

ỨNG DỤNG PHÂN TÁN

IV. Định danh

Nhóm chuyên môn UDPT – Trường Công nghệ thông tin Phenikaa
(Trần Đăng Hoan, Đỗ Quốc Trường, Nguyễn Thành Trung, Phạm Kim Thành, Nguyễn Hữu Đạt)

NỘI DUNG

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

VÍ DỤ: XÃ HỘI

Tên
Nơi sinh

Số Định danh cá
nhân

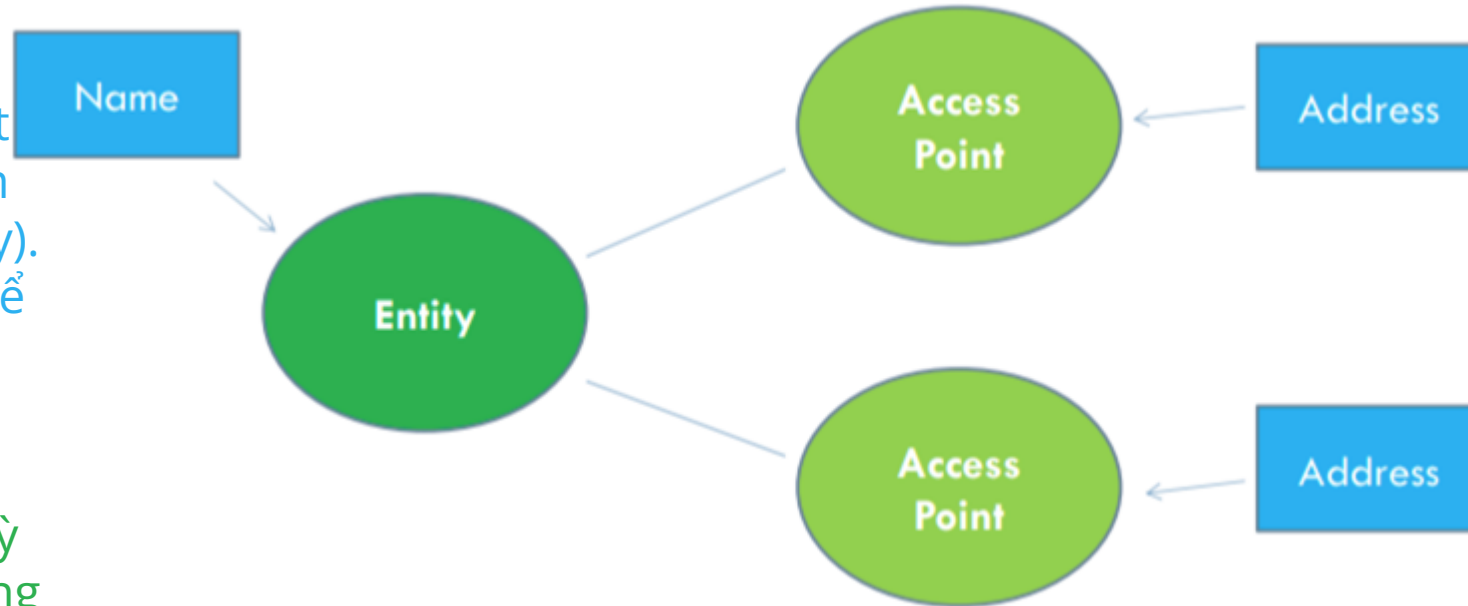


THỰC THỂ VÀ TÊN

Name (Tên): một chuỗi các bit hoặc các ký tự dùng để tham chiếu đến một thực thể (entity). Số lượng tên cho mỗi thực thể không bị hạn chế.

Entity (Thực thể): bất kỳ những gì có trong hệ thống như host, printer, file, người dùng, tiến trình.

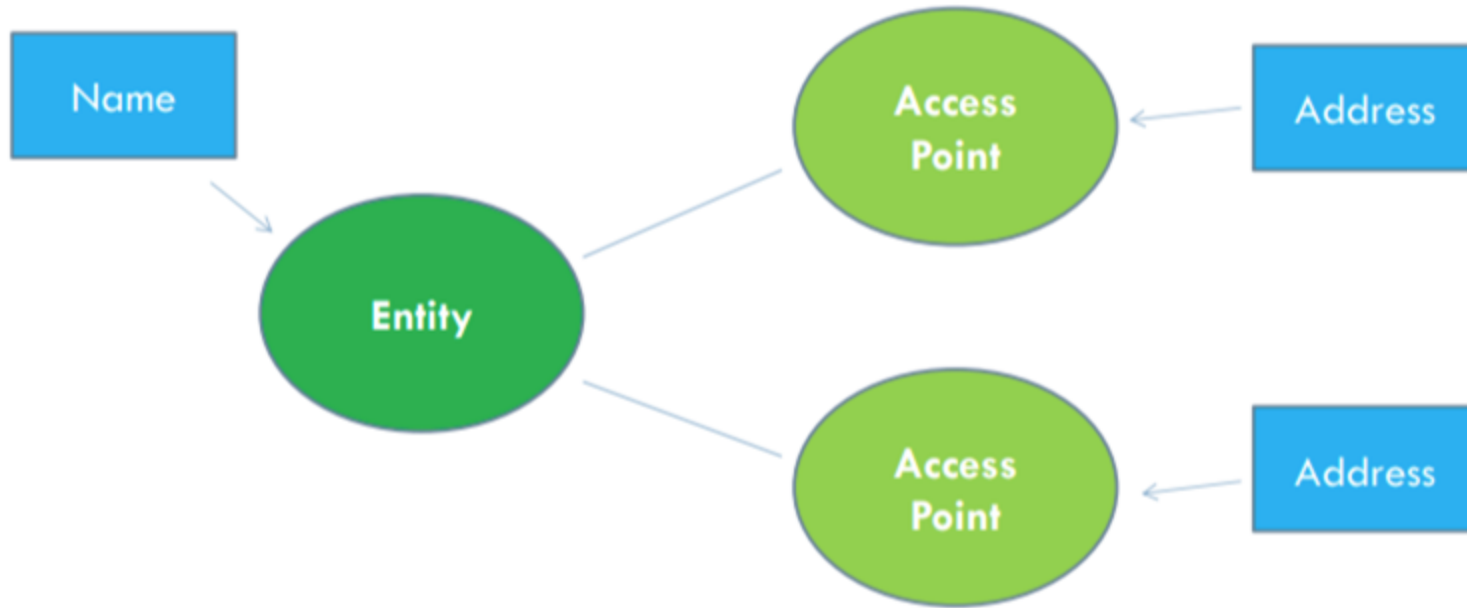
- **Access point (Điểm truy cập):** điểm truy nhập đến thực thể, 1 thực thể có thể có nhiều điểm truy cập



