Agenda

- **Last time**
  - Web Services
  - Assignment #3 out (due in 1 week)
- **This time (Tues 3/27)**
  - Finish naming: DNS
  - P2P
  - PA#2 back
- **Next time**
  - P2P
  - Midterm back
  - Assignment #3 due, Assignment #4 out
- ***Next next time (Thurs night / Fri afternoon)***
  - Time and global states

Schedule

Sun		Tues		Thurs	Fri	
		Midterm – closed books, closed notes 20		Web services (19) PA#3 out 22		
		DNS P2P (10) 27		P2P, #3 due,#4 out Time/global(11) 29	Time/global (11) 30	
		Time/global PA#5 Out 3		Coordination/agreement (12) 5		
		Coordination/agreement PA#4 due 10		Transactions (13) PA#5 proposal due 12		
		Transactions (13) Dis transactions (14) 17		Replication (15) 19		
		Student presentations PA#5 due 24		Student presentations 26		

Chapter 9: Naming

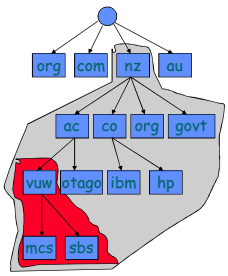
- **Basic ideas:**
  - Naming: names, objects, object addresses, binding
    - "Get the data from DataServ-1" – where is DataServ-1? What is it?
  - Name services
  - Location-independent naming: pros/cons?
  - Do names have to be unique: pros/cons?
  - Can names be reused: pros/cons?
  - DNS: partitioning, replication, caching
  - Iterative vs. recursive name resolution: pros/cons

DNS

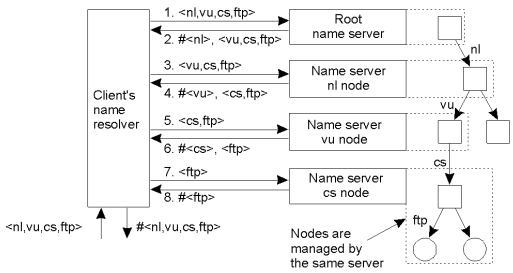
- **Problem: humans could not remember 192.0.34.166**
  - Solution circa 1985: /etc/hosts.txt
- **Problem:**
  - Did not scale
  - Local orgs wanted to administer their own namespaces
  - Only applicable to hostname→computer address
- **Solution: DNS (Mockapetris 1987)**
  - Partitioning, replication, caching
  - Generally used for looking up hosts and for determining electronic mail hosts
    - humphrey@cs.virginia.edu vs. humphrey@viper.cs.virginia.edu
  - DNS client is called a *resolver*

Implementing DNS

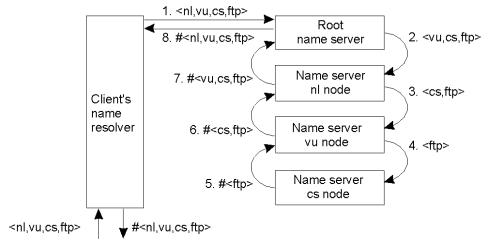
- **A global naming service**
- **Global**
  - Few nodes, many replicas
  - Slow (sec) resolution
  - Lazy updates
- **Administrative**
  - National
  - Many nodes, few replicas
  - Fast resolution (msec)
- **Organizational**
  - Many nodes, no replicas
  - Immediate resolution



DNS Iterative Name Resolution



DNS Recursive Name Resolution



The Zone Database

- **Data for a single domain**
  - Less delegated sub-domains
- **Names and addresses of <u>authoritative servers</u>**
  - for this domain
  - for delegated sub-domains (zones)
- **Management data**
  - time-to-live (TTL) parameter

Resource Records

- **Attached to nodes in the tree**
- **All terminal nodes have RRs**
- **Most non-terminal nodes have RRs**
- **All RRs in a zone have the zone's class**
- **RRs have:**
  - Domain name
  - RR TTL (time to live)
  - RR Type
  - RR Data

