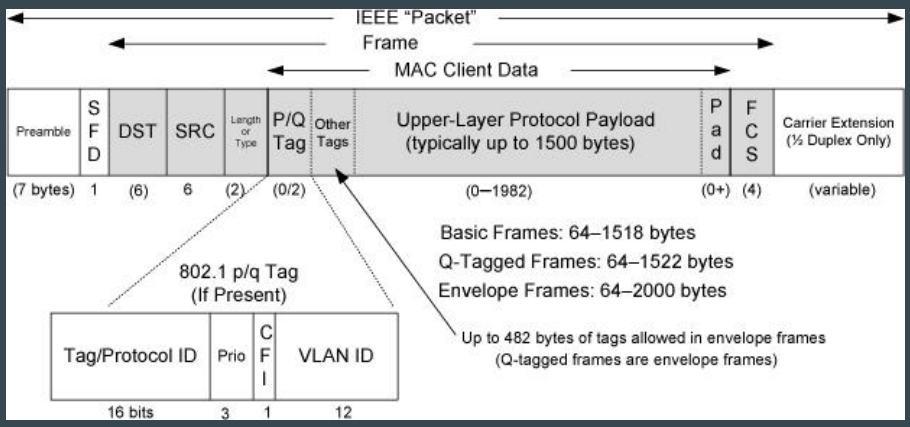
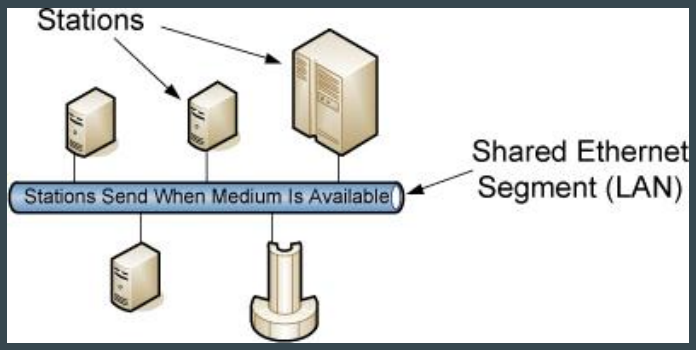
Ethernet Frame Format



(Shared) Ethernet

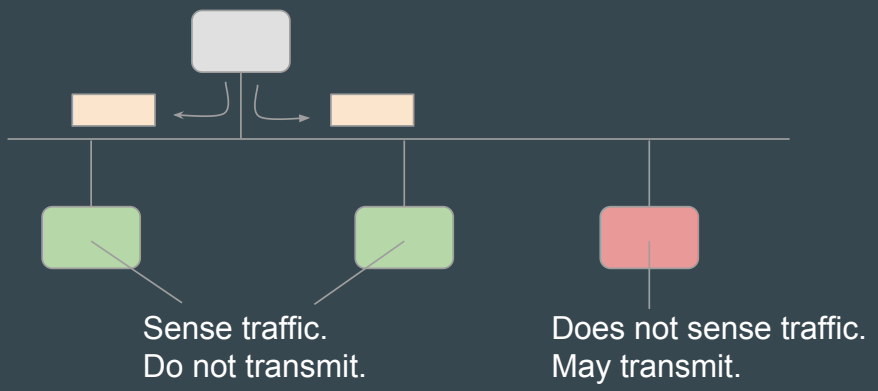


Usually the ethernet exists as hub that all other devices are connected. Any packet that gets put on the bus just gets sent to every port on the bus.

CSMA/CD

Carrier Sense Multiple Access with Collision Detection

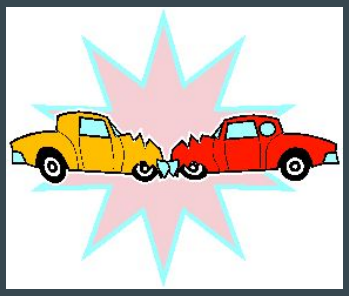
Carrier Sense: do not transmit when you know someone else is.



CSMA/CD is carrier sense multiple access collision detection. This means that before you submit a packet to the bus you check if the carrier has traffic on it. When you send a packet you can only send it one bit at a time which can be problematic as someone else can start submitting while you are still sending. This is called **trasmission delay**. Another problem that occurs is that it takes time for the bit to actually move down the wire which can cause a collision. This is called **propogation delay**.

