



# CHƯƠNG 5: ĐỊNH DANH TRONG HPT

TS. TRẦN HẢI ANH

Tham khảo bài giảng của PGS. TS. Hà Quốc Trung

# Nội dung

2

1. Tên, định danh, và địa chỉ
2. Không gian tên phẳng
3. Không gian tên có cấu trúc
4. Không gian tên theo thuộc tính

3

# 1. Tên, định danh, thực thể, tên chức năng

Định danh

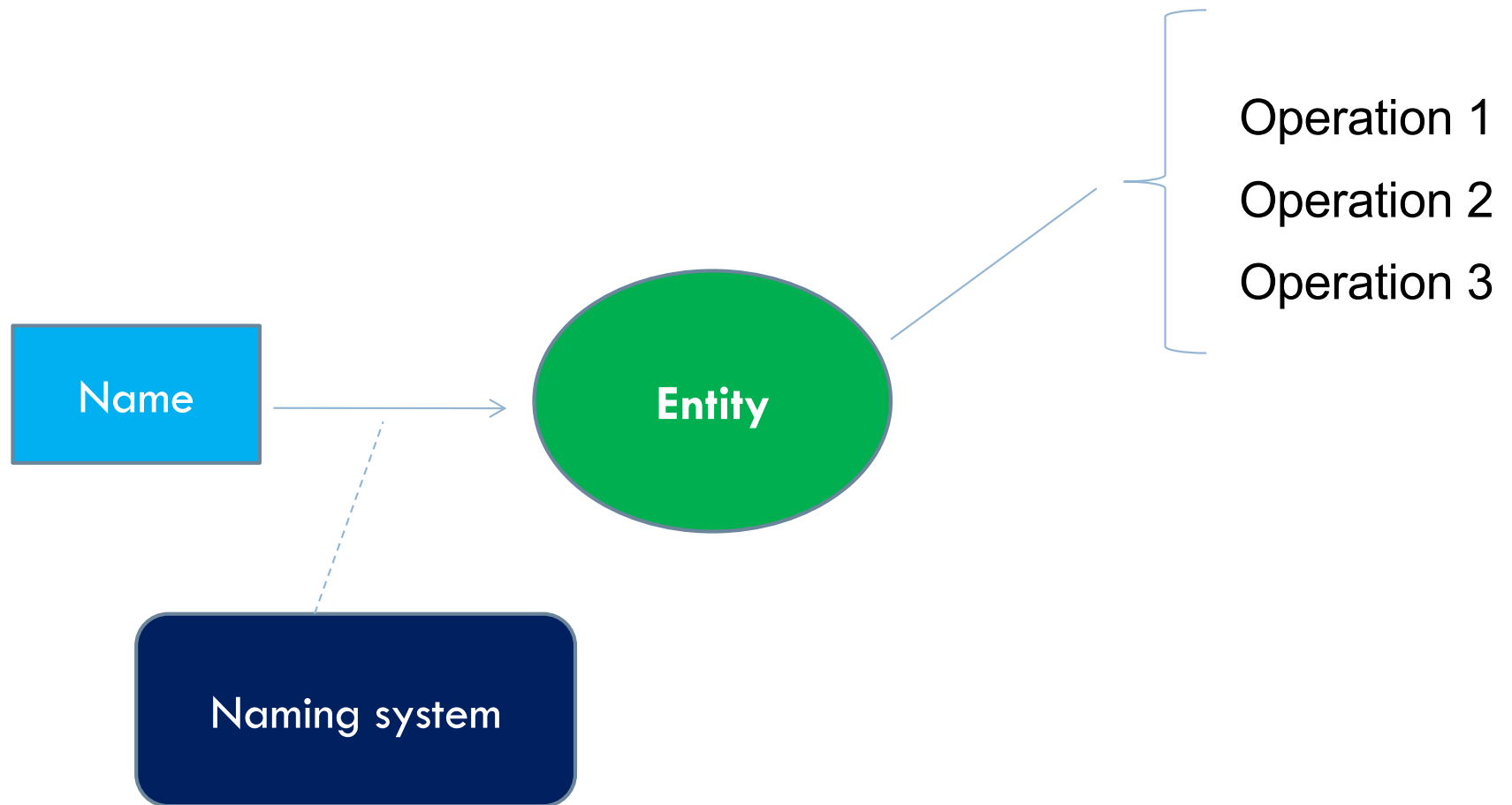
Tên

Địa chỉ

Phân giải tên

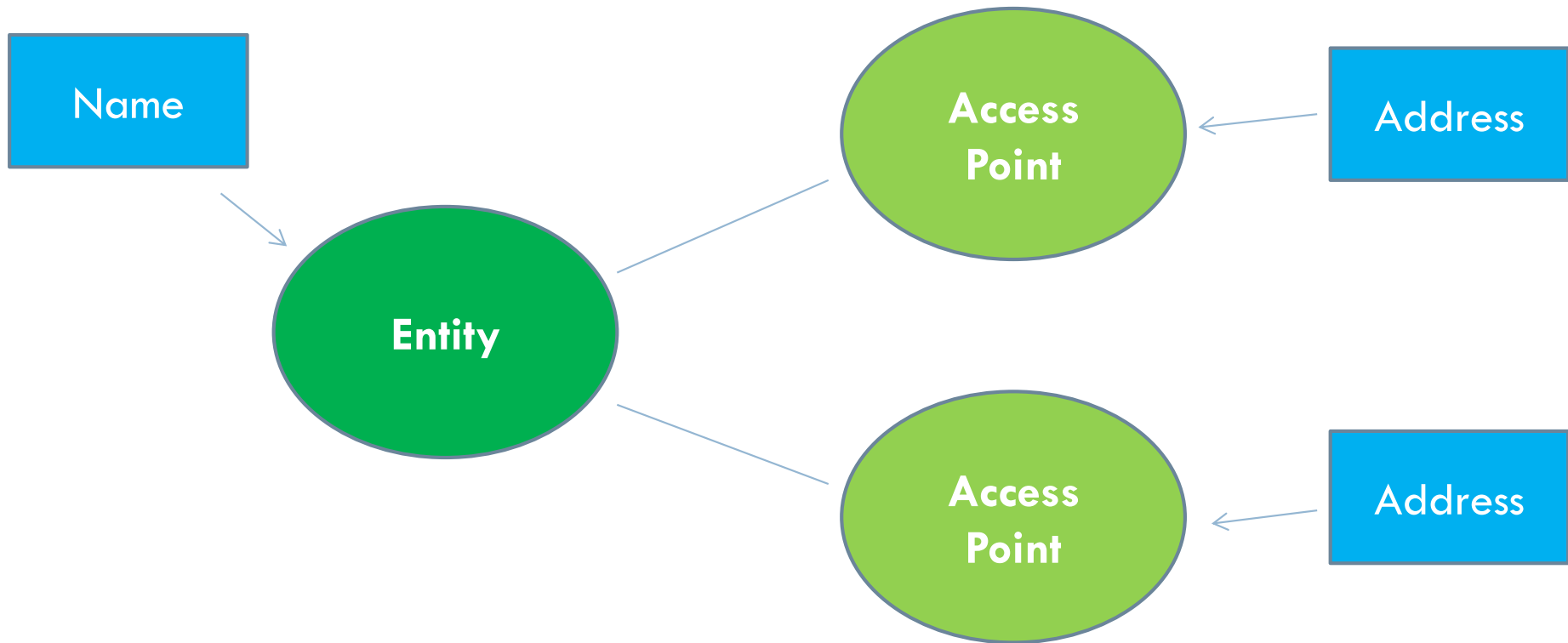
# Thực thể & tên

4



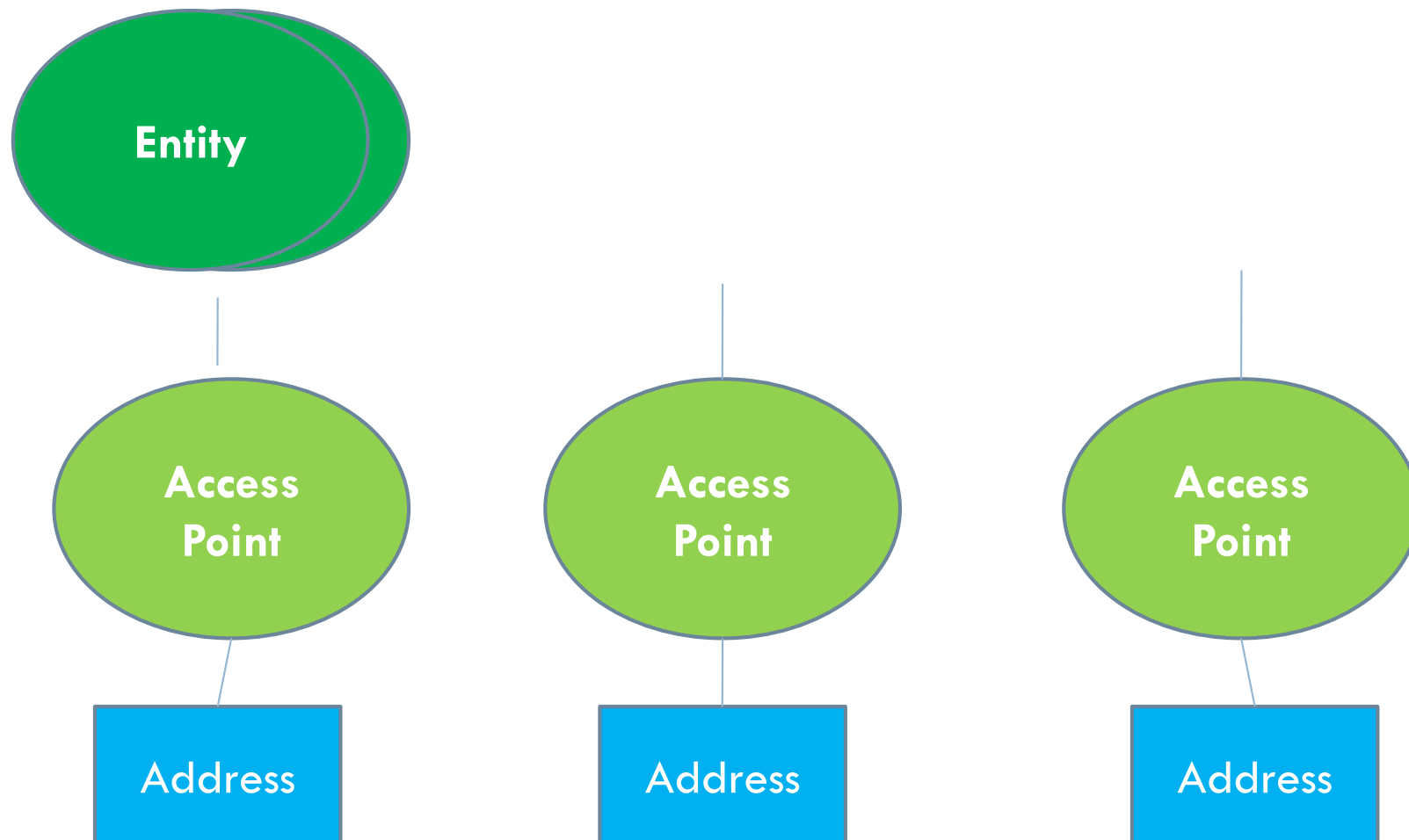
# Thực thể, A.P

5



# Tách biệt dịch vụ và vị trí

6



# Định danh

7

- 1 định danh chỉ đến nhiều nhất 1 thực thể
- Mỗi thực thể chỉ được xác định bởi 1 định danh
- Một định danh mãi mãi chỉ trỏ đến 1 thực thể.

# Phân giải *tên & định danh* thành *địa chỉ*

8

- Bảng ánh xạ tập trung *tên-địa chỉ*
- Vấn đề: không phù hợp với hệ thống mạng cỡ lớn

## Các hệ thống phân giải tên

- Yêu cầu của dịch vụ tên
  - ▣ Qui mô: vô hạn về tên và miền tên
  - ▣ Bền vững: chịu được các thay đổi
  - ▣ Sẵn sàng, chịu lỗi, chịu rủi ro bảo mật

□

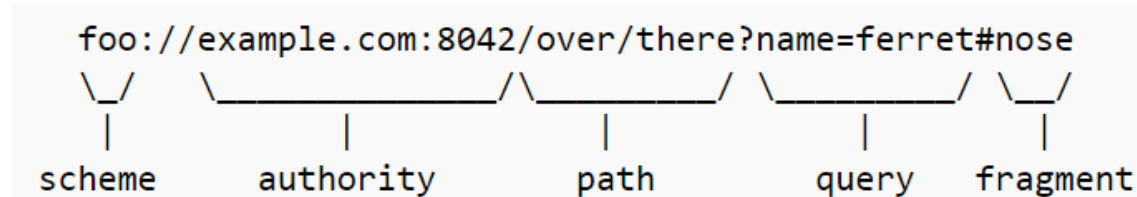


# URI, URL và URN

9

## □ URI:

- Xâu các ký tự để định danh tên của tài nguyên. Với sự biểu diễn tài nguyên trong 1 mạng, với các giao thức cụ thể. Được phân loại như là URL hoặc URN.
- 5 phần: scheme (sự xếp đặt), authority (nhà cung cấp), path (đường dẫn), query (truy vấn) và fragment (phần mảnh)



## □ URN:

- Chỉ số ISBN 0486275574 (`urn:isbn:0-486-27557-4`)

## □ URL:

- `file:///home/username/RomeoAndJuliet.pdf`

# Nội dung

10

1. Tên, định danh, và địa chỉ
2. Không gian tên phẳng
3. Không gian tên có cấu trúc
4. Định danh theo thuộc tính

## 2. Không gian tên phẳng

- 2.1. Khái niệm
- 2.2. Các giải pháp thông thường
- 2.3. Giải pháp Home-based
- 2.4. Giải pháp sử dụng hàm băm phân tán
- 2.5. Giải pháp phân cấp

## 2.1. Khái niệm

12

- ▣ Chuỗi bit, chuỗi ký tự không cấu trúc
- ▣ Không cho biết thông tin về vị trí
- ▣ Nhiệm vụ: cho biết tên, xác định vị trí
  1. Các giải pháp thông thường
  2. Home-base (dựa vào Home Agent)
  3. DHT
  4. Cách tiếp cận phân cấp

## 2.2. Các giải pháp thông thường

13

- 2.2.1. Quảng bá/thông báo nhóm
- 2.2.2. Chuyển tiếp con trỏ (Forwarding pointers)

## 2.2.1. Quảng bá/thông báo nhóm

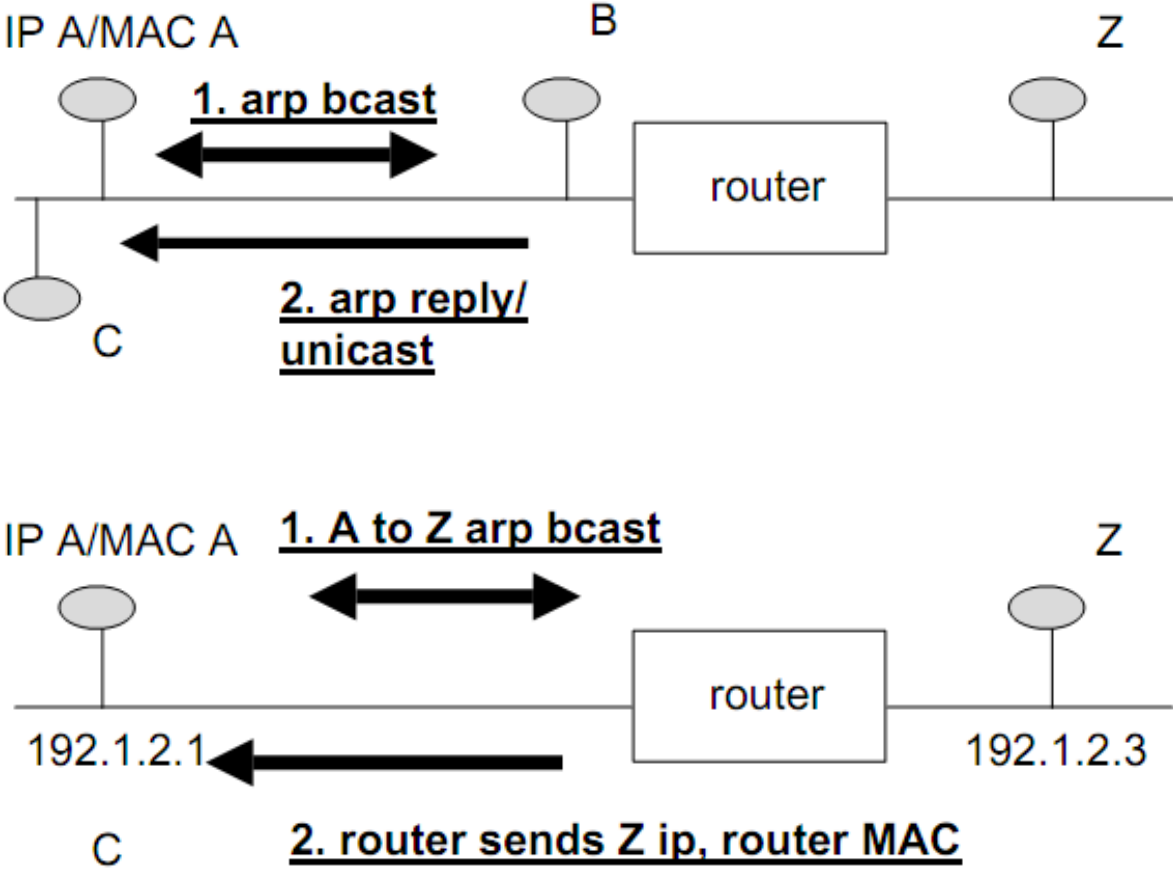
14

- ĐK: hệ phân tán hỗ trợ việc trao đổi thông tin thông qua quảng bá
  - ▣ Một thông báo có chứa định danh cần phân giải được quảng bá tới tất cả các thực thể trong hệ thống.
  - ▣ Thực thể nào có đúng định danh trong thông báo nhận được sẽ quảng bá một thông báo chứa định danh và địa chỉ của thực thể.
  - ▣ Tất cả các thực thể khác sẽ nhận được thông báo này và có được ánh xạ giữa định danh và địa chỉ của thực thể nói trên.

