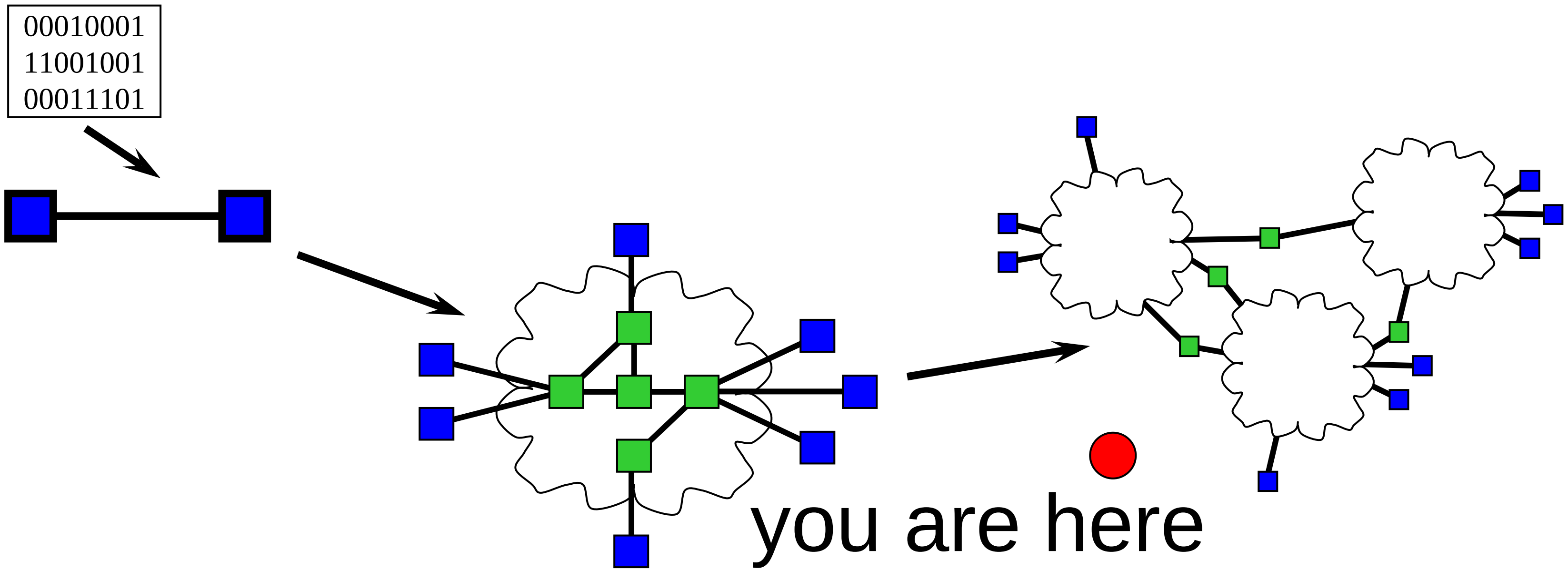
Lecture 9:
Internet

CS/ECE 438: Communication Networks

Prof. Matthew Caesar

March 10, 2010

The Big Picture



Topics

- Overview of the Internet
  - Message formats, topology, history, interconnection
- Internet Routing
  - Delivery models: Unicast, Multicast, Anycast
  - Forwarding (MTU, tunneling/VPNs), ICMP
  - Network management (TE)
- Internet Addressing/Naming
  - DNS, NAT; ARP, DHCP; Mobility

Overview of the Internet

What is the Internet?

- 30,925 ISPs, 2,600,000 routers, 289,989 subnets, 625,226,456 hosts
- Rich array of systems and protocols
- ISPs compete for business, hide private information, end-hosts misbehave, complex failure modes, cross-dependencies and oscillations
- Yet everyone must cooperate to ensure reachability
  - Relies on complex interactions across multiple systems and protocols

Internetworking

- You currently understand
  - How to build a network on one physical medium
  - How to connect networks (except routing)
- You have experimented with
  - Construct a reliable byte stream
  - Deal with
    - Finite frame length
    - Corrupt frames
    - Frame loss
- Now: Internetworking
  - Address heterogeneity of networks
  - Address rapid growth of Internet (scalability issues)

Internetworking

- Dealing with simple heterogeneity issues
  - Defining a service model
  - Defining a global namespace
  - Structuring the namespace to simplify forwarding
  - Building forwarding information (routing)
  - Translating between global and local (physical) names
  - Hiding variations in frame size limits
- Dealing with global scale
- Moving forward with IP

Internetworks

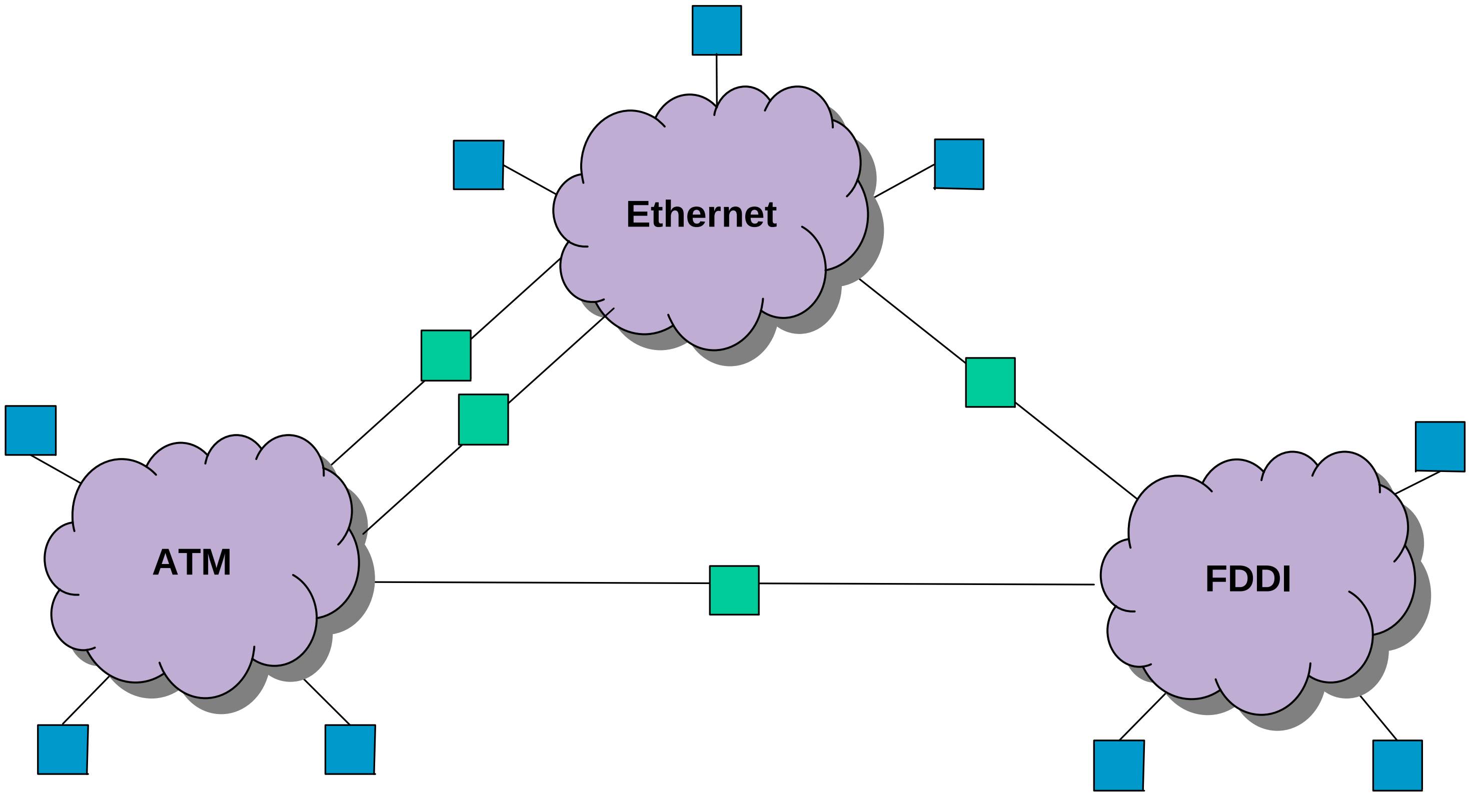
reading: Peterson and Davie, Ch. 4 + Sect. 9.1

- Basics of internetworking (heterogeneity)
  - The IP protocol, address resolution, and control messages
- Routing
- Global internets (scale)
  - Address assignment and translation
  - Hierarchical routing
  - Name translation and lookup
  - Multicast traffic
- Future internetworking: IPv6

Basics of Internetworking

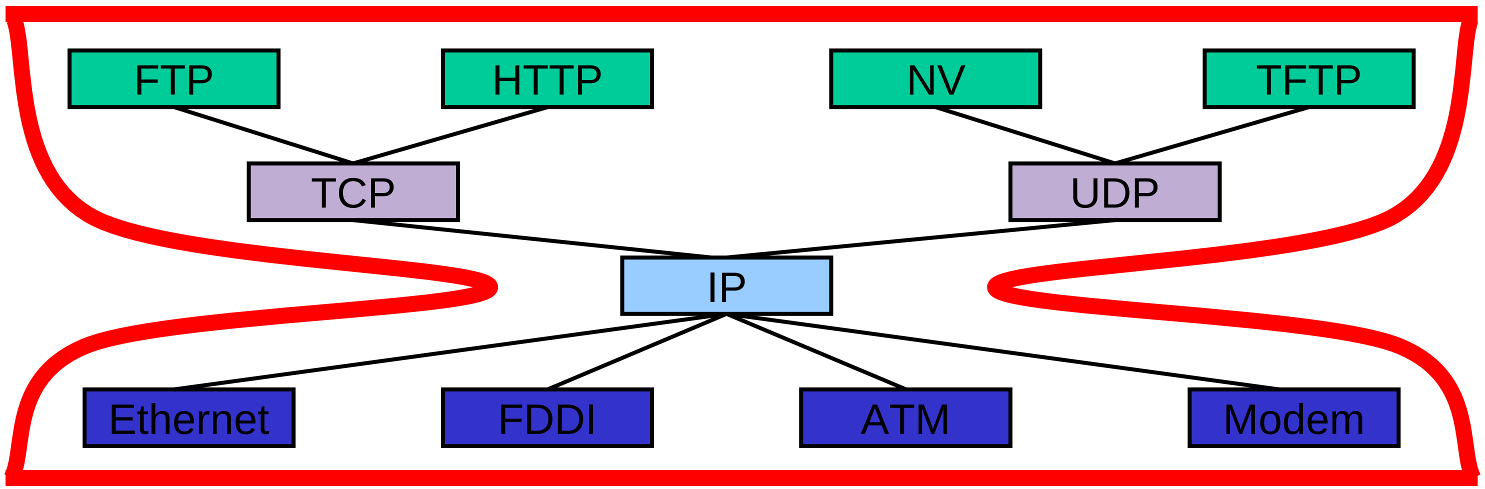
- What is an internetwork
  - Illusion of a single (direct link) network
  - Built on a set of distributed heterogeneous networks
  - Abstraction typically supported by software
- Properties
  - Supports heterogeneity
    - Hardware, OS, network type, and topology independent
  - Scales to global connectivity
- The Internet is the specific global internetwork that grew out of ARPANET

Internetworking

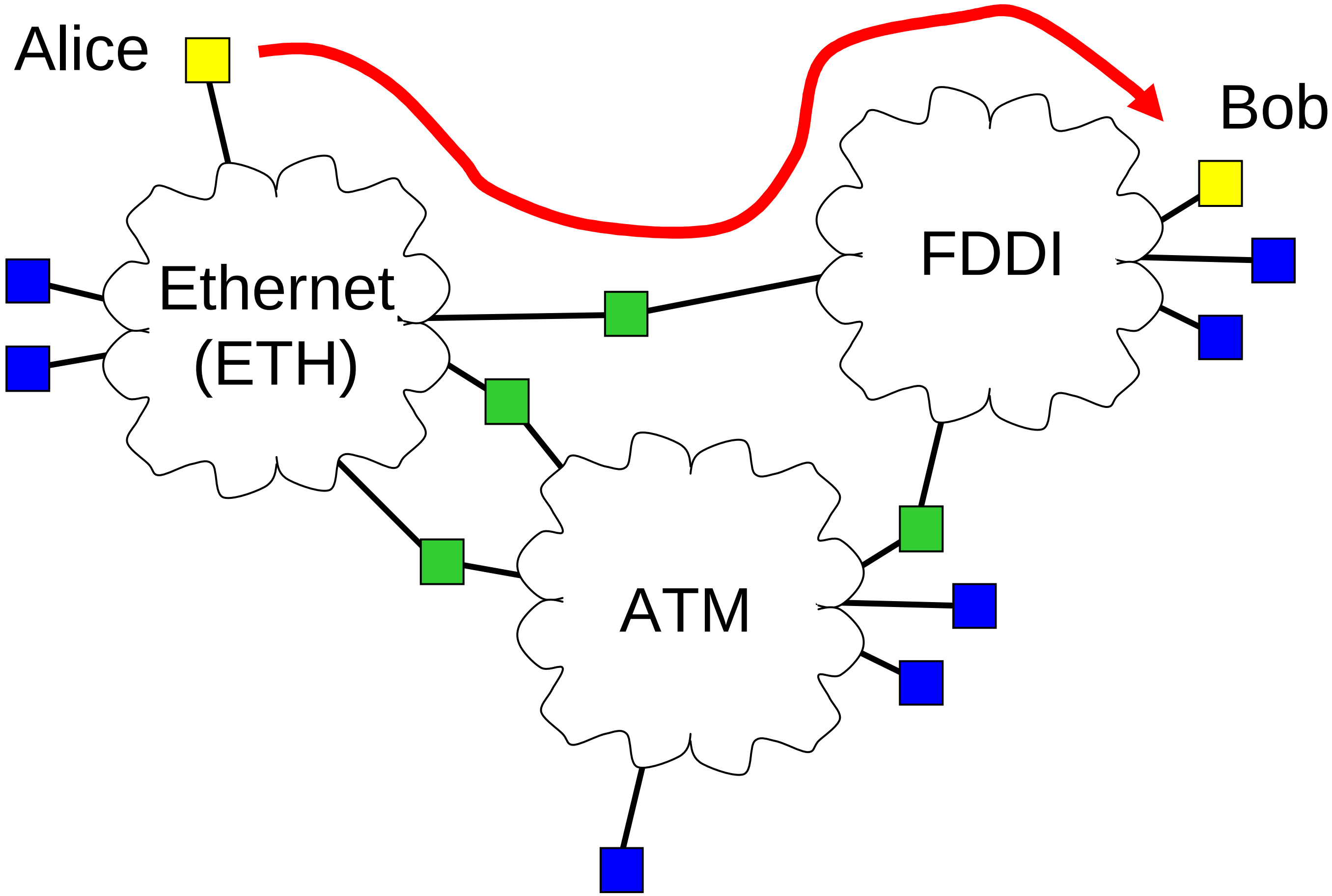


Internet Protocol (IP)

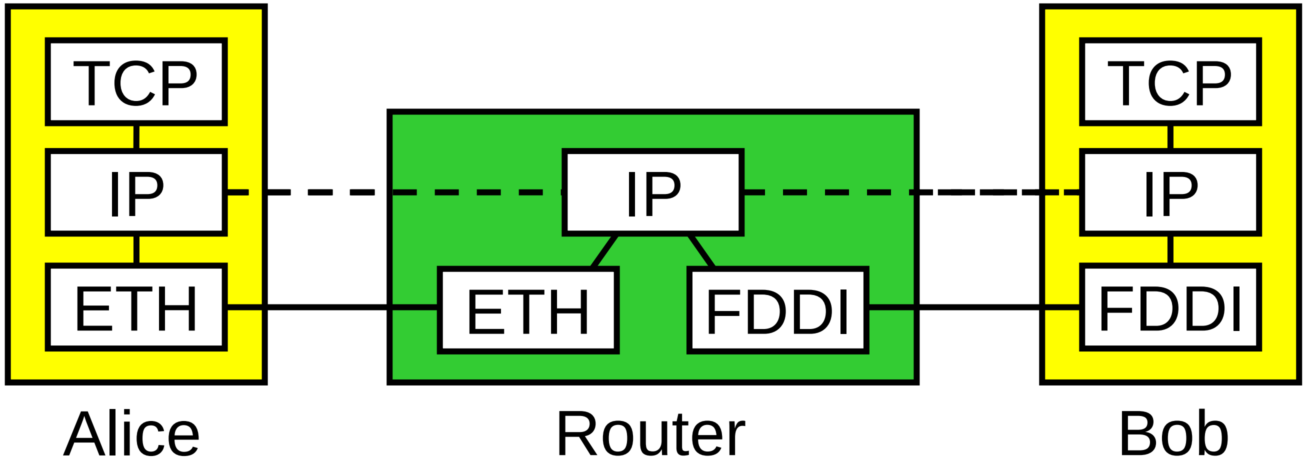
- Network-level protocol for the Internet
- Operates on all hosts and routers
  - Routers are nodes connecting distinct networks to the Internet



Message Transmission



Message Transmission



1. Alice/application finds Bob's IP address, sends packet
2. Alice/IP forwards packet to Router
3. Alice/IP looks up Router's Ethernet address and sends
4. Router/IP forwards packet to Bob
5. Router/IP looks up Bob's FDDI address and sends

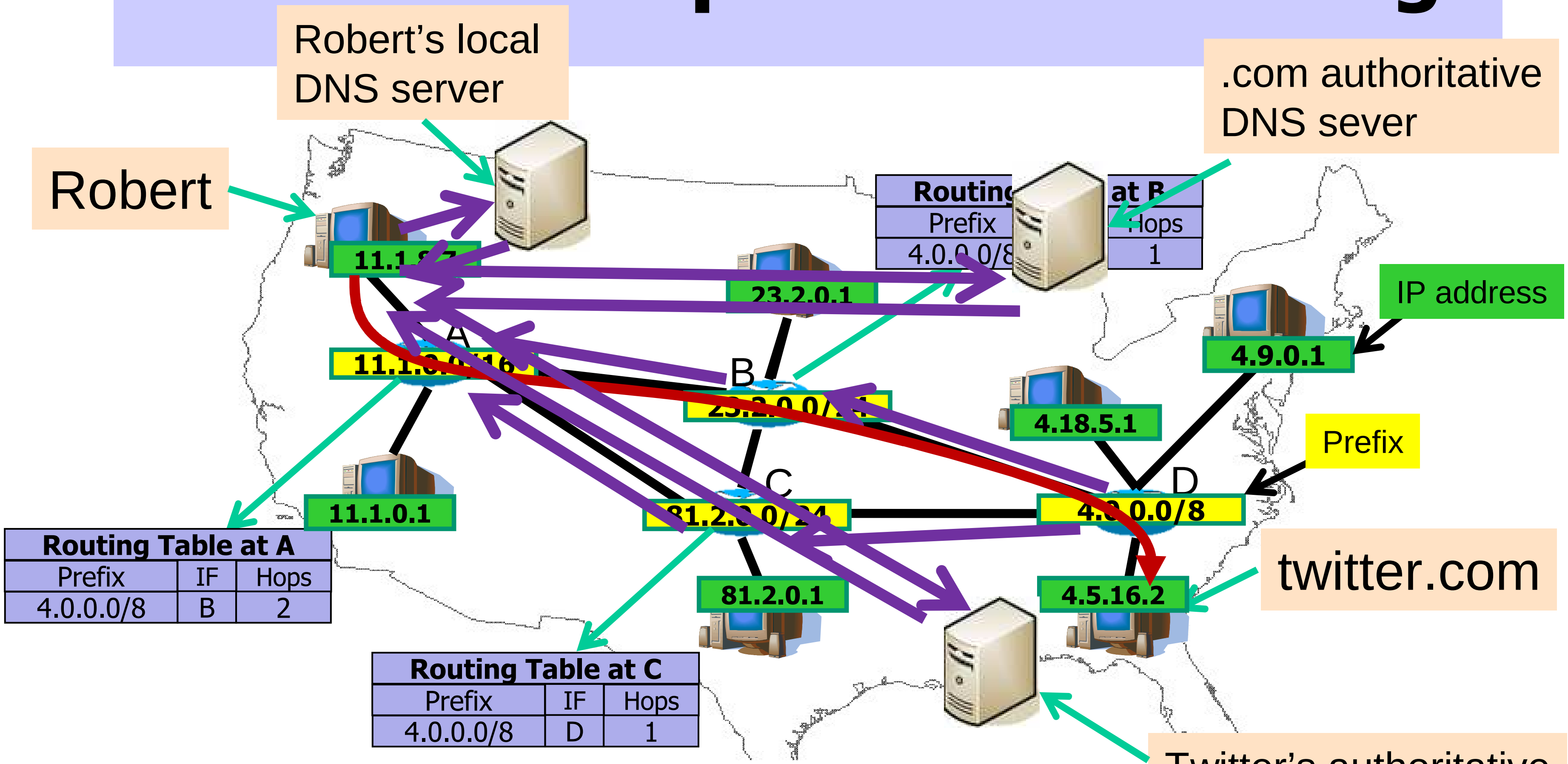
Internet Protocol Service Model

- Service provided to transport layer (TCP, UDP)
  - Global name space
  - Host-to-host connectivity (connectionless)
  - Best-effort packet delivery
- Not in IP service model
  - Delivery guarantees on bandwidth, delay or loss
- Delivery failure modes
  - Packet delayed for a very long time
  - Packet loss
  - Packet delivered more than once
  - Packets delivered out of order

Simple Internetworking with IPv4

- Host addressing
- Forwarding
- Fragmentation and reassembly
- Error reporting/control messages

Overview of packet forwarding



- Hosts assigned topology-dependent addresses
- Routers advertise address blocks ("prefixes")
- Routers compute "shortest" paths to prefixes
- Map IP addresses to names with DNS
- More on "Routing" and "Naming" later

Routing and Forwarding

Roadmap

- IP Forwarding
  - Fragmentation and reassembly, ICMP, VPNs
  - Delivery models
- IP Routing
  - Routing across ISPs
  - How inter- and intra-domain routing work together
  - How Ethernet and intra-domain routing work together

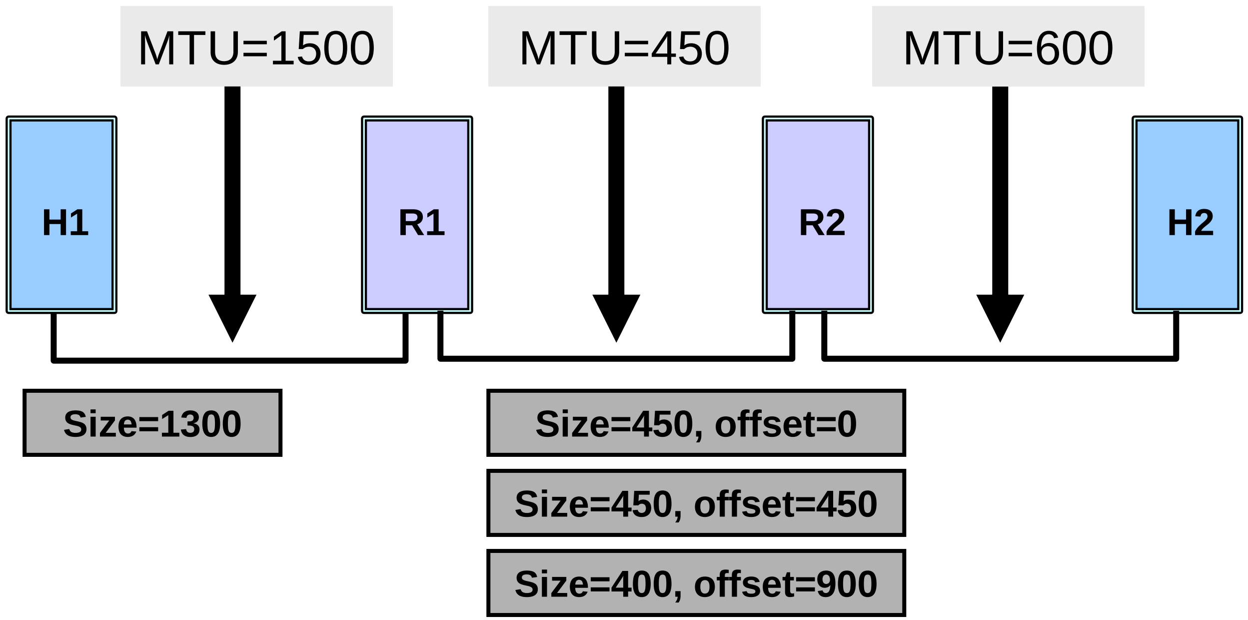
Datagram Forwarding with IP

- Hosts and routers maintain forwarding tables
  - List of <prefix, next hop> pairs
  - Often contains a default route
    - Pass unknown destination to provider ISP
  - Simple and static on hosts, edge routers
    - Complex and dynamic on core routers
- Packet forwarding
  - Compare network portion of address with <network/host, next hop> pairs in table
    - Send directly to host on same network
    - Send to indirectly (via router on same network) to host on different network
  - Use ARP to get hardware address of host/router

IP Packet Size

- Problem
  - Different physical layers provide different limits on frame length
    - Maximum transmission unit (MTU)
  - Source host does not know minimum value
    - Especially along dynamic routes

IP Fragmentation and Reassembly

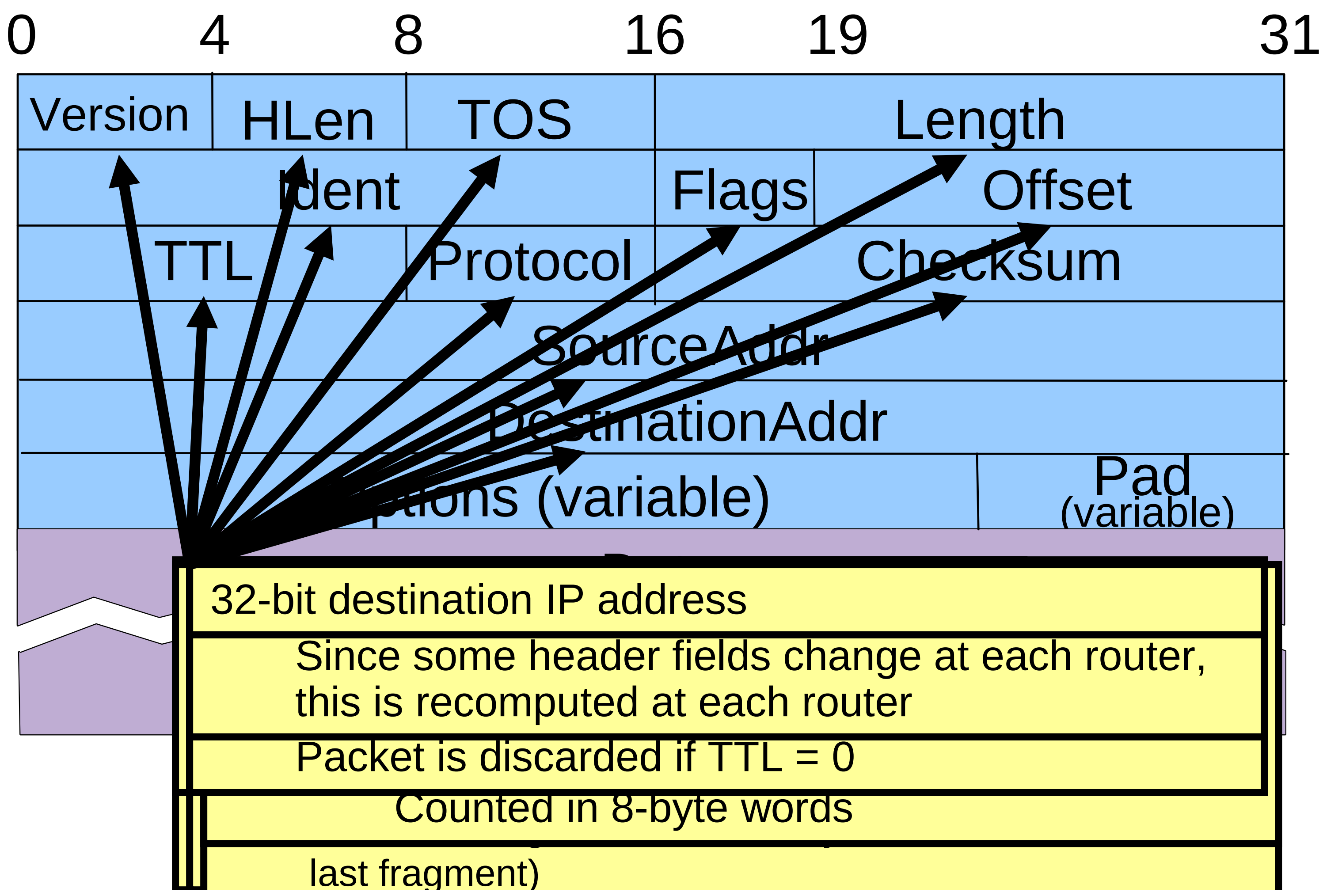


- Solution
  - When necessary, split IP packet into acceptably sized packets prior to sending over physical link
- Questions
  - Where should reassembly occur?
  - What happens when a fragment is damaged/lost?

