CCNA – KỲ 2

KHÁI NIỆM & GIAO THỨC ĐỊNH TUYẾN

# ****MỤC LỤC****

[MỤC LỤC 2](#_Toc310282170)

[CHƯƠNG 1 - GIỚI THIỆU VỀ ĐỊNH TUYẾN VÀ CHUYỂN TIẾP GÓI 5](#_Toc310282171)

[I. Cấu tạo của Router 6](#_Toc310282172)

[1. Các thành phần cấu tạo bên trong Router 7](#_Toc310282173)

[2. Hệ điều hành: 10](#_Toc310282174)

[3. Các cổng giao diện của Router 10](#_Toc310282175)

[4. Quá trình khởi động Router 15](#_Toc310282176)

[5. Router và mô hình OSI 17](#_Toc310282177)

[II. Cấu trúc giao diện dòng lệnh CLI 20](#_Toc310282178)

[1. Các yếu tố giao diện dòng lệnh CLI 20](#_Toc310282179)

[2. CLI Help 20](#_Toc310282180)

[3. Hotkey 22](#_Toc310282181)

[4. Các câu lệnh show kiểm tra cac vùng bộ nhớ 22](#_Toc310282182)

[III. Cấu hình cơ bản 23](#_Toc310282183)

[IV. Tổng kết 26](#_Toc310282184)

[CHƯƠNG 2 - ĐỊNH TUYẾN TĨNH 28](#_Toc310282185)

[I. Mạng kết nối trực tiếp 28](#_Toc310282186)

[1. Giao diện Ethernet 30](#_Toc310282187)

[2. Giao diện Serial 31](#_Toc310282188)

[II. Mạng kết nối trực tiếp 34](#_Toc310282189)

[1. Kết nối trực tiếp 34](#_Toc310282190)

[2. Giao thức CDP 38](#_Toc310282191)

[III. Định tuyến tĩnh 40](#_Toc310282192)

[1. Mục đích của định tuyến tĩnh 40](#_Toc310282193)

[2. Cấu hình định tuyến tĩnh 41](#_Toc310282194)

[IV. Ứng dụng định tuyến tĩnh 51](#_Toc310282195)

[1. Tuyến đường mặc định (default route) 51](#_Toc310282196)

[2. Tuyến đường tổng hợp (summary route) 52](#_Toc310282197)

[V. Tổng kết 55](#_Toc310282198)

[CHƯƠNG 3 - ĐỊNH TUYẾN ĐỘNG 56](#_Toc310282199)

[I. Sơ lược về định tuyến động 56](#_Toc310282200)

[1. Lịch sử các giao thức định tuyến động 56](#_Toc310282201)

[2. Định tuyến động vs Định tuyến tĩnh 58](#_Toc310282202)

[3. Mục đích của giao thức định tuyến động 59](#_Toc310282203)

[II. Phân loại giao thức định tuyến động 61](#_Toc310282204)

[1. Phân loại dựa trên miền quản trị – IGP vs EGP 62](#_Toc310282205)

[2. Phân loại dựa trên cách thức cập nhật thông tin định tuyến – Distance-vector vs Link-State 65](#_Toc310282206)

[3. Phân loại dựa trên thông tin cập nhật mạng đích – Classful vs Classless 68](#_Toc310282207)

[4. Hội tụ - Convergence 70](#_Toc310282208)

[III. Lựa chọn tuyến đường tối ưu 72](#_Toc310282209)

[1. Chỉ số ưu tiên AD – Administrator Distance 72](#_Toc310282210)

[2. Metric – Thước đo tuyến đường 76](#_Toc310282211)

[3. Cân bằng tải (Load-Balancing) 77](#_Toc310282212)

[IV. Tổng kết: 81](#_Toc310282213)

[CHƯƠNG 4 - GIAO THỨC ĐỊNH TUYẾN DISTANCE-VECTOR 82](#_Toc310282214)

[I. Giới thiệu giao thức định tuyến Distance-vector 82](#_Toc310282215)

[1. Các đặc điểm cập nhật (update) của giao thức Distance-vector 83](#_Toc310282216)

[2. Ưu điểm và nhược điểm của giao thức Distance-vector 83](#_Toc310282217)

[II. Học định tuyến 84](#_Toc310282218)

[1. Cold Start - Khởi động nguội 84](#_Toc310282219)

[2. Bắt đầu trao đổi bản tin định tuyến 85](#_Toc310282220)

[3. Cập nhật theo tin đồn (Update by rumor) 86](#_Toc310282221)

[4. Nhận xét về tốc độ và khả năng hội tụ: 88](#_Toc310282222)

[III. Duy trì bảng định tuyến 89](#_Toc310282223)

[1. Cập nhật định kỳ - Periodic Update 89](#_Toc310282224)

[2. Cập nhật hạn chế - Bounded Update 90](#_Toc310282225)

[3. Cập nhật kích hoạt – Triggered Update 92](#_Toc310282226)

[4. Cập nhật đồng bộ - Synchroniaztion Update 93](#_Toc310282227)

[IV. Loop định tuyến – Routing loop 94](#_Toc310282228)

[1. Giới thiệu về loop định tuyến 95](#_Toc310282229)

[2. Ảnh hưởng của loop định tuyến 95](#_Toc310282230)

[3. Giải pháp thứ nhất – Giới hạn hop-count 96](#_Toc310282231)

[4. Giải pháp thứ hai – Holddown timer 100](#_Toc310282232)

[5. Giải pháp thứ ba – Split horizon kết hợp Route poisoning 103](#_Toc310282233)

[6. Giải pháp thứ tư – Poison Reverse 107](#_Toc310282234)

[V. Tổng kết 109](#_Toc310282235)

[CHƯƠNG 5 - RIPv1 110](#_Toc310282236)

[I. Giới thiệu giao thức RIP 110](#_Toc310282237)

[1. Lịch sử phát triển 110](#_Toc310282238)

[2. Đặc điểm của RIPv1 111](#_Toc310282239)

[3. Các thông số thời gian của RIP 111](#_Toc310282240)

[II. Vận hành của RIPv1 112](#_Toc310282241)

[1. Cấu trúc bản tin RIP 112](#_Toc310282242)

[2. Vận hành RIPv1 113](#_Toc310282243)

[III. Cấu hình RIPv1 115](#_Toc310282244)

[1. Quảng bá 1 mạng vào RIP 116](#_Toc310282245)

[2. Cấu hình passive-interface 122](#_Toc310282246)

[3. Quá trình auto-summary 124](#_Toc310282247)

[4. Vấn đề mạng không liên tục – Discontigous network 131](#_Toc310282248)

[5. Cấu hình default route 137](#_Toc310282249)

[Tổng kết 140](#_Toc310282250)

[CHƯƠNG 6 - GIAO THỨC ĐỊNH TUYẾN KHÔNG PHÂN LỚP. VLSM VÀ CIDR. 140](#_Toc310282251)

[I. Classful vs Classless 141](#_Toc310282252)

[1. Classful và Classless Addressing 141](#_Toc310282253)

[2. Classful Routing 141](#_Toc310282254)

[3. Classless Routing 144](#_Toc310282255)

[II. VLSM và CIDR 145](#_Toc310282256)

[1. VLSM 145](#_Toc310282257)

[2. CIDR 146](#_Toc310282258)

[CHƯƠNG 7 - RIPv2 148](#_Toc310282259)

[I. Các tính năng RIPv2 148](#_Toc310282260)

[1. Đặc điểm chung 148](#_Toc310282261)

[2. Bản tin RIPv2 149](#_Toc310282262)

[3. Các cải tiến so với RIPv1 151](#_Toc310282263)

[4. Gộp tuyến thủ cong – Manual Summarization 152](#_Toc310282264)

[5. Hỗ trợ xác thực 152](#_Toc310282265)

[II. Cấu hình RIPv2 152](#_Toc310282266)

[1. Chạy RIPv2 155](#_Toc310282267)

[2. Tắt auto-summary 157](#_Toc310282268)

[3. RIPv2 và CIDR – redistribution 160](#_Toc310282269)

[CHƯƠNG 8 - CHI TIẾT BẢNG ĐỊNH TUYẾN 163](#_Toc310282270)

[I. Cấu trúc bảng định tuyến chạy Classful Addressing 163](#_Toc310282271)

[1. Các bản ghi (entry) 164](#_Toc310282272)

[2. Cấp độ tuyến đường 164](#_Toc310282273)

[II. Cấu trúc bảng định tuyến chạy Classless Addressing 169](#_Toc310282274)

[III. Quá trình tra cứu bảng định tuyến 171](#_Toc310282275)

[1. Thứ tự tra cứu bảng định tuyến: 173](#_Toc310282276)

[2. Tuyến phù hợp nhất – So sánh chiều dài khớp (longest match) 175](#_Toc310282277)

[3. Tra cứu bảng định tuyến, khớp hay không khớp 179](#_Toc310282278)

[CHƯƠNG 9 - GIAO THỨC ĐỊNH TUYẾN LINK-STATE 186](#_Toc310282279)

[I. Giới thiệu giao thức Link-State 186](#_Toc310282280)

[II. Các tiến trình được sử dụng bởi giao thức Link-State 188](#_Toc310282281)

[1. Học từ kết nối trực tiếp 189](#_Toc310282282)

[2. Thiết lập các quan hệ cần thiết 190](#_Toc310282283)

[3. Xây dựng bản tin Link-State (Link-state packet) 191](#_Toc310282284)

[4. Flood bản tin update cho tất cả hàng xóm 192](#_Toc310282285)

[5. Xây dựng bảng cơ sở dữ liệu và tính toán tuyến đường tốt nhất 193](#_Toc310282286)

[6. Tổng kết 195](#_Toc310282287)

[III. Triển khai giao thức Link-State 197](#_Toc310282288)

[I. Yêu cầu phần cứng 197](#_Toc310282289)

[II. Yêu cầu thiết kế 197](#_Toc310282290)

[III. Lựa chọn 1 giao thức Link-State 198](#_Toc310282291)

[IV. Tổng kết 199](#_Toc310282292)

[CHƯƠNG 10 - OSPF 200](#_Toc310282293)

[I. Giới thiệu giao thức OSPF 200](#_Toc310282294)

[1. Lịch sử phát triển 200](#_Toc310282295)

[2. Các đặc điểm cơ bản của OSPF 201](#_Toc310282296)

[II. Vận hành của giao thức OSPF 201](#_Toc310282297)

[1. Các tiến trình vận hành của OSPF 201](#_Toc310282298)

[2. Các bản tin OSPF 202](#_Toc310282299)

[3. Giao thức Hello 205](#_Toc310282300)

[4. Hàng xóm và quan hệ Adjacency 207](#_Toc310282301)

[5. Bình bầu DR/BDR 209](#_Toc310282302)

[6. OSPF Metric 216](#_Toc310282303)

[7. OSPF và các loại mạng 221](#_Toc310282304)

[III. Cấu hình OSPF 221](#_Toc310282305)

[1. Quảng bá một mạng vào OSPF 221](#_Toc310282306)

[2. Router-ID 228](#_Toc310282307)

[3. Điều chỉnh metric 232](#_Toc310282308)

[4. Điều chỉnh bình bầu DR/BDR 236](#_Toc310282309)

[5. Default route 243](#_Toc310282310)

[6. Điều chỉnh OSPF timer 247](#_Toc310282311)

[IV. Tổng kết 248](#_Toc310282312)

# CHƯƠNG 1 - GIỚI THIỆU VỀ ĐỊNH TUYẾN VÀ CHUYỂN TIẾP GÓI

Mạng máy tính ngày nay có vai trò quan trọng đối với cuộc sống của chúng ta, ảnh hưởng cách con người giao tiếp, làm việc và giải trí. Mạng máy tính, hay mở rộng ra là Internet đưa ra 1 cách thức tương tác hiệu quả hơn và thông minh hơn. Mạng máy tính có thể triển khai ở khắp mọi nơi và con người sử dụng thông qua nhiều hình thức: trình duyệt web, 1 ứng dụng desktop, hội nghị truyền hình, game online, thương mại điện tử, giáo dục từ xa, thoại IP, …

Theo đó, trung tâm của mạng máy tính là Router – bộ định tuyến. Một cách ngắn gọn, Router kết nối 1 mạng tới 1 mạng khác, vì thế Router chịu trách nhiệm vận chuyển dữ liệu giữa các mạng khác nhau. Địa chỉ đích của 1 gói tin có thể là địa chỉ 1 máy chủ Web đặt tại 1 quốc gia khác hay 1 máy chủ email đặt ngay tại trong mạng cục bộ của doanh nghiệp. Hiệu quả của quá trình truyền thông liên mạng, được đánh giá theo những quy mô khác nhau, phụ thuộc vào khả năng chuyển tiếp gói tin của Router theo những cách thức tối ưu nhất

Phạm vi hoạt động của Router ngày càng vươn xa, hiện nay các bộ định tuyến có thể triển khai trên các vệ tinh ngoài không gian, cho phép vùng bao phủ rộng hơn và ngày càng cải tiến về tốc độ định tuyến. Những Router trên vệ tinh ngoài không gian có khả năng chuyển tiếp gói tin giữa các vệ tinh bên ngoài Trái Đất, nhưng với những cách thức (hay giao thức) tương tư như hoạt động định tuyến trên mặt đất, do đó giảm thiểu độ trễ và tăng tính linh hoạt khi triển khai hạ tầng mạng.

Các Router hiện nay đã vượt qua những yêu cầu tối thiểu là định tuyến, dần được tích hợp các dịch vụ nâng cao như:

* Tăng tính sẵn sàng, lên tới 24x7 (24 giờ/ngày và 7 ngày/ tuần). Đảm bảo kết nối mạng xuyên suốt, tăng thời gian uptime lên tối đa cho phép, Router hỗ trợ các giải pháp chuyển tiếp gói kể cả khi thiết bị tạm thời ngừng định tuyến
* Cung cấp các dịch vụ tích hợp, xu hướng mạng hội tụ như dữ liệu, thoại và video, thông qua đồng thời cả hạ tầng mạng dây lẫn không dây. Router được triển khai chặt chẽ các kỹ thuật QoS – chất lượng dịch vụ để phân loại và ưu tiên hóa gói tin dựa trên ứng dụng và yêu cầu người dùng
* Tích hợp ngày càng mạnh mẽ các chức năng bảo mật để đối phó triệt để với tội phạm mạng. Các kỹ thuật lọc và điều khiển gói tin đa dạng có thể chống lại nhiều kiểu tấn công khác nhau

Tất cả các dịch vụ này được xây dựng xung quanh Router và chức năng cơ bản của bộ định tuyến là chuyển tiếp gói liên mạng. Chúng ta nhớ 1 đặc điểm của mọi hệ thống mạng, đó là trừ khi có định tuyến, các thiết bị thuộc các mạng khác nhau mới có thể giao tiếp. Chương đầu tiên của kỳ 2, giới thiệu về Router bao gồm cấu trúc phần cứng và phần mềm và chức năng của Router trong mạng, bao gồm các tiến trình định tuyến

## Cấu tạo của Router

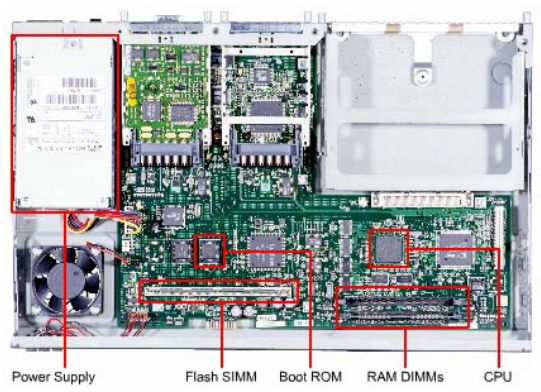
Router hay bộ định tuyến, làm nhiệm vụ đẩy gói tin dọc theo các tuyến đường (route) giữa 2 mạng. Tuyến đường có thể đi qua một hoặc nhiều Router. Trong nhiều trường hợp, sẽ có nhiều tuyến đường khác nhau để đến cùng một mạng đích, khi đó Router sẽ thực hiện một loạt tiến trình để *lựa chọn tuyến đường tốt nhất*. Các tiến trình được sử dụng để lựa chọn tuyến đường tốt nhât và chia sẻ thông tin tuyến đường cho các Router khác được định nghĩa trong các giao thức định tuyến (routing protocol)

Trong suốt quá trình phát triển của mình, Router có nhiều tên gọi và bản thân tên gọi phản ánh chức năng của Router trong một hệ thống. Thời kỳ đầu của mạng Internet, Router được gọi là *IMP* – giao diện xử lý gói tin (Interface Message Processes), làm nhiệm vụ chuyển tiếp (switch) dữ liệu từ 1 mạng tới 1 mạng khác. Trong mô hình mạng LAN, Router được gọi là *default Gateway*, đóng vai trò cổng ra mặc định cho các thiết bị người dùng cuối để tới các mạng khác. Trong mô hình OSI, Router được gọi là *IS* (Intermediate System), đóng vai trò thiết bị trung gian cung cấp kết nối đầu cuối của quá trình truyền dữ liệu

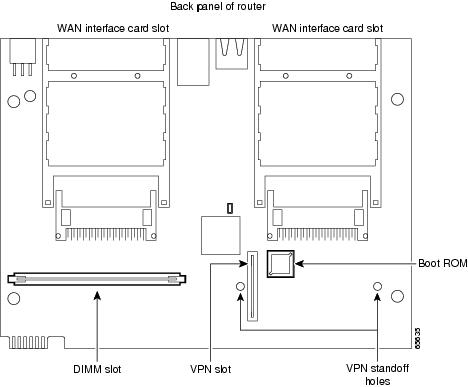
### Các thành phần cấu tạo bên trong Router

Router là một loại *máy tính* đặc biệt. Nó cũng có các thành phần cơ bản giống như máy tính: CPU, bộ nhớ, system bus và các cổng giao tiếp. Tuy nhiên router được thiết kế để thực hiện một số chức năng đặc biệt. Ví dụ: router kết nối hai hệ thống mạng với nhau và cho phép hai hệ thống này có thể liên lạc với nhau, ngoài ra chức năng chính của Router là thực hiện việc chọn lựa đường đi tốt nhất cho dữ liệu

Cũng giống như máy tính cần phải có hệ điều hành để chạy các ứng dụng thì Router cũng cần phải có hệ điều hành để chạy các tập tin cấu hình. Tập tin cấu hình chứa các câu lệnh và các thông số để điều khiển các ứng dụng hay chức năng trên router. Router sử dụng giao thức định tuyến để quyết định chọn đường đi tốt nhất cho các gói dữ liệu. Do đó, tập tin cấu hình cũng chứa các thông tin để cài đặt và chạy các giao thức định tuyến trên Router



*Hình 1.1 – cấu tạo phần cứng bên trong 1 Router*



*Hình 1.2 – Bo mạch chủ Cisco Router 1700*

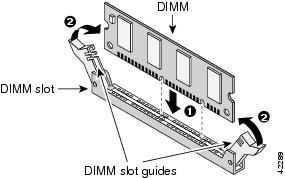
Cấu tạo bên trong của Router bao gồm các thành phần:

**CPU -** Hướng dẫn hệ điều hành của Router, thực hiện được các chức năng từ khởi động hệ thống cho tới thực hiện định tuyến liê mạng, và chuyển tiếp gói giữa các cổng giao tiếp

**RAM –** RAM được sử dụng để lưu trữ dữ liệu có thể được sử dụng bởi CPU như

* Bảng định tuyến
* Bảng ARP
* Buffer hàng đợi
* File cấu hình (running config)
* Hệ điều hành IOS

Trong đa số router, hệ điều hành Cisco IOS chạy trên RAM. RAM thường được chia thành hai phần: Bộ nhớ xử lý chính và bộ nhớ chia sẻ xuất/nhập. Bộ nhớ chia sẻ xuất/nhập được chia cho các cổng giao tiếp làm nơi lưu trữ tạm các gói dữ liệu.Toàn bộ nội dung trên RAM sẽ bị xoá khi mất điện. RAM trên router thường là RAM động (DRAM – Dynamic RAM) và sử dụng giao tiếp DIMM nên dễ dàng nâng cấp



*Hình 1.3 – Khe cắm DIMM để nâng cấp RAM*

**ROM (Read Only Memory) -** Là nơi lưu chương trình kiểm tra (bootstrap) khi khởi động. Nhiệm vụ chính của ROM là kiểm tra phần cứng của router khi khởi động, sau đó chép phần mềm Cisco IOS từ flash vào RAM. ROM sử dụng firmware, là phần mềm nhúng bên trong mạch tích hợp. Firmware bao gồm các chức năng hỗ trợ tiến trình khởi động nên thường không hỗ trợ sửa hay tùy biến

Bootstrap là chương trình hỗ trợ cách ly và xử lý các vấn đề phần cứng khi cài đặt Router. Phần mềm Bootstrap hỗ trợ các công cụ và câu lệnh kiểm tra

**Flash** - Bộ nhớ Flash được sử dụng để lưu toàn bộ hệ điều hành Cisco IOS. Mặc định khi khởi động router sẽ tìm IOS trong flash. Chúng ta có thể nâng cấp hệ điều hành bằng cách chép phiên bản mới hơn vào flash. Phần mềm IOS có thể ở dạng nén hoặc không nén. IOS được chép lên RAM trong quá trình khởi động router (Một số router hỗ trợ chạy trực tiếp IOS trên flash mà không cần chép lên RAM). Tương tự bộ nhớ RAM, bộ nhớ Flash có thể được nâng cấp qua khe SIMM hay card PCMCIA

**NVRAM (Non-volative Random-access Memory) -** Là bộ nhớ RAM không bị mất thông tin khi mất điện. NVRAM được sử dụng bởi Cisco IOS như 1 bộ nhớ vĩnh cử để lưu trữ tập tin startup config – tập tin cấ hình khởi tạo. Tât cả thay đổi cấu hình được lưu trên RAM trong tập tin cấu hình, tuy nhiên 1 vài ngoại lệ cho phép những thay đổi này thực thi trực tiếp vào IOS. Để lưu trữ thông tin cấu hình trong trường hợp thiết bị khởi động lại hoặc mất điện, tập tin cấu hình cần được lưu lại trong NVRAM vào tập tin cấu hình khởi tạo. Nội dung của tập tin cấu hình khởi tạo giữ nguyên ngay cả khi thiết bị khởi động lại

**Bus -** Phần lớn các router đều có bus hệ thống và CPU bus. Bus hệ thống được sử dụng để thông tin liên lạc giữa CPU với các cổng giao tiếp và các khe mở rộng. Loại bus này vận chuyển dữ liệu và các câu lệnh đi và đến các địa chỉ của ô nhớ tương ứng.

**Nguồn điện** - Cung cấp điện cho các thành phần của router, một số Router lớn có thể sử dụng nhiều bộ nguồn hoặc nhiều card nguồn. Còn ở một số router nhỏ, nguồn điện có thể là bộ phận nằm ngoài router.

### Hệ điều hành:

Hệ điều hành được sử dụng bởi Router Cisco được gọi là *Cisco IOS* – Internetwork Operating System. Giống như bất kỳ hệ điều hành trên máy tính, Cisco IOS quản lý các tài nguyên phần cứng và phần mềm trên Router bao gồm việc phân cấp bộ nhớ, quản lý và vận hành các tiến trình, bảo mật và duy trì file hệ thống. Cisco IOS là hệ điều hành *đa nhiệm* được tích hợp nhiều dịch vụ cho phép triển khai nhiều chức năng cùng 1 lúc bao gồm: định tuyến, chuyển tiếp gói, giao tiếp liên mạng và các chức năng truyền thông... *Cisco IOS là các file ảnh,* và có nhiều phiên bản khác nhau. Các phiên bản này được sử dụng phụ thuộc vào model của Router cũng như các tính năng mong muốn hỗ trợ. Thông thường các nhiều chức năng thì file ảnh IOS càng lớn và vì thế chiếm dụng nhiều bộ nhớ flash và RAM, và thời gian khởi động

Tương tự các hệ điều hành khác, Cisco IOS có giao diện người dùng riêng, 1 vài Router có hỗ trợ giao diện đồ họa – GUI (Graphical User Interface), nhưng giao diện dòng lệnh CLI (Command line Interface) mới là phương tiện chính được sử dụng để cấu hình chức năng cho Router. Chúng ta sẽ làm quen và đi sâu vào các câu lệnh cấu hình khi học cụ thể về các tính năng trên Router

Trong quá trình khởi động, file cấu hình khởi tạo (Startup config) được chép vào file cấu hình (running config) trên RAM, IOS sẽ thực thi các lệnh cấu hình lên file cấu hình này. Bất kỳ thay đổi nào của người quản trị đều được lưu trữ trong file cấu hình và ngay lập tức thực hiện bởi IOS.

### Các cổng giao diện của Router

Các cổng giao diện trên Router được chia làm 2 nhóm:

* Các cổng quản lý: Console và AUX
* Các cổng giao tiếp: Ethernet và Serial



*Hình 1.4 – Mặt ngoài phía sau của Router bao gồm các cổng giao tiếp và quản lý*

1. **Các cổng quản lý**

Là các cổng kết nối vật lý chỉ dùng để cấu hình Router. Nếu các cổng giao diện LAN, WAN là kết nối chuyển tiếp được dùng để Router nhận và gửi dữ liệu, thì cổng quản lý cung cấp kết nối đầu cuối để người quản trị có thể cấu hình hoặc trực tiếp trên giao diện dòng lệnh của Router. Cổng quản lý là cổng console hoặc cổng AUX (Auxilliary).

**Cổng Console -** Là cổng giao tiếp bất đồng bộ cho phép kết nối qua giao diện EIA/TIA-232, thường là từ cổng COM. Trên PC, chúng ta sử dụng chương trình mô phỏng thiết bị đầu cuối để thiết lập phiên kết nối dạng văn bản (text-based) trực tiếp, out-of-band, tới giao diện dòng lệnh của Router. Cổng Console được dùng để cấu hình Router.

**Cổng AUX –** Phần lớn Router Cisco có 1 loại cổng quản lý thứ 2 nằm phía sau được gọi là auxiliary (hay cổng AUX) – Tương tự cổng Console, cổng AUX dùng để kết nối trực tiếp, *out -of-band* tới Router. Cổng AUX cho phép người quản trị truy cập và cấu hình qua kết nối quay số tới modem.

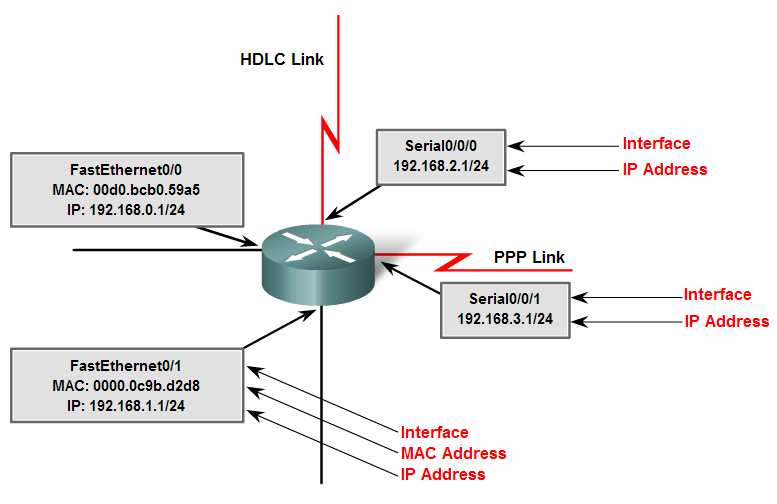
Out-of-band là truyền tải dữ liệu và truyền tải tín hiệu điều khiển trên 2 đường khác nhau, AUX và Console là kiểu quản lý out-of-band. Ngược lại là In-of-band, hay truyền tải dữ liệu và tín hiệu điều khiển trên cùng 1 đường, quản lý in-of-band sử dụng ứng dụng telnet hoặc SSH

1. **Các cổng giao tiếp**

Các cổng giao tiếp (interface) là các giao diện kết nối vật lý dùng để gửi hoặc nhận các gói tin. Mỗi giao diện của Router là thành viên của 1 mạng IP, do đó mỗi interface cần được cấu hình 1 địa chỉ IP và mặt nạ mạng tương ứng, nhưng không được chồng lấn dải địa chỉ lên nhau.

