



ISO 9001:2000

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

HUYỀN NGUYỄN CHÍNH



GIÁO TRÌNH

MẠNG MÁY TÍNH NÂNG CAO



NHÀ XUẤT BẢN
ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
THÀNH PHỐ HỒ CHÍ MINH**

HUỲNH NGUYỄN CHÍNH

**GIÁO TRÌNH
MẠNG MÁY TÍNH NÂNG CAO**

**NHÀ XUẤT BẢN ĐẠI HỌC QUỐC GIA
THÀNH PHỐ HỒ CHÍ MINH**

LỜI NÓI ĐẦU

Tài liệu này được biên soạn dựa theo đề cương chi tiết của chương trình đào tạo mới (150TC) của trường ĐH Sư phạm Kỹ thuật TP HCM. Tài liệu được biên soạn nhằm cung cấp những kiến thức nền tảng, giúp sinh viên nắm vững và vận dụng được các kỹ thuật phổ biến trên hạ tầng mạng. Từ đó, sinh viên có thể tự học các kiến thức chuyên sâu hơn.

Trong tài liệu này, tác giả sử dụng tập lệnh trên thiết bị Cisco làm các ví dụ minh họa cho các công nghệ được trình bày. Qua việc nắm bắt các kiến thức và kỹ năng thực hành cơ bản, sinh viên có thể vận dụng triển khai, cấu hình trên các thiết bị khác.

Tác giả rất mong nhận được những ý kiến đóng góp từ phía người đọc để tài liệu ngày một hoàn thiện hơn.

MỤC LỤC

| | |
|---|-----------|
| Chương 1. ĐỊNH TUYẾN..... | 7 |
| 1. Giới thiệu..... | 7 |
| 2. Phân loại định tuyến..... | 8 |
| 3. Định tuyến tĩnh..... | 11 |
| 4. RIP..... | 13 |
| 5. OSPF..... | 17 |
| 6. EIGRP..... | 22 |
| 7. Phân phối giữa các giao thức định tuyến..... | 26 |
| 8. Tổng kết chương..... | 30 |
| 9. Câu hỏi và bài tập..... | 31 |
| | |
| Chương 2. VLAN..... | 35 |
| 1. Giới thiệu..... | 35 |
| 2. VLAN..... | 36 |
| 3. Phân loại..... | 37 |
| 4. Cấu hình VLAN..... | 38 |
| 5. Đường trunk..... | 40 |
| 6. VLAN Trunking protocol (VTP)..... | 42 |
| 7. Định tuyến giữa các VLAN..... | 46 |
| 8. Giao thức STP (Spanning Tree Protocol)..... | 51 |
| 9. Tổng kết chương..... | 54 |
| 10. Câu hỏi và bài tập..... | 55 |
| | |
| Chương 3. ACL..... | 59 |
| 1. Giới thiệu..... | 59 |
| 2. Phân loại và hoạt động của ACL..... | 60 |
| 3. Cấu hình ACL..... | 60 |
| 4. Standard ACL..... | 62 |

| | |
|---------------------------------------|------------|
| 5. Extended ACL..... | 65 |
| 6. Named ACL | 67 |
| 7. Tổng kết chương | 70 |
| 8. Câu hỏi và bài tập..... | 70 |
| Chương 4. NAT | 75 |
| 1. Giới thiệu..... | 75 |
| 2. Static NAT..... | 76 |
| 3. Dynamic NAT | 78 |
| 4. NAT Overload..... | 79 |
| 5. Tổng kết chương | 81 |
| 6. Câu hỏi và bài tập..... | 81 |
| Chương 5. CÁC DỊCH VỤ WAN..... | 87 |
| 1. Giới thiệu..... | 87 |
| 2. Kết nối serial Point-to-Point..... | 88 |
| 3. Frame Relay | 94 |
| 4. Tổng kết chương | 106 |
| 5. Câu hỏi và bài tập..... | 107 |
| Chương 6. VPN | 113 |
| 1. Giới thiệu..... | 113 |
| 2. Các thành phần của VPN | 114 |
| 3. Các loại VPN..... | 116 |
| 4. Tổng kết chương | 122 |
| 5. Câu hỏi và bài tập..... | 122 |

Chương 1

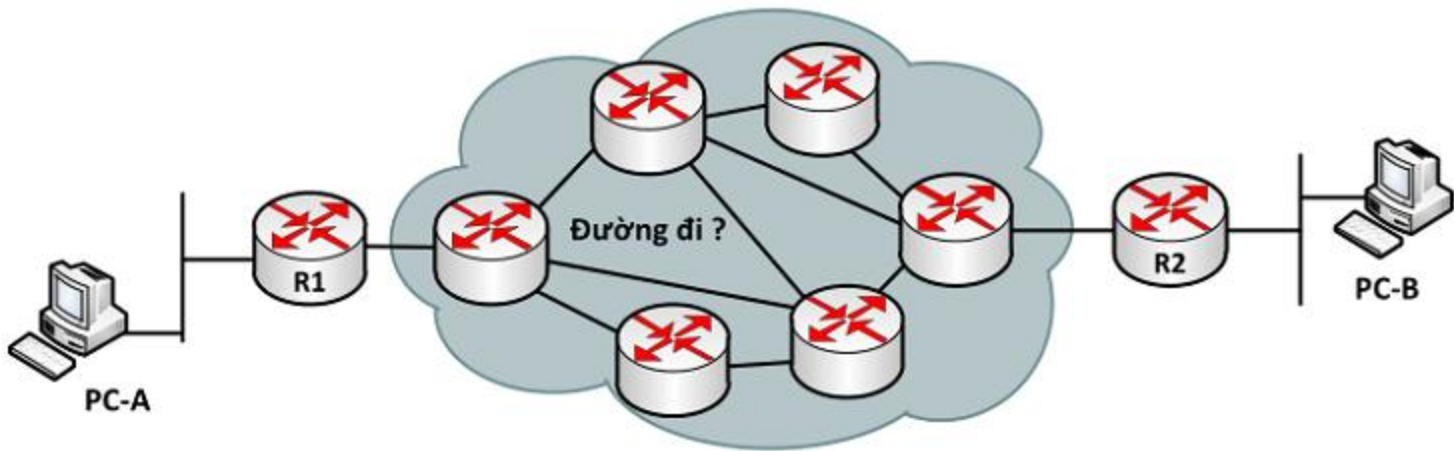
ĐỊNH TUYẾN

Chương này trình bày một số vấn đề cơ bản về định tuyến, phân loại định tuyến, đặc điểm của một số giao thức định tuyến phổ biến và cách cấu hình trên thiết bị của Cisco. Học xong chương này, người học có khả năng:

- Phân biệt được định tuyến tĩnh và định tuyến động
- Phân biệt được giao thức định tuyến dạng distance-vector, link-state, classful và classless
- Trình bày được đặc điểm của các giao thức RIP, OSPF, EIGRP
- Cấu hình định tuyến tĩnh, định tuyến động bằng các giao thức RIP, OSPF, EIGRP

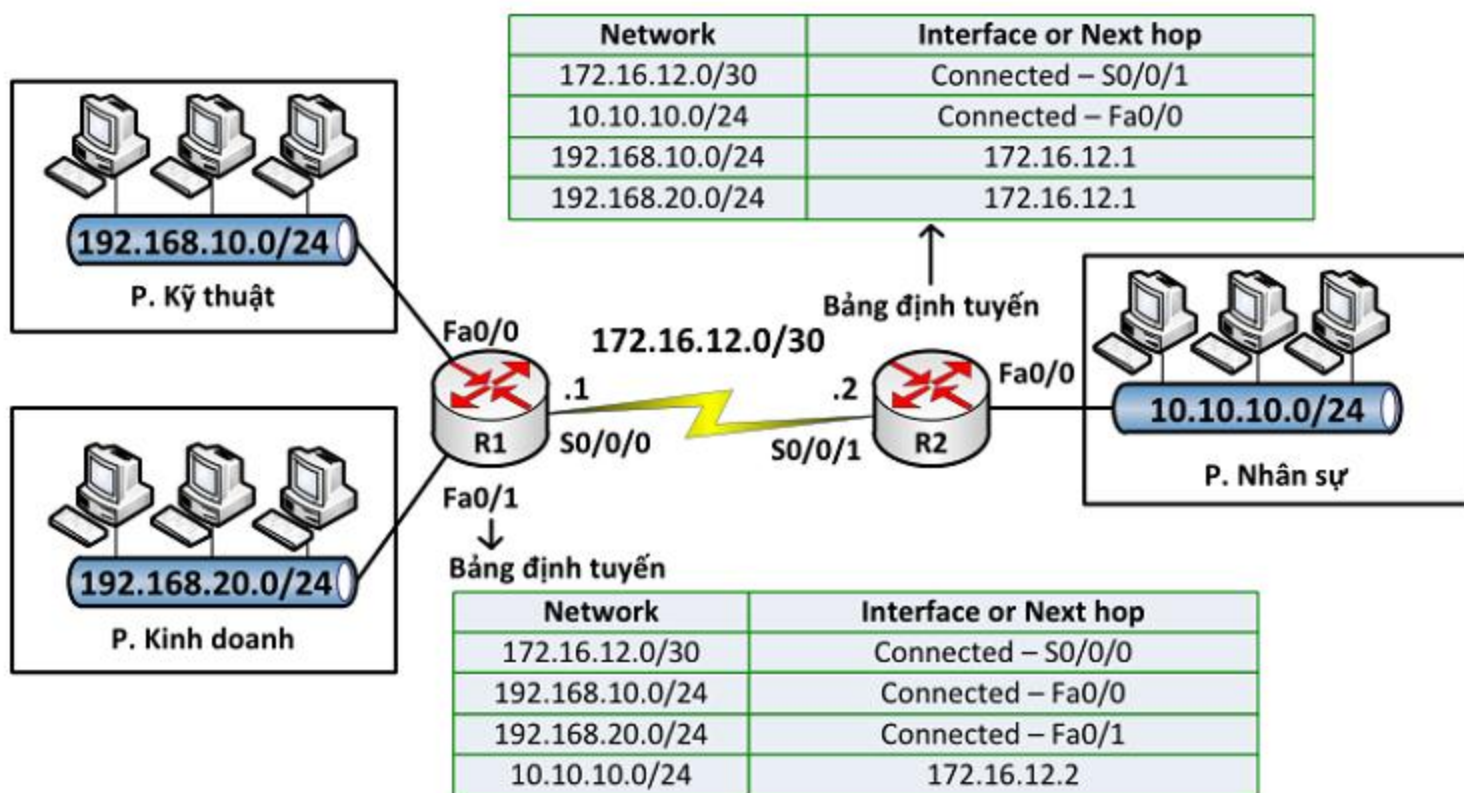
1. GIỚI THIỆU

Định tuyến là chức năng của router giúp xác định đường đi cho các gói tin từ nguồn tới đích thông qua hệ thống mạng.



Hình 1.1 Mô hình hệ thống mạng

Router dựa vào địa chỉ IP đích (destination IP) trong các gói tin và sử dụng bảng định tuyến (routing table) để xác định đường đi cho chúng.



Hình 1.2 Bảng định tuyến trên router

Trong bảng định tuyến, mỗi mạng mà router có thể chuyển đi (mạng đích) thể hiện bằng một dòng. Mỗi mạng này có được có thể do chúng đang kết nối trực tiếp với router đang xét hay router học được thông qua việc cấu hình định tuyến.

2. PHÂN LOẠI ĐỊNH TUYẾN

Có hai loại định tuyến: định tuyến tĩnh và định tuyến động.

• Định tuyến tĩnh

Định tuyến tĩnh là loại định tuyến mà trong đó router sử dụng các tuyến đường đi tĩnh để vận chuyển dữ liệu đi. Các tuyến đường đi tĩnh này có được do người quản trị cấu hình thủ công vào các router.

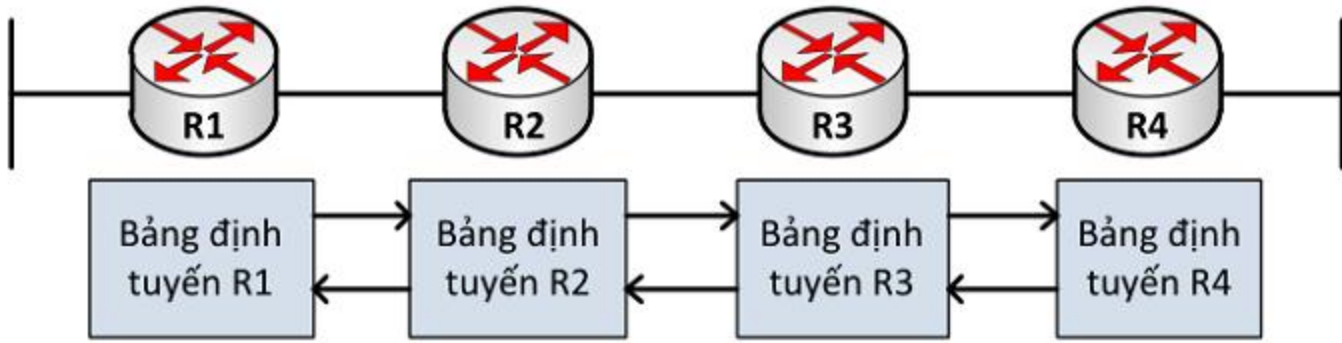
• Định tuyến động

Định tuyến động là loại định tuyến mà trong đó router sử dụng các tuyến đường đi động để vận chuyển dữ liệu đi. Các tuyến đường đi động này có được do các router sử dụng các giao thức định tuyến động trao đổi thông tin định tuyến với nhau tạo ra.

Một số giao thức định tuyến động phổ biến: RIP, OSPF, BGP,...

- ❖ Trong định tuyến động, người ta chia ra thành 2 loại: *distance-vector* và *link-state*

✓ *Distance vector*



Hình 1.3 Trao đổi thông tin định tuyến dạng distance-vector

Các router định tuyến loại “*distance vector*” thực hiện gửi định kỳ toàn bộ bảng định tuyến của mình và chỉ gửi cho các router láng giềng kết nối trực tiếp với mình.

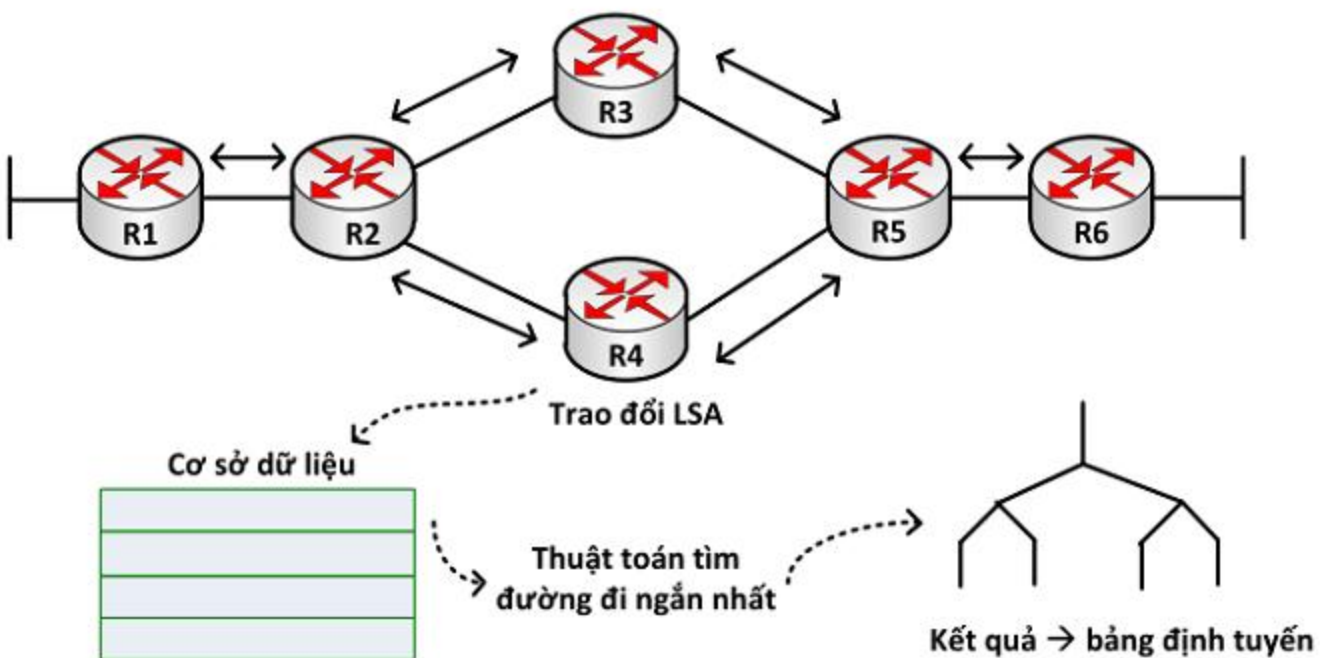
Các router định tuyến theo dạng này không biết được đường đi đến đích một cách cụ thể, không biết về các router trung gian trên đường đi và cấu trúc kết nối giữa chúng.

Bảng định tuyến là nơi lưu kết quả chọn đường tốt nhất của mỗi router. Do đó, khi chúng trao đổi bảng định tuyến với nhau, các router chọn đường dựa trên kết quả đã chọn của router láng giềng. Mỗi router nhìn hệ thống mạng theo sự chi phối của các router láng giềng.

Các router định tuyến theo “*distance vector*” thực hiện cập nhật thông tin định tuyến theo định kỳ nên tốn nhiều băng thông đường truyền. Khi có sự thay đổi xảy ra, router nào nhận biết sự thay đổi đầu tiên sẽ cập nhật bảng định tuyến của mình trước rồi chuyển bảng định tuyến cập nhật cho các router láng giềng.

Giao thức định tuyến thuộc loại này: RIP

✓ *Link-state*



Hình 1.4 Trao đổi thông tin định tuyến dạng link-state

Trong các giao thức định loại *link-state*, các router sẽ trao đổi các LSA (*link state advertisement*) với nhau để xây dựng và duy trì cơ sở dữ liệu về trạng thái các đường liên kết hay còn gọi là cơ sở dữ liệu về cấu trúc mạng (*topology database*). Các thông tin trao đổi được gửi dưới dạng multicast.

Như vậy mỗi router đều có một cái nhìn đầy đủ và cụ thể về cấu trúc của hệ thống mạng. Từ đó mỗi router sẽ dùng thuật toán tìm đường đi ngắn nhất (SPF - Shortest Path First) để tính toán chọn đường đi tốt nhất đến từng mạng đích.

Khi các router định tuyến theo *link state* đã hội tụ xong, nó không thực hiện cập nhật định tuyến định kỳ mà chỉ cập nhật khi nào có sự thay đổi xảy ra. Do đó, thời gian hội tụ nhanh và ít tốn băng thông.

Giao thức định tuyến theo *link-state* có hỗ trợ CIDR, VLSM nên chúng là một chọn lựa tốt cho các mạng lớn và phức tạp. Nhưng đồng thời nó đòi hỏi dung lượng bộ nhớ lớn và khả năng xử lý mạnh của CPU trên các router.

Để đảm bảo cho các cơ sở dữ liệu cập nhật thông tin mới, trong các LSA này được đánh thêm chỉ số tuần tự "*sequence*". Chỉ số "*sequence*" được bắt đầu từ giá trị *initial* đến giá trị *max-age*. Khi một router nào đó tạo ra một LSA, nó sẽ đặt giá trị *sequence* bằng *initial*. Mỗi khi router gửi ra một phiên bản LSA, nó sẽ tăng giá trị đó lên 1. Như vậy, giá trị *sequence* càng cao thì thông tin LSA càng mới. Nếu giá trị *sequence* này đạt đến *max-age*, router sẽ gửi LSA ra cho tất cả các router còn lại, sau đó router đó sẽ đặt lại giá trị *sequence* về *initial*.

Một số giao thức định tuyến thuộc loại này: OSPF, IS-IS

- ❖ Ngoài cách phân loại như trên, người ta còn chia giao thức định tuyến động theo 2 dạng: "*classful routing protocol*" và "*classless routing protocol*"

✓ **Giao thức định tuyến dạng Classfull**

Các giao thức định tuyến nhóm *classfull* không quảng bá *subnet-mask* cùng với địa chỉ mạng quảng bá trong các gói tin cập nhật định tuyến. Do đó, khi router nhận được các cập nhật này, router phải lấy giá trị *network-mask* mặc định có cùng với địa chỉ lớp mạng của địa chỉ đích. Nếu địa chỉ mạng đó được kết nối trực tiếp với router, *network-mask* được lấy cùng với *network-mask* được cấu hình trên cổng kết nối đến mạng đó. Nếu địa chỉ mạng đích không nối trực tiếp, router sẽ lấy địa chỉ *subnet-mask* mặc định của địa chỉ mạng đích.

Các giao thức thuộc loại này không hỗ trợ mạng VLSM, tóm tắt các tuyến tự động. Giao thức định tuyến thuộc dạng này: RIPv1.

✓ *Giao thức định tuyến dạng Classless*

Các giao thức định tuyến thuộc nhóm *classless* sẽ quảng bá thông tin “*subnet mask*” cùng với địa chỉ mạng quảng bá trong các gói tin cập nhật định tuyến, hỗ trợ VLSM, cho phép tóm tắt các tuyến một cách thủ công.

Các giao thức định tuyến thuộc dạng này: RIPv2, OSPF, EIGRP

❖ Hai tham số dùng trong định tuyến: Metric và AD

✓ *Metric*

Là tham số được sử dụng để chọn đường tốt nhất cho việc định tuyến. Đây là giá trị mà bất kỳ giao thức định tuyến nào cũng phải dùng để tính toán đường đi đến mạng đích.

Trong trường hợp có nhiều đường đi đến một mạng đích thì đường đi nào có *metric* thấp nhất sẽ được lựa chọn để đưa vào bảng định tuyến. Mỗi giao thức định tuyến có một kiểu *metric* khác nhau.

✓ *AD*

AD (Administrative Distance) là giá trị quy ước dùng để chỉ độ tin cậy của các giao thức định tuyến, giao thức nào có AD nhỏ hơn sẽ được xem là đáng tin cậy hơn. Trong trường hợp router học được một mạng đích thông qua nhiều giao thức định tuyến khác nhau, thì tuyến của giao thức định tuyến nào có AD nhỏ nhất thì sẽ được lựa chọn và đưa vào bảng định tuyến.

3. ĐỊNH TUYẾN TĨNH

Trong cấu hình định tuyến tĩnh, người quản trị phải cấu hình thủ công chỉ ra đường đi đến tất cả các mạng đích trên các router trong hệ thống. Định tuyến tĩnh không có hoạt động gửi thông tin cập nhật như các giao thức định tuyến động.

Lưu ý: mặc định router sẽ biết được đường đi đến các mạng đích đang kết nối trực tiếp với nó.

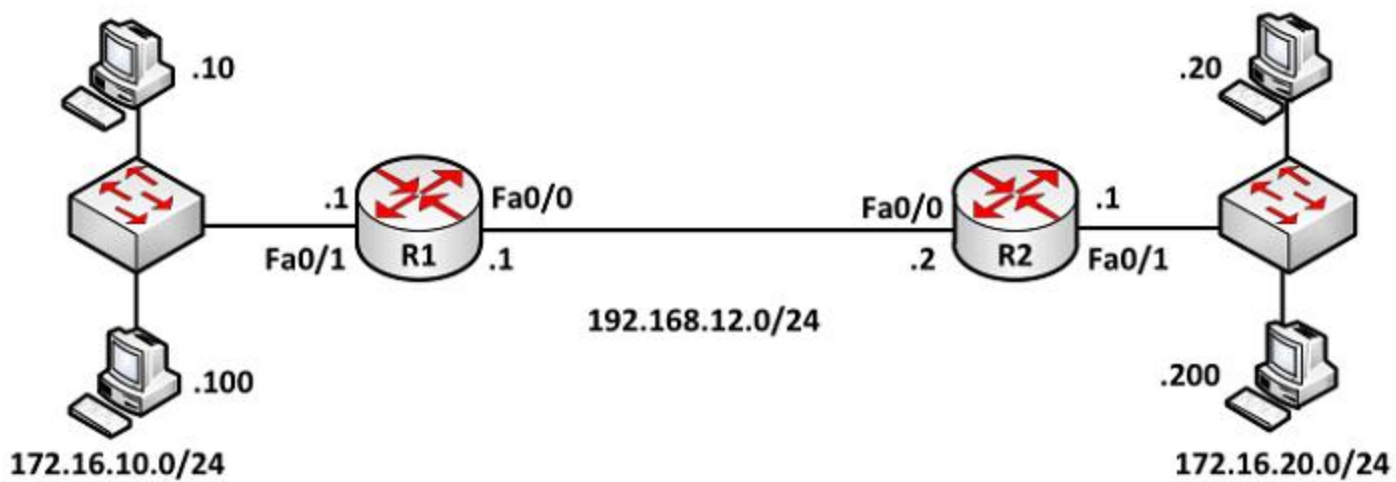
Để cấu hình định tuyến tĩnh, chúng ta sử dụng cú pháp sau:

```
Router(config)#ip route <destination-network>  
<subnet-mask> {next-hop-address|out-bound-interface}  
[distance]
```

Trong đó:

- *Destination-network*: là địa chỉ mạng đích cần đi tới
- *Subnet-mask*: Subnet mask của *destination-network*
- *Next-hop-address*: địa chỉ IP của cổng trên router kế tiếp có kết nối trực tiếp với router đang xét.
- *Out-bound-interface*: cổng của router sẽ gửi dữ liệu ra
- *Distance*: thay đổi giá trị AD cho tuyến này. Mặc định các tuyến tĩnh có AD=1.

Ví dụ: Cấu hình định tuyến tĩnh cho mô hình mạng sau:



Nhận xét: Trong mô hình mạng đã cho có 3 mạng: 172.16.10.0/24, 192.168.12.0/24 và 172.16.20.0/24. Để hệ thống mạng liên thông với nhau thì trong bảng định tuyến của các router R1 và R2 phải có đường đi đến tất cả các mạng này. Do mặc định các router biết được đường đi đến các mạng đang kết nối trực tiếp với nó nên:

- Router R1: đã biết được đường đi đến 2 mạng đang kết nối trực tiếp là 172.16.10.0/24 và 192.168.12.0/24. Đối với mạng 172.16.20.0/24, chúng ta cấu hình định tuyến tĩnh như sau:

```
R1(config)#ip route 172.16.20.0 255.255.255.0  
fa0/0
```

Hoặc

```
R1(config)#ip route 172.16.20.0 255.255.255.0  
192.168.12.2
```

- Router R2: tương tự router R1, mặc định R2 biết được đường đi đến 2 mạng đang kết nối trực tiếp với nó là 192.168.12.0/24 và 172.16.20.0/24. Chúng ta cần chỉ ra đường đi đến mạng 172.16.10.0/24 bằng định tuyến tĩnh như sau:

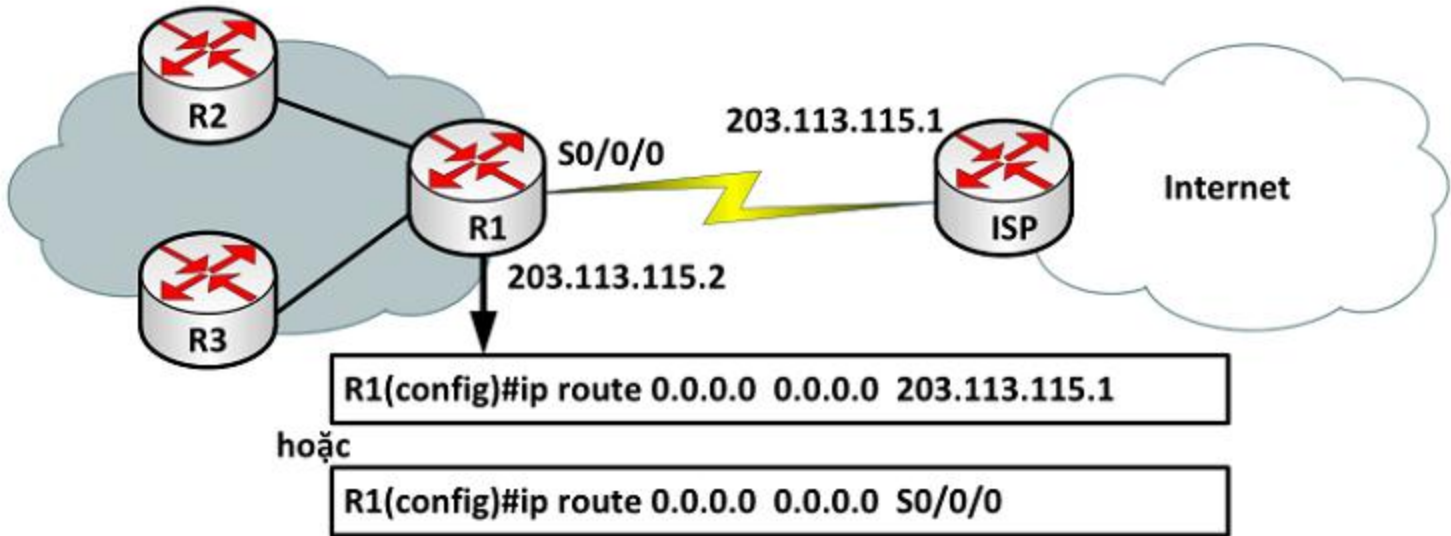
```
R2 (config)#ip route 172.16.10.0 255.255.255.0
fa0/0
```

Hoặc

```
R2 (config)#ip route 172.16.10.0 255.255.255.0
192.168.12.1
```

❖ Default route

Default route nằm ở cuối bảng định tuyến và được sử dụng để gửi các gói tin đi trong trường hợp mạng đích không tìm thấy trong bảng định tuyến. Nó rất hữu dụng trong các mạng dạng “*stub network*” như kết nối từ mạng nội bộ ra ngoài Internet.



Hình 1.5 Cấu hình *default route*

4. RIP

RIP là một giao thức định tuyến theo kiểu “*distance-vector*”. “*Hop count*” được sử dụng làm *metric* cho việc chọn đường. Nếu có nhiều đường đến cùng một đích thì RIP sẽ chọn đường nào có số *hop-count* (số router đi qua) ít nhất. RIP hỗ trợ tính năng chia tải (load balancing) mặc định là 4 đường, tối đa 16 đường.

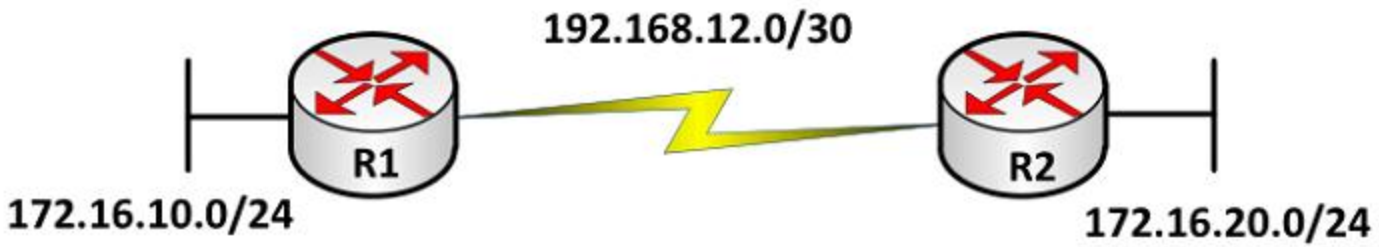
Nếu *hop-count* lớn hơn 15 thì các gói tin sẽ bị loại bỏ. Mặc định thời gian cập nhật định tuyến là 30 giây. Giá trị *AD* mặc định của RIP là 120. RIP có hai phiên bản là RIPv1 và RIPv2.

Bảng so sánh giữa RIPv1 và RIPv2

| Đặc điểm | RIPv1 | RIPv2 |
|--|----------|-----------|
| Loại định tuyến | Classful | Classless |
| Hỗ trợ VLSM và mạng không liên tục | Không | Có |
| Gửi kèm <i>Subnet-mask</i> trong bản tin cập nhật định tuyến | Không | Có |

| | | |
|-----------------------------------|-----------|-------------------------|
| Quảng bá thông tin định tuyến | Broadcast | Multicast |
| Hỗ trợ tóm tắt các tuyến thủ công | Không | Có |
| Hỗ trợ chứng thực | Không | Có |
| Định nghĩa trong RFC | RFC 1058 | RFC 1721, 1722, 2453 |

Mạng không liên tục (discontiguous network): là mạng mà trong đó các mạng con (subnet) của cùng một mạng lớn (major network: là mạng theo đúng lớp) bị ngăn cách bởi “major-network” khác.



Hình 1.6 Mạng không liên tục

Hai mạng con của cùng một “major network” là 172.16.0.0 bị ngăn cách bởi một “major network” khác là 192.168.12.0 tạo nên mạng không liên tục.

❖ Cấu hình

- Khởi tạo tiến trình định tuyến RIP

```
Router(config)#router rip
```

- Bật chế độ RIPv2

```
Router(config-router)#version 2 //sử dụng cho RIPv2
```

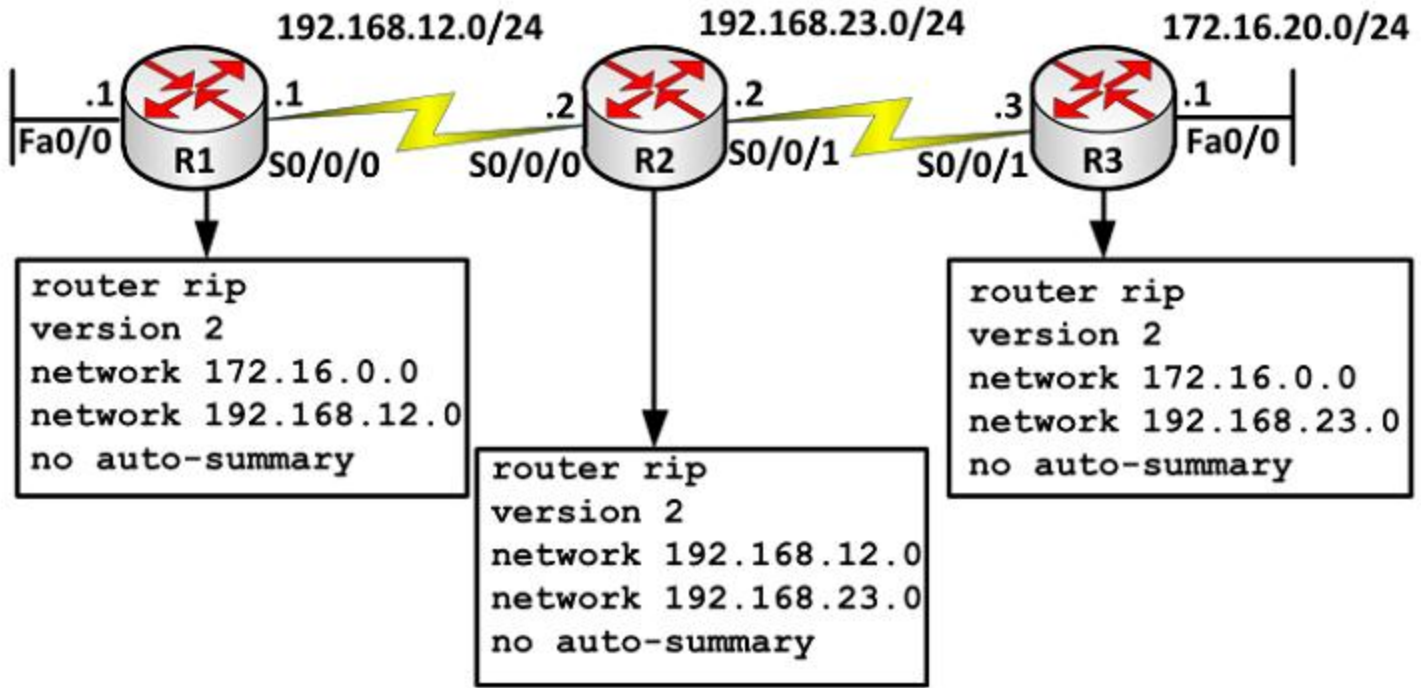
- Chọn cổng tham gia vào quá trình trao đổi thông tin định tuyến

```
Router(config-router)#network <major-classful-network>
```

- Tắt tính năng tự động tóm tắt các tuyến

```
Router(config-router)#no auto-summary //sử dụng cho RIPv2
```

❖ Ví dụ



❖ Chứng thực trong RIPv2

Chứng thực trong định tuyến là cách thức bảo mật trong việc trao đổi thông tin định tuyến giữa các router. Nếu có cấu hình chứng thực thì các router phải vượt qua quá trình này trước khi các thông tin trao đổi định tuyến được thực hiện. RIPv2 hỗ trợ hai kiểu chứng thực là: “Plain text” và “MD5”

- Chứng thực dạng “Plain Text”: còn gọi là “Clear text”

Quá trình chứng thực chỉ đơn giản là các router được cấu hình một khóa (password) và trao đổi chúng để so khớp. Các khóa này được gửi dưới dạng không mã hóa trên đường truyền.

Các bước cấu hình:

Bước 1. Tạo bộ khóa

```
Router(config)#key chain <name>
```

Bước 2. Tạo các khóa

```
Router(config-keychain)#key <key-id>
```

```
Router(config-keychain-key)#key-string <password>
```

Bước 3. Áp đặt vào cổng gửi chứng thực

```
Router(config)#interface <interface>
```

```
Router(config-if)#ip rip authentication key-chain <name>
```

Ví dụ: Cấu hình chứng thực trong định tuyến RIPv2 dạng “Plain Text”



```
R1 (config) #key chain newstar
R1 (config-keychain) #key 1
R1 (config-keychain-key) #key-string ccna
R1 (config) #interface S0/0/0
R1 (config-if) #ip rip authentication key-chain
newstar
R2 (config) #key chain newstar2
R2 (config-keychain) #key 1
R2 (config-keychain-key) #key-string ccna
R2 (config) #interface S0/0/0
R2 (config-if) #ip rip authentication key-chain
newstar2
```

• Chứng thực dạng MD5

Dạng chứng thực này sẽ gửi thông tin về khóa đã được mã hóa giúp các thông tin trao đổi được an toàn hơn. Các bước cấu hình tương tự như dạng “Plain Text”, chỉ có khác ở bước 3 phải thêm 1 lệnh sau:

```
Router (config-if) #ip rip authentication mode md5
```

Ví dụ: Sử dụng lại mô hình mạng trong ví dụ chứng thực dạng “Plain Text”, chúng ta sẽ cấu hình chứng thực định tuyến RIPv2 bằng MD5 với tên bộ khóa là “*spkt*” và mật khẩu là “123456” trên R1 và tên bộ khóa là “*cntt*” và mật khẩu là “123456” trên R2

```
R1 (config) #key chain spkt
R1 (config-keychain) #key 1
R1 (config-keychain-key) #key-string 123456
R1 (config) #interface S0/0/0
R1 (config-if) #ip rip authentication mode md5
R1 (config-if) #ip rip authentication key-chain
spkt
```



```
R2 (config)#key chain cntt
R2 (config-keychain)#key 1
R2 (config-keychain-key)#key-string 123456
R2 (config)#interface S0/0/0
R2 (config-if)#ip rip authentication mode md5
R2 (config-if)#ip rip authentication key-chain
cntt
```

❖ Các lệnh kiểm tra cấu hình

```
R#debug ip rip
R#show ip route
```

5. OSPF

OSPF (*Open Shortest Path First*) là một giao thức định tuyến dạng *link-state*, sử dụng thuật toán Dijkstra để xây dựng bảng định tuyến.

OSPF mang những đặc điểm của giao thức *link-state*. Nó có ưu điểm là hội tụ nhanh, hỗ trợ được mạng có kích thước lớn và không xảy ra “*routing loop*”. OSPF đồng thời là giao thức định tuyến dạng *classless* nên hỗ trợ VLSM và mạng không liên tục. OSPF sử dụng địa chỉ multicast 224.0.0.5 và 224.0.0.6 (DR và BDR router) để gửi các thông điệp *hello* và *update* trong quá trình cập nhật định tuyến.

Bên cạnh đó OSPF còn được thiết kế theo dạng phân cấp, sử dụng các *area* để giảm yêu cầu về CPU, bộ nhớ của router. OSPF hỗ trợ chứng thực dạng *Plain-Text* và dạng *MD5*.

❖ Metric của OSPF

OSPF sử dụng *metric* là *cost*. *Cost* của toàn tuyến được tính theo cách cộng dồn *cost* dọc theo tuyến đường đi của packet. Cách tính *cost* được IETF đưa ra trong RFC 2328.

Cost được tính dựa trên băng thông sao cho tốc độ kết nối của đường kết nối càng cao thì *cost* càng thấp dựa trên công thức $10^8/\text{bandwidth}$ với giá trị *bandwidth* được cấu hình trên mỗi cổng của router và đơn vị tính là *bps*.

Tuy nhiên, chúng ta có thể thay đổi giá trị *cost*. Nếu router có nhiều đường đến đích mà chi phí bằng nhau thì router sẽ cân bằng tải trên các đường đó, mặc định trên 4 đường, tối đa là 16 đường. Những tham số bắt buộc phải giống nhau trong các router chạy OSPF trong một hệ thống

mạng, đó là *Hello/dead interval*, *Area – ID*, *authentication password* (nếu có), *stub area flag*.

❖ Các loại môi trường OSPF

- Multiple access (ethernet)
- Point-to-Point
- NBMA (Non-Broadcast Multiple Access)

❖ Quá trình xây dựng bảng định tuyến của OSPF

- Các OSPF gửi các gói *hello* định kỳ để thiết lập quan hệ láng giềng. Gói tin *hello* mang các thông tin thương lượng với các router láng giềng trước khi thiết lập quan hệ *adjacency*. Trong mạng đa truy cập, giao thức *hello* sẽ bầu ra DR và BDR. DR và BDR sẽ thiết lập mối quan hệ *adjacency* với tất cả các router khác và những router này chỉ trao đổi thông tin với DR và BDR. Trong mạng Point-to-Point không cần chọn DR và BDR.
- Mỗi router nhận một LSA từ láng giềng với cơ sở dữ liệu về trạng thái các đường liên kết (link-state database) của láng giềng đó và gửi một bản sao của LSA tới tất cả các láng giềng khác của nó.
- Bằng cách gửi các LSA cho toàn bộ một *area*, tất cả router sẽ xây dựng chính xác cơ sở dữ liệu về trạng thái liên kết. Khi cơ sở dữ liệu được hoàn tất, mỗi router sử dụng thuật toán SPF để xây dựng nên cây SPF.
- Mỗi router sẽ xây dựng nên bảng định tuyến từ cây SPF. Kết quả là mỗi router sẽ có thông tin về đường đến tất cả các mạng đích trong hệ thống mạng.

❖ Quá trình bầu chọn DR và BDR

Quá trình bầu chọn liên quan đến 2 tham số: độ ưu tiên (*priority*) và *router-ID*. Tham số *priority* được chọn trước tiên, giá trị *priority* nằm trong khoảng từ 0 đến 255. Nếu *priority* đặt là 0 thì router này sẽ không tham gia vào quá trình bầu chọn DR/BDR. Router nào có độ ưu tiên cao nhất sẽ được chọn là DR, cao thứ hai sẽ là BDR. Mặc định giá trị *priority* OSPF là 1. Khi giá trị *priority* đều bằng nhau thì OSPF sẽ bầu chọn DR dựa vào tham số thứ hai là *router-ID*.

Trong hệ thống mạng dùng OSPF không cấu hình cổng *loopback* thì giá trị *router-ID* được chọn là giá trị địa chỉ IP lớn nhất của các cổng đang hoạt động trên router. Nếu có cổng *loopback* thì cổng *loopback*

được chọn, trường hợp có nhiều cổng *loopback* thì chọn cổng *loopback* nào có địa chỉ IP cao nhất.

❖ Cấu hình OSPF

- Khởi tạo tiến trình định tuyến OSPF

```
Router(config)#router ospf <process-id>
```

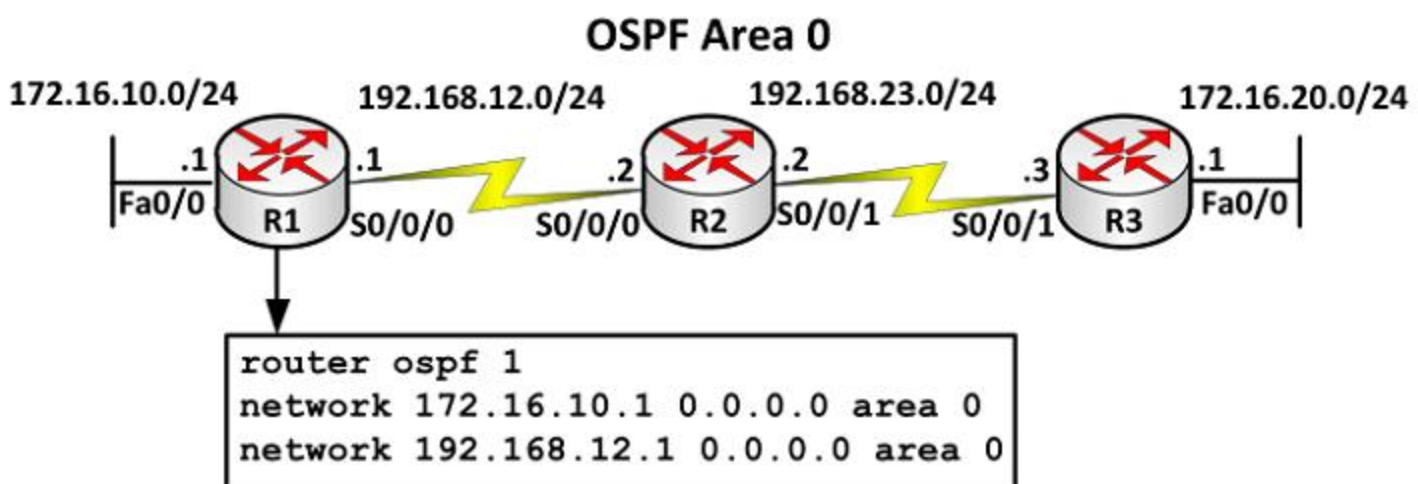
- Chọn cổng tham gia vào quá trình trao đổi thông tin định tuyến

```
Router(config-router)#network <address>  
<wildcard-mask> area <area-id>
```

Trong đó:

- *Process-id*: chỉ số tiến trình của OSPF, mang tính chất cục bộ, có giá trị 1 đến 65535.
- *Address*: địa chỉ cổng tham gia định tuyến
- *Wildcard mask*: điều kiện kiểm tra giữa địa chỉ cấu hình trong *address* và địa chỉ các cổng trên router, tương ứng bit 0 – phải so khớp, bit 1 – không cần kiểm tra.
- *Area-id*: vùng mà cổng tương ứng thuộc về trong kiến trúc OSPF.