Important Resource Records

- **SOA --- Start Of Authority**
- **NS --- Name Server**
- **A --- Address**
- **MX --- Mail eXchange**
- **CNAME --- alias for the Canonical NAME**
- **PTR --- Pointer to another node**
- **TXT --- arbitrary TeXT**

Example RRs

- **Start of Authority**

```
@  IN  SOA  mcs.vuw.ac.nz mark.comp.vuw.ac.nz (
            199610140  ; Serial number
            28800      ; Refresh 8 hours
            7200       ; Retry 2 hours
            604800     ; Expire 7 days
            86400 )    ; Minimum 24 hours
```

- **Name Server**

```
IN   NS    downstage.mcs.vuw.ac.nz.
```


Example RRs cont.

- **Address**

```
embassy   IN   A    130.195.6.15
circa     IN   A    130.195.5.12
```

- **Canonical Name**

```
www    IN   CNAME   circa
```

DNS spoofing ("DNS cache poisoning")

- **DNS server accepts and uses incorrect information from a host that has no authority giving that information.**

One approach:

1. Attacker runs his own domain with HACKED DNS server in MyAttack.com (having "cnn.com=81.18.18.81")
2. Attacker sends request to DNS server asking to resolve www.MyAttack.com
3. The real DNS server does not know, so it asks YOU
4. Hacked machine replies, along with all of the records

Another attack:

- You ask UVA cs for IP of CNN.com
- Before UVa CS gets a legitimate reply from cnn.com, you send your own packet!

Secure DNS

- **Data Origin Authentication and Integrity for DNS data (digital signatures)**
- **Transaction and Request Authentication**

Name Services: Summary

- **Naming: underappreciated issue**
- DNS: partitioning, replication, caching
- **LDAP: Directory service for machines, humans**
  - Microsoft's Active Directory supports it

Chapter 10: P2P

- **Client/Server vs. P2P**
  - Where is the information and/or computational capacity?
- **Range of architectures from Client/Server to P2P**
- **5 general properties/characteristics of P2P systems**
  1. Each user contributes resources to the system
  2. Although each may contribute different resources, every node generally has the same functional capabilities and responsibilities
  3. Their correct operation does not depend on the existence of any centrally-administered systems
  4. Nodes arrive and depart continuously
  5. *Key issue: how to (dynamically) place/locate data across nodes?*

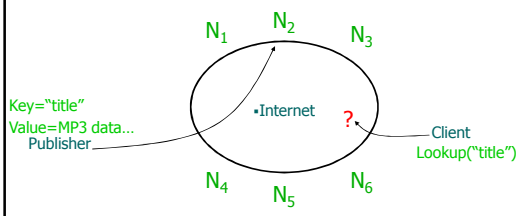
Generations of P2P systems

- **Gen I: Exploit PC resources**
  - Napster
- **Gen II: Eliminate centralized components**
  - Freenet, Gnutella, KaZaa
- **Gen III: Use structured overlays to guarantee number of network hops to find information**
  - Pastry, Tapestry, CAN, Chord

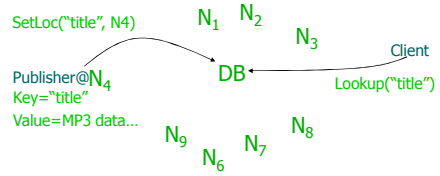
P2P: Overlay Networks

- **Overlays: all in the application layer**
  - Flexibility: protocol flexibility, messaging over TCP or UDP
  - Underlying physical net is transparent to the developer

P2P: The lookup problem

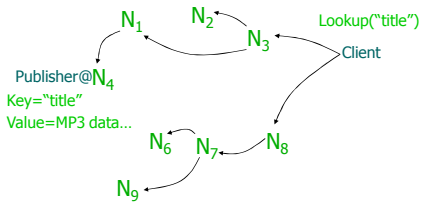


Centralized lookup (Napster)



- Simple, but O(N) state and a single point of failure

Flooded queries (Gnutella)



- Robust, but worst case O(N) messages per lookup