Lecture 3: Data Center and Enterprise Network Security

CS 598: Network Security

Matthew Caesar

January 29, 2013

Why secure data centers?

- Consolidation brings many benefits
  - Easier management, statistical multiplexing
- Consolidation brings threats
  - Homogeneity and shared vulnerabilities
  - Centralization
- Problems are getting worse
  - Increasing desirability of targets: military, business, resource infrastructures, etc moving to clouds
  - Increasing power of attackers: governments, organized crime
- Commercial sector isn’t acting to protect against these threats

Today: Security of the MAC Layer

- How Ethernet works
  - Broadcast, Learning switches, Spanning Tree, ARP, VLANs
  - DHCP, HSRP/VRRP, Power over Ethernet
- Vulnerabilities and Countermeasures
  - LAN protocols were designed without security in mind
  - Automated tools (e.g., Yershina) bring these attacks to the hands of unskilled adversaries
- Securing L2 is important in itself
  - Applicability beyond data centers
  - First protocol-aware layer in stack
  - First line of defense against adversaries

Core LAN Protocols (Ethernet)

Overview of Ethernet

- Dominant wired LAN technology
  - Pretty much obsoleted token ring, optical LANs, ATM

- Defines a spectrum of techniques
  - Physical wiring, contention resolution (CSMA/CD), framing, encoding, devices (hubs/switches/bridges), forwarding, addressing

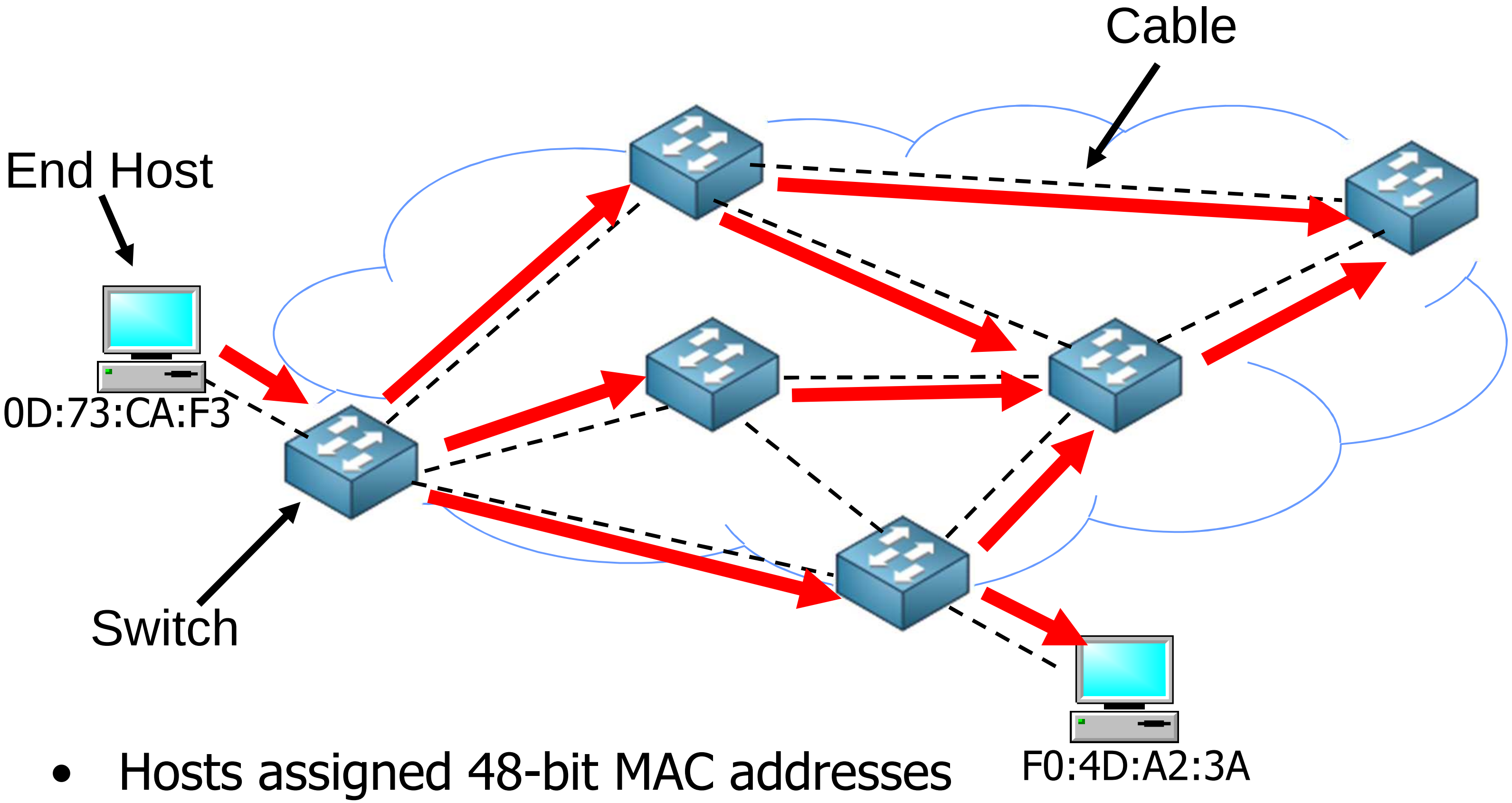
Overview of Ethernet

- Ethernet uses CSMA/CD
  - Carrier sense, collision detection, random access
- Limitations on Ethernet length
  - Need to ensure collisions are detected before sender is done transmitting a packet
- Frame structure
  - Preamble for synchronization

Overview of Ethernet

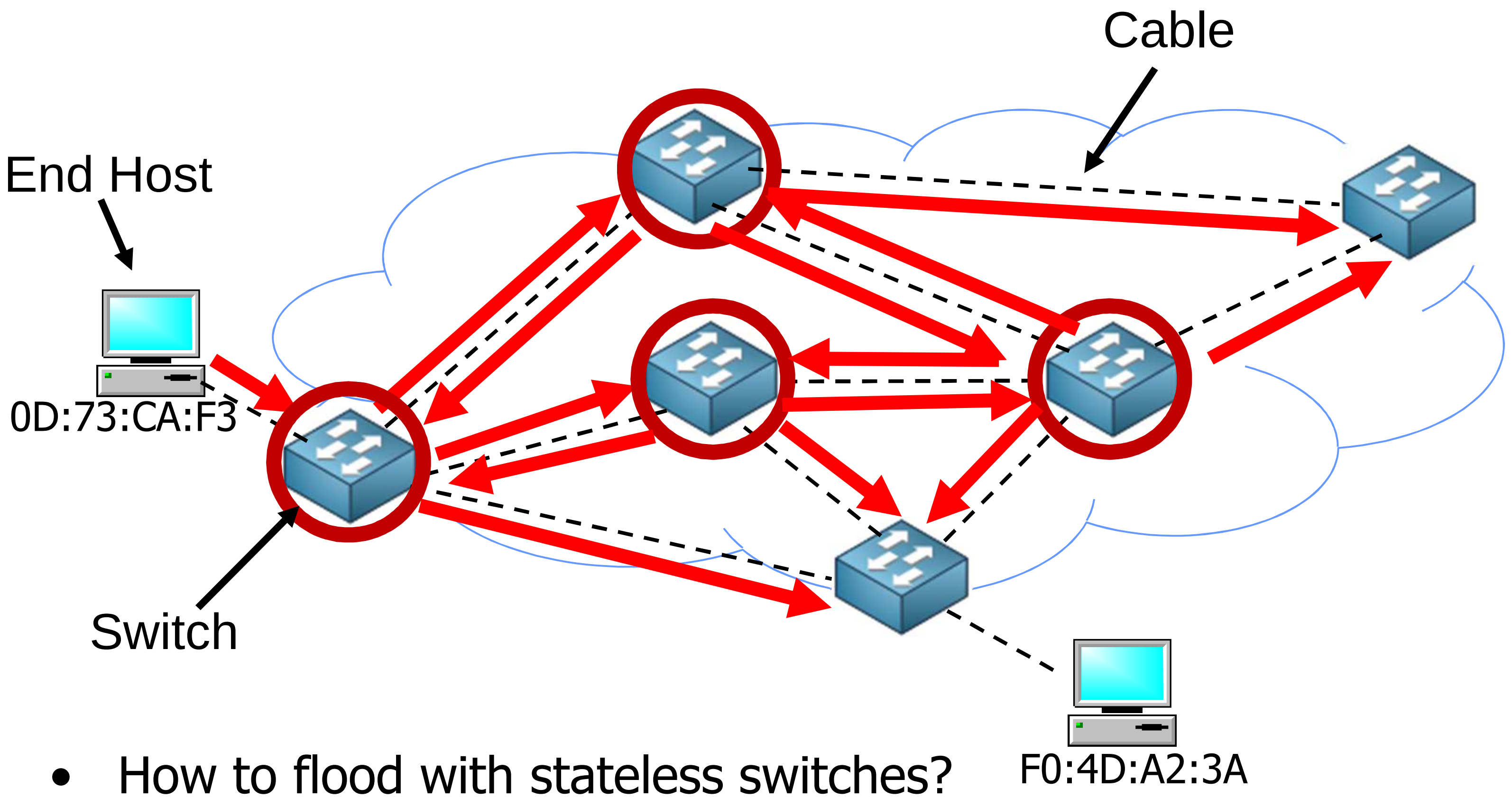
- Device types
  - Hubs: physical layer repeaters (obsolete?)
  - Switch: store and forward, breaks subnet into isolated LAN segments, learning
- Semantics: Unreliable, Connectionless
- Benefits: easy to administer and maintain, plug-and-play
- Downsides: scaling, security

Ethernet Forwarding



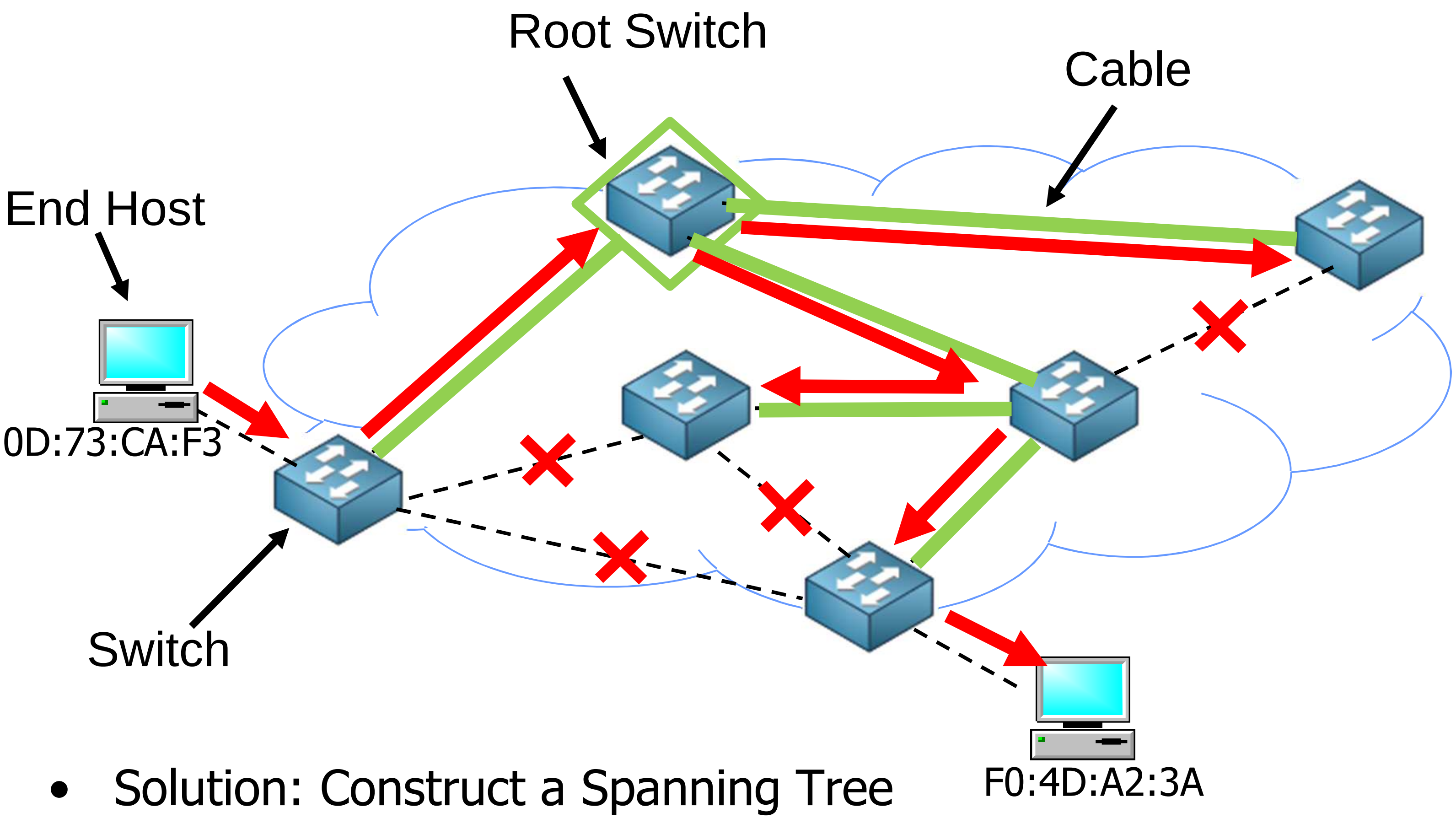
- Hosts assigned 48-bit MAC addresses
- Forwarding by “flooding”

Ethernet Forwarding



- How to flood with stateless switches?

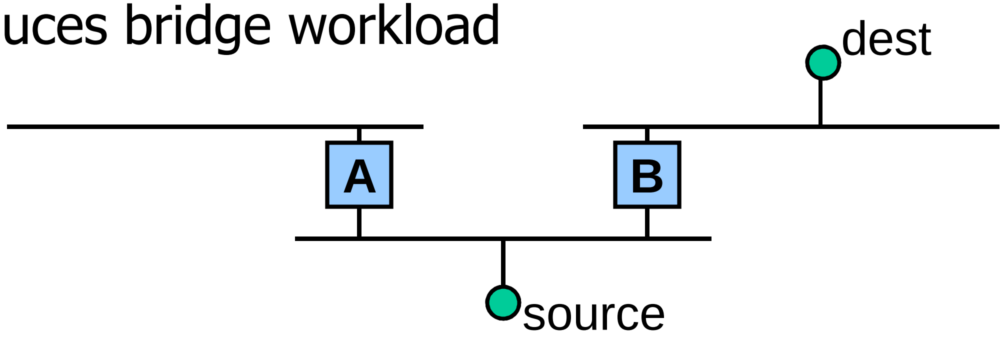
Ethernet Forwarding



- Solution: Construct a Spanning Tree
  - Elect a “root” switch
  - Root-facing ports are active, others disabled

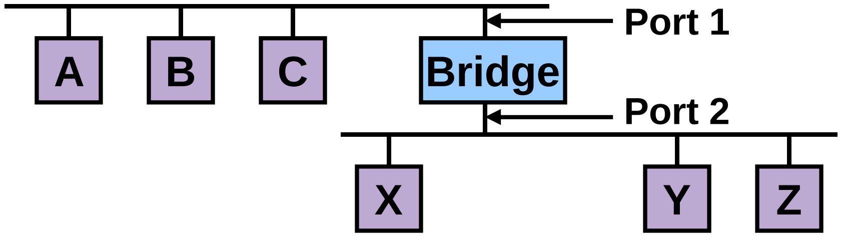
Learning Bridges

- Suppose source sends a frame to a destination
  - Which LANs should a frame be forwarded on?
- Trivial algorithm
  - Forward all frames on all (other) LAN’s
  - Potentially heavy traffic and processing overhead
- Optimize by using address information
  - “Learn” which hosts live on which LAN
  - Maintain forwarding table
  - Only forward when necessary
  - Reduces bridge workload



Learning Bridges

- Bridge learns table entries based on source address
  - When receive frame from A on port 1
    add A to list of hosts on port 1
  - Time out entries to allow movement of hosts
- Table is an "optimization", meaning it helps performance but is not mandatory
- Always forward broadcast frames