## 2.2.1. Quảng bá/thông báo nhóm

- Kém hiệu quả khi kích thước mạng tăng.
  - ▣ Băng thông bị bận, các thực thể liên tục xử lý các yêu cầu không phải của mình
- Thay thế quảng bá bằng truyền thông nhóm trên mạng điểm điểm. Khi một thực thể gửi một thông báo nhóm, các bộ định tuyến sẽ thực hiện theo chính sách nỗ lực tối đa để chuyển các thông báo này tới đích

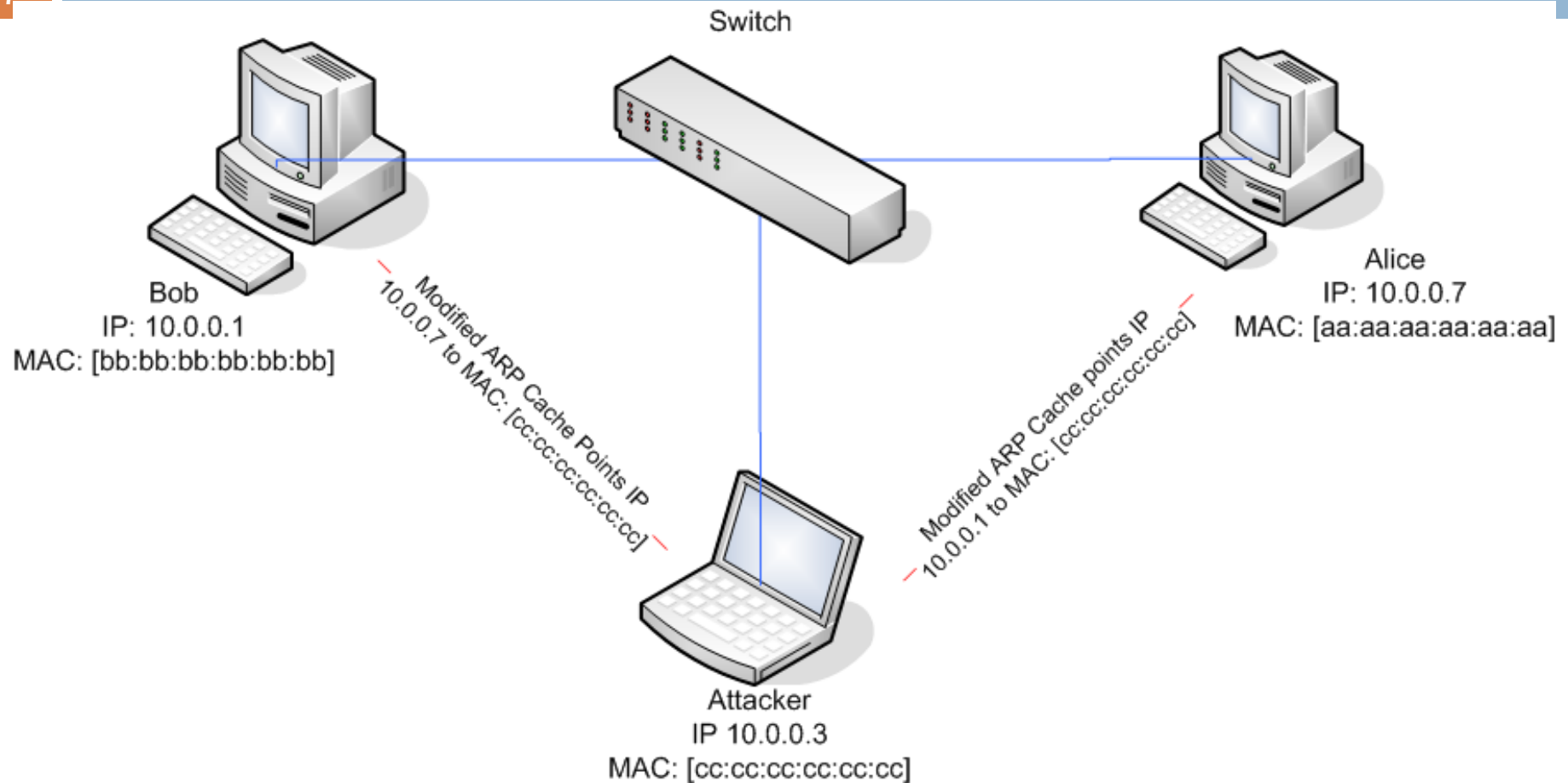
# VD: ARP





# ARP-Spoofing

17

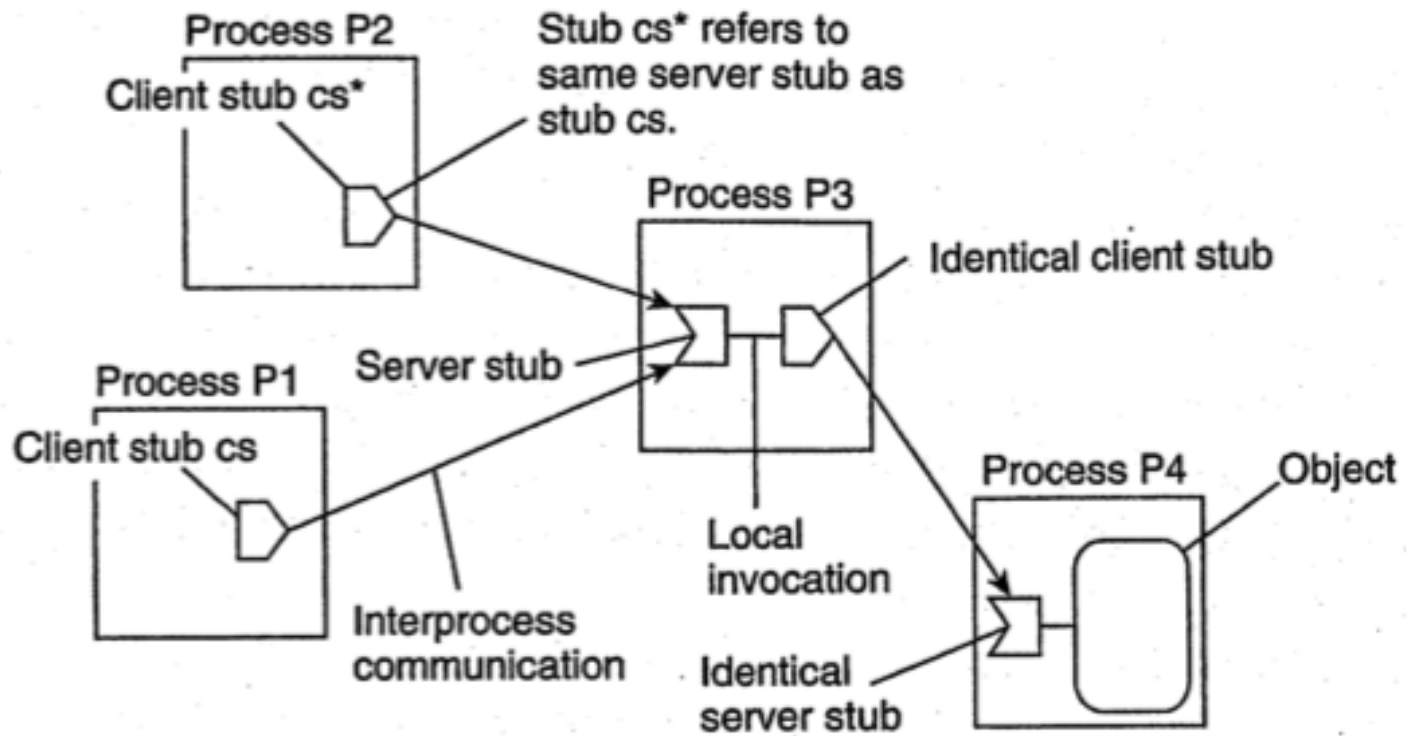


## 2.2.2. Chuyển tiếp con trỏ (Forwarding pointer)

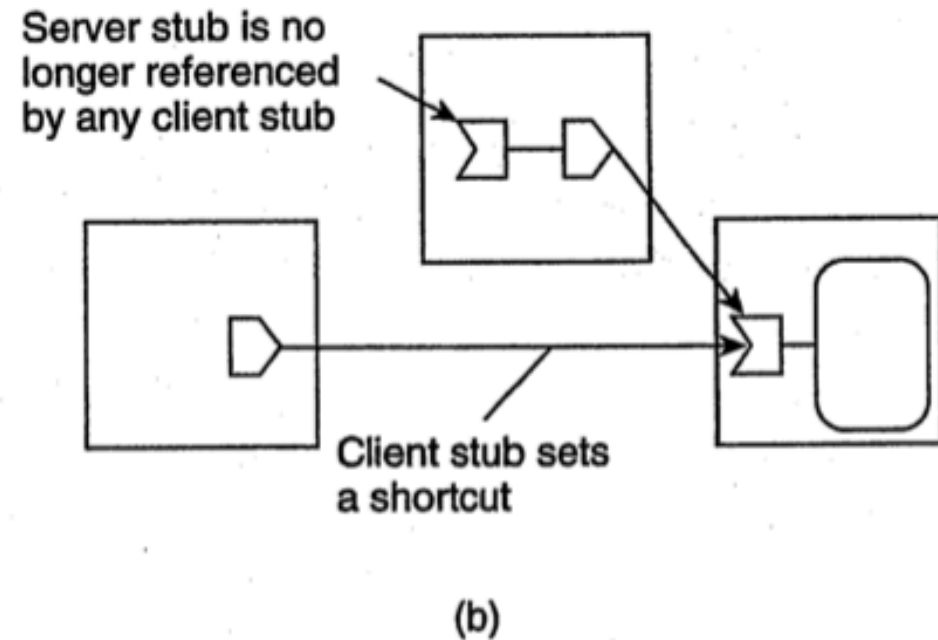
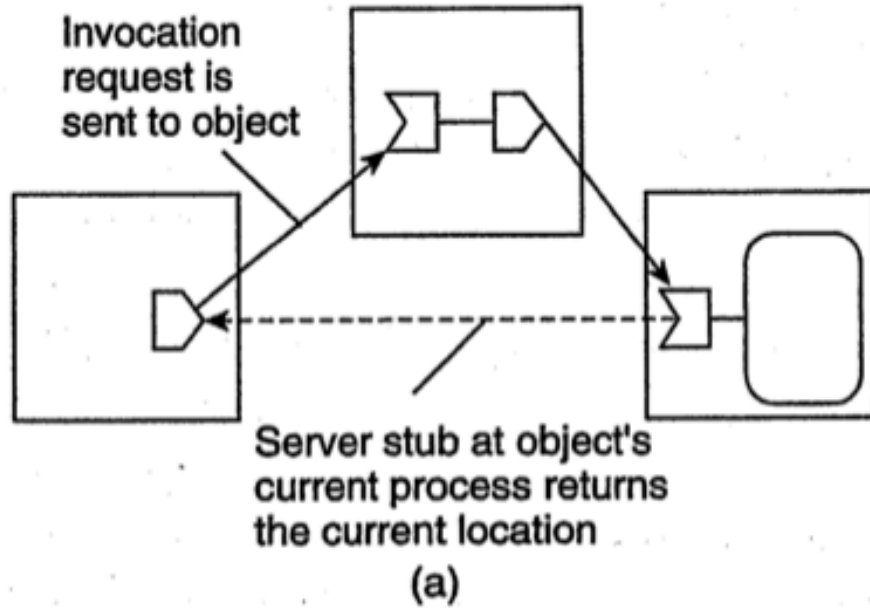
18

- Khi chuyển vị: để lại tham chiếu mới tại địa chỉ cũ
- Quản lý các pointer
- Duy trì các pointer
- Quản lý chuỗi các pointer
  - ▣ Client stub
  - ▣ Server stub

# Cơ chế hoạt động



# Tái định hướng con trỏ



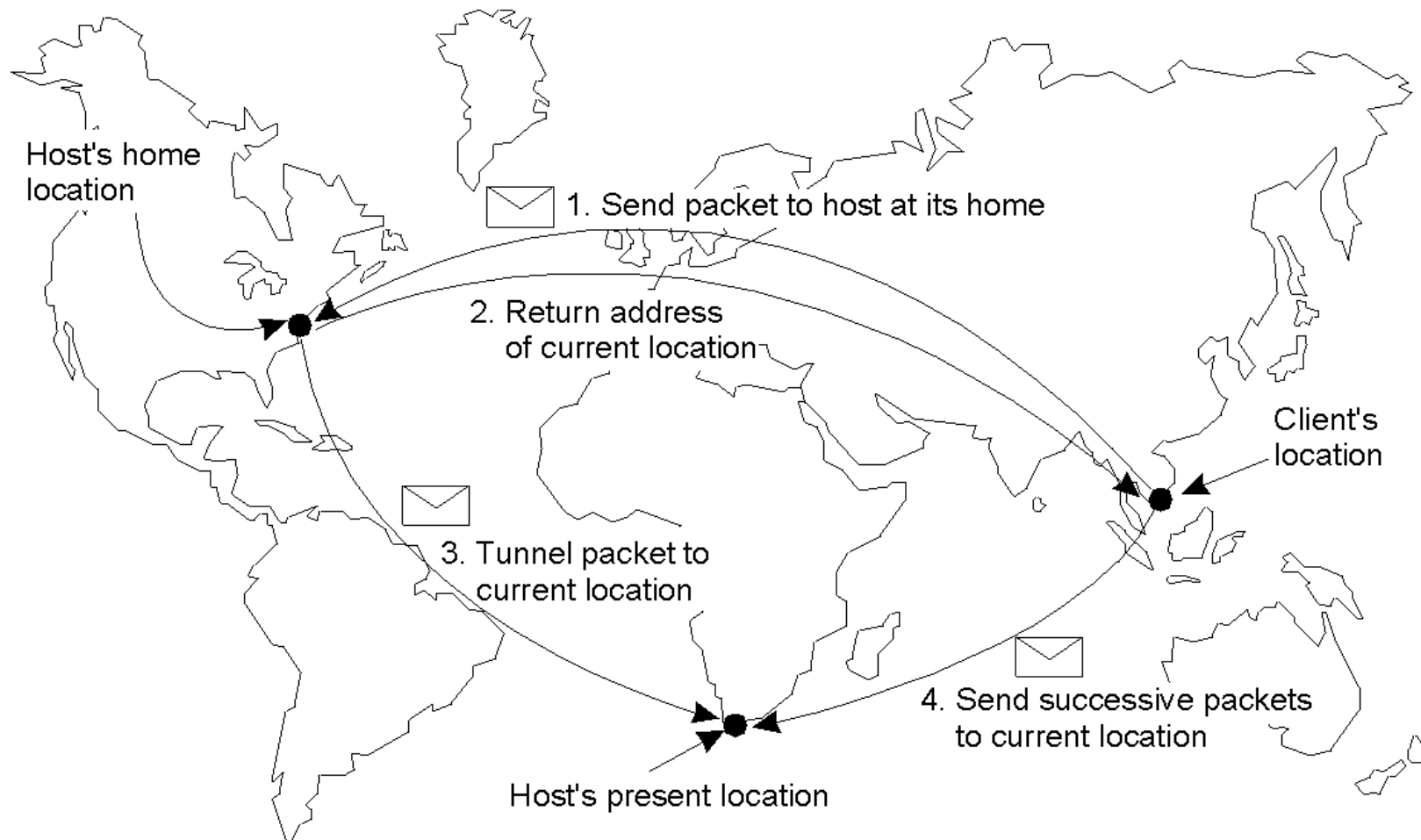
# Vấn đề

21

- Chuỗi dài vô hạn
  - ▣ Giải pháp: sử dụng các short cut
  - ▣ có khả năng có chuỗi không tham chiếu được
- Lưu trữ vô số các tham chiếu
  - ▣ Giải pháp: loại bỏ các tham chiếu
  - ▣ Khi nào có thể loại bỏ các tham chiếu
  - ▣ Bài toán toàn cục
- => Giải pháp home-based