Address (Địa chỉ): tên của access point. Là một loại name đặc biệt:

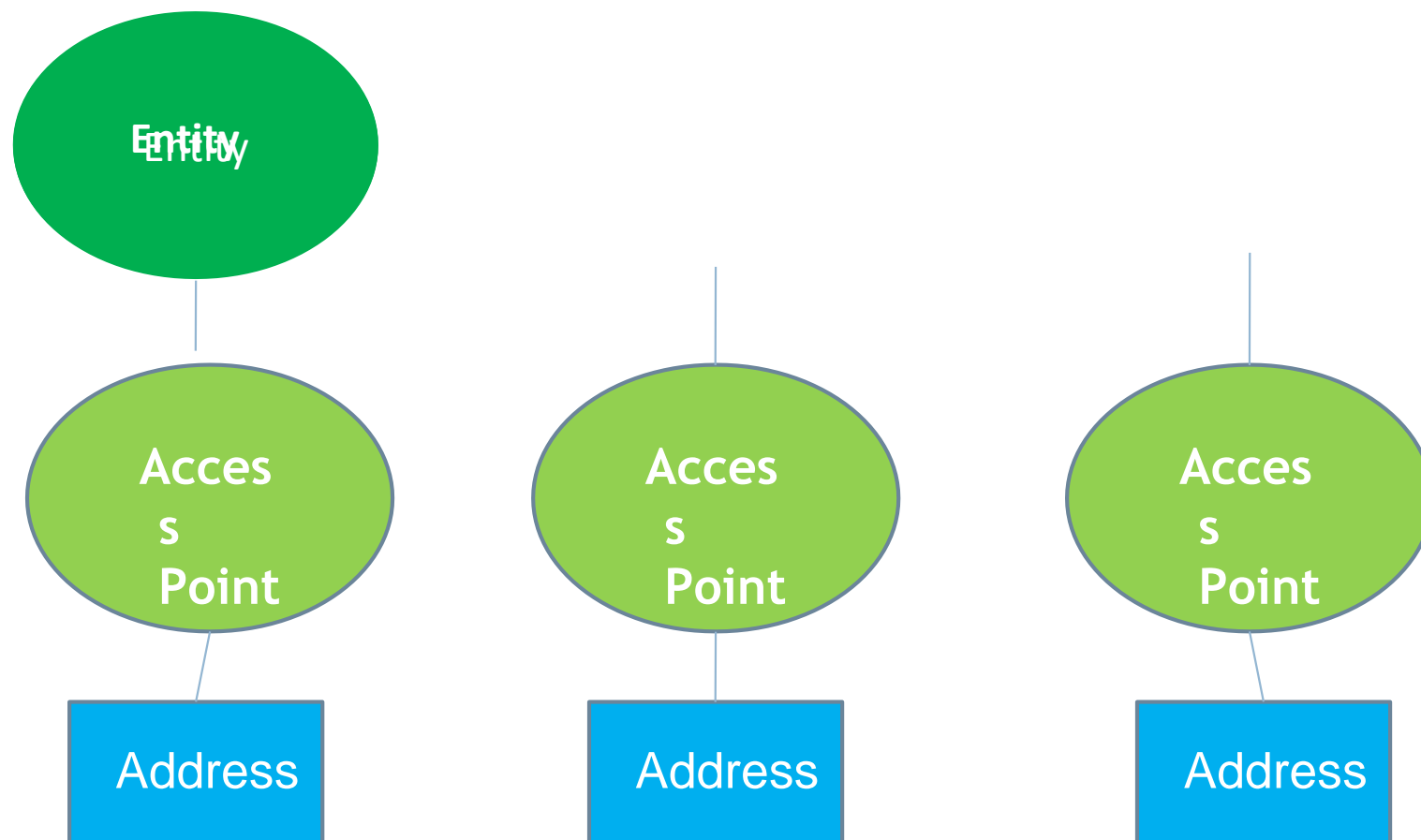
- Một tên có thể gồm nhiều địa chỉ
- Thực thể có thể thay đổi địa chỉ trong quá trình tồn tại
- Một địa chỉ có thể trỏ đến các thực thể khác nhau trong các thời điểm khác nhau
- Đảm bảo có thể tham chiếu tới các tài nguyên bằng tên

THỰC THỂ VÀ TÊN



- Ví dụ: Một máy tính có thể kết nối với 2 mạng khác nhau nếu nó có 2 interface network

THỰC THỂ VÀ TÊN



IDENTIFIER (Định danh)

- **Identifier(Định danh)**: một loại tên đặc biệt, phải đáp ứng ba yêu cầu sau:
 - 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.



truy xuất tới các thực thể trong hệ thống máy tính

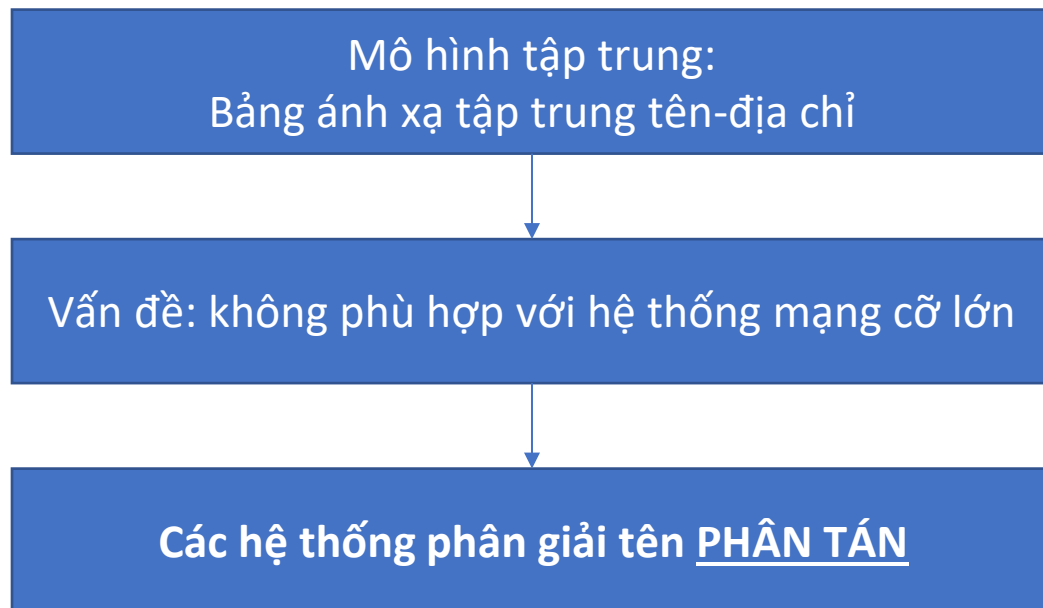
Vấn đề: Cạn kiệt không gian tên

Giải pháp:

- Mở rộng không gian tên
- Tái sử dụng

IDENTIFIER (Định danh)

Phân giải tên và định danh thành địa chỉ



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

MỖI KHÔNG GIAN TÊN CẦN MỘT CƠ CHẾ ĐỊNH DANH KHÁC NHAU

KHÔNG GIAN TÊN PHẪNG

1. Khái niệm
2. Các giải pháp thông thường
3. Giải pháp Home-based
4. Giải pháp sử dụng hàm băm phân tán
5. Giải pháp phân cấp

KHÔNG GIAN TÊN PHẪNG

1. Khái niệm

- Là chuỗi các bit, chuỗi ký tự ngẫu nhiên không cấu trúc.
- Không chứa bất kỳ thông tin nào về cách xác định vị trí các điểm truy cập của thực thể đã được liên kết của chúng.
- Nhiệm vụ: cho biết tên và xác định điểm truy nhập tới các thực thể.
- Các giải pháp để xác định vị trí thực thể trong KGTP:
 - Các giải pháp thông thường
 - Home-base (dựa vào Home Agent)
 - DHT (hàm băm phân tán)
 - Cách tiếp cận phân cấp

KHÔNG GIAN TÊN PHẪNG

2. Các giải pháp thông thường

Thực hiện bằng cách gửi định danh cần tìm theo:

- Phương pháp quảng bá, gửi cho các thành viên của nhóm.
- Sử dụng con trỏ chuyển tiếp.

KHÔNG GIAN TÊN PHẪNG

2. Các giải pháp thông thường

- Phương pháp quảng bá, gửi cho các thành viên của nhóm
 - Điều kiện: hệ phân tán hỗ trợ việc trao đổi thông tin thông qua quảng bá
- B1: Một thông báo có chứa định danh cần phân giải được quảng bá tới tất cả các thực thể trong hệ thống.
- B2: 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.
- B3: 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.

KHÔNG GIAN TÊN PHẪNG

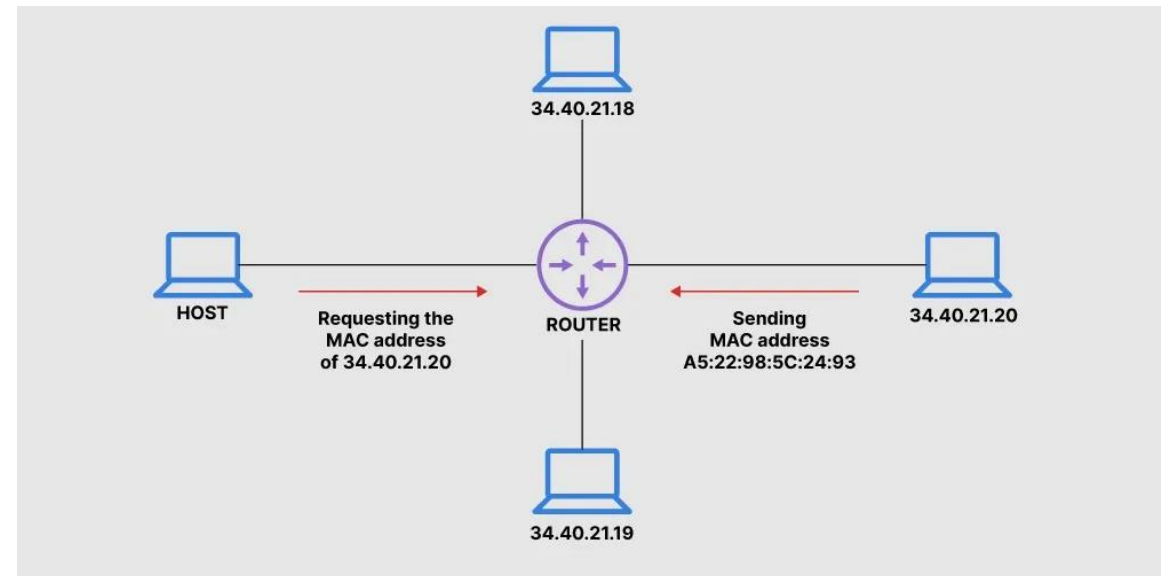
2. Các giải pháp thông thường

- Phương pháp quảng bá, gửi cho các thành viên của nhóm
 - Kém hiệu quả khi kích thước mạng tăng (\Rightarrow mạng cỡ lớn)
 - 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, thay vì gửi tới tất cả các nút.
 - 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

