CSE/EE 461

Ratul Mahajan
ratul@cs.washington.edu

21st Nov, 2003
Remember your friend, DNS?

- Hierarchical name space
  - hierarchy of name servers
- Decentralized administration
  - every domain manages its own namespace
- Caching to improve performance
DNS Evolution

- 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
  - What if I don’t care what server a Web page comes from, as long as it’s the right page?
    - e.g., a yahoo page might be replicated
Akamai

- Use the DNS to effect selection of a nearby Web cache

- Beware DNS caching
Network Address Translation (NAT)

- Organizations want to have their own private IPv4 address space
  - address space shortage
    - your home networks, for example
  - portable addressing
    - your home networks, again
  - sense of security
NATs (2)

- Hosts are assigned private addresses 10.1.1.1, ........
- Internal communication occurs normally
- All external communication goes through NAT
- NAT transforms each packet to maintain illusion of globally reachable addresses
NAT Mechanics

◆ Host wants to access an external web service
  • from 10.1.1.1 (home) to 128.208.3.88 (www.cs.washington.edu)

◆ NAT transforms outgoing packets
  • rewrite source address
    ▪ 10.1.1.1, port 4567 → 12.63.1.27, port 3456
  • the source address is dynamically allocated on connection setup; forgotten on connection teardown

◆ NAT transforms incoming packets
  • rewrite destination address
    ▪ 12.63.1.27, port 3456 → 10.1.1.1, port 4567
  • incoming destination address is matched to the transformation state
Implications of NAT

- No more universal connectivity
  - cannot have servers behind NAT

- No more fate sharing
  - NAT failure leads to connection failure

- Breaks some other protocols
Load Balancers

- A highly available, scalable web service uses thousands of servers
  - Want to balance load between them
- Original approach:
  - use DNS to translate service name to IP addresses of individual servers
  - load balance by round robin among servers
- Want server failures to be transparent
  - Use DNS to translate name to a single address
  - Load balancer forwards incoming requests to servers
    - Translates each incoming/outgoing packet (like NATs do)
Proxies

◆ A process that sits between the client and the server
  • it’s a server for the client
  • it’s a client for the server
  • it’s application-specific

◆ Applications of proxies: caches, transcoding, security
Proxies (2)

Proxy-based firewall to enhance security

- Two possible designs: classical vs. transparent
Peer-to-peer file sharing

- Want to share files among a large number of peers; each peer serves a subset of files
  - need to map files to servers
  - can we do this using DNS?
    - flat namespace
    - root is the central point of failure

- Three possible solutions
  - Napster (directory server)
  - Gnutella (flooding-based search)
  - Chord/CAN/Pastry (distributed hash tables)
Napster

- Centralized directory
- Peers register files they are willing to serve
- Peers query the directory for files they want

- Conceptually simple
- Single point of attack/failure
  - legal and technical
Gnutella

- No centralized directory
- Peers organize themselves in an ad hoc graph
- Flood queries on this graph
  - controlled flooding
  - can be scoped using hop count
- Peers with the file (if any) responds to the seeker

- Scalability limitations due to query flooding
  - Kazza is a hierarchical (scalable) version of Gnutella
Distributed Hash Tables (DHTs)

- No exhaustive search
- Examples: Chord/CAN/Pastry
- Hash peers and filenames onto a big ID-space
- Organize peers into a pre-defined topology based on their IDs
- Search for files over this topology using their IDs
Chord (an example DHT)
Summary

- The Internet is still evolving (very fast)
  - all kinds of boxes come together to make our applications run
    - DNS, NATs, load balancers, proxies, firewalls
  - new communication modes
    - p2p networks
  - overlay networks (next class)