Naming

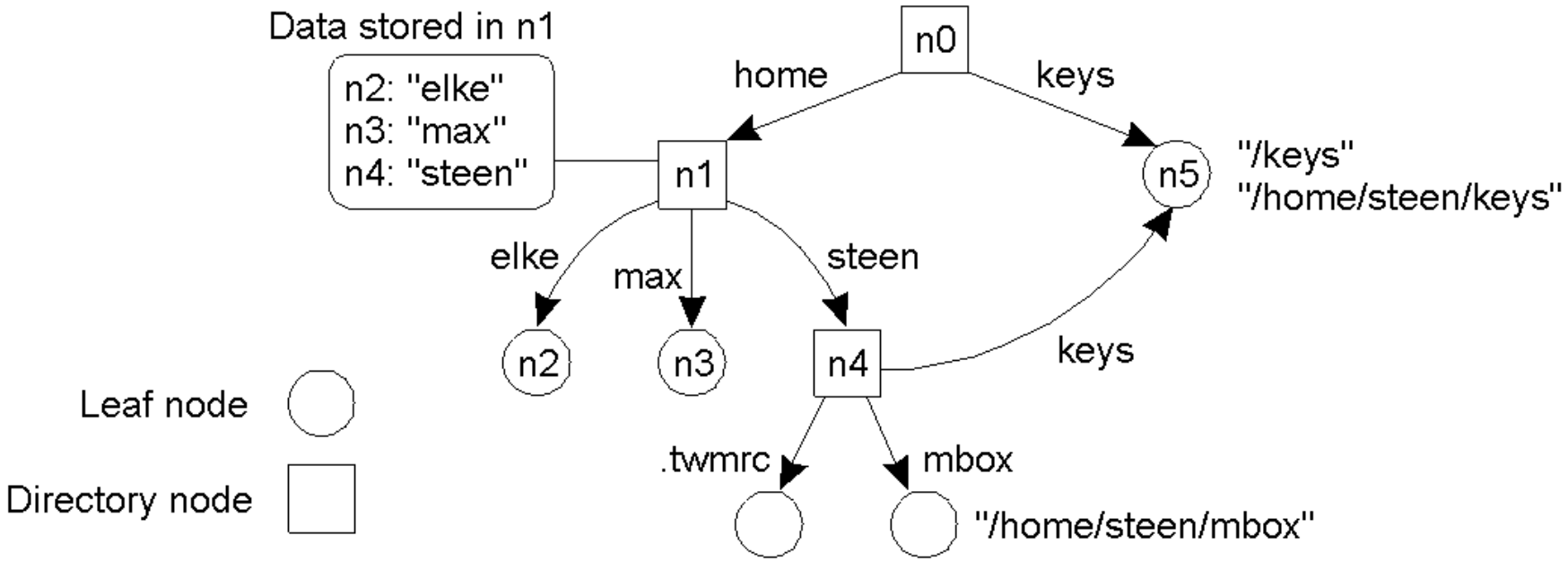
Chapter 4: Distributed Systems, Tanenbaum

D Hamed Vahdat-Nejad, University of Birjand

- Name:
  - String of characters
  - Used to refer to an entity
- Entity:
  - Resources: computers, printers, disks, files
  - Others: processes, users, emails, newsgroups, webpages, messages, network connections, etc
- Access point:
  - To operate on an entity
  - Name of an access point is called address
  - An entity may have several Access Points

Name Spaces

- Names in a distributed system are organized into name space
- Name space can be represented in a labeled directed graph
  - Leaf node: named entity
- A general naming graph with a single root node.



