

CSSE4004-Lecture 4

Naming

Learning Objectives

After this week you should:

- understand naming and name resolution concepts
- be able to describe and compare various naming approaches used in DS
- understand naming concepts which can be applied to mobile entities
- be able to describe well known naming services

2

Outline

- Names, Identifiers and Addresses
- Flat Naming
- Structured Naming
- Attribute-based Naming

3

Names, Identifiers and Addresses

- Names refer to entities
 - e.g., processes, users, mailboxes, network connections
- Special types of names exist:
 - Addresses
 - Identifiers
 - Human-friendly names
- Addresses
 - To use (operate on) an entity you need an access point
 - The name of an access point is an *address*
 - Entities can have more than one access point at a time
 - An entity may change access points e.g., when it changes location
 - *Location-independent names* are not tied to an address

4

Names, Identifiers and Addresses (2)

- Identifiers
 - are a special type of name that:
 - Refers to at most one entity
 - Always refers to the same entity
 - Are limited to one per entity
 - are often represented in machine-readable form
 - usually flat names
- Human-friendly names:
 - Generally represented as a character string e.g., www.news.com.au or /root/ryan/slides/

5

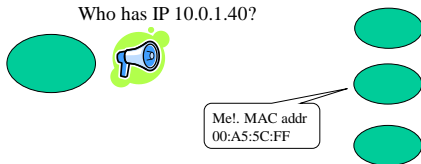
Flat Naming

- Flat names:
 - Random bit strings (name doesn't help locate access point)
- Names must be resolved to address
- There are many name resolution approaches for flat names:
 - Broadcasting (or multicasting) approaches
 - Forwarding pointers
 - Home-based approaches (e.g., Mobile IP)
 - Distributed Hash Tables (DHTs)
 - Hierarchical approaches

6

Resolving Flat Names - Broadcasting

- Broadcasting (or multicasting) the ID requesting the entity to return its current address
 - e.g., ARP (Address Resolution Protocol)



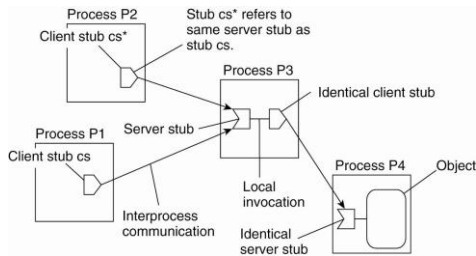
7

Resolving Flat Names - Forwarding Pointers (1)

- When entity moves, it leaves a pointer to its new location
- To find entity, must follow trail of pointers to entity's current location
- Problems:
 - Can get long chains of pointers (not scalable)
 - If chain breaks, can't find entity
- Potential solution:
 - Stub Scion Pair (SSP) chains

8

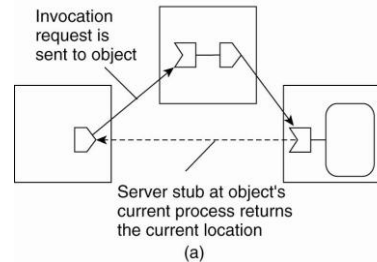
Resolving Flat Names - Forwarding Pointers (2)



The principle of forwarding pointers (using client stub, server stub) pairs

9

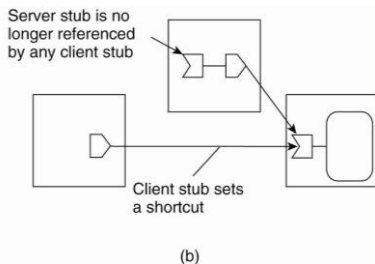
Resolving Flat Names - Forwarding Pointers (3)



Redirecting a forwarding pointer by storing a shortcut in a client stub

10

Resolving Flat Names - Forwarding Pointers (4)



Redirecting a forwarding pointer by storing a shortcut in a client stub

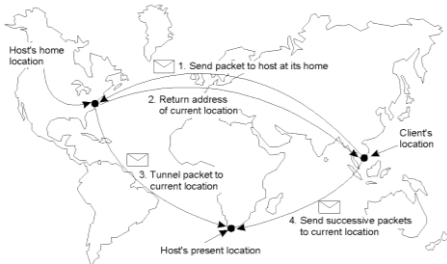
11

Resolving Flat Names -Home-Based Approach (2)

- Approach designed for mobile entities
- Entities have home location that keeps track of entity's current location
- Example of home-based approach is Mobile IP:
 - Mobile entity has fixed IP address
 - All traffic to mobile entity goes to entity's Home Agent (home location)
 - Home Agent forwards traffic to Care-of-Address (mobile entity's IP in its current network)
 - Whenever mobile entity changes network, it gets new Care-of-Address, which it registers with Home Agent.

