

ICS 362 Distributed Systems

Distributed Systems: Part 10

Lecturer: Toby Daniel

Naming

J
A
K
S
O
L
R
V
O
I
D
E
N
K
N
G
E
M
E
N
I
R
A
K
T
R
E
V
O
P

Naming

- What is naming?
- Why do we need to use names in Distributed Systems?

Why we need names...

- Any entity within a system needs a name – a string of bits/characters referring to an entity.
 - As entities can be operated upon, we need a way of identifying it.
- To operate on an entity, we need an access point – the access point is an address of the entity.
 - Entities may have several access points, and hence several addresses – in just the same way we might have more than one phone number.

Addresses



- The address of an entity may change over time;
 - A new IP address when you move your laptop.
- Addresses however rarely are the same as the name of the entity to which they refer.
 - Machines may be reassigned leading to inappropriate naming.
 - If a machine has more than one access point, which name should be assigned.
- Entity names which are independent of their addresses are easier and more flexible to use – these names are ‘location independent’.

Identifiers

- A different type of name is one which uniquely identifies an entity;
 - An identifier refers to at most 1 entity.
 - An entity is referred to by at most 1 identifier.
 - And identifier always refers to the same entity.
- Identifiers provide a way of unambiguously referring to an entity.
 - “John Smith” would not be an identifier.
 - A telephone would not be an identifier.

Human Friendly Names

- A final type of name is a human friendly name.
 - Names within the distributed system are actually strings of bits – perhaps a 32 bit string.
 - These are clearly difficult for humans to remember or use.
- Human friendly names are names that are translated for instance using a DNS.

