# Unit-5 Naming

# Content

Naming, Identifiers, and Address

Structured Naming

Attribute-based naming

**Naming:**

Names, Identifiers, and Addresses

Naming is about mapping between names, addresses, identifiers and the referred entities

Names (a bit-or character-string referring to an entity)

**e.g. John Smith or ftp-server**

Can be human-friendly(or not) and location dependent(or not)

"**A name in a distributed system is a string of bits or characters that is used to refer to an entity.**"

- An entity in a distributed system can be practically anything.
- Typical examples include resources such as hosts, printers, disks, and files.

To operate on an entity, it is necessary to access it, for which we need an access point.

An access point is yet another, but special, kind of entity in a distributed system.

The name of an access point is called an address.

The address of an access point of an entity is also simply called an address of that entity.

As a comparison, a telephone can be viewed as an access point of a person, whereas the telephone number corresponds to an address. Indeed, many people nowadays have several telephone numbers, each number corresponding to a point where they can be reached.

In a distributed system, a typical example of an access point is a host running a specific server, with its address formed by the combination of, for example, an IP address and port number (i.e., the server's transport level address).

An entity may change its access points in the course of time.

• For example: when a mobile computer moves to another location, it is often assigned a different IP address than the one it had before. Likewise, when a person moves to another city or country, it is often necessary to change telephone numbers as well.

# Identifiers (unique identifiers)

A (true) identifier is a name with the following properties

1. Each identifier refers to at most 1 entity and
2. Each entity is referred to by at most 1 identifier
3. An identifier always refers to the same entity (never reused)

**e.g. John Smith + social security number, or MAC address**

a name that uniquely identifies an entity

the identifier is unique and refers to only one entity

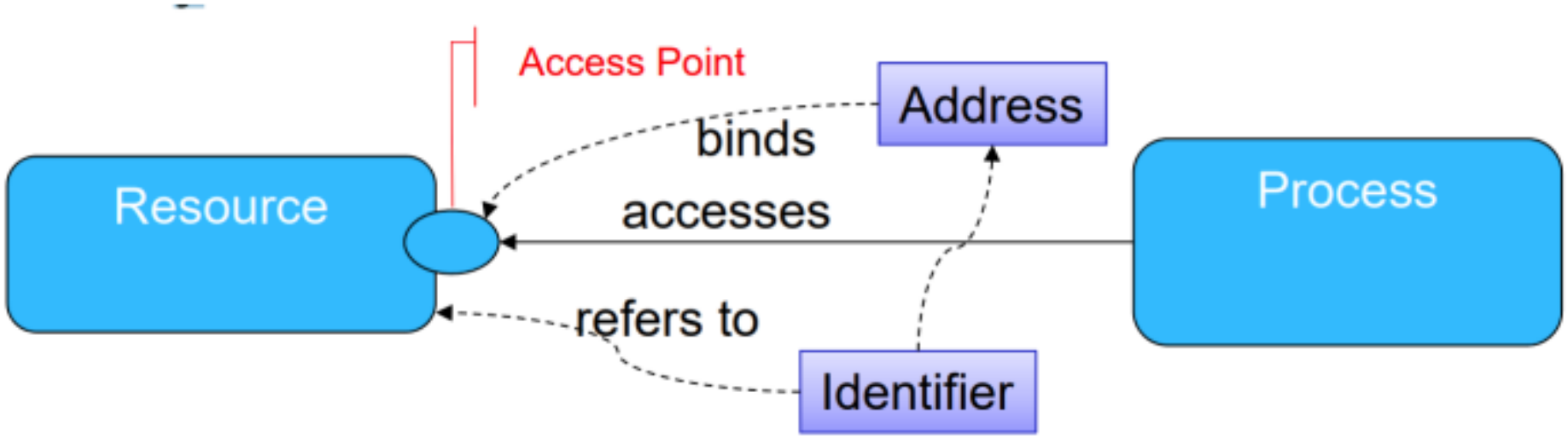
**Special type of (usually, computer readable) name with the following properties:**

**An id refers to at most one entity**

**Each entity is referred by at most one id**

**An id always refers to the same entity (never reused)**

**Address:** It is nothing but the name of an access point, the location of an entity.



**Classes of naming systems**

1. FLAT NAMING / UNSTRUCTURED NAMING

2. STRUCTURED NAMING

3. ATTRIBUTE-BASED NAMING

**FLAT NAMING / UNSTRUCTURED NAMING**

- Identifiers are convenient to uniquely represent entities.