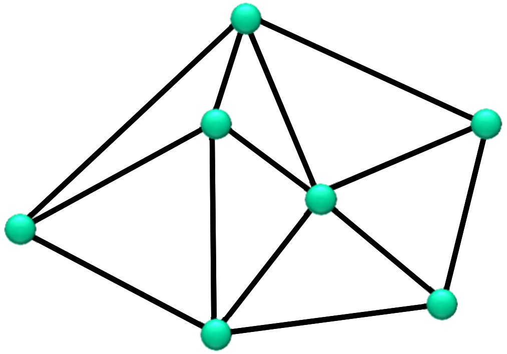


Host	Port
A	1
B	1
C	1
X	2
Y	2
Z	2

Virtualized Networking with VLANs

Network-wide broadcasts aren’t always desirable

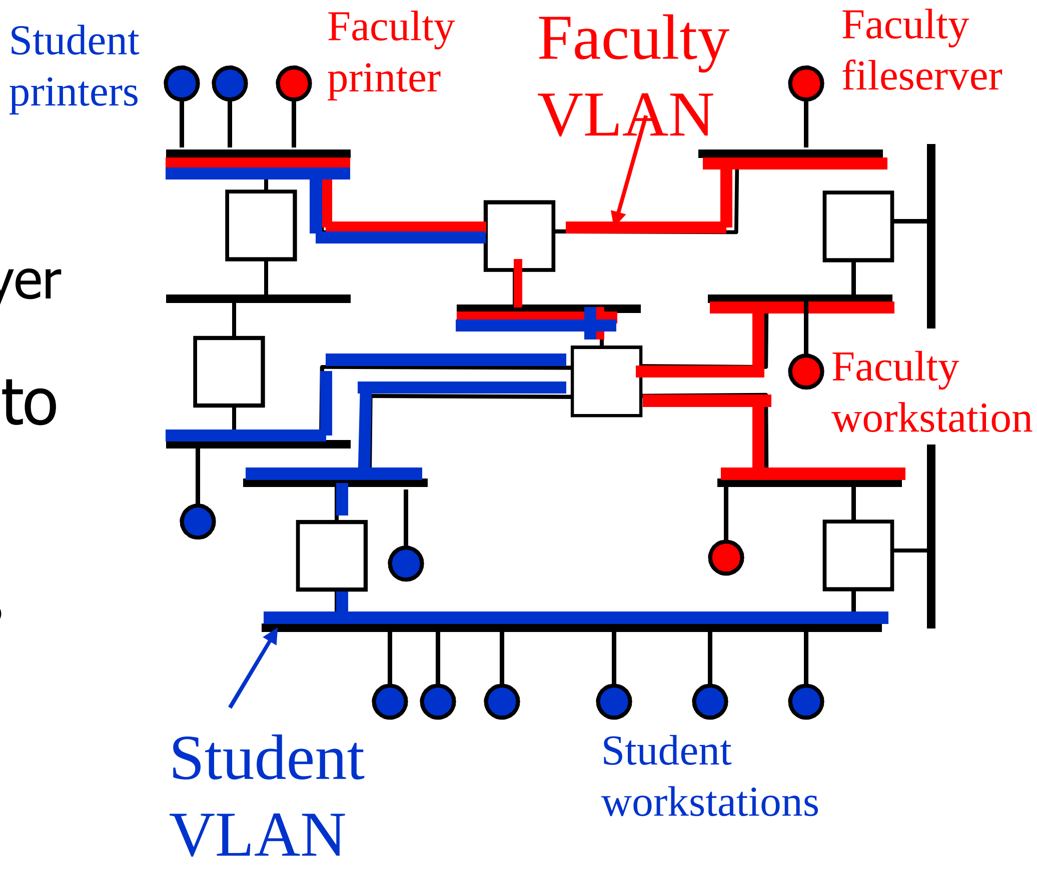
- Flooding packets throughout network introduces problems
  - Scalability, privacy, resource isolation, lack of access contro



- Scalability requirement is growing very fast
  - Large enterprises: 50k end hosts
  - Data centers: 100k servers, 5k switches
  - Metro-area Ethernet: over 1M subscribers

Scaling Ethernet with VLANs

- Divide up hosts into logical groups called **VLANs**
  - VLANs isolate traffic at layer 2
- Each VLAN corresponds to IP subnet, single broadcast domain
- Ethernet packet headers have VLAN tag
- Bridges forward packet only on subnets on corresponding VLAN

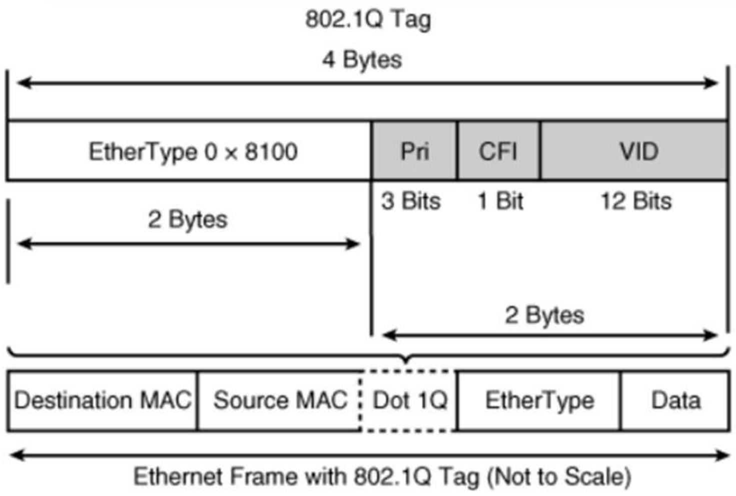


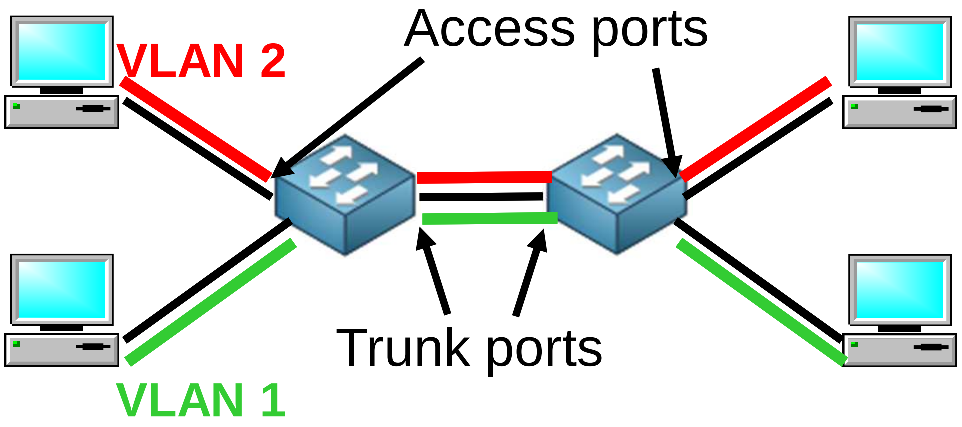
Virtual LANs

- Downsides of VLANs
  - Are manually configured, complicates network management
  - Hard to seamlessly migrate across VLAN boundaries due to addressing restrictions

- Upsides of VLANs
  - Limits scope of broadcasts
  - Logical separation improves isolation, security
  - Can change virtual topology without changing physical topology
    - E.g., used in data centers for VM migration

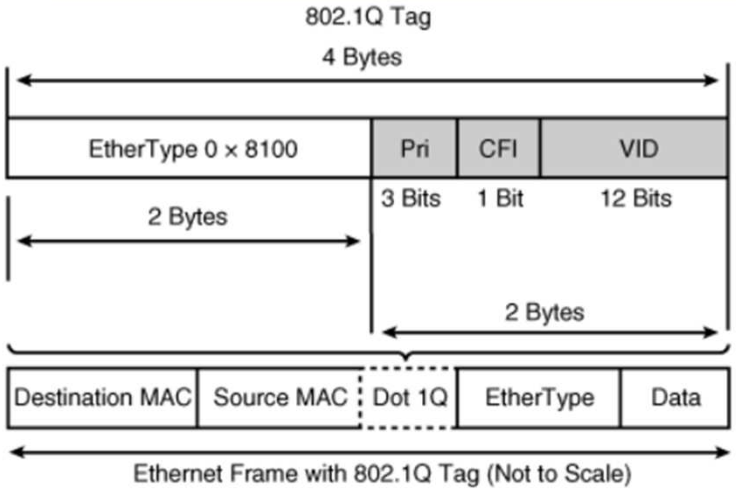
How VLANs are implemented

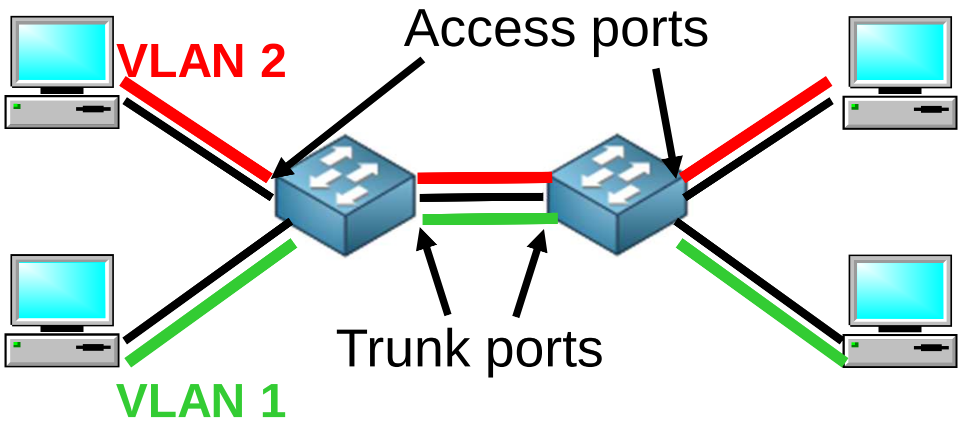




- Packets are annotated with 12-bit VLAN tags
  - Up to 4096 VLANs can be encapsulated within a single VLAN ID
- LAN switches can configure ports as access ports or trunk ports
  - Access ports append tags on packets
  - VLAN membership almost always statically encoded in access switch’s configuration file
  - Trunk ports can multiplex several VLANs

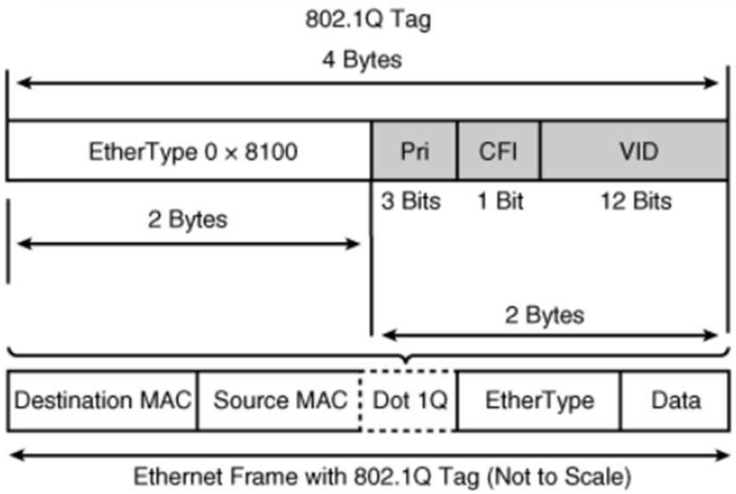
How VLANs are implemented

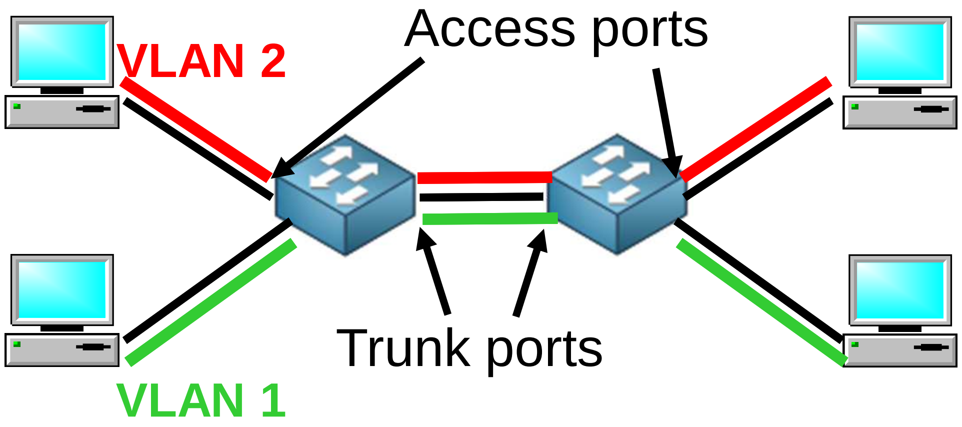




- 802.1Q (VLAN spec) defines a few other fields too
  - Ethertype of 0x8100 instructs switch to decode next 2 bytes as VLAN header
  - 3 bits of priority (like IP ToS)
  - 1 bit for compatibility with token ring
- What if 4096 VLANs isn't enough?
  - QinQ (802.1ad) – can encapsulate VLANs within VLANs by stacking VLAN tags
  - Up to 4096 VLANs can be multiplexed within a single VLAN ID→ 4096^2 combinations

How VLANs are implemented



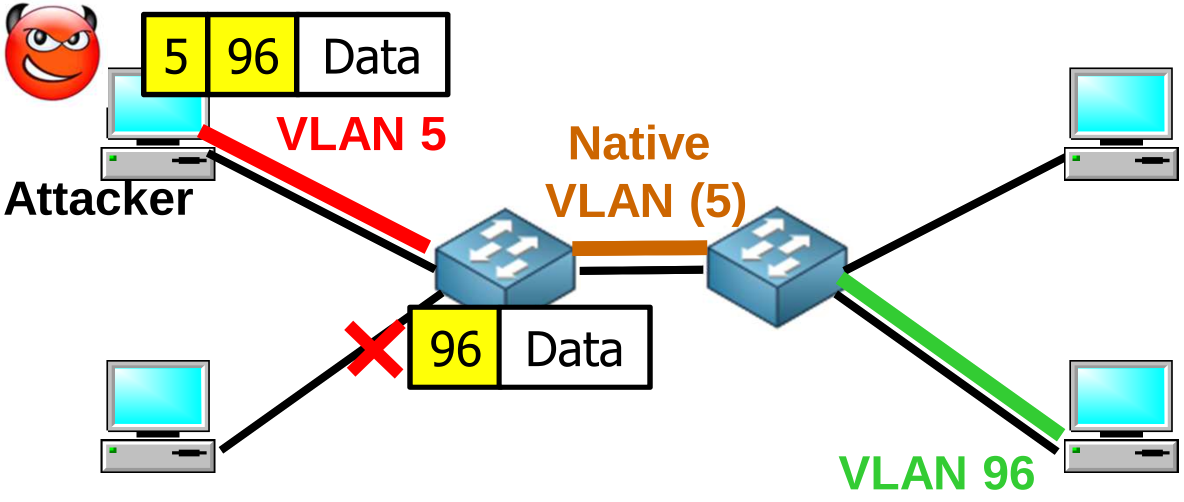


- Native mode
  - IEEE likes to make specs that are backwards compatible
  - 802.1Q allows trunk ports to carry both tagged and untagged frames
  - Frames with no tags are said to be part of the switch’s native VLAN

Attacks on VLANs

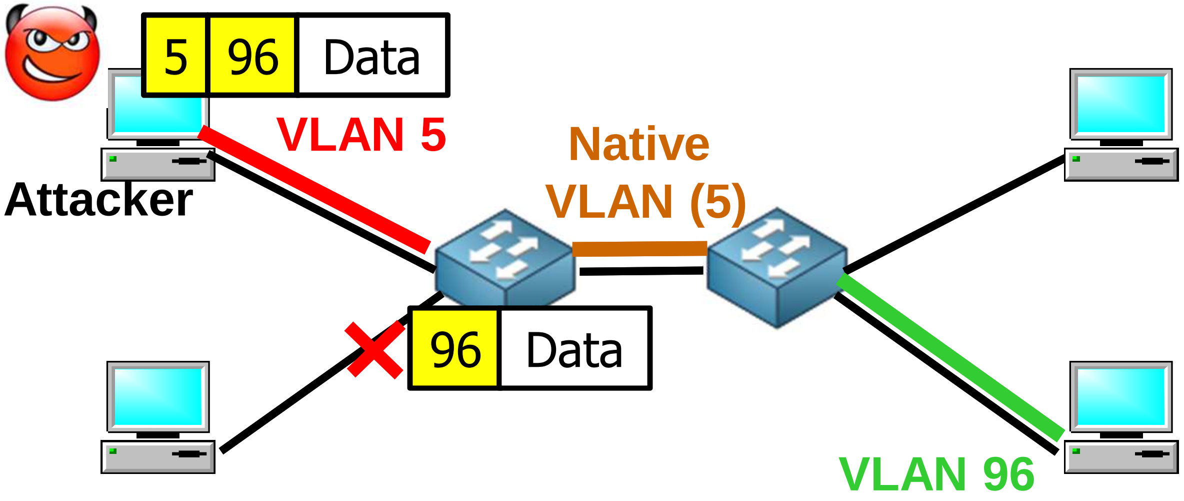
- VLANs are a very important building block for network security
  - Access control: hosts on one VLAN prevented, at layer 2, from reaching hosts on other VLANs
  - E.g., keep sensitive corporate records on a "private" VLAN
  - VLANs also provide resource isolation through QoS mechanisms
- Attack: VLAN hopping
  - Main idea: trick switches into forwarding attacker's packets onto the wrong VLAN
  - This could happen due to misconfigurations
    - Native VLANs make misconfigurations more prevalent)
  - Unfortunately, this could happen in networks without misconfigurations too