Ví dụ:



❖ Các câu lệnh kiểm tra cấu hình OSPF

```
Router#show ip protocol  
Router#show ip route  
Router#show ip ospf interface  
Router#show ip ospf neighbor  
Router#debug ip ospf events
```

```
Router#debug ip ospf packet
```

❖ Chứng thực trong OSPF

Giao thức OSPF hỗ trợ hai dạng chứng thực là: “Plain Text” và MD5

- Chứng thực bằng “Plain Text”

Cấu hình giữa hai cổng của 2 router nối trực tiếp với nhau để chứng thực giữa chúng trước khi trao đổi thông tin định tuyến. Mật khẩu gửi chứng thực không được mã hóa.

```
R(config)#interface <interface>
R(config-if)#ip ospf authentication
R(config-if)#ip ospf authentication-key <password>
```

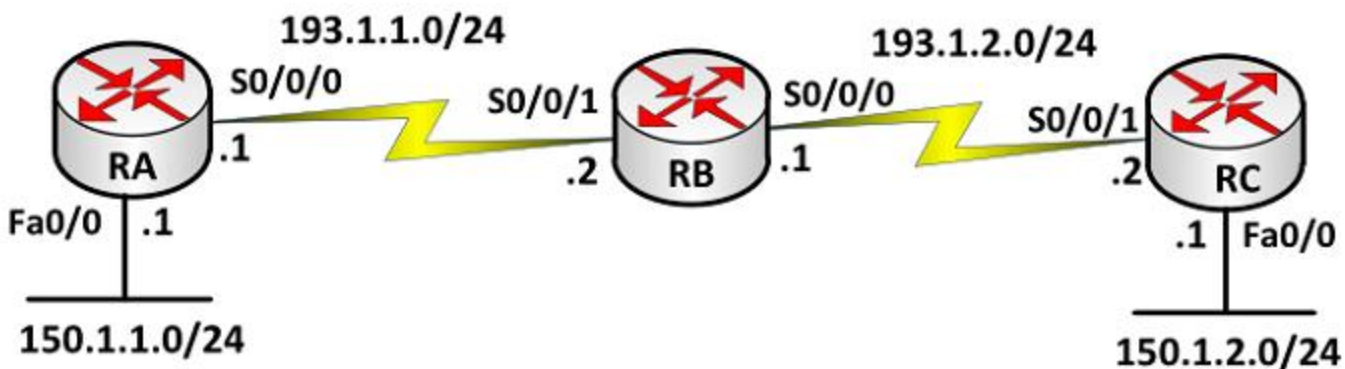
- Chứng thực bằng MD5

Trên cổng của router gửi thông tin chứng thực cấu hình lệnh sau:

```
R(config)#interface <interface>
R(config-if)#ip ospf authentication message-digest
R(config-if)#ip ospf message-digest-key 1 md5
<password>
```

Ví dụ 1: Cho mô hình mạng sau.

Yêu cầu: Cấu hình OSPF cho các router RA, RB và RC (Area 0) trong mô hình mạng sau để quảng bá các thông tin định tuyến. Cấu hình chứng thực dạng “Plain Text” và MD5 giữa 2 router: RA và RB với mật khẩu là “cisco”.



Hướng dẫn cấu hình:

Bước 1: Cấu hình cơ bản (đặt hostname, địa chỉ IP cho các cổng: Serial, FastEthernet)

Bước 2: Cấu hình giao thức định tuyến OSPF trên mỗi router

```
RA(config)#router ospf 1
RA(config-router)#network 150.1.1.0 0.0.0.255 area 0
RA(config-router)#network 193.1.1.0 0.0.0.255 area 0
```

```

RB(config)#router ospf 1
RB(config-router)#network 193.1.1.0 0.0.0.255 area 0
RB(config-router)#network 193.1.2.0 0.0.0.255 area 0

RC(config)#router ospf 1
RC(config-router)#network 150.1.2.0 0.0.0.255 area 0
RC(config-router)#network 193.1.2.0 0.0.0.255 area 0

```

Bước 3.1. Cấu hình chứng thực dạng “Plain Text” giữa 2 router: RA và RB

```

RA(config)#int S0/0/0
RA(config-if)#ip ospf authentication
RA(config-if)#ip ospf authentication-key cisco

RB(config)#int S0/0/1
RB(config-if)#ip ospf authentication
RB(config-if)#ip ospf authentication-key cisco

```

Bước 3.2 Cấu hình chứng thực dạng MD5 giữa 2 router: RA và RB

```

RA(config)#int S0/0/0
RA(config-if)#ip ospf authentication message-digest
RA(config-if)#ip ospf messages-digest-key 1 md5 cisco

RB(config)#int S0/0/1
RB(config-if)#ip ospf authentication message-digest
RB(config-if)#ip ospf messages-digest-key 1 md5 cisco

```

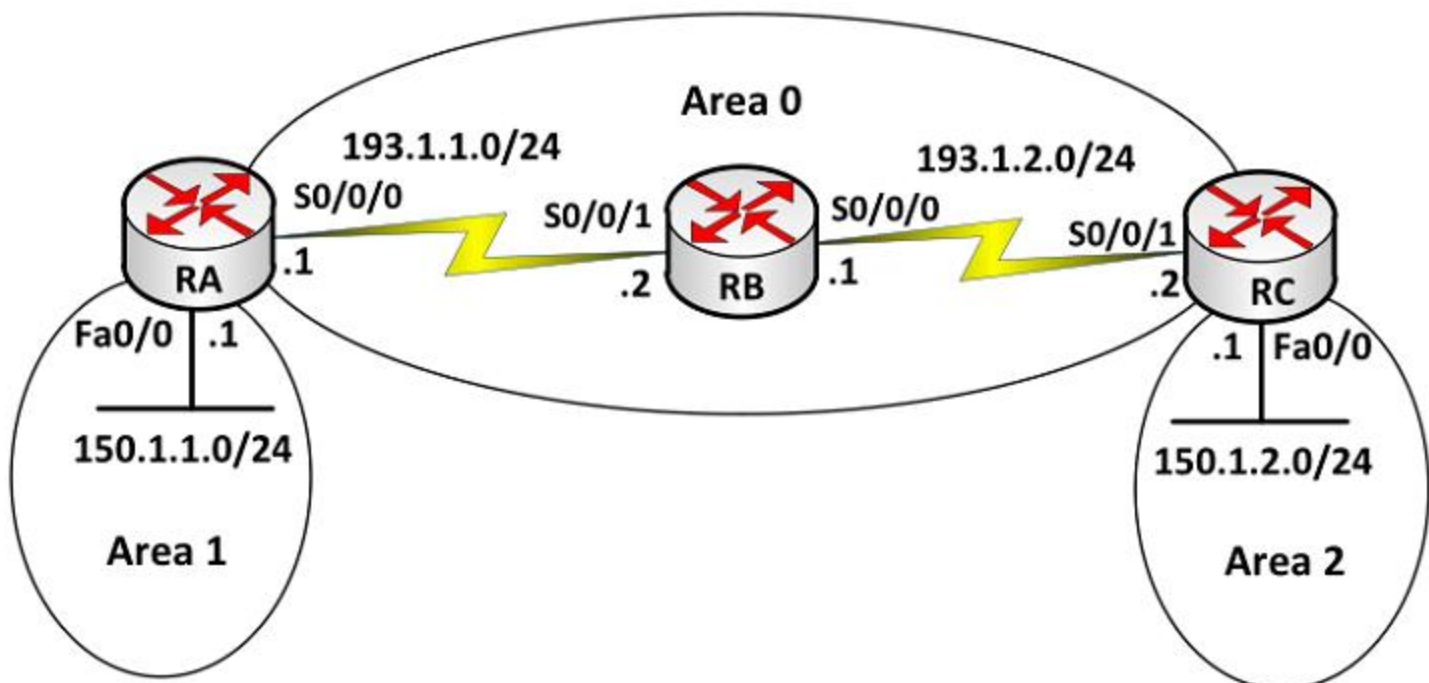
Bước 4. Kiểm tra cấu hình

Thực hiện các câu lệnh sau để kiểm tra cấu hình

show ip route: xem bảng định tuyến

debug ip ospf event: xem quá trình cập nhật định tuyến của OSPF

Ví dụ 2: Định tuyến động – OSPF



❖ Mô tả

- RA, RB, RC sử dụng OSPF để quảng bá thông tin định tuyến
- Các router cấu hình OSPF và quảng bá tất cả các mạng nối trực tiếp. Từ Router RA, RB và RC ta ping được hết các địa chỉ trong mạng.

❖ Các bước thực hiện

- Đặt hostname, địa chỉ IP cho các cổng trên router.
- **Cấu hình giao thức định tuyến RIP trên mỗi router**

```
RA(config)#router ospf 1
```

```
RA(config-router)#network 150.1.1.0 0.0.0.255  
area 1
```

```
RA(config-router)#network 193.1.1.0 0.0.0.255  
area 0
```

```
RB(config)#router ospf 1
```

```
RB(config-router)#network 193.1.1.0 0.0.0.255  
area 0
```

```
RB(config-router)#network 193.1.2.0 0.0.0.255  
area 0
```

```
RC(config)#router ospf 1
```

```
RC(config-router)#network 150.1.2.0 0.0.0.255  
area 2
```

```
RC(config-router)#network 193.1.2.0 0.0.0.255  
area 0
```

✓ Kiểm tra cấu hình

Thực hiện các câu lệnh sau để kiểm tra cấu hình

```
Router#show ip route: xem bảng định tuyến
```

```
Router#ping: kiểm tra kết nối
```

6. EIGRP

EIGRP là giao thức định tuyến do Cisco tạo ra, chỉ hoạt động trên các thiết bị của Cisco. EIGRP là một giao thức định tuyến lai, nó vừa mang những đặc điểm của “*distance vector*” vừa mang một số đặc điểm của “*link-state*”. EIGRP là dạng định tuyến “*classless*”.

EIGRP hỗ trợ VLSM và CIDR nên sử dụng hiệu quả không gian địa chỉ, sử dụng địa chỉ multicast (224.0.0.10) để trao đổi thông tin cập nhật định tuyến.

❖ Cách tính metric của EIGRP

$$metric_{EIGRP} = \left[K1 * BW + \frac{K2 * BW}{(256 - load)} + K3 * Delay \right] * \frac{K5}{(reliability + K4)}$$

Với K1, K2, K3, K4, K5 là hằng số

Mặc định: K1=1, K2=0, K3=1, K4=0, K5=0

Do đó, ta có:

$$metric = bandwidth + delay$$

Những xử lý cơ bản của EIGRP trong việc học các mạng đích:

- Các router phát hiện các láng giềng của nó, danh sách các láng giềng được lưu giữ trong “*neighbor table*”.
- Mỗi router sẽ trao đổi các thông tin về cấu trúc mạng với các láng giềng của nó.
- Router đặt những thông tin về cấu trúc hệ thống mạng học được vào cơ sở dữ liệu về cấu trúc mạng (topology table).
- Router chạy thuật toán DUAL với cơ sở dữ liệu đã thu thập được ở bước trên để tính toán tìm ra đường đi tốt nhất đến mỗi một mạng trong cơ sở dữ liệu.
- Router đặt các đường đi tốt nhất đến mỗi mạng đích vào bảng định tuyến.
- Trong EIGRP có hai tuyến ta cần quan tâm là “*successor route*” và “*fessible successor route*”.
 - ✓ **Successor route**: là tuyến đường đi chính được sử dụng để chuyển dữ liệu đến đích, được lưu trong bảng định tuyến. EIGRP cho phép chia tải tối đa trên 16 đường (mặc định là 4 đường) đến mỗi mạng đích.
 - ✓ **Fessible successor route**: là đường đi dự phòng cho đường đi chính và được lưu trong bảng cấu trúc mạng (*topology table*).

❖ EIGRP chống “routing loop”

“Routing loop” là một trở ngại rất lớn trong các giao thức định tuyến dạng “*distance vector*”. Các giao thức định tuyến dạng “*link-state*”

vượt qua vấn đề này bằng cách mỗi router đều nắm giữ toàn bộ cấu trúc mạng. Trong giao thức EIGRP, khi tuyến đường đi chính gặp sự cố, router có thể kịp thời đặt đường đi dự phòng vào bảng định tuyến đóng vai trò như đường đi chính.

Trường hợp không có đường đi dự phòng, EIGRP sử dụng thuật toán DUAL cho phép router gửi các yêu cầu và tính toán lại các đường đi đến đích.

❖ Cấu hình EIGRP

- Bước 1. Kích hoạt giao thức định tuyến EIGRP

```
Router(config)#router eigrp <autonomous-system>
```

Trong đó: *autonomous-system*: có giá trị từ 1 đến 65535, giá trị này phải giống nhau ở tất cả các router trong hệ thống chạy EIGRP

- Bước 2. Chọn cổng tham gia vào quá trình trao đổi thông tin định tuyến

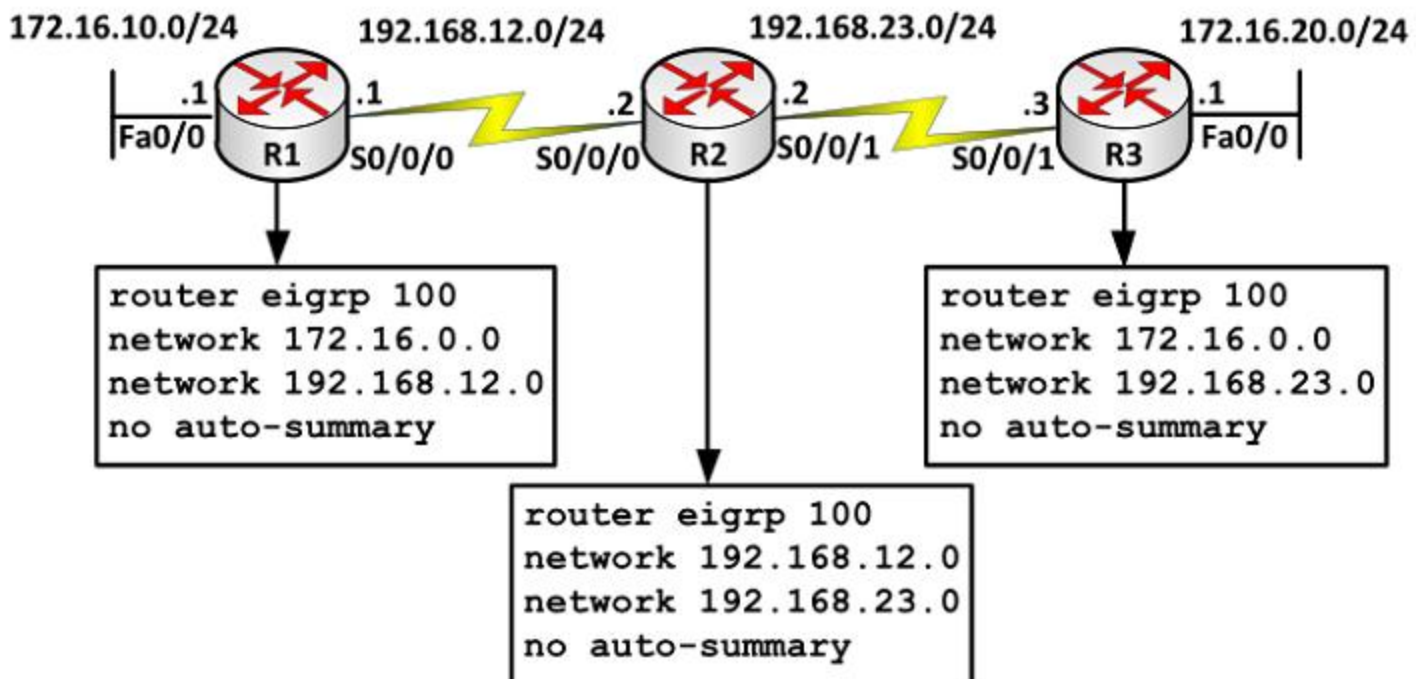
```
Router(config-router)#network <network-number>
```

Trong đó: *network-number* là địa chỉ cổng theo đúng lớp mạng của nó.

Để quảng bá các mạng con và hỗ trợ mạng không liên tục, chúng ta phải sử dụng lệnh sau:

```
Router(config-router)#no auto-summary
```

Ví dụ: Cấu hình định tuyến EIGRP cho mô hình mạng sau



❖ Các câu lệnh kiểm tra cấu hình EIGRP

```
Router#show ip eigrp neighbors
```

```
Router#show ip eigrp topology
```



```
Router#show ip route eigrp
Router#show ip protocols
Router#show ip eigrp traffic
```

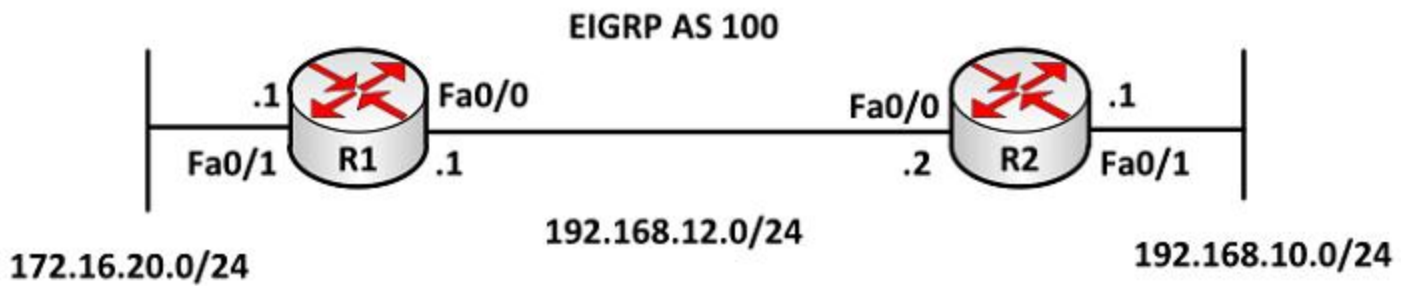
❖ Chứng thực trong EIGRP

EIGRP chỉ hỗ trợ một dạng chứng thực là MD5.

Trên công của router gửi thông tin chứng thực cấu hình lệnh sau:

```
R(config)#key chain <keychain>
R(config-keychain)#key <key-id>
R(config-keychain-key)#key-string <password>
R(config)#interface <interface>
R(config-if)#ip authentication mode eigrp <AS> md5
R(config-if)#ip authentication key-chain eigrp <AS>
<keychain>
```

Ví dụ: Cấu hình chứng thực cho giao thức định tuyến EIGRP giữa hai router R1 và R2.



• Hướng dẫn cấu hình

- Cấu hình cơ bản: hostname, địa chỉ IP cho các công trên các router.

- Cấu hình định tuyến EIGRP AS 100

```
R1(config)#router eigrp 100
R1(config-if)#network 192.168.12.0
R1(config-if)#network 172.16.0.0
R1(config-if)#no auto-summary
```

```
R2(config)#router eigrp 100
R2(config-if)#network 192.168.12.0
R2(config-if)#network 192.168.10.0
R2(config-if)#no auto-summary
```

- Cấu hình chứng thực

```
R1(config)#key chain my_keychain1
R1(config-keychain)#key 1
R1(config-keychain-key)#key-string cisco
```

```

R1(config)#interface fa0/0
R1(config-if)#ip authentication mode eigrp 100 md5
R1(config-if)#ip authentication key-chain eigrp 100
my_keychain1

R2(config)#key chain my_keychain2
R2(config-keychain)#key 1
R2(config-keychain-key)#key-string cisco

R2(config)#interface fa0/0
R2(config-if)#ip authentication mode eigrp 100 md5
R2(config-if)#ip authentication key-chain eigrp 100
my_keychain2

```

- **Kiểm tra cấu hình:** Dùng các lệnh sau

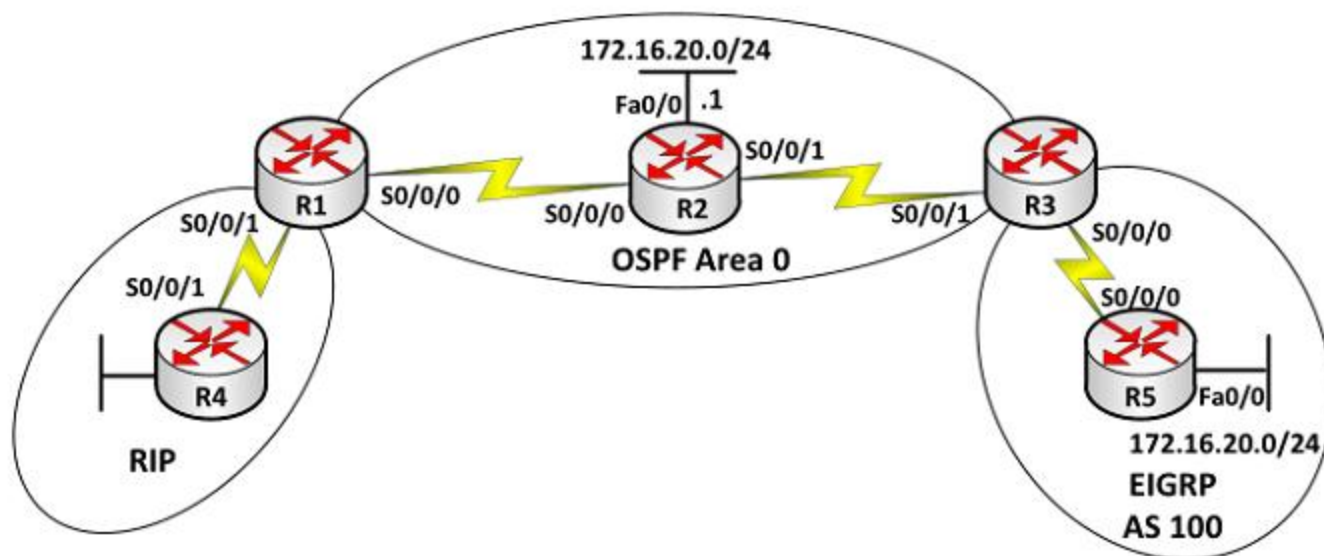
```

show ip eigrp neighbors
show ip eigrp interfaces details
show key chain

```

7. PHÂN PHỐI GIỮA CÁC GIAO THỨC ĐỊNH TUYẾN

Nếu một hệ thống mạng chạy nhiều hơn một giao thức định tuyến, người quản trị cần một vài phương thức để phân phối các đường đi của một giao thức này vào một giao thức khác. Quá trình đó gọi là phân phối giữa các giao thức định tuyến (*redistribution*).



Hình 1.7 Phân phối giữa các giao thức định tuyến

Phân phối định tuyến định nghĩa cách thức trao đổi thông tin định tuyến giữa các giao thức định tuyến. Mỗi giao thức định tuyến có cách tính toán “*metric*” khác nhau, do đó khi thực hiện quá trình phân phối thì dạng “*metric*” sẽ được chuyển đổi sao cho phù hợp với giao thức định tuyến đó để các giao thức đó có thể quảng bá các đường đi cho nhau.

- Phân phối định tuyến giữa RIP và OSPF
- Quảng bá các tuyến học được từ OSPF vào RIP

```
Router(config)#router rip
```

```
Router(config-router)#redistribute ospf 1 metric
<number>
```

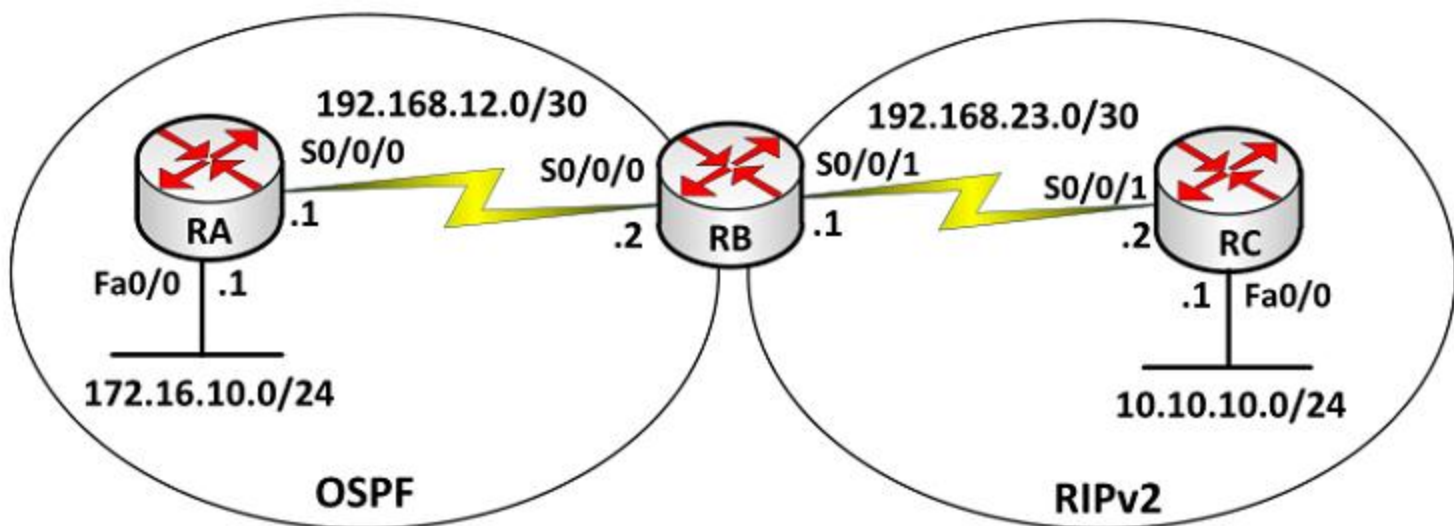
Lưu ý: Do RIP sử dụng *metric* có giá trị tối đa là 15 nên giá trị *<number>* trong lệnh trên cũng phải nhỏ hơn 15.

- Quảng bá các tuyến học được từ RIP vào OSPF

```
Router(config)#router ospf <process-id>
```

```
Router(config-router)#redistribute rip metric
<metric> subnets
```

Ví dụ: Cho mô hình mạng sau



Mô tả yêu cầu:

- ✓ RA, RB sử dụng OSPF để quảng cáo thông tin định tuyến
- ✓ RB, RC sử dụng RIP để quảng cáo thông tin định tuyến
- ✓ Từ RA, RB, RC *ping* được hết các địa chỉ trong mạng

Các bước thực hiện:

- ✓ Đặt hostname, địa chỉ IP cho các cổng trên router.
- ✓ Cấu hình giao thức định tuyến OSPF trên mỗi RA và RB

```
RA(config)#router ospf 1
```

```
RA(config-router)#network 172.16.10.0 0.0.0.255
area 0
```

```
RA(config-router)#network 192.168.12.0 0.0.0.3
area 0
```

```
RB(config)#router ospf 1
RB(config-router)#network 192.168.12.0 0.0.0.3
area 0
RB(config)#router rip
RB(config-router)#version 2
RB(config-router)#network 192.168.23.0
RB(config-router)#no auto-summary
RC(config)#router rip
RC(config-router)#version 2
RC(config-router)#network 192.168.23.0
RC(config-router)#network 10.0.0.0
RC(config-router)#no auto-summary
```

✓ Cấu hình phân phối định tuyến

Để RC thấy được RA, ta thực hiện các lệnh sau:

```
RB(config)#router rip
RB(config-router)#redistribute ospf 1 metric 3
```

Tương tự: để RA thấy RC

```
RB(config)#router ospf 1
RB(config-router)#redistribute rip metric 100
subnets
```

✓ Kiểm tra cấu hình

Thực hiện các câu lệnh sau để kiểm tra cấu hình

Router#show ip route: xem bảng định tuyến

Router#ping: kiểm tra kết nối

- Phân phối định tuyến giữa RIP và EIGRP

- Quảng bá các tuyến học được từ EIGRP vào RIP

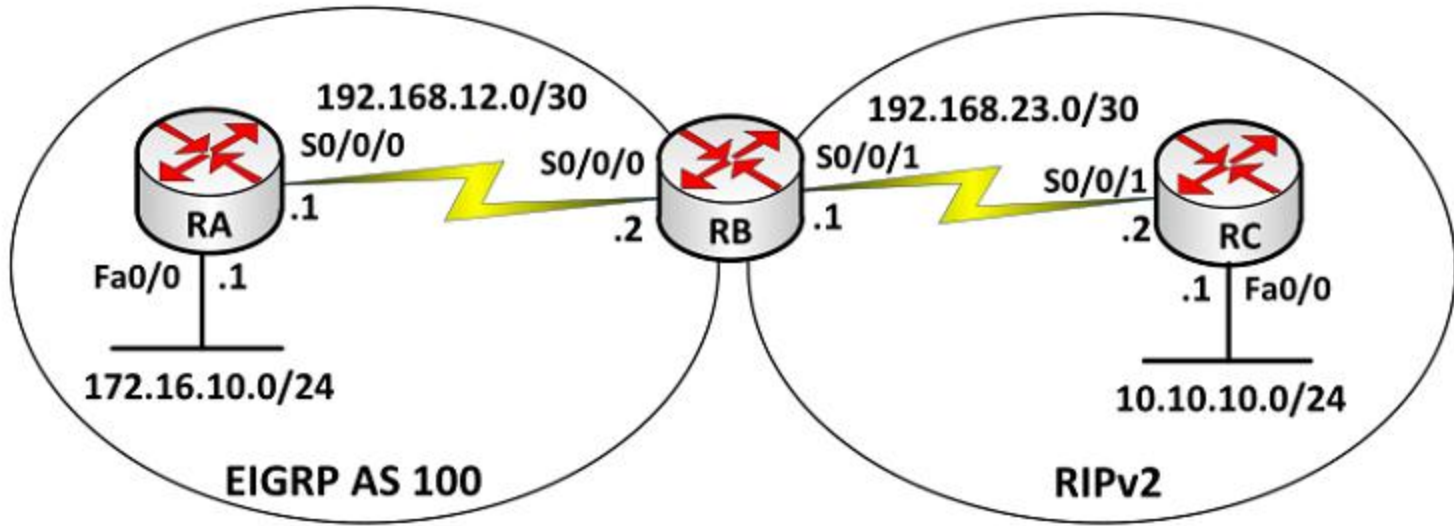
```
RB(config)#router rip
RB(config-router)#redistribute eigrp <AS> metric
<number>
```

- Quảng bá các tuyến học được từ RIP vào EIGRP

```
RB(config)#router eigrp <AS>
RB(config-router)#redistribute rip metric BW DL
L R MTU
```

Trong đó: BW, DL, L, R, MTU tương ứng với các thông số trong metric của EIGRP (trừ MTU). Tương tự, chúng ta có thể suy luận ra phương pháp để phân phối các tuyến học được từ một giao thức này sang một giao thức khác là phải tuân theo thông số về *metric* của giao thức mà ta sẽ phân phối vào.

Ví dụ:



Mô tả

- ✓ RA, RB sử dụng EIGRP để quảng cáo thông tin định tuyến
- ✓ RB, RC sử dụng RIP để quảng cáo thông tin định tuyến
- ✓ Từ RA, RB, RC *ping* được hết các địa chỉ trong mạng

Các bước thực hiện

- ✓ Đặt Hostname, địa chỉ IP cho các cổng trên router.
- ✓ Cấu hình giao thức định tuyến EIGRP trên mỗi RA và RB

```

RA(config)#router eigrp 100
RA(config-router)#network 172.16.0.0
RA(config-router)#network 192.168.12.0
RA(config-router)#no auto-summary
RB(config)#router eigrp 100
RB(config-router)#network 192.168.12.0
RB(config-router)#no auto-summary
RB(config)#router rip
RB(config-router)#version 2
RB(config-router)#network 192.168.23.0
RB(config-router)#passive interface S0/0/0
    
```

```
RC(config)#router rip
RC(config-router)#version 2
RC(config-router)#network 10.0.0.0
RC(config-router)#network 192.168.23.0
```

Để RC thấy được RA, ta thực hiện các lệnh phân phối định tuyến:

```
RB(config)#router rip
RB(config-router)#redistribute eigrp 100 metric 3
```

Tương tự: để RA thấy RC

```
RB(config)#router eigrp 100
RB(config-router)#redistribute rip metric 100 1
255 255 1500
```

✓ Kiểm tra

Thực hiện các câu lệnh sau để kiểm tra cấu hình

```
show ip route: xem bảng định tuyến
ping: kiểm tra kết nối
```

8. TỔNG KẾT CHƯƠNG

Trong chương này đề cập đến một số vấn đề cơ bản và định tuyến. Định tuyến là quá trình tìm đường đi cho các gói tin từ nơi gửi đến nơi nhận. Quá trình định tuyến được phân làm 2 loại là định tuyến tĩnh và định tuyến động.

Định tuyến tĩnh là quá trình định tuyến sử dụng các tuyến được cấu hình thủ công bởi người quản trị mạng, còn định tuyến động là quá trình định tuyến được thực hiện bằng các giao thức định tuyến động.

Trong định tuyến động người ta phân ra làm 2 dạng: *distance-vector* và *link-state*. Ngoài ra, định tuyến còn được chia thành 2 kiểu là *classful* và *classless*.

RIPv1 là giao thức định tuyến động theo dạng *distance-vector* và là giao thức định tuyến theo kiểu *classful*, có nghĩa là không mang theo thông tin *subnet-mask* trong các thông tin cập nhật định tuyến và mỗi router nhìn hệ thống mạng theo sự chi phối của các router láng giềng. RIPv2 là một giao thức định tuyến cải tiến từ RIPv1, mang các đặc điểm của loại giao thức *distance-vector*. Tuy nhiên, RIPv2 thuộc nhóm giao thức *classless*. Do đó, nó còn mang một số tính chất như hỗ trợ VLSM và mạng không liên tục.

OSPF là một giao thức định tuyến loại *link-state*, thuộc nhóm giao thức *classless*. OSPF được sử dụng trên các mạng lớn, phức tạp. EIGRP là một giao thức lai giữa *distance-vector* và *link-state*. EIGRP thuộc nhóm giao thức *classless*.

9. CÂU HỎI VÀ BÀI TẬP

9.1 Các câu nào sau đây nói về đặc điểm của giao thức định tuyến loại "distance vector" và "link-state"?

- A. Các giao thức định tuyến loại "distance vector" gửi toàn bộ bảng định tuyến cho các router láng giềng có kết nối trực tiếp với nó.
- B. Các giao thức định tuyến loại "distance vector" gửi bản cập nhật thay đổi đến các mạng được liệt kê trong bảng định tuyến
- C. Các giao thức định tuyến loại "link-state" gửi toàn bộ bảng định tuyến cho các router khác trong mạng
- D. Các giao thức định tuyến loại "link-state" gửi các cập nhật về trạng thái kết nối cho các router khác

9.2 Giao thức định tuyến nào mặc định sử dụng hai tham số "bandwidth" và "delay" làm metric?

- A. RIP
- B. OSPF
- C. EIGRP
- D. Định tuyến tĩnh

9.3 Các câu nào sau đây là đúng cho các giao thức định tuyến thuộc loại "classless"?

- A. Không chạy được trong mạng không liên tục (discontiguous network)
- B. Hỗ trợ mạng VLSM
- C. RIPv1 là giao thức thuộc "classless"
- D. RIPv2 là giao thức thuộc loại "classless"

9.4 "Default route" dùng để làm gì?

- A. Nó được sử dụng khi các giao thức định tuyến không sử dụng được.

- B. Nó được nhà cung cấp dịch vụ (ISP) cấu hình để gửi dữ liệu cho các đối tác qua mạng.
 - C. Nó được sử dụng khi gói tin gửi tới mạng đích không có trong bảng định tuyến.
 - D. Nó được cấu hình thủ công đến các mạng khác mà các giao thức định tuyến chưa cấu hình.
- Nó được dùng để gửi các gói tin đến các "stub network".

9.5 Các câu nào sau đây mô tả đúng về giao thức định tuyến RIP?

- A. RIPv1 không hỗ trợ chứng thực trong cập nhật thông tin định tuyến. RIPv2 hỗ trợ chứng thực trong cập nhật thông tin định tuyến.
- B. RIPv1 không hỗ trợ quảng bá các đường đi qua mạng WAN. RIPv2 hỗ trợ quảng bá các đường đi qua LAN và WAN.
- C. RIPv1 không gửi kèm thông tin "subnet-mask" trong bản tin cập nhật định tuyến. RIPv2 gửi kèm thông tin "subnet-mask" trong bản tin cập nhật định tuyến.
- D. RIPv1 định thời gửi thông tin cập nhật định tuyến qua địa chỉ IP multicast: 224.0.0.10. RIPv2 định thời gửi thông tin cập nhật định tuyến qua địa chỉ IP multicast: 224.0.0.9.
- E. RIPv1 sử dụng "hold-down timer" và "split horizon" để chống tình trạng "routing loop". RIPv2 không yêu cầu "hold-down timer" hay "split horizon" để chống "routing loop".

9.6 Một router học được đường đi đến một mạng đích 131.107.4.0/24 bằng RIP và OSPF. Bạn cũng cấu hình thêm bảng định tuyến tĩnh trên router này đến mạng 131.107.4.0/24. Router sẽ chọn đường đi như thế nào để chuyển dữ liệu đi?

- A. Đường đi học được từ RIP
- B. Đường đi học được từ OSPF
- C. Đường đi học được từ định tuyến tĩnh
- D. Chia tải đi trên cả 3 đường này

9.7 Giá trị AD (Administrative Distance) mặc định của OSPF là

- A. 1
- B. 90

- C. 110
- D. 120

9.8 Các đặc điểm nào dưới đây là của các giao thức định tuyến thuộc loại *link-state*?

- A. Cung cấp thông tin toàn diện về hệ thống mạng
- B. Trao đổi bảng định tuyến với các láng giềng.
- C. Tính toán đường đi ngắn nhất
- D. Có tính năng “trigger update”
- E. Có tính năng cập nhật định thời

9.9 Router R1 đang chạy giao thức định tuyến RIP, thông tin bảng định tuyến được hiển thị như sau:

```
Gateway of last resort is 10.1.2.2 to network
0.0.0.0
      10.0.0.0/24 is subnetted, 2 subnets
R      10.1.3.0 [120/1] via 10.1.2.2, 00:00:00,
Serial0/0
C      10.1.2.0 is directly connected, Serial0/0
C      10.1.5.0 is directly connected, Serial0/1
C      10.1.6.0 is directly connected,
FastEthernet0/0
R*    0.0.0.0/0 [120/1] via 10.1.5.5, 00:00:00,
Serial0/1
```

Dựa vào thông tin trên, nếu ta thực hiện lệnh *ping* đến địa chỉ 10.1.8.5 từ máy tính có địa chỉ 10.1.6.100, thì router R1 xử lý các gói tin ICMP này như thế nào?

- A. Các gói tin sẽ bị loại bỏ
- B. Các gói tin sẽ được chuyển ra cổng S0/0
- C. Các gói tin sẽ được chuyển ra cổng S0/1
- D. Các gói tin sẽ được chuyển ra cổng Fa0/0
- E. Các gói tin sẽ được chuyển đến gateway 10.1.2.2

9.10 Giao thức định tuyến OSPF dùng khái niệm khu vực (area). Đặc điểm của OSPF area là gì?

- A. Mỗi OSPF area cần cấu hình cổng loopback
- B. Các area có thể được gán giá trị từ 0 đến 65535
- C. Area 0 còn gọi là area backbone
- D. Kiến trúc phân cấp OSPF không yêu cầu nhiều area
- E. Các area phải kết nối về area 0
- F. OSPF đơn vùng phải cấu hình là area 1

Chương 2

VLAN

Chương này đề cập đến một số kỹ thuật được triển khai trên Switch như VLAN, VTP, STP. Học xong chương này, người học có khả năng:

- Phân biệt giữa miền đụng độ và miền quảng bá
- Trình bày được khái niệm và đặc điểm của VLAN, VTP, STP
- Cấu hình VLAN, VTP, STP trên switch
- Cấu hình định tuyến giữa các VLAN

1. GIỚI THIỆU

- **Collision domain:** miền đụng độ

Đụng độ xảy ra khi có hai hay nhiều máy truyền dữ liệu đồng thời trong một mạng chia sẻ. Khi đụng độ xảy ra, các gói tin đang được truyền đều bị phá hủy, các máy đang truyền sẽ ngưng việc truyền dữ liệu và chờ một khoảng thời gian ngẫu nhiên theo quy luật của CSMA/CD. Nếu đụng độ xảy ra quá nhiều mạng có thể không hoạt động được.

Miền đụng độ là khu vực mà dữ liệu được phát ra có thể bị đụng độ. Tất cả các môi trường mạng chia sẻ là các miền đụng độ.

- **Broadcast domain:** miền quảng bá

Các thông tin liên lạc trong mạng được thực hiện theo ba cách: unicast, multicast và broadcast.

- Unicast: gửi trực tiếp từ một máy đến một máy.
- Multicast: được thực hiện khi một máy muốn gửi gói tin cho một nhóm máy.
- Broadcast: được thực hiện khi một máy muốn gửi cho tất cả các máy khác trong mạng.

Khi một thiết bị muốn gửi một gói quảng bá thì địa chỉ MAC đích của gói tin đó sẽ là FF:FF:FF:FF:FF:FF. Với địa chỉ như vậy, mọi thiết bị đều nhận và xử lý gói quảng bá.

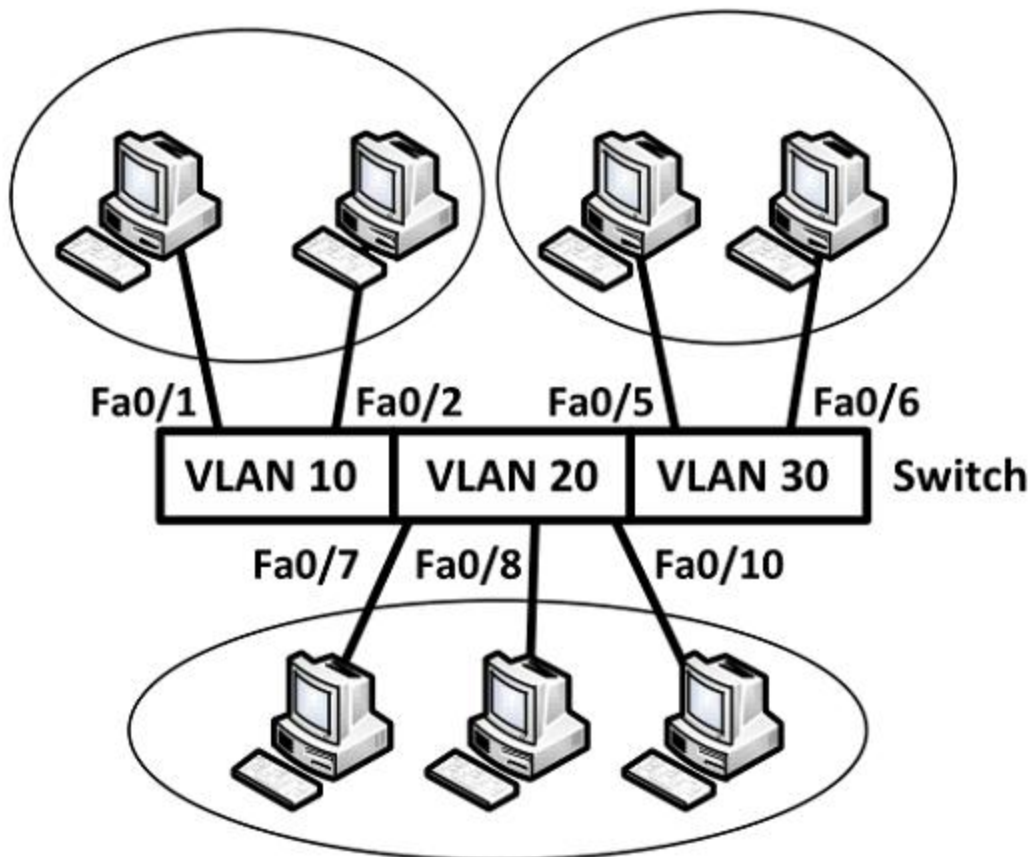
Miền quảng bá là miền bao gồm tất cả các thiết bị có thể nhận được gói tin quảng bá từ một thiết bị nào đó trong LAN.

Switch là thiết bị hoạt động ở tầng liên kết dữ liệu, khi Switch nhận được gói quảng bá thì nó sẽ gửi ra tất cả các cổng của nó trừ cổng nhận gói tin vào. Mỗi thiết bị nhận được gói quảng bá đều phải xử lý thông tin nằm trong đó.