Có 2 loại cổng giao tiếp tương ứng với loại kết nối LAN hoặc WAN:

* Giao diện LAN – chẳng hạn Ethernet, fast Ethernet
* Giao diện WAN – Chẳng hạn Serial, ISDN, hoặc Frame-relay



*Hình 1.5 – Mỗi interface của Router được nối tới các mạng IP khác nhau*

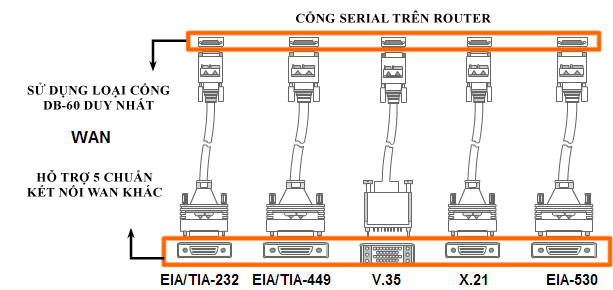
**Giao diện LAN -** Cổng giao tiếp LAN cho phép router kết nối vào mạng cục bộ LAN. Tương tự cách 1 PC sử dụng Ethernet NIC, cổng giao tiếp LAN hiện nay thường là các cổng hỗ trợ chuẩn Ethernet. Là cổng Ethernet nên giao diện LAN sẽ đi kèm với địa chỉ MAC tương ứng, do đó giống như 1 host trong LAN, Router có thể gửi ra bản tin Broadcast ARP hoặc phản hổi ARP. Giao diện LAN của Router sử dụng cáp UTP với đầu nối RJ-45 – 2 loại cáp UTP được sử dụng hiện nay với giao diện Ethernet là cáp thẳng và cáp chéo



*Hình 1.6 - đầu nối Cáp UTP chạy Ethernet*

**Cổng diện WAN –** Giao diện WAN được sử dụng để kết nối Router với 1 mạng ngoài, thường có khoảng cách địa lý lớn. Khác với LAN, giao diện WAN sử dụng nhiều kiểu đóng gói lớp 2 khác nhau tùy vào công nghệ được sử dụng như HDLC, PPP, Frame-relay. Tương tự với giao diện LAN, giao diện WAN sẽ được gán địa chỉ IP và subnet mask tương ứng, tuy nhiên tùy vào công nghệ lớp 2, mà địa chỉ lớp 2 sẽ được sử dụng để đóng gói, không phải địa chỉ MAC

Kết nối WAN cung cấp kết nối thông qua các nhà cung cấp dịch vụ đến các chi nhánh xa hoặc kết nối vào Internet. Với kết nối WAN, Router Cisco hỗ trợ các chuẩn EIA/TIA-232, EIA/TIA-449, V.35, X.21, và EIA/TIA-530 với cáp Serial thông qua duy nhất cổng DB-60, do đặc điểm có thể hỗ trợ 5 chuẩn WAN, nên cổng Serial còn được gọi là cổng 5-trong-1.



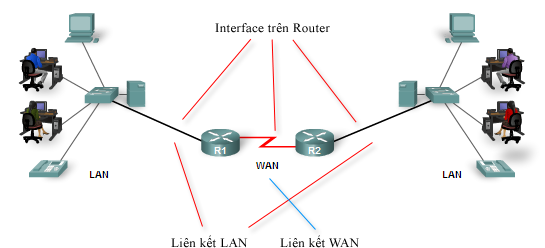
*Hình 1.7 - Giao tiếp WAN trên Router*

Ngoài cổng Serial, Cổng giao tiếp WAN có thể là cổng ISDN, cổng tích hợp đơn vị dịch vụ kênh CSU (Chanel Service Unit).



*Hình 1.8 – Cáp DTE Serial DB60 được dùng cho kết nối WAN*

Các cổng giao tiếp được gọi chung là **interface,** và từ giờ trở đi, chúng ta sẽ sử dụng khái niệm *interface* thay cho giao diện hoặc cổng



*Hình 1.9 – Router hỗ trợ các loại interface khác nhau cho các kết nối khác nhau*

LAN là mạng kết nối các thiết bị cuối như: máy tính,máy in và máy chủ,… trong một miền địa lý hẹp, công nghệ LAN được dùng phổ biến nhất hiện nay là Ethernet. WAN hay mạng diện rộng, được sử dụng để kết nối các mạng LAN, 1 kết nối WAN thường kéo từ mạng LAN khách hàng tới ISP

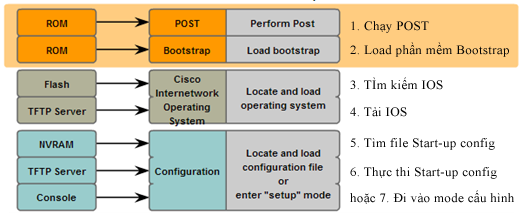
### Quá trình khởi động Router

Router khởi động bằng cách tải bootstrap, hệ điều hành và tập tin cấu hình. Nếu router không tìm thấy tập tin cấu hình thí sẽ tự động vào chế độ cài đặt. Khi bạn hoàn tất việc cấu hình trong chế độ cài đặt thì tập tin cấu hình đó sẽ được lưu trong NVRAM. Để cho router bắt đầu hoạt động, quá trình khởi động phần mềm Cisco IOS thực hiện 3 công đoạn sau:

**Bước 1 - Kiểm tra phần cứng** của trên Router. Bước này gồm 2 bước phụ là chạy POST (Power-on-self-test) ngay khi thiết bị khởi động để kiểm tra trạng thái phần cứng, và tải phần mềm Bootstrap, tất cả công việc được thực hiện trên ROM

**Bước 2 - Tìm và tải phần mềm Cisco IOS**, mặc định Router sẽ tìm kiếm IOS trong bộ nhớ Flash, tuy nhiên có thể triển khai để Router tìm IOS thông qua TFTP Server

**Bước 3 - Tìm kiếm và tải tập tin cấu hình khởi** **tạo** trong NVRAM và nạp vào RAM thành tập tin cấu hình. Nếu trong NVRAM không có, Router sẽ cố tìm kiếm từ các nguồn khác nhau như TFTP Server, nếu các phương pháp khác không tìm ra tập tin cấu hình khởi tạo, Router sẽ nhảy vào chế độ cài đặt



*Hình 1.10 – Các tiến trình khi khởi động Router*

Khi router mới được bật điện lên, sẽ thực hiện quá trình tự kiểm tra POST (Power on self test). Trong quá trình này, router chạy trình điều khiển từ ROM để kiểm tra tất cả các thành phần phần cứng trên router, ví dụ kiểm tra hoạt động của CPU, bộ nhớ, nhiệt độ và các cổng giao tiếp. Sau khi hoàn tất quá trình này, router bắt đầu thực hiện khởi động phần mềm.

Sau quá trình POST, router sẽ thực hiện các bước sau:

**Bước 1 -** Chạy chương trình nạp bootstrap từ ROM

**Bước 2-** Tìm IOS. Giá trị khởi động trên thanh ghi cấu hình sẽ quyết định việc tim IOS ở đâu. Nếu giá trị này cho biết là tải IOS từ flash hay từ 1 máy tính khác thi các câu lệnh boot system trong tập tin cấu hình sẽ cho biết chính xác vị trí và tên của IOS.

**Bước 3 -** Tải hệ điều hành

**Bước 4 -** Tải tập tin cấu hình khởi tạo (Startup config) lưu trong NVRAM vào RAM và thực thi từng dòng lệnh. Các câu lệnh cấu hình được thực hiện:

* Đặt địa chỉ cho các cổng giao tiếp mạng
* Định tuyến
* Đặt mật khẩu
* Triển khai các tiến trình khác nhau trong file cấu hình khởi tạo

**Bước 5 -** Nếu không tìm thấy tập tin cấu hình trong NVRAM thì hệ điều hành sẽ đi tìm TFTP server. Nếu cũng không tìm thấy một TFTP server nào thì chế độ cài đặt sẽ được khởi động.

*Chế độ cài đặt cho phép người quản trị mạng cài đặt một cấu hình tối thiểu cho router khi không thể tìm được tập tin cấu hình từ những nguồn khác. Trong chế độ cài đặt, câu trả lời mặc định được đặt trong dấu ngoặc vuông [] ở sau mỗi câu hỏi.*

Để kiểm tra quá trình khởi động Router, chúng ta sử dụng câu lệnh show version trong giao diện CLI:

Router>show version

Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE SOFTWARE (fc2) 🡪 Phiên bản IOS

Technical Support: http://www.cisco.com/techsupport

Copyright (c) 1986-2007 by Cisco Systems, Inc.

Compiled Wed 18-Jul-07 04:52 by pt\_team

ROM: System Bootstrap, Version 12.3(8r)T8, RELEASE SOFTWARE (fc1) 🡪 bootstrap

System returned to ROM by power-on

System image file is "flash:c1841-advipservicesk9-mz.124-15.T1.bin" 🡪 vị trí IOS

<output omitted>

Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.

🡪 Dung lượng bộ nhớ RAM

Processor board ID FTX0947Z18E

M860 processor: part number 0, mask 49 🡪 CPU

2 FastEthernet/IEEE 802.3 interface(s) 🡪 Các interface

191K bytes of NVRAM. 🡪 dung lượng NVRAM

63488K bytes of ATA CompactFlash (Read/Write) 🡪 dung lượng Flash

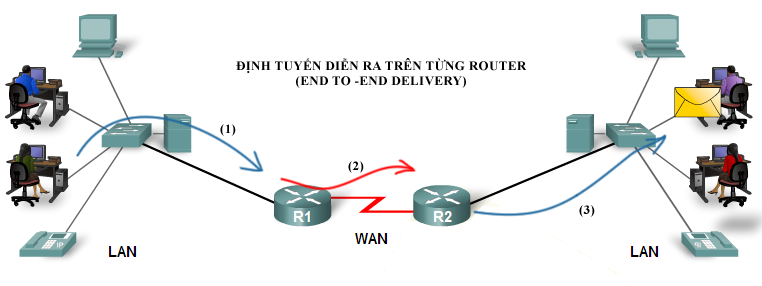
Configuration register is 0x2102 🡪 thanh ghi

### Router và mô hình OSI

1. **Vị trí trung tâm mạng IP**

Router là thiết bị trung tâm của mạng máy tính, mở rộng hơn là mạng Internet. Người dùng thường không có nhận thức về sự hiện diện của Router trong hạ tầng mạng, họ quan tâm tới các ứng dụng, và liệu có thể truy cập tới Server nào đó để lấy xuống thông tin mình cần. Trên con đường kết nối giữa người dùng tới Server, hoặc giữa các người dùng với nhau là sự hiện diện của Router. Nhiệm vụ của Router là chuyển tiếp gói giữa cac mạng, đưa gói tin từ 1 nguồn tơi đích.

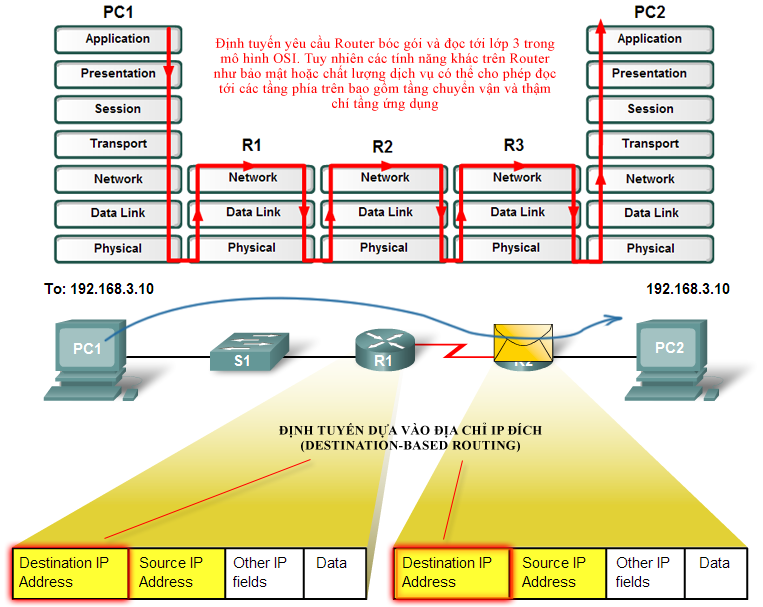
Router là một điểm kết nối các mạng. Điều này có nghĩa là một router sẽ có nhiều interface và mỗi interface nối tới một mạng. Các mạng trên các interface của Router phải *độc lập* với nhau, hay dải địa chỉ IP không được chồng lấn lên nhau. Khi một Router nhận được gói tin IP trên một interface, Router sẽ xác định interface được sử dụng để chuyển gói tin đó đến đúng mạng đich. Interface được chọn có thể là interface nối trực tiếp tới mạng đích, hoặc nối tới 1 Router khác có tuyến đường hoặc kết nối trực tiếp với mạng đích. Interface trên Router có thể kết nối tới một mạng cục bộ (LAN) hoặc mạng diện rộng (WAN).



*Hình 1.11 – Mô tả quá trình định tuyến trên Router*

1. **Vai trò của Router**

Như mọi thiết bị, Router sẽ xử lý 1 gói tin thông qua các quá trình đóng gói và bỏ đóng gói, với thứ tự từ trên xuống hoặc từ dưới lên dọc mô hình OSI tùy thuộc chiều là nhận hay gửi gói tin. Tuy nhiên Router vẫn được gọi là thiết bị lớp 3, vì nhiệm vụ cơ bản của Router (định tuyến) là làm việc với địa chỉ IP, đọc địa chỉ đích và thực hiện định tuyến, hay nói cách khác, Router đưa ra quyết định chuyển tiếp gói dựa vào thông tin lớp 3



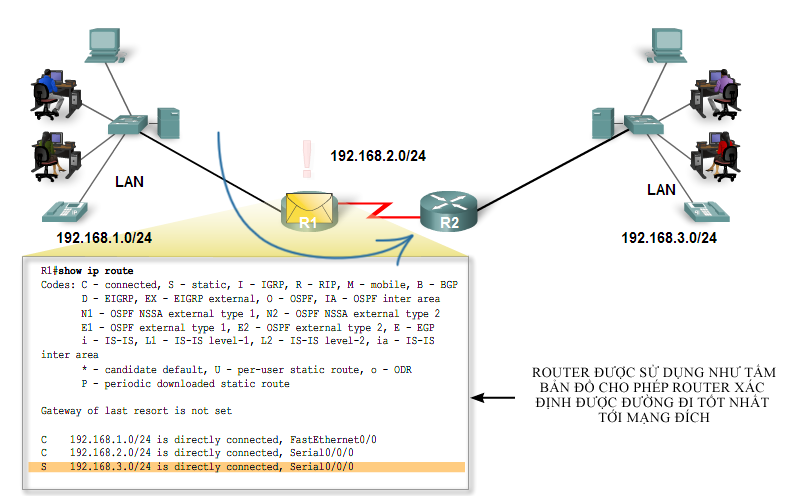
*Hình 1.12 – Router là thiết bị lớp 3*

Định tuyến trên Router yêu cầu đọc tới thông tin tầng 3, tuy nhiên các dịch cao cấp hơn như bảo mật hoặc chất lượng dịch vụ cho phép Router đọc thông tin tầng 4, và cao hơn thậm chí tầng 7 để có thể đưa ra quyết định chuyển tiếp gói (có thể khác với định tuyến)- Nên cách nói “thiết bị lớp 3” chỉ nhằm nhấn mạnh vào công việc định tuyến của Router

Router thực hiện định tuyến để chuyển gói tin từ nguồn tới đích thông qua 2 tiến trình

* Tìm đường đi ngắn nhất cho gói tin
* Chuyển tiếp gói tin giữa các Interface

Router sử dụng bảng định tuyến của mình để xác định đường đi tốt nhất và chuyển tiếp gói. Khi Router nhận được 1 gói tin, nó sẽ kiểm tra địa chỉ IP đích và tra cứu xem liệu có tuyến đường tới mạng đích phù hợp trong bảng định tuyến. Thông tin tuyến đường bao gồm cả interface được sử dụng để chuyển tiếp gói đi. Tất nhiên trước đó để có thể đọc được phần packet, Router phải bóc gói lớp 1 và lớp 2. Sau khi Router xác định được interface, các tiến trình đóng gói sẽ diễn ra ngược lại, packet giữ nguyên được đóng vào frame mới tương ứng, sau đó đẩy ra môi trường dưới dạng tín hiệu phù hợp với môi trường truyền



*Hình 1.13 – Router sử dụng bảng định tuyến để tìm đường đi  
và xác định interface chuyển tiếp*

Các interface của Router thuộc các mạng khác nhau, và môi trường truyền trên các interface có thể khác nhau, do đó quá trình đóng gói lại sẽ xác định xem môi trường là gì, và header trong frame sẽ được đóng gói lại tương ứng. Trong ví dụ trên, Router R1 nhận gói tin có frame Ethernet, sẽ đóng gói lại thành 1 frame PPP để gửi sang Router R2

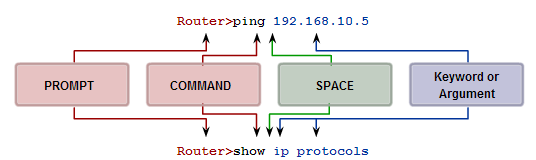
Tuyến đường được lựa chọn bởi Router có thể là tuyến tĩnh (static route – sẽ học trong chương 2) hoặc tuyến động (dynamic route – sẽ học trong chương 3), tuy nhiên chúng đều mang các thông tin cơ bản về mạng đích và interface dùng để chuyển tiếp. Chúng ta sẽ bàn trong các chương tiếp theo

## Cấu trúc giao diện dòng lệnh CLI

### Các yếu tố giao diện dòng lệnh CLI

**Cấu trúc**

Cấu trúc cơ bản của dòng lệnh trong IOS bao gồm:



*Hình 1.14 – Cấu trúc câu lệnh*

* **Prompt –** Xác định hostname của thiết bị và dấu nhắc lệnh [dấu nhắc lệnh thể hiện mode cấu hình]
* **Command –** Là cú pháp câu lệnh
* **Space –** Khoảng trắng giữa các từ trong cú pháp hoặc cú pháp và từ khóa
* **Argument –** Các thông số chi tiết nhập vào

### CLI Help

**Cisco IOS CLI help** là 1 công cụ hỗ trợ câu hình vô cùng đắc lực, hỗ trợ người quản trị trong quá trình thực thi câu lệnh. Các công cụ bao gồm:

**[?] –** Sử dụng dấu ? để gợi ý các câu lệnh có thể thực hiện tại dấu nhắc lệnh, ví dụ:

R3>show ip ?

arp IP ARP table

bgp BGP information

dhcp Show items in the DHCP database

eigrp IP-EIGRP show commands

interface IP interface status and configuration

nbar Network-Based Application Recognition

ospf OSPF information

protocols IP routing protocol process parameters and statistics

rip IP RIP show commands

route IP routing table

ssh Information on SSH

Ngoài ứng dụng hiển thị các câu lệnh có thể thực thi tại dấu nhắc lệnh, ? được sử dụng để hiển thị các câu lệnh có thể gõ vào nếu bắt đầu bằng 1 số ký tự cho trước, chẳng hạn:

R3#show ip interface vi?

Virtual-Access Virtual-Template

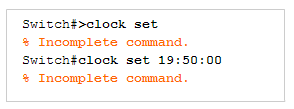
Hiên thị các câu lệnh có thể gõ nếu bắt đầu bằng ký tự {vi}

**[Tab] –** phím tab được sử dụng để làm đầy câu lệnh khi người dùng không nhớ câu lệnh hoàn chỉnh, chẳng hạn:

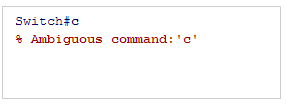
R3>show ip inter 🡨 tab gõ ở đây

R3>show ip interface 🡨 câu lệnh được làm đầy

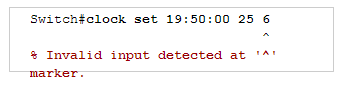
**Cú pháp báo lỗi –** Hiển thị sau khi 1 câu lệnh được gõ và thông báo liệu câu lện có lỗi, và nếu lỗi thì lỗi ở đâu, loại lõi là gì:



*Hình 1.15 – câu lệnh chưa hoàn chỉnh*



*Hình 1.16 – Câu lệnh vô nghĩa*



*Hình 1.17– Argument nhập vào không chính xác, lỗi tại vị trí dấu ^*

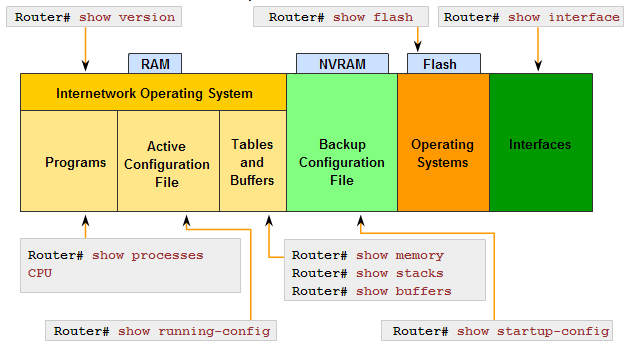
### Hotkey

Hotkey được sử dụng để tăng tốc độ cấu hình và giúp việc nhập câu lênh dễ dàng hơn:

Chúng ta có bảng hotkey sau:

|  |  |
| --- | --- |
| Hotkey | Ý nghĩa |
| Tab | Làm đầy câu lệnh |
| Ctrl-Z | Quay lại mode EXEC |
| Ctrl-R | Hiển thị lại 1 dòng trước |
| Mũi tên lên/xuống | History |
| Ctrl-shift-6 | Ngừng các tiến trình hiện tại như ping |
| Ctrl-C | Thoát khỏi mode cấu hình |
| Ctrl-A | Quay lại đầu dấu nhăc lệnh |
| Ctrl-E | Quay về cuối dấu nhăc lệnh |
| Ctrl-U | Xóa toàn bộ ký tự từ dấu nhắc tới ký tự đầu |

### Các câu lệnh show kiểm tra cac vùng bộ nhớ



*Hình1 .18 – các thành phần bộ nhớ và câu lệnh kiểm tra trạng thái*

|  |  |
| --- | --- |
| Show | Ý nghĩa |
| Show version | Xem phiên bản IOS và các thành phần bộ nhớ |
| Show flash | Xem các thư mục và file trong bộ nhơ flash |
| Show interface | Kiểm tra trạng thái và thông số trên interface |
| Show processes | Xem các tiến trình đang chạy trên thiết bị |
| Show running-config | Kiểm tra file cấu hình đang chạy trong RAM |
| Show startup-config | Xem file cấu hình khởi tạo trong NVRAM |
| Show memory | Xem các thành phần bộ nhớ bị chiếm dụng |

## Cấu hình cơ bản

Công việc cấu hình cơ bản 1 Router bao gồm

* Đặt tên cho Router
* Thiết lập Password
* Cấu hình biểu ngữ (Banner)
* Lưu file cấu hình khởi tạo
* Xác nhận kiểm tra cấu hình cơ bản
* Exec-timeout và Logging Synchronous

Công việc cấu hình cơ bản thường chỉ diễn ra lần đầu tiên hoặc khi Router không lưu file cấu hình khởi tạo. Cấu hình cơ bản tương đối quan trọng và có thể có 1 chút khác biệt liên quan đến thói quen người dùng

Khi người dùng mới login vào Router, họ sẽ ở mode User EXEC, mode User cho phép người dùng theo dõi một vài trạng thái hiện tại cơ bản trên Router, nhưng không thể cấu hình. Dấu nhắc lệnh tại mode User là [>]

RouterX>

Để có thể cấu hình, người dùng cấn bước vào mode Privilege EXEC. Cú pháp để đi vào mode Privilege từ mode User là enable Dấu nhắc lệnh tại mode Privilege là [#]

RouterX>enable

RouterX#

Để có thể bắt đầu cấu hình, người dùng phải đi vào mode cấu hình trong mode Privilege EXEC, cú pháp vào mode cấu hình là configure terminal. Dấu nhắc lệnh là (config)#

RouterX#config terminal

RouterX(config)#

**Cấu hình tên cho Router**

Cú pháp để cấu hình tên cho Router là hostname {tên}, trong mode cấu hình

RouterX(config)#hostname R1

R1(config)#

**Cấu hình Passoword**

Mật khẩu được sử dụng để hạn chế người dùng truy cập vào Router trái phép. Do có thể truy cập vào Router thông qua nhiều đường khác nhau, nên sẽ có các loại mật khẩu khác nhau cần được cấu hình – 3 mật khẩu cơ bản nên cấu hình trên mọi thiết bị là:

* Mật khẩu truy cập qua console
* Mật khẩu truy cập từ xa qua kết nối đầu cuối ảo - VTY
* Mật khẩu đi vào mode Privilege EXEC

**Mật khẩu truy cập qua Console** được cấu hình tại sub-mode line console của Router. Cú pháp là password {mật khẩu}. Cấu hình ví dụ:

R1(config)#line console 0

R1(config-line)#password cisco

R1(config-line)#login

Trong đó

* *line console 0* để đi vào sub-mode cấu hình các thuộc tính của line Console
* *password* là mật khẩu sẽ hỏi khi người dùng truy cập qua đường console
* *login* được sử dụng để yêu cầu Router hỏi password khi truy cập

**Mật khẩu truy cập qua VTY** được cấu hình tại sub-mode line vty của Router. Cú pháp là password {mật khẩu}. Truy cập qua vty có thể là Telnet hoặc SSH, hay các kết nối in-of-band, cho phép người dùng truy cập và cấu hình từ xa qua 1 hạ tầng mạng. Cấu hình ví dụ:

R1(config)#line vty 0 15

R1(config-line)#password cisco

R1(config-line)#login

Trong đó

* *line vty 0 15* để đi vào sub-mode cấu hình các thuộc tính của line vty. Cisco Router hỗ trợ 15 kết nối vty đồng thời, do đó để tiết kiệm thời gian, cú pháp [0 15] để cấu hình cho 1 dải line vty
* *password* là mật khẩu sẽ hỏi khi người dùng truy cập qua đường vty
* *login* được sử dụng để yêu cầu Router hỏi password khi truy cập

**Mật khẩu truy cập vào Privilege mode** được cấu hình tại mode cấu hình của Router. Cú pháp là enable password. Tuy nhiên mật khẩu này sẽ được lưu vào file cấu hình dưới dạng clear-text, do đó, 1 giải pháp bảo mật hơn là sử dụng enable secret, khi đó mật khẩu lưu trong file cấu hình sẽ được mã hóa md5

Router(config)#enable password cisco

và

Router(config)#enable secret cisco

**Lưu file cấu hình khởi tạo**

Khi cấu hình Router, mọi câu lệnh cấu hình được chép vào trong file cấu hình running-config, file running-config được lưu trong RAM, nên sẽ mất khi Router mất điện hoặc khởi động lại. Để tránh trường hợp mất thông tin cấu hình đột ngột, chúng ta phải lưu file cấu hình vào file cấu hình khởi tạo Startup-config. (File cấu hình khởi tạo lưu trong NVRAM nên sẽ không bị mất khi Router khởi động lại)

R1#copy running-config startup-config

Ngoài việc copy file cấu hình vào startup-config, chúng ta có thể lưu file cấu hình vào các nguồn khác nhau như TFTP Server, FTP Server, HTTP Server, Flash,… Chúng ta xem các lựa chọn cho phép:

R1#copy running-config ?

archive: Copy to archive: file system

flash: Copy to flash: file system

ftp: Copy to ftp: file system

http: Copy to http: file system

https: Copy to https: file system

idconf Load an IDConf configuration file

null: Copy to null: file system

nvram: Copy to nvram: file system

pram: Copy to pram: file system

rcp: Copy to rcp: file system

running-config Update (merge with) current system configuration

scp: Copy to scp: file system

slot0: Copy to slot0: file system

startup-config Copy to startup configuration

syslog: Copy to syslog: file system

system: Copy to system: file system

tftp: Copy to tftp: file system

tmpsys: Copy to tmpsys: file system

xmodem: Copy to xmodem: file system

ymodem: Copy to ymodem: file system

Chúng ta theo dõi lựa chọn phổ biến đó là lưu file cấu hình vào TFTP Server, về nguyên tắc, để lưu file vào 1 thiết bị ngoài, ta phải đảm bảo kết nối giữa Router và thiết bị đó trước. Chúng ta giả sử kết nối đã thành công

Với TFTP Server

R1#copy running-config tftp:

Address or name of remote host []? 64.104.207.171

Destination filename [ce\_2-confg]? backup\_cfg\_for\_my\_router

!!

