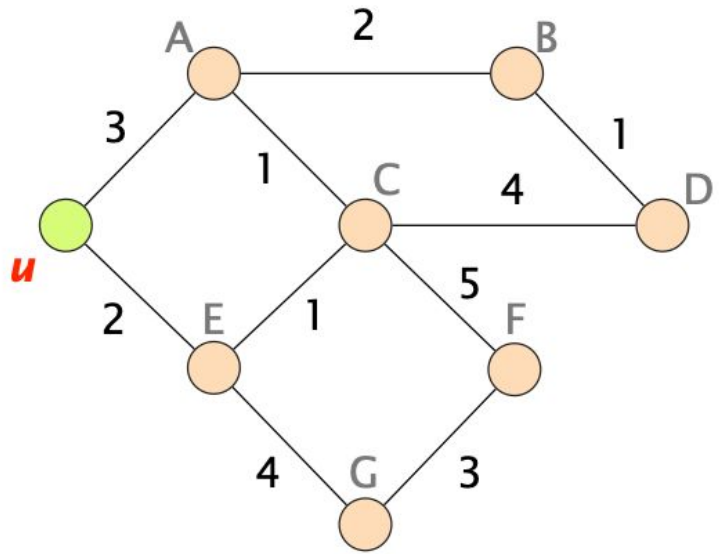
Review

Dijkstra's Algorithm for Shortest Path Search

S = {*u*} — set of nodes for which we know the shortest-path

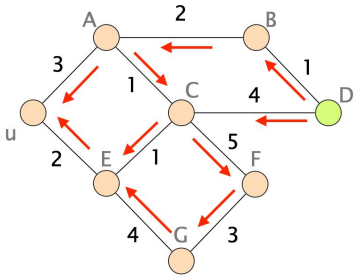
D(*v*) — the smallest distance currently known by *u* to reach *v*

c(*u*,*v*) — the weight of the link connecting *u* and *v*



Dijkstra's Algorithm -> Link State Routing

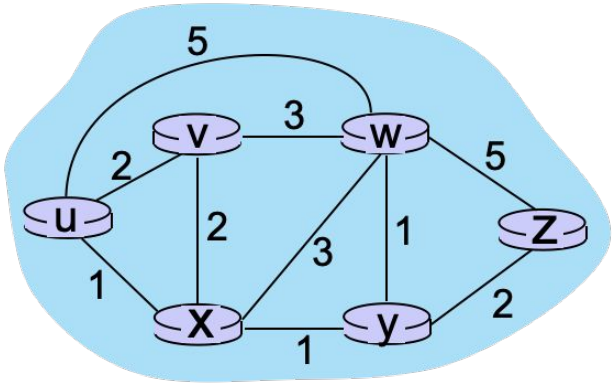
- Each router floods its link state information to other n routers in order to generate a global view
- Updates are sent when things change, and only the difference is sent, not everything
- Any drawbacks you can think of?
- ```
  U: {v=2, x=1, w=5}
  V: {u=2, x=2, w=3}
  W: {v=3, u=5, x=3, y=1, z=5}
  X: {u=1, v=2, w=3, y=1}
  Y: {x=1, w=1, z=2}
  Z: {w=5, y=2}
  ```



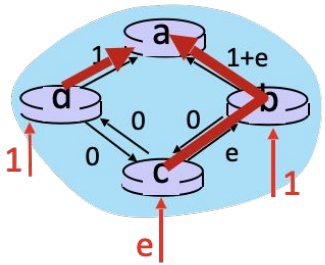
D's Advertisement

edge (D,B); cost: 1

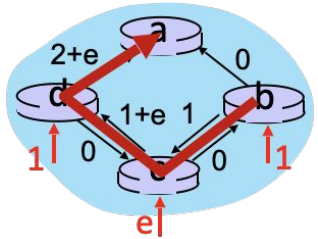
edge (D,C); cost: 4


  ```

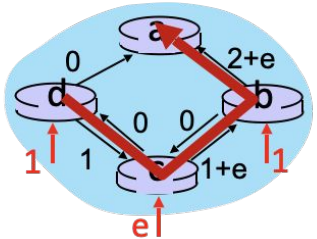
Dynamic Weights -> Route Oscillations



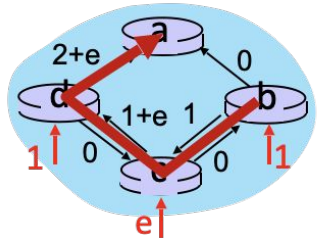
initially



given these costs,
find new routing....
resulting in new costs



given these costs,
find new routing....
resulting in new costs



given these costs,
find new routing....
