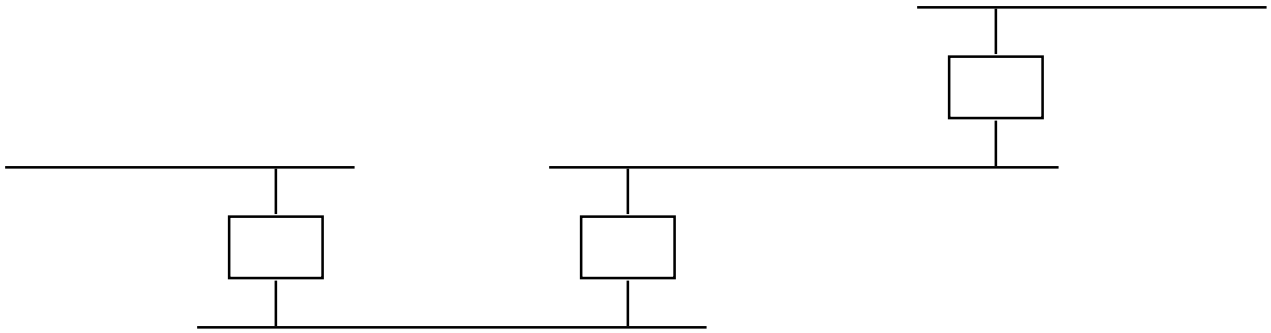
Copyright 1996. All rights reserved. Permission to reproduce this document for not for profit educational purposes is hereby granted. This document may not be reproduced for commercial purposes without the express written consent of the author.

Bridges and Extended LANs

Overview

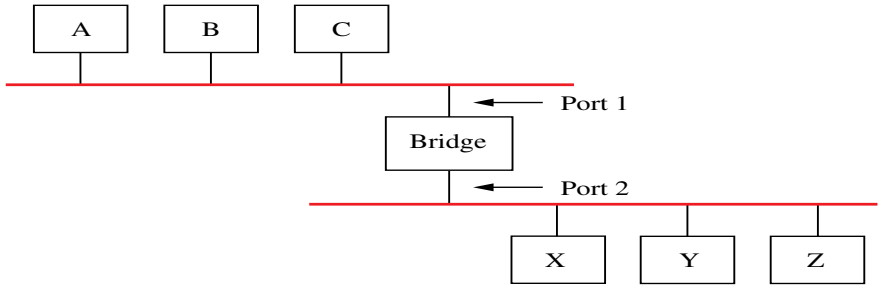
- LANs have physical limitations (e.g., 1500m Ethernet)
- Connect two or more LANs with a *bridge*
  - accept and forward strategy
  - level 2 connection (does not add packet header)



- Collection of LANs connected by bridges called an *exteneded LAN*

Learning Bridges

- Do not forward when unnecessary



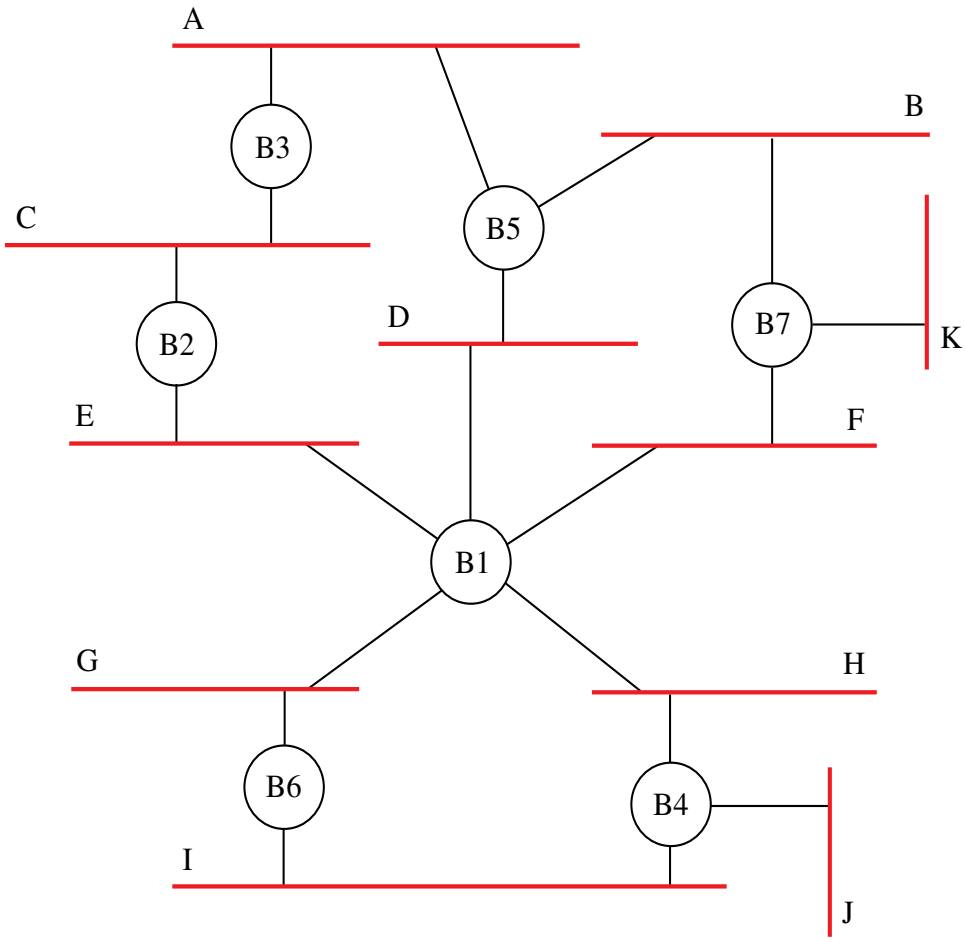
- Maintain forwarding table

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

- Learn table entries based on source address
- Table is an optimization; need not be complete
- Always forward broadcast frames

