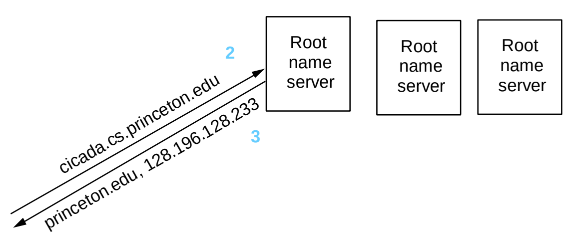
CSE/EE 461 Lecture 18
Naming and Multicast

Tom Anderson
tom@cs.washington.edu
Peterson, Chapter 9.1, 4.4

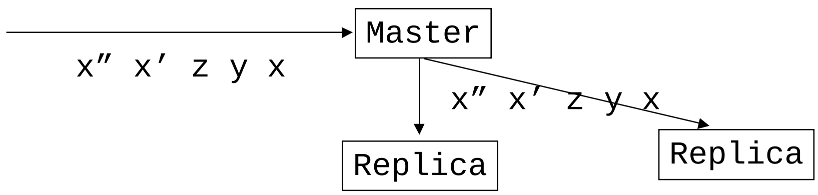
DNS Availability

- What happens if DNS service is not working?
- DNS servers are replicated
  - name service available if at least one replica is working
  - queries load balanced between replicas



Replica Consistency

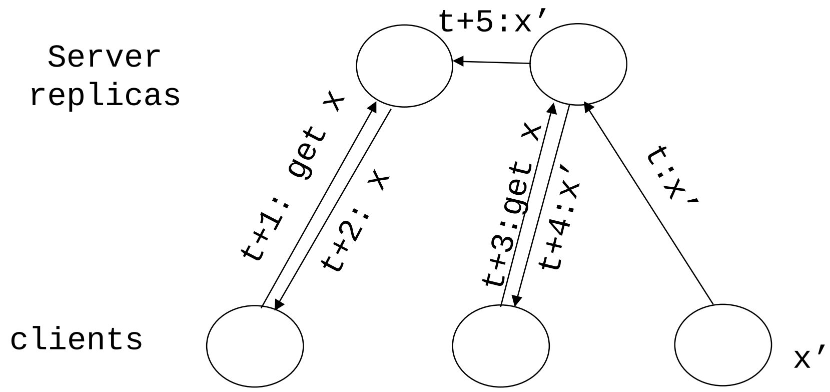
- How do we keep multiple copies of a database consistent?
- Apply same sequence of updates to each copy, *in the same order*
  - Example: send updates to master; master copies exact sequence of updates to each replica



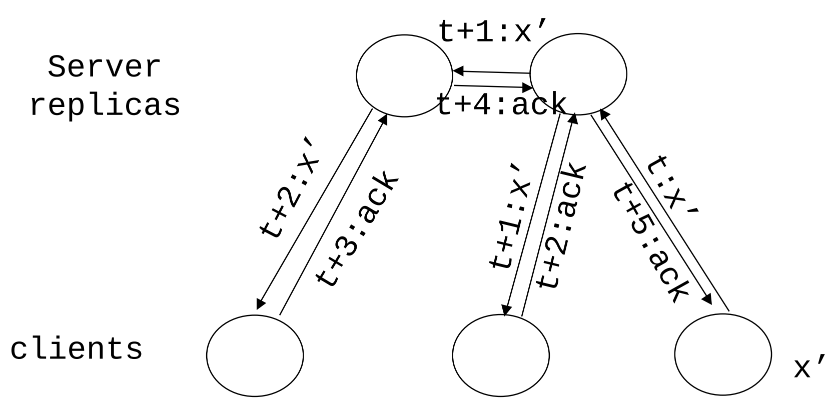
Replica Consistency

- While updates are propagating, which version(s) are visible?
- DNS solution: eventual consistency
  - changes made to a master server; copied in the background to other replicas
  - in meantime can get inconsistent results, depending on which replica you consult
- Alternative: strict consistency
  - before making a change, notify all replicas to stop serving the data temporarily (and invalidate any copies)
  - broadcast new version to each replica
  - when everyone is updated, allow servers to resume

Eventual Consistency Example



Sequential Consistency Example



Write doesn't complete until all copies invalidated or updated

Building on the DNS

- Email: tom@cs.washington.edu
  - DNS record for tom in the domain cs.washington.edu, specifying where to deliver the email
- Uniform Resource Locators (URLs) name for Web pages
  - e.g., www.cs.washington.edu/homes/tom
  - Use domain name to identify a Web server
  - Use “/” separated string for file name on the server (or program to run to generate the page)