KHÔNG GIAN TÊN PHẪNG

2. Các giải pháp thông thường

- Phương pháp quảng bá, gửi cho các thành viên của nhóm
- Ví dụ: Giao thức ARP



KHÔNG GIAN TÊN PHẪNG

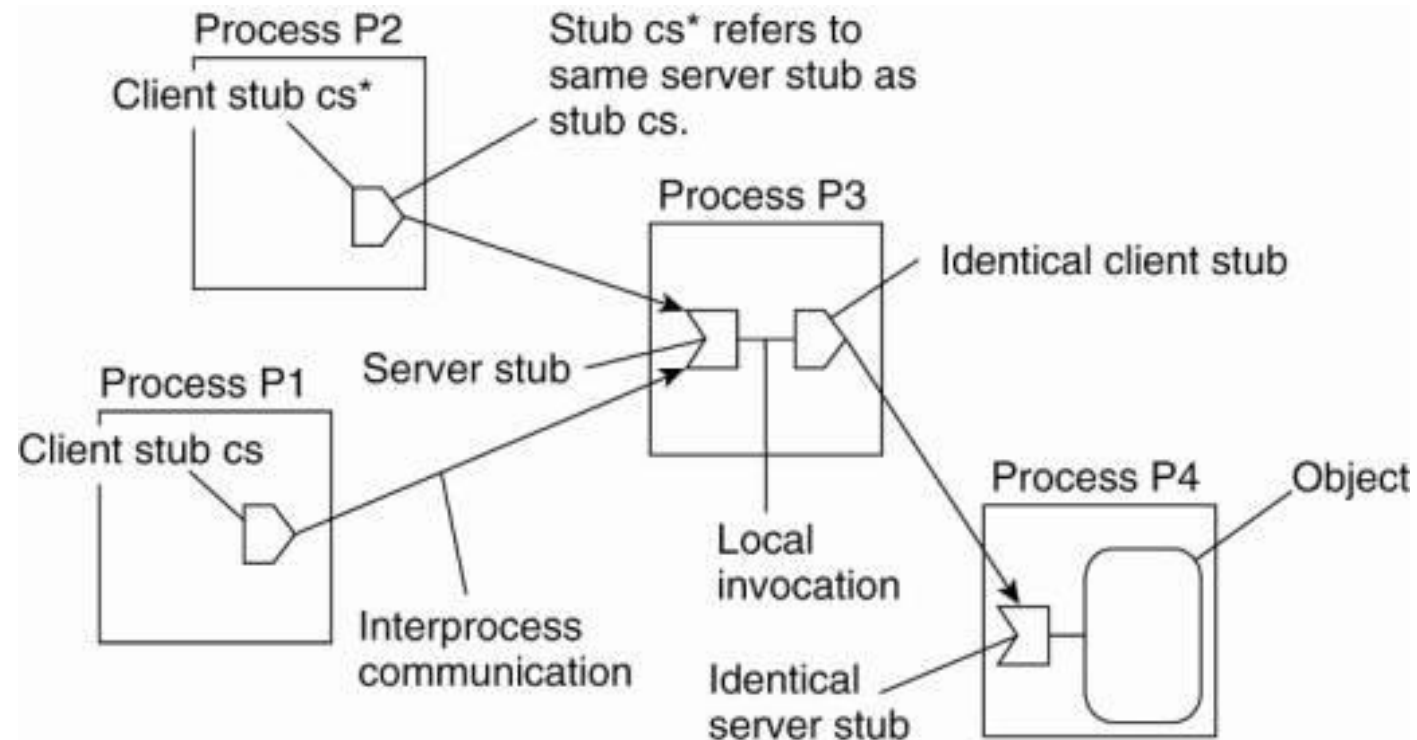
2. Các giải pháp thông thường

- Chuyển tiếp con trỏ (Forwarding pointer)
 - Phương pháp dựa trên nguyên tắc một thực thể khi rời sang vị trí khác thì phải để lại thông tin tham chiếu tới vị trí mới.
 - Khi cần tìm một thực thể, client có thể xác định ngay được địa chỉ hiện tại của thực thể này nhờ phương pháp dò tìm lưu vết.

KHÔNG GIAN TÊN PHẪNG

2. Các giải pháp thông thường

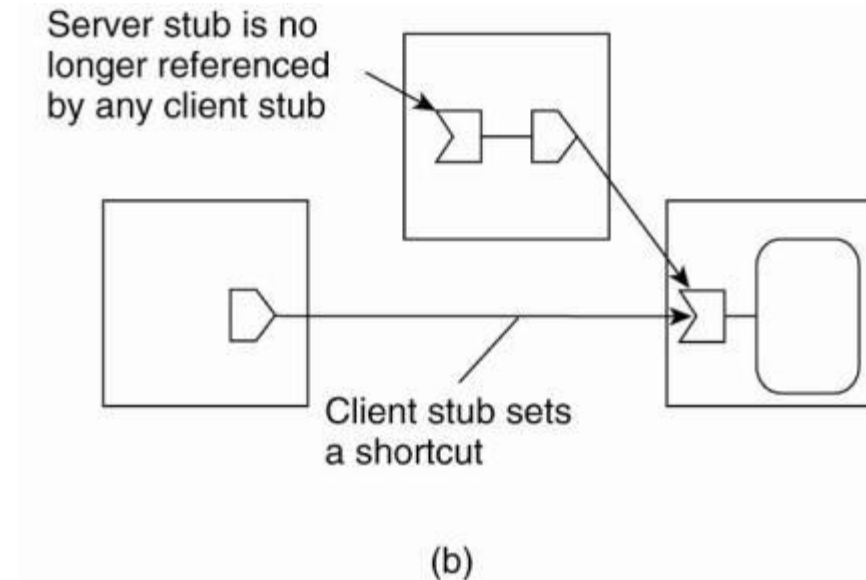
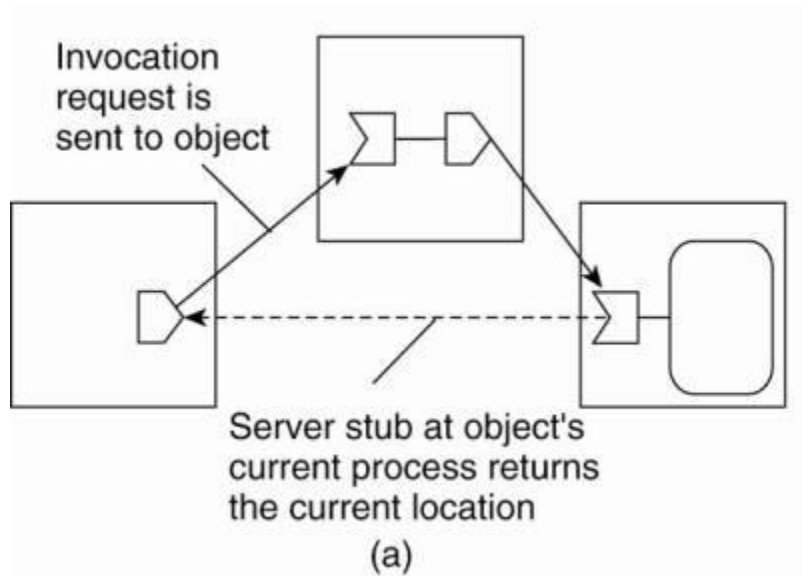
- Chuyển tiếp con trỏ (Forwarding pointer)
 - Cơ chế



