# CS5223
# Distributed Systems

Lecture 4: Naming

Instructor: YU Haifeng

# Today's Roadmap

- Chapter 5 of textbook
  - P2P and DHT is a important topic and will be covered in a separate lecture
- Basic concepts in naming
- Simple naming mechanisms
- Domain name service

# Motivation for Naming

- Example: We want to visit www.nus.edu.sg
  - The web server is located in some room that we don't know
  - There may be multiple web servers
  - ISP's may want to serve the content from their own machines – Akamai
  - www.nus.edu.sg → ip address → some physical machine (with specific MAC address)
- Example: Distributed file system
  - We know the file name (and full path), need to find out where the file is (i.e., on which machine, which disk sector, etc)
  - The file may migrate from one machine to another without changing its name

# Basic Concepts in Naming

- General definition for naming:
  - We have a “name” and we want to find/locate the “entity”
  - Definitions in the textbook is unnecessarily convoluted, and sometimes circular – please ignore those
  - A lot of things can be called “naming” – just a definition issue – no need to draw a black-and-white boundary

- Name space: The set of all possible names in the given context

- Naming system: The system that enables you to find the “entity”
  - Name resolution
  - The naming system can be either non-distributed or distributed
  - Same tradeoff as before…

# Different Kinds of Names

- Every name is a bit string
  - Some are human-friendly
- Hierarchical versus flat
- Alias
  - Multiple names for the same entity
  - Similar as symbolic links