- In many cases, identifiers are simply random bit strings. which we conveniently refer to as unstructured, or flat names.

- An important property of such a name is that it does not contain any information whatsoever on how to locate the access point of its associated entity.

**We will study these aspects:**

How flat names can be resolved ?

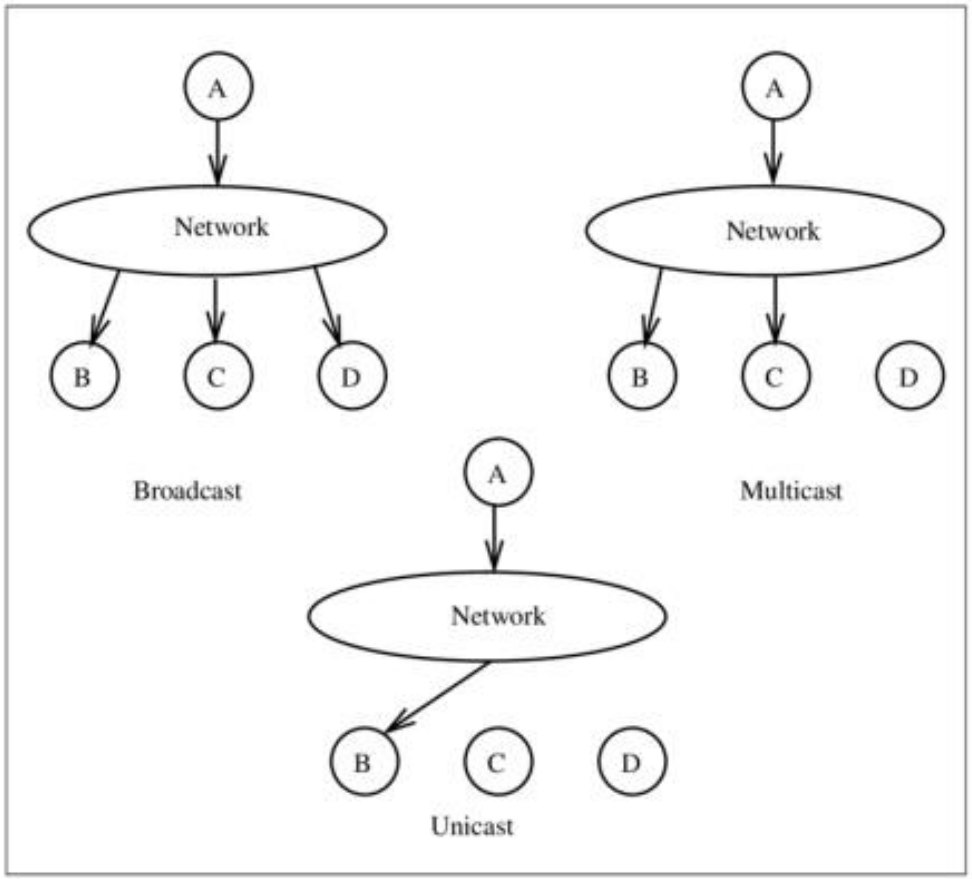
How we can locate an entity when given only its identifier ?

Simple Solution for locating an entity

- Broadcasting and Multicasting
- Forwarding Pointers

Both solutions are applicable only to local-area networks.

# Broadcasting and Multicasting




## BROADCAST VERSUS MULTICAST

| BROADCAST | MULTICAST |
| --- | --- |
| A method of transferring a message to all recipients simultaneously | A group communication where data transmission is addressed to a group of destination computers simultaneously |
| Packets are transmitted to all the connected devices in the network | Packets are transmitted to some of the devices in the network |
| There is no need for group management | Requires group management |
| Less secure | More secure |
| More traffic | Less traffic |
| Slower | Faster |

Consider a distributed system built on a computer network: that offers efficient broadcasting facilities.

Typically, such facilities are offered by local-area networks and LAN wireless in which all machines are connected to a single cable or equivalent.

Locating an entity in such an environment is simple: a message containing the identifier of the entity is broadcast to each machine and each machine is requested to check whether it has that entity.

Only the machines that can offer an access point for the entity send a reply message containing the address of that access point.

Example:

- a machine broadcasts a packet on the local network asking who is the owner of a given IP address.
- When the message arrives at a machine, the receiver checks whether it should listen to the requested IP address.
- If so, it sends a reply packet containing, for example, its Ethernet address.

**Problems with Broadcasting:**

Broadcasting becomes inefficient when the network grows.Not only is network bandwidth wasted by request messages, but, more seriously, too many hosts maybe interrupted by requests they cannot answer.