Nested VLAN Hopping (Tag Stack) Attack



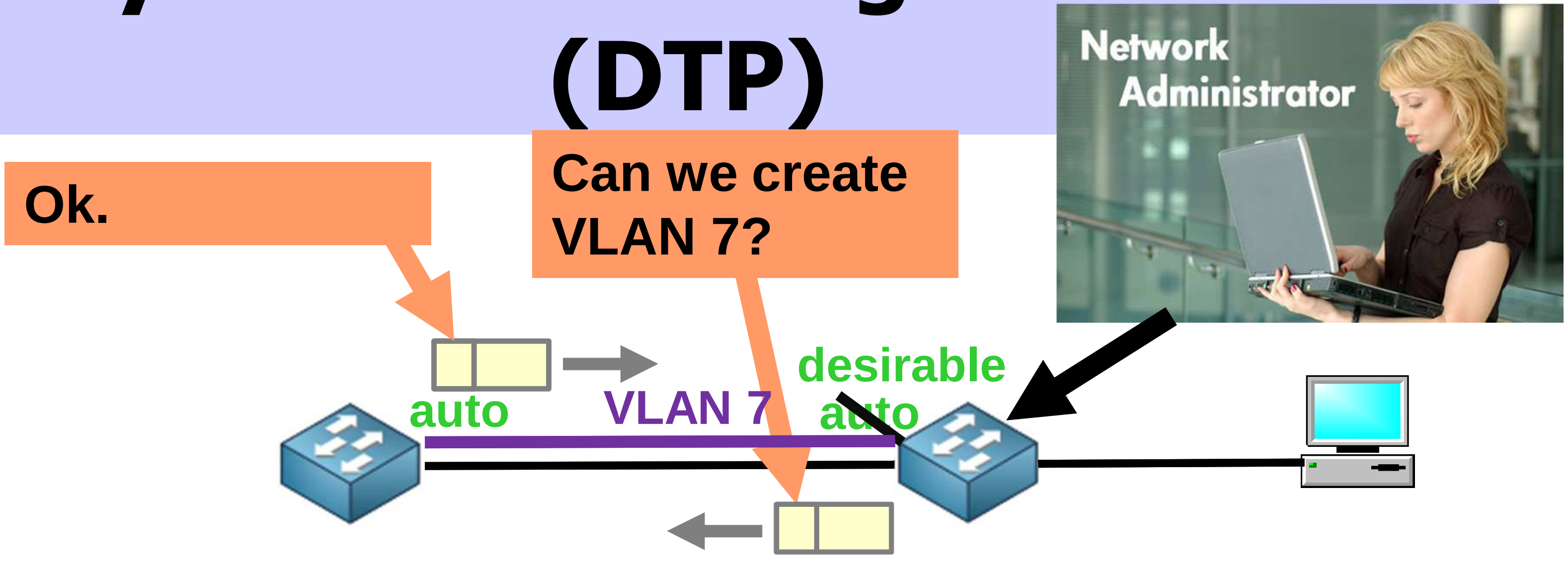
- Tag Stack Attack
  - In 802.11Q there is sometimes ambiguity about whether a tag is an internal tag or external tag
  - Adversary can “trick” switch by encapsulating a tag of the VLAN they want to hop to, and tricking a switch to decapsulating their correct VLAN tag
  - This attack is very difficult to trace

Nested VLAN Hopping: Countermeasures



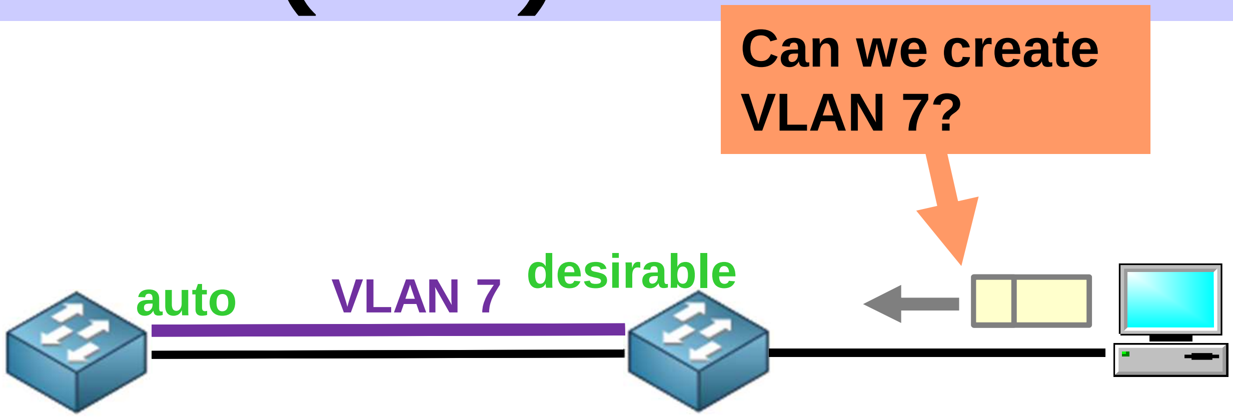
- Countermeasures:
  1. Ensure native VLAN is not assigned to any access port
  2. Clear the native VLAN from the trunk (not recommended)
  3. Force all traffic on the trunk to always carry a tag (preferred)
- Options 2 and 3 not available on all switches

Dynamic Trunking Protocol (DTP)



- Protocol to automate certain aspects of VLAN configuration
  - Determines whether two connected switches want to create a trunk
  - Automatically sets parameters such as encapsulation and VLAN range
- DTP transitions port through a set of states
  - Auto (port is willing to be trunked), On/Off (permanently forces link into/from trunking, even if neighbor disagrees), Desirable (attempts to make port a trunk; pursues agreement with neighbor)

Dynamic Trunking Protocol (DTP)



- Attack: Host can send DTP packets to switch, to trick it into joining itself into a VLAN
  - Countermeasure: do not leave user-facing ports in dynamic configuration mode
  - Hard code them as access ports and place them in a static VLAN

VLAN Trunk Protocol (VTP)

- Another protocol to automate VLAN configuration
- When you configure a VLAN on one switch, its information is disseminated via VTP to others
  - Eliminates need to manually configure each switch one by one
- You can configure different VTP modes
  - Server (can create VLANs), Client (can only receive configs), Transparent (just forward VTP advertisements), Off

VLAN Trunk Protocol (VTP)

- Attacks
  - Attacker can craft VTP packets, to disable a VLAN to do a DoS attack, enable a new VLAN across all switches to do a broadcast attack
  - Countermeasures: be careful to only enable VTP on trusted ports, use MD5 HMAC with shared key, have version numbers and only accept more recent copies (to mitigate replay attacks)

Spanning Tree Protocol

Overview of Spanning Tree Protocol (STP)

- Eliminates the possibility of forwarding loops by making the topology a tree (hierarchy)
- At the top of the tree is a root bridge
  - You want your root in the center of network as much as possible and to be a high-end device (why?)
  - Each switch has a "priority" (default=38464)
  - Lowest-priority switch becomes the root
  - If multiple switches have same priority, lowest MAC address becomes root (what's wrong with this?)
- Each switch disables (blocks) the port that is "furthest away" from the root
  - Each link has a "cost", which can (optionally) be automatically set based on link bandwidth
  - Automatically unblocks ports if necessary to recover from failure

Attacks on the Spanning Tree Protocol

- STP is trustful, stateless, and has no authentication mechanism
- STP is the foundation of most modern LANs
  - STP attacks are highly disruptive
  - Can lead to black holes, DoS, excessive flooding, hijacking of traffic, etc
- Automated tools (Yershina) bring attacks on STP to unskilled attackers

STP Attacks: Taking over as root bridge

- Taking over as the root bridge
  - Forces all traffic between two halves of network be sent to itself (MITM attacks), can cause major disruptions to ST
  - Attacker sends BPDU with same priority as root bridge (32767), but slightly lower numerical MAC address
    - Ensures a victory in root bridge selection process
  - Countermeasures:
    - Root guard: forces a particular port to be the desginated port. This enforces the position of the root bridge.
    - BPDU guard: prevents ports from processing BPDU traffic. Receipt of a BPDU disables the port. Not limited to root takeover attacks.

Attacks on the Spanning Tree Protocol

- DoS using Flood of Config BPDUs
  - BPDUs are processed in software
  - Yershina generates 25,000 BPDUs/sec on Pentium IV
    - Enough to bring a Catalyst 6500 to its knees, with 99% CPU utilization on the switch processor
    - Side effects: HSRP flapping
    - Hard to detect: STP doesn’t complain about excessive BPDU loads
- Countermeaures
  - BPDU guard
  - BPDU filtering
    - Yershina listens for real BPDUs to construct its fake ones
    - BPDU filtering discards incoming <u>and</u> outgoing
    - Potential to shoot yourself in the foot: enable on wrong port and loop conditions go undetected → you should only enable on end-station ports to be safe

Attacks on the Spanning Tree Protocol

- Simulating a dual-homed switch
  - Computer with two ethernet cards takes over as root bridge
  - Forces traffic to traverse attacker
- Countermeasure: BPDU guard

Defeating Switch Learning

Switch Learning Attacks

- Switch learning is what makes Ethernet scale

- Switch learning is what makes Ethernet private

- Two key attacks: MAC flooding and spoofing
  - Extremely simple to carry out, yet very potent
  - Can help attacker collect usernames/passwords, prevent proper operation of LAN, etc
  - Can turn a $50,000 switch into a $12 hub

Background on switch memory

Technology	Single chip density	$/MByte	Access speed	Watts/ chip
Dynamic RAM (DRAM) cheap, slow	64 MB	$0.50-$0.75	40-80ns	0.5-2W
Static RAM (SRAM) expensive, fast, a bit higher heat/power	4 MB	$5-$8	4-8ns	1-3W
Ternary Content Addressable Memory (TCAM) very expensive, very high heat/power, very fast (does parallel lookups in hardware)	1 MB	$200-$250	4-8ns	15-30W

- Vendors moved from DRAM (1980s) to SRAM (1990s) to TCAM (2000s)
- Vendors are now moving back to SRAM and parallel banks of DRAM due to power/heat

Limitations on switch memory

- High end switches can store hundreds of thousands of learning table entries
- What happens if learning table fills up?
- Depends on vendor
  - Most Cisco switches do not replace older entries with new ones
    - Need to “age out” entries (wait for them to time out)
  - Other switches circular buffer
    - Existing entries get overwritten

MAC Flooding Attack

- Problem: attacker can cause learning table to fill
  - Generate many packets to varied (perhaps nonexistant) MAC addresses
- This harms efficiency
  - Effectively transforms switch into hub
  - Wastes bandwidth, endhost CPU
- This harms privacy
  - Attacker can eavesdrop by preventing switch from learning destination of a flow
  - Causes flow’s packet to be flooded throughout LAN

MAC Spoofing Attack

- Host pretends to own the MAC address of another host
  - Easy to do: most ethernet adapters allow their address to be modified
  - Powerful: can immediately cause complete DoS to spoofed host
    - All learning table entries switch to point to the attacker
    - All traffic redirected to attacker
    - Can enable attacker to evade ACLs set based on MAC information

Switch Learning Attacks: Countermeasures

- Detecting MAC activity
  - Many switches can be config’d to warn administrator about many sudden MAC address moves
- Port Security
  - Ties a given MAC address to a port
  - On violation, can drop frames, disable port for specified duration, signal alarm, increment violation counter

Switch Learning Attacks: Countermeasures

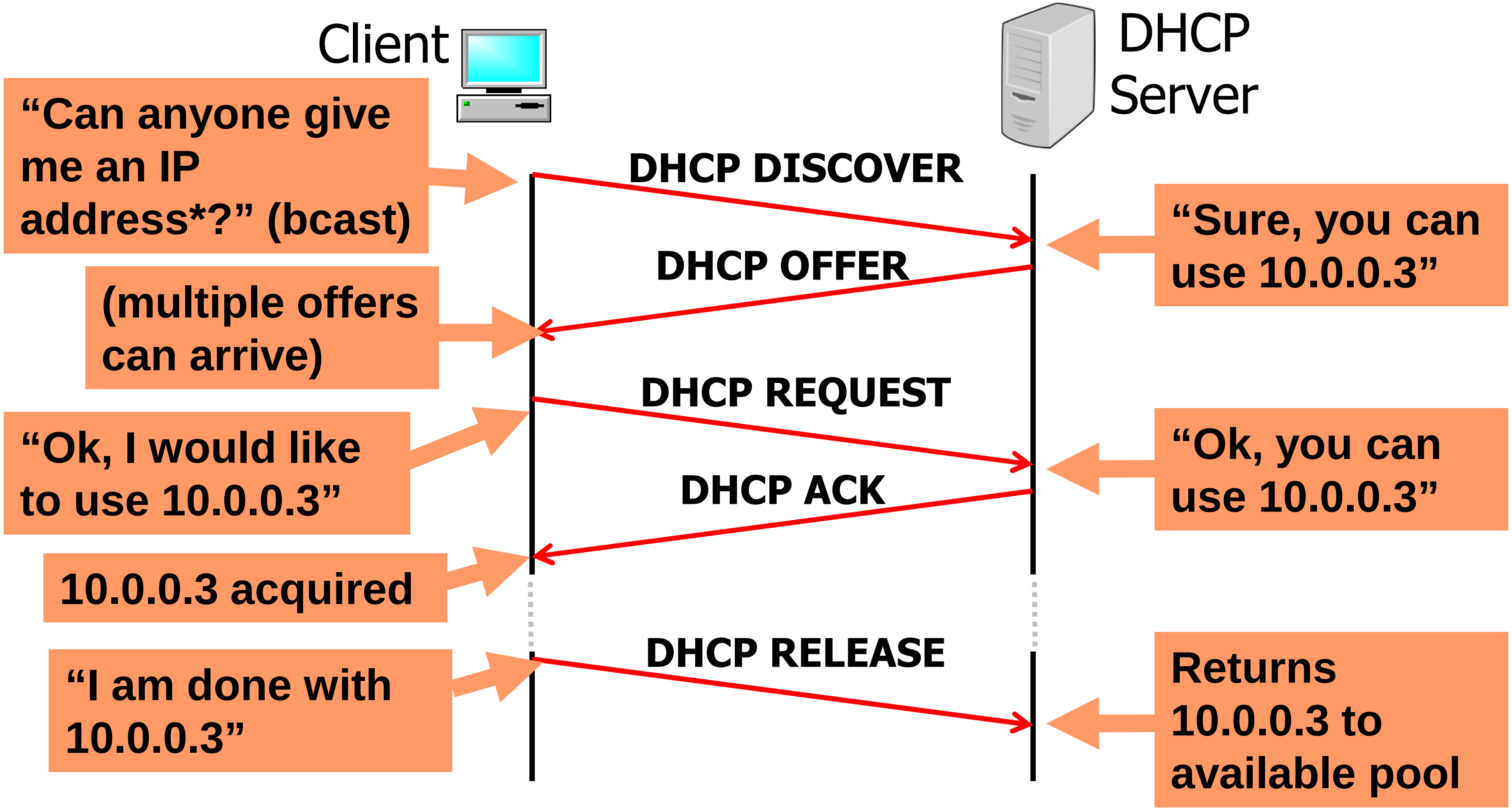
- Unicast Flooding Protection
  - Send alert when user-defined rate limit is exceeded
  - Can also filter traffic or shut down port generating excessive floods

Attacks on Addressing

Dynamic Host Configuration Protocol (DHCP)

- Automatically configure hosts
  - Assign IP addresses, DNS server, default gateway, etc.
  - Client listen on UDP port 68, servers on 67
- Very common LAN protocol
  - Rare to find a device that doesn’t support it
- Address is assigned for a lease time

Dynamic Host Configuration Protocol (DHCP)



Client

DHCP Server

“Can anyone give me an IP address*?” (bcast)

DHCP DISCOVER

“Sure, you can use 10.0.0.3”

DHCP OFFER

(multiple offers can arrive)

DHCP REQUEST

“Ok, I would like to use 10.0.0.3”

“Ok, you can use 10.0.0.3”

DHCP ACK

10.0.0.3 acquired

DHCP RELEASE

“I am done with 10.0.0.3”

