



Adama Science and Technology University
School of Electrical Engineering & Computing
Department of Computer Science and Engineering

Introduction to Distributed Systems

Text Book:

S. Tanenbaum and Maarten Van Steen, Distributed Systems: Principles and Paradigms, 2nd Ed.

Compiled by: Sufian K.

E-mail: sufainkedir156@gmail.com

Chapter - 4

Naming

Introduction

- **Names** play an important role in all **computer Systems**.
 - Names are used **to share resources, to uniquely identify entities, to refer location and more.**
- An important issue with naming is that a name can be **resolved** to the entity it refers to.
- **Name resolution** allows a process to access the named entity. To resolve names, its necessary to implement **a naming system.**
- Naming system is maintains **name-address binding** (**name, address**)

Introduction

- In a DS, the implementation of a naming system is itself often distributed across multiple machines.
- Efficiency and scalability of naming system is important.
- Names are used to locate **mobile entities**.
- Removing names no more used is called garbage collection.

Naming Entities

■ Names, Identifiers, and Addresses

- **Name** in a distributed system is a string of bits or characters that is used to refer to **an entity**
- **Entity** is can be practically anything; e.g., resources (hosts, printers, disks, files, objects, processes, users, ...)
- Entities can be **operated** on; e.g., a resource such as **printer** offers an interface containing operations for printing a document and like.

Naming Entities

- To operate on an entity, it is necessary to access it through its **access point**, itself special kind of an entity.
e.g. to access printer
- Name of an access point is called **address** (**IP address** and **port number** used by transport layer) **of entity**.
- The address of an access point of an entity is also simply called **an address of that entity**.
- **Access point** special kind entity in a DS to access resources.

...Cont'd

- An entity can offer more than one access point.
 - E.g. a telephone can be viewed as an access point of a person, whereas the telephone number corresponds to an address. Person may have several telephone numbers.
- An **entity** may change its access points in the course of time.
 - E.g. When a mobile computer moves to another locations, its often assigned **a different IP address** than one it had before.
 - When person move to other country, it often change telephone numbers as well.

...Cont'd

□ Address

- is special kind of name.
- It refer **access point** of an entity.
- each entity is referred by at most one address.
- Separating the name of an entity and its address makes it easier and more flexible; such a name is called **location independent**.

...Cont'd

Identifier

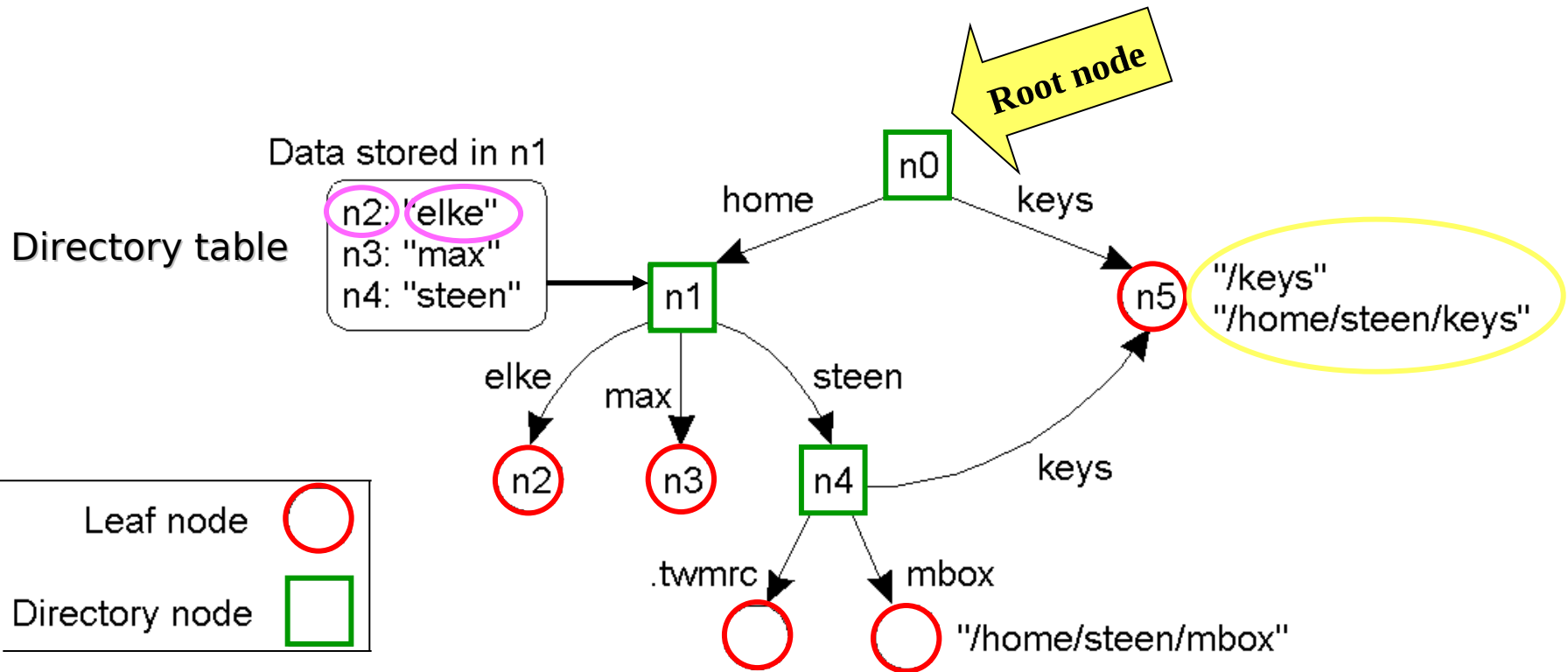
- A **true identifier** is a special type of name that **uniquely identifies** an entity and has the following properties:
 - refers to at **most one** entity
 - each entity is referred by at **most one** identifier
 - always refers to the same entity (**its never reused and always unique**)
 - allow to unambiguously refer to **an entity**
 - E.g. FTP Server Name (Entity) – URL