resulting in new costs

Link State Algorithms

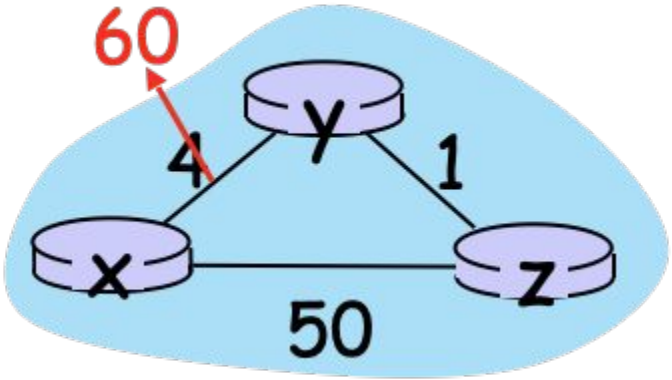
Pros

- Fast convergence
- Event-driven updates
- Every router can determine the best path

Cons

- Computationally expensive
- Memory intensive
- If a network is constantly changing, bandwidth can suffer from overhead of messages

Distance Vector Suffers From the “Count to Infinity” Problem



Count to Infinity

Node X:

	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5	1	0

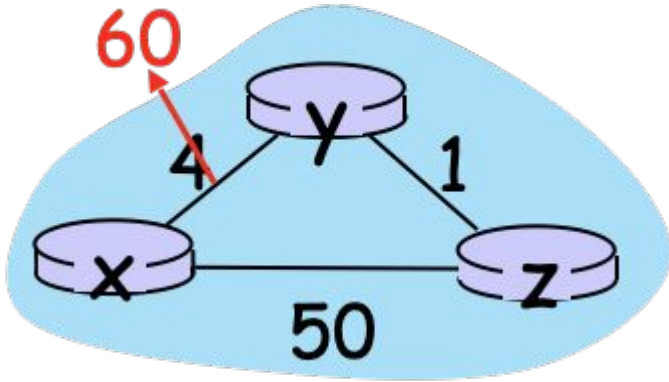
(Ignore X for simplicity)

Node Y:

	X	Y	Z
X	0	4	5
Y	**6**	0	1
Z	5	1	0

Node Z:

	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5	1	0



Count to Infinity

Node X:

	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5	1	0

(Ignore X for simplicity)

Node Y:

	X	Y	Z
X	0	4	5
Y	**6**	0	1
Z	5	1	0

Node Y:

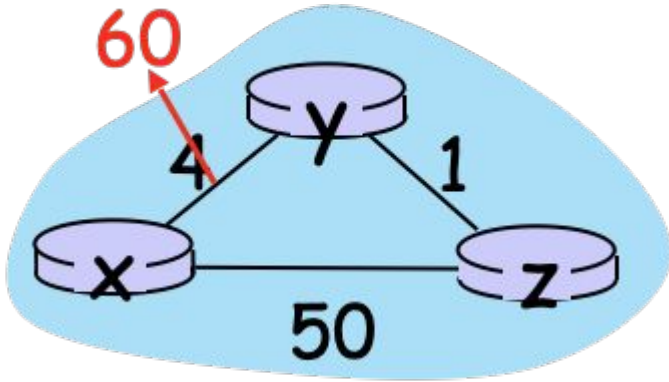
	X	Y	Z
X	0	4	5
Y	6	0	1
Z	5	1	0

Node Z:

	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5	1	0

Node Z:

	X	Y	Z
X	0	4	5
Y	6	0	1
Z	**7**	1	0



Count to Infinity

Node X:

	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5	1	0

(Ignore X for simplicity)

Node Y:

	X	Y	Z
X	0	4	5
Y	**6**	0	1
Z	5	1	0

Node Z:

	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5	1	0

Node Y:

	X	Y	Z
X	0	4	5
Y	6	0	1
Z	5	1	0

Node Z:

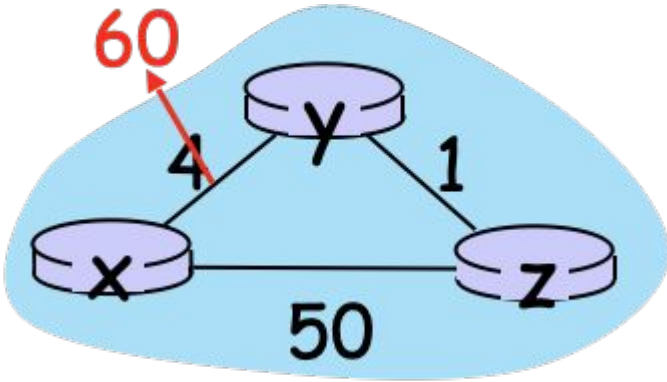
	X	Y	Z
X	0	4	5
Y	6	0	1