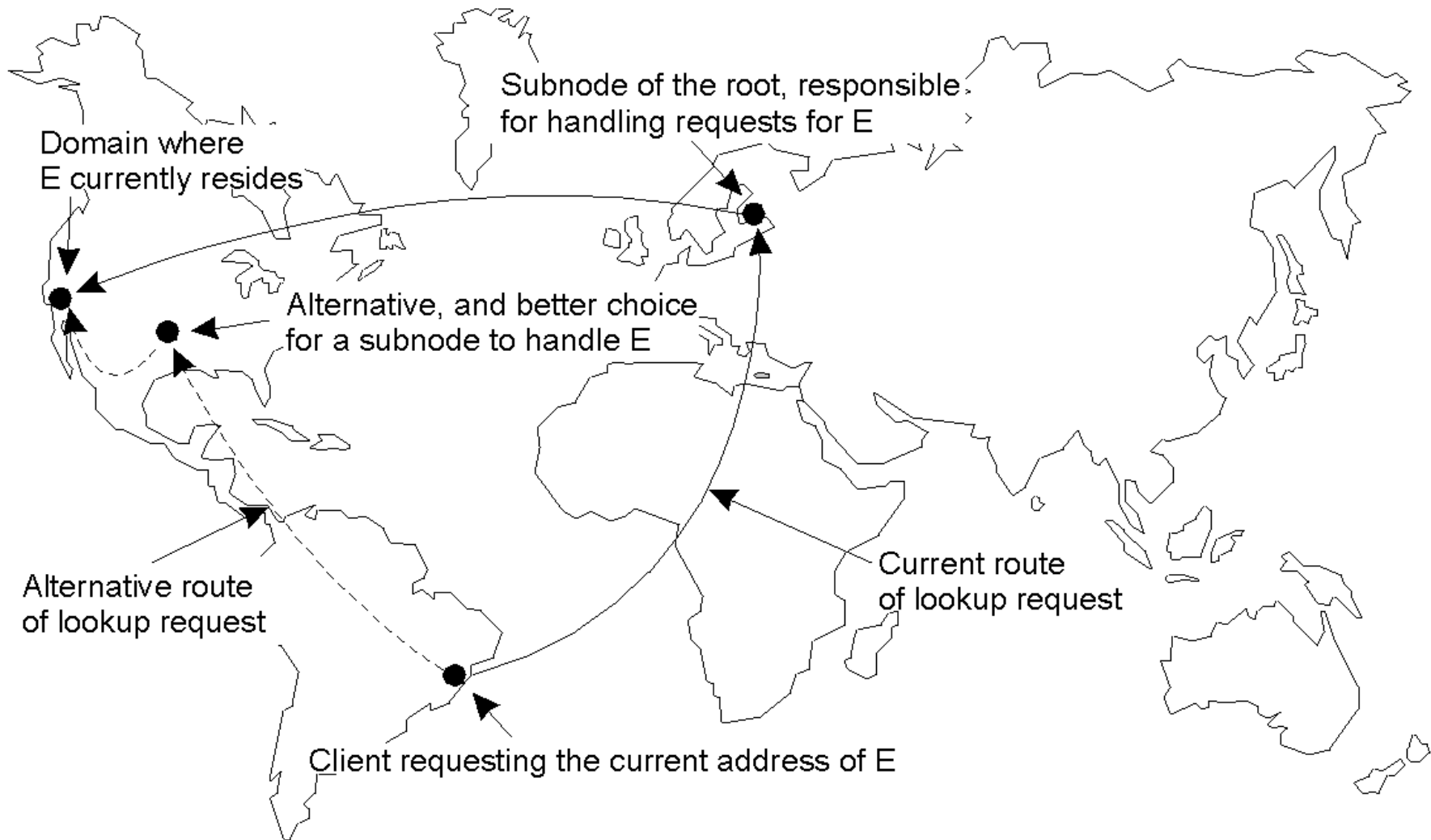
## 2.3. Giải pháp Home-based

22



# Vấn đề về qui mô

23



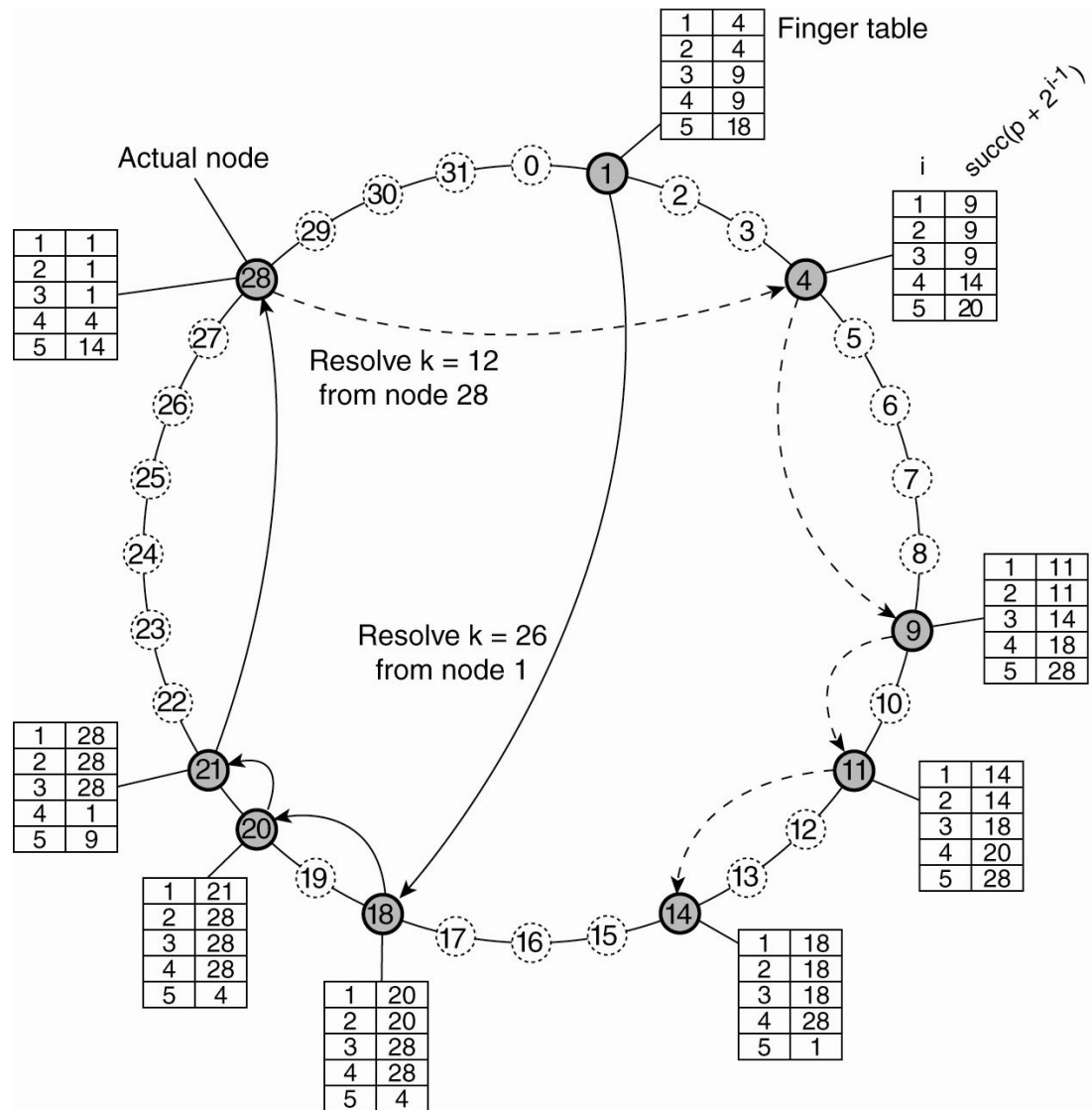
## 2.4. Giải pháp sử dụng hàm băm phân tán

24

- Xem xét hệ thống Chord
- Thiết lập vòng bằng các biến cục bộ  $prev(n)$  và  $succ(n)$
- Sử dụng bảng băm để xác định địa chỉ  $succ(k)$  của tên  $k$
- Với  $FT_p$  là finger table của node  $p$ :
$$FT_p[i] = succ(p + 2^{i-1})$$
- Khi cần tìm khóa  $k$ , node  $p$  sẽ gửi cho node  $q$ :
$$q = FT_p[j] \leq k < FT_p[j+1]$$
- Cập nhật bảng băm khi có các nút được thêm vào



# Bảng băm phân tán



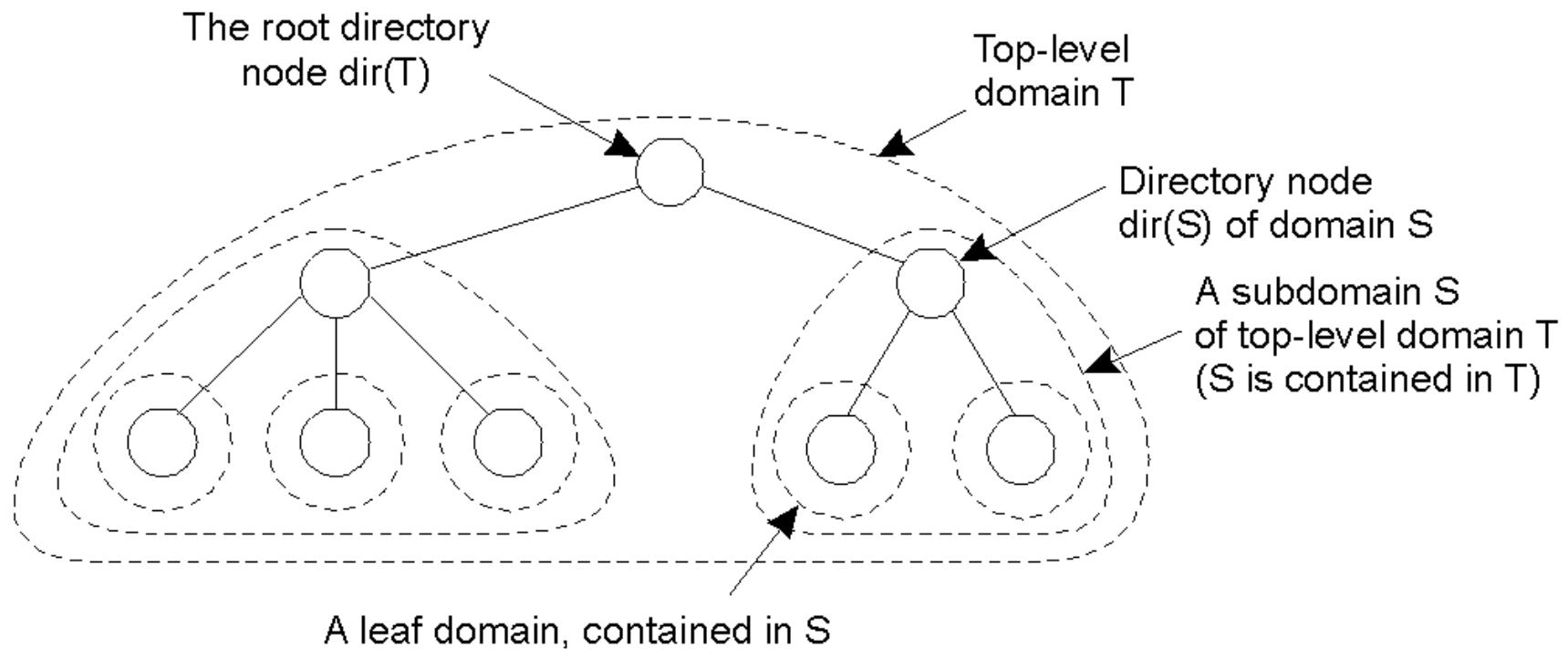
## 2.5. Giải pháp phân cấp

26

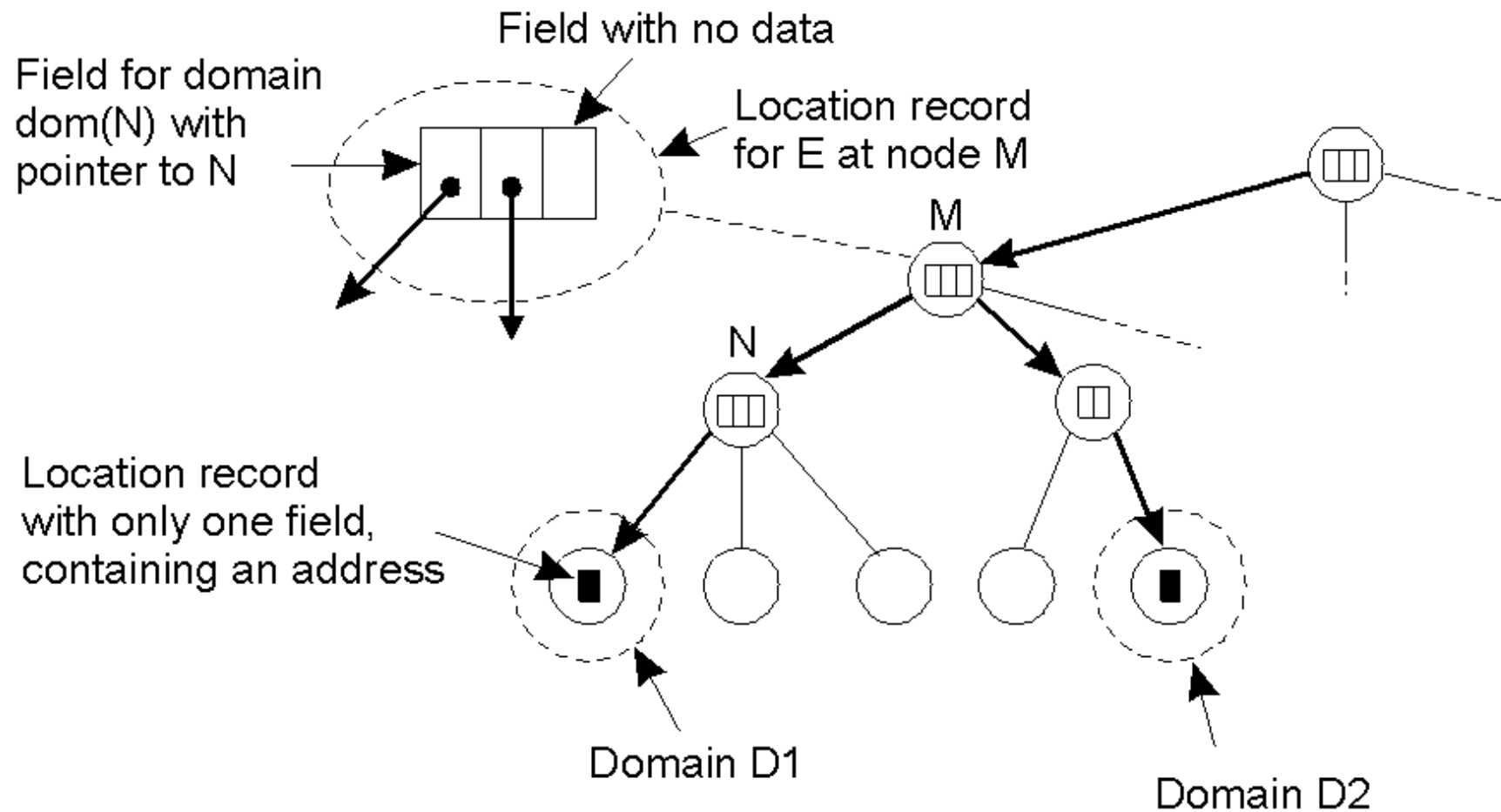
- Các domain phân cấp
- Domain lá: mạng cục bộ, cell
- Domain chứa các bản ghi (định danh, địa chỉ Domain con) của tất cả các nút

# Giải pháp phân cấp

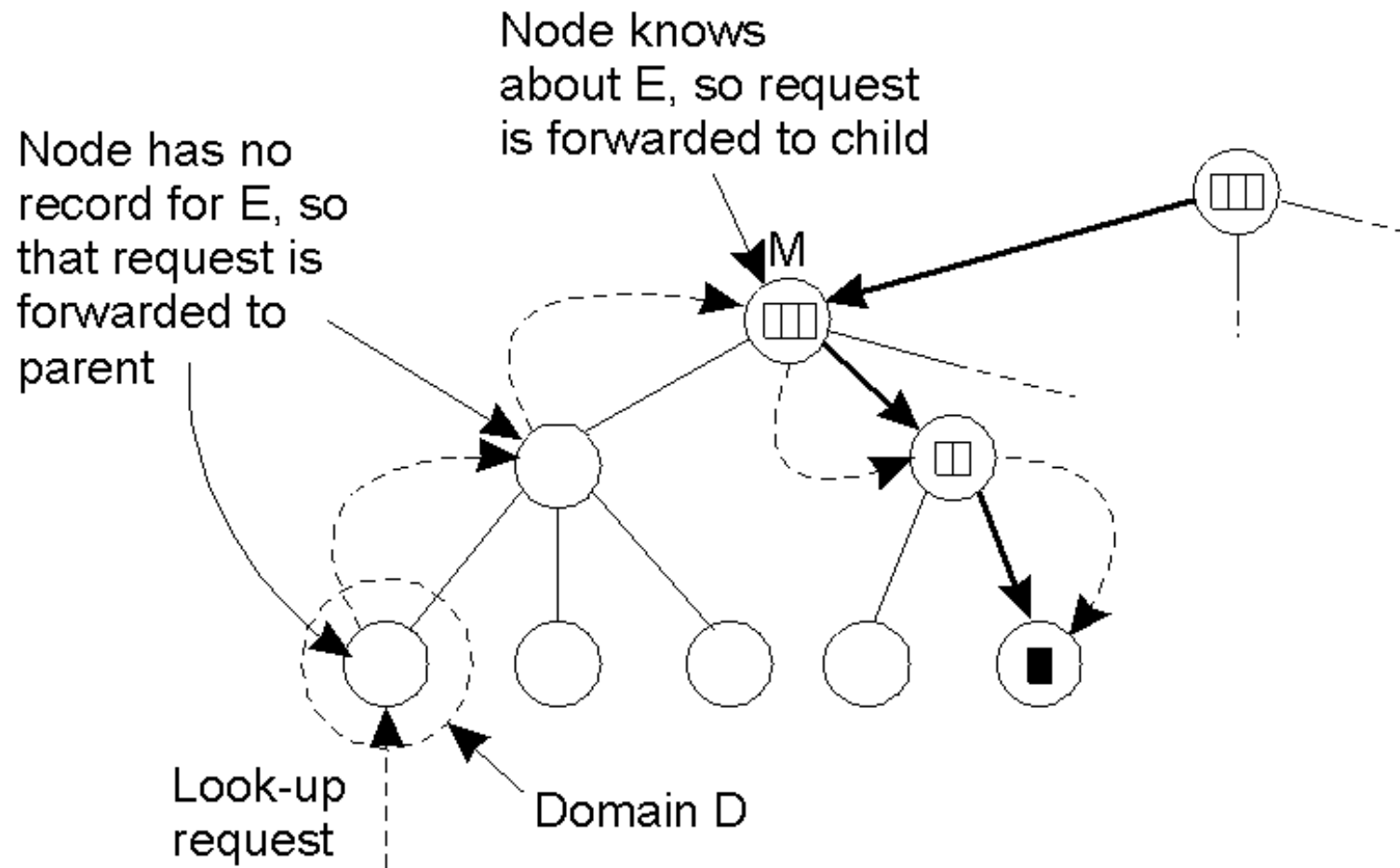
27



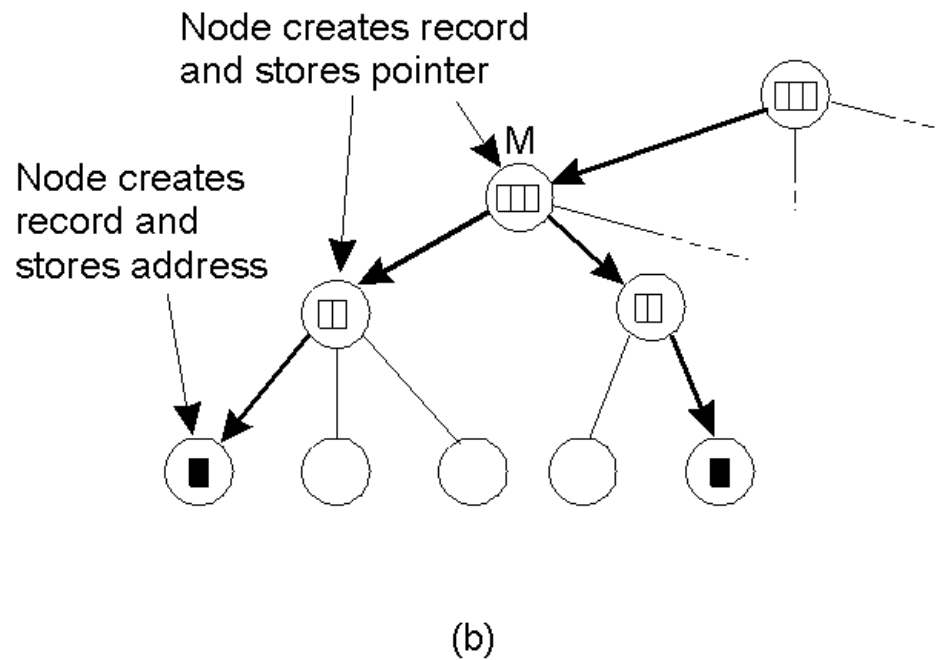
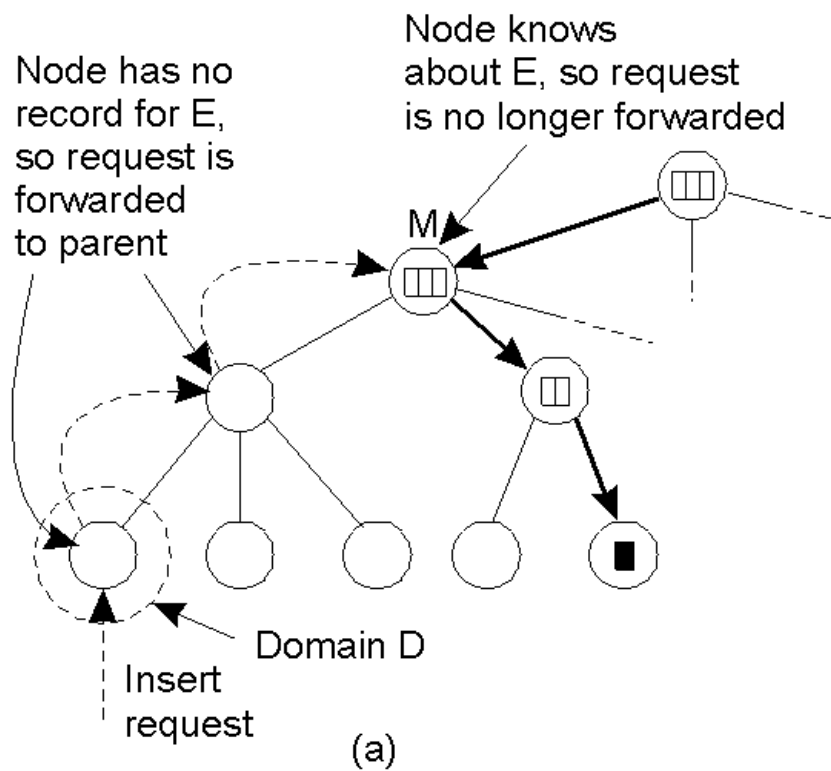
# Thực thể có 2 địa chỉ



