

CHƯƠNG 5: Định Danh

Họ tên sinh viên: Phan Thành Đạt

MSSV:20173001

Mã Lớp: 118636

Mã học phần: IT4611

Câu 1:

Không thể lấy địa chỉ Access Point làm địa chỉ thực thể vì 1 Access Point có thể có nhiều thực thể và các thực thể có thể di chuyển qua lại giữa các Access Point

Câu 2:

Các vấn đề của hệ thống định danh là :

- Một định danh chỉ gắn với 1 thực thể
- Một thực thể chỉ gắn với một định danh
- Một định danh mãi mãi chỉ trỏ đến 1 thực thể

Vấn đề:

- Không gian định danh hữu hạn.
- Cạn kiệt định danh.

Giải pháp:

- Sử dụng lại những định danh mà thực thể đó đã chết
- Có khả năng bị trùng lặp.

Câu 3:

Mô hình kết hợp cả dịch vụ tổ chức vị trí phân cấp và cơ chế chuyển tiếp con trỏ (forwarding pointers)

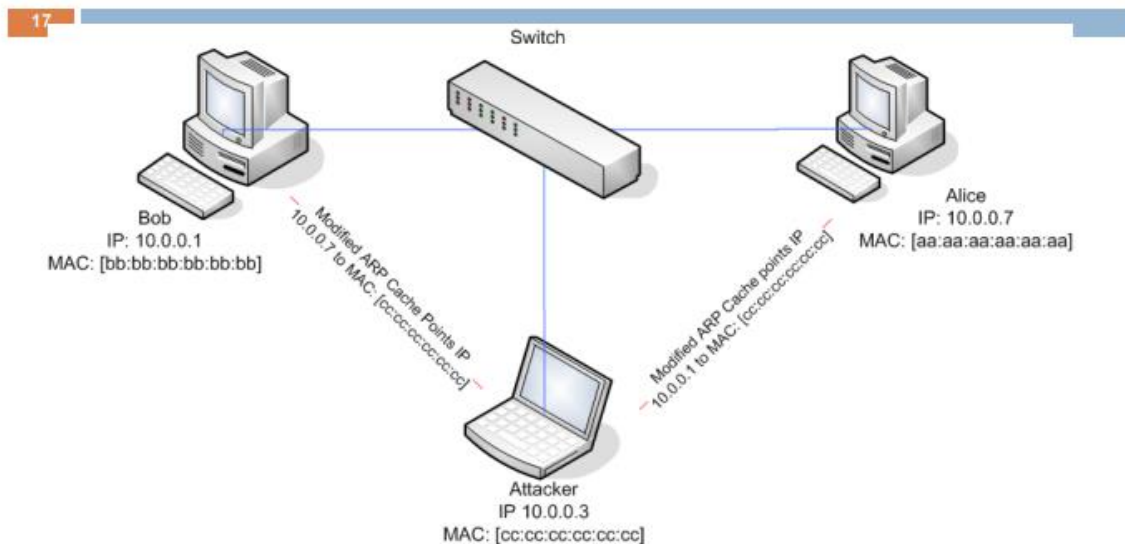
để có thể xác định được vị trí của thực thể khi nó ở các nút trung gian.

- Khi thông tin dịch chuyển vị trí trung gian A và B, để lại con trỏ khi đến B ghi thêm địa chỉ mới đó ==> Mô hình phân cấp.
- Sau khi thực thể đến B sẽ trỏ thẳng đến vị trí hiện tại của thực thể và xóa các con trỏ trung gian đi.

Câu 4:

ARP không có cơ chế bảo mật về tính xác thực, vì vậy hoàn toàn có thể có 1 kẻ tấn công đứng ở giữa đọc toàn bộ gói tin ARP và giả mạo cả 2 bên. Ví dụ trong hình là kẻ tấn công đứng ở giữa, đọc được gói tin của Bob cần tìm địa chỉ MAC của Alice, Attacker sẽ giả mạo địa chỉ của Alice là địa chỉ của mình, gửi response lại cho B, rồi giả danh Bob gửi cho Alice và nói rằng địa chỉ MAC của mình là địa chỉ của Bob, khi đó A và B gửi tin cho nhau thì Attacker đều nhận được.

ARP-Spoofing



Câu 5:

Các vấn đề với cơ chế chuyển tiếp con trỏ:

- Số lượng các bước trung gian nhiều, chuỗi dài vô hạn => lưu trữ và quá trình tìm kiếm rất thời gian
- Nếu như nút trung gian nào đó có vấn đề => mất địa chỉ hiện tại của thực thể.

Câu 6:

Nhược điểm của Home-based:

- Vấn đề về qui mô, địa lý: Giả sử thực thể A sinh ra ở Châu Mỹ thì sẽ gắn với Home-based ở Châu Mỹ, sau một thời gian thực thể A đã di chuyển đến châu Á nhưng vẫn gắn với Home-based ở Mỹ, một client ở châu Á muốn tìm thực thể sẽ phải sang tận Mỹ để hỏi mà trong khi thực thể đó đã di chuyển rất xa ở châu Mỹ và ngay ở gần mình.
- ⇒ Giải pháp: mỗi khi thực thể di chuyển đủ xa và đủ lâu thì sẽ gắn nó với Home-based gần nó hơn.

Câu 7:

Khi không áp dụng bảng băm, muốn truy vấn 1 tài nguyên nào đó thì ta phải truy vấn lần lượt qua từng host, khi đã có bảng băm thì ta sẽ dựa vào bảng băm mà nhảy cóc lần lượt qua các hosts trung gian để lấy được tài nguyên.

Câu 8:

Sử dụng bộ đệm sẽ giúp giảm thời gian tìm định danh, các định danh đã tìm thấy sẽ được lưu vào cache, lần tới sẽ không cần phải tìm kiếm nữa.

Không sử dụng cache	Sử dụng cache
<ul style="list-style-type: none"> • Đi từ lá -> node cha -> root -> xuống • Tới khi tìm thấy thì dừng. • Không lưu trữ kết quả. • Lần tới lặp lại các bước như trên tới khi tìm thấy 	<ul style="list-style-type: none"> • Đi từ lá -> node cha -> root -> xuống • Tới khi tìm thấy thì dừng. • Lưu kết quả tìm được vào cache. • Lần tới truy suất ngay kết quả thay vì tìm lại từ đầu.

Câu 9:

Liên kết vật lý	Liên kết biểu tượng
<ul style="list-style-type: none"> • Một liên kết vật lý là một quan hệ giữa tên tệp trong thư mục với một inode. • Có thể nhiều liên kết vật lý - 1 inode. 	<ul style="list-style-type: none"> • Khi tạo liên kết biểu tượng, một inode mới được tạo ra. • Inode này chứa tên của phần tử được trỏ tới. • Khi xóa một file, các file khác không ảnh hưởng.

<ul style="list-style-type: none"> • Liên kết cuối cùng tới inode bị xóa thì các khối liên quan tới inode cũng bị xóa 	
--	--

Câu 10:

Không cần phải cập nhật toàn bộ các mảng finger, chỉ cần cập nhật cho nút sau và nút trước

Vì trong hệ thống Chord, mỗi nút sẽ quản lí khóa của nút trước đó.

Câu 11:

Đệ quy	Không đệ quy
<ul style="list-style-type: none"> • Xử lí nhiều ở bên server. • Các name server tổ chức lưu trữ cache: Server lưu trữ kết quả và có thể trả lời nhanh cho client. 	<ul style="list-style-type: none"> • Xử lí nhiều ở bên client. • Nếu client ở xa độ trễ tăng do việc gửi đi gửi lại