Z	**7**	1	0

Node Y:

	X	Y	Z
X	0	4	5
Y	**8**	0	1
Z	7	1	0

Node Z:

	X	Y	Z
X	0	4	5
Y	6	0	1
Z	7	1	0



Count to Infinity

Node X:

	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5	1	0

(Ignore X for simplicity)

Node Y:

	X	Y	Z
X	0	4	5
Y	**6**	0	1
Z	5	1	0

Node Z:

	X	Y	Z
X	0	4	5
Y	4	0	1
Z	5	1	0

Node Y:

	X	Y	Z
X	0	4	5
Y	6	0	1
Z	5	1	0

Node Z:

	X	Y	Z
X	0	4	5
Y	6	0	1
Z	**7**	1	0

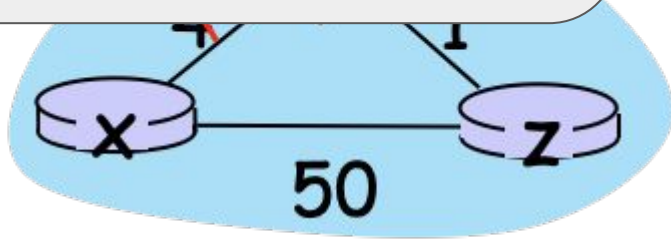
Node Y:

	X	Y	Z
X	0		
Y	**8**	0	1
Z	7	1	0

Node Z:

	X	Y	Z
X	0	4	5
Y	6	0	1
Z	7	1	0

This will continue until they realize that 50 (X <> Y) is cheaper



DV Routing

“Bad News Travels Slowly,
Good News Travels Fast”

Distance Vector Algorithms

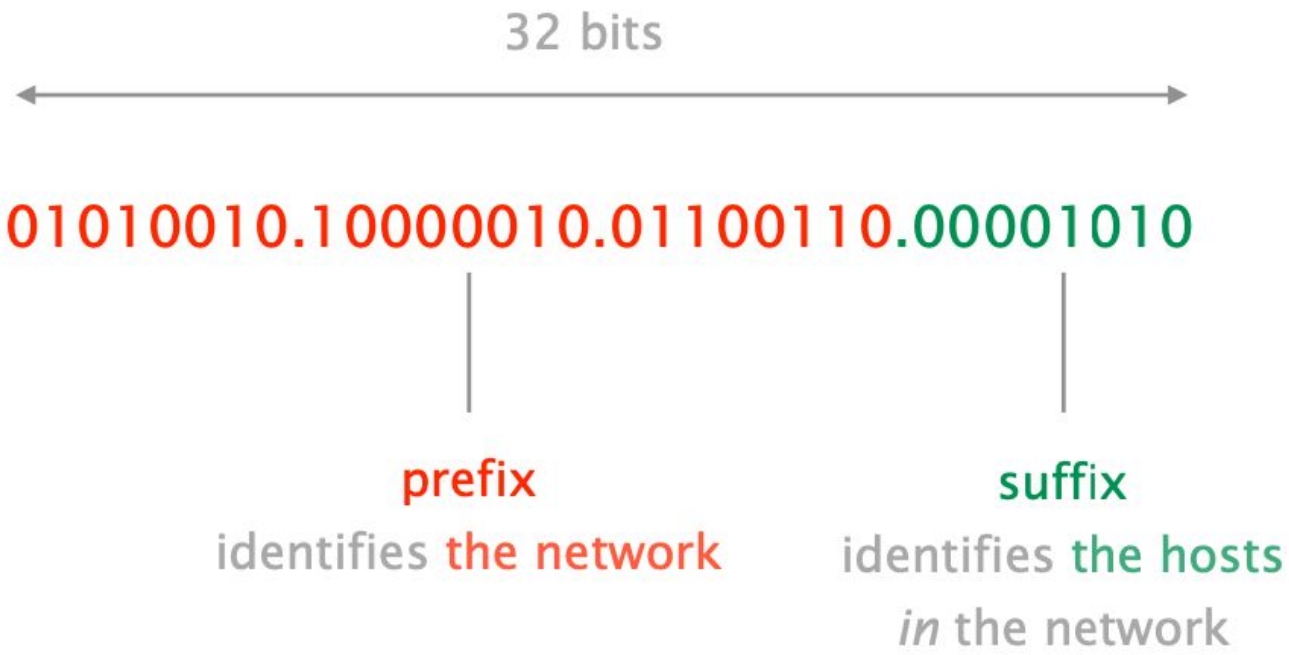
Pros

- Simple to configure / maintain
- Only need a local view of the world

Cons

- Slow to converge
- Loops are possible
- Count to infinity
- Wastes bandwidth - constant updates even when nothing changes

IPs are Hierarchically, Composed of Prefix (network address) and Suffix (host address)



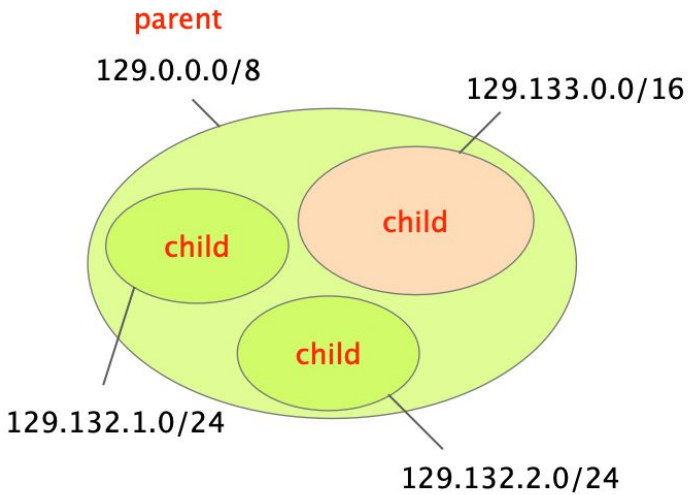
Prefixes Have Varying Lengths, Usually Written Using “slash notation”

Route Aggregation

Child prefixes can be removed from the table if they share the same output interface as the parent

Routing Table

IP prefix	Output Interface
…	
129.0.0.0/8	IF#2
~~129.132.1.0/24~~	~~IF#2~~
~~129.132.2.0/24~~	~~IF#2~~