One possible solution is to switch to multicasting, by which only a restricted group of hosts receives the request.

For example, Ethernet networks support data-link level multicasting directly in hardware.

**Multicasting can also** be used to locate entities in point-to-point networks. For example, the Internet supports network-level multicasting by allowing hosts to join a specific multicast group.

- Such groups are identified by a multicast address.
- When a host sends a message to a multicast address, the network layer provides a best-effort service to deliver that message to all group members.

A multicast address can be used as a general location service for multiple entities.

For example, consider an organization where each employee has his or her own mobile computer.

When such a computer connects to the locally available network. it is dynamically assigned an IP address.

In addition, it joins a specific multicast group. When a process wants to locate computer A, it sends a "where is A?" request to the multicast group.

If A is connected, it responds with its current IP address

**Forwarding Pointers**

• The principle is simple: when an entity moves from A to B, it leaves behind in A a reference to its new location at B.

• **Advantages:**

• simplicity: as soon as an entity has been located, for example by using a traditional naming service, a client can look up the current address by following the chain of forwarding pointer

• **Drawbacks**

• a chain for a highly mobile entity can become so long that locating that entity is prohibitively expensive.

• the vulnerability to broken links

# STRUCTURED NAMING

- Flat names: • good for machines, but are generally not very convenient for humans to use.
- structured names: • composed from simple, human-readable names.

➢Names are commonly organized into what is called a name space.

**Namespace**

Names are commonly organized into what is called a name space.

Name spaces for structured names can be represented as a labeled, directed graph with two types of nodes.

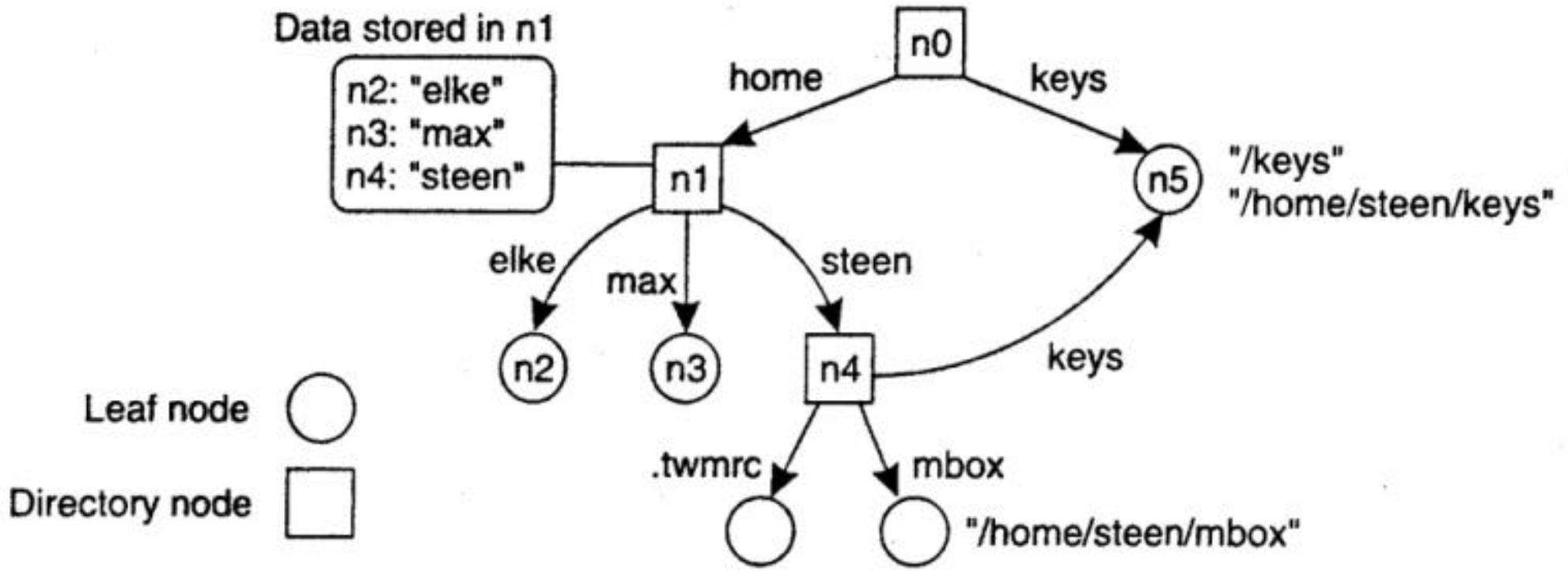



Figure 5-9. A general naming graph with a single root node.

