standards.ieee.org/oui/oui.txt

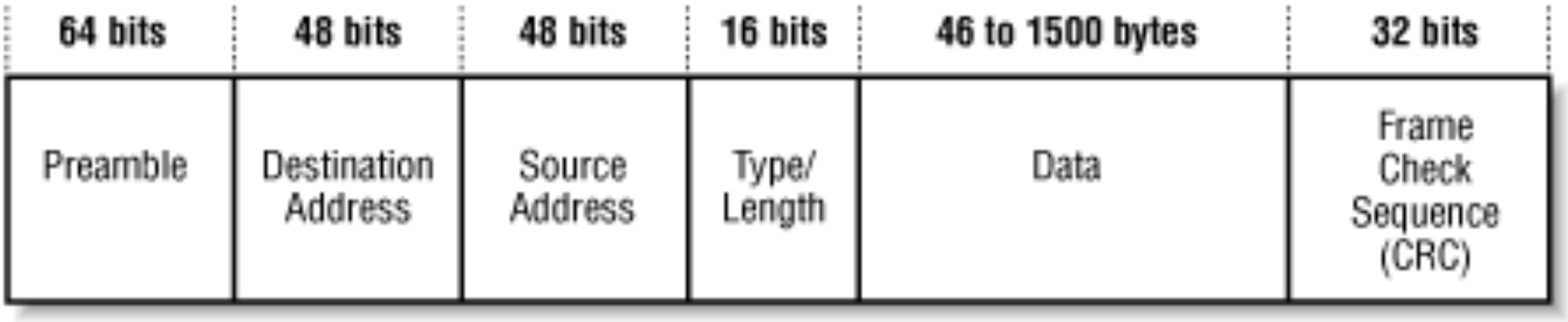


(‡ Ignoring Endianness)

X :
0 = physical
1= multicast

Y:
0 = registered
1 = “logical”

Ethernet

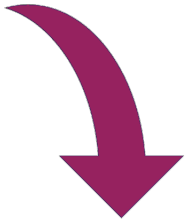

64 bits
48 bits
48 bits
16 bits
46 to 1500 bytes
32 bits
Preamble
Destination Address
Source Address
Type/ Length
Data
Frame Check Sequence (CRC)

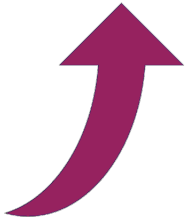
Message Addressing

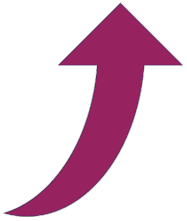
AA:AA:AA:11:11:11

BB:BB:BB:11:11:11

CC:CC:CC:11:11:11



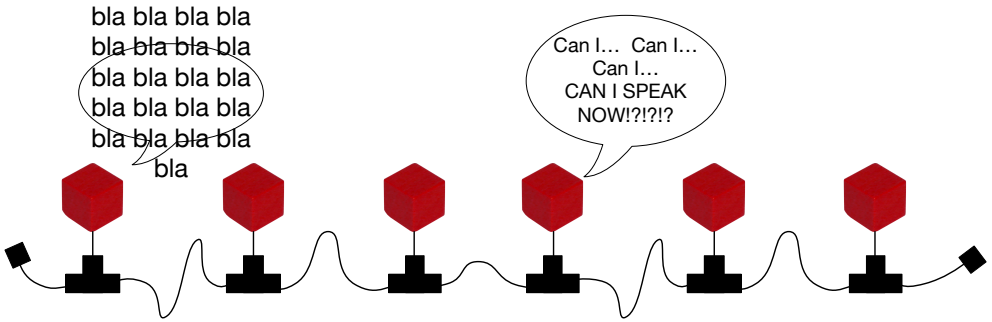




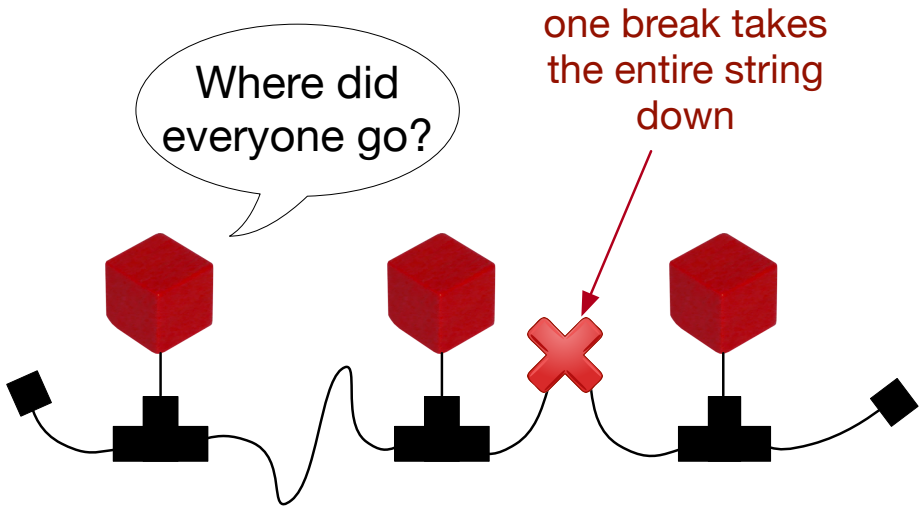
To: CC:CC:CC:11:11:11	From: AA:AA:AA:11:11:11	Data: Hi!

- Both ”B” and “C” get the message, but “B” ignores the message because it’s not for him, and “C” reads it
- If ”A” wants to see who all is there, it can send to [FF:FF:FF:FF:FF:FF] (Broadcast) and request everyone to respond back.

Problems



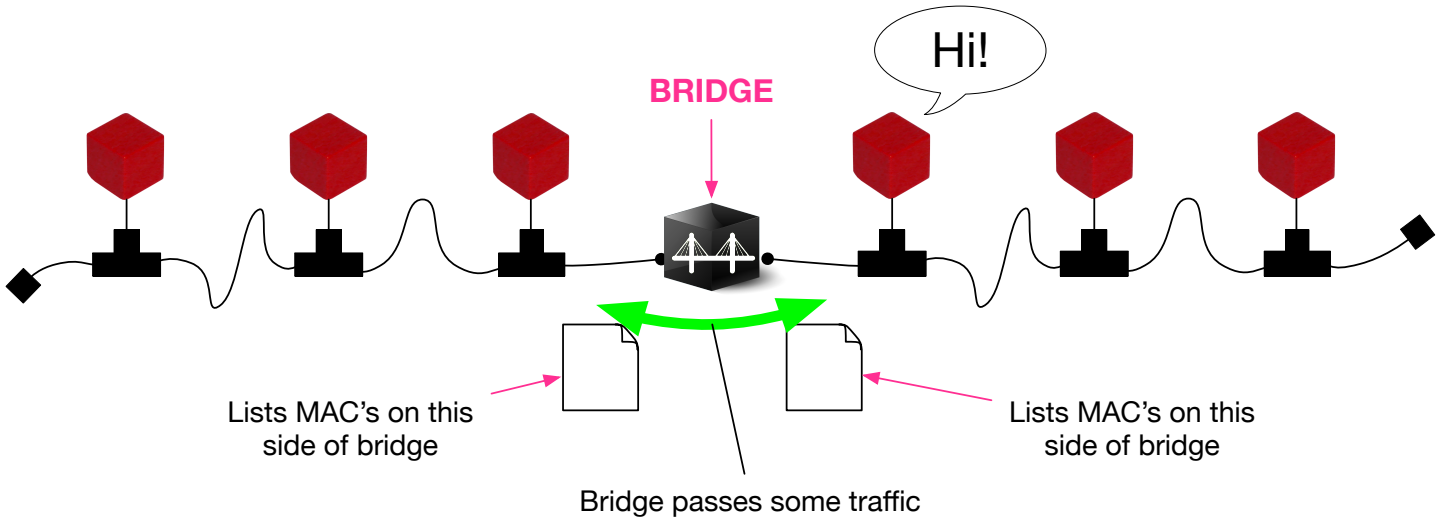
Contention: Too much chatter on the line will prevent others from speaking.



Faults: A single fault in the cable will prevent <u>everyone</u> from being able to talk.

What can tech do to fix this?

Bridge

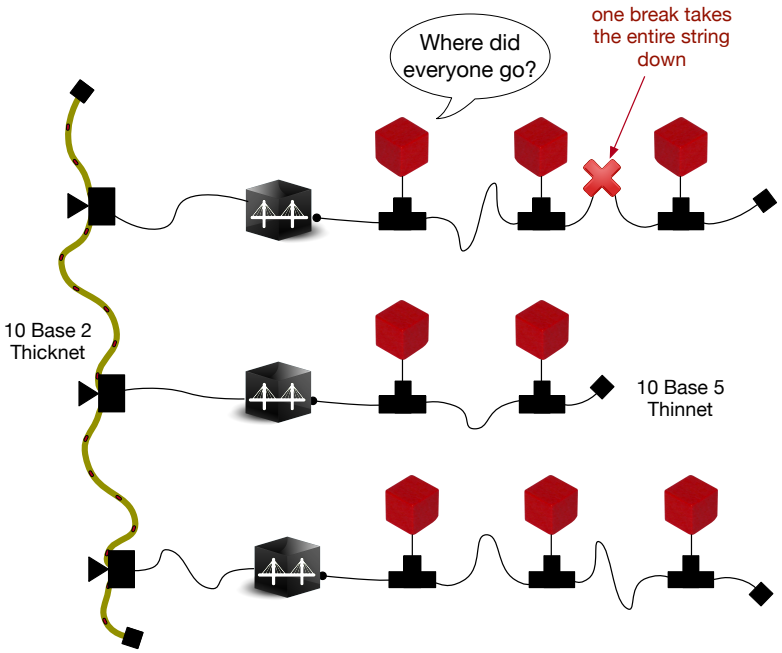


Bridge Passes:

- Broadcast Messages (to: FF:FF:FF:FF:FF:FF)
- Messages it knows are for the other side
- Messages it does not know the location of the receiver

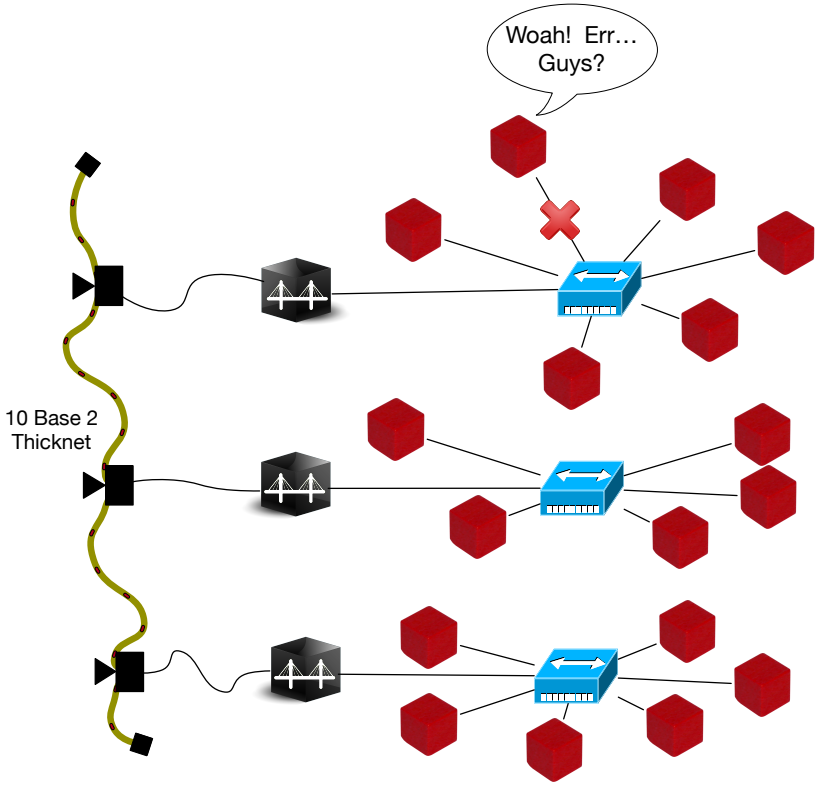
Hub

Old Badness
Chained Coax



We could limit network outages by using a lot of bridges, but if a string is damaged, all systems on it are off line.