12

Resolving Flat Names - Home-Based Approach (2)



Mobile IP example

13

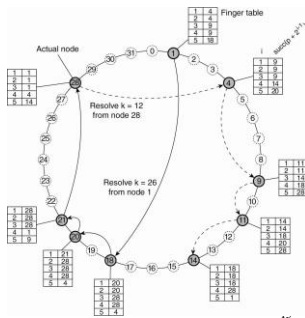
Resolving Flat Names - Distributed Hash Tables (1)

- Distributed Hash Tables (DHT):
 - Consist of many distributed nodes
 - Map data to a key value using a hash function
 - Each node is responsible for key values (and the associated data) in a particular range
 - Have fast lookup times
- Examples of DHTs include Chord, INS/Twine, Tapestry, Superstring

14

Resolving Flat Names - Distributed Hash Tables (2)

Example:
Resolving key 26 from node 1 and key 12 from node 28 in a Chord system



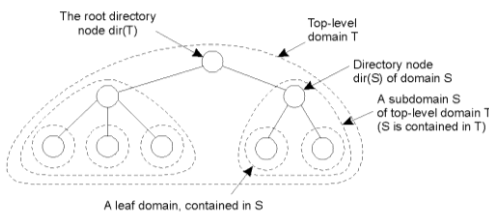
15

Resolving Flat Names - Hierarchical Approach (1)

- Network is divided into **domains**, with each domain subdivided into smaller subdomains
- **Leaf domain** is lowest level domain (typically LAN or Cell in mobile phone network)
- Each domain has a **directory node** that keeps track of entities in that domain
- Directory nodes for leaf domains store address of entities in that domain
- Directory nodes for non-leaf domains store reference to lower-level domain containing entities

16

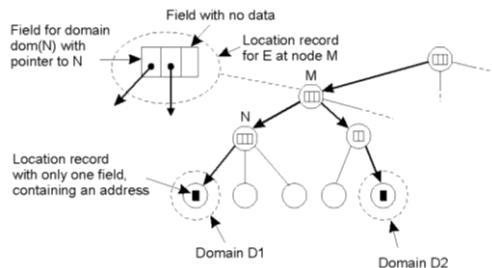
Resolving Flat Names - Hierarchical Approach (2)



Hierarchical organization of a location service into domains, each having an associated directory node (each entity in domain S is described in Dir(S))

17

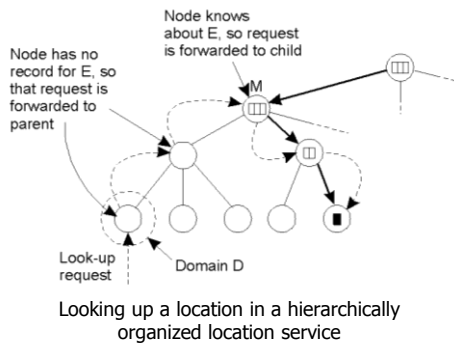
Resolving Flat Names - Hierarchical Approach (3)



An example of storing information of an entity having two addresses in different leaf domain (if the entity has more than one address)

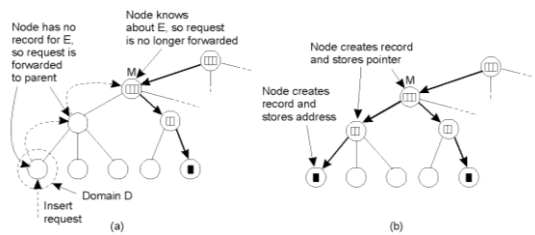
18

Resolving Flat Names - Hierarchical Approach (4)



19

Resolving Flat Names - Hierarchical Approach (5)



- An insert request for a replica in Domain D is forwarded to the first node that knows about entity E
- A chain of forwarding pointers to the leaf node is created

20

Structured Naming

- Flat names are not convenient for humans
- Structured names:
 - Composed from simple human-readable names
 - Generally supported by naming systems
 - Names are organised into a [name space](#)

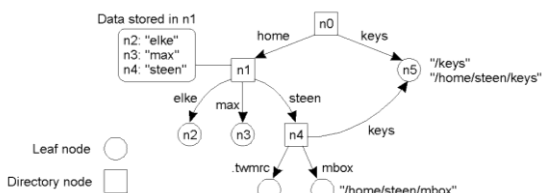
21

Structured Naming - Name Spaces (1)

- Name spaces for structured names are represented as a labeled, directed naming graph in which:
 - A **leaf node** represents a (named) entity
 - A **directory node** is an entity that refers to other nodes
 - Outgoing edge is represented as (*edge label*, *node identifier*)
 - Root node** has only outgoing edges
- Naming graphs are usually directed acyclic graphs
- Each path in a naming graph can be referred to by a sequence of edge labels separated by a special character e.g., `/root/ryan/slides`