Collision, Collision Detection

- Collision: > 1 pulse on the bus.
- Detection: hosts check for collision immediately after transmission.
- Minimum frame size (64 bytes) designed with this in mind.



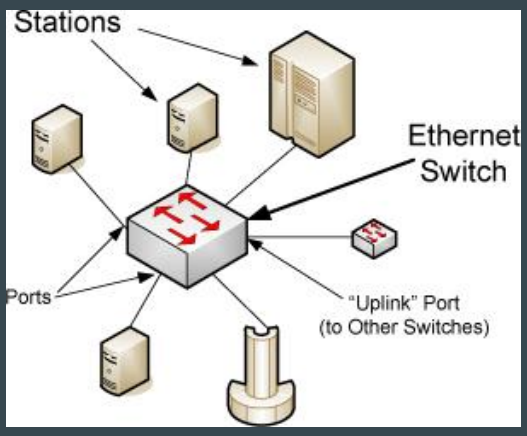
Exponential Back-off

- In attempting to send a frame, *f*, after c^{th} collision
  - Choose a uniform integer, $i \in [0, 2^c - 1]$
  - Delay by $i \times$ "slot time"
  - Usually, $c \in [0, 16]$. Give up after $c = 16$.

A collision is just when there is more than one bit moving on the bus. We can calculate the propagation delay and the size of our packet and such to know which fame caused a collision. If its ours we can resend the packet to try again. These retransmission attempts will occur 16 times (this is just a magic number). We delay by a random amount of time (chose number and multiply by the time it takes to traverse).

Switched Ethernet

- Switch learns and maintains a table.
- While not yet learned, switch acts as a hub.

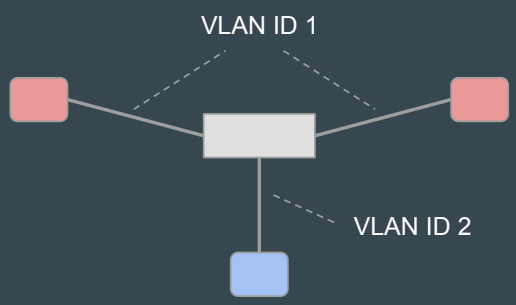


Connected to	*via Port*
c4:8e:f5:68:10:1e	5
2e:8d:11:ee:5e:8a	2
...	...

The performance of shared ethernet is god awful. This resulted in the invention of switched internet. For this the packet on the ethernet only gets sent to the port that it actually wants to. When it first gets turned on it behaves like a hub, but as packets are sent it builds a table of addresses for the various devices on the network. When a device sends something it notes its address.

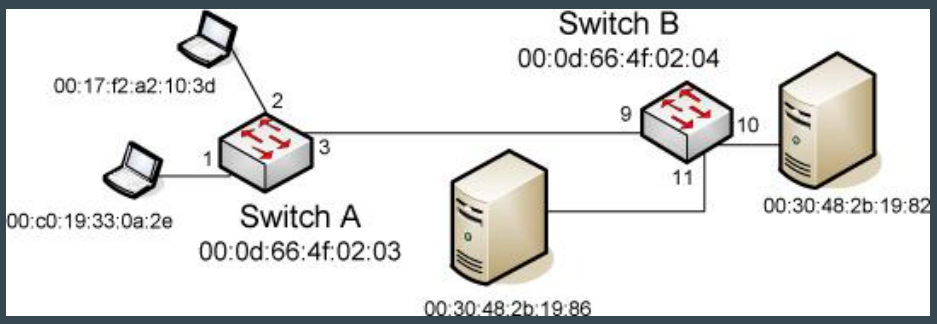
Virtual LAN (VLAN)

- Hosts connected to same switch may not be on same VLAN.
- One way to do this: switch binds a port to a VLAN ID.
- Traffic allowed to flow within VLAN only - extra column in switching table.



Over the years switches got hella powerful with tons of ports. You can extend your network using vlan. When the switch is building its table of devices it also records a vlan id for each one. If a device tries to send a packet to a device of a different vlan id the switch will not send it. This allows us to segregate our network.