129.133.0.0/16	IF#3
…	



NAT

- Share a single (public) address or a pool (CG NATs) between hosts
  - Port numbers (transport layer) are used to distinguish

- One of the main reasons why we can still use IPv4
  - Saved us from address depletion

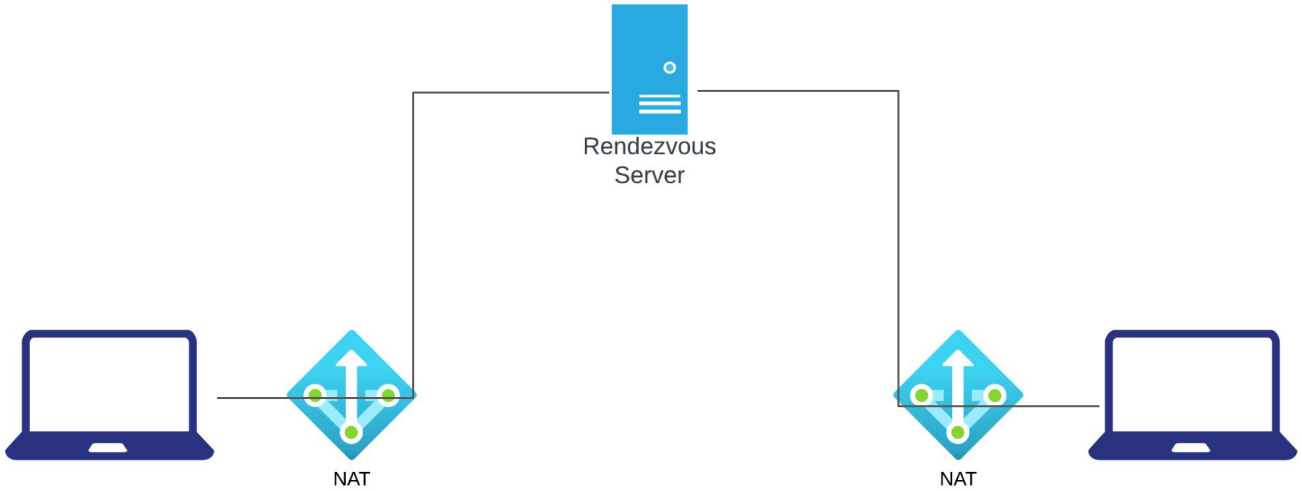
- Violates the general end-to-end principle of the Internet
  - A NAT box adds a layer of indirection

NAT Pros and Cons

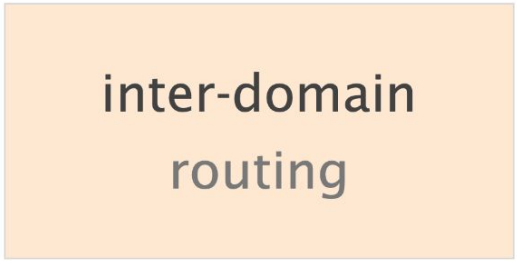
- Better privacy/anonymization
  - All hosts in one network get the same public IP
  - But, cookies, browser version, ... still identify hosts
- Better security
  - From the outside you cannot directly reach the hosts
  - Problematic e.g., for online gaming
- Limited scalability (size of the mapping table)
  - Example: Wi-Fi access problems in public places (e.g., lecture hall) often due to a full NAT table

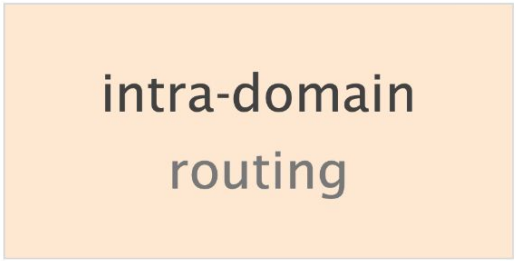
Hole Punching

Each machine sets up a connection to a publicly reachable rendezvous server, which then relays the streams for the clients. This is known as hole punching



Routing Comes in Two Flavors: *intra* and *inter*-domain routing

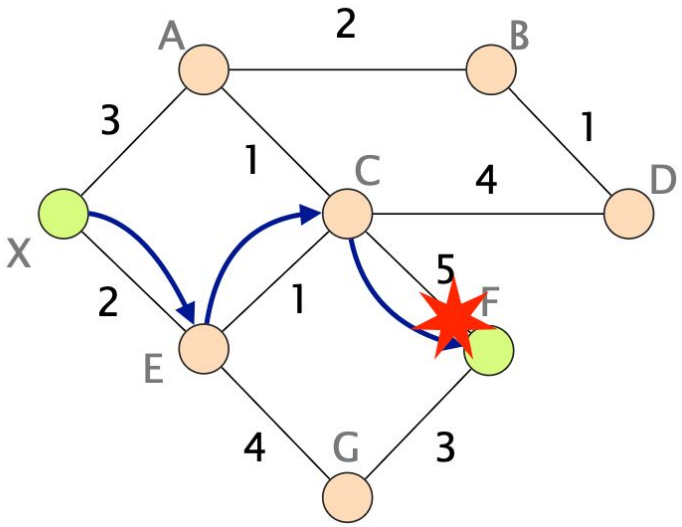




Find paths between networks

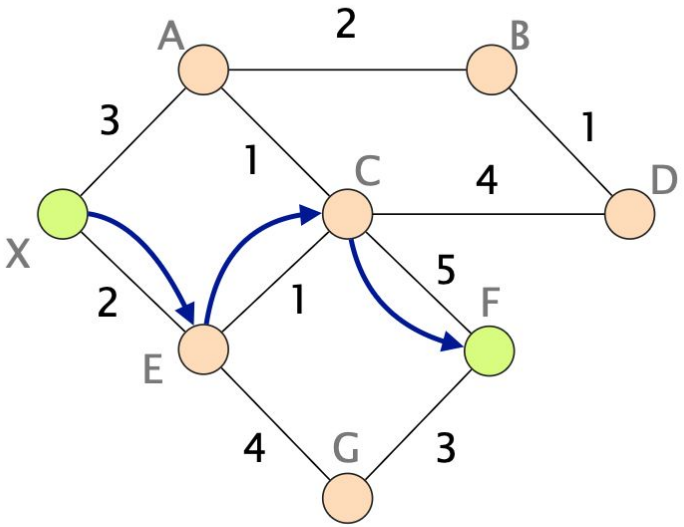
Find paths within a network

Black Holes - Due to Detection Delay as Routers Do Not Immediately Detect Failure



