CMSC 417 Spring 2016 Lecture #3 2/3/2016

Agenda:

⇒ tentative exam/project schedule

⇒ project 0 (how people are doing?) GitHub names

⇒ GitHub usernames to real names

⇒ scaling wires
- hubs / repeaters
- switches
- bridges

⇒ intro to routing protocols

CMSC417 Spring 2016 Lecture #3 2/3/2016

Scaling Wires:

when does this fail?

⇒ too many hosts (Ethernet is 1024?)

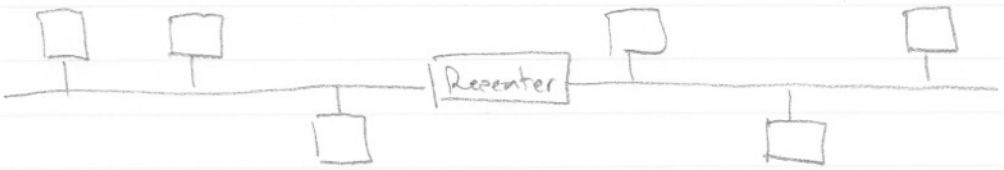
⇒ wires need to be too long (Ethernet is 500 m)

⇒ not enough capacity

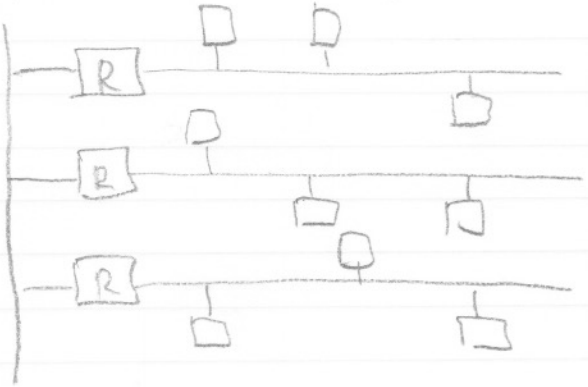
these are really the same

⇒ have to solder or otherwise physically tap into the wire

Repeaters: two wires and they repeat everything gotten on one out the other



Repeaters can be put on another wire



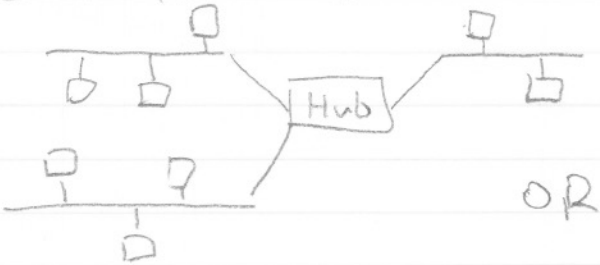
This is all at the physical layer

talking signals & voltages — not frames

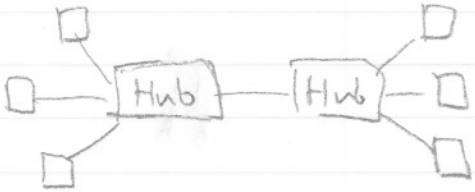
CMSC417 Spring 2016 Lecture #3 2/3/2016

Hubs

Like a repeater but with more than 2 wires



OR



Problems

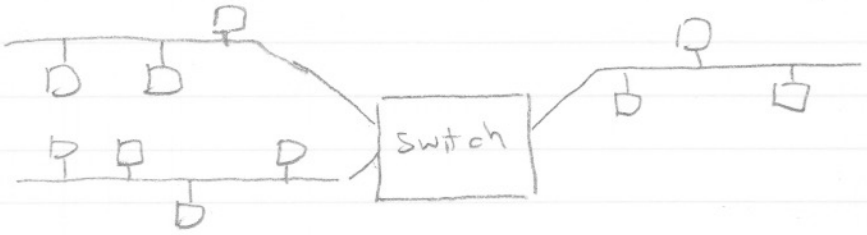
⇒ too many hosts → still there
⇒ wires can't be long enough → still there (now 2500m)
⇒ not enough capacity → still there
⇒ have to physically tap the wire → fixed
real problem is everyone hears every message!

