Chapter 11: Naming

Naming

- Objectives
  - To motivate the differences of addresses and names
  - To develop an understanding of naming types backed by example systems

- Topics
  - Address
  - Names
  - Naming types
    - Hierarchical Naming
    - Flat Naming
    - Attribute-based Naming

Accessing Entities: Address

- Need to operate on entities exists for
  - Hosts and users
  - Web pages and services
  - Files (*e.g.*, on an FTP server)
  - …
- To operate on an entity, access to it is essential
  - Each entity should have an access point, usually called address
    - Sometimes descriptive attributes or a service may be known only
- Various types of addresses exist, *e.g.*:
  - Phone number, IP address, port, URL
  - User ID, postal address

Names

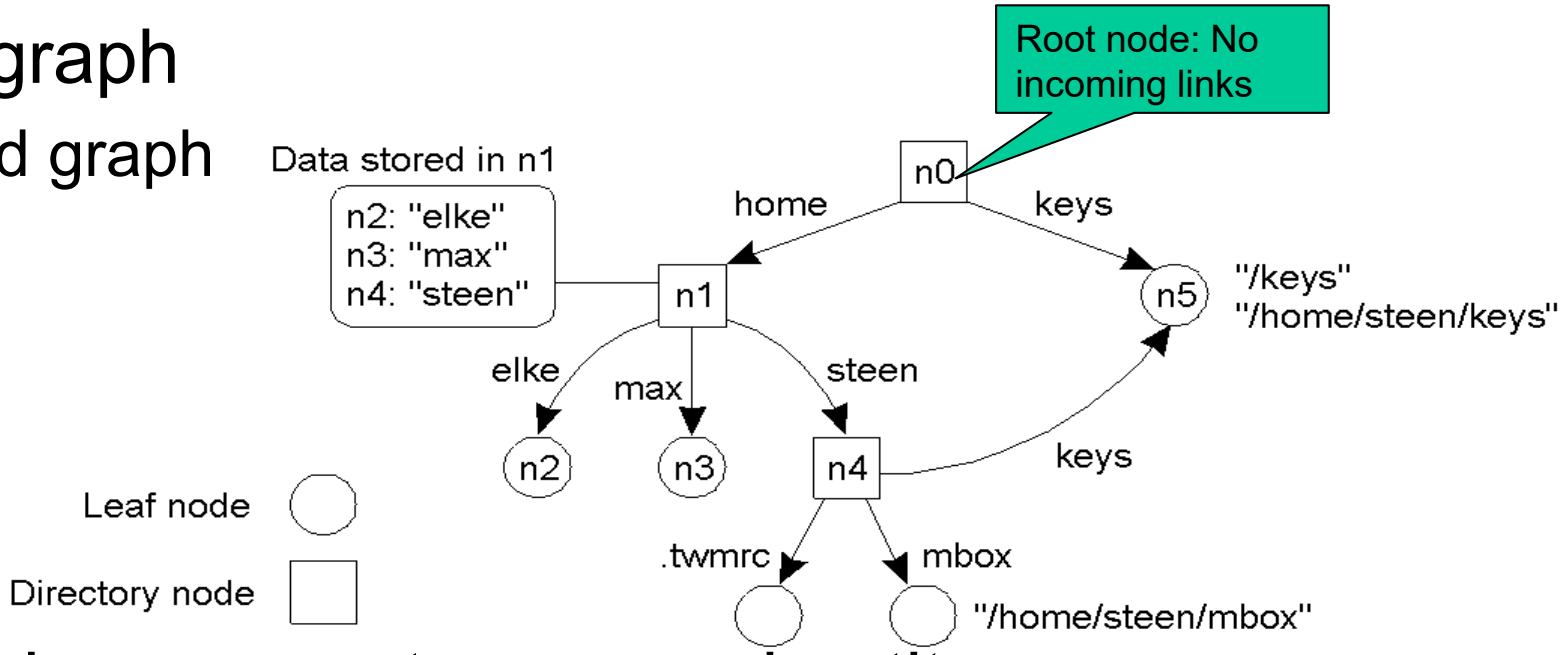
- If an address is sufficient to access an entity, what are names needed for?
- Various reasons:
  - Address change of an entity
    - *E.g.*, due to relocation (with a new IP address)
    - *E.g.*, due to reorganization (a Web server may be moved to a new host)
  - An entity may have more than one access point (replication)
- Naming provides an abstraction useful for
  - Location independence
  - Relocation of entities
  - Allowing a single reference to a set of alternative access points
  - Offering human-friendly names

The Three Naming Types

- Hierarchical Naming
  - DNS (Domain Name System)
  - Unix File System
- Flat Naming
  - Distributed Hash Tables (DHT)
- Attribute-based Naming
  - X.500