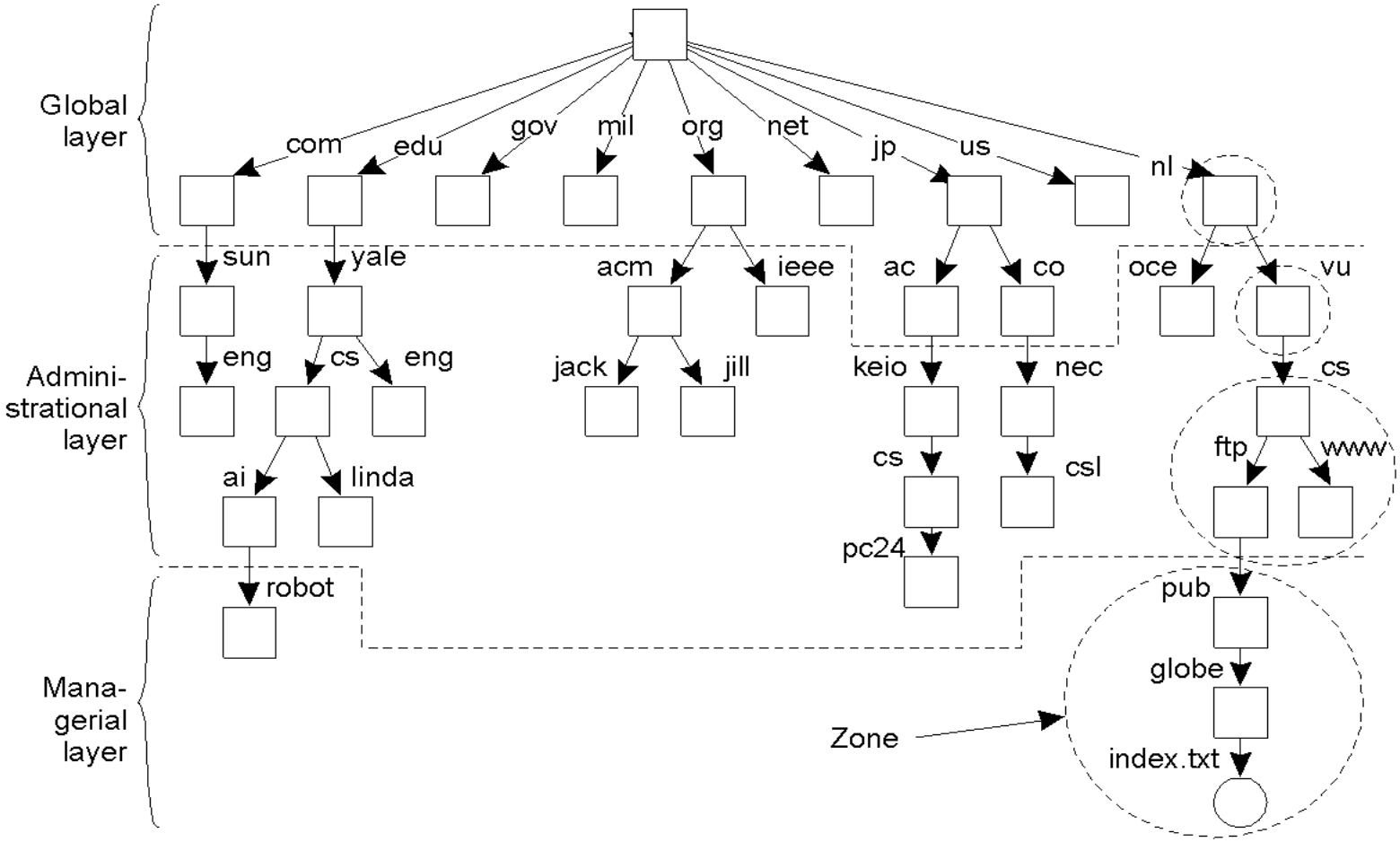
Name space implementation

- Naming service:
  - A service that allows users and processes to assign, remove and lookup names.
  - Implemented by name servers
  - Implemented centrally for a LAN
  - Distributed over multiple name servers in a large-scale distributed system

Name space distribution for a large scale distributed system

- Name space is organized hierarchically
- There is one root node.
- There are 3 logical layers:
  - Global layer
    - Formed by highest-level nodes: Root node and directory nodes close to the root that are its children
    - Nodes are stable: Directory tables are rarely changed.
  - Administrational layer
    - Formed by directory nodes that together are managed within a single organization
    - belong to the same administrational unit (organization): A directory node for each department in an organization
    - Nodes are relatively stable
  - Managerial layer
    - Consists of nodes that may change regularly: Nodes in a LAN-
    - Nodes may be user directories or files

Name Space Distribution (1)



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

Name Space Distribution (2)

- Zone:
  - Name space is divided into non-overlapping parts
  - Zone is implemented by a separate name server

Name Space Distribution (3)

Item	Global	Administrational	Managerial
Geographical scale of network	Worldwide	Organization	Department
Total number of nodes	Few	Many	Vast numbers
Responsiveness to lookups	Seconds	Milliseconds	Immediate
Update propagation	Lazy	Immediate	Immediate
Number of replicas	Many	None or few	None
Is client-side caching applied?	Yes	Yes	Sometimes

- A comparison between name servers for implementing nodes from a large-scale name space partitioned into a global layer, as an administrational layer, and a managerial layer.