# Tìm kiếm



# Cập nhật





# Loại bỏ các tên không sử dụng

32

- ❑ Đối tượng không được tham chiếu
- ❑ Con đếm tham chiếu
- ❑ Danh sách tham chiếu
- ❑ Các đối tượng không kết nối được

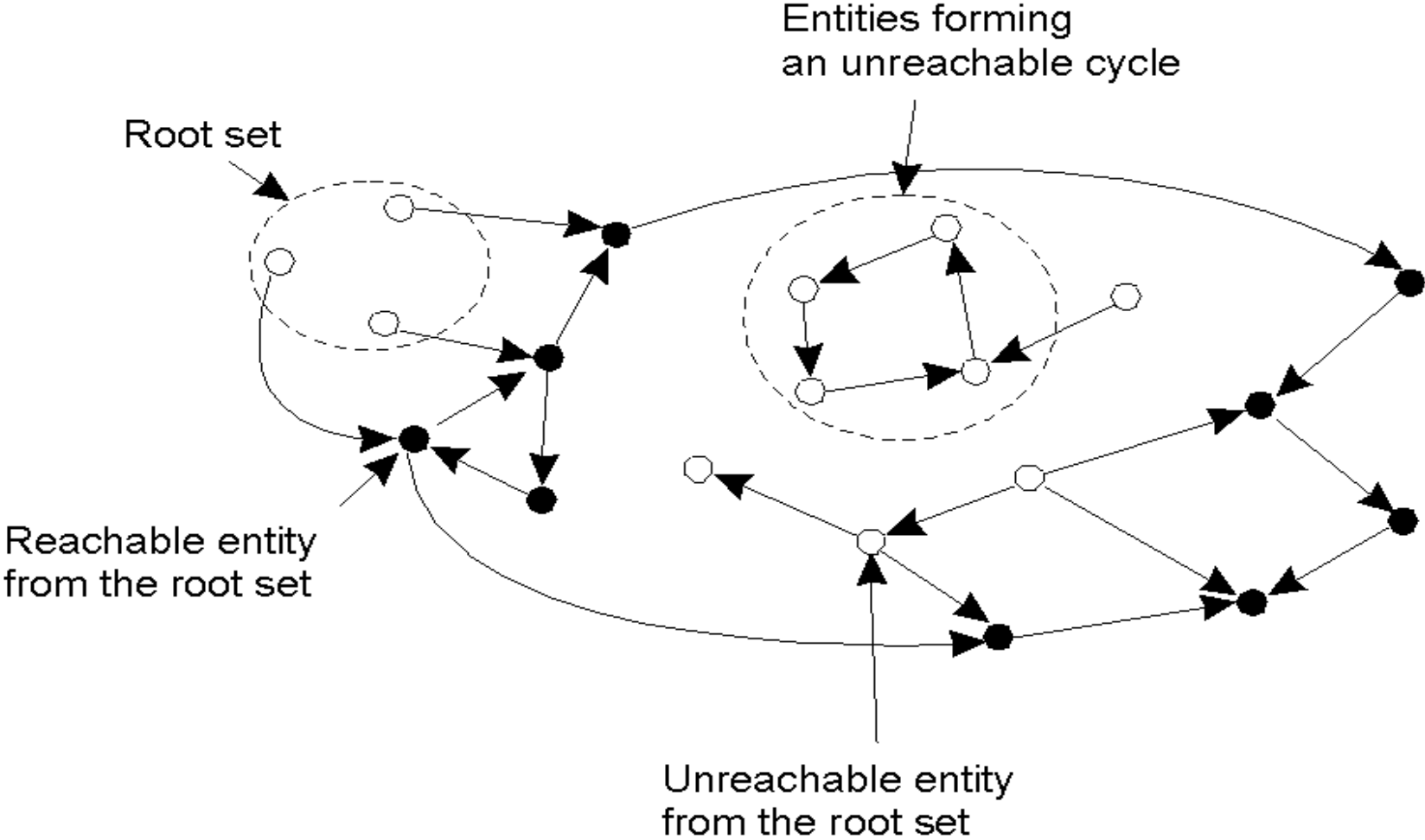


# Đối tượng không được tham chiếu

33

- Không sử dụng = loại bỏ
- Không tham chiếu  $\rightarrow$  không sử dụng
- Có tham chiếu ? Không sử dụng?
- Thực thể tham chiếu lẫn nhau và đều không được sử dụng 😊
- $\Rightarrow$  bài toán xác định các tham chiếu cần loại bỏ rất phức tạp

# Tham chiếu lẫn nhau

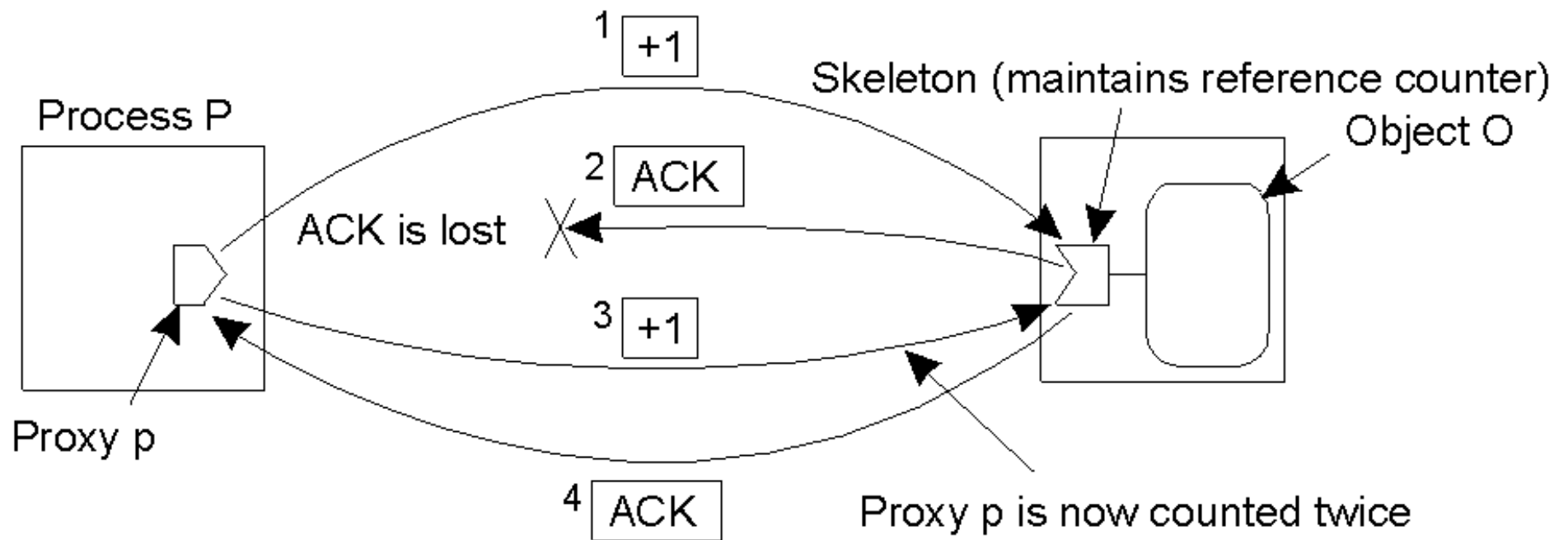


# Con đếm tham chiếu

35

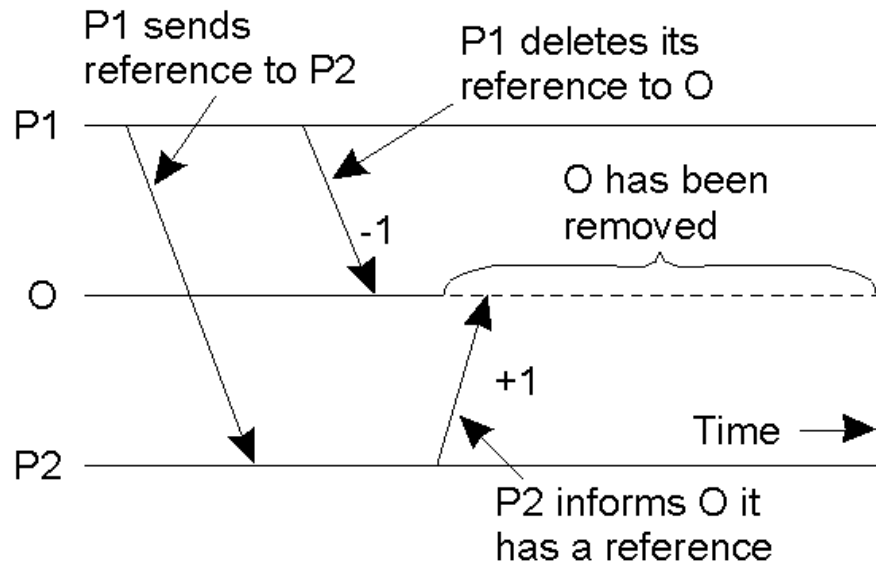
- Unix File System
- Đếm hai lần
- Đếm chậm

# Đếm 2 lần

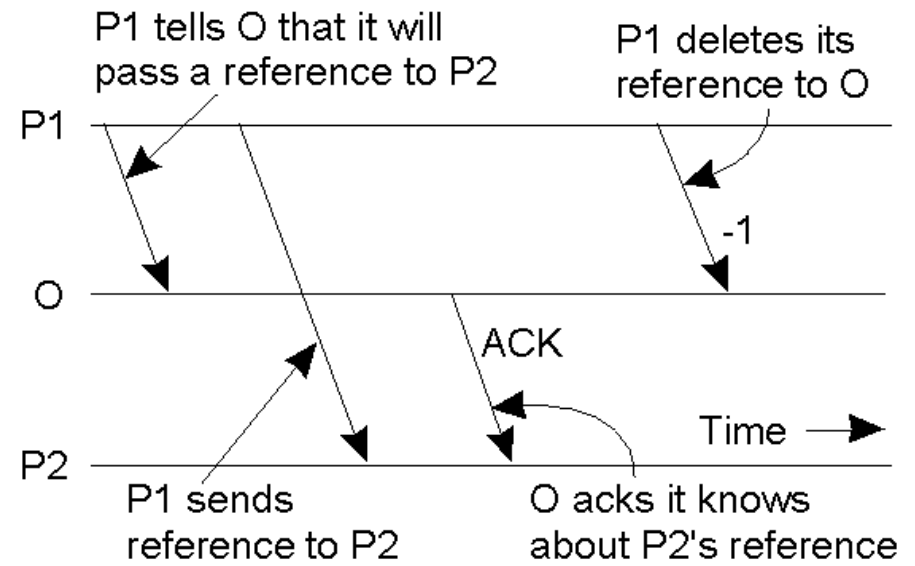


# Đếm chậm

37



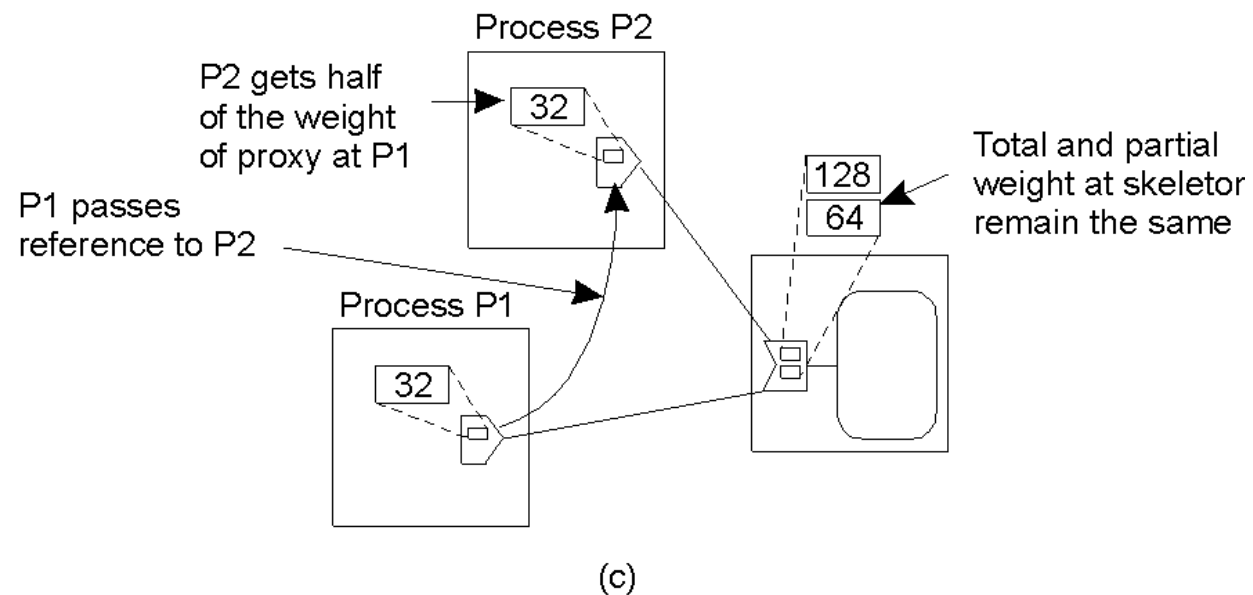
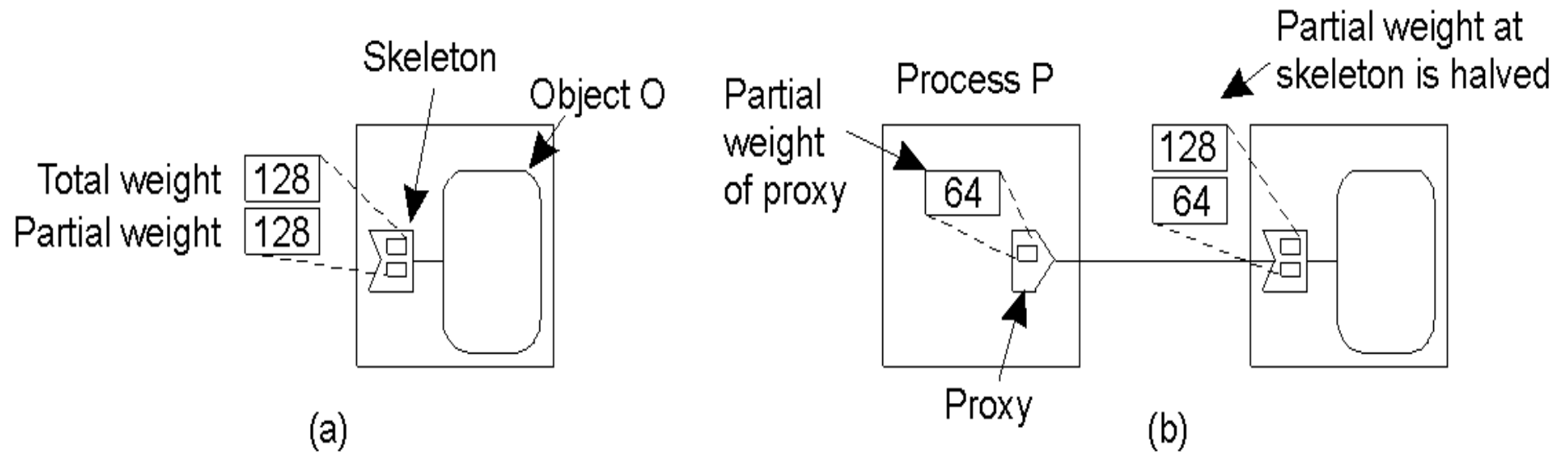
(a)



(b)