FTP Server Address - IP number:port number

Name Spaces and Name Resolution

- Names in a distributed system are organized into a **name space**.
- Name space is generally organized as **a labeled, directed graph** with **two** types of nodes
 - **Leaf node**: represents named entity and stores information (address or state of that entity)
 - **Directory node**: a special entity that has a number of **outgoing edges**, each labeled with **a name**.

...Cont'd



A general naming graph with a single root node.

...Cont'd

- If first node is **root** of naming graph, it is called an **absolute path name** (first node in path name is root); otherwise it is a **relative path name** (**first node can be any node**)
- Path Name is sequence of labels. $N: \langle \text{label1}, \dots, \text{labeln} \rangle$
- Instead of the **path name** $n0: \langle \text{home}, \text{steen}, \text{mbox} \rangle$, we often use its string representation `/home/steen/mbox`
- There may also be **several paths** leading to the same node, e.g., node $n5$ can be represented as `/keys` or `/home/steen/keys`

Name Resolution

- The process of looking up **a name** stored in node is called name resolution
- Given a path name, it should be possible to look up any information stored in the node referred to by that name.
- Finding address when the name is given (following path)
- Name resolution can take place only if **how** and **where to start**. This is known **closure mechanism**.

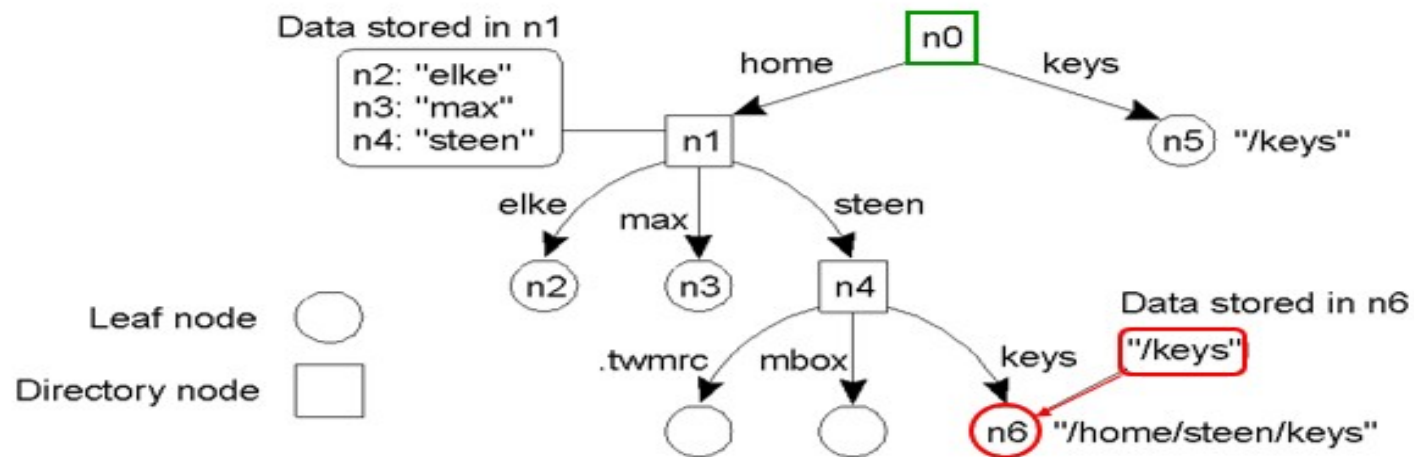
Name Resolution

Linking and Mounting

- **Linking**: giving another name for the same entity (**alias**)
- Two types of links (or two ways to implement alias):
 - **Hard link**: allow multiple **absolute path names** to refer to **same node** in a naming graph.
 - E.g., in the previous graph, there are two different path names for node n5:
 - ✓ */keys* and */home/steen/keys*
 - Either paths are used to access the node

...Cont'd

- **Symbolic link**: representing an entity by a **leaf node** and instead of storing **address or state of entity**, node stores an **absolute path name**.
- ✓ Use **information** stored on location to access the node.