Router là thiết bị hoạt động ở tầng mạng, router không chuyển các gói quảng bá. Router được sử dụng để chia mạng thành nhiều miền đưng độ và nhiều miền quảng bá.

2. VLAN

VLAN (Virtual LAN) là kỹ thuật được sử dụng trên Switch, dùng để chia một Switch vật lý thành nhiều Switch luận lý. Mỗi một Switch luận lý gọi là một VLAN hoặc có thể hiểu VLAN là một tập hợp của các cổng trên Switch nằm trong cùng một miền quảng bá. Các cổng trên Switch có thể được nhóm vào các VLAN khác nhau trên một Switch hoặc được triển khai trên nhiều Switch.



Hình 2.1 Chia VLAN trên switch

Khi có một gói tin quảng bá được gửi bởi một thiết bị nằm trong một VLAN sẽ được chuyển đến các thiết bị khác nằm trong cùng VLAN đó, gói tin quảng bá sẽ không được chuyển tiếp đến các thiết bị thuộc VLAN khác.

VLAN cho phép người quản trị tổ chức mạng theo luận lý chứ không theo vật lý. Sử dụng VLAN có ưu điểm là:

- ✓ Tăng khả năng bảo mật
- ✓ Thay đổi cấu hình LAN dễ dàng
- ✓ Di chuyển máy trạm trong LAN dễ dàng
- ✓ Thêm máy trạm vào LAN dễ dàng.

VLAN = broadcast domain = logical network

3. PHÂN LOẠI

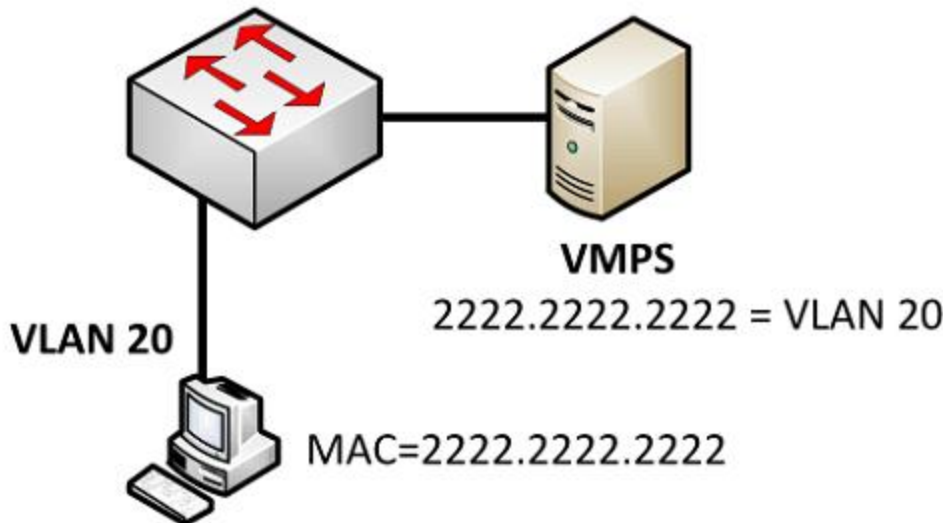
- VLAN tĩnh (Static VLAN)



Hình 2.2 VLAN tĩnh

Đối với loại này, các cổng của Switch được cấu hình thuộc về một VLAN nào đó, các thiết bị gắn vào cổng đó sẽ thuộc về VLAN đã định trước. Đây là loại VLAN dùng phổ biến.

- VLAN động (dynamic VLAN)



Hình 2.3 VLAN động

Loại VLAN này sử dụng một server lưu trữ địa chỉ MAC của các thiết bị và qui định VLAN mà thiết bị đó thuộc về, khi một thiết bị gắn vào Switch, Switch sẽ lấy địa chỉ MAC của thiết bị và gửi cho server kiểm tra và cho vào VLAN định trước.

4. CẤU HÌNH VLAN

Bước 1. Tạo VLAN

```
Switch(config)#vlan <vlan-id>  
Switch(config-vlan)#name <vlan-name>
```

Ví dụ:

```
Switch(config)#vlan 10  
Switch(config-if)#name P.KyThuat
```

Bước 2. Gán các cổng cho VLAN

- Gán 1 cổng vào LAN

```
Switch(config)#interface <interface>  
Switch(config-if)#switchport mode access  
Switch(config-if)#switchport access vlan <vlan-id>
```

Ví dụ:

```
Switch(config)#interface fa0/5  
Switch(config-if)#switchport mode access  
Switch(config-if)#switchport access vlan 10
```

- Gán 1 dãy các cổng liên tiếp

```
Switch(config)#interface range <start>-<end-intf>  
Switch(config-if-range)#switchport mode access  
Switch(config-if-range)#switchport access vlan  
<vlan-id>
```

Ví dụ:

```
Switch(config)#interface fa0/10 - 20  
Switch(config-if-range)#switchport mode access  
Switch(config-if-range)#switchport access vlan 10
```

- Gán nhiều cổng không liên tiếp

```
Switch(config)#interface range <interface1,  
interface2,...>  
Switch(config-if-range)#switchport mode access  
Switch(config-if-range)#switchport access vlan  
<vlan-id>
```

Ví dụ:

```
Switch(config)#interface fa0/7, fa0/9, fa0/2  
Switch(config-if-range)#switchport mode access  
Switch(config-if-range)#switchport access vlan 10
```

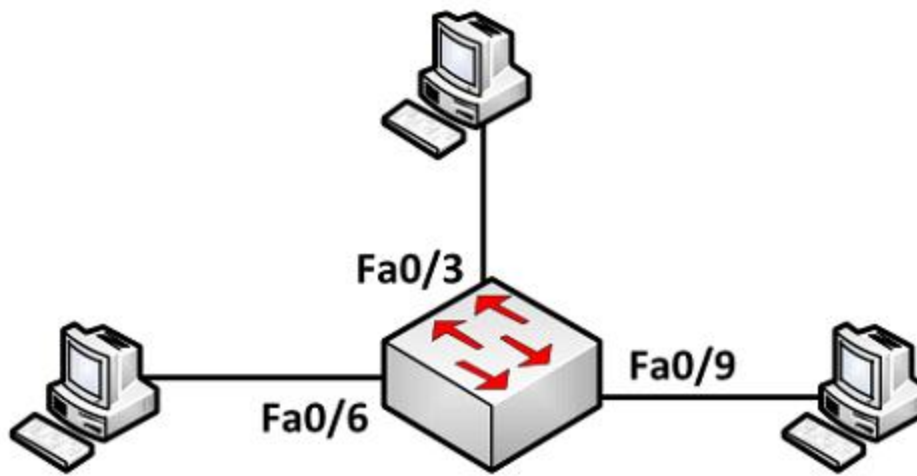
❖ **Xóa VLAN:** Xóa một VLAN trên switch bằng cách sử dụng lệnh “no” trước câu lệnh tạo VLAN.

❖ **Lệnh kiểm tra cấu hình VLAN**

```
Switch#show vlan
```

Lệnh này cho phép hiển thị các VLAN-ID (số hiệu VLAN), tên VLAN, trạng thái VLAN và các cổng được gán cho VLAN trên switch.

Ví dụ:



Mô tả yêu cầu:

- Cấu hình VLAN trên Switch
- Tạo 3 VLAN: VLAN 10, VLAN 20, VLAN 30
- Fa0/1 – Fa0/6: VLAN 10, Fa0/7 – Fa0/9: VLAN 20, Fa0/10 – Fa0/12: VLAN 30

Các bước thực hiện:

✓ **Tạo vlan:**

```
Switch(config)#vlan 10  
Switch(config)#vlan 20  
Switch(config)#vlan 30
```

✓ **Gán các cổng vào VLAN**

```
Switch(config)#interface range f0/1 - 6  
Switch(config-if-range)#switchport mode access  
Switch(config-if-range)#switchport access vlan 10
```

```

Switch(config)#interface range f0/7 - 9
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
Switch(config)#interface range f0/10 - 12
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 30

```

✓ **Kiểm tra cấu hình:**

Thực hiện các câu lệnh sau để kiểm tra cấu hình

```

Switch#show run
Switch#show vlan

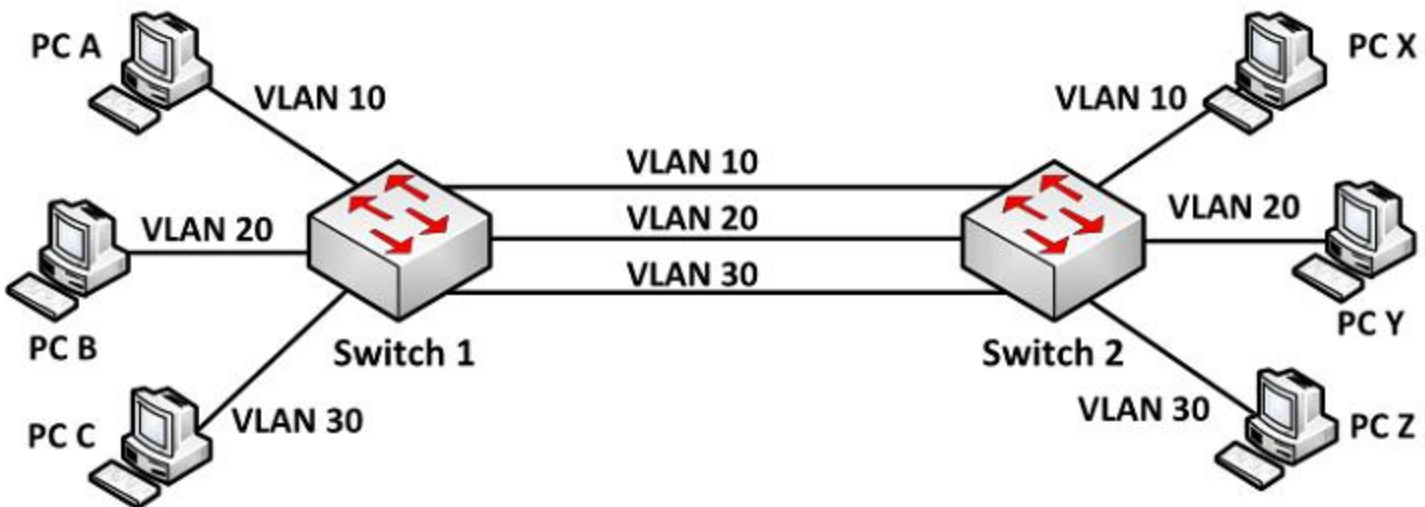
```

Gắn PC vào các cổng như trên sơ đồ, đặt IP cho các PC và dùng lệnh “ping” để kiểm tra kết nối.

5. ĐƯỜNG TRUNK

VLAN tổ chức trên nhiều switch như vậy làm sao các thiết bị thuộc cùng một VLAN nhưng nằm ở những switch khác nhau có thể liên lạc với nhau? Chúng ta có hai cách để giải quyết vấn đề này:

- Dùng mỗi kết nối cho từng VLAN



Hình 2.4 Sử dụng mỗi kết nối cho từng VLAN

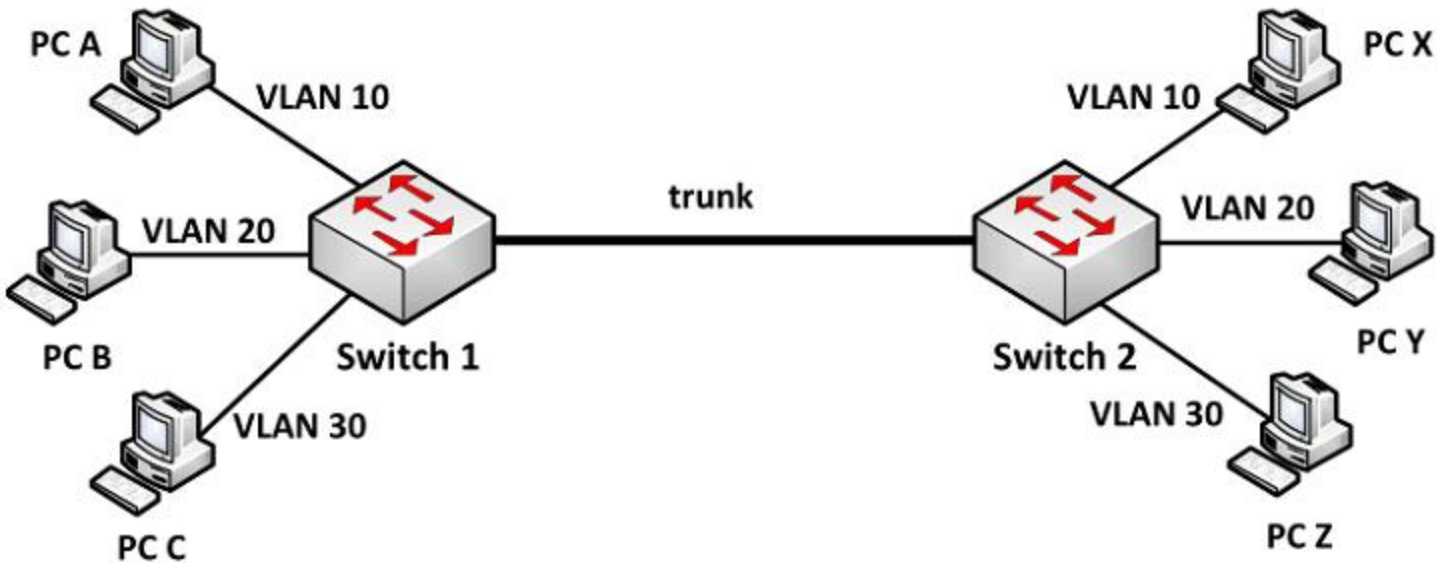
Có nghĩa là mỗi VLAN ở trên các switch sẽ được kết nối lại bằng một đường kết nối riêng. Theo mô hình trên ta thấy: nếu PC A trong VLAN 10 ở Switch 1 muốn liên lạc với PC X trong VLAN 10 ở Switch 2, ta phải có một kết nối vật lý nối Switch 1 với Switch 2 và hai cổng kết nối này phải thuộc cùng VLAN 10.

Tương tự đối với VLAN 2 và VLAN3, ta cần hai kết nối vật lý. Như vậy, với n VLAN được tạo ra tổng cộng ta phải dùng đến n dây nối

để các thành viên trong cùng VLAN có thể giao tiếp được với nhau. Điều này gây ra lãng phí.

- Kết nối trunk (đường trunk)

Một kỹ thuật khác để giải quyết vấn đề trên là dùng chỉ một kết nối cho phép dữ liệu của các VLAN có thể cùng lưu thông qua đường này. Người ta gọi kết nối này là đường *trunk*.



Hình 2.5 Kết nối trunk cho các VLAN

Theo như mô hình trên chúng ta chỉ dùng một dây nối Switch 1 với Switch 2, các thành viên trong cùng VLAN ở các Switch khác nhau vẫn có thể giao tiếp với nhau. Đường dây như thế gọi là liên kết trunk lớp 2.

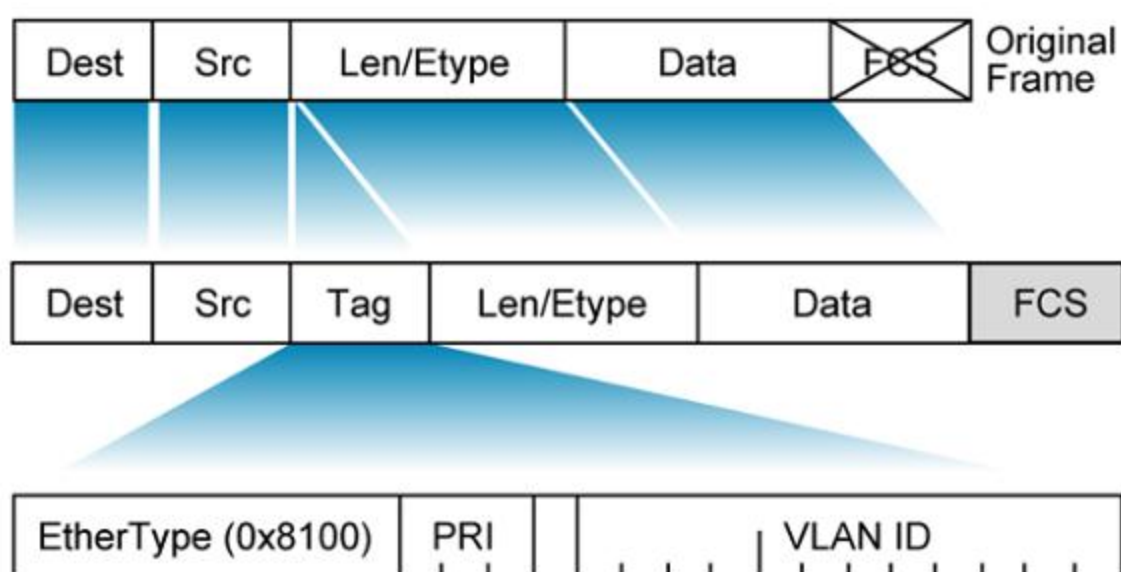
Mỗi thành viên trong cùng VLAN chỉ có thể thấy thành viên khác trong cùng VLAN với nó. Để PC A có thể giao tiếp với PC B hoặc C (không thuộc cùng VLAN), cần phải sử dụng thiết bị ở lớp 3 như router hay switch lớp 3 (Multilayer Switch hay Switch layer 3).

Kết nối “*trunk*” là liên kết Point-to-Point giữa các cổng trên switch với router hoặc với switch khác. Kết nối “*trunk*” sẽ vận chuyển dữ liệu của nhiều VLAN thông qua một liên kết đơn và cho phép mở rộng VLAN trên hệ thống mạng.

Vì kỹ thuật này cho phép dùng chung một kết nối vật lý cho dữ liệu của các VLAN đi qua nên để phân biệt được chúng là dữ liệu của VLAN nào, người ta gắn vào các gói tin một dấu hiệu gọi là “*tagging*”. Hay nói cách khác là dùng một kiểu đóng gói riêng cho các gói tin di chuyển qua đường “*trunk*” này. Giao thức được sử dụng là 802.1Q (dot1q).

❖ Giao thức 802.1Q

Đây là giao thức chuẩn của IEEE để dành cho việc nhận dạng các VLAN bằng cách thêm vào “frame header” đặc điểm của một VLAN. Phương thức này còn được gọi là gắn thẻ cho VLAN (frame tagging).



Hình 2.6 Frame được đóng gói theo kiểu 802.1Q

❖ Cấu hình VLAN trunking:

Để cấu hình đường “trunk”, chúng ta cấu hình 2 cổng “trunk” như sau:

```
switch(config)#interface <interface>
switch(config-if)#switchport mode trunk
switch(config-if)#switchport trunk encapsulation
dot1q
```

Lệnh cuối cùng là mặc định ở một số dòng switch

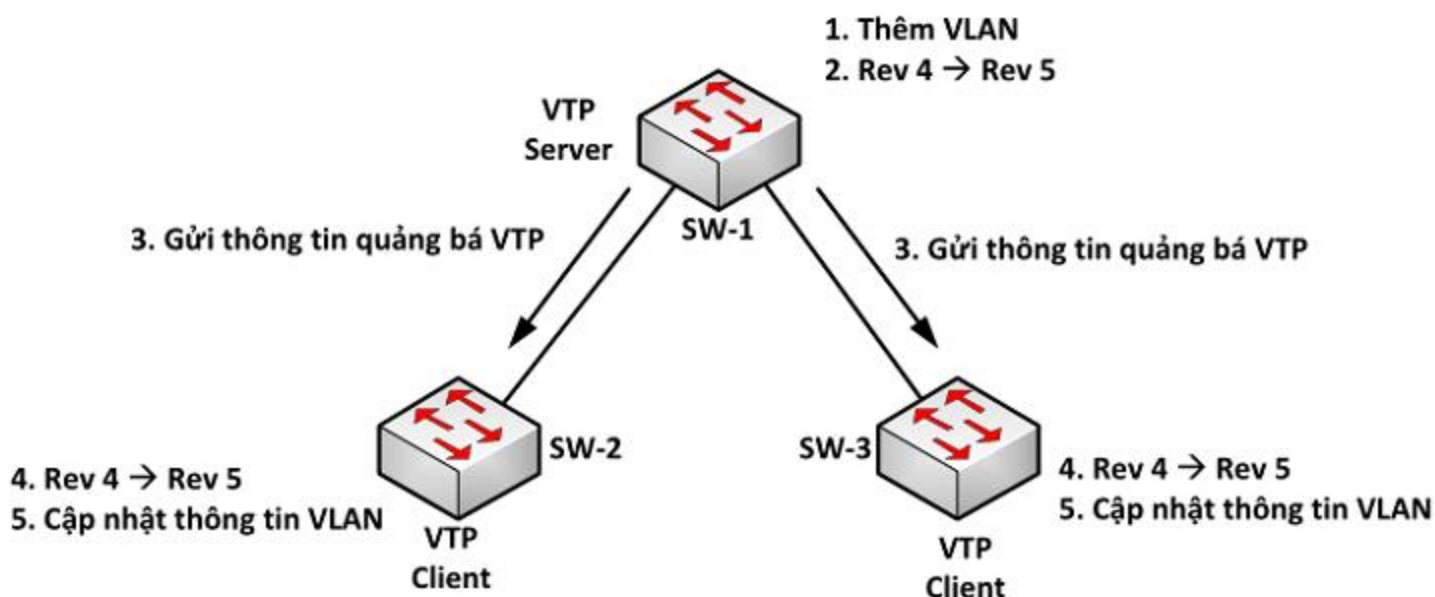
6. VLAN TRUNKING PROTOCOL (VTP)

VTP là giao thức hoạt động ở tầng liên kết dữ liệu trong mô hình OSI. VTP giúp cho việc cấu hình VLAN luôn đồng nhất khi thêm, xóa, sửa thông tin về VLAN trong hệ thống mạng.

❖ Hoạt động của VTP

VTP gửi thông điệp quảng bá qua “VTP domain” mỗi 5 phút một lần, hoặc khi có sự thay đổi xảy ra trong cấu hình VLAN. Một thông điệp VTP bao gồm “*revision-number*”, tên VLAN (VLAN name), số hiệu VLAN (VLAN number), và thông tin về các switch có cổng gắn với mỗi VLAN. Bằng sự cấu hình *VTP Server* và việc quảng bá thông tin VTP, tất cả các switch đều đồng bộ về tên VLAN và số hiệu VLAN của tất cả các VLAN.

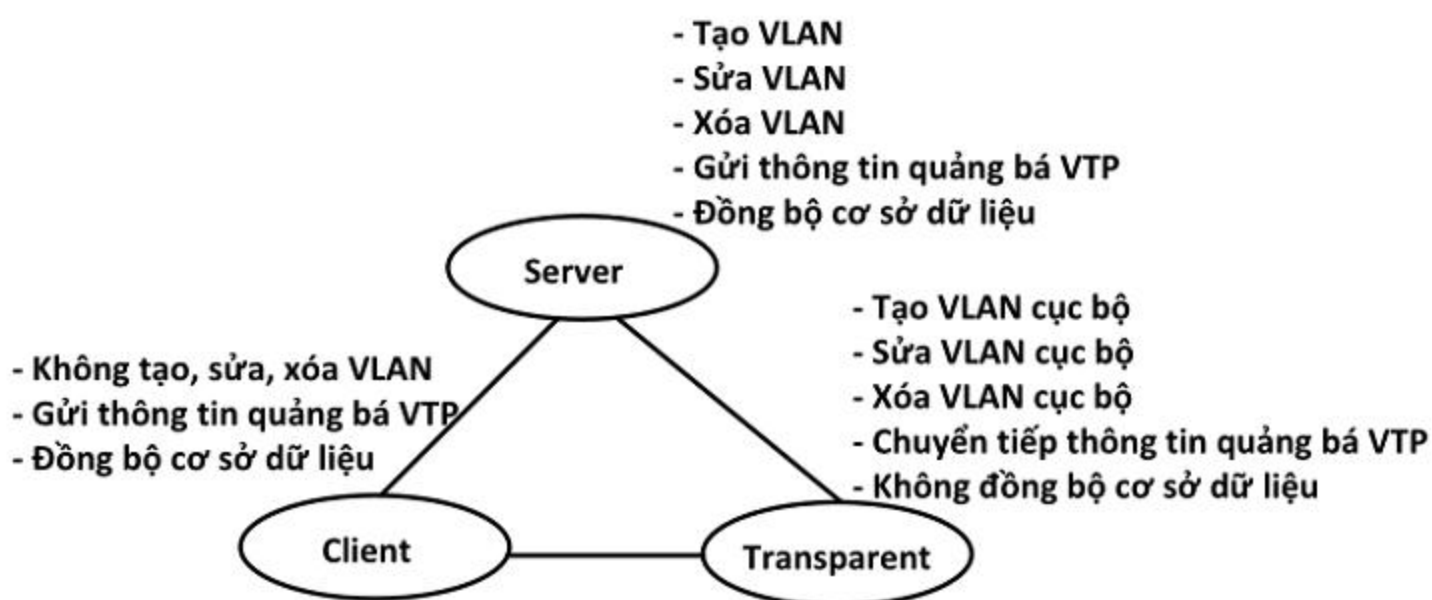
Một trong những thành phần quan trọng trong các thông tin quảng bá VTP là tham số “*revision number*”. Mỗi lần VTP server điều chỉnh thông tin VLAN, nó tăng “*revision-number*” lên 1, rồi sau đó *VTP Server* mới gửi thông tin quảng bá VTP đi. Khi một switch nhận một thông điệp VTP với “*revision-number*” lớn hơn, nó sẽ cập nhật cấu hình VLAN.



Hình 2.7 Hoạt động của VTP

❖ **VTP hoạt động ở một trong ba chế độ**

- Server
- Client
- Transparent



Hình 2.8 Các mode của VTP

Switch ở chế độ *VTP server* có thể tạo, chỉnh sửa và xóa VLAN. VTP server lưu cấu hình VLAN trong NVRAM của nó. *VTP Server* gửi thông điệp ra tất cả các cổng “trunk”.

Switch ở chế độ *VTP client* không tạo, sửa và xóa thông tin VLAN. *VTP Client* có chức năng đáp ứng theo mọi sự thay đổi của VLAN từ *Server* và gửi thông điệp ra tất cả các cổng “trunk” của nó. *VTP Client* đồng bộ cấu hình VLAN trong hệ thống.

Switch ở chế độ *transparent* sẽ nhận và chuyển tiếp các thông điệp quảng bá VTP do các switch khác gửi đến mà không quan tâm đến nội dung của các thông điệp này. Nếu “transparent switch” nhận được thông

tin cập nhật VTP nó cũng không cập nhật vào cơ sở dữ liệu của nó; đồng thời nếu cấu hình VLAN của nó có gì thay đổi, nó cũng không gửi thông tin cập nhật cho các switch khác. Trên “*transparent switch*” chỉ có một việc duy nhất là chuyển tiếp thông điệp VTP. Switch hoạt động ở “*transparent-mode*” chỉ có thể tạo ra các VLAN cục bộ. Các VLAN này sẽ không được quảng bá đến các switch khác.

❖ Cấu hình VTP

- Cấu hình VTP domain

```
Switch(config)#vtp domain <domain_name>
```

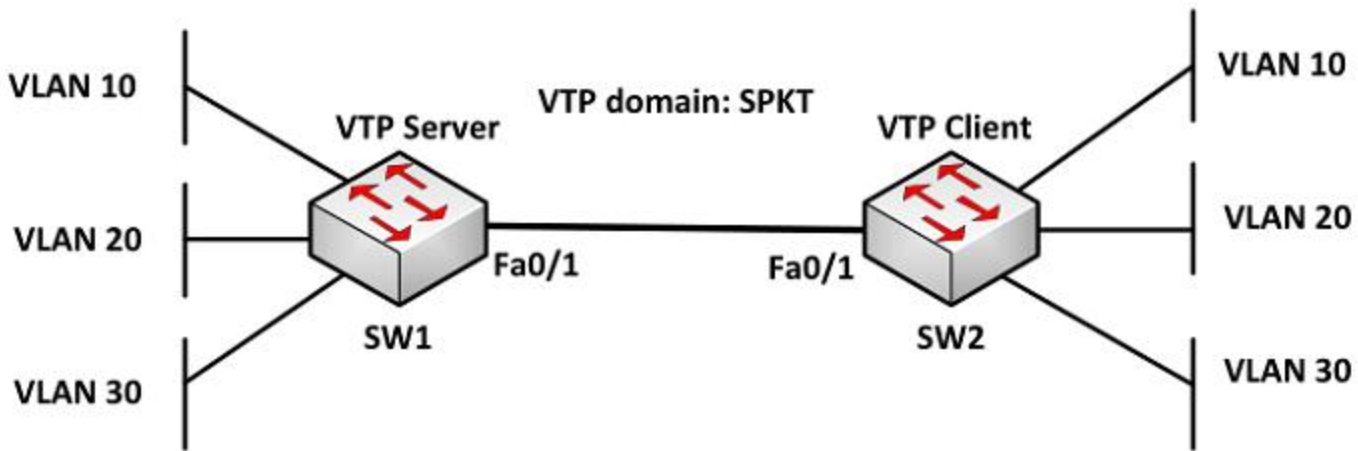
- Cấu hình VTP mode

```
Switch(config)#vtp [client| transparent| server]
```

- Lệnh xem cấu hình VTP

```
Switch#show vtp status
```

Ví dụ: Cho sơ đồ mạng



Mô tả

- ✓ Hai switch kết nối với nhau qua đường “trunk”.
- ✓ Tạo 3 vlan: VLAN 10, VLAN 20, VLAN 30 trên SW1
- ✓ Cấu hình VTP để các thông tin các VLAN trên SW1 cập nhật cho SW2
- ✓ Trên SW1: VLAN 10 (Fa0/2 – Fa0/4), VLAN 20 (Fa0/5 – Fa0/7), VLAN 30 (Fa0/8 – Fa0/10)
- ✓ Trên SW2: VLAN 10 (Fa0/4 – Fa0/6), VLAN 20 (Fa0/7 – Fa0/9), VLAN 30 (Fa0/10 – Fa0/12)

Các bước cấu hình

Cấu hình Sw1 làm VTP Server:

- ✓ Thiết lập VTP domain: SPKT, VTP mode Server, và tạo các VLAN

```
sw1#config terminal
sw1(config)#vtp mode server
sw1(config)#vtp domain SPKT
sw1(config)#vlan 10 name CNTT
sw1(config)#vlan 20 name TTTH
sw1(config)#vlan 30 name TTCLC
```

- ✓ Cấu hình đường trunk và cho phép tất cả các VLAN qua đường trunk

```
sw1(config)#interface f0/1
sw1(config-if)#switchport mode trunk
sw1(config-if)#switchport trunk encapsulation dot1q
```

- ✓ Gán các port vào các VLAN

```
sw1(config)#int range f0/2 - 4
sw1(config-if-range)#switchport mode access
sw1(config-if-range)#switchport access vlan 10

sw1(config-if)#int range f0/5 - 7
sw1(config-if-range)#switchport mode access
sw1(config-if-range)#switchport access vlan 20

sw1(config-if)#int range f0/8 - 10
sw1(config-if-range)#switchport mode access
sw1(config-if-range)#switchport access vlan 30
```

- ✓ **Kiểm tra cấu hình**

Sử dụng các lệnh: switch#show vlan
 switch# show vtp status

Cấu hình Sw2 làm VTP client:

- ✓ **Cấu hình vtp domain: SPKT, vtp mode: client**

```
SW2(config)#vtp domain SPKT
SW2(config)#vtp mode client
```

- ✓ **Cấu hình trunking trên cổng f0/1 của SW2**

```
SW2(config)#int f0/1
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk encapsulation dot1q
```

✓ **Gán các port vào các vlan**

```
sw2(config)#int range f0/4 - 6
sw2(config-if-range)#switchport mode access
sw2(config-if-range)#switchport access vlan 10

sw2(config)#int range f0/7 - 9
sw2(config-if-range)#switchport mode access
sw2(config-if-range)#switchport access vlan 20

sw2(config)#int range f0/10 - 12
sw2(config-if-range)#switchport mode access
sw2(config-if-range)#switchport access vlan 30
```

✓ **Kiểm tra**

Sử dụng các câu lệnh sau

```
switch#show vlan
switch#show int interface
switch#show vtp status
switch#show vtp counters : kiểm tra số lần gửi và nhận
thông tin trunking
```

7. ĐỊNH TUYẾN GIỮA CÁC VLAN

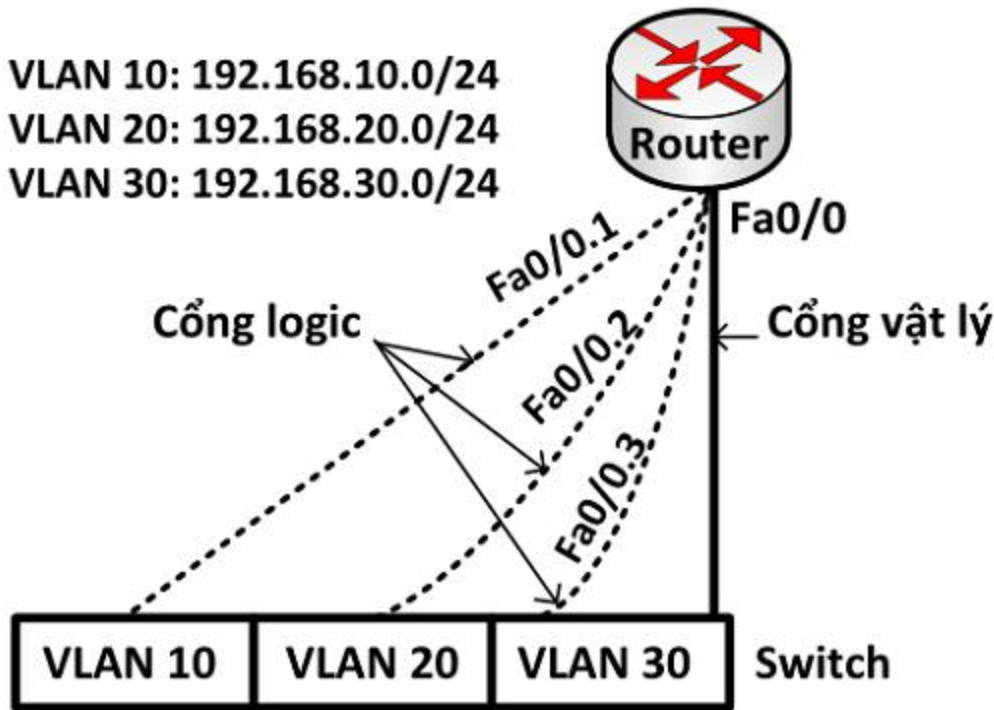
Mỗi VLAN là một miền quảng bá. Do đó, mỗi thiết bị trong VLAN chỉ liên lạc được với các thiết bị khác trong cùng một VLAN. Nếu một máy tính trong một VLAN muốn liên lạc với một máy tính thuộc một VLAN khác thì nó phải thông qua thiết bị định tuyến như là router.

Router trong cấu trúc VLAN thực hiện ngăn chặn quảng bá, bảo mật và quản lý các lưu lượng mạng. Switch layer 2 không thể chuyển dữ liệu giữa các VLAN với nhau. Dữ liệu trao đổi giữa các VLAN phải được định tuyến qua thiết bị hoạt động ở tầng mạng như router.

Giả sử trên switch tạo 3 VLAN, nếu ta dùng 3 cổng của router để định tuyến cho 3 VLAN này thì quá cồng kềnh và không tiết kiệm. Ta chỉ cần sử dụng 1 cổng trên router kết nối với một cổng trên switch và cấu hình đường này làm đường *trunk* (trunk layer 3) để định tuyến cho các VLAN.

Đường kết nối cho phép mang lưu lượng của nhiều VLAN gọi là kết nối *trunk lớp 3*. Nó không phải là của riêng VLAN nào. Ta có thể cấu hình một đường *trunk* để vận chuyển lưu thông cho tất cả VLAN hoặc một số VLAN cụ thể nào đó được chỉ ra trong cấu hình. *Trunking layer 3* đòi hỏi cổng trên VLAN phải có thể hoạt động ở tốc độ FastEthernet trở lên.

❖ Cổng vật lý và cổng logic



Hình 2.9 Định tuyến giữa các VLAN

Đường “trunk” có ưu điểm là làm giảm số lượng cổng cần sử dụng của router và switch. Điều này không chỉ tiết kiệm chi phí mà còn giúp cho cấu hình bớt phức tạp. Kết nối “trunk” trên router có khả năng mở rộng với số lượng lớn VLAN. Nếu mỗi VLAN phải có một kết nối vật lý thì không thể đáp ứng được khi số lượng VLAN lớn.

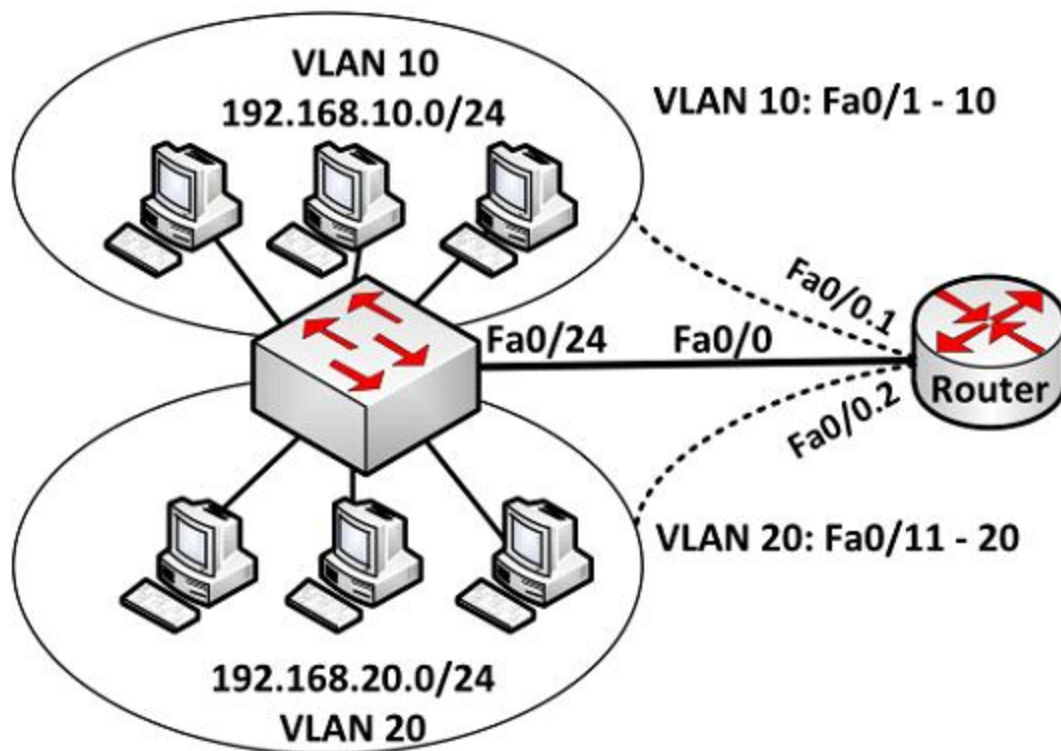
Một cổng vật lý có thể được chia thành nhiều cổng luận lý. Mỗi cổng luận lý tương ứng với một VLAN và được đặt một địa chỉ IP của vlan đó. Mỗi VLAN là một mạng riêng, do đó cổng luận lý thuộc VLAN nào thì có địa chỉ IP thuộc mạng của VLAN đó.

❖ Cấu hình định tuyến cho các VLAN dùng Router

Sử dụng các cổng luận lý được chia từ một cổng vật lý để cấu hình định tuyến giữa các VLAN, các câu lệnh được sử dụng như sau:

```
R(config)#interface <port-number.subintf-number>  
R(config-if)#encapsulation dot1q <vlan-id>  
R(config-if)#ip address <address> <subnet-mask>
```

Ví dụ: Cấu hình định tuyến giữa các VLAN



Yêu cầu

- Tạo 2 vlan: **VLAN 10** (P.KinhDoanh) và **VLAN 20** (P.KeToan)
- Các cổng Fa0/1–Fa0/10 thuộc VLAN 10, các cổng Fa0/11–Fa0/20 thuộc VLAN 20
- Cấu hình định tuyến cho phép hai VLAN này có thể liên lạc được với nhau.

Các bước thực hiện

• Cấu hình trên switch

✓ Tạo vlan

```
switch(config)#vlan 10
switch(config-vlan)#name P.KinhDoanh
switch(vlan)#vlan 20
switch(config-vlan)#name P.KeToan
```

✓ Gán các port vào vlan

```
switch(config)#interface range fa0/1 - 10
switch(config-if-range)#switchport mode access
switch(config-if-range)#switchport access vlan 10

switch(config)#int fa0/11 - 20
switch(config-if-range)#switchport mode access
switch(config-if-range)#switchport access vlan 20
```

✓ Cấu hình đường trunk

```
switch(config)#int fa0/24
switch(config-if)#switchport mode trunk
switch(config-if)#switchport trunk encapsulation dot1q
```


Lưu ý: Lệnh cuối là mặc định trên một số dòng switch.

- **Cấu hình trên router**

- ✓ Chọn cổng fa0/0 để cấu hình trunk

```
router(config)#interface fa0/0
router(config-if)#no shutdown
```

- ✓ Kích hoạt *trunk* trên subinterface fa0/0.1 và đóng gói bằng dot1q

```
router(config)#int fa0/0.1
router(config-if)#encapsulation dot1q 10
```

- ✓ Cấu hình thông tin lớp 3 cho sub-interface **fa0/0.1**

```
router(config-subif)#ip address 192.168.1.1
255.255.255.0
```

- ✓ Kích hoạt “trunk” trên sub-interface **fa0/0.2** và đóng gói bằng **dot1q**

```
router(config)#int fa0/0.2
router(config-subif)#encapsulation dot1q 20
```

- ✓ Cấu hình thông tin lớp 3 cho sub-interface **fa0/0.2**

```
router(config-subif)#ip address 192.168.2.1
255.255.255.0
```

- ✓ Lưu cấu hình

```
router#copy run start
```

- **Kiểm tra cấu hình**

Trên switch dùng các lệnh sau:

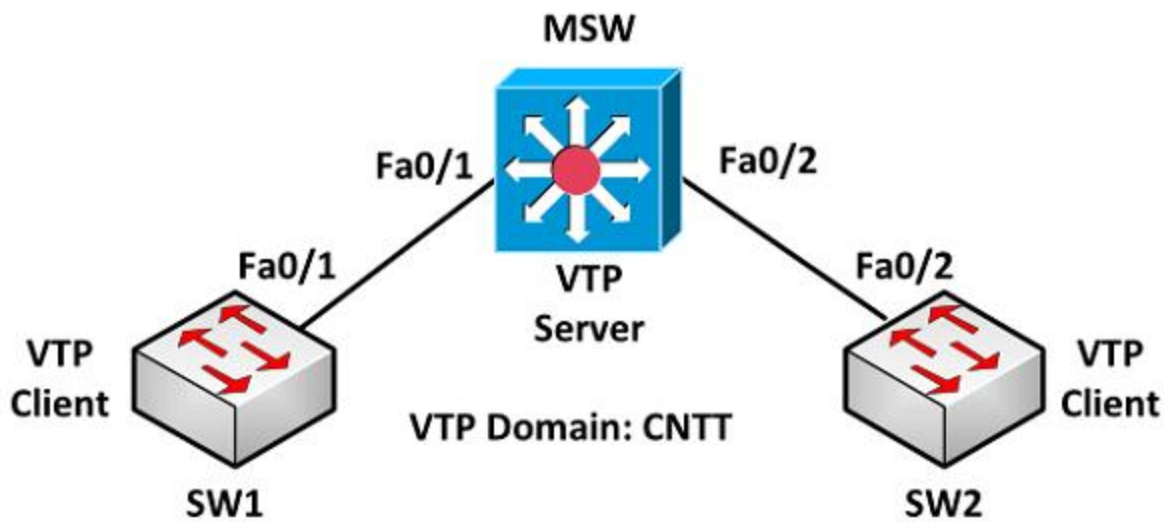
```
Switch#show interface interface
Switch#show vlan
Switch#show vtp status
```

Trên router dùng các lệnh sau

Router#show vlan: thông tin layer 2 và layer 3 cấu hình cho mỗi VLAN.

```
Router#show interfaces <interface>
```

- ❖ Định tuyến cho các VLAN dùng switch layer 3 (MSW)



VLAN10: 192.168.10.0/24

VLAN20: 192.168.20.0/24

VLAN30: 192.168.30.0/24

VLAN40: 192.168.40.0/24

Yêu cầu

- Cấu hình đường “trunk”
- Cấu hình VTP, VLAN
VTP domain: CNTT; MSW: VTP Server; SW1, SW2: VTP Client
- Cấu hình MSW để định tuyến cho 4 VLAN

Hướng dẫn cấu hình

✓ Cấu hình trunk

```
SW1(config)#interface fa0/1
SW1(config-if)#switchport mode trunk
SW1(config-if)#switchport trunk encapsulation dot1q
SW2(config)#interface fa0/2
SW2(config-if)#switchport mode trunk
SW2(config-if)#switchport trunk encapsulation dot1q
MSW(config)#interface fa0/1
MSW(config-if)#switchport mode trunk
MSW(config-if)#switchport trunk encapsulation dot1q
MSW(config)#interface fa0/2
MSW(config-if)#switchport mode trunk
MSW(config-if)#switchport trunk encapsulation dot1q
```

✓ Cấu hình VTP, VLAN

```
MSW(config)#vtp domain CNTT
MSW(config)#vtp mode server
```

```
SW1(config)#vtp domain CNTT
SW1(config)#vtp mode client
SW2(config)#vtp domain CNTT
SW2(config)#vtp mode client
MSW(config)#vlan 10
MSW(config)#vlan 20
MSW(config)#vlan 30
MSW(config)#vlan 40
```

✓ **Cấu hình MSW để routing giữa 4 VLAN**

```
MSW(config)#ip routing
MSW(config)#interface vlan 10
MSW(config-if)#ip address 192.168.10.1 255.255.255.0
MSW(config)#interface vlan 20
MSW(config-if)#ip address 192.168.20.1 255.255.255.0
MSW(config)#interface vlan 30
MSW(config-if)#ip address 192.168.30.1 255.255.255.0
MSW(config)#interface vlan 40
MSW(config-if)#ip address 192.168.40.1 255.255.255.0
```

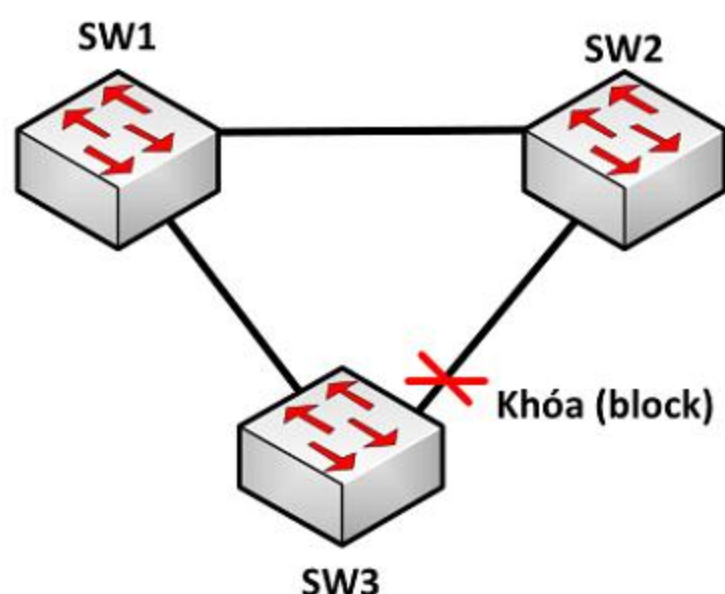
✓ **Kiểm tra cấu hình**

```
show interface trunk
show vtp status
show vlan brief
show ip route
```

8. GIAO THỨC STP (SPANNING TREE PROTOCOL)

Trong thiết kế mạng, việc tạo ra các kết nối dư thừa là cần thiết nhằm tạo khả năng dự phòng cho hệ thống. Tuy nhiên, khi thiết kế dự phòng trên Switch thì có 3 vấn đề cần xem xét là: bão quảng bá, nhiều gói tin được nhận giống nhau và bảng địa chỉ MAC trên các Switch không ổn định. Có thể gọi chung trường hợp này là “switching loop”.