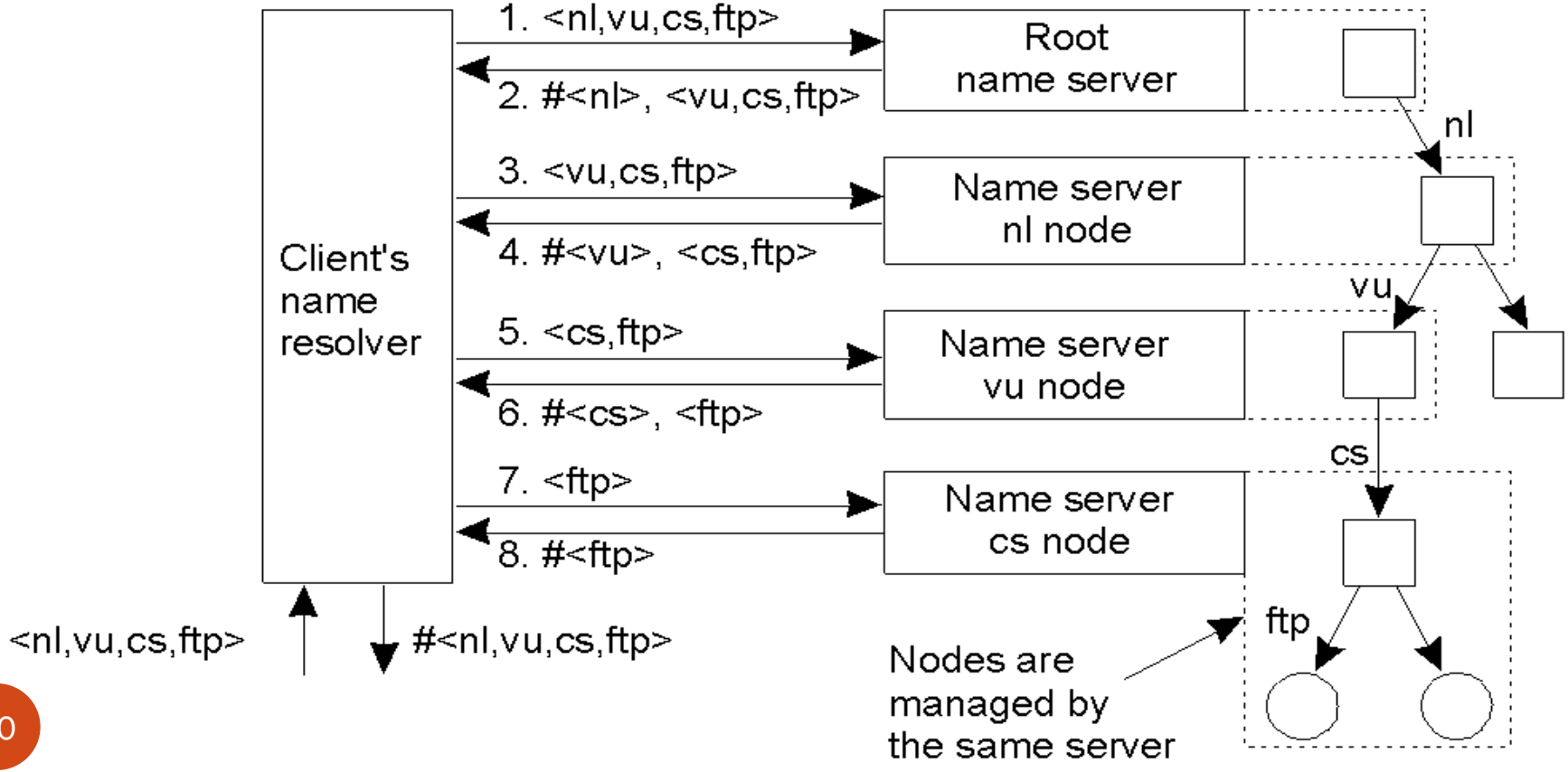
Name resolution

- Name resolution is the process of looking up a name
- Each client has access to a local name resolver
- There are 2 approaches to implement name resolution
  - Iterative: Client's name resolver iteratively contacts different name servers in a top-down manner
  - Recursive: A name server passes the result to the next name server it finds.

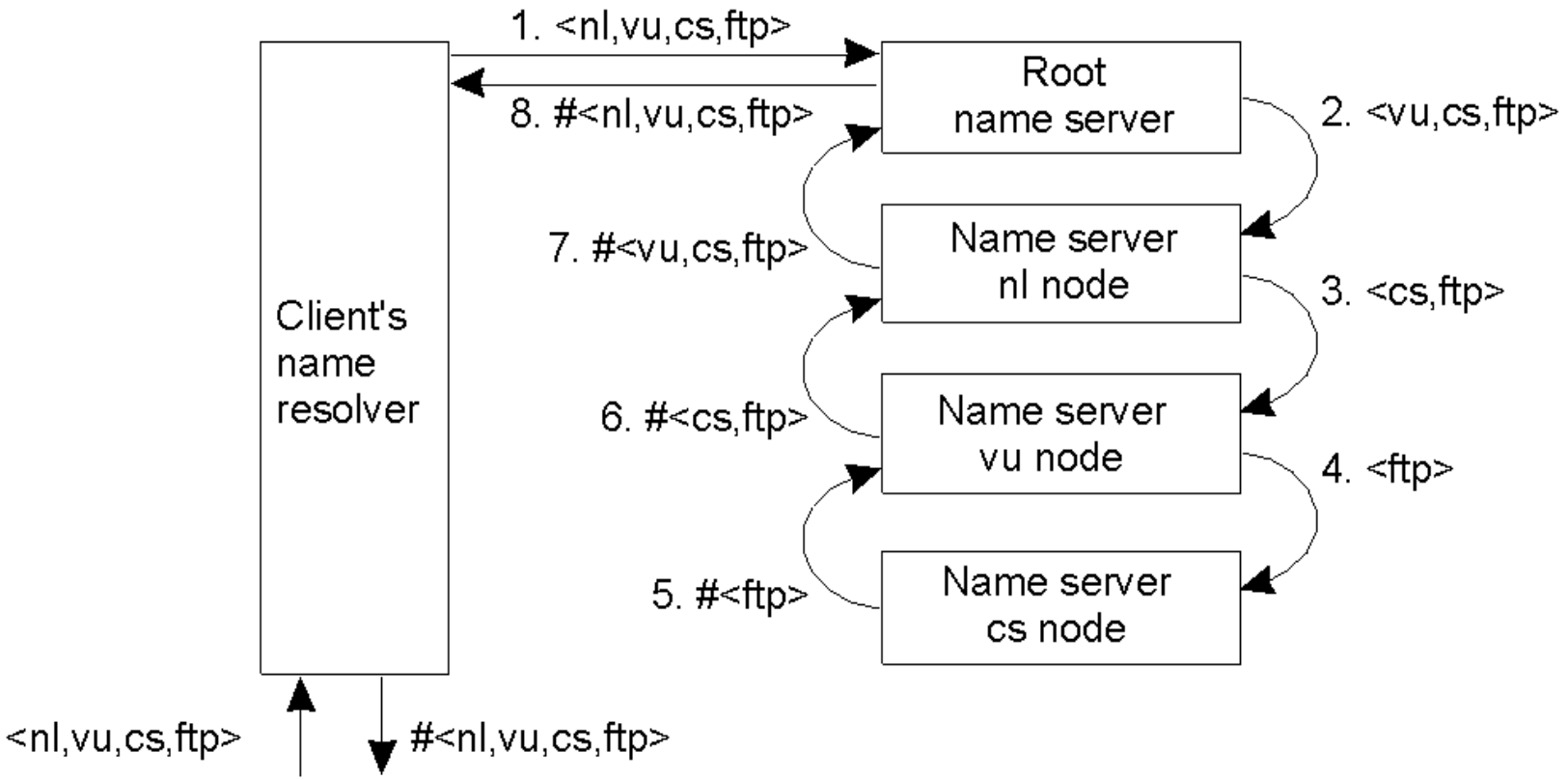
Implementation of Name Resolution (1)- Iterative

