



ĐẶT TÊN TRONG HỆ THỐNG PHÂN TÁN

Trình bày: Phạm Minh Dương – B24CHHT064

Ma Công Thành – NCS2024.20

Vũ Quỳnh Anh – B24CHHT056

Nội dung - Đặt tên có cấu trúc

Không gian tên (Name spaces)

Phân giải tên (Name resolution)

Gắn kết không gian tên

Phân phối không gian tên (Name space distribution)

Triển khai phân giải tên (Implementation of name resolution)

Nội dung - Đặt tên theo thuộc tính

Dịch vụ thư mục (directory services)

Cài đặt theo kiến trúc phân cấp
(Hierarchical implementations: LDAP)

Cài đặt theo kiến trúc ngang hàng



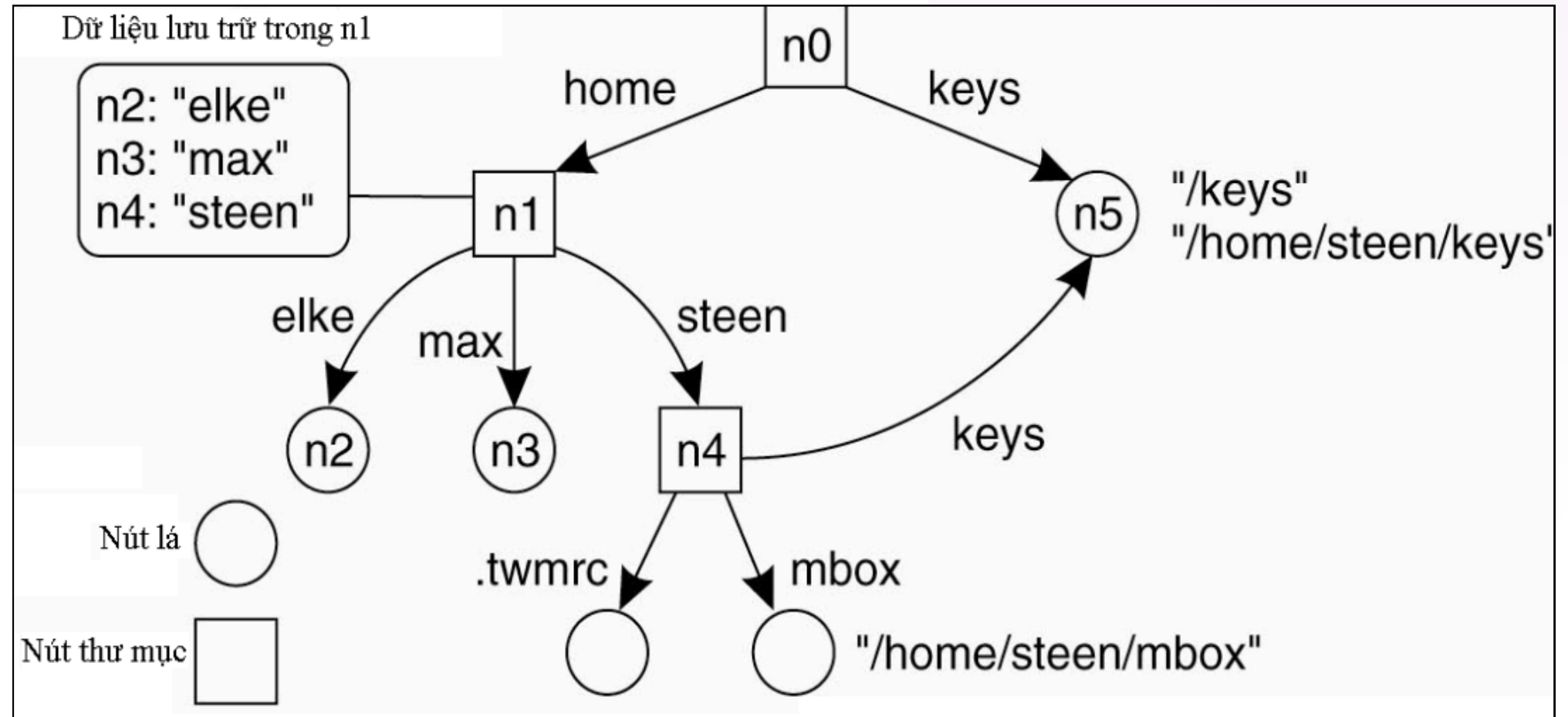
Đặt tên có cấu trúc (Structured Naming)

Nội dung - Đặt tên có cấu trúc

- Định danh phi cấu trúc thích hợp cho máy, nhưng không thuận tiện cho người sử dụng.
→ Ra đời hệ thống định danh hỗ trợ tên có cấu trúc được cấu tạo từ sự đơn giản, phù hợp với con người.
- Hệ thống này không chỉ hỗ trợ định danh tên file mà cả hệ thống tên host trên Internet.

Không gian tên

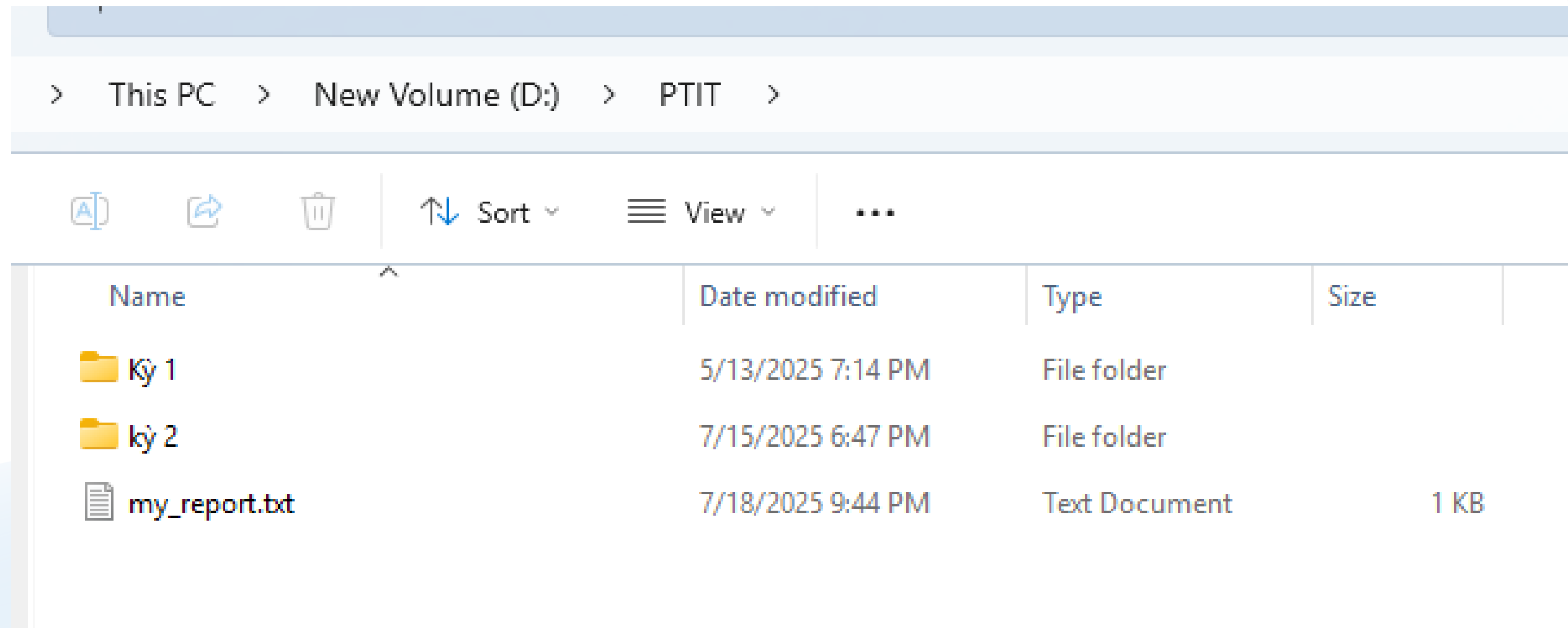
- Các tên thường được tổ chức thành không gian tên, có thể biểu diễn bằng đồ thị có hướng gắn nhãn.
 - Đồ thị này bao gồm các nút lá và nút thư mục.
 - Nút gốc (Root nodes): là điểm khởi đầu, không có đường đi vào
 - Nút lá (Leaf nodes):
 - Không có cạnh đi ra.
 - Đại diện cho thực thể được đặt tên.
 - Chứa thông tin liên quan đến thực thể.
 - Nút thư mục (Directory nodes):
 - Có cạnh đi ra.
 - Lưu trữ bảng thư mục chứa các cặp nhãn.
- The diagram illustrates a hierarchical namespace graph. It features several nodes: n0 (a square directory node at the top), n1 (a square directory node), n2, n3, and n4 (all square directory nodes), and n5 (a circular leaf node). Edges are labeled with names: 'home' from n0 to n1, 'keys' from n0 to n5, 'elke' from n1 to n2, 'max' from n1 to n3, 'steen' from n1 to n4, and 'keys' from n4 to n5. Node n4 has two outgoing edges labeled '.twmrc' and 'mbox' leading to two unlabeled circular leaf nodes. A box labeled 'Dữ liệu lưu trữ trong n1' (Data stored in n1) contains a list: n2: "elke", n3: "max", n4: "steen". A legend at the bottom left shows a circle for 'Nút lá' (Leaf node) and a square for 'Nút thư mục' (Directory node). Path labels on the right include '/keys' and '/home/' for n5, and '/home/steen/mbox' for the leaf node reached via n4 and 'mbox'.
- Hình 1: Đồ thị định danh tổng thể với duy nhất một (Giáo trình)