Extended LANs



Switching Table for Extended LAN

Station	Port
00:17:f2:a2:10:3d	2
00:c0:19:33:0a:2e	1
00:0d:66:4f:02:03	
00:0d:66:4f:02:04	3
00:30:48:2b:19:82	3
00:30:48:2b:19:86	3

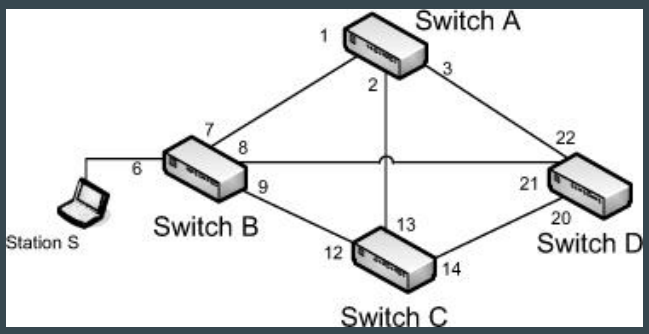
Switch A's Database

Station	Port
00:17:f2:a2:10:3d	9
00:c0:19:33:0a:2e	9
00:0d:66:4f:02:03	9
00:0d:66:4f:02:04	
00:30:48:2b:19:82	10
00:30:48:2b:19:86	11

Switch B's Database

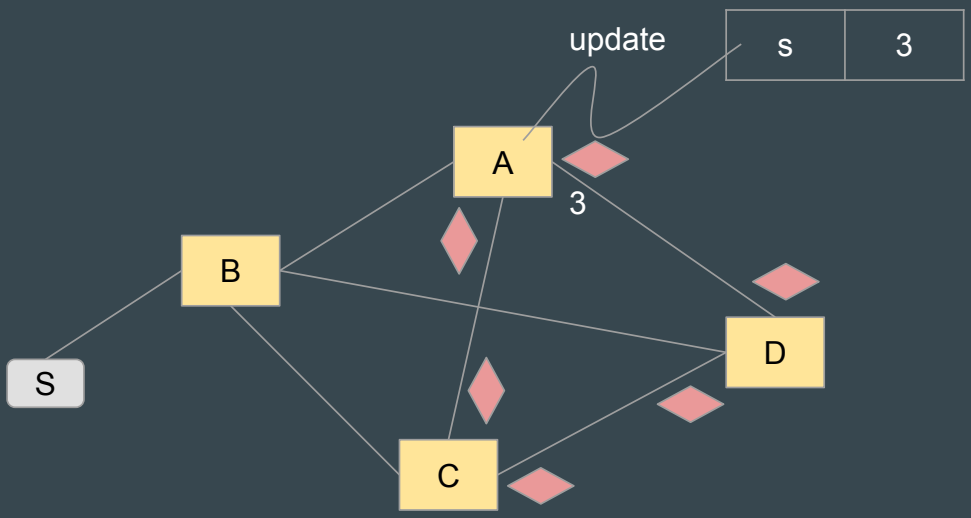
If you still want to send a packet to a different vlan you can do it by sending it through a higher level of protocol (like sending it through ip). You could also bridge the two devices. Basically making a special entry saying that its totally ok for those two to talk in the switch's table.

Spanning Tree Protocol (STP) - motivation



- What could go wrong?

STP - motivation, contd.



We can create connections between devices, by physically wiring them together, ignoring the switch. This makes it easier to talk since they don't have to make as many hops. We also have redundancy if someone where to remove one of those wires. On the flipside we can run into problems with loops. The devices don't have switching tables, so they have to build them. When a devices gets a packet it sends to out on all of its ports (it doesnt have a switch table built up yet). As the packet hits those devices they do the same thing which makes A update its switch table.

STP - motivation, contd.

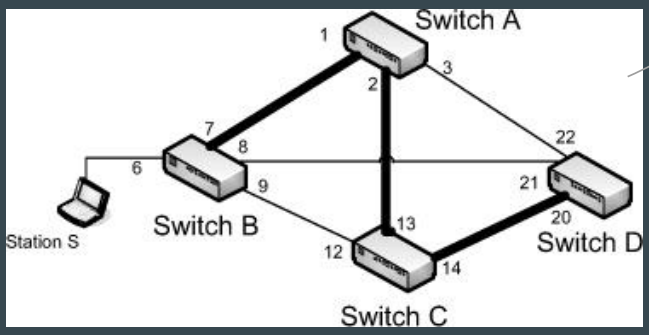
So two problems:

- Broadcast storm
- Oscillation of switching tables

This causes a shit ton of packets to get sent out. Because those packets are getting sent everywhere the switch tables stored on devices oscillate like crazy.