IP Fragmentation and Reassembly

- Fragments: self-contained IP datagrams
- Reassemble at destination
  - Minimizes refragmentation
- If one or more fragments are lost
  - Drop all fragments in packet
- Avoid fragmentation at source host
  - Transport layer should send packets small enough to fit into one MTU of local physical network
  - Must consider IP header

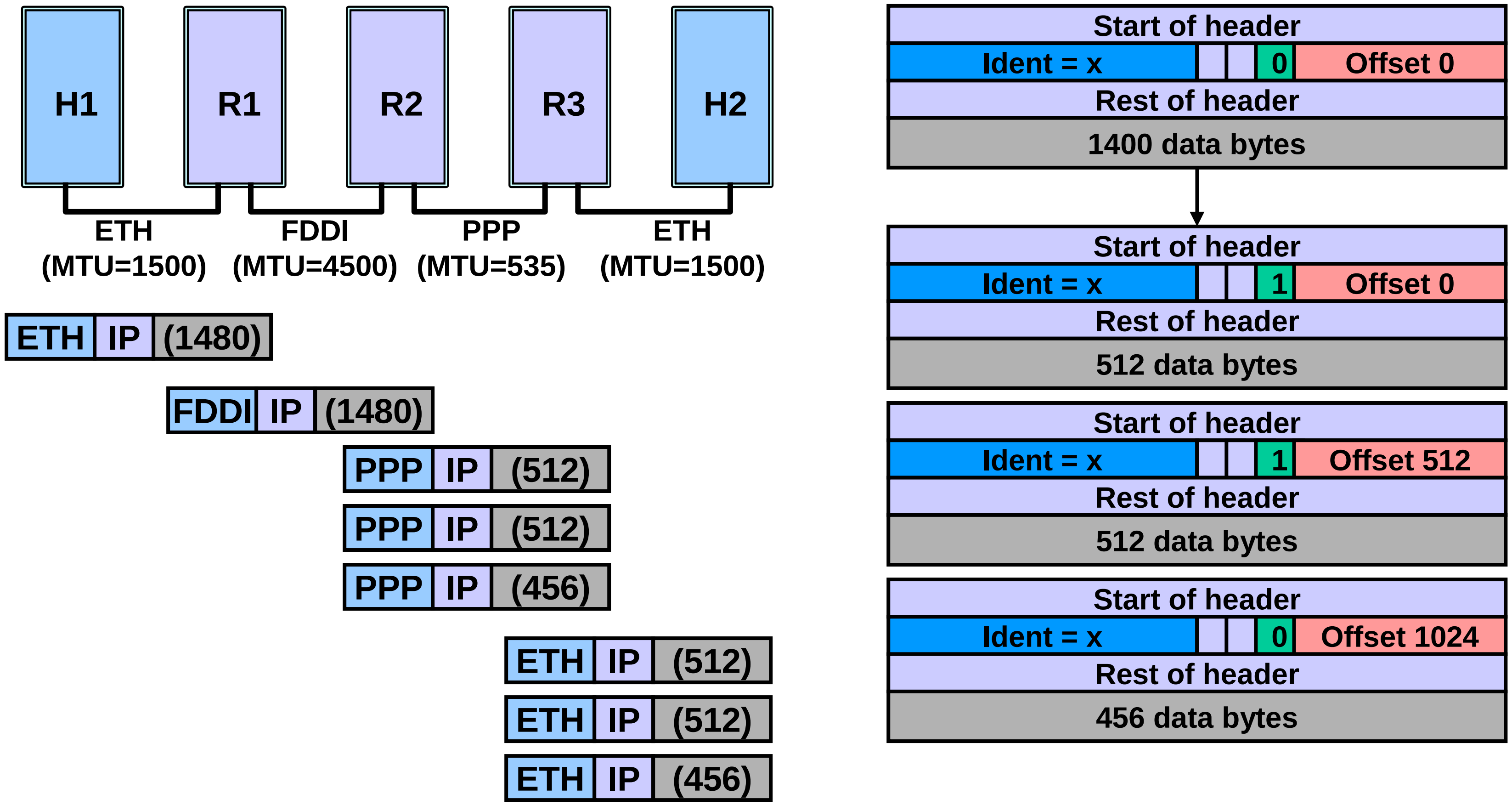
IP Packet Format



Path MTU discovery

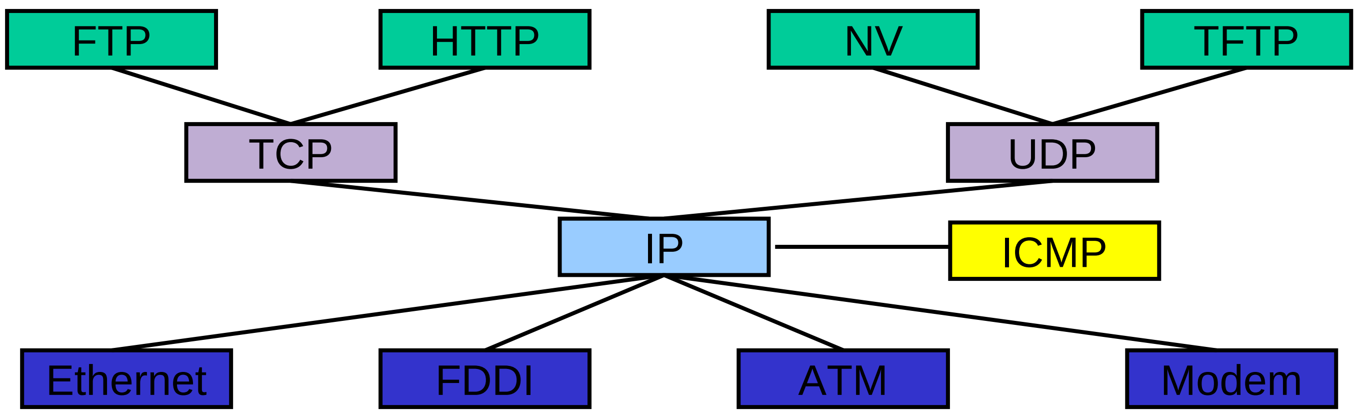
- Set “don’t fragment” bit in IP header, size is MTU of first hop
- Interface with too-small MTU responds back with “ICMP” message
  - Unfortunately, many networks drop ICMP traffic
- Reduce packet size, repeat until discover smallest MTU on path
- Binary search
  - Better yet: note there are small number of MTUs in the Internet

IP Fragmentation and Reassembly



Internet Control Message Protocol (ICMP)

- IP companion protocol
  - Handles error and control messages
  - Used for troubleshooting and measurement



ICMP

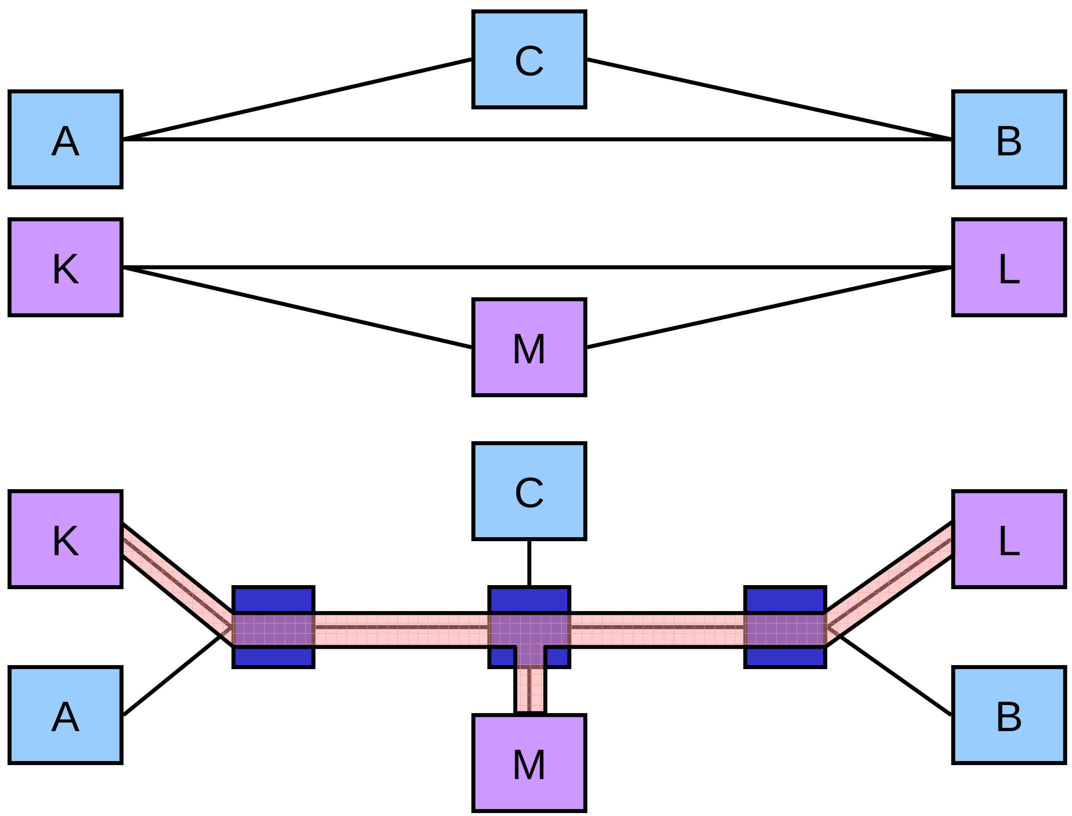
- Used for pings (probing remote hosts), traceroutes (learning set of routers along a path), etc.

- Error Messages
  - Host unreachable, fragmentation failed, TTL exceeded, invalid header
- Control Messages
  - Echo/ping request and reply, timestamps, route redirect

Virtual Private Networks

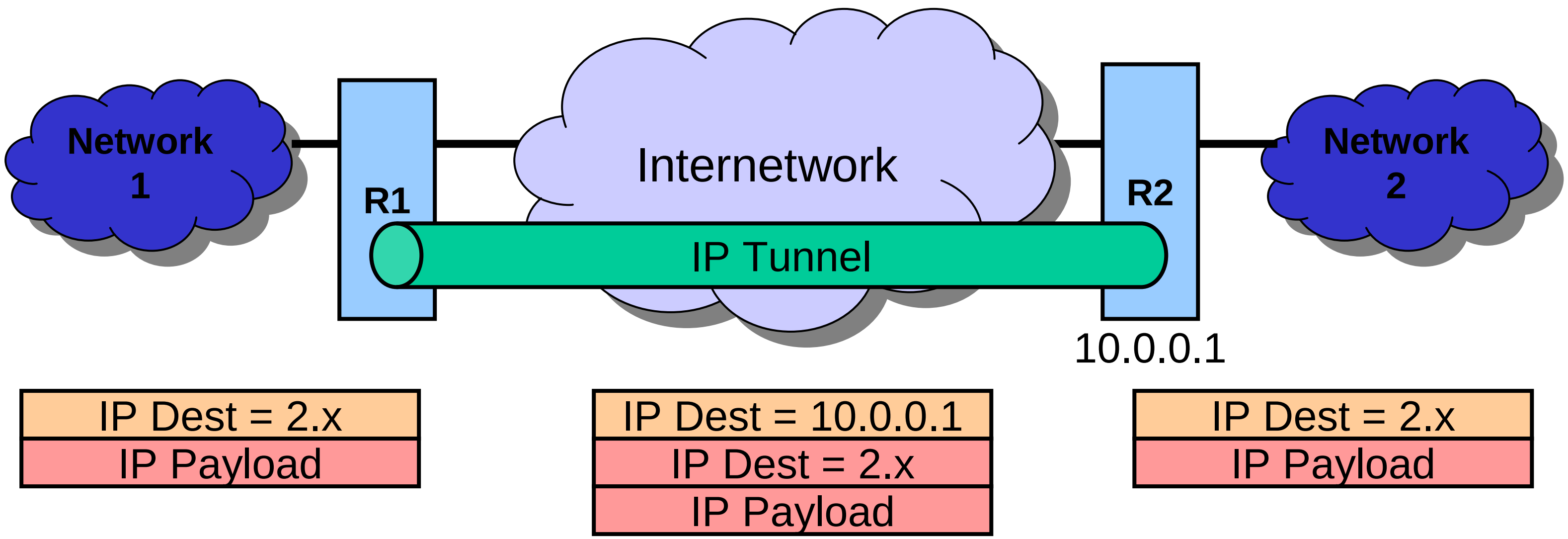
- Goals
  - Controlled connectivity
    - Restrict forwarding to authorized hosts
  - Controlled capacity
    - Change router drop and priority policies
    - provide guarantees on bandwidth, delay, etc.
- Virtual Private Network
  - A group of connected subnets
  - Connections may be over shared network
  - Similar to VLANs, but over IP allowing the use of heterogeneous networks

Virtual Private Networks



Tunneling

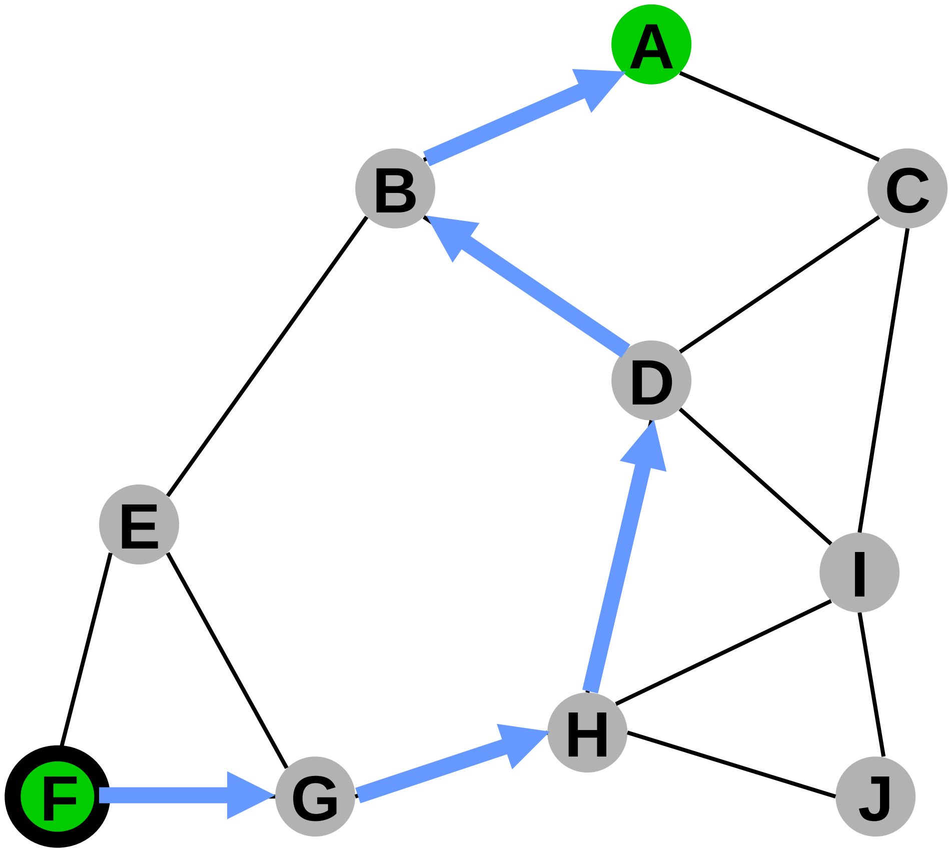
- IP Tunnel
  - Virtual point-to-point link between an arbitrarily connected pair of nodes



Tunneling

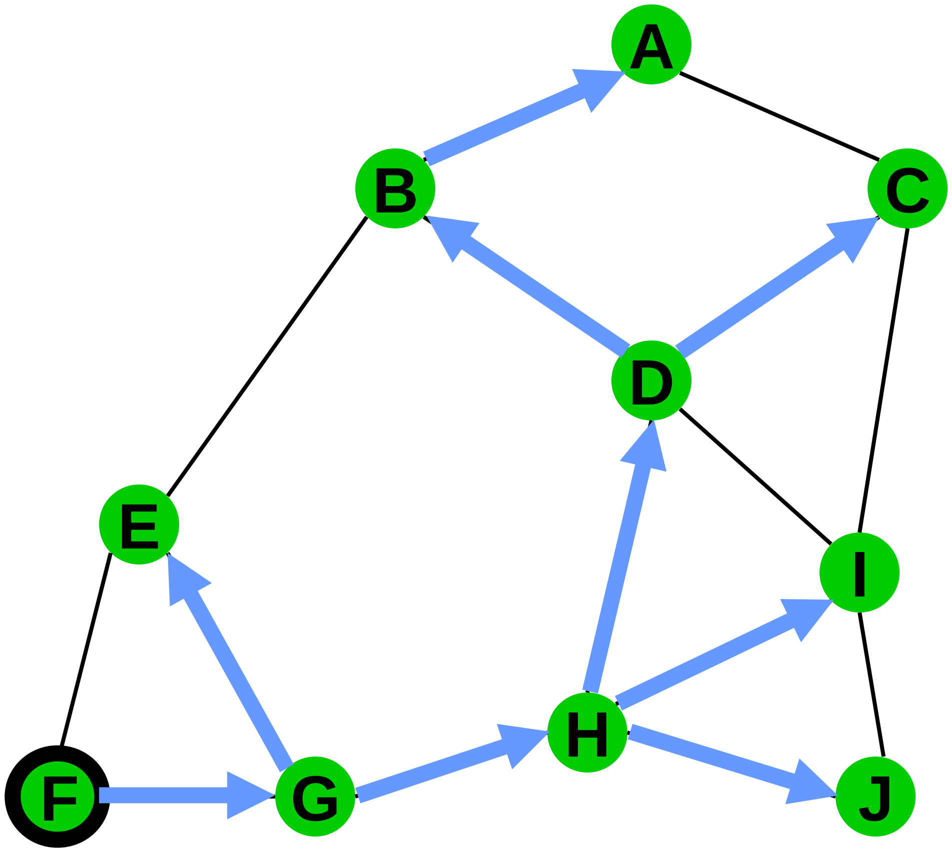
- Advantages
  - Transparent transmission of packets over a heterogeneous network
  - Only need to change relevant routers
- Disadvantages
  - Increases packet size
  - Processing time needed to encapsulate and unencapsulate packets
  - Management at tunnel-aware routers

Delivery models



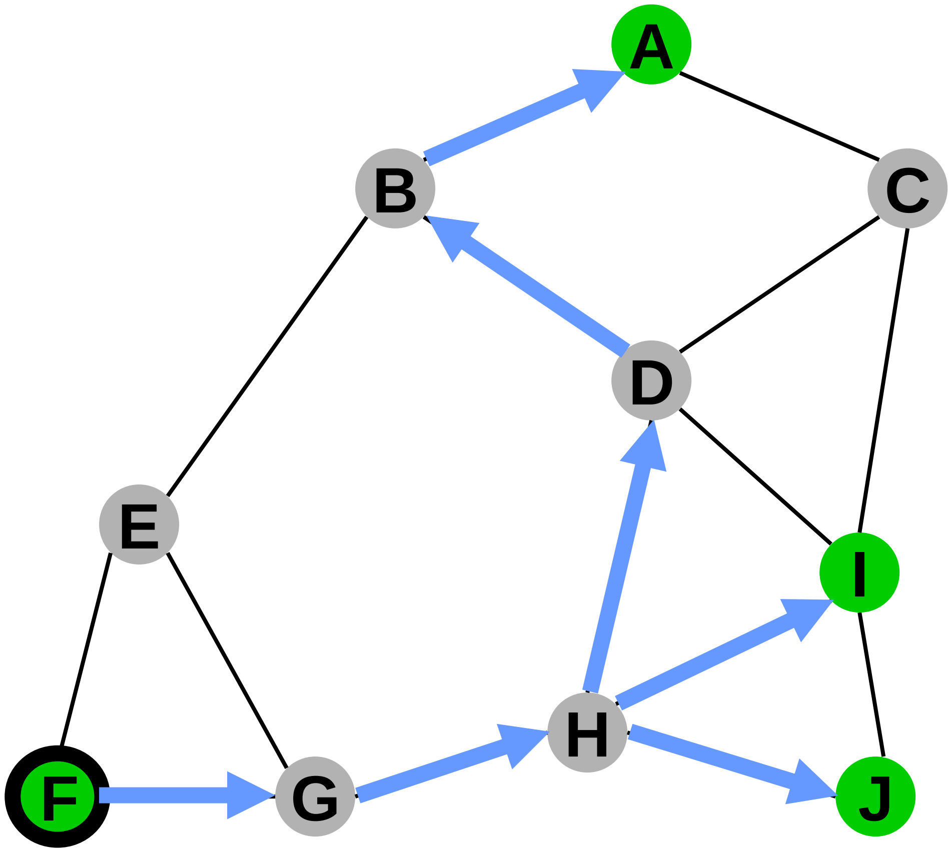
- Unicast
  - One source, one destination
  - Widely used (web, p2p, streaming, many other protocols)
- Broadcast
- Multicast
- Anycast

Delivery models



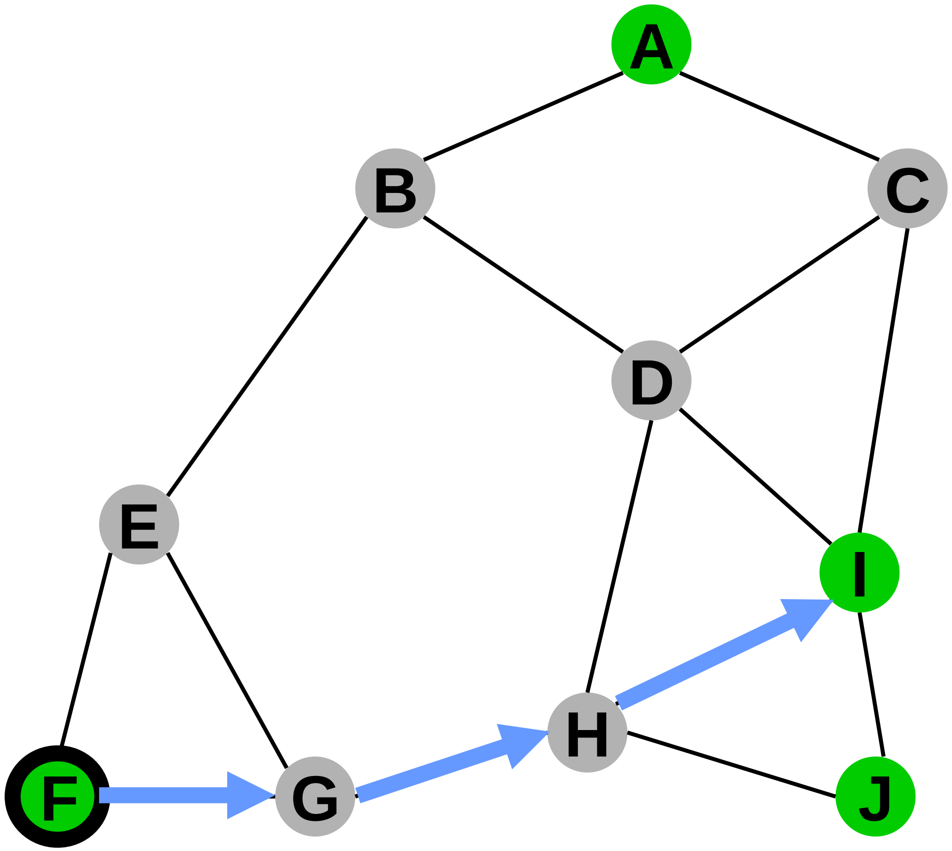
- Unicast
- Broadcast
  - One source, all destinations
  - Used to disseminate control information, perform service discovery
- Multicast
- Anycast

Delivery models



- Unicast
- Broadcast
- Multicast
  - One source, several (prespecified) destinations
  - Used within some ISP infrastructures for content delivery, overlay networks
- Anycast

Delivery models



- Unicast
- Broadcast
- Multicast
- Anycast
  - One source, route to “best” destination
  - Used in DNS, content distribution

Internet Multicast

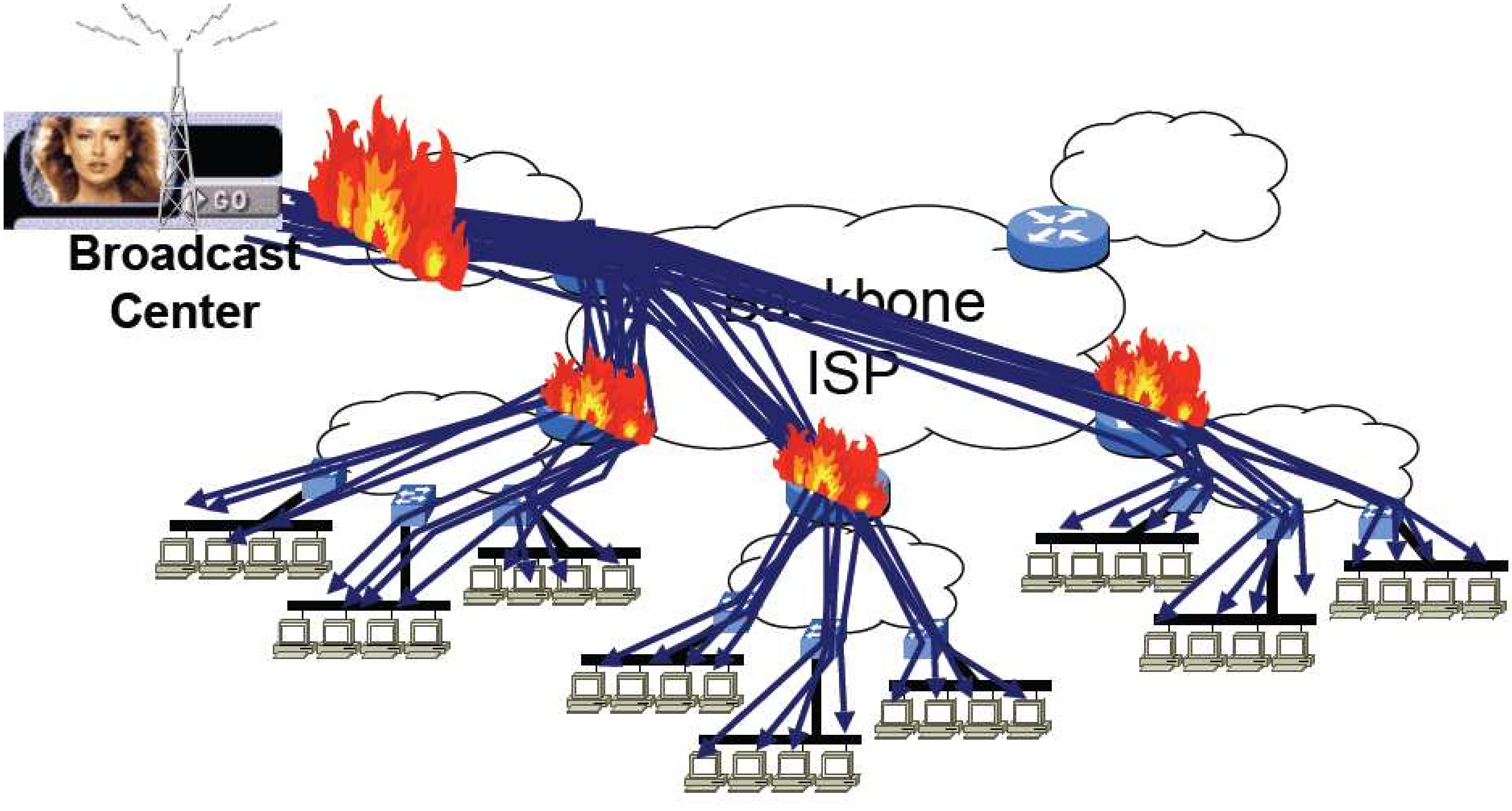
- Motivation and challenges
- Support strategy
- IP multicast service model
- Multicast in the Internet
- Multicast routing protocols
- Limitations