Root:<nl,vu,cs,ftp,pub,globe,index.txt>

ftp://ftp.cs.vu.nl/pub/globe/index.txt



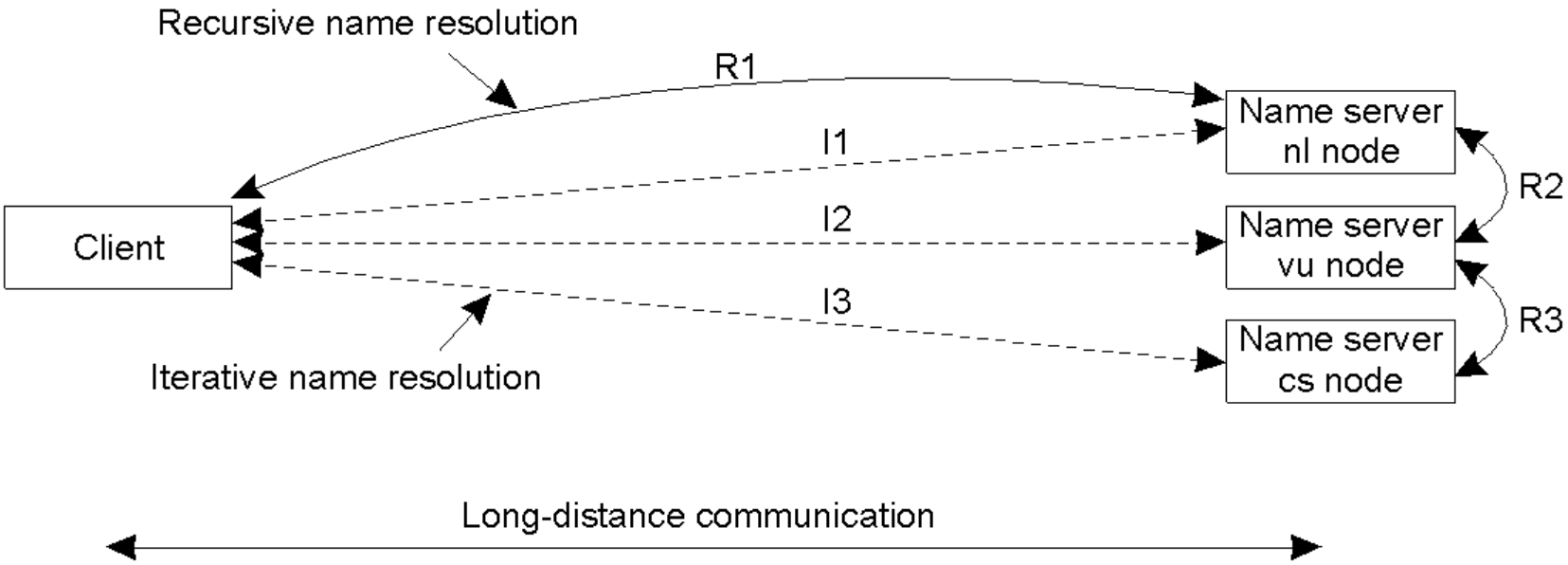
Implementation of Name Resolution (2)- Recursive



Comparison

- Recursive:
  - Puts a higher performance demand on each name server
  - Name servers in the global layer do not support recursive
  - Name servers use caching technique to enhance performance
- Iterative:
  - Caching is restricted to the client's name resolver
  - Name resolution process is the same for different clients
  - Organizations use a local intermediate name server that is shared by all clients
  - Communication is more expensive

Implementation of Name Resolution (4)



- The comparison between recursive and iterative name resolution with respect to communication costs.

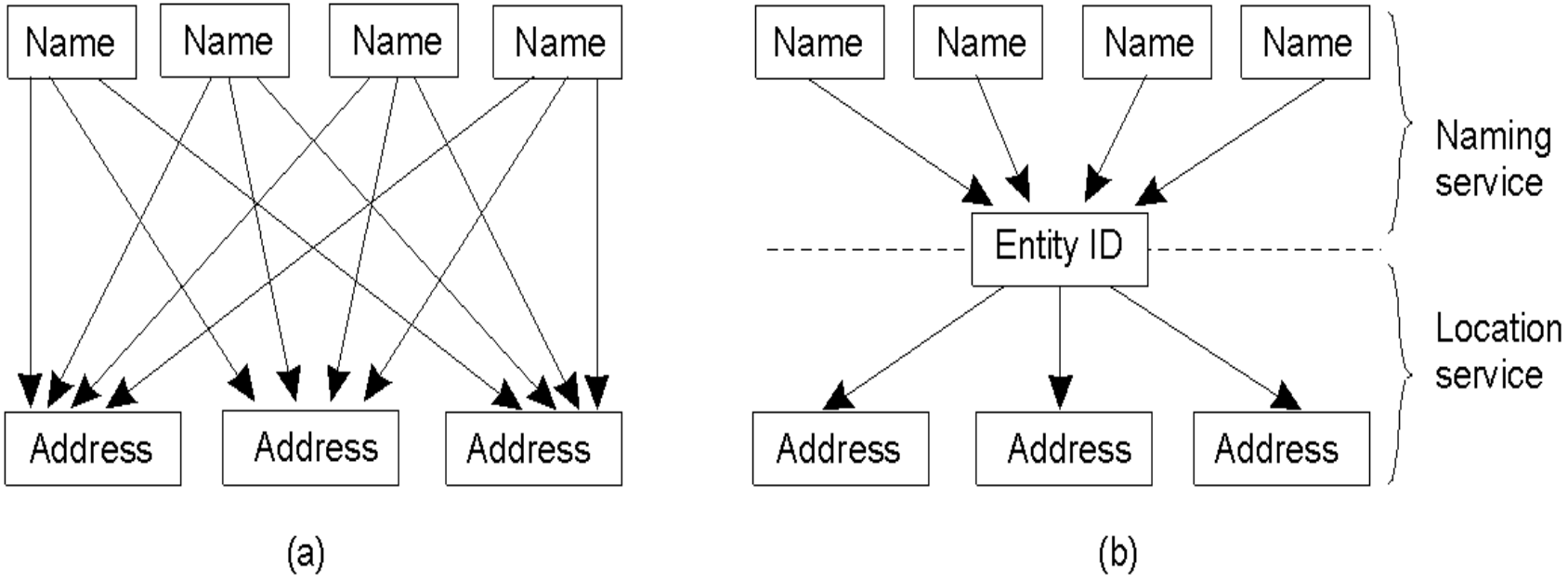
The DNS Name Space

- The largest distributed naming service is the Internet Domain Name System (DNS)