# Đếm tổng

3



# Nội dung

39

1. Tên, định danh, và địa chỉ
2. Không gian tên phẳng
3. Không gian tên có cấu trúc
4. Định danh theo thuộc tính

## 3. Không gian tên có cấu trúc

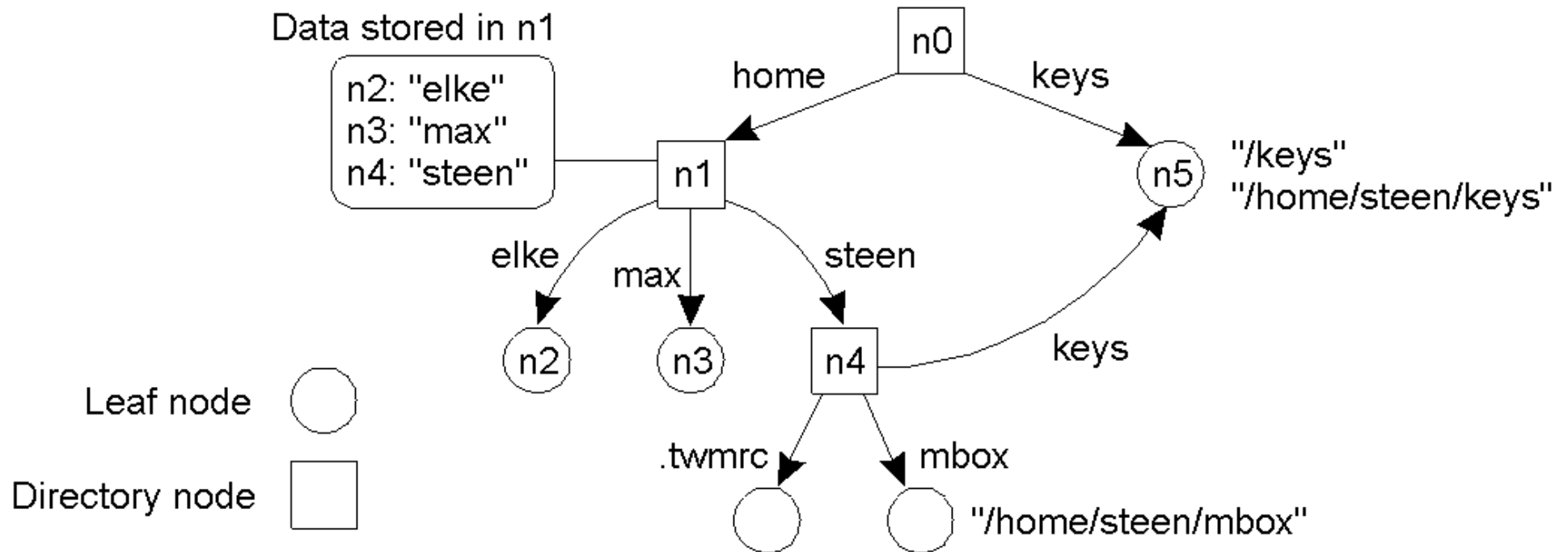
Cấu trúc không gian tên

Phân giải tên có cấu trúc

Dịch vụ tên có cấu trúc



# Không gian tên có cấu trúc



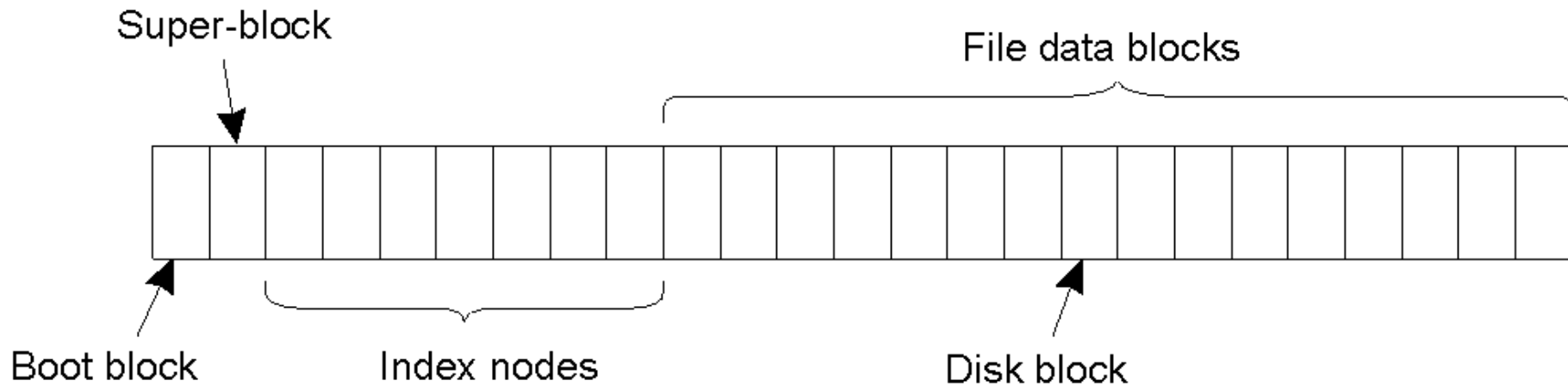
# Không gian tên có cấu trúc

42

- Nút lá:
  - ▣ Không có nhánh ra
  - ▣ Chứa địa chỉ của thực thể
- Nút thư mục:
  - ▣ Có các nhánh ra
  - ▣ Có chứa tên của các thực thể trong thư mục
- Đường dẫn tương ứng với một tên
- Tên toàn cục/đường dẫn tuyệt đối
- Tên cục bộ/đường dẫn tương đối

# Không gian tên (UNIX)

43



# Nội dung của tệp thư mục

44

- Theo hàng:
  - ▣ Chứa danh mục các tệp và thư mục con
- Theo cột
  - ▣ Tên
  - ▣ Chủ sở hữu/nhóm chủ sở hữu
  - ▣ Quyền truy cập
  - ▣ Inode

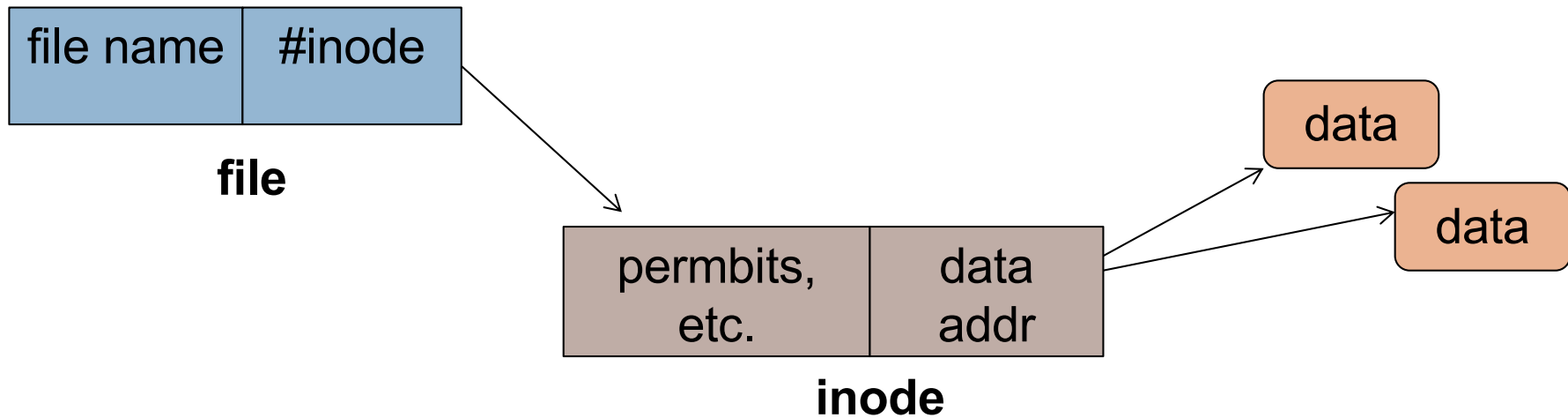
# Phân giải tên

45

- Tên
  - /home/trunghq/teaching/undergraduate/ds/
  - <home, trunghq,teaching, undegraduate, ds>
- Xác định địa chỉ
  - Cần biết được địa chỉ của thư mục gốc
  - Đọc nội dung của thư mục gốc
  - Xác định địa chỉ của home
  - Đọc nội dung của home
  - Xác định địa chỉ của trunghq
  - .....
  - Đọc nội dung của ds
- Luôn cần một điểm cố định: nguyên tắc bao đóng

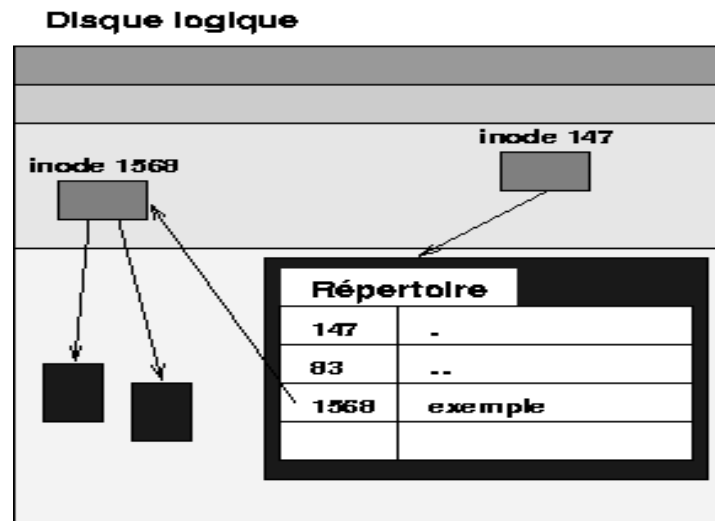
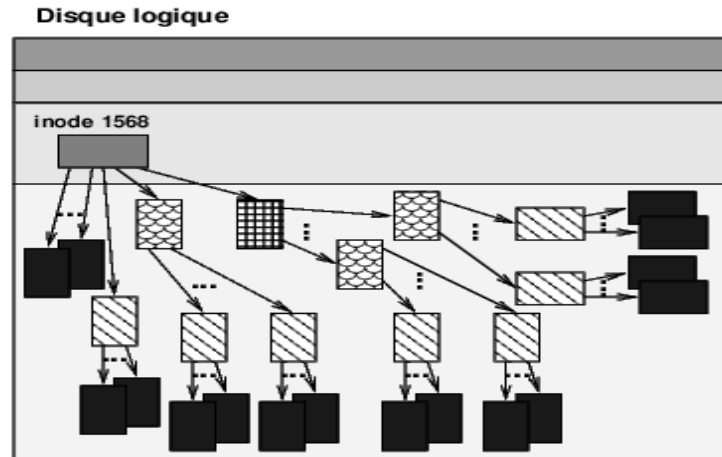
# Inode, soft link and hard link

- Một file trong Unix:



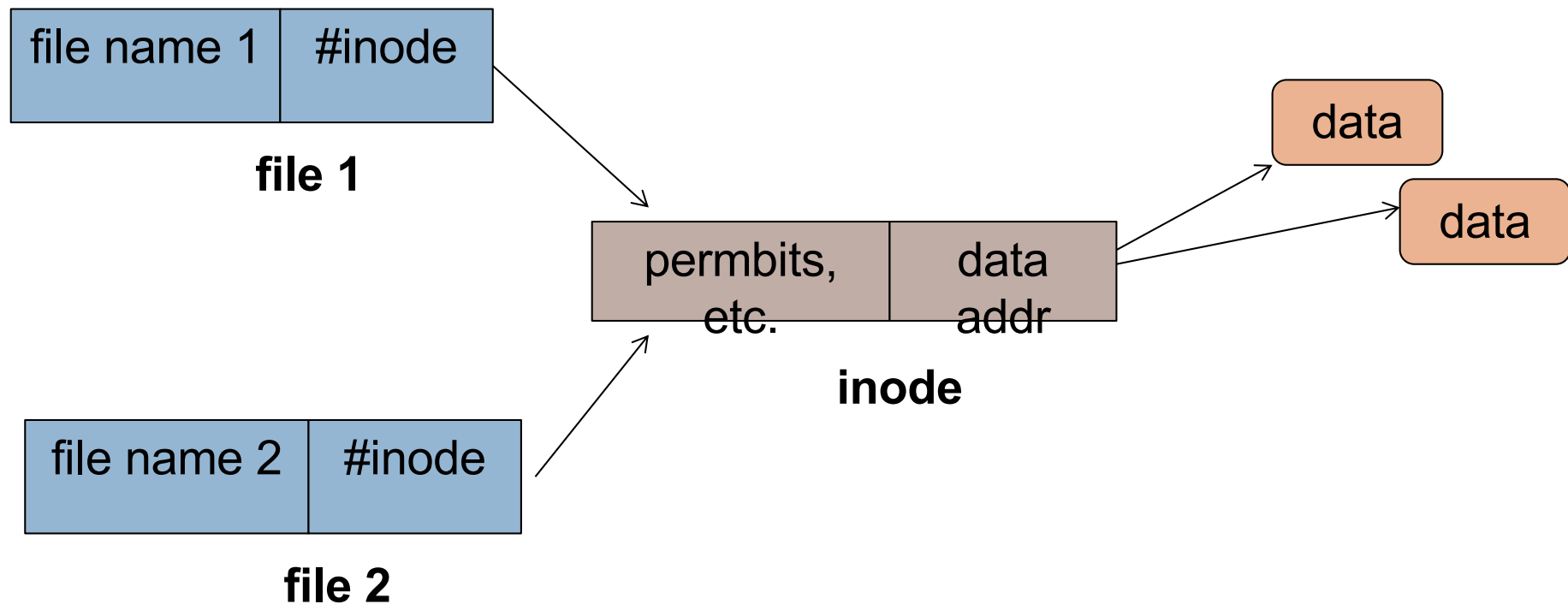
# Thư mục

- Một thư mục là một tập với nội dung là một bảng liên kết
  - ▣ một liên kết gắn một tên tệp với một inode của hệ thống tệp



# Inode, soft link and hard link (cont.)

- Liên kết vật lý (hard link)

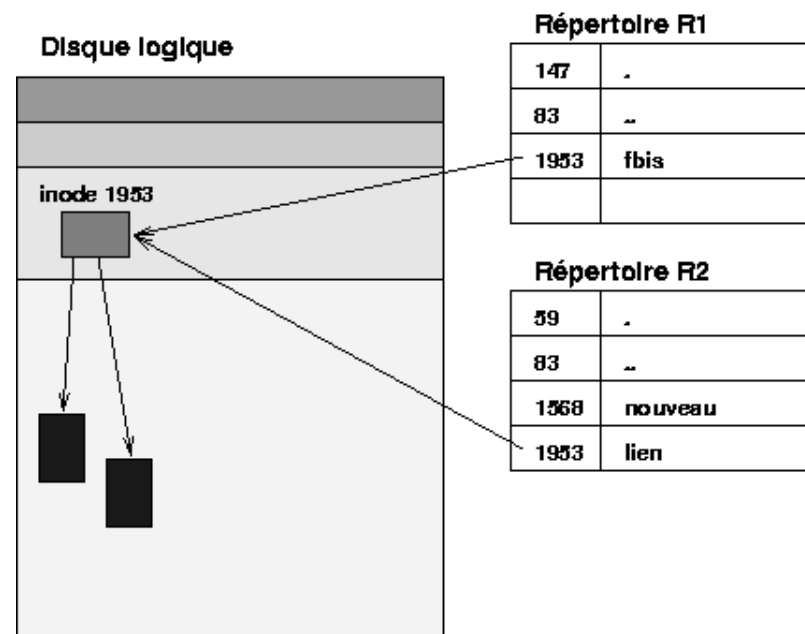




# Liên kết vật lý (cont.)

- Một liên kết vật lý là một quan hệ giữa tên tệp trong thư mục với một inode
- Có thể có nhiều liên kết vật lý đến cùng một inode
- Lệnh **ln** cho phép tạo một liên kết vật lý đến một inode (tệp) đã tồn tại
  - tệp mới chia sẻ cùng inode và khối dữ liệu của tệp ban đầu