A solution

- Switches first agree on a Spanning Tree.
- Spanning Tree Protocol (STP).
- Given an undirected connected graph G, a spanning tree T of G is a connected acyclic subgraph.



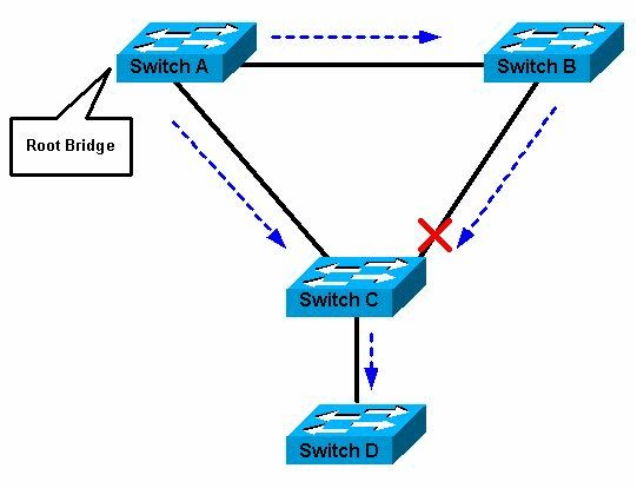
E.g., Switch B will forward packets on ports 6 and 7 only.

The solution to this is to build a spanning tree out of all of your connected devices. A super efficient way to do this is through a simple breadth first search. If you have weights and care about it you can use Prim's algorithm.

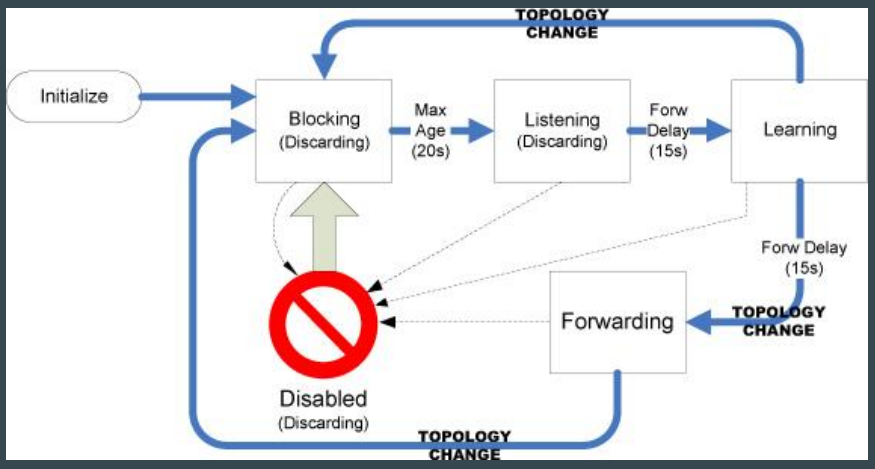
How STP works

- Special frames: Bridge Protocol Data Units (BPDUs).
- A rooted tree is built.

How STP works, contd.



State Machine for each Port



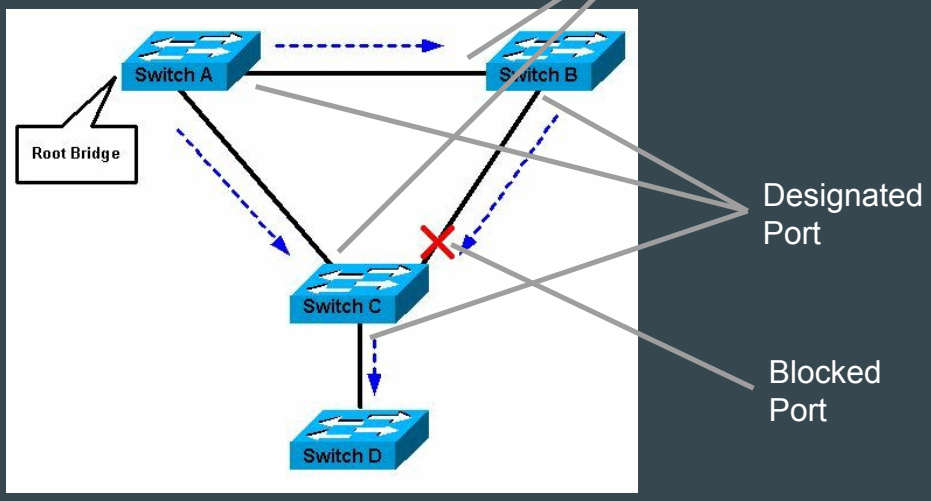
Use of the state machine (1)