Hình 1: Đồ thị định danh tổng thể với duy nhất một nút gốc
(Giáo trình)

Không gian tên

- D:\ là một nút gốc
- Kỳ 1, Kỳ 2 là các nút thư mục nằm dưới nút D:\
- my_report.txt là một nút lá



Hình 2: Ví dụ về đặt tên trên hệ thống máy tính

Không gian tên

Đường dẫn (path):

- Là chuỗi các nhãn liên tiếp mà bạn đi qua để đến được một nút từ một nút khác.
- **Đường dẫn tuyệt đối (Absolute Path):** Là đường dẫn bắt đầu từ nút gốc. Ví dụ: D:\PTIT\Documents\my_report.docx.
- **Đường dẫn tương đối (Relative Path):** Là đường dẫn bắt đầu từ một nút bất kỳ, không phải nút gốc. Ý nghĩa của nó phụ thuộc vào vị trí hiện tại của bạn. Ví dụ, nếu bạn đang ở thư mục Kỳ 1, thì Kỳ 2\my_report.docx là đường dẫn tương đối.

Không gian tên

Tên toàn cục và Tên cục bộ:

- Tên toàn cục (global name): Là tên biểu thị cùng một thực thể và có ý nghĩa giống nhau trên toàn bộ hệ thống. Đường dẫn tuyệt đối chính là một dạng của tên toàn cục. Ví dụ: `www.google.com` là một tên toàn cục trên Internet.
- Tên cục bộ (Local Name): Là tên mà ý nghĩa của nó chỉ có giá trị trong một phạm vi hoặc ngữ cảnh nhất định. Chỉ thư mục chứa nó mới biết ý nghĩa của nó.

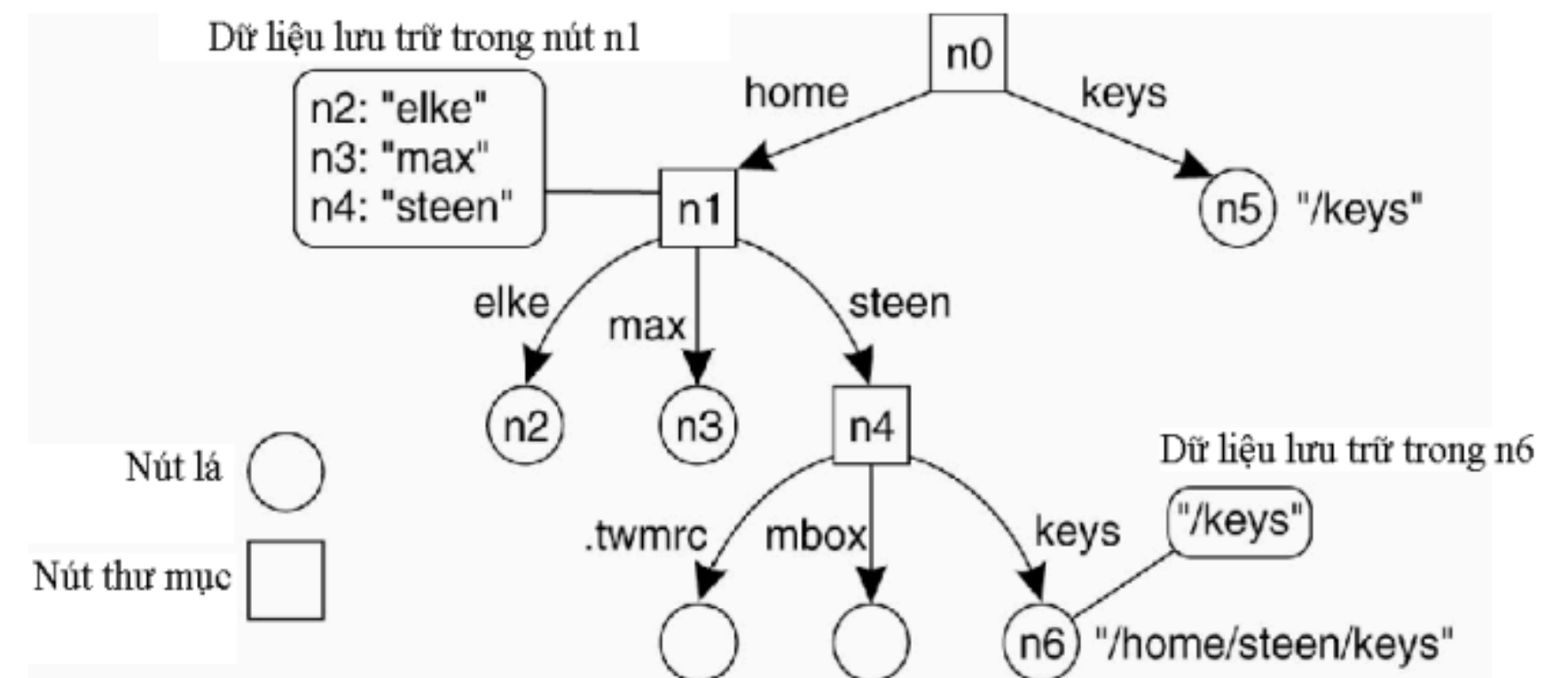
Phân giải tên

Là quá trình không gian tên cung cấp một quá trình thuận tiện cho việc lưu trữ và trích xuất thông tin về các đối tượng bằng tên.

Để phân giải cần bắt đầu từ một điểm đã biết (**Cơ chế đóng**). *Ví dụ:* khi nhìn vào dãy số 84986677028 ta sẽ không biết nó là cái gì trừ khi có thông tin nó là số điện thoại.

Bí danh là một tên khác của cùng một thực thể, có hai cách triển khai trong đồ thị tên:

- Cho phép nhiều đường dẫn tuyệt đối tham chiếu đến cùng một nút.
- Liên kết tên.