1030 bytes copied in 2.489 secs (395 bytes/sec)

R1#

Các thông số cần chú ý:

|  |  |
| --- | --- |
| Show | Ý nghĩa |
| Address or name of remote host []? | Đia chỉ TFTP Server |
| Destination filename [ce\_2-confg]? | Tên file được lưu |
| 1030 bytes | Dung lượng file đã lưu |

**Kiểm tra cấu hình trên Router**

Câu lệnh xem file cấu hình (running-config) cua Router, hay nói cách khác, kiểm tra các câu lệnh đã được Router thực thi:

R1#show running-config

## Tổng kết

Hoàn thành chương 1, chúng ta đã có được cái nhìn tương đối rõ ràng về:

* Cấu trúc phần cứng của Router
* Các thành phần bộ nhớ và quá trình khởi động
* Các interface và đầu nối hỗ trợ kết nối
* Giao diện cấu hình dòng lệnh CLI
* Cấu hình các thành phần cơ bản

# ****CHƯƠNG 2 -**** ĐỊNH TUYẾN TĨNH

Định tuyến là công việc cơ bản được thực hiện trong mọi mạng dữ liệu, làm nhiệm vụ đẩy gói tin từ mạng nguồn tới mạng đích. Router là các thiết bị chịu trách nhiệm định tuyến. Để có thể định tuyến, Router cần có thông tin tuyến đường. Để có thông tin tuyến đường, Router có thể học thông qua các nguồn khác nhau như kết nối trực tiếp, thông qua giao thức định tuyến động, hoặc cấu hình tĩnh bởi người quản trị. Thông tin định tuyến tĩnh có độ tin cậy cao, vì thế không yêu cầu xử lý nhiều thông tin, tuy nhiên do đặc điểm đơn giản, nên thiếu khả năng mở rộng và không có khả năng phản ứng trước sự cố.

Tuy nhiên, định tuyến tĩnh vẫn đóng vai trò quan trọng trong mọi sơ đồ mạng và hầu như luôn có mặt trong các bảng định tuyến của Router. Nắm được kiến thức về định tuyến tĩnh chính là nền tảng cơ bản để có thể triển khai một mạng IP.

## Mạng kết nối trực tiếp

**Topology:** Chúng ta sẽ sử dụng topology sau để làm ví dụ cấu hình địa chỉ IP trên các interface của Router:



**Yêu cầu:** Cấu hình địa chỉ IP cho các interface trên mỗi Router để R2 có thể giao tiếp với R1 và R3

**Cho trước** Quy hoạch địa chỉ IP của mạng như sau:

|  |  |  |
| --- | --- | --- |
| Router | Interface | Mạng |
| R1 | fastEthernet 0/0 | 172.16.3.0/24 |
| Serial 0/0/0 | 172.16.2.0/24 |
| R2 | fastEthernet 0/0 | 172.16.1.0/24 |
| Serial 0/0/0 | 172.16.2.0/24 |
| Serial 0/0/1 | 192.168.1.0/24 |
| R3 | fastEthernet 0/0 | 192.168.2.0/24 |
| Serial 0/0/0 | 192.168.1.0/24 |

**Kiểm tra:** Trước khi bắt đầu, chúng ta kiểm tra trạng thái ban đầu của các Router

Kiểm tra bảng định tuyến:

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

Ta thấy chưa có tuyến đường nào trong bảng định tuyến. Thứ nhất, do chưa triển khai định tuyến, thứ 2, do cac Router vẫn được cấu hinh các mạng kết nối trực tiếp, hay các interface vẫn chưa có địa chỉ IP

R1#show ip interface brief

Interface IP-Address OK? Method Status Protocol

FastEthernet0/0 unassigned YES manual administratively down down

FastEthernet0/1 unassigned YES manual administratively down down

Serial0/0/0 unassigned YES manual administratively down down

Serial0/0/1 unassigned YES manual administratively down down

Các interface chưa được gán địa chỉ IP. Trạng thái hiện tại của interface là *administratively down* và trạng thái line protocol cũng là *down.* *Administratively down* nghĩa là interface đó hiện đang ở trạng thái đóng (shutdown mode) hay interface đó chưa hoạt động. Trạng thái line protocol *down* nghĩa là interface chưa nhận được tín hiệu trên đường truyền, hoặc giao thức tầng liên kết dữ liệu và vật lý trên đường truyền đang có vấn đề, nguyên nhân trong trường hợp của chúng ta chính là vì trạng thái của interface hiện đang là đóng

R1#show running-config

Building configuration...

Current configuration : 460 bytes

!

version 12.4

<output omitted>

!

interface FastEthernet0/0

no ip address

duplex auto

speed auto

shutdown

!

interface serial0/0/0

no ip address

shutdown

!

<output omitted>

Trong ví dụ trên ta thấy trên các interface fastEthernet0/0 và Serial0/0/0 đều đang chưa có địa chỉ IP và trạng thái hiện thời là đóng (shutdown)

### Giao diện Ethernet

1. **Cú pháp:**

Để cấu hình địa chỉ IP cho 1 interface, chúng ta đi vào sub-mode interface từ mode global config

R1>ena

R1#config terminal

R1(config)#interface {loại cổng/slot}

Loại cổng có thể là serial hoặc ethernet/fastEthernet, slot là thứ tự cổng đó, chẳng hạn 0/0, hoặc 0/1, hoặc 0/0/0 hoặc 0/0/1. Trong ví dụ trên, R1 có 2 cổng fastEthernet0/0 và Serial0/0/0

Cú pháp để cấu hình địa chỉ IP cho 1 interface:

Ip address {địa chỉ IP host} {subnet mask}

Trong đó:

* *Địa chỉ ip host* là địa chỉ IP được dùng để gán cho interface, mặc dù interface đại diện cho 1 mạng nhưng địa chỉ gán cho interface là 1 địa chỉ host
* *Subnet mask* là mặt nạ mạng của mạng kết nối trực tiếp với Router qua interface đó

1. **Triển khai:**

Chúng ta triển khai cấu hình địa chỉ IP trên interface fastEthernet 0/0 của R1

R1(config)#interface fastethernet 0/0

R1(config-if)#ip address 172.16.3.1 255.255.255.0

R1(config-if)#no shutdown

Chúng ta nhận được log như sau

\*Mar 1 01:16:08.212: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up

\*Mar 1 01:16:09.214: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up

Chúng ta thấy có 2 bản tin thông báo. Bản tin đầu tiên chỉ ra trạng thái interface là mở (no shut), như vậy tầng vật lý của cổng fastEthernet hoạt động tốt. Bản tin thứ 2 chỉ ra tầng liên kết dữ liệu trên cổng fastEthernet hoạt động tốt vì nhận thấy tín hiệu sóng mang phía đầu xa (trong trường hợp này là Switch)

### Giao diện Serial

Tương tự chúng ta triển khai cấu hình địa chỉ IP trên interface serial0/0/0 của R1

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

R1(config-if)#ip address 172.16.2.1 255.255.255.0

R1(config-if)#no shutdown

Chúng ta nhận được log như sau

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to down

R1#show interfaces serial 0/0/0

Serial0/0/0 is administratively down, line protocol is down

Ta thấy mặc dù đã đặt địa chỉ IP và kich hoạt bằng no shutdown, nhưng trạng thái của interface Serial0/0/0 vẫn là down, cả administratively lẫn line protocol. Tương tự Ethernet, cổng Serial chỉ up khi nhận tín hiệu từ đầu xa. Chúng ta cấu hình địa chỉ IP và mở interface Serial0/0/0 trên Router R2

R2(config)#interface s0/0/0

R2(config-if)#ip address 172.16.2.2 255.255.255.0

R2(config-if)#no shut

%LINK-5-CHANGED: Interface Serial0/0/0, changed state to up

R2 show interface s0/0/0

Serial0/0/0 is up, line protocol is down (disabled)

Sau khi 2 đầu của link Serial mở (no shutdown) và được cấu hình địa chỉ IP. Trạng thái administrative lúc này đã up. Tuy nhiên line protocol vẫn DOWN. Để kết nối hoạt động, 2 đầu Serial phải được cung cấp cùng xung đồng hồ (clock rate) – Xung đồng hồ được thiết lâp trên đầu dây Serial đóng vai trò DCE, trong trường hợp này là Router R1

R1(config-if)#clock rate 64000

R1(config-if)#

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0, changed state to up

R1#show interface s0/0/0

Serial0/0/0 is up, line protocol is up (connected)

Sau khi được cung cấp clock rate. 2 interface trên R1 và R2 đã sẵn sàng hoạt động và bắt đầu trao đổi dữ liệu:

R1#

R1#ping 172.16.2.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.2.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/14 ms

**Xác nhận**

Để xác nhận cấu hình địa chỉ IP và trạng thái interface, chúng ta sử dụng các câu lệnh sau:

RouterX#show ip interface brief

* Kiểm tra địa chỉ IP và trạng thái (administrative và line protocol) của tất cả interface

R1#show ip interface brief

Interface IP-Address OK? Method Status Protocol

FastEthernet0/0 unassigned YES unset administratively down down

FastEthernet0/1 unassigned YES unset administratively down down

Serial0/0/0 172.16.2.1 YES manual up up

Serial0/0/1 unassigned YES unset administratively down down

Vlan1 unassigned YES unset administratively down down

RouterX#show interface {loại/slot}

* Xem tất cả thông tin trạng thái, các thông số, thuộc tính, cách đóng gói của 1 interface

R1#show interface serial0/0/0

Serial0/0/0 is up, line protocol is up (connected)

Hardware is HD64570

Internet address is 172.16.2.1/24

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

reliability 255/255, txload 1/255, rxload 1/255

Encapsulation HDLC, loopback not set, keepalive set (10 sec)

Last input never, output never, output hang never

Last clearing of "show interface" counters never

Input queue: 0/75/0 (size/max/drops); Total output drops: 0

Queueing strategy: weighted fair

Output queue: 0/1000/64/0 (size/max total/threshold/drops)

Conversations 0/0/256 (active/max active/max total)

Reserved Conversations 0/0 (allocated/max allocated)

Available Bandwidth 1158 kilobits/sec

5 minute input rate 8 bits/sec, 0 packets/sec

5 minute output rate 8 bits/sec, 0 packets/sec

10 packets input, 1280 bytes, 0 no buffer

Received 0 broadcasts, 0 runts, 0 giants, 0 throttles

0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort

10 packets output, 1280 bytes, 0 underruns

0 output errors, 0 collisions, 1 interface resets

0 output buffer failures, 0 output buffers swapped out

0 carrier transitions

DCD=up DSR=up DTR=up RTS=up CTS=up

RouterX#show run

* Kiểm tra file cấu hình, hiển thị những lệnh cấu hình đã thực hiện

<output omitted>

!

interface FastEthernet0/0

ip address 172.16.1.1 255.255.255.0

duplex auto

speed auto

!

interface Serial0/0/0

ip address 12.12.12.1 255.255.255.0

clock rate 64000

RouterX#show controller {Serial/slot}

* Kiểm tra cấu hình và trạng thái phần cứng, bao gồm clock rate trên link Serial

R1#show controllers serial 0/0/0

Interface Serial0/0/0

Hardware is PowerQUICC MPC860

DCE V.35, clock rate 64000

**Tỏng kết**

Như vậy chúng ta đã cấu hình thành công interface fastEthernet và interface Serial, toàn bộ cú pháp như sau:

Với interface fastEthernet:

RouterX(config-if)#ip address {địa chỉ IP} {subnet mask}

RouterX(config-if)#no shutdown

Với interface Serial

RouterX(config-if)#ip address {địa chỉ IP} {subnet mask}

RouterX(config-if)#clock rate {pulse}

RouterX(config-if)#no shutdown

## Mạng kết nối trực tiếp

### Kết nối trực tiếp

Mạng kết nối trực tiếp là mạng gắn trực tiếp trên 1 interface của Router. Như vậy, bất cứ khi nào trạng thái interface của Router trở thành UP/UP với 1 cặp địa chỉ IP/subnet mask hợp lệ, thì trên Router sẽ xác định có 1 mạng kết nối trực tiếp. 1 bản ghi sẽ xuất hiện trong bảng định tuyến mang thông tin tuyến đường tới mạng đó. Xét ví dụ như trên, chúng ta kiểm tra bảng định tuyến của Router R1 và R2, cú pháp câu lệnh kiểm tra bảng định tuyến:

RouterX#show ip route

* Kiểm tra thông tin tuyến đường, mạng đích, và interface chuyển tiếp gói

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 1 subnets

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.3.0 is directly connected, fastEthernet0/0

Các thông số cần quan tâm:

|  |  |
| --- | --- |
| Show | Ý nghĩa |
| C | Connected, mạng là kết nối trực tiếp |
| 172.16.2.0 | Địa chỉ mạng đích |
| Serial 0/0/0 | Interface chuyển tiếp gói |
| Directly connected | Đẩy gói tin thẳng ra interface chuyển tiếp |

Chúng ta cấu hình tất cả interface còn lại trên cả 3 Router R1, R2 và R3

Trên Router R2

R2(config)#interface s0/0/1

R2(config-if)#ip address 192.168.1.2 255.255.255.0

R2(config-if)#clock rate 64000

R2(config-if)#no shutdown

R2(config)#interface fastethernet 0/0/0

R2(config-if)#ip address 172.16.1.1 255.255.255.0

R2(config-if)#no shutdown

Trên Router R3

R3(config)#interface s0/0/0

R3(config-if)#ip address 192.168.1.1 255.255.255.0

R3(config-if)#no shutdown

R3(config)#interface fastethernet 0/0/0

R3(config-if)#ip address 192.168.2.1 255.255.255.0

R3(config-if)#no shutdown

Kiểm tra bảng định tuyến trên Router R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.1.0 is directly connected, fastEthernet0/0

C 192.168.1.0 is directly connected, Serial0/0/1

và Router R3

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

C 192.168.1.0/24 is directly connected, Serial0/0/0

C 192.168.2.0/24 is directly connected, fastEthernet0/0

Khi trên Router có những tuyến đường kết nối trực tiếp, nghĩa là Router có thể chuyển tiếp gói giữa những mạng kết nối trực tiếp với Router đó, hay nói cách khác:

* R1 có thể giao tiếp với mạng 172.16.3.0/24 và mạng 172.16.2.0
* R2 có thể giao tiếp với mạng 172.16.2.0/24, mạng 172.16.1.0/24 và mạng 192.168.1.0/24
* R3 có thể giao tiếp với mạng 192.168.1.0/24 và mạng 192.168.2.0/24

Tới đây chúng ta có 2 bài toán mới:

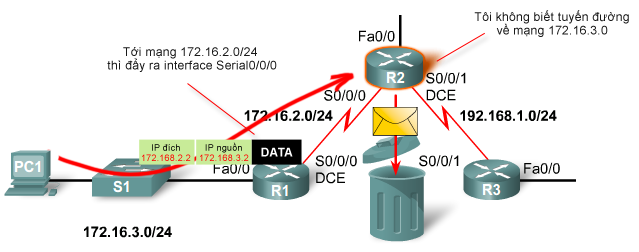
1- Điều gì sẽ xảy ra khi 1 thiết bị trong mạng LAN gửi gói tin tới kết nối WAN giữa R1-R2 ?

2- Điều gì sẽ xảy ra khi Router R1 gửi gói tin tới các mạng không phải kết nối trực tiếp, chẳng hạn mạng LAN của R2, mạng LAN của R3 hoặc kết nối WAN giữa R2-R3

**Với bài toán 1 –** Khi 1 PC trong mạng LAN gửi gói tin tới 1 địa chỉ khác mạng, trong trường hợp này chẳng hạn 172.16.2.2 là interface serial0/0/0 của R2. Gói tin IP:



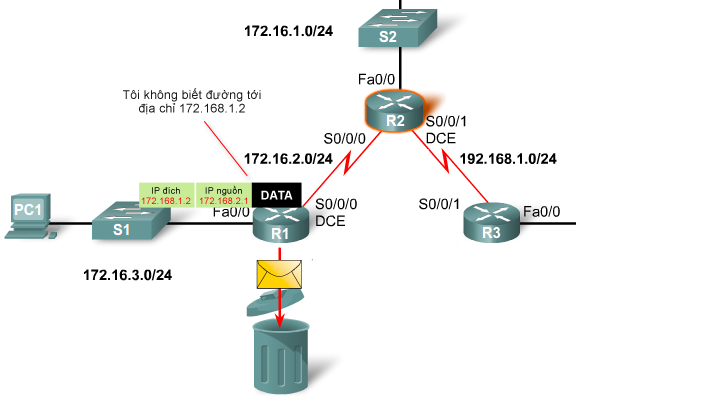
1. Gói tin sẽ được đẩy tới default gateway chính là interface fastEthernet 0/0 của R1
2. Router R1 tra cứu bảng định tuyến và biết rằng mạng 172.16.2.0/24 là kết nối trực tiếp
3. Router R1 chuyển gói tin sang Router R2
4. Router R2 tra cứu bảng định tuyến, nhưng không có thông tin về mạng nguồn là mạng LAN của R1: 172.168.3.0
5. Router R2 sẽ loại bỏ (drop) gói tin



*Hình 2.1 – Router R2 không biết tuyến đường phản hồi*

**Với bài toán 2 –** Khi Router R1 muốn gửi gói tin tới 1 mạng khác mạng kết nối trực tiếp. Chẳng hạn mạng LAN của của R2, gói tin IP: 

1. Router R1 sẽ tra cứu bảng định tuyến cho địa chỉ đích 172.168.1.1
2. Router R1 không tìm thấy thông tin tuyến đương tới mạng đích
3. Router R1 drop gói tin



*Hình 2.2– R1 không có tuyến đường tới mạng khác kết nối trực tiếp*

Để giải 2 bài toán này, bắt buộc trên Router R1 và R2 phải học tuyến đường về các mạng khác mạng kết nối trực tiếp của mình. Quá trình học tuyến đường có thể 2 diễn ra theo 1 trong 2 cách:

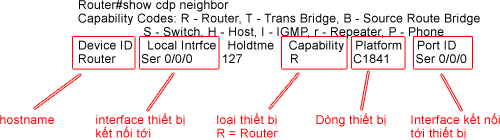
* Học tĩnh (static) – Người quản trị “vẽ” đường cho Router
* Học động (dynamic) – Các Router học thông tin tuyến đường của nhau

### Giao thức CDP

CDP - Cisco Discovery Protocol, là 1 giao thức mạnh mẽ hỗ trợ công việc theo dõi và gỡ rối kết nối giữa các thiết bị Cisco trong mạng. Nhiệm vụ của giao thức CDP là thu thâp thông tin về các thiết bị kết nối trực tiếp

Giao thức CDP chỉ sử dụng 1 bản tin là bản tin CDP advertisement được gửi bởi Router ra các interface có bật CDP. Bản tin CDP Advertisement mang các thông tin:

* Loại thiết bị (Router, hay Switch, hay IP Phone)
* Model của thiết bị (Cisco Router 1811 hoặc Catalyst Switch 3550,…)
* Tên thiết bị (hostname)
* Interface thiết bị kết nối tới
* Interface kết nối tới thiết bị



*Hình 2.3 – Các thông tin mang theo trong bản tin CDP advertisement*

CDP có 2 đặc điểm:

* Giao thức này hoạt động tại tầng liên kết dữ liệu
* CDP chỉ hoạt động giữa các giao thức Cisco

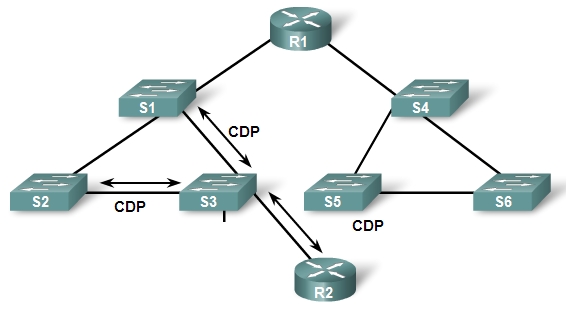
Về mặt vận hàng thì CDP là 1 giao thức rất đơn giản, tuy nhiên CDP đề cập tới khái niệm được sử dụng rất nhiều trong Networking, đó là khái niệm hàng xóm. Hàng xóm (hay Neighbor) là khái niệm chỉ *2 thiết bị mạng hay đối tượng chia sẻ chung đặc điểm nào đó, thể hiện quan hệ tương tác giữa 2 thiết bị đó.* Theo đó, kết hợp với mô hình OSI, chúng ta có 2 dạng hàng xóm hay gặp:

* Hàng xóm lớp 3
* Hàng xóm lớp 2

**Hàng xóm lớp 3 –** Là các thiết bị thuộc cùng 1 dải mạng với nhau. Chẳng hạn trong ví dụ ban đầu, R1 và R2 là hàng xóm vì cùng thuộc dải 172.168.2.0/24, R2 và R3 là hàng xóm vì cùng thuộc dải 192.168.1.0/24, nhưng R1 và R3 không phải hàng xóm vì chúng không chia sẻ dải mạng chung nào

**Hàng xóm lớp 2 –** Là các thiết bị chia sẻ chung kết nối *vật lý* trực tiếp. Chẳng hạn trong hình dưới, chúng ta có các cặp hàng xóm giữa Router-Switch, và giữa các Switch. Các thiết bị này không chia sẻ chung dải mạng, nhưng từng cặp có chung kết nối vật lý với nhau.

CDP được sử dụng để phát hiện các hàng xóm lớp 2 giữa các thiết bị Cisco



*Hình 2.4 – Giao thức CDP phát hiện hàng xóm giữa Router-Switch*

Trong ví dụ trên, R1 và R2 có thể là hàng xóm lớp 3 vì chúng kết nối với nhau qua 1 hạ tầng chuyển mạch, tuy nhiên chúng không phải hàng xóm lớp 2. Ngược lại, Router R2 và Switch S3 là 2 hàng xóm lớp 2 nhưng không phải hàng xóm lớp 3

Để kiểm tra hàng xóm kết nối trực tiếp qua CDP, chúng ta sử dụng lệnh

RouterX#show cdp neighbor

Để bật CDP trên từng interface, trong sub-mode interface chúng ta sử dụng lệnh

RouterX(config-if)#cdp enable

Để bật CDP trên toàn bộ interface, trong mode cấu hình chúng ta sử dụng lệnh

RouterX(config)#cdp run

Để tắt CDP tương ứng trên 1 interface hoặc trên tất cả interface, chúng ta thêm tiền tố no vào 2 câu lệnh trên

RouterX(config-if)#no cdp enable

RouterX(config)#no cdp run

## Định tuyến tĩnh

### Mục đích của định tuyến tĩnh

Mục đích của định tuyễn tĩnh, về cơ bản không đi ngoài mục đích của định tuyến: đó là học thông tin tuyến đường để tới 1 mạng đích. Trước khi xét đến các trường hợp sử dụng định tuyến tĩnh, ta xem xét các ưu điểm và nhược điểm của phương pháp này:

**Ưu điểm:**

* Giảm tải trên Router CPU, vì Router không phải tính toán hay xử lý nhiều dữ liệu định tuyến
* Không tốn băng thông để trao đổi thông tin định tuyến giữa các Router
* Có tính bảo mật, vì quá trình định tuyến hoàn toàn được điều khiển bởi admin

**Nhược điểm:**

* Vất vả, vì mỗi Router thêm thì người quản trị lại phải cấu hình lại tất cả câu lệnh định tuyến, càng nhiều tuyến đường, càng vất vả
* Khả năng mở rộng kém, mạng càng mở rộng càng khó quản lý tuyến đường
* Hỗ trợ kém các tính năng định tuyến động, chẳng hạn tự động thay đổi tuyến đường khi có sự cố xảy ra hoặc tự lựa chọn tuyến đường tối ưu hơn

Do các đặc điểm đó nên định tuyến tĩnh được triển khai với những mục đích cụ thể hơn như:

* **Định tuyến tới 1 mạng cụt (stub Network) –** Mạng cụt là mạng chỉ có kết nối tới từ 1 tuyến đường duy nhất. Khi đó chạy 1 giao thức định tuyến động giữa các Router sẽ là lãng phí tài nguyên, định tuyến tĩnh trong trường hợp này là phương án tối ưu
* **Tạo tuyến đường mặc định (default route) –** Tuyến đường mặc định không phải tuyến đường cụ thể tới 1 mạng cụ thể nào, tuyến đường mặc định sẽ được sử dụng khi Router không tìm được tuyến đường cụ thể để tới 1 mạng đích nào đó. Triển khai tuyến đường mặc định thường thông qua định tuyến tĩnh
* **Tạo tuyến đường tổng hợp (summary route) –** Tuyến đường tổng hợp là tuyến đường về mặt cú pháp tương tự tuyến đường tới mạng cụt, tuy nhiên dưới góc nhìn trong 1 topology cụ thể, tuyến đường tổng hợp là 1 tuyến đương có thể thay thế nhiều tuyến đường khác để tới các mạng khác nhau
* **Tạo tuyến đường null (null route) –** Null route hay Discard route là tuyến đường loại bỏ, hay nói cách khác các gói tin được định tuyến qua tuyến đường null thì sẽ bị loại bỏ mà nhưng đồng thời Router cũng không gửi bản tin thông báo nào về cho bên nhận

### Cấu hình định tuyến tĩnh

1. **Cú pháp định tuyến tĩnh**

Cú pháp để cấu hình định tuyến tĩnh là *ip route* tại mode *configure terminal*

Router(config)#ip route {địa chỉ IP mạng đích} {subnet mask} {địa chỉ IP next-hop | outgoing interface} {AD}

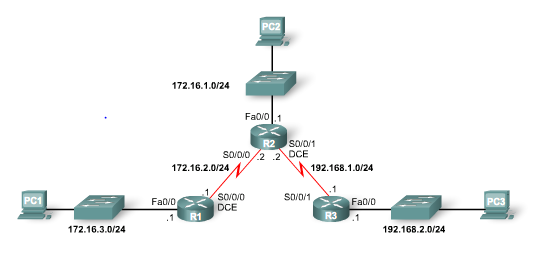
Trong đó:

* **{địa chỉ IP mạng đích}** – là địa chỉ mạng đầu xa cần định tuyến tới
* **{subnet mask}** – là mặt nạ của mạng đầu xa
* **{địa chỉ IP next-hop}** – là địa chỉ IP Router hàng xóm được dùng làm next-hop Router để tới mạng đích
* **{outgoing interface}** – là interface trên Router được sử dụng để đẩy gói tin ra khi định tuyến tới mạng đích
* **{AD} –** (optional) Chỉ số tin cậy để xác định mức độ ưu tiên cho tuyến đường, mặc định bằng 1

Các thông tin bắt buộc phải cung cấp để đảm bảo định tuyến tĩnh hợp lệ bao gồm *địa chỉ IP mạng đích, subnet mask* và *địa chỉ ip next-hop* hoặc *outgoing interface*

1. **Ví dụ minh họa**

Cho topology:



**Yêu cầu:** Cấu hình định tuyến tĩnh trên R1, R2 và R3 sao cho PC1, PC2 và PC3 có thể kết nối tới nhau

**Phân tích và triển khai:**

Để host PC1 có thể kết nối tới host PC2, tại Router R1 và R2 cần phải biết tuyến đường tới mạng đích tương ứng, R1 cần phải biết tuyến đường tới mạng 172.16.1.0/24, còn R2 cần phải biết tuyến đường tới mạng 172.16.3.0/24. Hiện tại trong bảng định tuyến của Router R1 và R2 tương ứng:

R1#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.3.0 is directly connected, FastEthernet0/0

R2#show ip route

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.1.0 is directly connected, FastEthernet0/0

C 172.16.2.0 is directly connected, Serial0/0/0

C 192.16.1.0 is directly connected, Serial0/0/1

Ta thấy R1 và R2 hiện tại chỉ đang có các mạng kết nối trực tiếp. Để PC1 có thể kết nối tới PC2, yêu cầu trên R1 phải có tuyến đường tới mạng 172.16.1.0/24 và R2 phải có tuyến đường tới mạng 172.16.3.0/24. Để thực hiện điều đó, ta triển khai định tuyến tĩnh trên R1 và R2 với cả 2 cách:

R1(config)#ip route 172.16.1.0 255.255.255.0 172.16.2.2

R2(config)#ip route 172.16.3.0 255.255.255.0 serial 0/0/0

*Cách thứ nhất* ta triển khai định tuyến tĩnh trên R1 với địa chỉ IP next-hop 172.16.2.2, đây là địa chỉ IP của interface s0/0/0 trên R2, hay chính là địa chỉ next-hop để tới mạng 172.16.1.0/24 từ R1.