Multicast: motivating example

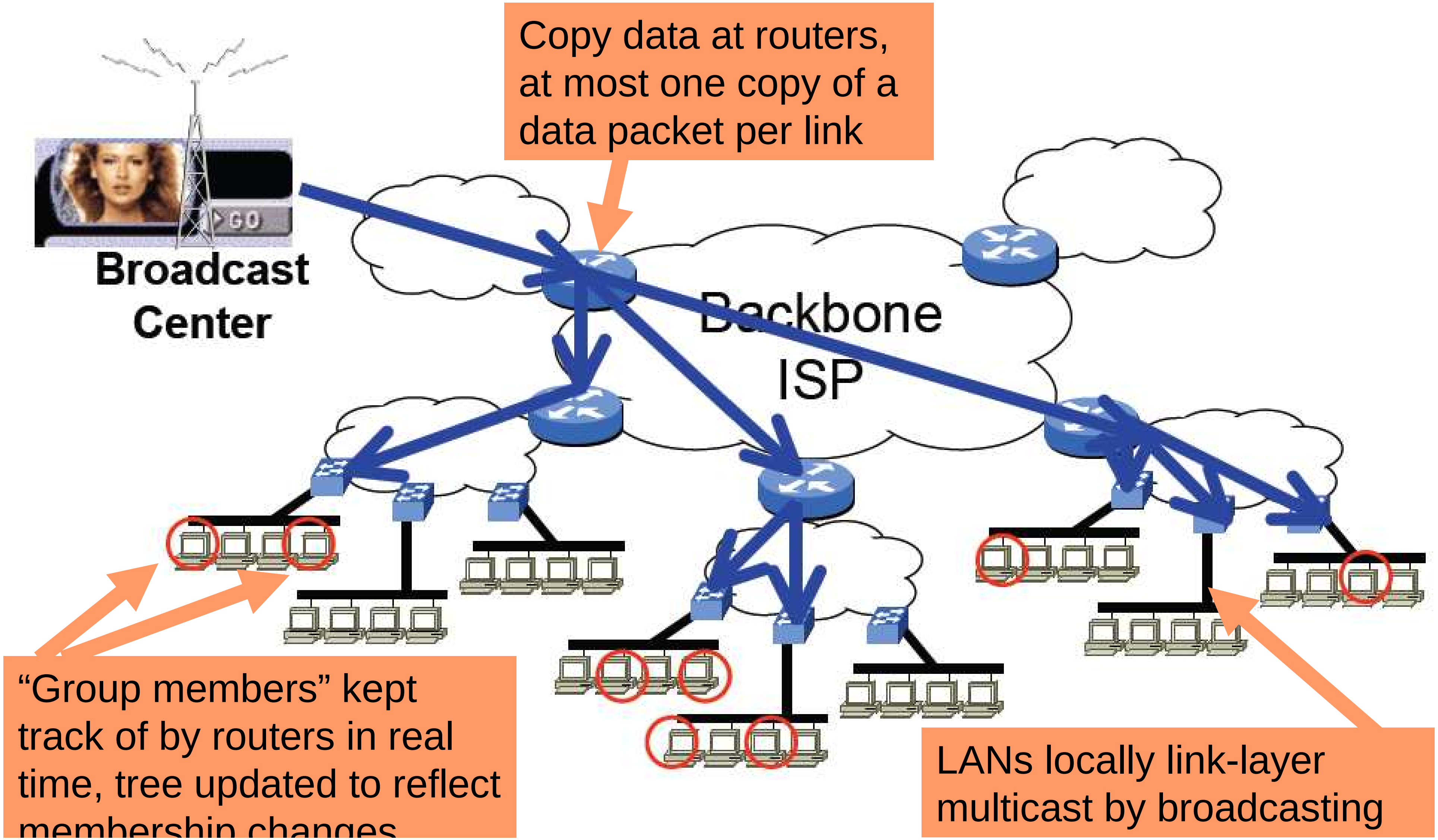
- Example: Live 8 concert
  - Send ~300 Kbps video streams
  - Peak usage > 100,000 simultaneous users
  - Consumes > 30 Gbps

- If 1000 people in UIUC, and if the concert is broadcast from a single location, then 1000 unicast streams are sent from that location to UIUC

Problem: this approach does not scale



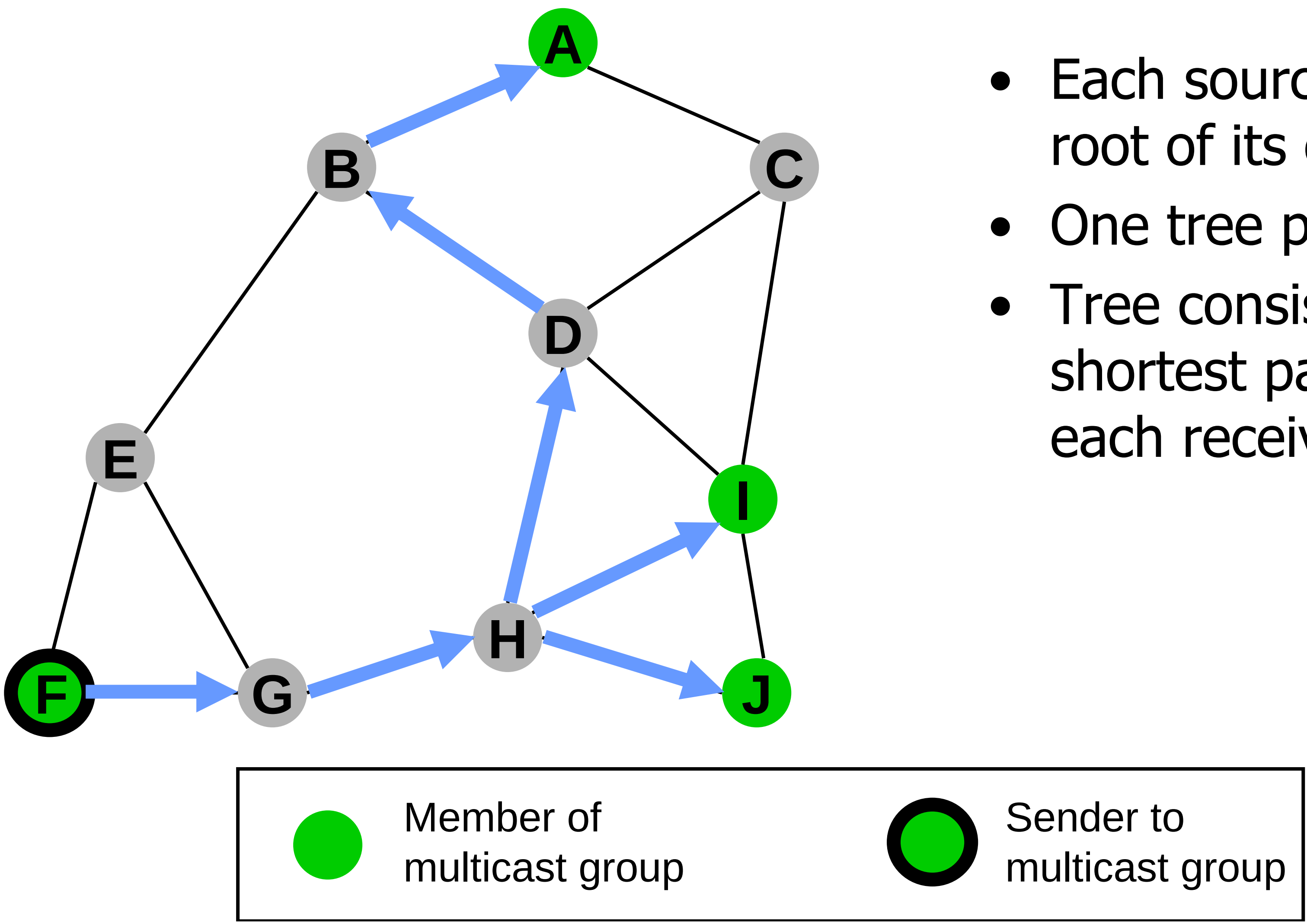
Alternative: build trees



Multicast routing approaches

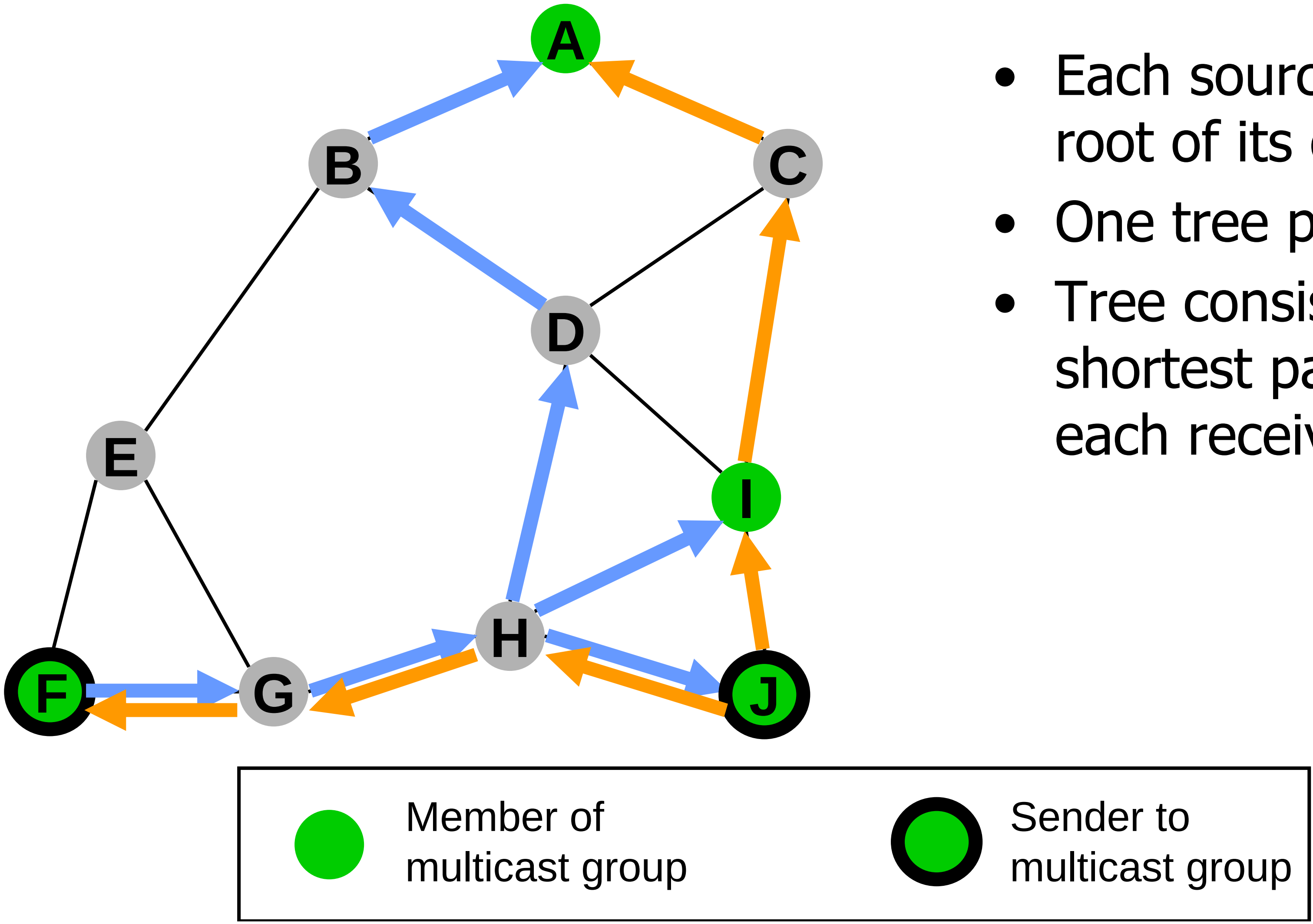
- Kinds of trees
  - Source-specific trees vs. Shared tree
- Layer
  - Data-link, network, application
- Tree computation methods
  - Link state vs. Distance vector

Source-specific trees



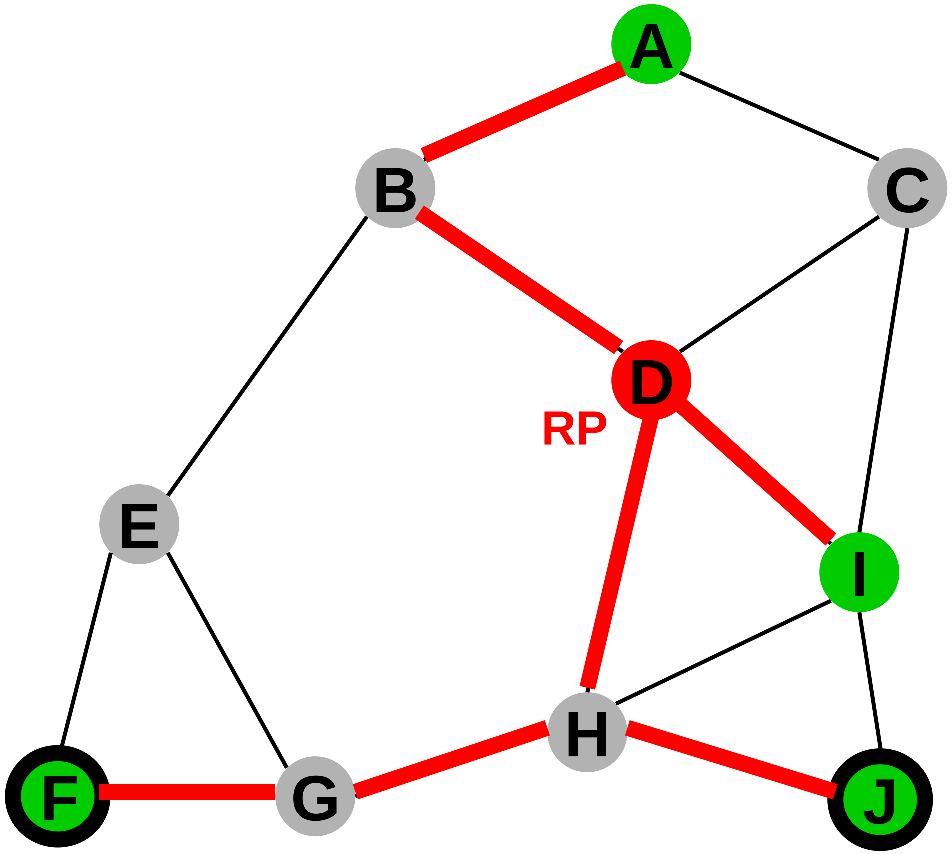
- Each source is the root of its own tree
- One tree per source
- Tree consists of shortest paths to each receiver

Source-specific trees



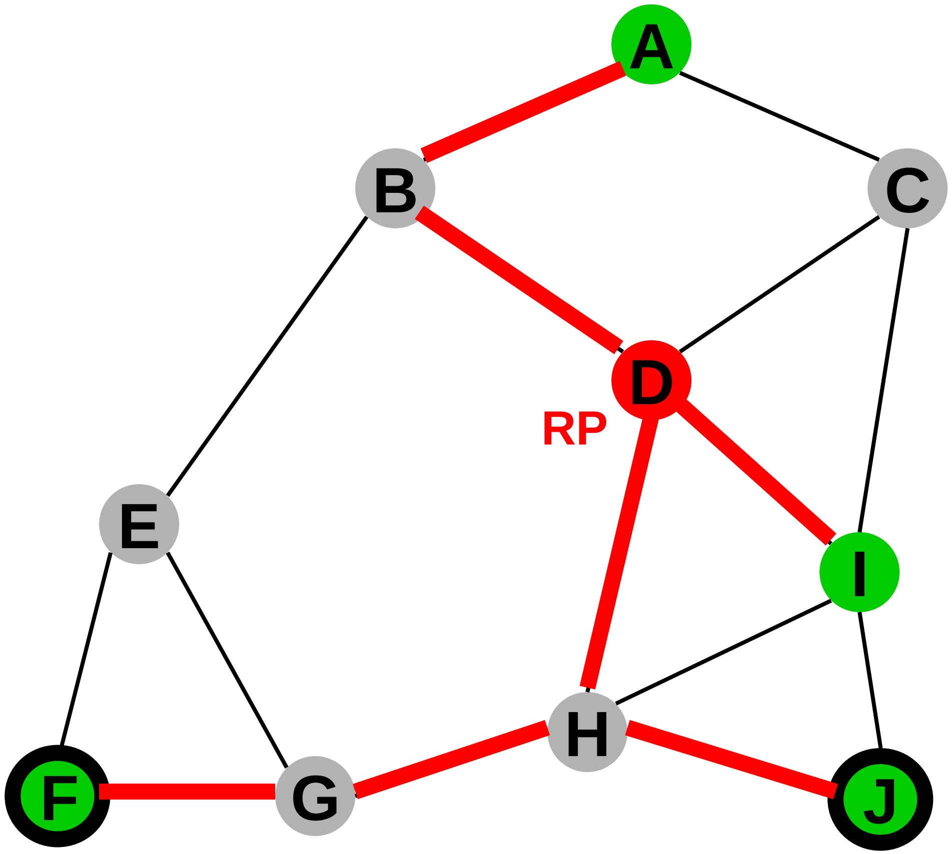
- Each source is the root of its own tree
- One tree per source
- Tree consists of shortest paths to each receiver

Shared Tree



- One tree used by all members of a group
- Rooted at "rendezvous point" (RP)
- Less state to maintain, but hard to pick a tree that's "good" for everybody!

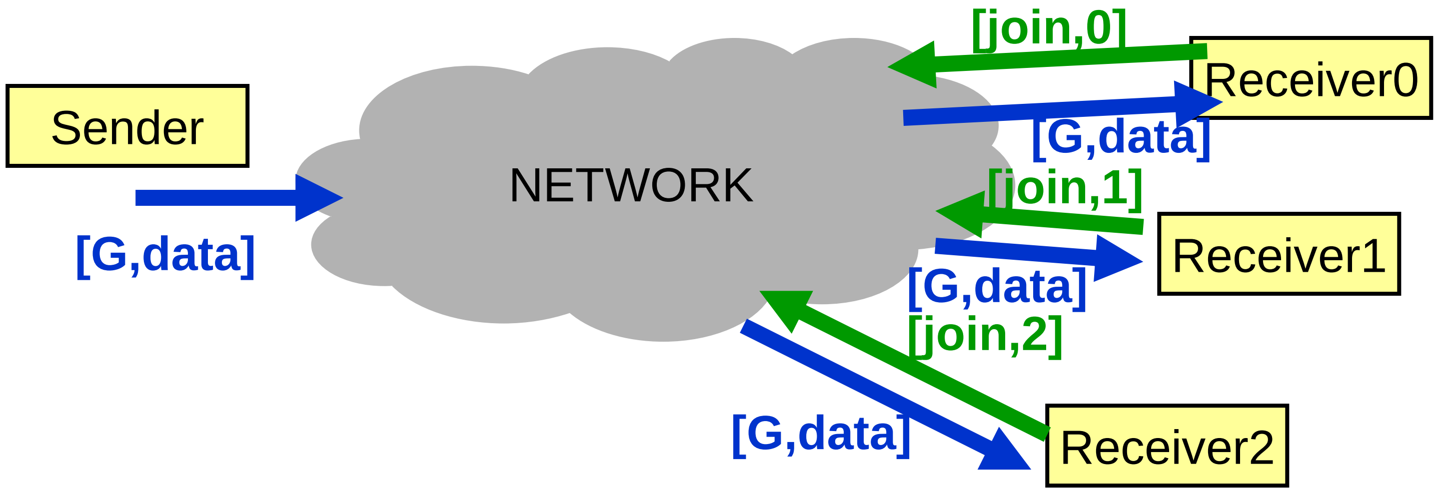
Shared Tree



- Ideally, find a "Steiner tree" minimum-weighted tree connecting **only** the multicast members
  - Unfortunately, this is NP-hard
- Instead, use heuristics
  - E.g., find a minimum spanning tree (much easier)

Multicast service model

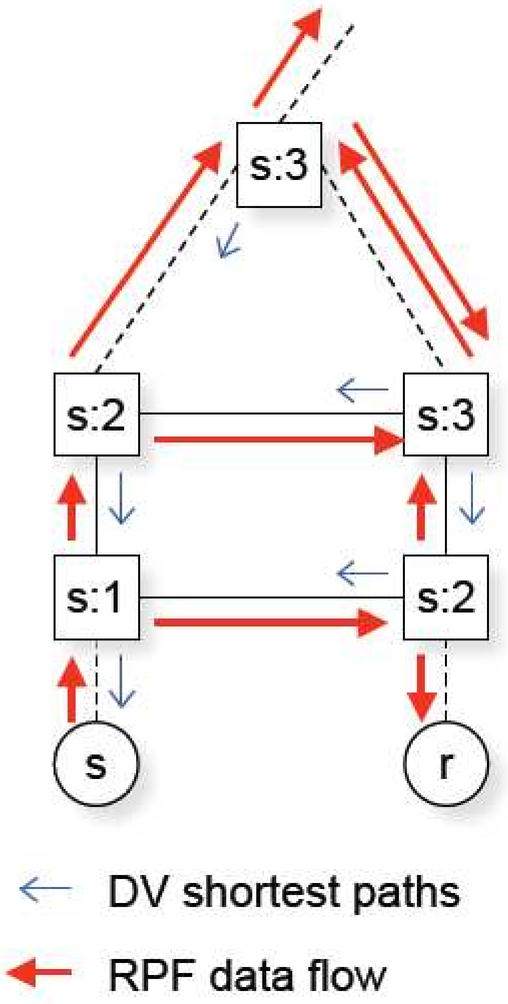
- Unicast: packets are delivered to one host
- Broadcast: packets are delivered to all hosts
- Multicast: packets are delivered to all hosts that have joined the multicast group
  - Multicast group identified by a multicast address



Concepts

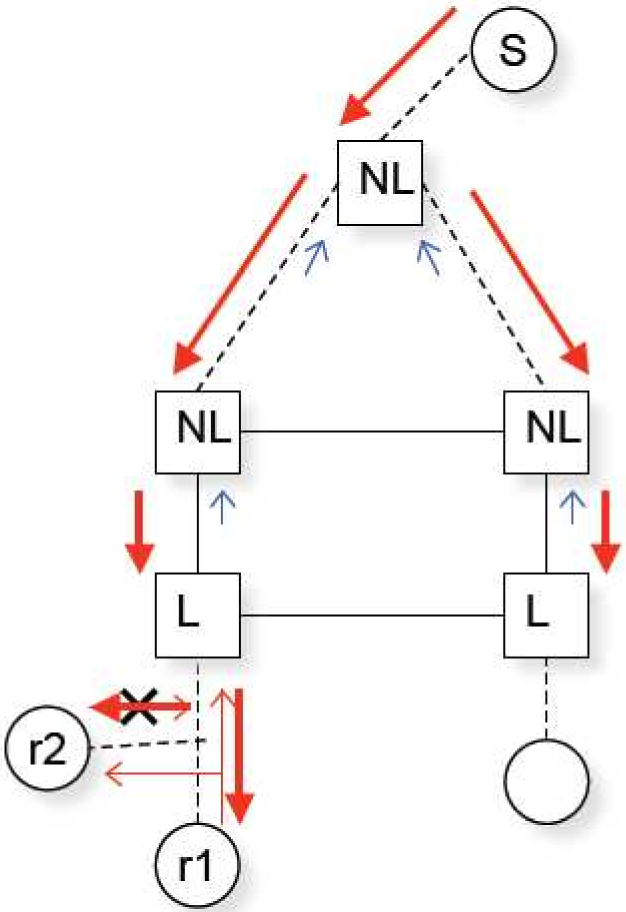
- Reverse-path forwarding
  - Regular routing protocols compute shortest path tree, so forward multicast packets along “reverse” of this tree
- Truncated reverse-path forwarding
  - Routers inform upstreams whether the upstream is on the router’s shortest path, to eliminate unnecessary broadcasting
- Flood-and-prune
  - Hosts must explicitly ask to not be part of multicast tree
  - Alternative: host explicitly sends “join” request to add self to tree

Reverse-path forwarding



- Extension to DV routing
- Packet forwarding
  - If incoming link is shortest path to source
  - Send on all links except incoming
  - Packets always take shortest path
    - Assuming delay is symmetric
- Issues
  - Routers/LANs may receive multiple copies

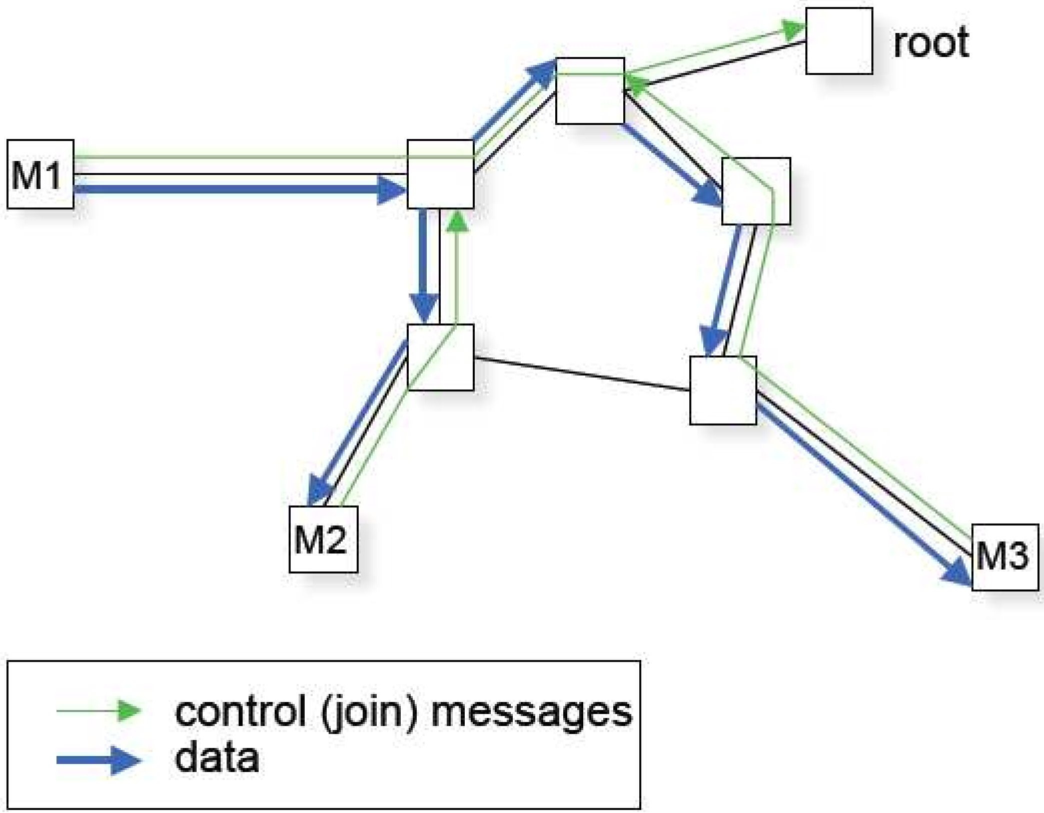
Truncated reverse-path forwarding



L – leaf node
NL – Non-leaf node

- Eliminate unnecessary forwarding
  - Routers inform upstreams if used on shortest path
  - Explicit group joining per-LAN
- Packet forwarding
  - If not a leaf router, or have members
  - Then send out all links except incoming

Core-based trees



- Pick a rendezvous point for each group (called the “core”)
- Unicast packet to core and bounce back to multicast groups
- Reduces routing table state
  - By how much?
  - O(S*G) to O(G)
- Finding optimal core location is hard (use heuristics)

Other IP Multicast protocols

- Three ways for senders and receivers to “meet”:
  - Broadcast membership advertisement from each receiver to entire network
    - example: MOSPF
  - Broadcast initial packets from each source to entire network; non-members prune
    - examples: DVMRP, PIM-DM
  - Specify “meeting place” to which sources send initial packets, and receivers join; requires mapping between multicast group address and “meeting place”
    - examples: PIM-SM
- What are some problems with IP-layer Multicast?