22

Structured Naming - Name Spaces (2)



A general naming graph with a single root node

23

Structured Naming - Name Resolution in a Name Space

- Need to know how and where to start -> need closure mechanism
- Closure mechanism deals with selecting initial node in a name space from which name resolution is to start
- Closure mechanisms are often implicit:
 - `/home/jaga/file` (starts in the root directory)
 - `61 7 3365 2542` (dial phone)

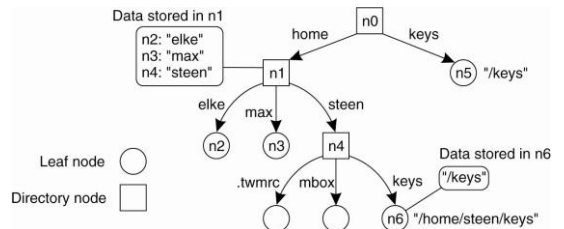
24

Structured Naming - Linking and Mounting (1)

- Aliases
 - Another name for the same entity
 - Symbolic links in UNIX enable more than one path for a file
- Mounting
 - Used to merge different name spaces transparently
 - E.g., directory node 1 stores identifier of directory node 2 (in different, foreign, name space)

25

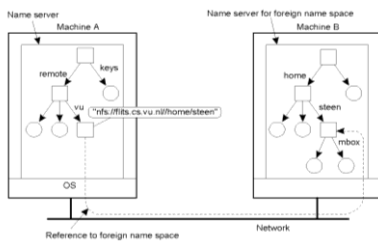
Structured Naming - Linking and Mounting (2)



The concept of a symbolic link explained in a naming graph

26

Structured Naming - Linking and Mounting (3)



Mounting remote name spaces through a specific access protocol

27

Structured Naming - Name Resolution

- Structured names resolved using naming service (implemented by name servers)
- Naming service allows addition, removal and look-up of names
- DS naming services are distributed
- Large-scale distributed naming services are usually hierarchical:
 - **Global level:** high-level directory nodes
 - **Administrational level:** mid-level directory nodes grouped into separate administrations
 - **Managerial level:** low-level directory nodes within a single administration
- The Domain Name System is a good example of a structured name resolution mechanism

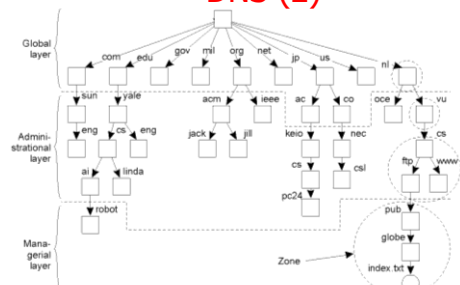
28

Structured Naming - DNS (1)

- Large distributed name service used by Internet
- DNS name space is hierarchical
- DNS labels use alphanumeric character strings separated by a "." (e.g., www.google.com)
- A path name in DNS is called a domain name
- DNS is primarily used to look-up IP addresses for hosts and mail servers (e.g., browser uses DNS to resolve www.itee.uq.edu.au to 130.102.79.1)

29

Structured Naming - DNS (2)



An example partitioning of the DNS (Domain Name System) name space, including Internet-accessible files, into three layers

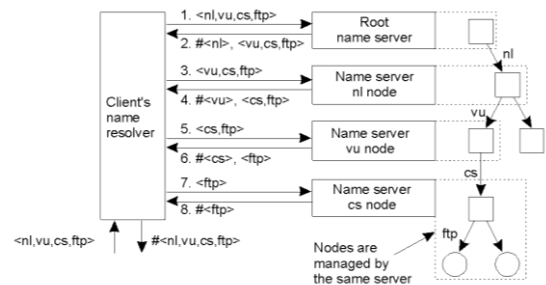
30

Structured Naming - DNS (3)

- Client (e.g., browser) contacts name resolver
- Name resolver uses either iterative or recursive approach
- Iterative name resolution:
 1. Name resolver contacts name servers for help
 2. Starts at root name server
 3. Each name server resolves as much of name as it can before referring name resolver to another name server who knows more
 4. Process repeats until name is fully resolved, or cannot be resolved anymore

31

Structured Naming - DNS (4)



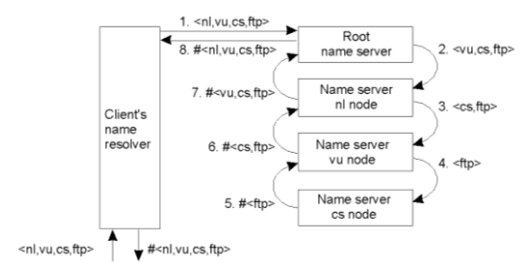
32

Structured Naming - DNS (5)