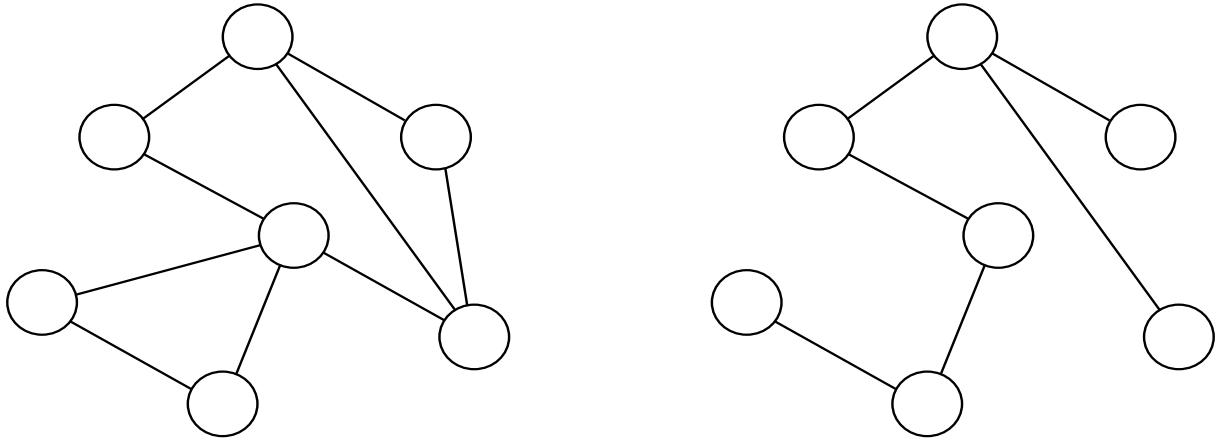
Spanning Tree Algorithm

Extended LANs sometimes have loops



Bridges run a distributed spanning tree algorithm

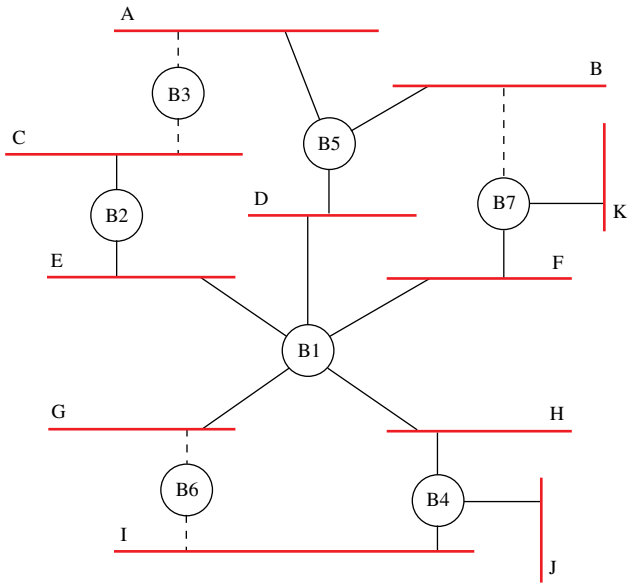
- Select which bridges actively forward frames
- Developed by Radia Perlman at DEC
- Now IEEE 802.1 specification



(a) (b)

Algorithm Overview

- Each bridge has a unique id (e.g., B1, B2, B3)
- Select bridge with smallest id as root
- Select bridge on each LAN that is closest to the root as that LAN's designated bridge (use id to break ties)
- Each bridge forwards frames over each LAN for which it is the designated bridge



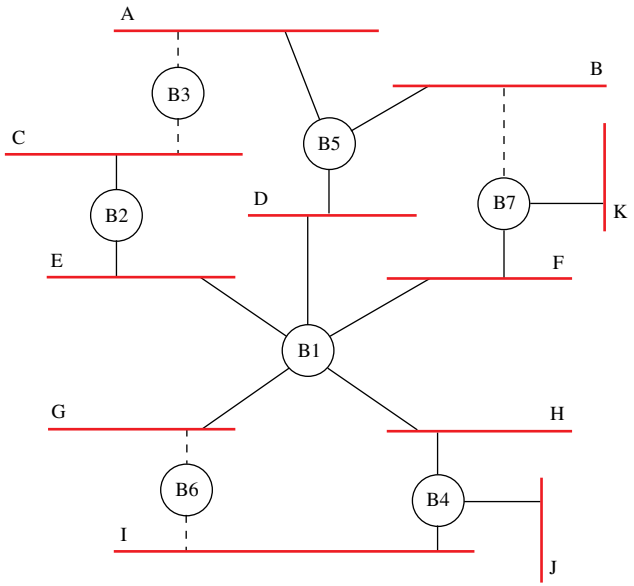
Algorithm Detail

- Bridges exchange configuration messages
  - id for bridge sending the message
  - id for what the sending bridge believes to be root bridge
  - distance (hops) from sending bridge to root bridge
- Each bridge records current best configuration message for each port
- Initially, each bridge believes it is the root
- When learn not root, stop generating configuration message
  - in steady state, only root generates configuration messages

- When learn not designated bridge, stop forwarding configuration messages
  - in steady state, only designated bridges forward configuration messages
- Root bridge continues to send configuration messages periodically
- If any given bridge does not receive configuration message after a period of time, starts generating configuration messages claiming to be to be the root

Broadcast and Multicast

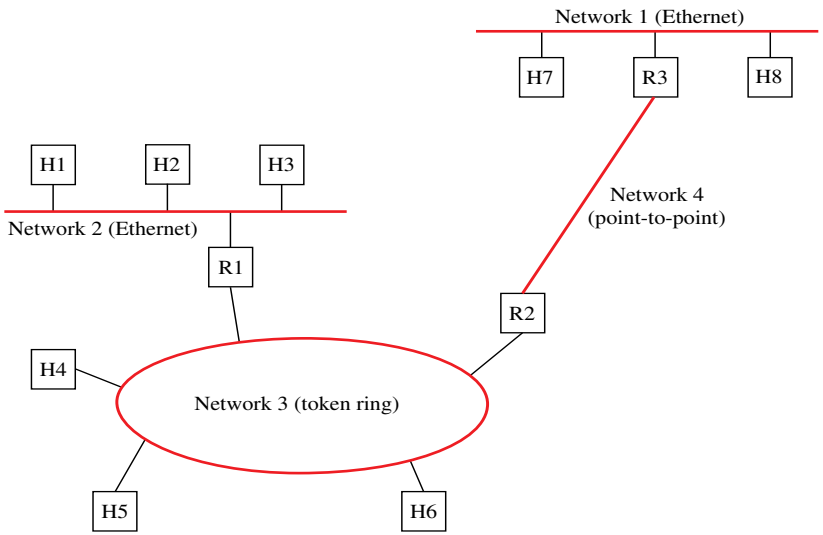
- Forward all broadcast/multicast frames (current practice)
- Learn when no group members downstream
- Accomplished by having each member of group G send a frame to bridge multicast address with G in source field

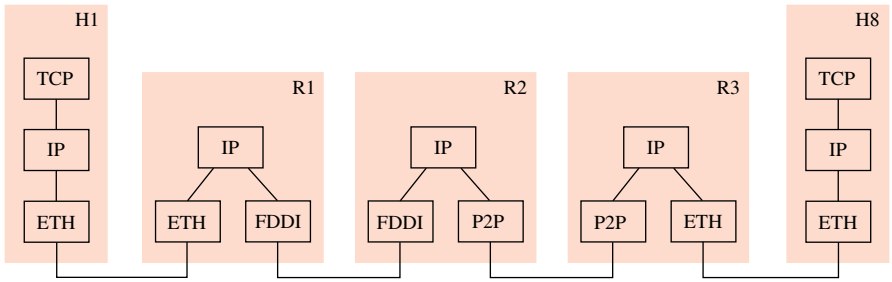


Limitations of Bridges

- Do not scale
  - spanning tree algorithm does not scale
  - broadcast does not scale
- Do not accommodate heterogeneity
- Caution: beware of transparency