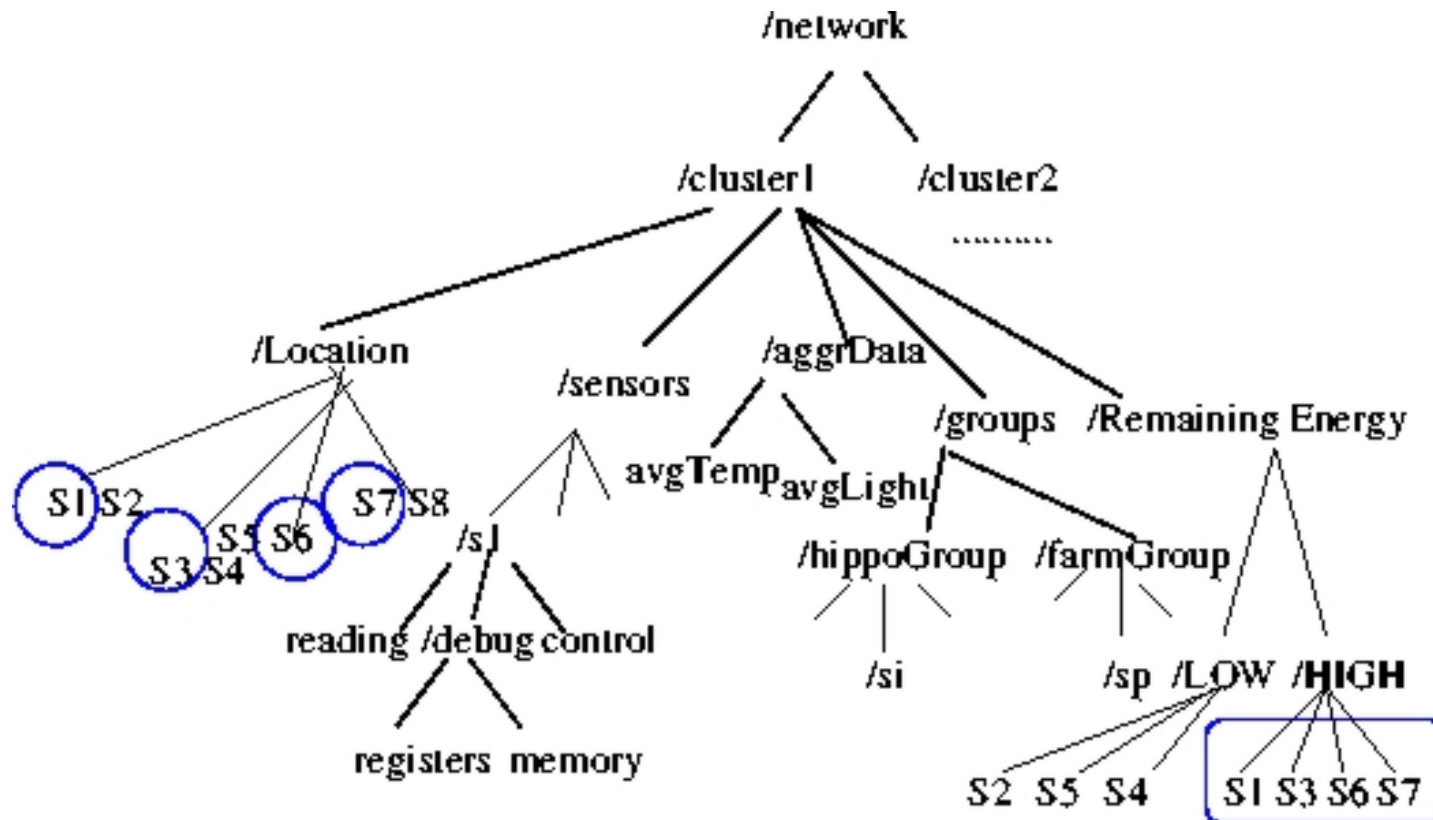
Example Names

What type of names are the following:

- Domain Names: payap.ac.th
- IP address: 64.233.187.99
- MAC address: 00-B0-D0-86-BB-F7
 - MAC addresses are sometimes called Burned In Addresses (BIA).

Namespaces

- Names are collected together into 'name spaces'.
 - A namespace could be viewed as a directed graph, comprised of leaf nodes (names) and directory nodes.



Paths

- Names can be found by following a path through the graph, for example;
 - N:<network, *cluster1*, *location*, S4>
 - or more commonly /network/cluster1/location/S4
- If the path begins at the root node, it is an absolute path name, or if it starts from somewhere else it is a relative path name.
- **Global names** are names which refer to the same entity wherever it is used.
- **Local names** are names which depend on where the name is used.

Name Resolution

- Name resolution simply refers to finding the node referred to, by following the path name.
 - Assuming we know where ‘/location’ is, we can then follow it to ‘/S4’ etc.
- Knowing where to start following a path is known as a ‘closure mechanism’.
 - Given 069176770, this number sequence means nothing unless you know it is a phone number and you can start to follow it using a telephone. (The telephone system deals with most of the resolution.)

Examples

Resource	Name	Identifiers
- File	Pathname	File within the file system
- Process	Process ID	Process on the computer
- Port	Port number	IP port on a given computer