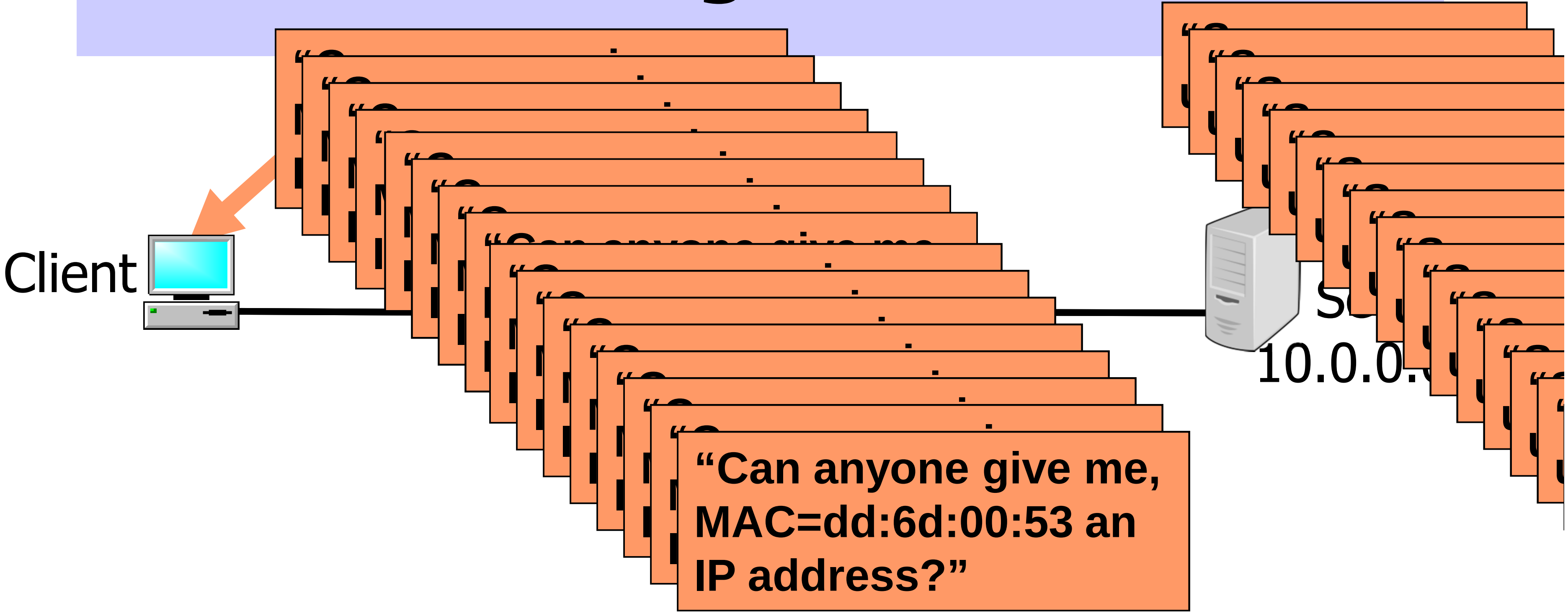
Returns 10.0.0.3 to available pool

*and other config information

Attacks on DHCP

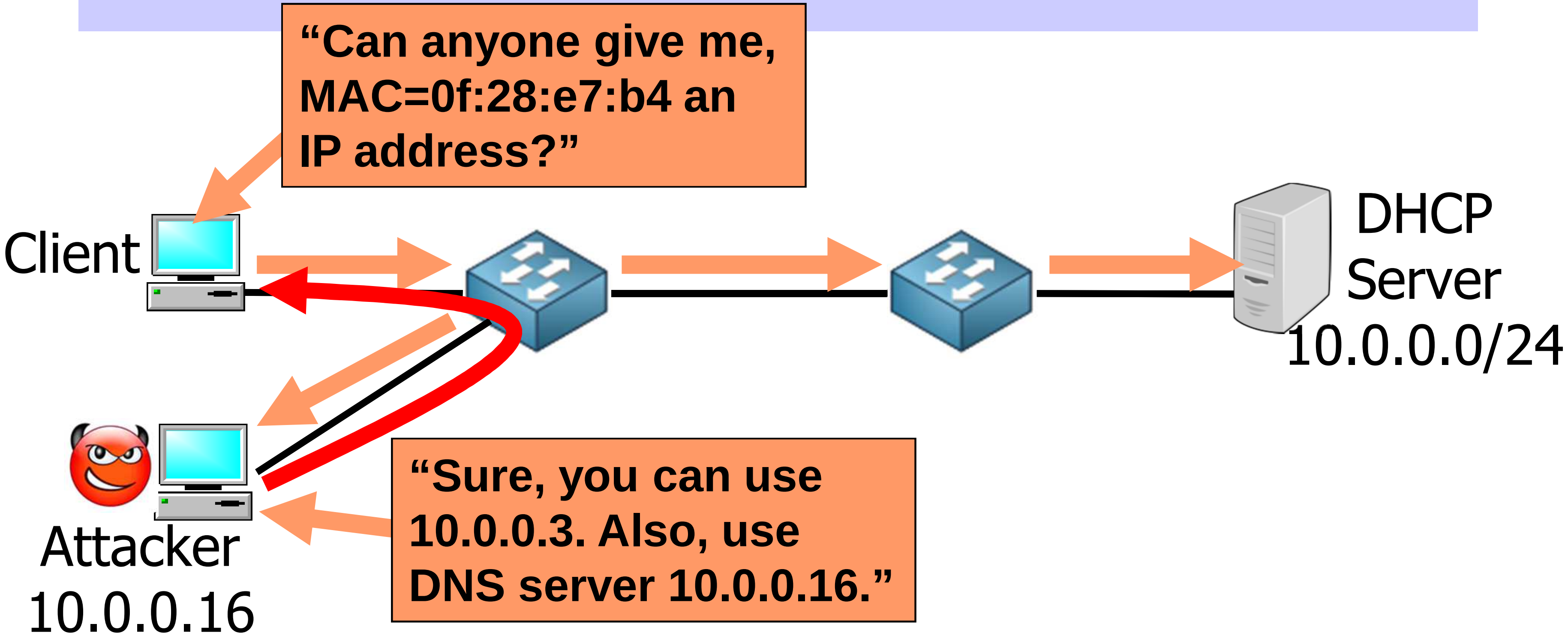
- Unfortunately, DHCP was designed without security in mind
  - Whoever requests an address is free to receive one
  - No authentication fields or any other security-inclined information in protocol

Attacks against DHCP



- DHCP Scope Exhaustion
  - Malicious client attempts to seize entire range of IP addresses
  - When legitimate client tries, it is abandoned with no IP connectivity

Attack: Rogue DHCP Server



“Can anyone give me, MAC=0f:28:e7:b4 an IP address?”

Client

DHCP Server 10.0.0.0/24

Attacker 10.0.0.16

“Sure, you can use 10.0.0.3. Also, use DNS server 10.0.0.16.”

- Installation of a Rogue DHCP Server
  - – Client uses offeror of prevoiusly-used IP address, if none then uses first-received response
    - Rogue can compromise all clients “near” itself

Countermeasures to DHCP Attacks

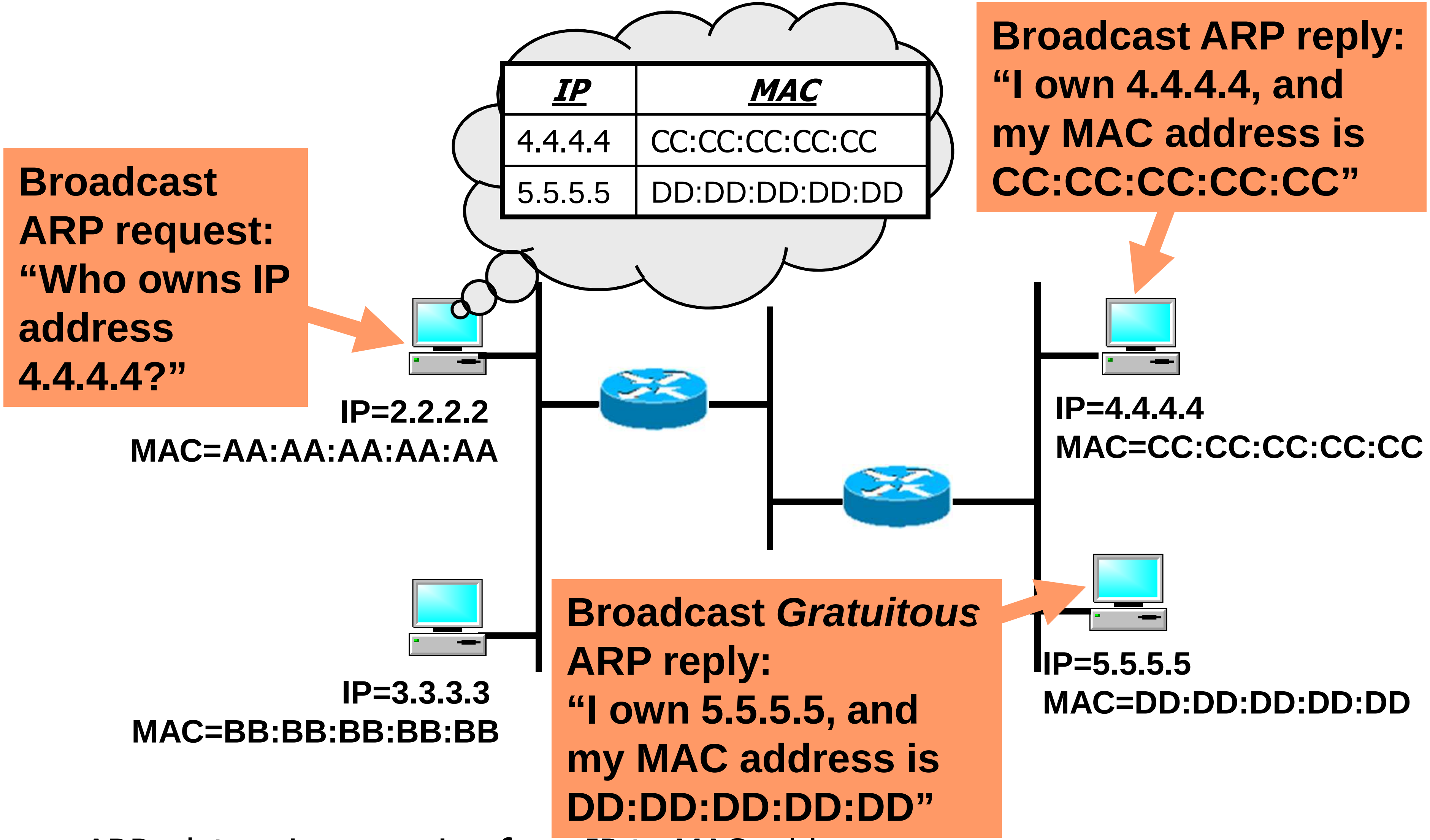
- Limit number or set of MAC addresses per port
  - This is called Port Security
  - Limit can be set manually or switch can be intructed to lock down on first dynamically learned address
- Limitations
  - DHCP lets you request multiple IP addresses from a single MAC address
  - DHCP lease time is usually several days but port-security timers are usually order of minutes
    - Attacker can change its MAC address slowly

Countermeasures to DHCP Attacks

- Prevent hosts from generating certain DHCP messages (DHCP Snooping)
  - Like a stateful firewall for DHCP
  - Runs on router's central management processor, to do deep packet inspection
  - Learns IP-to-MAC bindings by snooping on DHCP packets
  - Rules:
    - If port is connected to host, don't allow DHCPOFFER and DHCPACK packets
    - Don't allow DHCP packets that don't match learned bindings
    - Can also rate-limit DHCP messages per port, etc

Address Resolution Protocol (ARP)

- Networked applications are programmed to deal with IP addresses
- But Ethernet forwards to MAC address
- How can OS know the MAC address corresponding to a given IP address?
- Solution: Address Resolution Protocol
  - Broadcasts ARP request for MAC address owning a given IP address



IP	***MAC***
4.4.4.4	CC:CC:CC:CC:CC
5.5.5.5	DD:DD:DD:DD:DD

Broadcast ARP request: “Who owns IP address 4.4.4.4?”

Broadcast ARP reply: “I own 4.4.4.4, and my MAC address is CC:CC:CC:CC:CC”

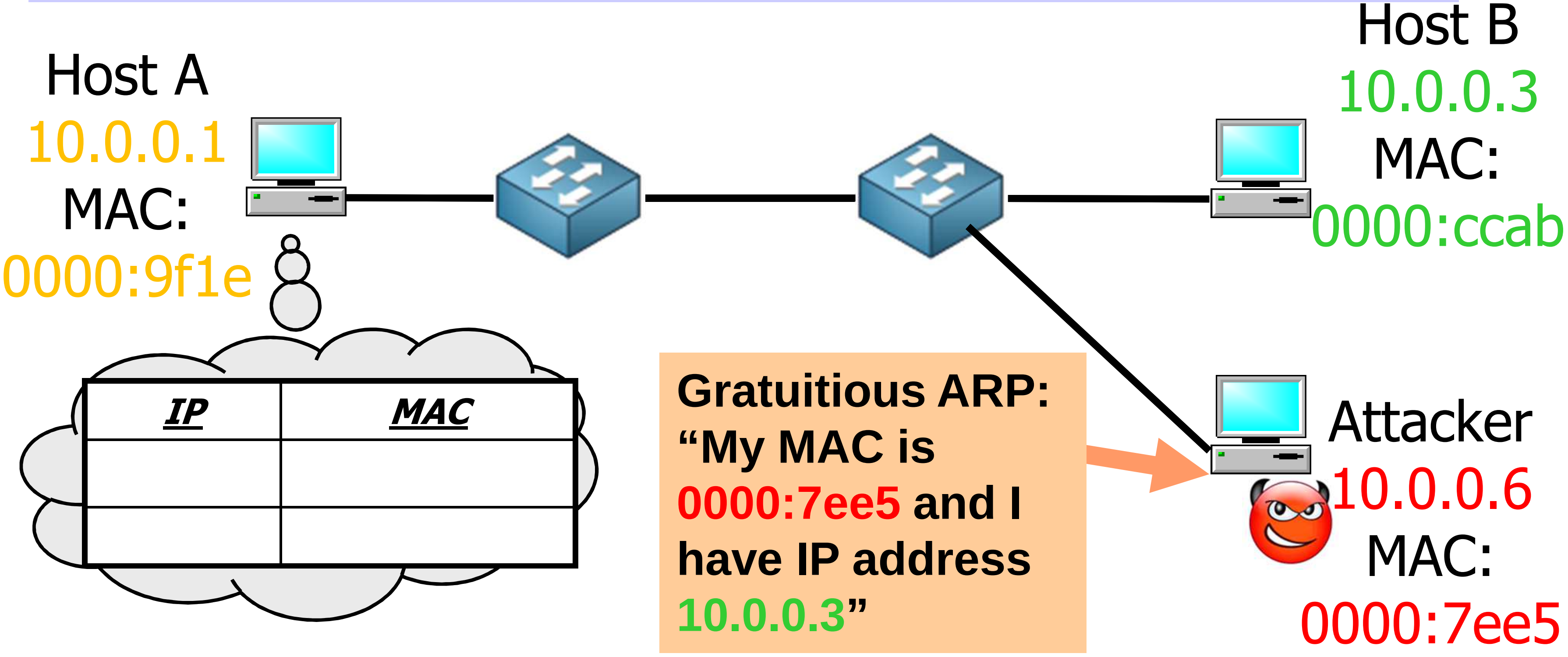
Broadcast *Gratuitous* ARP reply: “I own 5.5.5.5, and my MAC address is DD:DD:DD:DD:DD”

- ARP: determine mapping from IP to MAC address
- What if IP address not on subnet?
  - Each host configured with “default gateway”, use ARP to resolve its IP address
- Gratuitous ARP: tell network your IP to MAC mapping
  - Used to detect IP conflicts, IP address changes; update other machines’ ARP tables, update bridges’ learned information

Risk Analysis for ARP

- No authentication
  - Hosts do not sign ARP replies
- Information leak
  - All hosts in same VLAN learn the advertised <IP,MAC> mapping
  - All hosts discover querying host wishes to communicate with replying host
- Availability
  - All hosts on same LAN receive ARP request, must process it in software
  - Attacker could send high rate of spurious ARP requests, overloading other hosts

ARP Spoofing Attack



Host A
10.0.0.1
MAC:
0000:9f1e

IP	*MAC*

Host B
10.0.0.3
MAC:
0000:ccab

Gratuitious ARP: "My MAC is 0000:7ee5 and I have IP address 10.0.0.3"

Attacker
10.0.0.6
MAC:
0000:7ee5

- Attacker sends fake unsolicited ARP replies
  - Attacker can intercept forward-path traffic
  - Can intercept reverse-path traffic by repeating attack for source
  - Gratuitious ARPs make this easy
  - Only works within same subnet/VLAN

Countermeasures to ARP Spoofing

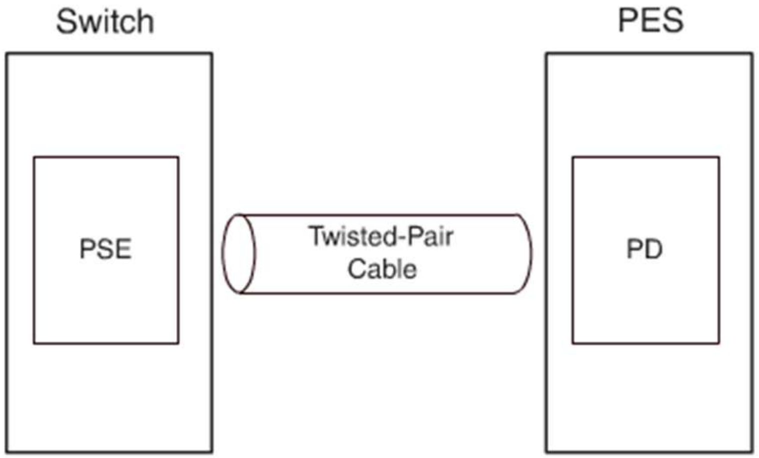
- Ignore Gratuitious ARP
  - Problems: gratuitious ARP is useful, doesn't completely solve the problem
- Dynamic ARP Inspection (DAI)
  - Switches record <IP,MAC> mappings learned from DHCP messages, drop all mismatching ARP replies
- Intrusion detection systems (IDS)
  - Monitor all <IP,MAC> mappings, signal alarms

Other Countermeasures

- Availability attacks
  - Control Plane Policing: rate-limit ARP messages sent to switch/host control planes
- Information leaks
  - No great solution
  - VLANs help

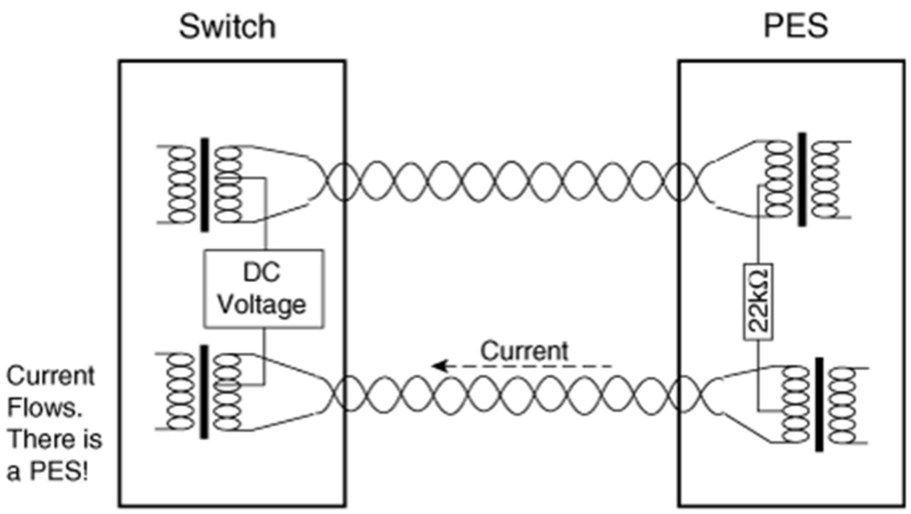
Attacks on
Power over Ethernet (PoE)