KHÔNG GIAN TÊN PHẪNG

2. Các giải pháp thông thường

- Chuyển tiếp con trỏ (Forwarding pointer)
 - Cơ chế



KHÔNG GIAN TÊN PHẪNG

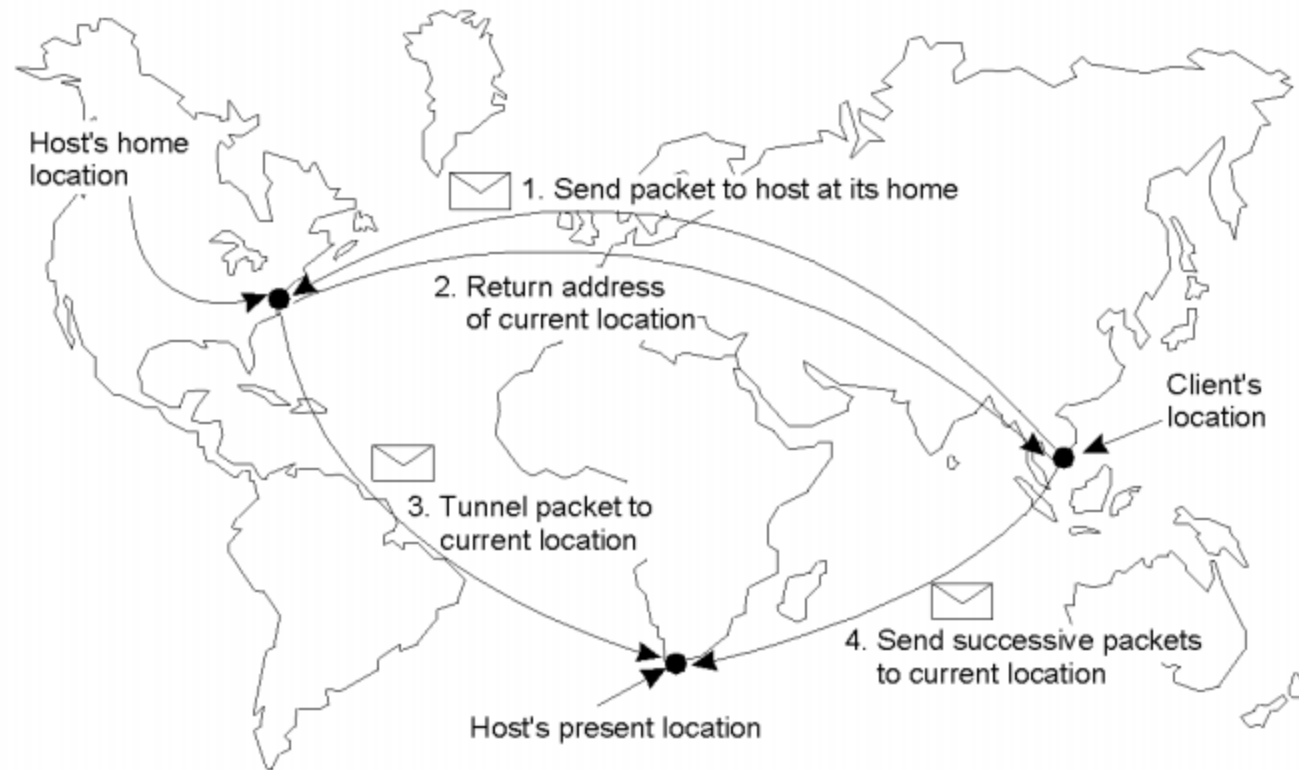
2. Các giải pháp thông thường

- Chuyển tiếp con trỏ (Forwarding pointer)
 - Ưu điểm: đơn giản.
 - Nhược điểm:
 - Chuỗi dài vô hạn
 - Giải pháp: sử dụng các short cut. Tuy nhiên có khả năng có chuỗi không tham chiếu được.
 - Lưu trữ vô số các tham chiếu
 - Giải pháp: loại bỏ các tham chiếu. Tuy nhiên khi nào có thể loại bỏ các tham chiếu