Internetworking





Service Model

- Packet Delivery Model
- Global Addressing Scheme

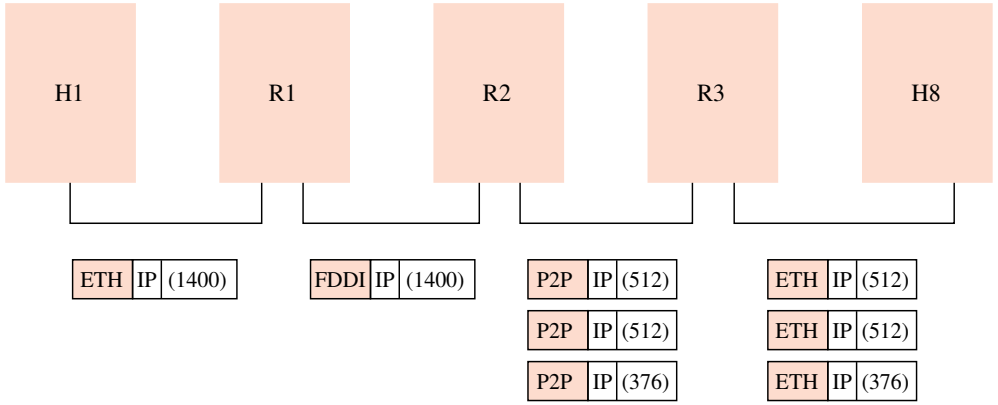
Packet Delivery Model

- Connectionless (datagram-based)
- Best-effort delivery (unreliable service)
  - packets are lost
  - packets are delivered out of order
  - duplicate copies of a packet are delivered
  - packets can be delayed for a long time

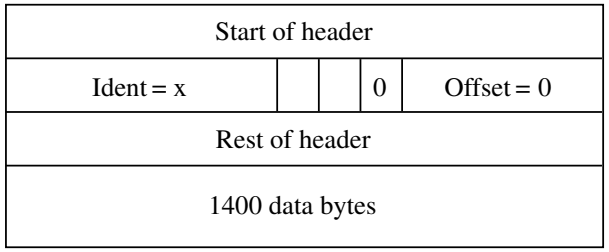
- Datagram format
  - Version (4): currently 4
  - Hlen (4): number of 32-bit words in header
  - TOS (8): type of service (not widely used)
  - Length (16): number of bytes in this datagram
  - Ident (16): used by fragmentation
  - Flags/Offset (16): used by fragmentation
  - TTL (8): number of hops this datagram has travelled
  - Protocol (8): demux key (TCP=6, UDP=17)
  - Checksum (16): of the header only
  - DestAddr & SrcAddr (32)

Fragmentation and Reassembly

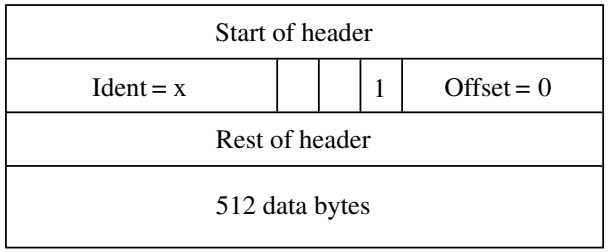
- Each network has some MTU
- Strategy
  - fragment when necessary (MTU < Datagram)
  - try to avoid fragmentation at source host
  - refragmentation is possible
  - fragments are self-contained datagrams
  - use CS-PDU (not cells) for ATM
  - delay reassembly until destination host
  - do not recover from lost fragments
- Example

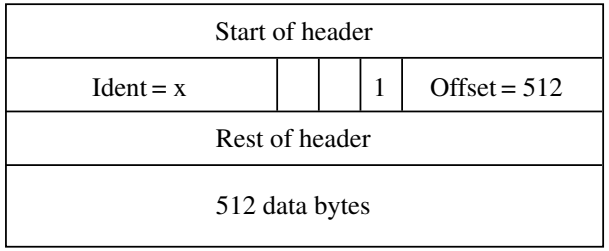


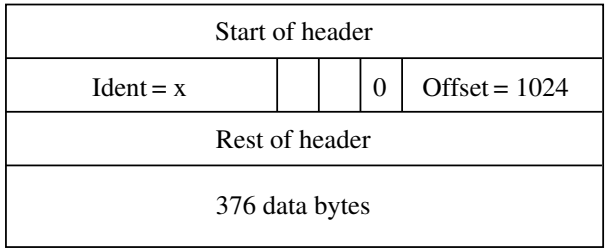
(a)


Start of header
Ident = x
0
Offset = 0
Rest of header
1400 data bytes

(b)

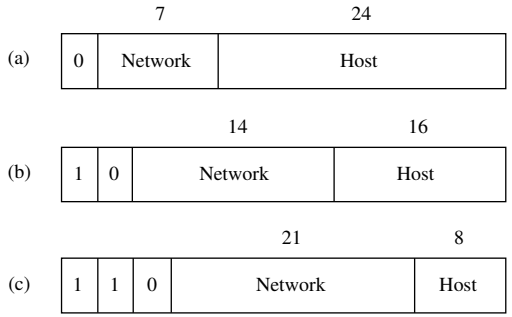

Start of header
Ident = x
1
Offset = 0
Rest of header
512 data bytes


Start of header
Ident = x
1
Offset = 512
Rest of header
512 data bytes


Start of header
Ident = x
0
Offset = 1024
Rest of header
376 data bytes

Global Addresses

- Properties
  - globally unique
  - hierarchical: network + host
- Format



- Dot notation
  - 10.3.2.4
  - 128.96.33.81
  - 192.12.69.77

Datagram Forwarding

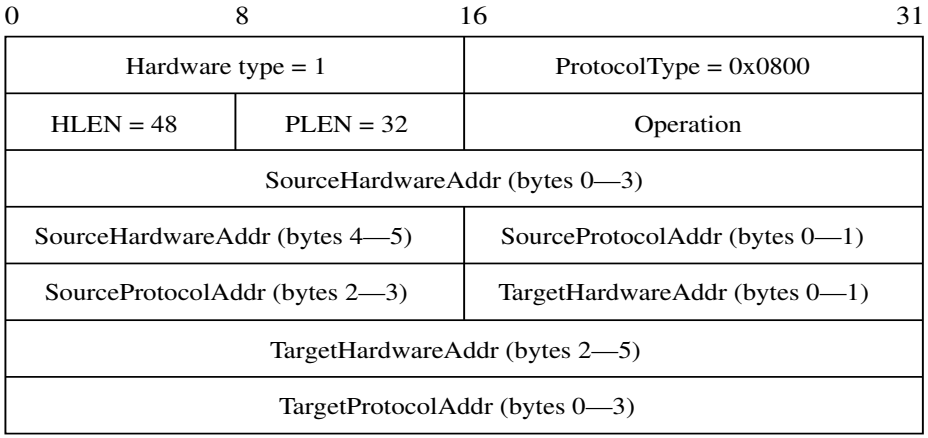
- Strategy
  - every datagram contains destination's address
  - if directly connected to destination network, then forward to host
  - if not directly connected to destination network, then forward to some router
  - forwarding table maps network number into next hop
  - each host has a default router
  - each router maintains a forwarding table
- Example (router R2)