Locating Mobile Entities

- Mobile entities
  - Entities that do not have fixed location
  - Name-to-address mappings change regularely.
- Types of names
  - Human-friendly names
  - Identifiers:
    - Unique and fixed
    - Optimized for machine processing only
  - Addresses
- Problem
  - Traditional name services maintain a direct mapping between human-friendly names and addresses of entities
  - A change in name or address results in change of the mapping

Naming versus Locating Entities



a) Direct, single level mapping between names and addresses.

b) Two-level mapping using identities.

Location service accepts an identifier as input and returns the current address of the entity

Broadcasting and multicasting

- Suitable for a LAN
- Locating is performed by broadcasting a message containing the identifier
- The machine that contains the entity responds with the address of the access point
- Inefficient for large-scale distributed systems

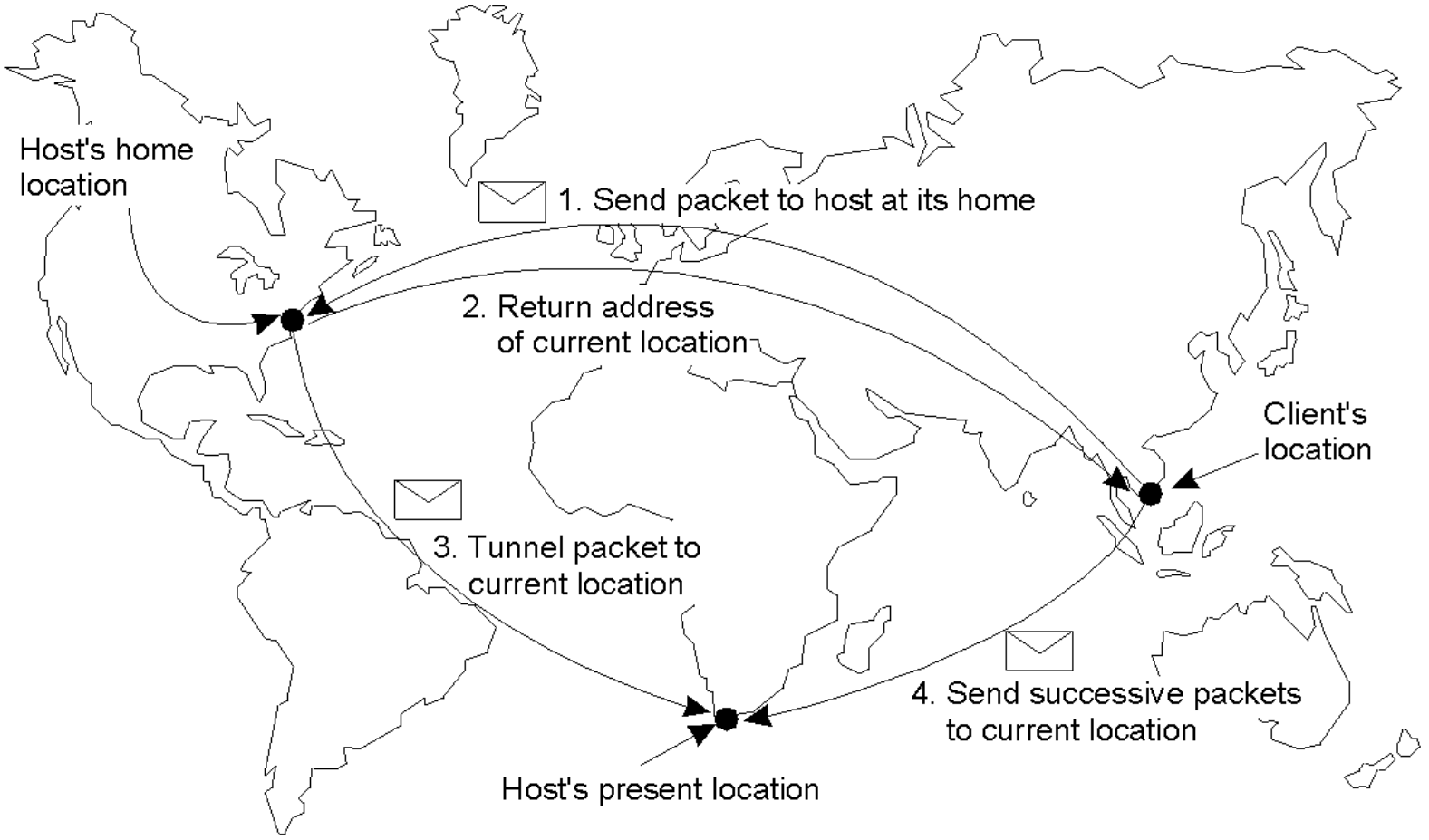
Forwarding Pointers

- When an entity moves from A to B, it leaves behind a reference to its new location at B.
- The approach is simple.
- It can use the traditional naming service.
- A client can lookup the current address by following the chain of forwarding pointers.
- A chain can become so long (Drawback).
- If a forwarding pointer is lost for whatever reason, the entity can no longer be reached (Drawback)

Home-based approach

- Introduce a home location for each mobile entity.
- The home location is responsible for keeping track of the current location of the entity.
- The home location could be the place where the entity is created.
- For example, each mobile host could use a fixed IP address.
- All communication to that entity is initially directed to that IP address.
- When the home location receives a packet, it looks up the mobile entities current location