Hierarchical Naming: Name Spaces

- Naming graph
  - Directed graph



- A leaf node represents a named entity
  - No outgoing links

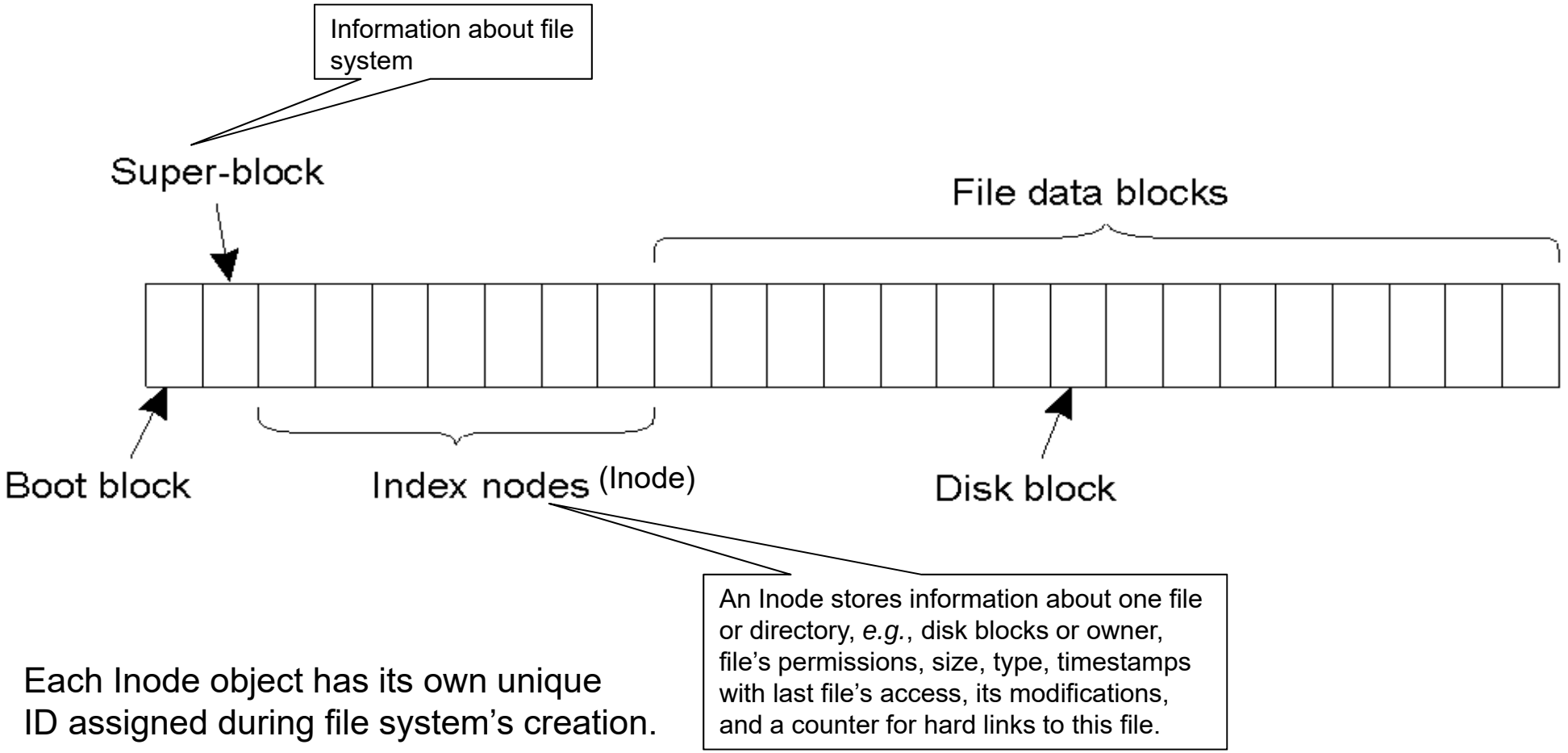
- A directory node stores links to other nodes
  - Leafs or other directory nodes

Hierarchical Naming Concept

- Based on the concept of "Name Spaces"
  - A directed graph of names
  - Ordered and defined by a fixed alphabet of characters
- Each name is a path in the naming graph
  - If starting from the root, we call it an absolute name
  - Otherwise, relative name
- Examples (of absolute names):
  - File system: /home/stiller/teaching/fs20/SSDS20/exam.pdf
  - DNS: www.csg.uzh.ch
  - URL: http://www.csg.uzh.ch/csg/en/teaching/fs20/ssds.html

Unix File System

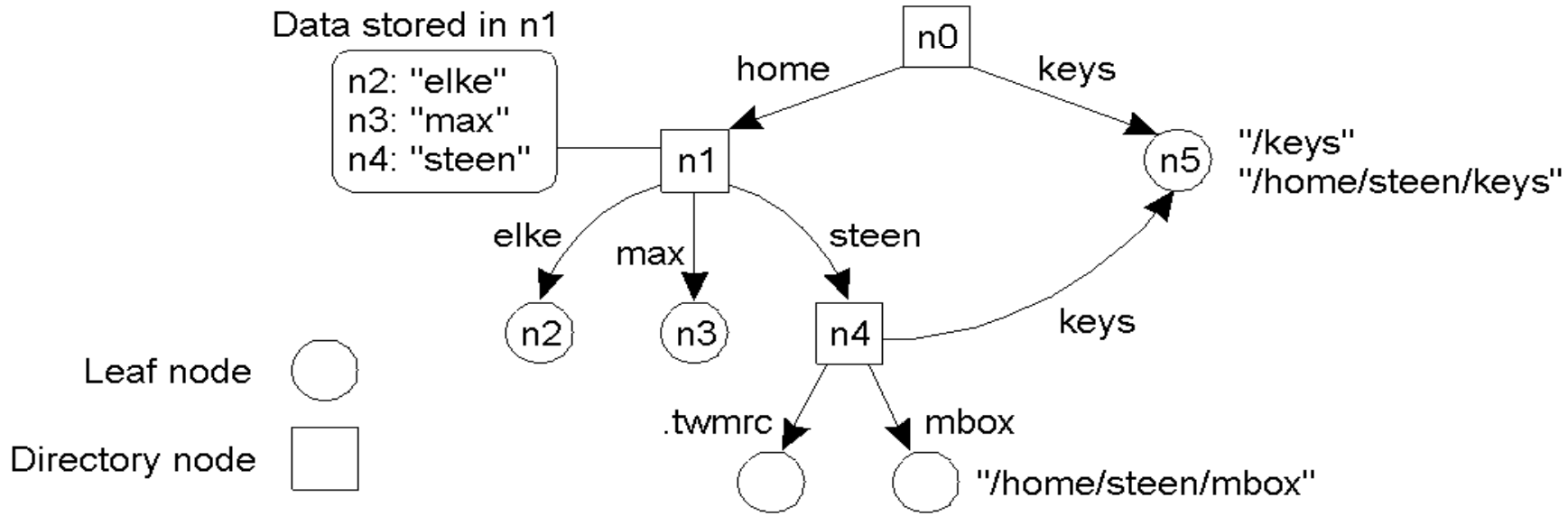
A logical disk of contiguous disk blocks



Each Inode object has its own unique ID assigned during file system's creation.