Giải pháp Home-base

KHÔNG GIAN TÊN PHẪNG

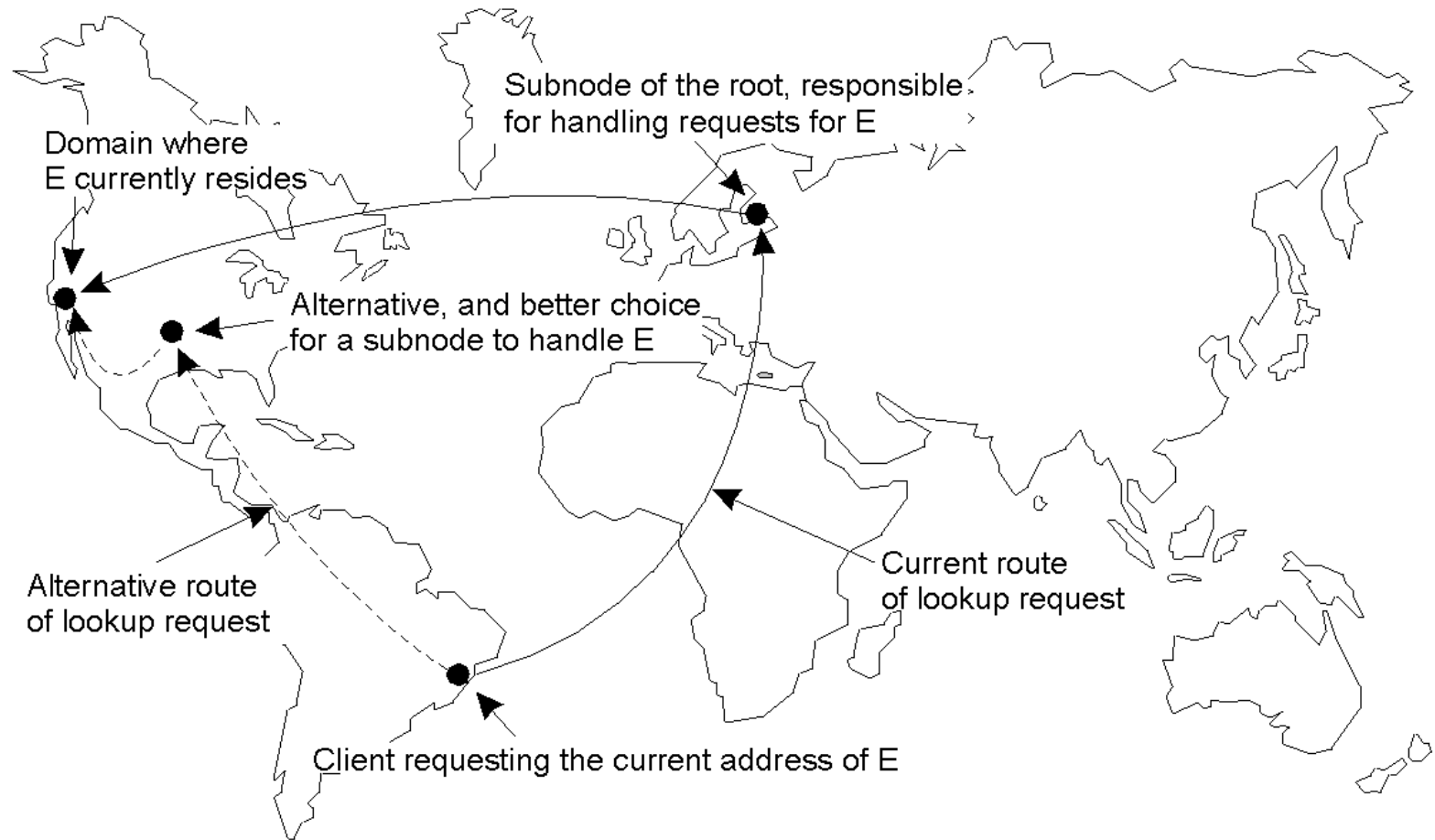
3. Giải pháp Home-base



KHÔNG GIAN TÊN PHẪNG

3. Giải pháp Home-base

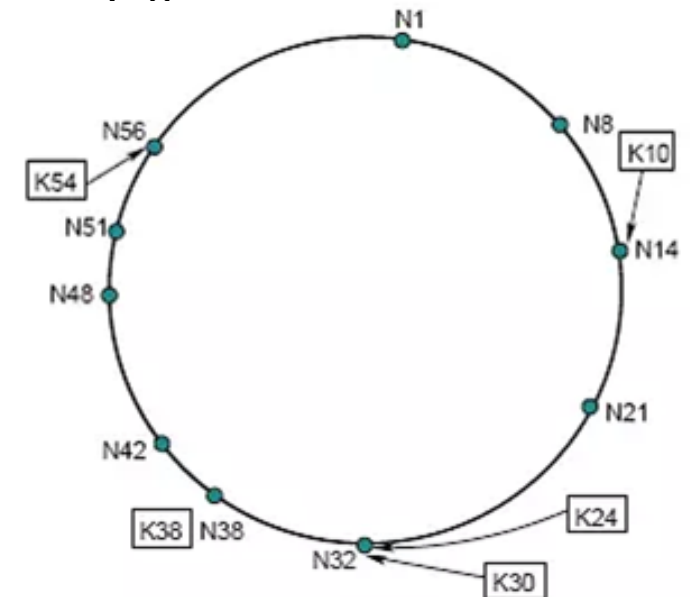
Vấn đề về quy mô



KHÔNG GIAN TÊN PHẪNG

4. Giải pháp sử dụng hàm băm phân tán

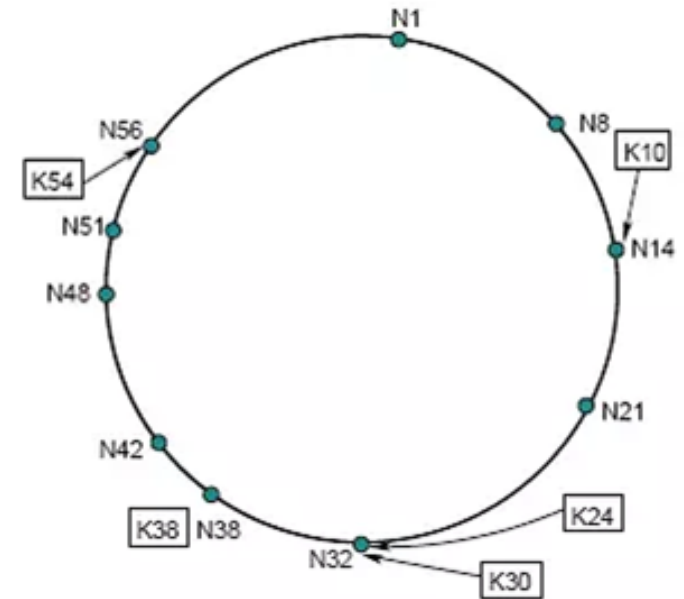
- Xem xét hệ thống Chord
- Chord là một trong những giao thức phổ biến nhất được sử dụng trong bảng băm phân tán.
- Mạng Chord hỗ trợ khả năng gán tương ứng một key cho trước với một nút mạng



KHÔNG GIAN TÊN PHẪNG

4. Giải pháp sử dụng hàm băm phân tán

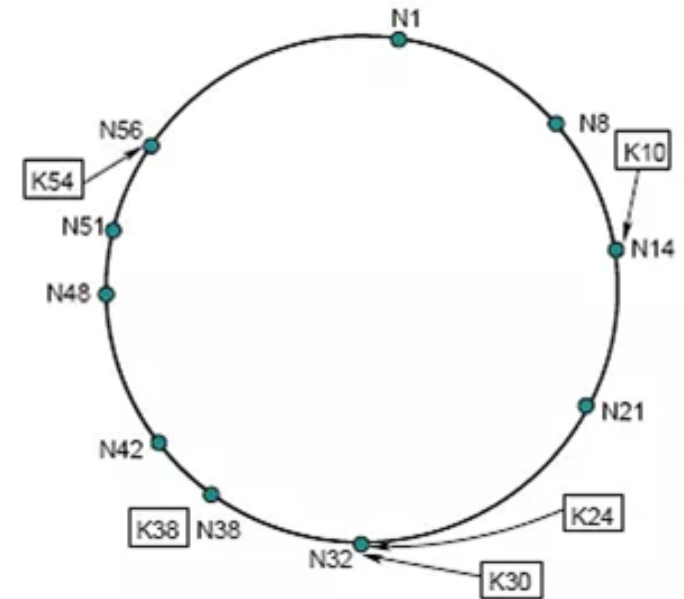
- Xem xét hệ thống Chord
- Các nút trong mạng Chord tạo lên một mạng logic dạng vòng tròn có các vị trí nút từ 0 đến $2^m - 1$.
- Khóa k được gán cho nút đầu tiên có định danh bằng hoặc lớn hơn định danh của k . Nút đó được gọi là nút successor của khóa k hay $\text{successor}(k)$.
- Trong vòng định danh của Chord, successor của một khóa chính là nút gần nhất theo chiều kim đồng hồ tính từ khóa k .



KHÔNG GIAN TÊN PHẪNG

4. Giải pháp sử dụng hàm băm phân tán

- Xem xét hệ thống Chord
- Ví dụ: Xét HT Chord với $m=6$. Vòng Chord có chứa 10 nút và 5 khóa.
 - Successor của định danh 10 là nút 14 do đó key 10 sẽ được đặt ở nút 14. Tương tự khóa 24 và 30 sẽ được đặt ở nút 32, khóa 38 tại nút 38 và khóa 54 tại nút 56.
 - Kỹ thuật consistent hashing được thiết kế để việc các nút tham gia hay rời khỏi mạng sẽ tạo ra ít ảnh hưởng nhất. Để duy trì bảng mapping khi một nút n tham gia vào mạng, một số khóa trước đây được đặt tại successor của n sẽ chuyển sang cho nút n . Trong ví dụ trên, nếu có một nút với định danh 26 tham gia vào mạng, nó sẽ nhận được khóa 24 chuyển từ nút 32.
- Successor của một nút là nút tiếp sau nút đó trên vòng Chord, predecessor là nút liền trước trên vòng Chord.