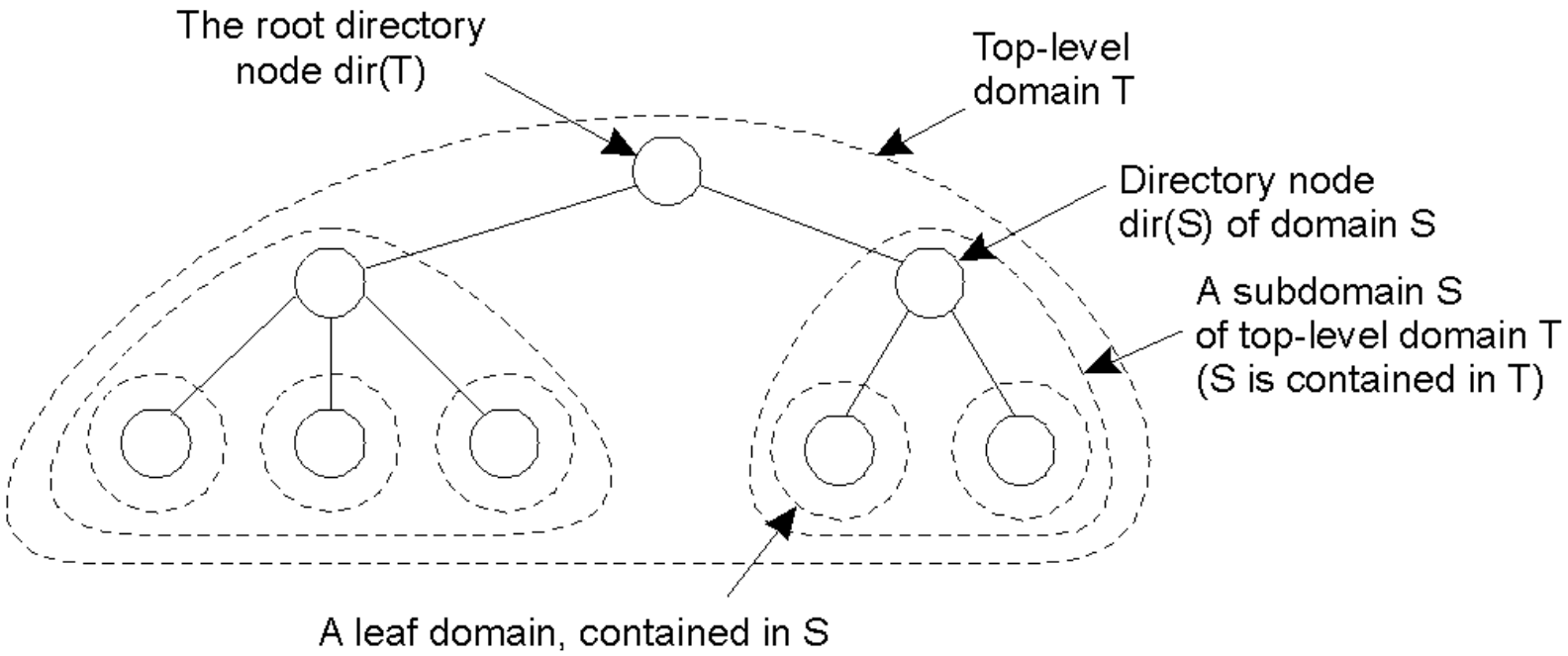
Home-Based Approach



Hierarchical Approach(1)

- A network is hierarchically divide into a collection of domains (Similar to DNS).
- There is a single top-level domain that spans the entire network.
- Each domain can be subdivided into multiple smaller domains.
- A lowest level domain (leaf domain) corresponds to a LAN or a cell (in Mobile phone network).
- Each domain D has an associated directory node dir(D) that keeps track of the entities in that domain.
- The directory node of the top-level domain (root directory node) knows about all entities.

Hierarchical Approach (2)