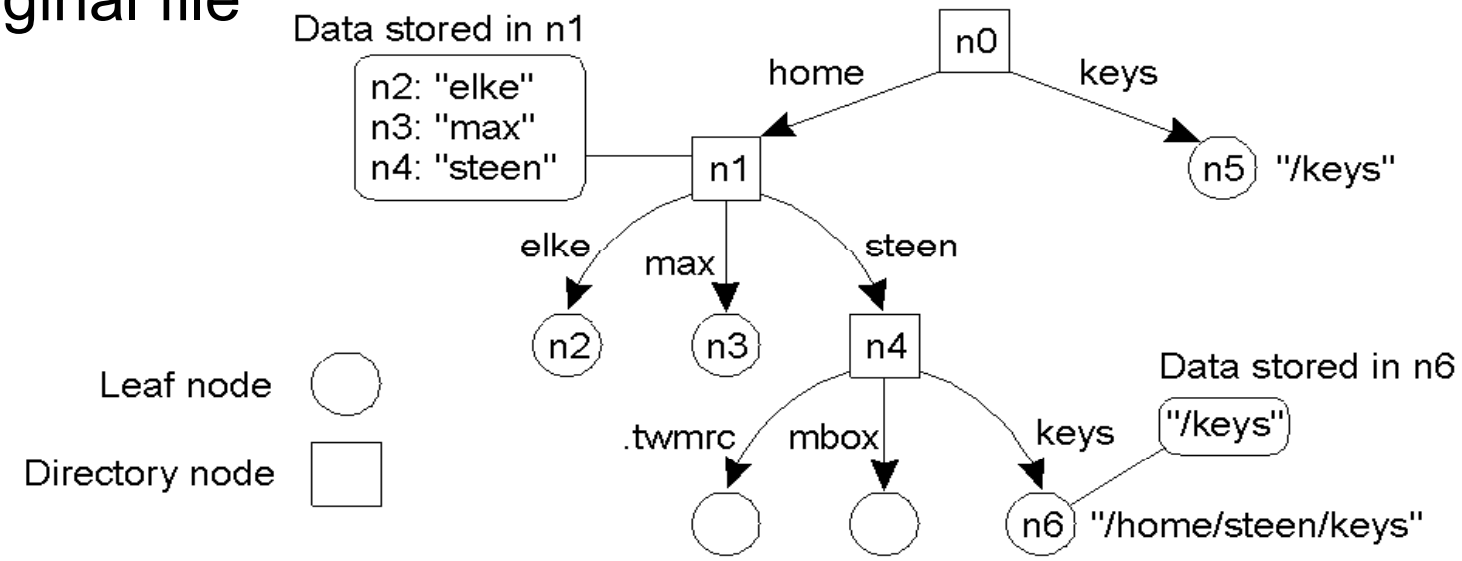
Multiple Names: Hard Links

- An entity may have multiple names within a name space
  - Multiple paths that lead to the same leaf node
    - The same, identical file with just another name
- In Unix they are called hard links
  - Pointing to the "inode"



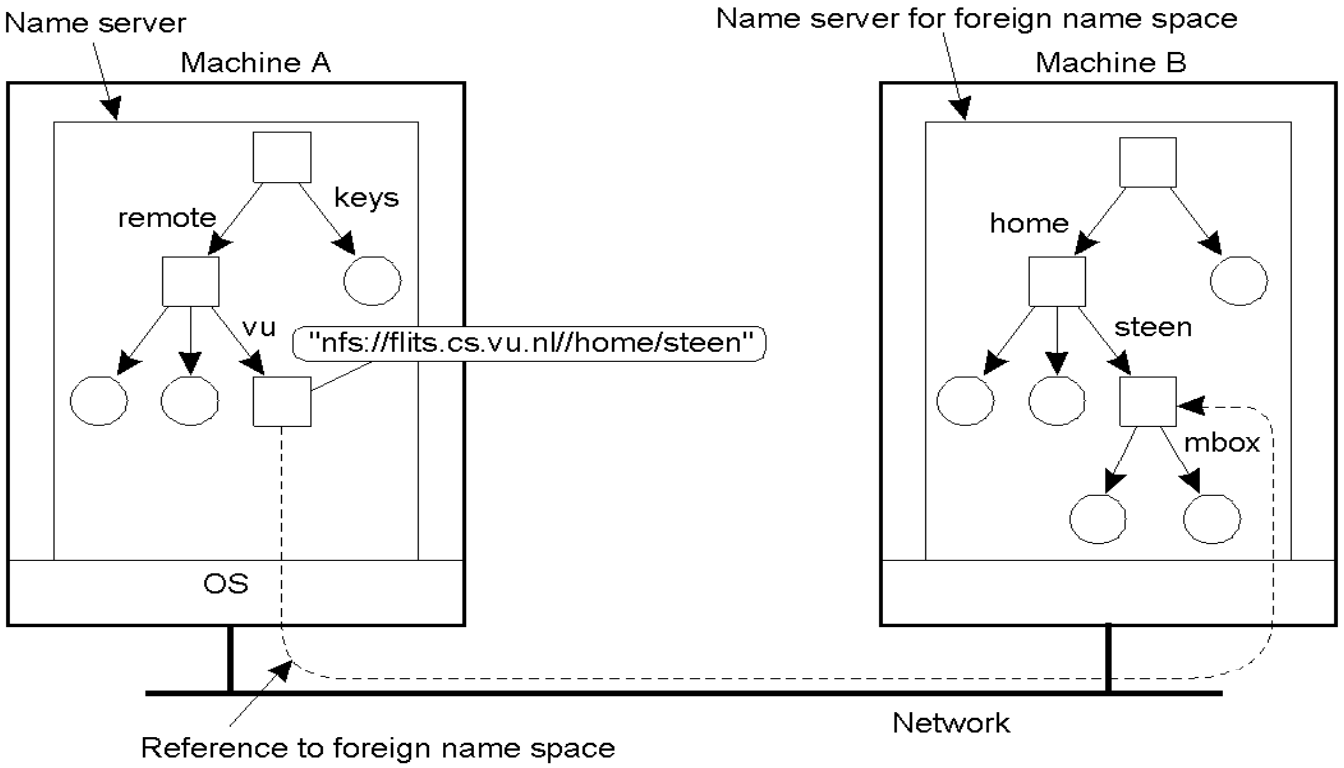
Multiple Names: Symbolic Links

- A special node contains the absolute (or relative) name of another node
- In Unix this is called "soft link" or "symbolic link"
  - Changing the symlink's name, attributes, or even delete or just "change a direction" to another file without affecting the original file