**A leaf node** represents a named entity and has the property that it has no outgoing

**A leaf node generally** stores information on the entity it is representing-for example, its address-so that a client can access it. edges.

Alternatively, it can store the state of that entity, such as in the case of file systems 'in which a leaf node actually contains the complete file it is representing.

**A directory node** has a number of outgoing edges, each labeled with a name

**A directory node** stores a table in which an outgoing edge is represented as a pair (edge label, node identifier). Such a table is called a directory table.

**Name Resolution**

- Name spaces offer a convenient mechanism for storing and retrieving information about entities by means of names.
- More generally, given a path name, it should be possible to look up any information stored in the node referred to by that name.
- The process of looking up a name is called name resolution.

**Name Space distribution:**

Name space for world wide distributed system are usually organized hierarchically. To make it more convenient partition it into logical layer and they are as follows.

**Global Layer:** It is formed by highest-level nodes ie, that is the root node and other directory node logically close to the root called children.

-characterized by stability.

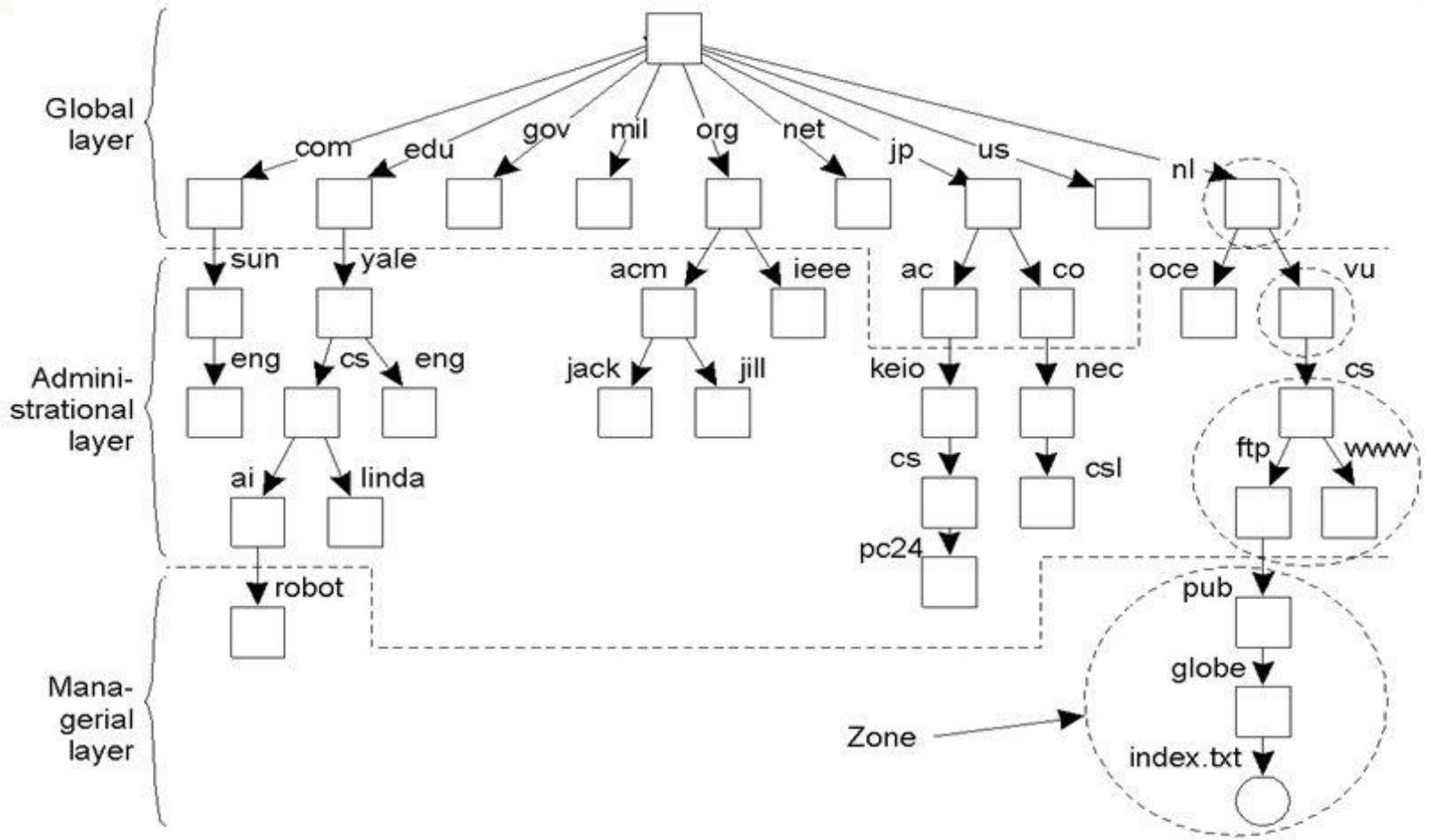
**Administrational layer:** This layer is formed by directory nodes that together managed with in a single organization.