Giao thức STP được sử dụng để giải quyết vấn đề này bằng cách khóa tạm thời một hoặc một số cổng để tránh tình trạng như trên.



❖ Hoạt động của STP qua các bước sau:

- Bầu chọn 1 switch làm “*Root switch*” còn gọi là “*Root bridge*”
- Chọn “*Root port*” trên các switch còn lại
- Chọn “*Designated port*” trên mỗi phân đoạn (segment) mạng
- Cổng còn lại gọi là “*Nondesigned port*” sẽ bị khóa

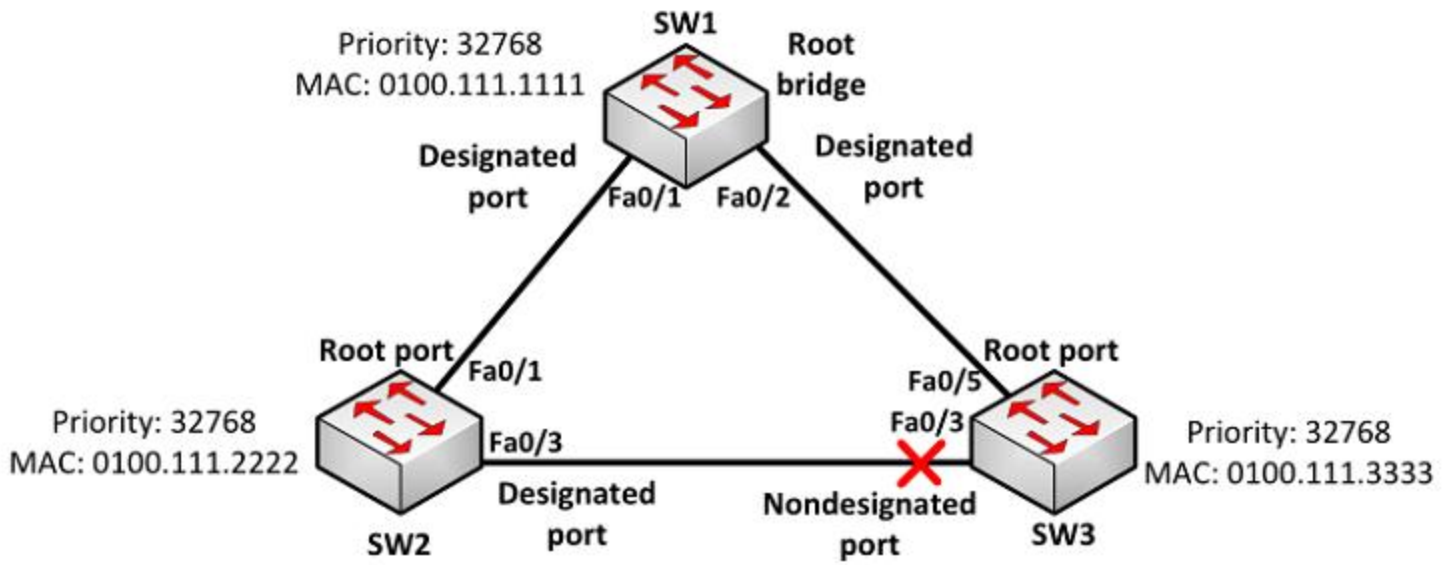
❖ Quá trình bầu chọn “root switch”

Mỗi switch có một giá trị “*Bridge-ID*” gồm 2 trường là “*Bridge priority*” và “*MAC address*” và được đặt vào trong BPDU và gửi quảng bá cho các switch khác mỗi 2 giây. Switch được chọn làm “root switch” là switch có giá trị “*Bridge-ID*” nhỏ nhất. Để so sánh, giá trị “*Bridge priority*” được dùng để so sánh trước, nếu tất cả các switch đều có giá trị này bằng nhau thì tham số thứ 2 là “*MAC address*” sẽ được dùng để so sánh.

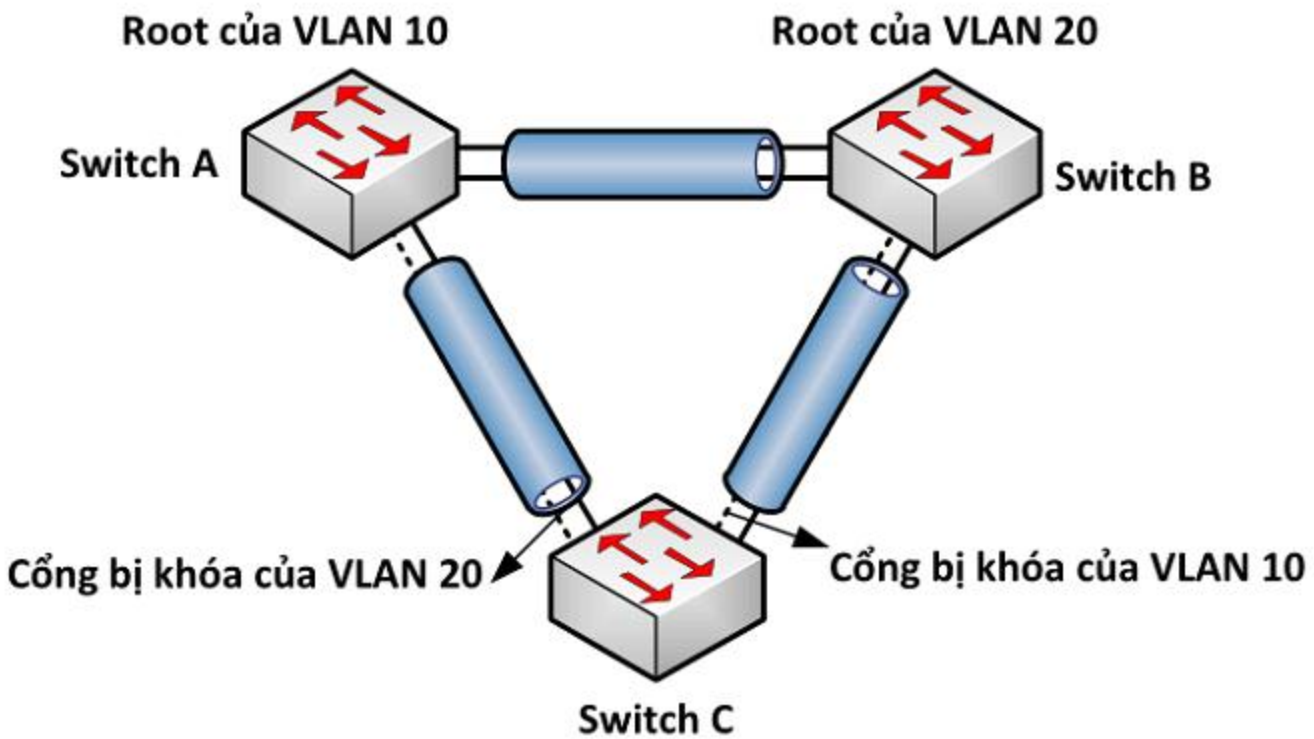
Các loại cổng khác “root port”, “designated port” sẽ lần lượt được bầu chọn dựa vào chi phí nhỏ nhất tính từ nó đến “root switch”. Dựa vào bảng sau để tính chi phí cho mỗi chặn.

| Tốc độ kết nối | Chi phí (Cost) |
|----------------|----------------|
| 10 Gb/s | 2 |
| 1 Gb/s | 4 |
| 100 Mb/s | 19 |
| 10 Mb/s | 100 |

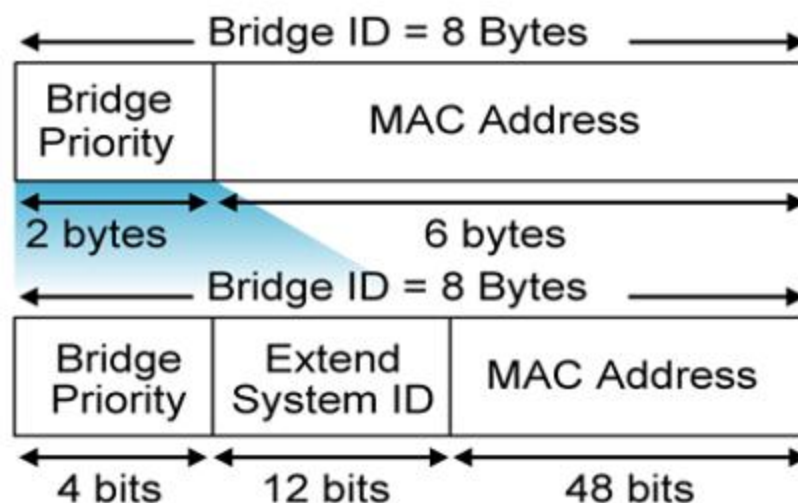
Ví dụ:



Một số dạng STP được cải tiến như: PVSTP+ (Per VLAN Spanning Tree Plus) dùng tạo cho mỗi VLAN một STP riêng.



Trong PVSTP+, *Bridge-ID* có thêm trường *System-ID* (VLAN-ID) để phân biệt cho từng VLAN.



Một số cải tiến khác như RSTP (Rapid Spanning Tree Protocol), MSTP.

Một số lệnh cấu hình để điều chỉnh giá trị “Bridge priority” mặc định của switch. Chọn switch làm “root switch” bằng lệnh sau:

```
Switch(config)#spanning-tree vlan <vlan-id> root primary
```

Hoặc

```
Switch(config)#spanning-tree vlan <vlan-id> priority  
<priority>
```

9. TỔNG KẾT CHƯƠNG

Trong môi trường Ethernet LAN, tập hợp các thiết bị cùng nhận một gói quảng bá bởi bất kỳ một thiết bị còn lại được gọi là một “*broadcast domain*”. Trên các switch không hỗ trợ VLAN, switch sẽ gửi tất cả các gói tin quảng bá ra tất cả các cổng, ngoại trừ cổng mà nó nhận gói tin vào. Kết quả là trên các cổng của loại switch này là cùng một “*broadcast domain*”. Nếu switch này kết nối đến các switch và các hub khác, các cổng trên switch này sẽ cùng “*broadcast domain*”.

VLAN cho phép kết hợp các cổng trên switch thành các nhóm để giảm lưu lượng broadcast. VLAN là một LAN theo logic dựa trên chức năng, ứng dụng của một tổ chức chứ không phụ thuộc vào vị trí vật lý hay kết nối vật lý trong mạng. Một VLAN là một miền quảng bá được tạo nên bởi một hay nhiều switch.

Giao thức VTP có vai trò duy trì cấu hình của VLAN và đồng nhất trên toàn mạng. VTP là giao thức sử dụng đường trunk để quản lý sự thêm, xóa, sửa các VLAN trên toàn mạng từ switch trung tâm được đặt trong *Server mode*. VTP hoạt động chủ yếu là đồng nhất các thông tin VLAN trong cùng một VTP domain giúp giảm đi sự cấu hình giống nhau trong các switch.

Kết nối *trunk* là liên kết Point-to-Point giữa các cổng trên switch với router hoặc với switch khác. Kết nối *trunk* sẽ vận chuyển thông tin của nhiều VLAN thông qua một liên kết đơn và cho phép mở rộng VLAN trên hệ thống mạng. Các VLAN được định tuyến sử dụng thiết bị ở tầng 3 như router hay “Switch layer 3”.

Giao thức STP được dùng trong trường hợp hệ thống mạng thiết kế các kết nối dự phòng trên Switch. STP chống tình trạng “switching loop” bằng cách khóa tạm một số cổng trong mạng. Một số phiên bản cải tiến từ STP truyền thống như PVSTP+, RSTP,...

10. CÂU HỎI VÀ BÀI TẬP

10.1 Một VLAN là một tập các thiết bị nằm cùng miền _____.

- A. Autonomous system
- B. Broadcast domain
- C. Bandwidth domain
- D. Collision domain

10.2 Thiết bị nào sau đây được dùng để kết nối các VLAN?

- A. Switch
- B. Bridge
- C. Router
- D. Hub

10.3 Giao thức nào sau đây được dùng để phân phối thông tin về cấu hình VLAN đến các Switch khác trong mạng?

- A. STP
- B. VTP
- C. EIGRP
- D. SNMP
- E. CDP

10.4 Giao thức STP (Spanning-Tree Protocol) dùng để làm gì?

- A. Dùng để cập nhật định tuyến trong môi trường Switch.
- B. Dùng để chống "routing loop" trong mạng
- C. Dùng để tránh "switching loop" trong mạng
- D. Dùng để quản lý việc thêm, xóa, sửa thông tin VLAN trong hệ thống có nhiều Switch.
- E. Dùng để phân hoạch mạng thành nhiều miền đưng độ

10.5 Để kiểm tra interface fa0/5 có được gán cho VLAN Sales không, thì ta sử dụng lệnh nào sau đây?

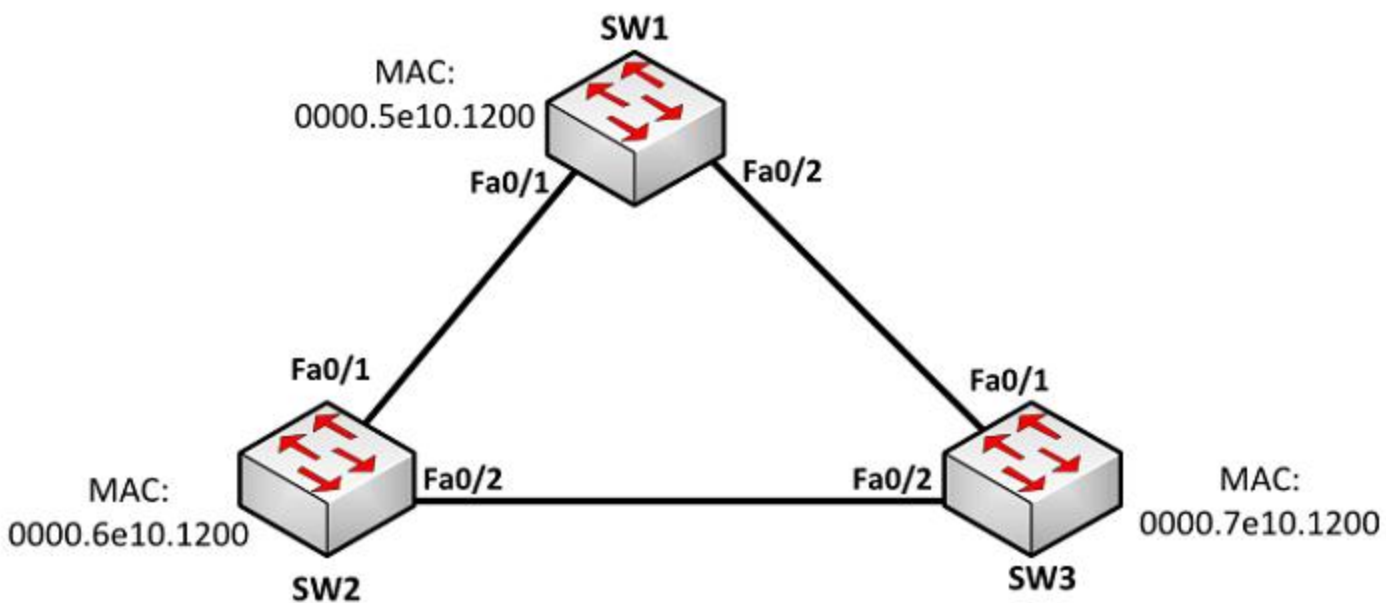
- A. show vlan
- B. show mac-address-table

- C. show vtp status
- D. show spanning-tree root
- E. show ip interface brief

10.6 Tại sao Switch không bao giờ học một địa chỉ “broadcast”?

- A. Frame broadcast không bao giờ được gửi tới Switch
- B. Địa chỉ broadcast sử dụng định dạng không đúng trong bảng chuyển mạch trên Switch
- C. Địa chỉ broadcast không bao giờ là địa chỉ nguồn trong một frame.
- D. Địa chỉ broadcast chỉ dùng trong layer 3
- E. Switch không bao giờ chuyển tiếp các gói tin broadcast

10.7 Cho mô hình mạng:



Tất cả các switch được cấu hình STP mặc định và tất cả các kết nối qua port FastEthernet. Port nào sẽ chuyển vào trạng thái "blocking"?

- A. Switch SW1 - Port Fa0/1
- B. Switch SW1 - Port Fa0/2
- C. Switch SW2 - Port Fa0/2
- D. Switch SW2 - Port Fa0/1
- E. Switch SW3 - Port Fa0/1
- F. Switch SW3 - Port Fa0/2

10.8 Cho mô hình mạng:

Router:

Interface fa0/1.1

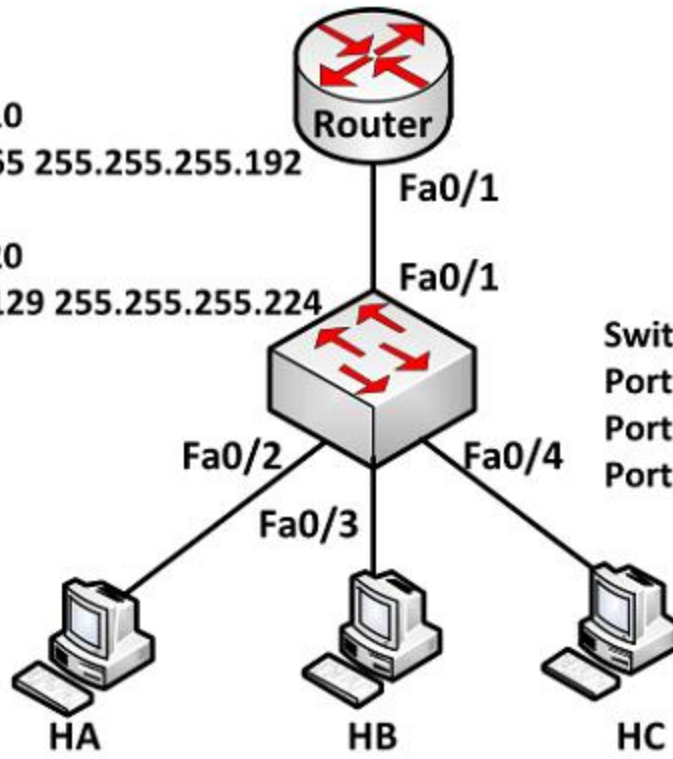
encapsulation dot1q 10

ip address 192.168.1.65 255.255.255.192

Interface fa0/1.2

encapsulation dot1q 20

ip address 192.168.1.129 255.255.255.224



Switch:

Port Fa0/1: trunk

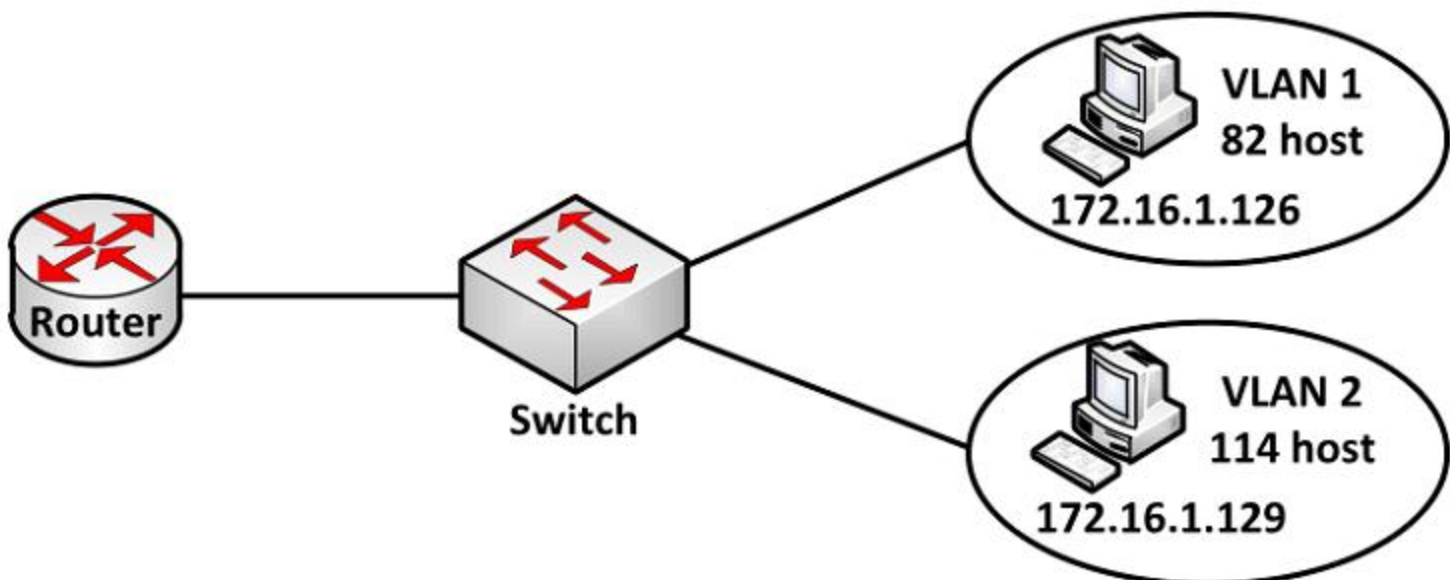
Port Fa0/2, Fa0/3: VLAN 10

Port Fa0/4: VLAN 20

Những thông tin cấu hình nào sau đây là đúng cho các host trong mô hình trên?

- A. Địa chỉ IP của HA: 192.1.1.65
- B. Subnet mask của HA: 255.255.255.224
- C. Địa chỉ IP của HB: 192.1.1.125
- D. Default gateway của HB: 192.1.1.65
- E. Địa chỉ IP của HC: 192.1.1.66
- F. Subnet mask của HC: 255.255.255.224

10.9 Cho mô hình mạng:

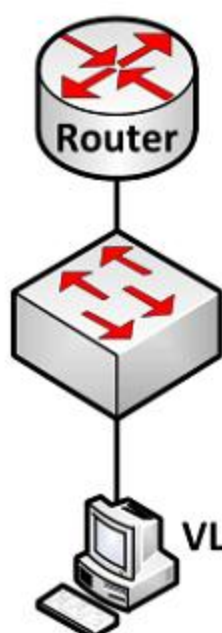


Những phát biểu nào sau đây là đúng trong mô hình mạng trên?

- A. Subnet mask được sử dụng là 255.255.255.192
- B. Subnet mask được sử dụng là 255.255.255.128

- C. Địa chỉ IP 172.16.1.25 có thể được gán cho các host thuộc VLAN1
- D. Địa chỉ IP 172.16.1.205 có thể được gán cho các host thuộc VLAN1
- E. Cổng LAN trên router được cấu hình với một địa chỉ IP
- F. Cổng LAN trên router được cấu hình với nhiều địa chỉ IP

10.10 Cho mô hình mạng:



```
R(config)#interface fastethernet 0/1.1
R(config-if)#encapsulation dot1q 1
R(config-if)#ip address 192.168.1.1
255.255.255.0
```

```
R(config)#interface fastethernet 0/1.2
R(config-if)#encapsulation dot1q 2
R(config-if)#Ip address 192.168.2.1
255.255.255.0
```

```
R(config)#interface fastethernet 0/1.3
R(config-if)#encapsulation dot1q 3
R(config-if)# Ip address 192.168.3.1
255.255.255.0
```

Router trong mô hình mạng được cấu hình như trên. Switch kết nối với router qua đường *trunk*. Trên Switch cấu hình 3 VLAN: VLAN1, VLAN2, and VLAN3. Một máy tính A kết nối vào VLAN2. Hỏi địa chỉ **default gateway** phải đặt cho máy tính này là địa chỉ nào sau đây?

- A. 192.168.1.1
- B. 192.168.1.2
- C. 192.168.2.1
- D. 192.168.2.2
- E. 192.168.3.1
- F. 192.168.3.2

10.11 Hai tham số được STP sử dụng để bầu chọn “root bridge”?

- A. Bridge priority
- B. Địa chỉ IP
- C. Địa chỉ MAC
- D. Phiên bản IOS
- E. Dung lượng RAM
- F. Tốc độ kết nối

Chương 3

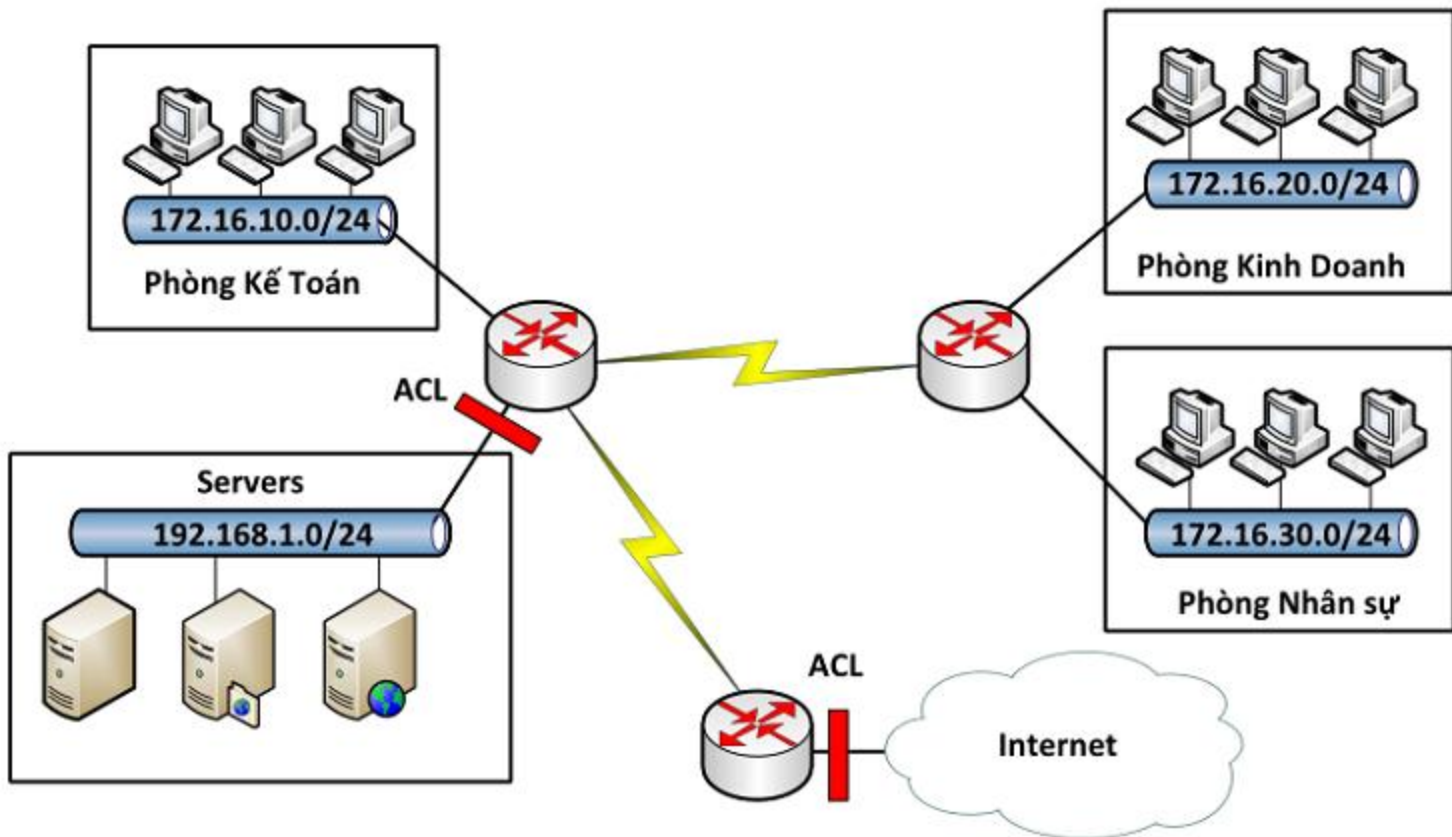
ACL

Chương này trình bày chức năng và đặc điểm của việc sử dụng ACL trong hệ thống mạng để điều khiển các truy cập, đặc điểm của các loại ACL và cách thức cấu hình trên thiết bị Cisco. Học xong chương này, người học có khả năng:

- Xác định được vai trò của ACL trong hệ thống mạng.
- Phân biệt và cấu hình được “Standard ACL” và “Extended ACL”.

1. GIỚI THIỆU

ACL là một danh sách các điều kiện được áp đặt vào các cổng của router để lọc các gói tin đi qua nó. Danh sách này chỉ ra cho router biết loại dữ liệu nào được cho phép (allow) và loại dữ liệu nào bị hủy bỏ (deny). Sự cho phép và hủy bỏ này có thể được kiểm tra dựa vào địa chỉ nguồn, địa chỉ đích, giao thức hoặc chỉ số cổng.



Hình 3.1 ACL trong mô hình mạng

Sử dụng ACL để quản lý các lưu lượng mạng, hỗ trợ ở mức độ cơ bản về bảo mật cho các truy cập mạng, thể hiện ở tính năng lọc các gói tin qua router.

2. PHÂN LOẠI VÀ HOẠT ĐỘNG CỦA ACL

❖ ACL được chia thành 2 loại:

- Standard ACL
- Extended ACL

❖ Hoạt động của ACL

ACL thực hiện việc kiểm tra theo trình tự của các điều kiện trong danh sách cấu hình. Nếu có một điều kiện được khớp trong danh sách thì nó sẽ thực hiện hành động tương ứng trong điều kiện đó, và các điều kiện còn lại sẽ không được kiểm tra nữa. Trường hợp tất cả các điều kiện trong danh sách đều không khớp thì một câu lệnh mặc định “deny any” được thực hiện, có nghĩa là điều kiện cuối cùng ngầm định trong một ACL mặc định sẽ là cấm tất cả. Vì vậy, trong cấu hình ACL cần phải có ít nhất một câu lệnh có hành động là “*permit*”.

Khi gói tin đi vào một cổng, router sẽ kiểm tra xem có ACL nào được đặt trên cổng để kiểm tra hay không, nếu có thì các gói tin sẽ được kiểm tra với những điều kiện trong danh sách. Nếu gói tin đó được cho phép bởi ACL, nó sẽ tiếp tục được kiểm tra trong bảng định tuyến để quyết định chọn cổng ra để đi đến đích.

Tiếp đó, router sẽ kiểm tra xem trên cổng dữ liệu chuyển ra có đặt ACL hay không. Nếu không thì gói tin đó có thể sẽ được gửi tới mạng đích. Nếu có ACL thì nó sẽ kiểm tra với những điều kiện trong danh sách ACL đó.

3. CẤU HÌNH ACL

Có 2 phương pháp cấu hình ACL:

- Dựa vào số (numbered ACL)
- Dựa vào tên (named ACL)

Tổng quát: để cài đặt một ACL, ta thực hiện các bước sau:

Bước 1: Tạo ACL

- ✓ Xác định loại ACL dựa vào số hiệu ACL (numbered ACL) hoặc tên (named ACL)
- ✓ Lựa chọn hành động cho từng điều kiện “*permit*” hay “*deny*” theo yêu cầu cụ thể

Bước 2: Gán ACL vào cổng của router

- ✓ Các ACL được gán vào một hoặc nhiều cổng và có thể được lọc theo chiều các gói tin đi vào hay đi ra.
- ✓ Một router với một ACL được đặt ở cổng dữ liệu vào phải kiểm tra mỗi gói tin để tìm xem nó có khớp các điều kiện trong danh sách ACL trước khi chuyển gói tin đó đến một cổng ra.

❖ Một số thuật ngữ

• Wildcard mask

“Wildcard mask” có 32 bit, chia thành 4 phần, mỗi phần có 8 bit, là tham số được dùng xác định các bit nào sẽ được bỏ qua hay buộc phải so trùng trong việc kiểm tra điều kiện. Bit ‘1’ trong “wildcard mask” có nghĩa là bỏ qua vị trí bit đó khi so sánh, và bit ‘0’ xác định vị trí bit đó phải giống nhau.

Với Standard ACL, nếu không thêm “wildcard-mask” trong câu lệnh tạo ACL thì mặc định “wildcard-mask” sẽ là 0.0.0.0

Mặc dù “Wildcard mask” có cấu trúc 32 bit giống với “Subnet mask” nhưng chúng hoạt động khác nhau. Các bit 0 và 1 trong một “Subnet mask” xác định phần “Network” và phần “Host” trong một địa chỉ IP. Các bit 0 và 1 trong một “wildcard-mask” xác định bit nào sẽ được kiểm tra hay bỏ qua cho mục đích điều khiển truy cập.

• Wildcard “host”

- ✓ “Wildcard mask” dùng cho một thiết bị hay còn gọi là “wildcard-host” có dạng: 0.0.0.0 (kiểm tra tất cả các bit)

Ví dụ: 172.30.16.29 0.0.0.0

- ✓ Ý nghĩa: khi kiểm tra ACL, nó sẽ kiểm tra tất cả các bit trong địa chỉ dùng để so khớp.
- ✓ “Wildcard mask” cho một thiết bị có thể được đại diện bằng từ khóa “host”

Ví dụ: host 172.30.26.29

Câu lệnh ACL cho phép một thiết bị như sau:

```
R(config)#access-list 1 permit 172.30.16.29 0.0.0.0
```

hoặc:

```
R(config)#access-list 1 permit host 172.30.16.29
```

- **Wildcard “any”**

- ✓ Wildcard mask cho tất cả các thiết bị được gọi là wildcard “any” có dạng: 255.255.255.255 (không kiểm tra tất cả các bit)
- ✓ Ý nghĩa: chấp nhận tất cả các địa chỉ
- ✓ “Wildcard mask” dùng cho tất cả các thiết bị có thể đại diện bằng từ khoá “any”

Ví dụ:

```
R(config)#access-list 1 permit 0.0.0.0 255.255.255.255
```

hoặc:

```
R(config)#access-list 1 permit any
```

- **Inbound và outbound**

Khi áp dụng ACL trên một cổng, phải xác định ACL đó được dùng cho luồng dữ liệu vào (inbound) hay ra (outbound). Chiều của luồng dữ liệu được xác định trên cổng của router.



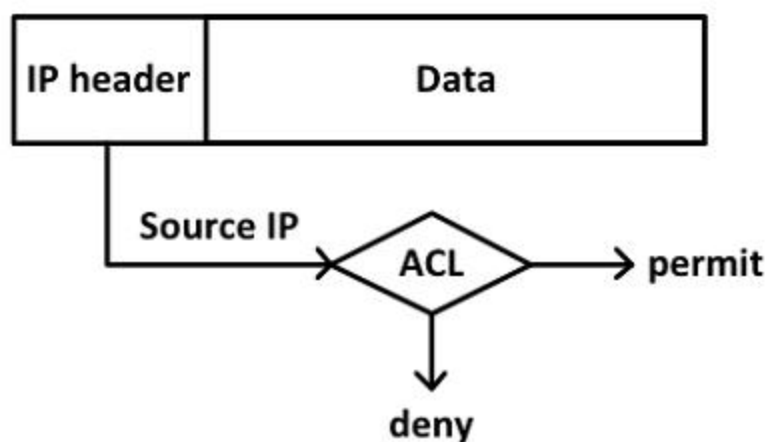
Hình 3.2 Hướng của các luồng dữ liệu

4. STANDARD ACL

Sử dụng “Standard ACL” khi ta muốn cấm hay cho phép tất cả các luồng dữ liệu từ một thiết bị hay một mạng xác định trên toàn bộ giao thức.

“Standard ACL” kiểm tra điều kiện dựa vào địa chỉ nguồn trong các gói tin và thực hiện hành động cấm hoặc cho phép tất cả các lưu lượng từ một thiết bị hay một mạng xác định nào đó.

Kiểm tra gói tin với “Standard ACL”:



Hình 3.3 Gói tin được kiểm tra bởi ACL

❖ Cấu hình Standard ACL

- Router(config)# **access-list** <ACL-number> {**permit**|**deny**} source [wildcast-mask]

Trong đó: *ACL-number*: có giá trị từ 1 đến 99, hoặc 1300-1999

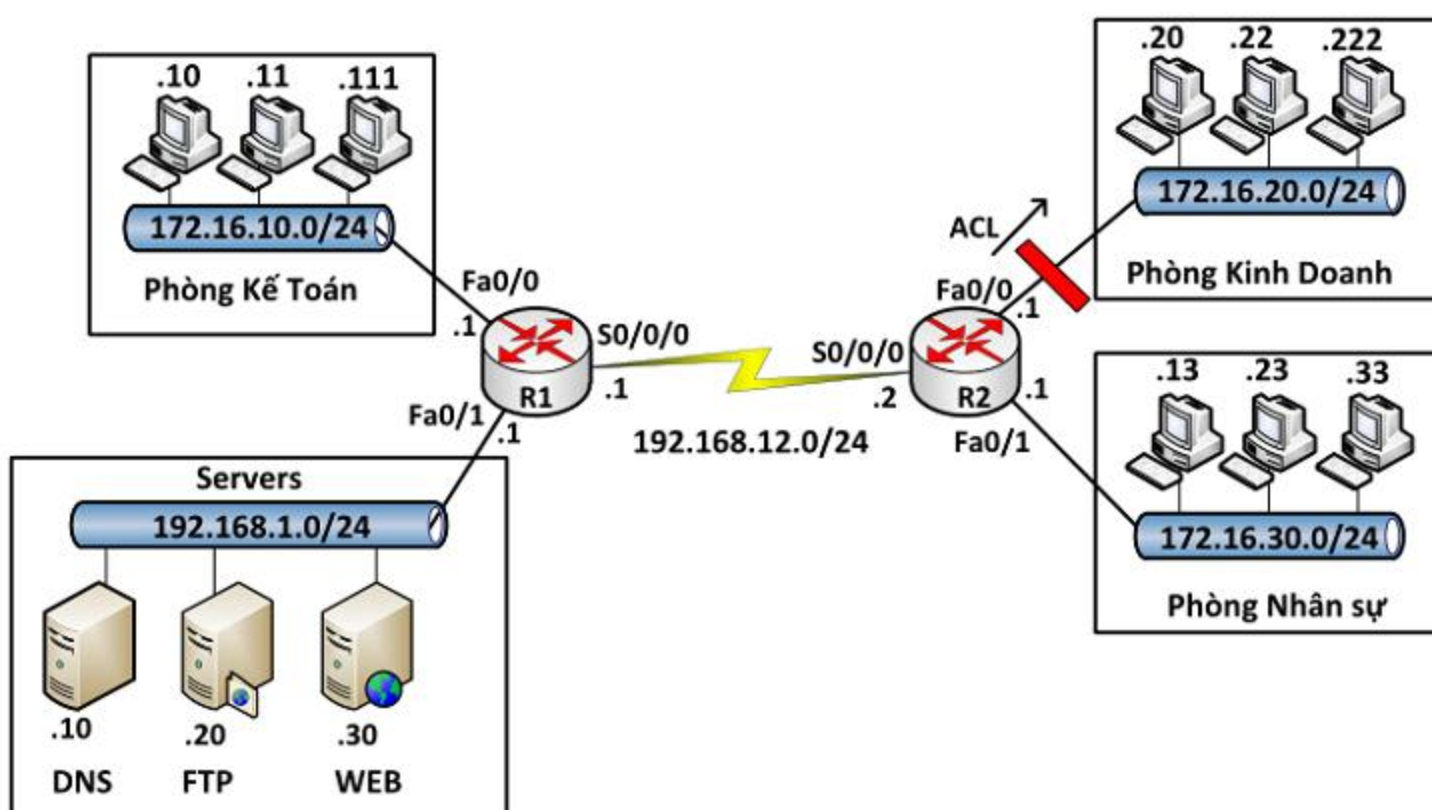
Wildcast-mask: nếu không được cấu hình sẽ lấy giá trị mặc định là:0.0.0.0

- Router(config-if)#**ip access-group** <ACL-number> {**in**|**out**}

Câu lệnh này có tác dụng gán ACL vào một cổng và đặt chế độ kiểm tra cho luồng dữ liệu đi vào hay đi ra khỏi cổng của router.

Dùng lệnh **no ip access-group** <ACL-number> để không áp đặt ACL vào cổng. Có nghĩa là hủy bỏ câu lệnh trên.

Ví dụ 1: Cấm các máy tính thuộc mạng 172.16.10.0/24 truy cập tới mạng 172.16.20.0/24.

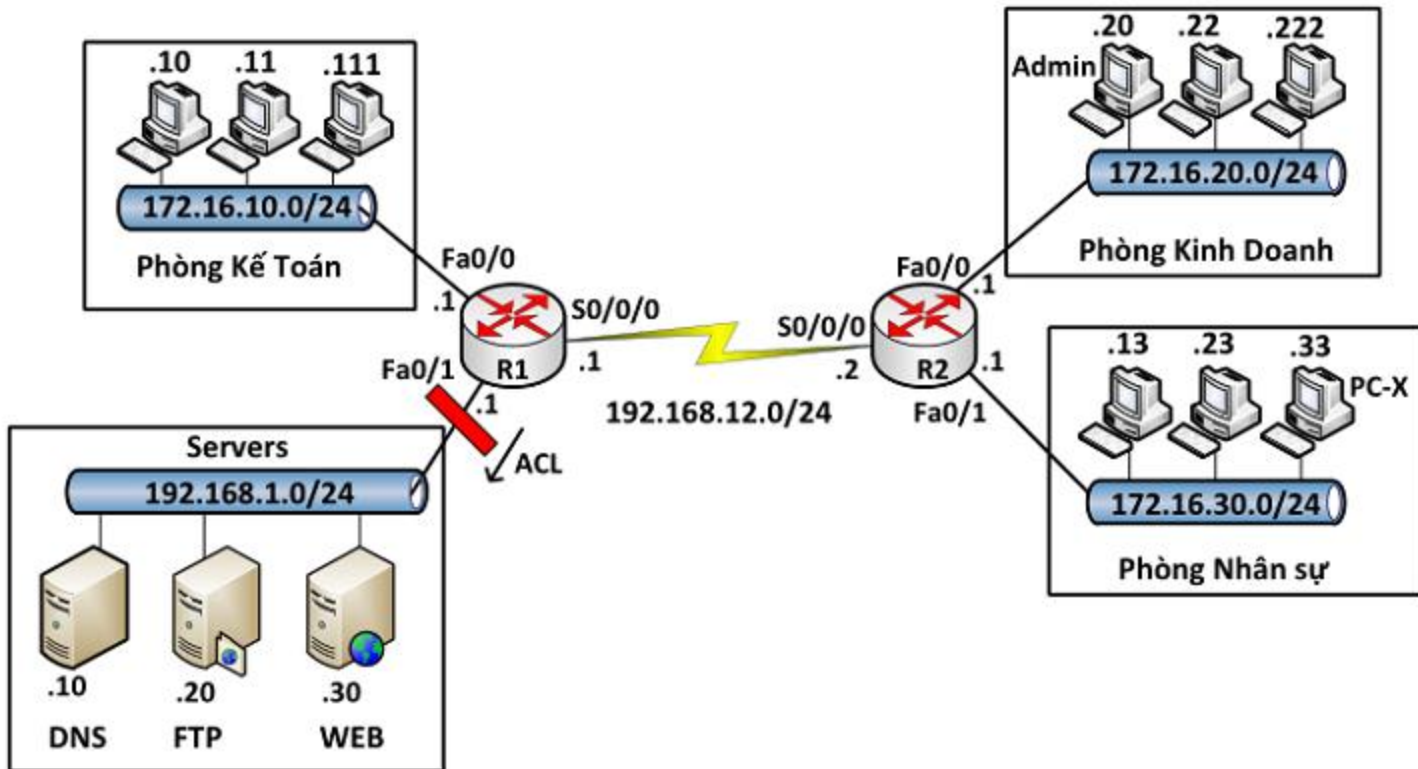


```

R2(config)#access-list 1 deny 172.16.10.0 0.0.0.255
R2(config)#access-list 1 permit any
R2(config)#interface fa0/0
R2(config-if)#ip access-group 1 out

```

Ví dụ 2: Cấm PC-X có địa chỉ 172.16.30.33/24 truy cập vào mạng 192.168.1.0/24



```

R1(config)# access-list 10 deny host 172.16.30.33
R1(config)# access-list 10 permit any
R1(config)#interface fa0/1
R1(config-if)#ip access-group 10 out

```

Ví dụ 3. Sử dụng lại mô hình trong ví dụ 2, viết ACL chỉ cho phép máy Admin có IP 172.16.20.20 telnet vào các router R1, R2.