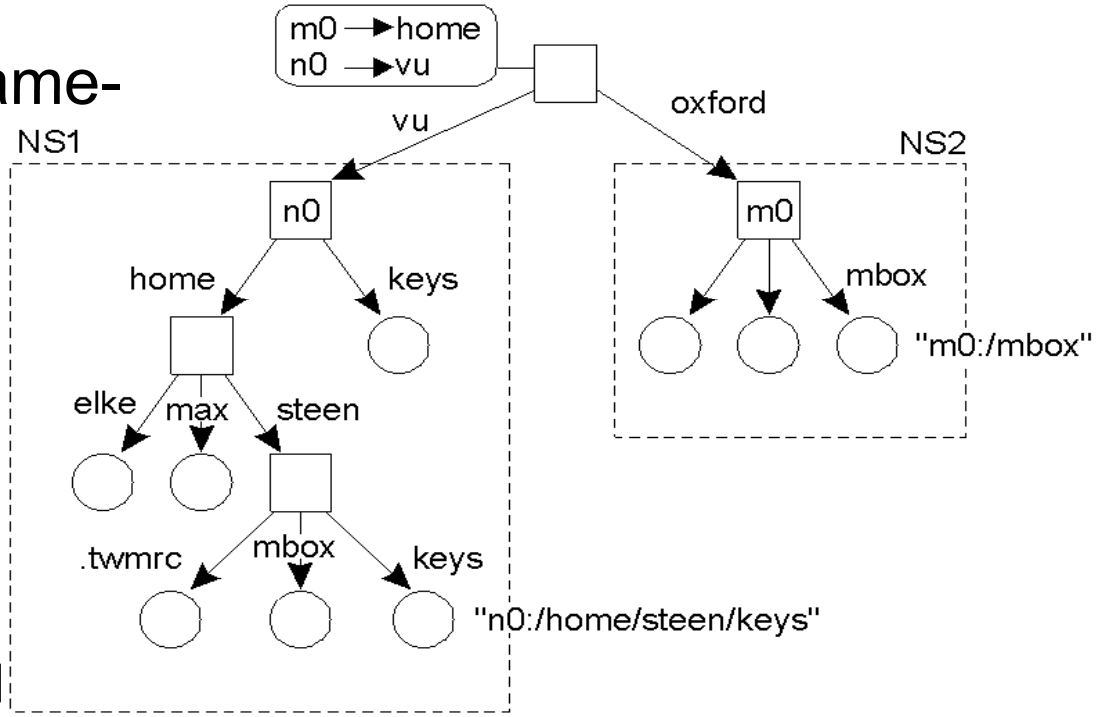
Mounting: Remote Linking

- Mounting
  - Setting a symbolic link referring to a remote name space
  - Performed through a specific process protocol



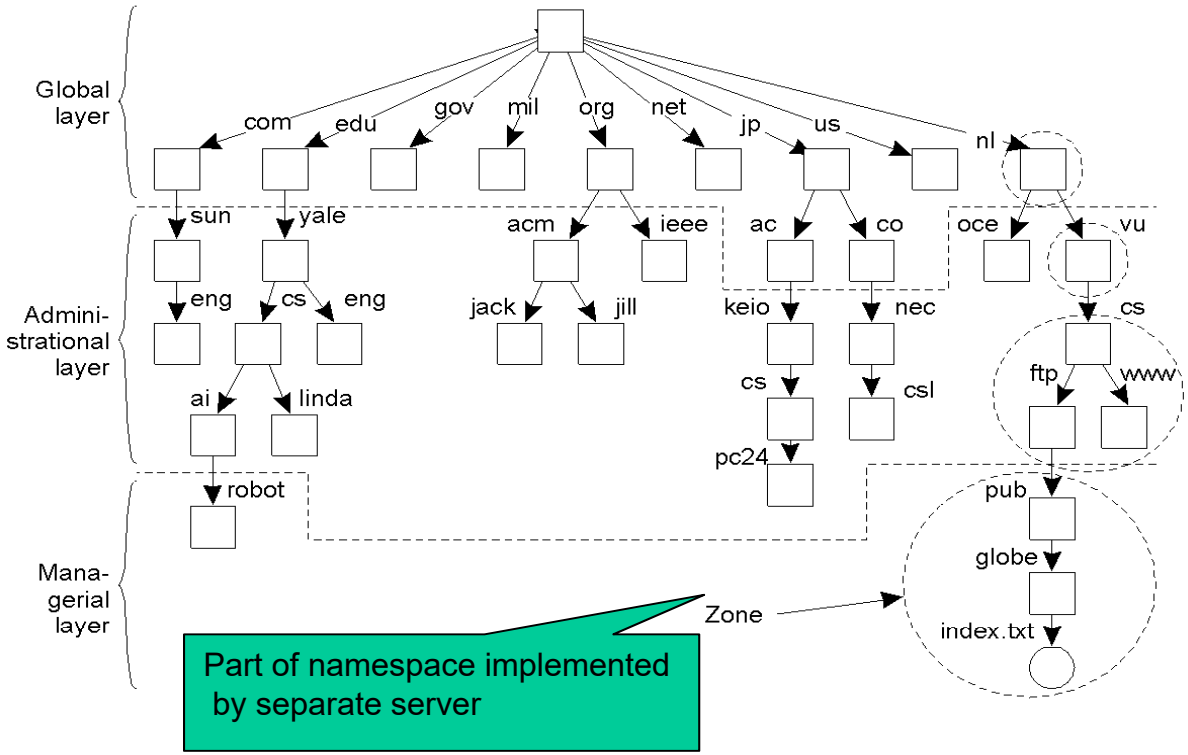
Merging Name Spaces

- Adding a new root and mounting two or more namespaces below it (*e.g.*, DEC Global Name Service)
- Problem
  - Absolute names of all name-spaces are changed
- Solution
  - At root node, cache original top-level names
    - *E.g.*, root remembers that home, keys map to /vu, mbox maps to /oxford
  - Only file systems with different top-level directories can be merged