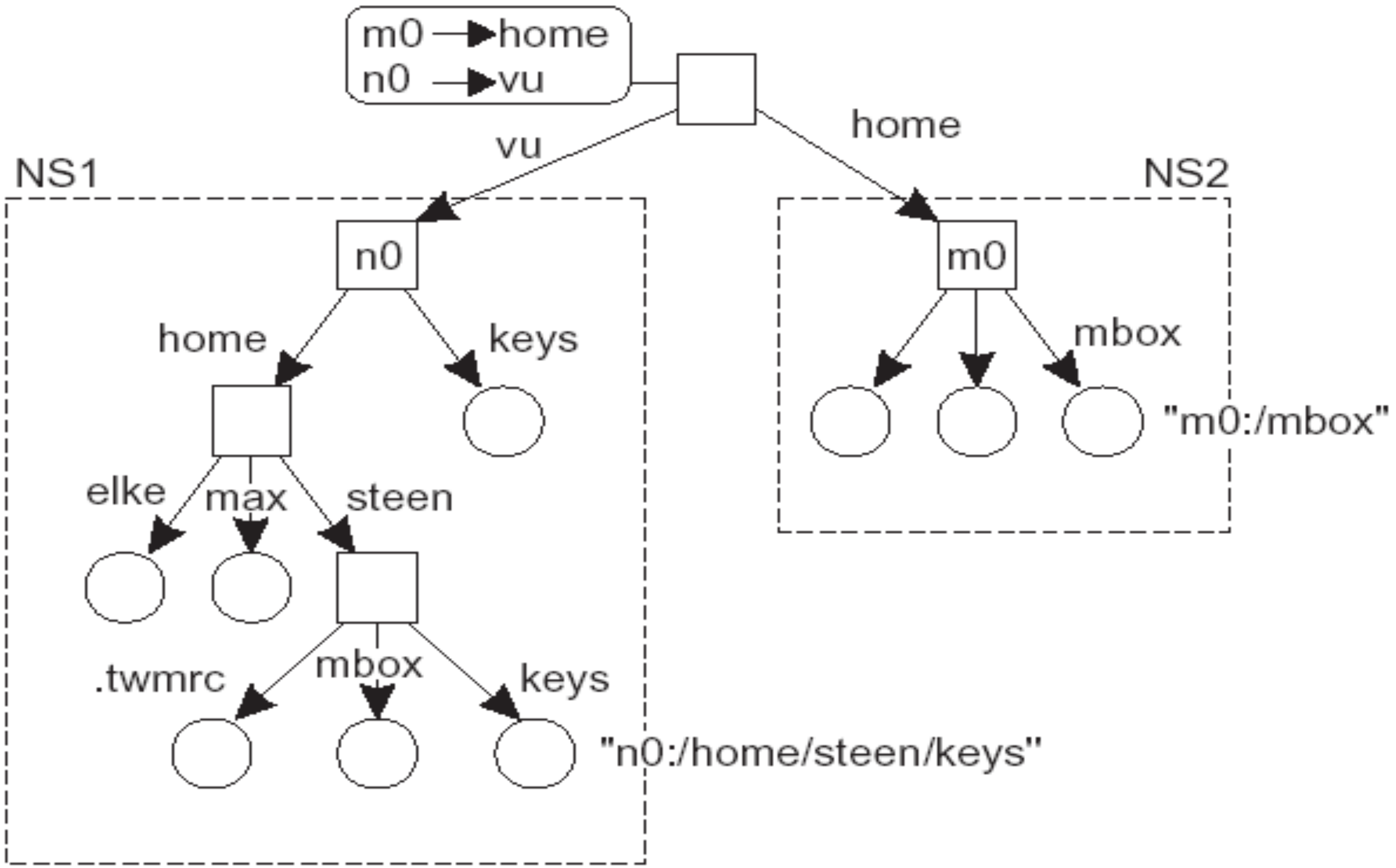
# Merging Name Spaces

- We have two different name spaces
  - Now we want to define a unified name space
  - Name space merging
  - Generalizes to multiple name spaces…how?
- Example: Merging two network file systems
- Example: Two users want to give each other access to files in their home directories
- Main issue to take care of: There can be name collisions

# Solution 1: New Root Node

- Merge by adding a new root node and make existing root nodes its children
- In principle, you always have to start in the new root
  - But you can avoid this by including the identifier of node from where resolution should start
  - Example, ~/XXX
- Problem: existing names need to be changed