- *Blocking* state - not allowed to transmit, only receive BPDUs.
- *Listening* - allowed to receive and transmit BPDUs.
- *Learning* - allowed to Learn switching table.
- *Forwarding* - (finally) allowed to forward data frame.

Defensive mindset. To avoid loops.

Use of state machine (2)

Each port converges to a *role*.



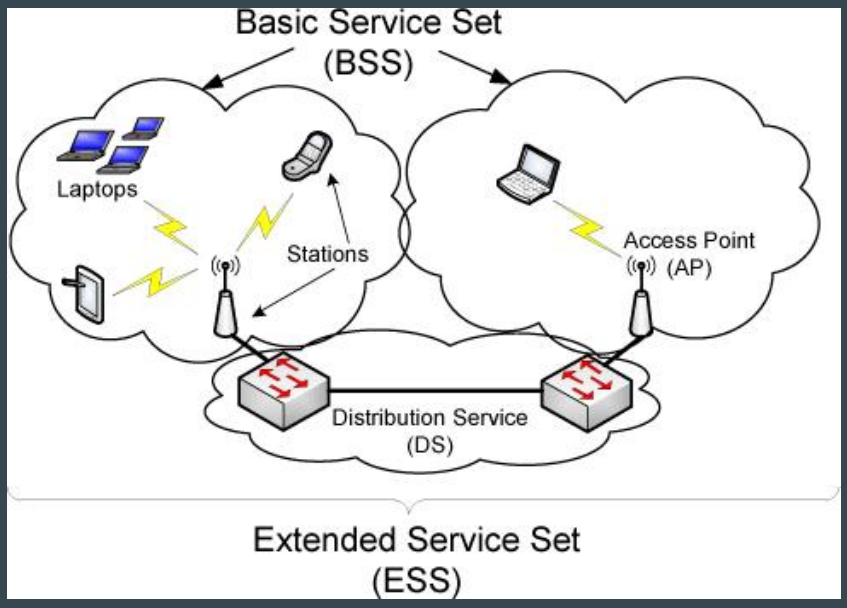
Root Port

Designated Port

Blocked Port

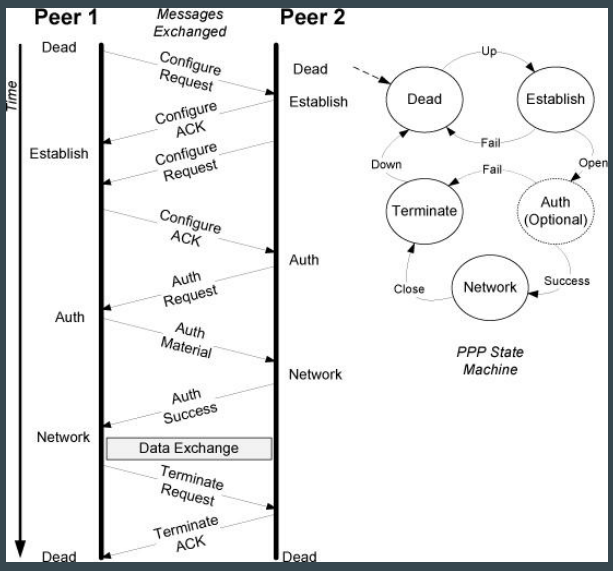
When we build the spanning three we send out a bunch of special packets that the switches know contain nothing and handle properly. With this we build a rooted tree. As we build we record numbers on the devices so that we can tell that the device with the smallest number is the root relative to you. Before it has received a packet all devices think they are the root. As packets go through they re-evaluate this based on the packet age. Devices create designated root ports and designated designated ports through which its children can send to the root through it. It takes about a minute of it building its shit when you reboot your ethernet switch.