Name Space Distribution

- Ok, so far, for small scale naming
  - *E.g.*, a company-wide file system
- But what happens when the scale grows to … global?!
  - *E.g.*, DNS
- Partitioning of the DNS name space, including Internet-accessible files, into three layers

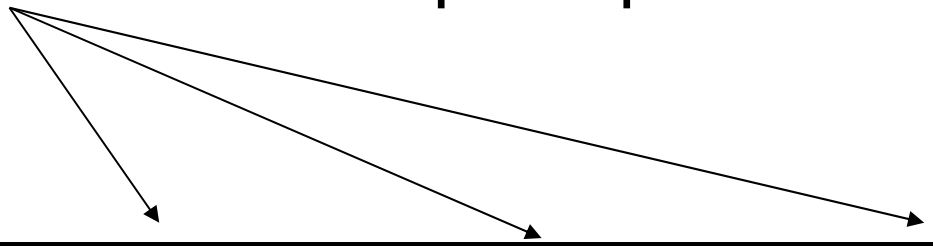


Domain Name Systems (DNS)

- How to map structured hostnames to IP addresses?
  - Old days: HOSTS.TXT file FTPed among hosts
- DNS operates as a distributed directory service
  - Hierarchical name space
  - One global root
    - Replicated across 13 root servers
    - All Denial-of-Service (DoS) attack on these root servers unsuccessful
      - Due to caching: queries to root servers are relatively rare
- DNS is most longstanding and stable global directory service, if not the only one!

DNS Layers

- A comparison between name servers for implementing nodes from a large-scale name space partitioned into 3 layers



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

DNS Data Base Entries

- For the zone cs.vu.nl

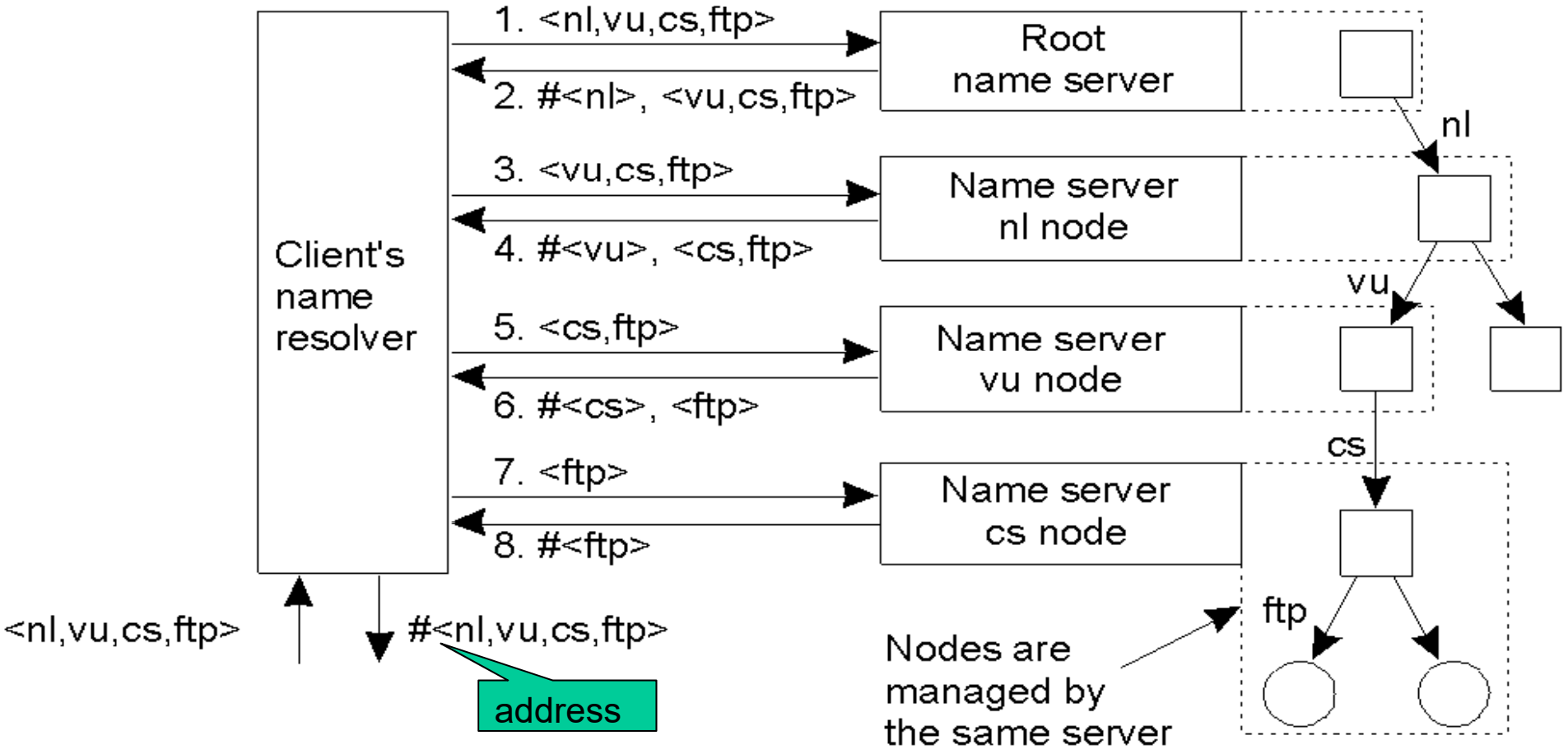
Name	Record type	Record value
cs.vu.nl	SOA	star (1999121502,7200,3600,2419200,86400)
cs.vu.nl	NS	star.cs.vu.nl
cs.vu.nl	NS	top.cs.vu.nl
cs.vu.nl	NS	solo.cs.vu.nl
cs.vu.nl	TXT	"Vrije Universiteit - Math. & Comp. Sc."
cs.vu.nl	MX	1 zephyr.cs.vu.nl
cs.vu.nl	MX	2 tornado.cs.vu.nl
cs.vu.nl	MX	3 star.cs.vu.nl
star.cs.vu.nl	HINFO	Sun Unix
star.cs.vu.nl	MX	1 star.cs.vu.nl
star.cs.vu.nl	MX	10 zephyr.cs.vu.nl
star.cs.vu.nl	A	130.37.24.6
star.cs.vu.nl	A	192.31.231.42
		Unix
		hyr.cs.vu.nl
		ado.cs.vu.nl
		1.231.66
		.cs.vu.nl
		.cs.vu.nl
		Unix
		ng.cs.vu.nl
		phyr.cs.vu.nl
		7.24.11
		S-DOS
		7.30.32
		37.130.in-addr.arpa
		7.26.0