Problems with Network Layer Multicast

- Scales poorly with number of groups
  - Routers must maintain state for every group
  - Many groups traverse core routers
- Higher-layer functionality is difficult
  - NLM: **best-effort** delivery
  - Reliability, congestion control, transcoding for NLM complicated
- Deployment is difficult and slow
  - ISPs reluctant to turn on NLM

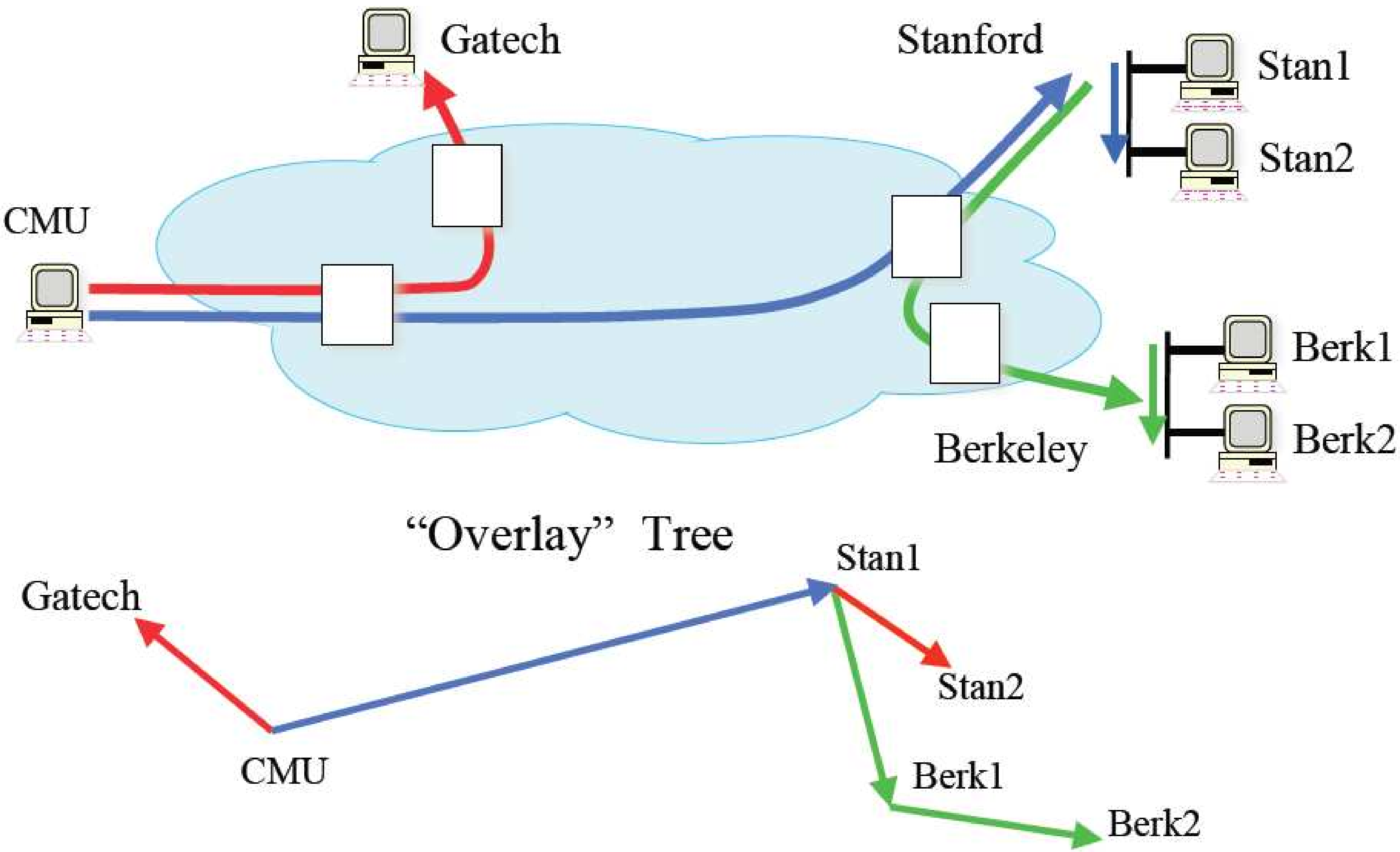
Problems with Network Layer Multicast

- Inconsistent with ISP charging model
  - Charging today is based on send rate of customer
  - But one multicast packet at ingress may cause millions to be sent on egress
- Troublesome security model
  - Anyone can send to group
  - Denial of service attacks on groups

Alternative: Application-layer Multicast

- Let hosts do all packet copying, tree construction
  - Only require unicast from network
  - Hosts construct unicast channels between themselves to form tree
- Benefits
  - No need to change IP, or for ISP cooperation
  - End hosts can prevent untrusted hosts from sending
  - Easy to implement reliability (per-hop retransmissions)
- Downsides
  - Stretch penalty (latency), Stress penalty (multiple retransmissions over same physical link

Example of Application-level Multicast



IPv6: proposed next generation of IP

- Problems with IPv4
  - Running out of address space
    - Projected depletion 2015-2032
  - Forwarding complicated by fragmentation, checksum computation, many unused fields
- IPv6 adopted by IETF in 1994
- IPv6 deployed incrementally, runs in parallel with IPv4
  - Routers distinguish packets based on version number

IPv6: Features → Larger address space

- 2^{128} (340,282,366,920,938,000,000, 000,000,000,000,000,000) addresses
- 2^{95} addresses for every person alive, 2^{52} addresses for every observable star in the universe
- But goal is to simplify addressing, not geographic saturation of devices
  - More hierarchical, systematic approach to allocation
  - Avoids need for splitting up prefixes, renumbering networks

IPv6: Features
→ Automatic host configuration

- IPv6 hosts probe network to discover gateway, acquire IPv6 address
- "Stateless": upstream router stores no per-host information
- Steps:
  - Host locally derives IP address from its MAC address
  - Broadcasts to make sure that IP address is not in use on the network
  - Contacts gateway router to get other configuration information
  - Host assigns itself IP address determined above

IPv6: Other features

- Simplified packet processing
  - Removed rarely used fields
  - Hosts must perform MTU discovery, fragmentation disallowed
  - IPv6 header has no checksum (integrity assumed to assured by transport level)
- Mobile IPv6 simplifies mobility
  - Maintains connectivity while end host moves
- Improved security
  - IPSec support is mandatory in IPv6

- What are the downsides of IPv6?

IPv6 Deployment challenges

- Requires infrastructure changes
  - What kind of changes?
  - Hardware: forwarding engines
  - Software: routing protocols

- However, certain strategies simplify deployment
  - Dual stack: routers run IPv4 and IPv6
  - IPv4 addresses can be mapped to IPv6 addresses
    - First 80 bits set to 0, next 16 set to one, final 32 are IPv4 address
  - Tunneling: IPv6 packets encapsulated in IPv4 packets

IPv6: Do we really need it?

- Larger address space
  - NAT reduces severity of address space depletion
  - Could just extend address sizes (IPv4+4)
- Simplified processing
  - Routers can do checksum processing in hardware at line speeds
- End host configuration, mobility, security
  - This functionality has been back-ported to IPv4
  - DHCP, IPSec, Mobile IP

Current state of IPv6

- *Prefix allocations* growing rapidly, but *traffic* showing no substantial growth
  - 500 allocations in 2004, 1200 allocations in 2009
  - Currently less than 1% of internet traffic is IPv6 (0.45% in USA, 0.24% in China)

- Possible reasons for slow growth
  - Lack of incentives (low demand from customers)
  - Not clear benefits are worth deployment efforts

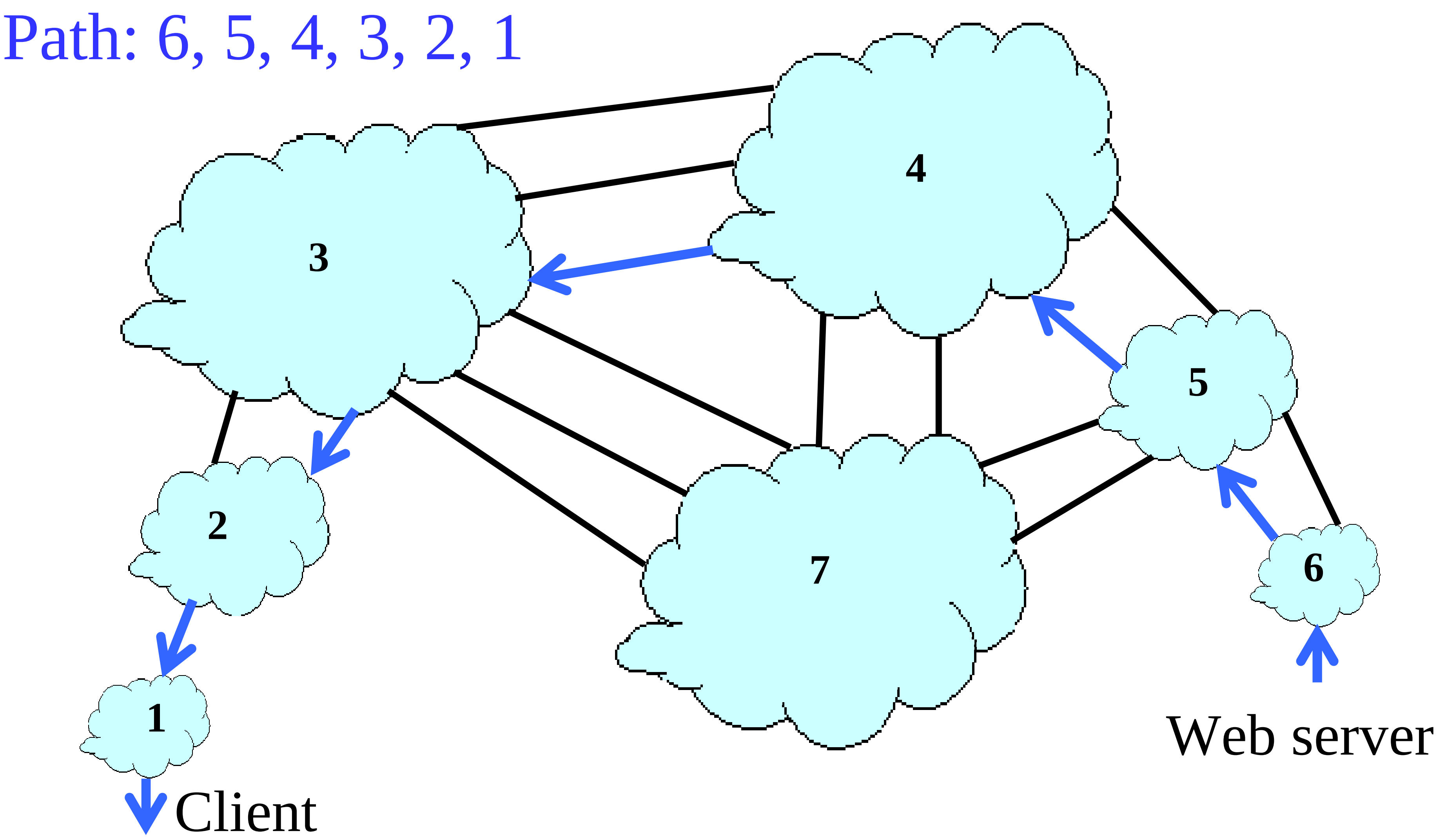
Roadmap

- IP Forwarding
  - Fragmentation and reassembly, ICMP, VPNs
  - Delivery models
- IP Routing
  - Routing across ISPs
  - How inter- and intra-domain routing work together
  - How Ethernet and intra-domain routing work together

Internet routing architecture

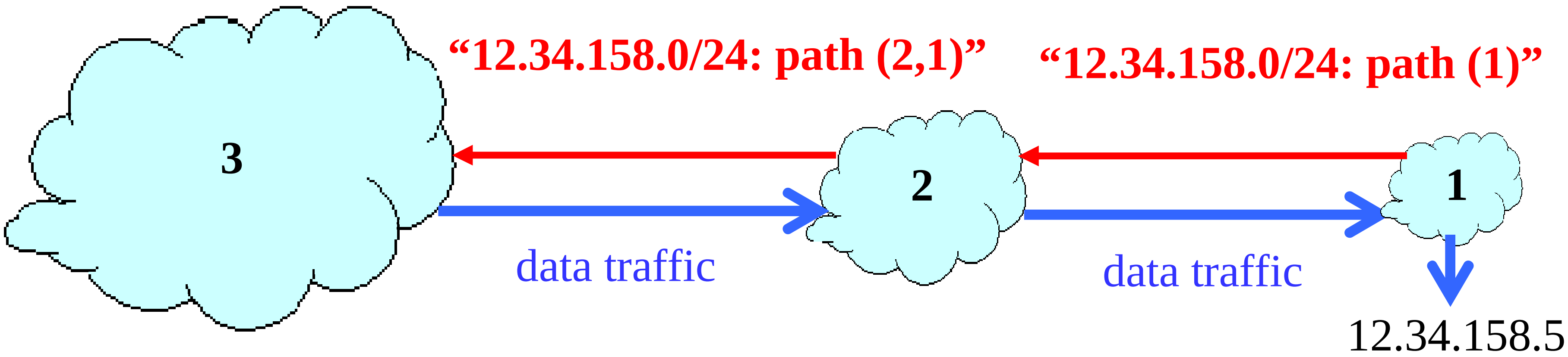
- Divided into ~30,000 *Autonomous Systems*
  - Distinct regions of administrative control
  - Routers/links managed by single "institution"
  - ISP, company, university
- Hierarchy of Autonomous Systems
  - Large, tier-1 providers with nationwide backbone
  - Medium-sized regional provider with smaller backbone
  - Small network run by company or university
- Interaction between Autonomous Systems
  - Internal topology is not shared between ASes
  - But, neighboring ASes interact to coordinate routing

Interdomain routing

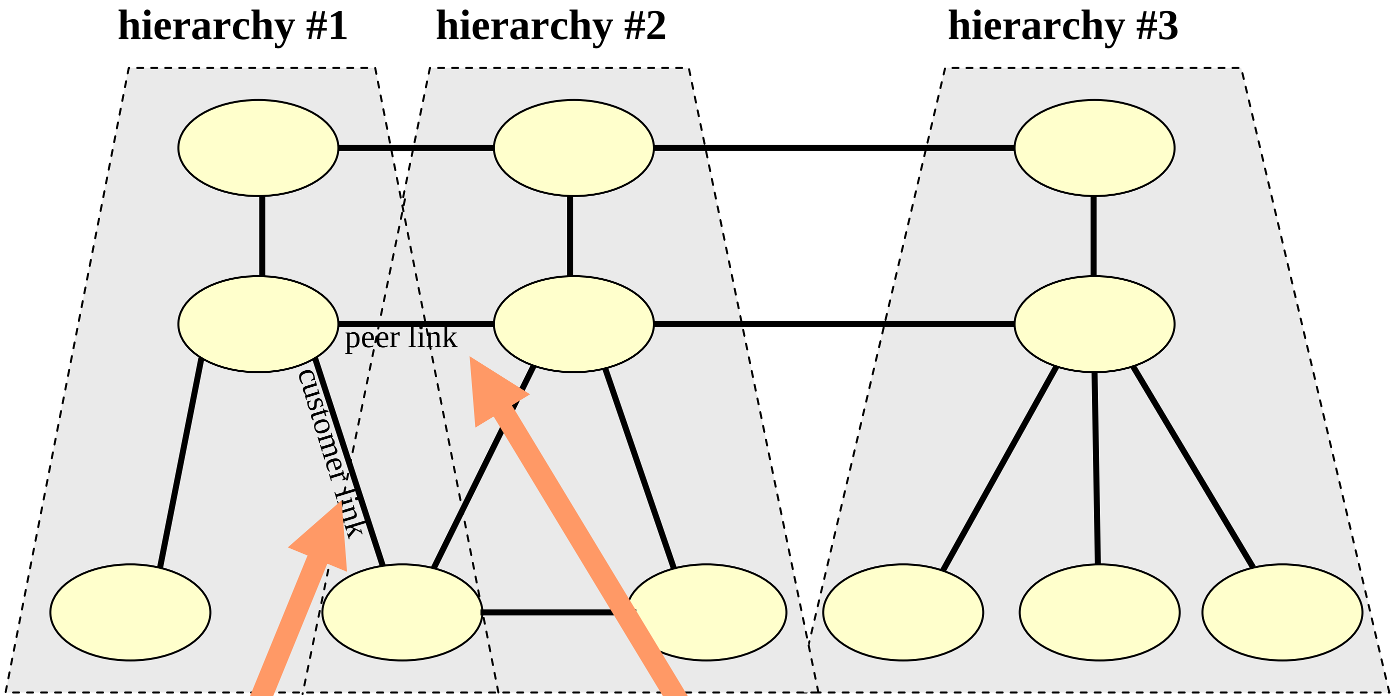


Interdomain routing: the Border Gateway Protocol (BGP)

- ASes exchange information about which IP prefixes they can reach using *path vector*
  - Propagate (IP prefix, AS-path) pairs
- Policies configured by AS's operator
  - Path selection: which of paths to use?
  - Path export: which neighbors to tell about path?



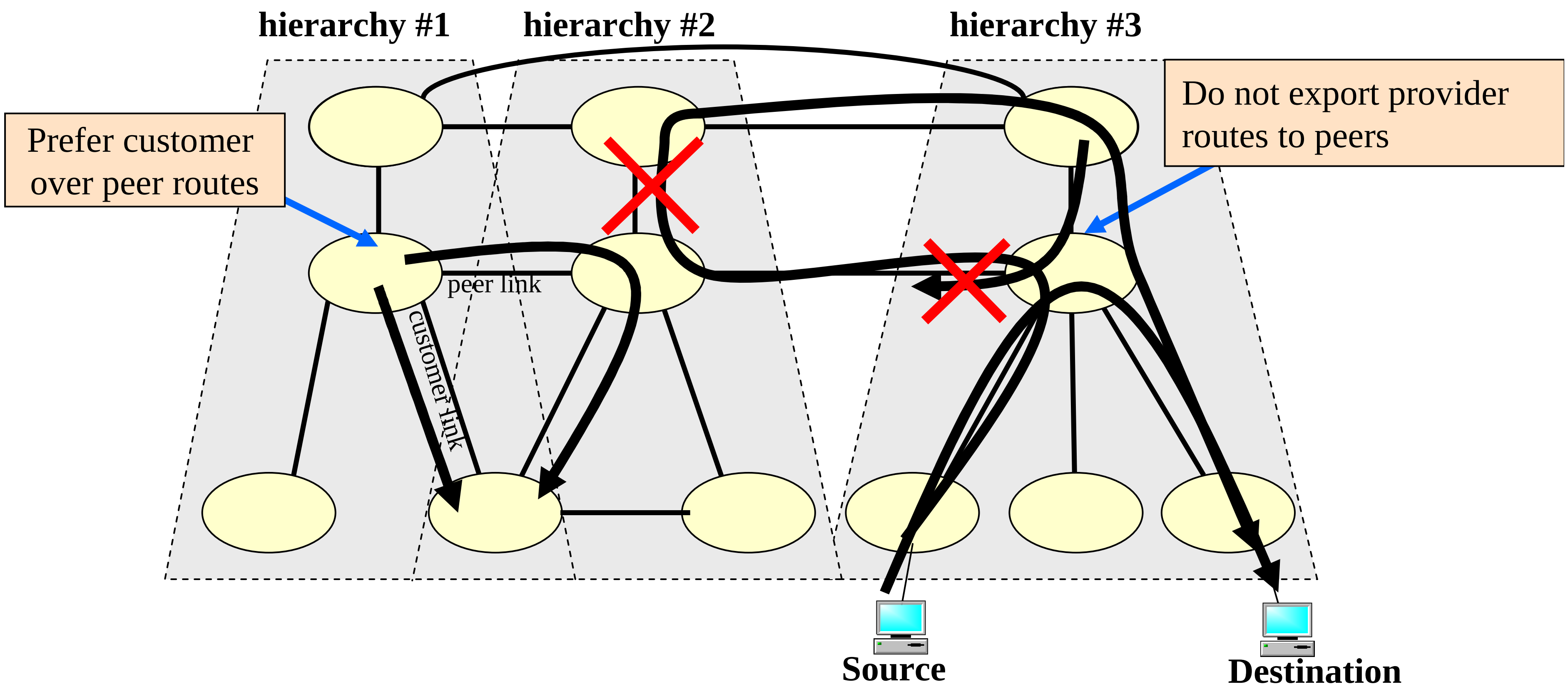
Types of AS relationships



Provider-customer: customer pays provider money to transit traffic

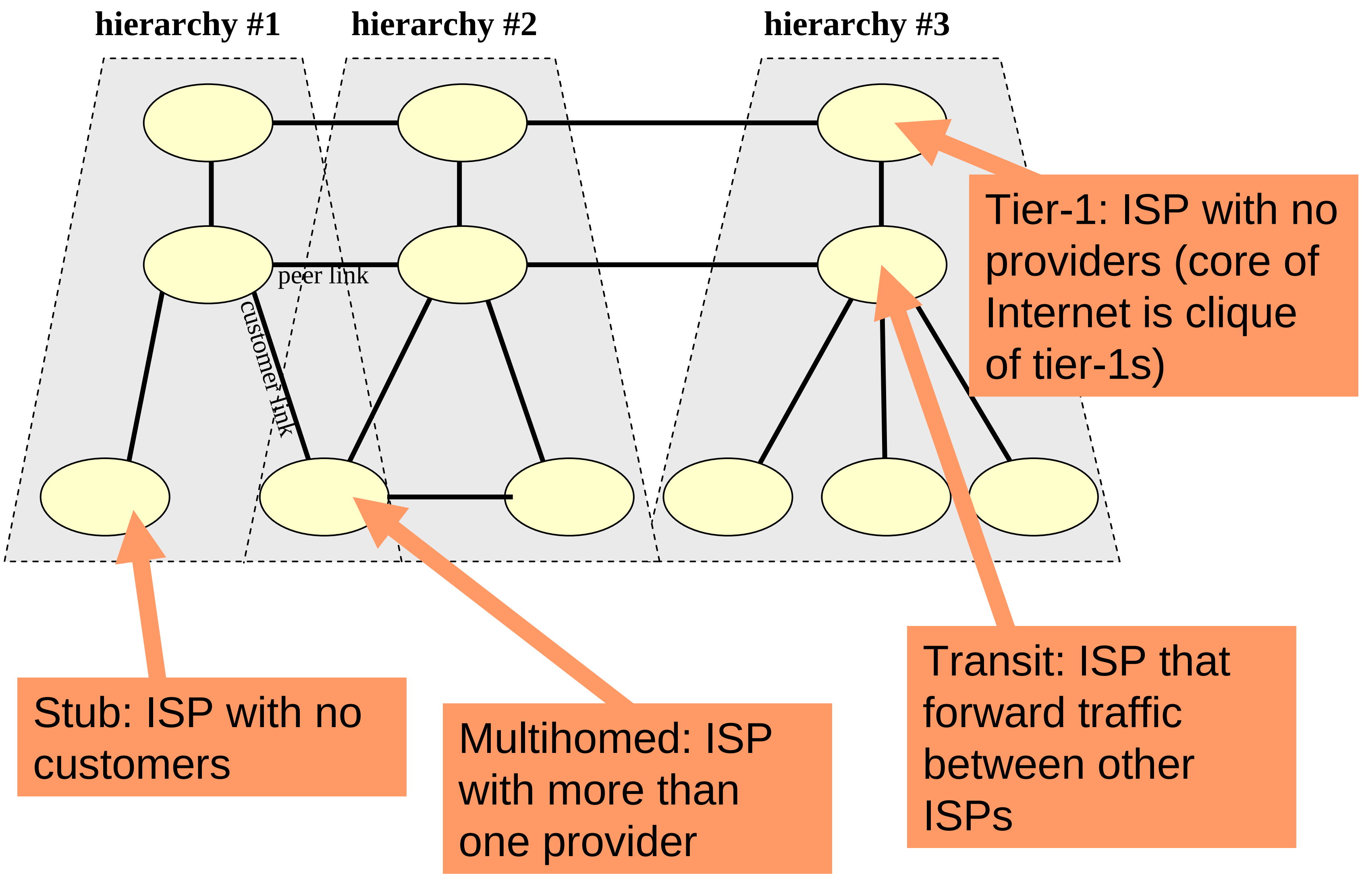
Peer link: ISPs form link out of mutual benefit, typically no money is exchanged

Policies between ISPs

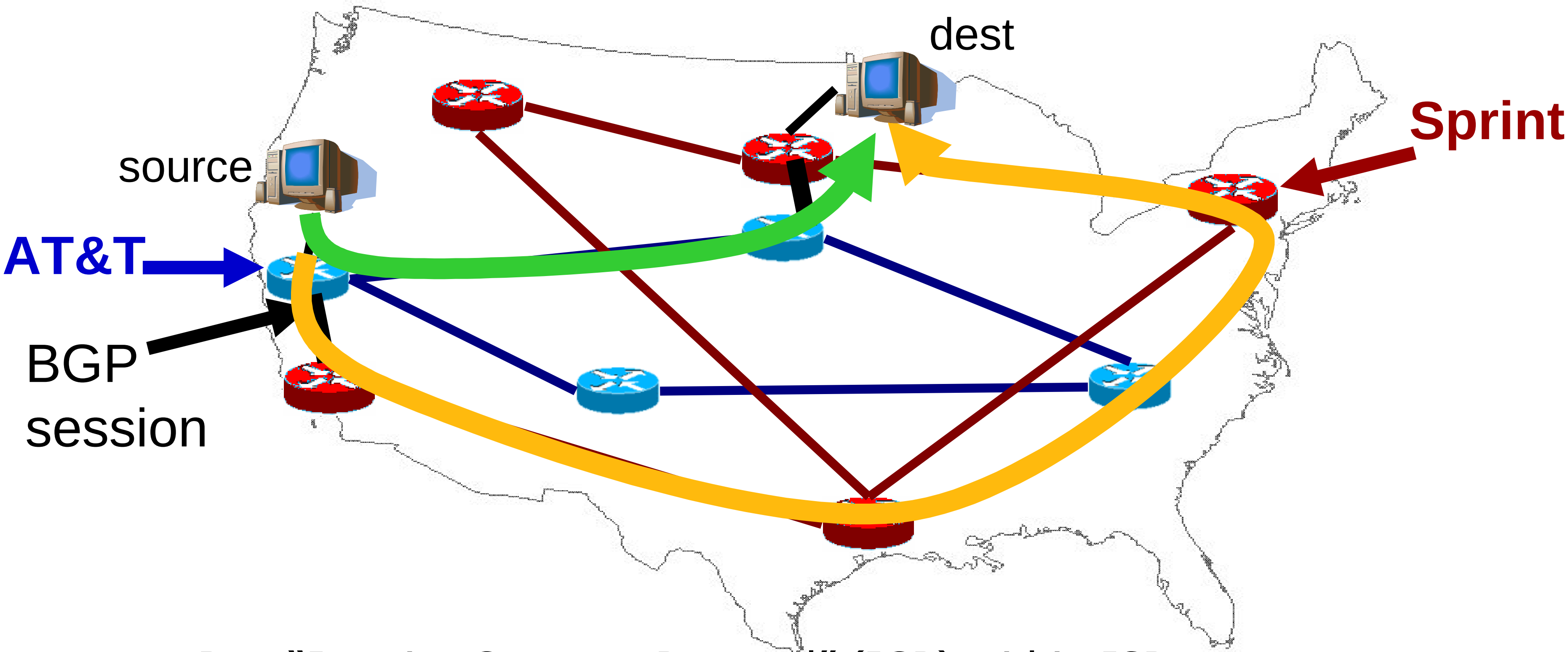


- Example policies: peer, provider/customer
- Also trust issues, security, scalability, traffic engineering