Network Number	Next Hop
1	R3
2	R1
3	interface 1
4	interface 0

Address Translation

- Map IP addresses into physical addresses
  - destination host
  - next hop router
- Techniques
  - encode physical address in host part of IP address
  - table-based
- ARP
  - table of IP to physical address bindings
  - broadcast request if IP address not in table
  - target machine responds with its physical address
  - table entries are discarded if not refreshed

- Request format



  - HardwareType: type of physical network (e.g., Ethernet)
  - ProtocolType: type of higher layer protocol (e.g., IP)
  - HLEN & PLEN: length of physical and protocol addresses
  - Operation: request or response
  - Source/Target Physical/Protocol addresses

- Notes
  - table entries timeout in about 10 minutes
  - update table with source when you are the target
  - update table table if already have an entry
  - do not refresh table entries upon reference

Internet Control Message Protocol

- Echo (ping)
- Redirect (from router to source host)
- Destination unreachable (protocol, port, or host)
- TTL exceeded (so datagrams don't cycle forever)
- Checksum failed
- Reassembly failed
- Cannot fragment

Global Internet

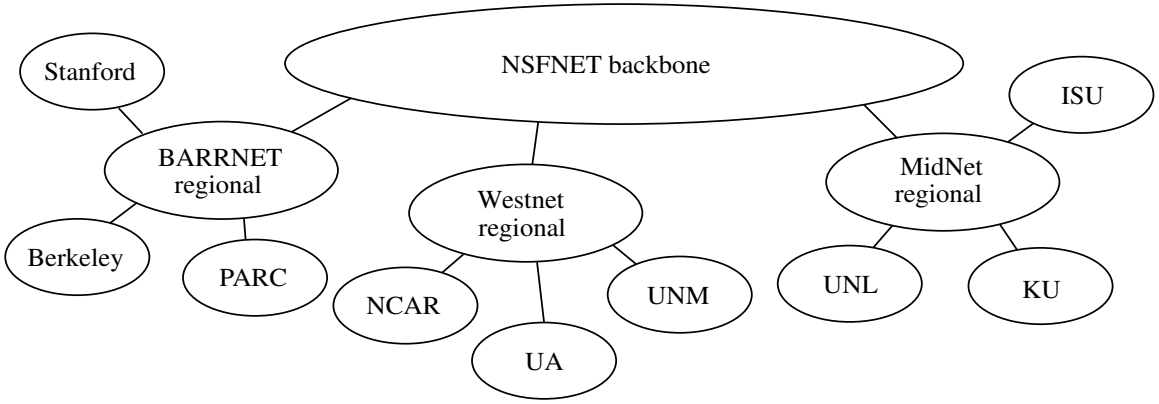
Scalability Issues

IP “hides” hosts in address hierarchy, but...

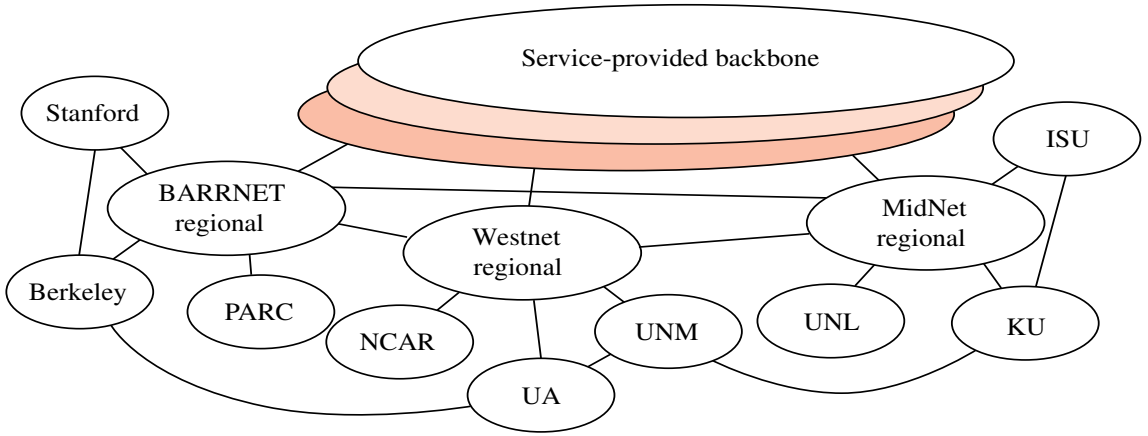
- Inefficient use of address space
  - class C network with 2 hosts (2/255 = 0.78% efficient)
  - class B network with 256 hosts (256/65535 = 0.39% efficient)
- Too many networks
  - today's Internet has tens of thousands of networks
  - routing tables do not scale
  - route propagation protocols do not scale

Internet Structure

Recent Past



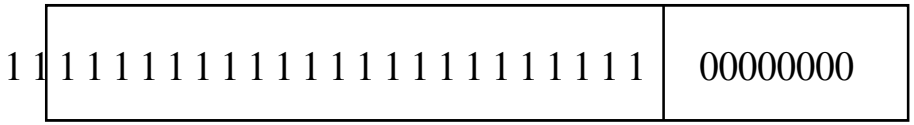
Today



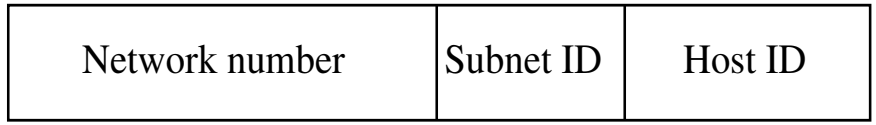
Subnetting

- Add another level to address/routing hierarchy: *subnet*
- *Subnet masks* define variable partition of host part of class A and B addresses
- Subnets visible only within site