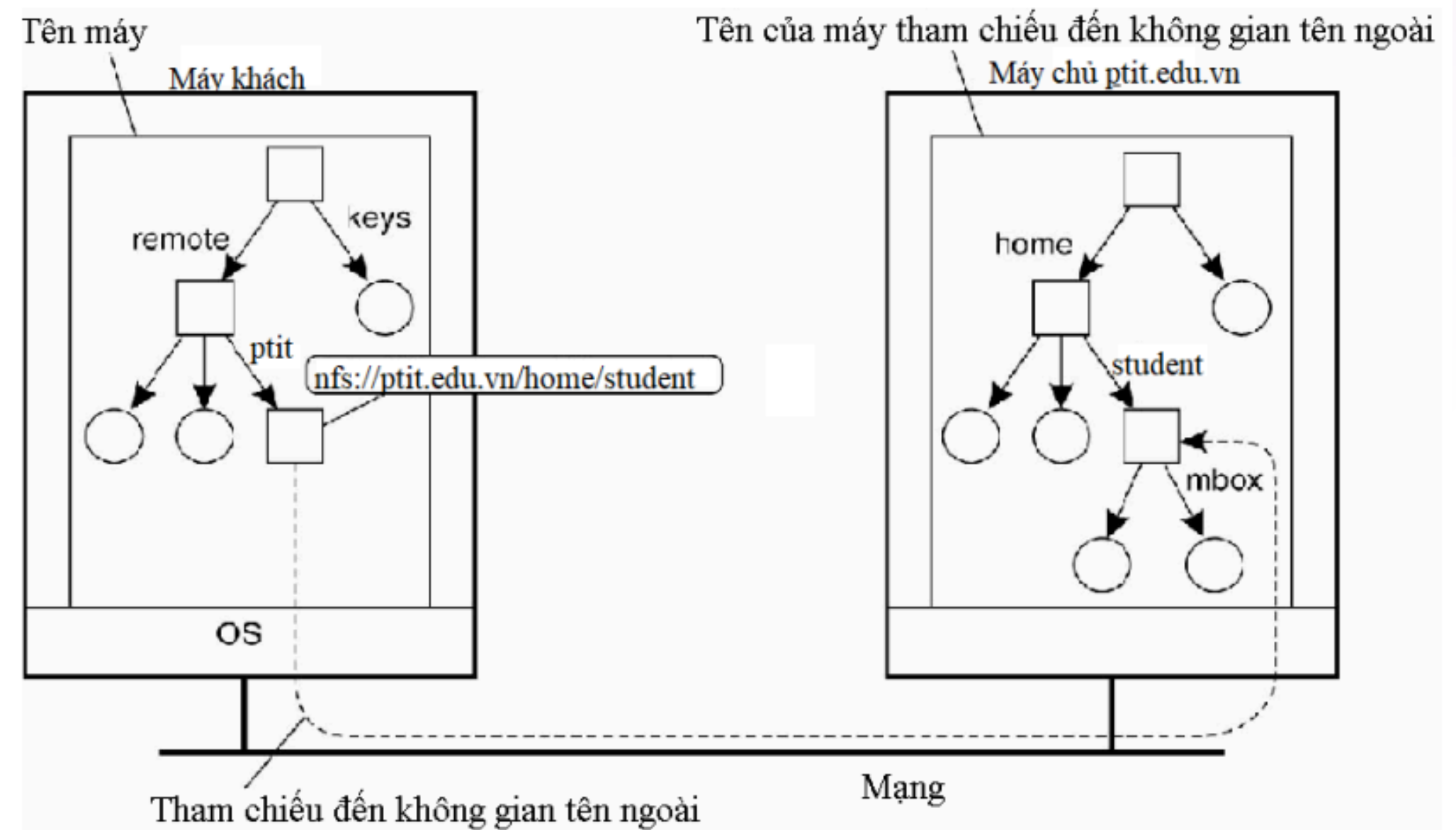


Hình 3: Liên kết trong đồ thị tên
(Giáo trình)

Gắn kết không gian tên

Trong một hệ thống phân tán, các máy tính hoặc dịch vụ khác nhau có thể có không gian tên riêng của chúng. Để làm việc với tài nguyên từ các không gian tên khác, chúng ta cần gắn kết chúng lại với nhau:

- Tên giao thức sử dụng
- Tên của server
- Tên của điểm truy nhập



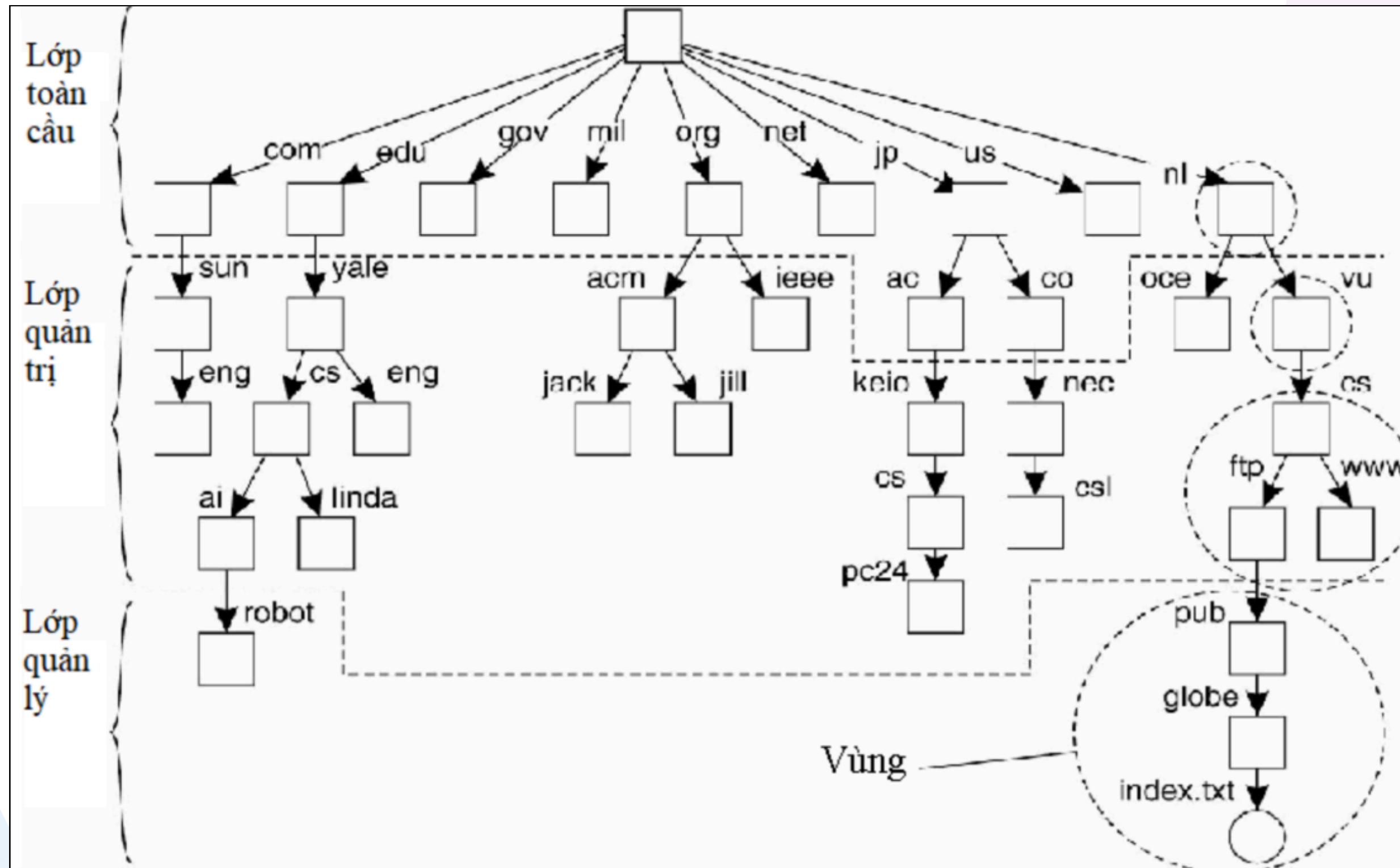
Hình 4: Gắn kết không gian tên (Giáo trình)

Phân phối không gian tên

Trong các hệ thống phân tán quy mô lớn, không gian tên (name space) thường được tổ chức theo cấu trúc phân cấp:

- Global (toàn cầu): gồm nút gốc và các nút thư mục cấp cao nhất. (ví dụ: .com, .vn, .edu).
- Administrative (quản trị): gồm các nút thư mục được quản lý bởi một tổ chức hoặc đơn vị quản trị duy nhất. (ví dụ: ptit.edu.vn, hcm.edu.vn)
- Managerial (quản lý): chứa những nút thư mục ở mức thấp, thay đổi thường xuyên. (ví dụ: tên máy chủ cục bộ, thư mục và tệp do người dùng định nghĩa).