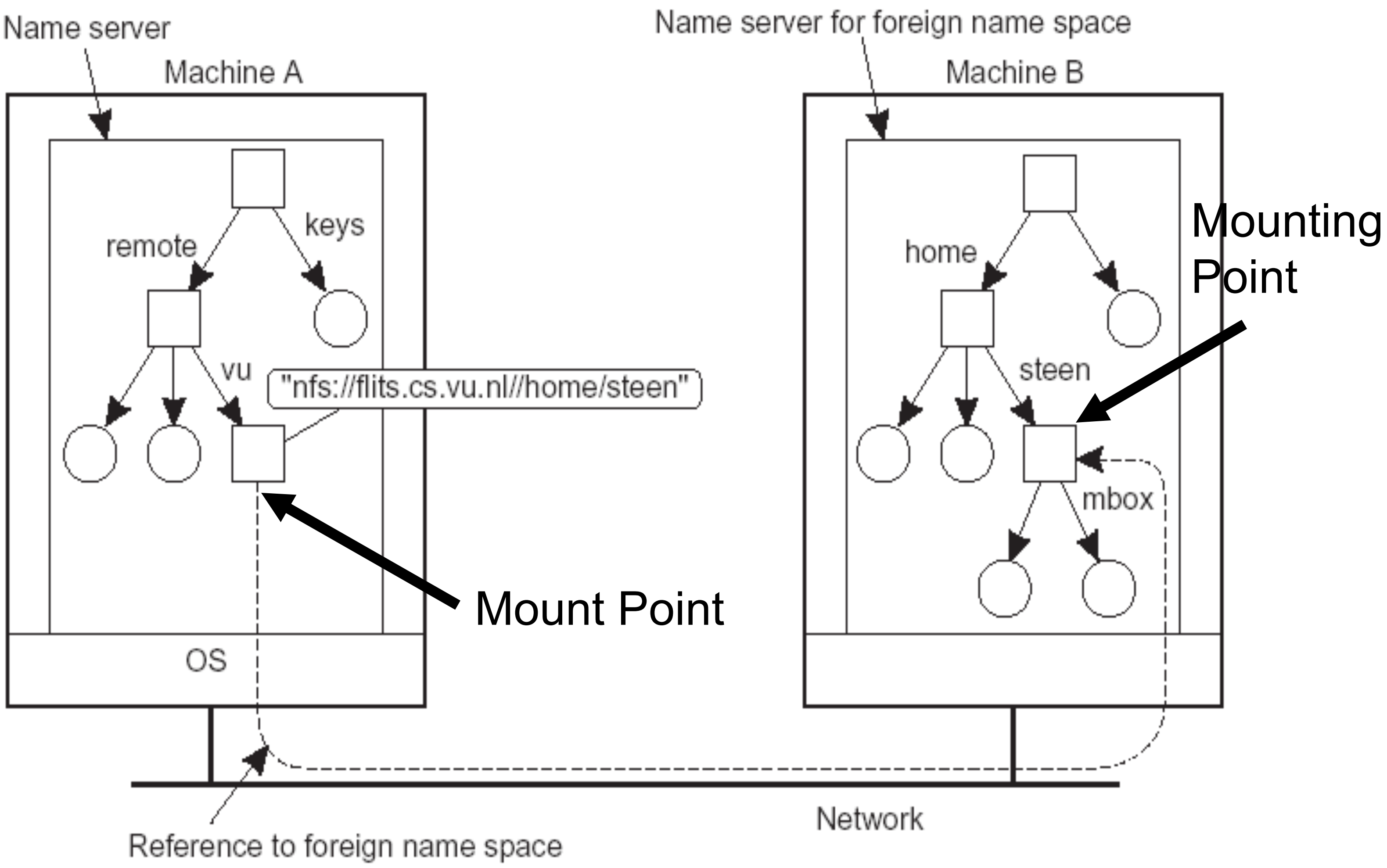
# Example: New Root Node



# Solution 2: Mounting

- Introduce nodes that contain the name of a node in a “foreign” name space
- Mount point: (Directory) node in naming graph that refers to other naming graph
- Mounting point: (Directory) node in other naming graph that is referred to

# Example of Mounting


Name server
Machine A
remote
keys
vu
"nfs://flits.cs.vu.nl//home/steen"
OS
Reference to foreign name space
Name server for foreign name space
Machine B
home
steen
mbox
Network

Mounting Point

Mount Point

# Simple Naming: Broadcast

- The “naming system” may or may not be a stand-alone software module
- Broadcast to resolve a name
  - Used in early day of computing
  - Simple is good!
- Drawback: performance
  - Alleviated but not fully addressed by using multicast

# Simple Naming: Forwarding Pointers

- Forwarding pointers: Same idea as mail forwarding

- The machine A with ip_addr1 was initially the web server for www.nus.edu.sg
  - We want to use another B machine now with ip_addr2 as the server