- It represents group of entities that belong to the same organization or administrational unit.

-for example, there may be a directory node for each department in an organization.

**Managerial layer:** It consists of nodes that may typically change regularly. for example: nodes representing hosts in the network. the layer includes nodes representing shared files, user-defined directories and files.

# Name Space Distribution (1)




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

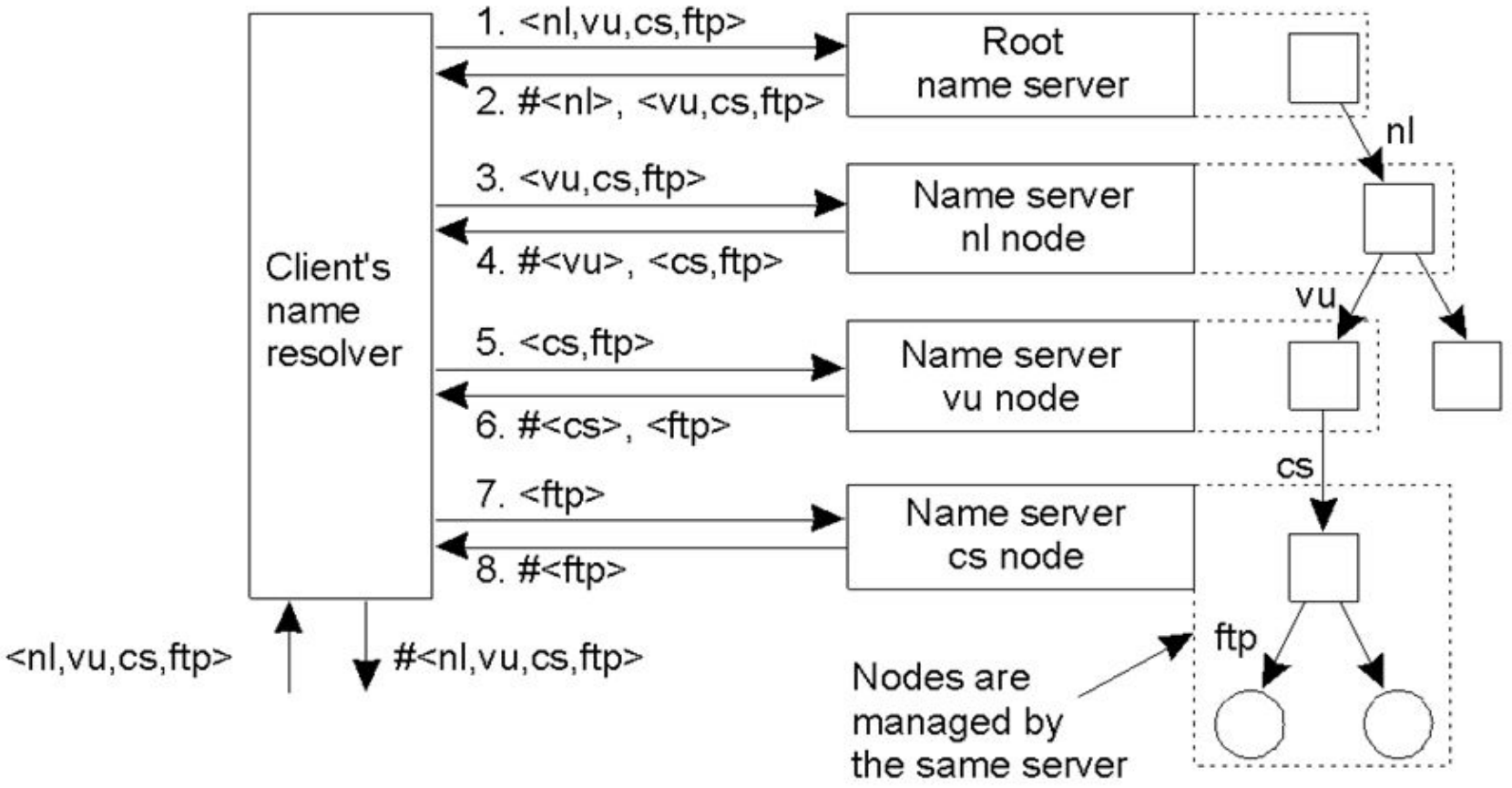
## Iterative Resolution

When a client sends an iterative request to a name server, the server responds back with either the answer to the request, or the name of another server that has the information or is closer to it. The original client must then iterate by sending a new request to this referred server, which again may either answer it or provide another server name. The process continues until the right server is found.