Phân phối không gian tên



Hình 5: Tổ chức tên miền Internet (Giáo trình)

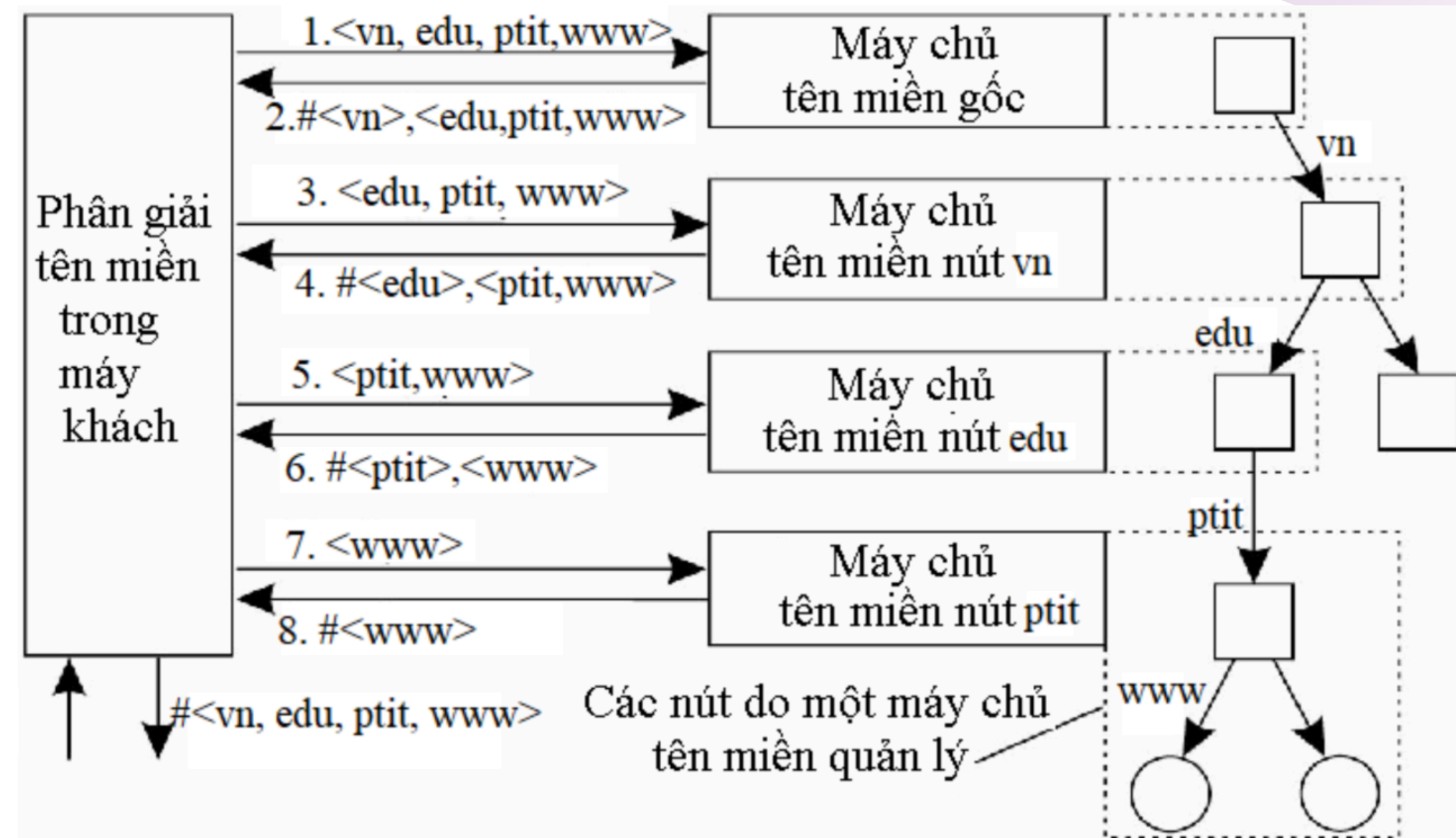
Phân phối không gian tên

Mục	Toàn cầu	Quản trị	Quản lý
Phạm vi địa lý	Toàn thế giới	Tổ chức	Đơn vị
Số lượng nút	Ít	Nhiều	Mili giây
Thời gian đáp ứng	Giây	Mili giây	Ngay lập tức
Cập nhật lan truyền	Ít	Ngay lập tức	Ngay lập tức
Số lượng bản sao	Nhiều	Ít	Không có
Lưu yêu cầu máy khách	Có	Có	Không cần

Bảng: So sánh đặc điểm máy chủ tên miền theo các mức (Giáo trình)

Triển khai phân giải tên

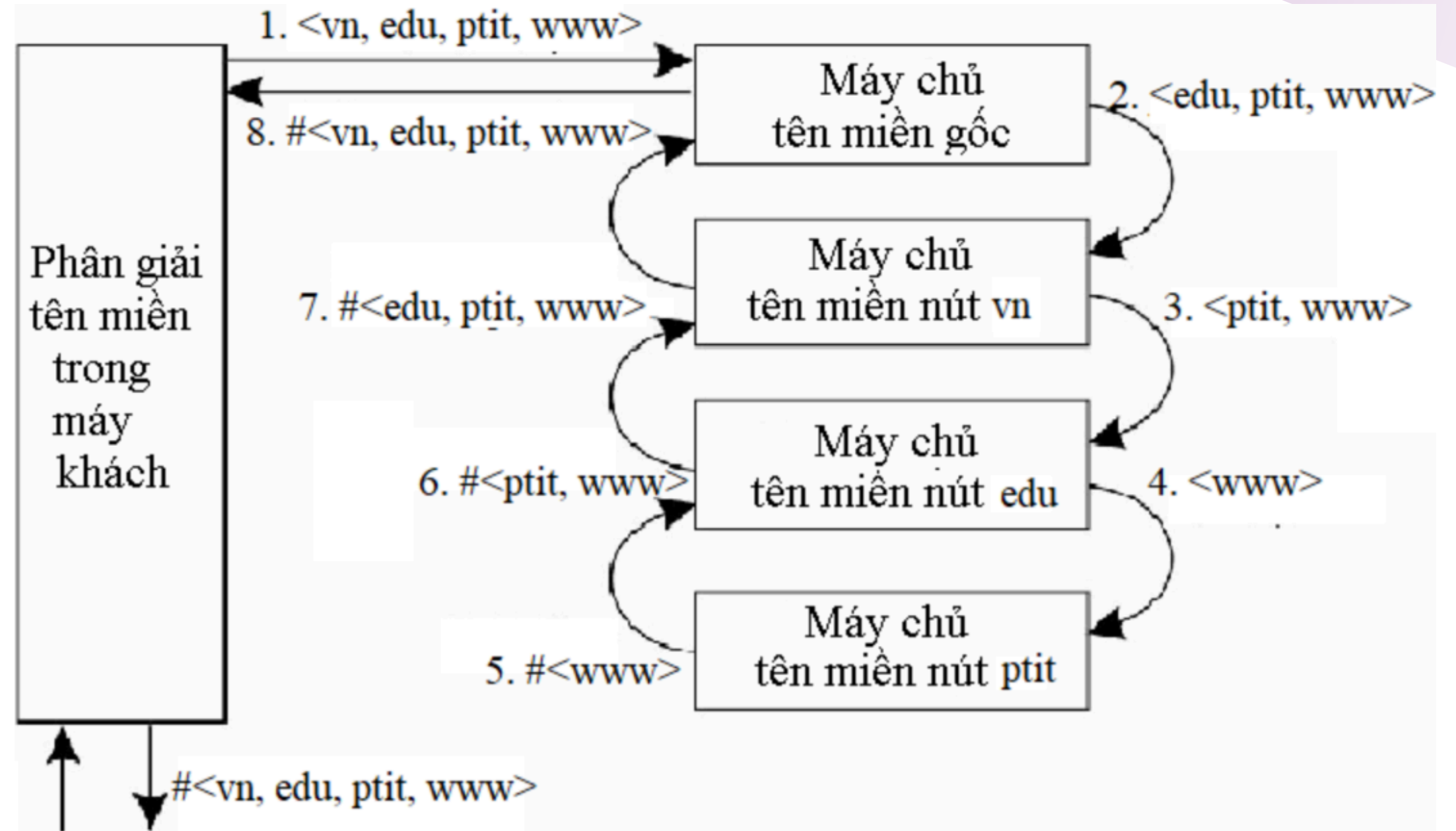
Phân giải tương tác: Máy khách tự mình liên hệ tuần tự với các máy chủ tên miền khác nhau theo chỉ dẫn để có được địa chỉ IP cuối cùng.