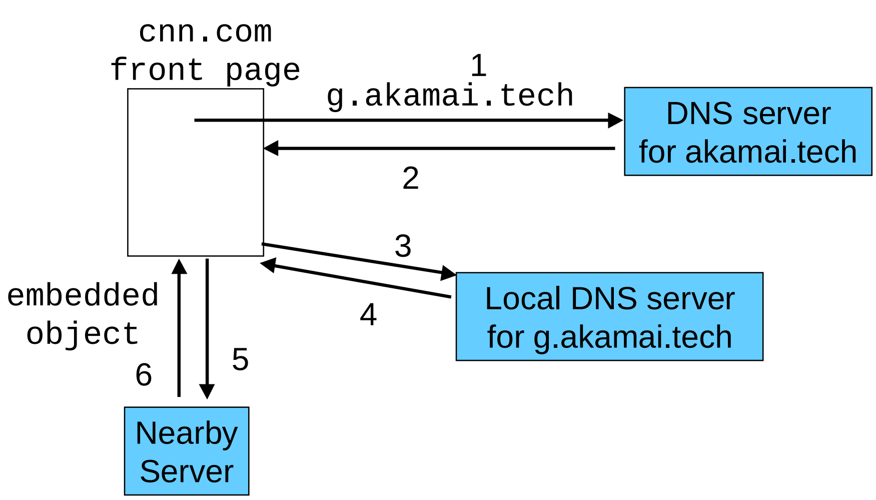
Future Evolution of the DNS

- Design constrains us in two major ways that are increasingly less appropriate
- Static host to IP mapping
  - What about mobility (Mobile IP) and dynamic address assignment (DHCP)?
- Location-insensitive queries
  - Many servers are geographically replicated; “yahoo.com” doesn’t refer to a single machine or even a single location (want closest server)

Akamai

- Use DNS to select a nearby Web cache
  - Front page points to g.akamai.tech
  - Special DNS server for akamai.tech, points to local akamai DNS server
    - return different server based on client location
    - use long TTL assuming clients don't move
  - Local DNS server points to local web server
    - use short TTL to allow rapid load balancing
  - Local server returns data
- Names no longer mean same thing everywhere

Akamai Example



Peer-to-Peer File Sharing

- Want to share files among large number of users; each serves subset of files
  - need to locate which servers have which files
  - would DNS be a good solution for this?
- Example: napster
  - centralized directory of all servers offering each file
  - users register files, make requests to napster central
  - napster returns list of servers hosting requested file
  - client directly connects to server to get file

Peer-to-Peer File Sharing (2)

- Can we locate files without a centralized directory?
  - for legal and privacy reasons
- Gnutella
  - organize servers into ad hoc graph
  - flood query to all servers, in breadth first search
    - – use hop count to control depth
  - if found, server replies back through path of servers
  - client makes direct connection to server to get file
- Freenet
  - same as gnutella, except depth first search, data goes back along request path, and servers in path cache files

Peer-to-Peer File Sharing (3)

- Can we locate files without an exhaustive search?
  - want to scale to thousands of servers
- Chord/CAN
  - organize servers into a predefined topology (e.g., k-dimensional hypercube or 2-D set of rings)
  - hash file names into search path

Multicast

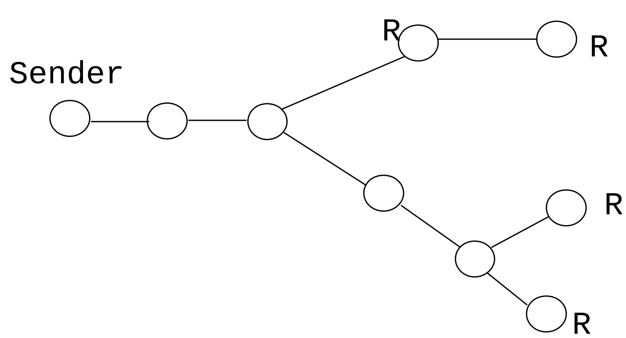
- Challenge: how do we efficiently send messages to a group of machines?
  - Need to revisit all aspects of networking
    - Routing
    - Autonomous systems
    - Address allocation
    - Congestion control
    - Reliable delivery
    - Ordered delivery

Multicast Motivation

- Send data to multiple receivers at once
  - broadcasting, narrowcasting
  - telecollaboration
  - software update
  - group coordination, subcasting
- Send question to unknown receiver
  - resource discovery
  - distributed database
  - anonymous directory services