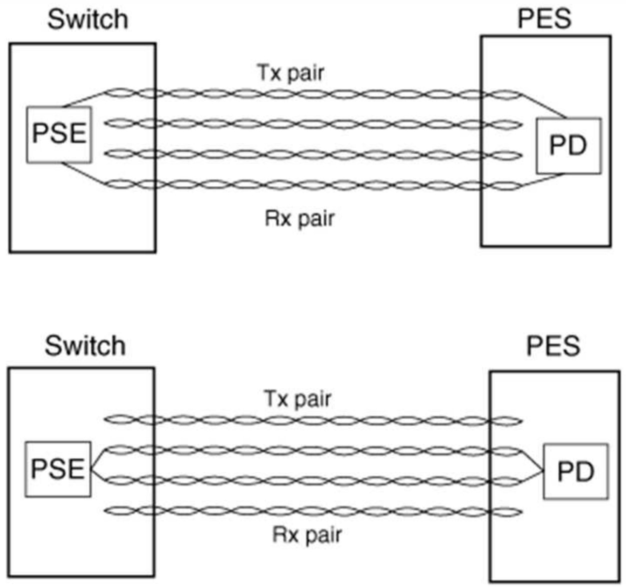
Power over Ethernet (IEEE 802.3af)



- Ethernet switch can provide power to attached stations, over Ethernet cable
- Eliminates need for separate cable
  - 12-45 V of galvanically isolated power
  - Improved economy and safety

Power over Ethernet





- Detection:
  - Apply voltage and see if resistance is between 19kΩ and 26.5kΩ
  - Device can send CDP packets to adjust voltage

- Powering:
  - Apply DC power
  - Switch has finite power limit
    - 600W limit means it can only power forty 15-Watt IP phones

Power over Ethernet: Attacks

- Power gobbling: Unauthorized devices connect and request so much power none is left for PES
- Power changing: Unauthorized device spoofs CDP packet requesting power decrease, shutting down PES
- Burning: Spoofs CDP to increase power, overloading PES
- Shutdown: Disabling switch disables power to PES

Countermeasures

- Power gobbling attacks
  - Static configuration of which ports can request power, and how much power they can request
- Burning, power-changing attacks
  - No easy way to mitigate
  - Can sometimes disable CDP
- Shutdown attacks
  - Add uninterruptable power supply to switches

Resilient Topology Design

Today's lecture: Internet topology

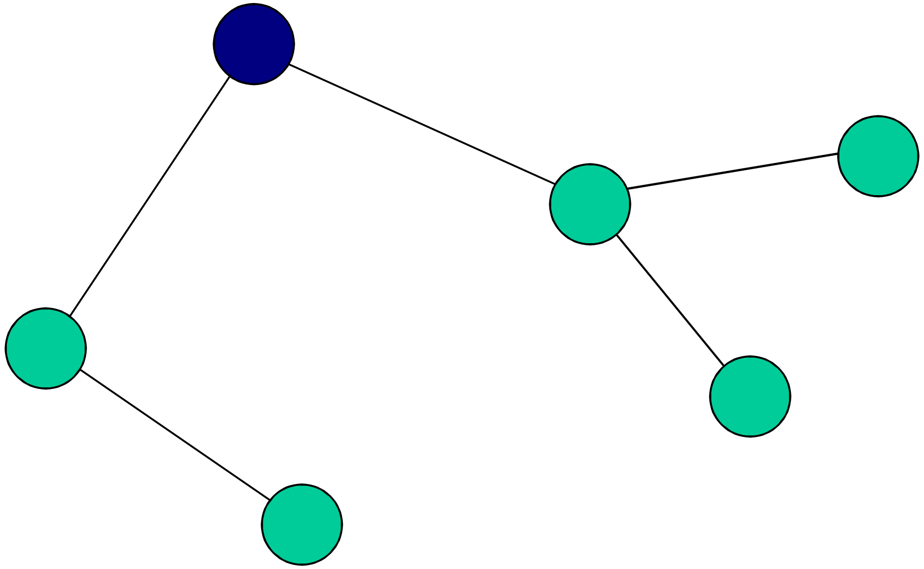
- How should I design my network's topology?
- What is the network topology of the Internet?
  - How can we measure the Internet topology?
- This lecture:
  - Preliminaries (Network elements: router/link design
  - Designing the topology (Hub-and-spoke, backbones, provider/peering

Today’s lecture: Internet topology

- Modeling the topology
  - Graph-based characterizations
- Measuring the topology
  - Traceroute probes, locating IP addresses

Problem Statement

Sender / Source



Build Network
(1) Low latency
(2) Low cost

Many Receivers

What is a node?

Links

Fibers

Coaxial Cable

Interfaces

Ethernet card

Wireless card

Switches/routers

Large router

Telephone switch

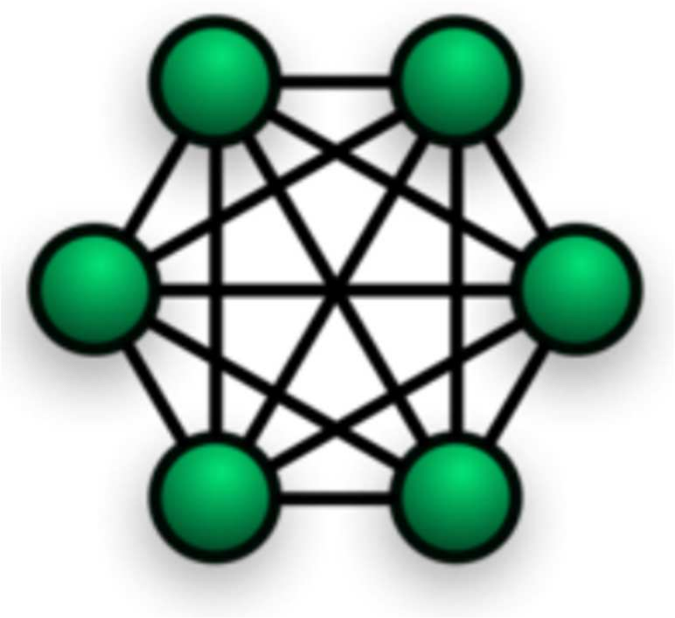
Formal Statement

- Given a graph G=(V,E)
- Each edge has c(e) and l(e)
- Each vertex has demand d(v)
- Compute graph such that
  - Minimize total c(e) of e $\in$ E
  - Minimize l(e) along (src,dst) paths

One approach: Optimization algorithms

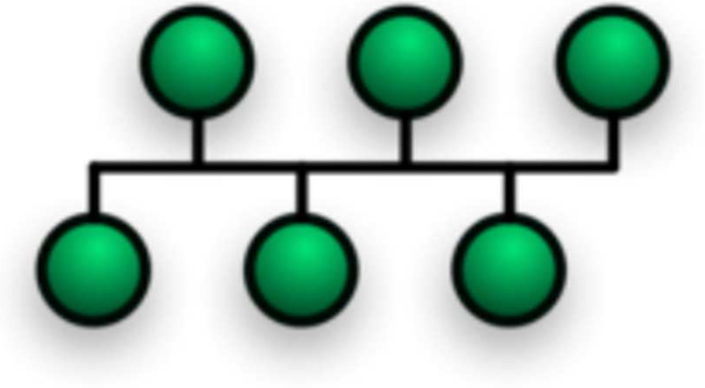
- Find value x such that f(x) is as large as possible
  - Linear/nonlinear convex/nonconvex optimization
  - Facility location problem
- Marathe et al, 1998
  - Bicriteria optimization of total c(e), max l(e)
  - Factors (log n, log n) where n=|D|
- Meyerson et al, 2000
  - Optimizes sum of c(e) + d(v)l(v→s)
  - Factor log n where n=|D|
- Various other results assuming c(e) and l(e) are somehow related

Fully connected topology



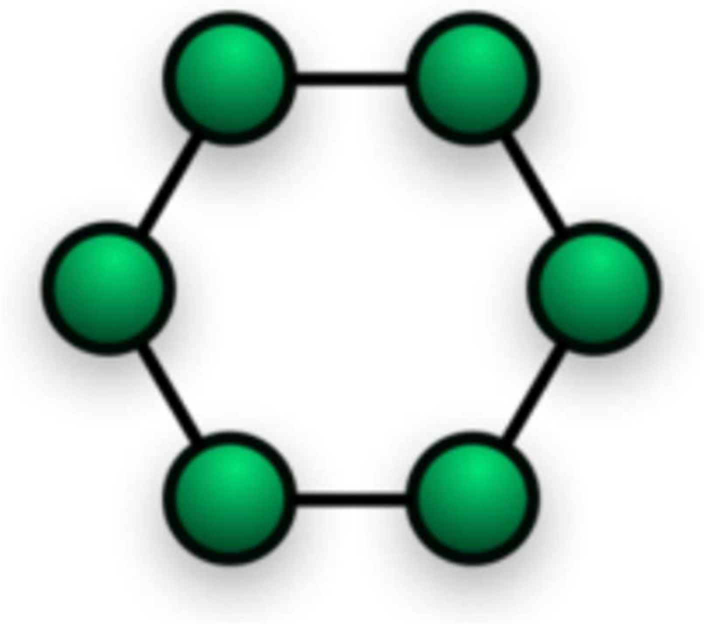
- All nodes connected to each other
- Doesn’t need switching or broadcasting
- However, number of connections grows quadratically with number of nodes

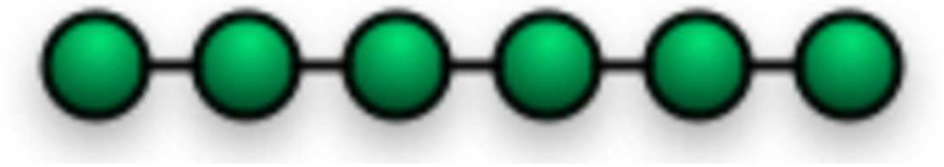
Bus topology



- All nodes connected to a single, shared cable
- Modern Ethernets are “logical” buses (hubs help propagate signal)
- Simple to manage, cost effective, easy to identify faults, reduced weight
- However, poor fault tolerance, performance low with heavy traffic, termination required

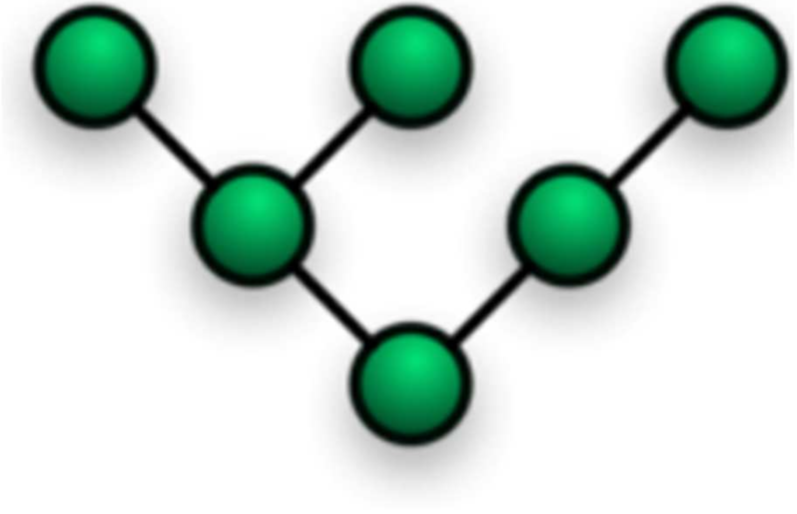
Ring and Daisy-chain topology





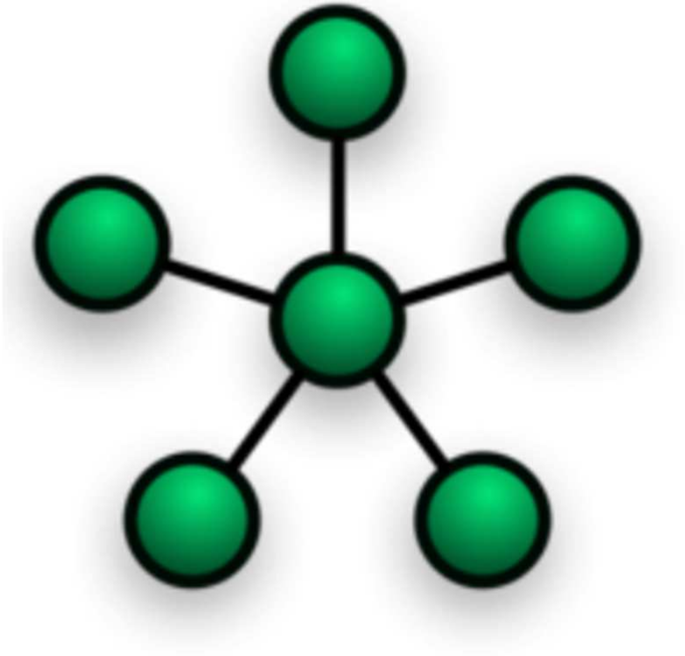
- Outperforms bus networks, simple to manage
- Ring networks can reduce number of transmitters by half, and can double resilience as compared to daisy chain
- Can pass around “token” to take turns transmitting

Tree topology



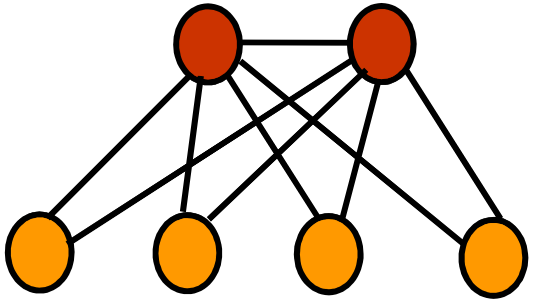
- Can exploit statistical aggregation
- Layout may follow physical/administrative constraints
- But, can be bottleneck at root
- Solution: “FAT Tree”
  - Increase bandwidth on links near the root

Hub-and-spoke topology

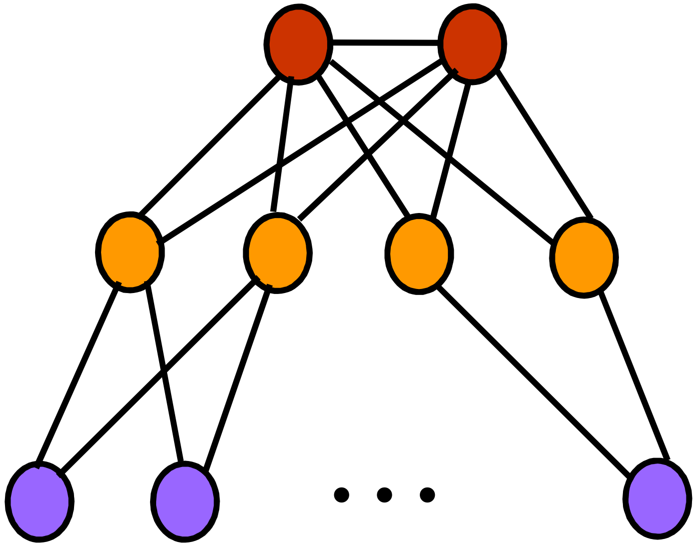


- Single hub node
- Common in enterprise networks
- Main location and satellite sites
- However, single point of failure, bandwidth limitations, high delay between sites, costs to backhaul and hub
- How can we improve upon hub and spoke?

Improvements to hub-and-spoke



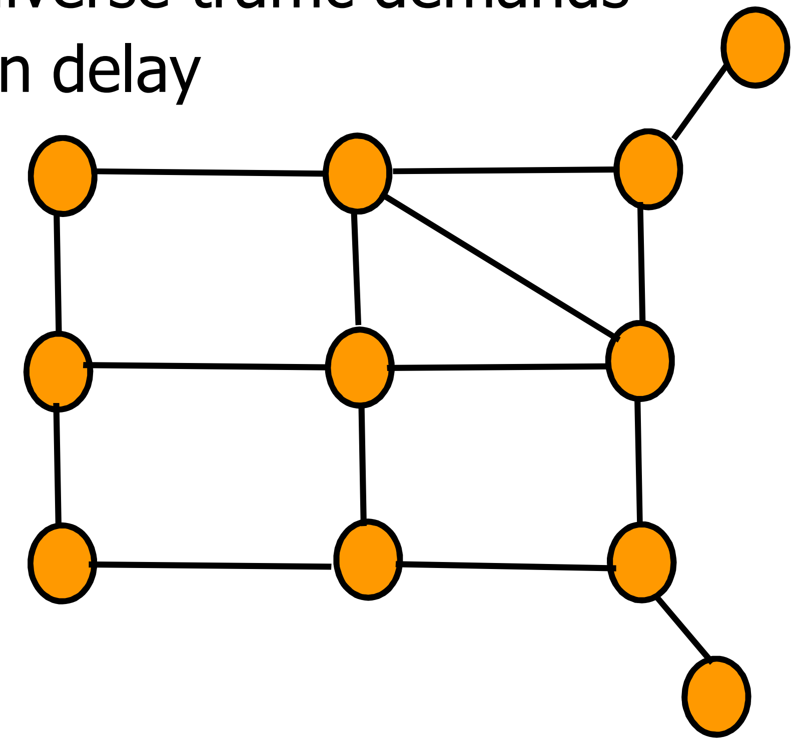
- Dual hub-and-spoke
  - Higher reliability
  - Higher cost
  - Good building block