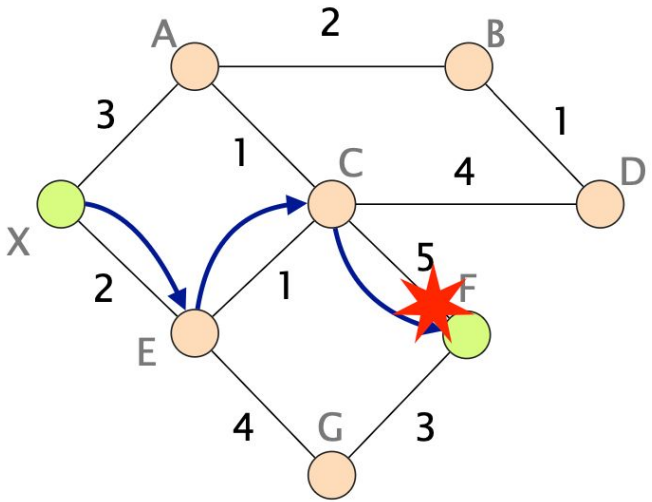
depends on the timeout for detecting lost hellos

Forwarding Loops - Due to Inconsistent Link-State DBs



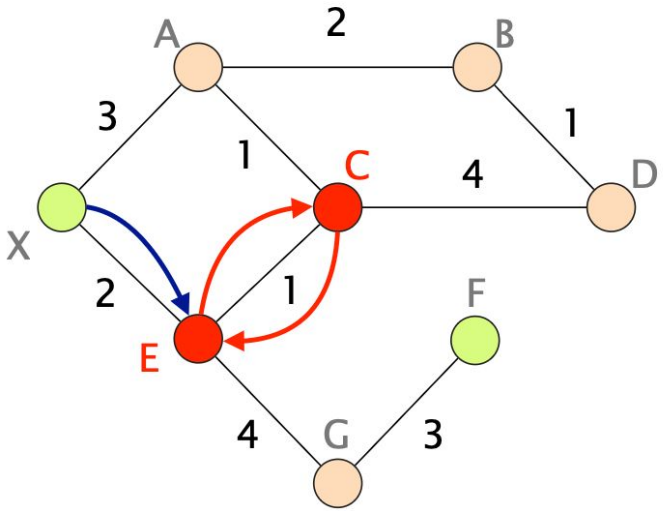
Initial forwarding state

Forwarding Loops - Due to Inconsistent Link-State DBs



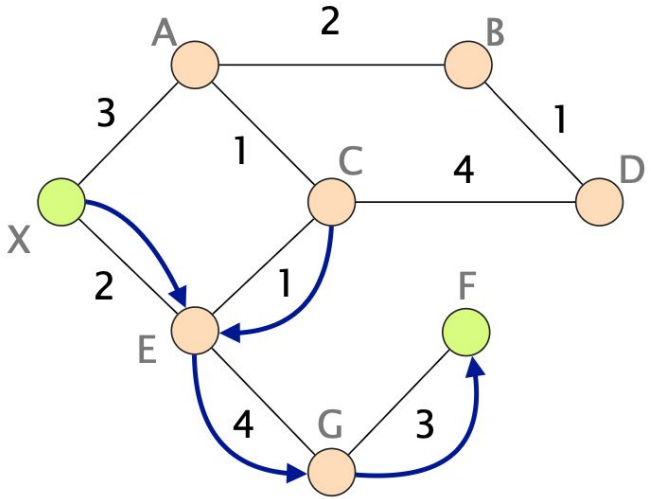
C learns about the failure
and immediately reroute to E

Forwarding Loops - Due to Inconsistent Link-State DBs



A loop appears as E
isn't yet aware of the failure

Forwarding Loops - Due to Inconsistent Link-State DBs



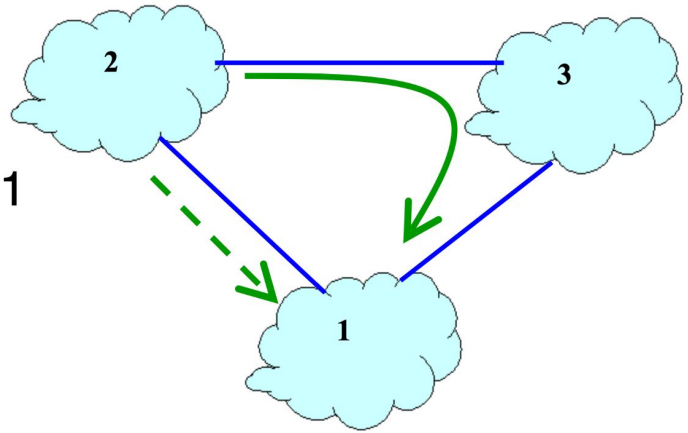
The loop disappears as soon as
E updates its forwarding table

BGP is similar to DV

But, four key differences:

1. BGP does not pick the shortest path routes
    a. BGP selects route based on **policy**, not shortest distance/least cost

Node 2 may prefer 2, 3, 1 over 2, 1



BGP is similar to DV

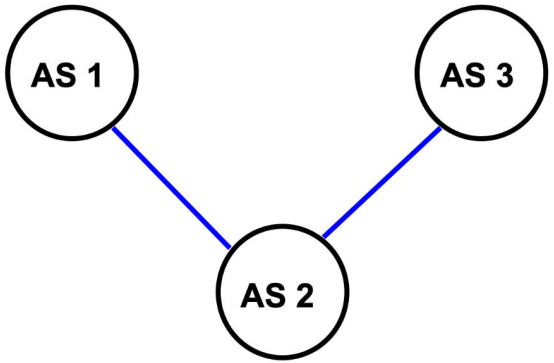
But, four key differences:

2. **Path-vector** Routing
    a. Benefits
        i. Loop avoidance is easy
        ii. Flexible policies based on entire path

BGP is similar to DV

But, four key differences:

3. Selective Route Advertisement
   a. For policy reasons, an AS may choose not to advertise a route to a destination
   b. As a result, **reachability is not guaranteed** even if the graph is connected

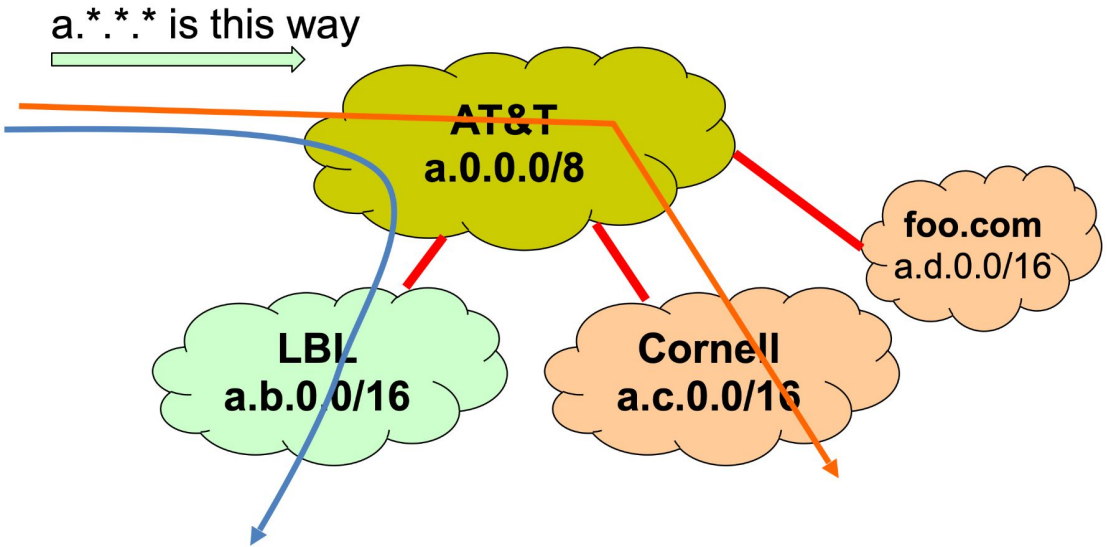