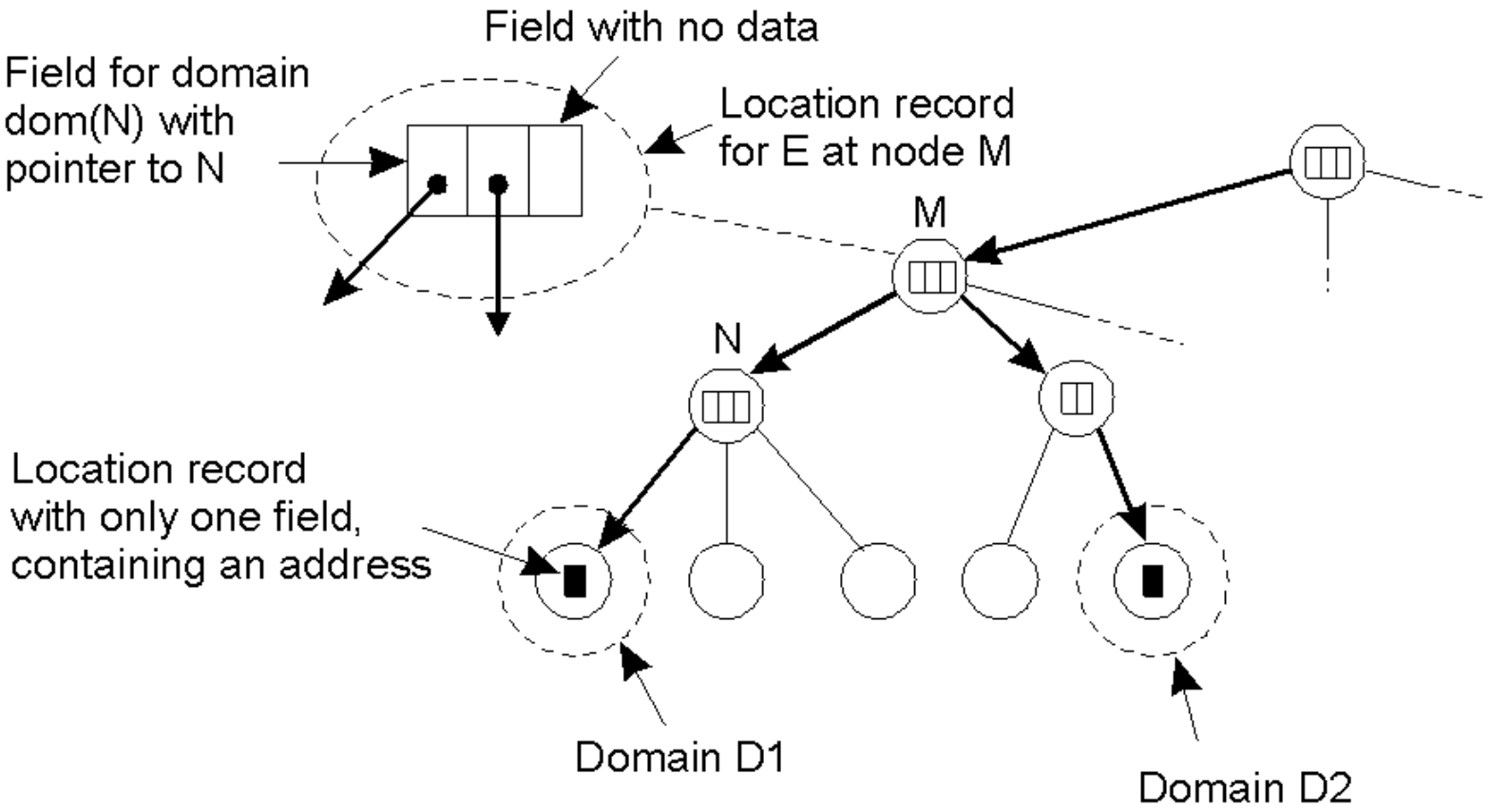


- Hierarchical organization of a location service into domains, each having an associated directory node.

Hierarchical Approach(3)

- Each entity currently located in a domain D, is represented by a location record in the directory node dir(D).
- A location record for entity E in the directory node N for a leaf domain D contains the entity's current address in that domain.
- The directory node N' for the next higher level domain D' that contains D, will have a location record for E containg only a pointer to N.
- If an entity has 2 addresses in domains D1 and D2, then the directory node of the smallest domain containing both of them, will have two pointers,

Hierarchical Approach (4)

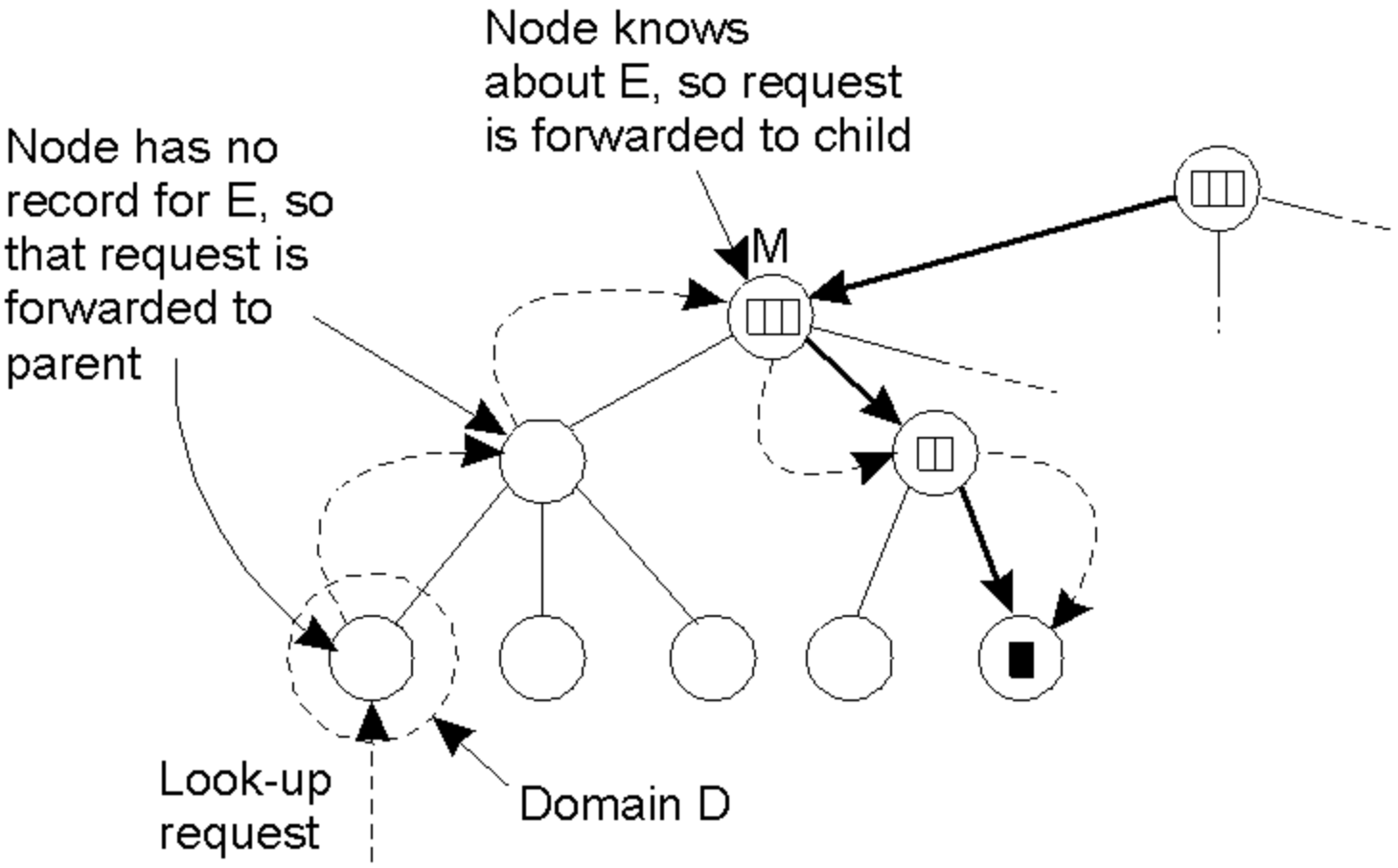


- An example of storing information of an entity having two addresses in different leaf domains.