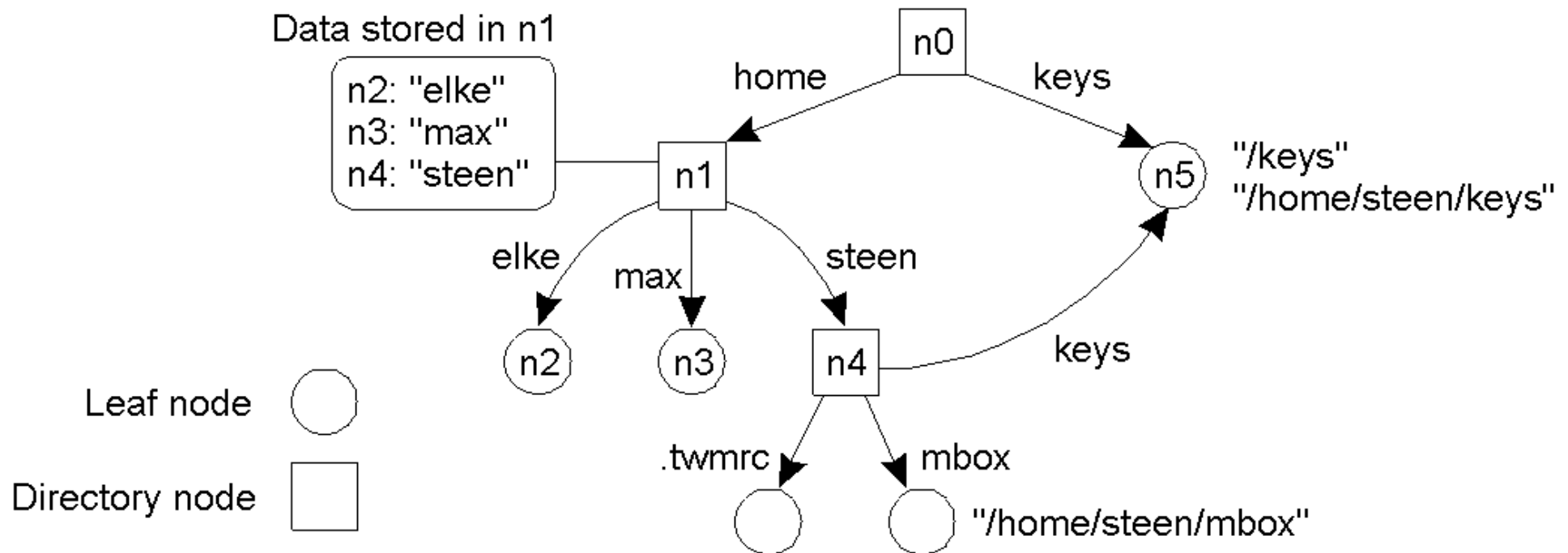
\$ln fbis lien



# Liên kết vật lý (cont.)

50

- 1 nút có 2 địa chỉ đường dẫn



# Liên kết vật lý (cont.)

- Số liên kết vật lý đến một inode có thể được xem bằng lệnh `ls -l`

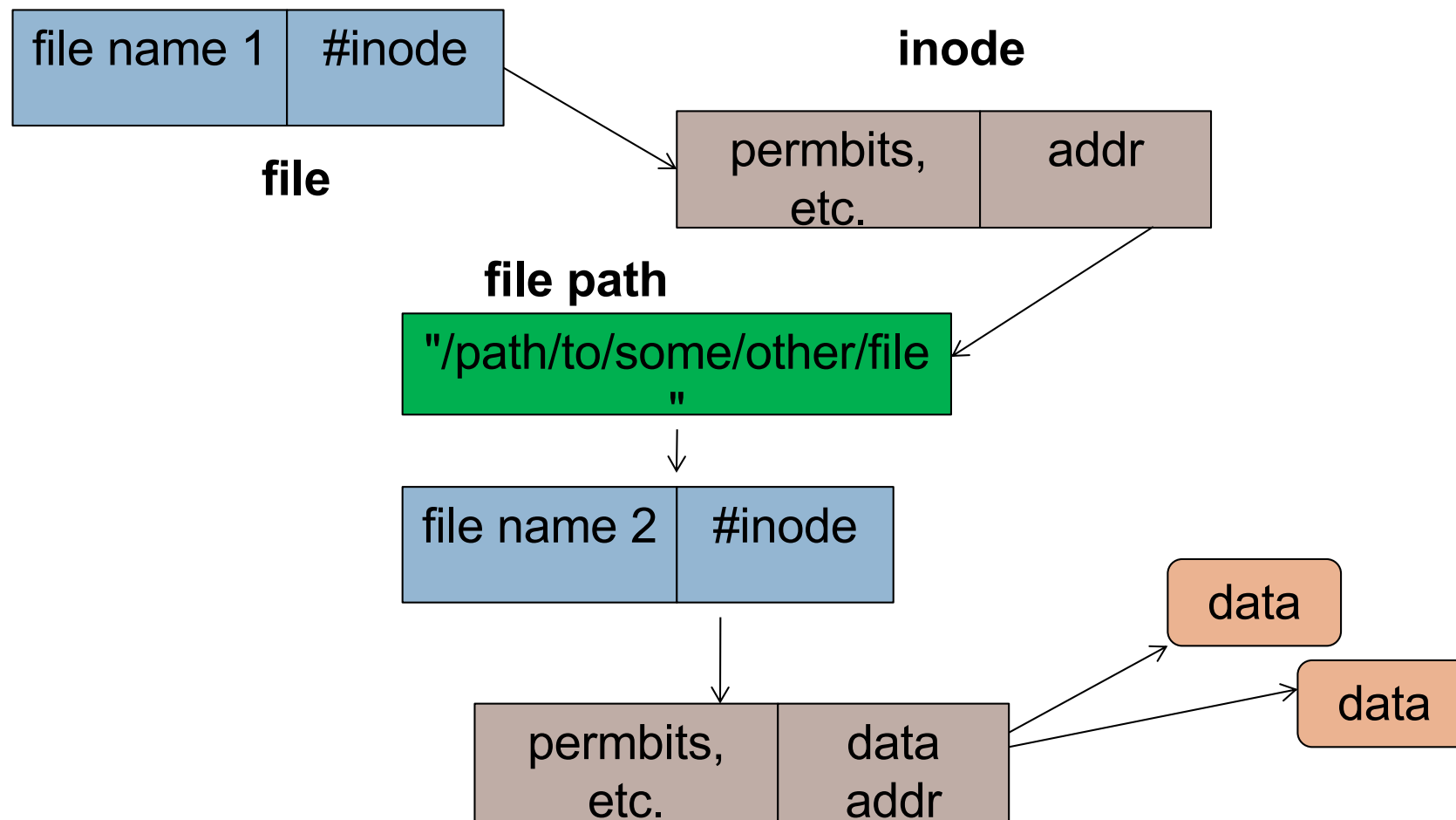
```
$ ls -l
```

```
-rw-rw-r-- 1 tuananh user1 0 Nov 12 15:19 file  
drwxr-xr-x 2 tuananh user1 4096 Dec 14 17:50 dir
```

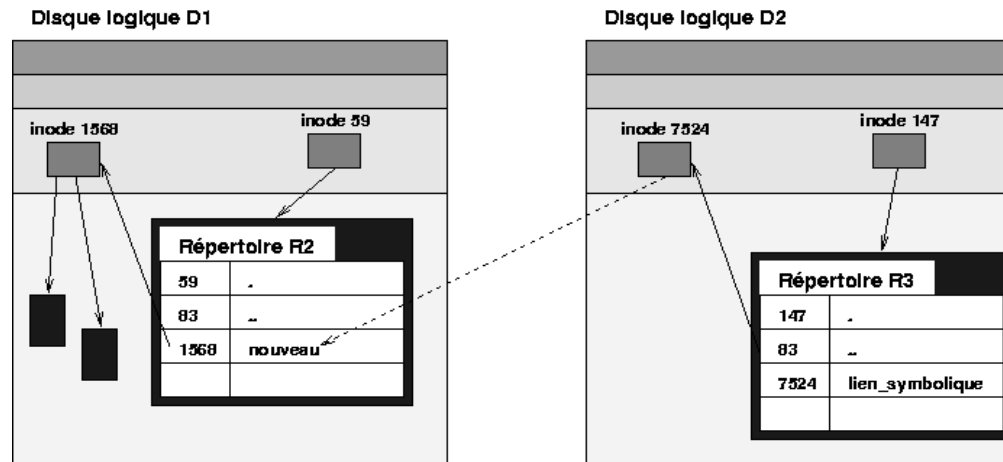
- Tại sao với một thư mục luôn có ít nhất 2 liên kết vật lý?
- Xoá một tệp (lệnh `rm`) đồng nghĩa với xoá một liên kết
  - Nếu là liên kết vật lý cuối cùng trở đến inode được xoá thì các khối liên quan đến inode cũng được xoá theo

# Inode, soft link and hard link (cont.)

- Liên kết biểu tượng (soft link)



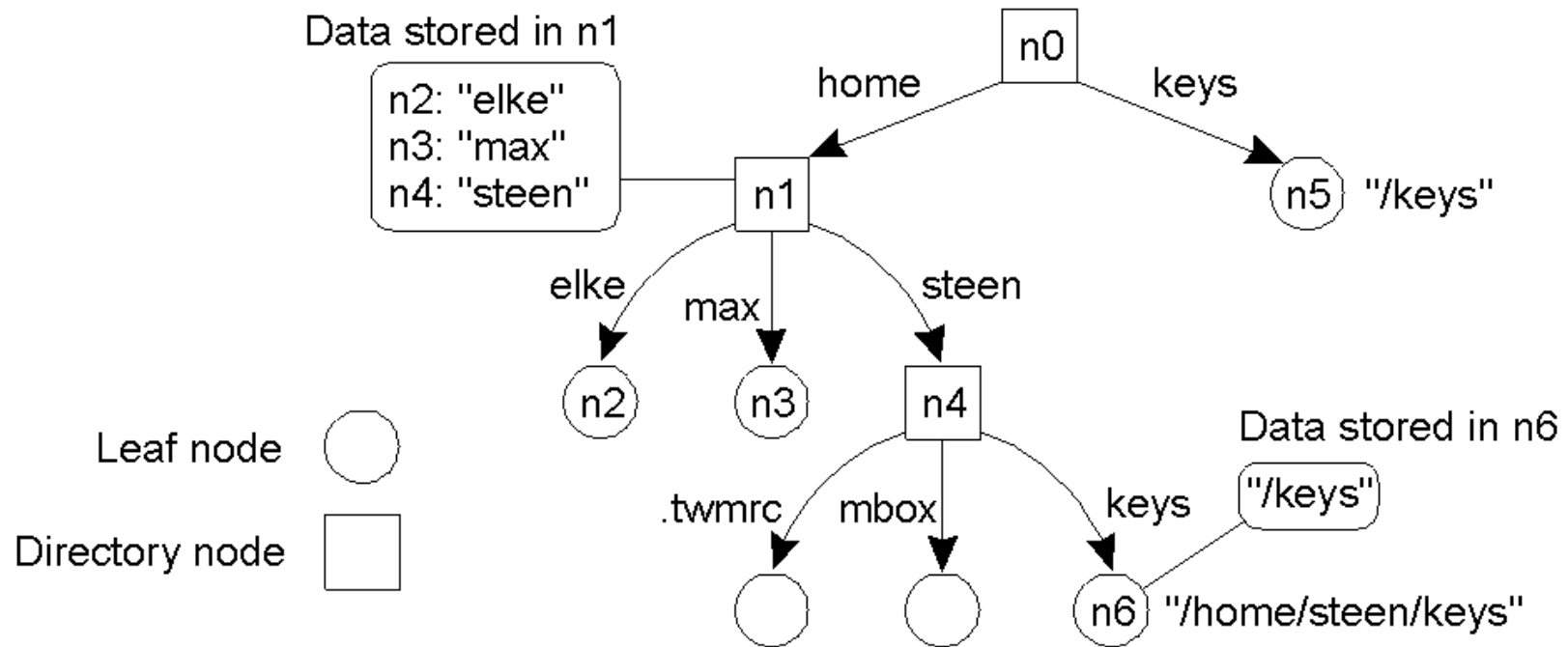
# Liên kết biểu tượng



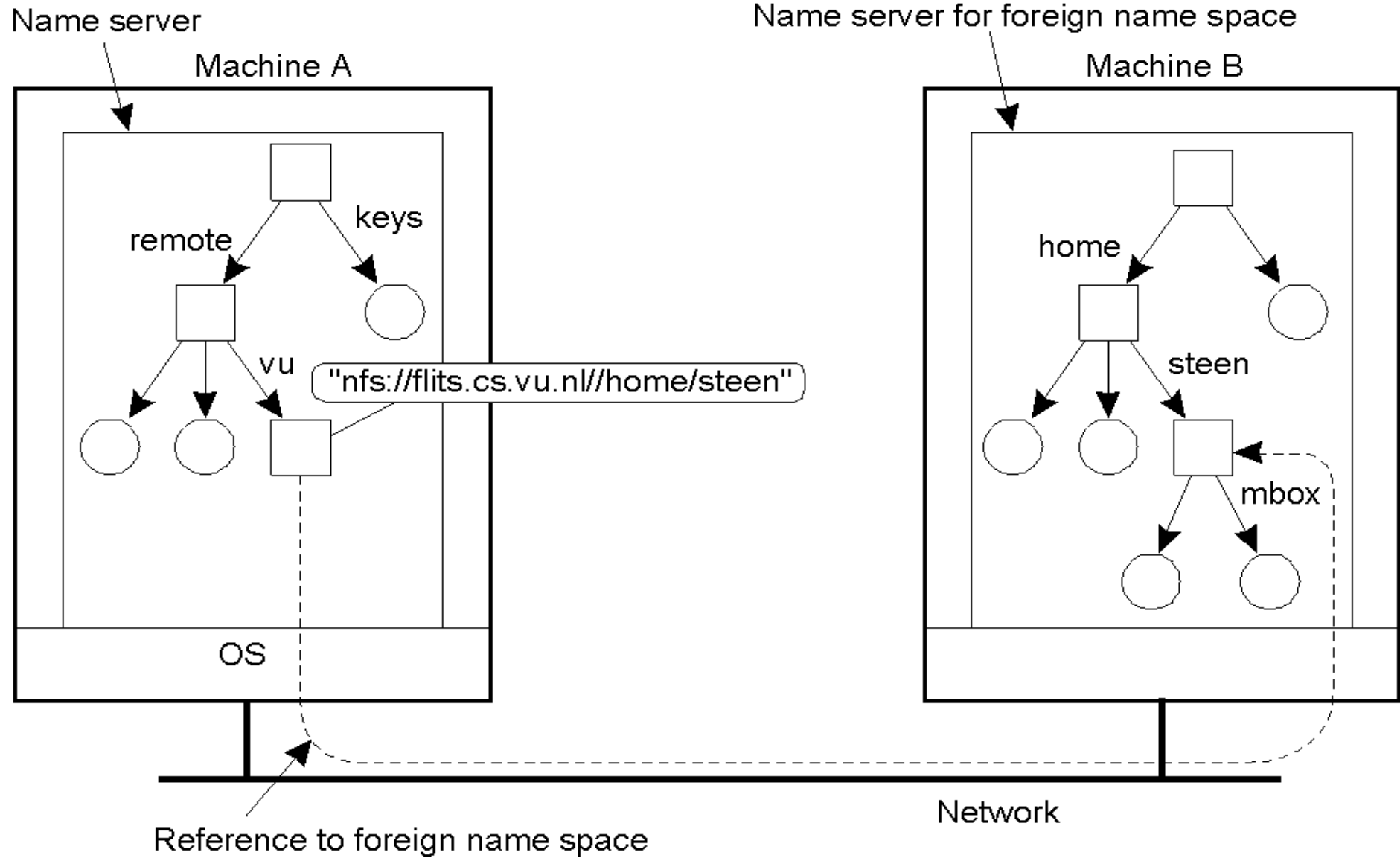
- In -s R2/nouveau R3/lien\_symbolique
  - ▣ khi tạo liên kết biểu tượng (tùy chọn -s) một inode mới được tạo ra
  - ▣ inode này chứa tên (dạng tuyệt đối hay tương đối) của phân tử được trỏ tới

# Liên kết biểu tượng

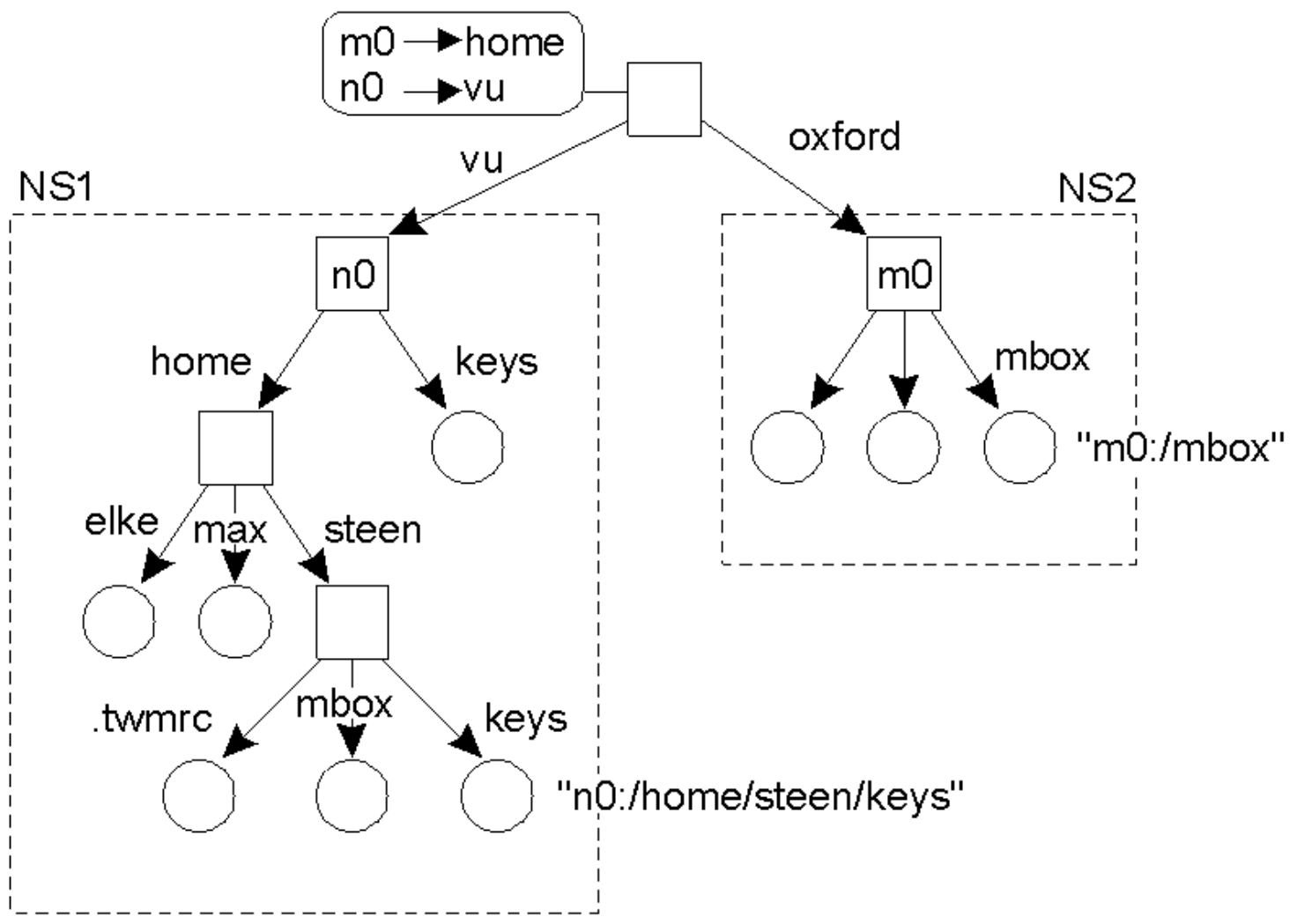
54



# Mounting



# Merging





# Dịch vụ tên

57

- Chức năng
  - ▣ Đăng ký, loại bỏ các định danh
  - ▣ Phân giải các định danh
  - ▣ Tìm kiếm các định danh
- Tính chất
  - ▣ Phân tán trên nhiều máy chủ khác nhau
- Cần phân tán không gian tên

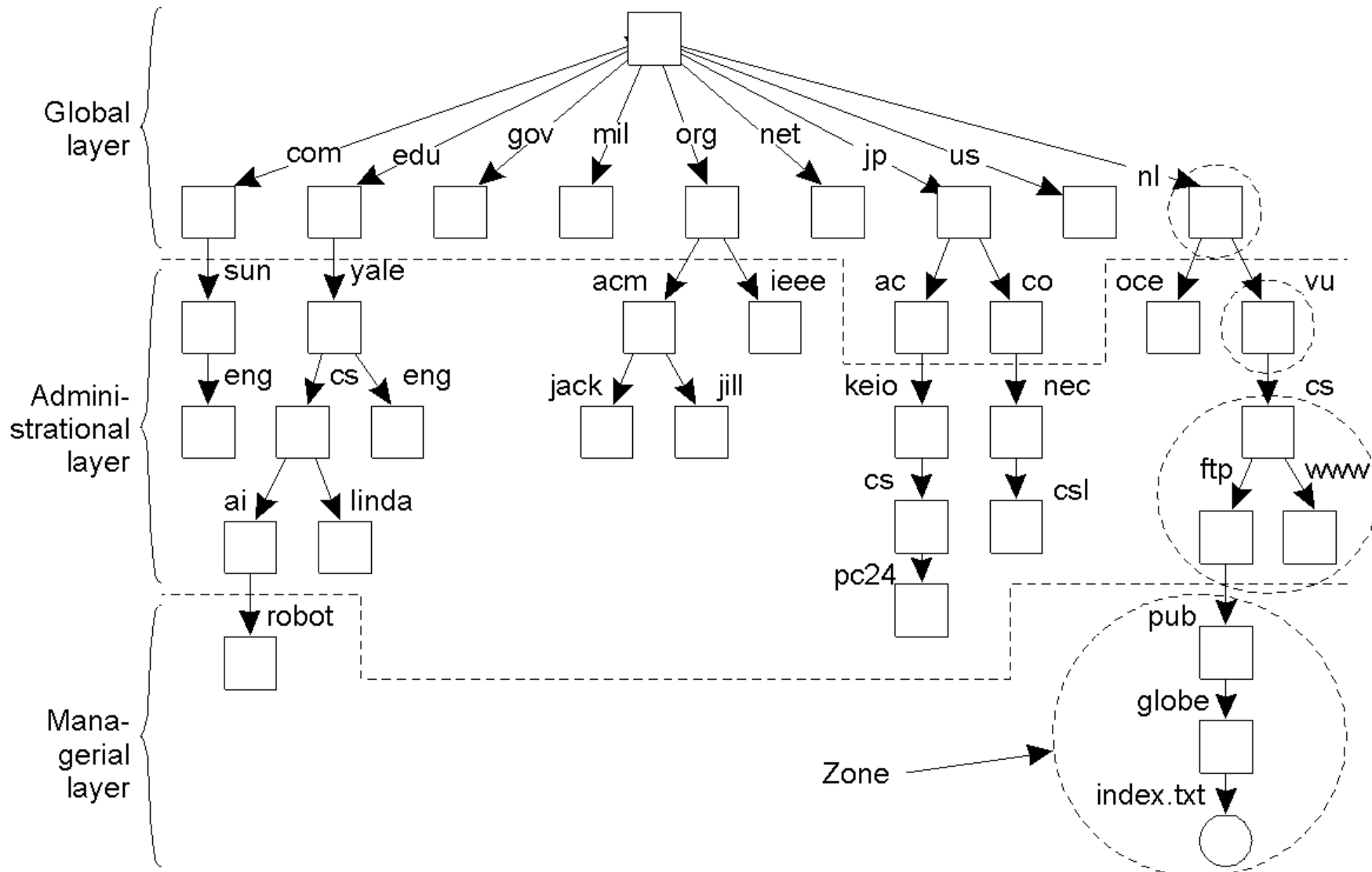
# Phân tán không gian tên

58

- Phân cấp
- Mức toàn thể
- Mức quản trị
- Mức quản lý
- Yêu cầu về hiệu năng khác nhau