*Cách thứ 2* ta triển khai định tuyến tĩnh trên R2 với outgoing interface serial0/0/0, đây là interface trên R2 được sử dụng để đẩy gói tin ra khi muốn tới mạng 172.16.3.0/24. Chúng ta sẽ kiểm tra quá trình định tuyến tĩnh bằng câu lệnh *debug*

R1#debug ip routing

R1#configure terminal

R1(config)#ip route 172.16.1.0 255.255.255.0 172.16.2.2

00:20:15: RT: add 172.16.1.0/24 via 172.16.2.2, static metric [1/0]

Khi ta cấu hình định tuyễn tĩnh cho mạng 172.16.1.0/24, ngay lập tức thông tin định tuyến được thêm vào bảng định tuyến (Routing table – RT). Chúng ta kiểm tra bảng định tuyến của 2 Router sau khi cấu hình

R1#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.3.0 is directly connected, FastEthernet0/0

S 172.16.1.0 [1/0] via 172.16.2.2

R2#show ip route

<output omitted> 172.16.0.0/24 is subnetted, 2 subnets

C 172.16.1.0 is directly connected, FastEthernet0/0

C 172.16.2.0 is directly connected, Serial0/0/0

C 192.16.1.0 is directly connected, Serial0/0/1

S 172.16.3.0/24 is directly connected, Serial0/0/0

Các thông tin cần quan tâm, ta chú ý tới 2 tuyến tĩnh (ký hiệu S)

|  |  |
| --- | --- |
| Show | Ý nghĩa |
| S | Staic - Tuyến tĩnh |
| 172.16.1.0 | Địa chỉ mạng đích |
| /24 | Tiền tố mạng của mạng đích |
| [1/0] | Chỉ số AD và metric |
| Via 172.16.2.2 | Địa chỉ IP next-hop |
| is directly connected, Serial0/0/0 | Outgoing interface trên R2 |

Ta thử kiểm tra kết nối từ R1 tới mạng 172.16.1.0/24 và từ R2 tới mạng 172.16.3.0/24 bằng câu lệnh ping:

R1#ping 172.16.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/40/40 ms

R2#ping 172.16.3.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 40/40/40 ms

Ta thấy lúc này R1 đã có thể kết nối tới mạng 172.16.1.0/24 và R3 có thể kết nối tới mạng 172.16.3.0/24. Như vậy PC1 có thể kết nối tới PC2

C:\Users\PC1>ping 172.16.1.1

Pinging 172.16.1.1 with 32 bytes of data:

Reply from 172.16.1.1: bytes=32 time<1ms TTL=128

Reply from 172.16.1.1: bytes=32 time<1ms TTL=128

Reply from 172.16.1.1: bytes=32 time<1ms TTL=128

Reply from 172.16.1.1: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.1.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

Chúng ta làm tương tự để tạo kết nối giữa PC2 và PC3 bằng cách tạo 2 tuyến định tuyến tĩnh:

* Trên R2 tới mạng 192.168.2.0/24
* Trên R3 tới mạng 172.16.1.0/24

Cuối cùng tạo kết nối giữa PC1 và PC3 bằng cách tạo 2 tuyến định tuyến tĩnh:

* Trên R1 tới mạng 192.168.2.0/24
* Trên R3 tới mạng 172.16.3.0/24

R1(config)#ip route 192.168.2.0 255.255.255.0 172.16.2.2

R2(config)#ip route 192.168.2.0 255.255.255.0 192.168.1.1

R3(config)#ip route 172.16.1.0 255.255.255.0 192.168.1.2

R3(config)#ip route 172.16.3.0 255.255.255.0 192.168.1.2

Sau đó chúng ta kiểm tra kết nối giữa PC1-PC3 và PC2-PC3 bằng câu lệnh ping

C:\Users\PC1>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time<1ms TTL=128

Reply from 192.168.2.1: bytes=32 time<1ms TTL=128

Reply from 192.168.2.1: bytes=32 time<1ms TTL=128

Reply from 192.168.2.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\PC2>ping 192.168.2.1

Pinging 192.168.2.1 with 32 bytes of data:

Reply from 192.168.2.1: bytes=32 time<1ms TTL=128

Reply from 192.168.2.1: bytes=32 time<1ms TTL=128

Reply from 192.168.2.1: bytes=32 time<1ms TTL=128

Reply from 192.168.2.1: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.2.1:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

**Chú ý :** Khi muốn thay đổi tuyến đường tĩnh tới cùng 1 mạng đích, chúng ta cần phải xóa câu lệnh cũ đi bằng tiền tố no, phía trước câu lệnh cũ, sau đó mới cấu hình câu lệnh mới

Bảng định tuyến trên 3 Router sau khi hoàn thành cấu hình:

R1#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.3.0 is directly connected, FastEthernet0/0

S 172.16.1.0 [1/0] via 172.16.2.2

S 192.168.2.0/24 [1/0] via 172.16.2.2

R2#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.1.0 is directly connected, FastEthernet0/0

C 172.16.2.0 is directly connected, Serial0/0/0

C 192.16.1.0 is directly connected, Serial0/0/1

S 172.16.3.0/24 is directly connected, Serial0/0/0

S 192.168.2.0/24 [1/0] via 192.168.1.1

R3#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

S 172.16.1.0 [1/0] via 192.168.1.2

S 172.16.3.0 [1/0] via 192.168.1.2

C 192.168.2.0/24 [1/0] is directly connected, fastEthernet0/0

C 192.168.1.0/24 [1/0] is directly connected, Serial0/0/0

1. **3 Nguyên tắc (tiên đề) bảng định tuyến**

Khi chúng ta triển khai định tuyến tĩnh trong trường hợp trên bằng cách sử dụng 2 câu định tuyến tĩnh trên mỗi Router, liệu có gì đảm bảo sau đó kết nối giữa các PC thành công. Liệu các gói tin từ PC1 và PC2 có thể tới được PC3. 3 nguyên tắc bảng định tuyến được mô tả bởi Alex Zinin sẽ giải thích câu hỏi trên

**Nguyên tắc 1 –** Mỗi Router đưa ra quyết định định tuyến một cách *độc lập*, dựa vào thông tin chúng có trong bảng định tuyến.

Trong ví dụ trên, R1 có 2 static route (tuyến tĩnh) R1 sẽ hoàn toàn chỉ dựa vào thông tin bảng định tuyến của R1 để thực hiện chọn đường đi, mà không dựa vào bảng định tuyến của bất kỳ Router nào khác

**Nguyên tắc 2 –** Bảng định tuyến của mỗi Router độc lập với nhau, hay nói cách khác, không có cách nào đảm bảo thông tin định tuyến trong bảng định tuyến của mỗi Router được đồng bộ với nhau (cho dù là định tuyến tĩnh hay động).

Trong ví dụ trên, R1 muốn tới mạng 192.168.2.0/24 sẽ đẩy gói tin tới IP next-hop là 172.16.2.2 (R2) nhưng R1 không cách nào xác nhận liệu R2 có tuyến đường tới mạng 192.168.2.0/24 hay không.

**Nguyên tắc 3 –** Thông tin tuyến đường để tới một mạng đích trên Router đầu gần không đồng nghĩa sẽ cung cấp thông tin tuyến đường quay ngược lại về mạng nguồn trên Router đầu xa. Hay nói cách khác, 1 kết nối thành công chỉ khi Router đầu gần có định tuyến tới mạng đích, và Router đầu xa có định tuyến về mạng nguồn.

Trong ví dụ trên, nếu R1 có kết nối tới 192.168.2.0/24 không đồng nghĩa với việc PC1 có thể kết nối tới PC3, điều đó chỉ có nghĩa PC1 có thể gửi gói tin tới PC3, nhưng không chắc R3 có thể gửi gói tin phản hồi từ PC3 về PC1. Vì thế, định tuyến tĩnh phải đảm bảo thông tin tuyến đường được cung cấp cho cả chiều đi và chiều về

1. **Tra cứu đệ quy (Recursive Route Lookup)**

Trong ví dụ trên, và từ cấu trúc câu lệnh định tuyến tĩnh, ta thấy có 2 cách để tạo 1 tuyến tĩnh, bằng cách trỏ ra IP next-hop hoặc outgoing interface. Vậy sự khác biệt của 2 cách này là gì? Và từng cách 1 có những hạn chế hay đặc điểm như thế nào?

Trước khi nói về cách sử dụng với IP next-hop hay outgoing interface, ta cần làm quen với 1 thuật ngữ mới **Recursive Route lookup** (hay tra cứu đệ quy). Toàn bộ quá trình định tuyến bao gồm 2 bước sau:

1. TÌm tuyến đường phù hợp trong bảng định tuyến
2. Xác định outgoing interface để đẩy gói tin

Như vậy mục đích cuối cùng vẫn là tìm ra **outgoing interface** phù hợp để đẩy gói tin tới đích. Ta sẽ chia làm 3 trường hợp, để xem xét cách Router xác định như thế nào:

**Trường hợp 1 –** Mạng đích là 1 mạng có kết nối trực tiếp với Router, entry trong bảng định tuyến có cấu trúc như sau:

C 192.16.2.0 is directly connected, Serial0/0/0

Ta thấy, với mạng kết nối trực tiếp, entry trong bảng định tuyến đã xác định được ngay lập tức outgoing interface để đẩy gói tin, trong trường hợp này tra cứu bảng định tuyến diễn ra một lần

**Trường hợp 2 –** Mạng đích không phải là mạng có kết nối trực tiếp với Router, entry trong bảng định tuyến có cấu trúc như sau:

S 192.168.2.0/24 [1/0] via 172.16.2.2

Ta thấy xác định được IP next-hop vẫn chưa xác định được outgoing interface, nên Router sẽ thực hiện tra cứu đệ quy thêm 1 lần nữa để xác định để tới được next-hop 172.16.2.2 thì cần đẩy gói tin ra giao diện này. Router tiếp tục tra bảng định tuyến

C 192.16.2.0 is directly connected, Serial0/0/0

Ta thấy, để tới next-hop 172.16.2.2 cần đẩy ra mạng 172.16.2.0, đây là mạng có kết nối trực tiếp, nên dễ dàng xác định outgoing interface là Serial 0/0/0. Đến lúc này, dữ liệu mới chính thức được chuyển tiếp. Trong trường hợp này, Router đã thực hiện tra cứu bảng định tuyến 2 lần.

Tra cứu đệ quy là tiến trình tra cứu bảng định tuyến một hoặc nhiều lần, được thực hiện bởi Router nhằm xác định được outgoing interface để có thể chuyển tiếp (forward) gói tin tới mạng đích. Ta xem xét lại bảng định tuyến của Router R1 trong ví dụ trên để có cái nhìn tổng quát hơn:

R1#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.2.0 is directly connected, Serial0/0/0 🡪 **Bước 2**

C 172.16.3.0 is directly connected, FastEthernet0/0

S 172.16.1.0 [1/0] via 172.16.2.2 🡪 **Bước 1**

Trong nhiều trường hợp, quá trình tra cứu đệ quy sẽ diễn ra nhiều lần thay vì chỉ 1 hoặc 2 lần, vì đặc điểm 1 tuyến đường có thể đi qua rất nhiều node trung gian, mỗi node là 1 Router và các Router trao đổi với nhau thông qua giao thức định tuyến động, khi đó mỗi Router sẽ trỏ next-hop sang Router hàng xóm hoặc các Router phía trước Router hàng xóm để tham khảo. Khi đó quá trình tra cứu bảng định tuyến sẽ diễn ra nhiều lần

R1(config)#ip route 192.168.2.0 255.255.255.0 192.168.1.1

R1(config)#ip route 192.168.1.0 255.255.255.0 172.16.2.2

R1(config)#exit

R1#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.2.0 is directly connected, Serial0/0/0 🡪 **Bước 3**

S 172.16.1.0 [1/0] via 172.16.2.2

C 172.16.3.0 is directly connected, FastEthernet0/0

S 192.168.1.0/24 [1/0] via 172.16.2.2 🡪 Bước 2

S 192.168.2.0/24 [1/0] via 192.168.1.1 🡪 **Bước 1**

**Ý nghĩa của tra cứu đệ quy –** Tra cứu đệ quy cho phép 1 Router xác định được outgoing interface từ địa chỉ IP next-hop. Không chỉ vậy, tra cứu đệ quy xác định liệu 1 tuyến tĩnh có được phép đưa vào bảng định tuyến tĩnh. Trong trường hợp với 1 tuyến tĩnh mà quá trình tra cứu đệ quy không thể xác định được outgoing interface từ IP next-hop, tuyến tĩnh đó sẽ không được đưa vào bảng định tuyến. Trong ví dụ trên, chúng ta có 3 tuyến đường tĩnh trên Router R1, và chúng đều được tra cứu đệ quy ra giao diện Serial 0/0/0. Trong trường hợp giao diện Serial0/0/0 không còn, hoặc không tồn tại, hãy xem điều gì xảy ra:

R1#debug ip routing

IP routing debugging is on

R1#config t

Enter configuration commands, one per line. End with CNTL/Z.

R1(config)#interface s0/0/0

R1(config-if)#shutdown

R1(config-if)#end

Is\_up: 0 state: 6 sub state: 1 line: 0

RT: interface serial0/0/0 removed from routing table

RT: del 172.16.2.0/24 via 0.0.0.0, connected metric [0/0]

RT: delete subnet route to 172.16.2.0/24

RT: del 192.16.1.0 via 172.16.2.2. static metric [1/0]

RT: delete network route to 192.168.1.0

RT: del 172.16.1.0 via 172.16.2.2. static metric [1/0]

RT: delete network route to 172.16.1.0

RT: del 192.168.2.0/24 via 192.168.1.1, static metric [1/0]

RT: delete subnet route to 192.168.2.0/24

Ta thấy, khi giao diện Serial0/0/0 bị đóng (shutdown) tất cả tuyến tĩnh được đệ quy tới đều bị xóa khỏi bảng định tuyến

1. **Outgoing interface thay vì IP Next-hop**

Chúng ta quay lại cách thứ 2 để cấu hình định tuyến tĩnh đã đề cập ở trên, ngoài cú pháp:

Router(config)#ip route {Địa chỉ mạng đích} {Subnet mask} {IP Next-hop}

Ta có thể sử dụng:

Router(Config)#ip route {Địa chỉ mạng đích} {Subnet mask} {Outgoing interface}

Như vậy, quay lại với topology trong ví dụ ban đầu, để cấu hình tuyến tĩnh tới mạng 172.16.1.0/24 trên R1, ta có 2 cách:

**Cách 1 –** R1(config)#ip route 172.16.1.0 255.255.255.0 172.16.2.2

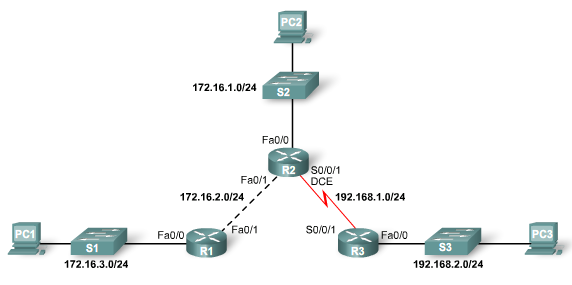
**Cách 2 –** R1(config)#ip route 172.16.1.0 255.255.255.0 serial0/0/0

Lợi ích rõ ràng nhất của việc sử dụng cách 2 đó là Router chỉ cần tra cứu bảng định tuyến 1 lần duy nhất thay vì 2 lần so với cách thứ 2. Khi đó dạng của tuyến tĩnh trong bảng định tuyến sẽ dưới dạng

S 192.168.2.0/24 is directly connected, Serial0/0/0

Chúng ta thấy Router sẽ không còn quá trình tra cứu từ next-hop ra outgoing interface nữa, mà trực tiếp xác định outgoing interface là Serial0/0/0 từ bảng định tuyến ngay từ lần tra cứu đầu tiên. Tuy vậy có sự khác biệt lớn giữa việc sử dụng IP Next-hop và outgoing interface

1. **Static route và topology**



Câu hỏi đặt ra là: **nếu Tuyến tĩnh trỏ ra outgoing interface, trong môi trường đa truy cập, vậy quá trình đóng gói sẽ diễn ra như thế nào để gói tin có thể tới đúng Router next-hop** ? Môi trường đa truy cập có thể là Ethernet, trước khi đề cập tới Static route và Ethernet, ta xem xét quá trình đóng gói trong ví dụ ban đầu, nghĩa là môi trường kết nối là điểm-điểm, giao thức chạy trên Serial là HDLC hoặc PPP (Point to Point protocol)

Đối với các giao thức hỗ trợ kết nối điểm-điểm chạy trên Serial như HDLC hoặc PPP, trường địa chỉ lớp 2 sẽ mang thông tin địa chỉ đích là 0xFF, hay địa chỉ Broadcast. Điều này hợp lý, vì kết nối điểm-điểm chỉ có 2 node nằm trên 2 đầu của kết nối, vì thế khi 1 gói tin đi ra Interface mặc nhiên sẽ phải tới node đầu bên kia. Quay lại với static route sử dụng outgoing interface, như vậy, với kết nối là điểm-điểm như Serial, Router sẽ không cần quan tâm tới địa chỉ đích lớp 2. Do đó, static route sử dụng outgoing interface là giải pháp lý tưởng cho các kết nối điểm-điẻm

Đối với môi trường kết nối là đa truy nhập như Ethernet, 1 gói tin được chuyển đi sẽ phải đóng gói địa chỉ MAC đích la node đầu xa, nếu địa chỉ MAC đích lớp 2 cũng là 0xFF thì không hợp lý vì môi trường đa truy cập cho phép nhiều node chia sẻ môi trường truyền. Vì thế yêu cầu bắt buộc là Router phải bằng cách nào đó xác định được địa chỉ MAC đích lớp 2 của Router next-hop, để làm được điều đó người ta sử dụng 1 giao thức **Proxy ARP** – Về mặt ý tưởng, Proxy-ARP phản hồi bản tin ARP hỏi về 1 mạng nằm ngoài mạng vừa nhận được bản tin ARP đó. Tuy nhiên chúng ta sẽ chưa đề cập tới kỹ thuật đó trong chương trình CCNA. Về mặt thiết kế, chúng ta phân biệt được các trường hợp để sử dụng định tuyến tĩnh một cách hợp lý.

Để có thể tối ưu đồng thời cả 2 tính năng: sử dụng outgoing interface để “tiết kiệm” 1 lần tra cứu đệ quy, và sử dụng địa chỉ IP next-hop để tránh việc Router không thể đóng gói Ethernet thành công, Cisco IOS đưa ra 1 cách thứ 3 tích hợp đồng thời cả 2 cách 1 và 2 bên trên:

**Cú pháp:**

Router(config)#ip route {địa chỉ mạng đích} {subnet mask} {outgoing interface} {IP next-hop}

Ta thấy với phương pháp này, tuyến đường trong bảng định tuyến đã xác định ngay lập tức giao diện đẩy gói tin ra, đồng thời địa chỉ MAC của Router hàng xóm dùng để đóng gói bản tin. Trong trường hợp Router chưa xác định được địa chỉ MAC của Router hàng xóm, giao thức ARP sẽ được sử dụng. So với giao thức Proxy-ARP, thì giao thức ARP đảm bảo về mặt bảo mật hơn

**Ví dụ** Trên Router R1

R1(config)#ip route 192.168.2.0 255.255.255.0 fastethernet 0/1 172.16.2.2

## Ứng dụng định tuyến tĩnh

### Tuyến đường mặc định (default route)

Tuyến đường mặc định là tuyến đường được sử dụng khi trong bảng định tuyến không có tuyến đường nào khác để tới 1 mạng đích.

Tuyến đường mặc định được sử dụng ở khắp mọi nơi, từ mô hình mạng LAN nhỏ SOHO cho tới mạng Campus, mạng Enterprise, và thậm chí là mạng của các ISP. Tuyến đường mặc định phản ảnh một đặc điểm cơ bản của định tuyến, đó là “Không một Router nào có thể học tất cả tuyến đường tới tất cả mạng đích” – Từ người dùng, sẽ trỏ default route tới ISP, từ ISP sẽ trỏ default route tới các ISP cấp cao hơn.

Điều này cũng phản ánh cấu trúc cấp phát địa chỉ IP từ IANA 🡪 tổ chức quản lý địa chỉ cấp vùng (Việt Nam thuộc sự quản lý APNIC) 🡪NIR (VNNIC) 🡪 ISP (Viettel)

Đối với doanh nghiệp, tuyến đường mặc định được sử dụng tại các Router biên và trỏ tới mạng của ISP. Cú pháp:

RouterX(config)#ip route 0.0.0.0 0.0.0.0 [IP next-hop | Outgoing interface ]

Trong đó 0.0.0.0 0.0.0.0 được sử dụng thay cho {địa chỉ IP mạng} {Mặt nạ mạng}. Địa chỉ 0.0.0.0 0.0.0.0 được gọi là địa chỉ *Quad-zero*, 1 địa chỉ đặc biệt khớp với mọi mạng đích, do đó khi có defalt route, Router sẽ luôn có thể chuyển tiếp gói tin, cho dù không có thông tin tới cụ thể tới mạng đích nào đó

Quay lại ví dụ ban đầu, và xét Router R1. Ta thấy, để mạng LAN của R1 có thể tới tất cả các mạng, Router R1 cần cấu hình 3 tuyến tĩnh tới các mạng 172.168.1.0/24, 192.168.1.0/24 và 192.168.2.0/24, như vậy bảng định tuyến có 3 bản ghi định tuyến tĩnh. Chúng ta thấy để tới cả 3 mạng đích trên, R1 đều chuyển tiếp gói tin sang Router R2. Như vậy, ta có thể thay 3 câu định tuyến tĩnh ban đầu bằng 1 default route duy nhất (chú ý xóa câu lệnh định tuyến cũ trước khi sử dụng câu lệnh định tuyến tĩnh mới)

R1(config)#no ip route 172.16.1.0 255.255.255.0 172.168.2.2

R1(config)#no ip route 192.168.1.0 255.255.255.0 172.168.2.2

R1(config)#no ip route 192.168.2.0 255.255.255.0 172.168.2.2

!

R1(config)#ip route 0.0.0.0 0.0.0.0 172.168.2.2

Chúng ta kiểm tra lại bảng định tuyến trên Router R1

R1#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.3.0 is directly connected, FastEthernet0/0

S\* 0.0.0.0/0 is directly connected, Serial0/0/0

Ký hiệu \* xác định đây là tuyến đường default route

### Tuyến đường tổng hợp (summary route)

Một trong những bài toán của định tuyến đó là ngoài việc đảm bảo kết nối tới tất cả các mạng, cần tối ưu không gian bảng định tuyến. Bảng định tuyến càng nhỏ quá trình tra cứu diễn ra càng hiệu quả và ngược lại. Ngoài ra, mọi Router đều có giới hạn số lượng bản ghi cho phép trong bảng định tuyến, ngay cả các Router mạnh nhất đặt tại trung tâm mạng Internet cũng không thể xem thường, số lượng tuyến đường được lưu giữ trong bảng định tuyến cũng chỉ … 400,000 tuyến (thay vì lẽ ra phải hàng triệu tuyến để tới tất cả các mạng trên Internet). Ý tưởng của summary route rất đơn giản, “tạo 1 tuyến đường mới có thể đại diện cho nhiều tuyến đường nhỏ hơn” – Hay nói cách khác, summary route là cách triển khai gộp tuyến “giảm nhẹ” so với default route, nếu default route đại diện cho *tất cả* tuyến đường, thì summary route đại diện cho 1 nhóm tuyến đường

Một so sánh đơn giản với summary route đó là các mạng major – hay các mạng được chia theo lớp (Classful Addressing) – có thể được dùng để đại diện cho tất cả mạng con của nó. Chẳng hạn, tập hợp các mạng 10.0.0.0/16, 10.1.0.0/16, 10.2.0.0/16, 10.3.0.0/16,… cho tới 10.255.0.0/16, đều có thể được đại diện bởi 1 mạng lớp A duy nhất 10.0.0.0/8. Theo logic đó, nếu 256 mạng con trên có thể truy cập tới thông qua 1 outgoing interface hoặc địa chỉ IP next-hop duy nhất, thì thay vì 256 tuyến đường tới các mạng con đó, ta chỉ cần 1 tuyến đường duy nhất tới mạng 10.0.0.0/8 với cùng outgoing interface hoặc IP next-hop. Kỹ thuật gộp 256 tuyến đường thành 1 tuyến đường đại diện như vậy được gọi là kỹ thuật gộp tuyên (route summarizaion)

**Cú pháp gộp tuyến**

Cú pháp của gộp tuyến đường không khác một chút nào so với cú pháp cấu hình định tuyễn tĩnh

Router(config)#ip route {địa chỉ mạng đích} {subnet mask} {outgoing interface | IP next-hop}

**Kỹ thuật gộp tuyến**

Hãy xét ví dụ ban đầu, Router R3 cần 3 câu lệnh định tuyến tĩnh để tới 3 mạng con 172.16.1.0/24, 172.16.2.0/24 và 172.16.3.0/24. Yêu cầu triển khai gộp tuyến sao cho chỉ cần 1 câu lệnh định tuyễn tĩnh duy nhất đại diện cho 3 tuyến đường trên.