RR: Resource Record (DNS Information Structure)

RR type	Entity	Description
SOA	Zone	“Start of Authority”: Holds information on the represented zone
A	Host	Contains an IP address (4 bytes) of the host this node represents
MX	Domain	Refers to a mail server to handle mail addressed to this node
NS	Zone	Refers to a name server that implements the represented zone
HINFO	Host	Holds information on the host this node represents
TXT	Any kind	Contains any human-readable information for an entity
…	..	..

Iterative Name Resolution

- IP address for ftp.cs.vu.nl?



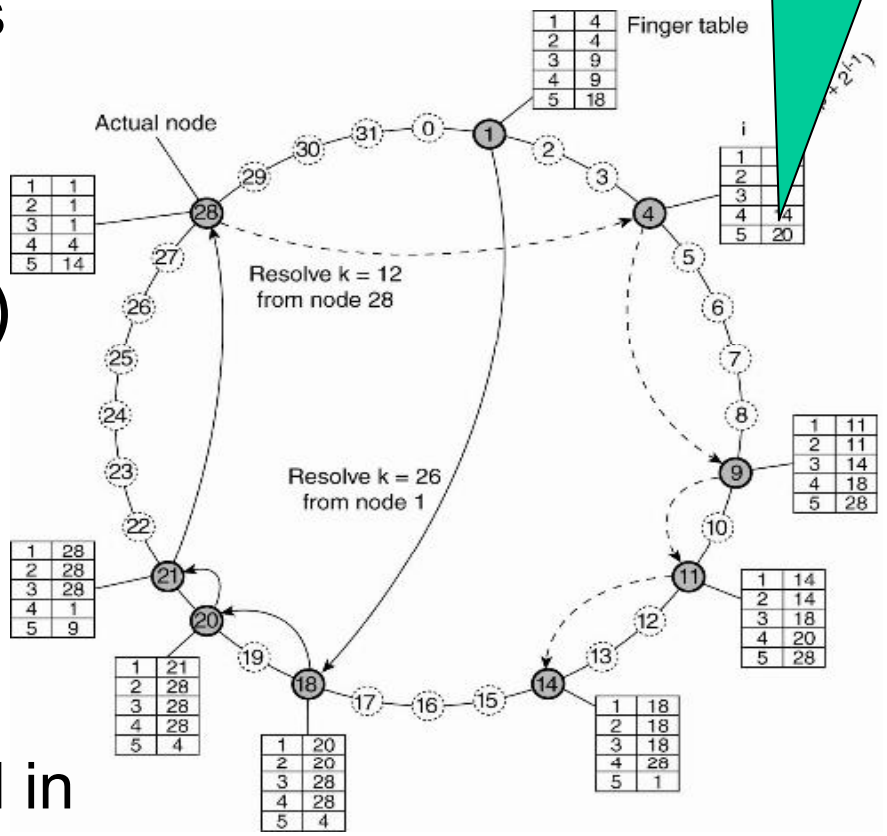
Flat Naming

- Useful for addressing space in a homogeneous way
  - *E.g.*, memory addressing
- Very common in centralized systems
  - *E.g.*, memory, low-level disk access
  - In decentralized systems very complicated
- Naïve approaches
  - Every node knows all names and addresses
  - Every node knows only some
    - Flood the network asking who has the name in question
    - The node that has that name replies
    - Reminds of peer-to-peer systems

Distributed Hash Tables (DHT)

- Chord DHT, supports many nodes by mapping
  - Key → value
  - Also useful for name → address
- Each node has a unique ID and maintains small routing table (pointers to other nodes)
- How to find a node holding the value for a key?
  - Find largest node ID ≤ key in routing table
  - If none exists: Values are stored in the node with the next higher ID in routing table

ith entry: smallest node Id larger by at least 2^{i-1}
i=1: 4 + 2^0 = 5 → 9
i=2: 4 + 2^1 = 6 → 9
i=3: 4 + 2^2 = 8 → 9
i=4: 4 + 2^3 = 12 → 14



Attribute-based Names (1)

- Example: X.500 Name Space

- Structured and attribute-based naming
  - Used for Lightweight Directory Access Protocol (LDAP) and for Microsoft's Active Directory