Example: AS#2 does not want to carry traffic between AS#1 and AS#3

BGP is similar to DV

But, four key differences:

4. BGP may **aggregate** routes



BGP Needs to Solve Three Challenges: Scalability, Privacy and Policy Enforcement

- There is a huge # of networks and prefixes
  - 1M prefixes, >70,000 networks, millions of routers
- Networks don’t want to divulge internal topologies or their business relationships
- Networks need to control where to send and receive traffic without an Internet-wide notion of a link cost metric

BGP Updates Carry an IP Prefix and Some Attributes

Attributes	Usage
NEXT-HOP	egress point identification
AS-PATH	loop avoidance outbound traffic control inbound traffic control
LOCAL-PREF	outbound traffic control
MED	inbound traffic control

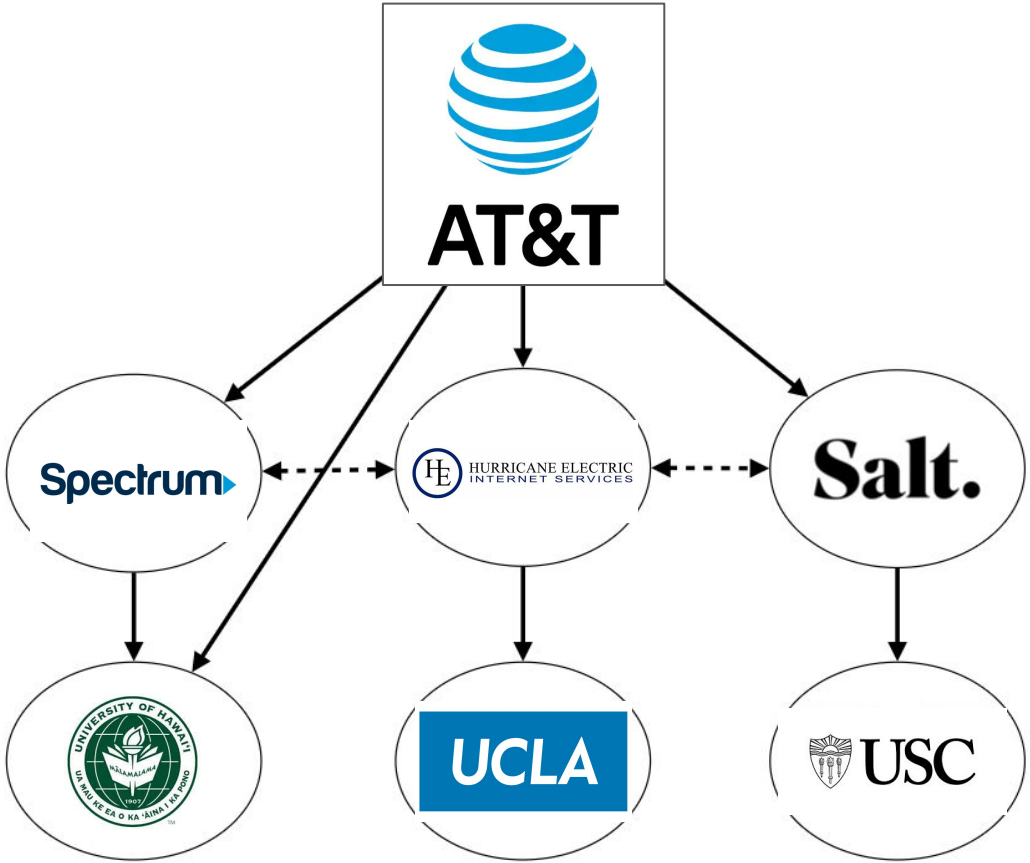
BGP route selection

- Default decision for route selection
  - Highest local pref, shortest AS path, lowest MED, prefer eBGP over iBGP, lowest IGP cost, router id
- Local Pref controls egress
- MED (attempts) to control ingress
- Sender always has the final say

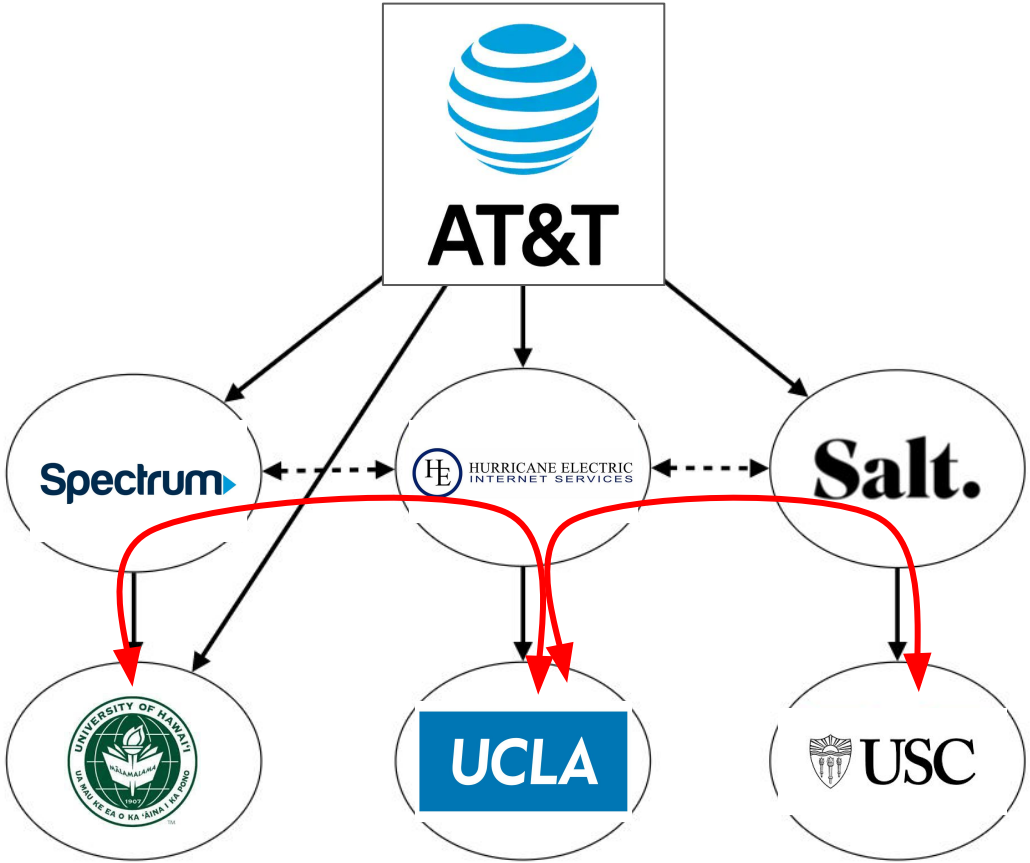
2 Main business relationships today

There are 2 main business relationships today:

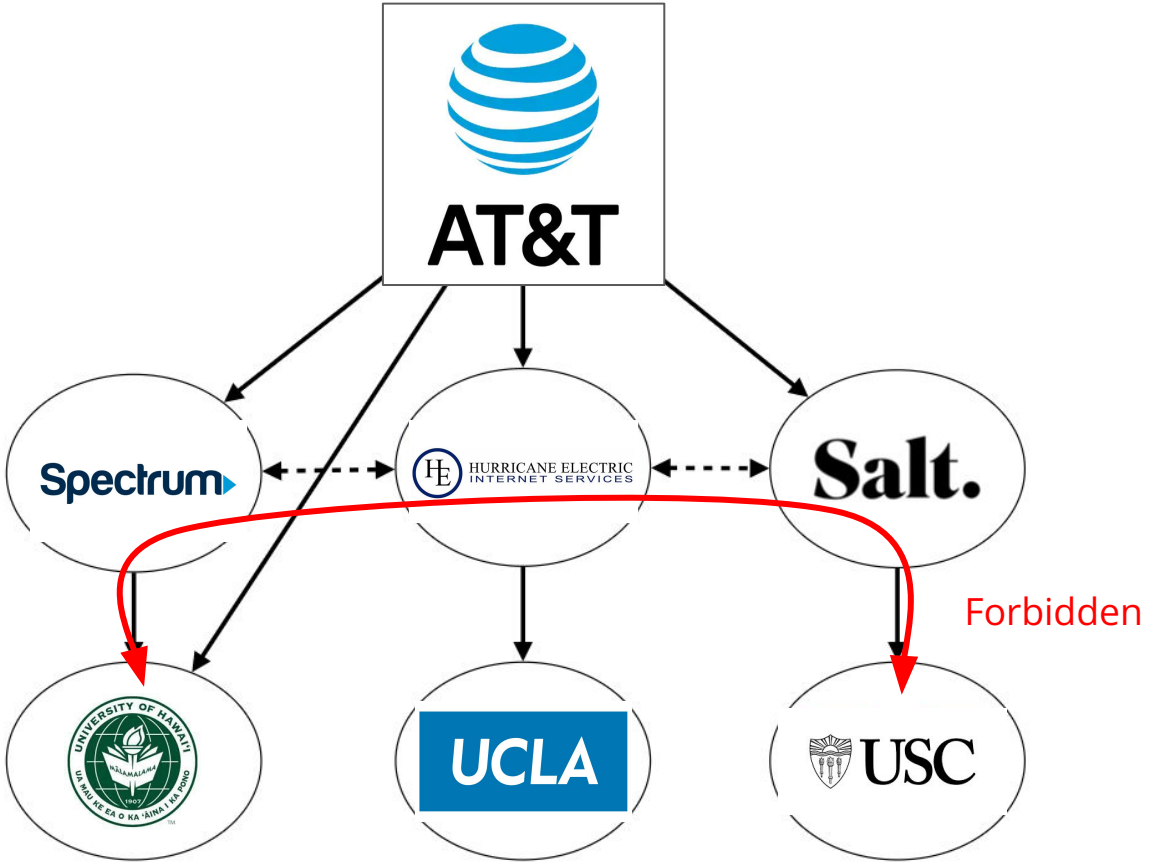
- customer/provider
- peer/peer


AT&T
Spectrum
HURRICANE ELECTRIC
INTERNET SERVICES
Salt.