Như vậy chúng ta cần gộp 3 mạng con: 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24 thành 1 mạng duy nhất:

🡪 Bước 1: viết tất cả địa chỉ mạng con dưới dạng nhị phân, thẳng hàng

|  |  |
| --- | --- |
| Thập phân | Nhị phân |
| 172.16.1.0 | 10101100.00010000.00000001.00000000 |
| 172.16.2.0 | **10101100.00010000.00000010.00000000** |
| 172.16.3.0 | **10101100.00010000.00000011.00000000** |

🡪 Bước 2: Đếm số bit trùng nhau nhiêu nhất từ trái qua, đó chính là subnet mask của mạng gộp

|  |  |
| --- | --- |
| Thập phân | Nhị phân |
| 172.16.1.0 | 10101100.00010000.000000**01.00000000** |
| 172.16.2.0 | **10101100.00010000.000000**10.00000000 |
| 172.16.3.0 | **10101100.00010000.000000**11.00000000 |

Khớp 22 bit, như vậy subnet mask của mạng gộp bằng 255.255.252.0

🡪 Bước 3: Giữ nguyên số bit trùng nhau, các bit còn lại gán bằng 0, xác định địa chỉ mạng gộp, chuyển ngược lại thập phân

|  |  |  |
| --- | --- | --- |
|  | Thập phân | Nhị phân |
| Mạng con 1 | 172.16.1.0 | 10101100.00010000.000000**01.00000000** |
| Mạng con 2 | 172.16.2.0 | **10101100.00010000.000000**10.00000000 |
| Mạng con 3 | 172.16.3.0 | **10101100.00010000.000000**11.00000000 |
| Mạng gộp | 172.16.0.0 | **10101100.00010000.00000000.00000000** |

Như vậy, mạng gộp cho 3 mạng con trên là 172.16.0.0/22

Như vậy ta cấu hình lại:

R3(config)#no ip route 172.16.1.0 255.255.255.0 192.168.1.2

R3(config)#no ip route 172.16.2.0 255.255.255.0 192.168.1.2

R3(config)#no ip route 172.16.3.0 255.255.255.0 192.168.1.2

!

R3(config)# ip route 172.16.0.0 255.255.252.0 192.168.1.2

Kiểm tra bảng định tuyến lúc trước

R3#show ip route

<output omitted>

172.16.0.0/24 is subnetted, 2 subnets

S 172.16.1.0 [1/0] via 192.168.1.2

S 172.16.2.0 [1/0] via 192.168.1.2

S 172.16.3.0 [1/0] via 192.168.1.2

C 192.168.2.0/24 [1/0] is directly connected, fastEthernet0/0

C 192.168.1.0/24 [1/0] is directly connected, Serial0/0/0

Bảng định tuyến lúc sau

R3#show ip route

<output omitted>

172.16.0.0/22 is subnetted, 2 subnets

S 172.16.0.0 [1/0] via 192.168.1.2

C 192.168.2.0/24 [1/0] is directly connected, fastEthernet0/0

C 192.168.1.0/24 [1/0] is directly connected, Serial0/0/0

Kiểm tra kết nối từ R3 tới các mạng sau khi cấu hình định tuyến gộp tuyến:

R3#ping 172.16.1.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 3/11/41 ms

!

R3#ping 172.16.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.2.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 3/11/41 ms

!

R3#ping 172.16.3.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.16.3.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 3/11/41 ms

## Tổng kết

Kết thúc chương này, chúng ta đã có cái nhìn khá chi tiết về:

* Mạng kết nối trực tiếp
* Cấu hình cơ bản interface fastEthernet và interface Serial
* Giao thức CDP được dùng để khám phá hàng xóm lớp 2
* Định tuyến tĩnh và ứng dụng

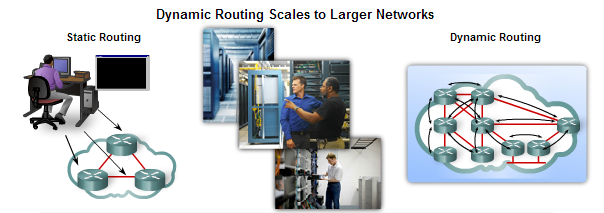
# ****CHƯƠNG 3 - ĐỊNH TUYẾN ĐỘNG****

Bài toán định tuyến là bài toán cơ bản trong thiết kế hạ tầng mạng, những yêu cầu nảy ra khi quy mô mạng lớn dần lên, định tuyến cần trở nên năng động hơn nữa và mạnh mẽ hơn nữa. Hàng trăm, hàng ngàn tuyến đường cần được học khiến cho định tuyến tĩnh “hụt hơi”. Những vấn đề liên quan đến dự phòng, tính thừa thãi để đảm bảo hạ tầng duy trì uptime trên 90%. Các chính sách và kỹ thuật làm việc với tuyến đường trở thành tất yếu để quản lý luồng dữ liệu. Tất cả những yếu tố đó đặt ra 1 yêu cầu về một cách thức định tuyến mới – Ta gọi đó là định tuyến động – Dynamic Routing

Định tuyến động, sinh ra là để giải những bài toán mà định tuyến tĩnh không thể làm được, và vẫn tiếp tục phát triển cho tới bây giờ. Rất nhiều giải pháp định tuyến động được đưa ra, và để có thể triển khai trong mạng doanh nghiệp lẫn ISP, người quản trị cần phải có kiến thức vững vàng về các giao thức định tuyến động này.

Trong bài này ta sẽ làm quen với :

* Mô tả khái niệm định tuyến động ( dynamic routing) và những nơi mà giao thức này được triển khai trong vùng mạng
* Phân loại các giao thức định tuyến động
* Các thành phần của mọi giao thức định tuyến động



*Hình 3.1 – Sử dụng định tuyến động thay thế định tuyến tĩnh*

## Sơ lược về định tuyến động

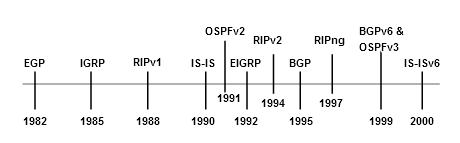
### Lịch sử các giao thức định tuyến động

Giao thức định tuyến động được sử dụng từ đầu những năm 1980. RIP là một trong những giao thức định tuyến động được sử dụng rộng rãi đầu tiên vào năm 1982, tuy nhiên thuật toán lúc đó thì gần giống với giao thức được triển khai trong mạng ARPANET (tiền thân của Internet) của những năm 1969, sau đó mới được chuẩn hóa bởi IETF thông qua RFC 1058 vào năm 1988. 4 năm sau RIP đưa ra phiên bản RIPv2, đưa vào thêm các tính năng khắc phục một số các vấn đề của RIPv1. Nhược điểm lớn nhất của RIP đó là khả năng mở rộng kém, không thể triển khai trong các mô hình mạng lớn. Nhìn ra vấn đề đó nên trong khoảng thời gian rất ngắn 4 năm giữa 2 phiên bản của RIP, các giao thức định tuyến mới mạnh mẽ hơn được phát triển và chuẩn hóa, bao gồm IS-IS (bởi ISO, năm 1990) OSPFv2 (cho Ipv4, bởi IETF năm 1991) EIGRP (bởi Cisco, năm 1992). Hiện nay các giao thức này là vẫn là các giao thức định tuyến động nền tảng của hầu hết mọi mô hình mạng doanh nghiệp hoặc lõi sử dụng Ipv4.

Internet phát triển với tốc độ chóng mặt, và các nhà phát triển cũng ngay lập tức nhận ra giới hạn của các giao thức định tuyến trên, do cách tiếp cận khác nhau – RIP, IS-IS, OSPF, hay EIGRP được thiết kế cho mô hình 1 mạng doanh nghiệp, có thể triển khai trong các mạng lõi của ISP, tuy nhiên Internet dẫn đến nhu cầu kết nối các liên mạng, số lượng tuyến đường yêu cầu được định tuyến trên các Router vượt quá khả năng xử lý. Nhu cầu tất yếu là cần một giao thức mới làm nhiệm vụ định tuyến liên mạng trên Internet, và BGP được chuẩn hóa vào năm 1995. Hiện nay BGP điều khiển toàn bộ định tuyến trên Internet và thường được triển khai bởi các ISP

Bài toán định tuyến xem như đã được giải quyết tạm thời, tuy nhiên nhu cầu kết nối không bao giờ là đủ. Ipv4 đạt đến giới hạn phát triển của mình và thực sự đã hoàn thành sứ mệnh lịch sử là xây dựng một nền tảng Internet mở, mạnh mẽ và ổn định. Ipv6 trở thành tương lai của Internet, giao thức IP thế hệ tiếp theo này đặt ra các bài toán mới, trong đó có bài toán định tuyến. Cần phải có các giao thức định tuyến riêng hỗ trợ Ipv6. Và ngay lập tức, dường như song song cùng sự phát triển của Ipv6 là các giao thức định tuyến riêng đi kèm bao gồm RIPng (nền tảng là RIPv2, năm 1997) OSPFv3 (nền tảng là OSPFv2, năm 1999) và BGPv6 cùng năm đó, IS-Isv6 (nền tảng từ IS-IS, năm 2000)

Tương lai của các giao thức định tuyến động không nên chỉ nhìn nhận ở khía cạnh tìm đường tối ưu nữa, thay vào đó là hỗ trợ triển khai các nền tảng ứng dụng và dịch vụ một cách hiệu quả, chẳng hạn Multicast, QoS, VPN hay Security.



*Hình 3.2 - Lịch sử các giao thức định tuyến động*

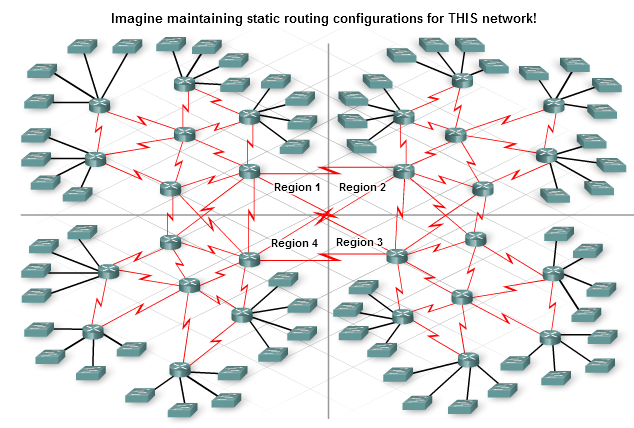
### Định tuyến động vs Định tuyến tĩnh

Trước khi đi sâu vào các giao thức định tuyến động, chúng ta thử so sánh những ưu điểm và nhược điểm của định tuyến động so với định tuyến tĩnh.

|  |  |  |
| --- | --- | --- |
| Yêu cầu | Định tuyến động | Định tuyến tĩnh |
| Mức độ cấu hình phức tạp | Không thường phụ thuộc vào kích thước mạng | Mạng càng phức tạp, triển khai càng vất vả |
| Yêu cầu kiến thức quản trị | Hiểu biết về cách vận hành và cấu hình của từng giao thức | Ít kiến thức, cấu hình đơn giản |
| Tốc độ phản xạ với thay đổi | Tự động thích ứng khi có thay đổi | Không tự thay đổi, yêu cầu người quản trị mạng phải tham gia |
| Khả năng mở rộng | Phù hợp cho các mọi quy mô mạng từ đơn giản tới phức tạp | Chỉ thích hợp cho các mô hình mạng rất nhỏ |
| Yêu cầu tài nguyên | Nhiều, bao gồm: CPU, bộ nhớ, băng thông | Không đáng kể |
| Tỉnh ổn định (bảo mật) | Ít ổn định hơn, Router tự điều chỉnh quá trình định tuyến | Rất ổn định, do quá trình lựa chọn tuyến đường bởi admin |
| Tính linh hoạt | Tuyến đường thay đổi phụ thuộc vào mô hình | Tuyến đường tới đich là cố định |

Ta đã phân tích các đặc điểm của định tuyến tĩnh trong bài trước, các vấn đề của định tuyến tĩnh đều được trực tiếp giải quyết bởi định tuyến động. Định tuyến động thông minh hơn và hỗ trợ nhiều tính năng hơn, nhưng ngược lại yêu cầu người quản trị phải có trình độ, vì không chỉ cấu hình, quá trình xác nhận, theo dõi, gỡ rối định tuyến động luôn phức tạp, triển khai càng ít đầu tư chất xám vào thiết kế, các bài toán gỡ rối càng trở nên khó khăn, chúng ta cũng biết rằng, không có một mạng nào ổn định tuyệt đối, mạng càng phức tạp, càng hay xảy ra vấn đề. Con người không phải mặt cần đầu tư duy nhất khi triển khai định tuyến động, Router phải mạnh mẽ hơn, tài nguyên mạng như băng thông phải nhiều hơn, bộ nhớ cũng theo đố cần phải tăng cường, hệ điều hành cũng phải nâng cấp. Vì thế, chúng ta không nên đứng ở khía cạnh định tuyến động là giải pháp thay thế cho định tuyến tĩnh, định tuyến động là giải pháp định tuyến duy nhất đối với mạng có xu hướng mở rộng, bất kể lúc đầu mạng đó có quy mô nhỏ hay lớn. Nâng cao hơn, định tuyến động chính là nền tảng cung cấp dịch vụ hỗ trợ kết nối cho các ứng dụng

Để thấy được vai trò và sức mạnh của định tuyến động so với định tuyến tĩnh, hãy thử xét 1 topology như sau:



*Hình 3.3 – Nếu phải định tuyến tĩnh với topology này thì quả khủng khiếp*

Liệu ai có thể hình dung cần bao nhiêu câu lệnh định tuyến tĩnh được cấu hình trên mỗi Router, và chúng ta có 28 Router trong topology trong hình bên. Hãy hình dung khi có thay đổi về 1 mạng, sẽ phải thay đổi bao nhiêu câu lệnh trước đó. Hãy hình dung khả năng admin có thể cấu hình lỗi 1 con số trong địa chỉ IP next-hop, hãy hình dung số lượng tuyến đường xuất hiện trong bảng định tuyến hoàn toàn phẳng, không có một cấu trúc đặc biệt nào để quản lý… Tất cả vấn đề đó dẫn đến kết luận tất yếu là cần phải triển khai định tuyến động.

### Mục đích của giao thức định tuyến động

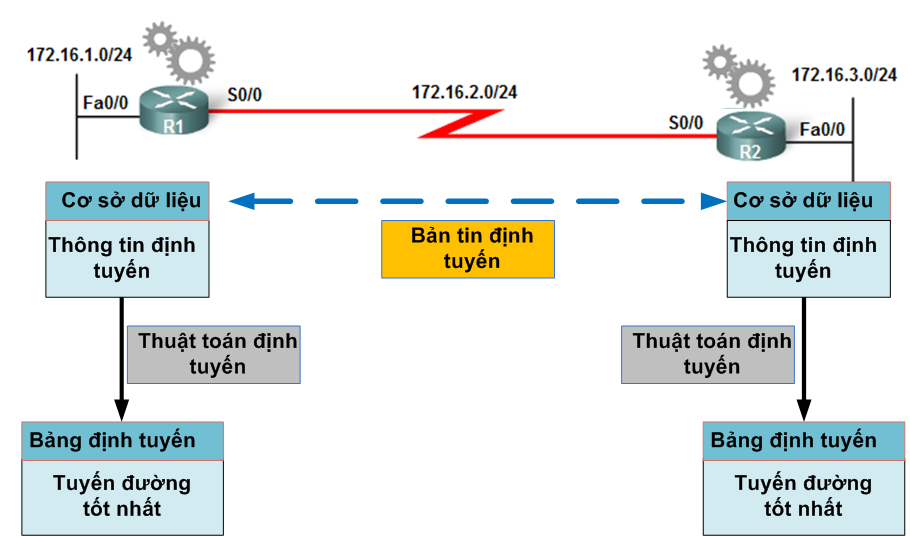
Định tuyến động, không phải là một tiến trình đơn lẻ, đó là một tập hợp các tiến trình. Mục đích cơ bản của một giao thức định tuyến động không nằm ngoài mục đích của định tuyến: đó là tìm đường đi tốt nhất và chuyển tiếp gói tin giữa các mạng. Điều khác biệt cốt lõi nhất, đó là với định tuyến động, người quản trị không bao giờ phải trực tiếp điều khiển Router “Muốn tới 1 mạng đích A, hãy đẩy gói tin ra interface B hoặc đẩy gói tin sang Next-hop x.x.x.x” về cơ bản, người quản trị sẽ bảo Router “Hãy quảng bá các mạng của cậu, học thông tin về các mạng khác, và tự xây dựng tuyến đường tới các mạng đó” Như thế, việc lựa chọn outgoing interface thuộc thẩm quyền của Router, Router sẽ tự biết khi nào cần sử dụng outgoing interface nào, và khi nào mạng đích không tồn tại hoặc không hợp lệ để có thể xóa tuyến đường tới đó trong bảng định tuyến. Đó là tính “động” thứ nhất của Định tuyến động

Tính “Động” thứ 2, nằm ở đặc điểm, để có thể thực hiện được các mục tiêu trên, các Router sẽ phải trao đổi thông tin định tuyến với nhau, một cách chủ động, thông qua các bản tin điều khiển và bản tin định tuyến. Do đặc điểm đó nên Router luôn xây dựng được một bảng cơ sở dữ liệu về các mạng đích và tuyến đường tới các mạng đích đó.

Tính “Động” thứ 3, ít được đề cập, nhưng quan trọng không kém, đó là khả năng duy trì độ ổn định trong mạng. Khác với định tuyến tĩnh phụ thuộc vào người quản trị, thì định tuyến động do Router hoàn toàn phụ trách việc chọn lựa tuyến đường, nếu 1 tuyến đường lỗi hoặc không thể dẫn gói tin tới được mạng đích (loop) thì Router cũng không biết cách để sửa lại, vì thế đối với 1 giao thức định tuyến, bài toán phải giải quyết không chỉ lựa chọn tuyến đường, mà còn phải có các cơ chế phòng chống những tình huống không mong muốn xảy ra

Để đảm bảo 3 tính “động” như trên, 1 giao thức định tuyến động thường bao gồm 3 thành phần:

* **Cấu trúc dữ liệu (*Data s*tructures)** – Chính là bảng cơ sở dữ liệu về mạng đích và các tuyến đường tới mạng đích đó. Bảng cơ sở dữ liệu được xây dựng dựa trên các thông tin định tuyến được trao đổi, sẽ cung cấp cho Router các lựa chọn để tìm ra tuyến đường tốt nhất. Thông tin này thường được lưu trữ trong RAM nên sẽ bị mất khi thiết bị ngừng hoạt động
* **Thuật toán (Algorithm)** – Thuật toán chính là bộ não của 1 giao thức định tuyến. Bao gồm một danh sách các bước giới hạn (finite state machine) Router phải làm để thực hiện một tác vụ. Tùy vào mức độ phức tạp của thuật toán sẽ quyết định mức độ phức tạp và khả năng xử lý của 1 giao thức định tuyến động. Các thuật toán làm nhiệm vụ điều khiển trao đổi thông tin định tuyến, lựa chọn tuyến đường tối ưu và chống loop
* **Bản tin định tuyến (Routing protocol messages)** – Bản tin định tuyến là công cụ để các Router trao đổi thông tin định tuyến. Tùy vào giao thức định tuyến mà số loại cũng như mục đích của các bản tin định tuyến sẽ khác nhau



*Hình 3.4 – Các thành phần của 1 giao thức định tuyến động*

Dựa vào các thành phần cơ bản mọi giao thức định tuyến động phải có, quá trình vận hành cơ bản của một giao thức định tuyến bao gồm các bước sau:

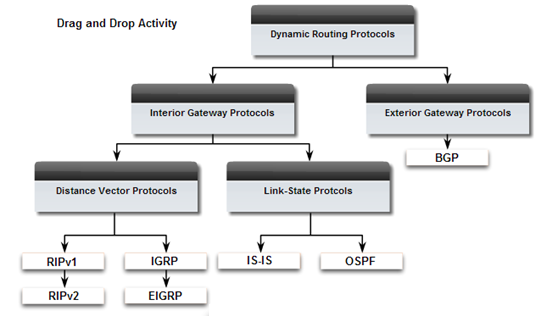
1. Các Router có kết nối trực tiếp (vật lý hoặc logic) chạy cùng 1 giao thức định tuyến sẽ thiết lập các mối quan hệ cần thiết
2. Khi mối quan hệ hình thành, các Router sẽ bắt đầu chia sẻ các thông tin tuyến đường về các mạng đích thông qua các bản tin định tuyến
3. Nhận thông tin định tuyến từ Router hàng xóm, xử lý và xây dựng cơ sở dữ liệu định tuyến
4. Dựa trên cơ sở dữ liệu, xác định tuyến đường tối ưu với từng mạng đích
5. Khi mạng xảy ra xử cố, thông báo sự cố tới tất cả các Router chạy cùng giao thức định tuyến về sự cố, đồng thời đưa ra các điều chỉnh phù hợp
6. Khi 1 Router rời mạng hoặc ngừng tham gia định tuyến, thông tin định tuyến được quảng bá bởi Router đó sẽ bị xóa trên các Router hàng xóm, và cập nhật dần trên toàn mạng

Mọi giao thức định tuyến, đều thực hiện tất cả các bước như trên, trong toàn bộ tiến trình.

## Phân loại giao thức định tuyến động

Các giao thức định tuyến có thể được phân loại dựa vào các đặc điểm của chúng. Các giao thức định tuyến phổ biến bao gồm

* RIPv1 và RIPv2
* IGRP
* OSPF
* IS-IS
* EIGRP
* BGP



*Hình 3.5 – Phân loại các giao thức định tuyên động*

### Phân loại dựa trên miền quản trị – IGP vs EGP

Trước khi đi vào các giao thức định tuyến, chúng ta làm quen với khái niệm **miền tự trị-AS (autonomous system).** Khái niệm AS tương đối trừu tượng:

Theo định nghĩa của VNNIC: AS là một tập hợp các mạng có cùng chính sách định tuyến và thường thuộc quyền quản lý, khai thác của một chủ thể.

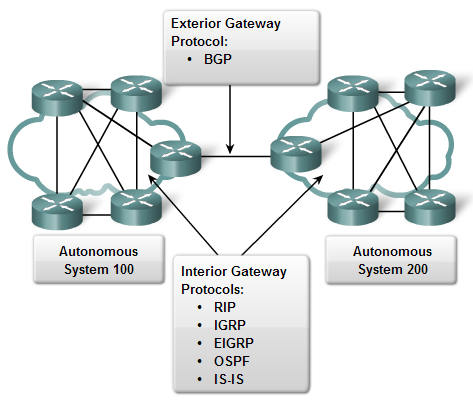
Theo IETF: AS là một tập hợp các Router dưới cùng một quyền quản lý kỹ thuật, sử dụng một hoặc nhiều giao thức định tuyến nội (IGP) với một tập hợp metric để thực hiện quá trình định tuyến gói tin bên trong AS đó, và sử dụng 1 giao thức liên-AS để thực hiện định tuyến giữa các AS – Khái niệm miền tự trị nhấn mạnh vào việc quản trị định tuyến bên trong 1 AS là độc lập với 1 AS khác

Như vậy, ta hiểu: AS, thứ nhất, phải là một miền định tuyến (routing domain), như vậy cấu trúc topology phải là tập hợp một cách có hệ thống các Router kết nối với nhau, có thể triển khai một hoặc nhiều giao thức định tuyến, nhưng phải dựa trên 1 chính sách quản trị nhất quán, hay chịu sự quản lý độc lập của 1 đơn vị chủ thể. Một AS được đại diện bằng số hiệu mạng – ASN (Autonomous System Number)

ASN được quản lý và cấp phát bởi Tổ chức cấp phát số hiệu Internet (IANA-Internet Assigned Numbers Authority). Các số ASN được cấp phát cho các Tổ chức đǎng ký vùng (regional registry) và các tổ chức này sẽ chịu trách nhiệm phân bổ lại cho các đơn vị sử dụng khác. Hiện tại có 4 Tổ chức đǎng ký vùng là : APNIC (Asia Pacific Network Information Center) chịu trách nhiệm đối với khu vực Châu á Thái Bình Dương, ARIN (American Registry for Internet Numbers) chịu trách nhiệm đối với khu vực gồm Bắc Mỹ, Nam Mỹ, Biển Caribe, RIPE NCC (Réseaux IP Européens Network Coordination Centre) chịu trách nhiệm đối với khu vực Châu Âu và AfriNic là một tổ chức đăng ký vùng mới thành lập chịu trách nhiệm cho khu vực châu Phi. Ở Việt Nam, việc cấp phát và quản lý số hiệu mạng được thực hiện bởi Trung tâm Internet Việt nam (VNNIC). Các số hiệu mạng do VNNIC cấp sẽ thuộc vào nhóm các số hiệu mạng do APNIC quản lý.

Theo đó, dựa vào miền quản trị, ta có 2 nhóm giao thức định tuyến:

* **IGP (Interior Gateway Protocol) -** Giao thức định tuyến nội : được sử dụng để định tuyến ở bên trong các miền AS. Về mặt quản trị các mạng được định tuyến bởi IGP thường nằm trong sự điều khiển của 1 tổ chức như doanh nghiệp, khách sạn, bệnh viện,… Các giao thức IGP có thể kể tới:
  + RIP
  + IGRP và EIGRP
  + OSPF
  + IS-IS
* **EGP (Exterior Gateway Protocol) -** Giao thức định tuyến ngoại : được sử dụng để định tuyến giữa các miền AS với nhau (liên AS), giao thức định tuyến ngoại duy nhất được sử dụng hiện nay là BGP. Do đặc điểm định tuyến liên AS, nên BGP thường được triển khai giữa ISP và khách hàng và giữa các ISP với nhau. BGP là giao thức định tuyến duy trì Internet hiện nay



*Hình 3.6 – Vị trí sử dụng IGP và EGP*

Xét theo miền quản lý, ta có IGP và EGP. Tuy nhiên do đặc thù vị trí sử dụng của mình nên IGP và EGP phản ánh các đặc điểm đặc trưng rất rõ ràng của 2 nhóm giao thức này. Câu hỏi đặt ra là: Liệu IGP có thể sử dụng để định tuyến liên AS và EGP có thể được sử dụng để định tuyến bên trong 1 AS ? Nếu có thể, tại sao lại phân chia thành 2 nhóm như vậy ?

**Với câu hỏi thứ nhất:** Câu trả lời là được, xét theo mục tiêu cơ bản của các giao thức định tuyến động là xác định tuyến đường tối ưu tới 1 mạng đích và chuyển tiếp gói. Thực tế là chúng ta có thể hoán đổi vị trí của các giao thức IGP và EGP. Vậy điều gì dẫn đến việc phát triển 2 nhóm giao thức có thể thực hiện cùng tác vụ ? Ta xem xét việc trả lời câu hỏi thứ 2

Vấn đề nằm ở quy mô và các yêu cầu chính sách định tuyến. EGP được triển khai liên AS, phủ khắp Internet, phạm vi toàn cầu, đồng nghĩa với số lượng tuyến đường cực kỳ lớn (tới cuối năm 2011 là hơn 400,000 tuyến đường, con số này tăng nhanh từ 50,000 tuyến đường vào năm 1999) và sẽ còn tăng nhanh nữa. Yêu cầu đối với EGP không chỉ nằm ở quản lý lượng tuyến đường lớn mà còn có thể triển khai các chính sách phức tạp để tối ưu điều khiển tuyến đường. Với EGP, độ ổn định quan trọng hơn tốc độ hội tụ. Ngược lại với IGP, một giao thức định tuyến triển khai trong 1 tổ chức, số lượng tuyến đường quản lý không quá lớn, do đó sẽ tập trung tối ưu tốc độ hội tụ và khả năng phản ứng với thay đổi topology. Theo đó, một giao thức IGP không phù hợp để triển khai liên AS và ngược lại.