Types of ASes

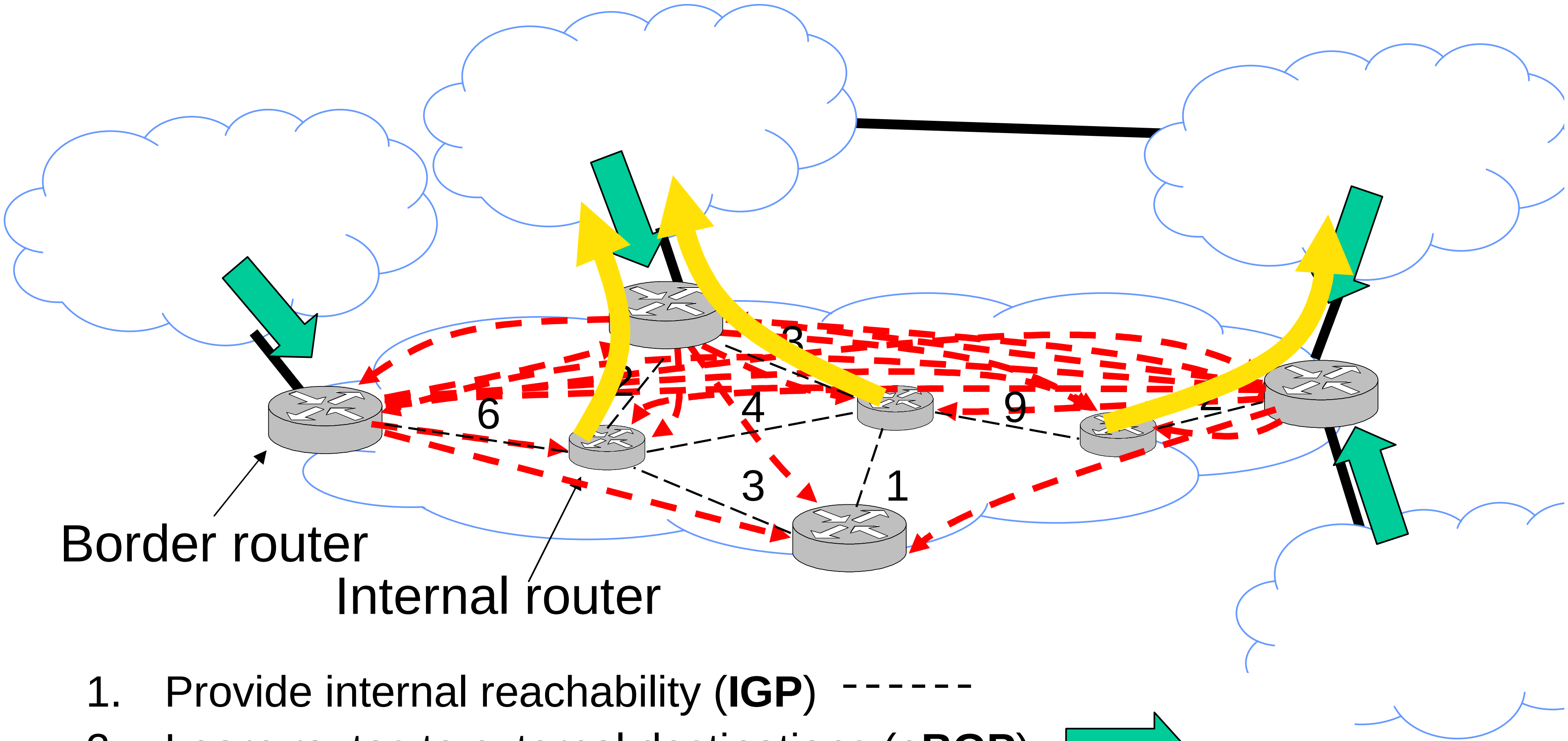


Intra- vs. Inter-domain routing



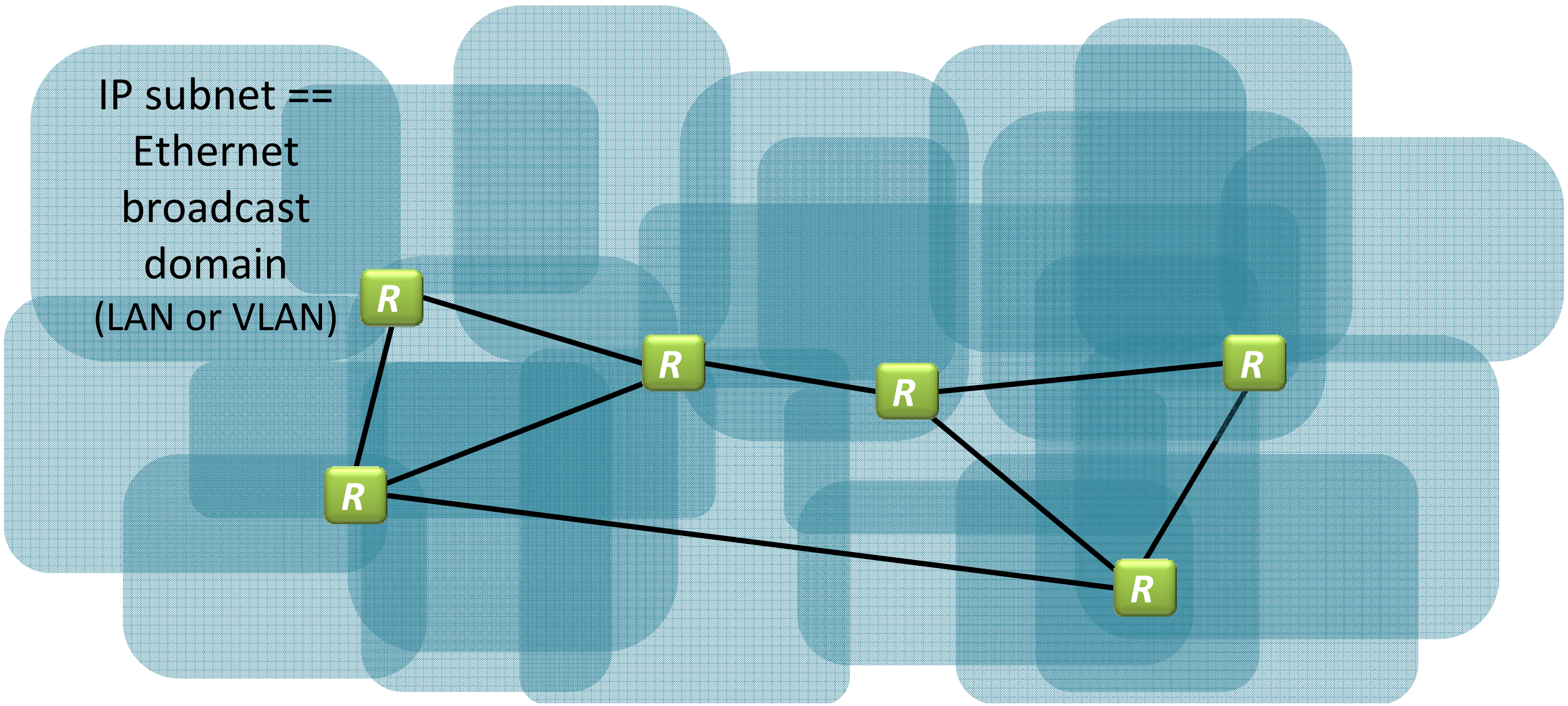
- Run "Interior Gateway Protocol" (IGP) within ISPs
  - OSPF, IS-IS, RIP
- Use "Border Gateway Protocol" (BGP) to connect ISPs
  - To reduce costs, peer at exchange points (AMS-IX, MAE-EAST)

How inter- and intra- domain routing work together



1. Provide internal reachability (**IGP**) ------
2. Learn routes to external destinations (**eBGP**)
3. Distribute externally learned routes internally (**iBGP**)
4. Select closest egress (**IGP**) ------

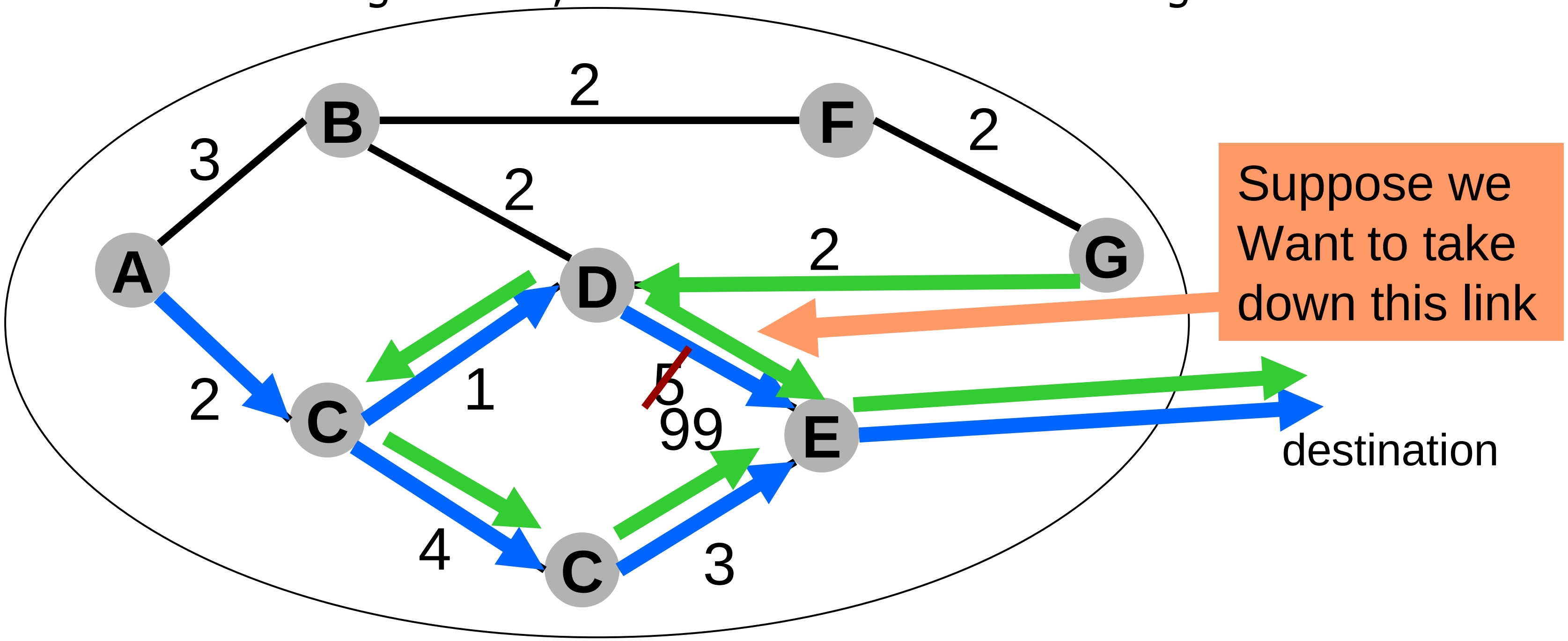
How Ethernet and Intra-domain routing work together



Current practice: a hybrid architecture comprised of small Ethernet-based IP subnets connected by routers

“Costing out” of equipment

- Increase cost of link to high value
  - Triggers immediate flooding of LSAs
- Leads to new shortest paths avoiding the link
  - While the link still exists to forward during convergence
- Then, can safely disconnect the link
  - New flooding of LSAs, but no influence on forwarding

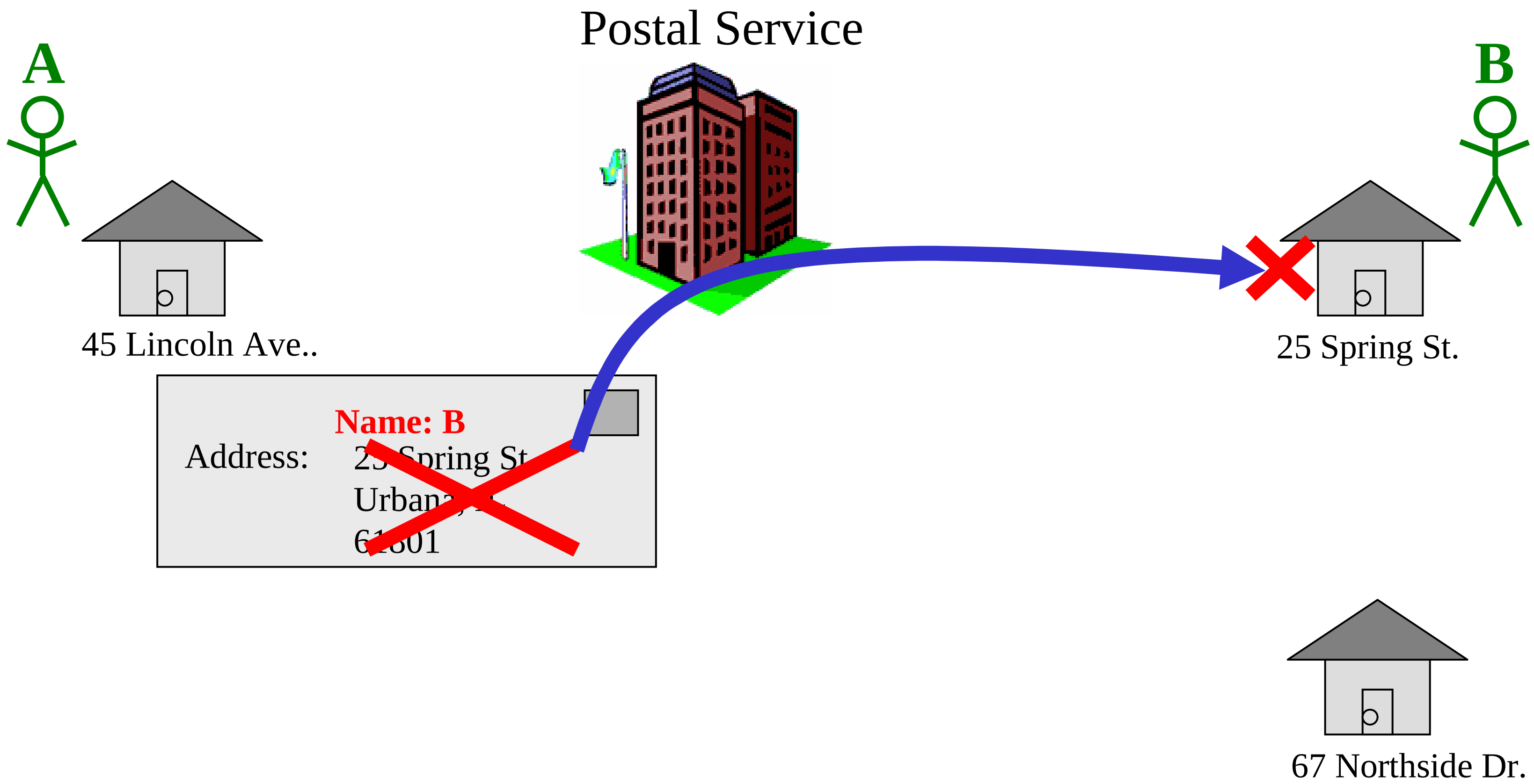


Naming and Addressing

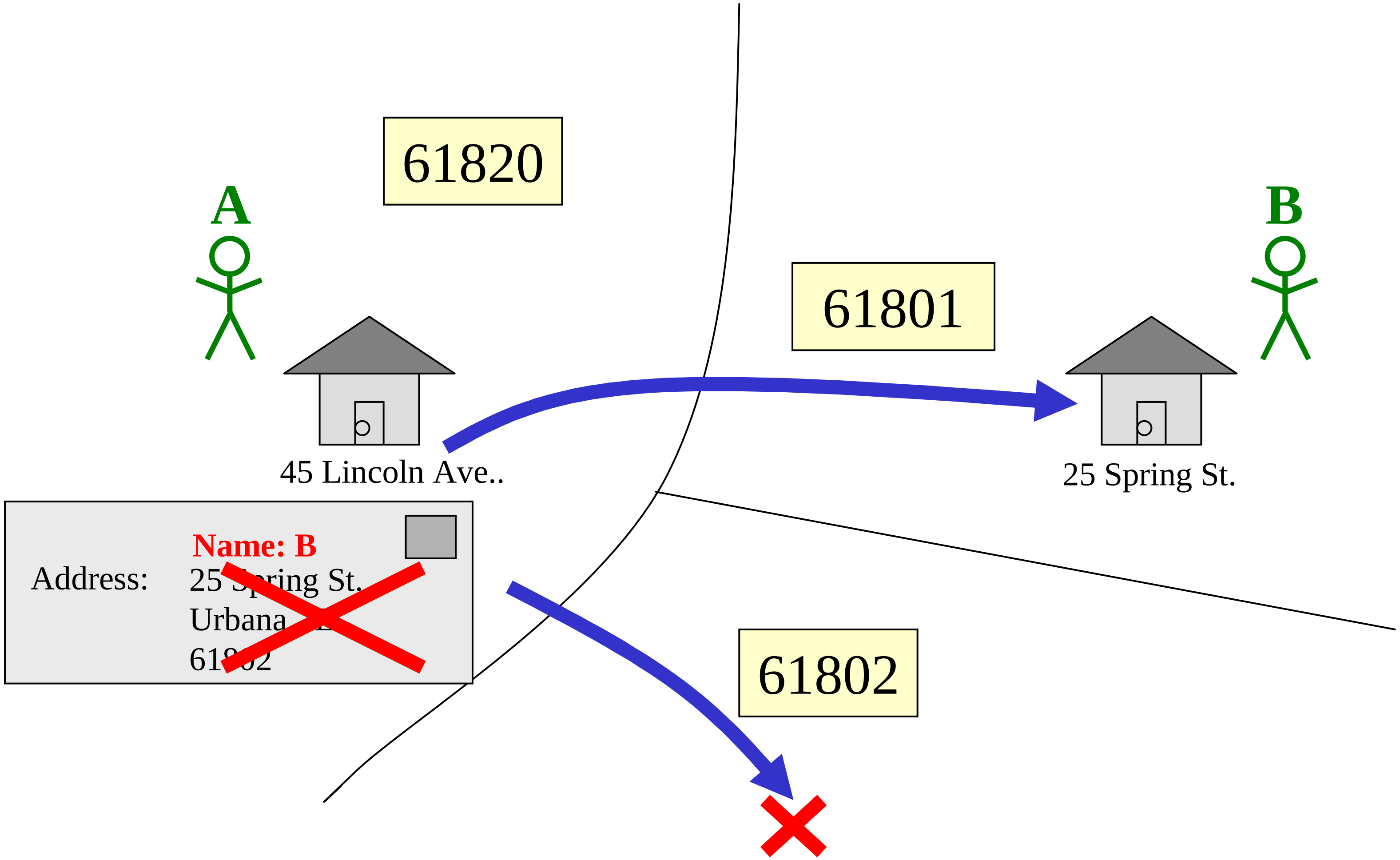
Roadmap

- Addresses
  - Assignment: CIDR, DHCP
  - Translation: ARP, NAT
  - Host mobility
- Names
  - Translation to addresses

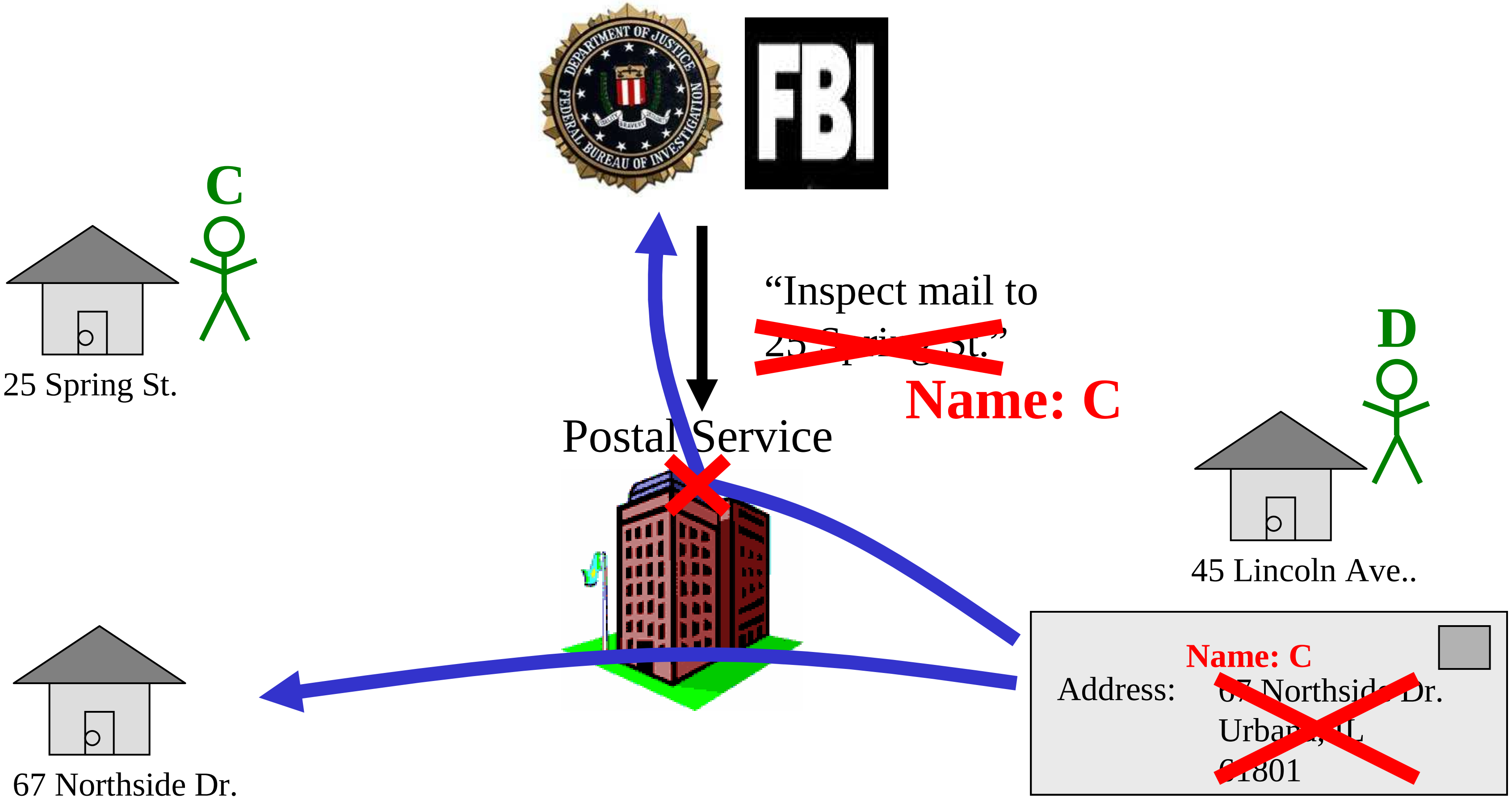
Scenario: Sending a Letter



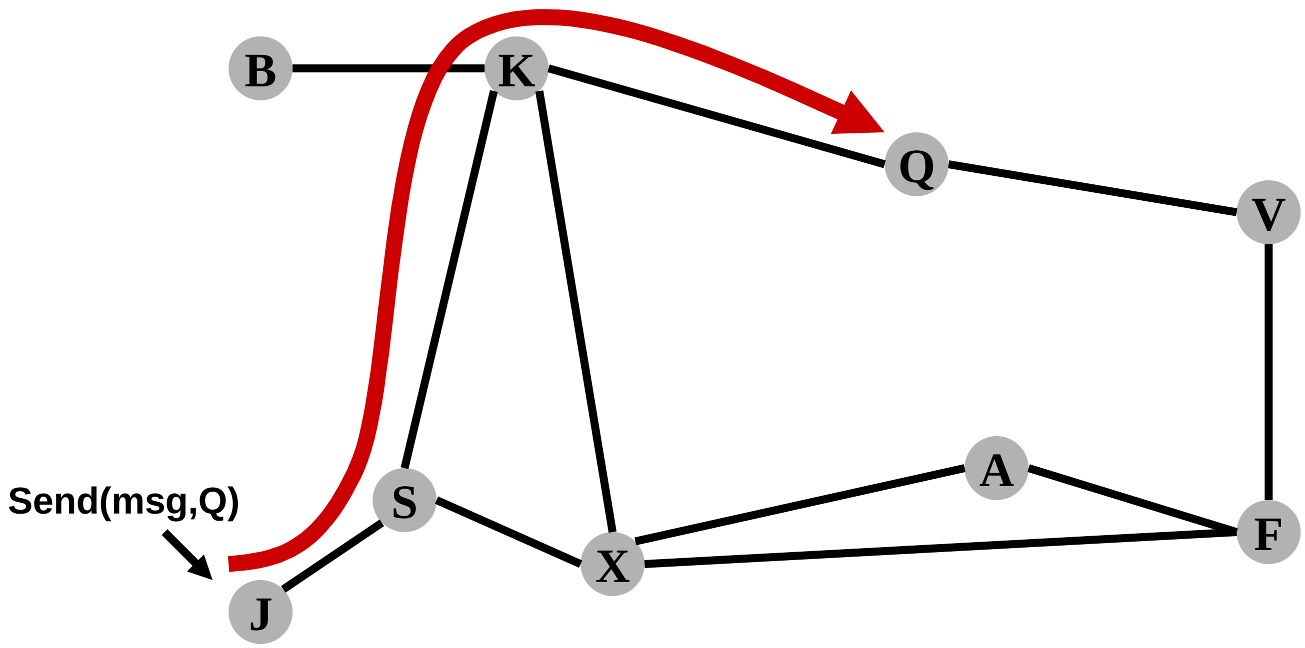
Scenario: Address Allocation



Scenario: Access Control

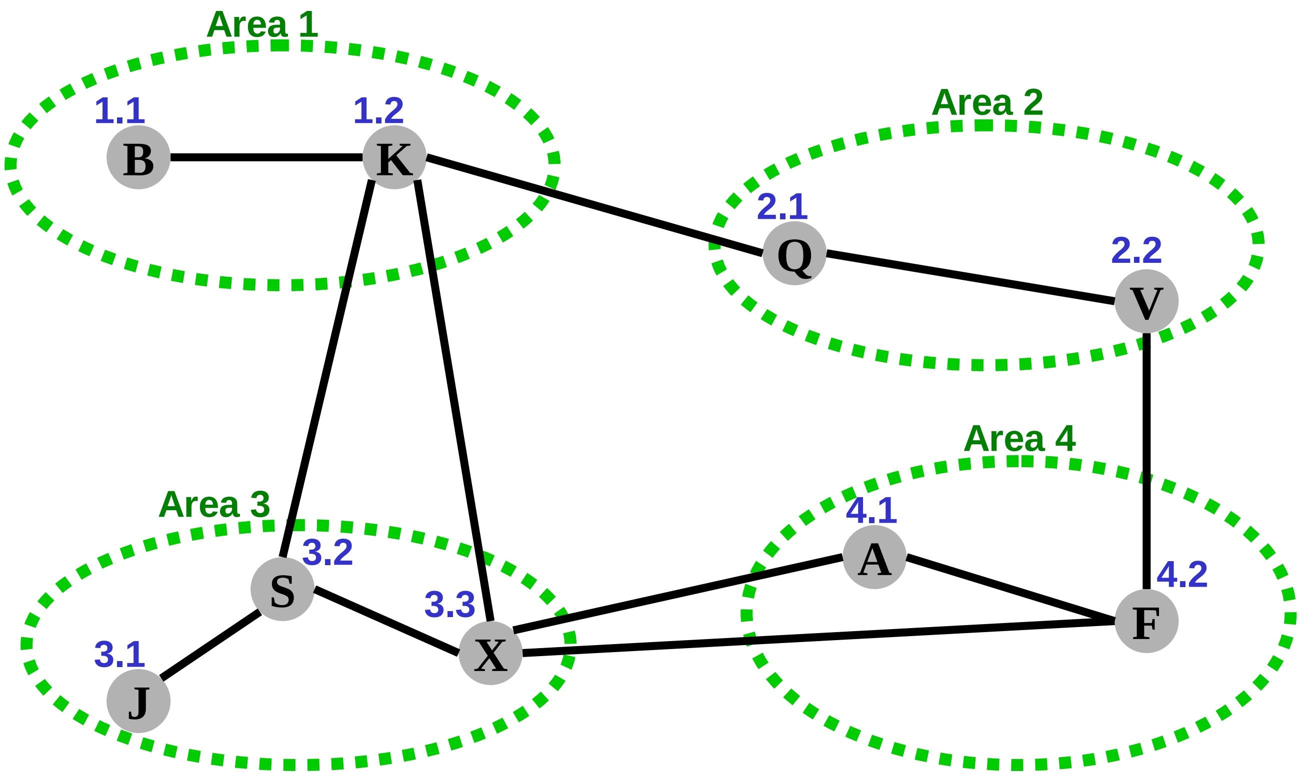


How Routing Works Today



- Each node has an address
- Goal: find path to destination

Scaling Requires Aggregation



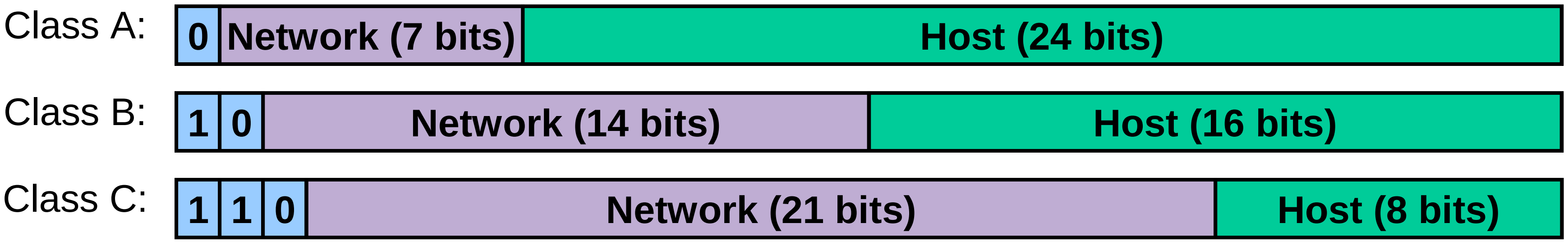
- Pick addresses that depend on location
- Aggregation provides excellent scaling properties
- Key is topology-dependent addressing!