UNIVERSITY OF HAWAI'I
UCLA
USC

Providers transit traffic for their customers



Providers DO NOT transit traffic between each other



Business relationships condition route selection

For a destination *p*, **prefer routes coming from**

- **customers** over
- **peers** over
- **providers**

Network Adapter Bootstrap

Two problems to solve:

1. Who am I? How do I acquire an IP address?
   Dynamic Host Configuration Protocol

2. Who are you? Given an IP, how do I find which MAC to send to?
   Address Resolution Protocol

DHCP

- Used to request IP addresses for the network layer
- Uses link-layer broadcast address: ff:ff:ff:ff:ff:ff
- Lease times avoid pool exhaustion

ARP

ARP is stateless

- Hosts will automatically cache any ARP replies they receive, regardless of whether network hosts requested them.
- Even ARP entries that have not yet expired will be overwritten when a new ARP reply packet is received.
- There is no method in the ARP protocol by which a host can authenticate the peer from which the packet originated.
    - Allows ARP spoofing

Constructing a Spanning Tree

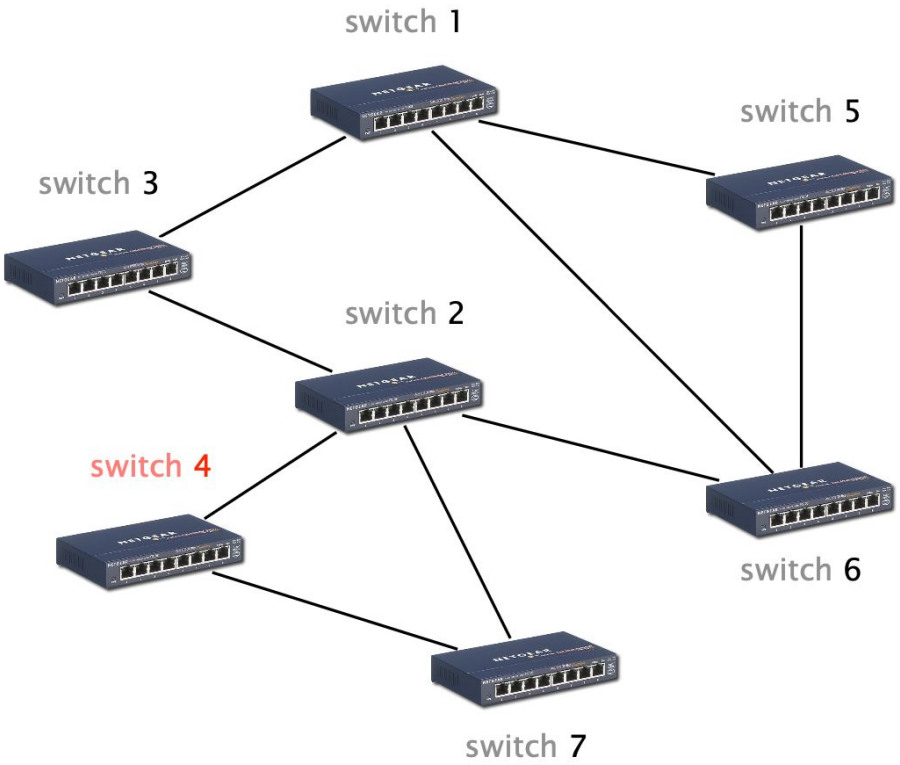
Switches...

- elect a root switch