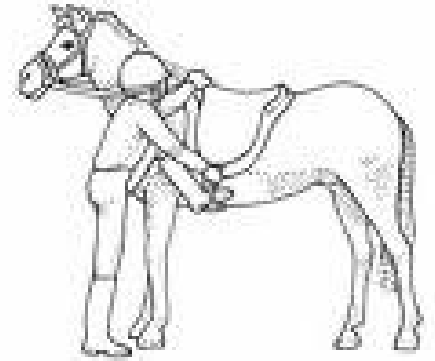
Aliases



Aliases

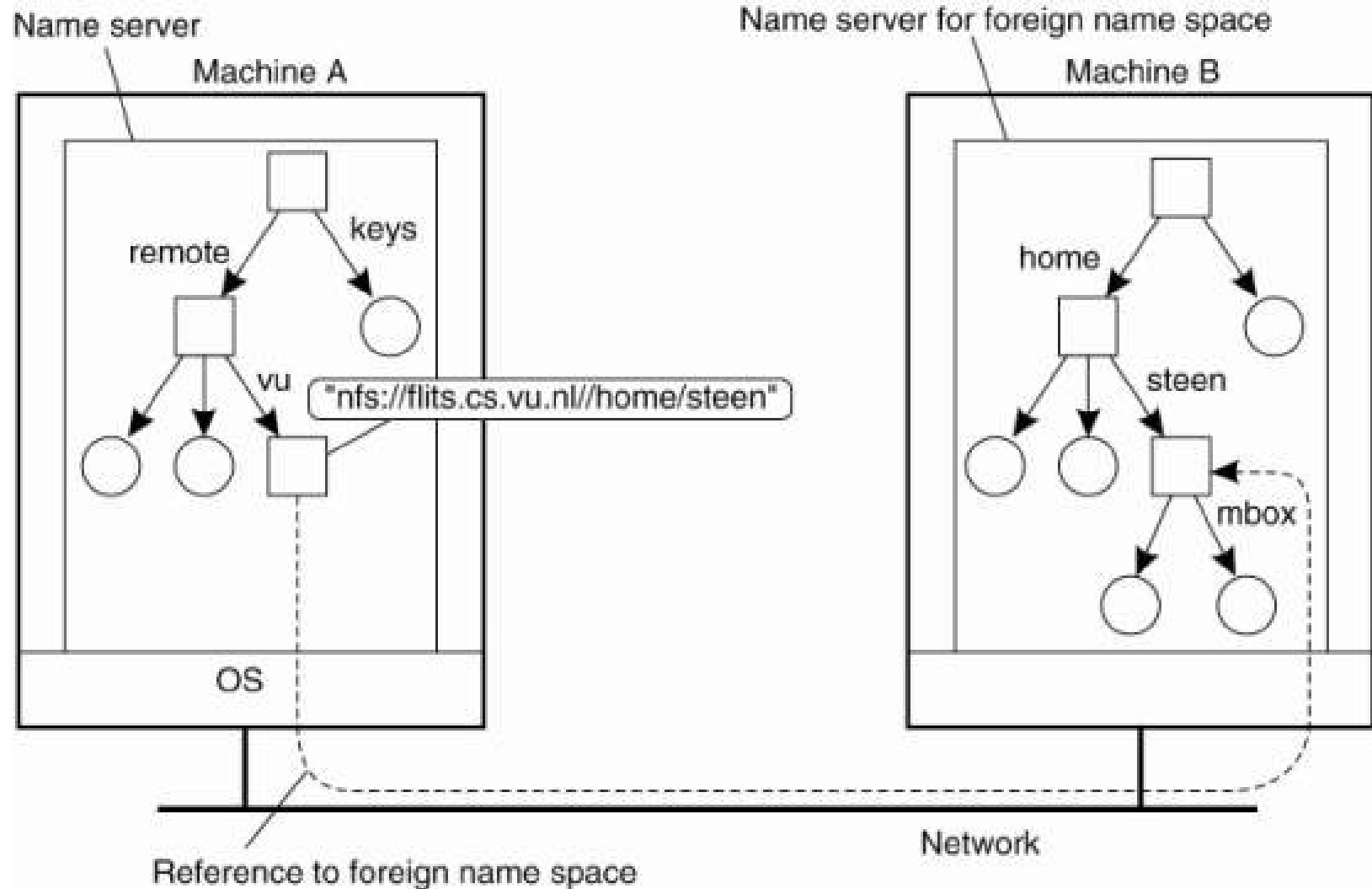
- An alias is an alternative name for the same entity
- What is the point of using Aliases?