Hướng dẫn cấu hình: trước tiên, cấu hình mở telnet trên R1 và R2.

ACL thực hiện yêu cầu đầu bài: trên R1 và R2 sử dụng ACL sau

```

R(config)#access-list 20 permit host 172.16.20.20
R(config)#line vty 0 4
R(config-line)#access-class 20 in

```


❖ Dùng “Standard ACL” để điều khiển telnet

Trên router có các “virtual terminal port” được dùng để cấu hình cho mục đích cho phép telnet vào router. Telnet cũng là một cách thức cho phép người quản trị cấu hình hay theo dõi thiết bị từ xa. Ta có thể lọc các địa chỉ truy xuất vào các cổng này bằng “Standard ACL”.

Cấu hình: thực hiện hai bước chính sau

- Chọn các thiết bị hoặc mạng được phép telnet vào các thiết bị dùng *Standard ACL*
- Gán ACL đã được cài đặt ở trên vào cổng telnet.

- Các câu lệnh cấu hình:

```
Router(config)#line vty {vty-number|vty-range}  
Router(config-line)#access-class <access-list-number> {in|out}
```

Trong đó:

vty-number: có giá trị 0 đến 4 (mặc định trên Router), có giá trị 0 đến 15 (mặc định trên Switch)

vty-range: là một dãy liên tiếp các port vty được sử dụng. Trong cấu hình ta sẽ cấu hình như sau: `line vty start-number end-number`

access-list-number: ACL gán vào các cổng **vty** để điều khiển truy cập

Ví dụ:

```
access-list 12 permit 192.168.1.0  
0.0.0.255  
(implicit deny all)  
!  
line vty 0 4  
access-class 12 in
```

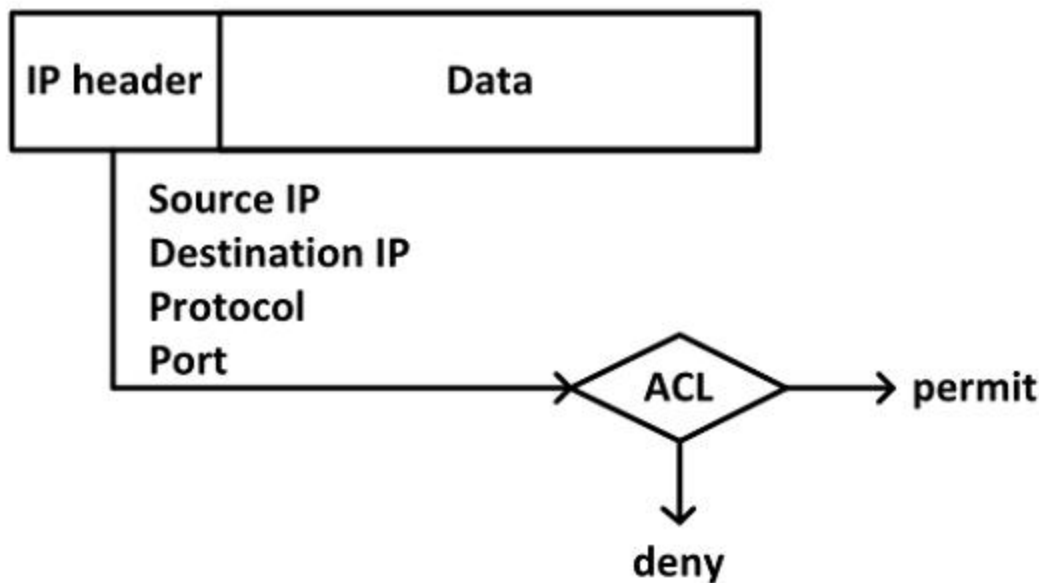
Các câu lệnh cấu hình trên có nghĩa là: chỉ cho phép các thiết bị thuộc mạng 192.168.1.0/24 có thể kết nối vào router thông qua telnet.

5. EXTENDED ACL

“Extended ACL” cung cấp sự điều khiển linh hoạt hơn “Standard ACL”. Nó kiểm tra cả địa chỉ nguồn, địa chỉ đích, giao thức, chỉ số cổng ứng

dụng. “Extended ACL” thực hiện hành động cấm hay cho phép ở một số ứng dụng xác định.

Kiểm tra các gói tin với “Extended ACL”:



Hình 3.4 Gói tin được kiểm tra bởi “Extended ACL”

❖ Cấu hình Extended ACL

- Router(config)#**access-list** <access-list-number>
{**permit**|**deny**} <protocol> <source-address>
<source-wildcard> <destination-address>
<destination-wildcard> <operation> <operand>

Trong đó: *access-list-number*: có giá trị từ 100 – 199 hoặc 2000 - 2699

protocol: là ip, udp, tcp, icmp,...

operator: thường dùng là **eq**

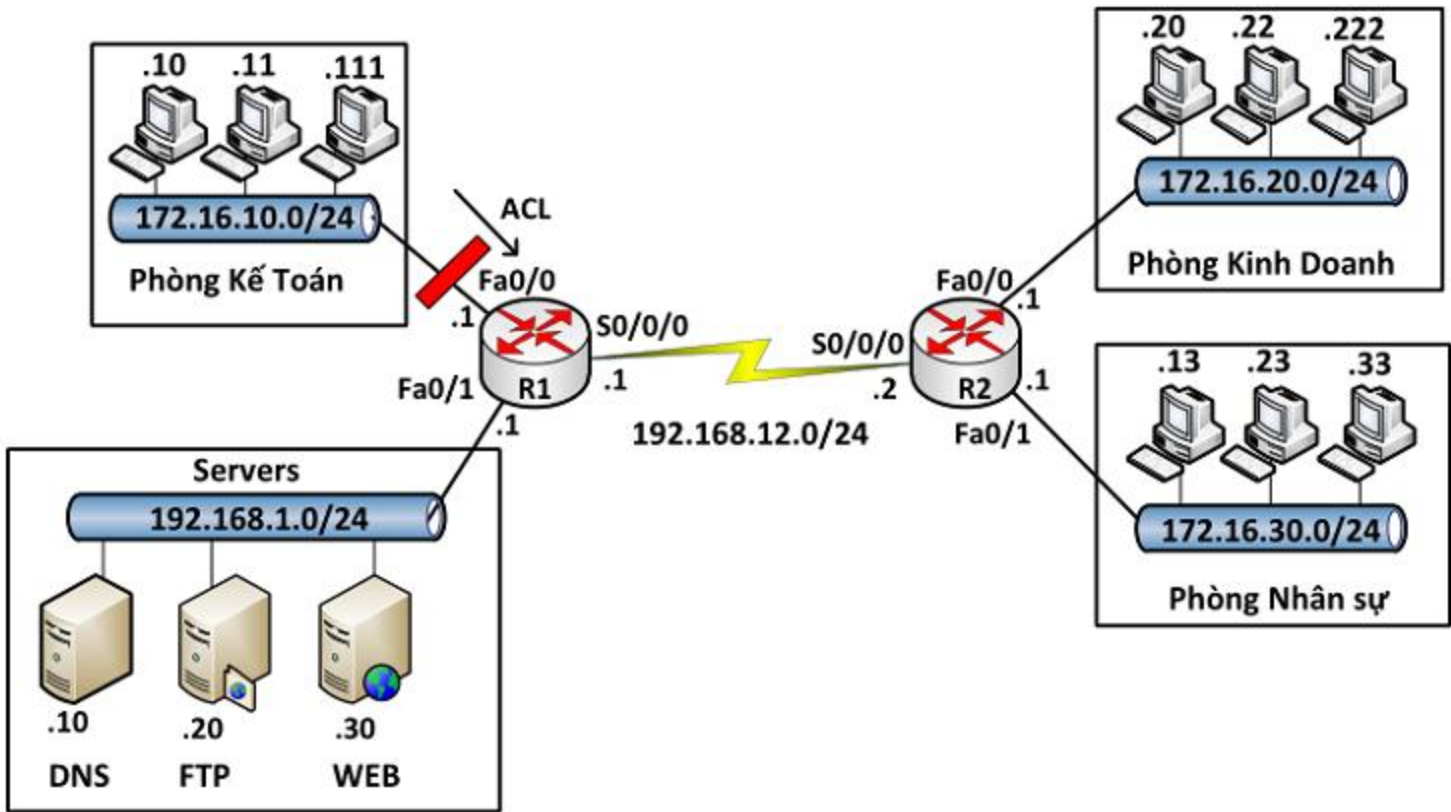
operand: là chỉ số port của dịch vụ hay tên của dịch vụ. Ví dụ: ta có thể dùng chỉ số port **23** hay có thể dùng tên dịch vụ là **telnet**

Câu lệnh trên được dùng để tạo một điều kiện (ACL entry) trong một ACL *access-list-number*

- Router(config-if)#**ip access-group** *access-list-number* {**in**|**out**}

Trong đó, *access-list-number* là số hiệu (có giá trị 100 – 199 hoặc 2000 - 2699) chỉ danh sách ACL ta đã tạo. Câu lệnh này có ý nghĩa là gán danh sách ACL vào interface và chọn hướng (*inbound* hoặc *outbound*) các traffic sẽ được kiểm tra

Ví dụ 1: Cấu hình trên router trong mô hình mạng dưới đây để cấm các FTP traffic từ các host thuộc subnet 172.16.10.0 đến FTP server có IP 192.168.1.20/24, cho phép tất cả các traffic còn lại hoạt động bình thường.



```
R1(config)#access-list 100 deny tcp 172.16.10.0
0.0.0.255 host 192.168.1.20 eq 20
R1(config)#access-list 100 deny tcp 172.16.10.0
0.0.0.255 host 192.168.1.20 eq 21
R1(config)#access-list 100 permit ip any any
R1(config)#interface fa0/0
R1(config-if)#ip access-group 100 in
```

- **Vị trí đặt ACL**

Nên đặt *extended ACL* gần nguồn của traffic muốn cấm và nên đặt *Standard ACL* gần đích đến của traffic.

6. NAMED ACL

Named-ACL cho phép *Standard* và *Extended ACL* được định danh bởi một tên thay vì đại diện bởi một con số. Loại ACL này có thể cho phép xóa một số dòng (điều kiện) trong một danh sách đã được cấu hình.

Named-ACL không tương thích với các Cisco IOS phiên bản trước 11.2 và không thể sử dụng cùng một tên cho nhiều ACL. ACL của các loại giao thức khác nhau không thể có cùng một tên.

- **Các câu lệnh cấu hình Name ACL**

```
Router(config)#ip access-list {standard | extended}
name
```

```
Router(config{std-|ext-}nacl)#[sequence-number]
{permit|deny} {ip access list test conditions}

Router(config-if)#ip access-group name {in | out}
```

❖ **Một số lệnh kiểm tra cấu hình ACL**

```
Router#show access-list {access-list-number |
name}
```

Sau đây một ví dụ về kết quả hiển thị của lệnh *show access-lists*

```
Router#show access-lists
```

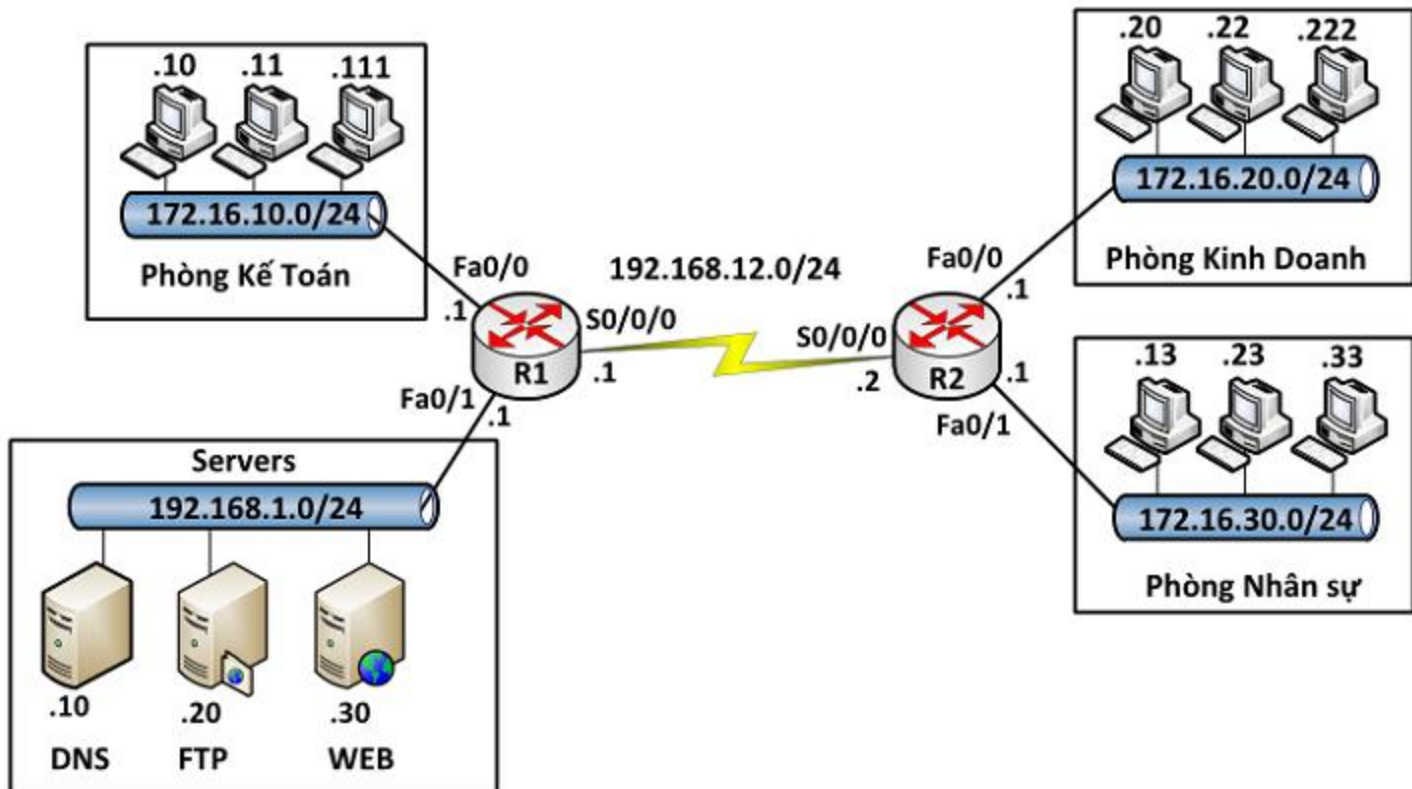
```
Standard IP access list 1
```

```
    permit 10.2.2.1
    permit 10.3.3.1
    permit 10.4.4.1
    permit 10.5.5.1
```

```
Extended IP access list 101
```

```
    permit tcp host 10.22.22.1 any eq telnet
    permit tcp host 10.33.33.1 any eq ftp
    permit tcp host 10.44.44.1 any eq ftp-data
```

Ví dụ:



❖ **Yêu cầu:**

- (1) Cấu hình standard ACL cấm các máy tính thuộc phòng Kinh doanh truy cập tới phòng Kế toán

- (2) Cấm các máy tính thuộc phòng Kế toán truy cập tới Web server bằng dịch vụ www
- (3) Cấm các máy tính thuộc phòng Nhân sự ping tới DNS server

❖ **Hướng dẫn cấu hình**

Bước 1: Cấu hình hostname, địa chỉ IP cho các cổng trên các thiết bị, cấu hình định tuyến cho hệ thống mạng trên với giao thức định tuyến tùy chọn.

Bước 2: Cấu hình ACL theo yêu cầu

- (1) Có thể dùng standard ACL và extended ACL cho yêu cầu này

- Dùng “Standard ACL”

```
R1(config)#ip access-list standard abc
R1(config-std-nacl)# deny 172.16.20.0 0.0.0.255
R1(config-std-nacl)# permit any
R1(config)#interface fa0/0
R1(config-if)#ip access-group abc out
```

- Dùng “Extended ACL” (có thể cấu hình trên R1 hoặc R2)

```
R2(config)#ip access-list extended xyz
R2(config-ext-nacl)#deny ip 172.16.20.0
0.0.0.255 172.16.10.0 0.0.0.255
R2(config-ext-nacl)#permit ip any any
R2(config)#interface fa0/0
R2(config-if)#ip access-group xyz in
```

- (2) Cấm các máy tính thuộc phòng Kế toán truy cập tới Web server bằng dịch vụ www

```
R1(config)#ip access-list extended spkt
R1(config-ext-nacl)#deny tcp 172.16.10.0
0.0.0.255 host 192.168.1.30 eq 80
R1(config-ext-nacl)#permit ip any any
R1(config)#interface fa0/1
R1(config-if)#ip access-group spkt out
```

- (3) Cấm các máy tính thuộc phòng Nhân sự ping tới DNS server

```
R2(config)#ip access-list extended cntt
```

```
R2(config-ext-nacl)#deny icmp 172.16.30.0
0.0.0.255 host 192.168.1.10

R2(config-ext-nacl)#permit ip any any

R2(config)#interface fa0/1

R2(config-if)#ip access-group cntt in
```

❖ Kiểm tra

Dùng lệnh *ping*, trình duyệt Web để kiểm tra kết quả, dùng các câu lệnh show trên router để kiểm tra cấu hình

```
show run
show ip route
show access-lists
```

7. TỔNG KẾT CHƯƠNG

ACL có thể xem như là một tường lửa nhỏ định ra một tập luật để chặn các truy cập bất hợp pháp được cấu hình trên các router.

ACL được chia làm hai loại: *standard ACL* và *Extended ACL*. Trong đó, *standard ACL* thường được đặt ở gần đích, còn *Extended ACL* thường đặt ở gần nguồn cần cấm luồng dữ liệu.

ACL hoạt động theo trình tự cấu hình được thiết lập, khi một điều kiện được so khớp thì các câu lệnh còn lại sẽ không được kiểm tra nữa và cuối danh sách luôn có câu lệnh mặc định là “*deny all*”.

8. CÂU HỎI VÀ BÀI TẬP

8.1 ACL sau đây được áp đặt vào cổng fa0/0 theo chiều outbound:

```
access-list 123 deny tcp 192.168.1.8 0.0.0.7 eq 20 any
access-list 123 deny tcp 192.168.1.9 0.0.0.7 eq 21 any
```

Cho biết ý nghĩa của ACL trên?

- A. Tất cả các gói tin sẽ được cho phép đi qua cổng fa0/0 trừ các gói tin FTP.
- B. Cấm các gói tin FTP xuất phát từ 192.168.1.22 đến bất kỳ đâu
- C. Cấm các gói tin FTP xuất phát từ 192.168.1.9 đến bất kỳ đâu
- D. Tất cả các gói tin đi qua cổng fa0/0 đều bị cấm.

E. Cấm các gói tin FTP từ bất kỳ đâu đến mạng 192.168.1.8/29

8.2 Standard ACL lọc các gói tin dựa vào thành phần nào trong gói tin?

- A. Dựa vào địa chỉ IP nguồn và IP đích
- B. Dựa vào chỉ số port đích
- C. Dựa vào địa chỉ IP nguồn
- D. Tất cả các câu trên

8.3 ACL nào sau đây được sử dụng để cấm telnet xuất phát từ mạng 210.93.105.0/24 đến mạng 223.8.151.0/24?.

- A.

```
access-list one deny 210.93.105.0 0.0.0.0 any eq 23
access-list one permit any
```
- B.

```
access-list 100 deny tcp 210.93.105.0 0.0.0.255
223.8.151.0 0.0.0.255 eq 23
```
- C.

```
access-list 100 deny ip 223.8.151.0 0.0.0.255
any 23
access-list 100 permit ip any any
```
- D.

```
access-list 100 deny tcp 210.93.105.0 0.0.0.255
223.8.151.0 0.0.0.255 eq telnet
access-list 100 permit ip any any
```

8.4 Câu nào sau đây là “Standard ACL”?.

- A.

```
access-list 10 permit 192.168.1.0 0.0.0.255
```
- B.

```
access-list 100 deny host 192.168.1.100
```
- C.

```
access-list 101 permit ip any 192.168.1.0
0.0.0.255
```
- D.

```
access-list 10 permit tcp 192.168.1.0 0.0.0.255
any
```

8.5 Công ty XYZ sử dụng *Subnet mask /29*. *Wildcard mask* được sử dụng để cấu hình ACL để *permit* hay *deny* truy cập cho mạng này?

- A. 255.255.255.224
- B. 255.255.255.248
- C. 0.0.0.224
- D. 0.0.0.8

E. 0.0.0.7

F. 0.0.0.3

8.6 Một ACL được cấu hình như sau:

```
access-list 10 permit 172.29.16.0 0.0.0.255
```

```
access-list 10 permit 172.29.17.0 0.0.0.255
```

```
access-list 10 permit 172.29.18.0 0.0.0.255
```

```
access-list 10 permit 172.29.19.0 0.0.0.255
```

Lệnh nào sau đây có thể thay thế cho tất cả các lệnh trên?

A. Access-list 10 permit 172.29.16.0 0.0.0.255

B. Access-list 10 permit 172.29.16.0 0.0.1.255

C. Access-list 10 permit 172.29.16.0 0.0.3.255

D. Access-list 10 permit 172.29.16.0 0.0.15.255

E. Access-list 10 permit 172.29.0.0 0.0.255.255

8.7 ACL nào sau đây là ví dụ dùng để cấm các gói tin xuất phát từ một host cụ thể?

A. router(config)#access list 1 deny 17231.212.74

B. router(config)#access list 1 deny 10.6.111.48
host

C. router(config)#access list 1 deny 172.16.4.13
0.0.0.0

D. router(config)#access list 1 deny 192.168.14.132
255.255.255.0

E. router(config)#access list 1 deny
192.168.166.127 255.255.255.255

8.8 ACL nào sau đây được dùng để cấm tất cả các gói tin telnet đến mạng 10.10.1.0/24?

A. access-list 15 deny telnet any 10.10.1.0
0.0.0.255 eq 23

B. access0list 115 deny udp any 10.10.1.0 eq
telnet

- C. `access-list 15 deny tcp 10.10.1.0 255.255.255.0 eq telnet`
- D. `access-list 115 deny tcp any 10.10.1.0 0.0.0.255 eq 23`
- E. `access-list 15 deny udp any 10.10.1.0 255.255.255.0 eq 23`

8.9 ACL được cấu hình trong router như sau:

```

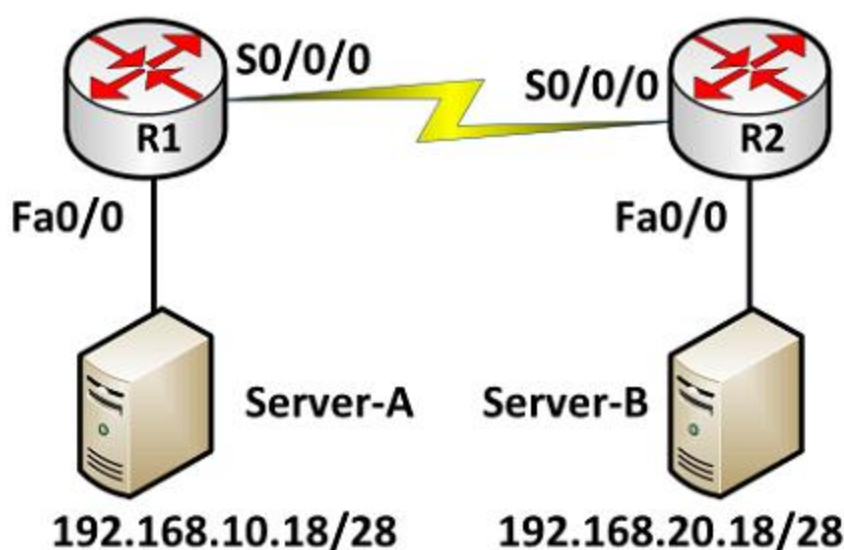
router#show access-lists
Extended IP access list 110
10 deny tcp 172.16.0.0 0.0.255.255 any eq telnet
20 deny tcp 172.16.0.0 0.0.255.255 any eq smtp
30 deny tcp 172.16.0.0 0.0.255.255 any eq http
40 permit tcp 172.16.0.0 0.0.255.255 any

```

Hãy cho biết router sẽ thực hiện hành động gì khi các gói tin HTTP từ Internet đến 172.16.12.10 nếu các gói HTTP này được ACL kiểm tra.

- A. Các gói tin này sẽ bị hủy bởi so khớp với điều kiện có số thứ tự 30
- B. Các gói tin này sẽ cho phép đi qua bởi so khớp với điều kiện có số thứ tự 40
- C. Các gói tin này sẽ bị hủy bởi vì lệnh ngầm định cấm tất cả ở cuối ACL
- D. Các gói tin này sẽ cho phép đi qua bởi vì địa chỉ nguồn không thuộc trong ACL

8.10 Cho mô hình mạng sau



Để điều khiển truy cập trong mạng, người quản trị tạo ACL như sau:

```
access-list 101 permit tcp 192.168.10.16 0.0.0.15  
192.168.20.16 0.0.0.15 eq 23
```

Cho biết ý nghĩa của ACL trên và nên đặt ACL này trên router nào, cổng nào và theo hướng nào.

- A. Cho phép các gói tin Telnet từ 192.168.1.16/28 đến 192.168.2.16/28.
- B. Cho phép các gói tin SMTP từ 192.168.1.16/28 đến 192.168.2.16/28.
- C. ACL cho phép các gói tin từ một host này đến một host khác.
- D. ACL nên đặt vào cổng fa0/0 trên Router R1 theo hướng inbound.
- E. ACL nên đặt vào cổng fa0/0 trên router R1 theo hướng outbound.

Chương 4

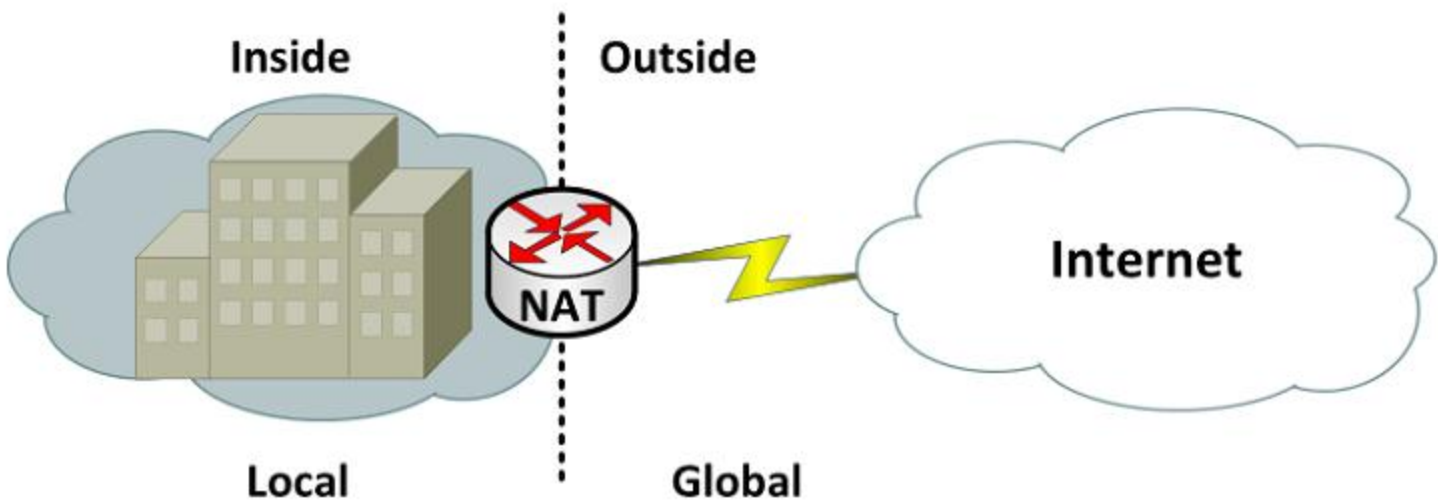
NAT

Chương này trình bày một số đặc điểm của NAT, phân loại và cấu hình trên thiết bị Cisco. Học xong chương này, người học có khả năng:

- Trình bày được một số khái niệm dùng trong kỹ thuật NAT
- Phân loại và trình bày được đặc điểm của mỗi loại NAT
- Cấu hình NAT

1. GIỚI THIỆU

NAT (*Network Address Translation*) là một kỹ thuật cho phép chuyển đổi từ một địa chỉ IP này thành một địa chỉ IP khác. Thông thường, NAT được dùng phổ biến trong mạng sử dụng địa chỉ cục bộ, cần truy cập đến mạng công cộng (Internet). Vị trí thực hiện NAT là router biên kết nối giữa hai mạng.

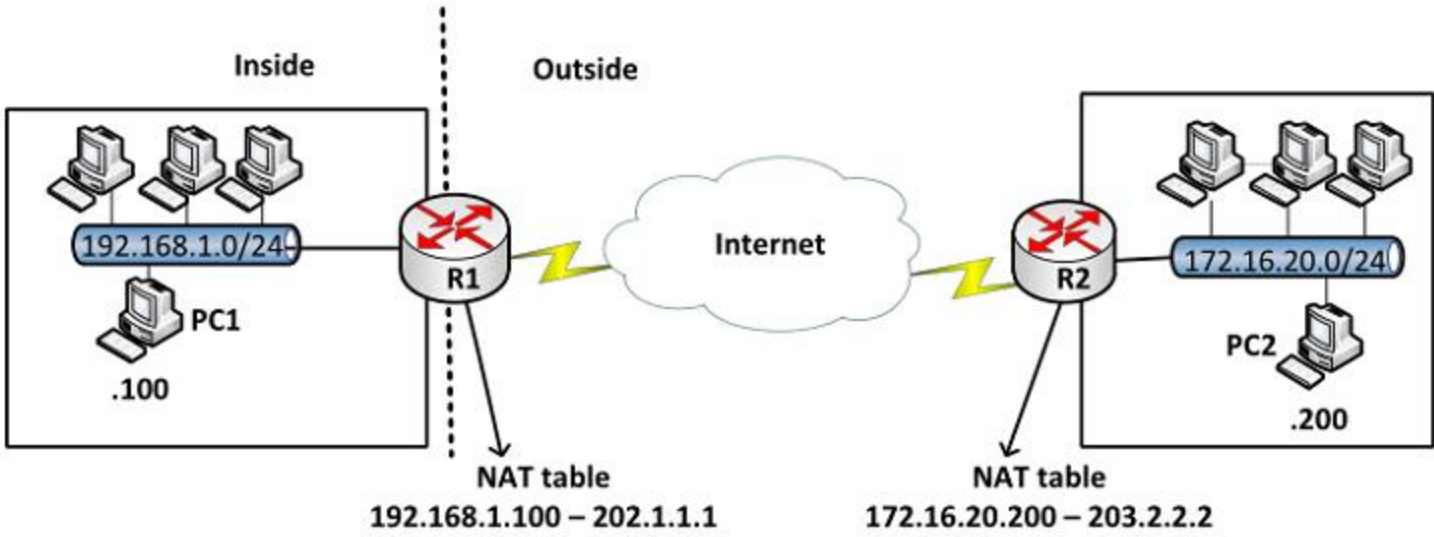


Hình 4.1 Mô hình thực hiện NAT

❖ Địa chỉ *private* và địa chỉ *public*

- Địa chỉ *private*: được định nghĩa trong RFC 1918
 - ✓ 10.0.0.0 – 10.255.255.255
 - ✓ 172.16.0.0 – 172.31.255.255
 - ✓ 192.168.0.0 – 192.168.255.255
- Địa chỉ *public*: các địa chỉ còn lại. Các địa chỉ *public* là các địa chỉ được cung cấp bởi các tổ chức có thẩm quyền.

❖ Một số thuật ngữ



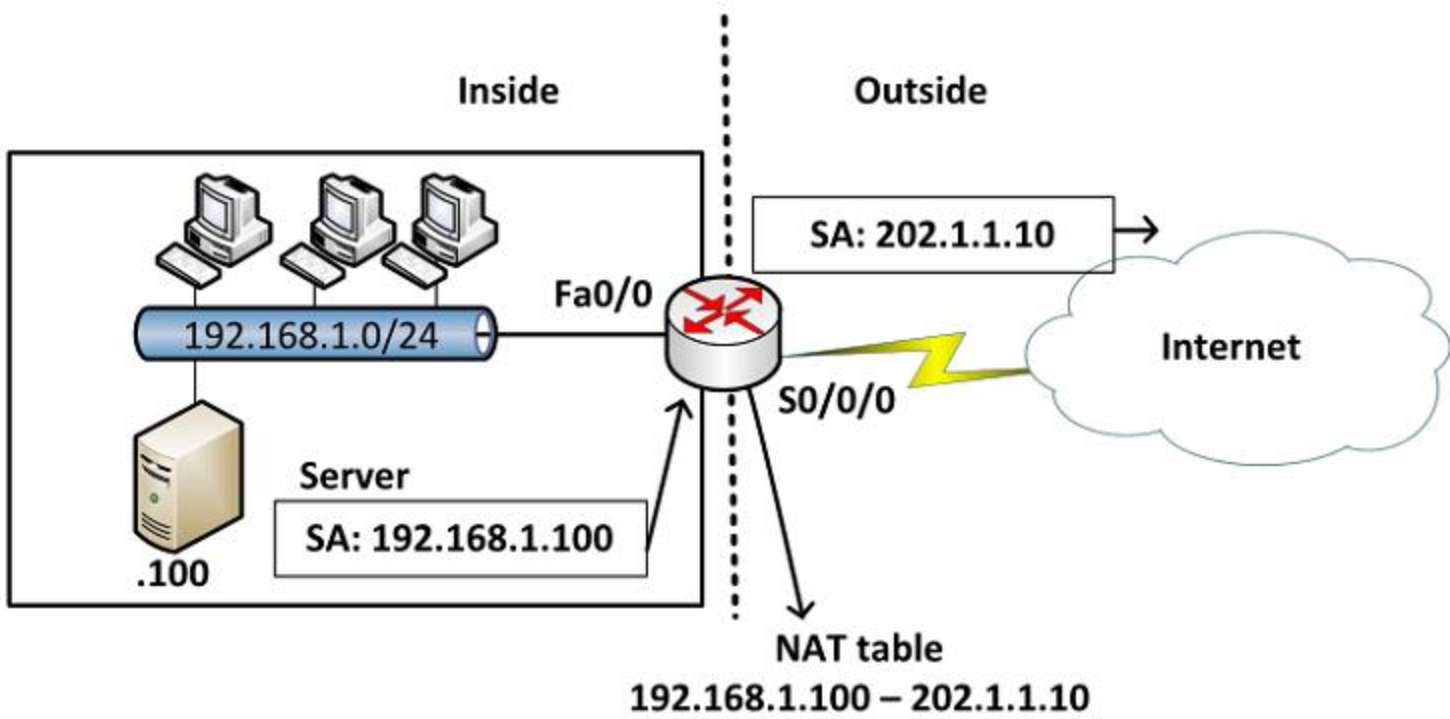
Hình 4.2 Địa chỉ inside và outside

- Địa chỉ **inside local**: là địa chỉ IP gán cho một thiết bị ở mạng bên trong. Địa chỉ này hầu như không phải địa chỉ được cấp bởi NIC (Network Information Center) hay nhà cung cấp dịch vụ.
- Địa chỉ **inside global**: là địa chỉ đã được đăng ký với NIC, dùng để thay thế một hay nhiều địa chỉ IP *inside local*.
- Địa chỉ **outside local**: là địa chỉ IP của một thiết bị bên ngoài khi nó xuất hiện bên trong mạng. Địa chỉ này không nhất thiết là địa chỉ được đăng ký, nó được lấy từ không gian địa chỉ bên trong.
- Địa chỉ **outside global**: là địa chỉ IP gán cho một thiết bị ở mạng bên ngoài. Địa chỉ này được lấy từ địa chỉ có thể dùng để định tuyến toàn cầu hay từ không gian địa chỉ mạng.

2. STATIC NAT

Static NAT được dùng để chuyển đổi một địa chỉ IP này sang một địa chỉ khác một cách cố định, thông thường là từ một địa chỉ cục bộ sang một địa chỉ công cộng và quá trình này được cài đặt thủ công, nghĩa là địa chỉ ánh xạ và địa chỉ được ánh xạ được chỉ định rõ ràng tương ứng duy nhất.

Static NAT rất hữu ích trong trường hợp những thiết bị cần phải có địa chỉ cố định để có thể truy cập từ bên ngoài Internet. Những thiết bị này phổ biến là những Server như Web, Mail,...



Hình 4.3 Chuyển dịch địa chỉ dạng tĩnh

❖ Cấu hình Static -NAT

- ✓ Thiết lập mối quan hệ chuyển đổi giữa địa chỉ nội bộ bên trong và địa chỉ đại diện bên ngoài.

```
Router(config)#ip nat inside source static
local-ip global-ip
```

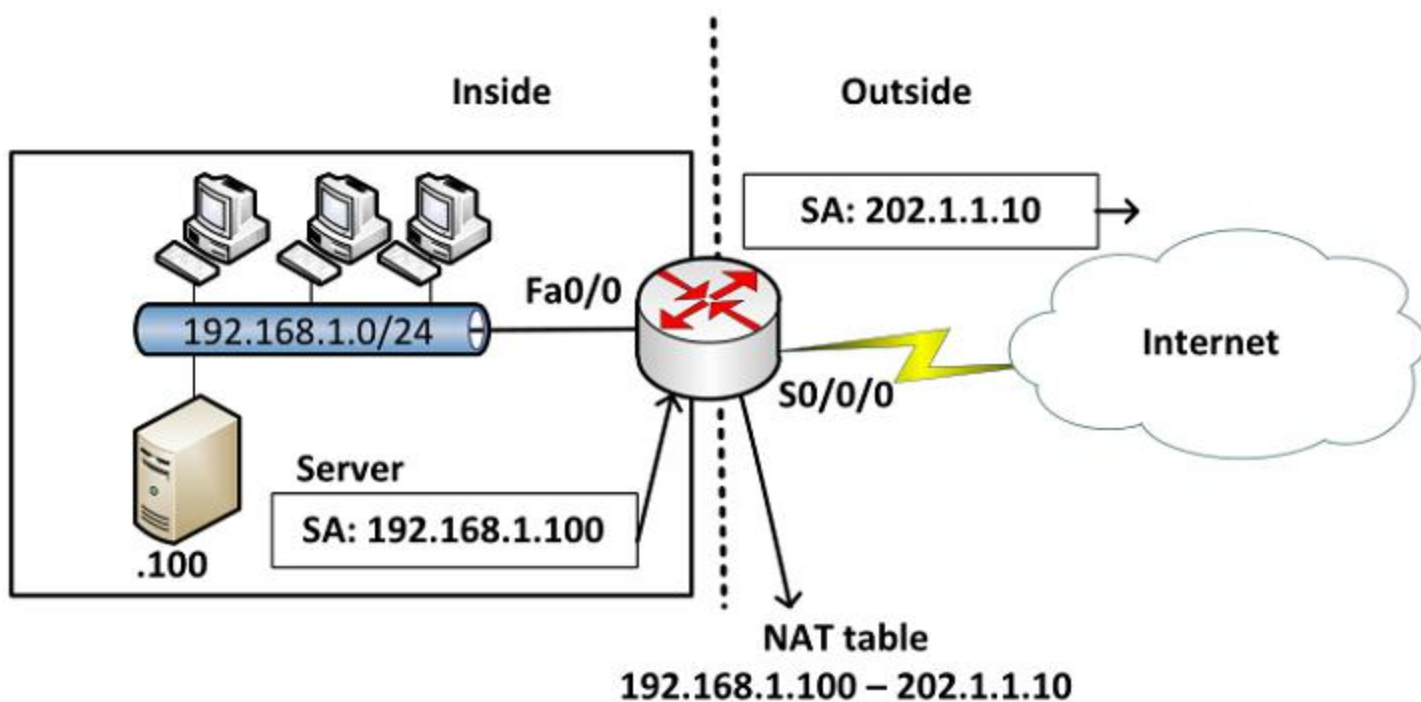
- ✓ Xác định các cổng kết nối vào mạng bên trong và thực hiện lệnh

```
Router(config-if)#ip nat inside
```

- ✓ Xác định các cổng kết nối ra mạng công cộng bên ngoài và thực hiện lệnh

```
Router(config-fi)#ip nat outside
```

Ví dụ:



```

Router(config)#ip nat inside source static
192.168.1.100 202.1.1.10

Router(config)#interface fa0/0

Router(config-if)#ip nat inside

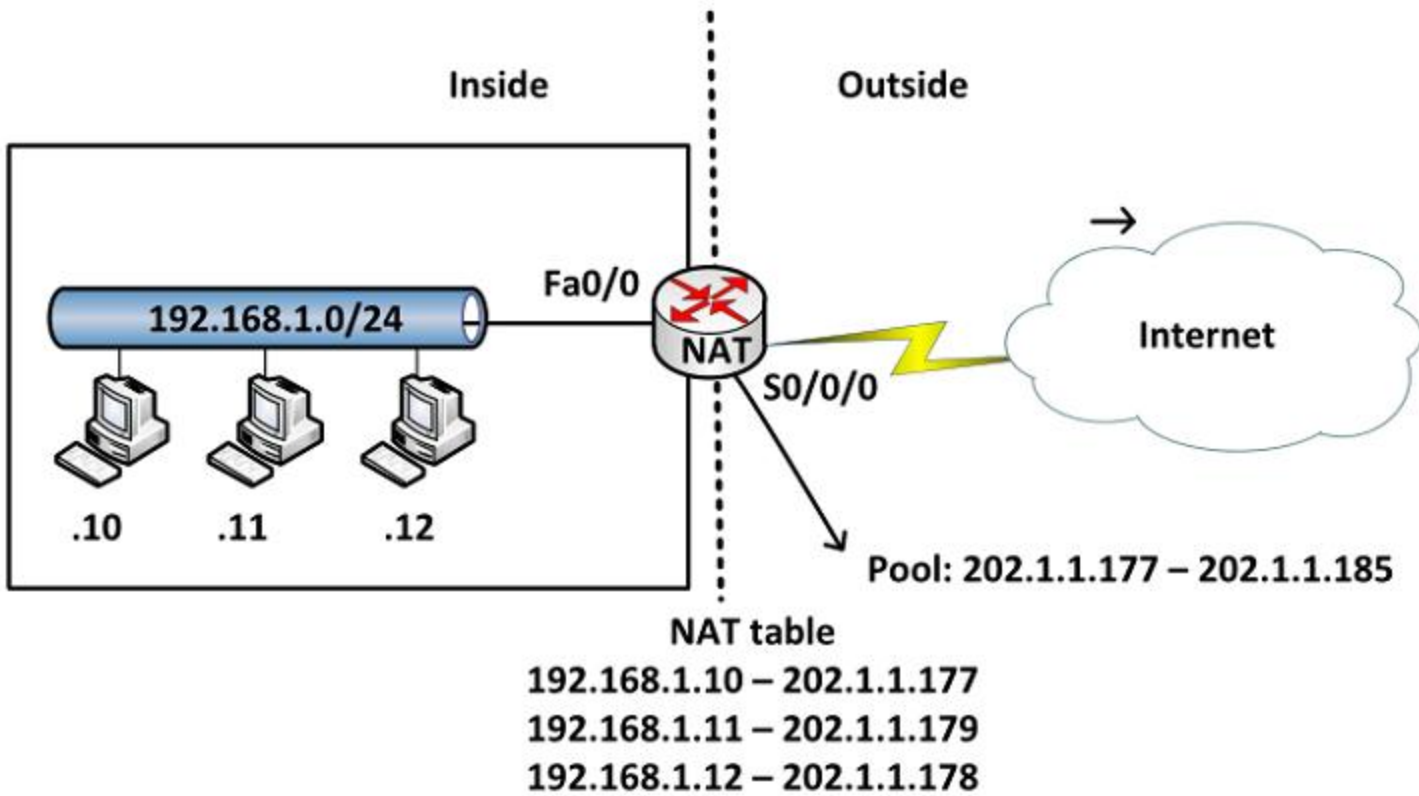
Router(config)#interface S0/0/0

Router(config-if)#ip nat outside

```

3. DYNAMIC NAT

Dynamic NAT được dùng để ánh xạ một địa chỉ IP này sang một địa chỉ khác một cách tự động, thông thường là ánh xạ từ một địa chỉ cục bộ sang một địa chỉ được đăng ký. Bất kỳ một địa chỉ IP nào nằm trong dải địa chỉ IP công cộng đã được định trước đều có thể được gán cho một thiết bị bên trong mạng.