Class B address

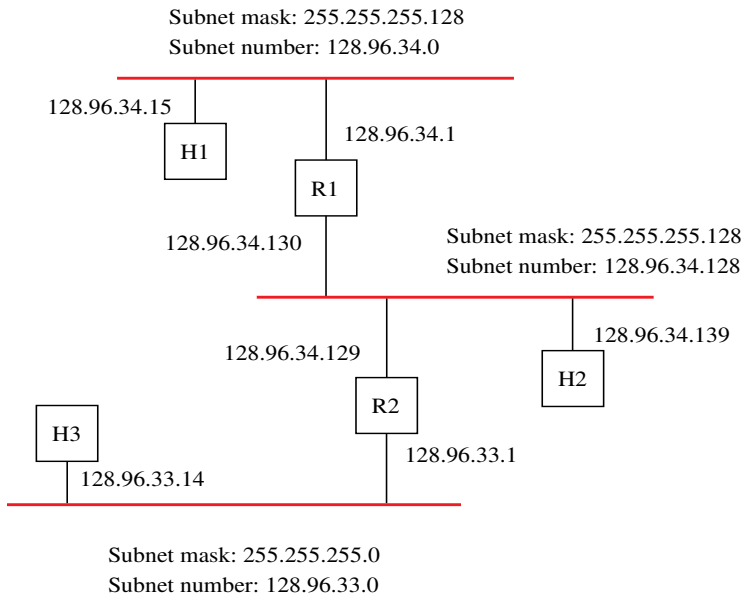


Subnet mask (255.255.255.0)



Subnetted address

Subnet Example



Forwarding table at router R1

Subnet Number	Subnet Mask	Next Hop
128.96.34.0	255.255.255.128	interface 0
128.96.34.128	255.255.255.128	interface 1
128.96.33.0	255.255.255.0	R2

Forwarding Algorithm

```
D = destination IP address
for each entry < SubnetNum, SubnetMask, NextHop>
   D1 = SubnetMask & D
   if D1 = SubnetNum
      if NextHop is an interface
         deliver datagram directly to destination
      else
         deliver datagram to NextHop (a router)
```

Notes

- Would use a default router if nothing matches
- Not necessary for all ones in subnet mask to be contiguous
- Can put multiple subnets on one physical network
- Subnets not visible from the rest of the Internet

Supernetting

- Assign block of contiguous network numbers to near-by networks
- Called CIDR: Classless Inter-Domain Routing
- Represent blocks with a single pair

  `<first_network_address, count>`
- Restrict block sizes to powers of 2
- Use a bit mask (CIDR mask) to identify block size
- All routers must understand CIDR addressing

Route Propagation

Idea: Impose a second hierarchy on the network that limits what routers talk to each other. (The first hierarchy is the address hierarchy that governs how packets are forwarded.)

- Autonomous System (AS)
  - corresponds to an administrative domain
  - examples: University, company, backbone network
  - assign each AS a 16-bit number
- Two-level route propagation hierarchy
  - interior gateway protocol (each AS selects its own)
  - exterior gateway protocol (Internet-wide standard)

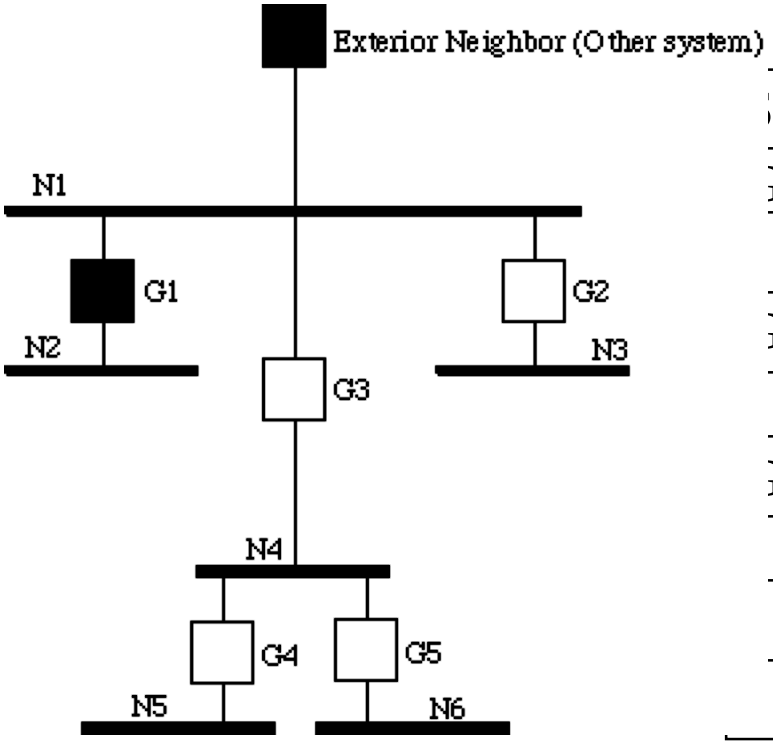
Popular Interior Gateway Protocols

- RIP: Route Information Protocol
  - developed for XNS
  - distributed with Unix
  - distance-vector algorithm
  - based on hop-count
- OSPF: Open Shortest Path First
  - recent Internet standard
  - uses link-state algorithm
  - supports load balancing
  - supports authentication

EGP: Exterior Gateway Protocol

- Overview
  - designed for tree-structured Internet
  - concerned with *reachability*, not optimal routes
- Protocol messages
  - neighbor acquisition: one router requests that another be its peer; peers exchange reachability information
  - neighbor reachability: one router periodically tests to see if the other router is still reachable; exchange HELLO/ACK messages; uses a k-out-of-n rule
  - routing updates: peers periodically exchange their routing tables (distance-vector)

EGP Example



ource Net: N1
1
N2
2
N3
3
N4
N5
N6

BGP-4: Border Gateway Protocol

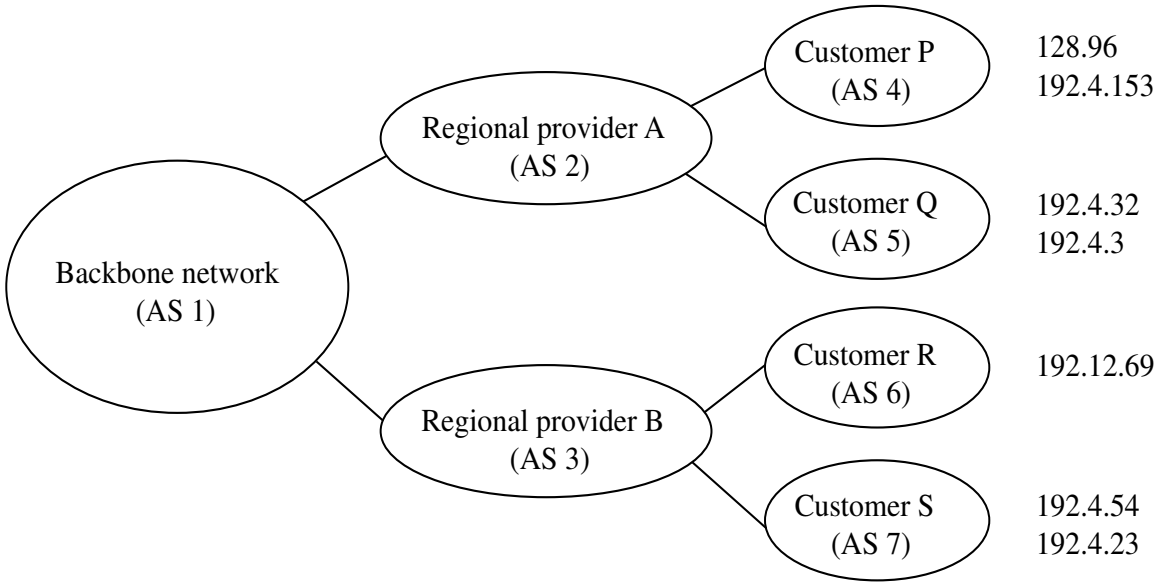
Assumes the Internet is an arbitrarily interconnected set of AS's. Define*local traffic* as traffic that originates at or terminates on nodes within an AS, and *transit traffic* as traffic that passes through an AS, we can classify AS's into three types:

- Stub AS: an AS that has only a single connection to one other AS; such an AS will only carry local traffic.
- Multihomed AS: an AS that has connections to more than one other AS, but refuses to carry transit traffic.
- Transit AS: an AS that has connections to more than one other AS, and is designed to carry both transit and local traffic.