"New" Hotness
Twisted Pair to a repeater



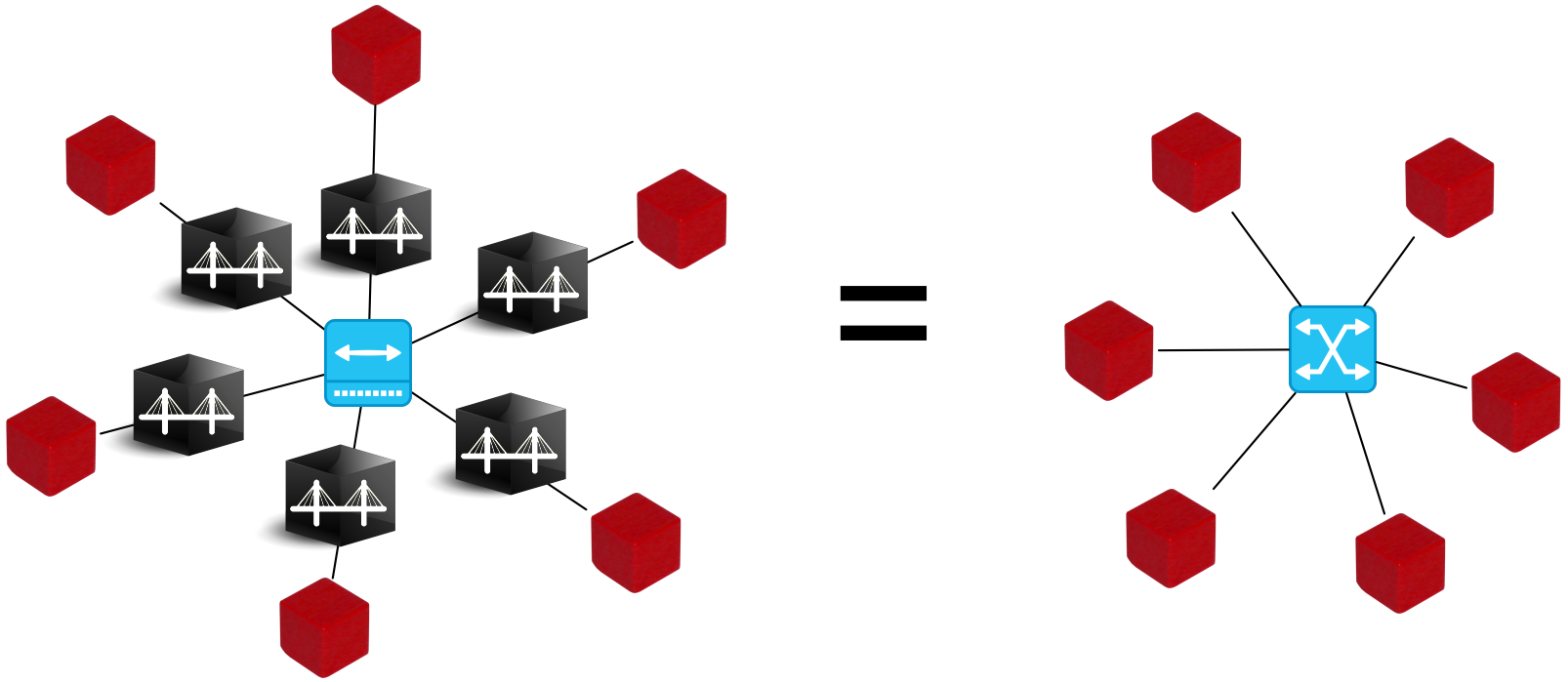
Replace the strings with a hub (just a repeater). One cable issue only takes out the host connected to it.

Bridges got so popular, that their pricing got really cheap, and they started making them with multiple ports!

Switch

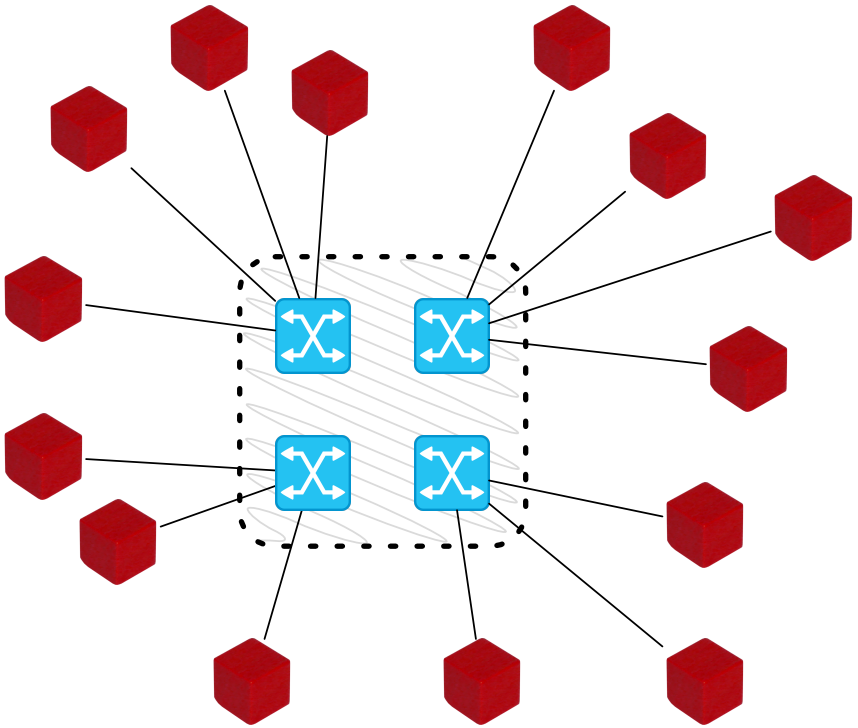
Hub with bridges at each port

Switch



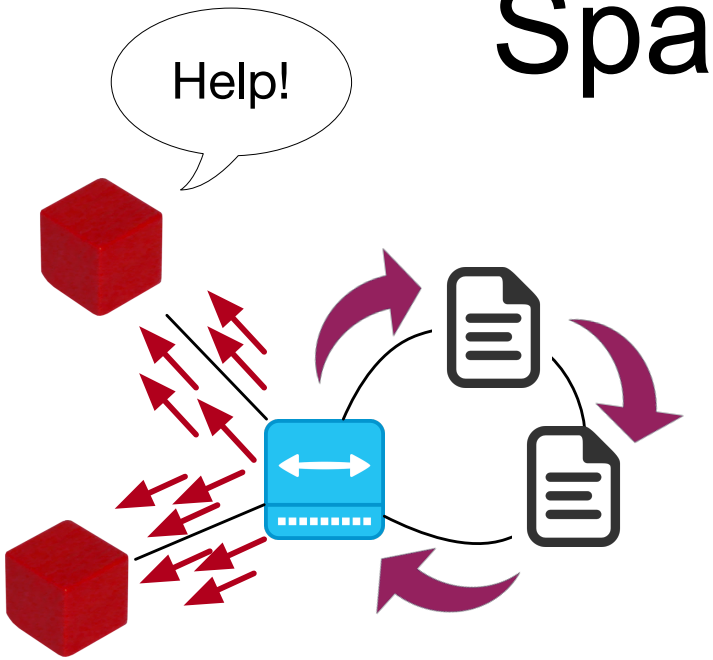
- A switch is basically a multi port bridge (or a hub with a bridge at each port).

VLANs

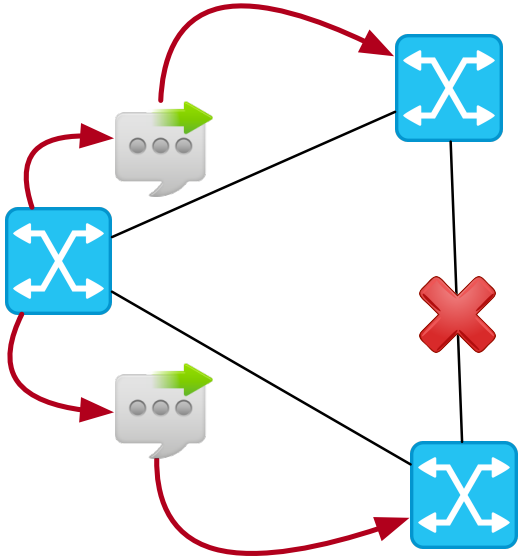


- Can break one switch into a bunch of “virtual” switches
- Each switch (vlan) cannot talk to each other

Spanning Tree



- Loops are ~*very*~ bad!!!
- Every time a broadcast packet comes into a port, it gets sent to all other ports
- They can be weird, messy errors that are hard to troubleshoot



Spanning Tree Details:

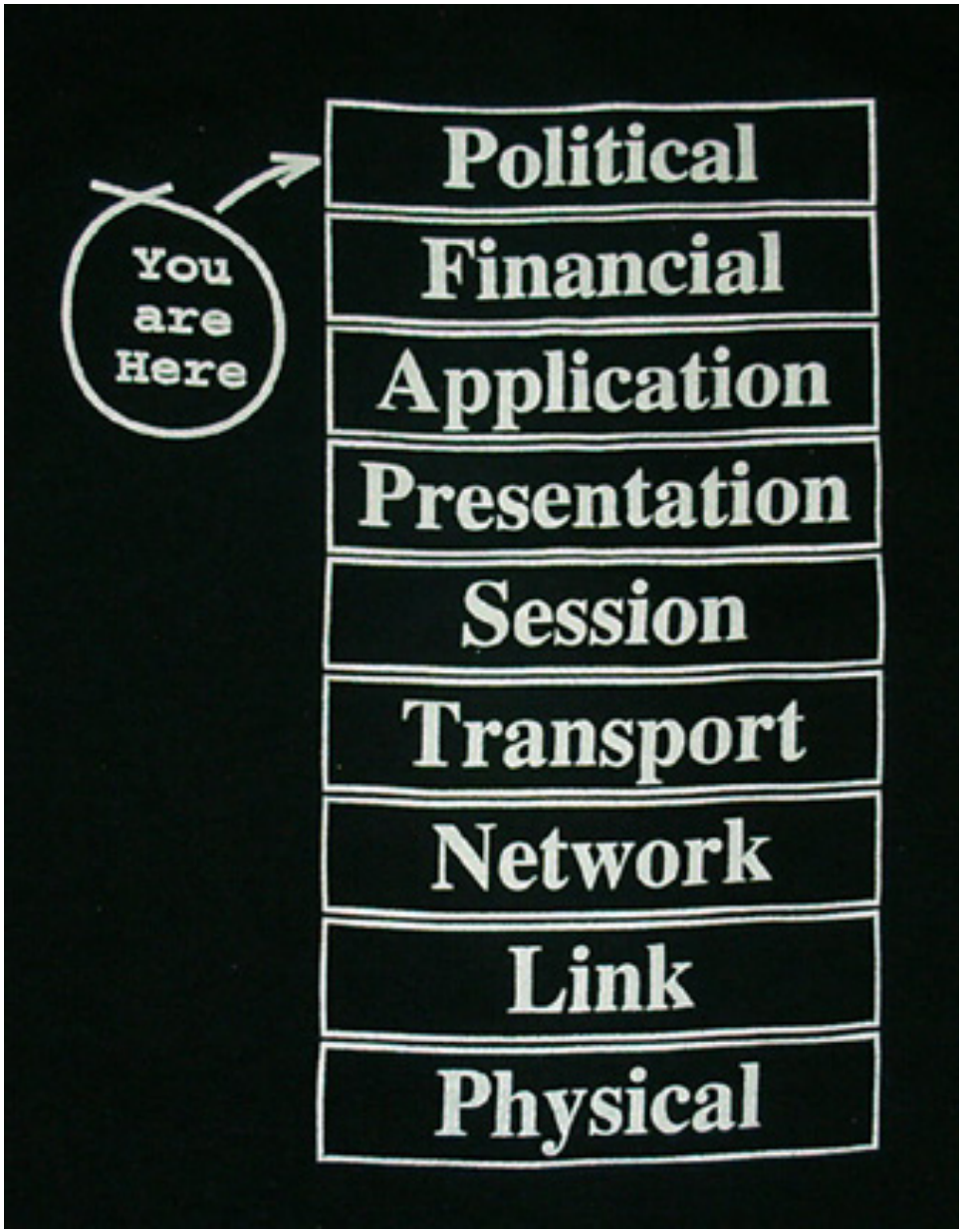
- BPDU Messages
- Blocking, Listening, Learning, Forwarding (50sec)
- Port Guard Ports

Ethernet Summery

- Broadcast
- Data is analyzed at the NIC
- MAC address (OUI)
- Switch, VLAN, Spanning Tree