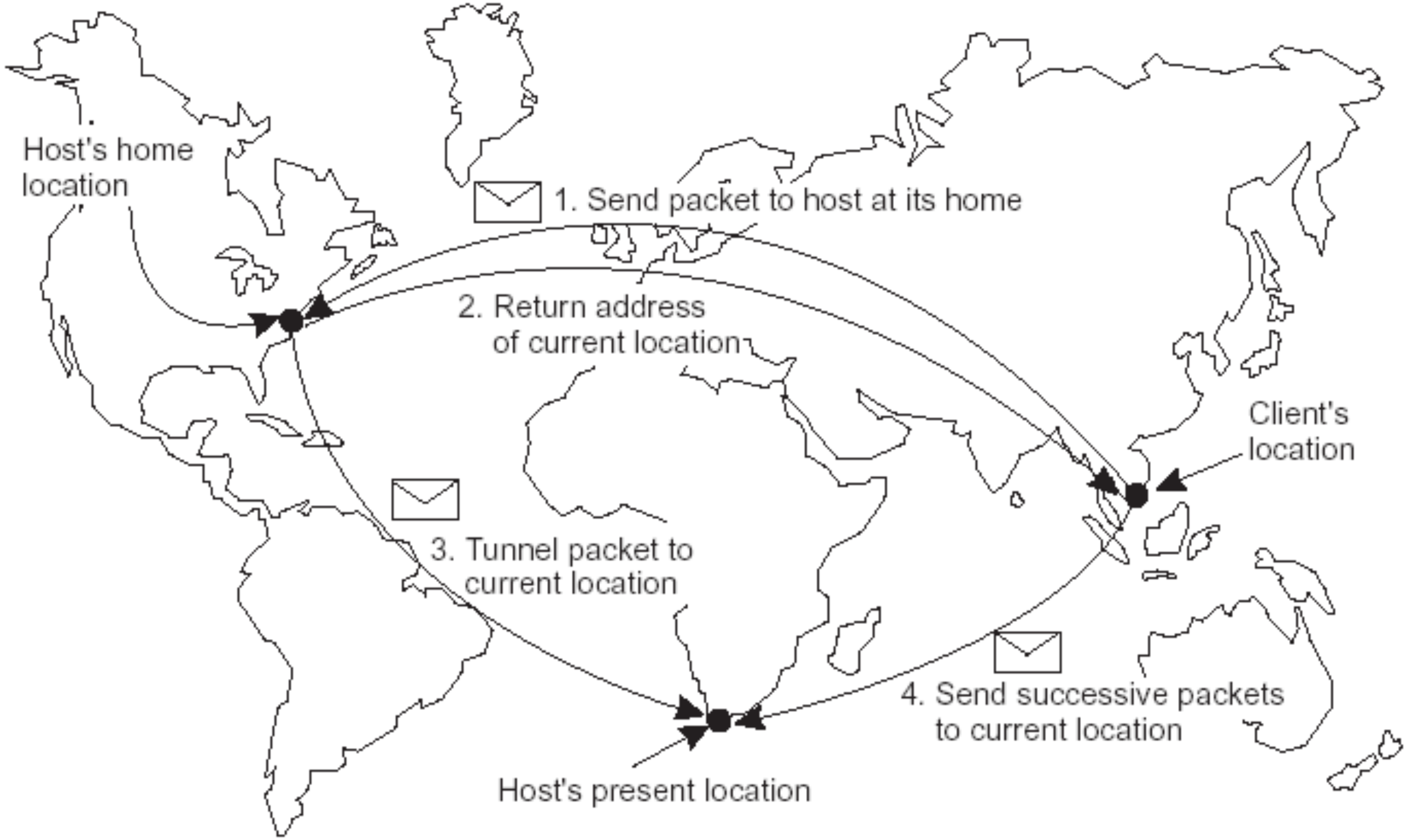
- A will forward the HTTP requests to B
- Or A can inform the client of B's address

- Drawback:
  - A needs to stay there forever
  - The forwarding chain can become long

# Simple Naming: Home-Based Approach

- Each entity has a “home”
  - The entity will register its IP address (which may change) with its home
- To find the entity, we contact the home first
- Only minor difference from forwarding pointers
  - Maximum one-hop forwarding
- Drawback:
  - The home needs to be there forever (i.e., have a fixed IP address)

# Example: Home-Based Approach


Host's home location
1. Send packet to host at its home
2. Return address of current location
Client's location
3. Tunnel packet to current location
4. Send successive packets to current location
Host's present location

# Domain Name System: DNS

- For looking-up IP addresses
  - Example: what ip address www.nus.edu.sg corresponds to

- DNS dates back to over 30 years ago
  - Replaces the original Internet naming scheme that keep everything in a master file (lack of scalability and freedom for each domain to assign internal names)

- Overall, DNS has been a very successful distributed system
  - Many of the early design decisions turn out to be critical later
  - Additional reading (non-compulsory): “Signposts in cyberspace: The Domain Name System and Internet Navigation” by R. Levien, National Academic Research Council, 2005
- We will study it as a case study…