# Phân tán không gian tên

5



# Phân tán không gian tên

60

Item	Toàn cục	Quản lý	Quản trị
Phạm vi địa lý	Địa cầu	Nước /tổ chức lớn	Tổ chức nhỏ /Thành viên
Số lượng nút (servers)	Ít (a-m)	Nhiều (số nước, số tổ chức toàn cầu)	Rất nhiều
Thời gian đáp ứng	s	ms	Tức khắc
Phổ biến thay đổi	Chậm	Ngay	Immediate
Số lượng bản sao	Nhiều	Không có hoặc ít	Không có
Bộ đệm trên client	Có	Có	Có/Không

# Cài đặt cơ chế phân giải tên

61

- Phụ thuộc vào việc phân tán không gian tên
- Phân giải tên máy khác
- Đệ qui
- Không đệ qui

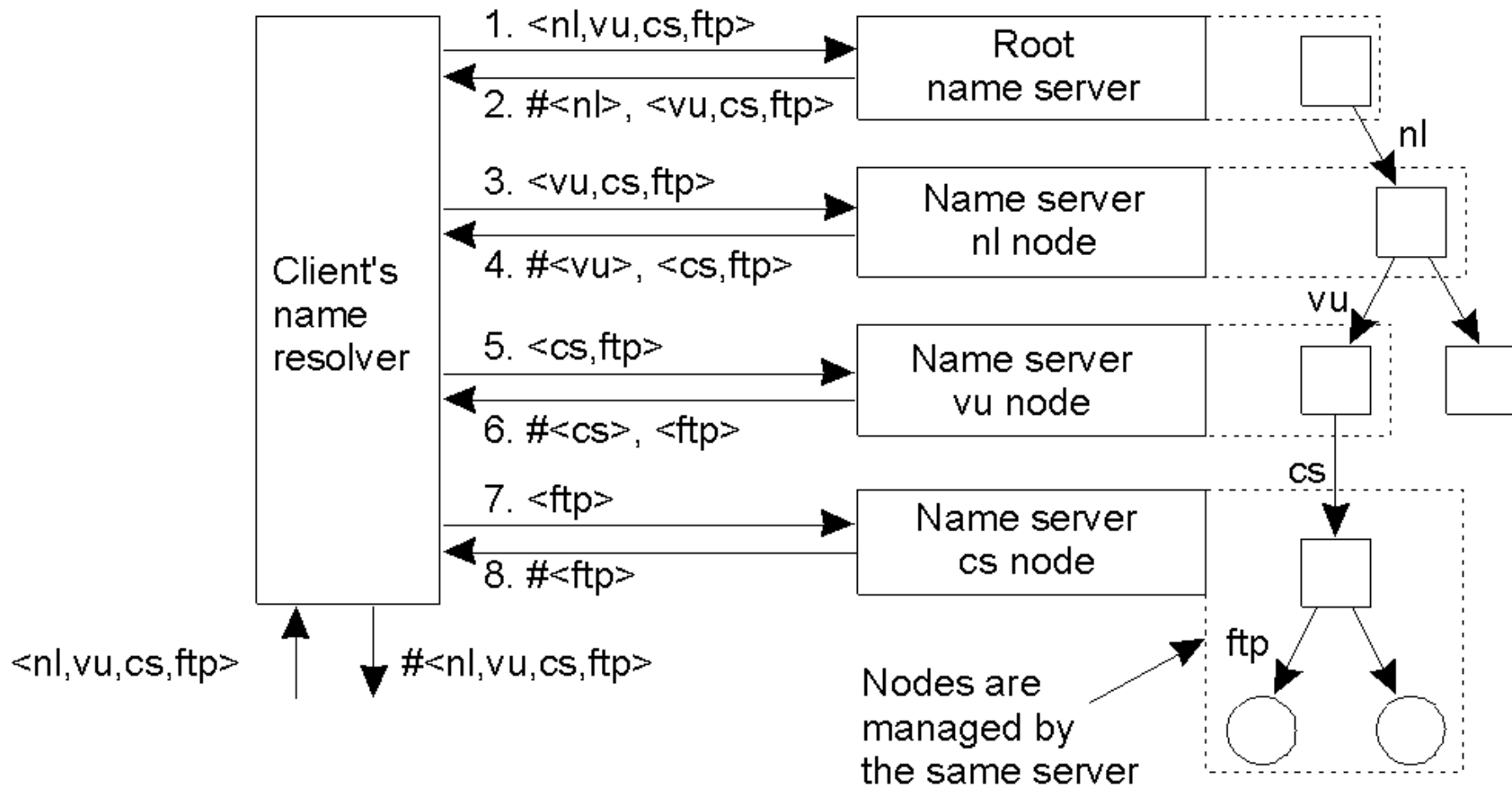
# Chức năng của các server có liên quan

62

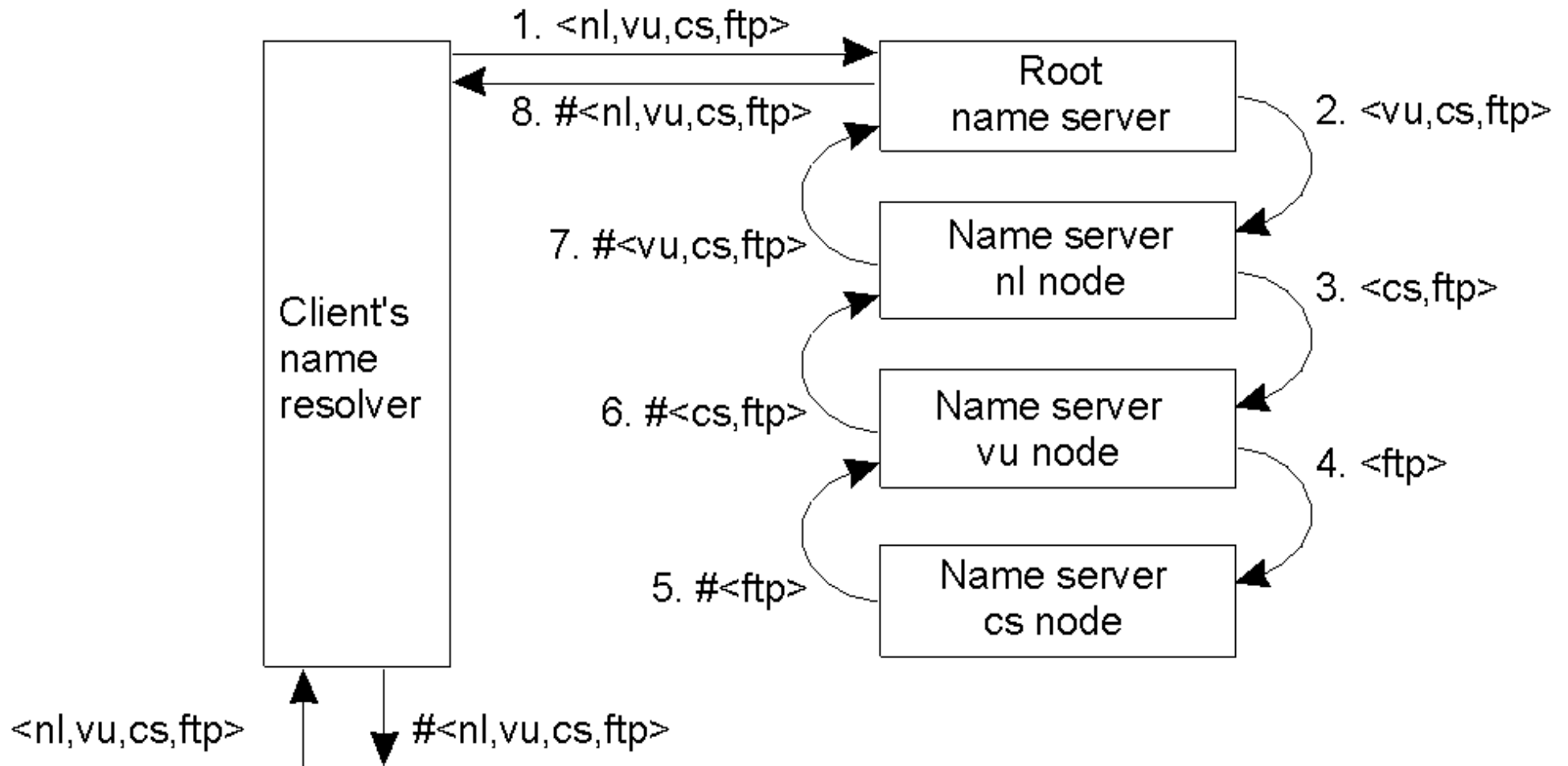
Server for node	Should resolve	Looks up	Passes to child	Receives and caches	Returns to requester
cs	<ftp>	#<ftp>	--	--	#<ftp>
vu	<cs,ftp>	#<cs>	<ftp>	#<ftp>	#<cs> #<cs, ftp>
nl	<vu,cs,ftp>	#<vu>	<cs,ftp>	#<cs> #<cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp>
root	<ni,vu,cs,ftp>	#<nl>	<vu,cs,ftp>	#<vu> #<vu,cs> #<vu,cs,ftp> >	#<nl> #<nl,vu> #<nl,vu,cs> #<nl,vu,cs,ftp>

# Phân giải tên không đệ quy

63



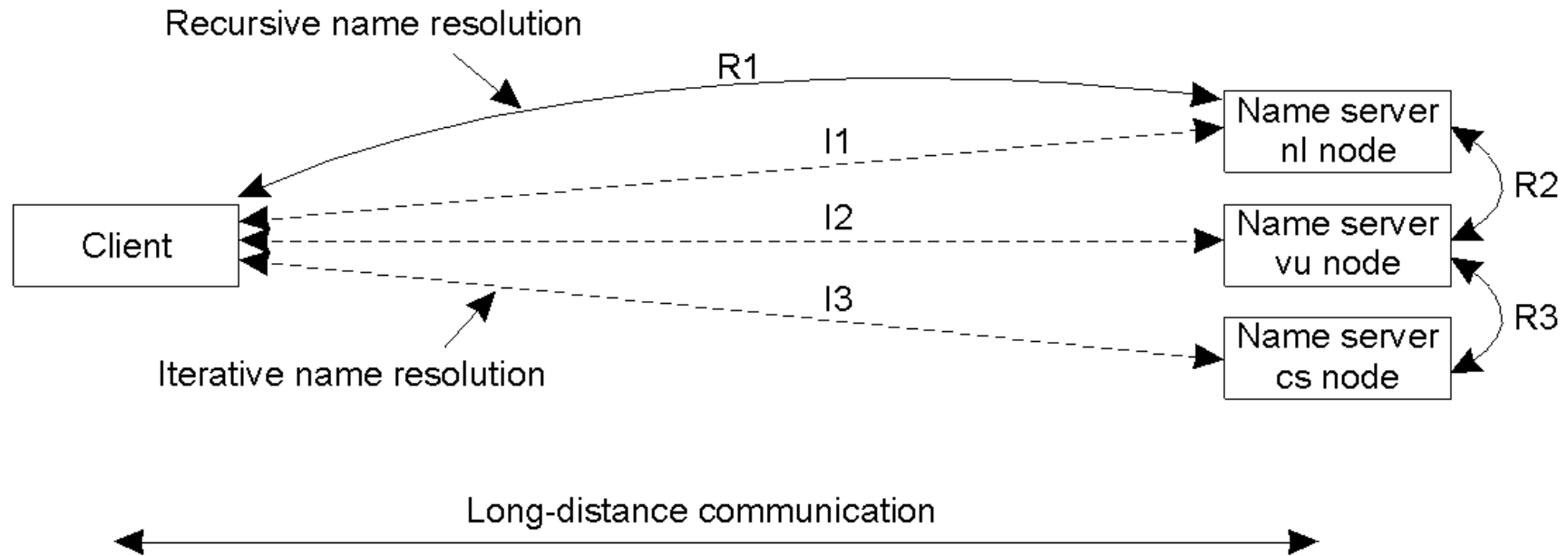
# Phân giải tên đệ qui





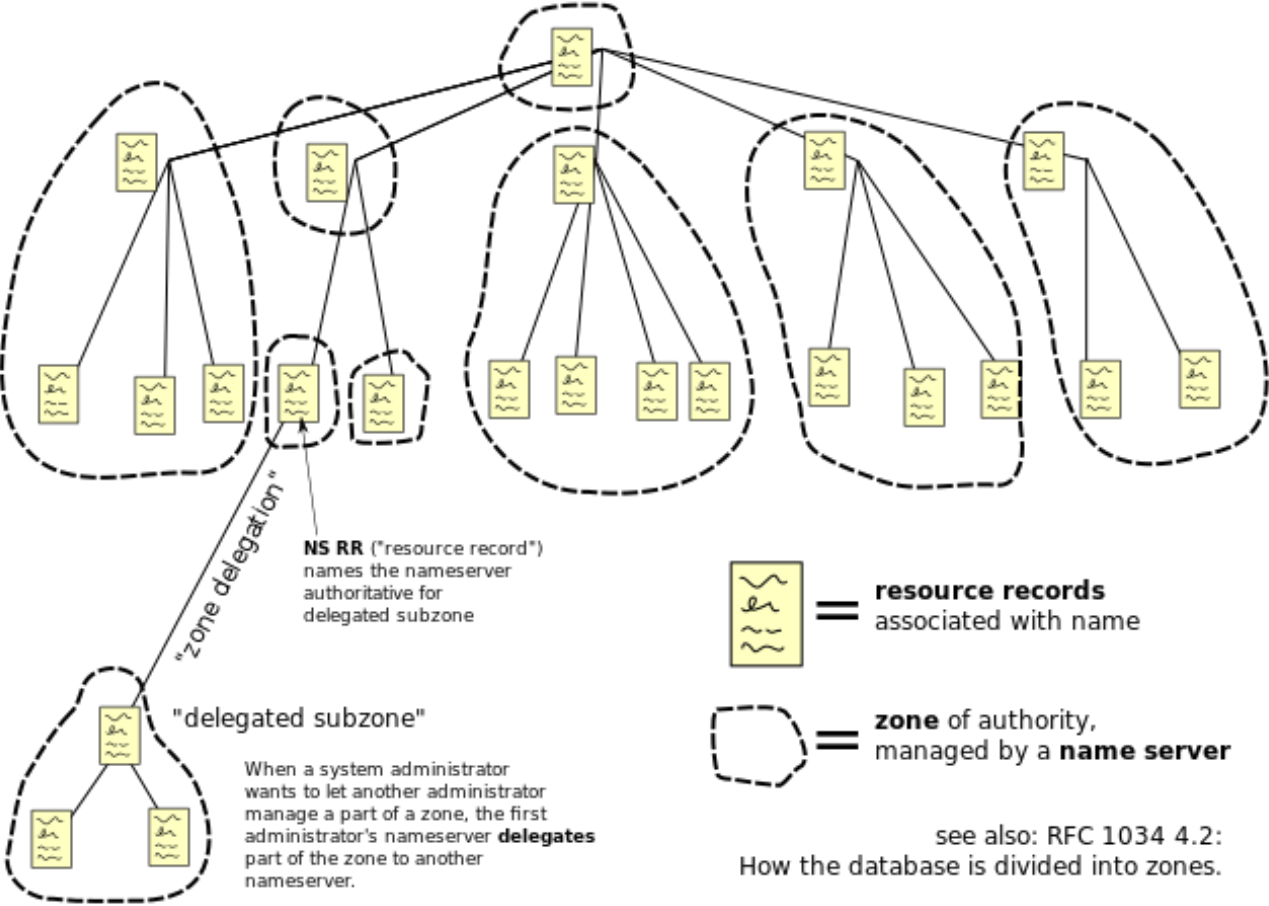
# So sánh phân giải đệ qui/không đệ qui

65



# Ví dụ: DNS

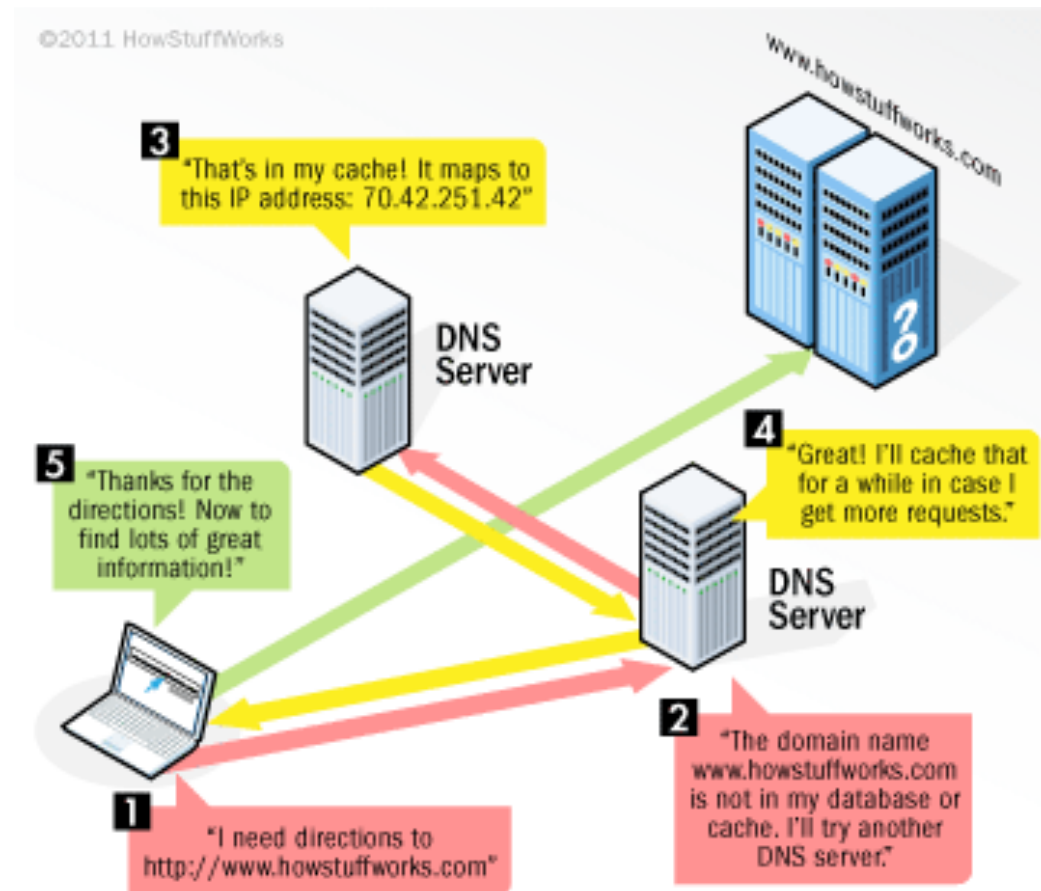
## Domain Name Space



[source: wikipedia.org]