Each AS has:

- One or more border routers
- One BGP *speaker* that advertises:
  - local networks
  - other reachable networks (transit AS only)
  - gives *path* information

BGP Example



- Speaker for AS 2 advertises reachability to P and Q
  Network 128.96, 192.4.153, 192.4.32, and 192.4.3, can be reached directly from AS 2.
- Speaker for backbone network then advertises
  Networks 128.96, 192.4.153, 192.4.32, and 192.4.3 can be reached along the path <AS 1, AS 2>.
- Speaker can also cancel previously advertised paths

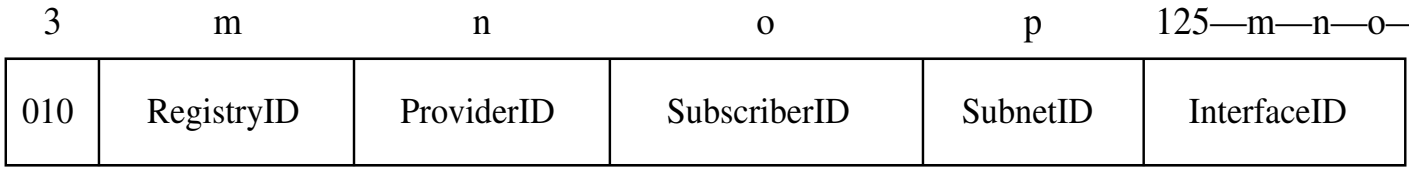
Next Generation IP (IPv6)

Major Features

- 128-bit addresses
- Multicast
- Real-time service
- Authentication and security
- Autoconfiguration
- End-to-end fragmentation
- Protocol extensions

IPv6 Addresses

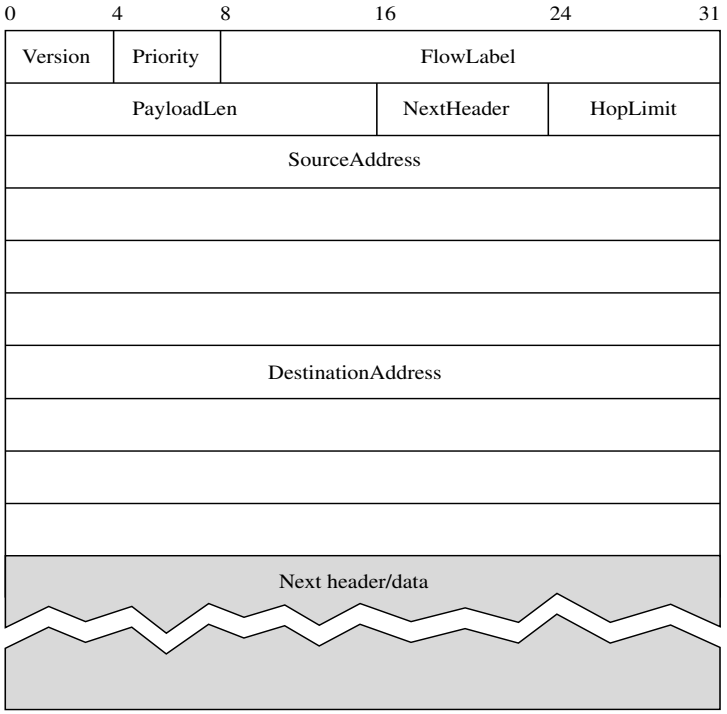
- Classless addressing/routing (similar to CIDR)
- Notation: x:x:x:x:x:x:x:x (x = 16-bit hex number)
  - contiguous 0s are compressed: 47CD::A456:0124
  - IPv6 compatible IPv4 address: ::128.42.1.87
- Address assignment
  - provider-based
  - geographic



Prefix	Use
0000 0000	Reserved
0000 0001	Unassigned
0000 001	Reserved for NSAP Allocation
0000 010	Reserved for IPX Allocation
0000 011 0000 1 0001 001	Unassigned Unassigned Unassigned Unassigned
010	Provider-Based Unicast Address
011	Unassigned
100	Reserved for Geographic-Based Unicast Addresses
101 110 1110 1111 0 1111 10 1111 110 1111 1110 0	Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned Unassigned
1111 1110 10	Link Local Use Addresses
1111 1110 11	Site Local Use Addresses
1111 1111	Multicast Addresses

IPv6 Header

- 40-byte “base” header
- Extension headers (fixed order, mostly fixed length)
  - fragmentation
  - source routing
  - authentication and security
  - other options



Internet Multicast

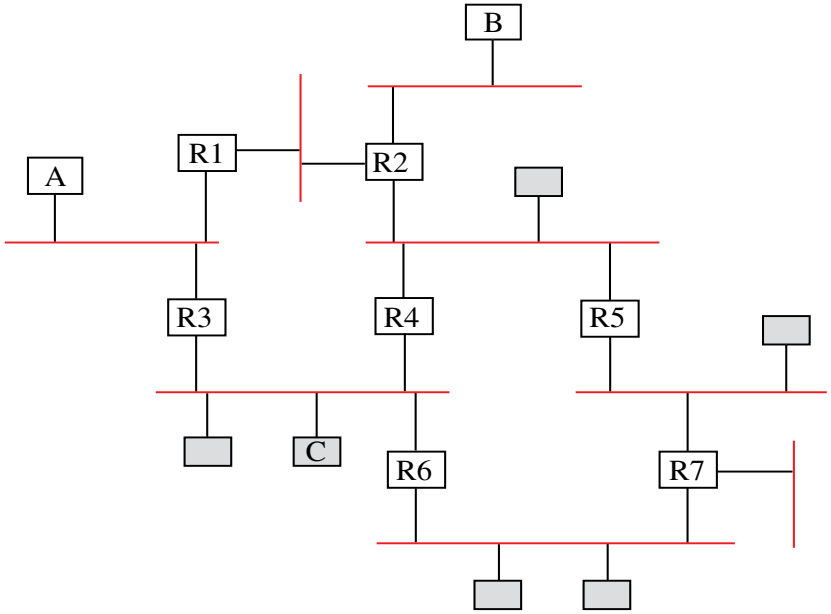
Overview

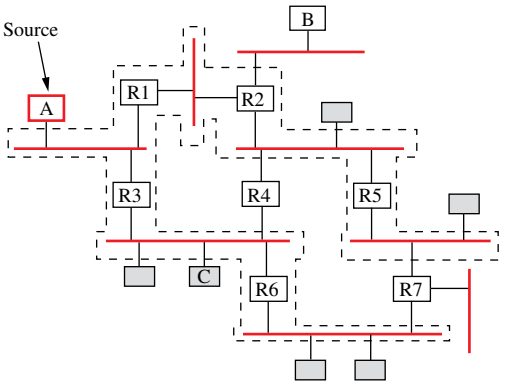
- IPv4
  - class D addresses
  - demonstrated with MBone
  - uses tunneling
- Integral part of IPv6
  - problem is making it scale