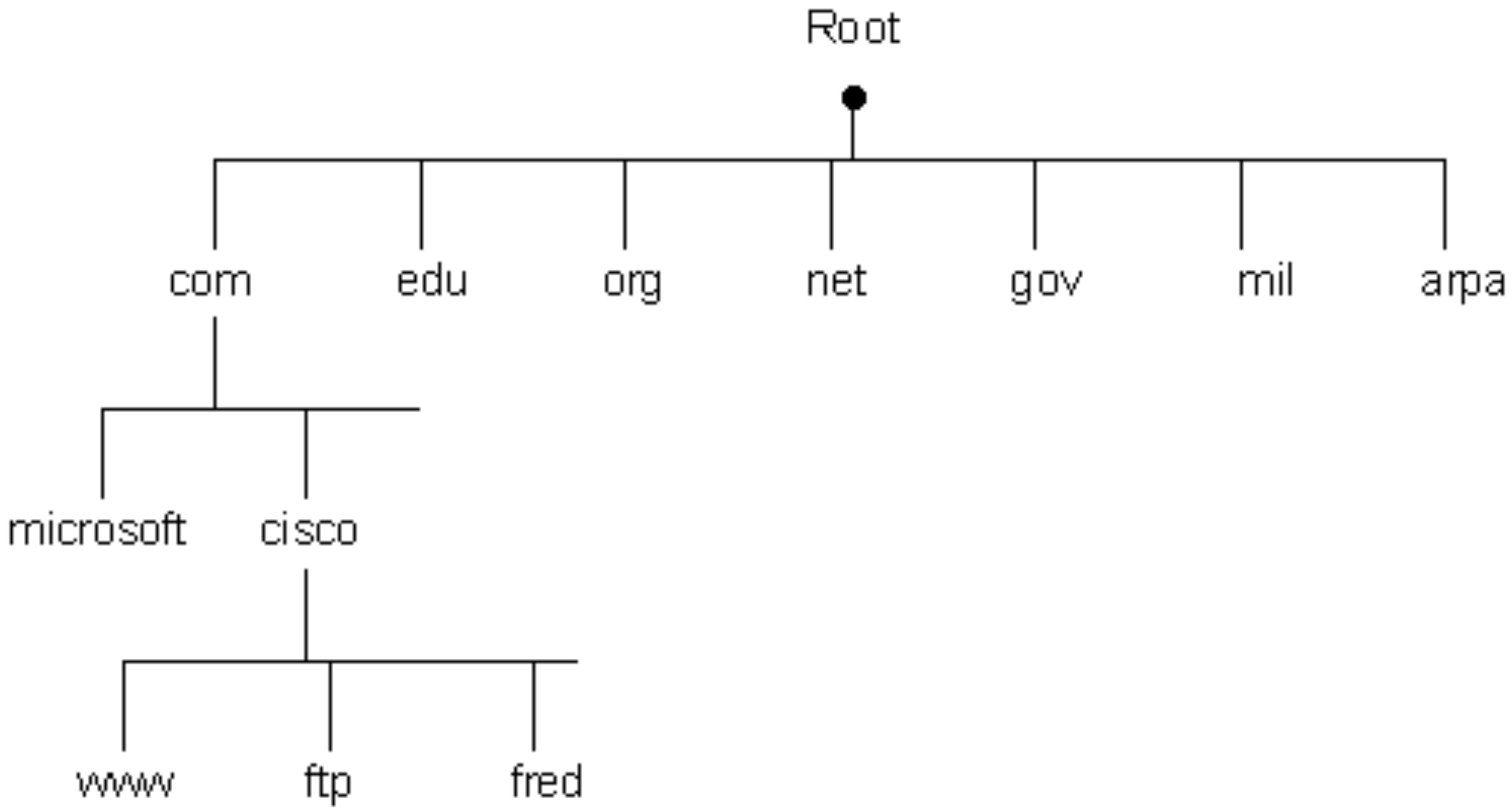
# DNS Name Space

$$\overbrace{\text{www.}\underbrace{\text{comp}}_{\text{label (max length 63)}}\text{.nus.edu.sg}}^{\text{DNS name (max length 255)}}\text{/}\sim\text{yuhf}$$

- Top-level domains
  - com, edu, net, org, etc.
  - sg, cn, fr, etc.