# Cơ chế của DNS

67



# Một số bản ghi quan trọng

68

Type of record	Associated entity	Description
SOA	Zone	Holds information on the represented zone
A	Host	Contains an IP address of the host this node represents
MX	Domain	Refers to a mail server to handle mail addressed to this node
SRV	Domain	Refers to a server handling a specific service
NS	Zone	Refers to a name server that implements the represented zone
CNAME	Node	Symbolic link with the primary name of the represented node
PTR	Host	Contains the canonical name of a host
HINFO	Host	Holds information on the host this node represents
TXT	Any kind	Contains any entity-specific information considered useful

Name	Record type	Record value
cs.vu.nl.	SOA	star.cs.vu.nl. hostmaster.cs.vu.nl. 2005092900 7200 3600 2419200 3600
cs.vu.nl.	TXT	"Vrije Universiteit - Math. & Comp. Sc."
cs.vu.nl.	MX	1 mail.few.vu.nl.
cs.vu.nl.	NS	ns.vu.nl.
cs.vu.nl.	NS	top.cs.vu.nl.
cs.vu.nl.	NS	solo.cs.vu.nl.
cs.vu.nl.	NS	star.cs.vu.nl.
star.cs.vu.nl.	A	130.37.24.6
star.cs.vu.nl.	A	192.31.231.42
star.cs.vu.nl.	MX	1 star.cs.vu.nl.
star.cs.vu.nl.	MX	666 zephyr.cs.vu.nl.
star.cs.vu.nl.	HINFO	"Sun" "Unix"
zephyr.cs.vu.nl.	A	130.37.20.10
zephyr.cs.vu.nl.	MX	1 zephyr.cs.vu.nl.
zephyr.cs.vu.nl.	MX	2 tornado.cs.vu.nl.
zephyr.cs.vu.nl.	HINFO	"Sun" "Unix"
ftp.cs.vu.nl.	CNAME	soling.cs.vu.nl.
www.cs.vu.nl.	CNAME	soling.cs.vu.nl.
soling.cs.vu.nl.	A	130.37.20.20
soling.cs.vu.nl.	MX	1 soling.cs.vu.nl.
soling.cs.vu.nl.	MX	666 zephyr.cs.vu.nl.
soling.cs.vu.nl.	HINFO	"Sun" "Unix"
vucs-das1.cs.vu.nl.	PTR	0.198.37.130.in-addr.arpa.
vucs-das1.cs.vu.nl.	A	130.37.198.0
inkt.cs.vu.nl.	HINFO	"OCE" "Proprietary"
inkt.cs.vu.nl.	A	192.168.4.3
pen.cs.vu.nl.	HINFO	"OCE" "Proprietary"
pen.cs.vu.nl.	A	192.168.4.2
localhost.cs.vu.nl.	A	127.0.0.1

# Nội dung

70

1. Tên, định danh, và địa chỉ
2. Không gian tên phẳng
3. Không gian tên có cấu trúc
4. Định danh dựa trên thuộc tính

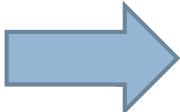
## 4. Định danh dựa trên thuộc tính

Dịch vụ thư mục

Dịch vụ thư mục phân tán

# 4.1. Dịch vụ thư mục

- *In software engineering, a directory is a map between **names** and **values**.*
- Lưu trữ thông tin của các thực thể theo thuộc tính
- Có thể sử dụng tập thuộc tính cứng/động
  - ▣ Thuộc tính cứng: tập thuộc tính tối ưu
  - ▣ Thuộc tính động: khung mô tả tập thuộc tính (Resource Description Framework)
    - cặp bộ 3 (subject, predicate, object)
    - Vd: (Person, name, Alice)

 cần các kỹ thuật để áp dụng vào HT mà dữ liệu được phân tán nhiều máy.



# Ví dụ RDF

73

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:contact=http://www.w3.org/2000/10/swap/pim/contact#>

  <contact:Person rdf:about="http://www.w3.org/People/EM/contact#me">
    <contact:fullName>Eric Miller</contact:fullName>
    <contact:mailbox rdf:resource="mailto:em@w3.org"/>
    <contact:personalTitle>Dr.</contact:personalTitle>
  </contact:Person>

</rdf:RDF>
```

# LDAP (Lightweight directory access protocol)

74

<b>Attribute</b>	<b>Abbr.</b>	<b>Value</b>
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
FTP_Server	--	130.37.21.11
WWW_Server	--	130.37.21.11

# LDAP (cont.)

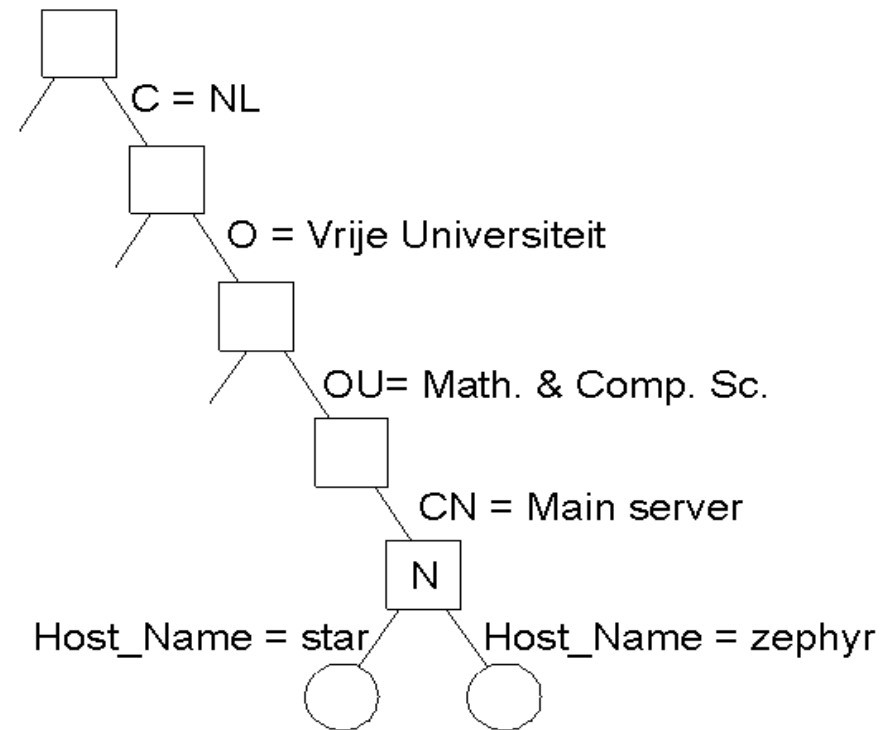
75

- DIB (Directory Information Base)
- RDN (Relative Distinguished Name)
- /C=NL/O=Vrije Universiteit/OU=Comp. Sc.

# DIT (Directory Information Tree)

76

- Mỗi node biểu diễn 1 bản ghi
- Node N thành node cha (khi thêm RDN Host\_Name)
- Lệnh *read* và *list*
- DSA (Directory Service Agents)
- DUA (Directory User Agents)



# Lệnh *read* và *list*

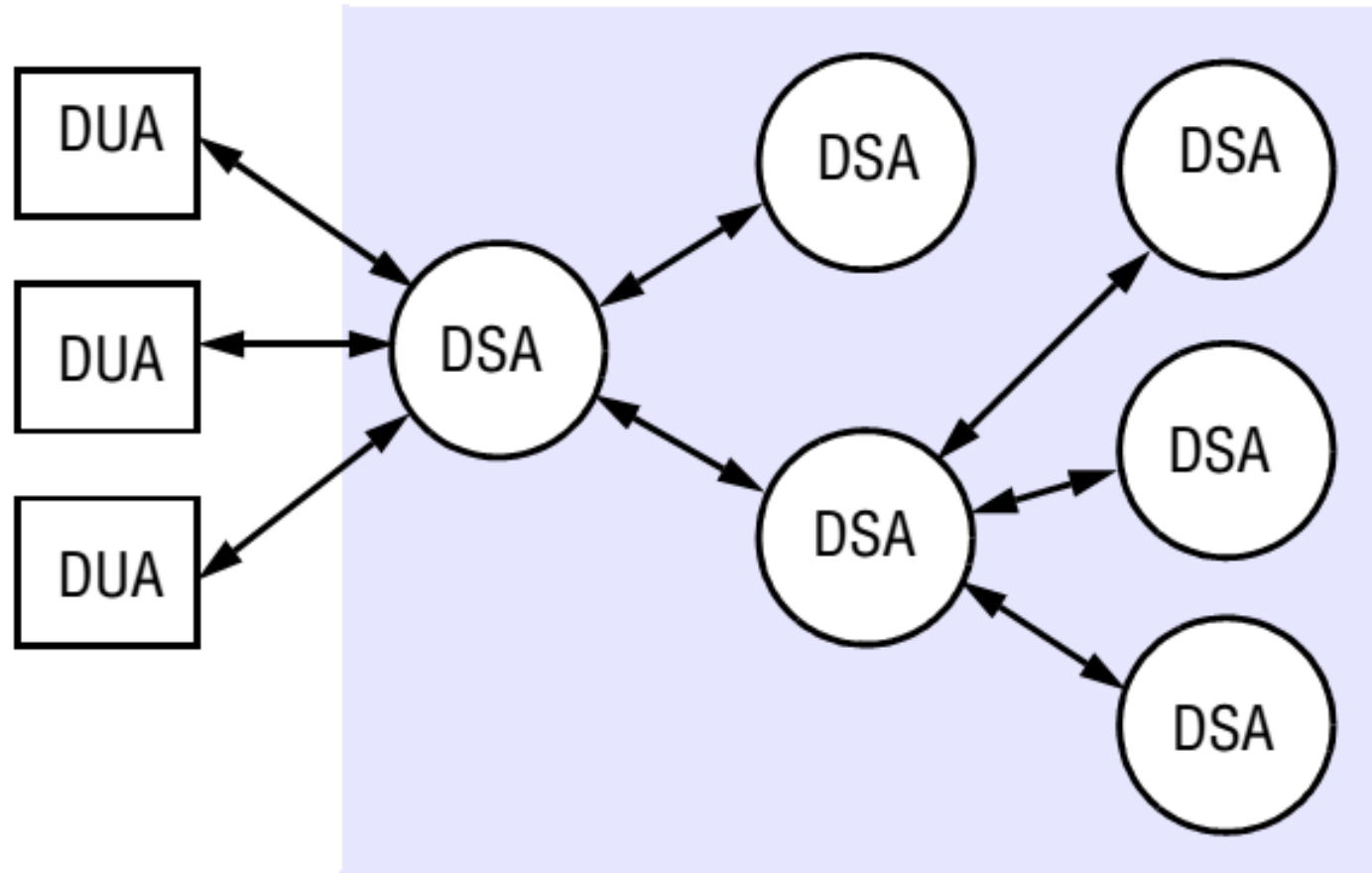
77

Attribute	Value
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Math. & Comp. Sc.
CommonName	Main server
Host_Name	star
Host_Address	192.31.231.42

Attribute	Value
Country	NL
Locality	Amsterdam
Organization	Vrije Universiteit
OrganizationalUnit	Math. & Comp. Sc.
CommonName	Main server
Host_Name	zephyr
Host_Address	192.31.231.66

# Kiến trúc dịch vụ X500

78



## 4.2. Dịch vụ thư mục phân tán

79

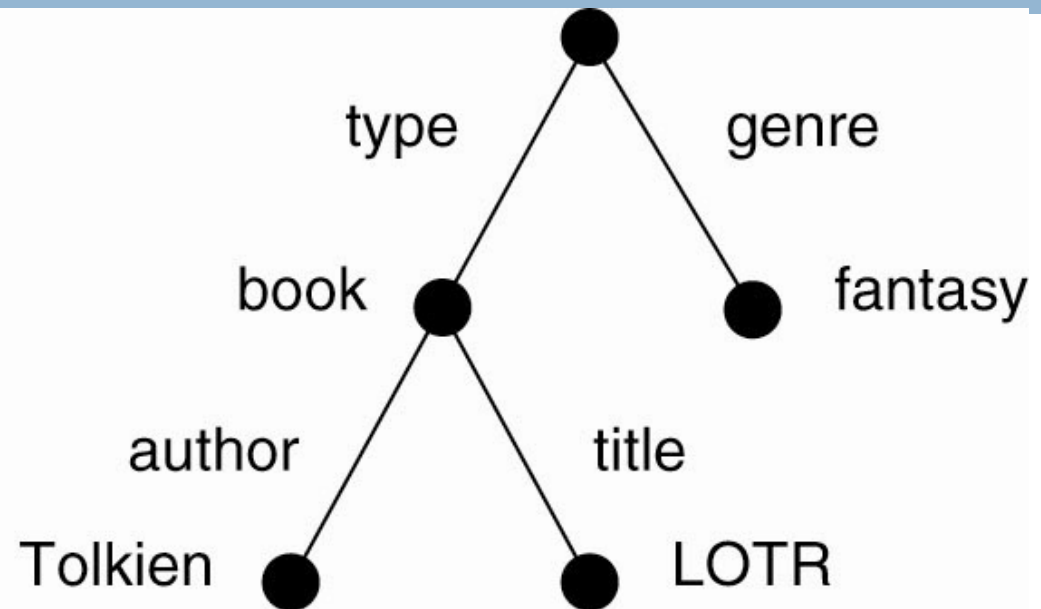
- Ánh xạ vào bảng băm phân tán
  - ▣ Tìm kiếm chính xác
  - ▣ Tìm kiếm theo khoảng
- Mạng overlay ngữ nghĩa

# Ảnh xạ vào bảng băm phân tán

80

```
description {  
  type = book  
  description {  
    author = Tolkien  
    title = LOTR  
  }  
  genre = fantasy  
}
```

(a)



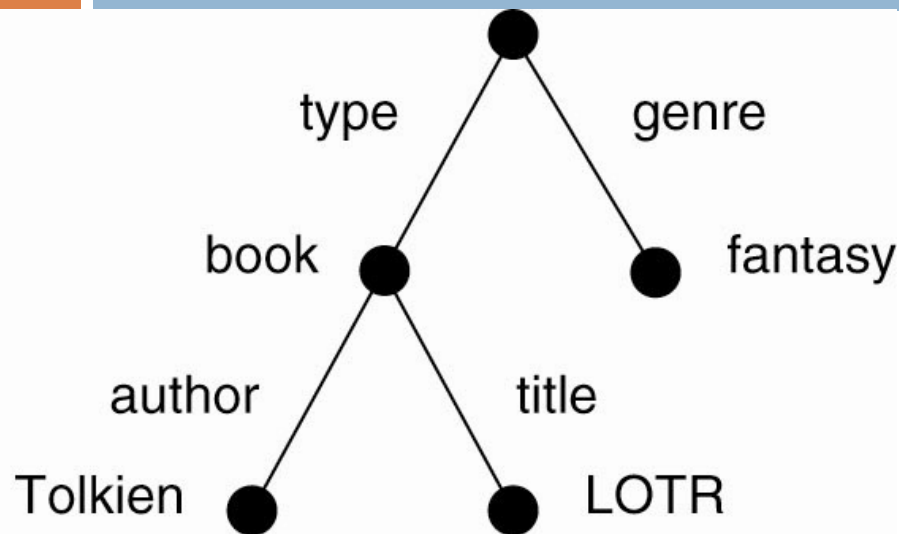
(b)

**AVTree (attribute-value tree)**

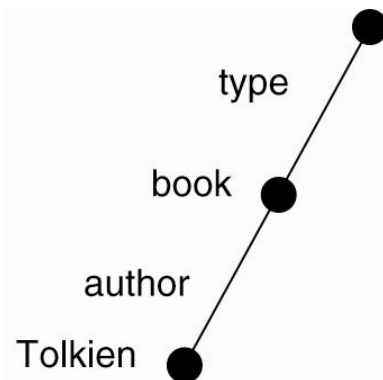


# Tìm kiếm chính xác: Các giá trị hàm hash

81



(b)



$h_1$ : hash(type-book)  
 $h_2$ : hash(type-book-author)  
 $h_3$ : hash(type-book-author-Tolkien)  
 $h_4$ : hash(type-book-title)  
 $h_5$ : hash(type-book-title-LOTR)  
 $h_6$ : hash(genre-fantasy)

$h_1$ : hash(type-book)  
 $h_2$ : hash(type-book-author)  
 $h_3$ : hash(type-book-author-Tolkien)

# Mạng overlay ngữ nghĩa

83

- Phân tán hóa → tăng tính tự trị
- → Xây dựng một hệ thống mà các nodes liên kết với nhau dựa trên các tài nguyên tương tự nhau

## □ **Semantic Overlay Network**

### □ Các hướng tiếp cận:

- ▣ Coi có sự gần giống nhau của các thông tin mà các node lưu trữ → SAI
- ▣ Dựa trên đặc tả của tên file
- ▣ Sử dụng *hàm khoảng cách ngữ nghĩa (semantic proximity function)* được xây dựng dựa trên sự giống nhau giữa các danh sách file

# Mô hình gossiping 2 tầng

84