Wireless LAN - WiFi



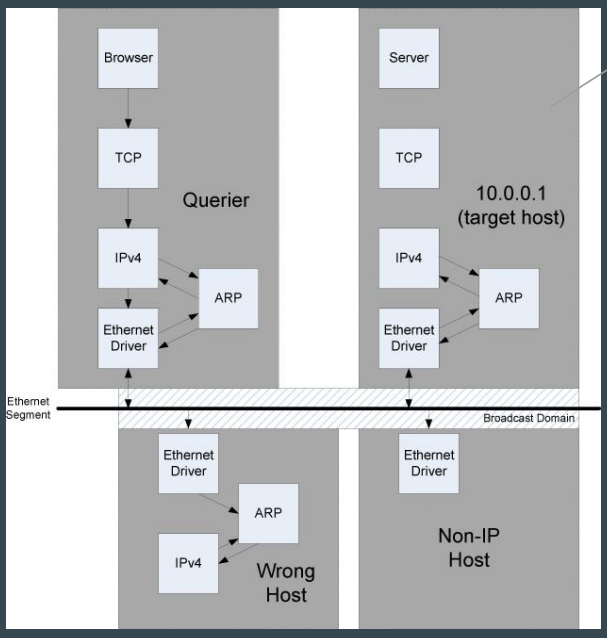
Lower-layer protocols can be complex

E.g., Link Control Protocol for Point-to-Point links



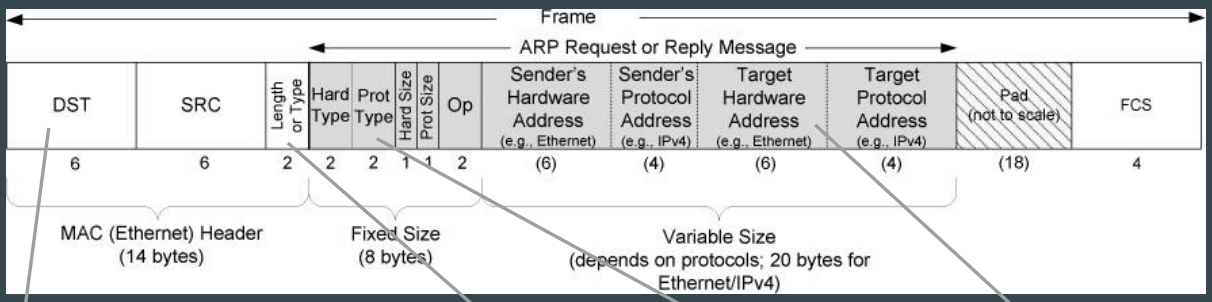
Even in point to point connections shit can get complicated. We send a bunch of configuration and authentication requests with a bunch of acknowledgements and responses. In this image you can see all the crap bouncing back and forth for a simple device to device connection. Stevens

Address Resolution Protocol (ARP) - Chapter 4



May be a router, instead

ARP Frame encapsulated in Ethernet Frame



DST = link-layer broadcast address in a request. Unicast in a response.

ARP = 0x0806

IP = 0x0800

0's in request

ARP is how we resolve conflicting addresses. We have a special box that checks for any matching addresses. If there aren't any the ARP box gives the go-ahead and the ethernet driver can construct the ethernet frame and send its shit out.

ARP works by building a dummy frame. It includes ethernet information (destination, source, and type). Then you have the ARP frame. It contains the sender and target's hardware address (the ethernet address) and protocol address(the ipaddress). The sender just fills that in with its own information. If you are send it locally, you know that the target's protocol address is the same as yourse (since it is on the same network). Problem is, you don't know the target's hardware address. This is why we have a request or response value stored in the op field. In a request the target hardware address is all 0s since you dont know it. If you don't know the destination address you can fill in all 1s which the protocol knows that its a broadcast frame, so the device that it hits just absorbs it. It then populates the target hardware address and sends a response. This will be unicast (the destination address is the device we are looking at).

We **resolve** addresses, not translate the.