The concept of a **symbolic link** explained in a naming graph within a **single name space**.

...Cont'd

- Name resolution can also be used to merge different **name spaces** in transparent way using the concept of **mount points**.
 - ✓ **Mounting and adding a new root node and making existing root nodes its children.**

1. Mounting

- To mount **a remote name space** in a distributed system requires at least the following information:
 - Name of an **access protocol** (for communication)
 - Name of **the server**
 - Name of the **mounting point** in the foreign name space

...Cont'd

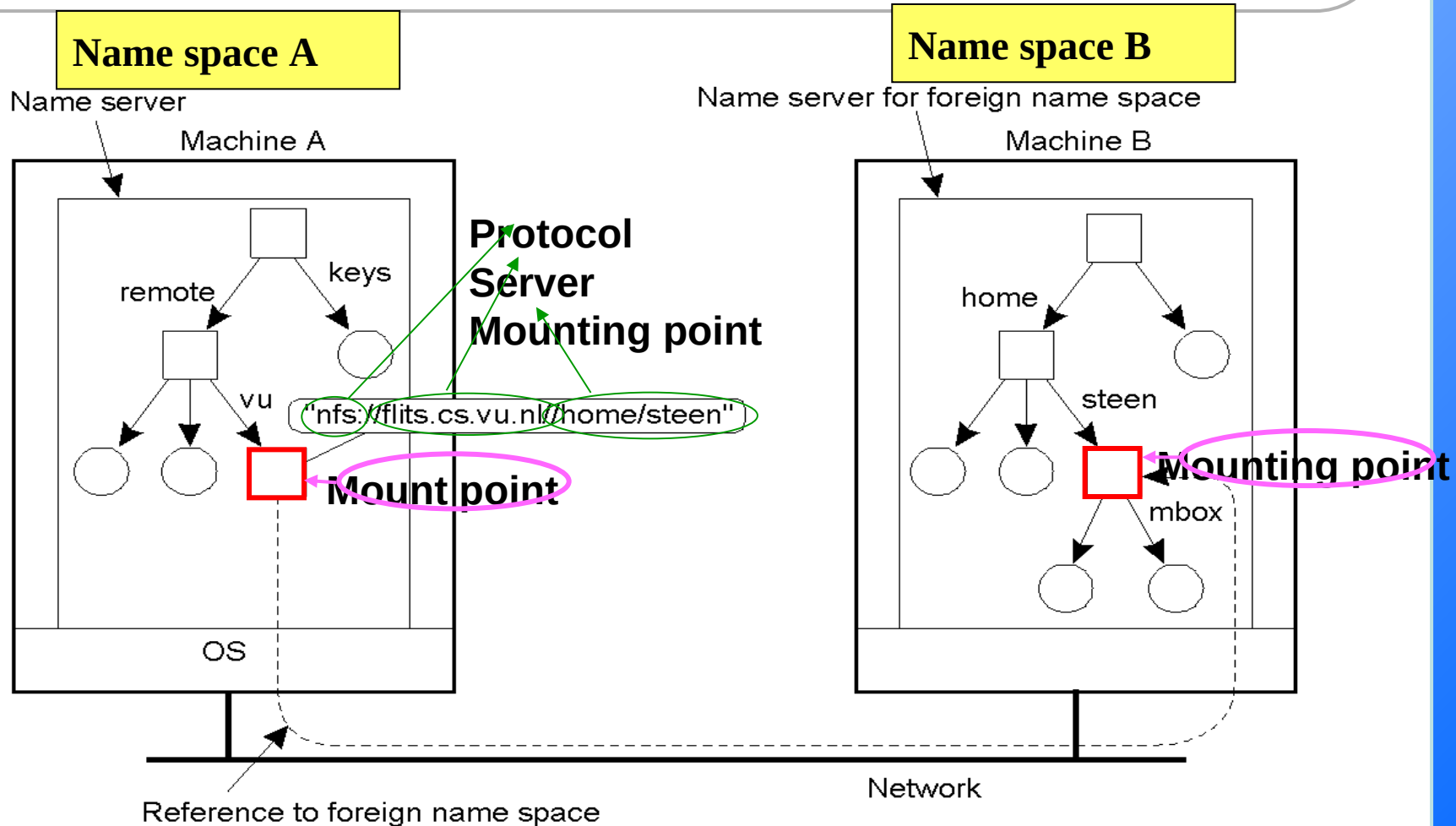
ftp://www.asu.edu.et/cs/cd

Protocol | Resolved using DNS | Resolved by Server



- Known server is used to resolve name (server will've protocol).
- Directory node **storing node identifier** is called **mount point**.
- Directory node in **foreign name space** is called **mounting point**

...Cont'd



Mounting remote name spaces through a specific process protocol (i.e. in NFS) to **merge** different name spaces.