Hình 4.4 Quá trình chuyển địa chỉ của gói tin trong mạng

❖ Cấu hình Dynamic NAT

- ✓ Xác định dải địa chỉ đại diện bên ngoài (public): các địa chỉ NAT
 Router(config)#**ip nat pool** name start-ip end-ip
 [netmask netmask/prefix-length prefix-length]
- ✓ Thiết lập ACL cho phép những địa chỉ nội bộ bên trong nào được chuyển đổi: các địa chỉ được NAT

```
Router(config)#access-list access-list-number  
permit source [source-wildcard]
```

- ✓ Thiết lập mối quan hệ giữa địa chỉ nguồn đã được xác định trong ACL với dải địa chỉ đại diện ra bên ngoài

```
Router(config)#ip nat inside source list <acl-number>  
pool <name>
```

- ✓ Xác định các cổng kết nối vào mạng nội bộ

```
Router(config-if)# ip nat inside
```

- ✓ Xác định các cổng kết nối ra bên ngoài

```
Router(config-if)#ip nat outside
```

Ví dụ: Cấu hình cho mô hình trong hình trên

```
Router(config)#ip nat pool abc 202.1.1.177  
202.1.1.185 netmask 255.255.255.0
```

```
Router(config)#access-list 1 permit 192.168.1.0  
0.0.0.255
```

```
Router(config)#ip nat inside source list 1 pool abc
```

```
Router(config)#interface fa0/0
```

```
Router(config-if)#ip nat inside
```

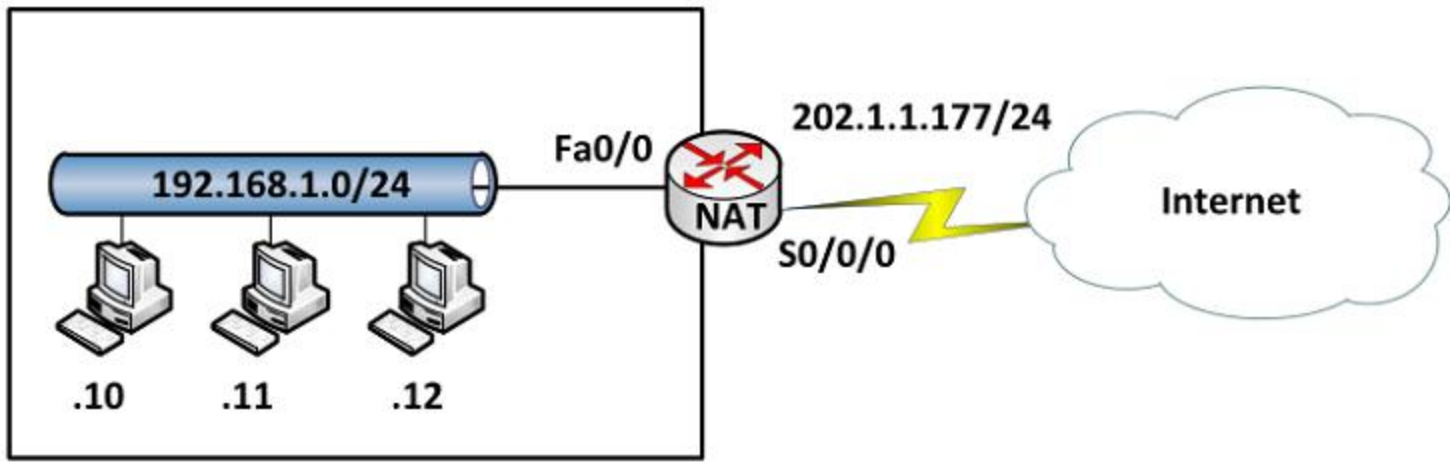
```
Router(config)#interface S0/0/0
```

```
Router(config-if)#ip nat outside
```

4. NAT OVERLOAD

NAT Overload là một dạng của *Dynamic NAT*, nó thực hiện ánh xạ nhiều địa chỉ IP thành một địa chỉ (many – to – one) và sử dụng các chỉ số cổng khác nhau để phân biệt cho từng chuyển đổi. NAT Overload còn có tên gọi là PAT (*Port Address Translation*).

Chỉ số cổng được mã hóa 16 bit, do đó có tới 65536 địa chỉ nội bộ có thể được chuyển đổi sang một địa chỉ công cộng.



NAT table
 192.168.1.10 – 202.1.1.177:1030
 192.168.1.11 – 202.1.1.177:1031
 192.168.1.12 – 202.1.1.177:1032

Hình 4.5 Chuyển dịch địa chỉ của “NAT Overload”

❖ Cấu hình NAT Overload

- Xác định dãy địa chỉ bên trong cần chuyển dịch ra ngoài (*private ip addresses range*)

```
Router(config)#access-list <ACL-number> permit
<source> <wildcard>
```

- Cấu hình chuyển đổi địa chỉ IP sang cổng nối ra ngoài

```
Router(config)#ip nat inside source list <ACL-
number> interface <interface> overload
```

- Xác định các cổng nối vào mạng bên trong và nối ra mạng bên ngoài

Đối với các cổng nối vào mạng bên trong:
 router(config-if)#**ip nat inside**

Đối với nối ra mạng bên ngoài:
 router(config-if)#**ip nat outside**

Ví dụ:

Giả sử hệ thống mạng công ty mô tả như sơ đồ trên, công ty thuê một đường kết nối Internet qua cổng S0/0/0 của router. Công ty muốn tất cả các thành viên trong công ty đều có thể truy cập được Internet.

Trong trường hợp này, người quản trị mạng thực hiện cấu hình PAT (NAT Overload) trên router để cho phép người dùng trong công ty có thể truy cập ra ngoài bằng địa chỉ được đăng ký trên cổng S0/0/0 của router.

Các lệnh cấu hình NAT như sau:


```
R(config)#access-list 1 permit 192.168.1.0 0.0.0.255
R(config)#ip nat inside source list 1 interface
s0/0/0 overload
R(config)#interface fa0/0
R(config-if)#ip nat inside
R(config)#interface S0/0/0
R(config-if)#ip nat outside
```

❖ Các lệnh kiểm tra cấu hình

R#show ip nat translation → hiển thị bảng NAT đang hoạt động

R#show ip nat statistics → hiển thị trạng thái hoạt động của NAT

R#clear ip nat translation * → xóa bảng NAT

R#debug ip nat → kiểm tra hoạt động của NAT, hiển thị các thông tin chuyển đổi NAT bởi router.

5. TỔNG KẾT CHƯƠNG

Cisco IOS NAT cho phép một tổ chức với những địa chỉ không đăng ký (địa chỉ local) có thể kết nối Internet bằng cách chuyển những địa chỉ này thành những địa chỉ đã được đăng ký (public).

NAT có ưu điểm là tiết kiệm địa chỉ đăng ký (public). Tuy nhiên, sử dụng NAT cũng có khuyết điểm là làm tăng thời gian trễ do phải thực hiện việc chuyển đổi địa chỉ trong các gói dữ liệu.

Ba kỹ thuật NAT được dùng là: *Static NAT*, *Dynamic NAT* và *NAT Overload (PAT)*. *Static NAT* được sử dụng để ánh xạ địa chỉ theo kiểu “one-to-one” và được chỉ định bởi người quản trị. *Dynamic NAT* là kiểu chuyển dịch địa chỉ dạng “one-to-one” một cách tự động. *NAT Overload* là kiểu chuyển dịch địa chỉ dạng “many-to-one” một cách tự động, sử dụng các chỉ số cổng (port) để phân biệt cho từng chuyển dịch.

6. CÂU HỎI VÀ BÀI TẬP

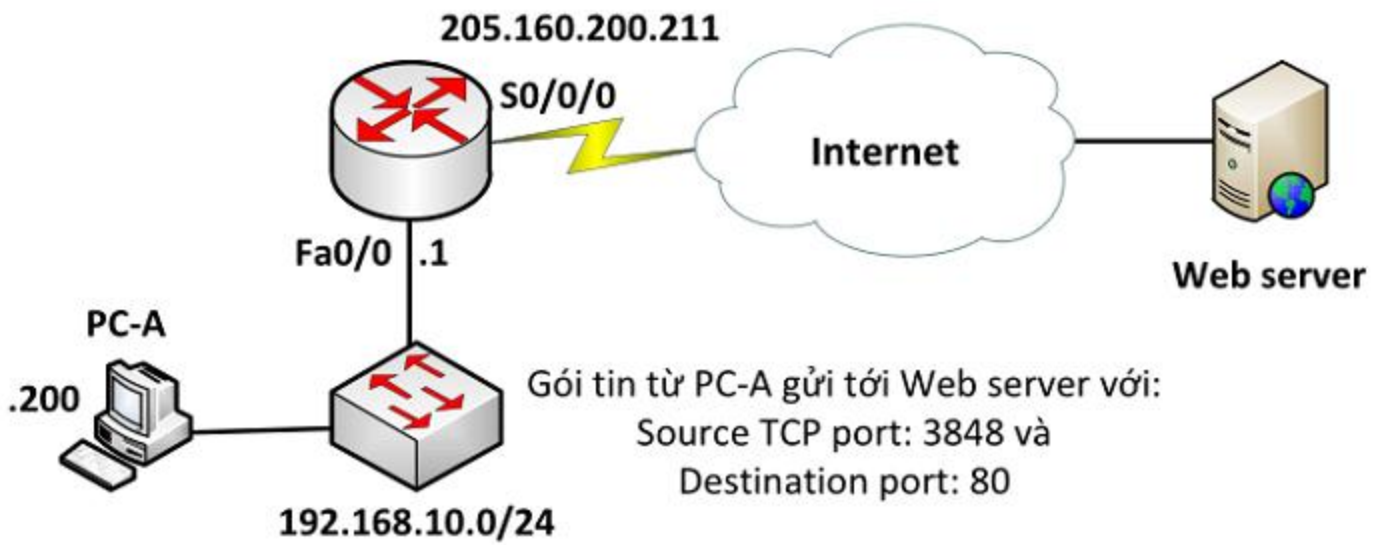
6.1 Địa chỉ "Inside Global" trong cấu hình NAT có ý nghĩa gì?

- A. Là địa chỉ MAC được các máy tính sử dụng để kết nối ra ngoài.
- B. Là địa chỉ tóm tắt đại diện cho tất cả các mạng bên trong.

C. Là địa chỉ cục bộ gán cho máy tính ở mạng bên trong.

D. Là địa chỉ được đăng ký (public) đại diện cho các máy tính bên trong khi đi ra mạng bên ngoài.

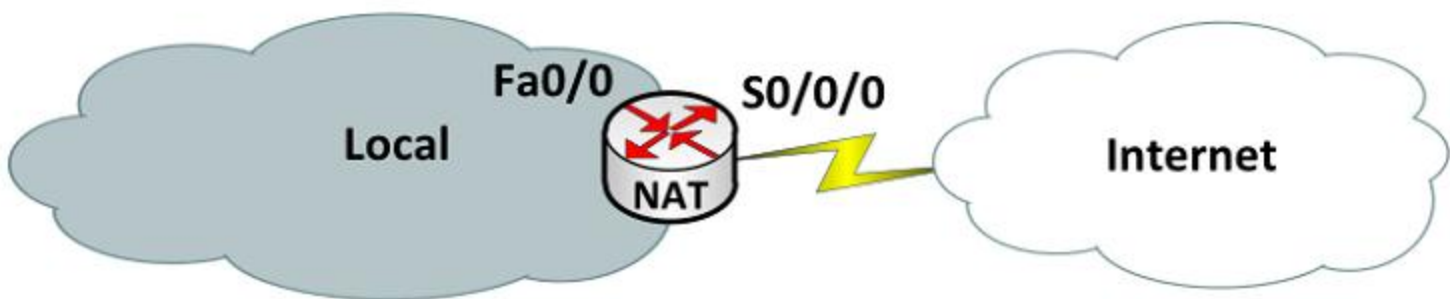
6.2 Cho mô hình mạng



NAT Overload đã được cấu hình trên router, phát biểu nào sau đây là đúng khi máy tính PC-A giao tiếp với Web server?

- A. Web server sử dụng địa chỉ IP đích là 205.160.200.211 và port đích là 80 khi gửi gói tin đến cho PC-A
- B. Máy tính PC-A sử dụng địa chỉ IP đích là 192.168.10.1 và port nguồn là 80 khi gửi các gói tin đến Web server.
- C. Web server sử dụng địa chỉ IP đích là 205.160.200.211 và port đích là 3848 khi gửi gói tin đến cho PC-A
- D. Máy tính PC-A sử dụng địa chỉ IP đích là 205.160.200.211 và port đích là 3848 khi gửi các gói tin đến Web server.

6.3 Cho mô hình mạng



Lệnh nào sau đây được cấu hình trên cổng S0/0/0 của Router NAT khi cấu hình NAT trên router này?

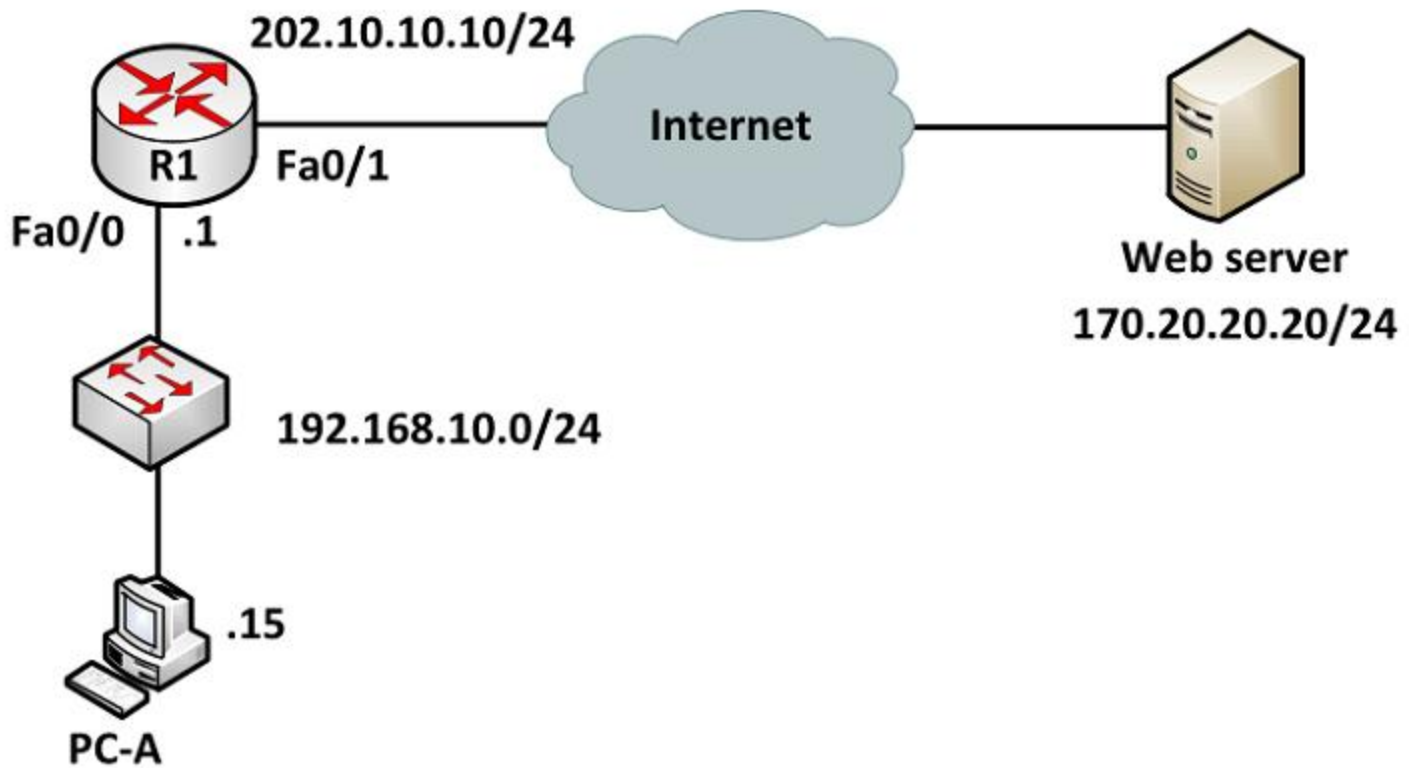
- A. `ip nat inside`
- B. `ip nat outside`

- C. ip nat inside
- D. ip nat outside

6.4 Hai phát biểu nào sau đây là đúng cho loại Static NAT

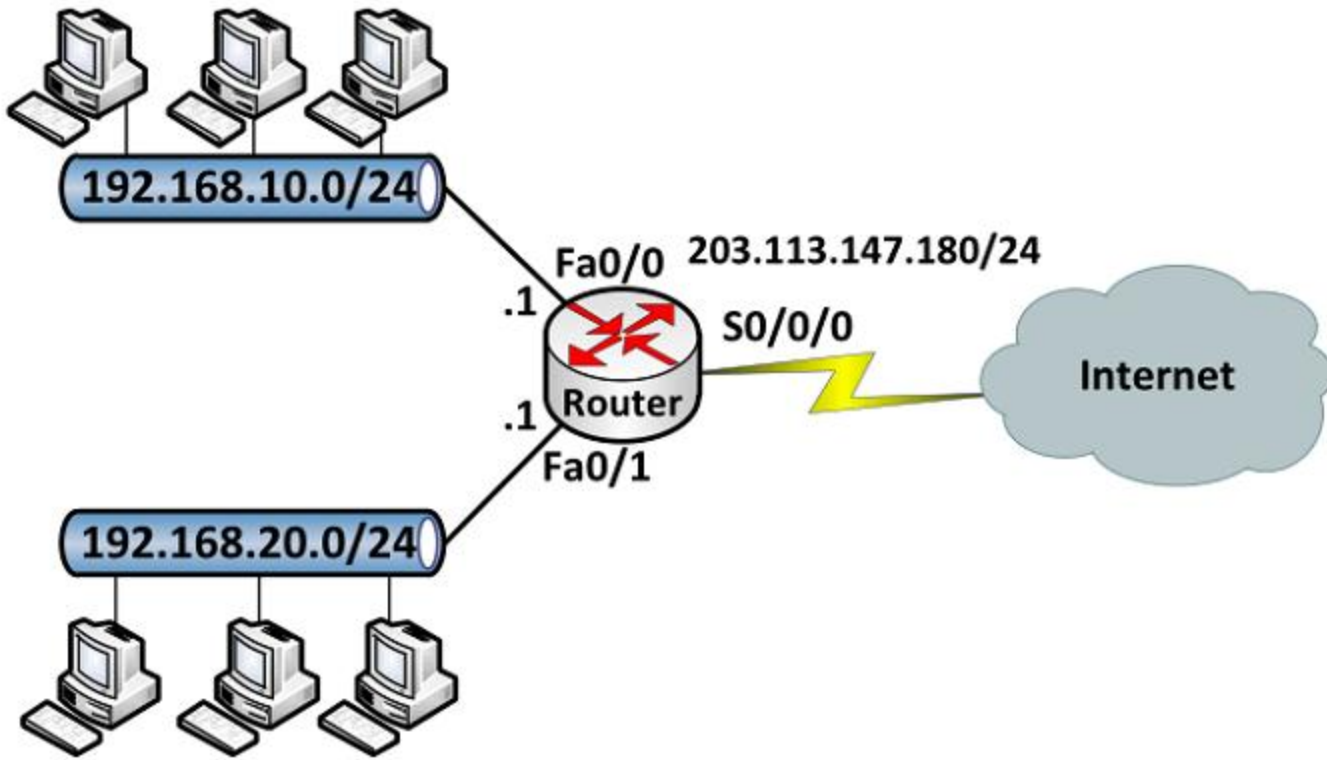
- A. Loại này cho phép từ bên ngoài có thể khởi tạo kết nối vào bên trong
- B. Loại này không yêu cầu phải chỉ ra cổng nào gắn với mạng ngoài và cổng nào gắn với mạng bên trong ở router thực hiện NAT
- C. Loại này có thể dùng ACL để cho phép nhiều kết nối khởi tạo từ mạng bên ngoài
- D. Loại này luôn được hiển thị trong bảng NAT

6.5 Cho mô hình mạng



Trong mô hình trên đã cấu hình NAT overload trên router R1. PC-A đang truy cập tới Web server. Hãy cho biết các địa chỉ: *inside local*, *inside global*, *outside local*, *outside global*.

6.6 Cho mô hình mạng



Router được cấu hình như sau:

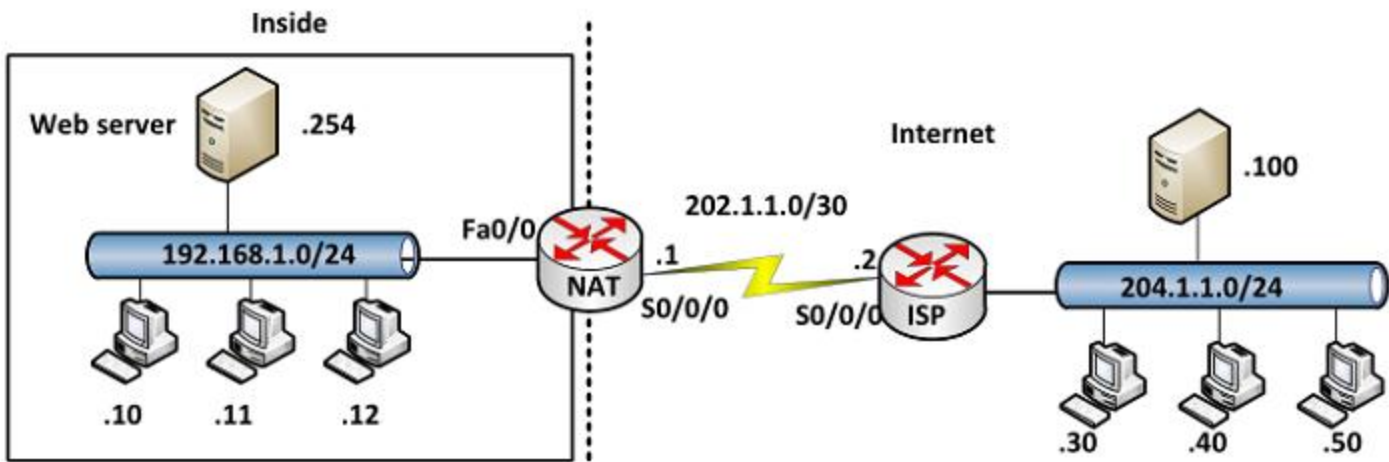
```
interface FastEthernet0/0
 ip address 192.168.10.1 255.255.255.0
 ip nat outside
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 192.168.20.1 255.255.255.0
 ip nat inside
 duplex auto
 speed auto
!
interface Serial0/0/0
 ip address 203.113.147.180 255.255.255.0
 ip nat inside
 clock rate 64000
!
interface Serial0/0/1
 no ip address
 shutdown
!
interface Vlan1
 no ip address
!
ip nat inside source list 1 interface Serial0/0/0
 overload
 ip classless
```

```
ip route 0.0.0.0 0.0.0.0 Serial0/0/0
!
access-list 1 permit 192.168.10.0 0.0.0.255
access-list 1 permit 192.168.20.0 0.0.0.255
```

Trên Router đã cấu hình NAT sai ở đâu?

- A. ACL cấu hình chưa đúng
- B. Cổng S0/0/0 và Fa0/0
- C. Cổng Fa0/1
- D. Lệnh default route cấu hình sai

6.7 Cho sơ đồ mạng



❖ Mô tả yêu cầu

Công ty thuê một IP public để dùng cho local Web server là 202.2.2.254 và đang kết nối ra Internet qua cổng S0/0/0 của Router. Yêu cầu cấu hình để cho các máy bên ngoài Internet có thể truy cập vào Web server và các máy bên trong có thể ra ngoài Internet net.

❖ Hướng dẫn cấu hình

- Các máy tính đặc địa chỉ IP và **default gateway** cho phù hợp
- Giữa router NAT và ISP không cấu hình bất kỳ giao thức định tuyến nào

- Router NAT tạo đường “default route” lên ISP

```
NAT(config)#ip route 0.0.0.0 0.0.0.0 s0/0/0
```

- Cấu hình public cho Web server

```
NAT(config)#ip nat inside source static
192.168.1.254 202.1.1.254
```

- Cấu hình cho phép các máy tính bên trong ra ngoài Internet

```
NAT(config)#access-list 1 permit 192.168.1.0
0.0.0.255
NAT(config)#ip nat inside source list 1
interface S0/0/0 overload
NAT(config-if)#int Fa0/0
NAT(config-if)#ip nat inside
NAT(config-if)#int S0/0/0
NAT(config-if)#ip nat outside
```

❖ **Kiểm tra cấu hình bằng các lệnh đã học**

- ✓ Thực hiện ping giữa các máy tính, phân tích các gói truyền nhận bằng lệnh *debug ip packet* trên router trước khi thực hiện lệnh *ping*.
- ✓ Sử dụng lệnh *debug ip nat* để xem quá trình hoạt động của quá trình NAT

Chương 5

CÁC DỊCH VỤ WAN

Chương này sẽ đề cập đến một số giao thức hoạt động trên môi trường WAN và một số dịch vụ WAN phổ biến. Học xong chương này, người học có khả năng:

- *Trình bày được khái niệm về WAN*
- *Trình bày được một số đặc điểm cơ bản của giao thức PPP, HDLC*
- *Phân biệt và cấu hình hai giao thức chứng thực trên PPP (PAP, CHAP)*
- *Phân biệt và cấu hình một số kỹ thuật WAN: Serial Point-to-Point, Frame Relay*

1. GIỚI THIỆU

Một WAN là một mạng trao đổi dữ liệu, nó hoạt động vượt ra ngoài phạm vi vật lý của LAN. WAN hoạt động trên một miền địa lý rộng lớn, kết nối hệ thống máy tính của cùng một đơn vị giữa các tỉnh, các quốc gia hay châu lục...

WAN sử dụng các liên kết dữ liệu như là Frame Relay, ATM, MPLS hỗ trợ các dịch vụ để truy cập băng thông vượt qua vùng địa lý rộng lớn.

Các tính năng kỹ thuật WAN nằm ở ba tầng cuối cùng của mô hình OSI.

- Các kiểu kết nối WAN (layer 1): đường thuê riêng (leased line), chuyển mạch kênh (circuit switched), chuyển mạch gói (packet-switched).
- Các giao thức đóng gói WAN (Layer 2): HDLC, PPP, ATM, Frame Relay, VPN, MPLS
- Một số kỹ thuật WAN: trước đây một số kỹ thuật được dùng như ISDN, X.25, ATM, các kỹ thuật đang sử dụng nhiều hiện nay như DSL, Leased lined, MPLS,...

Trong chương này, chúng ta sẽ tìm hiểu về 2 kỹ thuật WAN: PPP và Frame-Relay.

2. KẾT NỐI SERIAL POINT-TO-POINT

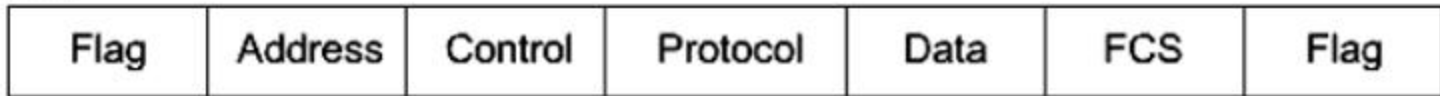
Hai giao thức liên kết dữ liệu (data link) WAN sử dụng trong mạng WAN kết nối *Serial Point-to-Point* được dùng phổ biến là HDLC và PPP.

HDLC



Hình 5.1 Frame HDLC

PPP



Hình 5.2 Định dạng của frame PPP

PPP là một giao thức thường được chọn để triển khai trên một kết nối WAN nối tiếp. PPP có hỗ trợ quá trình xác thực PAP và CHAP.

❖ Quá trình chứng thực trong PPP

PPP tổ chức gồm 2 giao thức sau:

- *Link Control Protocol (LCP)*: sử dụng cho việc thiết lập, cấu hình và kiểm tra kết nối ở tầng liên kết dữ liệu.
- *Network Control Protocol (NCP)*: sử dụng cho việc thiết lập và cấu hình các giao thức tầng mạng khác nhau.

❖ Quá trình thiết lập kết nối PPP

Quá trình thiết lập kết nối PPP qua 4 bước: Thiết lập kết nối và thương lượng cấu hình; quyết định chất lượng kết nối; thương lượng cấu hình giao thức tầng mạng và kết thúc kết nối.

- *Thiết lập kết nối và cấu hình*

Mỗi thiết bị PPP gửi gói tin LCP để cấu hình và thiết lập kết nối ở tầng liên kết dữ liệu. Gói tin LCP chứa các trường: “MTU”, “compression”, và giao thức chứng thực kết nối. LCP đầu tiên mở kết nối và thương lượng các tham số cấu hình. Giai đoạn này hoàn tất khi các gói tin thông nhất cấu hình (ACK) được gửi và nhận.

- *Quyết định chất lượng kết nối*

Liên kết được kiểm tra xem có tốt không để chuyển các giao thức lên tầng mạng hay không. Sau đó *Client* có thể được chứng thực. Việc chứng thực diễn ra trước giai đoạn cấu hình giao thức tầng mạng. PPP hỗ trợ hai giao thức chứng thực là: PAP và CHAP.

- *Thương lượng cấu hình tầng mạng*

Các thiết bị PPP gửi gói tin NCP để chọn và cấu hình một hoặc nhiều giao thức tầng mạng (ví dụ như IP). Khi giao thức tầng mạng được cấu hình, các gói tin từ giao thức tầng mạng có thể được gửi qua liên kết. Nếu LCP kết thúc kết nối, nó cung cấp các giao thức tầng mạng để có thể có những hành động phù hợp.

- *Kết thúc kết nối*

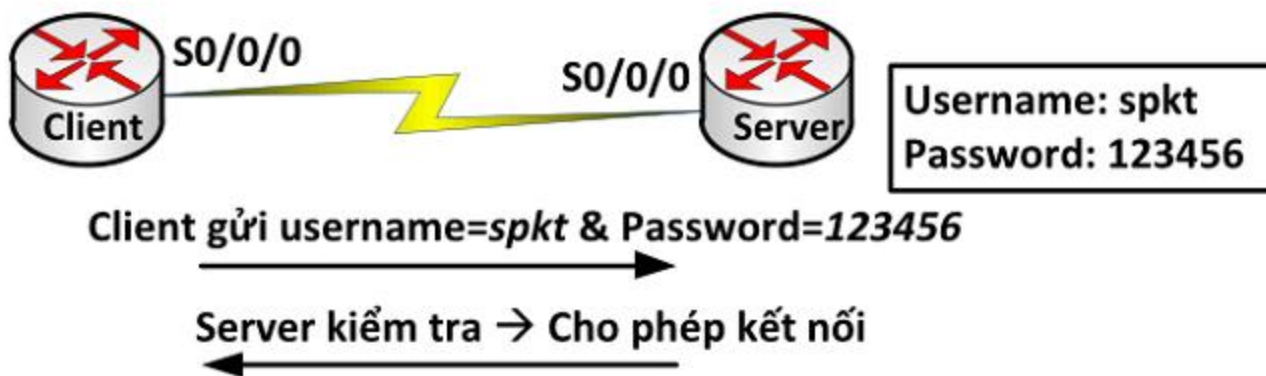
LCP có thể kết thúc kết nối bất cứ lúc nào. Điều này luôn được thực hiện ở yêu cầu của người dùng. Kết thúc kết nối cũng có thể xảy ra do sự cố vật lý, như là đứt kết nối hay vượt quá thời gian qui định (timeout).

❖ **Giao thức chứng thực PAP và CHAP**

- *Chứng thực PPP bằng PAP*

PAP sử dụng cơ chế bắt tay 2 bước. Đầu tiên *Client* sẽ gửi *username* và *password* cho *Server* để xác thực. *Server* sẽ tiến hành kiểm tra, nếu thành công thì sẽ thiết lập kết nối; ngược lại sẽ không thiết lập kết nối với *Client*.

Password được gửi dưới dạng không được mã hóa (clear – text) và *username/password* được gửi đi kiểm tra một lần khi thiết lập kết nối.



Hình 5.3 Chứng thực PAP

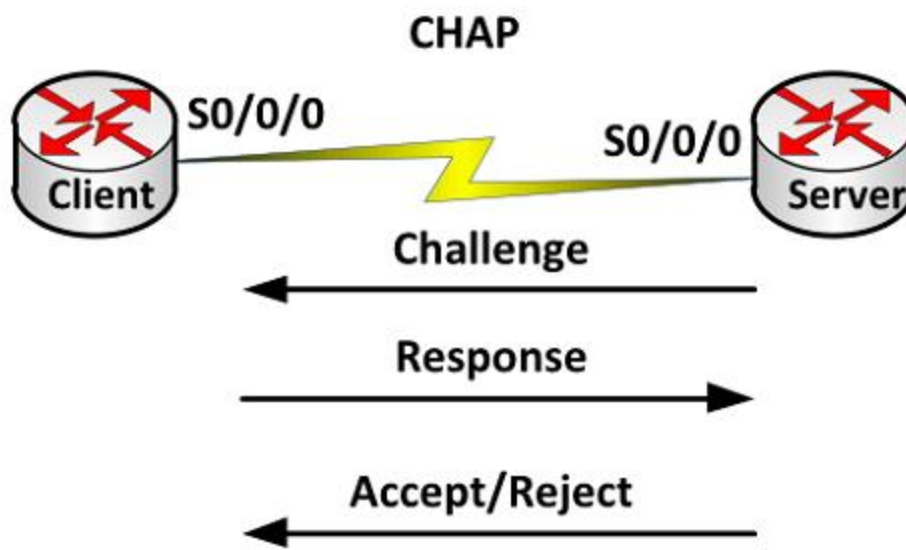
- *Chứng thực PPP bằng CHAP*

Sử dụng kỹ thuật 3 bước bắt tay (three-way handshake). CHAP được thực hiện ở lúc bắt đầu thiết lập kết nối và luôn được lặp lại trong suốt quá trình kết nối được duy trì.

Client muốn thiết lập kết nối với *Server*, *Server* gửi một thông điệp “challenge” yêu cầu *Client* gửi giá trị để *Server* chứng thực. Thông điệp gửi từ *Server* có chứa một số ngẫu nhiên dùng làm đầu vào cho thuật toán “hash”.

Client nhận được thông điệp yêu cầu của *Server*. Nó sẽ sử dụng thuật toán “hash” với đầu vào là *hostname*, *password* và số ngẫu nhiên vừa nhận được và tính toán ra một giá trị nào đó và gửi giá trị này qua cho *Server*.

Server sẽ kiểm tra danh sách “username” (nếu cấu hình nhiều username) để tìm ra “username” nào giống với *hostname* của *Client*. Sau khi tìm được “username” đó, nó dùng thuật toán “hash” để mã hóa *password* tương ứng và số ngẫu nhiên trong thông điệp “challenge” ban đầu mà nó gửi cho *Client* để tính ra một giá trị nào đó. Và giá trị này sẽ so sánh với giá trị do *Client* gửi qua, nếu giống nhau thì xác thực thành công; nếu không thì kết nối sẽ bị xóa ngay.



Hình 5.4 Chứng thực CHAP

Một cách đơn giản, ta cần nắm ý tưởng sau khi cấu hình CHAP: mỗi đầu kết nối phải có khai báo *username* và *password*. *Username* bên R1 phải là *hostname* của R2 và *username* khai báo bên R2 là *hostname* của R1, *password* hai bên phải giống nhau.

❖ Cấu hình PPP

- Cấu hình PPP

```
Router(config)#interface <interface>
Router(config-if)#encapsulation ppp
```

- Cấu hình chứng thực PPP PAP

Bước 1: Tạo username và password trên Server

```
Router(config)#username <username> password
<password>
```

Bước 2: Enable PPP

```
Router(config-if)#encapsulation ppp
```

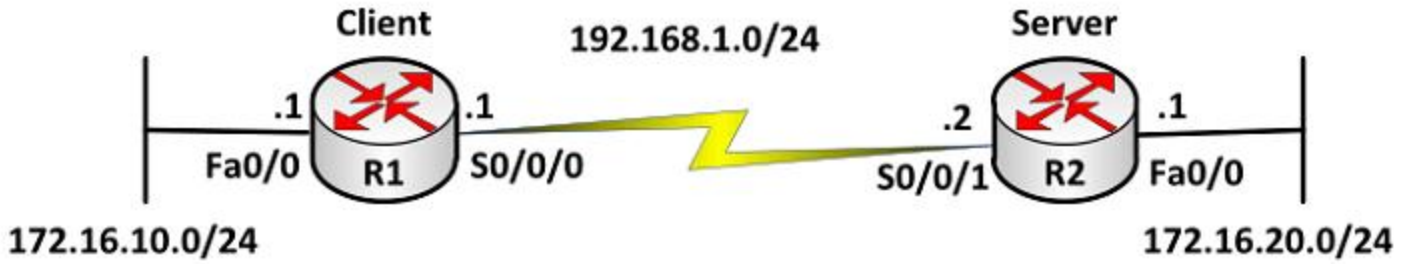
Bước 3: Cấu hình xác thực

```
Router(config-if)#ppp authentication  
{pap|chap|pap-chap|chap-pap}
```

Bước 4: PAP phải được enable trên interface bằng lệnh

```
Router(config-if)#ppp pap sent-username  
<username> password <password>
```

Ví dụ 1: Cấu hình PPP chứng thực bằng PAP



- **Mô tả**

Router R2 sẽ chứng thực cho router R1 bằng giao thức PAP

- **Hướng dẫn cấu hình**

- **Cấu hình cơ bản**

```
R1(config)#int S0/0/0  
R1(config-if)#ip address 192.168.1.1  
255.255.255.0  
R1(config-if)#exit  
R2(config)#int S0/0/1  
R2(config-if)#ip address 192.168.1.2  
255.255.255.0  
R2(config-if)#exit
```

- **Cấu hình chứng thực PAP**

```
R1(config)#int S0/0/0  
R1(config-if)#encapsulation ppp  
R1(config-if)#ppp pap sent-username cisco  
password cisco  
R2(config)#username cisco password cisco  
R2(config)#int S0/0/1  
R2(config-if)#encapsulation ppp  
R2(config-if)#ppp authentication pap
```

- **Cấu hình định tuyến:** tùy chọn giao thức

- **Kiểm tra cấu hình**

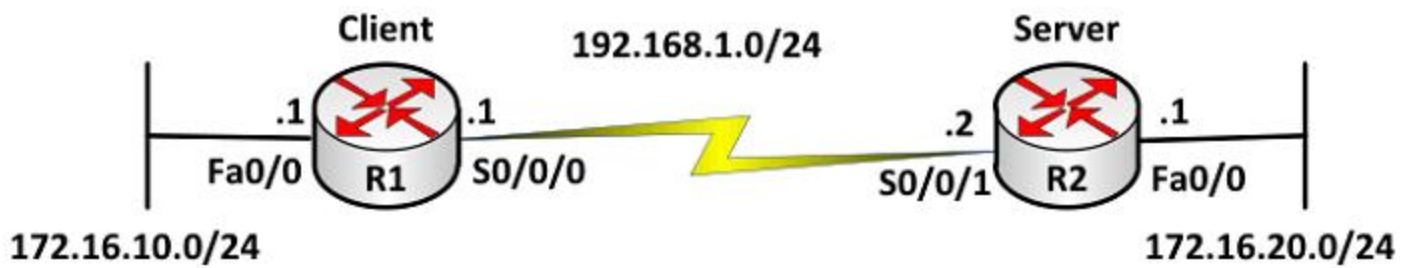
Sử dụng các lệnh sau:

```
ping
```

```
debug ppp authentication
```

- ❖ **Cấu hình chứng thực PPP CHAP**

Trường hợp 1: Các router dùng hostname để chứng thực



- **Mô tả**

Router R2 chứng thực cho router R1 bằng giao thức CHAP. Trường hợp mặc định, router gửi hostname để chứng thực.

- **Các bước cấu hình**

- **Cấu hình cơ bản**

```
R1 (config)#int S0/0/0
```

```
R1 (config-if)#ip address 192.168.1.1  
255.255.255.0
```

```
R1 (config-if)#exit
```

```
R2 (config)#int S0/0/1
```

```
R2 (config-if)#ip address 192.168.1.2  
255.255.255.0
```

```
R2 (config-if)#exit
```

- **Cấu hình chứng thực CHAP**

```
R1 (config)#username R2 password cisco
```

```
R1 (config)#int S0/0/0
```

```
R1 (config-if)#encapsulation ppp
```

```
R2 (config)#username R1 password cisco
```

```
R2 (config)#interface serial 0/0/1
```

```
R2 (config-if)#encapsulation ppp
```

```
R2 (config-if) #ppp authentication chap
```

- **Cấu hình định tuyến:** tùy chọn giao thức

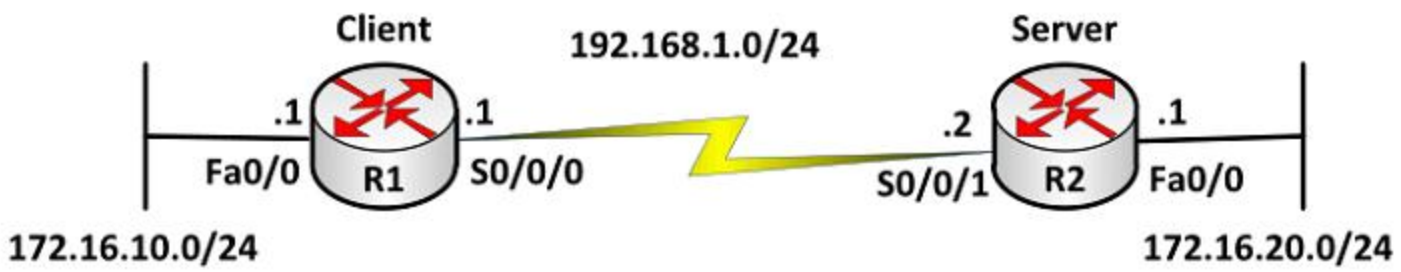
- **Kiểm tra cấu hình**

Sử dụng các lệnh sau:

```
Router#ping
```

```
Router#debug ppp authentication
```

Trường hợp 2: Các router gửi username & password bất kỳ



- **Yêu cầu**

Router R2 sẽ chứng thực cho router R1 bằng giao thức CHAP trường hợp router gửi hostname và password được chỉ ra.