Layer 3 = IP

9+ Layer (cough) OSI Model

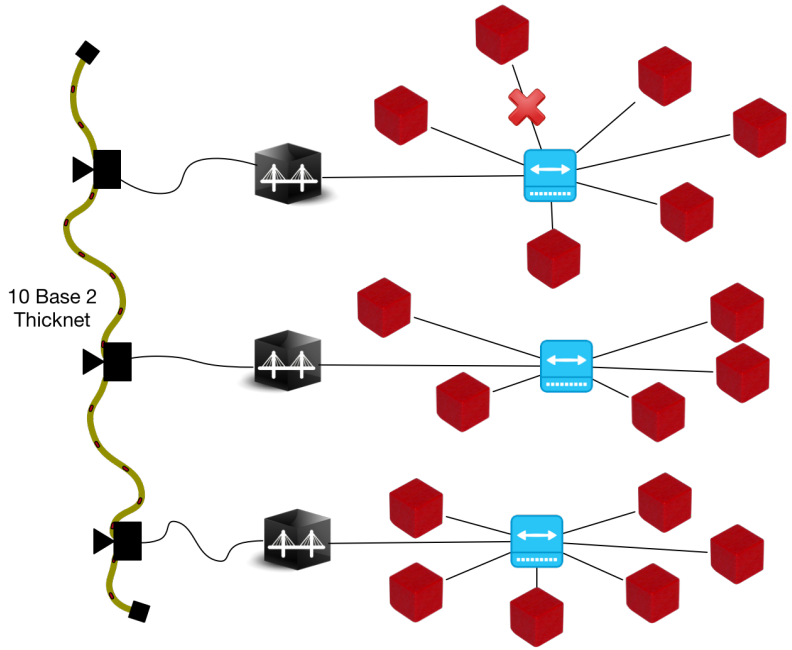


3

IP: Packets and Messaging

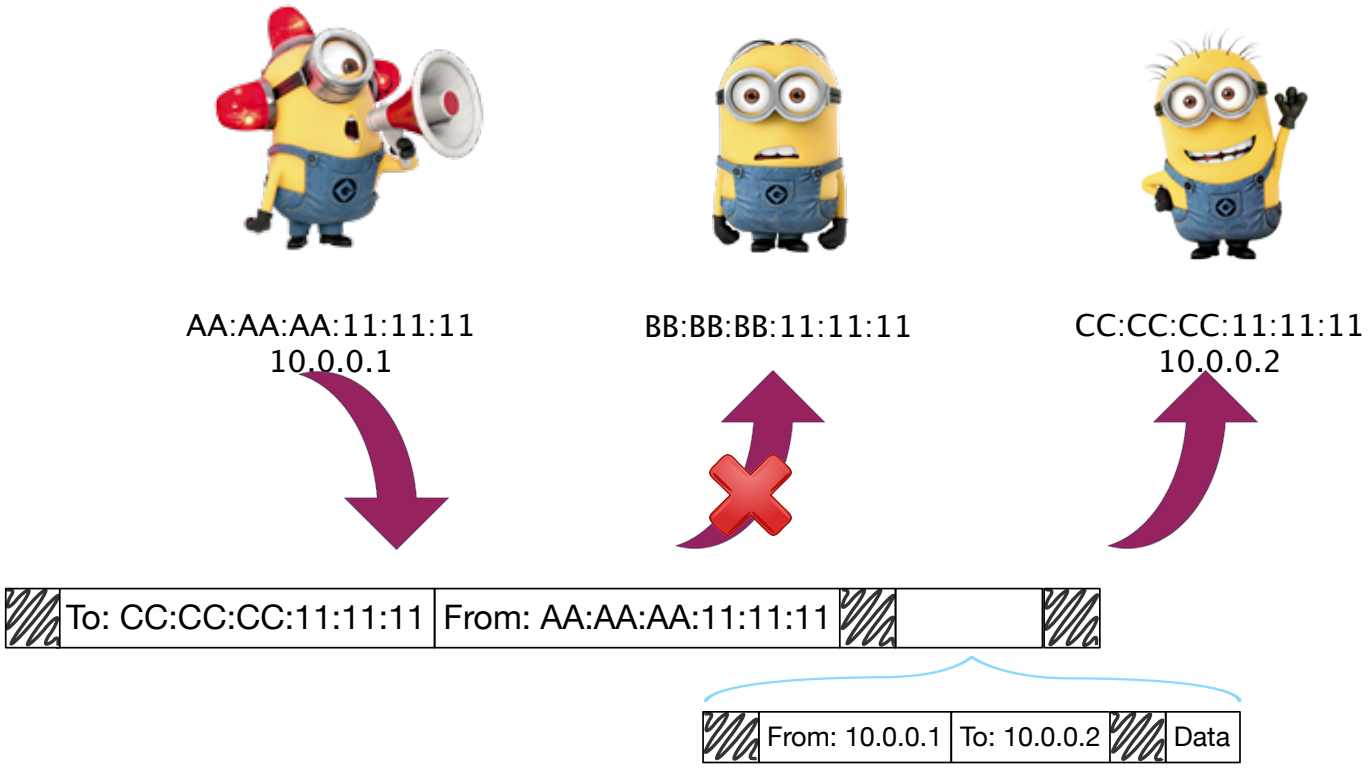
- How packets are sent around and what the packet looks like.

Ethernet Review (challenges)



- Need to know the MAC what you want to talk to.
- Broadcasts limit total # of hosts on network
- Other then a spanning tree blocked loop, no way to route around issues

IP packets in Ethernet



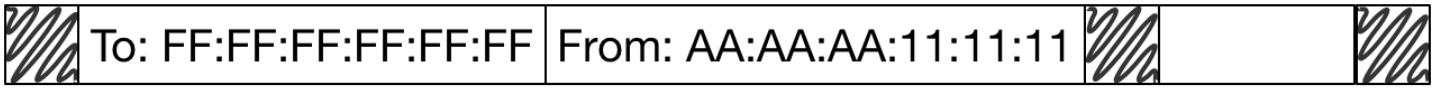
- IP fits within the Ethernet packet
- Provides an additional level of abstraction

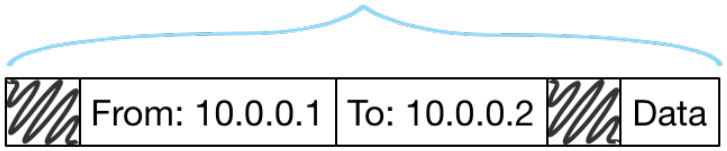
If you know IP, but not MAC

AA:AA:AA:11:11:11
10.0.0.1

BB:BB:BB:11:11:11

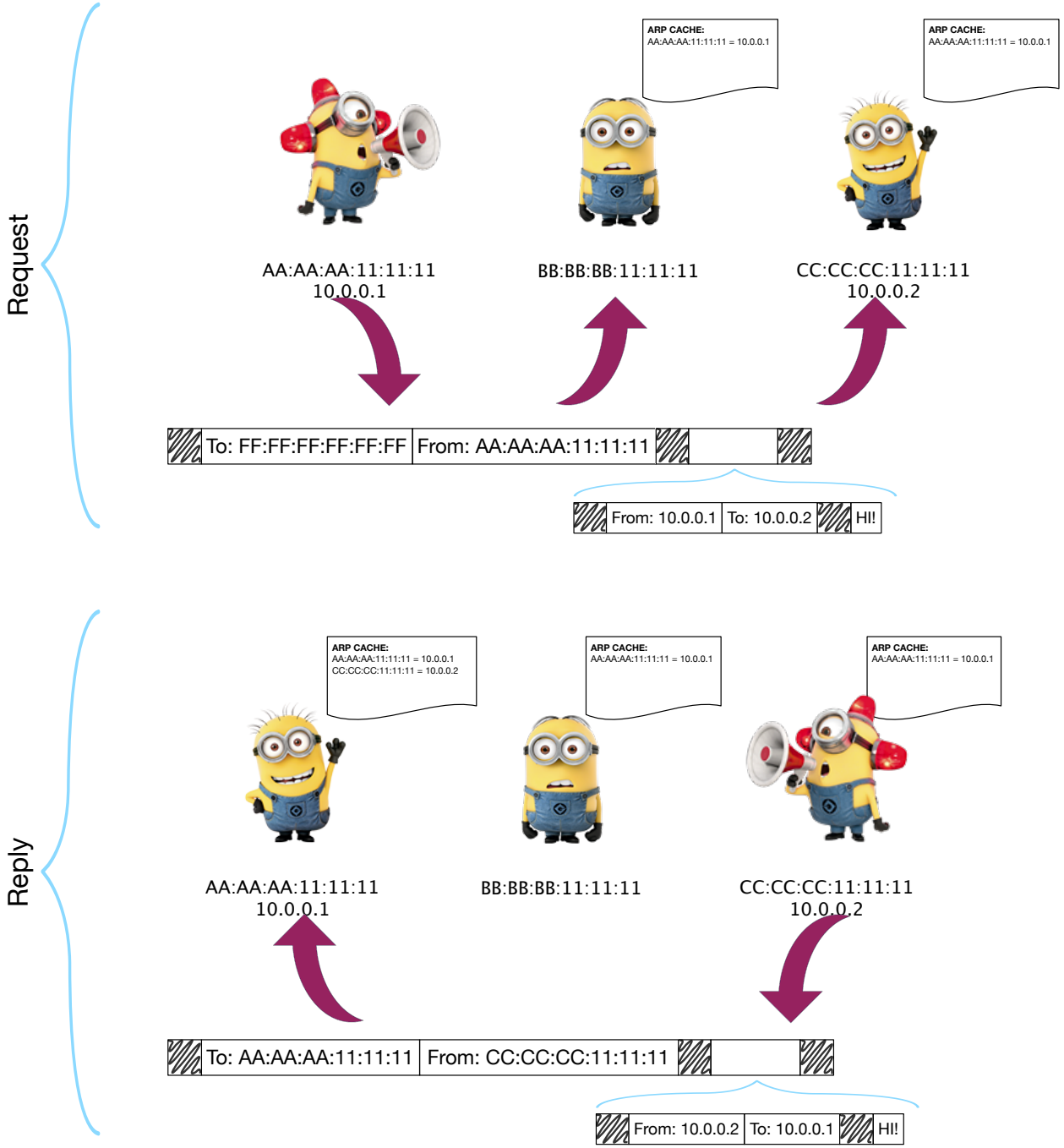
CC:CC:CC:11:11:11
10.0.0.2





- BROADCAST! Can send to everyone, and ask if they are the correct destination.