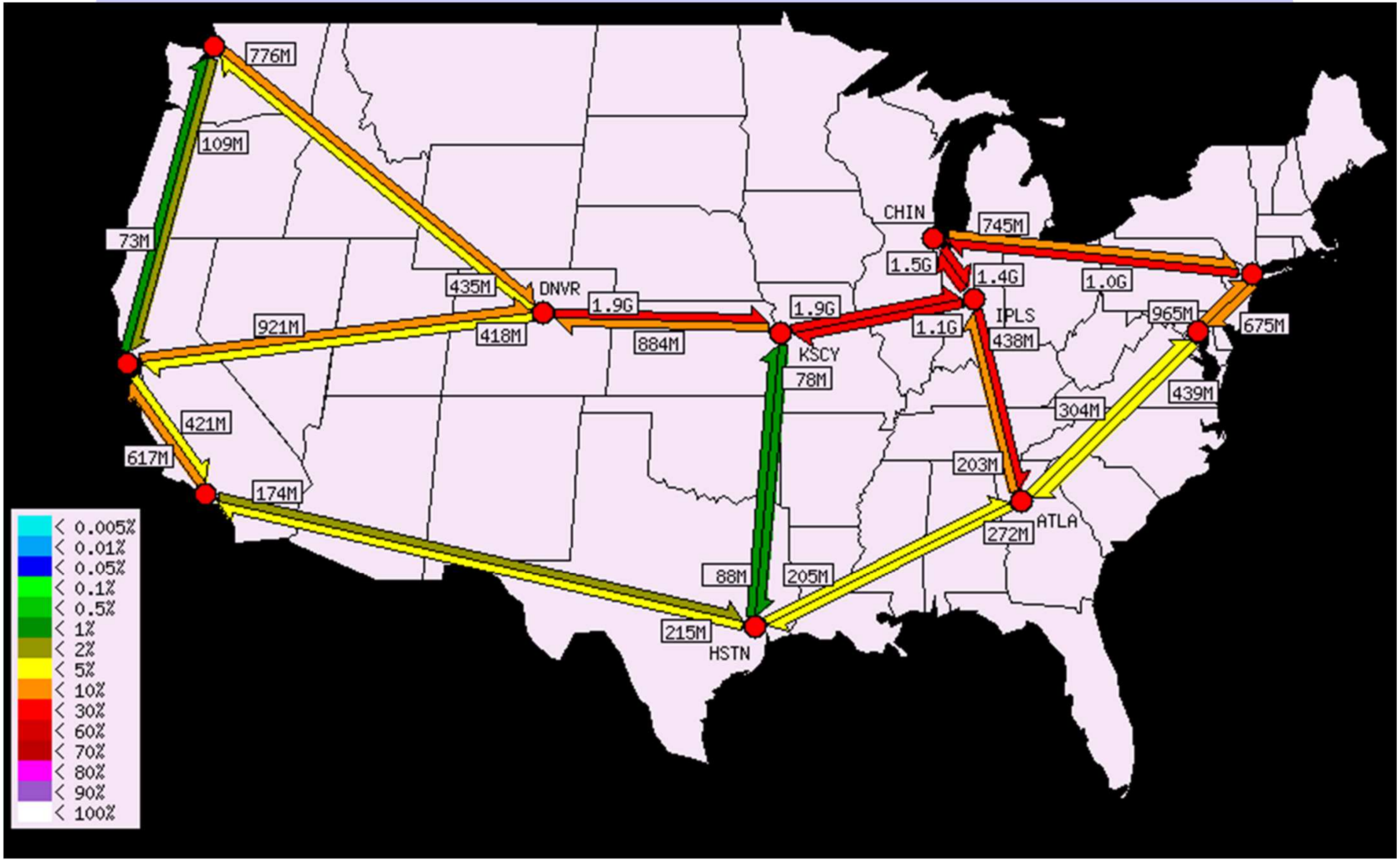
- Levels of hierarchy
  - Reduce backhaul cost
  - Aggregate the bandwidth
  - Shorter site-to-site delay

Backbone Networks

- Backbone networks
  - Multiple Points-of-Presence (PoPs)
    - Each with (easily) 40 routers
  - Lots of communication between PoPs
  - Need to accommodate diverse traffic demands
  - Need to limit propagation delay

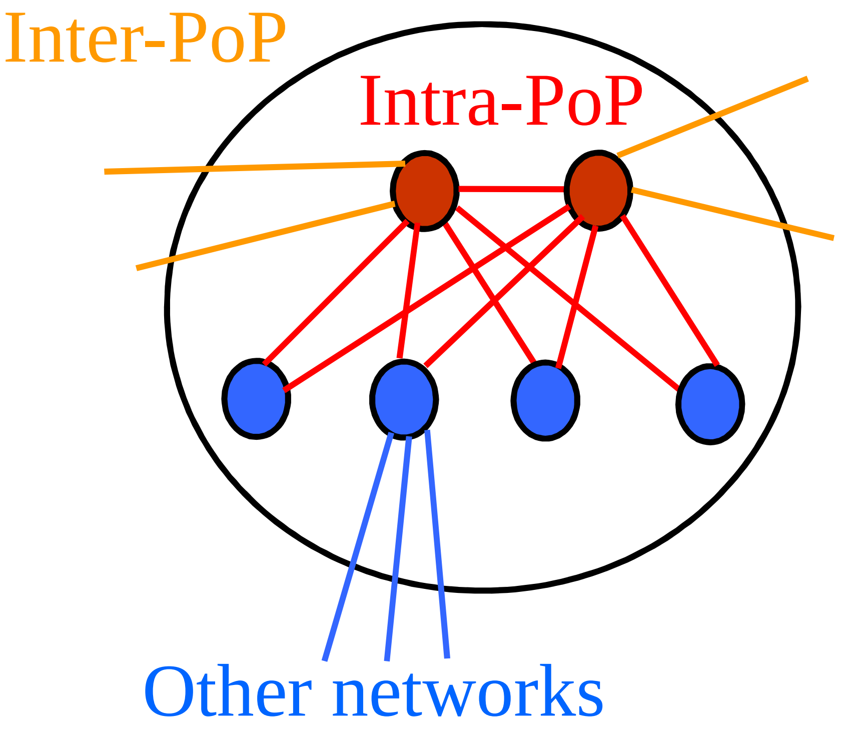


Abilene Internet2 Backbone



Points-of-Presence (PoPs)

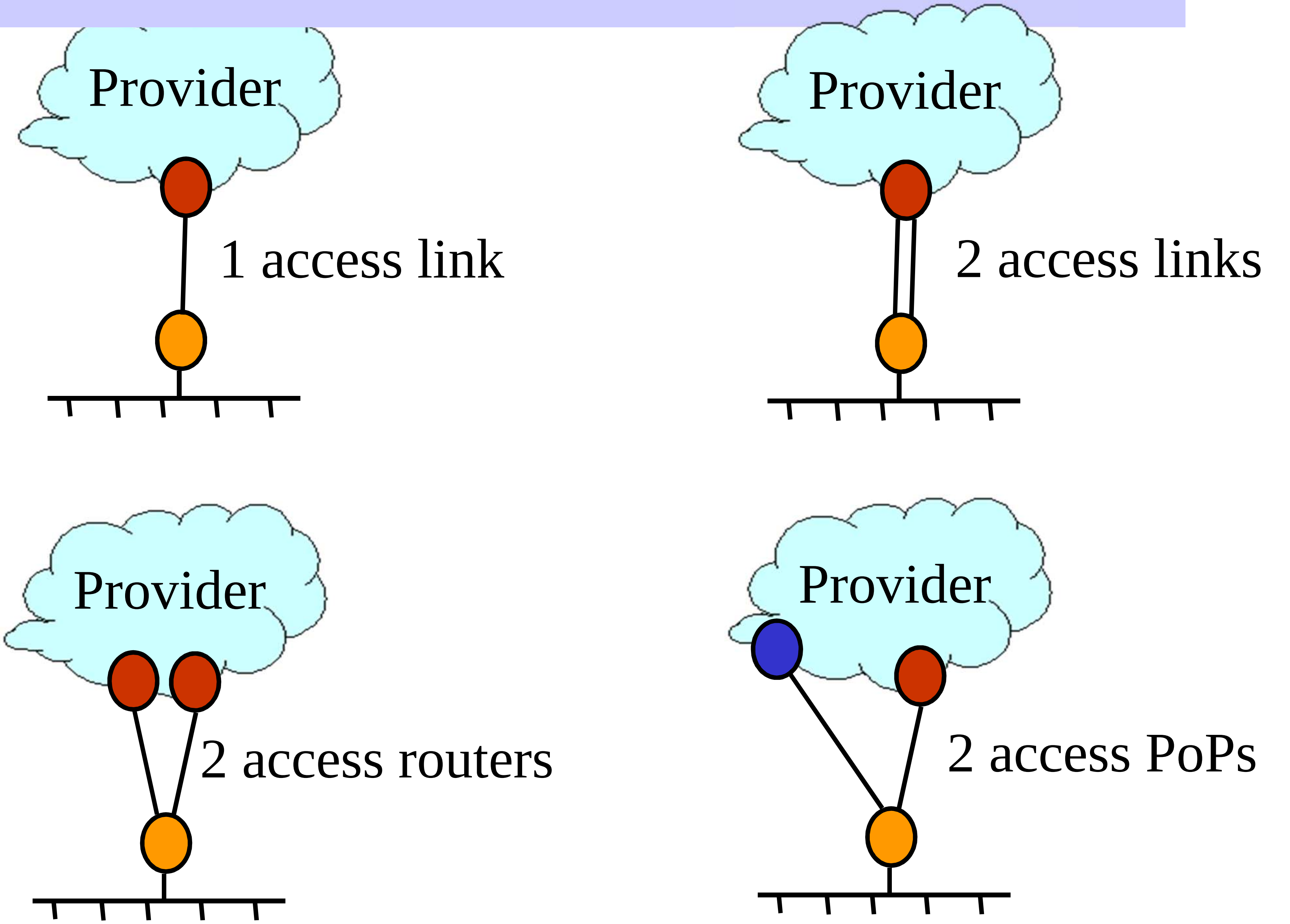
- Inter-PoP links
  - Long distances
  - High bandwidth
- Intra-PoP links
  - Short cables between racks or floors
  - Aggregated bandwidth
- Links to other networks
  - Wide range of media and bandwidth



Deciding Where to Locate Nodes and Links

- Placing Points-of-Presence (PoPs)
  - Large population of potential customers
  - Other providers or exchange points
  - Cost and availability of real-estate
  - Mostly in major metropolitan areas
- Placing links between PoPs
  - Already fiber in the ground
  - Needed to limit propagation delay
  - Needed to handle the traffic load

Customer Connecting to a Provider



Provider

1 access link

Provider

2 access links

Provider

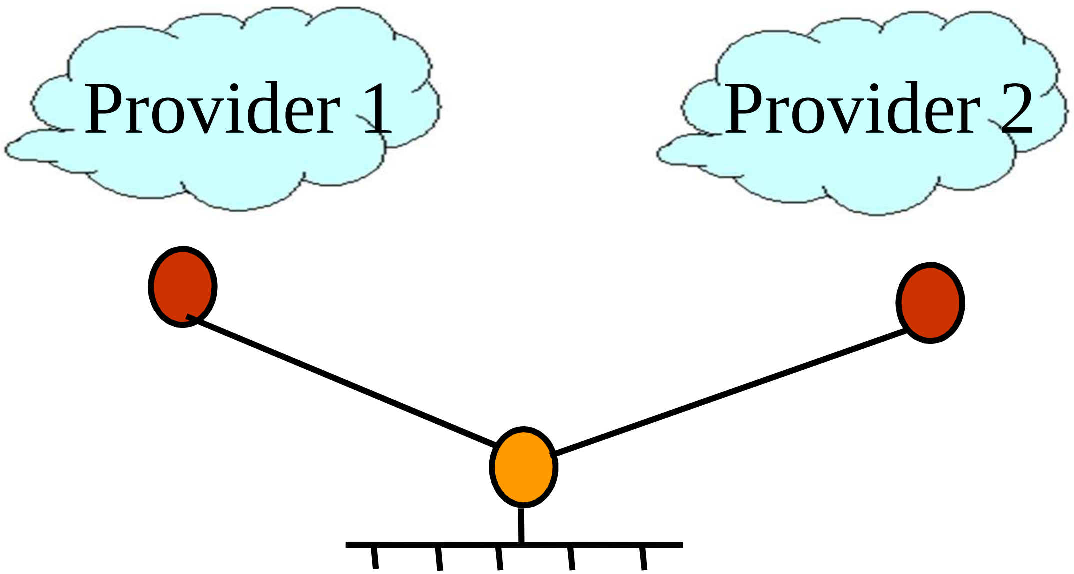
2 access routers

Provider

2 access PoPs

Multi-Homing: Two or More Providers

- Motivations for multi-homing
  - Extra reliability, survive single ISP failure
  - Financial leverage through competition
  - Better performance by selecting better path
  - Gaming the 95th-percentile billing model

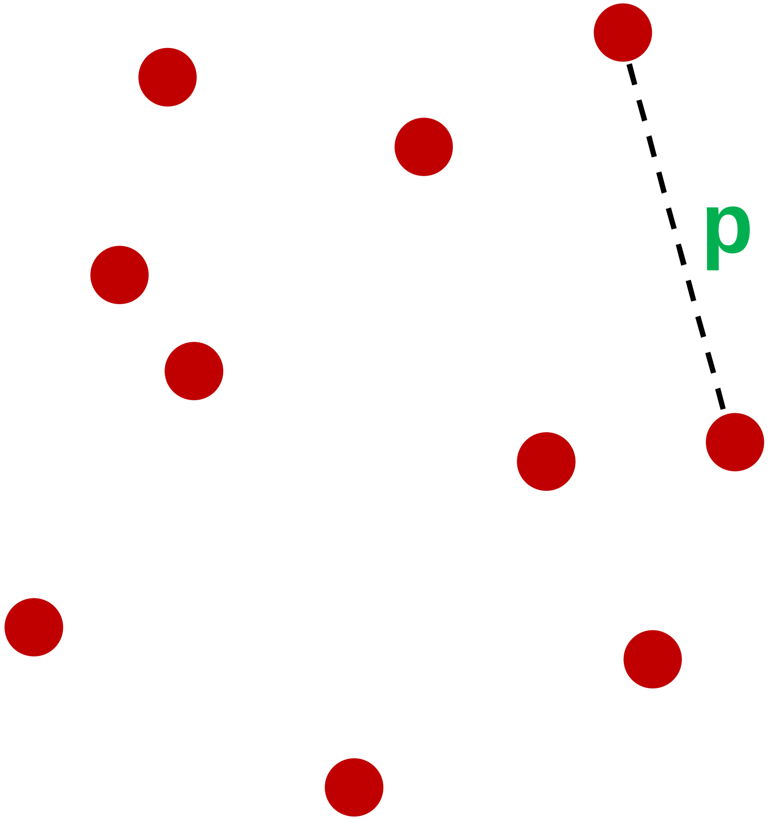


Modeling the Topology

Characterizing the Internet topology

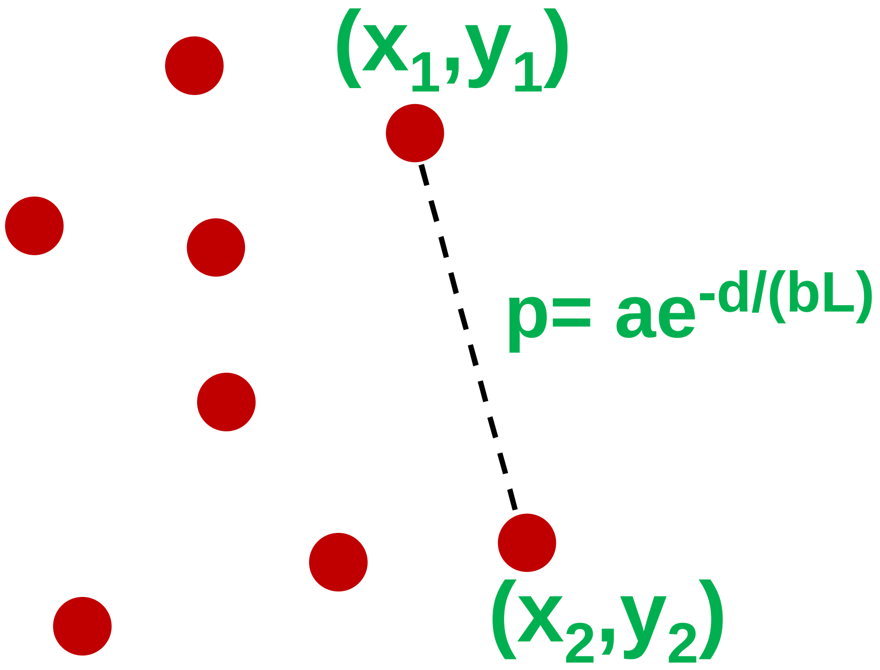
- Can we characterize the Internet’s topology?
  - Build understanding to inform protocol/architecture design
  - Create models to inform provisioning, perform accurate simulations
- Approach: abstract network as a graph
  - Intradomain: node=router, edge=link
  - Interdomain: node=AS, edge=peering

Erdős–Rényi model



- Edge exists between each pair of nodes with an equal probability **p**
- Edge probability independent of other edges
- Easy to mathematically analyze, but not the most accurate model for real-world networks