- **Các bước cấu hình**

- **Cấu hình cơ bản:**

```
R1 (config) #interface serial 0/0/0
```

```
R1 (config-if) #ip address 192.168.1.1  
255.255.255.0
```

```
R1 (config-if) #exit
```

```
R2 (config) #interface serial 0/0/1
```

```
R2 (config-if) #ip address 192.168.1.2  
255.255.255.0
```

```
R2 (config-if) #exit
```

- **Cấu hình chứng thực CHAP**

```
R1 (config) #int S0/0/0
```

```
R1 (config-if) #encapsulation ppp
```

```
R1 (config-if) #ppp chap hostname abc
```

```
R1 (config-if) #ppp chap password cisco
```

```
R2 (config) #username abc password cisco
```

```
R2 (config)#int S0/0/1
R2 (config-if)#encapsulation ppp
R2 (config-if)#ppp authentication chap
```

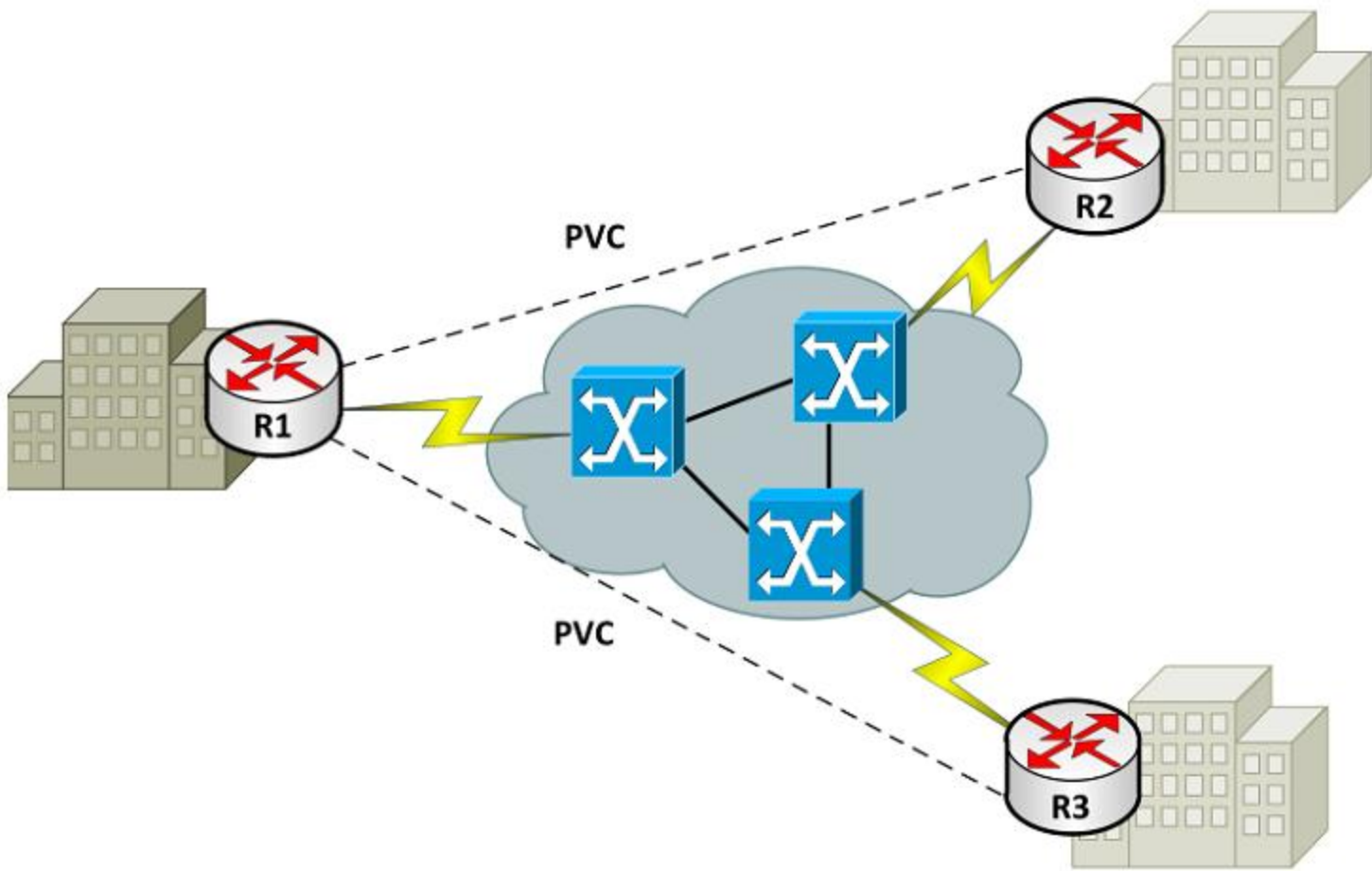
- **Cấu hình định tuyến:** tùy chọn giao thức

- **Kiểm tra cấu hình**

Sử dụng các lệnh sau:

```
Router#ping
Router#debug ppp authentication
```

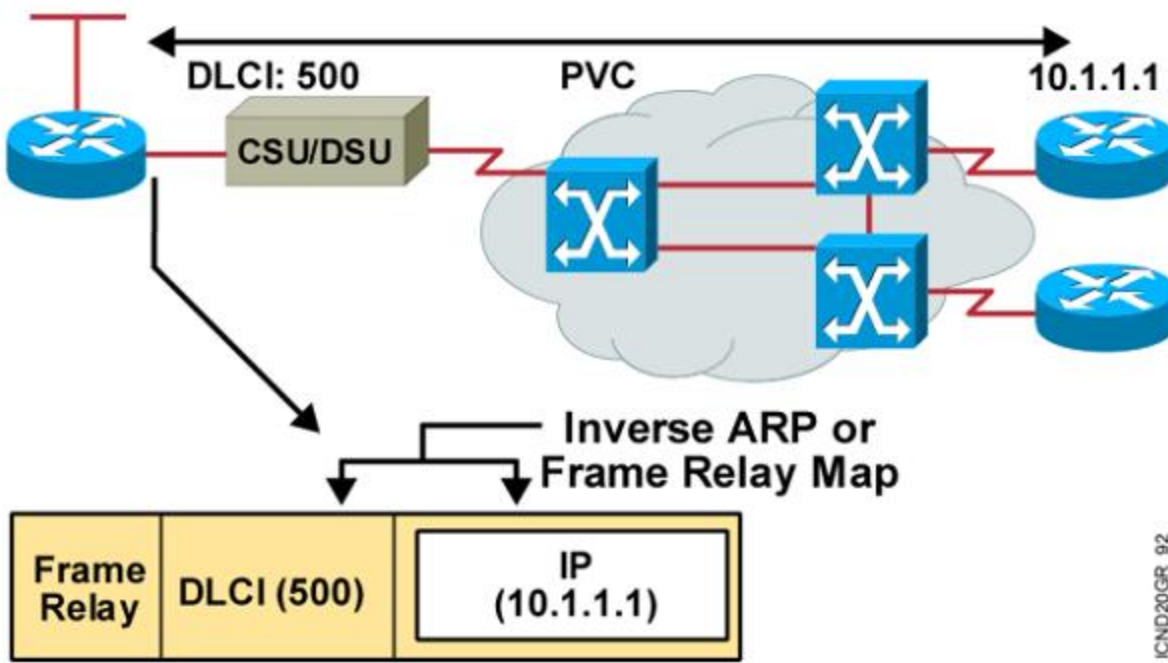
3. FRAME RELAY



Hình 5.5 Mô hình mạng Frame Relay

Frame Relay là dịch vụ WAN chuyên mạch gói theo hướng kết nối, hoạt động ở tầng liên kết dữ liệu và sử dụng các mạch ảo (virtual circuit) để tạo các kết nối.

Sự khác nhau giữa *Frame Relay* và *point-to-point* là thiết bị ở nhà cung cấp dịch vụ kiểm tra các gói tin gửi bởi router. Mỗi “*frame header*” giữ một trường địa chỉ gọi là *DLCI (Data-Link Connection Identifier)*. WAN Switch chuyển dữ liệu dựa vào *DLCI*, thông qua mạng nhà cung cấp dịch vụ để đến đầu bên kia của mạng.

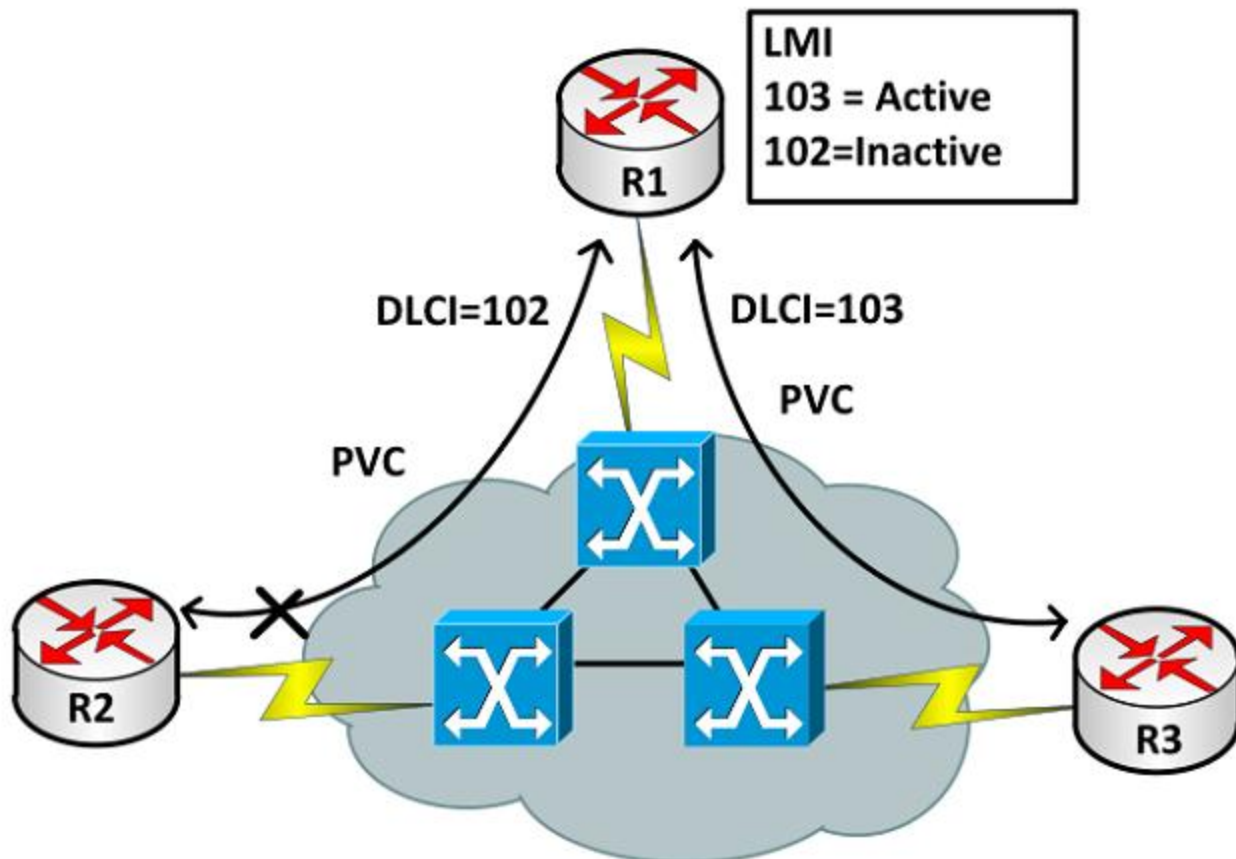


Hình 5.6 Ảnh xạ DLCI và địa chỉ IP

Các kết nối ảo trên cùng một đường truyền vật lý được phân biệt với nhau bởi chỉ số DLCI. Chỉ số DLCI được ghi trong mỗi gói dữ liệu truyền đi và chỉ có ý nghĩa cục bộ.

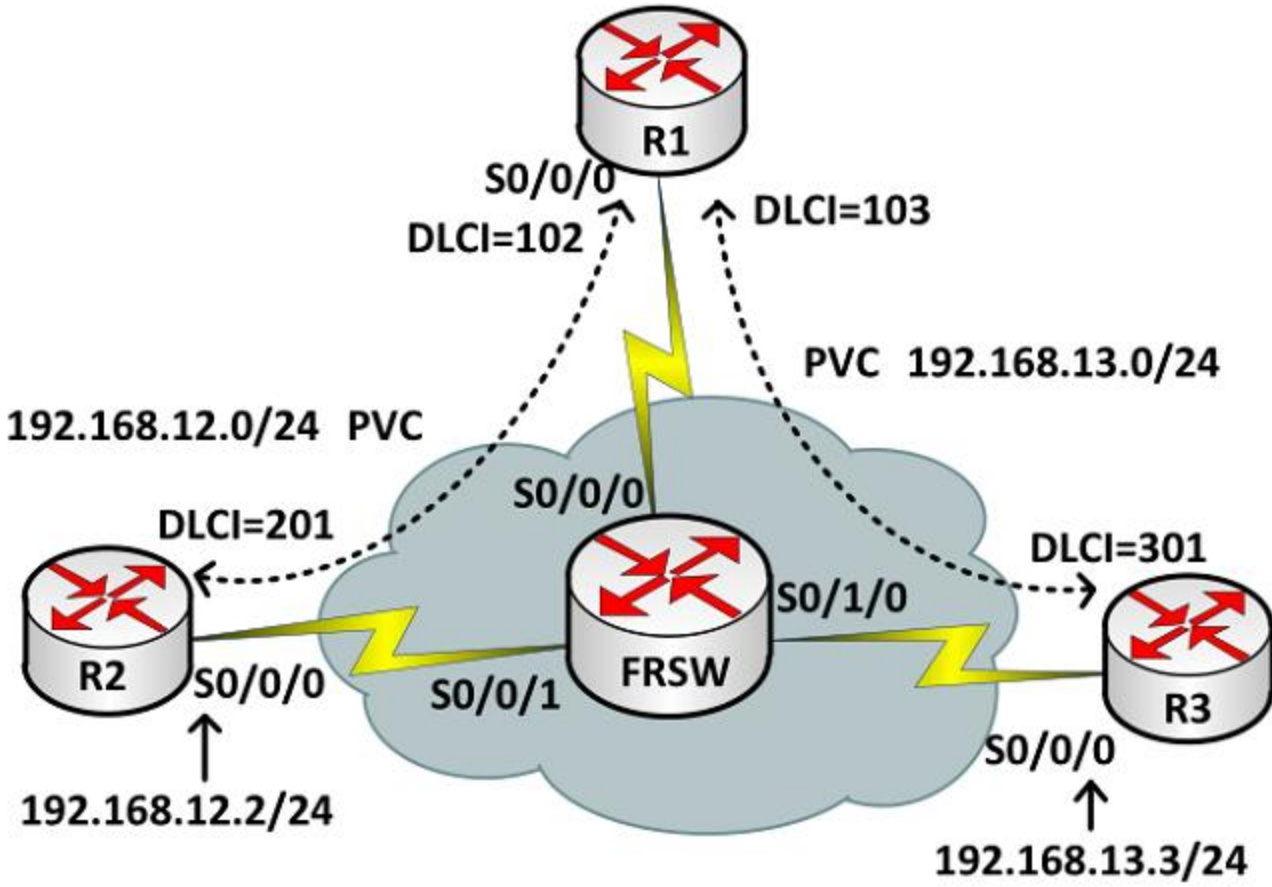
❖ Chức năng của LMI

- Quyết định trạng thái hoạt động của các PVC khác nhau
- Báo cho router biết PVC nào đang sẵn sàng.
- Ba loại LMI có thể được sử dụng là: ansi, cisco, và q933a.



Hình 5.7 Hoạt động của LMI

❖ **Frame Relay map**



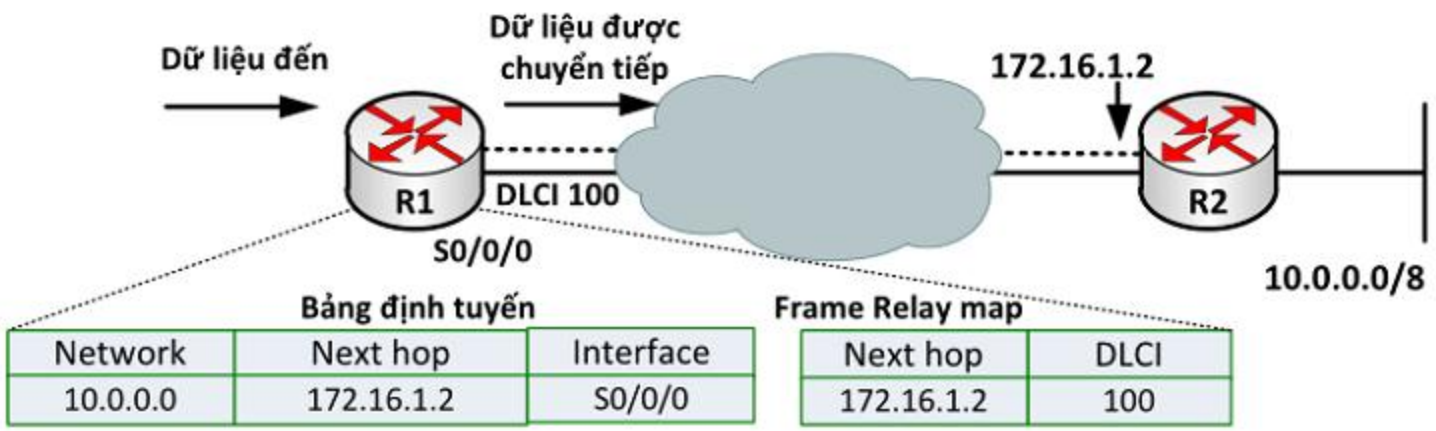
“Frame Relay map” trên R1

| IP của router kế tiếp | DLCI |
|-----------------------|------|
| 192.168.12.2 | 102 |
| 192.168.13.3 | 103 |

Trên FRSW thiết lập bảng chuyển mạch

| IN PORT | IN DLCI | OUT PORT | OUT DLCI |
|---------|---------|----------|----------|
| S0/0/0 | 102 | S0/0/1 | 201 |
| S0/0/0 | 103 | S0/1/0 | 301 |
| S0/0/1 | 201 | S0/0/0 | 102 |
| S0/1/0 | 301 | S0/0/0 | 103 |

- Bảng định tuyến được sử dụng để cung cấp địa chỉ của “next-hop” hoặc là DLCI cho các lưu lượng mạng.
- Các thông tin này được thực hiện thông qua một cấu trúc dữ liệu gọi là **Frame Relay map**. Cấu trúc dữ liệu này có thể được cấu hình tĩnh trên router, hoặc cấu hình tự động bằng cách sử dụng tính năng *inverse ARP*.

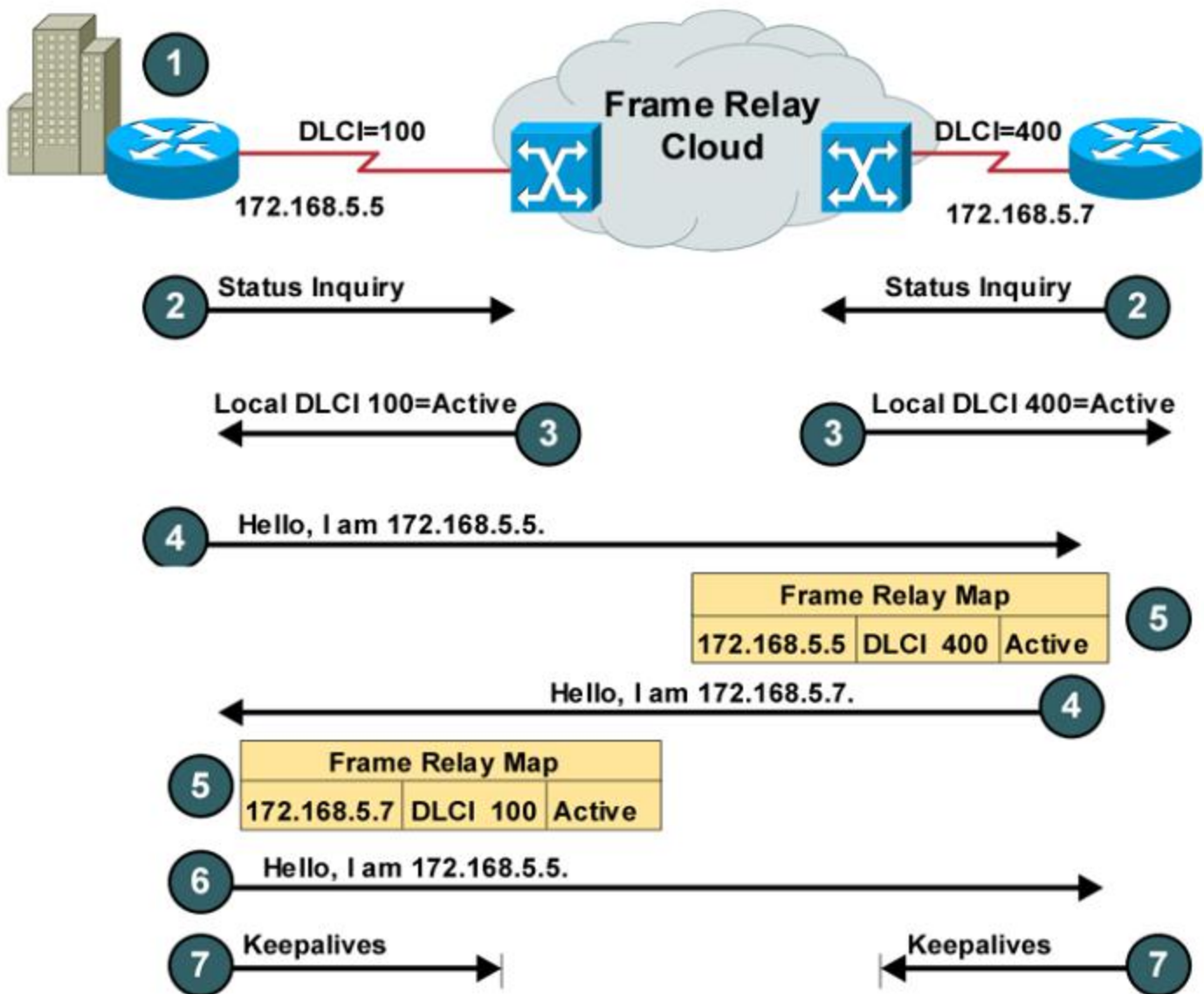


Hình 5.8 Frame relay map

❖ Inverse ARP

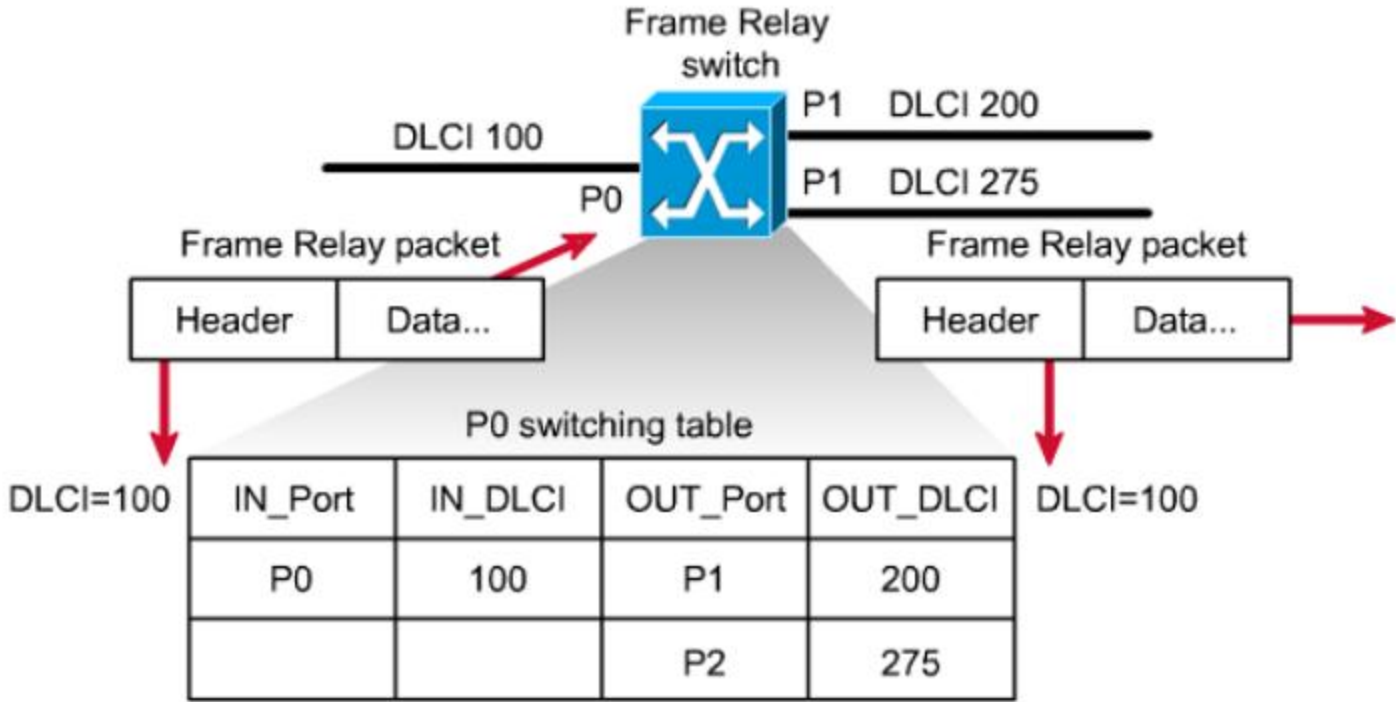
Cơ chế *inverse ARP* cho phép router xây dựng tự động bảng “*frame relay map*”

- Router học các DLCI đã được dùng trong suốt quá trình khởi tạo trao đổi LMI
- Sau đó router gửi một “*inverse-ARP request*” đến mỗi DLCI được cấu hình trên cổng.
- Những thông tin phản hồi từ *inversr-ARP* được sử dụng để xây dựng *Frame relay map*.



Hình 5.9 Quá trình xây dựng bảng “*frame relay map*”

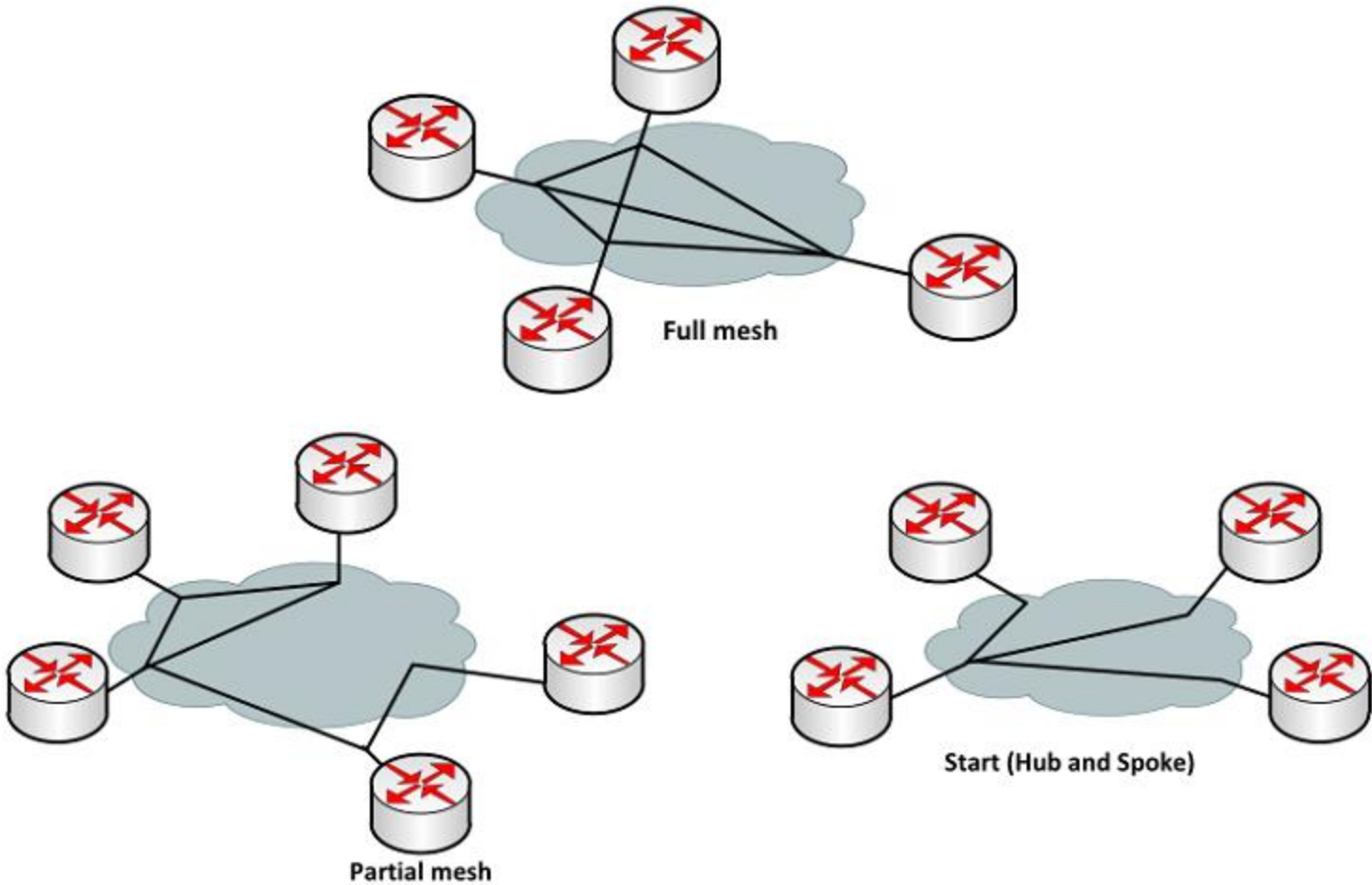
- ❖ Bảng chuyển mạch Frame Relay: bảng chuyển mạch Frame Relay bao gồm bốn thành phần: 2 cho port và DLCI vào (*incoming*), 2 cho port và DLCI ra (*outgoing*)



Hình 5.10 Bảng chuyển mạch Frame Relay

❖ **Kiến trúc mạng Frame Relay**

Có 3 mô hình phổ biến áp dụng trong mạng Frame Relay



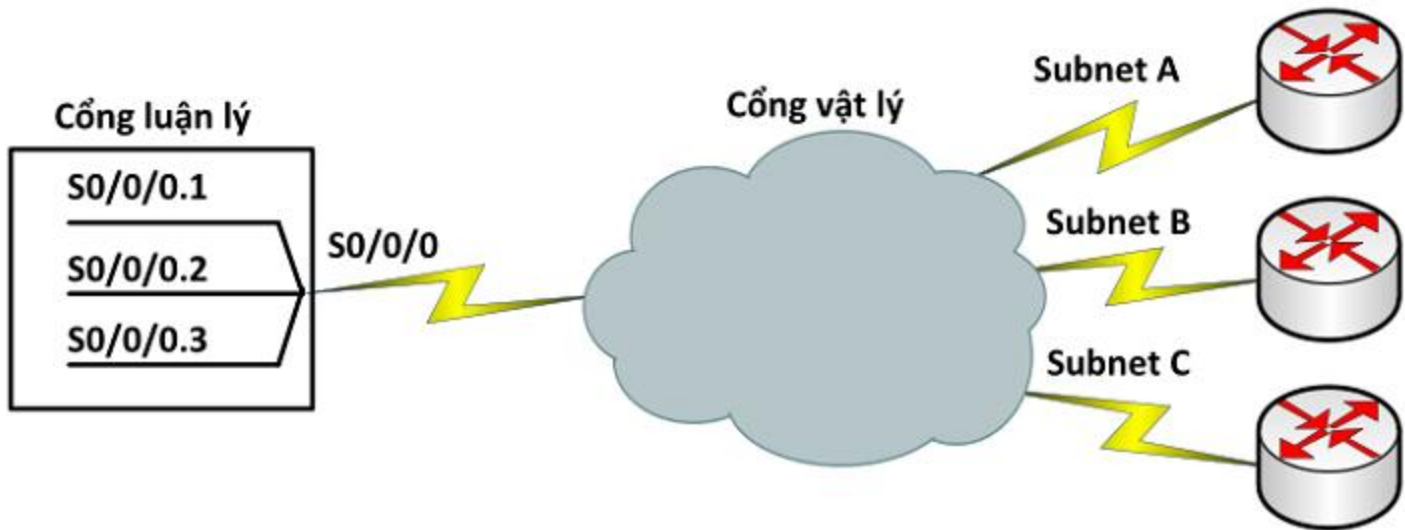
Hình 5.11 Kiến trúc mạng Frame Relay

Mặc định Frame Relay sử dụng môi trường NBMA (Non-broadcast Multiaccess)

❖ Frame Relay sub-interface là gì ?

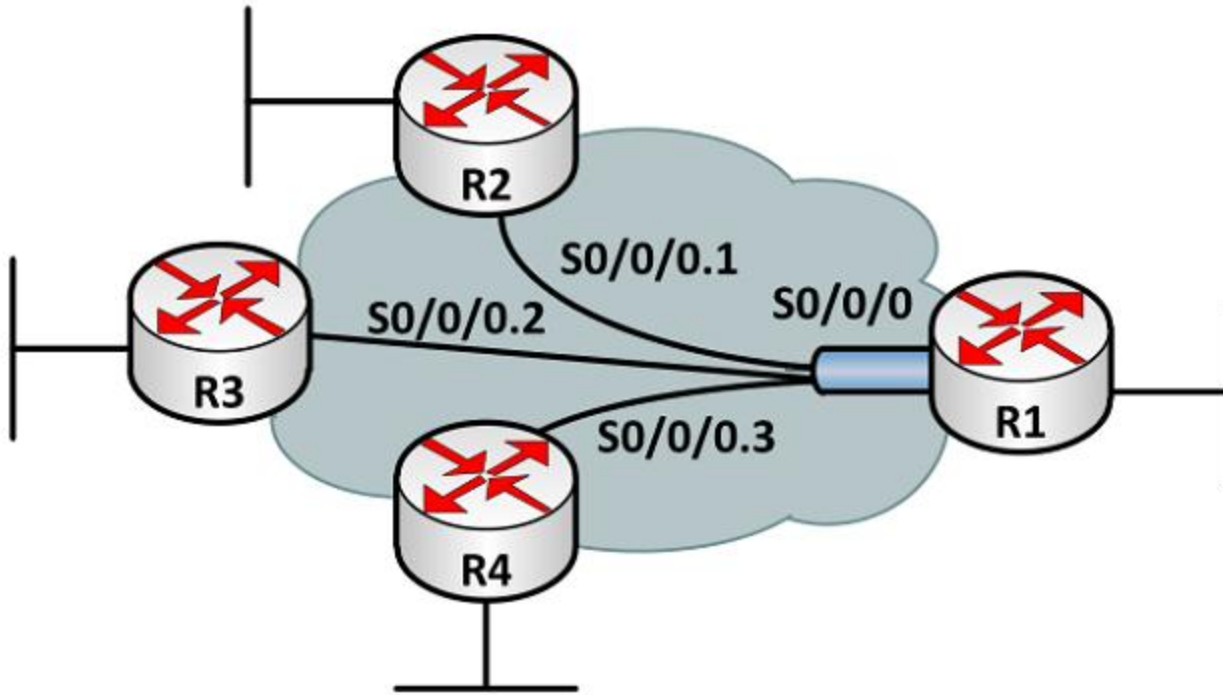
Sub-interface là sự chia luận lý từ một cổng vật lý. Trong cấu hình Frame Relay, mỗi PVC có thể được cấu hình như là một kết nối point-to-point sử dụng mỗi *sub-interface* hoạt động như là một đường kết nối trực tiếp.

Dùng nhiều *sub-interface* giúp giảm chi phí cho việc triển khai mạng Frame Relay.



Hình 5.12 Chia cổng vật lý thành các cổng luận lý

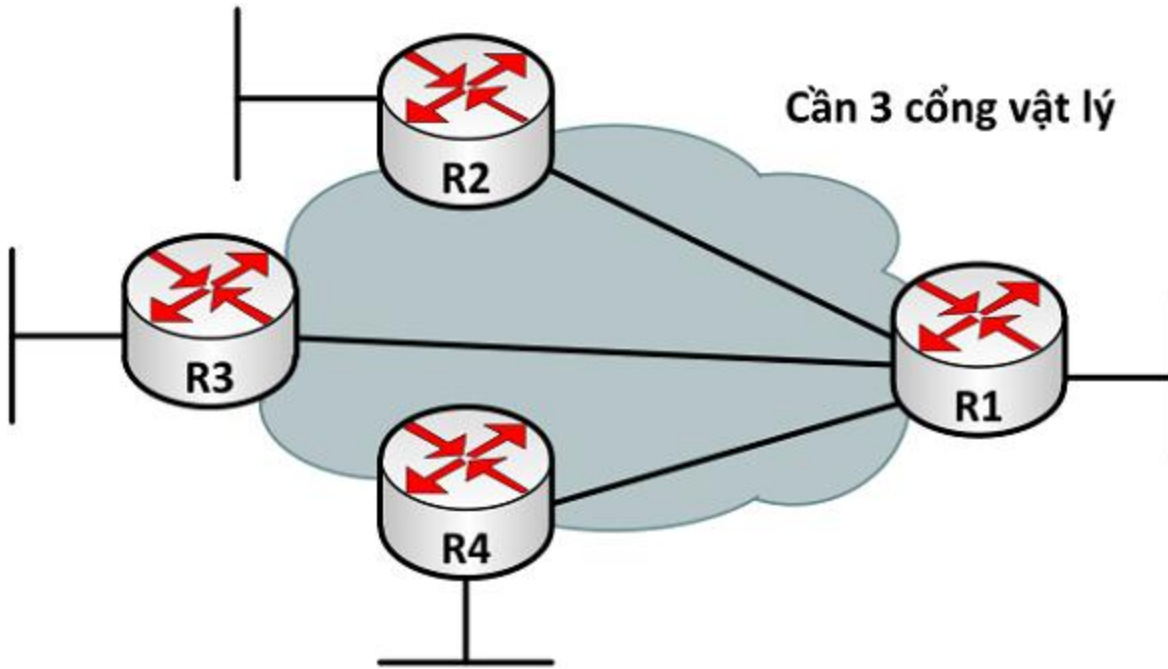
❖ Frame Relay dùng sub-interface



Hình 5.13 Frame Relay dùng subinterface

Sub-interface Frame Relay có thể cấu hình dùng trong các kết nối đa điểm (multipoint). Mỗi *sub-interface* kết nối đa điểm thiết lập nhiều kết nối PVC đến nhiều router khác nhau. Tất cả các router nối đều nằm trong cùng một mạng (subnet). Điều này giúp tiết kiệm được địa chỉ mạng và có ý nghĩa trong trường hợp không sử dụng VLSM. Tuy nhiên, *sub-interface* kết nối đa điểm lại không giải quyết được vấn đề “*split-horizon*”.

❖ Frame Relay không dùng sub-interface



Hình 5.14 Frame relay không dùng cổng luận lý

❖ Cấu hình Frame Relay căn bản

- Cấu hình Frame Relay DTE

- Chọn cổng cấu hình

```
Router(config)#interface <interface>
```

- Cấu hình địa chỉ mạng

```
Router(config-if)# ip address <network-number> <subnet-mask>
```

Ví dụ: Router(config-if)#ip address 192.168.1.1
255.255.255.0

- Chọn kiểu đóng gói

```
Router(config-if)#encapsulation frame-relay {cisco | IETF}
```

- Cấu hình LMI

```
Router(config-if)#frame-relay lmi-type {ansi | cisco | q933a}
```

- Một số câu lệnh kiểm tra

```
Router#show interfaces interface
```

```
Router#show frame-relay pvc
```

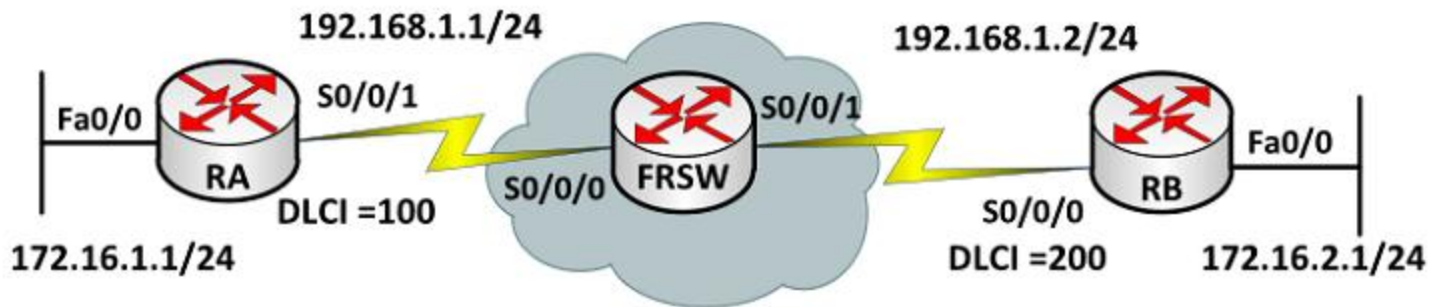
```
Router#show frame-relay map
```

```
Router#show frame-relay lmi
```

- Cấu hình Frame Relay Switch

- Enable Frame Relay switching
`FRSW(config)#frame-relay switching`
- Cấu hình trên interface
`FRSW(config-if)#encapsulation frame-relay`
`FRSW(config-if)#frame-relay intf-type dce|dte`
`FRSW(config-if)#clock rate clock-rate`
`FRSW(config-if)#frame-relay lmi-type cisco|ansi|q933a`
- Cấu hình FR route (tạo PVC – Switching table)
`FRSW(config-if)#frame-relay route <input-dlci> interface <output-interface> <output-dlci>`

Ví dụ 1: Frame Relay căn bản



- **Yêu cầu**

Cấu hình mạng Frame Relay cho mô hình mạng trên.

- **Các bước thực hiện**

- Cấu hình Frame Relay Switch

```
FRSW#configure terminal
FRSW(config)#frame-relay switching
FRSW(config)#interface s0/0/0
FRSW(config-if)#no ip address
FRSW(config-if)#encapsulation frame-relay
FRSW(config-if)#clock rate 64000
FRSW(config-if)#frame-relay intf-type dce
FRSW(config-if)#frame-relay route 100 interface
s0/0/1 200
```

```
FRSW(config-if)#exit
FRSW(config)#interface s0/0/1
FRSW(config-if)#no ip address
FRSW(config-if)#encapsulation frame-relay
FRSW(config-if)#clock rate 64000
FRSW(config-if)#frame-relay intf-type dce
FRSW(config-if)#frame-relay route 200 interface
serial 0/0/0 100
FRSW(config-if)#exit
```

- **Cấu hình Router A:**

```
RA(config)#interface S0/0/1
RA(config-if)#ip address 192.168.1.1 255.255.255.0
RA(config-if)#encapsulation frame-relay
RA(config-if)#no shutdown
RA(config-if)#exit
RA(config)#interface Fa0/0
RA(config-if)#ip address 172.16.1.1 255.255.255.0
RA(config-if)#no shut
RA(config-if)#exit
RA(config)#router eigrp 100
RA(config-router)#network 172.16.0.0
RA(config-router)#network 192.168.1.0
```

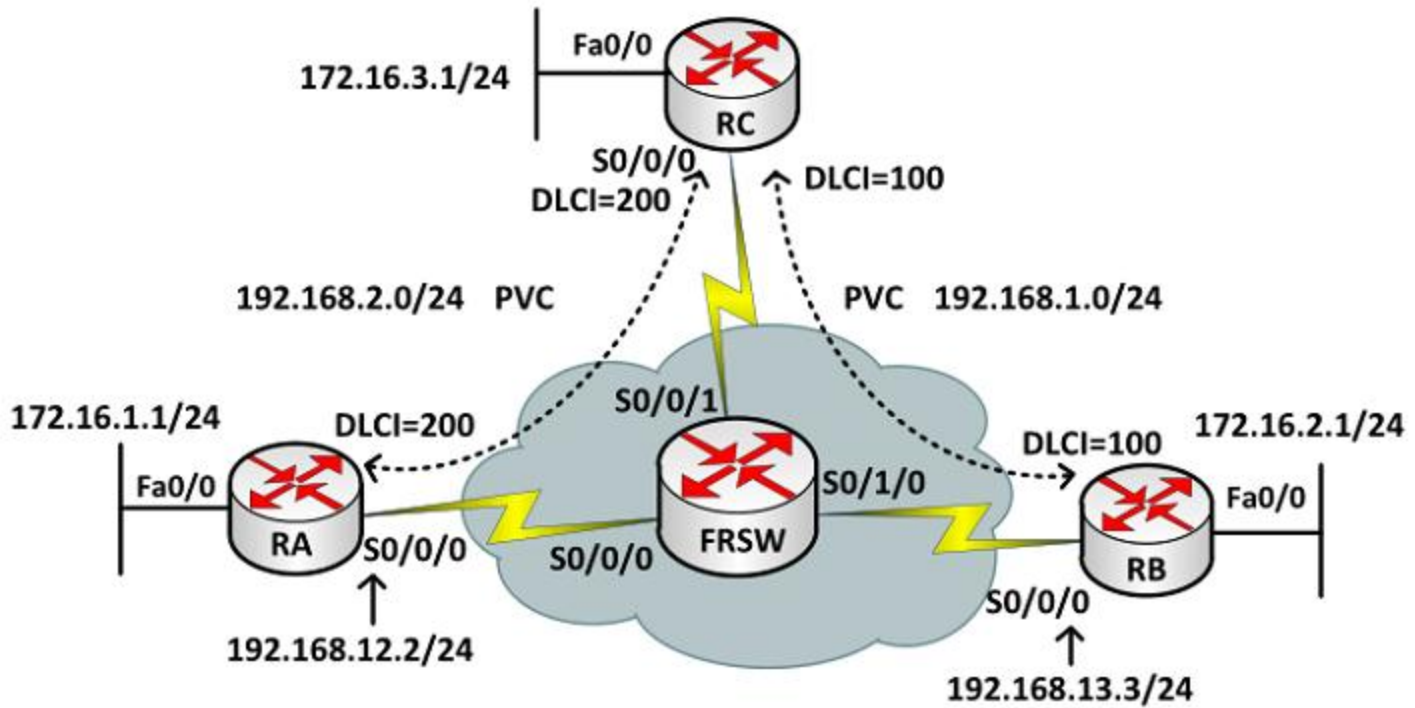
- **Cấu hình Router B:**

```
RB(config)#interface S0/0/0
RB(config-if)#ip address 192.168.1.2 255.255.255.0
RB(config-if)#encapsulation frame-relay
RB(config-if)#no shutdown
RB(config)#interface Fa0/0
RB(config-if)#ip address 172.16.2.1 255.255.255.0
RB(config-if)#no shut
RB(config)#router eigrp 100
RB(config-router)#network 172.16.0.0
RB(config-router)#network 192.168.1.0
```

- **Kiểm tra:** trên Router A và Router B

```
Router#show frame-relay pvc
Router#show frame-relay lmi
Router#show frame-relay route
Router#show frame-relay map
```

Ví dụ 2: Frame-Relay Sub-interface Point-to-Point



- **Yêu cầu**

Cấu hình mạng Frame Relay theo như mô hình trên.