### Phân loại dựa trên cách thức cập nhật thông tin định tuyến – Distance-vector vs Link-State

Các giao thức định tuyến nội IGP được chia thành 2 nhóm dựa trên cách thức cập nhật thông tin định tuyến là Distance-vector (Vector khoảng cách) Link-state (Trạng thái kết nối)

1. **Distance-vector**

Khái niệm vector trong toán học là một phần tử được xác định bởi 3 yếu tố: điểm đầu, hướng và độ lớn. Một giao thức distance-vector sẽ xem mạng đích là điểm đầu, giao diện tới Router quảng bá thông tin định tuyến là hướng, và metric (thước đô tuyến đường) là độ lớn. Như vậy, 1 giao thức định tuyến distance-vector sẽ chỉ quan tâm đến 2 yếu tố: Mạng đích và Router hàng xóm quảng bá tuyến đường. Với mỗi giao thức khác nhau sẽ dẫn đến thuật toán sử dụng, và cách tính metric khác nhau. Chẳng hạn RIP sử dụng metric là hop-count (số lượng Router dọc tuyến), trong khi IGRP xem xét đồng thời các yếu tố: băng thông, độ trễ, tải, và độ tin cậy của tuyến đường để tính metric.

Kết quả cuối cùng, 1 Router chạy Distance-vector sẽ có bảng cơ sở dữ liệu là 1 tập hợp các mạng đích, và Router quảng bá tuyến đường tới các mạng đích đó, kết hợp với metric. Thông tin tuyến đường được đưa trực tiếp vào bảng định tuyến

Các giao thức đại diện của Distance-vector bao gồm: RIPv1, RIPv2, IGRP và EIGRP

1. **Link-state :**

Khác với Distance-vector, giao thức Link-state xem xét 1 mạng đích được đại diển bởi 1 Interface của Router (link), vì thế thông tin mà các Router chạy giao thức Link-state trao đổi với nhau không phải là thông tin tuyến đường để tới một mạng đích, mà chính là thông tin trạng thái về link kết nối trực tiếp trên Router đó. Nếu 1 Router có 5 interface chạy giao thức link-state, thì Router sẽ chỉ quảng bá duy nhất trạng thái của 5 interface đó cho các Router hàng xóm. Dựa vào đó, và tùy thuộc vào thuật toán của mỗi giao thức link-state, Router sẽ xây dựng lại toàn bộ topology bao gồm các node có trong mạng và các link gắn với mỗi node. Như vậy, giao thức link-state quan tâm đến 3 yếu tố: node trong mạng, link kết nối trên node đó và trạng thái của link đó. Cũng như Distance-vector, giao thức Link-state cũng sử dụng metric riêng để đo khoảng cách tuyến đường tới các link.

Kết quả cuối cùng, mỗi Router chạy Link-State trong cùng 1 mạng sẽ có 1 bảng cơ sở dữ liệu giống nhau, là một tập hợp tất cả các node chạy cùng giao thức, link kết nối trên mỗi node và trạng thái của các link đó. Thông tin tuyến đường sẽ được trích xuất từ trạng thái của các link và kết nối giữa các node để đưa vào bảng định tuyến

Các giao thức đại diện của Link-State bao gồm: OSPF, và IS-IS

1. **So sánh giữa Distance-vector và Link-State**

Giao thức định tuyến distance-vector chỉ trao đổi thông tin tuyến đường, nên thông tin định tuyến về cơ bản chính là bảng định tuyến trên mỗi Router. Giao thức Link-state sẽ tự xây dựng tuyến đường dựa trên bảng cơ sở dữ liệu, vì thế chúng chỉ trao đổi thông tin trạng thái của link, chính là các “mảnh” dữ liệu sau đó được dùng để xây dụng bảng cơ sở dữ liệu. Do các đặc điểm đó, Distance-vector thường hội tụ nhanh hơn các giao thức Link-state. **Đó là khác biệt về thông tin cập nhật**

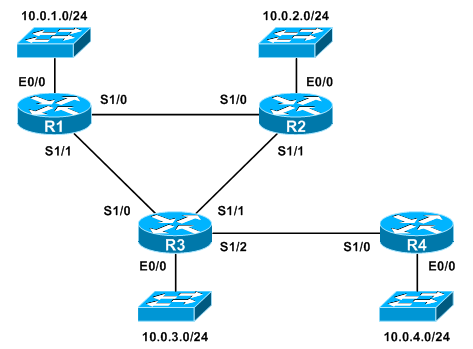
Do đặc điểm đó, 1 Router chạy distance-vector không nắm được tại 1 thời điểm, tuyến đường tới 1 mạng đích có thực sự tồn tại hay không, vì vậy Distance-vector sẽ định kỳ gửi thông tin update cho các Router hàng xóm để duy trì định tuyến. Ngược lại giao thức Link-state luôn xác định được trạng thái hiện tại của link kết nối trực tiếp, do đó thông tin update sẽ chỉ gửi khi có sự thay đổi về trạng thái. Do đó, các giao thức Link-state phản ứng với thay đổi hiệu quả hơn so với Distance-vector. **Đó là khác biệt về cách thức cập nhật thông tin**

Giao thức Distance-vector sẽ nhìn thấy mạng hoàn toàn phẳng với tập hợp các tuyến đường tới mạng đích, trong khi giao thức Link-state có thể xây dựng lại topology định tuyến của mạng, hay có thể xác định cấu trúc lớp của mạng đó, cấu trúc lớp xác định vai trò và vị trí các node trong mạng. Do đó, Distance-vector phù hợp trong các mạng có quy mô nhỏ và vừa, còn Link-State hướng đến các mạng yêu cầu cao về khả năng mở rộng. **Đó là khác biệt về cách tiếp cận thiết kế định tuyến trong mạng.**

Giao thức Distance-vector trao đổi thông tin tuyến đường, phần lớn trường hợp Router nhận update không có khái niệm về các mạng được quảng bá, hay nói cách khác Router đó phải “gửi trọn niềm tin” vào Router đã gửi quảng bá, do đó các vấn đề về loop có thể xảy ra mà Router không biết được, theo đó mỗi giao thức Distance-vector luôn được phát triển với kỹ thuật chống loop riêng biệt. Ngược lại giao thức Link-state xây dựng cơ sở dữ liệu dựa trên các phần thông tin định tuyến mà chúng tin cậy, do đó với giao thức Link-state, không bao giờ xảy ra loop, bản thân cách thức update đã là 1 kỹ thuật chống loop. Do đó, giao thức Link-state thường nhìn nhận là tin cậy với các mạng phức tạp hơn so với giao thức Distance-vector. **Đó là khác biệt về khả năng chống loop định tuyến trong mạng.**

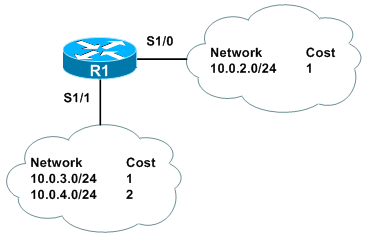
4 khác biệt trên là 4 khác biệt cơ bản giữa họ giao thức Distance-vector và họ giao thức Link-State.

Hãy xem cách thức 2 nhóm giao thức này làm việc, cho topology như sau:

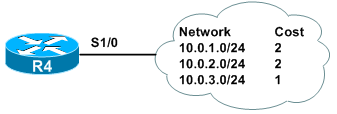


Chúng ta xem xét sự khác nhau trong việc xây dựng thông tin định tuyến và tuyến đường giữa Link-State và Distance-Vector thông qua theo dõi trên 2 Router R1 và R4.

**Với Distance Vector**



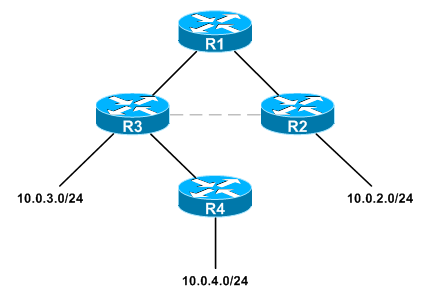
*Hình 3.7 – Cách thức Router R1 cập nhật tuyến đường*



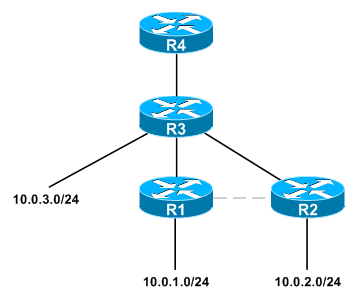
*Hình 3.8 – Cách thức Router R4 cập nhật tuyến đường*

Giao thức Distance-Vector hoàn toàn không biết gì về topology của hệ thống, tất cả chỉ là thông tin tới mạng đích

**Với Link-State**



*Hình 3.9 – Cách thức Router R1 sử dụng SPF xác định tuyến đường*



*Hình 3.10 – Topology trong mắt Router R4*

### Phân loại dựa trên thông tin cập nhật mạng đích – Classful vs Classless

Chúng ta nhớ lại về địa chỉ IP trong tầng Network và khái niệm lớp địa chỉ A, B, và C. Với Classful Addressing, 1 địa chỉ IP sẽ có tiền tố mạng tương ứng với lớp địa chỉ, chẳng hạn 1 địa chỉ lớp A như 10.0.0.1 sẽ có tiền tố mạng là /8, tương tự, 1 địa chỉ lớp B sẽ có tiền tố mạng /16 và 1 địa chỉ lớp C sẽ có tiền tố mạng /24. Như vậy, với classful addressing, chúng ta chỉ có 3 tiền tố mạng cho 3 lớp địa chỉ. Tuy nhiên cách đánh địa chỉ dựa vào Classful Addressing là không tối ưu, lãng phí và cứng nhắc, nên IETF đưa ra khái niệm Classless Addressing. Classless addressing là chia địa chỉ không theo lớp, hay tiền tố mạng có thể bằng bất kỳ giá trị nào từ 1 tới 32 thay vì chỉ có thể bằng 8, 16 hoặc 24.

Trong định tuyến, thông tin mạng đích sẽ được mang theo trong các bản tin cập nhật, vì thế tương ứng với mỗi cách đánh địa chỉ sẽ có cách thức định tuyến khác nhau. Tương ứng với Classful và Classless Addressing, chúng ta có Classful và Classless Routing

* 1. **Classful**

Về mặt kỹ thuật, các giao thức định tuyến Classful sẽ không gửi mặt nạ mạng kèm với địa chỉ mạng đích trong bản tin update. Do đặc điểm lịch sử, các giao thức Classful là các giao thức thời kỳ đầu, khi các mạng được quy hoạch dựa vào lớp A, B hoặc C. Khi đó mặt nạ mạng là không cần thiết mang theo trong bản tin update vì Router có thể xác định dựa vào octet đầu tiên của địa chỉ mạng.

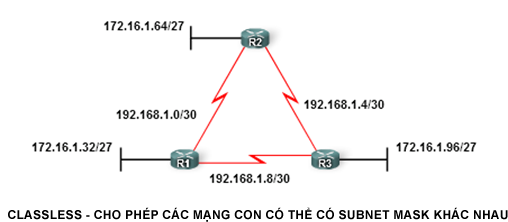
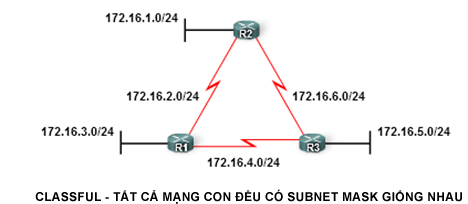
Đại diện của giao thức định Classful bao gồm RIPv1 và IGRP

* 1. **Classless**

Ngược lại với các giao thức classful, các giao thức Classless hỗ trợ gửi mặt nạ mạng kèm theo địa chỉ mạng. Các giao thức định tuyến Classless được thiết kế để hỗ trợ VLSM, do đó tất cả các giao thức định tuyến được sử dụng phổ biến hiện nay đều là giao thức Classless.

Các giao thức tiêu biểu thuộc nhóm này là RIPv2, EIGRP, OSPF, IS-IS, BGP

Chúng ta thấy, một mạng chia địa chỉ theo Classful Addressing có thể triển khai định tuyến theo Classful hoặc Classless, nhưng ngược lại thì không ổn, vì tất cả thông tin mạng đích sẽ bị thay đổi dựa vào lớp địa chỉ. Thông tin về mạng 172.168.1.0/24 sẽ trở thành 172.16.0.0/16 là một ví dụ minh họa. Các vấn đề cụ thể sẽ được giới thiệu trong các chương sau, nhưng chúng ta thấy, việc sử dụng 1 giao thức classful hay classless sẽ ảnh hưởng tới thiết kế và quy hoạch địa chỉ trong 1 mạng



*Hình 3.11 – Loại giao thức định tuyến quyết định cách chia địa chỉ*

### Hội tụ - Convergence

Hội tụ là một trạng thái của mạng định tuyến, trong đó quá trình cập nhật thông tin định tuyến đã hoàn thành trên các Router và quá trình định tuyến trong mạng đã ổn định. Như vậy, chúng ta nói 1 mạng đã hội tụ, cần hội đủ 2 yếu tố:

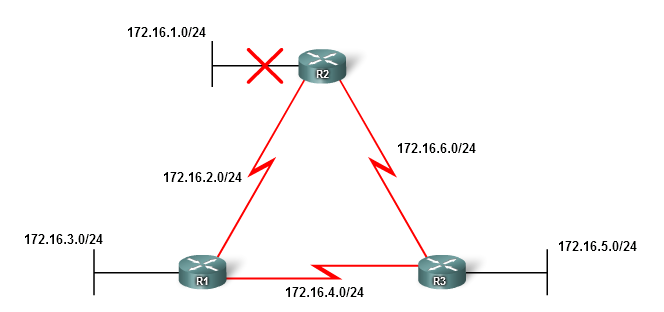
* Tất cả Router cập nhật đầy đủ thông tin định tuyến
* Tất cả yêu cầu kết nối đều thành công

Các Router chạy cùng giao thức định tuyến mới trao đổi thông tin định tuyến với nhau. Tuy nhiên trong phần lớn trường hợp mạng triển khai đồng thời nhiều giao thức định tuyến cùng 1 lúc, khi đó kể cả trong trường hợp mọi thông tin định tuyến đã trao đổi đầy đủ, và mạng tưởng như ổn định, vấn đề loop vẫn có thể xảy ra. Hoặc trường hợp xấu hơn là định tuyến thiếu ổn định, tại các thời điểm khác nhau Router lựa chọn các tuyến đường khác nhau để tới cùng mạng đích. Những trường hợp như vậy được xem là hội tụ chưa thành công

Trong trường hợp mạng chỉ sử dụng 1 giao thức định tuyến, tốc độ hội tụ của mạng cũng chính là tốc độ hội tụ của giao thức định tuyến. Tốc độ hội tụ là một yếu tố quan trọng để đánh giá hiệu quả của 1 giao thức định tuyến. Tốc độ hội tụ không chỉ được đánh giá dựa vào lần hội tụ đầu tiên, mà còn được đánh giá dựa trên tốc độ phản ứng của Router khi lựa chọn tuyến đường đối với sự thay đổi trạng thái của mạng đích. Theo đó, để đánh giá tốc độ hội tụ của 1 giao thức định tuyến, chúng ta xem thời điểm đầu tiên và thời điểm mạng xảy ra sự cố tại các điểm khác nhau.

Dựa trên 2 yếu tố đánh giá hội tụ, chúng ta thấy có 2 thuộc tính ảnh hưởng tới tốc độ hội tụ, đó là thời gian trao đổi thông tin định tuyến, và cách thức tính toán đường đi tối ưu của giao thức định tuyến.

Cho ví dụ topology:



Trong ví dụ trên, Router R2 có mạng kết nối trực tiếp 172.16.1.0/24. Sau khi mạng hội tụ lần đầu tiên, tất cả Router đều có tuyến đường tới mạng 172.16.1.0/24. Hãy xét trường hợp nếu mạng 172.16.1.0/24 DOWN, vì 1 lý do nào đó. Và do Router R2 là Router duy nhất biết trạng thái về mạng 172.16.1.0/24 nên Router R2 sẽ là thiết bị cập nhật lại thông tin định tuyến:

* Nếu giao thức định tuyến được sử dụng là Distance-vector, R2 sẽ không gửi thông tin cập nhật luôn, mà sẽ chờ đến quá trình cập nhật định kỳ với chu kỳ 30 giây, như vậy mạng sẽ chỉ hội tụ lại sau hơn 30 giây
* Nếu giao thức định tuyến là Link-State, R2 sẽ gửi cập nhật kích hoạt ngay khi phát hiện mạng mất kết nối. Do đó mạng sẽ hội tụ lại rất nhanh, trong khoảng vài giây.

Sự khác nhau giữa tốc độ hội tụ có thể khó đánh giá hơn nhiều nếu trong mạng triển khai nhiều giao thức định tuyến, hoặc mức độ phức tạp của thiết kế định tuyến cao hơn (chẳng hạn kiến trúc OSPF đa vùng so với đơn vùng) hoặc kết hợp các yếu tố ngoài định tuyến. Tuy nhiên, xét 1 cách tổng thể, giao thức Link-State hội tụ hiệu quả hơn so với giao thức Distance-Vector.

Ngoài ra, một số giao thức có quan điểm khác nhau đối với vấn đề hiệu quả của hội tụ, chẳng hạn giao thức định tuyến ngoại như BGP sẽ đánh giá cao độ ổn định của tuyến đường hơn so với tốc độ hội tụ, ngược lại, các giao thức định tuyến nội luôn quan tâm tới tốc độ hội tụ.

## Lựa chọn tuyến đường tối ưu

Như đã đề cập ở trên, mọi giao thức định tuyến động sau khi trao đổi thông tin định tuyến sẽ đều tập hợp vào 1 bảng cơ sở dữ liệu mang thông tin mạng đích và tất cả tuyến đường có thể sử dụng để tới mạng đích. 1 Router có thể có nhiều tuyến đường để tới 1 mạng đích, bài toán đưa ra là Router cần phải lựa chọn tuyến đường tối ưu nhất. Về cơ bản, Router sẽ xem xét 3 yếu tố để lựa chọn tuyến đường tối ưu theo thứ tự như sau:

* Chiều dài khớp mạng đích là lớn nhất - Chúng ta sẽ đề cập tới ưu tiên này trong chương 8 - bài về bảng định tuyến
* AD (Administrator Distance) – chỉ số tin cậy
* Metric – thước đo tuyến đường

### Chỉ số ưu tiên AD – Administrator Distance

1 Router có thể học tuyến đường tới 1 mạng đích (phía xa, khác kết nối trực tiếp) thông qua:

* Định tuyến tĩnh
* Định tuyến động với rất nhiều giải pháp với các giao thức định tuyến khác nhau

Việc triển khai định tuyến trong mạng không đơn giản, trong nhiều trường hợp buộc phải sử dụng nhiều hơn 1 giao thức định tuyến, khi đó hoàn toàn có thể xảy ra việc 1 mạng đích được học từ nhiều giao thức khác nhau, cả tĩnh và động, và cả giữa các giao thức định tuyến động. Những tuyến đường khác nhau này có chỉ số thước đo (metric) khác nhau, vậy làm sao để Router xác định được tuyến đường nào là tốt nhất, tuyến đường nào là tối ưu hơn ?

Để xem xét các tuyến đường tới cùng 1 mạng đích học từ nhiều nguồn khác nhau, chúng ta xem xét thông tin nào “đáng tin cậy” hơn, giao thức nào cần được ưu tiên hơn ? Khái niệm “tin cậy” trong trường hợp này chỉ mang tính tương đối, tuy nhiên nó cũng phản ánh phần nào khả năng của các giao thức định tuyến. Trong trường hợp này, độ ưu tiên được đo bằng 1 chỉ số được gọi là *AD – Administrator-Distance*

AD phản ánh độ ưu tiên của Router đối với 1 nguồn định tuyến, 1 nguồn định tuyến, như đã đề cập, có thể là 1 tuyến tĩnh, thậm chí 1 mạng kết nói trực tiếp, hoặc có thể là 1 tuyến động, được học qua bất kỳ 1 giao thức định tuyến động nào. Chỉ số AD càng nhỏ càng có mức độ ưu tiên cao hơn, chỉ số AD là các số nguyên từ 0 tới 255. Chúng ta có bảng giá trị AD như sau:

|  |  |
| --- | --- |
| Nguồn định tuyến | Giá trị AD |
| Kết nối trực tiếp | 0 |
| Tuyến tĩnh | 1 |
| EIGRP gộp tuyến | 5 |
| BGP ngoại tuyến | 20 |
| EIGRP nội tuyến | 90 |
| IGRP | 100 |
| OSPF | 110 |
| IS-IS | 115 |
| RIP | 120 |
| EIGRP ngoại tuyến | 170 |
| BGP nội tuyến | 200 |
| Không hợp lệ | 255 |

Ví dụ bảng định tuyến của 1 Router triển khai đồng thời nhiều giao thức định tuyến:

Sydney#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is 192.168.1.5 to network 0.0.0.0

10.0.0.0/16 is subnetted, 1 subnets

D 10.10.0.0 [90/2172416] via 192.168.1.5, 05:00:21, Serial0/0

172.16.0.0/24 is subnetted, 2 subnets

C 172.16.4.0 is directly connected, FastEthernet0/0

O 172.16.15.0 [110/65] via 192.168.2.2, 00:01:21, Serial0/1

192.168.1.0/30 is subnetted, 1 subnets

C 192.168.1.4 is directly connected, Serial0/0

192.168.2.0/30 is subnetted, 1 subnets

C 192.168.2.0 is directly connected, Serial0/1

S\* 0.0.0.0/0 [1/0] via 192.168.1.5

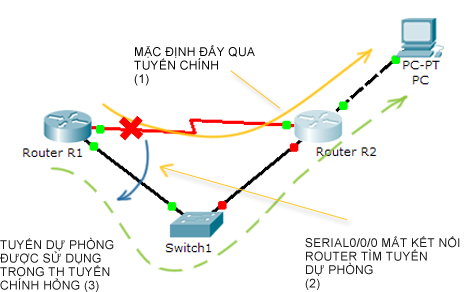
**Tuyến kết nôi trực tiếp (AD = 0) –** Tuyến kết nối trực tiếp có AD bằng, và luôn luôn bằng 0, AD bằng 0 là giá trị AD tốt nhất, và được ưu tiên hơn so với bất kỳ tuyến đường nào khác. Điều này hợp lý, vì không ai biết rõ hơn bản thân Router về các mạng kết nối trực tiếp của nó

**Tuyến tĩnh (AD = 1)** – **Ứng dụng backup route -** Qua các bảng định tuyến trong chương trước, chúng ta đã thấy giá trị AD của tuyến tĩnh bằng 1. ĐIều này phản ánh thực tế rằng tuyến tĩnh là tuyến được cấu hình trực tiếp bởi người quản trị, do đó hiển nhiên (cho dù đúng hay sai) thì tuyến đường phải được ưu tiên hơn các tuyến khác. Tuy nhiên, đôi lúc không phải, vì tuyến tĩnh không thể chủ động cập nhật thông tin, thêm vào đó, giả sử người quản trị muốn triển khai 1 tuyến dự phòng bằng định tuyến tĩnh, thì làm sao có thể làm được nếu AD của tuyến tĩnh *luôn* cao hơn những tuyến khác.

Chúng ta nhớ lại cú pháp cấu hình tuyến tĩnh:

Router(config)#ip route {địa chỉ IP mạng đích} {subnet mask} {địa chỉ IP next-hop | outgoing interface} {AD}

Trong đó trường cuối cùng, trường {AD}, được sử dụng để điều chỉnh AD cho 1 tuyến tĩnh. Các tuyến tĩnh khi được tạo ra đều có giá trị AD mặc định bằng 1, giả sử nếu có 2 tuyến tĩnh tới cùng 1 mạng, 1 tuyến là chính, 1 tuyến là dự phòng, vậy làm thế nào ?



*Hình 3.12– Định tuyến tĩnh điều khiển AD để làm tuyến dự phòng (backup route)*

Ứng dụng triển khai tuyến dự phòng bằng cách thay đổi thông số AD là 1 kỹ thuật đơn giản nhưng hiệu quả, trong ví dụ trên, chẳng hạn Router R1 cần 2 tuyến mặc định qua Router R2:

R1(config)#ip route 0.0.0.0 0.0.0.0 serial0/0/0

R1(config)#ip route 0.0.0.0 0.0.0.0 fastEthernet0/0 20

Như vậy R1 sẽ đánh giá tuyến đường qua Serial0/0/0 tối ưu hơn và sẽ sử dụng tuyến đường này trong bảng định tuyến, trong trường hợp nếu interface serial0/0/0 thay đổi trạng thái DOWN, tra cứu đệ quy thất bại, tuyến đường mặc định chính sẽ bị loại khỏi định tuyến. Lúc đó tuyến đường thứ 2 qua fastEthernet sẽ được sử dụng để làm tuyến đường mặc định. Xem bảng định tuyến của Router R1 trước và sau khi thay đổi tuyến đường:

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

S\* 0.0.0.0/0 [1/0] via 12.12.12.2

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

S\* 0.0.0.0/0 [20/0] via 21.21.21.2

**1 giao thức nhiều giá trị AD –** AD là 1 thông số có thể triển khai rất linh hoạt để điều khiển quá trình chọn tuyến đường. Một vài giao thức bản thân trong cách thiết kế sử dụng nhiều hơn 1 giá trị AD để đánh giá các tuyến đường của mình, chẳng hạn giao thức EIGRP coi các tuyến đường học từ EIGRP là các tuyến nội và có AD bằng 90, trong khi các tuyến đường trong EIGRP nhưng học từ 1 giao thức khác, là các tuyến ngoại và có AD bằng 170. Việc sử dụng nhiều giá trị AD có ý nghĩa quan trọng về mặt thiết kế giao thức định tuyến đó, bản thân cũng là 1 cách chống loop.

**Điều chỉnh AD –** AD không phải một bảng giá trị tham chiếu, mà là 1 cơ chế đánh giá tuyến đường, do đó, AD có thể điều chỉnh trong mọi giao thức định tuyến. Chẳng hạn mặc định OSPF (AD = 110) không được ưu tiên bằng EIGRP nội tuyến (AD = 90), tuy nhiên ta có thể điều chỉnh AD của OSPF bằng 80 để lật ngược quyết định. Các giao thức hiện tại còn mạnh hơn thế, cho phép người quản trị thay đổi AD cho 1 nhóm các tuyến đường, nhóm tuyến đường này do người quản trị thoải mái lựa chọn, có thể là 1 dải địa chỉ IP, hay cũng có thể là các loại tuyến khác nhau giống như EIGRP với tuyến nội và tuyến ngoại. Chi tiết hơn sẽ được đề cập trong giáo trình khác

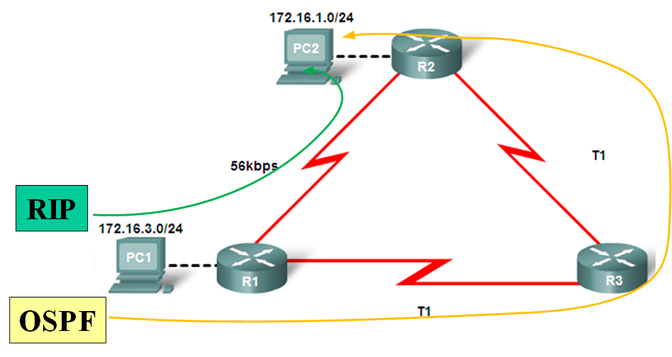
**Tuyến không hợp lệ, AD 255 –**1 tuyến đường với AD bằng 255 sẽ được coi là không hợp lệ và không được đưa vào bảng định tuyến. Khi nào 1 tuyến đường nhận được lại có AD bằng 255, khi người quản trị đặt giá trị AD cho tuyến đường đó bằng 255. Ý nghĩa duy nhất của việc sử dụng AD 255 đó là 1 kỹ thuật lọc tuyến đường.