...Cont'd

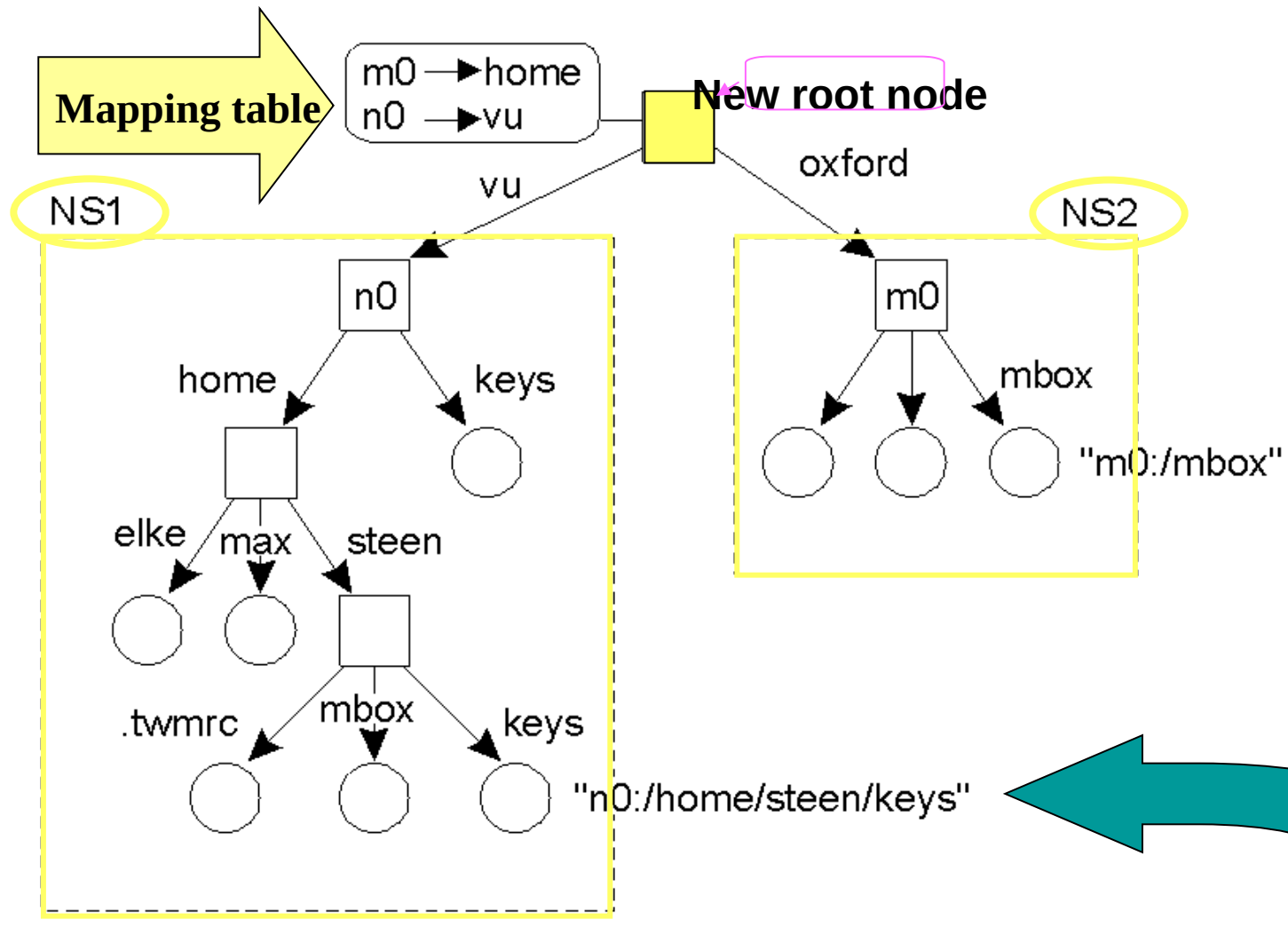
2. Add a new root node and make the existing root nodes its children.

- Method followed in GNS (Global Name Service)
- Problem: existing names need to be changed

E.g., **absolute path name** `/home/steen` has now changed to a **relative path name** and corresponds to the absolute path name `/vu/home/steen`

- Hence the system must expand `no:/home/steen` to `/vu/home/steen` without the awareness of users
- i.e. `n0:/home/steen = /vu/home/steen`

...Cont'd



Names in GNS **always include** the **ID** of the node from **where resolution should start**

The Implementation of a Name Space

- A name space forms the heart of a **naming service**.
- A naming service allows users and processes to **add, remove,** and **lookup** names.
- A naming service is implemented by **name servers**
- If a DS is restricted to a LAN, it's often feasible to implement a naming service by means of only **a single name server**,
 - Whereas for a large-scale distributed system the implementation of **a name space** is distributed over multiple **name servers**.