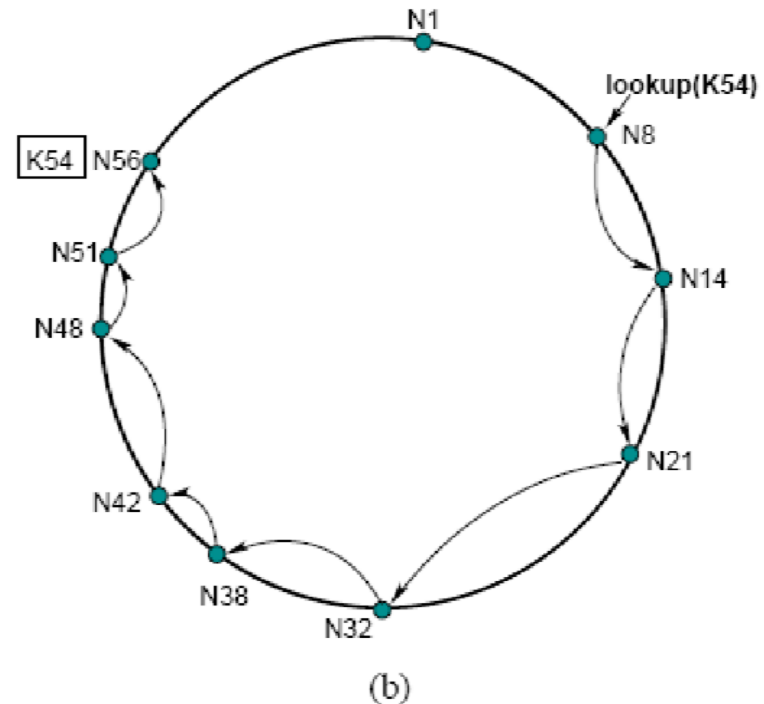


KHÔNG GIAN TÊN PHẪNG

4. Giải pháp sử dụng hàm băm phân tán

- Xem xét hệ thống Chord

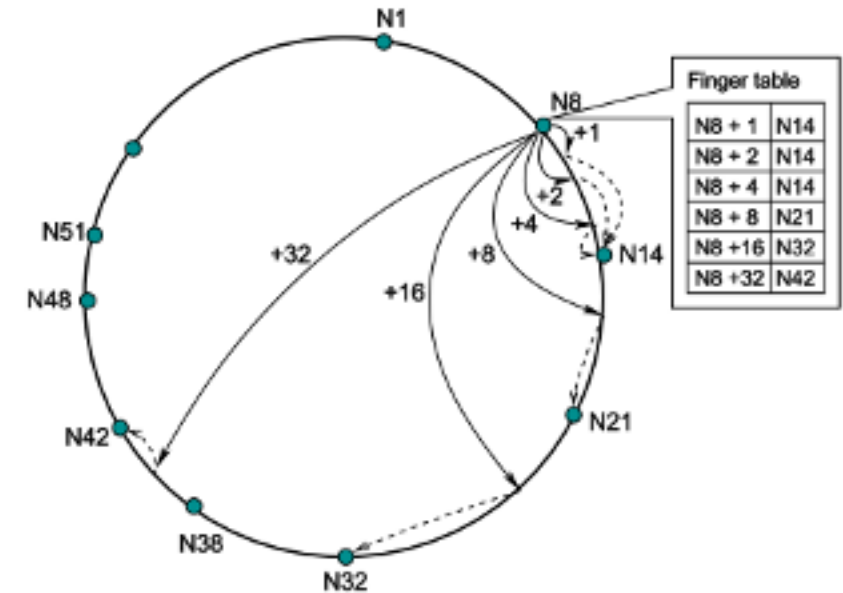
Tìm kiếm đơn giản



KHÔNG GIAN TÊN PHẪNG

4. Giải pháp sử dụng hàm băm phân tán

- Xem xét hệ thống Chord
 - Để tăng tốc độ quá trình tìm kiếm Chord sử dụng thêm một số thông tin định tuyến. Giả sử định danh của mỗi nút và khóa có độ dài m bit. Mỗi nút n duy trì một bảng định tuyến chứa m mục, được gọi là bảng finger



KHÔNG GIAN TÊN PHẪNG

4. Giải pháp sử dụng hàm băm phân tán

- 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 :