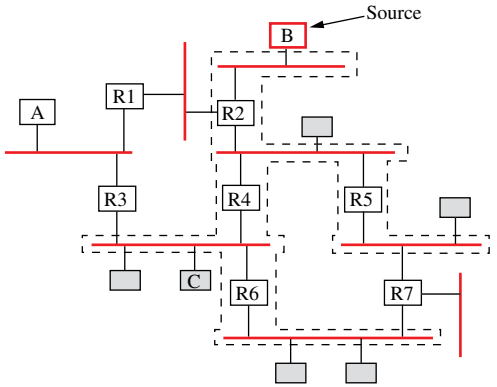
Link-State Multicast

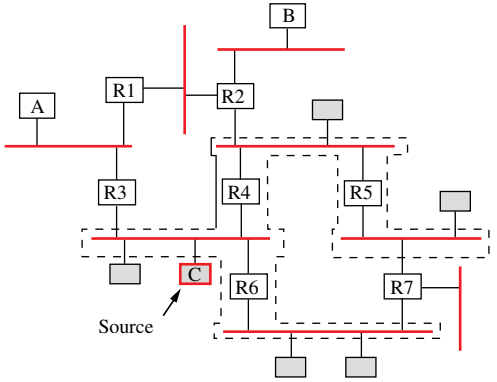
- Each host on a LAN periodically announces the groups it belongs to (IGMP).
- Augment update message (LSP) to include set of groups that have members on a particular LAN.
- Each router uses Dijkstra's algorithm to compute shortest-path spanning tree for each source/group pair.
- Each router caches tree for currently active source/group pairs.

Example


B
A
R1
R2
R3
R4
R5
C
R6
R7


Source
B
A
R1
R2
R3
R4
R5
C
R6
R7


Source
B
A
R1
R2
R3
R4
R5
C
R6
R7


B
A
R1
R2
R3
R4
R5
C
Source
R6
R7

Distance-Vector Multicast

Reverse Path Broadcast (RPB)

- Each router already knows that shortest path to destination S goes through router N.
- When receive multicast packet from S, forward on all outgoing links (except the one on which the packet arrived), iff packet arrived from N.
- Eliminate duplicate broadcast packets by only letting
- “parent” for LAN (relative to S) forward
  - shortest path to S (learn via distance vector)
  - smallest address to break ties

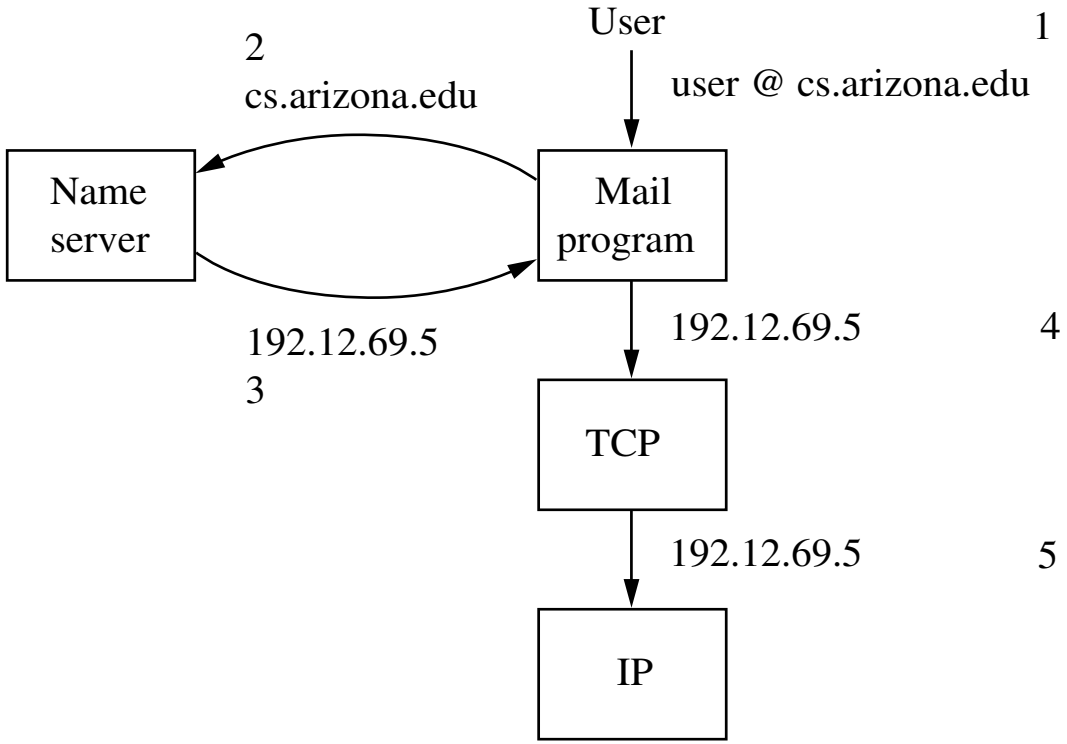
Reverse Path Multicast (RPM)

- Goal: Prune networks that have no hosts in group G
- Step 1: Determine of LAN is a *leaf* with no members in G
  - leaf if parent is only router on the LAN
  - determine if any hosts are members of G using IGMP
- Step 2: Propagate “no members of G here” information
  - augment **`<Destination, Cost>`** update sent to neighbors with set of groups for which this network is interested in receiving multicast packets.
  - only happens with multicast address becomes active.