...Cont'd

- In large scale DSs, it is necessary to distribute **name service** over multiple **name servers**, usually organized **hierarchically**.
 - such name space has only single **root node** .
- **Name service** can be partitioned into logical layers
- The following **three layers** can be distinguished

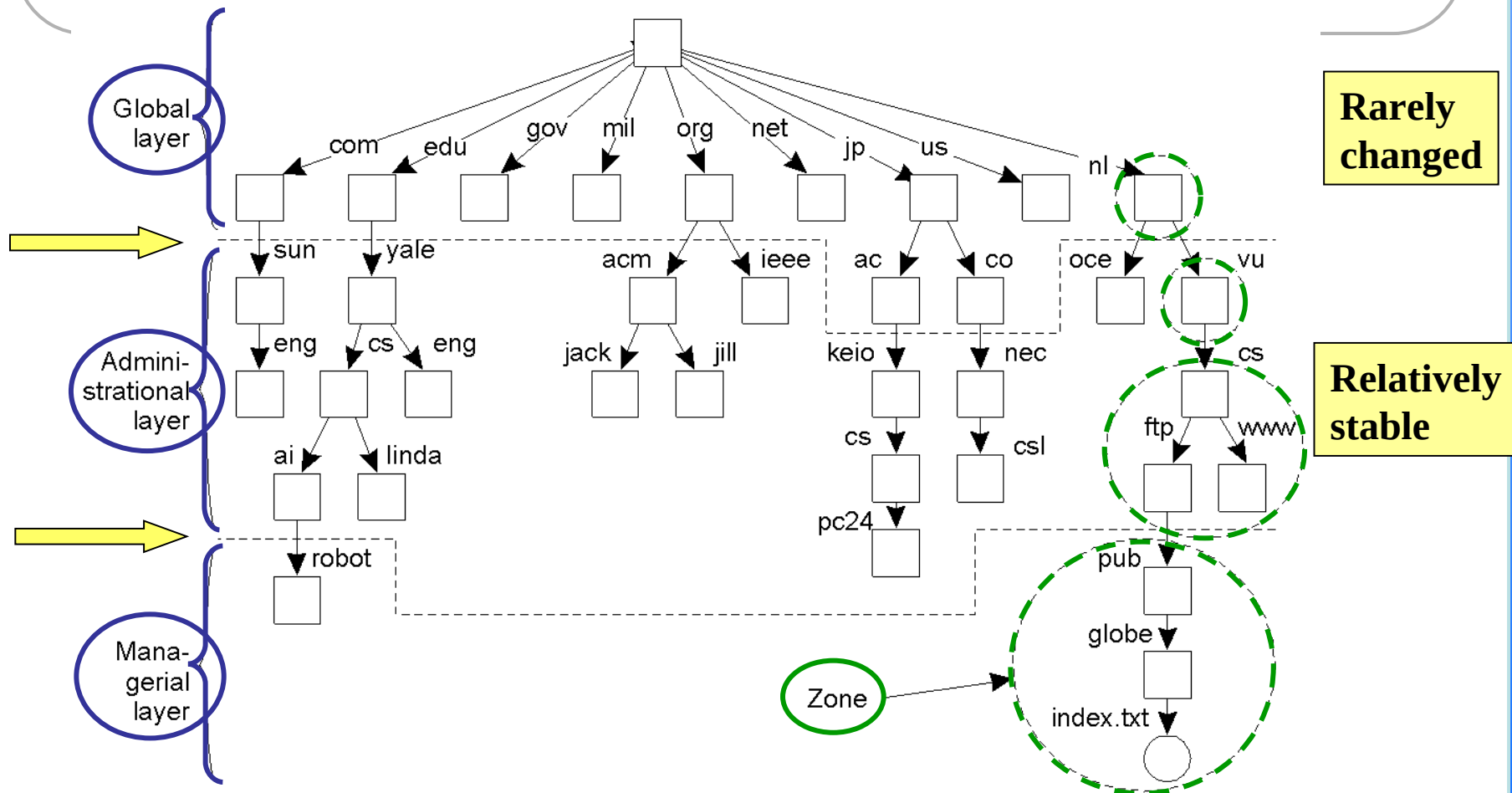
...Cont'd

- **Global layer** is formed by highest level nodes (root node and nodes close to it namely its **children**)
 - nodes on this layer are characterized by their **stability**, i.e., directory tables are **rarely changed**
 - they may **represent** organizations, groups of organizations, ..., where names are stored in the **name space**.

...Cont'd

- **Administrational layer** is formed by directory nodes that together are managed within **a single organization**.
- It represent groups of entities that belong to same organization or administrative unit, e.g., departments
- Node, although changes generally occur more frequently than to nodes in the global layer.
- **Managerial layer** is consist of nodes that may change regularly, e.g., nodes representing hosts of a LAN, shared files such as libraries or binaries, ...
 - nodes are managed not only by system administrators, but also by **end users**.

Name Space Distribution



- Example partitioning of the DNS name space, including Internet-accessible files, into **three layers**.