### Metric – Thước đo tuyến đường

Thước đo tuyến đường được sư dụng để đánh giá độ tối ưu của các tuyến đường khác nhau cua cùng 1 giao thức định tuyến tới cùng 1 mạng đich. Giữa 2 tuyến đường được học bởi cùng 1 giao thức, tuyến đường nào có metric tốt hơn thì sẽ được lựa chọn là tuyến đường tối ưu. Các giao thức định tuyến khác nhau có các tiêu chí lựa chọn tuyến đường tốt nhất khác nhau. Chúng ta xem xét metric chỉ là cách thể hiện hóa của 1 đặc điểm nào đó của tuyến đường. Các giao thức khác nhau xem xét các yếu tố ưu tiên khác nhau nên có trị số khác nhau như sau:

* **RIP :** Sử dụng hop-count. Hop count là số lượng Router gói tin sẽ đi qua để tới đích. Như vậy 1 cách định tính, tuyến đường tốt nhất đối với RIP là tuyến đường mà trên đó gói tin đi qua ít Router nhất. Cách tiếp cận của RIP rất dễ hiểu
* **IGRP và EIGRP:** 2 giao thức được phát triển bởi Cisco này lại tính toán metric dựa trên rất nhiều thông số, bao gồm: băng thông nhỏ nhất dọc tuyến đường, tổng độ trễ trên từng liên kết dọc tuyến đường, độ tải trên interface của các Router, độ tin cậy của interfafce. Với IGRP hay EIGRP 1 tuyến đường “dài” hay “ngắn” không quan trọng bằng tốc độ và trạng thái đường truyền thật sự, do đó ta cũng có thể thấy IGRP và EIGRP thông minh hơn RIP
* **OSPF :** OSPF sử dụng 1 thông số gọi là cost để làm metric. Cost được tính tỷ lệ nghich với băng thông trên mỗi link giứa các Router. Và metric của tuyến đường là tổng cost dọc tuyến đường. Như vậy OSPF đánh giá 1 tuyến đường là tốt hay xấu dựa trên khả năng truyền dữ liệu của đường truyền đó

Chúng ta mô tả sự khác nhau về mặt ý nghĩa của metric với ví dụ dưới đây:



*Hình 3.13 – Các giao thức khác nhau dùng metric khác nhau,   
đánh giá tuyến đường tối ưu khác nhau*

Ta thấy để mạng LAN của Router R1 có thể kết nối tới mạng LAN của Router R2, trên Router R1 có 2 lựa chọn là 2 tuyến đường: 1 tuyến đi trực tiếp qua R2 (tuyến thứ nhất), 1 tuyến là đi qua R3 rồi từ R3 chuyển gói tin sang R2 (tuyến thứ hai). Về mặt định tính, chúng ta thấy tuyến thứ nhất tối ưu hơn vì gói tin đi trực tiếp sang R2.

* **Trường hợp thứ nhất – Triển khai RIP:** Chúng ta thấy trong giới thiệu ở trên, RIP sử dụng hop-count là metric, do đó RIP nhìn thấy tuyến đường thứ nhất có metric bằng 2, còn tuyến thứ 2 có metric bằng 3. Vậy RIP sẽ lựa chọn tuyến đường đi trực tiếp qua Router R2
* **Trường hớp thứ hai – Triển khai OSPF:** Chúng ta thấy OSPF đánh giá tuyến đường dựa trên băng thông. Tuyến thứ nhất băng thông 56kbps, tuyến thứ 2 có 1544/2 Mbps = 772 Mbps = 772,000 kbps (Tuyến T1 có băng thông 1544 Mbps nhưng phải phải đi qua 2 link nên có thể xem như băng thông dọc tuyến chỉ còn 1 nửa) Vậy OSPF thấy rằng tuyến thứ 2 có thể dài hơn, nhưng đẩy gói tin đi nhanh hơn tuyến đường thứ nhất

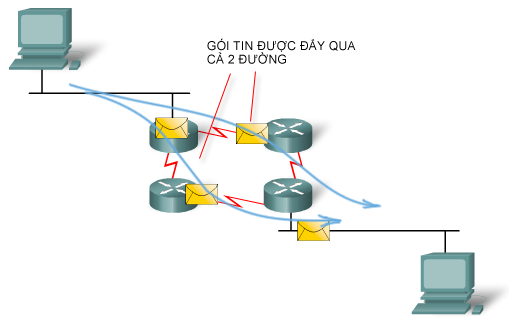
Do các đặc điểm lựa chọn tuyến đường khác nhau như vậy, việc lựa chọn giao thức định tuyến trở thành 1 bài toán không dễ, vì thế nào là tối ưu, thế nào là hiệu quả, là phải do người quản trị đánh giá và quyết định sẽ sử dụng giao thức nào.

Đối với định tuyến tĩnh, chúng ta nhớ trong ví dụ trước, metric luôn bằng 0, tương tự với tuyến tới mạng kết nối trực tiếp. Hay nói cách khác, metric là “đặc sản” của định tuyến động

### Cân bằng tải (Load-Balancing)

Cân bằng tải là 1 kỹ thuật chia tải trên nhiều đường khá nhau để tối ưu tài nguyên mạng trong việc điều khiển lưu lượng.

*Router sẽ hành xử thế nào khi có 2 tuyến đường cùng metric ?*Khi Router có 2 tuyến đường tới cùng 1 mạng đích và có metric bằng nhau, Router sẽ không thể xác định được tuyến đường nào là tối ưu hơn, do đó, để an toàn, Router triển khai định tuyến trên đồng thời cả 2 tuyến đường. Kỹ thuật điều khiển mà Router sử dụng nhiều hơn 1 tuyến đường để gửi gói tin tới 1 mạng đích được gọi là cân bằng tải (hay Load-balancing)



*Hình 3.14 – cân bằng tải diễn ra khi Router xác định nhiều hơn 1 tuyến đường “tốt nhất”*

Cân bằng tải chỉ diễn ra trong phạm vi 1 giao thức định tuyến, vì 2 tuyến đường chỉ có thể so sánh nếu cùng thước đo, hay cùng dùng 1 kiểu metric. Nếu 2 tuyến đường tới cùng mạng đích được học qua 2 giao thức khác nhau, thì 1 thông số khác sẽ được dùng để so sánh, đó là AD – administrative-distance

Định tuyến tĩnh không sử dụng metric để “đo” tuyến đường, do đó chỉ cần có 2 tuyến đường tĩnh tới cùng 1 mạng đích thì chắc chắn cân bằng tải sẽ diễn ra, cho dù 2 tuyến đường đó thực sự có các thông số về hop-count, băng thông,… khác nhau. Xét 1 ví dụ cân bằng tải với định tuyến tĩnh:

RouterX#show ip route

<output omitted>

C 192.168.1.0/24 [1/0] via 192.168.2.1, 00:00:24, Serial0/0/0

[1/0] via 192.168.4.2, 00:00:12, Serial0/0/1

Để xác định độ ưu tiên cho 1 tuyến tĩnh so với các tuyến động, hoặc giữa các tuyến tĩnh khác nhau, người ta chỉ điều khiển AD.

Ngoài định tuyến tĩnh, các giao thức định tuyến động cũng đều hỗ trợ cân bằng tải. Xét ví dụ:

RouterX#show ip route

<output omitted>

R 192.168.1.0/24 [120/4] via 192.168.2.1, 00:00:24, Serial0/0/0

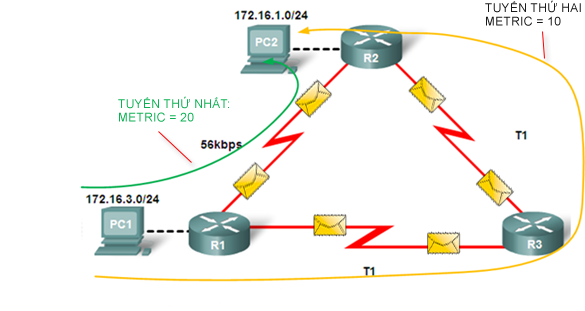
[120/4] via 192.168.4.2, 00:00:12, Serial0/0/1

[120/4] via 192.168.3.2, 00:00:01, fastEthernet0/0

Giao thức được sử dụng là RIP, với AD bằng 120 và metric bằng 4 – Ta thấy cân bằng tải diễn ra đồng thời trên cả 3 interface Serial0/0/0, Serial0/0/1 và fastEthernet 0/0. Các giao thức định tuyến luôn cho phép cân bằng tải diễn ra trên nhiều hơn 2 tuyến đường, tùy thuộc vào phiên bản IOS mà RIP có thể hỗ trợ tối đa 4, 6 hoặc 16 tuyến đường, tương tự với các giao thức định tuyến động khác, việc lựa chọn số lượng tuyến đường tối đa thậm chí có thể điều khiển. Mặc dù quá trình cân bằng tải diễn ra dựa vào việc so sánh metric, nhưng đưa ra quyết định cân bằng tải không phải công việc của giao thức định tuyến. Giao thức định tuyến chỉ chuyển thông tin tuyến đường vào bảng định tuyến, và bảng định tuyến, qua quá trình tra cứu, mới thực hiện cân bằng tải.

**Cân bằng tải (load-balancing) và chia tải (load-sharing) –** Cân bằng tải có thể được triển khai theo 1 trong 2 dạng cân bằng tải cân bằng(equal-load-balancing), hoặc cân bằng tải bất cân bằng (unequal-load-balancing), sự khác nhau giữa 2 kiểu triển khai này nằm ở cách thức chia tải (load-sharing)

* Cân bằng tải cân bằng chỉ diễn ra khi 2 tuyến đường tới cùng 1 mạng đích có cùng metric
* Cân bằng tải bất cân bằng cho phép Router chia tải trên 2 tuyến đường tới cùng 1 mạng đích nhưng khác metric



*Hình 3.15 – Tỷ lệ chia tải giữa 2 tuyến ½ , cân bằng tải bất cân bằng*

Kỹ thuật chia tải cho phép tối ưu hiệu năng mạng khi tận dụng nhiều kết nối để chuyển tiếp gói, tỷ lệ tải sẽ phụ thuộc vào metric của mỗi tuyến đường, chẳng hạn tuyến đường thứ nhất có metric bằng 10, và tuyến thứ 2 có metric bằng 20, vậy tỷ lệ chia tải sẽ là ½, hay cứ 1 gói tin được chuyển qua tuyến đường thứ hai sẽ có 2 gói tin được đẩy qua tuyến thứ nhất. Kỹ thuật chia tải là 1 kỹ thuật nâng cao, mạng càng hội tụ, yêu cầu về tải trên các kết nối càng cần phải được tối ưu. Kỹ thuật cân bằng tải cân bằng có thể xem như chia tải với tỷ lệ 1:1

Cân bằng tải cân bằng:

RouterX#show ip route 1.0.0.0

Routing entry for 1.0.0.0/8

Known via "rip", distance 120, metric 1

Redistributing via rip

Advertised by rip (self originated)

Last update from 192.168.75.7 on Serial1, 00:00:00 ago

Routing Descriptor Blocks:

\* 192.168.57.7, from 192.168.57.7, 00:00:18 ago, via Serial0

Route metric is 1, traffic share count is 1

192.168.75.7, from 192.168.75.7, 00:00:00 ago, via Serial1

Route metric is 1, traffic share count is 1

Cân bằng tải bất cân bằng:

RouterX#show ip route 150.1.2.2

Routing entry for 150.1.2.0/24

Known via "eigrp 100", distance 90, metric 5120, type internal

Redistributing via eigrp 100

Last update from 155.1.13.3 on Serial0/0.13, 00:00:04 ago

Routing Descriptor Blocks:

155.1.13.3, from 155.1.13.3, 00:00:04 ago, via Serial0/0.13

Route metric is 16640, traffic share count is 37

Total delay is 650 microseconds, minimum bandwidth is 128 Kbit

Reliability 255/255, minimum MTU 1500 bytes

Loading 1/255, Hops 2

\* 155.1.12.2, from 155.1.12.2, 00:00:04 ago, via Serial0/0.12

Route metric is 5120, traffic share count is 120

Total delay is 200 microseconds, minimum bandwidth is 1544 Kbit

Reliability 255/255, minimum MTU 1500 bytes

Loading 1/255, Hops 1

Các giao thức hỗ trợ cân bằng tải bất cân bằng là IGRP, EIGRP và BGP. EIGRP chia tải dựa vào metric (metric của EIGRP được xác định dựa vào băng thông, độ trễ, tải và độ tin cậy) BGP chia tải dựa vào băng thông trên link. Một vài giao thức chỉ cho phép duy nhât cân bằng tải cân bằng như OSPF hoặc RIP

## Tổng kết:

Trong chương này chúng ta đã có cái nhìn khái quát về định tuyến động bao gồm:

* Lịch sử phát triển và mục đích của các giao thức định tuyến động
* Phân loại các nhóm giao thức định tuyến động
* Ý nghĩa của hội tụ
* Các thông số lựa chọn tuyến đường bao gồm thước đô tuyến đường và chỉ số tin cậy
* Cân bằng tải cân bằng và bất cân bằng

# ****CHƯƠNG 4 - GIAO THỨC ĐỊNH TUYẾN**** DISTANCE-VECTOR

Như chương trước đã đề cập, xét các IGP có thể chia thành 2 nhóm giao thức: Distance-vector và Link-state. Các đặc điểm cơ bản của 2 nhóm này đã được đề cập trong chương trước. Chương này chúng ta sẽ đi sâu xem xét các đặc điểm của nhóm giao thức thứ nhất – Distance-vector

Các vấn đề liên quan tới giao thức định tuyến Distance-vector trong bài:

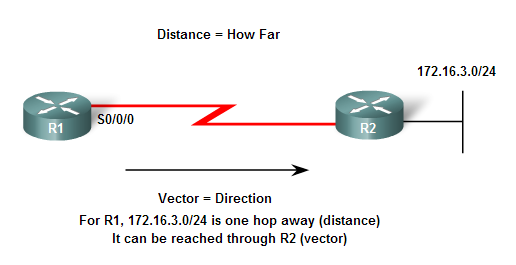
* Các đặc điểm và tính năng
* Cách thức vận hành của giao thức họ Distance-vector bao gồm quá trình cập nhật định tuyến, duy trì bảng định tuyến và triển khai chống loop
* Đánh giá yêu cầu khi thiết kế định tuyến mạng với giao thức Distance-vector

## Giới thiệu giao thức định tuyến Distance-vector

Chúng ta đã đề cập đến ý nghĩa của tên giao thức Distance-vector trong chương trước. Các giao thức distance-vector sẽ không có khái niệm về cấu trúc topology trong mạng, thay vào đó Router chạy Distance-vector sẽ xác định:

* **Hướng –** hay outgoing interface hoặc IP next-hop để đẩy gói tin. IP next-hop xác định địa chỉ IP của Router hàng xóm có kết nối trực tiếp
* **Khoảng cách –** hay còn được gọi là metric, xác định độ dài tuyến đường tới mạng đích

2 thông tin trên là các dữ kiện duy nhất mà Router chạy định tuyến distance-vector biết để xác định tuyến đường tốt nhất



*Hình 4.1 – Giao thức Distance-Vector*

### Các đặc điểm cập nhật (update) của giao thức Distance-vector

Các giao thức Distance-vector, mặc định sẽ có các đặc điểm đặc thù liên quan đến quá trình update thông tin định tuyến cho Router hàng xóm. Các đặc điểm đặc thù này được thiết kế dựa trên cách tiếp cận của giao thức Distance-vector. Chúng ta sẽ đi sâu vào từng đặc điểm và cố gắng phân tích:

**Update định kỳ (Periodic Update)** – Update định kỳ mô tả một Router sẽ chủ động gửi ra thông tin update theo 1 chu kỳ (mặc định là 30 giây với RIP và 90 giây với IGRP). Kể cả trong khi topology không có sự thay đổi nào trong thời gian dài, bản tin update định kỳ vẫn liên tục được gửi cho các Router hàng xóm

**Hàng xóm –** Router hàng xóm là Router có kết nối trực tiếp với Router đang xét và cũng chạy cùng giao thức định tuyến trên liên kết trực tiếp đó. Với mỗi các giao thức Distance-vector, mỗi quan hệ hàng xóm là mối quan hệ duy nhất giữa các Router. Giao thức Distance-vector chỉ nhận thức được sự tồn tại của mạng đích, và tuyến đường tới mạng đích đó qua các Router hàng xóm, chứ không có cái nhìn tổng thể về topology bao gồm cấu trúc tất cả Router chạy cùng giao thức định tuyến trong mạng

**Update quảng bá (Broadcast update) –** Update quảng bá mô tả việc Router chạy Distance-vector gửi bản tin update ra với địa chỉ đích là địch chỉ quảng bá giới hạn 255.255.255.255, các Router chạy cùng giao thức định tuyến sẽ xử lý, còn các thiết bị khác sẽ loại bỏ bản tin update. Tuy nhiên cách tiếp cận đó không thực sự hiệu quả, phần lớn các giao thức định tuyến Distance-vector gửi quảng bá ra địa chỉ Multicast đặc biệt dành riêng cho giao thức đó, chẳng hạn 224.0.0.10 là bản tin multicast của EIGRP, 224.0.0.9 là bản tin multicast của RIP.

**Update toàn bộ bảng định tuyến –** Các giao thức Distance-vector được thiết kế để gửi ra bản tin update với nội dung chính là toàn bộ phần bảng định tuyến chạy giao thức Distance-vector đó. Chẳng hạn 1 Router chạy RIP sẽ gửi update là toàn bộ tuyến đường RIP học được. Router hàng xóm nhận được update, sẽ sử dụng những thông tin cần thiết, những thông tin cập nhật hơn và loại bỏ phần còn lại. Cách tiếp cận này cùng với đặc điểm update định kỳ và update quảng bá chính là một vấn đề của các giao thức Distance-vector thời kỳ đầu, lãng phí lưu lượng không cần thiết.

### Ưu điểm và nhược điểm của giao thức Distance-vector

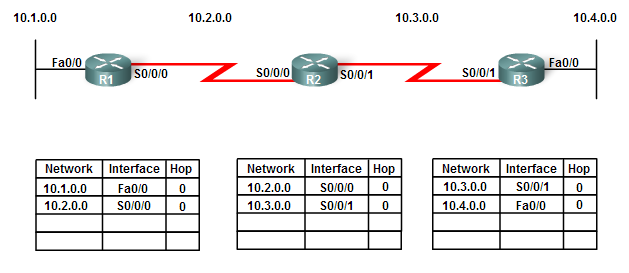
Các ưu điểm và nhược điểm của Distance-Vector có thể mô tả qua bảng sau

|  |  |
| --- | --- |
| Ưu điểm | Nhược điểm |
| Vận hành Đơn giản | Tốc độ hội tụ chậm |
| Tiêu tốn ít tài nguyên Router | Khả năng mở rộng kém |
| Tiêu tốn ít băng thông trên Link | Dễ xảy ra loop |
| Dễ cấu hình |  |

## Học định tuyến

### Cold Start - Khởi động nguội

Khởi động nguội là trạng thái đầu tiên khi Router bắt đầu khởi động, việc đầu tiên là Router xác định các interface nào đang ở trạng thái UP thông qua việc kiểm tra kết nối tại tầng liên kết dữ liệu và tầng vật lý mô hình OSI. Thông tin tầng mạng của các interface này bao gồm địa chỉ mạng kết nối trực tiếp và mặt nạ mạng sau đó được cập nhật vào bảng định tuyến



*Hình 4.2 – Cold-start, tự cập nhật mạng kết nối trực tiếp vào bảng định tuyến*

Trong ví dụ trên, sau khi Cold Start, các Router bắt đầu cập nhật thông tin các mạng kết nối trực tiếp vào bảng định tuyến, bao gồm thông tin địa chỉ mạng, interface và metric bằng 0 để chỉ kết nối trực tiếp

Trên R1

R1#show ip route

<output omitted>

10.0.0/24 is subnetted, 2 subnets

C 10.2.0.0 is directly connected, Serial0/0/0

C 10.1.0.0 is directly connected, FastEthernet0/0

Trên R2

R2#show ip route

<output omitted>

10.0.0/24 is subnetted, 2 subnets

C 10.2.0.0 is directly connected, Serial0/0/0

C 10.3.0.0 is directly connected, Serial0/0/1

Trên R3

R3#show ip route

<output omitted>

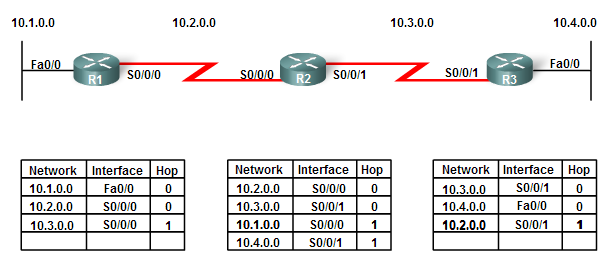
10.0.0/24 is subnetted, 2 subnets

C 10.3.0.0 is directly connected, Serial0/0/1

C 10.4.0.0 is directly connected, FastEthernet0/0

### Bắt đầu trao đổi bản tin định tuyến

Sau khi cập nhật các kết nối trực tiếp vào bảng định tuyến, các Router bắt đầu trao đổi thông tin định tuyến với nhau. Ban đầu, mỗi Router sẽ chỉ gửi ra update về các mạng kết nối trực tiếp trên Router đó, khi nhận được update tuyến đường, Router sẽ kiểm tra, bất kỳ tuyến đường tới mạng đích nào chưa tồn tại trong bảng định tuyến sẽ được thêm vào



*Hình 4.3 – Distance Vector trao đổi thông tin tuyến đường*

Trên R1 : Xây dựng bản tin update cho 2 mạng kết nối trực tiếp với giá trị metric bằng 1 (chúng ta chưa bàn cụ thể đến các giá trị metric trong ví dụ này, metric sẽ được tính toán trên R1 hoặc R2 tùy vào giao thức định tuyến được sử dụng là gì, chúng ta chỉ nhìn nhận metric được sử dụng để đánh giá thước đo tuyến đường), rồi gửi sang cho Router hàng xóm có kết nối trực tiếp là R2, đồng thời nhận update về mạng kết nối trực tiếp của R2 bao gồm mạng 10.2.0.0/24 và mạng 10.3.0.0/24. R1 giữ lại thông tin về mạng 10.3.0.0/24 và xóa bản tin update về mạng 10.2.0.0/24

R1#show ip route

<output omitted>

10.0.0/24 is subnetted, 2 subnets

C 10.2.0.0 is directly connected, Serial0/0/0

C 10.1.0.0 is directly connected, FastEthernet0/0

R 10.3.0.0 [120/1] via 10.2.0.2, 00:00:01, Serial0/0/0

Trên R2 : Tương tự R1, R3 xây dựng bản tin update cho 2 mạng kêt nối trực tiếp với giá trị metric bằng 1, rồi gửi sang cho R1 và R3, R2 nhận update từ R1 và R3, loại bỏ thông tin về tuyến kết nối trực tiếp, lưu 2 tuyến đường tới mạng 10.1.0.0/24 và 10.1.4.0/24 vào bảng định tuyến

R2#show ip route

<output omitted>

10.0.0/24 is subnetted, 2 subnets

C 10.2.0.0 is directly connected, Serial0/0/0

C 10.3.0.0 is directly connected, Serial0/0/1

R 10.1.0.0 [120/1] via 10.2.0.1, 00:00:01, Serial0/0/0

R 10.4.0.0 [120/1] via 10.3.0.3, 00:00:01, Serial0/0/1

Trên R3 : Tương tự R1, R3 nhận update tuyến đường tới mạng 10.2.0.0/24 từ R2

R3#show ip route

<output omitted>

10.0.0/24 is subnetted, 2 subnets

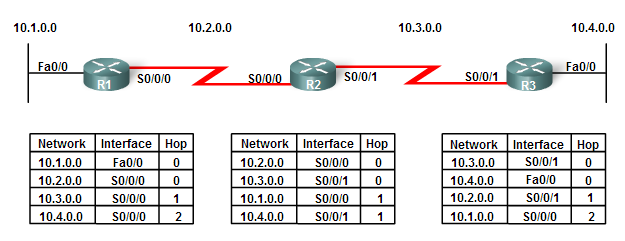
C 10.3.0.0 is directly connected, Serial0/0/1

C 10.4.0.0 is directly connected, FastEthernet0/0

R 10.2.0.0 [120/1] via 10.3.0.2, 00:00:01, Serial0/0/1

### Cập nhật theo tin đồn (Update by rumor)

Sau lần trao đổi bản tin định tuyến đầu tiên, các Router sẽ thực hiện tiến trình cập nhật định kỳ tiếp theo. Bản tin cập nhật mang đầy đủ các tuyến đường được học bởi giao thức định tuyến, trong nhiều trường hợp (trong ví dụ) đó là toàn bộ bảng định tuyến. Các Router nhận được bản tin update, xử lý và cập nhật những tuyến đường mới chưa có hoặc tối ưu hơn để tới 1 mạng đích, các tuyến đường còn lại bị loại bỏ



*Hình 4.4 – Các Router hoàn toàn tin tưởng hàng xóm của mình*

Trên R1 : Lặp lại các bước như trước đó với tất cả tuyến đường trong bảng định tuyến. Tuyến 10.1.0.0/24 được cập nhật metric bằng 1, 10.2.0.0/24 với metric bằng 1, và 10.3.0.0/24 với metric bằng 2, được đưa vào bản tin update gửi sang Router 2. Đồng thời nhận bản tin update từ R2 với thông tin tuyến đường tương ứng cho các mạng 10.1.0.0, 10.2.0.0, 10.3.0.0 và 10.4.0.0, R1 giữ lại mạng 10.4.0.0 (metric 2) cập nhật vào bảng định tuyến, cac tuyến khác đã có sẵn sẽ không được cập nhật

R1#show ip route

<output omitted>

10.0.0/24 is subnetted, 2 subnets

C 10.2.0.0 is directly connected, Serial0/0/0

C 10.1.0.0 is directly connected, FastEthernet0/0

R 10.3.0.0 [120/1] via 10.2.0.2, 00:00:01, Serial0/0/0

R 10.4.0.0 [120/2] via 10.2.0.2, 00:00:01, Serial0/0/0

Trên R2 : R2 nhận update từ R1 cho các tuyến 10.1.0.0, 10.2.0.0 và 10.3.0.0 nhưng không chấp nhận tuyến đường nào vì metric kém tối ưu hơn (mạng 10.3.0.0 trong bảng định tuyến của R2 có metric bằng 0), tương tự với update từ R3. Đồng thời quảng bá các tuyến đường trong bảng định tuyến của mình sang cho R1 và R3. Ta thấy R1 nhận update cho tuyến tới mạng 10.4.0.0 và loại bỏ các tuyến còn lại.