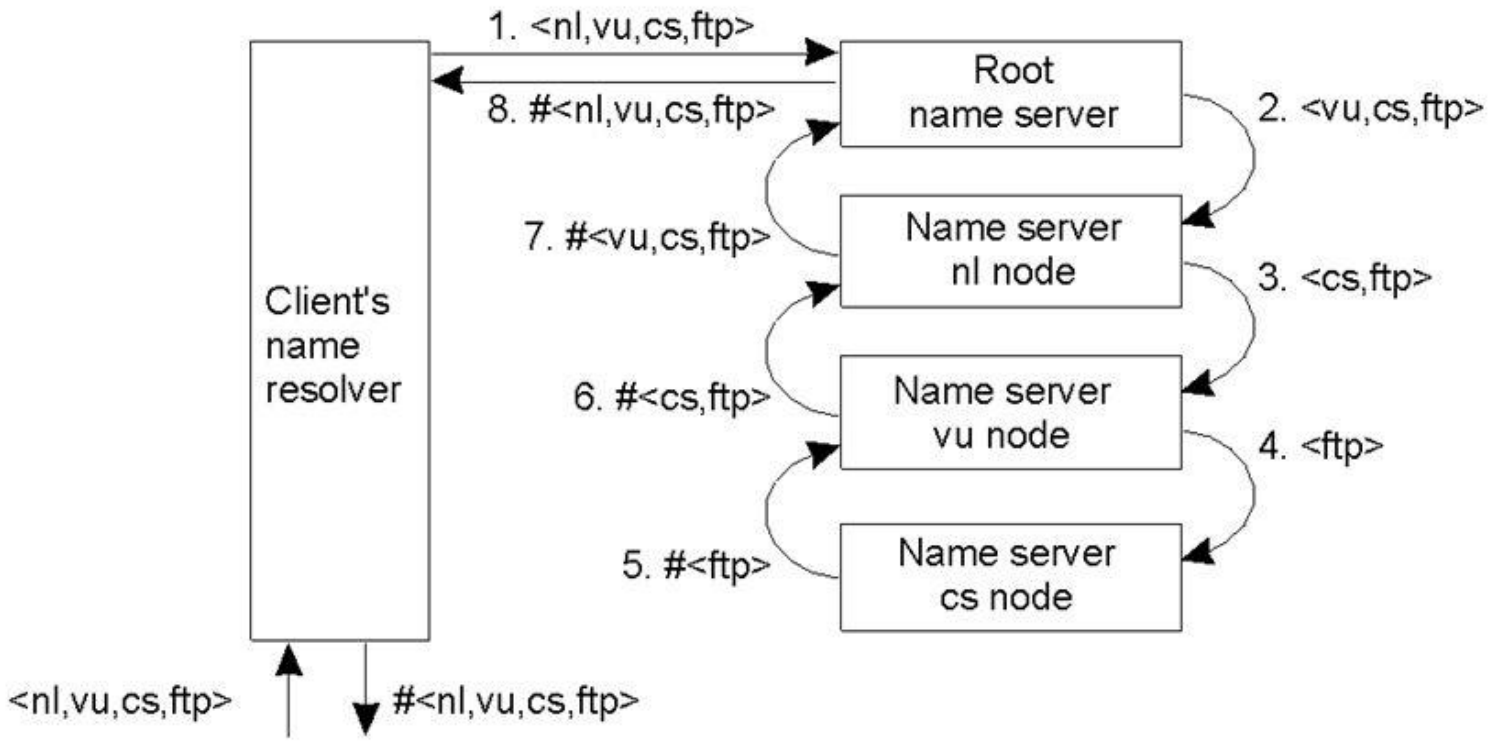
## Recursive Resolution

When  a client sends a recursive request to a name server, the server responds back with the answer if it has the information sought. If it does not, the server takes responsibility for finding the answer by becoming a client only sends one request, and eventually gets the information it wants.

# Iterative Name Resolution



# Recursive Name Resolution (2)



# DNS

An application layer protocol defines how the application processes running on different systems, pass the messages to each other.