Hierarchical Approach(5)

- Lookup
  - Client issues a lookup request at the directory node of its current leaf domain D.
  - If the directory node does not store a location record for the entity, it forwards the request to its parent (and so on).
  - When request reaches a directory node M that sores a location record for the entity .
  - The request then forwarded to the directory node of the containing subdomain (and so on).
  - When the request reaches a leaf node, the address of the entity is returned.

Hierarchical Approach (6)

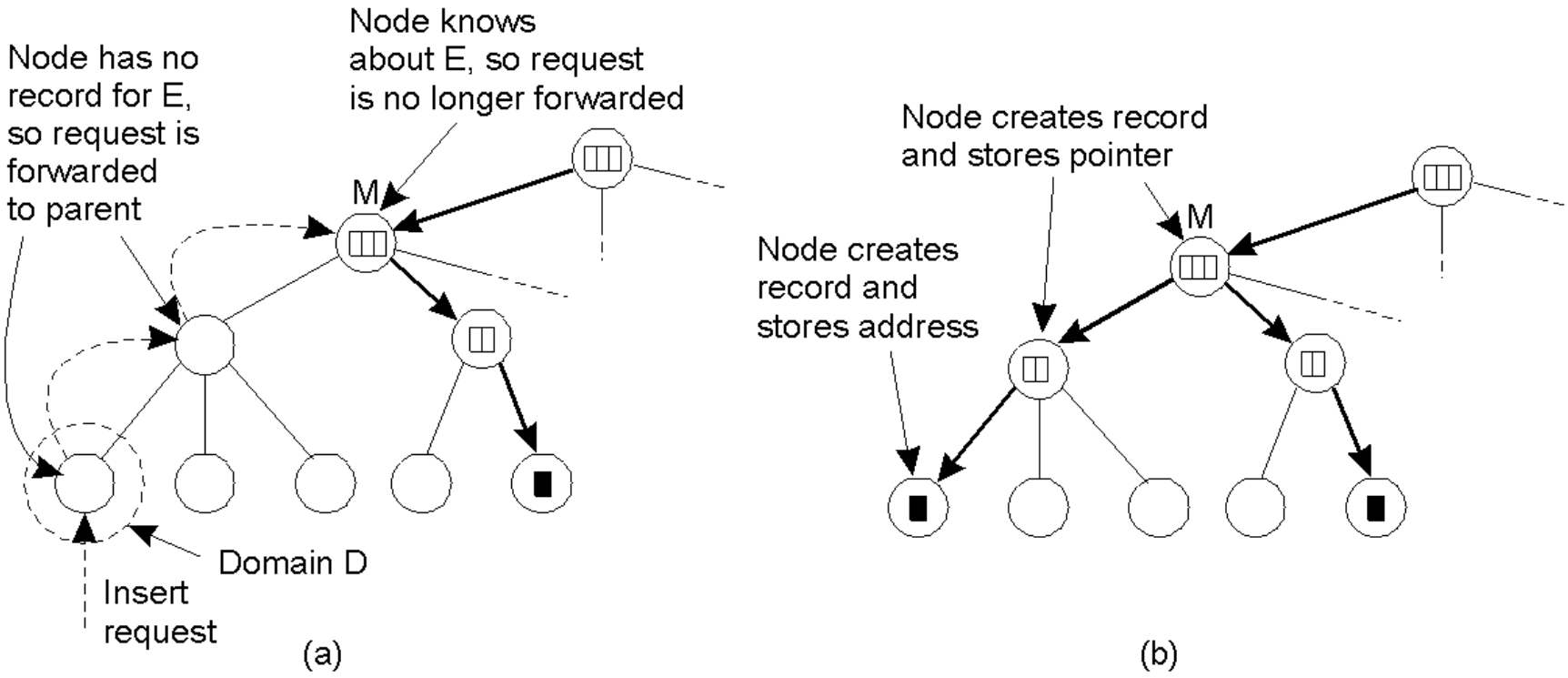


- Looking up a location in a hierarchically organized location service.

Hierarchical Approach(7)

- Update propagation is performed similar to the lookup process.
- Consider an entity that has created a replica in leaf domain D
- The insertion is initiated at the leaf node dir(D).
- Dir(D) forwards the insertion to its parent (and so on), until it reaches a directory node M that already stores a location record for the entity.
- Node M updates the insertion record by inserting its child node (and so on).
- Intermediate nodes create a location record for the entity referring to their child. (Until it reaches the initiating leaf node)

Hierarchical Approach (8)



a) An insert request is forwarded to the first node that knows about entity *E*.

b) A chain of forwarding pointers to the leaf node is created.

Hierarchical Approach(9)

- Delete operation is similar to insert operation.
- When address of an entity in leaf node needs to be removed, dir(D) is requested to remove the address from the corresponding record.
- If the record becomes empty (no other addresses for E), the record can be removed.
- The parent of dir(D) is requested to remove the pointer to dir(D).
- The process continues until reaching root (or reaching a directory node with more than one entries)