ARP Cache

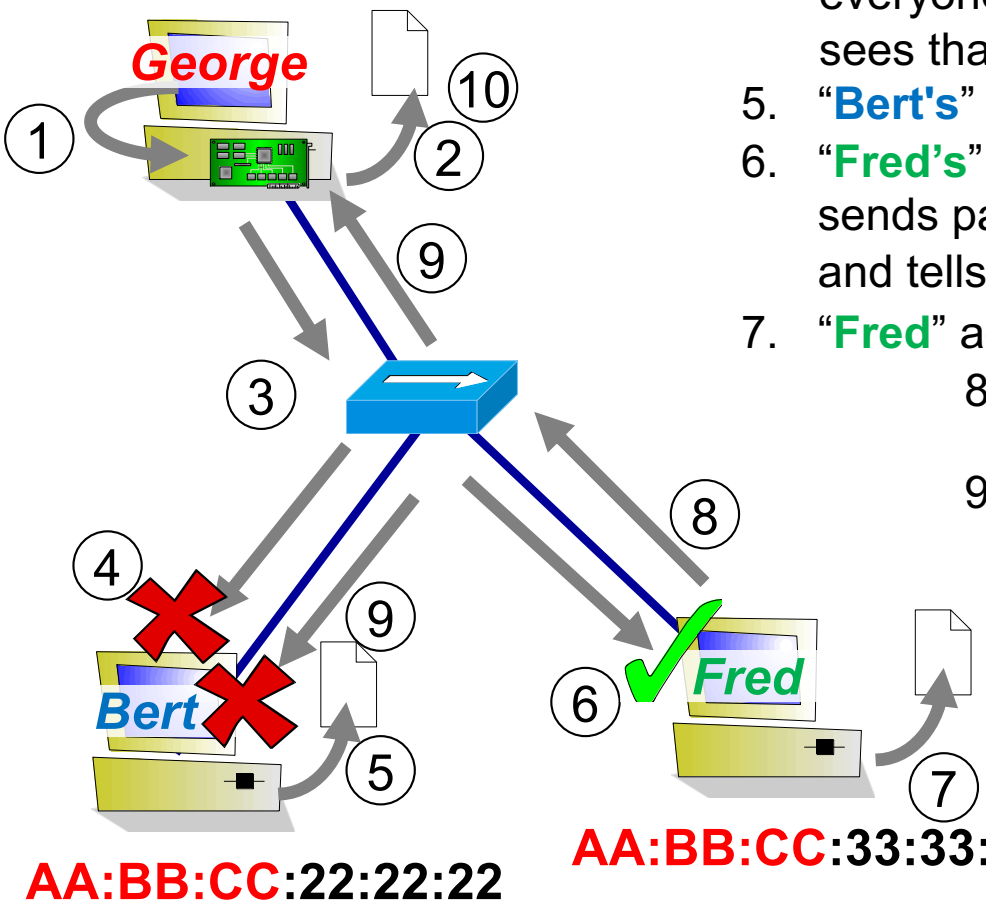


- Tracks MAC / IP combo in a local database

A Layer 3 Conversation

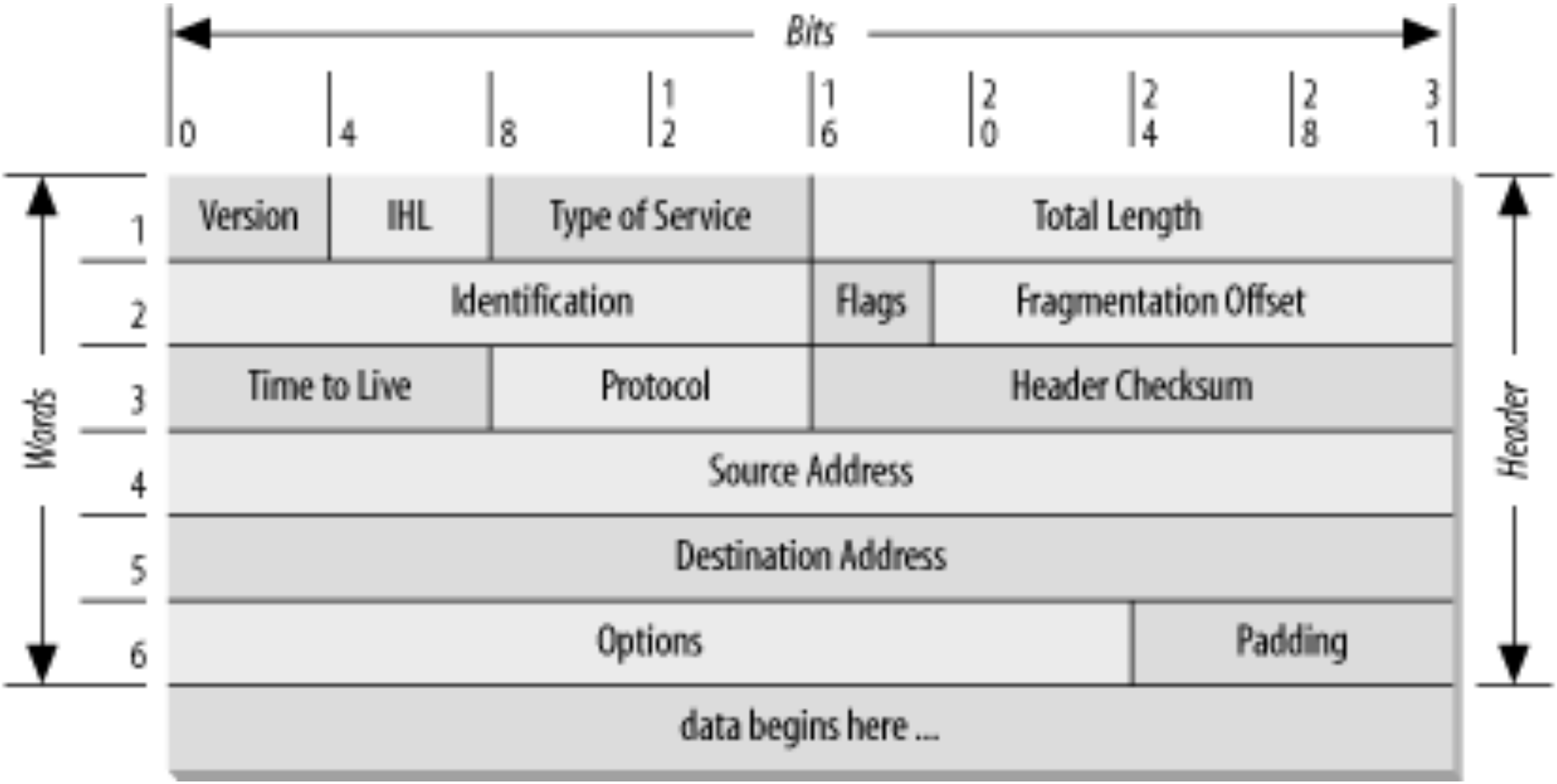
('George' " Hi " to 'Fred', 'Fred' says Hi back.)

(Names are alias's to IPs)



1. "George" says Hi to "Fred"
2. Do I know "Fred's" MAC? (no)
3. "George" sends to everyone's MAC (in payload, say looking for Fred)
4. "Bert's" NIC gets the message. Since it's "to" everyone, sends payload to the computer. "Bert" sees that it's for "Fred" so it ignores the data.
5. "Bert's" adds "George's" MAC to it's ARP Cache
6. "Fred's" NIC gets the message. Since it's "to" everyone, sends payload to computer. "Fred" sees the message, and tells NIC to says Hi back.
7. "Fred" adds "George's" MAC to it's ARP Cache
8. "Fred's" NIC sends message to "George's" MAC only, saying "Hi".
9. Message is received by "Bert's" NIC, and ignored
10. Message is received by "George" NIC and it's given to "George", and "Bert's" MAC is added to "George's" ARP Cache.

IP (internet protocol)



IP: Addressing

- Network IP
- Host IP
- Mask
- Binary vs. Decimal
- Classful and Classless Networks

Everything is binary

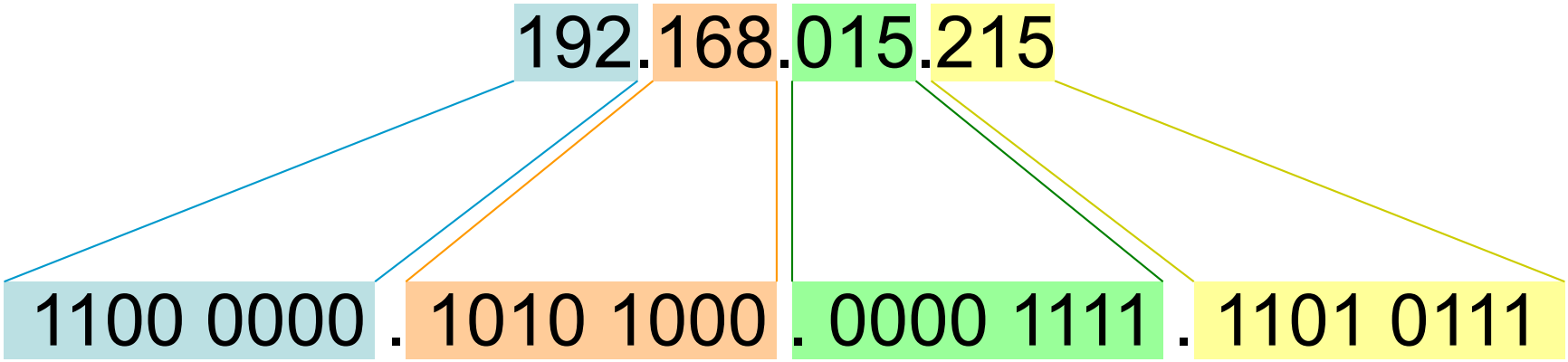
- We (humans) like numbers to be in Decimal; (Base 10 = B_{10}) “set {0123456789}”.
- Computers ONLY deal with numbers in binary; (Base 2 = B_2) “set {01}”. To make life easier for you, they will “talk” decimal for you, but their logic is based on binary.

B_{10}	B_2
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010

$$2^7\ 2^6\ 2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0$$

$$0011\ 1010 = 2^5+2^4+2^3+2^1 = 32+16+8+2 = 58$$

The IP Address



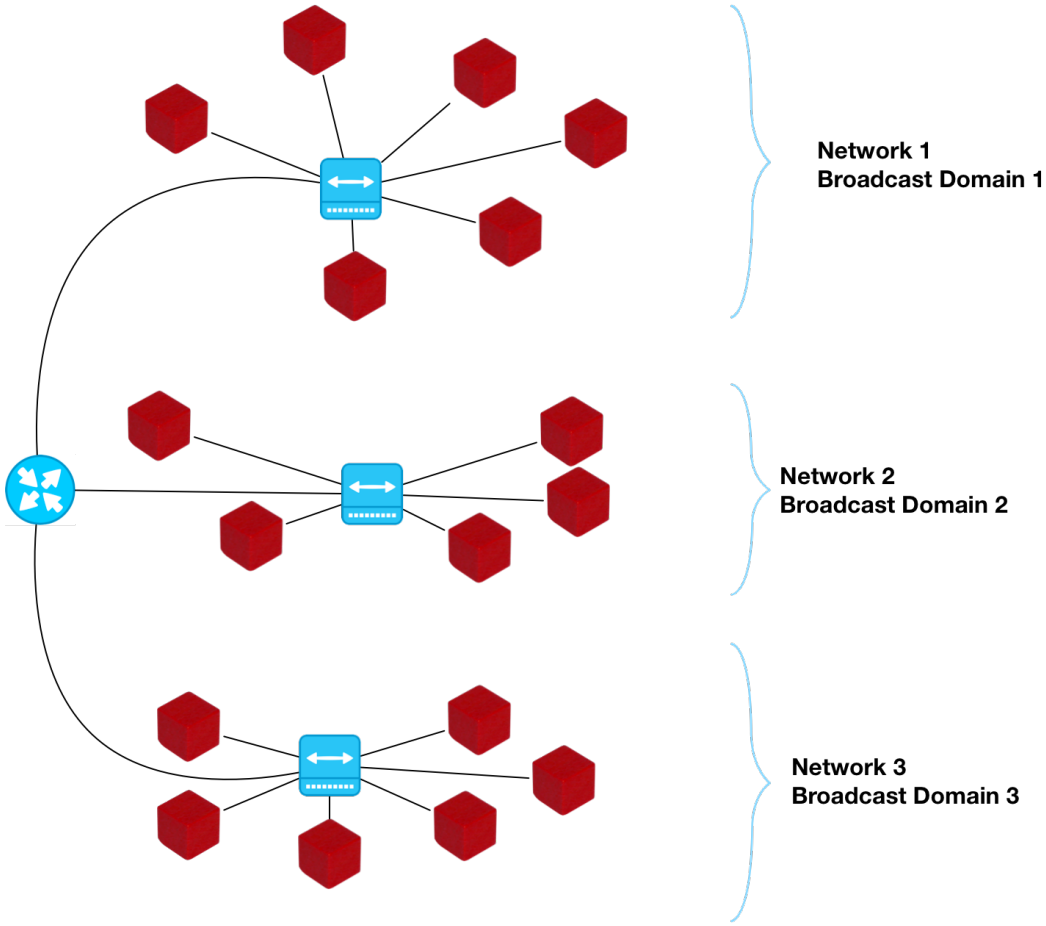
It’s really a binary number!

What is “The Local Network”?

IP sits on top of Ethernet (which is a broadcast technology).

Because of this, IP expects separate “networks” or “broadcast domains”.

Understanding the size of your “local network” is a critical piece of working in an IP network.