...Cont'd

- Name space is divided into non overlapping parts, called **zones** in DNS
- A zone is a part of name space that is implemented by a separate **name server**.
- Some requirements of servers at different layers
 - **performance** (responsiveness to lookups),
 - **availability** (failure rate), etc.
 - Performance may be enhanced by **client side caching** and **replication**;

Implementation of Name Resolution

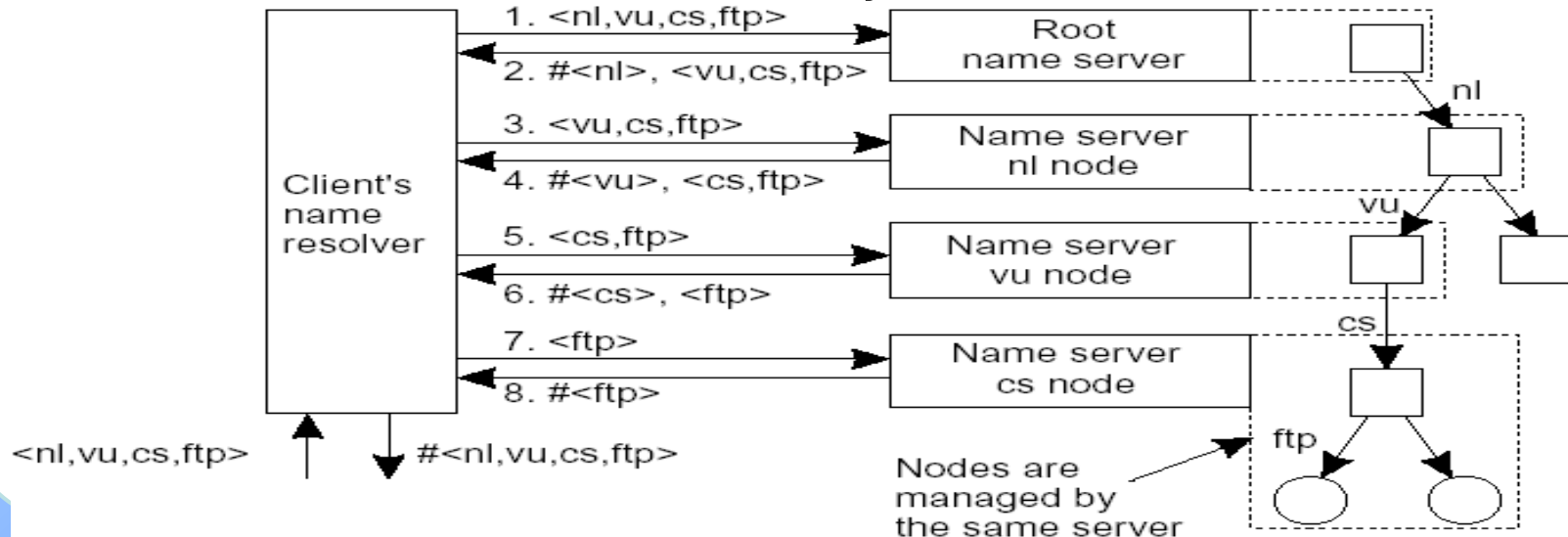
- Name resolution consists of finding the **address** when **name** is given.
 - Assume that name servers are not replicated and that no client-side caches are allowed.
- Each client has access to a **local name resolver**, responsible for ensuring that the name resolution process is carried out.
- E.g., assume the path name
root:<nl, vu, cs, ftp, pub, globe, index.txt> is to be resolved or using a URL notation, this **path name** would correspond to **ftp://ftp.cs.vu.nl/pub/globe/index.txt**

...Cont'd

Two ways of implementing name resolution

1. Iterative name resolution

- Name resolver **hands over complete name** to the **root name server**
- Root server will **resolve the name** as far as it can and return the result to the client; (ping www)
- Client calls the first level **name server**, then the second, ..., until it finds the **address** of the entity.