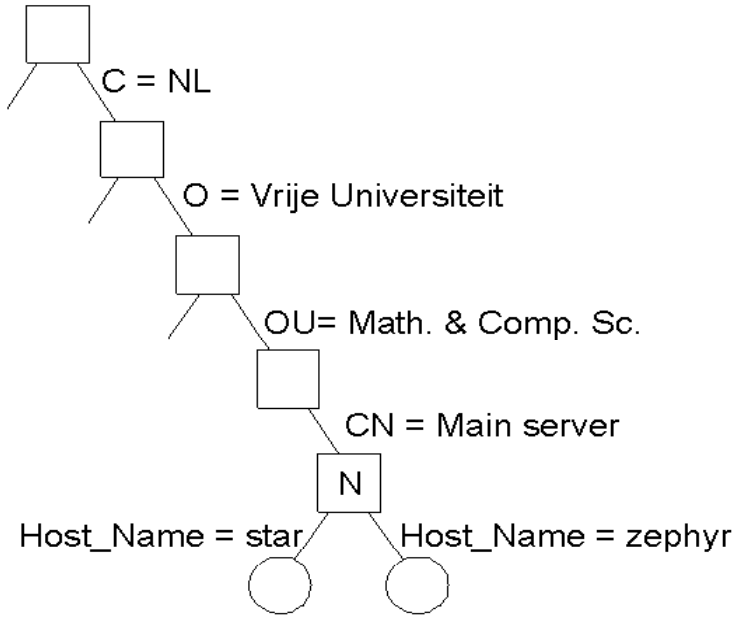
- Each attribute is called a Relative Distinguished Name (RDN)
  - Sequence of RDNs gives globally unique name
  - Better searching possible than with DNS with queries
    - *E.g.*, Find "C = NL and CN = Main server"

Attribute-based Names (2)

- Example
  - X.500 directory entry using X.500 naming conventions

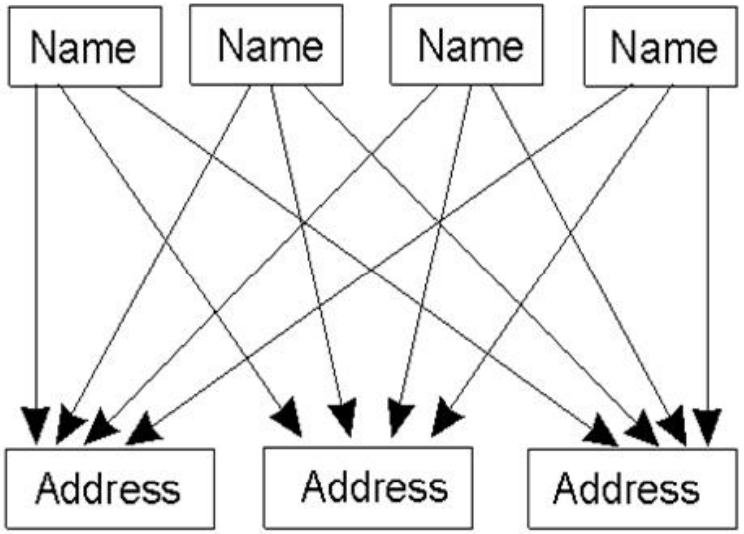
Attribute	Abbr.	Value
Country	C	NL
Locality	L	Amsterdam
Organization	O	Vrije Universiteit
OrganizationalUnit	OU	Math. & Comp. Sc.
CommonName	CN	Main server
Mail_Servers	--	130.37.24.6, 192.31.231,192.31.231.66
WWW_Server	--	130.37.21.11

  - X.500 naming graph



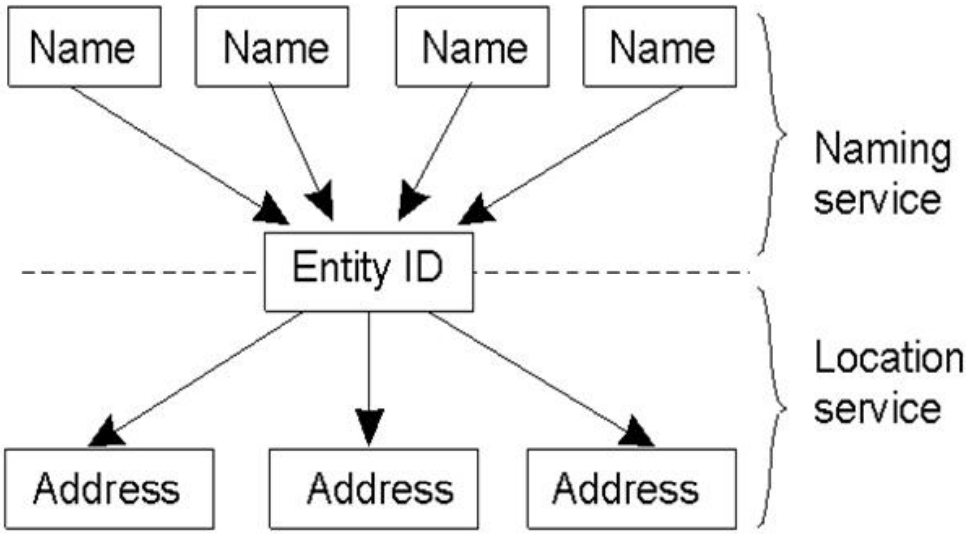
Naming vs. Locating Entities

- Names do not generally indicate, where an object is located



(a)