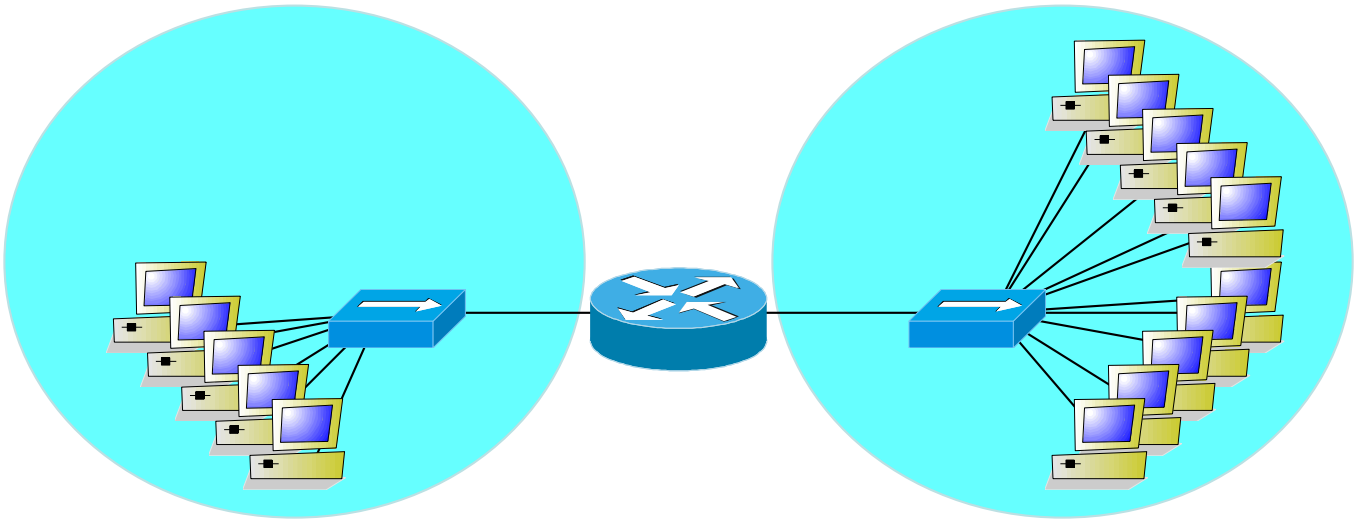


Natural Masks

192.168.015.215

1100 0000 . 1010 1000 . 0000 1111 . 1101 0111

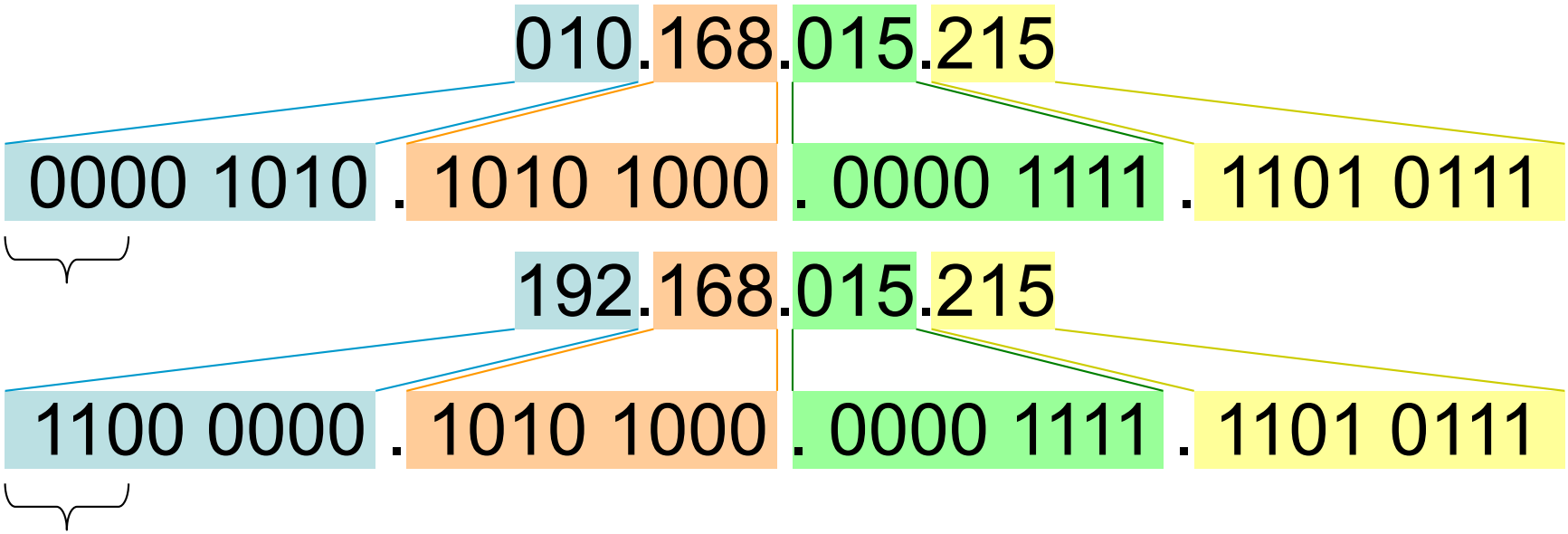
These three bits define how big the network can be!



Small Network

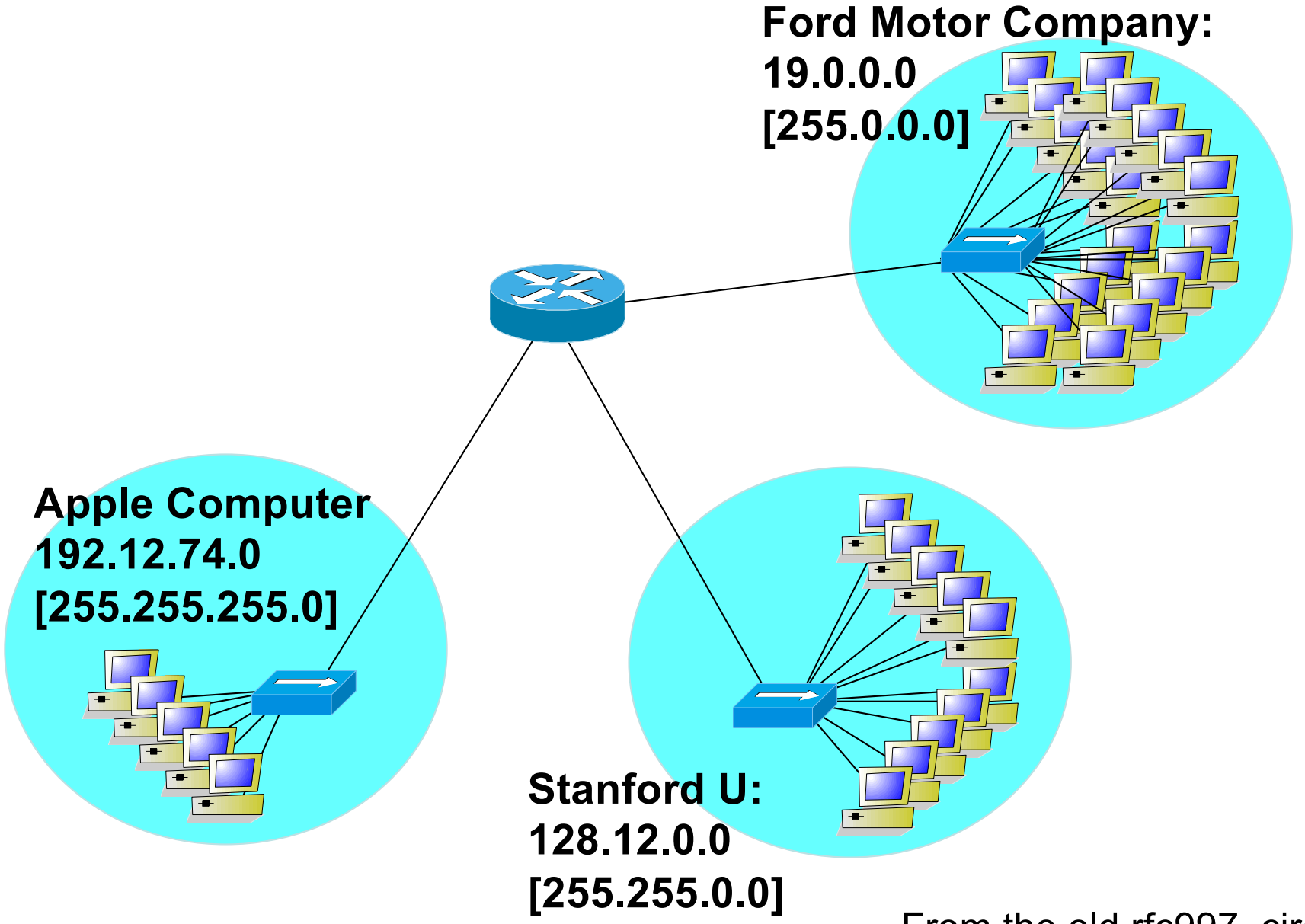
Big Network

Natural Masks



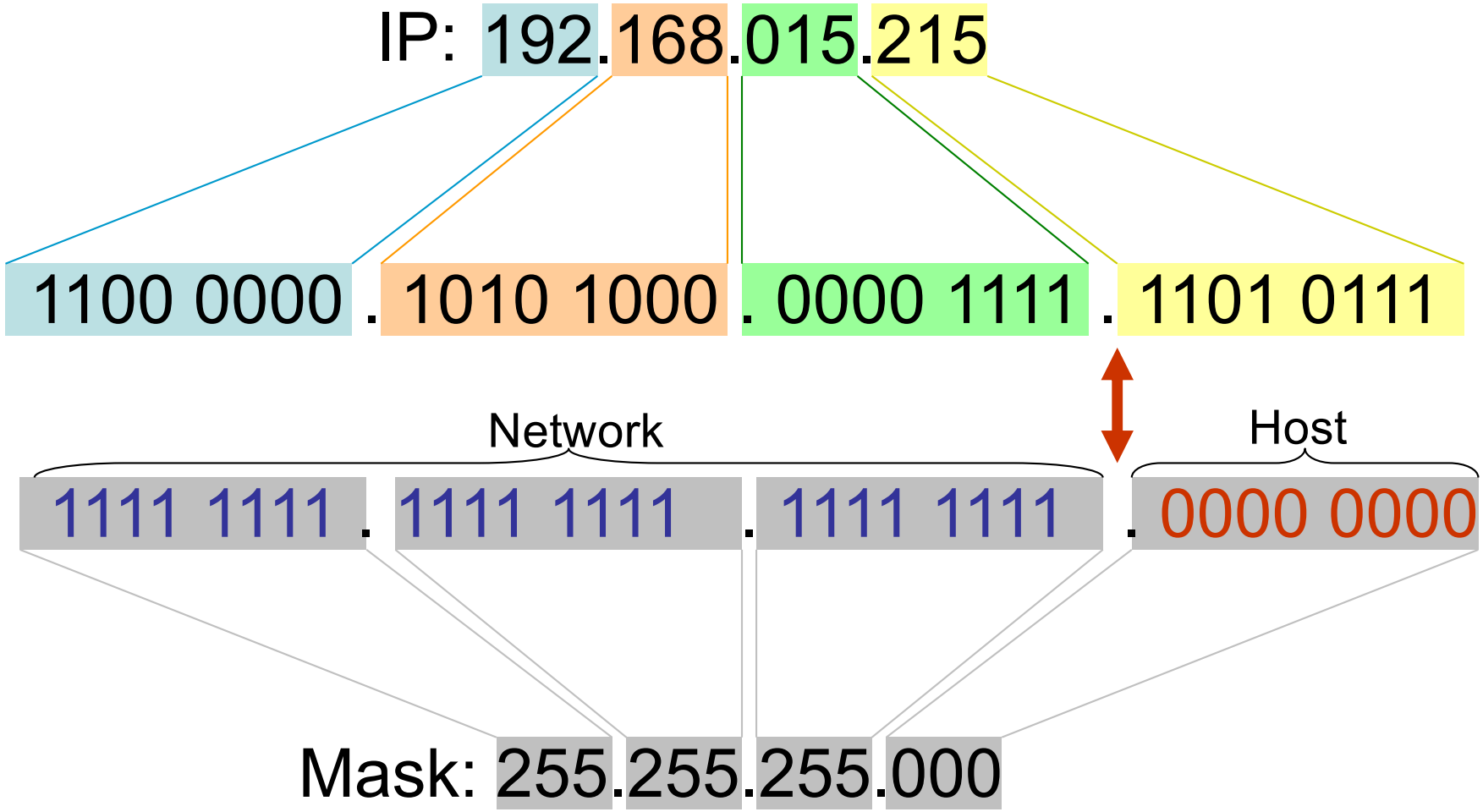

```
0*** = Class A (first octet = net)
       range = 0.0.0.0 - 127.255.255.255 (* 127)

10** = Class B (first two octets = net)
       range = 128.0.0.0 - 191.255.255.255

110* = Class C (first three octets = net)
       range = 192.0.0.0 - 223.255.255.255

1110 = Multicast Address
```