Mounting



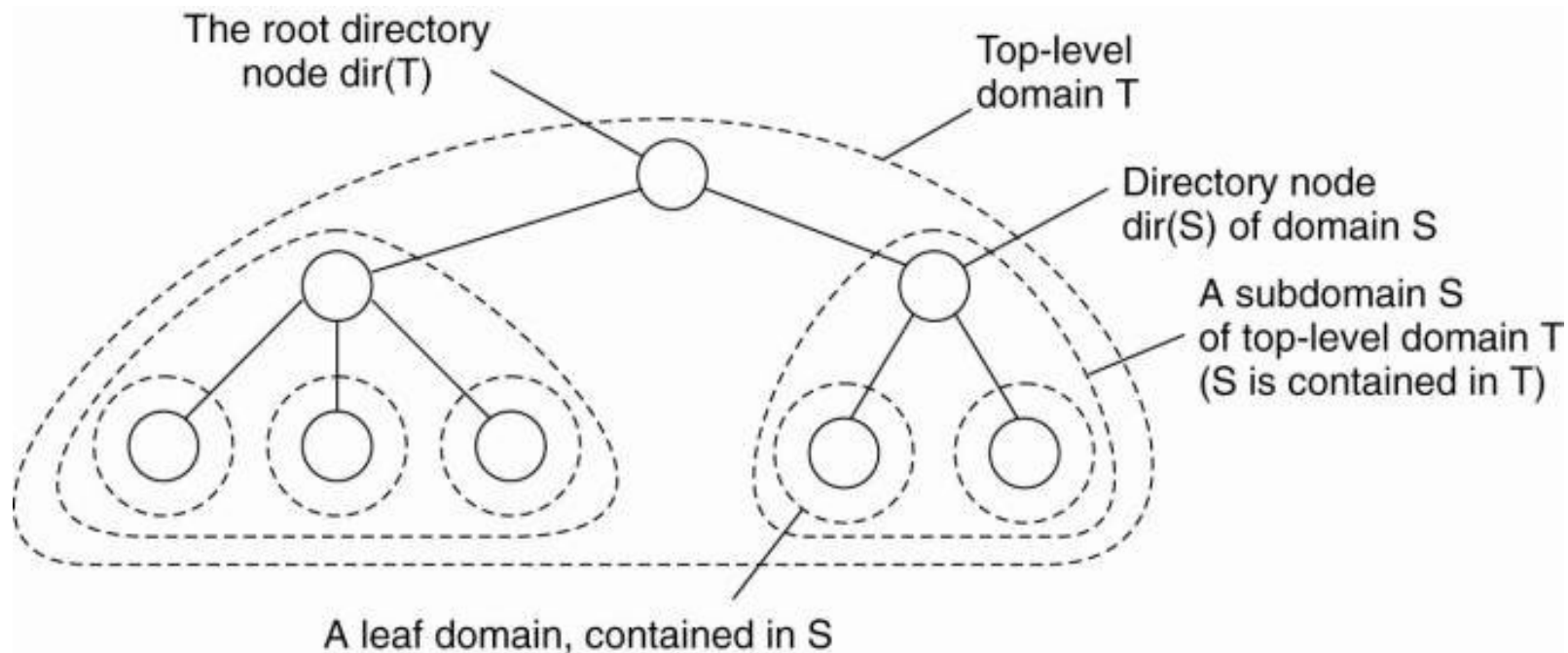
- We can mount to a 2nd namespace (foreign) on another machine if we have;
 - the name of an access protocol.
 - the name of the server.
 - the name of the mounting point in the foreign namespace.