IPv4 Address Model

- Properties
  - 32 bits, often written in dotted decimal notation (e.g. 72.14.205.147)
  - Maps to logically unique network adaptor
    - Exceptions: NAT, load balancing servers
  - Assigned based on position in topology
    - Simplifies routing
- Routers advertise *IP prefixes*
  - Aggregated blocks of IP addresses
  - Used to reduce router state requirements
  - Written in form *network/subnet* (e.g. 72.14.0.0/16) or with *number/mask* (e.g. 72.14.0.0/255.255.0.0)

Assigning IPv4 addresses

- Allocation: IANA → regional registries (eg. ARIN) → ISPs → ISP's customers (eg. ISPs or enterprises)
- Pre-1993: Classful addressing



- Downside of classful addressing:
  - Wasted address space
  - Renumbering is time consuming and can interrupt service

CIDR

- Current approach: Classless Inter-Domain Routing (CIDR)
  - Allows variable-length classes
  - When two alternatives, route to longer prefix match
- Subnetting
  - Share one address (network number) across multiple physical networks
- Supernetting/Aggregation
  - Aggregate multiple addresses (network numbers) for one physical network
- Downsides:
  - Supernetting often disabled to avoid unintended side effects, bloating routing tables
  - Multihomed sites don’t benefit from supernetting

Longest prefix match forwarding

- Multiple prefixes may "cover" the packet's destination
  - Used for load-balancing, failover
  - Router identifies longest-matching prefix, sends to corresponding interface

forwarding table

destination

12.34.158.5 →

4.0.0.0/8
4.83.128.0/17
12.0.0.0/8
12.34.158.0/24 →
126.255.103.0/24

outgoing link

Serial0/0.1

CIDR

- Allows hierarchical development
  - Assign a block of addresses to a regional provider
    - Ex: 128.0.0.0/9 to BARRNET
  - Regional provider subdivides address and hands out block to sub-regional providers
    - Ex: 128.132.0.0/16 to Berkeley
  - Sub-regional providers can divide further for smaller organizations
    - Ex: 128.132.32.0/1 to Berkeley Computer Science Department

Subnetting

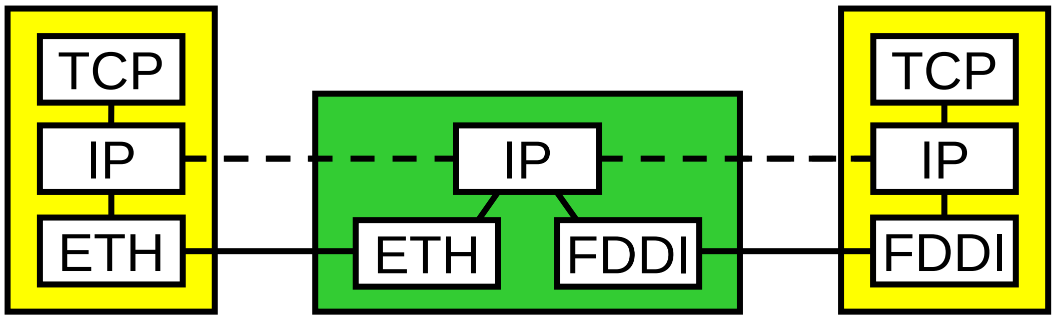
- Simple IP
  - All hosts on the same network must have the same *network* number
- Assumptions
  - Subnets are close together
    - Look like one network to distant routers
- Idea
  - Take a single IP network number
  - Allocate the IP addresses to several physical networks (subnets)
- Subnetting
  - All hosts on the same network must have the same *subnet* number

Subnet Example

- Solution
  - Partition the 65,536 address in the class B network
    - 256 subnets each with 256 addresses
    - Subnet mask: 255.255.255.0
  - If 135.104.5.{1,2,3} are all on the same physical network reachable from router 135.105.4.1
    - There only needs to be one routing entry for 135.104.5.* pointing to 135.105.4.1 as next hop

IPv4 Address Translation support

- IP addresses to LAN physical addresses
- Problem
  - An IP route can pass through many physical networks
  - Data must be delivered to destination's physical network
  - Hosts only listen for packets marked with physical interface names
    - Each hop along route
    - Destination host

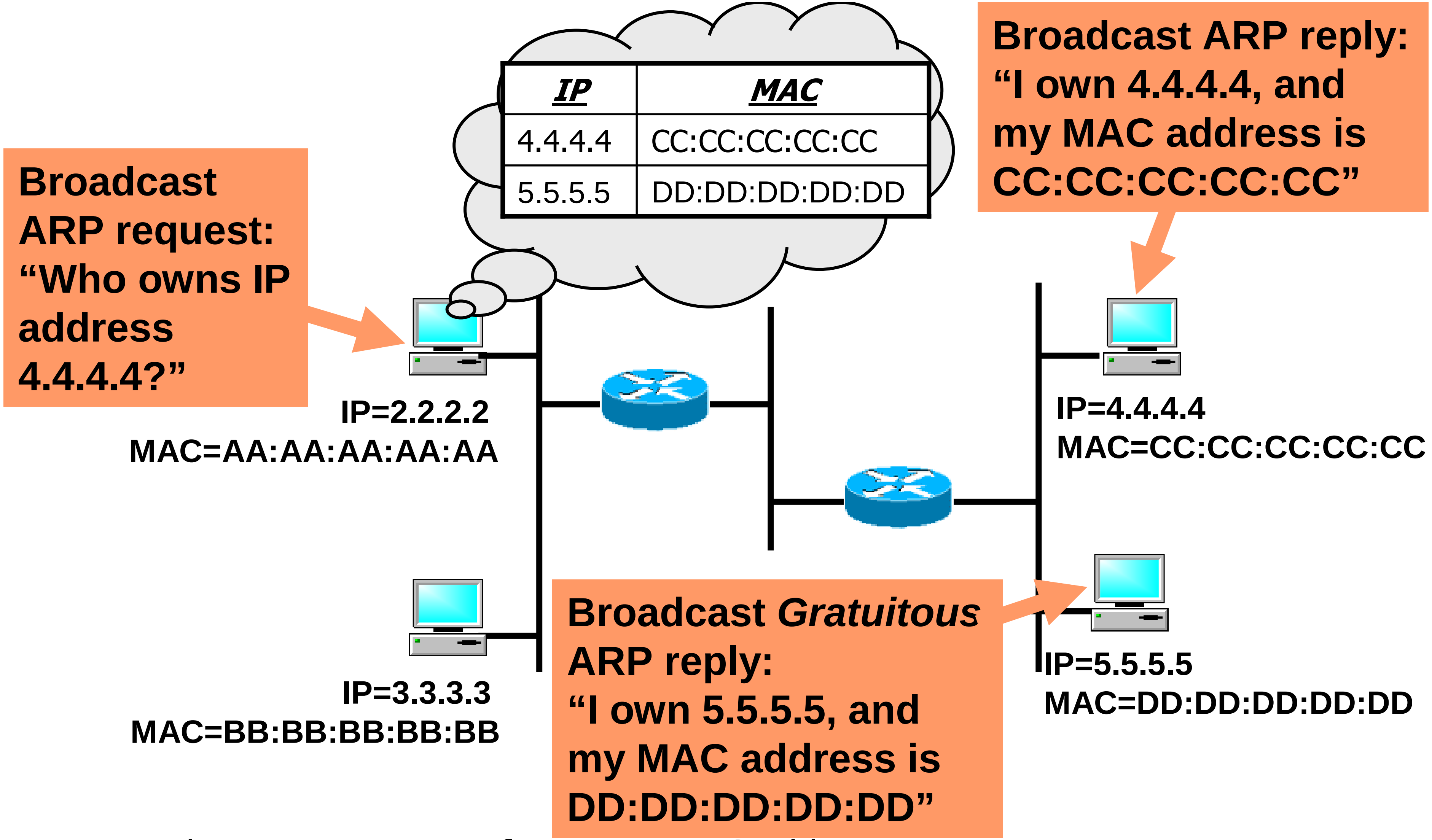


IP to Physical Address Translation

- Hard-coded
  - Encode physical address in IP address
  - Ex: Make IP address equal to lower 32 bits of Ethernet address.
  - Problems:
    - Uniqueness, hard to associate address with topology
- Fixed table
  - Maintain a central repository and distribute to hosts.
  - Problems:
    - Bottleneck for queries and updates
- Solution: Automatically generated table
  - Use ARP to build table at each host
  - Use timeouts to clean up table

ARP: Address Resolution Protocol

- Check ARP table for physical address
- If address not present
  - Broadcast an ARP query, include querying host's translation
  - Wait for an ARP response
- Upon receipt of ARP query/response
  - Targeted host responds with address translation
  - If address already present
    - Refresh entry and reset timeout
  - If address not present
    - Add entry for requesting host
    - Ignore for other hosts
- Timeout and discard entries after O(10) minutes



- 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

Host Configuration

- Plug new host into network
  - Host needs an IP address
  - Host must also
    - Send packets out of physical (direct) network
    - Thus needs physical address of “gateway” router

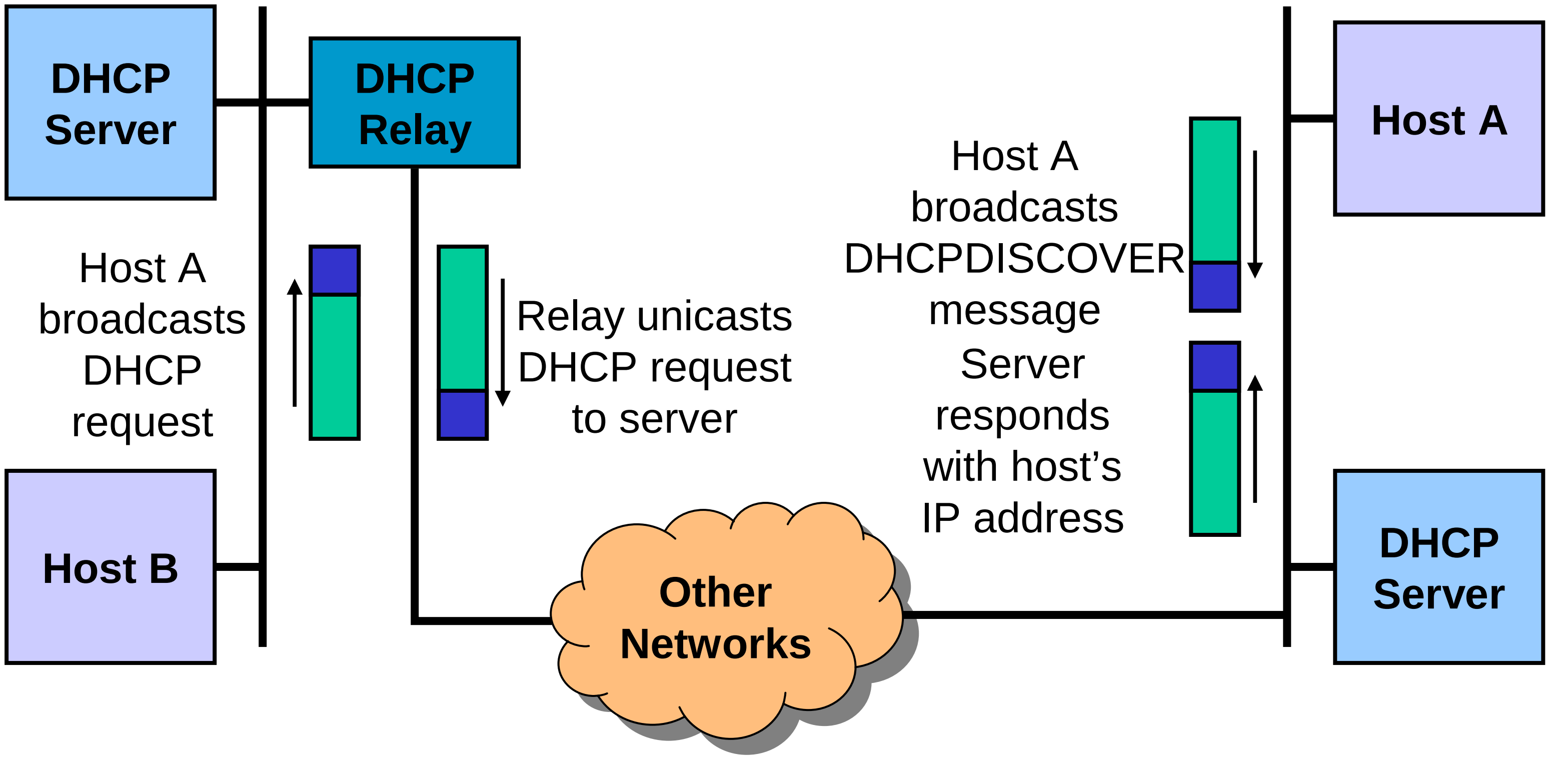
Dynamic Host Configuration Protocol (DHCP)

- A simple way to automate host configuration
  - Network administrator does not need to enter host IP address by hand
  - Good for large and/or dynamic networks
- New machine sends request to DHCP server for assignment and information

Dynamic Host Configuration Protocol (DHCP)

- Server receives
  - Directly if new machine given server’s IP address
  - Through broadcast if on same physical network
  - Or via DHCP relay nodes
    - Forward requests onto the server’s physical network
- Server assigns IP address and provides other info
- Can be made secure
  - Present signed request or just a “valid” physical address
- Remaining challenge: configuring DHCP servers
  - Need to ensure consistency across servers, between servers and network, address assignment across routers
  - But simpler than directly managing end hosts

DHCP


DHCP Server
DHCP Relay
Host B
Host A broadcasts DHCP request
Relay unicasts DHCP request to server
Other Networks
Host A broadcasts DHCPDISCOVER message
Server responds with host's IP address
Host A
DHCP Server

DNS

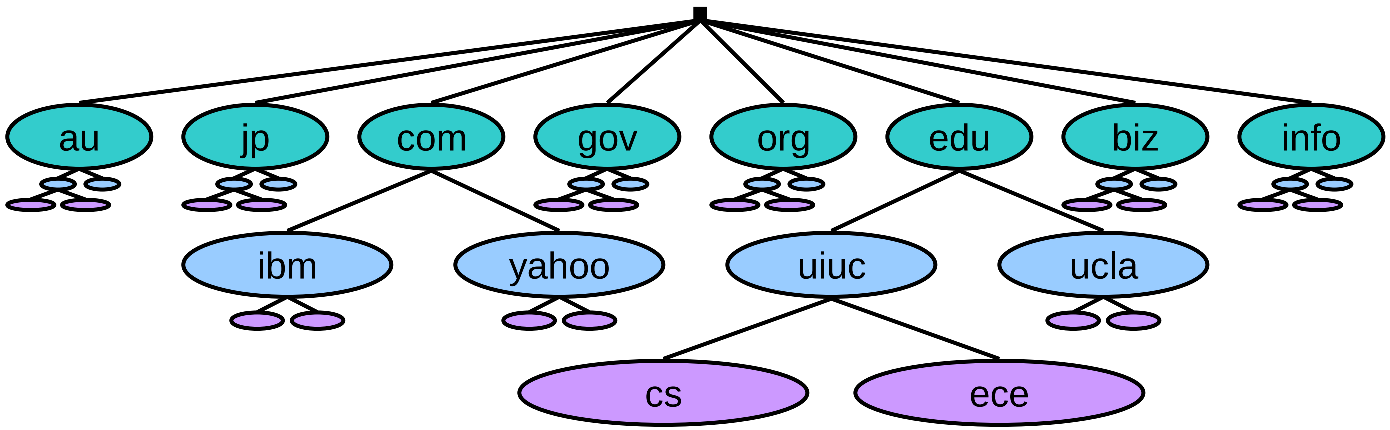
- A system to resolve from human readable names to IP addresses
  - E.g., www.yahoo.com → 209.191.93.52
- Namespace (set of possible names)
  - Host names, domain names
- History: pre-1983, hosts used to retrieve HOSTS.TXT from computer at SRI
  - What's wrong with this?

Comparison of domain names and IP addresses

- Internet domain names
  - Human readable
  - Variable length
  - Hierarchy used to ease administrative effort in allocating names
- IP Addresses
  - Easily handled by routers
  - Fixed length
  - Hierarchy used to reduce routing table size

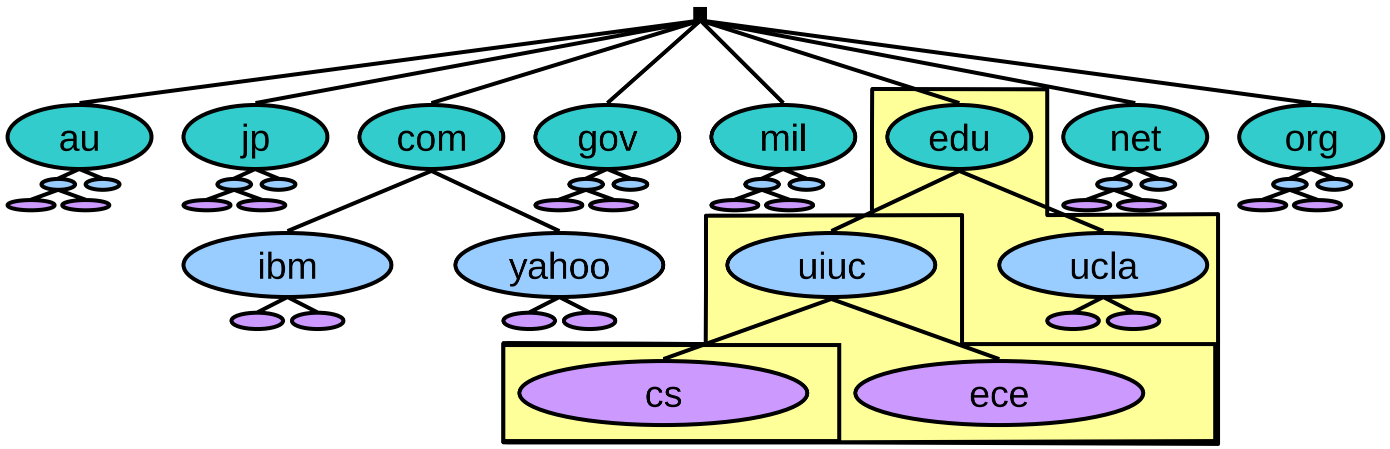
DNS – Name Space

- Domain name hierarchy
  - Structure
    - Period separated identifiers
    - Host name first
    - Each subsequent component is a larger group



DNS – Name Space

- Implementation
  - Each identifier (after host name) denotes a zone
  - Translation for each zone supported by 2+ name servers



DNS - Bindings

- Name servers maintain
  - Collection of resource records (5-tuples)
    - (name, value, type, class, TTL)
- Type is how name/value should be interpreted
  - type=A:
    - name=full domain name; value=IP addr
    - Implements name-to-address mapping
  - type=NS:
    - name=zone name; value=zone name server's domain name
    - Value is nameserver that can resolve this particular zone
    - Root nameserver has an NS record for each TLD
  - type=CNAME:
  - type=MX:

DNS - Bindings

- Name servers maintain
  - Collection of resource records (5-tuples)
    - (name, value, type, class, TTL)
- Type is how name/value should be interpreted
  - type=A:
  - type=NS:
  - type=CNAME:
    - name=domain name alias; value=canonical domain name for host
    - Can create multiple aliases for a single physical host
    - E.g., ftp.cs.uiuc.edu, www.cs.uiuc.edu point to different ports on srv1.cs.uiuc.edu
  - type=MX:
    - name=zone name; value=maildrop host's full domain name
    - Value field gives the domain name for a host that is running a mail server for the specified domain

DNS - Bindings

- Resource Record
  - Class
    - Generally set to IN (= Internet)
    - Allows use of DNS for other purposes
    - Rarely used
  - TTL
    - How long resource is valid (in seconds)
    - Used for caching, eviction after TTL expiry

DNS - Bindings

- Example resource records at a root name server
  - `< arizona.edu, telcom.arizona.edu, NS, IN >`
  - `< telcom.arizona.edu, 128.196.128.233, A, IN >`
  - `< bellcore.com, thumper.bellcore.com, NS, IN >`
  - `< thumper.bellcore.com, 128.96.32.20, A, IN >`

DNS - Bindings

- Examples of resource records at Arizona's name server

```
< cs.arizona.edu, optima.cs.arizona.edu, NS, IN >
< optima.cs.arizona.edu, 192.12.69.5, A, IN >
< ece.arizona.edu, helios.ece.arizona.edu, NS, IN >
< helios.ece.arizona.edu, 128.196.28.166, A, IN >
< jupiter.physics.arizona.edu, 128.196.4.1, A, IN >
< saturn.physics.arizona.edu, 128.196.4.2, A, IN >
```

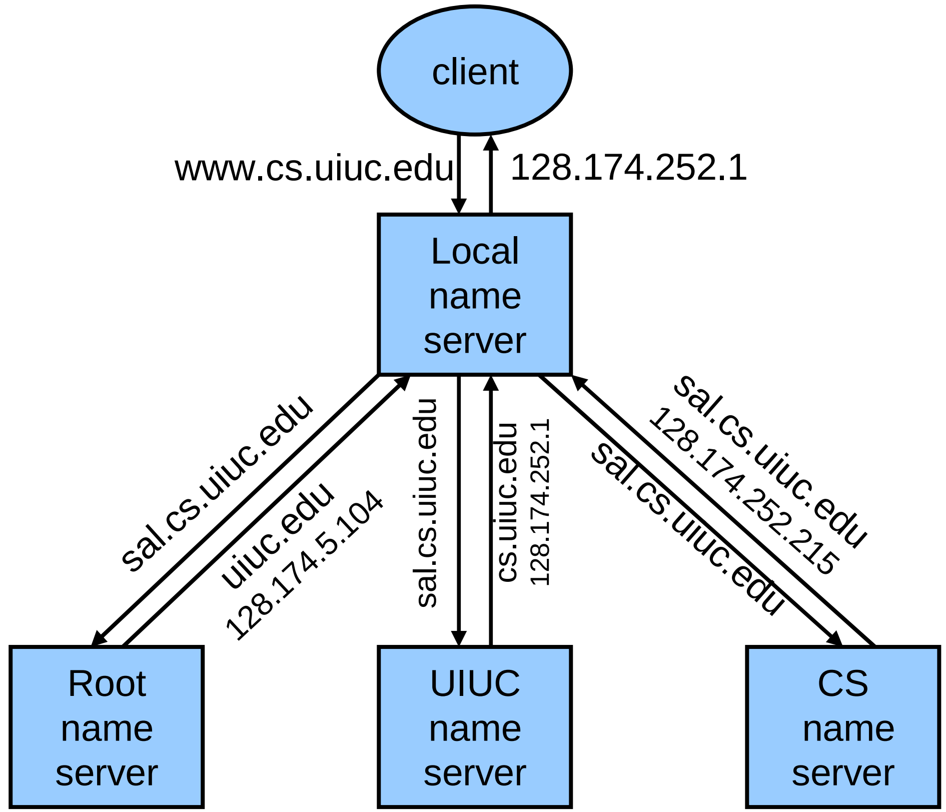
DNS - Bindings

- Examples of resource records at Arizona's CS name server

```
< cs.arizona.edu, optima.cs.arizona.edu, MX, IN >
< cheltenham.cs.arizona.edu, 192.12.69.60, A, IN >
< che.cs.arizona.edu,cheltenham.cs.arizona.edu,CNAME, IN >
< optima.cs.arizona.edu, 192.12.69.5, A, IN >
< opt.cs.arizona.edu, optima.cs.arizona.edu, CNAME, IN >
```

DNS – Name Server

- Name Resolution
  - Strategies
    - Iterative
    - Recursive
  - Local Server
    - Need to know root at only one place
    - Site-wide cache



DNS: Domain Name System

- Internet hosts
  - IP address (32 bit)
    - Used for addressing datagrams
  - Host name (e.g., www.yahoo.com)
    - Used by humans
- DNS: provides translation between host name and IP address
  - Distributed database implemented in hierarchy of many name servers
  - Distributed for scalability & reliability

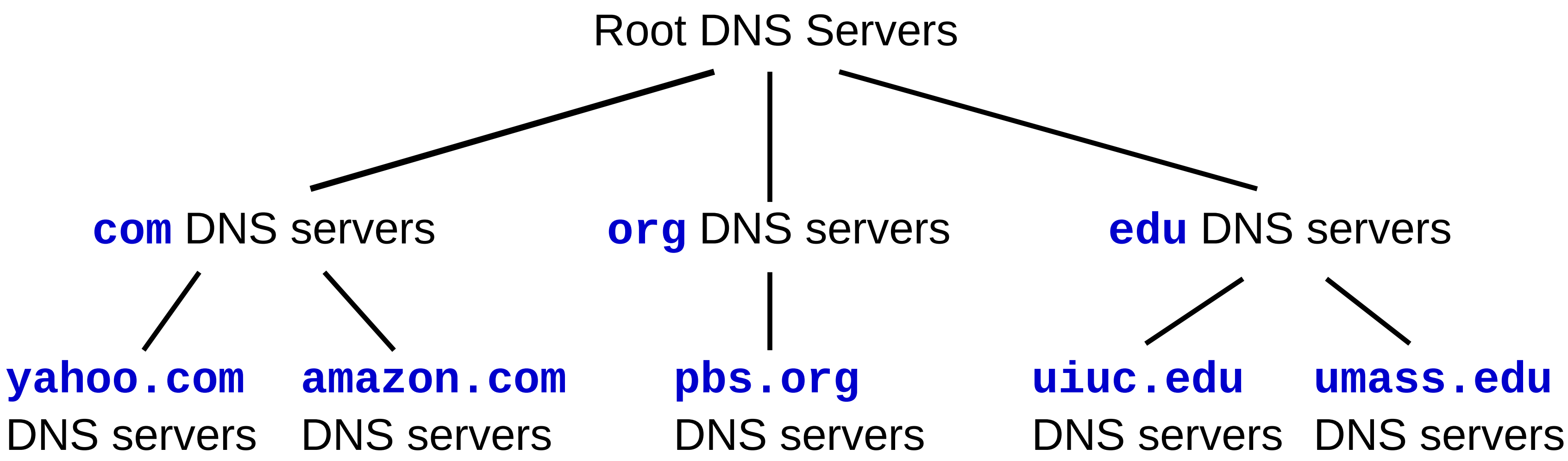
DNS

- DNS services
  - Hostname to IP address translation
  - Host aliasing
    - Canonical, alias names
  - Mail server aliasing
  - Load distribution
    - Replicated Web servers: set of IP addresses for one canonical name

- Why not centralize DNS?
  - Single point of failure
  - Traffic volume
  - Distant centralized database
  - Maintenance

- Doesn’t scale!