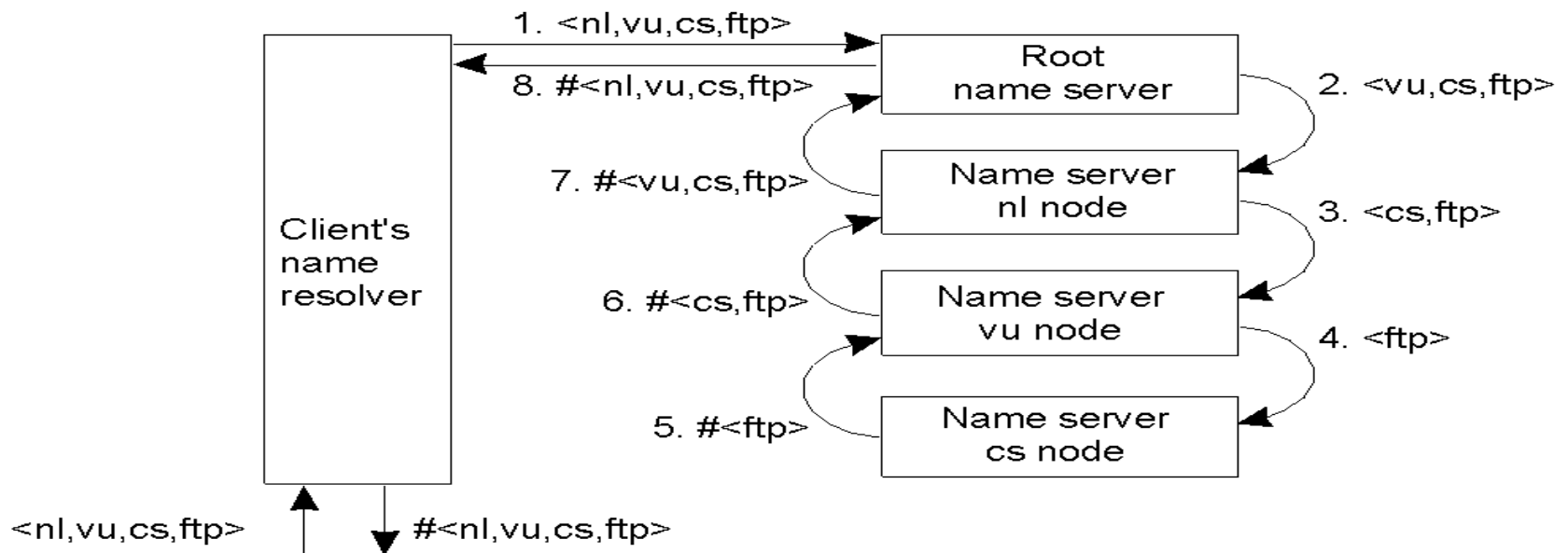
...Cont'd

- **Advantage**
 - Less burden on name sever
- **Disadvantage**
 - More communication cost

...Cont'd

2. Recursive name resolution

- Name resolver hands over the whole name to the root name server
- Root server will try to resolve the name and if it can't, it requests **the first level** name server to resolve it and to return **the address**.
- First level will do the same thing recursively



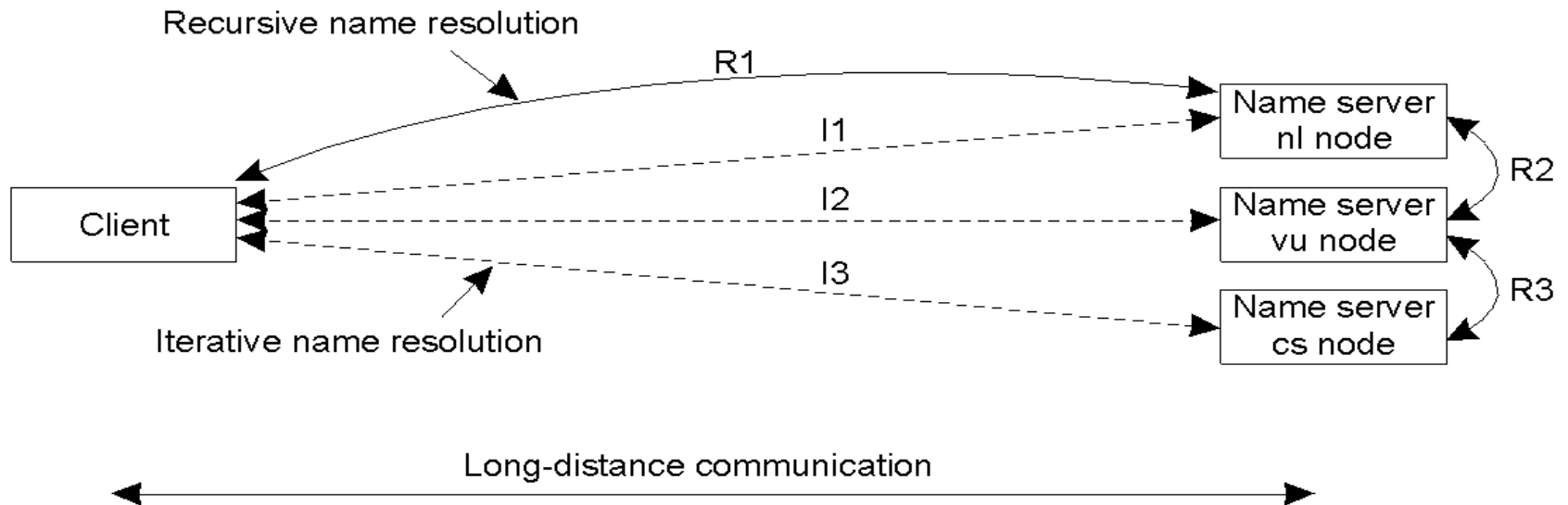
...Cont'd

❑ Advantages and drawbacks

- **Caching is more effective** with **recursive**; each name server gradually learns the address of each name server (handle lookup)
- **Communication costs may be reduced in recursive**
- **A higher performance demand on each name server;**
(name servers in global layer support only iterative)

...Cont'd

Method	Advantage(s)
Recursive	Less Communication cost; Caching is more effective
Iterative	Less performance demand on name servers



The comparison between **recursive** and **iterative** name resolution

The Domain Name System (DNS)