Foreign aliases



Alternative to foreign aliases

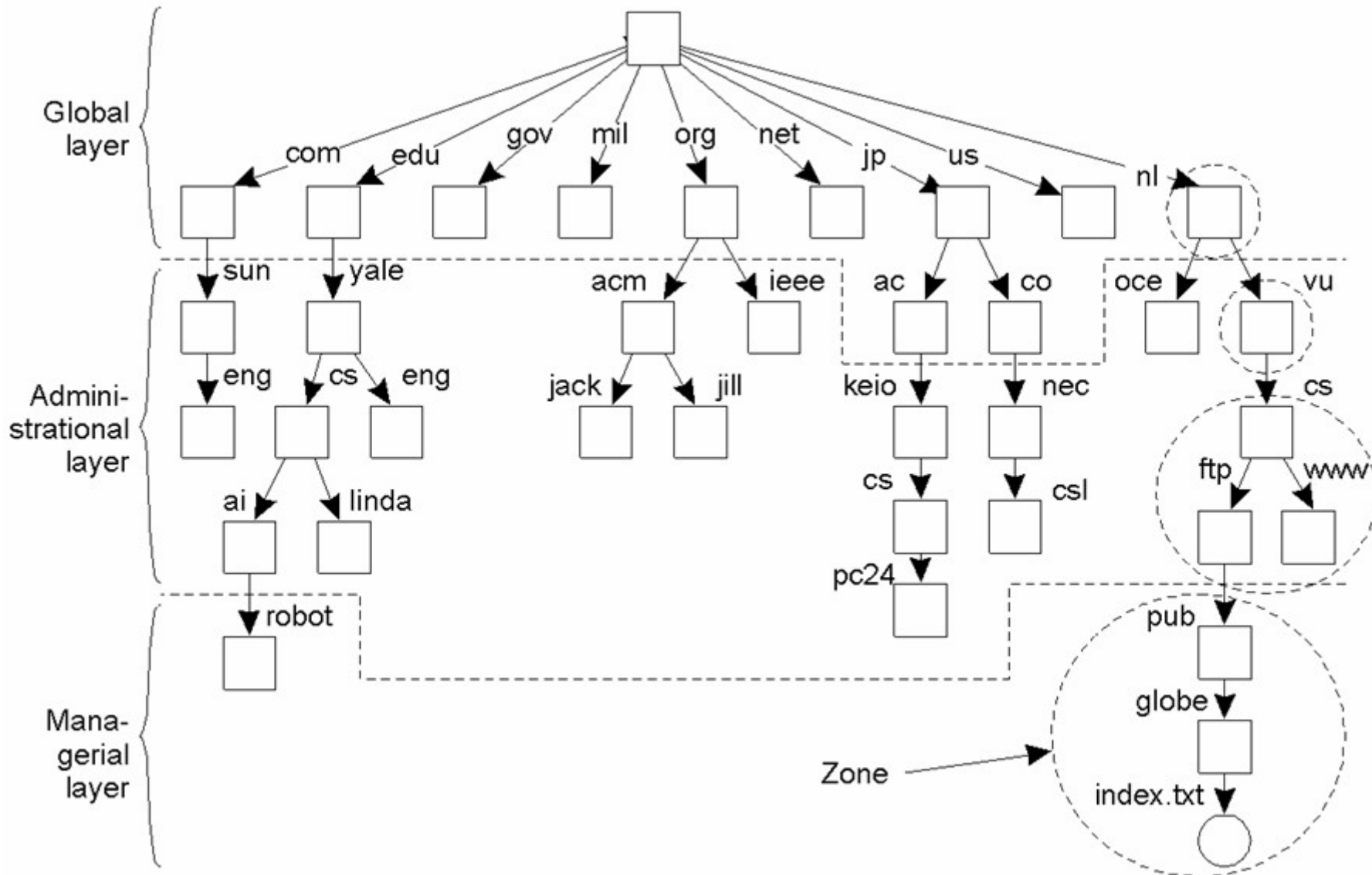
- An alternative would be to use a global naming service, where 2 namespaces can be merged under a new root node. However this can lead to issues with absolute path names.



Namespace implementation

- Namespaces for worldwide distributed systems are normally organised hierarchically, and the hierarchy can be broken into layers;
 - Global Layer
 - Highest level nodes, the root and its children.
 - Administrative Layer
 - Directory nodes managed within an organisation.
 - Managerial Layer
 - Nodes which may change regularly.

DNS Layers



Performance and Availability

- Breaking the namespace into layers has effects on the required performance and availability of the servers used to look up addresses.
 - Servers in the global layer need to have high availability, otherwise large parts of the system will become unavailable, however as their data changes rarely, it can be cached locally by clients requiring perhaps less server performance.