Hình 6: Phân giải tên miền tương tác
(Giáo trình)

Triển khai phân giải tên

Phân giải đệ quy: Máy khách chỉ gửi một yêu cầu duy nhất tới máy chủ tên miền của nó, và máy chủ này sẽ tự liên hệ các máy chủ cấp thấp hơn và trả về kết quả cuối cùng cho máy khách.



Hình 7: Phân giải tên miền đệ quy
(Giáo trình)

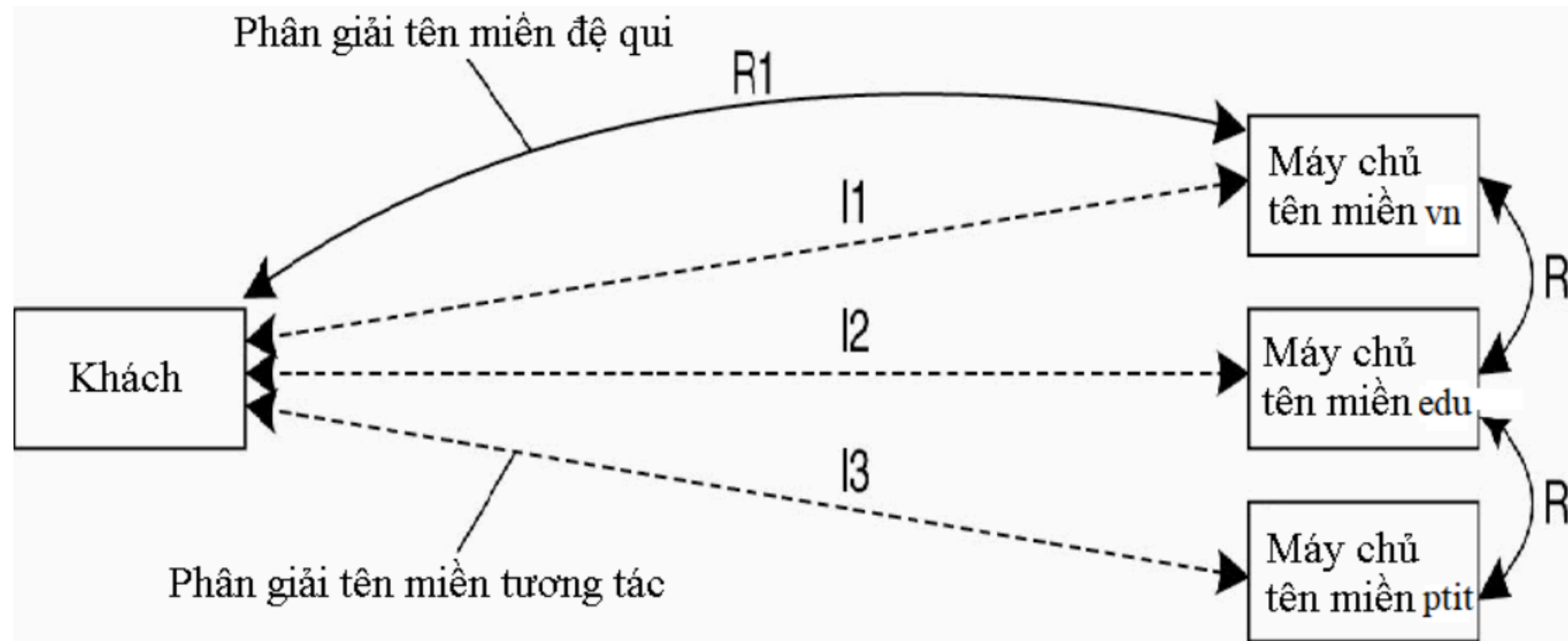
Triển khai phân giải tên

- **Phân giải tương tác:**
 - **Ưu điểm:** Phân tán tải tốt cho các máy chủ, mỗi máy chỉ "chỉ dẫn" bước tiếp theo.
 - **Nhược điểm:** Hiệu quả bộ đệm thấp, máy khách phải tự liên hệ nhiều lần.
- **Phân giải đệ quy:**
 - **Ưu điểm:** Hiệu quả bộ đệm cao (máy chủ lưu trữ kết quả đầy đủ), giảm lưu lượng mạng cho máy khách.
 - **Nhược điểm:** Tải cao hơn cho máy chủ thực hiện đệ quy.

Triển khai phân giải tên

Máy chủ	Yêu cầu phân giải	Tìm kiếm	Gửi cấp thấp hơn	Nhận và lưu bộ nhớ đệm	Trả kết quả cho máy khách
ptit	<www>	#<www>	-	-	#<www>
edu	<ptit,www>	#<ptit>	<www>	#<www>	#<ptit> #<ptit, www>
vn	<edu,ptit,www>	#<edu>	<ptit,www>	#<www> #<ptit, www>	#<edu> #< edu,ptit> #<edu,ptit,www>
Gốc	<vn,edu,ptit,www>	#<vn >	<edu,ptit, www>	#<vn> #<vn,edu > #<vn,edu,ptit>	#<vn> #<vn,edu> #<vn,edu,ptit> #<vn,edu,ptit,www>

Triển khai phân giải tên



Hình 8: So sánh hai giải pháp phân giải tên miền (Giáo trình)

Ví dụ: Hệ thống Tên miền DNS

- DNS là hệ thống đặt tên phân tán lớn nhất hiện nay.
- Giúp ánh xạ các tên miền thân thiện với con người sang địa chỉ IP mà máy tính có thể hiểu được để định tuyến thông tin.
- **Cấu trúc:** Tổ chức theo cây phân cấp với Tên miền gốc (.), các TLDs (Top-Level Domains) và các tên miền cấp dưới (Subdomains).
- **Máy chủ tên miền:** Cung cấp dịch vụ phân giải. Có 13 hệ thống máy chủ tên miền gốc phân tán toàn cầu để đảm bảo tính sẵn sàng.
- **Quá trình phân giải:** Máy khách gửi yêu cầu đến máy chủ DNS cục bộ, máy chủ này thực hiện phân giải đệ quy hoặc tương tác với các máy chủ DNS khác để tìm IP, sau đó trả về và lưu vào cache.