  - the one with the smallest identifier
- determine if each interface is on the shortest-path from the root
  - disable it if not

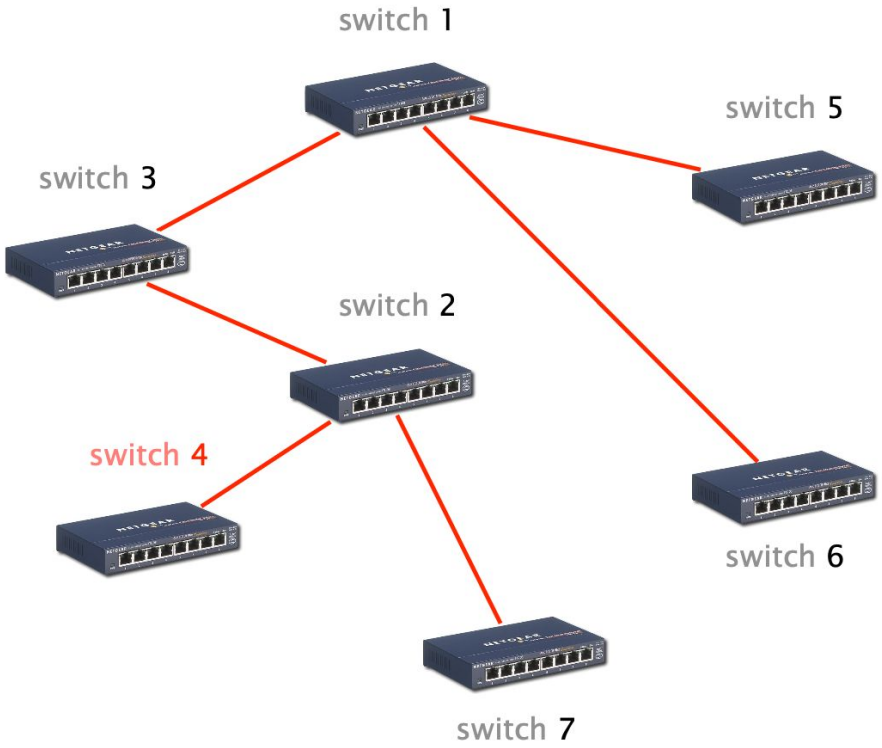
Constructing a Spanning Tree

- Select a root bridge (typically lowest ID, can be configured)
- All ports on bridge become “designated”
- All ports on other switches facing the root become “root” ports
- Lower ID switches block loop ports

Constructing a Spanning Tree



Constructing a Spanning Tree

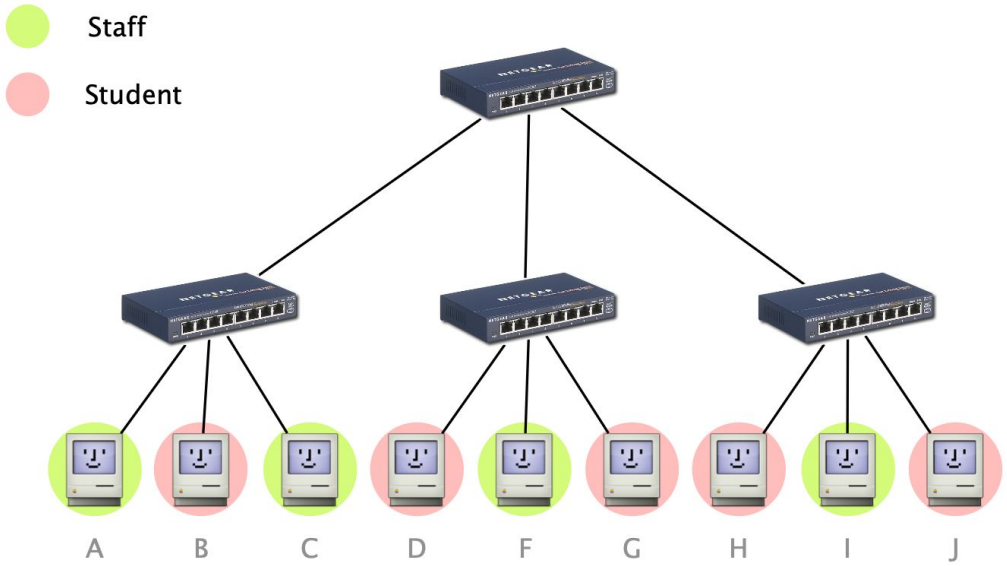


STP Must React to Failure

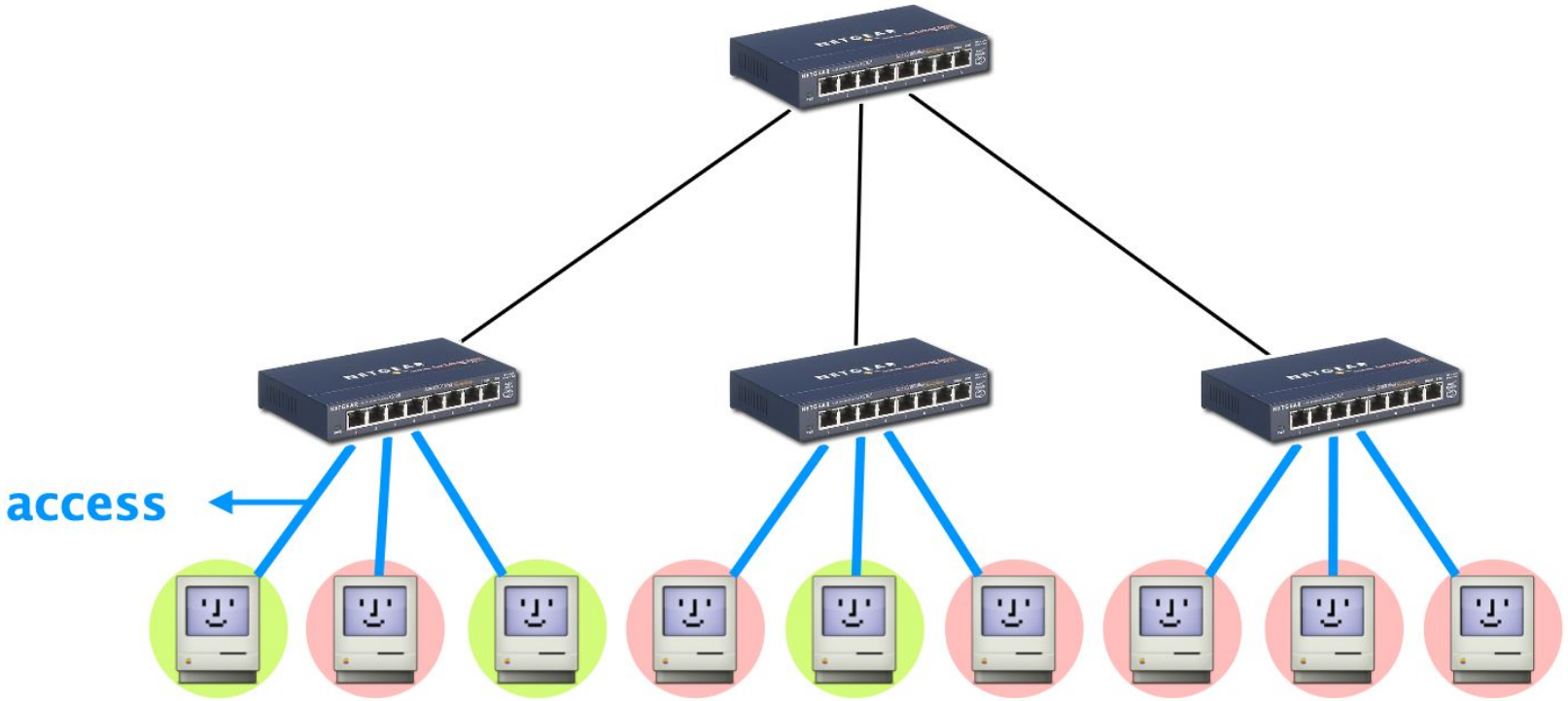
- Any switch, link or port can fail
  - including the root switch
- Root switch continuously sends messages
  - announcing itself as the root, others forward it
- Failures detected through timeout (soft state)
  - if no word from root in X, times out and claims to be the root

VLANs (Virtual Local Area Networks)

A VLAN logically identifies a set of ports attached to one (or more) Ethernet switches, forming one broadcast domain



Access Links Only Belong to One VLAN and Do Not Carry 802.1q Headers



Trunk Links Carry Traffic for More Than One VLAN and Use 802.1q Headers