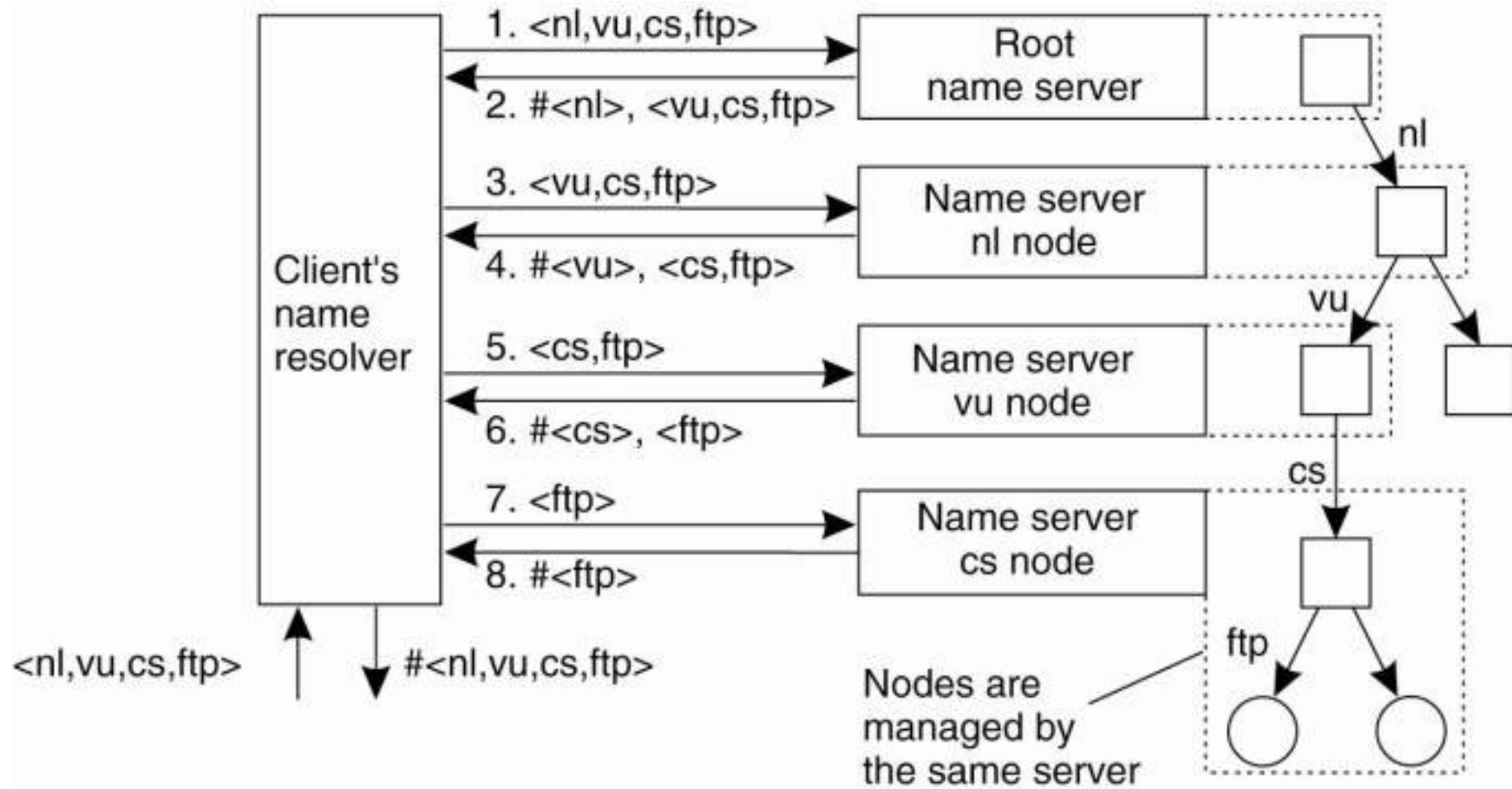
Name Resolution

- There are 2 key approaches to resolving a name:
 - Iterative
 - recursive

Iterative Resolution

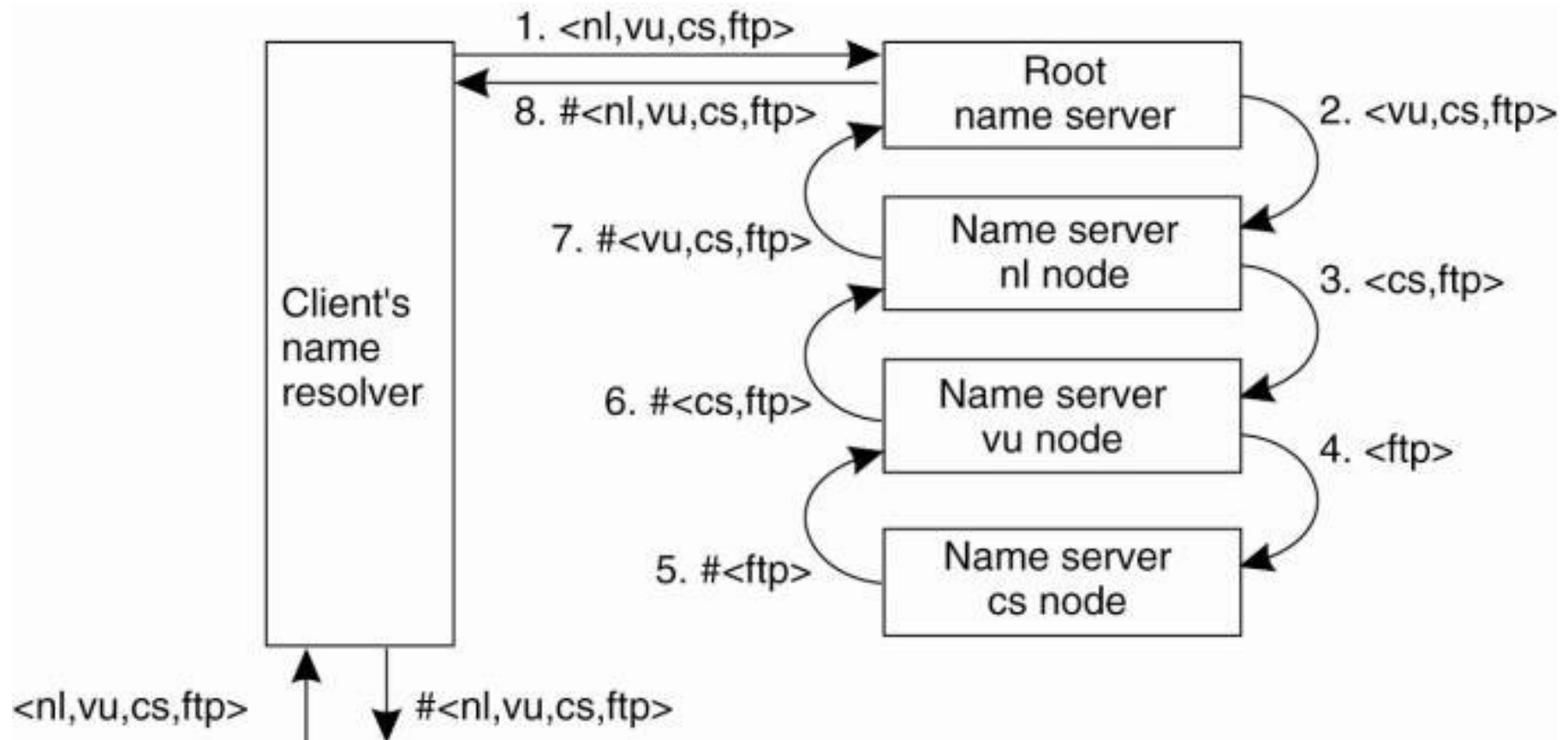
- In iterative resolution we start by requesting the location of the 2nd level node from the root server, we then contact the 2nd level node to request the third level node etc.
- Example- payap.ac.th
 - Start with root server to find .th
 - Request .ac from .th
 - Request payap from .ac

Iterative Resolution



Recursive Resolution

- In recursive resolution, we request the name resolution from the root server, which in turn contacts the next node.



Iterative vs Recursive

Which is best?



Scalability Issues

Size scalability:

- We need to ensure that servers can handle a large number of requests per time unit - high-level servers are in big trouble.