- Recursive name resolution:
 1. Client (e.g., browser) contacts name resolver
 2. Name resolver sends name to root name server
 3. Intermediate result not passed to client, rather it is sent to next name server
 4. Process repeats until name is fully resolved, then servers pass fully resolved name back
 5. Root name server passes fully resolved name to name resolver

33

Structured Naming - DNS (6)



34

Structured Naming - DNS (7)

- Comparison between Iterative and Recursive:
 - Recursive approach has higher performance demand on name servers
 - Recursive approach can use result caching more effectively
 - Iterative approach has higher communication costs

35

Attribute-Based Naming

- Each entity is described by (*attribute, value*) pairs
- Attribute-based queries:
 - Users specify attributes they are looking for
 - Naming system should return one (or more) entities with the specified attributes
- Attribute-based naming systems are commonly known as [directory services](#)
- Lightweight Directory Access Protocol (LDAP) is a common directory service

36

Attributed-Based Naming - LDAP (1)

- Derived from OSI X.500 directory service
- LDAP directory service stores *directory entries*
- Directory entries consist of (attribute,value) pairs.
- Directory Information Base (DIB) is collection of all directory entries in an LDAP service
- Each LDAP directory entry has globally unique name (Directory Information Tree) based on hierarchy of naming attributes
 - e.g., /C=NL/O=Vrije Universiteit/OU=Comp.Sc.

37

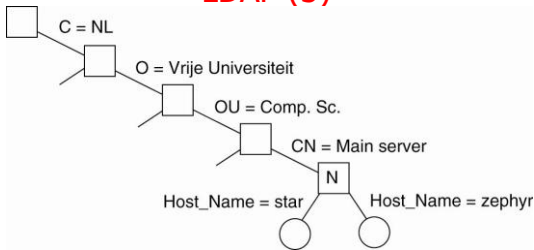
Attributed-Based Naming - LDAP (2)

Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Comp. Sc.
CommonName	CN	Main server
Mail_Server	—	137.37.20.3, 130.37.24.6, 137.37.20.10
FTP_Server	—	130.37.20.20
WWW_Server	—	130.37.20.20

A simple example of an LDAP directory entry using LDAP naming conventions

38

Attributed-Based Naming - LDAP (3)



Part of a Directory Information Tree (DIT)

39

Attribute-Based Naming - Decentralised Schemes

- Driven by advent of peer-to-peer
- Need efficient mapping of (attribute,value) pairs to avoid exhaustive search of network
- We examine two approaches:
 - Distributed Hash Tables (INS/Twine)
 - Semantic Overlay Networks

40

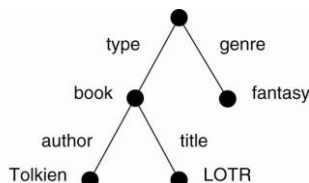
Attribute-Based Naming - Distributed Hash Tables (1)

- In INS/TWINE each entity (resource) is described by a hierarchical attribute-value tree (AVTree)

```

description {
  type = book
  description {
    author = Tolkien
    title = LOTR
  }
  genre = fantasy
}
    
```

General description of a resource



Description as an AVTree

Attribute-Based Naming - Distributed Hash Tables (2)

- Every path from root of AVTree gets unique hash value
- For example:
 - hash(type-book) = 5
 - hash(type-book-author) = 7
- Node in DHT responsible for hash value will keep reference to actual resource
- Query for (type-book) will get hashed to value 5 and sent to node responsible for storing hash value 5

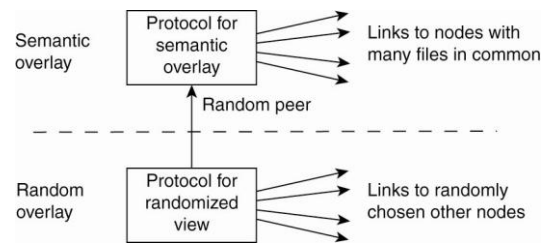
42

Semantic Overlay Networks

- When there is no organised attribute-based naming resolution scheme, nodes must discover for themselves where resources are located
- To make queries efficient, nodes can track nodes with similar resources
- Measuring similarity based on attributes is difficult => different nodes have different definitions of attributes
- Possibly ignore attributes and use file names
 - Similarity measured as number of files in common

43

Semantic Overlay Networks (2)



Example approach with two layers:

- Top layer maintains links between similar nodes
- Bottom layer is random overlay network

44

Summary

- Names are used to identify, access, discover and locate objects
- Mobile entities have to be supported by appropriate naming systems
- Structured names are organised into name spaces represented by a naming graph
 - Name resolution involves traversing the name graph
- Attribute-based naming is commonly used to support querying for services with particular attributes
- Reading: Chapter 5

45