Multicast Efficiency

- Send data only once down link shared by multiple receivers



Multicast Deployment

- How do we add multicast services to the Internet?
- IP multicast
  - special IP addresses to represent groups of receivers
  - receivers subscribe to specific channels
  - modify routers to support multicast sends
- Overlay network
  - PC routers, forward multicast traffic by tunneling over Internet
  - Works on existing Internet, with no router modifications

IP Multicast Service Model

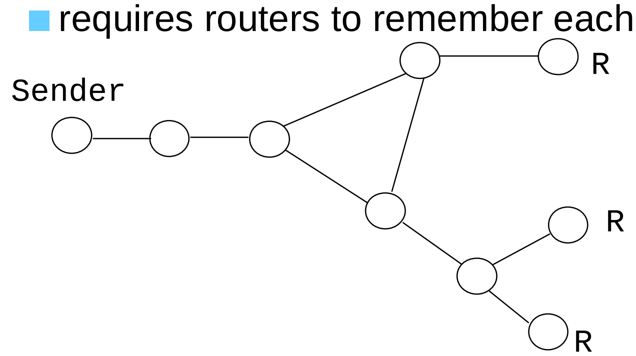
- Provided by internetwork, with help from LAN
- Best effort delivery (unreliable, unordered, …)
  - Packets addressed to group address (allocated from special range)
- Receivers
  - zero, one or many receivers
  - dynamic -- anyone can join, leave
- Senders
  - Any number of senders -- just send packet to group address

Internet Multicast Routing

- How do we distribute packets across thousands of LANs?
  - Each router responsible for its attached LAN
- Reduces to:
  - How do we forward packets to all interested routers? (DVMRP, M-OSPF, MBone)
  - How do hosts declare interest to their routers? (IGMP)

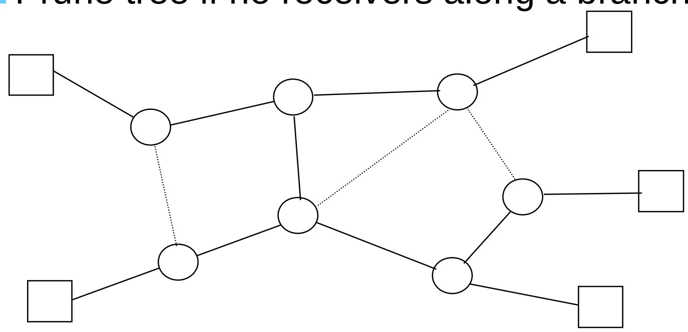
Why not Simple Flooding?

- If haven't seen a packet before
  - forward it on every link but incoming
  - requires routers to remember each pkt!



Multicast via Spanning Tree

- Send copies along the spanning tree
  - Ensures every host gets a copy
  - Prune tree if no receivers along a branch

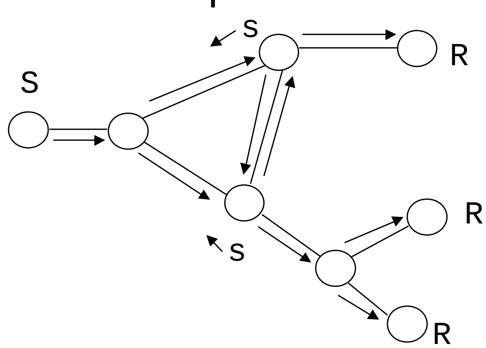


Distance Vector Multicast

- Intuition: unicast routing tables form inverse tree from senders to destination
  - why not use backwards for multicast?
  - Various refinements to eliminate useless transfers
- Implemented in DVMRP (Distance Vector Multicast Routing Protocol)

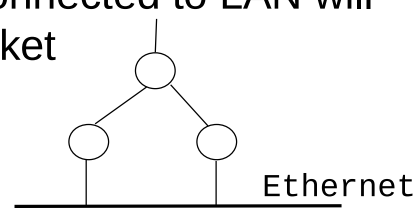
Reverse Path Flooding (RPF)

- Router forwards packet from S iff packet came via shortest path back to S



Redundant Sends

- RPF will forward packet to router, even if it will discard
  - each router gets pkt on all of its input links!
- Each router connected to LAN will broadcast packet



Reverse Path Broadcast (RPB)

- With distance vector, neighbors exchange routing tables
- Only send to neighbor if on its shortest path back to source
- Only send on LAN if have shortest path back to source
  - break ties arbitrarily

Truncated RPB

- End hosts tell routers if interested
- Routers forward on LAN iff there are receivers
- Challenges:
  - robust to router/host failures
  - avoid overloading LAN with announcements