Solution:

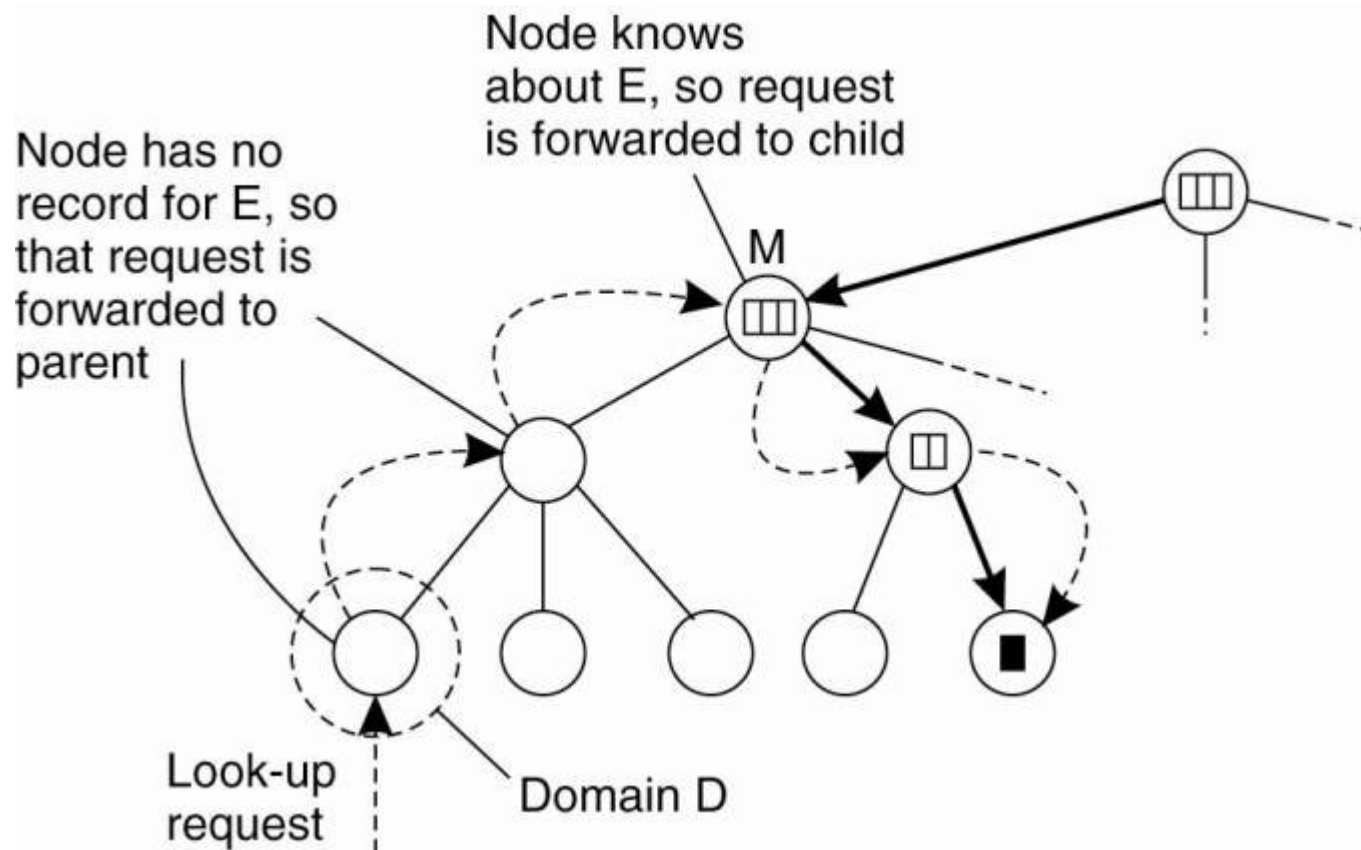
- Assume (at least at global and administrative level) that content of nodes hardly ever changes.
- In that case, we can apply extensive replication by mapping nodes to multiple servers, and start name resolution at the nearest server.

Scalability Issues

- Observation:
 - An important attribute of many nodes is the address where the represented entity can be contacted. Replicating nodes makes large-scale traditional name servers unsuitable for locating mobile entities.
- Geographical scalability:
 - We need to ensure that the name resolution process scales across large geographical distances.

Name Lookup

- Looking up a location in a hierarchically organized location service.



Next Lecture

Naming... what if resources move?