Waxman model



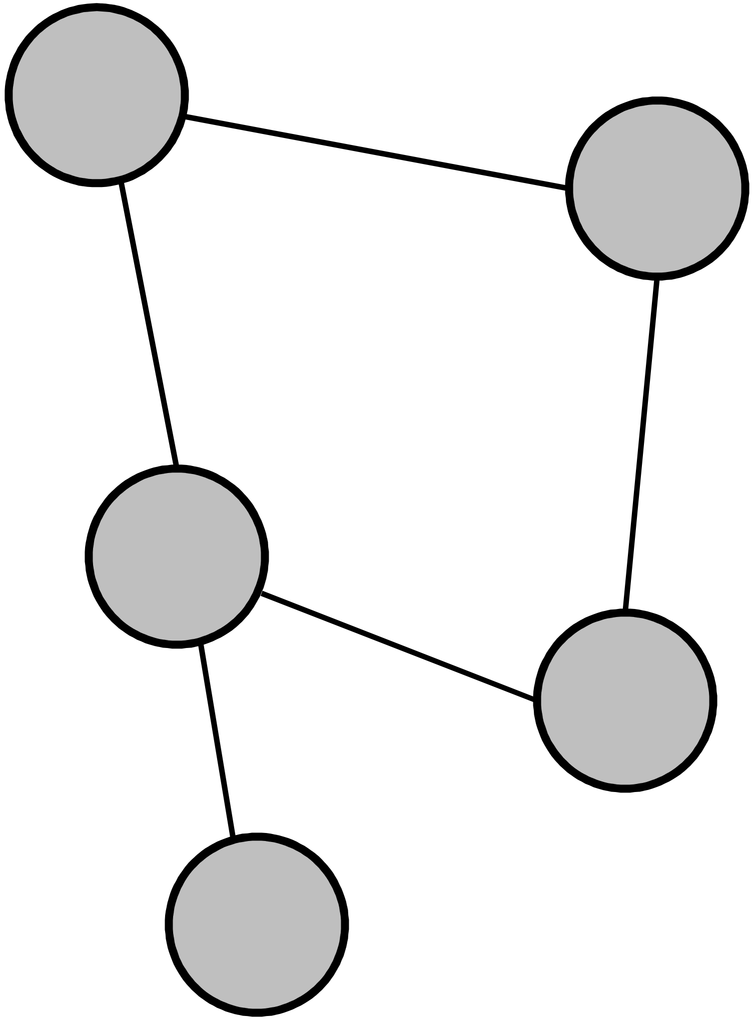
- Place nodes in plane
- Probability of edge depends on distance between nodes
- Aims to reflect geographic layout of network
  - See also: gravity model for internet traffic

d: distance

L: max distance between any two nodes

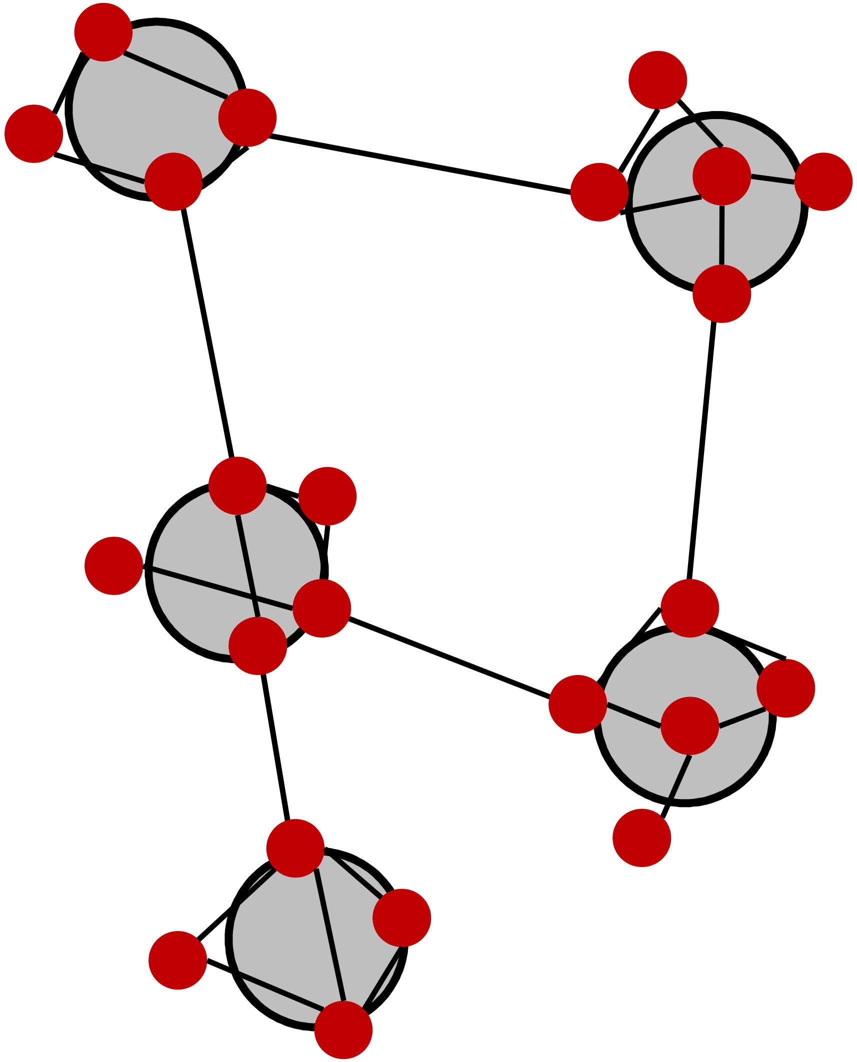
Parameters **a**>0, **b**<=1

Transit-stub model



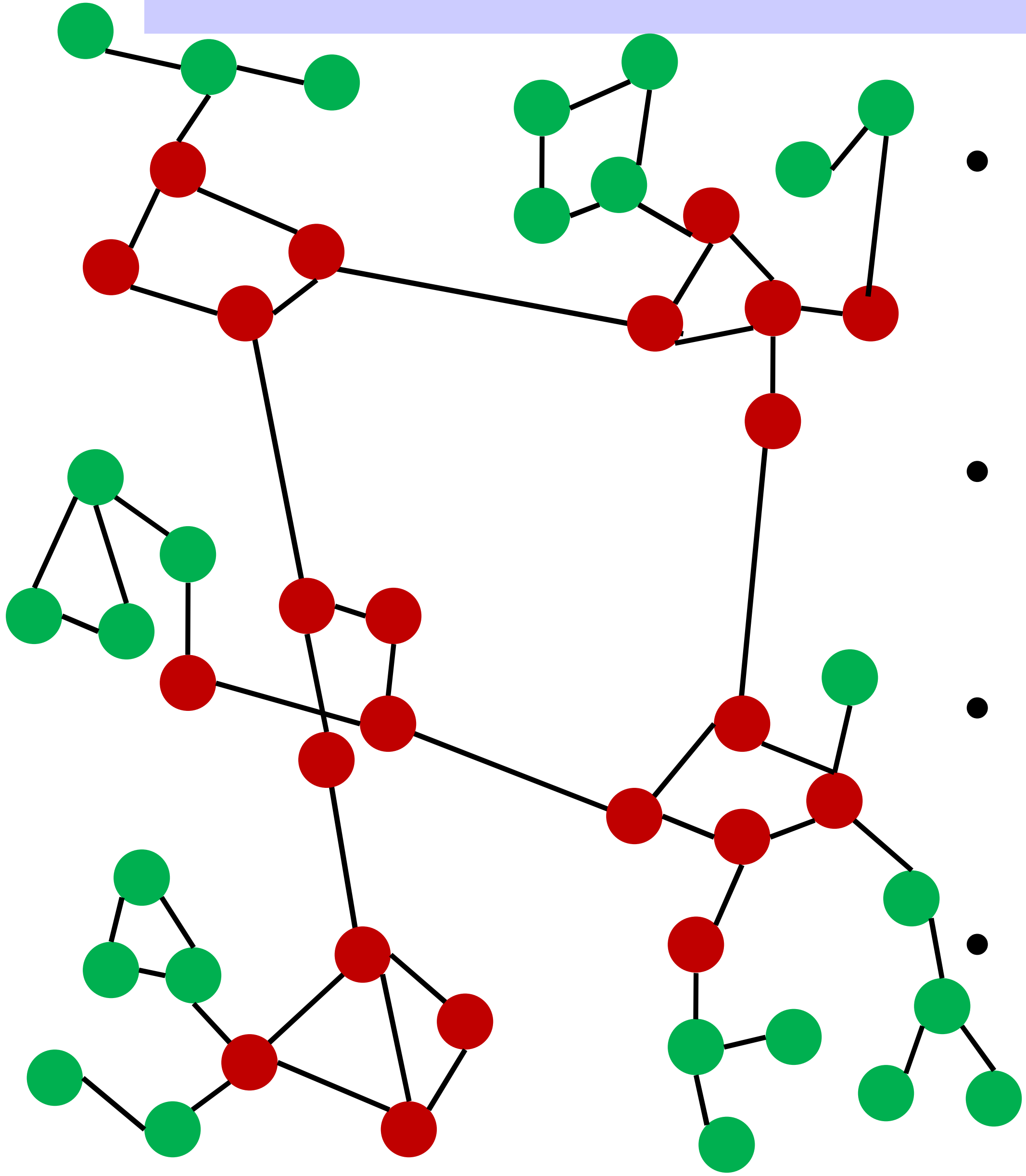
- Aims to model structural properties such as network backbones
- Randomly generate a graph using Waxman’s method

Transit-stub model



- Aims to model structural properties such as network backbones
- Randomly generate a graph using Waxman's method
- Expand each node to form a random graph (transit domain)

Transit-stub model



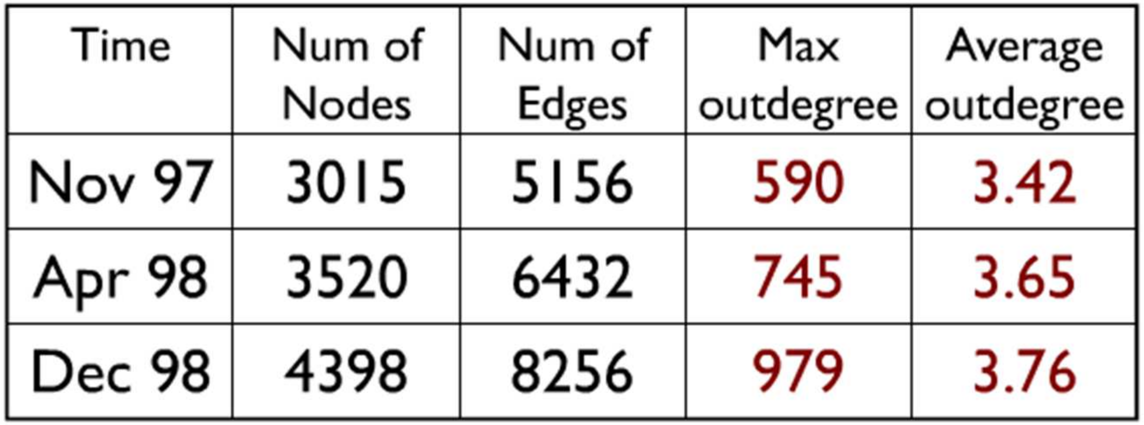
- Aims to model structural properties such as network backbones
- Randomly generate a graph using Waxman’s method
- Expand each node to form a random graph (transit domain)
- Connect stub domains to each transit domain

Transit stub in practice

- Transit-stub looks good, but is it close to the real thing?
- How to even answer this question?
- One way: write down a set of “metrics”, compare these metrics for generated graph against real Internet traces
  - Diameter, distribution of outdegree, mixing time, cut size, density, …
- This approach was taken by “On the power-law relationships of the Internet topology,” Faloutsos, Faloutsos, Faloutsos, Sigcomm 1999.

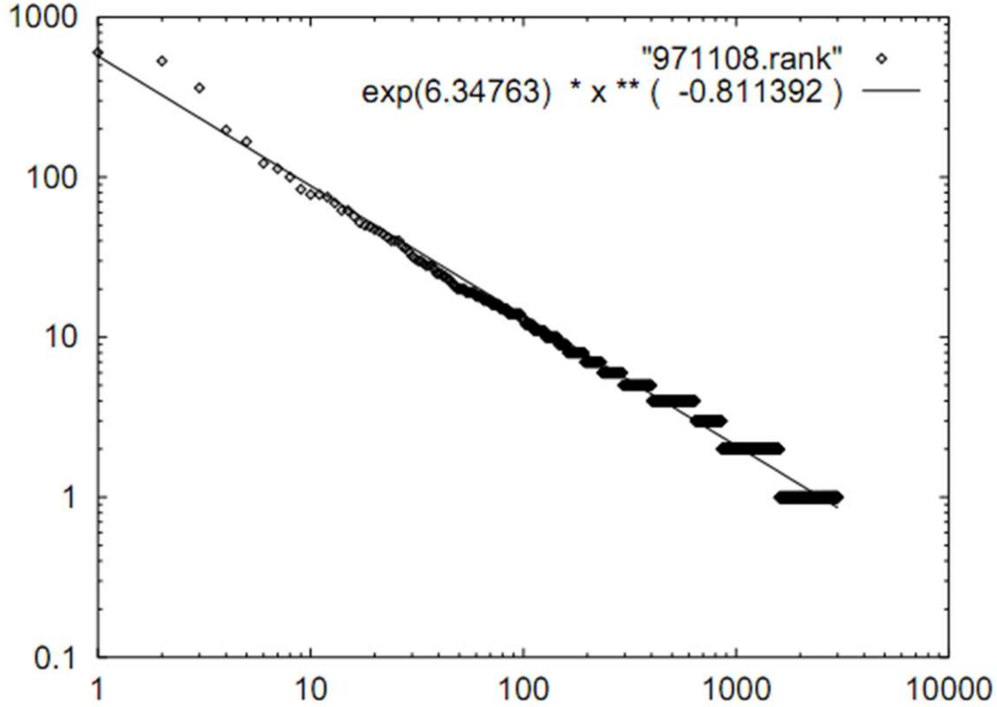
Faloutsos et al.'s findings

- Graphs can be decomposed into two components: trees and core
  - 40-50% of nodes are in trees
  - Maximum observed depth of 3
  - >80% of trees are of depth 1
- Outdegree is highly skewed



Time	Num of Nodes	Num of Edges	Max outdegree	Average outdegree
Nov 97	3015	5156	590	3.42
Apr 98	3520	6432	745	3.65
Dec 98	4398	8256	979	3.76

Router outdegrees are highly skewed



- Plot [router outdegree] vs [rank, in order of decreasing outdegree]
- Exhibits *Power Law* distribution

Do Waxman/Transit-stub give a power-law distribution?

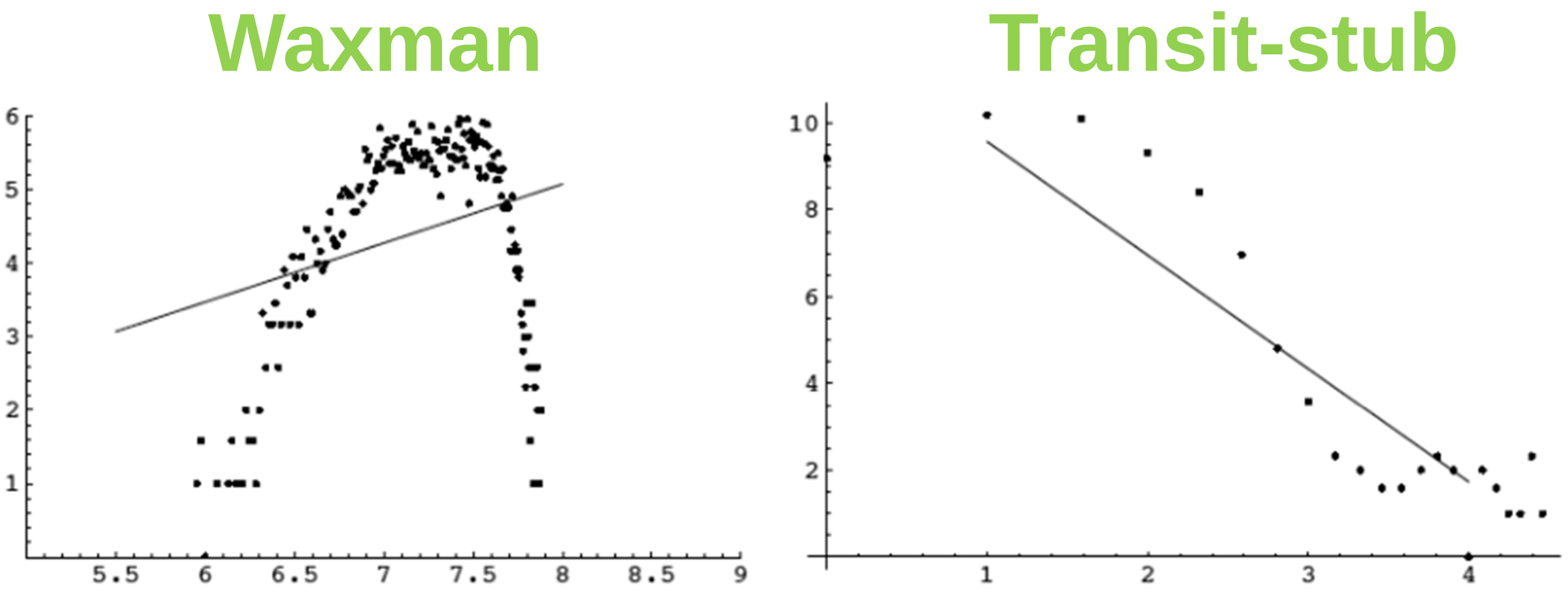
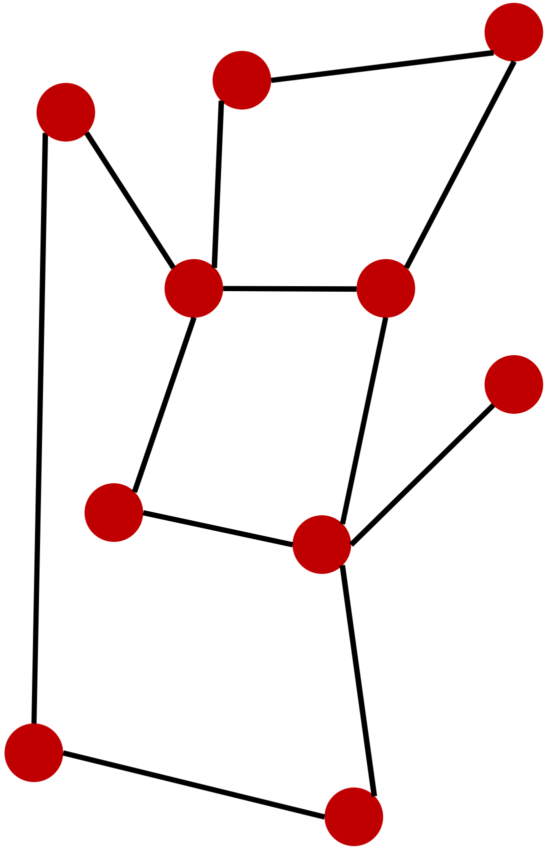


Figure 1: Log-log plot of frequency f_d vs. outdegree d for a 5000-node Waxman topology (left) and a 6660-node Transit-Stub topology (right). The correlation coefficient is 0.4 for the Waxman topology, and 0.9 for the Transit-Stub topology.