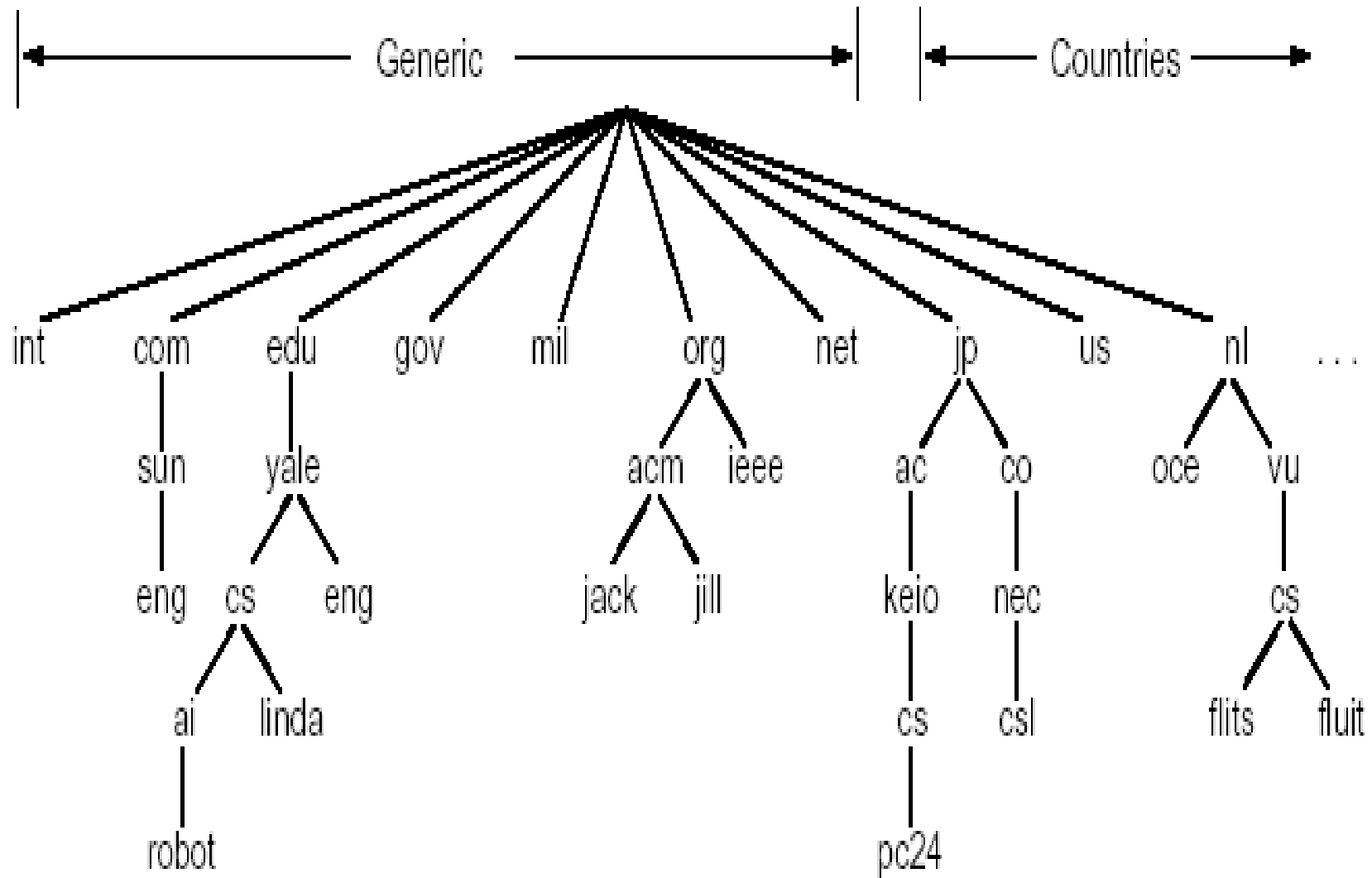
- Domain or domain names is the location of a website.
- Humans access information online through **domain names** like google.com, cnn.com etc.
- The DNS is directory (phonebook) of the Internet.
 - DNS translates **domain names and host names** to **IP addresses** so browsers can load Internet resources.
 - Connecting **web browsers with web server** (user website from web server through **DNS**)
 - It maps website to corresponding IP
 - Map IP address -> Naming || Naming IP address.



...Cont'd

- Internet DNS is One of the largest distributed naming services.
 - ✓ It is used for looking up **host addresses** and **mail servers**.
 - ✓ Hierarchical, defined in an **inverted tree structure** with the **root** at the top.
 - ✓ The tree can have only 128(0 to 127) levels

...Cont'd



Review Questions on Chapter Four

- Why naming?
- Discuss type of Naming System..?
- Define name, entity, address, access point, identifier.
- Explain the 2 methods of name resolution.
- Discuss the 2 types of linking.
- Explain the 2 ways of name resolution implementation.
- What are the advantages and drawbacks of iterative and recursive name resolution.
- Explain what a DNS is.