Đặt tên dựa trên thuộc tính (Attribute-based Naming)

Dịch vụ thư mục (Directory services) (1)

- Hay còn được gọi là hệ thống **Đặt tên dựa trên thuộc tính**
- Là phương pháp tìm kiếm thực thể bằng cách mô tả những gì bạn đang tìm kiếm.
- Sử dụng các cặp (thuộc tính, giá trị) để mô tả tài nguyên.
- Ví dụ:

Đặt tên theo cấu trúc	Đặt tên theo thuộc tính
<i>/home/alice/photo.jpg</i>	Tìm ảnh có (chủ đề, "bãi biển") và (năm, "2024")
<i>nl.vu.cs</i>	Tìm máy chủ có (Tổ chức, "VU University")

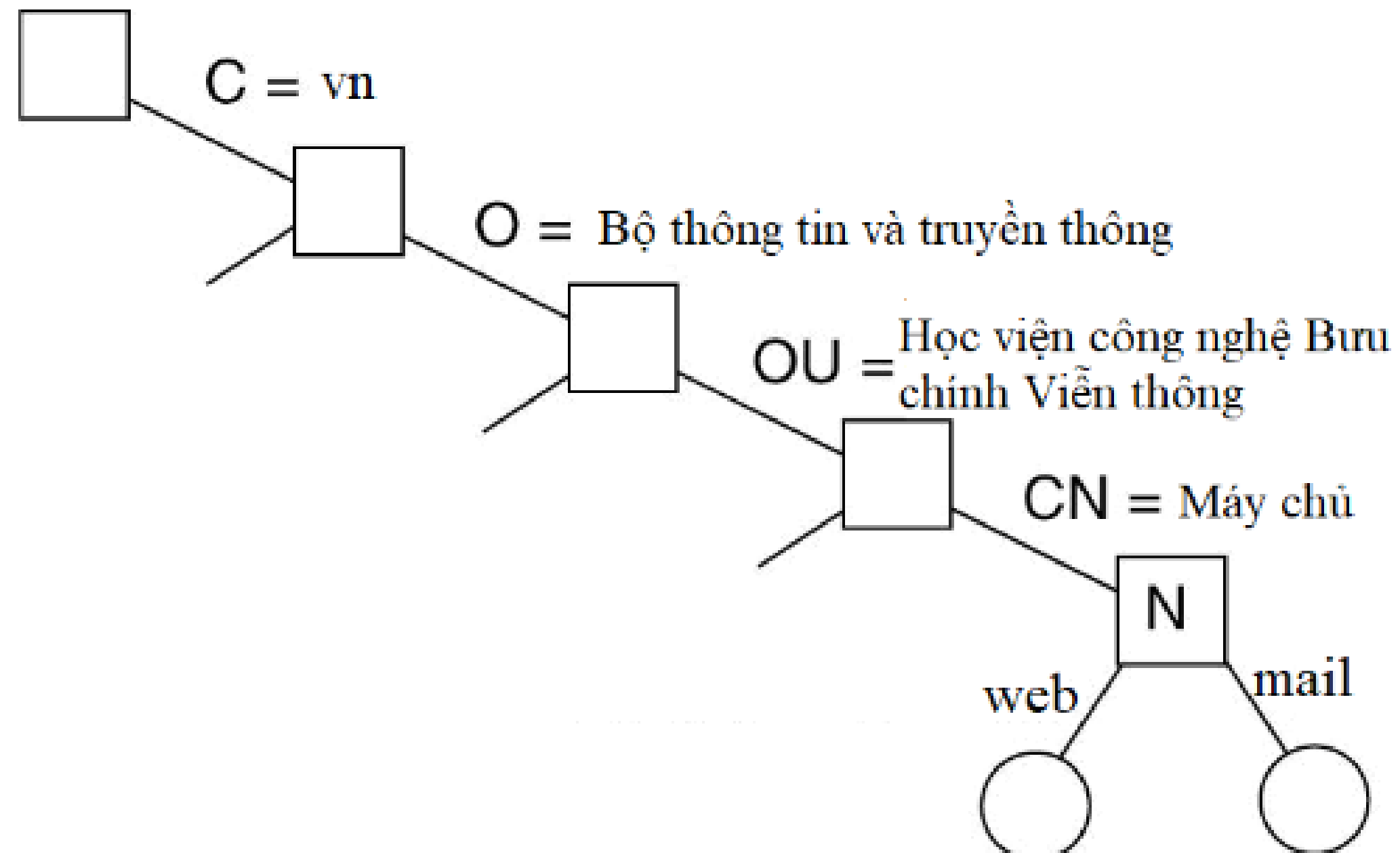
Dịch vụ thư mục (Directory services) (2)

- **Thách thức:** Khó khăn trong việc thiết kế thuộc tính và đảm bảo tính nhất quán khi nhập dữ liệu.
- **Giải pháp:** Áp dụng các chuẩn như **RDF (Resource Description Framework)** để mô tả dữ liệu bằng cấu trúc bộ ba (*chủ thể, vị ngữ, đối tượng*).
- **Tìm kiếm:**
 - Việc tìm kiếm trong hệ thống đặt tên dựa trên thuộc tính thường tốn thời gian => Có thể lập chỉ mục (index) cho dữ liệu
 - Tuy nhiên, khi gửi lượng lớn yêu cầu tìm kiếm đến hàng trăm máy chủ => **Cần có giải pháp tối ưu và hiệu quả hơn!**

Triển khai phân cấp - Giao thức LDAP (1)

- Cách tiếp cận phổ biến để xây dựng dịch vụ thư mục phân tán là kết hợp việc đặt tên theo cấu trúc và đặt tên theo thuộc tính.
- **LDAP (Lightweight Directory Access Protocol)** là một ví dụ điển hình cho phương pháp này, được sử dụng rộng rãi trong các hệ thống như Active Directory của Microsoft.
- Các khái niệm cốt lõi của LDAP
 - Mục nhập thư mục (**Directory Entry**)
 - Cấu trúc phân cấp (**DIT - Directory Information Tree**)

Triển khai phân cấp - Ví dụ LDAP (2)



Hình 9: Tổ chức bản ghi dịch vụ thư mục (Giáo trình)

Triển khai phân cấp - Ưu/nhược điểm của LDAP (3)

	Khía cạnh	Mô tả
Ưu điểm	Tìm kiếm linh hoạt	Hỗ trợ truy vấn theo nhiều thuộc tính trong Cơ sở thông tin thư mục (DIB), vượt trội hơn DNS.
	Truy vấn phức tạp	Cho phép lọc mục nhập theo nhiều tiêu chí như quốc gia, tổ chức, đơn vị, loại thực thể, v.v.
	Ví dụ thực tế	<code>search("(C=VN)(O=PTIT)(OU=*)(CN=Mail Server)")</code> → Truy vấn tất cả máy chủ thư điện tử thuộc tổ chức PTIT tại Việt Nam, không phân biệt đơn vị trực thuộc.
Nhược điểm	Chi phí truy vấn cao	Truy vấn có thể phải truy cập nhiều DSA để tổng hợp kết quả → tốn thời gian và tài nguyên.
	Không tối ưu như DNS	DNS chỉ cần truy cập một nút lá, còn LDAP có thể phải quét toàn bộ hoặc nhiều nhánh trong cây.

Triển khai phi tập trung - Giới thiệu (1)

- *Mục tiêu*
 - Loại bỏ sự phụ thuộc vào máy chủ trung tâm trong đặt tên dựa trên thuộc tính.
 - Hỗ trợ tìm kiếm tài nguyên hiệu quả trong **hệ thống ngang hàng (P2P)**.
- *Thách thức chính*: Làm sao để tìm kiếm thực thể theo (thuộc tính, giá trị) mà **không phải quét toàn bộ mạng**.
- *Các cách tiếp cận*:
 - Chỉ mục phân tán (Distributed Index)
 - Đường cong lấp đầy không gian (Space-Filling Curves)
 - Cây thuộc tính-giá trị (Attribute-Value Trees - AVTree)

Triển khai phi tập trung - LT Chỉ mục phân tán (2)

- *Ý tưởng:*
 - Mỗi **thuộc tính** (vd: tác giả, thể loại, năm) được gán cho **một máy chủ chỉ mục riêng**.
 - Máy chủ lưu danh sách các thực thể (files/tài nguyên) có giá trị tương ứng với thuộc tính đó
- *Quy trình tìm kiếm:*
 - Truy vấn dạng kết hợp nhiều thuộc tính (e.g., tác giả = A \wedge thể loại = B) \rightarrow gửi tới tất cả các máy chủ chỉ mục tương ứng.
 - **Client tự kết hợp kết quả** từ các máy chủ để tìm giao (intersection) các thực thể phù hợp
- *Nhược điểm chính:*

• Chi phí giao tiếp cao	Truy vấn với k thuộc tính \rightarrow cần liên hệ với k máy chủ khác nhau.
• Tải xử lý dồn về client	Client phải xử lý nhiều tập dữ liệu lớn, dễ quá tải (vd: lastName = Smith).
• Không hỗ trợ range query	Truy vấn dạng price $\in [1000-2500]$ rất khó thực hiện do không có tổ chức thứ tự.