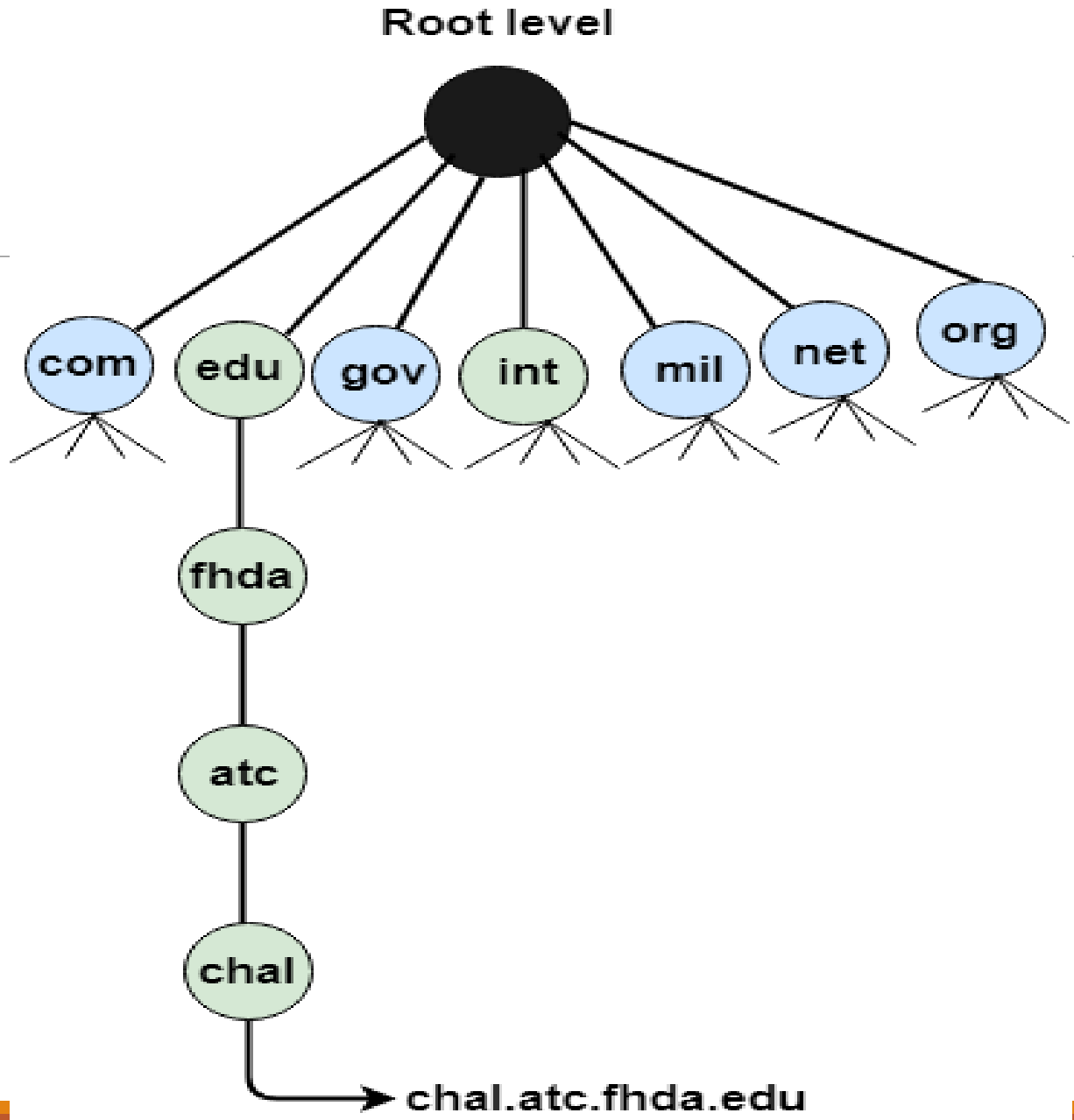
DNS stands for Domain Name System.

DNS is a directory service that provides a mapping between the name of a host on the network and its numerical address.

DNS is required for the functioning of the internet.

Each node in a tree has a domain name, and a full domain name is a sequence of symbols specified by dots.

DNS is a service that translates the domain name into IP addresses. This allows the users of networks to utilize user-friendly names when looking for other hosts instead of remembering the IP addresses.