Example of Classfull Networks



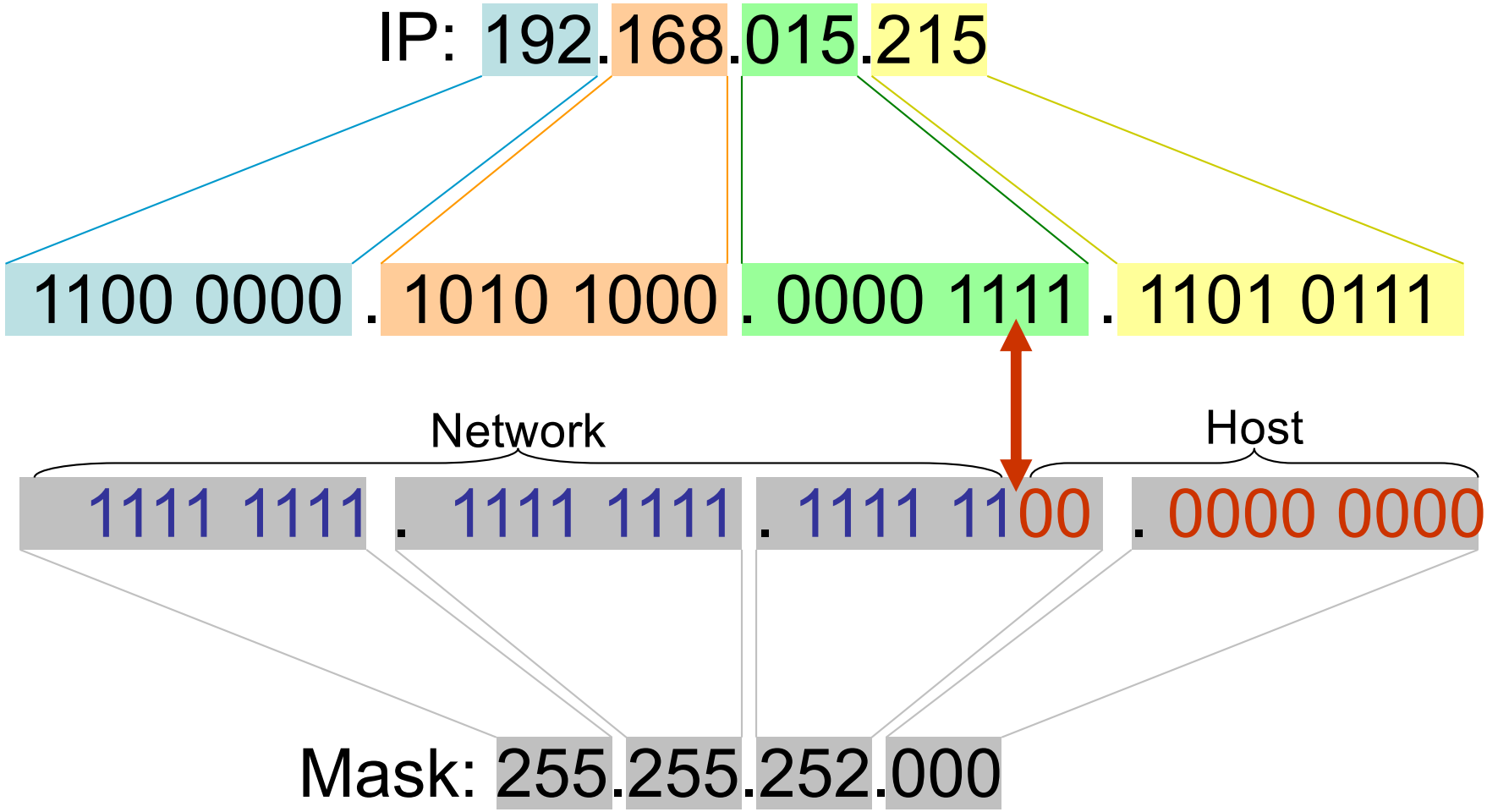
From the old rfc997, circa 1987 ☺

Masks (looking at them)

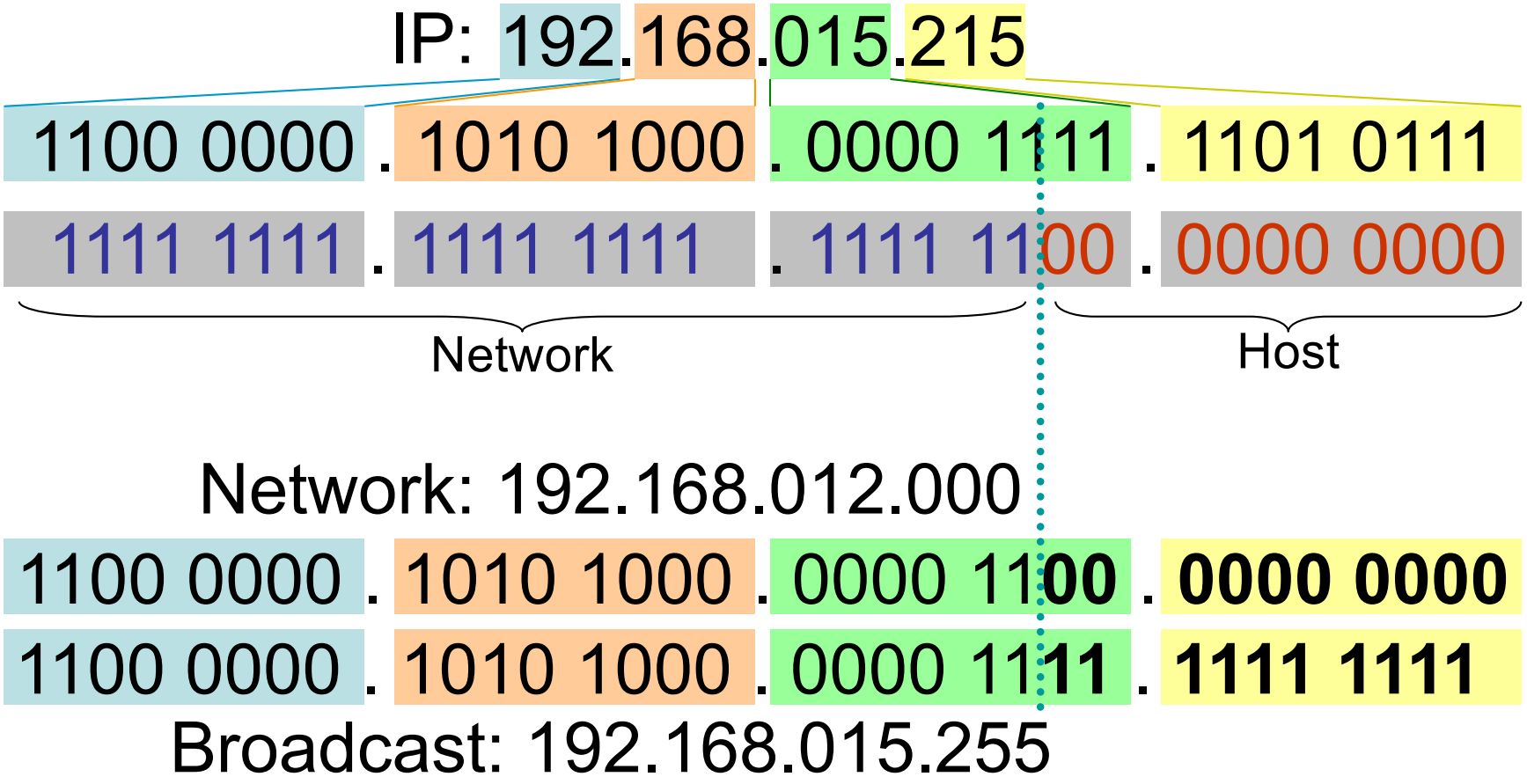


In the Mask, what's important is where the one's stop and the zero's continue.

CIDR Masks (looking at them)



CIDR Masks (looking at them)

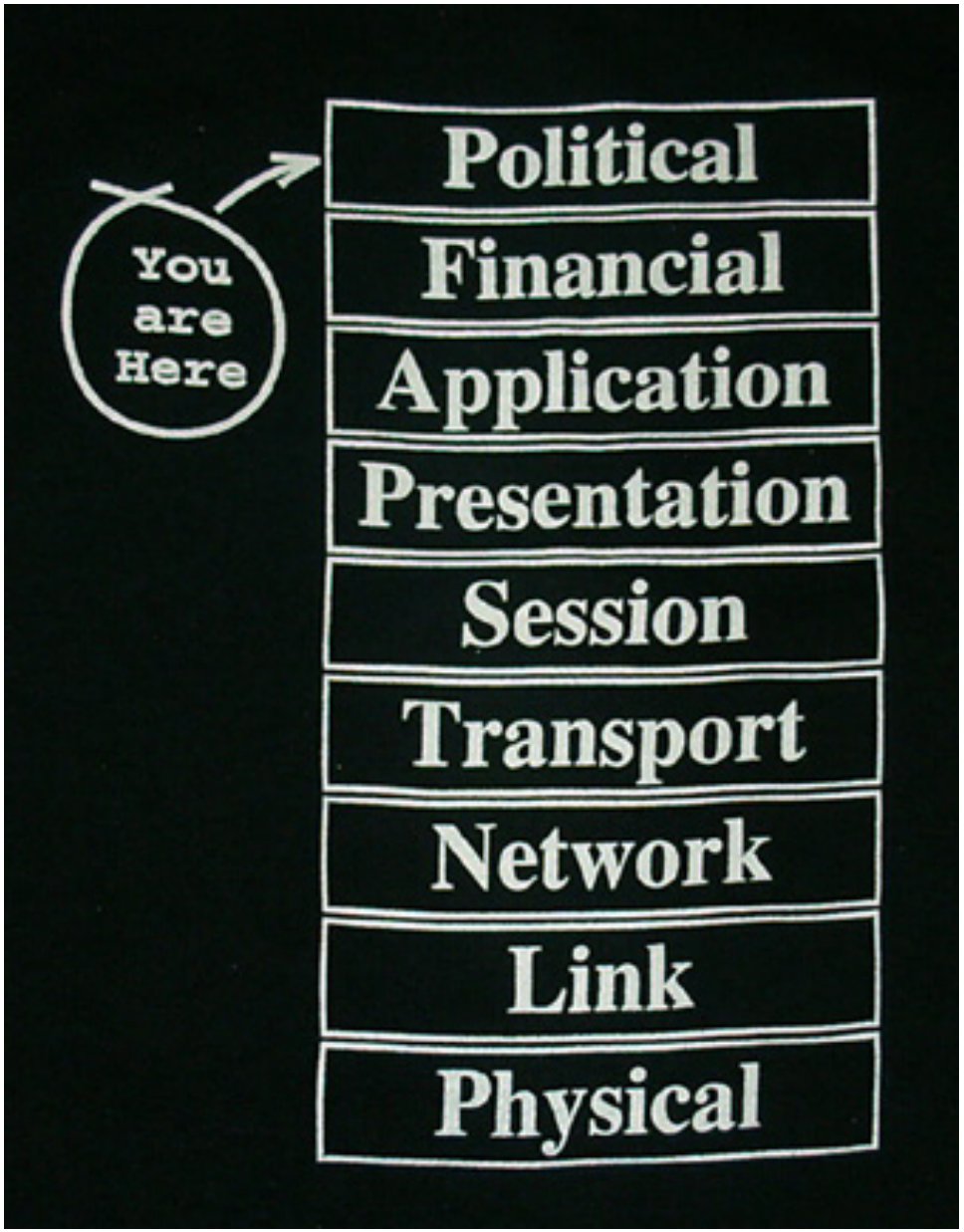


IP Summery

- Encapsulated within Ethernet packet
- It only makes sense in Binary
- Natural Masks
- CIDR Masks