Triển khai phi tập trung - VD Chỉ mục phân tán (3)

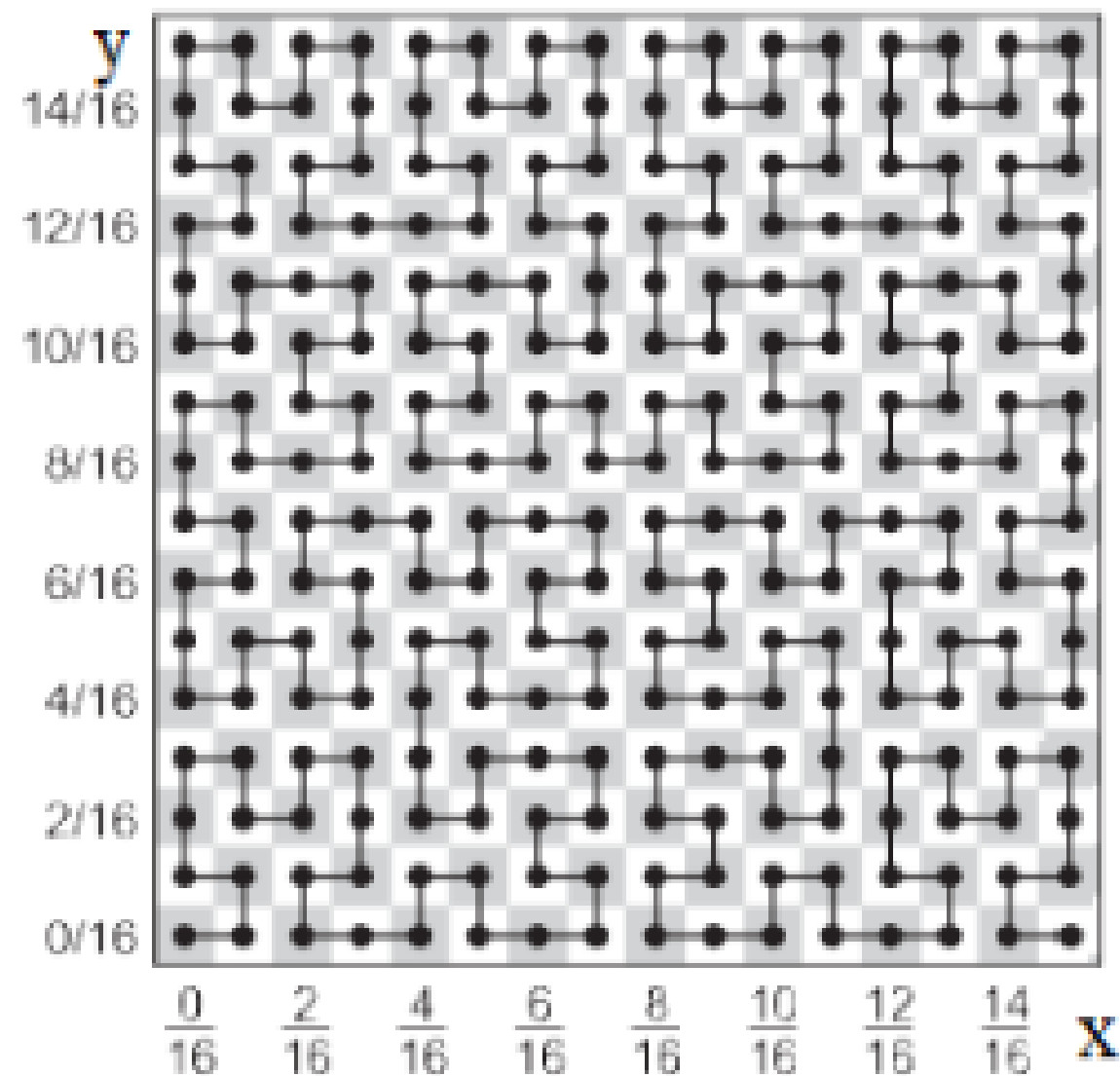
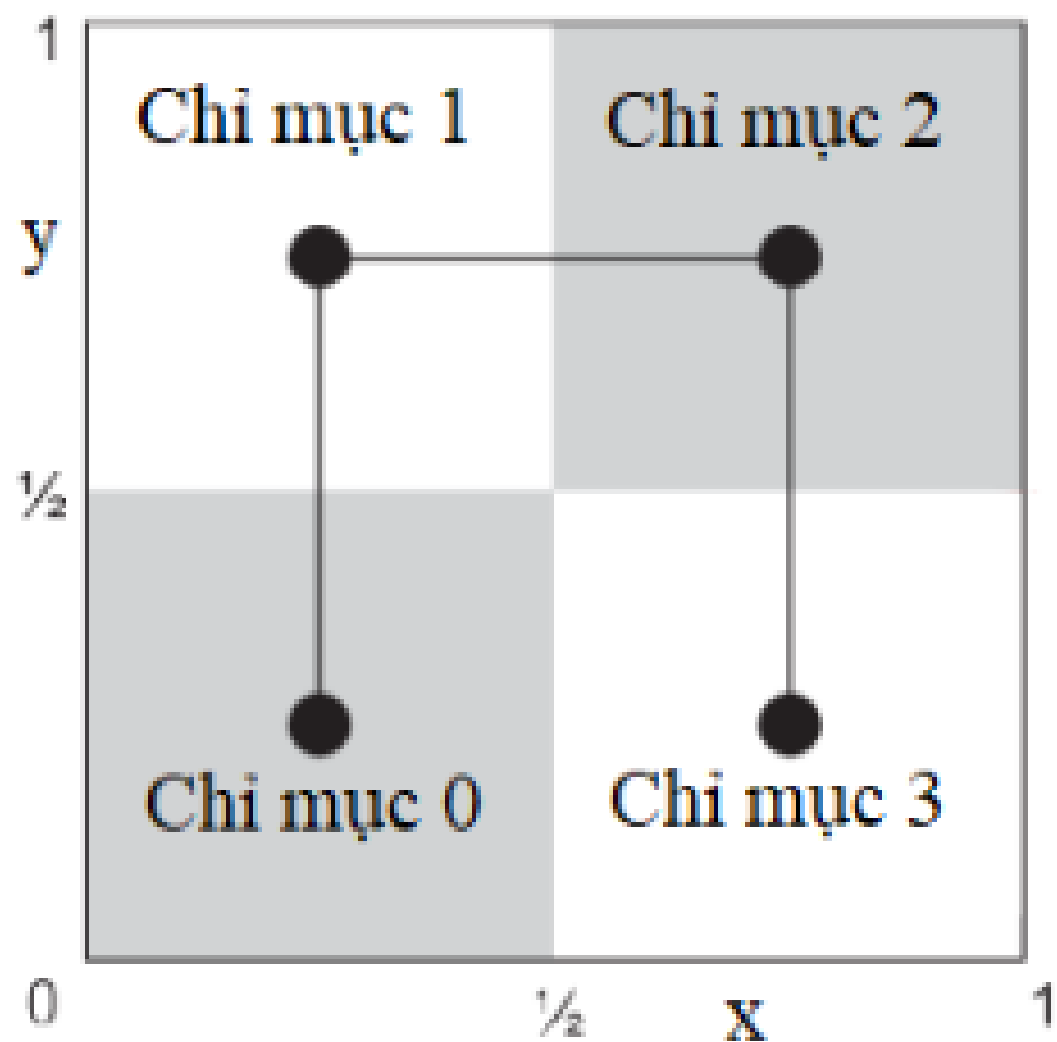
- Ví dụ: Trong một hệ thống phân tán lưu trữ sách điện tử, mỗi tài nguyên (cuốn sách) được mô tả bằng nhiều thuộc tính: Tác giả (author), Thể loại (genre), Năm xuất bản (year). Tìm các tài nguyên thỏa mãn:
 - Tác giả = Tolkien
 - Thể loại = Fantasy
- Cách hệ thống xử lý:

• Máy chủ chỉ mục 1	Phụ trách thuộc tính Tác giả, lưu: (Tolkien → [Book1, Book2])
• Máy chủ chỉ mục 2	Phụ trách thuộc tính Thể loại, lưu: (Fantasy → [Book1, Book3, Book4])
• Truy vấn từ client	Gửi song song đến cả hai máy chủ chỉ mục.
• Kết hợp kết quả tại client	Book1 xuất hiện trong cả hai danh sách → là kết quả hợp lệ.

Triển khai phi tập trung - LT Đường cong lấp đầy không gian (4)

- *Ý tưởng cốt lõi:*
 - **Ánh xạ không gian thuộc tính N chiều** (mỗi chiều là một thuộc tính: giá, vị trí, thời gian...) → thành **không gian 1 chiều** bằng một **đường cong liên tục**.
 - Ví dụ phổ biến: Đường cong Hilbert
- *Đặc tính quan trọng:*
 - **Bảo toàn tính cục bộ:** Hai điểm gần nhau trong không gian N chiều → ánh xạ thành hai chỉ số gần nhau.
 - **Hiệu quả tìm kiếm:** Hỗ trợ truy vấn theo vùng (range query) nhanh, hạn chế truy vấn toàn mạng.
 - **Tối ưu phân phối dữ liệu:** Dữ liệu được gán cho các chỉ số liên tiếp → dễ phân vùng và cân bằng tải.
- *Lợi ích trong hệ thống phân tán:* Giúp ánh xạ và phân phối dữ liệu hiệu quả, giảm số nút cần truy cập khi truy vấn và tăng hiệu suất so với các chỉ mục phân tán truyền thống

Triển khai phi tập trung - VD Đường cong lấp đầy không gian (5)



Hình 10: Cách đánh chỉ mục trong không gian Hilbert (Giáo trình)

Triển khai phi tập trung - LT Cây thuộc tính-giá trị (6)

- Ý tưởng cốt lõi:
 - Mô tả tài nguyên được tổ chức thành một cây phân cấp các thuộc tính → gọi là **Attribute-Value Tree (AVTree)**
 - Mỗi đường dẫn trong cây được mã hóa thành một khóa băm, lưu trữ trong mạng DHT.
- Lợi ích chính:
 - **Dư thừa có chủ đích:** Một tài nguyên được lưu dưới nhiều khóa → tăng khả năng truy vấn.
 - **Hỗ trợ truy vấn một phần:** Cho phép tìm kiếm linh hoạt khi chỉ biết một phần mô tả.

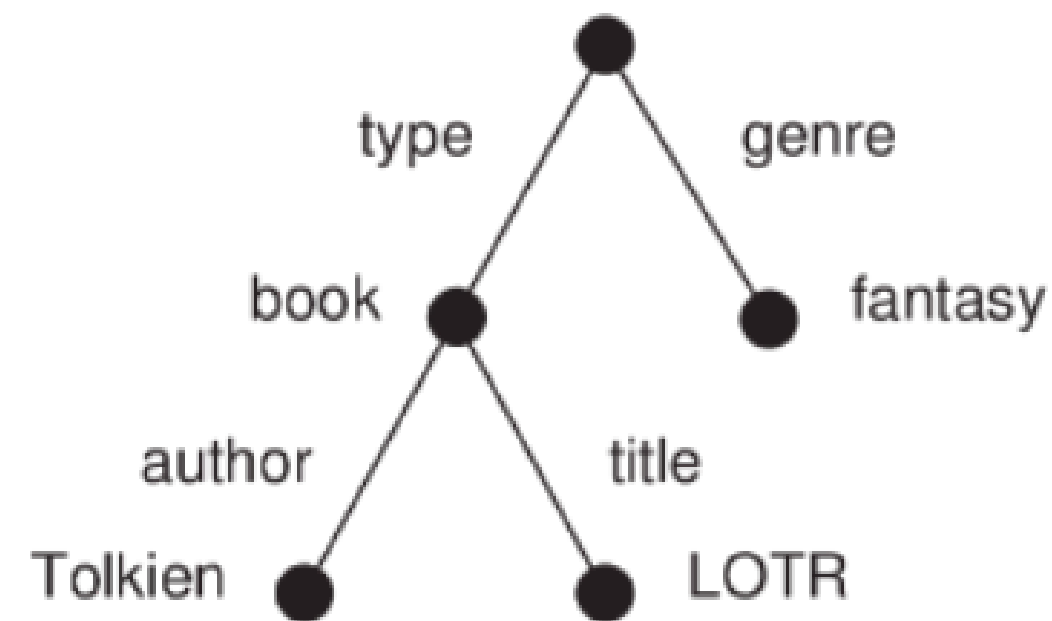
Triển khai phi tập trung - VD Cây thuộc tính-giá trị (7)

Ví dụ minh họa:

- Tài nguyên: Sách "Lord of the Rings" được mô tả: (type: book), (author: Tolkien), (title: LOTR)
 - Sinh các khóa:
 - hash(type-book)
 - hash(type-book-author)
 - hash(type-book-author-Tolkien)
 - Truy vấn "author = Tolkien" sẽ ánh xạ đến đúng tài nguyên.

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

Hình 11. Mô tả thông thường của tài nguyên (Tài liệu)



Hình 12. Biểu diễn của tài nguyên đó dưới dạng cây thuộc tính-giá trị (AVTree) (Tài liệu)

Tổng kết

Tiêu chí	Đặt tên theo cấu trúc (Structured Naming)	Đặt tên theo thuộc tính (Attribute-Based Naming)
Định danh thực thể	Theo cây thư mục, đường dẫn (ví dụ: /org/dep/user)	Theo tập (thuộc tính, giá trị), không cần tên cụ thể
Tính thân thiện người dùng	Cao (tên dễ nhớ, có tổ chức)	Trung bình (cần mô tả chính xác thuộc tính)
Tìm kiếm thực thể	Theo đường dẫn cố định, nhanh nếu biết rõ vị trí	Linh hoạt theo mô tả – hỗ trợ tìm kiếm theo nội dung
Hiệu quả truy vấn	Cao với truy vấn chính xác	Thấp nếu không có chỉ mục – cần tối ưu (indexing, phân vùng...)
Khả năng mở rộng	Hạn chế trong môi trường động hoặc phân tán rộng	Tốt trong hệ thống phân tán lớn, đặc biệt với P2P
Khó khăn khi triển khai	Quản lý cấu trúc tên, phân phối nút trong cây	Thiết kế thuộc tính chuẩn hóa, phân bố chỉ mục, xử lý phân tải
Khả năng mô tả linh hoạt	Hạn chế – phụ thuộc vào cấu trúc cây	Cao – hỗ trợ mô tả đa chiều, mở rộng dễ dàng
Ứng dụng điển hình	DNS, hệ thống file phân cấp, LDAP	Resource discovery, hệ thống P2P, dịch vụ thư mục nâng cao



Học viện
Công nghệ Bưu chính Viễn thông

THANK YOU!