Root level
com
edu
gov
int
mil
net
org
fhda
atc
chal
chal.atc.fhda.edu

**DNS record types**

The following are the five major DNS record types:

A record

AAAA record

CNAME record

Name server (NS) record

Mail exchange (MX) record

**1. A record**

The A record is the most important DNS record type. The "A" in A record stands for "address." An A record shows the IP address for a specific hostname or domain. For example, a DNS record lookup for the domain example.com returns the following result:

| Type | Domain Name | IP Address | TTL |
|---|---|---|---|
| A | example.com | 93.184.216.34<br>Verizon Business (AS15133) | 24 hrs |

**2. AAAA record**

AAAA record, just like A record, point to the IP address for a domain. However, this DNS record type is different in the sense that it points to IPV6 addresses.

IPV6 is an upgrade over IPV4 as it offers more IP addresses. As a result, IPV6 solves the issue of running out of unique IP addresses. An IPV6 address looks something like the following:

```
3001:0db7:3c5d:0024:0000:0000:1a2f:3c1b
```

**3. CNAME record**

CNAME—or, in full, "canonical name"—is a DNS record that points a domain name (an alias) to another domain. In a CNAME record, the alias doesn't point to an IP address. And the domain name that the alias points to is the canonical name. For example, the subdomain ng.example.com can point to example.com using CNAME. Here example.com points to the actual IP address using an A record.

**3. NS:** A name server is a server in the DNS that translates domain names into IP addresses. Name servers store and organize DNS records, each of which pairs a domain with one or more IP addresses. These servers act as the bridge between domain names, which we humans can remember, with IP addresses, which computers can process

**5. MX record**

A mail exchanger record specifies the mail server responsible for accepting email messages on behalf of a domain name

**In addition to the five DNS record types covered so far, here are some other record types:**

**SOA record:** SOA stands for "start of authority." It's an important DNS record type that stores admin information about a domain.

**PTR record:** A pointer (PTR) record provides a domain name for reverse lookup. It's the opposite of an A record as it provides the domain name linked to an IP address instead of the IP address for a domain.

**SRV record:** Using this DNS record type, it's possible to store the IP address and port for specific services.

**CERT record:** This record type stores public keys certificates.

## Attribute-Based Naming (Directory services)

Flat naming and structured naming are good upto some extent but if we get more data as input the attribute-based naming comes into play.

Mainly it is used for authentication, group and user management.

---

It is derived from OSI X500 directory services.

**Lightweight Directory Access Protocol (LDAP)** is an internet protocol works on TCP/IP, used to access information from directories. LDAP protocol is basically used to access an active directory.

**Features of LDAP:**

Functional model of LDAP is simpler due to this it omits duplicate feature.

It is easier to understand and implement.

It uses strings to represent data

**Directories:**
Directories are set of object with similar attributes, organised in a logical and hierarchical manner. For example, Telephonic Directories. It is a distributed database application used to manage attributes in a directory.

## Features of good naming system

### 1. Location transparency

Location transparency means that the name of an object should not reveal any hint as to the physical location of the object.

### 2. Location independency

For performance, reliability, availability and security reasons, distributed systems provide the facility of object migration that allows the movement and relocation of objects dynamically among the various nodes of a system.

### 3. Scalability

Distributed systems vary in size ranging from one with a few nodes to one with many nodes. Distributed systems are normally open systems, and their size changes dynamically.

### 4. Uniform naming convention

In many existing systems, different ways of naming objects called naming conventions, are used for naming different types of objects. For example, filenames typically differ from user names and process names.

### 5. Multiple user-defined names for the same object

For a shared object, it is desirable that different users of the object can use their own convenient names for accessing it. A naming system must provide the flexibility to assign multiple user-defined names to the same object.

### 6. Group naming

A naming system should allow many different objects to be identified by the same name. Such a facility is useful to support broadcast facility or to group objects for conferencing or other applications.

**7. Meaningful names**

A name can be simply any character string identifying some object. For users, meaningful names are preferred to lower level identifiers such as memory pointers, disk block numbers, or network addressess.

**8. Performance**

The most important performance measurement of a naming system is the amount of time needed to map an object's name to its attributes, such as its location.

**9. Fault tolerance**

A naming system should be capable of tolerating, to some extent, faults that occur due to the failure of a node or a communication link in a distributed system network. The naming system should continue functioning, perhaps in a degraded form, in the event of these failures.

**10. Replication transparency**

In a distributed system, replicas of an object are generally created to improve performance and reliability.

**11. Locating the nearest replica**

When a naming system supports the use of multiple copies of the same object, it is important that the object-locating mechanism of the naming system should always supply the location of the nearest replica of the desired object.