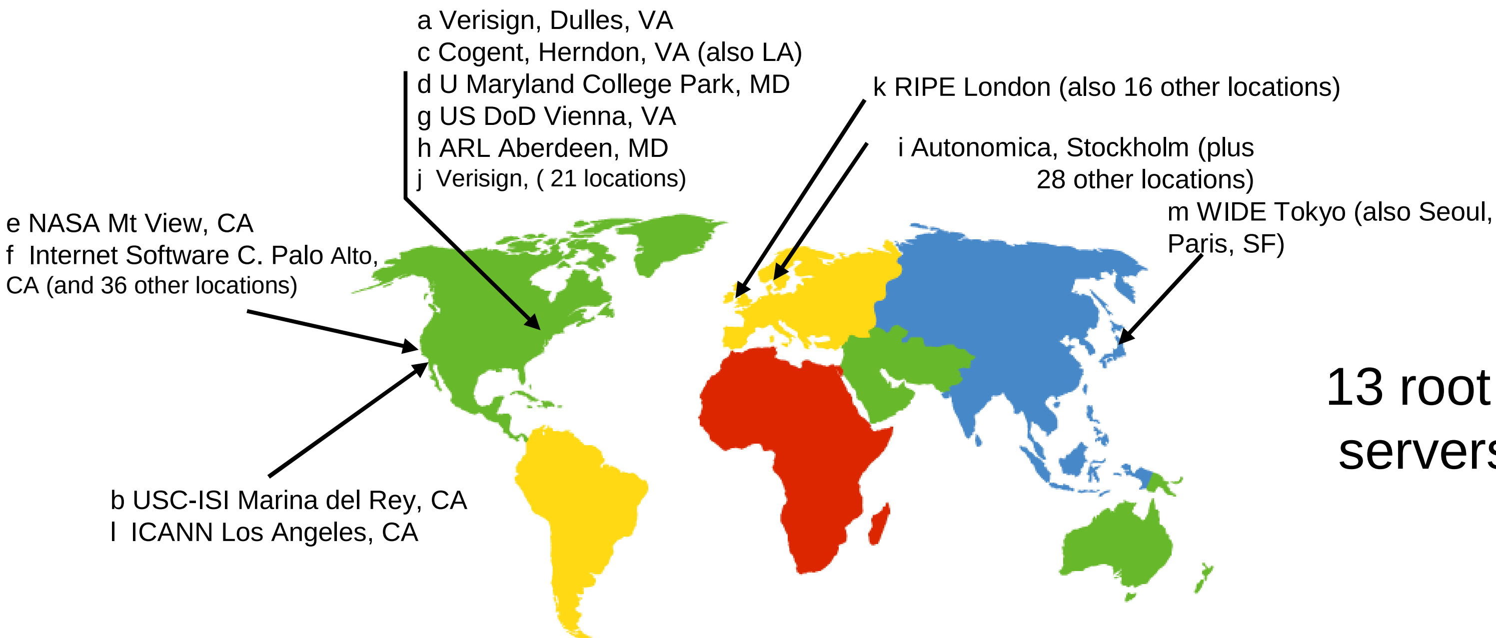
Distributed, Hierarchical Database



- Client wants IP for www.amazon.com
  - Client queries a root server to find **com** DNS server
  - Client queries **com** DNS server to get **amazon.com** DNS server
  - Client queries **amazon.com** DNS server to get IP address for **www.amazon.com**

DNS: Root Name Servers

- Contacted by local name server that can not resolve name, Contacts/redirects to authoritative name server if mapping not known
- Only 2% of queries reaching root are legitimate (incorrect caching is 75%, 7% were lookups for IPs as domain names, 12.5% were for unknown TLDs)



13 root name servers worldwide

TLD and Authoritative Servers

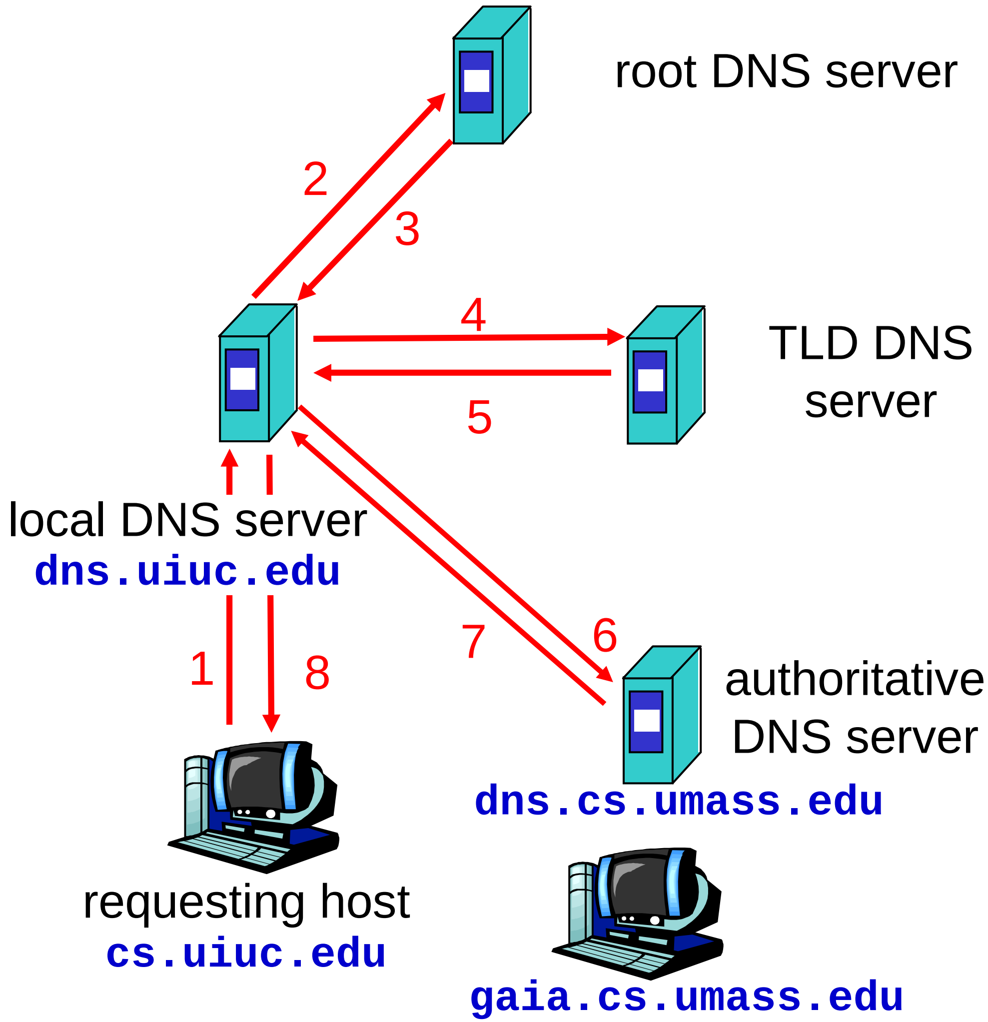
- Top-level domain (TLD) servers
  - Responsible for **com**, **org**, **net**, **edu**, etc, and all top-level country domains **uk**, **fr**, **ca**, **jp**.
    - Network Solutions maintains servers for **com** TLD
    - Educause for **edu** TLD
- Authoritative DNS servers
  - Organization's DNS servers
  - Provide authoritative hostname to IP mappings for organization's servers (e.g., Web, mail).
  - Can be maintained by organization or service provider

Local Name Server

- When host makes DNS query, query is sent to its local DNS server
  - Acts as proxy, forwards query into hierarchy
  - Uses caching to reduce lookup latency for commonly searched hostnames

DNS name resolution example

- Host at cs.uiuc.edu wants IP address for gaia.cs.umass.edu
- Iterated query
  - Contacted server replies with name of server to contact
  - “I don’t know this name, but ask this server”
  - Alternative: recursive queries (typically used only for local DNS, as requires state to be stored)



DNS: Caching

- Once (any) name server learns mapping, it caches mapping
  - Cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited

Domain Name Service (DNS)

- Large scale dynamic, distributed application
  - Replaced Network Information Center (NIC)
- RFC 1034 and 1035
- Outline
  - Comparison of domain names and addresses
  - Domain name hierarchy
  - Implementation of hierarchy
  - Name resolution

Network Address Translation (NAT)

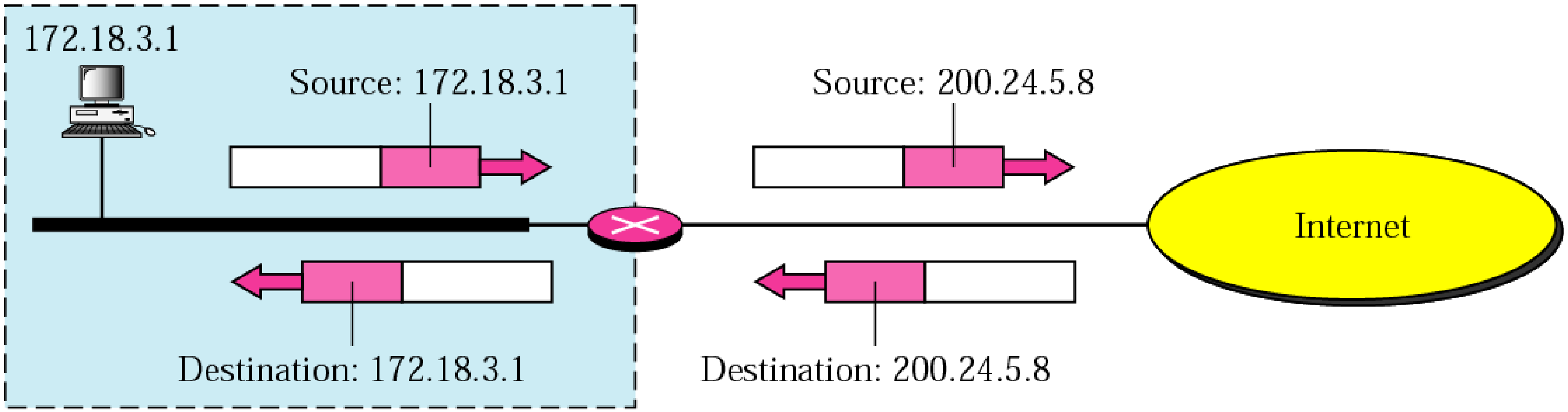
- Deals with problem of limited address space
  - Use locally unique addresses inside an organization
  - For communication outside the organization, use a NAT box
    - Translate from locally unique address to globally unique NAT address
    - Saves addresses if only a few hosts are ever communicating outside the organization
- Problem
  - Breaks IP service model
  - Lots of debate over whether this is a “permanent” or “temporary” solution

NAT: Network Address Translation

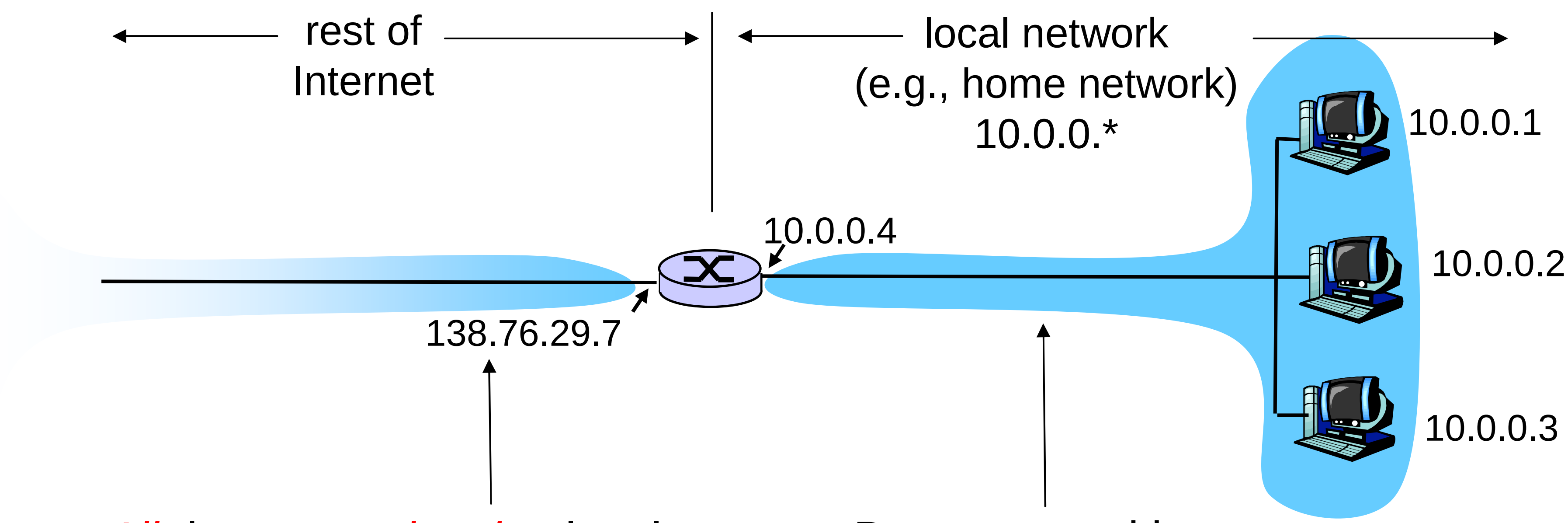
- Change IP Headers
  - IP addresses (and possibly port numbers) of IP datagrams are replaced at the boundary of a private network
  - Enables hosts on private networks to communicate with hosts on the Internet
  - Run on routers that connect private networks to the public Internet

NAT: Network Address Translation

- Outgoing packet
  - Source IP address (private IP) is replaced by one of the global IP addresses maintained by the NAT router
- Incoming packet
  - Destination IP address (global IP of the NAT router) is replaced by the appropriate private IP address



NAT: Network Address Translation



All datagrams *leaving* local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Datagrams with source or destination in this network have 10.0.0.* address for source, destination (as usual)

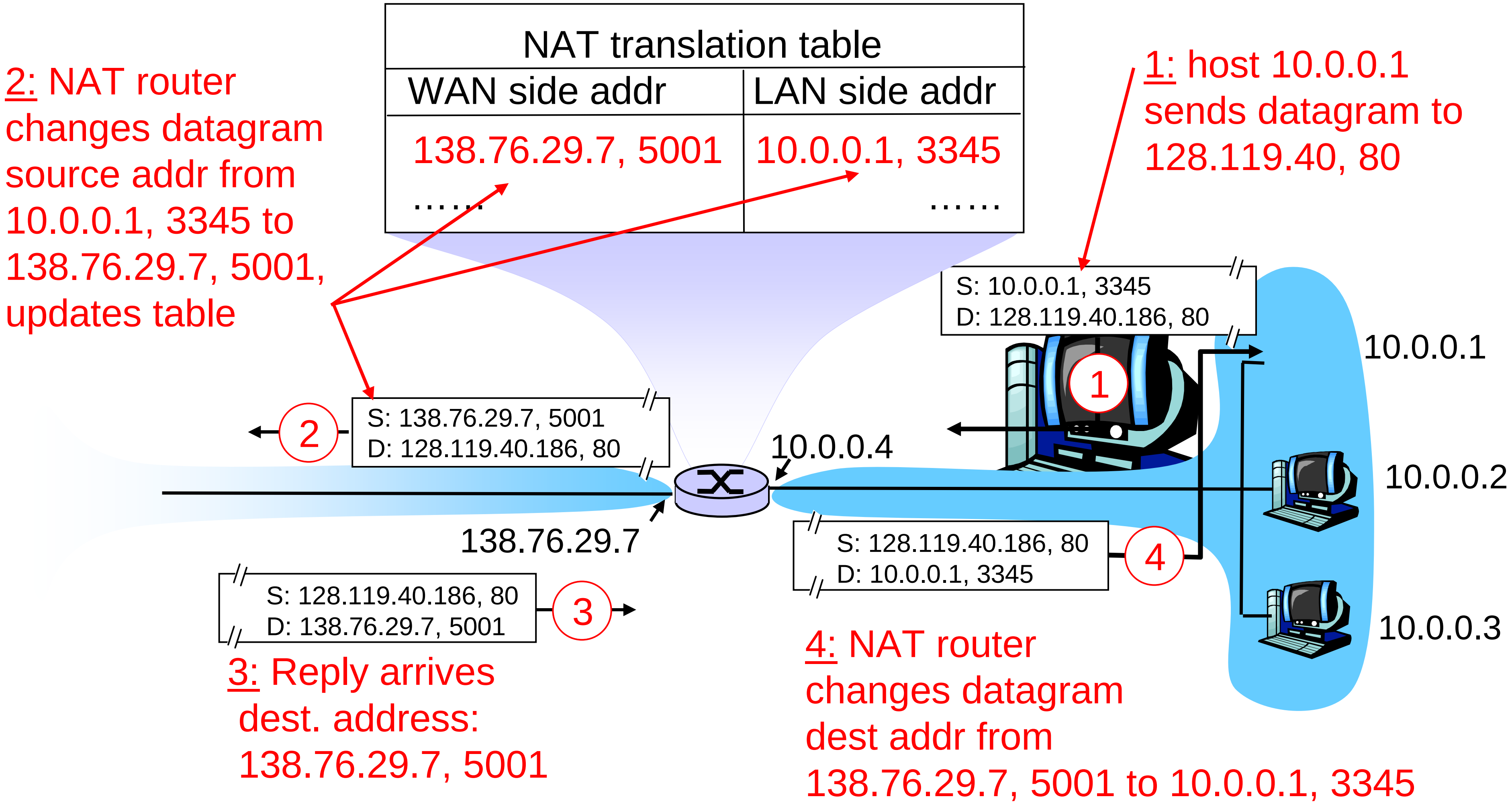
NAT: Network Address Translation

- Motivation: local network uses just one IP address as far as outside word is concerned
  - No need to be allocated range of addresses from ISP
    - Just one IP address is used for all devices
  - Can change addresses of devices in local network without notifying outside world
  - Can change ISP without changing addresses of devices in local network
  - Devices inside local net not explicitly addressable, visible by outside world (a security plus).

NAT: Network Address Translation

- Outgoing datagrams
  - replace (source IP address, port #) of every outgoing datagram with (NAT IP address, new port #)
    - Remote clients/servers respond using (NAT IP address, new port #) as destination addr
- Cache (in NAT translation table)
  - Every (source IP address, port #) to (NAT IP address, new port #) translation pair
- Incoming datagrams
  - Replace (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

NAT: Network Address Translation



NAT: Network Address Translation

- Address Pooling
  - Corporate network has many hosts
  - Only a small number of public IP addresses
- NAT solution
  - Manage corporate network with a private address space
  - NAT, at boundary between corporate network and public Internet, manages a pool of public IP addresses
  - When a host from corporate network sends an IP datagram to a host in public Internet, NAT picks a public IP address from the address pool, and binds this address to the private address of the host

NAT: Network Address Translation

- IP masquerading
  - Single public IP address is mapped to multiple hosts in a private network
- NAT solution
  - Assign private addresses to the hosts of the corporate network
  - NAT device modifies the port numbers for outgoing traffic
  - Modifying the IP header by changing the IP address requires that NAT boxes recalculate the IP header checksum
  - Modifying port number requires that NAT boxes recalculate TCP checksum

NAT: Network Address Translation

- Load balancing
  - Balance the load on a set of identical servers, which are accessible from a single IP address
- NAT solution
  - Servers are assigned private addresses
  - NAT acts as a proxy for requests to the server from the public network
  - NAT changes the destination IP address of arriving packets to one of the private addresses for a server
  - Balances load on the servers by assigning addresses in a round-robin fashion

NAT: Network Address Translation

- 16-bit port-number field
  - 60,000 simultaneous connections with a single LAN-side address!
- End-to-end connectivity
  - NAT destroys universal end-to-end reachability of hosts on the Internet
  - A host in the public Internet often cannot initiate communication to a host in a private network
  - The problem is worse, when two hosts that are in different private networks need to communicate with each other

NAT: Network Address Translation

- IP address in application data
  - Applications often carry IP addresses in the payload of the application data
  - No longer work across a private-public network boundary
  - Hack: Some NAT devices inspect the payload of widely used application layer protocols and, if an IP address is detected in the application-layer header or the application payload, translate the address according to the address translation table

Roadmap

- Addresses
  - Assignment: CIDR, DHCP
  - Translation: ARP, NAT
- Names
  - Translation to addresses

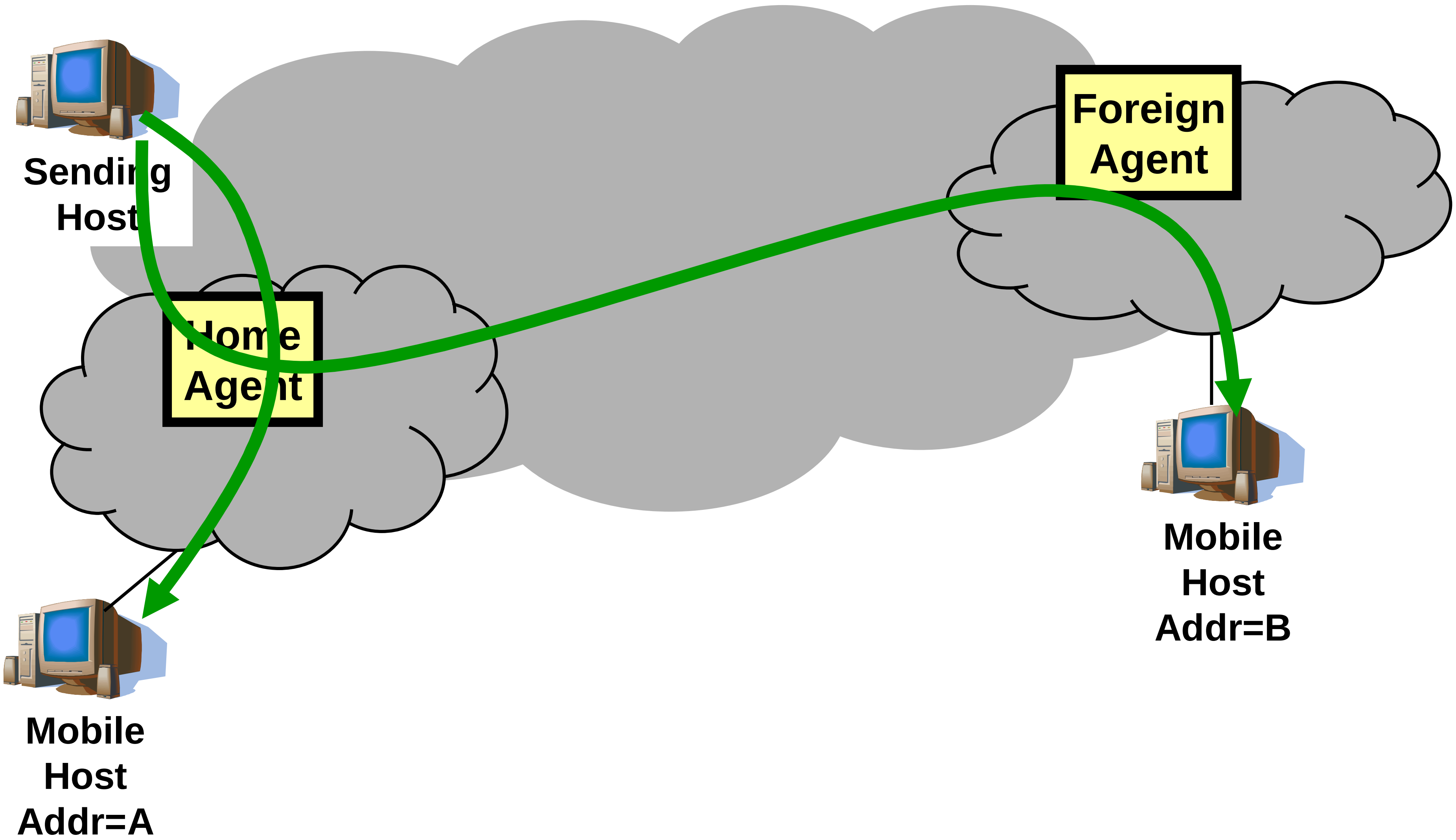
Routing For Mobile Hosts

- Scenarios
  - Mobile hosts, fixed infrastructure
    - Cellular networks
    - 802.11 enterprise networks
  - Mobile hosts, dynamic infrastructure
    - Ad hoc networks
- Problem
  - How can mobility be supported in view of the fact that a portion of an IP address is a network address?
  - Solution: Mobile IP

IP Address Problem

- Internet hosts/interfaces are identified by IP address
  - Domain name service translates host name to IP address
  - IP address identifies host/interface and locates its network
  - Mixes naming and location
- Moving to another network requires different network address
  - But this would change the host's identity
  - How can we still reach that host?

Routing for Mobile Hosts with Mobile IP



Why Mobile IP?

- Goal
  - IP-based protocol which allows network connectivity across host movement
- Features
  - Doesn't require global changes to deployed router software, etc.
  - Compatible with large installed base of IPv4 networks/hosts
  - Confines changes to mobile hosts and a few support hosts which enable mobility

Basic Mobile IP

- Features
  - Transparent routing of packets to a mobile host
  - No modification of existing routers or non-mobility supporting hosts
- Problem
  - Indirect routing places unnecessary burden on the internet and significant increases latency

Components

- Mobile Host (MH):
  - Assigned a unique home address within its home network
- Corresponding Hosts (CH):
  - Other hosts communicating with the MH
  - Always use MH’s home address

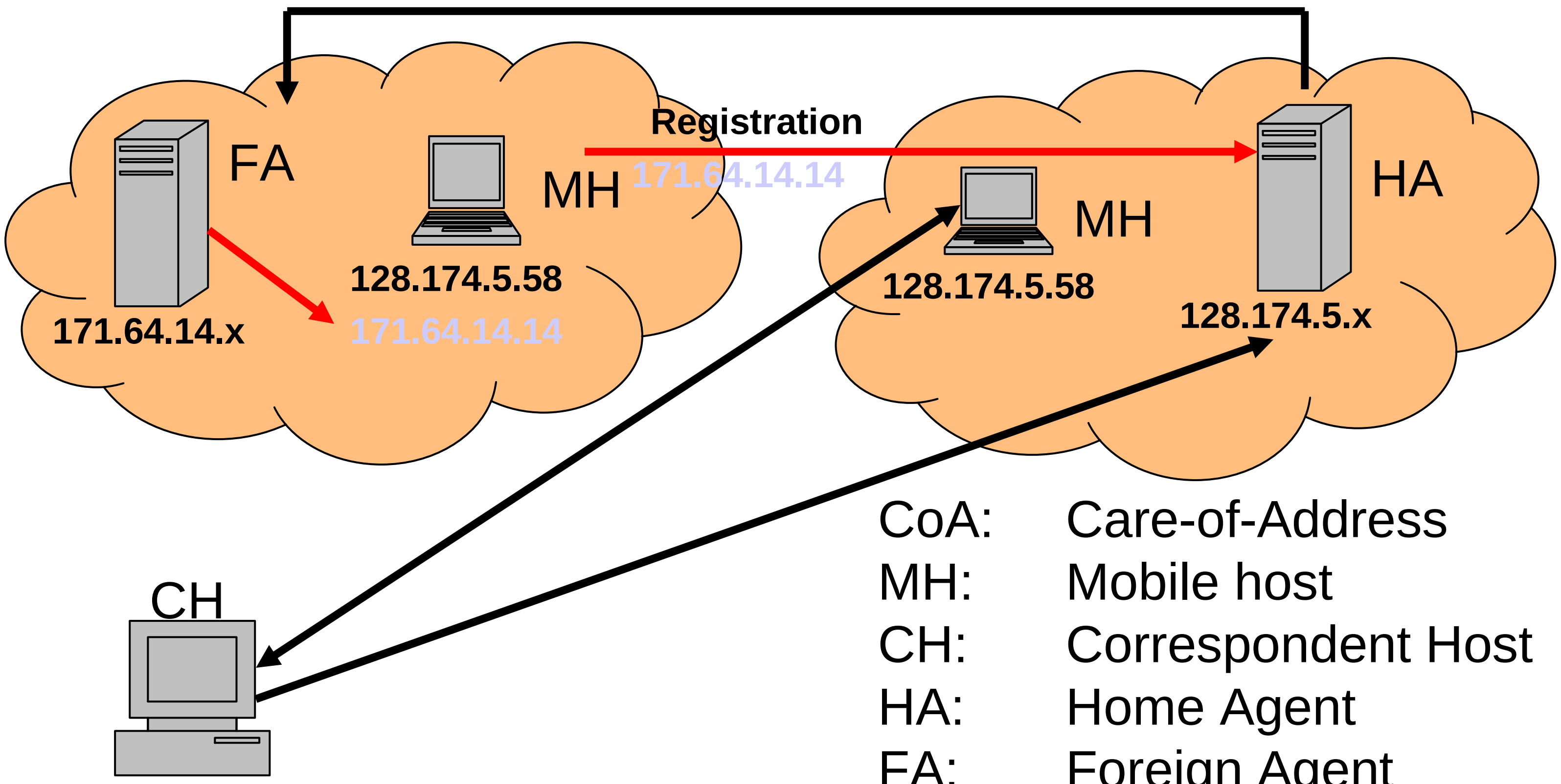
Routing for Mobile Hosts

- Home Agent (HA):
  - An agent on the MH's home network
  - Maintains registry of MH's current location
  - Mobility binding is the connection between the MH's home address and *care-of-address* (MH's remote address)
  - Each time the MH establishes a new care-of-address, it must register with its HA
- Foreign Agent (FA):
  - An agent on the MH's local network
  - Maintains a mapping from the MH's home address to its care-of-address

Issues

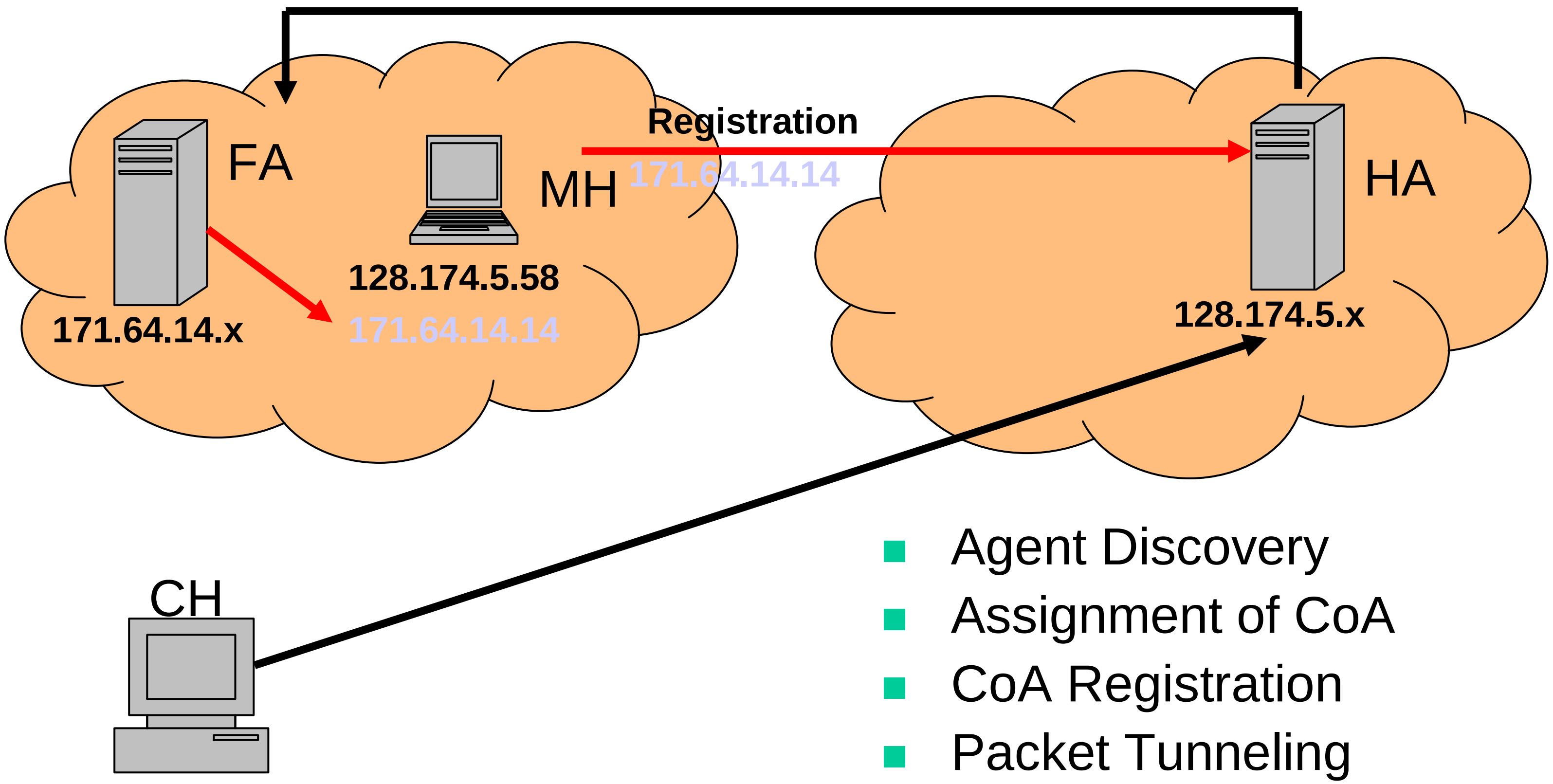
- Scenario
  - CH sends packet to home network
- Challenges
  - How does the MH get a local IP address?
  - How can a mobile host tell where it is?
  - How does the HA intercept a packet that is destined for the MH?
  - How does the HA then deliver the packet to the FA?
  - How does the FA deliver the packet to the MH?