Layer 4 = Transport

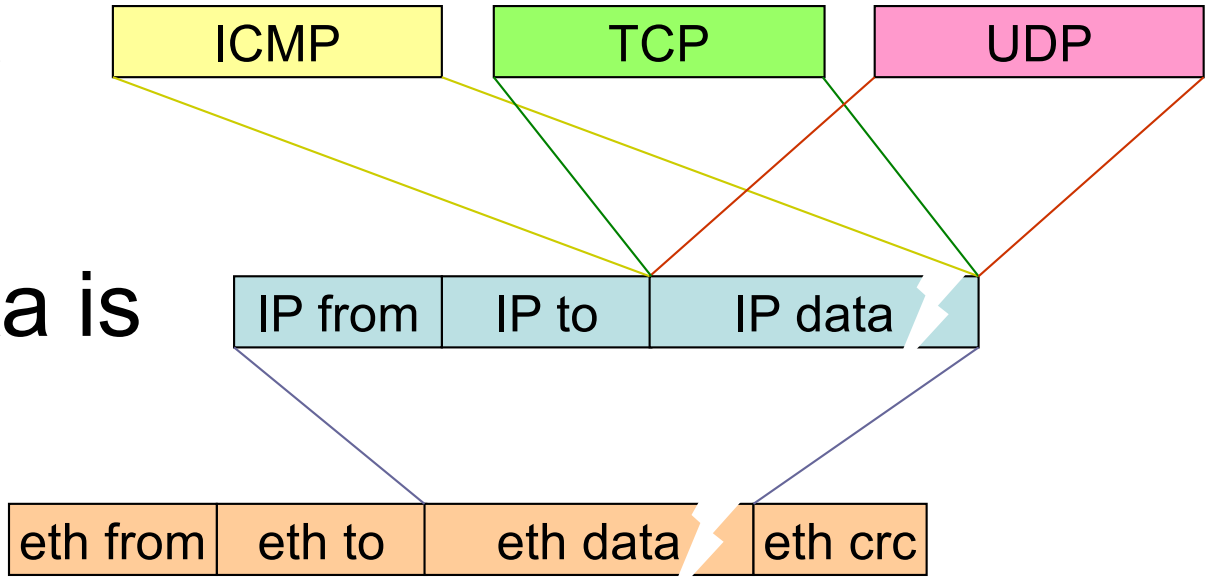
9+ Layer (cough) OSI Model



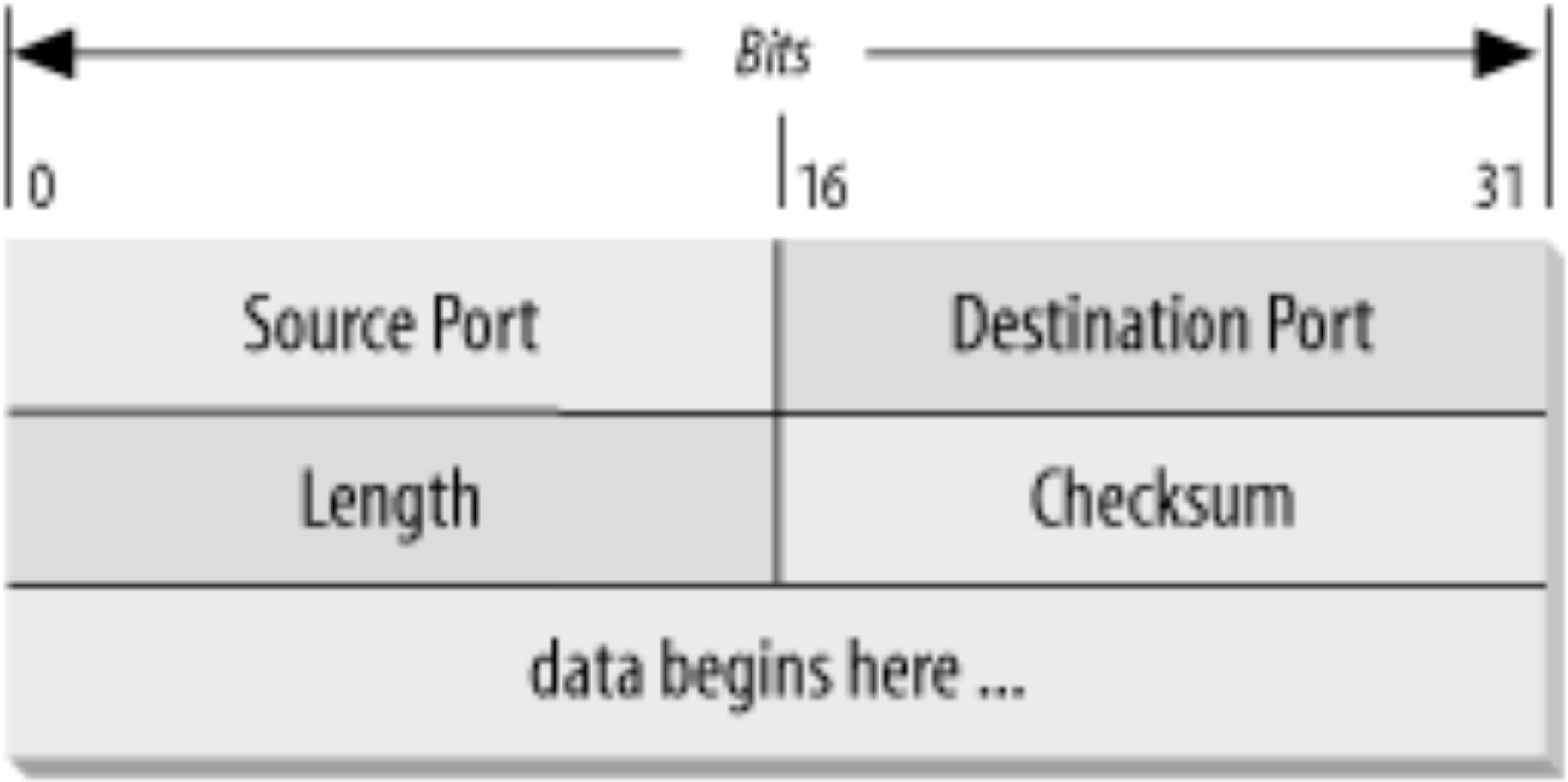
4

The 4th Layer:

- ICMP is a control protocol
- TCP guarantees data transport
- UDP no guarantee data is received



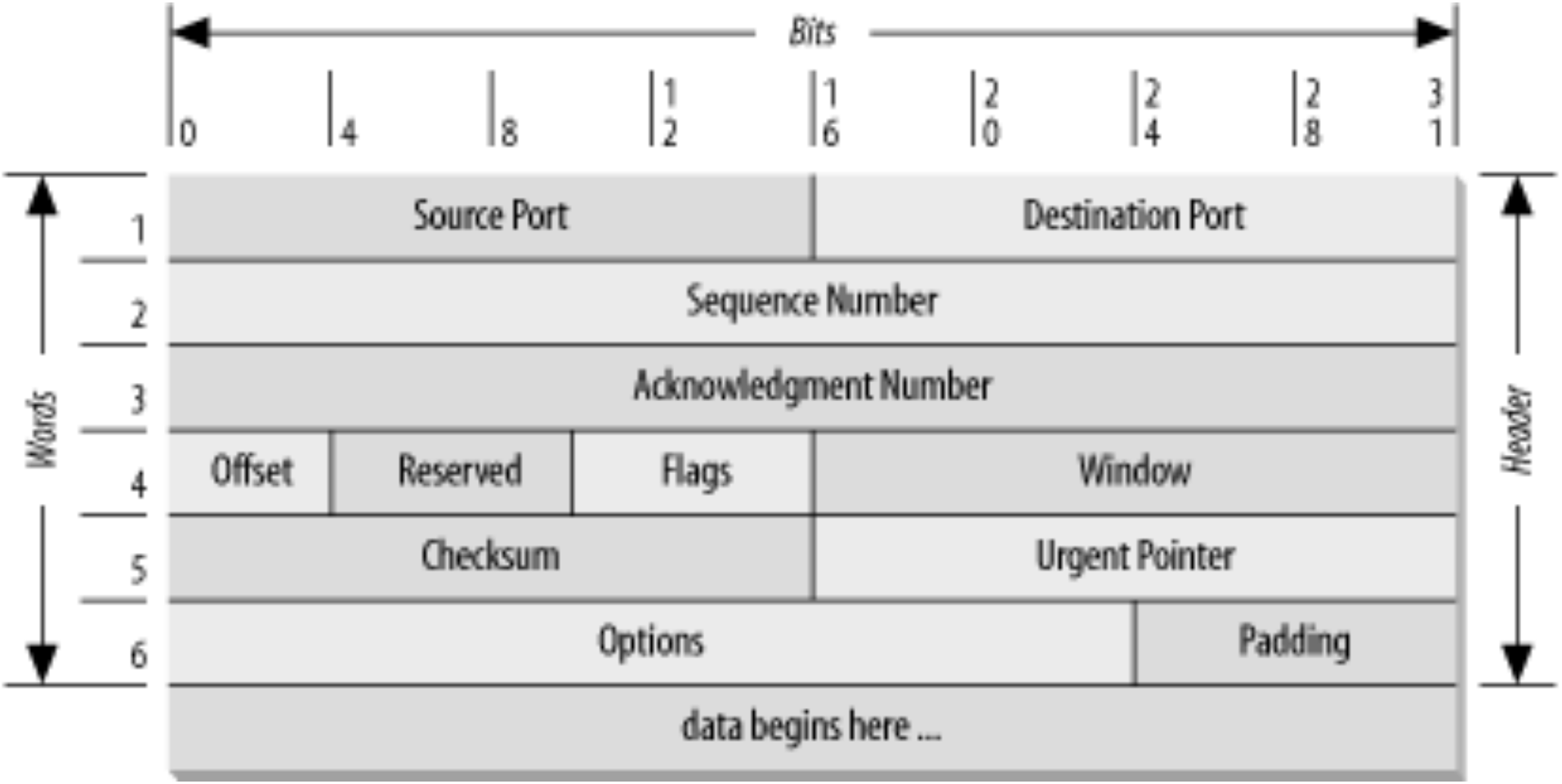
UDP (ip type 17)



UDP

- “Connectionless”
- Very basic way of sending data.
- Header really only includes ports
- No guarantees that data is received or without errors. (higher level apps are expected to handle this if necessary.)
- Fast (low overhead)
- Good for video/audio streaming

TCP (ip type 6)

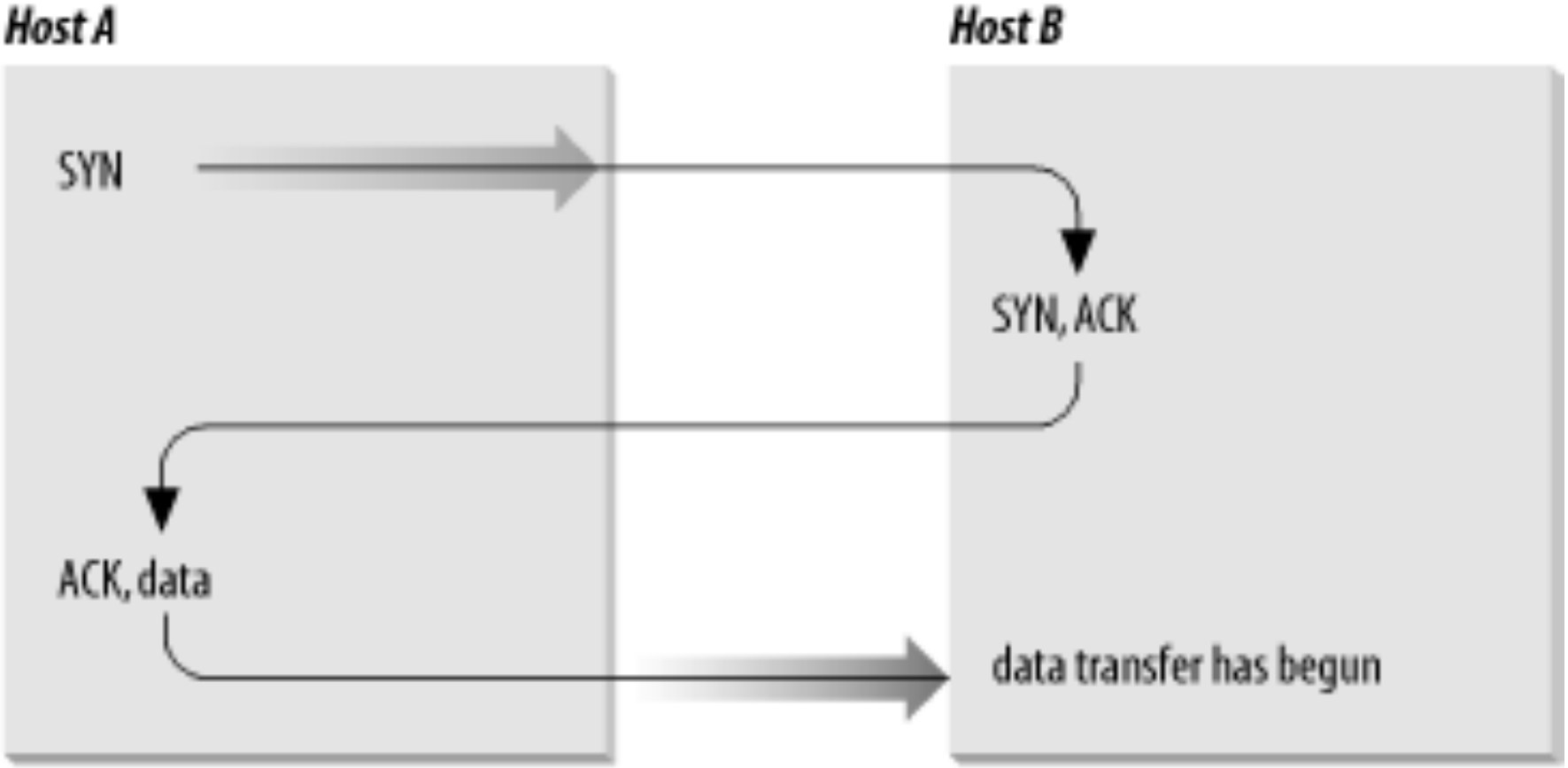

Bits
0
4
8
12
16
20
24
28
31
Words
1
2
3
4
5
6
Source Port
Destination Port
Sequence Number
Acknowledgment Number
Offset
Reserved
Flags
Window
Checksum
Urgent Pointer
Options
Padding
data begins here ....
Header