To enable efficient collision detection, Ethernet requires at most 4 repeaters in any given path between 2 hosts

CMSC 417 Spring 2016 Lecture #3 2/3/2016

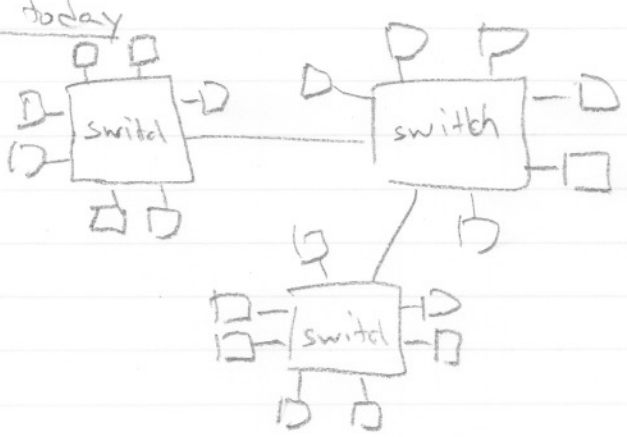
Switching!

⇒ a switch is a hub that tries to forward a frame out only the "right" link (actually the interface on the switch, which is called a port)



⇒ has a buffer, operates on frames

more commonly today



How do we figure out what port is "right"?

⇒ Assuming only that addresses on the network are unique.

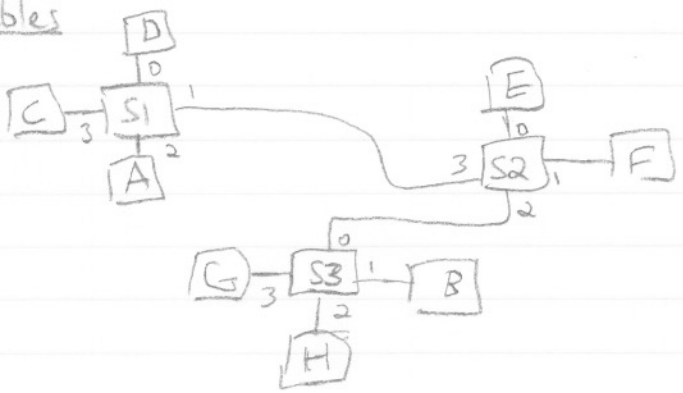
side-note: this is true in Ethernet b/c Network card vendors get a unique prefix and can then keep from repeating what comes after.

CMSC 417 Spring 2016 Lecture #3

Switching Cont'd

finding the "right" port

static tables



S1	S2	S3
A → 2	A → 3	A →
B → 1	B → 2	B → 1
C → 3	C → 3	C → 0
D → 0	D → 3	D → 0
E → 1	E → 0	E → 0
F → 1	F → 1	F → 0
G → 1	G → 2	G → 3
H → 1	H → 2	H → 2

build them by hand knowing the network

Why is this bad?

⇒ hosts move

⇒ people make mistakes typing

⇒ ...

CMSC 417 Spring 2016 Lecture #3 2/3/2016

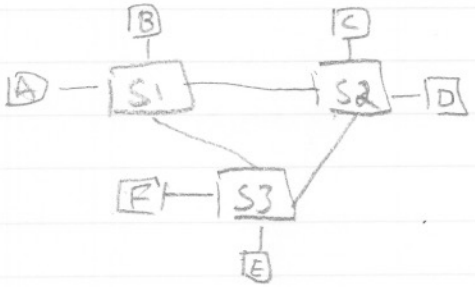
Learning Bridges

⇒ start with an empty table
⇒ on a miss, send out every port but the one it came in on
⇒ record the source of every frame as being out that port
⇒ automatically populate tables as host send frames
⇒ allows for more capacity

A → C, D → E, F → B, and G → H can all send at full speed assuming switches support it

if A → E and C → F at the same time, buffers fill at S1 and eventually both can only send at half rate

What if there is a loop?



⇒ tables start empy
⇒ A sends to D
⇒ S1 sends to S2 & S3, S2 & S3 send to each other, 2 frames will exist forever