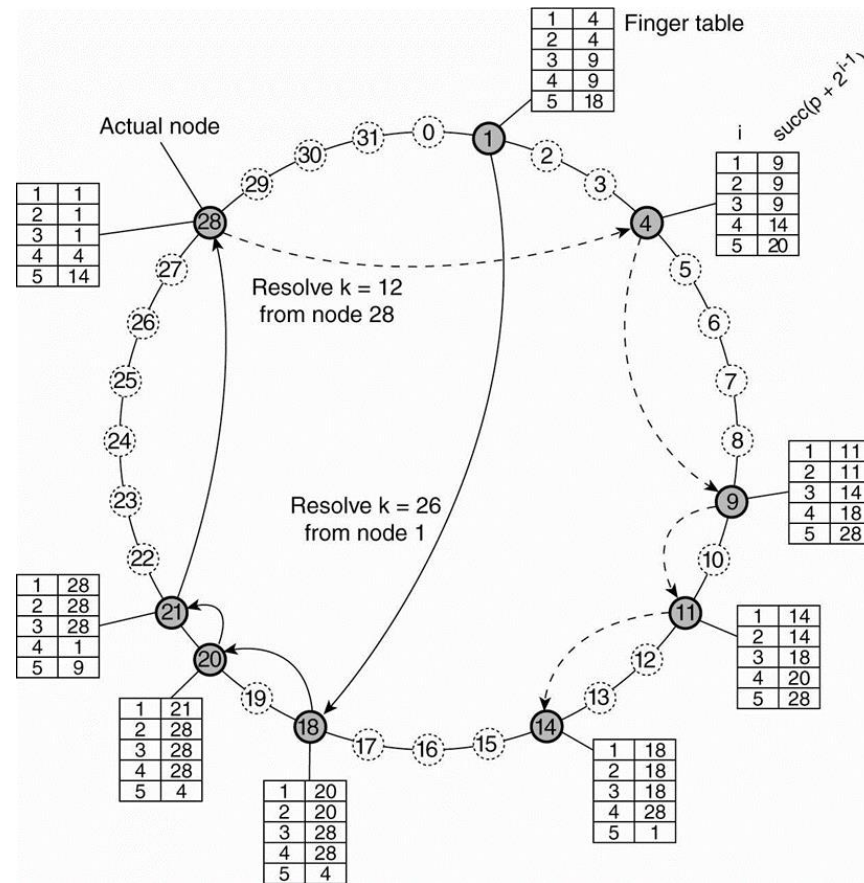
$$q = FT_p[j] \leq k < FT_p[j+1]$$

- Cập nhật bảng băm khi có các nút được thêm vào

KHÔNG GIAN TÊN PHẪNG

4. Giải pháp sử dụng hàm băm phân tán

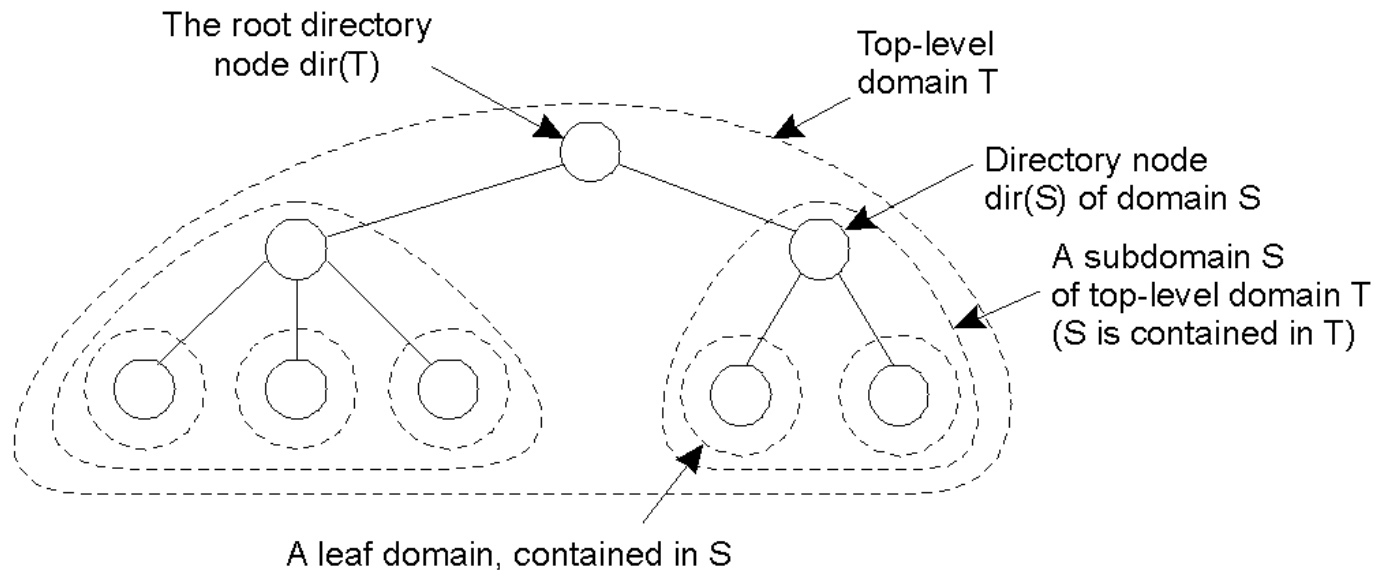
Bảng băm phân tán



KHÔNG GIAN TÊN PHẪNG

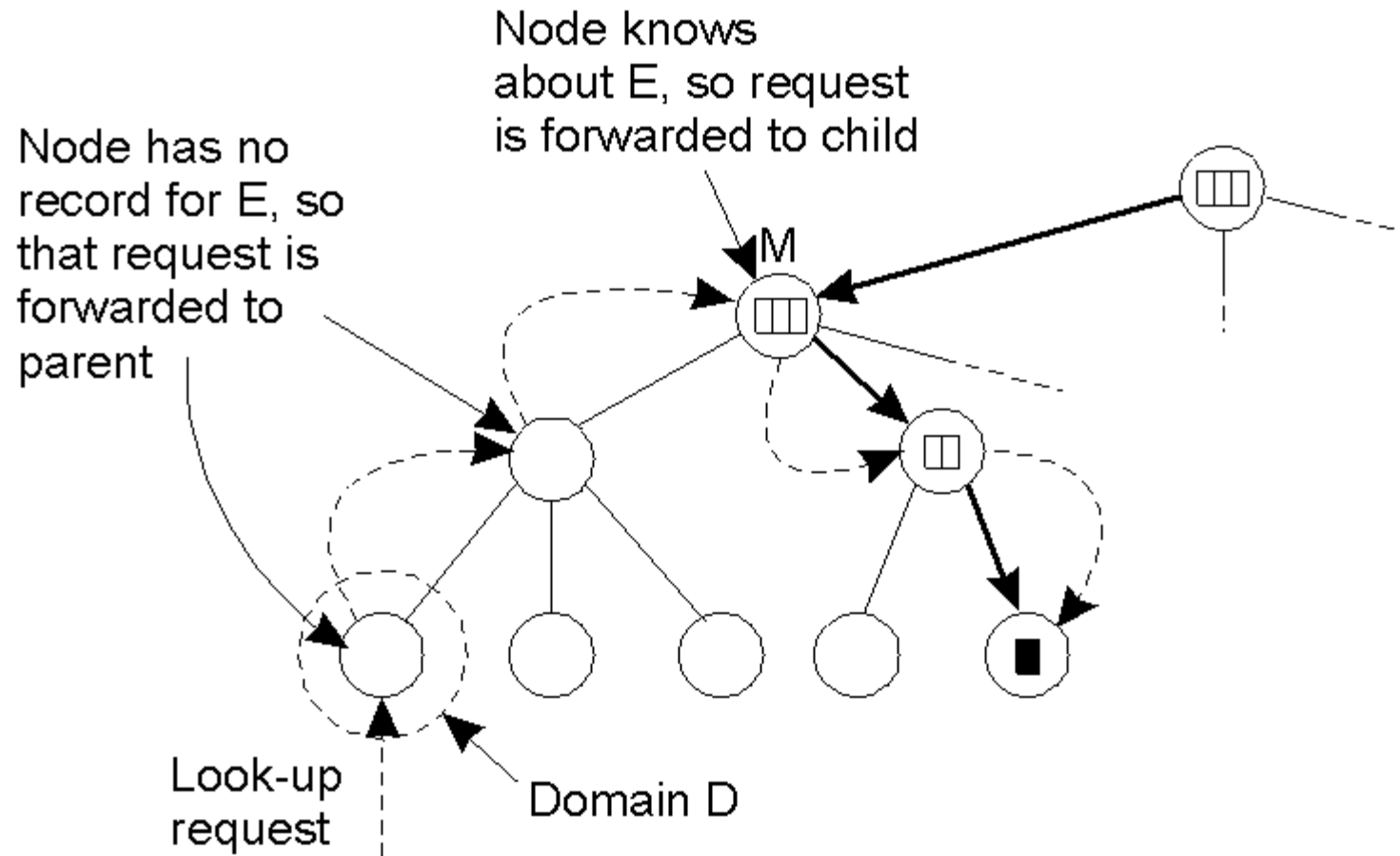
5. Giải pháp phân cấp

- 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



KHÔNG GIAN TÊN PHẪNG

5. Giải pháp phân cấp Tìm kiếm



Bài tập

- Bài tập 1: Tìm hiểu các tài liệu (Không sử dụng và copy các tool LLMs như chatGPT, Perplexity, ClaudeAI, Gemini ...) và diễn giải quy trình khi chúng ta duyệt một trang web www.motvidu.com. Nêu rõ khái quát ngắn gọn về quá trình liên quan đến tra cứu DNS, resolving DNS, caching.
 - Lưu ý: Trích dẫn nguồn tìm hiểu vào trong bài làm
- Bài tập 2: Viết thuật toán Chord bằng ngôn ngữ tự chọn (chẳng hạn như python), viết test case cho thuật toán và viết thành báo cáo. Nêu rõ
 - Ví dụ cụ thể về Chord
 - Ví dụ về kết quả của test case, được tính toán như nào
 - Code thực nghiệm