Where do power laws come from?

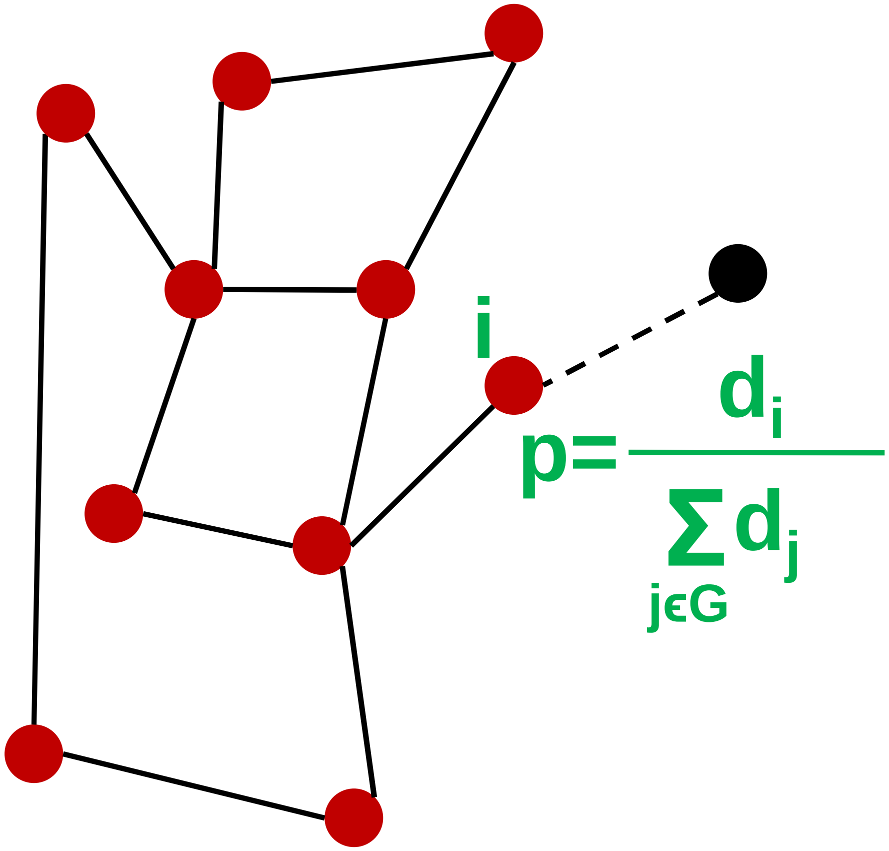
- Power laws observed in WWW, social networks, co-authorship of papers, actors appearing in same movie, interactions between proteins, etc.
- In these environments, there are “popular” nodes that are more desirable to connect to
- Idea of preferential attachment
  - A new node prefers to attach to an existing node that already has many connections
  - Eventually leads to system dominated by hubs

Approach taken by the BRITE topology generator



- Randomly generate a small graph

Approach taken by the BRITE topology generator



$$p = \frac{d_i}{\sum_{j \epsilon G} d_j}$$

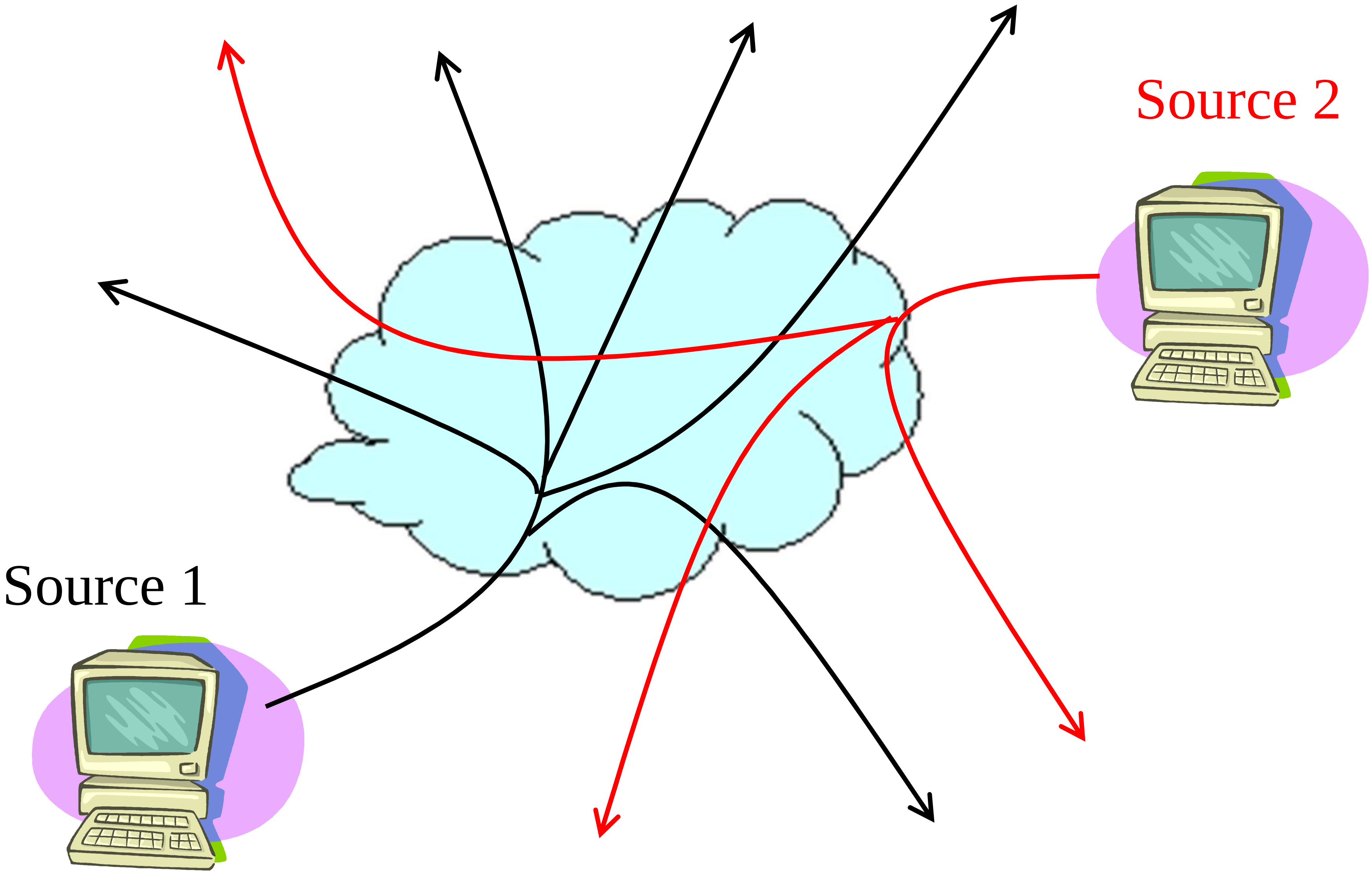
- Randomly generate a small graph
- Incrementally add a node
- Connect to other nodes with probability proportional to neighbor’s outdegree

Measuring the Topology

Motivation for Measuring the Topology

- Business analysis
  - Comparisons with competitors
  - Selecting a provider or peer
- Scientific curiosity
  - Treating data networks like an organism
  - Understand structure and evolution of Internet
- Input to research studies
  - Network design, routing protocols, …
- Interesting research problem in its own right
  - How to measure/infer the topology

Basic Idea: Measure from Many Angles

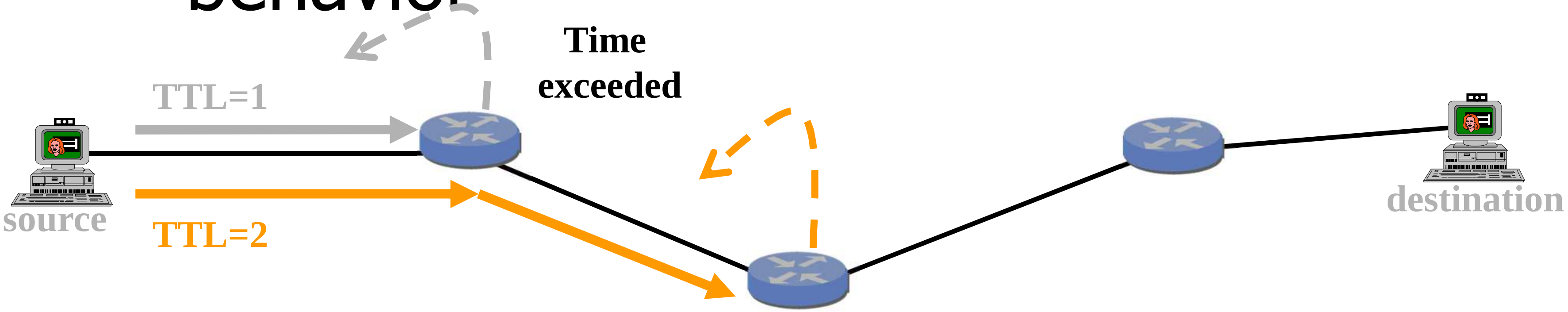


Where to Get Sources and Destinations?

- Source machines
  - Get accounts in many places
    - Good to have a lot of friends
  - Use an infrastructure like PlanetLab
    - Good to have friends who have lots of friends
  - Use public traceroute servers (nicely)
    - http://www.traceroute.org
- Destination addresses
  - Walk through the IP address space
    - One (or a few) IP addresses per prefix
  - Learn destination prefixes from public BGP tables
    - http://www.route-views.org

Traceroute: Measuring the Forwarding Path

- Time-To-Live field in IP packet header
  - Source sends a packet with a TTL of *n*
  - Each router along the path decrements the TTL
  - "TTL exceeded" sent when TTL reaches *0*
- Traceroute tool exploits this TTL behavior



Send packets with TTL=1, 2, 3, … and record source of "time exceeded" message

Example Traceroute Output
(Berkeley to CNN)

Hop number, IP address, DNS name

```
1  169.229.62.1        inr-daedalus-0.CS.Berkeley.EDU
2  169.229.59.225      soda-cr-1-1-soda-br-6-2
3  128.32.255.169      vlan242.inr-202-doecev.Berkeley.EDU
4  128.32.0.249        gigE6-0-0.inr-666-doecev.Berkeley.EDU
5  128.32.0.66         qsv-juniper--ucb-gw.calren2.net
6  209.247.159.109     POS1-0.hsipaccess1.SanJose1.Level3.net
7  *                   ?
8  64.159.1.46         ?
9  209.247.9.170       pos8-0.hsa2.Atlanta2.Level3.net
10  66.185.138.33      pop2-atm-P0-2.atdn.net
11  *                  ?
12  66.185.136.17      pop1-atl-P4-0.atdn.net
13  64.236.16.52       www4.cnn.com
```

No response from router

No name resolution

Problems with Traceroute

- Missing responses
  - Routers might not send “Time-Exceeded”
  - Firewalls may drop the probe packets
  - “Time-Exceeded” reply may be dropped
- Misleading responses
  - Probes taken while the path is changing
  - Name not in DNS, or DNS entry misconfigured
  - Forward path can differ from reverse path
- Mapping IP addresses
  - Mapping interfaces to a common router
  - Mapping interface/router to Autonomous System
- Angry operators who think this is an attack

Map Traceroute Hops to ASes

Traceroute output: (hop number, IP)

```
1  169.229.62.1      AS25
2  169.229.59.225    AS25      Berkeley
3  128.32.255.169    AS25
4  128.32.0.249      AS25
5  128.32.0.66       AS11423   Calren
6  209.247.159.109   AS3356
7  *                 AS3356    Level3
8  64.159.1.46       AS3356
9  209.247.9.170     AS3356
10  66.185.138.33    AS1668
11  *                AS1668    AOL
12  66.185.136.17    AS1668
13  64.236.16.52     AS5662    CNN
```

Need **accurate** IP-to-AS mappings (for network equipment).

Candidate Ways to Get IP-to-AS Mapping

- Routing address registry
  - Voluntary public registry such as whois.radb.net
  - Used by prtraceroute and “NANOG traceroute”
  - Incomplete and quite out-of-date
    - Mergers, acquisitions, delegation to customers
- Origin AS in BGP paths
  - Public BGP routing tables such as RouteViews
  - Used to translate traceroute data to an AS graph
  - Incomplete and inaccurate… but usually right
    - Multiple Origin ASes (MOAS), no mapping, wrong mapping

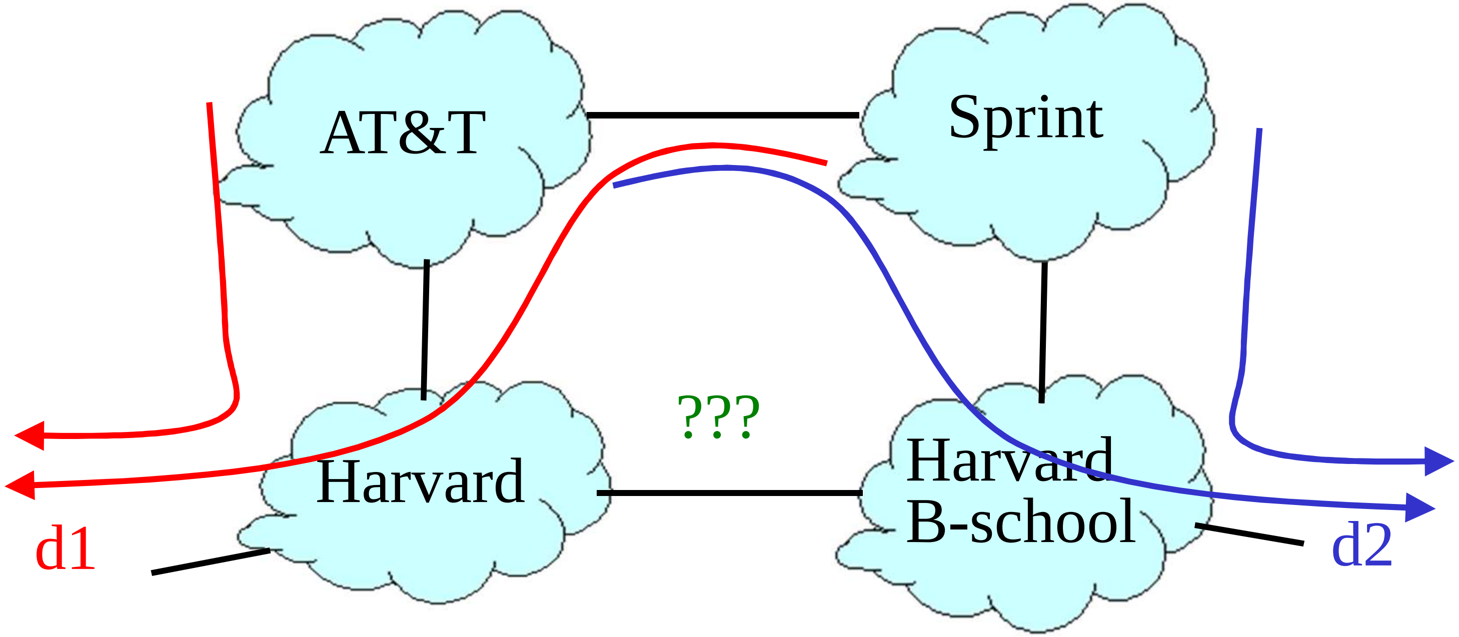
Example: BGP Table ("show ip bgp" at RouteViews)

```
   Network          Next Hop          Metric  LocPrf  Weight  Path
*  3.0.0.0/8        205.215.45.50                          0  4006 701 80 i
*                   167.142.3.6                            0  5056 701 80 i
*                   157.22.9.7                             0  715 1 701 80 i
*                   195.219.96.239                         0  8297 6453 701 80 i
*                   195.211.29.254                         0  5409 6667 6427 3356 701 80 i
*>                  12.127.0.249                           0  7018 701 80 i
*                   213.200.87.254       929               0  3257 701 80 i
*  9.184.112.0/20   205.215.45.50                          0  4006 6461 3786 i
*                   195.66.225.254                         0  5459 6461 3786 i
*>                  203.62.248.4                           0  1221 3786 i
*                   167.142.3.6                            0  5056 6461 6461 3786 i
*                   195.219.96.239                         0  8297 6461 3786 i
*                   195.211.29.254                         0  5409 6461 3786 i
```

AS 80 is General Electric, AS 701 is UUNET, AS 7018 is AT&T
AS 3786 is DACOM (Korea), AS 1221 is Telstra

Problem of Missing Edges

- Limited collection of paths
  - Some edges might never be traversed
  - Especially low in the AS hierarchy
  - … and backup links
- Example: paths from two tier-1 ISPs miss an edge


AT&T
Sprint
???
Harvard
Harvard
B-school
d1
d2