- The machine corresponding to XXX.XXX.sg may or may not be in Singapore

# The DNS Tree



(from http://www.rhyshaden.com/dns.htm)

# Relative Names in DNS

- Not supported in DNS specification

- However:
  - Your local software may automatically append the omitted part
  - Example, you can type ssh suna on any SoC unix machine, and you will connect to suna.comp.nus.edu.sg
  - If suna.comp.nus.edu.sg does not exist, ssh will try to connect to “suna” (which does not exist either)
  - Sometimes the local software may try appending different suffixes

# Domains in DNS

- A domain is a name space for which there is a single overall administrative authority to assignment the DNS names
  - Example, the collection of all XXX.nus.edu.sg is a domain
- A domain's name is the common suffix of all the DNS names in the collection
  - Example, nus.edu.sg is the domain's name
  - By definition, a domain's name is also a DNS name
- Potential confusion
  - A DNS name is sometime called domain name, which is different from domain's name defined above
  - We will only use DNS name and domain's name to avoid confusion

# Resolving DNS Names: The Perspectives of a Client

- Web browser
  - Make query to DNS server (how to find the DNS server?) to resolve www.nus.edu.sg into an IP address
  - Send HTTP request to that IP address at port 80
- ftp, telnet, ssh, smtp(email) are all similar
- Reverse resolution:
  - Translate an IP address into a DNS name
  - Only for IP addresses within the local domain

# Resolving a DNS Request

- The client needs to know the “local DNS server”
  - The IP address of this server is given to you by your ISP
  - Bootstrapping
- Client sends DNS query to local DNS server S
  - Containing the DNS name e.g., www.nus.edu.sg
- S checks whether it has either authoritative RR or cached RR for
  - www.nus.edu.sg
  - nus.edu.sg
  - sg

# Root DNS servers

- If no, S will send quest to the root DNS server
  - How does S know where to find the root DNS server?
- Root servers with well-known IP addresses
- https://www.iana.org/domains/root/servers

# Resolving a DNS Request

- The root DNS server sends back to S the IP address of the DNS server responsible for the domain “sg”

- S contact that DNS server
  - And then recursively the DNS server for “edu.sg”, “nus.edu.sg”, …

- After a DNS name is resolved, the RR will be cached

# Critical Design Decisions in DNS

- Delegation of responsibility
  - Allow individual domains to assignment their own DNS names
  - Hierarchical naming strcuture
- Caching and replication
  - Controls the overhead of DNS lookups
  - Controls the load on the root DNS servers

# Vulnerability of DNS

- The root servers – “single points” of failure
  - Target of DoS attacks
  - Already happened

- Alternatives:
  - Completely decentralized distributed naming system for the Internet
  - Already possible technically, but backward-compatibility issues need to be addressed

# History Readings (Non-compulsory)

- DNS root servers under attack
  - http://en.wikipedia.org/wiki/DNS_Backbone_DDoS_Attacks

# The Fundamental Need for Bootstrapping in Naming

- To resolve a name, you need to know at least one bootstrapping point
  - Broadcast: Bootstrapping point not needed (or every machine is a bootstrapping point?)
  - Forwarding pointers: The first machine you contact
  - Home-base approach: The home
  - DNS: Your local DNS server and the root DNS servers
- Called “closure mechanism” on the textbook
- Bootstrapping mechanism significantly influences the “goodness” of a naming system

# Today's Summary

- Basic concepts in naming
- Simple naming mechanisms
- Domain name service