Direct, single level mapping between names and addresses

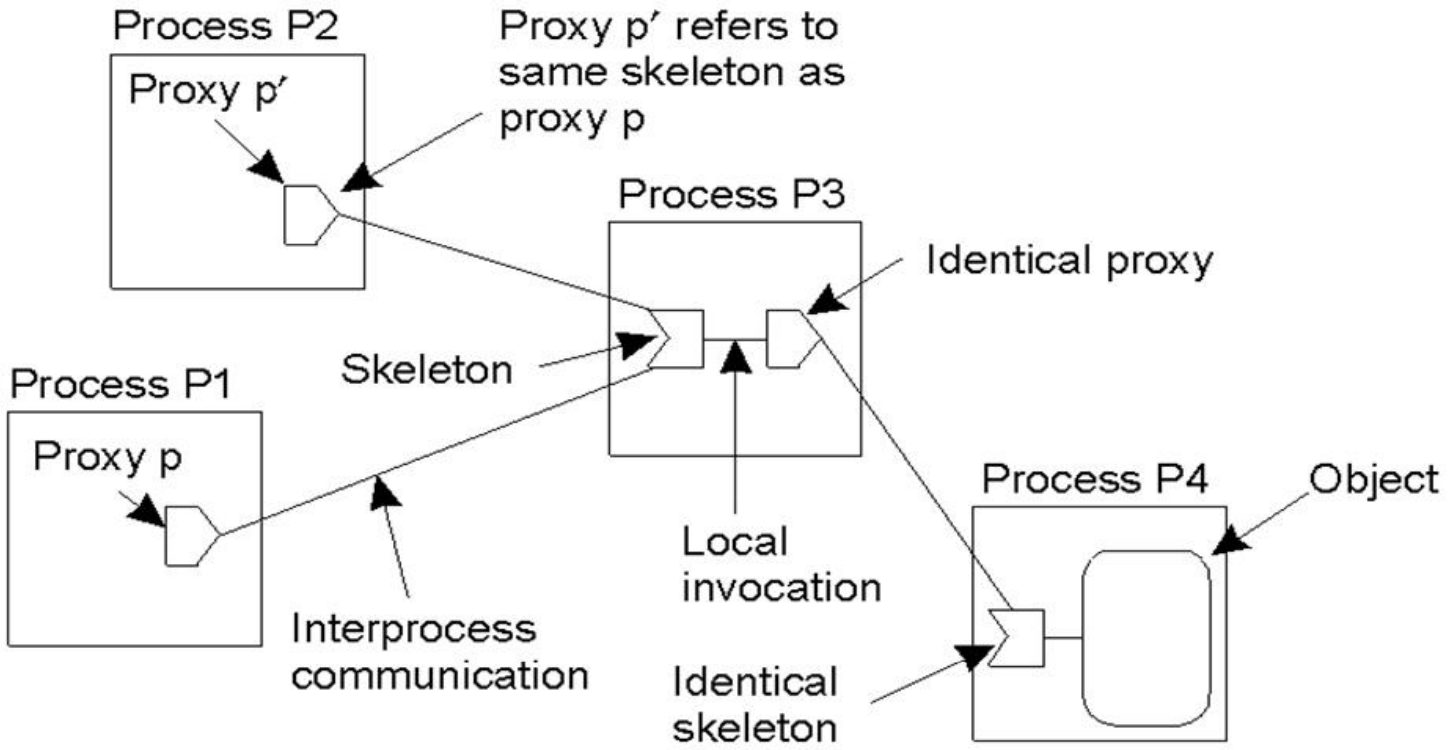


(b)

T-level mapping using identities

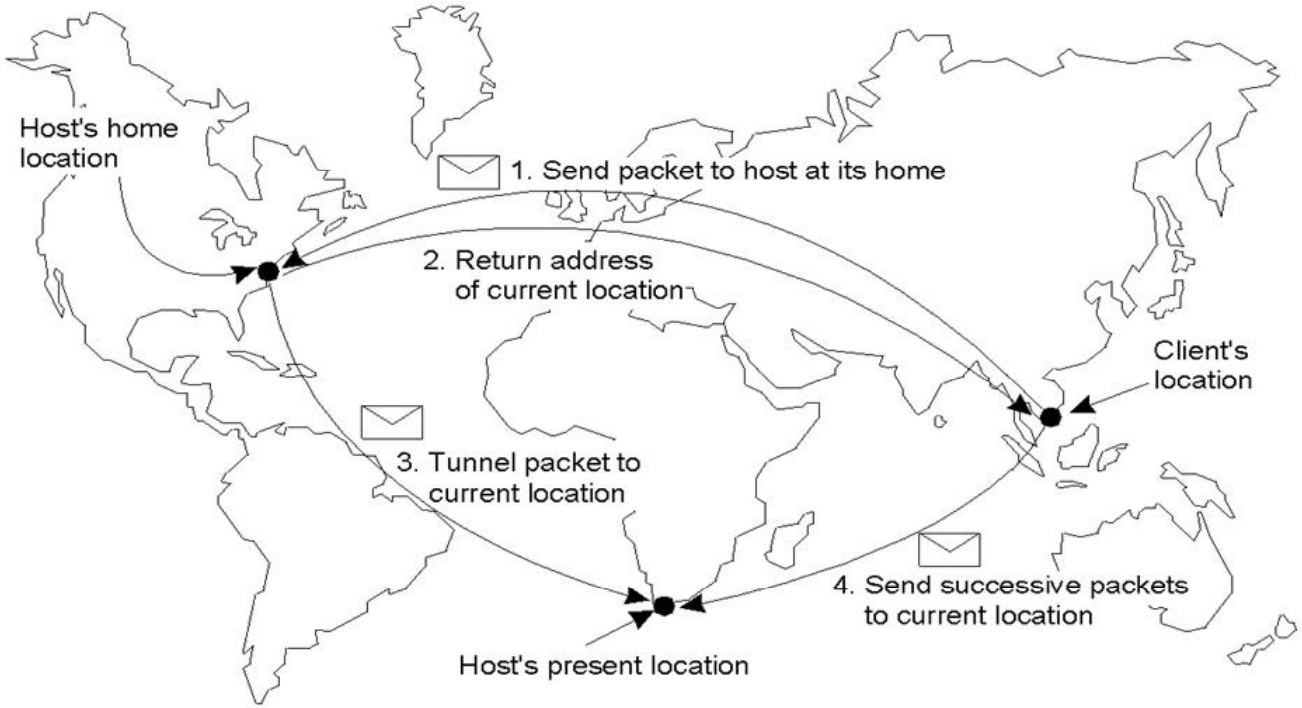
Forwarding Pointers (Movements)

- Forwarding pointer uses `(proxy, skeleton)` pairs
  - When objects move, they leave a `(proxy, skeleton)` pair
  - Calls just follow the chain to object



Home–based Approaches (Comparison)

- Each host has a fixed home location, *e.g.*, an IP address
  - When a host moves to a new network, it receives a new (temporary) IP address (Mobile IP)
  - The temporary IP address is stored at the home location



Hierarchical Location Services

- Locations are split into domains
  - Each having an associated directory node