Basic Mobile IP



CoA:	Care-of-Address
MH:	Mobile host
CH:	Correspondent Host
HA:	Home Agent
FA:	Foreign Agent

Basic Mobile IP



- Agent Discovery
- Assignment of CoA
- CoA Registration
- Packet Tunneling

Addressing

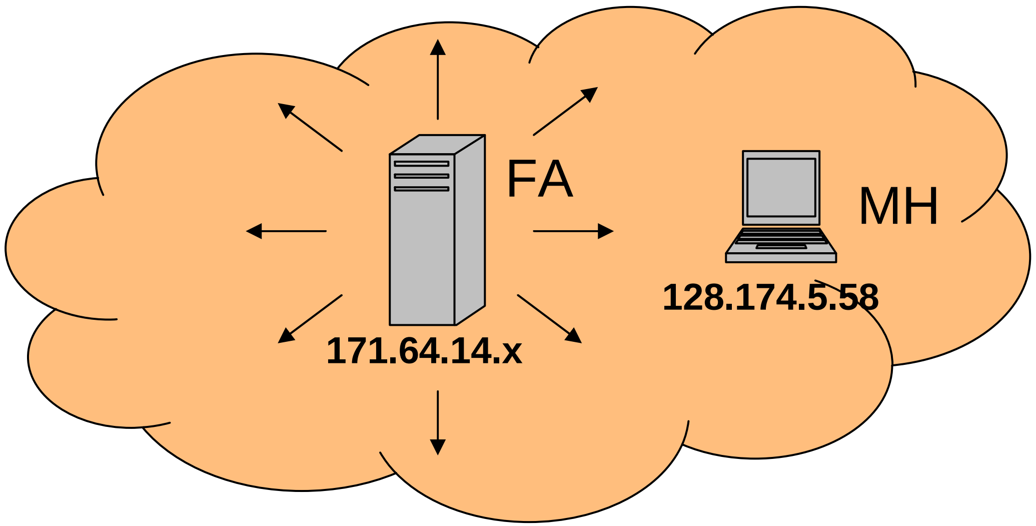
- How does the mobile host get a remote IP address?
  - Listen for router advertisements
  - Use DHCP
  - Manual assignment
- Assigning *care-of-address*
  - MH discovers *foreign agent* (FA) using an agent discovery protocol
  - MH registers with FA and FA's address becomes MH's *care-of-address*
  - MH obtains a temporary IP address from FA or via DHCP-like procedures

Location

- How can a mobile host tell where it is?
  - Am I at home?
  - Am I visiting a foreign network?
  - Have I moved?
  - Again, listen for router advertisements
  - Put network interface into promiscuous mode and watch traffic

Agent Discovery

- How can a mobile host tell where it is?
  - Extension of ICMP protocol
  - Allows MH to detect when it has moved from one network to another, or to home
  - FA Periodically broadcasts agent advertisement message



Agent Discovery

- MH determines a suitable FA (or its HA) with which to register
- If MA has not received a broadcast for a period of time, it can send an agent solicitation message

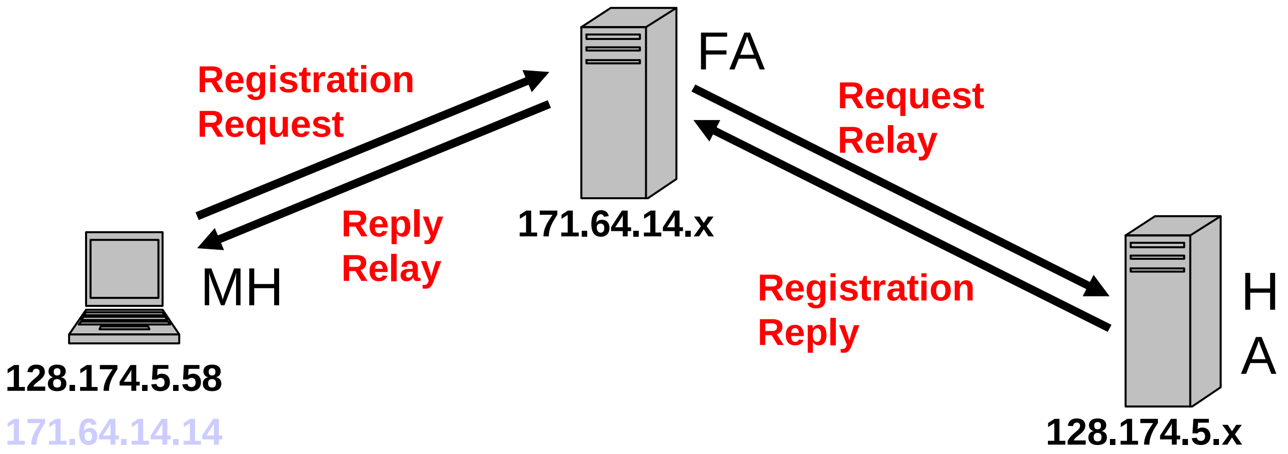
Packet Delivery

- How does the HA intercept a packet that is destined for the MH?
- While MH in foreign location
  - HA intercepts all packets for MH
    - Using proxy ARP
  - HA tunnels all packets to FA
    - IPIP - "IP within IP"
    - Upon receipt of an IP datagram
      - Packet is encapsulated in an IP packet of type IPPROTO_IPIP and sent to FA
      - FA strips IPIP header and sends packet to MH using local IP address
  - FA strips packet and forwards to MH

Registration

- MA must register with FA and tell HA its new care-of-address
  - MH sends registration request message to FA
  - FA forwards request to HA
  - HA returns registration reply message to FA
  - FA forwards reply to MH
- Registration may have a set lifetime

Care-of-Address Registration

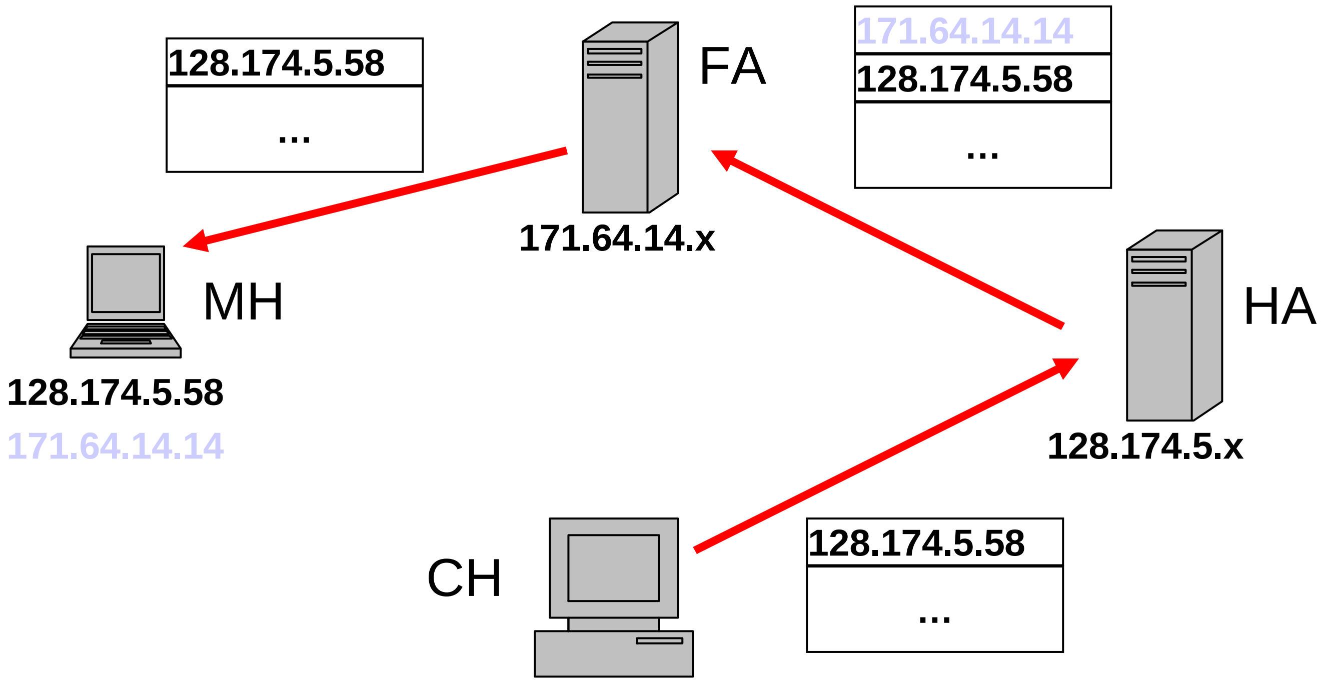


128.174.5.58
171.64.14.14
Timeout

Network Layer

- IPIP - “IP within IP”
  - Tunnel IP datagrams from one subnet to another
  - Upon receipt of an IP datagram
    - Packet is encapsulated in an IP packet of type IPPROTO_IPIP and sent to remote MSS
    - Remote MSS strips IPIP header and sends packet to MH using “real” IP address

Tunneling



128.174.5.58
...
FA
171.64.14.14
128.174.5.58
...
171.64.14.x
MH
HA
128.174.5.58
171.64.14.14
128.174.5.x
CH
128.174.5.58
...

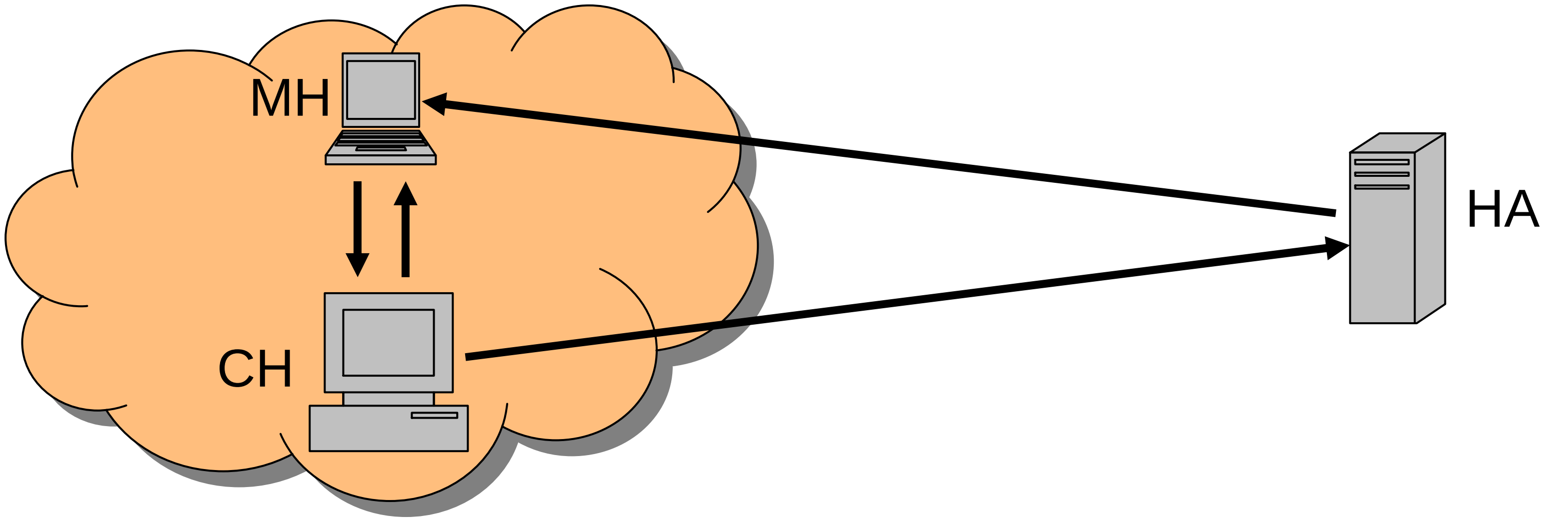
Tunneling Using IP-in-IP Encapsulation

<table>
<tr><th>0</th><th>4</th><th>8</th><th>16</th><th>19 … 31</th></tr>
<tr><td>Vers</td><td>IHL</td><td>TOS</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">IP Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>IP in IP</td><td colspan="2">IP Header Checksum</td></tr>
<tr><td colspan="5">Tunnel Source IP Address</td></tr>
<tr><td colspan="5">Care-of-Address</td></tr>
<tr><td>Vers</td><td>IHL</td><td>TOS</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">IP Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Orig. Protocol</td><td colspan="2">IP Header Checksum</td></tr>
<tr><td colspan="5">Original Source IP Address</td></tr>
<tr><td colspan="5">IP Address of Mobile Host</td></tr>
<tr><td colspan="5">TCP/UDP/etc</td></tr>
</table>

Tunneling Using Minimal Tunneling Protocol

<table>
<tr><th>0</th><th>4</th><th>8</th><th></th><th>16</th><th>19</th><th>31</th></tr>
<tr><td>Vers</td><td>IHL</td><td colspan="2">TOS</td><td colspan="3">Total Length</td></tr>
<tr><td colspan="4">IP Identification</td><td>Flags</td><td colspan="2">Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td colspan="2">Min Encap</td><td colspan="3">IP Header Checksum</td></tr>
<tr><td colspan="7">Tunnel Source IP Address</td></tr>
<tr><td colspan="7">Care-of-Address</td></tr>
<tr><td colspan="2">Orig. Protocol</td><td>S</td><td></td><td colspan="3">Tunnel Header Checksum</td></tr>
<tr><td colspan="7">IP Address of Mobile Host</td></tr>
<tr><td colspan="7">Original Source IP Address (only present if S is set)</td></tr>
<tr><td colspan="7">TCP/UDP/etc</td></tr>
</table>

Triangle Routing Problem



- Routing through HA increases latency

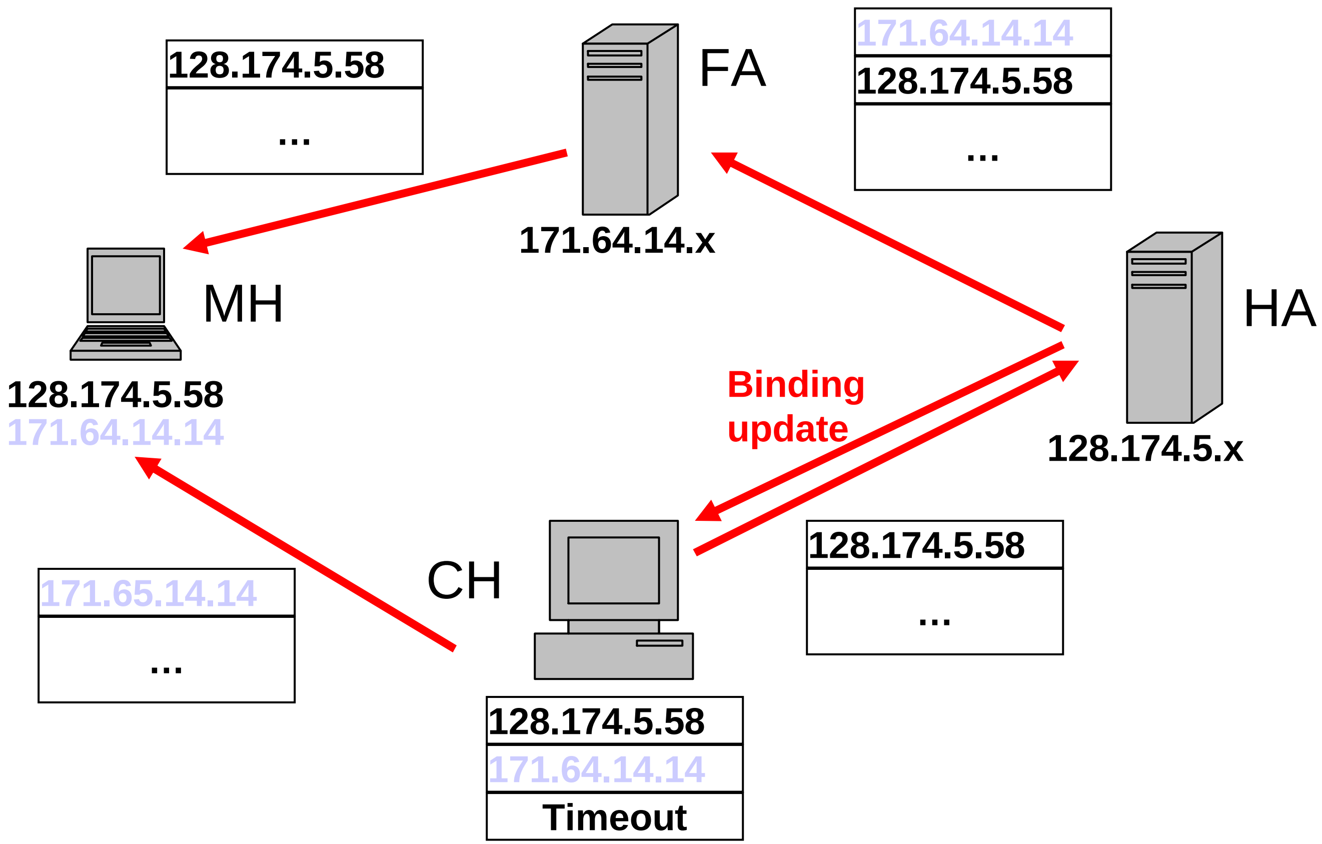
Route Optimization

- Basic Mobile IP routes all packets for a MH through its home network and HA
  - Limits performance
  - Potential bottleneck
  - Not scalable
- Solution
  - Cache MH location and care-of address

Location Caching

- Binding Cache
  - Maintains location information about MHs
  - Binding cache entry
    - Packet is tunneled directly to MH's care-of-address
  - No binding cache entry
    - Packet is sent to MH's HA
    - HA sends new entry
  - Can support networks with no Mobile IP by putting binding cache in router

Binding Cache



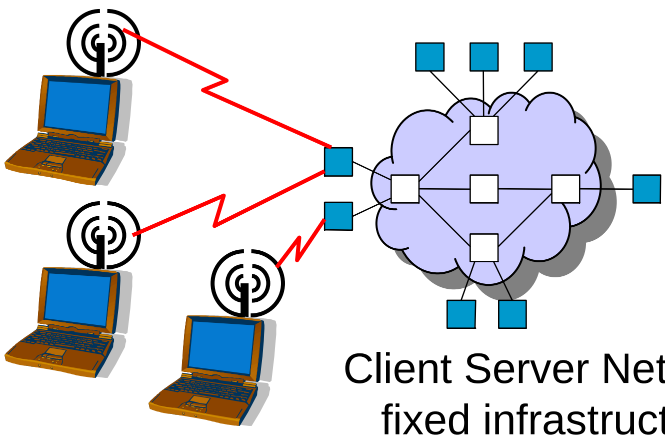
Ad Hoc Networks

Based on a Tutorial by
Nitin Vaidya

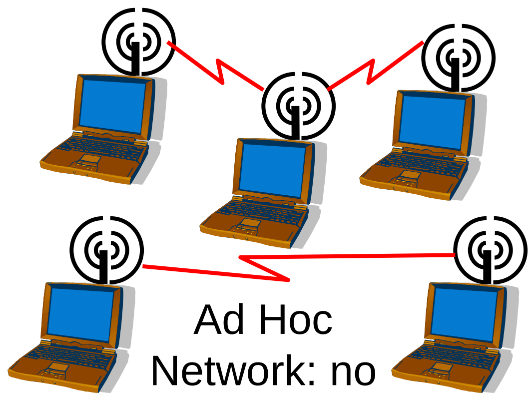
© Robin Kravets, M. Caesar, UIUC - Spring 2007

Routing in Ad Hoc Networks

- Ad hoc network
  - A collection of mobile nodes with wireless interfaces that form a temporary network without the aid of any established or centralized administration



Client Server Network: fixed infrastructure



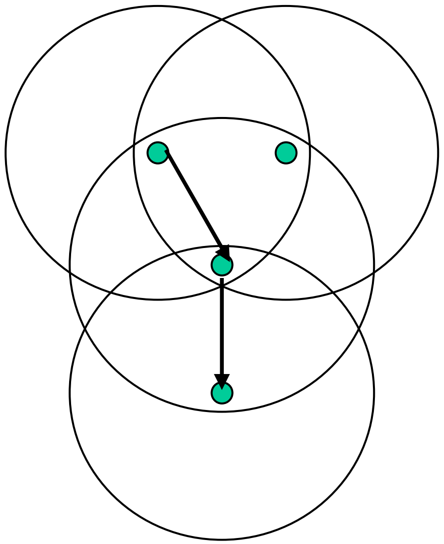
Ad Hoc Network: no servers or access points

Mobile Ad Hoc Networks

- Formed by wireless hosts which may be mobile
- Without (necessarily) using a pre-existing infrastructure
- Routes between nodes may potentially contain multiple hops

Multi-Hop Wireless

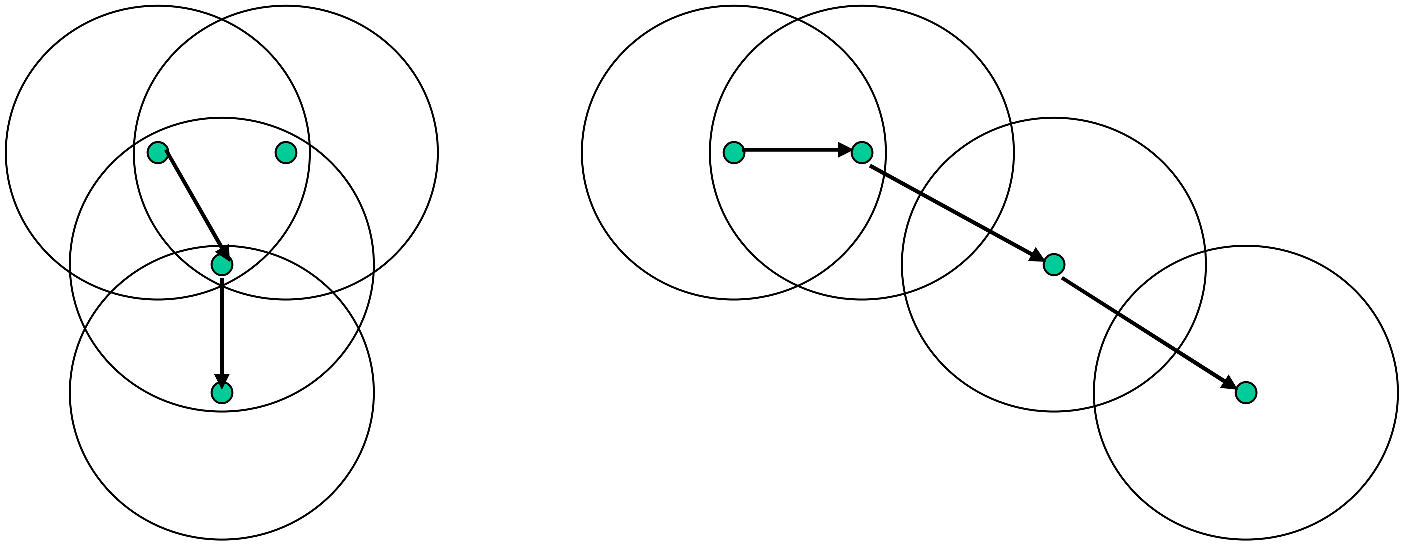
- May need to traverse multiple links to reach a destination



© Robin Kravets, M. Caesar, UIUC - Spring 2007

Multi-Hop Wireless

- Mobility
  - Mobile Ad Hoc Networks (MANET)
  - Mobility causes route changes



Why Ad Hoc Networks ?

- Ease of deployment
- Speed of deployment
- Decreased dependence on infrastructure

Challenges

- Limited wireless transmission range
- Broadcast nature of the wireless medium
  - Hidden terminal problem
- Packet losses due to transmission errors
- Mobility-induced route changes
- Mobility-induced packet losses
- Battery constraints
- Potentially frequent network partitions
- Ease of snooping on wireless transmissions

© Robin Kravets, M. Caesar, UIUC - Spring 2007

Why is Routing in MANET different ?

- Host mobility
  - Link failure/repair due to mobility may have different characteristics than those due to other causes
- Rate of link failure/repair may be high when nodes move fast
- New performance criteria may be used
  - Route stability despite mobility
  - Energy consumption

Routing in Ad Hoc Networks

- Periodic Protocols:
  - Driven by timer based mechanisms
  - Distance-Vector and Link-State protocols send periodic routing advertisements
  - Link status detection is beacon-based
- Concerns:
  - Periodic updates waste bandwidth and power (especially if nothing changes)
  - Topology changes may be too dynamic to be captured by periodic updates
  - Routes may not work (some links may be unidirectional)
  - Shortest path may not be best path (signal strength, energy consumption)

© Robin Kravets, M. Caesar, UIUC - Spring 2007

Routing Protocols

- Proactive protocols
  - Determine routes independent of traffic pattern
  - Traditional link-state and distance-vector routing protocols are proactive
- Reactive (on-demand) protocols
  - Maintain routes only if needed

© Robin Kravets, M. Caesar, UIUC - Spring 2007

Routing in Ad Hoc Networks

- On-demand Protocols:
  - Actions driven by data packets requiring delivery
  - Obtain a route only when needed
  - Link status detection performed only when forwarding data
  - Allow new metrics
  - Ex:
    - Dynamic Source Routing Protocol (DSR)
    - Ad Hoc On-Demand Distance Vector Protocol (AODV)

Routing in Ad Hoc Networks

- On-demand Protocols
  - Path/Route discovery
    - used to set up forward and reverse paths
  - Route table management
    - Route tables are soft-state
- Performance Concerns
  - Latency to set up route
  - Overhead for route maintenance

© Robin Kravets, M. Caesar, UIUC - Spring 2007

Trade-Off

- Latency of route discovery
  - Proactive protocols
    - May have lower latency since routes are maintained at all times
  - Reactive protocols
    - May have higher latency because a route from X to Y will be found only when X attempts to send to Y

Trade-Off

- Overhead of route discovery/maintenance
  - Reactive protocols
    - May have lower overhead since routes are determined only if needed
  - Proactive protocols
    - Can (but not necessarily) result in higher overhead due to continuous route updating
- Which approach achieves a better trade-off depends on the traffic and mobility patterns