- **Các bước cấu hình**

- Cấu hình Frame Relay Switch

```
FRSW#configure terminal
FRSW(config)#frame-relay switching
FRSW(config)#interface s0/0/0
FRSW(config-if)#no ip address
FRSW(config-if)#encapsulation frame-relay
FRSW(config-if)#clock rate 64000
FRSW(config-if)#frame-relay intf-type dce
FRSW(config-if)#frame-relay route 200 interface
s0/0/1 200
FRSW(config-if)#exit
```

```

FRSW(config)#interface s0/0/1
FRSW(config-if)#no ip address
FRSW(config-if)#encapsulation frame-relay
FRSW(config-if)#clock rate 64000
FRSW(config-if)#frame-relay intf-type dce
FRSW(config-if)#frame-relay route 200 interface
s0/0/0 200
FRSW(config-if)#frame-relay route 100 interface
s0/1/0 100
FRSW(config-if)#exit
FRSW(config)#interface s0/1/0
FRSW(config-if)#no ip address
FRSW(config-if)#encapsulation frame-relay
FRSW(config-if)#clock rate 64000
FRSW(config-if)#frame-relay intf-type dce
FRSW(config-if)#frame-relay route 100 interface
s0/0/1 100

```

- Cấu hình RA:

```

RA(config)#interface s0/0/0
RA(config-if)#no ip address
RA(config-if)#encapsulation frame-relay
RA(config-if)#no shutdown
RA(config-if)#exit
RA(config)#interface serial 0/0/0.1 point-to-point
RA(config-if)#ip address 192.168.1.2 255.255.255.0
RA(config-if)#frame-relay interface dlci 200
RA(config-if)#no shutdown
RA(config-if)#exit
RA(config)#interface Fa0/0
RA(config-if)#ip address 172.16.1.1 255.255.255.0
RA(config-if)#no shut
RA(config-if)#exit

```



```
RA(config)#router eigrp 100
RA(config-router)#network 172.16.0.0
RA(config-router)#network 192.168.1.0
```

- **Cấu hình RB:**

```
RB(config)#interface s0/0/0
RB(config-if)#no ip address
RB(config-if)#encapsulation frame-relay
RB(config-if)#no shutdown
RB(config-if)#exit
RB(config)#interface serial 0/0/0.1 point-to-point
RB(config-subif)#ip address 192.168.2.2
255.255.255.0
RB(config-subif)#frame-relay interface dlci 100
RB(config-subif)#no shutdown
RB(config-subif)#exit
RB(config)#interface Fa0/0
RB(config-if)#ip address 172.16.2.1 255.255.255.0
RB(config-if)#no shut
RB(config-if)#exit
RB(config)#router eigrp 100
RB(config-router)#network 172.16.0.0
RB(config-router)#network 192.168.1.0
```

- **Cấu hình RC**

```
RC(config)#interface serial 0/0/0
RC(config-if)#no ip address
RC(config-if)#encapsulation frame-relay
RC(config-if)#no shutdown
RC(config-if)#exit
RC(config)#interface serial 0/0/0.1 point-to-point
RC(config-subif)#ip address 192.168.1.1
255.255.255.0
RC(config-subif)#frame-relay interface dlci 100
```

```

RC (config-subif) #no shutdown
RC (config-subif) #exit
RC (config) #interface serial 0/0/0.2 point-to-point
RC (config-subif) #ip address 192.168.2.1
255.255.255.0
RC (config-subif) #frame-relay interface dlci 200
RC (config-subif) #no shutdown
RC (config-subif) #exit
RC (config) #interface Fa0/0
RC (config-if) #ip address 172.16.3.1 255.255.255.0
RC (config-if) #no shut
RC (config-if) #exit
RC (config) #router eigrp 100
RC (config-router) #network 172.16.0.0
RC (config-router) #network 192.168.1.0
RC (config-router) #network 192.168.2.0

```

- **Kiểm tra:** trên RA, RB, RC

```

Router#show frame-relay pvc
Router#show frame-relay lmi
Router#show frame-relay route
Router#show frame-relay map

```

4. TỔNG KẾT CHƯƠNG

Các kỹ thuật WAN hoạt động trong phạm vi rộng và phức tạp hơn trong các mạng LAN. Các kỹ thuật WAN được sử dụng phổ biến như: ISDN, leased line, X.25, Frame Relay, ATM, DSL, ... Các kiểu đóng gói thường dùng trong mạng WAN là: HDLC, PPP, Frame Relay, ...

Hai giao thức dùng để chứng thực trên PPP trong môi trường WAN là PAP và CHAP. PAP có độ bảo mật kém vì nó gửi *username/password* dưới dạng không mã hóa và việc chứng thực chỉ diễn ra một lần. Đối với CHAP, tham số chứng thực được gửi đi dưới dạng mã hóa và việc chứng thực được lặp lại trong suốt quá trình kết nối.

5. CÂU HỎI VÀ BÀI TẬP

5.1 CHAP sử dụng thuật toán nào sau đây để tạo giá trị gửi đến cho “remote peer” trong các bước chứng thực.

- A. 3DES
- B. DES
- C. SHA
- D. MD5

5.2 Những kiểu đóng gói ở tầng Data-Link nào sau đây sử dụng trên các công WAN?

- A. Ethernet
- B. PPP
- C. Token Ring
- D. HDLC
- E. Frame Relay
- F. POTS

5.3 Các loại LMI nào sau đây sử dụng trong Frame Relay?

- A. Q.931
- B. IEEE
- C. Cisco
- D. IETF
- E. Q933a
- F. ANSI

5.4 Trong mạng Frame Relay; mục đích của “Inverse ARP” là gì?

- A. Nó được để ánh xạ giữa một địa chỉ IP với địa chỉ MAC
- B. Nó được để ánh xạ giữa một DLCI với địa chỉ MAC
- C. Nó được để ánh xạ giữa một địa chỉ MAC với địa chỉ IP
- D. Nó được để ánh xạ giữa một DLCI với địa chỉ IP
- E. Nó được để ánh xạ giữa một địa chỉ MAC với DLCI

5.5 Sau khi các router được cấu hình Frame Relay, quản trị dùng lệnh *show frame relay map*, kết quả như sau:

```
Router#show frame-relay map
Serial0/0/0 (up): ip 192.168.12.1 dlci 100 (0x64,
0x1840), dynamic
                Broadcast, status defined, active
```

Hãy cho biết ý nghĩa của từ *dynamic* trong kết quả trên

- A. Cổng Serial 0/0/0 đang gửi các gói tin
- B. DLCI 100 được router tự động sinh ra
- C. Cổng Serial 0/0/0 được DHCP server gán IP là 192.168.12.1
- D. DLCI 100 sẽ tự động thay đổi để đáp ứng yêu cầu thay đổi trên mạng Frame Relay
- E. Một ánh xạ giữa DLCI 100 và “remote router” có IP là 192.168.12.1 được học qua “Inverse ARP”

5.6 Cho mô hình mạng sau:



Hai router đã cấu hình chính xác địa chỉ IP như mô tả trên hình và thông tin trên cổng Serial0/0/0 của các router khi thực hiện lệnh *show interface S0/0/0* như sau:

```
R1#show interface s0/0/0
Serial0/0/0 is up, line protocol is down
Hardware is HD64570
Internet address is 192.168.12.1/24
MTU 1500 bytes, BW 1433 Kbit
Reliable 255/255
Encapsulation HDLC, loopback not set
Keepalive set(10sec)
R2#show interface s0/0/0
Serial0/0/0 is up, line protocol is down
```

```
Hareward is HD64570
Internet address is 192.168.12.2/24
MTU 1500 bytes, BW 1433 Kbit
Reliable 255/255
Encapsulation PPP, loopback not set
Keepalive set(10sec)
```

Hai router trên không liên lạc với nhau được. Bạn hãy cho biết nguyên nhân vì sao?

- A. PCP không ở trạng thái “open”
- B. Subnet mask cấu hình sai
- C. Đóng gói không tương thích
- D. Băng thông được cấu hình quá thấp
- E. Địa chỉ IP cấu hình chưa đúng

5.7 Trong một hệ thống mạng cấu hình giao thức PPP và chứng thực bằng CHAP qua kết nối WAN. Lệnh nào sau đây dùng để hiển thị các chứng thực CHAP trong thời gian thực?

- A. show pp authentication
- B. debug pap authentication
- C. debug ppp authentication
- D. show chap authentication

5.8 Cho mô hình mạng sau:



Hai router này đã được cấu hình chứng thực PPP CHAP như sau:

```
Saigon(config)#username Hanoi password spkt@123
Saigon(config)#interface Serial0/0/0
Saigon(config-if)#encapsulation ppp
Saigon(config-if)#ppp authentication chap
```

```
Hanoi (config)#username Saigon password spkt@124
Hanoi (config)#interface Serial0/0/0
Hanoi (config-if)#encapsulation ppp
Hanoi (config-if)#ppp authentication chap
```

Hai router không chứng thực thành công. Hỏi lý do tại sao?

- A. Cấu hình username không đúng trên hai router
- B. Password cấu hình không giống nhau trên hai router
- C. Chứng thực CHAP không thể cấu hình được trên cổng Serial
- D. Các router không thể tạo kết nối được từ cổng Serial 0/0/0 đến Serial 0/0/0
- E. Chứng thực CHAP không cho phép các router chứng thực lẫn nhau

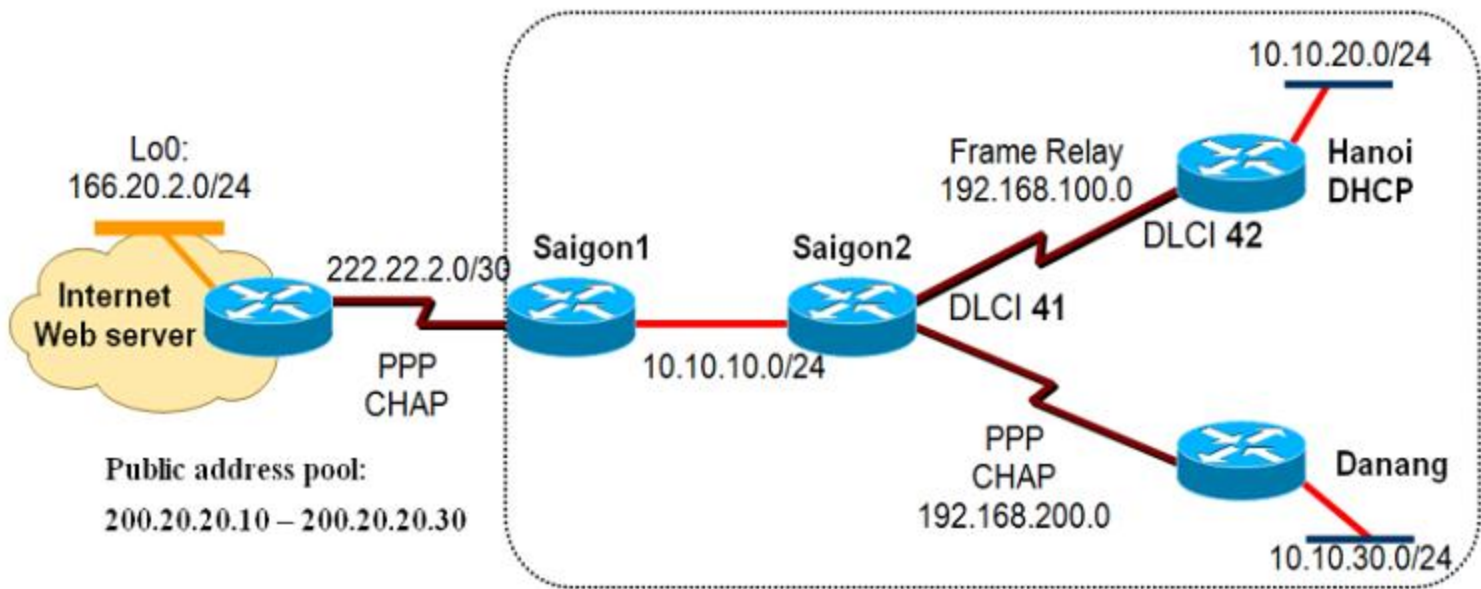
5.9 Mạng frame relay sử dụng DLCI cho mỗi PVC. DLCI có ý nghĩa gì?

- A. Nó được dùng để chọn loại đóng gói trong Frame Relay
- B. Nó dùng để xác định kết nối logic giữa local router và FR switch
- C. Nó đại diện cho địa chỉ vật lý của router.
- D. Nó đại diện cho kết nối của mỗi PVC

5.10 Các lệnh nào sau đây dùng trên cổng WAN nhưng không dùng được trên cổng LAN?

- A. IP address
- B. Encapsulation PPP
- C. No shutdown
- D. PPP authentication CHAP
- E. Speed

5.11 Trong bài tập này, yêu cầu học viên phải hoạch định và gán địa chỉ IP phù hợp các thiết bị, cấu hình định tuyến trên các router, cấu hình Frame Relay, PPP, NAT và DHCP.



Yêu cầu 1: Cấu hình cơ bản trên Router và Switch

- Đặt địa chỉ IP cho các cổng trên Router
- Cấu hình hostname cho các Router
- Cấu hình telnet trên các Router và Switch

Yêu cầu 2: Cấu hình Frame-Relay

- Sử dụng DLCI như trên mô hình
- Xác định kiểu đóng gói Frame Relay là IETF và kiểu LMI là ANSI.
- Giả sử chức năng Inverse-ARP đã bị khóa, cấu hình map tĩnh cho các địa chỉ *remote IP* và *local DLCI*.

Yêu cầu 3: Cấu hình PPP

- Cấu hình đóng gói PPP trên cổng Serial giữa các router như trong mô hình
- Cấu hình chứng thực bằng CHAP giữa 2 router sử dụng password là **spkt**.

Yêu cầu 4: Cấu hình định tuyến

- Cấu hình định tuyến cho hệ thống mạng trên sử dụng giao thức định tuyến tùy chọn.
- Trên Router **Saigon1** quảng bá **default network** cho các router bên trong để cho phép hệ thống mạng bên trong có thể truy cập ra ngoài Internet.
- **Không cấu hình định tuyến trên router ISP**, sử dụng **default route** hoặc **static route** để gửi các gói tin đến nó.

Yêu cầu 5: Cấu hình NAT

- Cấu hình NAT trên router Saigon1 để chuyển đổi các IP bên trong ra IP bên ngoài.

Yêu cầu 6: Cấu hình DHCP

- Cấu hình DHCP trên router Hanoi để cấp IP cho LAN.

Chương 6

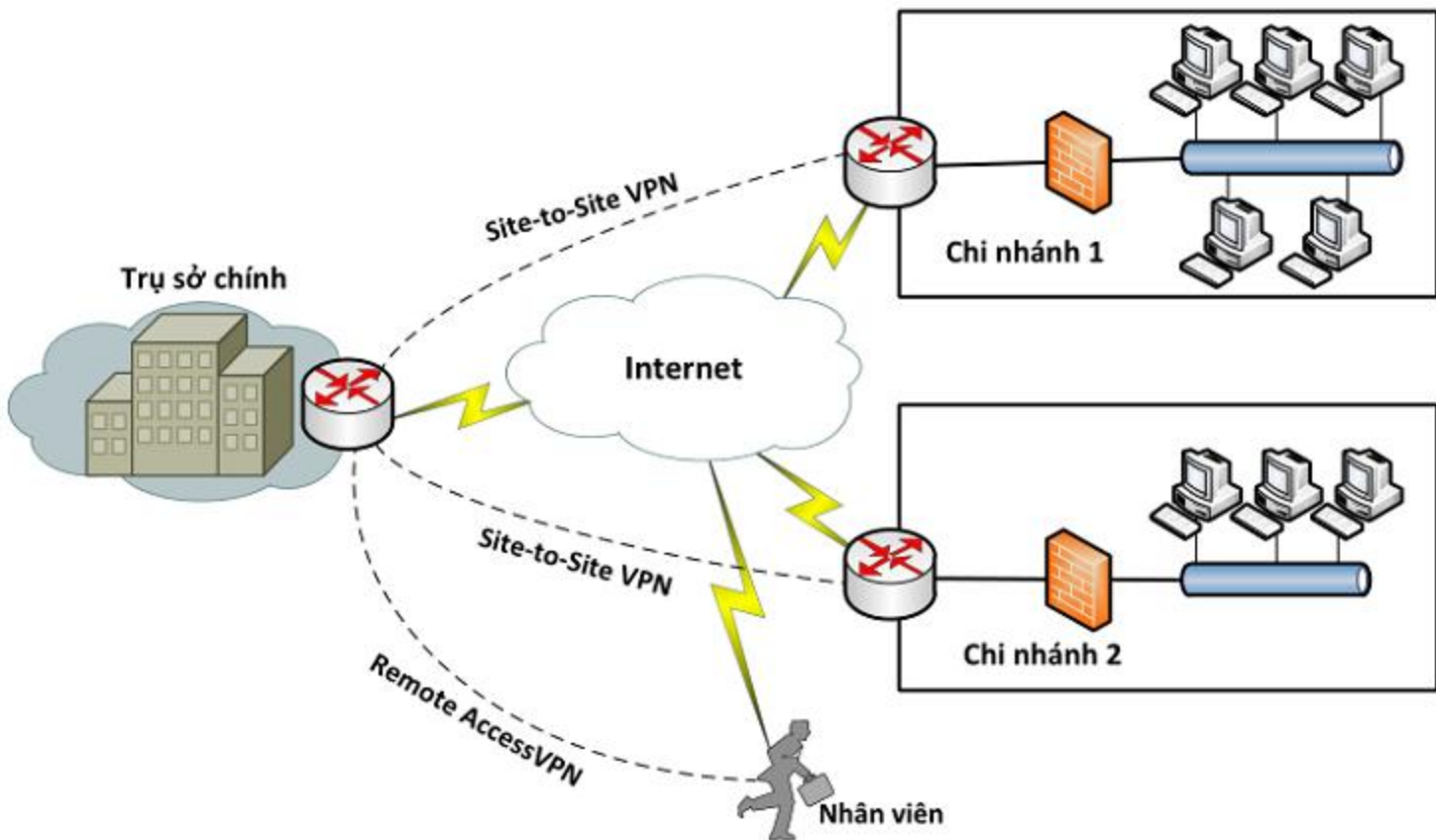
VPN

Trong chương này trình bày tổng quan về mạng riêng ảo VPN (Virtual Private Network). Học xong chương này, người học có khả năng:

- Trình bày được khái niệm VPN
- Trình bày được những lợi ích và đặc điểm của VPN
- Phân biệt được các loại VPN
- Trình bày được các thành phần của VPN

1. GIỚI THIỆU

VPN là sự mở rộng của một mạng riêng (private network) thông qua mạng công cộng (internet), được dùng để kết nối các văn phòng chi nhánh, người từ xa kết nối về văn phòng chính.



Hình 6.1 Mô hình VPN

VPN có thể được tạo ra bằng cách sử dụng phần cứng, phần mềm hay kết hợp cả hai để tạo ra một kết nối ảo bảo mật giữa hai mạng riêng thông qua mạng công cộng. Lợi ích của công nghệ VPN là đáp ứng nhu cầu trao đổi thông tin, truy cập từ xa và tiết kiệm chi phí.

❖ Các mode kết nối VPN

Có hai chế độ kết nối VPN để chuyển dữ liệu giữa hai thiết bị là

- Tunnel mode
- Transport mode

Cả hai mode này định nghĩa quá trình đóng gói được sử dụng để di chuyển dữ liệu một cách an toàn giữa hai thực thể.

• Transport mode

Một kết nối ở *mode transport* được sử dụng địa chỉ IP nguồn và đích thật sự của các thiết bị trong các gói tin để truyền dữ liệu.

• Tunnel mode

Hạn chế của *transport mode* là không có khả năng mở rộng. Do đó, nếu chúng ta có nhiều thiết bị ở hai vị trí riêng biệt cần nói chuyện với nhau trong chế độ bảo mật, ta nên sử dụng *tunnel mode* thay vì *transport-mode*. Trong *tunnel mode*, các thiết bị nguồn-đích thực thông thường sẽ không bảo vệ dữ liệu, thay vào đó các thiết bị trung gian được sử dụng để bảo vệ luồng dữ liệu. Các thiết bị này được gọi là các *VPN gateway*.

Tunnel mode cung cấp nhiều tính năng ưu việt hơn *transport mode*:

- *Tính mở rộng*: ta có thể chọn một thiết bị phù hợp để thực hiện việc xử lý bảo vệ.
- *Tính linh động*: không cần phải thay đổi gì trong cấu hình VPN khi thêm vào một thiết bị mới sau *VPN Gateway*.
- *Tính ẩn của các giao tiếp*: các lưu lượng được các *VPN Gateway* đại diện trao đổi với nhau, vì vậy sẽ che dấu nguồn và đích thật sự của kết nối.
- *Sử dụng địa chỉ cục bộ*: các thiết bị đích và nguồn thực có thể sử dụng địa chỉ được đăng ký (public) hay cục bộ bởi vì các gói tin được đóng gói bởi các *VPN Gateway*.
- Sử dụng các chính sách bảo mật hiện có: các chính sách bảo mật được thực hiện trên các thiết bị tường lửa và bộ lọc gói tin.

2. CÁC THÀNH PHẦN CỦA VPN

❖ Chứng thực

Có 2 loại chứng thực là: Chứng thực thiết bị và chứng thực người dùng

- Chứng thực thiết bị: cho phép hạn chế các truy cập vào hệ thống mạng dựa vào các thông tin cung cấp bởi các thiết bị VPN đầu xa.

Có 2 dạng chứng thực kiểu này là: *Pre-shared key*, *Digital signature* hoặc *certificate*.

Pre-shared key được sử dụng trong các môi trường VPN nhỏ. Một hay nhiều khóa được cấu hình và dùng để chứng thực để nhận dạng một thiết bị. *Digital signature*, *digital certificate* được sử dụng để chứng thực thiết bị trong các môi trường triển khai VPN lớn.

- Chứng thực người dùng: chỉ cho phép người dùng hợp lệ kết nối và truy cập hệ thống VPN. Người dùng phải cung cấp *username* và *password*.

❖ Phương pháp đóng gói

Làm thế nào mà thông tin người dùng, dữ liệu được đóng gói và vận chuyển qua mạng. Các câu hỏi cần đặt ra là: Các trường (field) gì sẽ tồn tại trong VPN header và VPN trailer, thứ tự xuất hiện các trường, kích thước của các trường?

❖ Mã hóa dữ liệu

Mã hóa dữ liệu giải quyết vấn đề dữ liệu bị đánh cắp trên đường truyền. Mã hóa dữ liệu chỉ đơn giản là lấy dữ liệu, một giá trị khóa và chạy thuật toán mã hóa để làm cho dữ liệu trở thành dạng khác với nội dung ban đầu. Chỉ có thiết bị có cùng khóa mới có thể giải mã được thông tin về dạng ban đầu. Một số thuật toán mã hóa như *DES*, *3DES*, *RSA*, *AES*, *RC4*...

❖ Toàn vẹn dữ liệu

Có thể xảy ra tình trạng có các gói tin giả làm tăng sự hoạt động lãng phí của CPU. VPN cung cấp một cơ chế để khắc phục là kiểm tra sự toàn vẹn của dữ liệu, hay còn gọi là “packet authentication”. Với “packet authentication”, một chữ ký (signature) được đóng vào các gói tin. *Signature* được tạo ra bằng cách lấy nội dung từ gói tin, một “share-key” và chạy thông tin này qua một hàm băm và xuất ra một giá trị gọi là *digital signature*. *Signature* này được thêm vào các gói tin và gửi đi đến đích. Ở đích đến sẽ kiểm tra “signature”, và nếu “signature” được kiểm tra là chính xác, nó sẽ giải mã nội dung gói tin.

Hai trong số các hàm băm được sử dụng cho việc kiểm tra toàn vẹn dữ liệu là *SHA* và *MD5*.

❖ Quản lý khóa

Chúng ta đã đề cập đến 3 thành phần VPN có sử dụng khóa là: *chứng thực*, *mã hóa* và *hàm băm*. Việc quản lý khóa trở nên quan trọng trong các kết nối VPN. Ví dụ như: làm thế nào để phân phối các khóa,

chúng được cấu hình tĩnh hay phát sinh ngẫu nhiên, các khóa được tạo lại bao nhiêu lần để tăng tính bảo mật?

❖ **Non-repudiation**

Repudiation là nơi ta không thể chứng thực các giao tiếp xảy ra (như là việc thiết lập kết nối). *Non-repudiation* trái ngược với điều này: ta có thể chứng thực một giao tiếp xảy ra giữa hai bên kết nối.

Ví dụ khi ta vào một cửa hàng online như Amazone.com và mua một quyển sách và thanh toán bằng credit card. Amazone sẽ phải thu gom thông tin cá nhân khi ta điền vào đơn đặt hàng như tên, địa chỉ, số điện thoại, thông tin về credit card... Khi đó, Amazone sẽ kiểm tra những thông tin đó với công ty phân phối *credit card* và lưu giữ lại các thông tin giao dịch như ngày, tháng, ...

❖ **Hỗ trợ ứng dụng và giao thức**

Khi lựa chọn cài đặt VPN, đầu tiên chúng ta cần phải xác định loại dữ liệu nào cần được bảo vệ. Ví dụ như loại dữ liệu IP hay IPX hoặc cả hai, hoặc là chỉ cần bảo vệ một số loại dữ liệu cho một số chương trình ứng dụng nào đó như Web hay Email,...

❖ **Quản lý địa chỉ:**

Quản lý địa chỉ là một vấn đề quan trọng trong việc hoạch định địa chỉ cho toàn hệ thống mạng của công ty.

3. CÁC LOẠI VPN

Có 2 loại VPN thông dụng:

- *Site-to-Site VPN*
- *Remote Access VPN*

Một số giao thức được sử dụng trong VPN: PPTP, L2TP, IPSec,...

❖ **Remote Access VPN**

Remote access VPN thường được sử dụng cho các kết nối có băng thông thấp giữa một thiết bị của người dùng như là PC, Ipad,... và một thiết bị *Gateway VPN*. *Remote access VPN* thông thường sử dụng *tunnel mode* cho các kết nối.

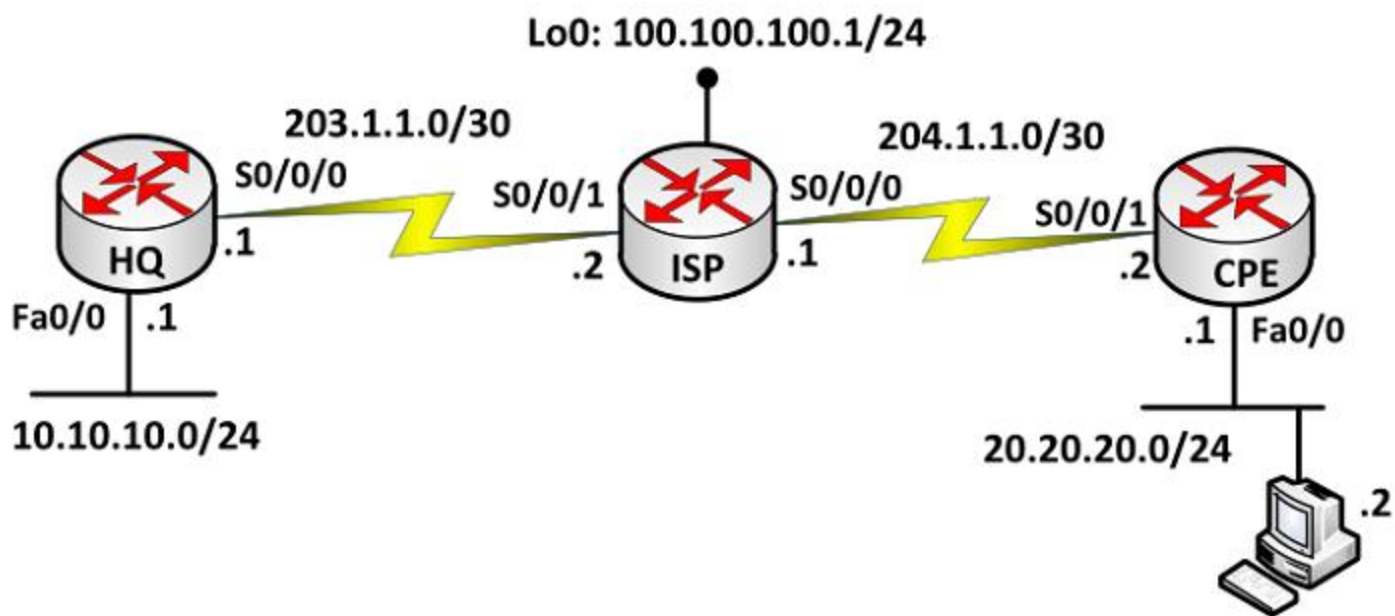
Người dùng ở xa sử dụng các phần mềm VPN để truy cập vào mạng của công ty thông qua Gateway hoặc VPN concentrator (bản chất là một server), giải pháp này thường được gọi là client/server. Trong giải

pháp này, người dùng thường sử dụng các công nghệ truyền thống để tạo lại các tunnel về mạng của họ.

Một phần quan trọng của thiết kế này là việc thiết kế quá trình xác thực ban đầu nhằm đảm bảo là yêu cầu được xuất phát từ một nguồn tin cậy. Thường thì giai đoạn ban đầu này dựa trên cùng một chính sách về bảo mật của công ty.

Trong Remote Access VPN có nhiều kỹ thuật được sử dụng để bảo mật trong việc trao đổi dữ liệu: IPSec, SSL,...

Ví dụ 1: Cấu hình “remote access VPN” (IPSec)



Hướng dẫn cấu hình

```
HQ(config)#aaa new-model
HQ(config)#username cisco privilege 15 password cisco
HQ(config)#aaa authentication login default local
HQ(config)#aaa authentication login my_authen local
HQ(config)#aaa authorization network my_autho local
```

- ISAKMP phase 1:

```
HQ(config)#crypto isakmp policy 1
HQ(config-isakmp)#encryption 3des
HQ(config-isakmp)#hash sha
HQ(config-isakmp)#group 2
HQ(config-isakmp)#authentication pre-shared
```

```
HQ(config)#crypto isakmp client configuration group
my_VPN
```

```
HQ(config-isakmp-group)#key cisco
HQ(config-isakmp-group)#pool VPN_pool
```

```
HQ(config)#ip local pool VPN_pool 10.10.10.2
10.10.10.50
```

- **Cấu hình phase 2:**

```
HQ(config)#crypto ipsec transform-set my_set esp-3des
esp-sha-hmac
```

```
HQ(config)#crypto dynamic-map my_dyn 1
```

```
HQ(config-crypto-map)#set transform-set my_set
```

```
HQ(config-crypto-map)#reverse route
```

- **Cấu hình phase 1.5**

```
HQ(config)#crypto map my_map isakmp authorization list
my_autho
```

```
HQ(config)#crypto map my_map client authentication
list my_authen
```

```
HQ(config)#crypto map my_map client configuration
address respond
```

```
HQ(config)#crypto map my_map 20 ipsec-isakmp dynamic
my_dyn
```

Vào cổng kết nối đến ISP:

```
HQ(config-if)#crypto map my_map
```

PC:

Dùng phần mềm VPN Client để kết nối với HQ

Khai báo:

Group: my_VPN

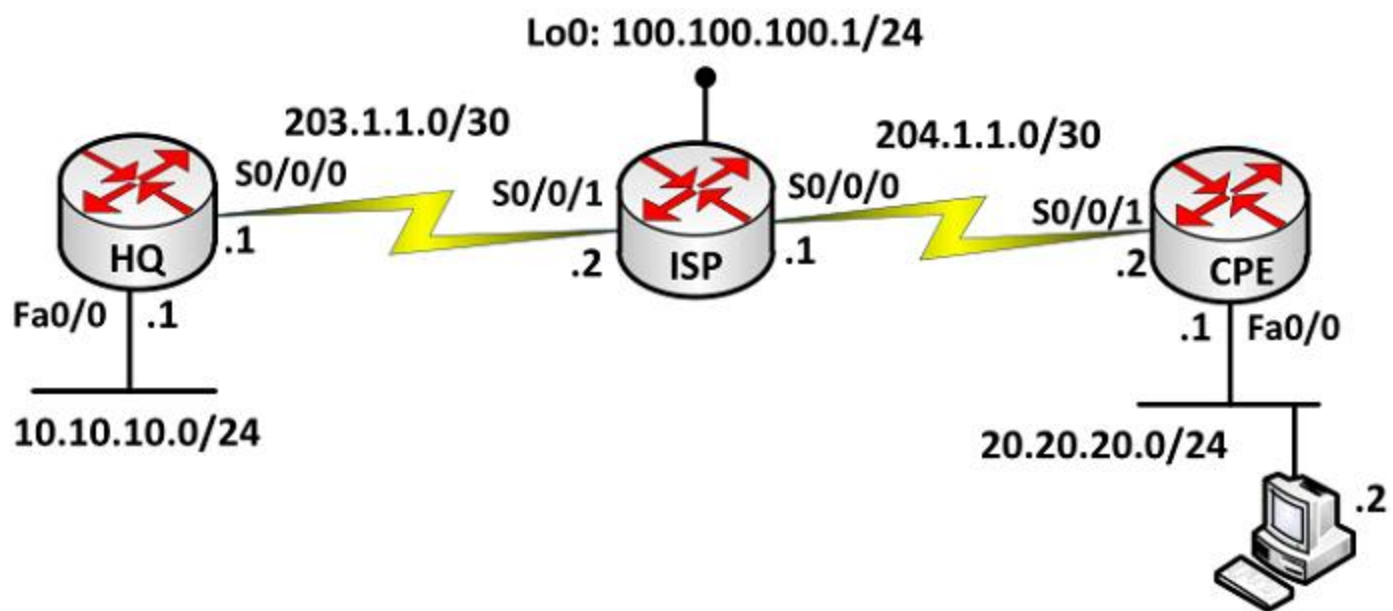
Password: cisco (key trong my_VPN)

Kết nối tới VPN HQ: xuất hiện popup window để nhập username/password

Username: cisco

Password: cisco

Ví dụ 2: Ở ví dụ 1, chúng ta dùng phần mềm trên PC để kết nối VPN. Trong bài này, chúng ta sẽ dùng router kết nối VPN về HQ.



- **Hướng dẫn cấu hình**

Cấu hình HQ:

Giống như bài trước, có thêm các câu lệnh sau:

```
HQ(config)#crypto isakmp client configuration
group my_VPN
HQ(config-isakmp-group)#save-password
```

Cấu hình cho Remote:

```
Remote(config)#crypto ipsec client ezvpn
my_remote
Remote(config-ipsec)#group my_VPN key cisco
Remote(config-ipsec)#connect manual
Remote(config-ipsec)#peer 203.1.1.1
Remote(config-ipsec)#mode client
Remote(config-ipsec)#username cisco password
cisco
Remote(config-ipsec)#xauth userid mode local
```

- **Áp đặt lên cổng:**

- **Cổng nối ra Internet**

```
Remote(config-if)#crypto ipsec client ezvpn
my_remote
```

- **Cổng nối vào LAN**

```
Remote(config-if)#crypto ipsec client ezvpn
my_remote inside
```

- **Kiểm tra cấu hình**

- Thực hiện lệnh sau trên Remote để connect

```
Remote#crypto ipsec client ezvpn connect
```

Lưu ý: Trên Remote tự động thực hiện NAT

```
Remote#show ip nat translation
```

- Disconnect VPN bằng 2 cách:

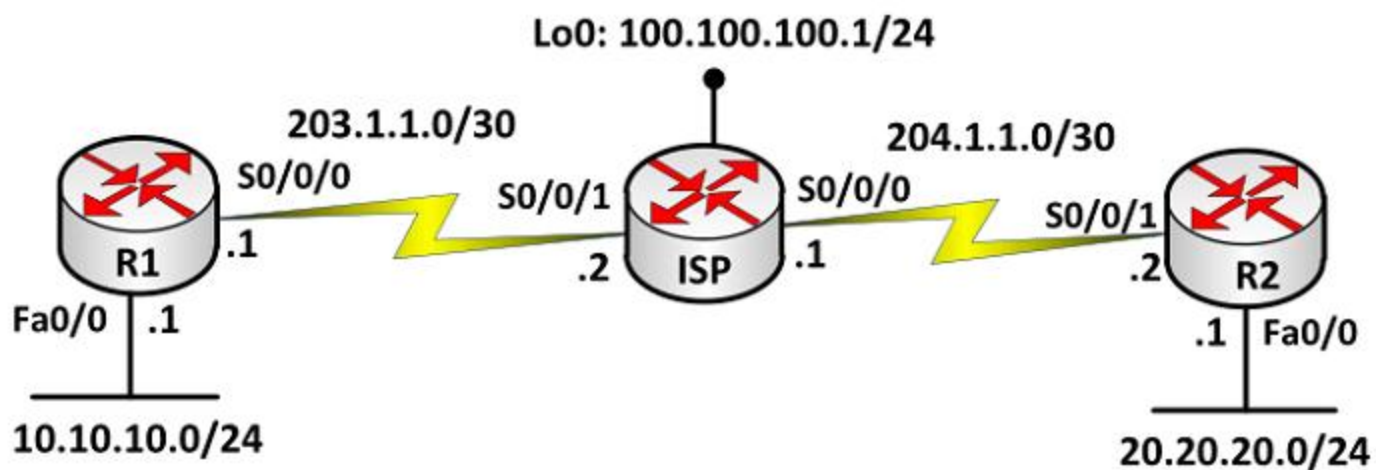
```
Remote#clear crypto sa
```

```
Remote#clear crypto session
```

- ❖ **Site-to-Site VPN**

Site-to-site VPN (LAN-to-LAN) là kỹ thuật kết nối các hệ thống mạng (site) của cùng một công ty ở các nơi khác nhau tạo thành một hệ thống mạng thống nhất thông qua môi trường mạng công cộng. Trong trường hợp này, quá trình xác thực ban đầu cho những người dùng cần phải được kiểm soát chặt chẽ bởi các thiết bị ở các site tương ứng. Các thiết bị này hoạt động như *Gateway*, truyền lưu lượng một cách an toàn cho đầu bên kia.

Ví dụ 1: Cấu hình VPN site-to-site giữa R1 và R2 (IPSec).



Cấu hình cơ bản: cấu hình hostname và địa chỉ IP cho các router theo mô hình

```
R1(config)#ip route 0.0.0.0 0.0.0.0 203.1.1.2
```

```
R2(config)#ip route 0.0.0.0 0.0.0.0 204.1.1.1
```

Cấu hình VPN site-to-site (R1 và R2)

Phase 1:

```
Router(config)#crypto isakmp policy 1
```

```
Router(config-isakmp)#encryption 3des
```

```
Router(config-isakmp)#hash sha
```



```
Router(config-isakmp)#authentication pre-shared
Router(config-isakmp)#group 2
Router(config)#crypto isakmp key cisco address
A.B.C.D → địa chỉ IP của Router bên site kia.
```

Phase 2:

```
Router(config)#crypto ipsec transform-set myset
esp-3des esp-sha-hmac
Router(config-crypto-transform-set)#exit | mode
transport //(mode tunnel default)
Router(config)#access-list 100 permit ip
10.10.10.0 0.0.0.255 20.20.20.0 0.0.0.255
Router(config)#crypto map mymap 1 ipsec-isakmp
Router(config-crypto)#match address 100
Router(config-crypto)#set transform-set myset
Router(config-crypto)#set peer A.B.C.D
```

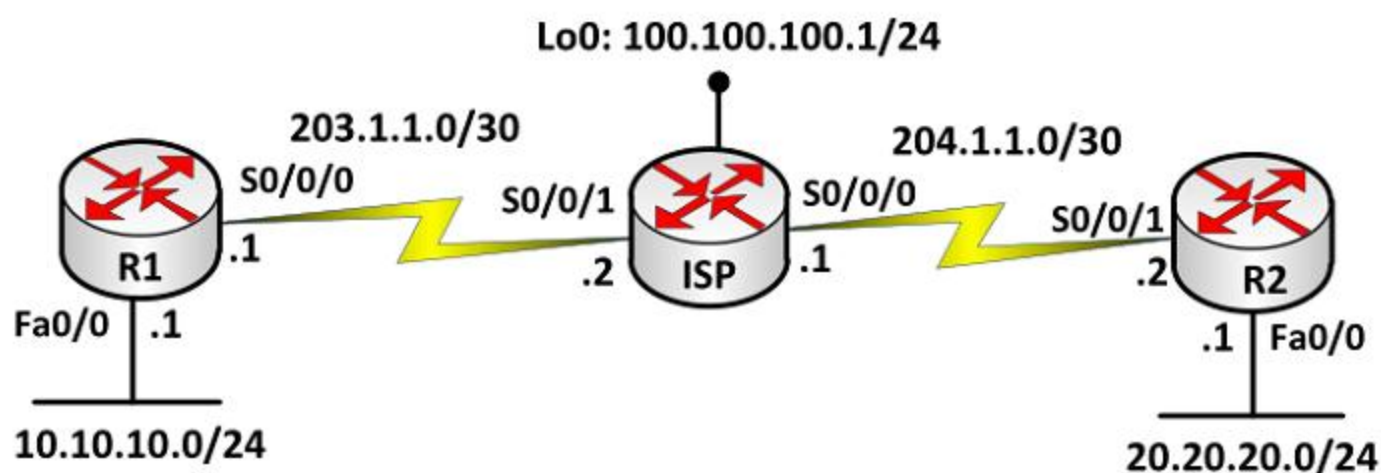
Gắn vào cổng:

```
Router(config-if)#crypto map mymap
```

Kiểm tra cấu hình:

```
Router#show crypto isakmp policy
Router#show crypto ipsec sa
```

Ví dụ 2: VPN kết hợp với NAT



Yêu cầu: Cấu hình cho 2 mạng riêng liên lạc bằng VPN. Các mạng còn lại sẽ đi bằng NAT overload.

Hướng dẫn cấu hình

```
R1(config)#access-list 101 deny ip 10.10.10.0
0.0.0.255 20.20.20.0 0.0.0.255
```

```
R1(config)#access-list 101 permit ip 10.10.10.0  
0.0.0.255 any
```

```
R2(config)#access-list 101 deny ip 20.20.20.0  
0.0.0.255 10.10.10.0 0.0.0.255
```

```
R2(config)#access-list 101 permit ip 20.20.20.0  
0.0.0.255 any
```

4. TỔNG KẾT CHƯƠNG

Công nghệ VPN được sử dụng phổ biến hiện nay. Nó cung cấp kết nối an toàn và hiệu quả để truy cập vào tài nguyên nội bộ từ nhân viên ở bên ngoài thông qua Internet vào hệ thống mạng công ty hay kết nối các chi nhánh của công ty để trao đổi dữ liệu với nhau.

Có nhiều cách phân loại VPN. Trong giáo trình này trình bày 2 loại VPN thông dụng: *Remote access VPN*, *Site-to-Site VPN*.

5. CÂU HỎI VÀ BÀI TẬP

5.1 Hai mode cơ bản được sử dụng trong việc chuyển dữ liệu trong VPN là gì?

- A. Tunnel mode
- B. Server mode
- C. Transport mode
- D. Transparent mode
- E. Client mode

5.2 Những tính năng nào sau đây thể hiện những ưu điểm của tunnel mode so với transport mode?

- A. Khả năng mở rộng
- B. Che dấu các giao tiếp
- C. Chuyển dữ liệu nhanh
- D. Hỗ trợ VLAN

5.3 Loại VPN nào sử dụng tunnel mode kết nối giữa hai VPN gateway để bảo vệ luồng dữ liệu giữa 2 site?.

- A. Remote Access VPN
- B. Site – to – Site VPN
- C. Firewall VPN
- D. User – to – User VPN

5.4 Loại VPN nào được sử dụng để kết nối các nhân viên bên ngoài vào hệ thống mạng nội bộ của công ty?

- A. Remote Access VPN
- B. Site – to – Site VPN
- C. Firewall VPN
- D. User – to – User VPN

TÀI LIỆU THAM KHẢO

- [1] Wendell Odom, CCIE No.1624 – *CCENT/CCNA ICND1 Official Exam Certification Guide*, Second Edition - Cisco Press, 2008
- [2] Wendell Odom, CCIE No.1624 – *CCNA ICND2 Exam Certification Guide*, Second Edition - Cisco Press, 2008
- [3] Todd Lammle, *CCNA - Cisco Certified Network Associate Study Guide*, Six Edition -Sybex Press, 2007
- [4] Richard Deal, *The Complete Cisco VPN Configuration Guide*, Cisco Press, 2005

Giáo trình
MẠNG MÁY TÍNH NÂNG CAO
ThS. Huỳnh Nguyên Chính

Nhà xuất bản ĐHQG-HCM và tác giả/đồng tác liên kết giữ bản quyền ©
Copyright © by VNU-HCM Publishing House and author/co-partnership
All rights reserved

NHÀ XUẤT BẢN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
Khu Phố 6, Phường Linh Trung, Quận Thủ Đức, TPHCM
Số 3, Công trường Quốc tế, Quận 3, TP Hồ Chí Minh
ĐT: 38239171 – 38225227 - 38239172
Fax: 38239172 - Email: vnuhp@vnuhcm.edu.vn

PHÒNG PHÁT HÀNH NHÀ XUẤT BẢN
ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
Số 3 Công trường Quốc tế - Quận 3 – TPHCM
ĐT: 38239170 – 0982920509 – 0913943466
Fax: 38239172 – Website: www.nxbdhqghcm.edu.vn

Chịu trách nhiệm xuất bản:
NGUYỄN HOÀNG DŨNG

Chịu trách nhiệm nội dung:
HUỲNH BÁ LÂN

Tổ chức bản thảo và chịu trách nhiệm về tác quyền
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

Biên tập:
THÙY DƯƠNG

Sửa bản in:
PHẠM THỊ BÌNH

Trình bày bìa
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM

Mã số ISBN: 978-604-73-1689-2

Số lượng 300 cuốn; khổ 16 x 24cm.
Số đăng ký kế hoạch xuất bản: 126-2013/CXB/164-07/ĐHQGTPHCM.
Quyết định xuất bản số: 144 ngày 25/07/2013 của NXB ĐHQGTPHCM.
In tại Công ty TNHH In và Bao bì Hưng Phú. Nộp lưu chiểu quý III năm 2013.