TCP

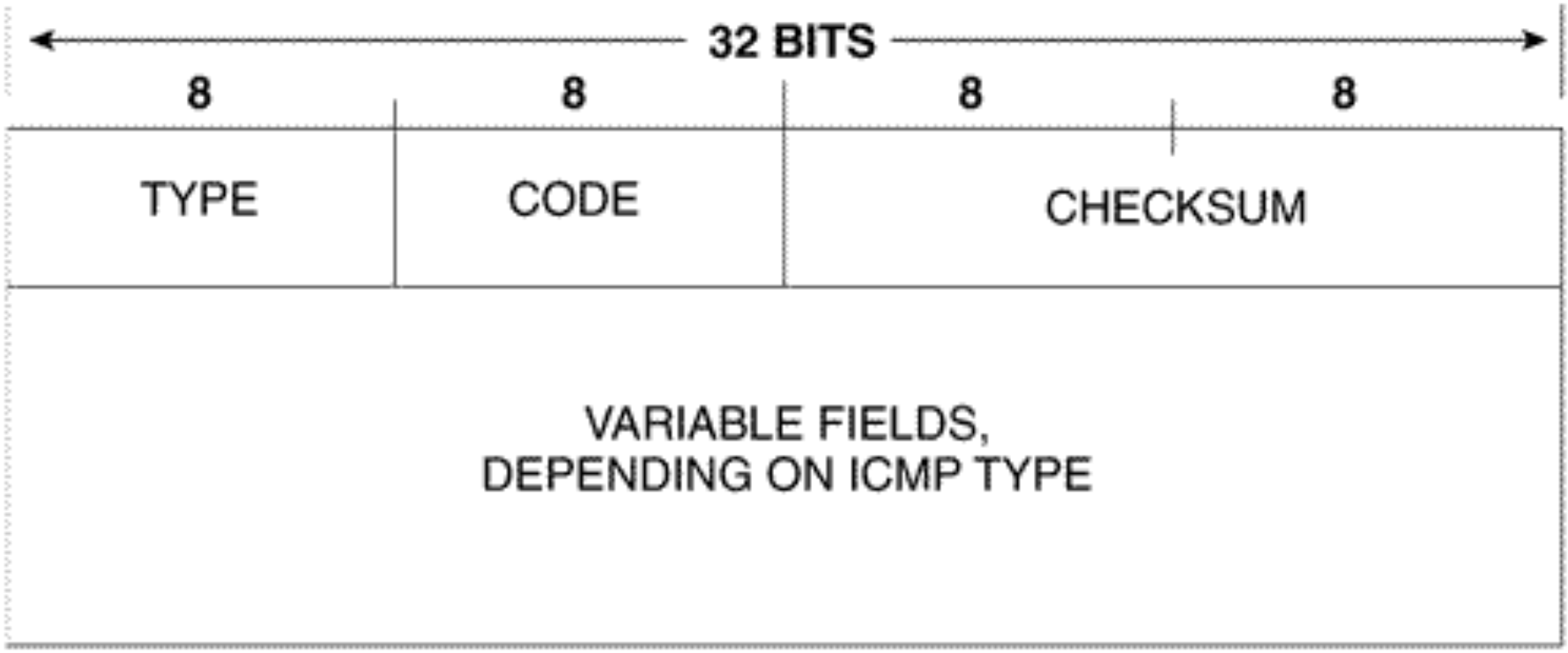
- “Connection Based”
- Protocol makes sure packet was received and that it is correct. - Will resend if there are errors
- Slow (lots of overhead)
- Good for ftp, http, and other file transfers

TCP

Establishing a connection



ICMP (ip type 1)

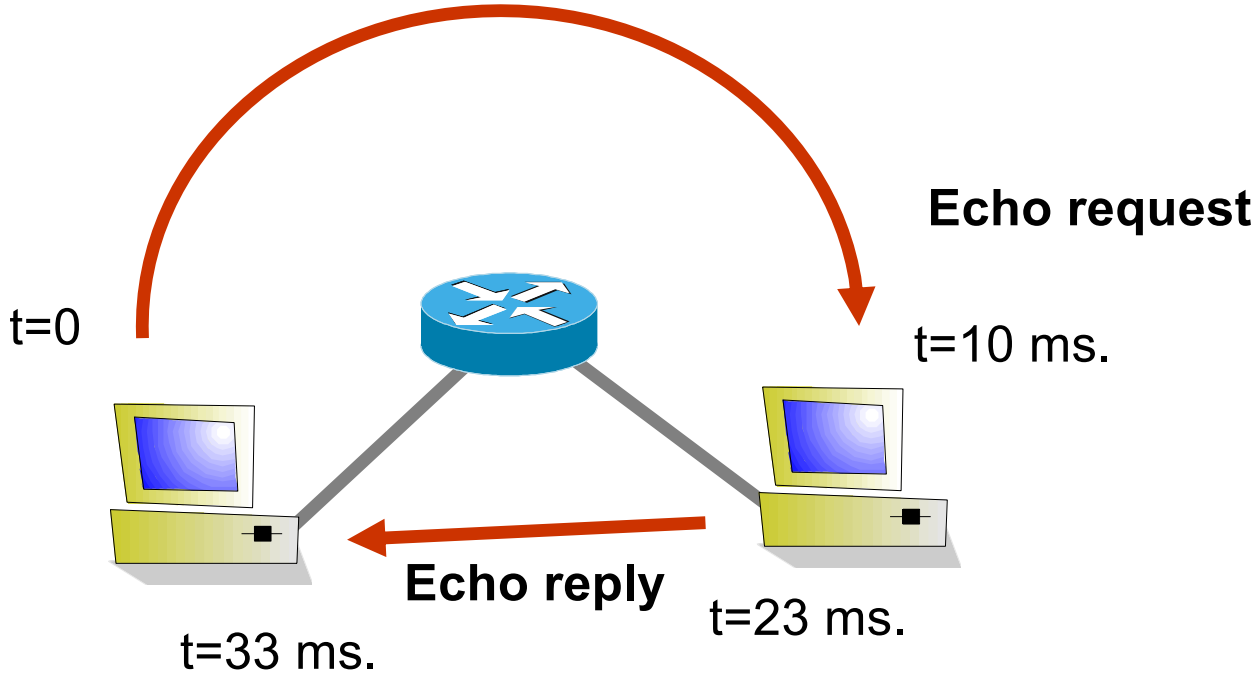


ICMP

- Control mechanism for ip.
- Can check to see if a host is reachable (ping)
- Can tell host traffic is taking to “long” to transmit (ttl || time)
- Tell host to hold off (or quench) sending data.
- (+ Lots of other tools)

ICMP

- Ping (echo request), the most famous icmp tool

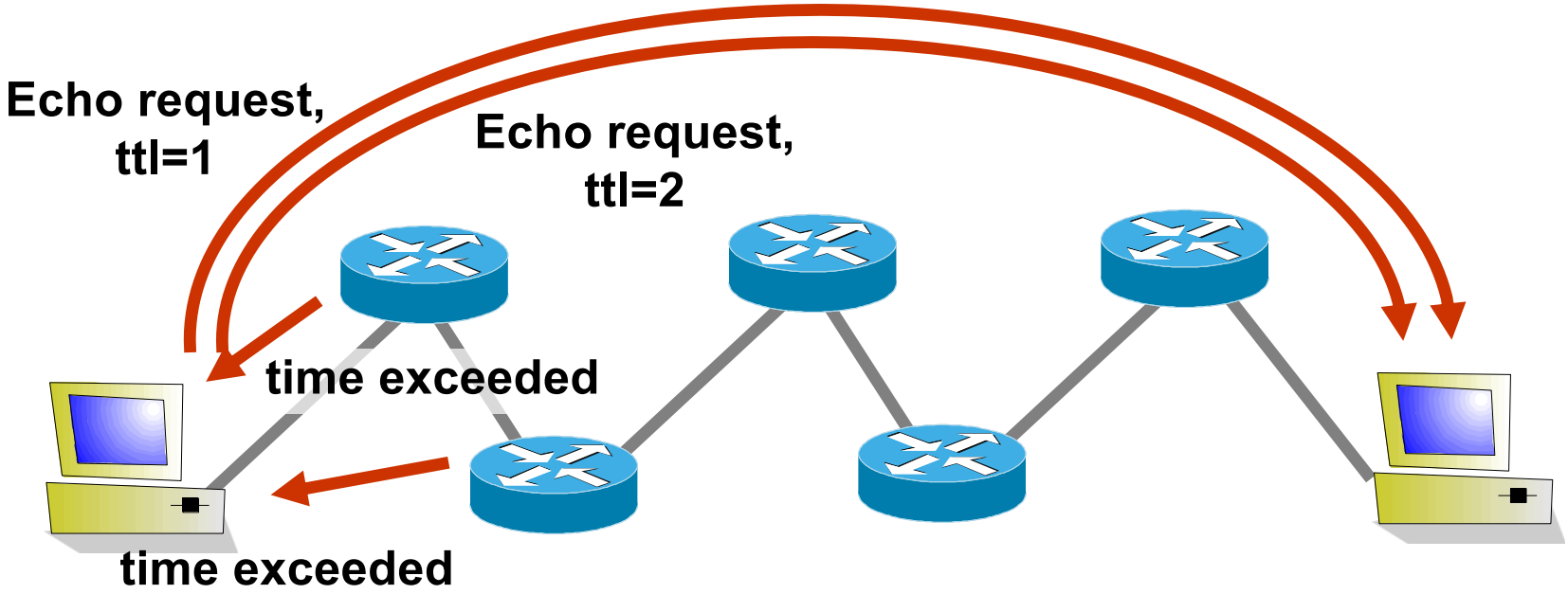



```
mowgli@taco mowgli $ ping printer
PING hp4000n.cmed.us (192.168.1.25) 56(84) bytes of data.
64 bytes from 192.168.1.25: icmp_seq=1 ttl=60 time=2.77 ms
64 bytes from 192.168.1.25: icmp_seq=2 ttl=60 time=1.44 ms
```

- How much delay is in the network? (trick question)

ICMP

- Traceroute: Ping with TTL's




```
chuck # traceroute google.com
traceroute to 216.239.57.99 (216.239.57.99), 30 hops max, 40 byte packets
 1  snv2-bdc-msfc2-vlan128.ariba.com (10.11.128.3)  0.274 ms  0.174 ms  0.188 ms
 2  snv3-core-msfc2-vlan822.ariba.com (10.11.248.2)  1.361 ms  1.528 ms  0.174 ms
 3  dgw-snv-inetgate.ariba.com (205.180.14.1)  0.590 ms  0.537 ms  0.443 ms
 4  65.90.7.25 (65.90.7.25)  2.353 ms  2.254 ms  2.227 ms
```

- Why doesn't all the reply times add up to the total ping reply time?