Host Names (DNS)

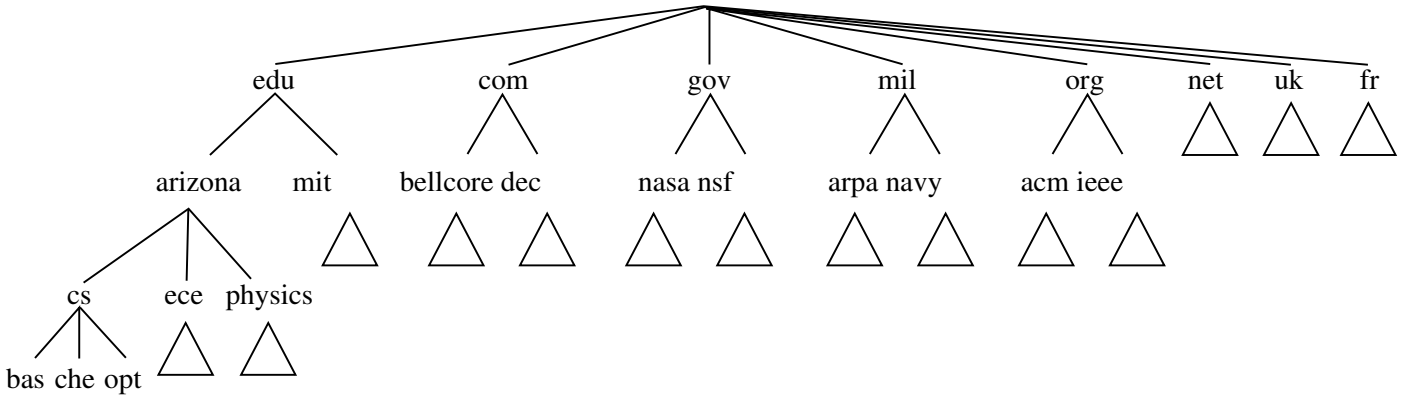
Overview

- Names versus Addresses
  - names are variable length, mnemonic, easy for humans to remember
  - addresses are fixed length, tied to routing, and easy for computers to process
- Name Space
  - defines set of possible names
  - flat versus hierarchical
  - consists of a set of name to value *bindings*


User
1
2
cs.arizona.edu
user @ cs.arizona.edu
Name
server
Mail
program
192.12.69.5
3
192.12.69.5
4
TCP
192.12.69.5
5
IP

Domain Hierarchy

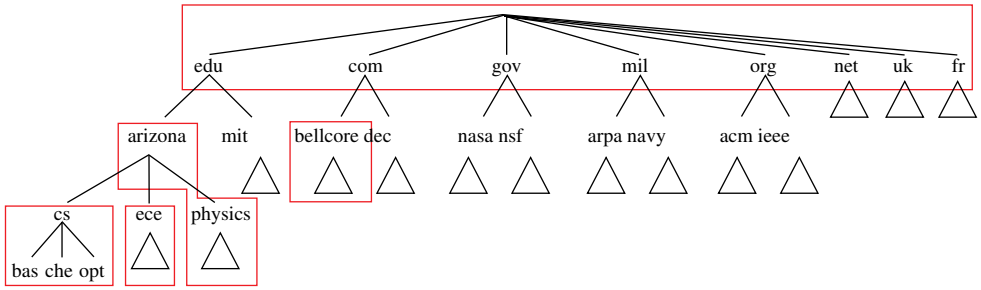
- Example hierarchy



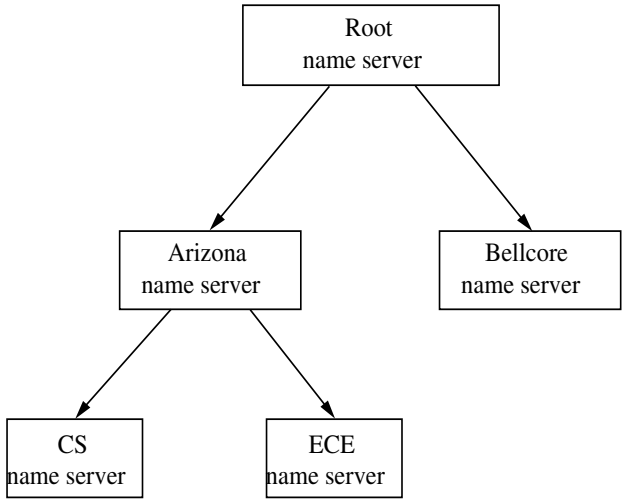
- Example name: **`cheltenham.cs.arizona.edu`**

Name Servers

- Partition hierarchy into *zones*



- Each zone implemented by two or more *name servers*



Resource Records

- Each name server maintains a collection of *resource records*

  <Name, Value, Type, Class, TTL>

- **Name/Value**: not necessarily host names to IP addresses
- **Type**
  - **NS**: the **Value** field gives the domain name for a host running a name server that knows how to resolve names within the specified domain.
  - **CNAME**: the **Value** field gives the canonical name for a particular host; it is used to define aliases.
  - **MX**: the **Value** field gives the domain name for a host running a mail server that accepts messages for the specified domain.
- **Class**: allow other entities to define types
- **TTL**: how long the resource record is valid

Example

Root 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>
```

Arizona 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>
<mars.physics.arizona.edu, 128.196.4.3, A, IN>
<venus.physics.arizona.edu, 128.196.4.4, A, IN>
```

CS server:

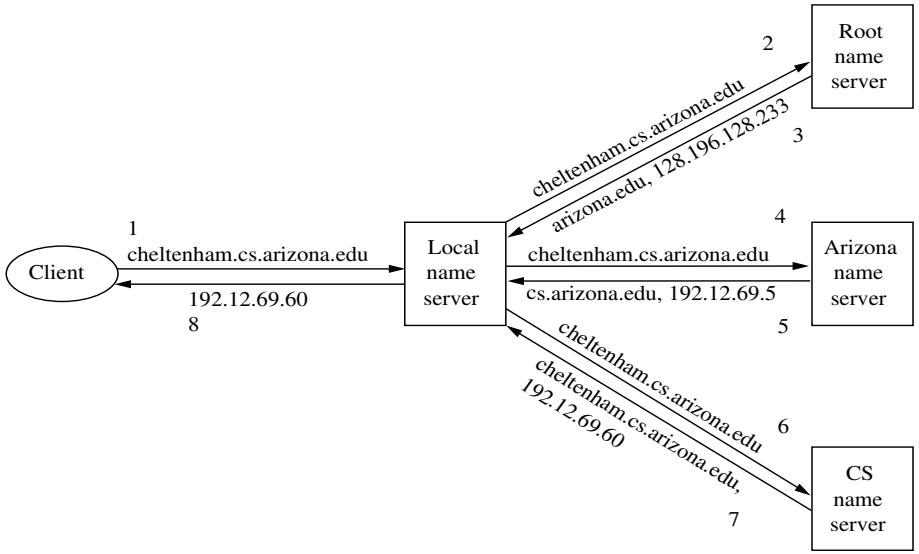
```
<cs.arizona.edu, optima.cs.arizona.edu, MX, IN>

<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>

<baskerville.cs.arizona.edu, 192.12.69.35, A, IN>
<bas.cs.arizona.edu, baskerville.cs.arizona.edu,
  CNAME, IN>
```

Name Resolution



- Strategies
  - forward
  - iterative
  - recursive
- Local server
  - need to know root at only one place (not each host)
  - site-wide cache