R2#show ip route

<output omitted>

10.0.0/24 is subnetted, 2 subnets

C 10.2.0.0 is directly connected, Serial0/0/0

C 10.3.0.0 is directly connected, Serial0/0/1

R 10.1.0.0 [120/1] via 10.2.0.1, 00:00:01, Serial0/0/0

R 10.4.0.0 [120/1] via 10.3.0.3, 00:00:01, Serial0/0/1

Trên R3 : Tương tự R1, R3 nhận update tuyến đường tới mạng 10.1.0.0/24 từ R2 với metric bằng 2, đồng thời gửi bản tin update sang cho R2, trong đó có tuyến 10.2.0.0 kém tối ưu hơn

R3#show ip route

<output omitted>

10.0.0/24 is subnetted, 2 subnets

C 10.3.0.0 is directly connected, Serial0/0/1

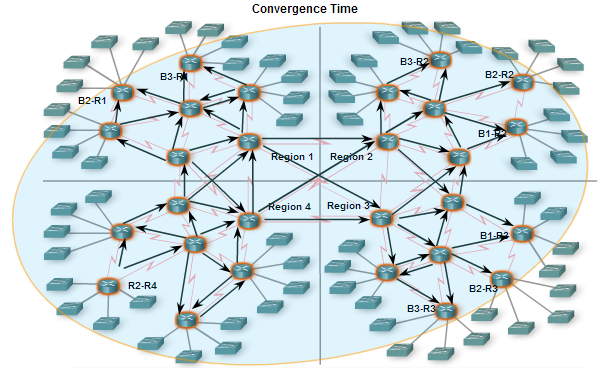
C 10.4.0.0 is directly connected, FastEthernet0/0

R 10.2.0.0 [120/1] via 10.3.0.2, 00:00:01, Serial0/0/1

R 10.1.0.0 [120/2] via 10.3.0.2, 00:00:01, Serial0/0/1

Ta thấy đến bước này, mạng chính thức hội tụ. Mỗi Router đều có tuyến đường tới tất cả mạng đích trong bảng định tuyến. Có thể kiểm tra kết nối bằng công cụ ping. Cơ chế trao đổi định tuyến của giao thức Distance-vector đều có điểm chung như thế, đó là mỗi Router gửi update tuyến đường tới các mạng xa mà Router đó có thể không biết, phương pháp như vậy được gọi là định tuyến theo tin đồn (Routing by rumor). Chúng ta đã phân tích đặc điểm của phương pháp này trong các mục trước. Về cơ bản, định tuyến theo tin đồn nhanh, đơn giản và phù hợp với mạng phẳng, không có cấu trúc phức tạp, trong phần lớn trường hợp hoạt động độc lập không gây ra vấn đề gì. Ngược lại với ưu điểm đó là khả năng có thể xảy ra loop trong mạng. Chúng ta sẽ bàn đến các bài toán cụ thể và cách thức chống loop trong bài sau

### Nhận xét về tốc độ và khả năng hội tụ:



*Hình 4.5 – Distance Vector sẽ hội tụ khởi động rất nhanh, nhưng tái hội tụ chậm*

Thời gian để 1 mạng hội tụ thường tỷ lệ thuận trực tiếp với quy mô của mạng đó. Trong ví dụ trên với việc sử dụng giao thức định tuyến Distane Vector, để mạng hội tụ sẽ cần tối thiểu 5 chu kỳ cập nhật định kỳ để bản tin cập nhật có thể đi từ Router biên bên này tới Router biên xa nhất phía bên kia. Tốc độ hội tụ của 1 giao thức định tuyến là rât quan trọng, vì thông số đó quyết định tính hiệu quả của giao thức định tuyến đó, yêu cầu về tính sẵn sàng của hạ tầng định tuyến nhìn vào tốc độ hội tụ.Khi đánh giá tốc độ hội tụ của 1 mạng, chúng ta cần xem xét các yếu tố:

* Mất bao lâu để Router quảng bá sự thay đổi trong mạng cho Router hàng xóm
* Tốc độ tính toán định tuyến để các tuyến đường được cập nhật lại khi có sự thay đổi

1 hệ thống mạng sẽ chỉ vận hành khi định tuyến hội tụ, do đó người quản trị mạng luôn ưu tiên các giao thức định tuyến có thời gian hội tụ ngắn

## Duy trì bảng định tuyến

Bảng định tuyến là công cụ để Router thực hiện tra cứu tuyến đường, độ ổn định của bảng định tuyến đồng nghĩa với độ ổn định mạng vì định tuyến sai dẫn tới kết nối thất bại. Bảng định tuyến được xây dựng dựa trên các bản tin định tuyến, các bản tin update trao đổi giữa các Router. Vì thế 1 giao thức định tuyến luôn cần những kỹ thuật để duy trì sự ổn định của bảng định tuyến. Bên cạnh đó, những thay đổi trong mạng sẽ yêu cầu giao thức định tuyến hội tụ lại, quá trình hội tụ lại yêu cầu phản ứng nhanh và chính xác, vì khoảng thời gian hội tụ lại càng lâu đồng nghĩa với hiệu quả của mạng càng giảm. Thay đổi trong mạng có thể kể tới như:

* Interface down
* Một interface mới tham gia định tuyến
* Router ngừng hoạt động hoặc khởi động lại
* Thay đổi thuộc tính trên interface

Đối với họ giao thức định tuyến Distance-vector, chúng ta xem xét các kỹ thuật thời gian được sử dụng để duy trì bảng định tuyến

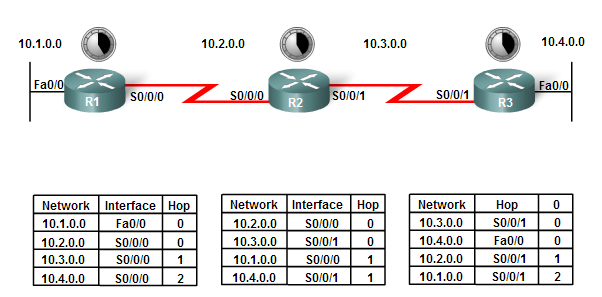
### Cập nhật định kỳ - Periodic Update

Chúng ta đã đề cập tới cập nhật định kỳ như 1 đặc điểm cơ bản của giao thức Distance-vector trong phần đầu của chương. Cập nhật định kỳ là khái niệm Router sẽ gửi bản tin update theo một chu kỳ cố định được xác định từ trước. Đối với RIP, khoảng thời gian cập nhật định kỳ là 30 giây. Cập nhật định kỳ diễn ra độc lập với sự thay đổi trong mạng, nghĩa là bộ đếm trong Router sẽ kích hoạt quá trình gửi update, cho dù trước đó có sự thay đổi hay không.

Quá trình cập nhật định kỳ có 2 ý nghĩa chính:

* Thông báo về sự thay đổi trong mạng và yêu cầu cập nhật lại nếu có sự thay đổi về tuyến đường
* Làm tươi thời gian hợp lệ của bản ghi trong bảng định tuyến. Chúng ta xét 1 trường hợp Router nhận update từ hàng xóm và cập nhật trong bảng định tuyến, sau đó vì một lý do nào đó Router không nhận được các bản tin update tiếp theo từ Router hàng xóm nữa, liệu các bản ghi trước đó có còn hợp lệ. Thời gian sống của bản ghi trong bảng định tuyến chính là một yếu tố để duy trì độ ổn định của bảng định tuyến

Quá trình cập nhật định kỳ có thể dẫn tới các vấn đề lãng phí tài nguyên trong mạng, tuy nhiên do đặc điểm “định kỳ” nên quá trình cập nhật theo chu kỳ vẫn được luôn được sử dụng trong bất kỳ giao thức định tuyến nào, cho dù dưới các dạng khác nhau. Định kỳ trao đổi bản tin, chính là yếu tố đảm bảo kết nối logic và mối quan hệ hàng xóm trong định tuyến giữa các Router, ta gọi những cơ chế có tính chất đó là keepalive. Bản tin cập nhật định kỳ là cơ chế keepalive của Distance-vector, trong các giao thức nâng cao như Link-State, người ta dùng 1 loại bản tin keepalive khác gọi là Hello. Chúng ta sẽ đề cập chi tiết trong các chương sau



*Hình 4.6 – Giao thức Distance-Vector sử dụng update timer cố định 30 giây*

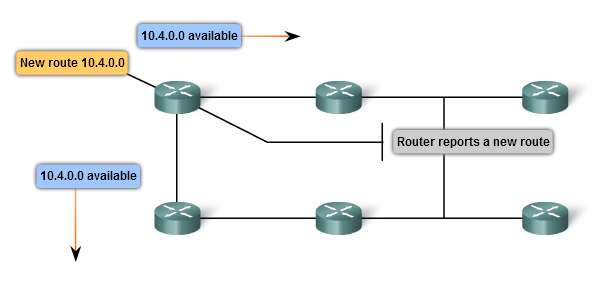
### Cập nhật hạn chế - Bounded Update

Chúng ta thấy hạn chế lớn nhất của cập nhật định kỳ chính là gửi ra toàn bộ thông tin định tuyến và gửi tới tất cả Router hàng xóm. Tuy không trực tiếp ảnh hưởng tới độ ổn định của quá trình định tuyến, nhưng cách thức này không tối ưu tài nguyên mạng, dẫn đến lãng phí băng thông không cần thiết. Một giải pháp được đưa ra và triển khai trên giao thức EIGRP được gọi là cập nhật hạn chế (Bounded update) để giải quyết vấn đề trên.

**Ý tưởng của Bounded update là:**

* Khi có sự thay đổi trong mạng, chỉ có thông tin tuyến đường thay đổi được gửi đi, thay vì toàn bộ thông tin định tuyến như trước đó
* Bản tin cập nhật hạn chế chỉ được gửi cho Router cần nhận update, thay vì tất cả Router

Về mặt vận hành của cập nhật hạn chế, việc chỉ gửi đi thông tin thay đổi sẽ hoàn toàn xóa đi khái niệm “định kỳ” trong việc trao đổi bản tin update. Hay nói cách khác, bản tin update sẽ không còn được sử dụng như cơ chế keepalive. Do đó, giao thức EIGRP sử dụng 1 bản tin khác để thực hiện keepalive, bản tin đó được gọi là bản tin Hello. Ý tưởng về bản tin hello được sử dụng riêng cho mục đích keepalive nên ngoài một vài thông tin liên quan đến thiết lập quan hệ hàng xóm, bản tin Hello không mang bất cứ thông tin nào khác, do đặc điểm kích cỡ nhẹ, nên Hello được trao đổi định kỳ mà không ảnh hưởng nhiều tới tài nguyên mạng.



*Hình 4.7 – Chỉ những thông tin mới được cập nhật*

Đặc điểm thứ hai của cập nhật hạn chế là chỉ gửi cho Router cần nhận update. Như vậy có 2 trường hợp:

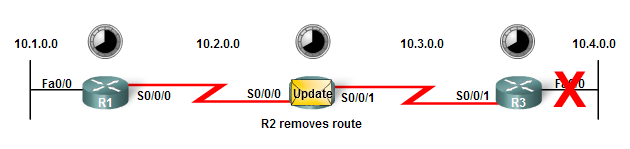
* **Trường hợp thứ nhất –** Nếu đó là tuyến đường tới 1 mạng mới, hoặc thay đổi trực tiếp trên interface, vậy như bản tin cập nhật thông thường, mọi Router vẫn sẽ nhận được cập nhật về tuyến đường
* **Trường hợp thứ hai –** Khi một Router mất thông tin tuyến đường tới 1 mạng đích, Router đó sẽ hỏi (bằng bản tin querry) các Router hàng xóm, và Router hàng xóm sẽ chỉ gửi cập nhật cho riêng Router gửi ra querry. Đó chính là lợi ích trực tiếp của cập nhật hạn chế

Cập nhật hạn chế là kỹ thuật được sử dụng bởi giao thức EIGRP, sẽ đề cập trong các tài liệu khác.

### Cập nhật kích hoạt – Triggered Update

Quay trở lại với trường hợp của cập nhật định kỳ. Chúng ta thấy quá trình cập nhật làm 2 việc, đó là thực hiện cơ chế keepalive và trao đổi thông tin định tuyến. Hãy xét trường hợp khi mạng có thay đổi nhưng quá trình cập nhật định kỳ vừa diễn ra ngay trước đó, các Router sẽ chờ hết 30 giây mới gửi ra update, và cứ thế cứ thế bắt đầu từ Router biết được sự thay đổi trong mạng cho đến Router cuối cùng. Một mạng ổn định không thể chấp nhận việc chờ vài phút để thông tin định tuyến cập nhật trên tất cả Router khi có một sự thay đổi. Để giải quyết vấn đề đó, cập nhật kích hoạt được sử dụng song song với cập nhật định kỳ.

*Ý tưởng của cập nhật kích hoạt* – Khi trên 1 Router phát hiện có sự thay đổi (thay đổi có thể là trạng thái interface thay đổi, 1 tuyến đường mới được đưa vào bảng định tuyến, hoặc ngược lại, 1 tuyến đường cũ bị xóa hoặc không hợp lệ) Router sẽ ngay lập tức gửi ra bản tin update để thông báo sự thay đổi cho các Router hàng xóm, khi Router hàng xóm nhận được bản tin triggered update, Router đó sẽ gửi cho tất cả Router hàng xóm khác trừ Router hàng xóm đã gửi ra bản tin triggerd update đó.



*Hình 4.8 – Có thay đổi, là gửi ngay lập tức*

Cập nhật kích hoạt đã làm tăng tốc độ hội tụ trong mạng lên rất nhiều, tăng tốc độ phản ứng của Router đối với thay đổi trong mạng và tất nhiên tăng hiệu quả sử dụng mạng. Tuy nhiên, bên cạnh những lợi ích rõ nhất về tốc độ hội tụ của cập nhật kích hoạt, có thể kể ra 2 vấn đề có thể nảy sinh:

* **Vấn đề thứ nhất -** Khi một mạng đủ lớn và số lượng interface trên mỗi Router nhiều hơn, đồng nghĩa với xác suất có sự thay đổi trong mạng cao hơn. Bản tin cập nhật kích hoạt được gửi cho tất cả Router hàng xóm, hãy xem nếu sự thay đổi tuyến đường xảy ra hàng loạt, đồng nghĩa với rất nhiều bản tin cập nhật được đẩy đi trong thời gian rất ngắn. Khi đó giải pháp về tốc độ hội tụ lại dẫn đến vấn đề về sự ổn định của định tuyến cũng như tối ưu sử dụng tài nguyên trong mạng. Hãy xét ví dụ trên, nếu tại 1 thời điểm người quản trị “lỡ tay” gõ shutdown interface f0/0 trên R3, rồi sau đó sửa sai ngay lập tức với no shutdown, chỉ trong 1 khoảng thời gian rất ngắn đã có 2 bản tin cập nhật kích hoạt gửi ra bởi R3, một cách không cần thiết.
* **Vấn đề thứ hai –** Do đặc điểm cập nhật kích hoạt được sử dụng song song với cập nhật định kỳ, có thể dẫn tới định tuyến không chính xác. Hãy quay lại ví dụ trên khi R3 phát hiện thấy interface f0/0 DOWN, R3 sẽ kích hoạt triggered update. Tuy nhiên thực tế quá trình đó không diễn ra ngay lập tức, giả sử trước khi gửi ra bản tin cập nhật kích hoạt, R3 nhận được cập nhật định kỳ từ R2 cho mạng đích 10.4.0.0. R3 sẽ cập nhật lại bảng định tuyến bằng cách xét tuyến đường tới mạng 10.4.0.0 qua R2. Tới đây, định tuyến đã thất bại, loop xảy ra giữa R2 và R3

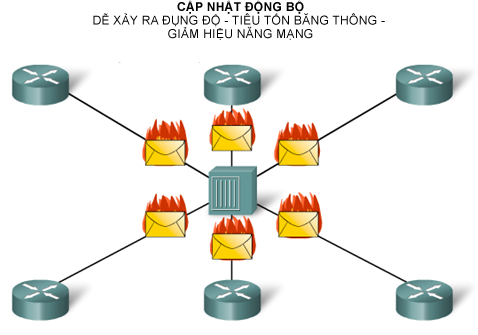
Tuy nhiên, cập nhật kích hoạt vẫn là cách thức gửi update tối ưu được sử dụng bởi tất cả giao thức định tuyến hiện nay, của cả họ Distance-vector và Link-state. Để hạn chế các nhược điểm trên, một số kỹ thuật khác được triển khai “ngầm” bởi giao thức định tuyến. Chẳng hạn với RIP, vấn đề thứ 2 mặc định sẽ không xảy ra với cập nhật kích hoạt do RIP triển khai 1 kỹ thuật chống loop được gọi là split-horizon sẽ được đề cập trong phần sau.

### Cập nhật đồng bộ - Synchroniaztion Update

Khác với 3 kỹ thuật cập nhật ở trên, cập nhật đồng bộ không phải một giải pháp duy trì bảng định tuyến, ngược lại đó là vấn đề trong mạng sử dụng Hub làm thiết bị trung tâm. Chúng ta đã đề cập tới các vấn đề của mạng sử dụng hub trong kỳ trước:

* Hub là thiết bị lớp 1 sẽ nhân bản và gửi bản tin ra mọi cổng trừ cổng nhận bản tin ban đầu
* Môi trường chia sẻ sử dụng Hub tồn tại vấn đề đụng độ bản tin và cực kỳ kém tối ưu với tài nguyên băng thông cũng như thông lượng

Đặt Hub là thiết bị trung tâm kết nối các Router là 1 tối kiến, nhưng điều tồi tệ nhất xảy ra khi tất cả các Router cùng gửi ra bản tin cập nhật định kỳ cùng một thời điểm. Một điều chắc chắn không thể bàn cãi đó là đụng độ gói tin và mất thông tin định tuyến. Hiện tượng trên, tất cả Router cùng gửi ra update tại 1 thời điểm được gọi là cập nhật đồng bộ



*Hình 4.9 – Vấn đề với cập nhật đồng bộ*

Để giải quyết vấn đề trên, chúng ta có một số phương án sau:

* **Phương án tối ưu nhất là thay đổi thiết kế của mạng**, chuyển từ Hub sang Switch. Vì Hub hoàn toàn không phù hợp để đứng vị trí trung tâm kết nối các Router
* **Phương án khả dĩ thứ hai,** đó là thay đổi khoảng thời gian update định kỳ trên các Router, với các khoảng thời gian không giống nhau, xác suất xảy ra cập nhật đồng bộ sẽ rất thấp
* **Phương án thứ 3**, đó là tùy vào việc triển khai giao thức định tuyến nào, sẽ có các kỹ thuật đặc thù đi kèm. Chẳng hạn với RIP, Cisco đưa ra 1 cơ chế được gọi là *RIP\_JITTER*. RIP\_JITTER sẽ thêm vào khoảng thời gian cập nhật định kỳ 1 giá trị ngẫu nhiên có độ lớn từ 0-15% khoảng thời gian cập nhật định kỳ. Như vậy, trên lý thuyết, mặc định bản tin định kỳ sẽ được gửi ra không phải 30 giây, mà là từ 25-30 giây tùy vào gia trị ngẫu nhiên.

## Loop định tuyến – Routing loop

Loop (vòng lặp) là một vấn đề thường hay xảy ra trong quá trình chuyển tiếp gói tin. Như vậy loop có thể xảy ra tại các thiết bị lớp 3 như Router, ta gọi đó là loop định tuyến, hoặc cũng có thể xảy ra tại các thiết bị lớp 2 như Switch, ta gọi đó là loop chuyển mạch. Về mặt nguyên nhân có thể khác nhau, nhưng hình thức thì tương tự.

### Giới thiệu về loop định tuyến

Loop định tuyến là trạng thái 1 gói tin được chuyển tiếp liên tục qua 1 hoặc nhiều Router trong mạng lặp đi lặp lại mà không tới được mạng đích. Loop định tuyến xảy ra khi 2 hoặc nhiều Router học sai thông tin định tuyến dẫn đến việc tìm đường thất bại. Các nguyên nhân có thể xảy ra loop định tuyến:

* Cấu hình tuyễn tĩnh (static route) sai
* Cấu hình phân phối lại (redistribution) không chính xác giữa nhiều giao thức định tuyến khác nhau
* Bảng định tuyến không nhất quán do nhiều nguyên nhân, trong đó có thể mạng hội tụ chậm, các thuộc tính định tuyến của 1 giao thức bị thay đổi trên 1 hoặc nhiều Router,…
* Cấu hình tuyến null (discard route) lỗi – chúng ta sẽ đề cập tới tuyến null trong phần sau

Các giao thức định tuyến Distance-vector có cách thức vận hành rất đơn giản, và sự đơn giản đó dễ dẫn tới vấn đề liên quan tới loop. Thực tế với cơ chế học tuyến đường theo kiểu Distance-vector, loop là vấn đề rất tự nhiên và rất dễ xảy ra, do đó bản thân bên trong mỗi giao thức Distance-vector đều triển khai các phương án chống loop, những phương án này mang tính chắp vá, là những điều kiện hạn chế loop có thể xảy ra. Ngược lại, các giao thức Link-State tiếp cận việc cập nhật tuyến đường theo 1 cách khác, mà theo đó loop hầu như không bao giờ xảy ra

Bản thân giao thức IP cũng có một cơ chế chống loop, đó là sử dụng trường TTL - Time- to-Live để ngăn trường hợp 1 gói tin được chuyển tiếp trong mạng mãi mãi. Cứ đi qua 1 Router (hop), trường TTL giảm đi 1, cho đến khi trường TTL bằng 0, gói tin sẽ bị loại bỏ

### Ảnh hưởng của loop định tuyến

Loop là 1 vấn đề hay gặp nhưng có ảnh hưởng rất tiêu cực tới hệ thống mạng, không chỉ làm giảm hiệu năng của mạng, loop định tuyến đồng nghĩa với đứt kết nối, có thể dẫn tới sập hạ tầng định tuyến trong mạng:

* Tiêu tốn rất nhiều băng thông không cần thiết trong mạng
* Giảm hiệu năng CPU trầm trọng khi phải xử lý những gói tin loop
* Gói tin loop cũng làm ảnh hưởng tiêu cực đến quá trình hội tụ lại trong mạng

Khi loop xảy ra, mạng bị nghẽn, các gói tin update có thể không được xử lý và chuyển tiếp hợp lý và đúng lúc. Mạng trở nên thiếu ổn định và có thể dẫn đến các kết quả xấu hơn: như mất kết nối hàng xóm, hoặc trạng thái interface up/down không ổn định

Các cơ chế chống loop được sử dụng bởi giao thức Distance-vector bao gồm:

* Giới hạn hop-count
* Holddown timer
* Split horizon
* Route poisoning
* Cập nhật kích hoạt

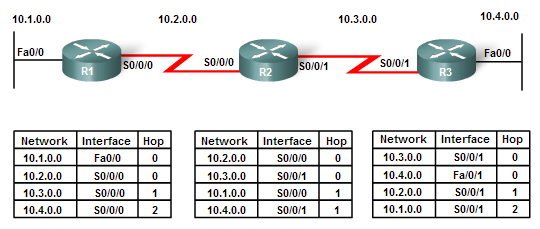
Trong phần tiếp theo chúng ta sẽ đi sâu vào từng phương pháp chống loop, tuy nhiên các bài toán đề nghị đều là các bài toán mang tính mô tả, không có thực, vì thực tế các giao thức Distance-vector đã triển khai sẵn các phương pháp trên để ngăn chặn loop. Chúng ta sẽ xem xét bài toán khi chưa có các phương pháp điều khiển kia và cách thức từng phương pháp giải quyết loop trong các trường hợp cụ thể

### Giải pháp thứ nhất – Giới hạn hop-count

1. **Bài toán**

Đề cập tới đặc điểm của giao thức Distance-vector, chúng ta thấy 1 Router sẽ gửi update là toàn bộ bảng định tuyến sang cho Router hàng xóm, Router hàng xóm sẽ đọc thông tin định tuyến, những tuyến đường chưa có trong bảng định tuyến hoặc có nhưng có metric tối ưu hơn sẽ được cập nhật, những tuyến đường có metric kém tối ưu hơn sẽ bị loại bỏ. Chúng ta xem xét bài toán dưới đây:

**Cho topology:**



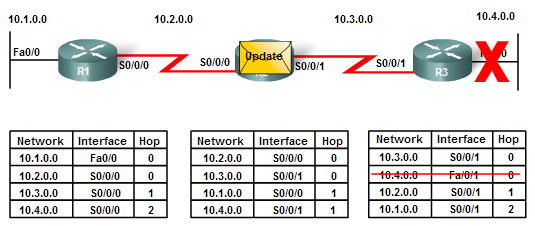
3 Router R1, R2, R3 nối với nhau như topology và triển khai định tuyến Distance-vector. Giao thức được sử dụng minh họa là RIP, sẽ được giới thiệu trong chương 5. RIP sử dụng hop-count (chiều dài của tuyến đường được xác định bởi số lượng Router đi qua) là metric, nghĩa là tuyến đường đi qua càng ít Router sẽ càng tối ưu hơn. Mạng hội tụ, các Router cập nhật thông tin tuyến đường tới các mạng đích và metric của từng tuyến đường

**Vấn đề:**

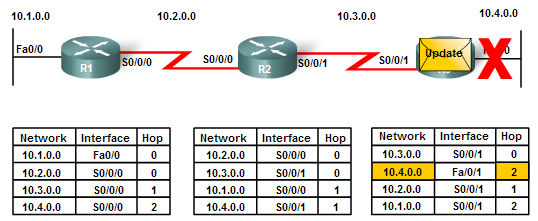
* 1. **Bước 1 -** R3 phát hiện mât kết nối tới mạng 10.4.0.0 trên interface fastEthernet0/1. R3 xóa bản ghi tuyến đường về mạng 10.4.0.0 trong bảng định tuyến và cố gắng gửi cập nhật sang cho R2 (thông qua cập nhật kích hoặc cập nhật định kỳ)



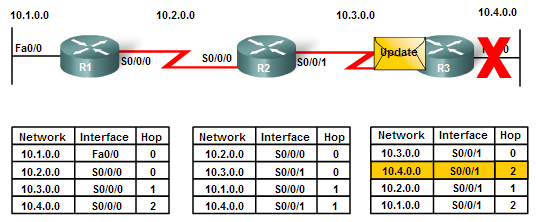
* 1. **Bước 2 -** Trước khi R3 gửi cập nhật mới nhất sang cho R2, R2 gửi cập nhật định kỳ là toàn bộ bảng định tuyến sang cho R3, trong đó có tuyến 10.4.0.0 cũ với metric bằng 2



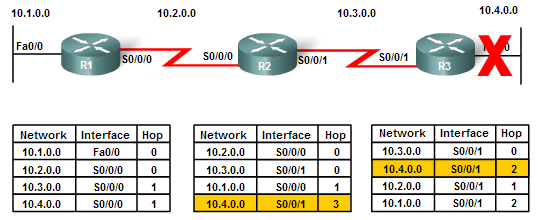
* 1. **Bước 3 -** R3 cập nhật tuyến đường tới mạng 10.4.0.0 qua R2 (interface Serial0/0/1) với metric bằng 2. R3 xem như đây là cập nhật mới cho tuyến đường tới mạng 10.4.0.0 nên thêm tuyến đường với metric bằng 2 vào bảng định tuyến



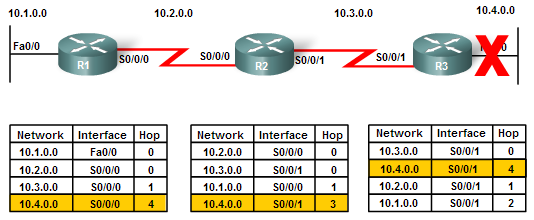
* 1. **Bước 4 –** Sau 30 giây, R3 gửi cập nhật định kỳ sang cho R2, trong đó có mạng 10.4.0.0 với metric bằng 3 (metric cũ bằng 2)



* 1. **Bước 5 –** R2 nhận cập nhật từ R3 về mạng 10.4.0.0 với metric mới bằng 3. R2 chấp nhận cập nhật này, vì bản ghi trong bảng định tuyến về mạng 10.4.0.0 ban đầu là nhận từ R3 qua interface Serial0/0/1. R2 cập nhật lại bảng định tuyến cho tuyến đường tới mạng 10.4.0.0 lúc này có metric bằng 3



* 1. **Bước 6 –** R2 lại gửi cập nhật định kỳ sang cho R1 và R3, lúc này tuyến đường tới mạng 10.4.0.0 được gửi đi với metric bằng 4. Cả R1 và R3 đều cập nhật lại bảng định tuyến vì tuyến đường tới mạng 10.4.0.0 lúc này đang là cập nhật từ R2 (R1 và R3 tin tưởng R2, trong khi R2 vẫn đinh ninh tin tưởng R3)



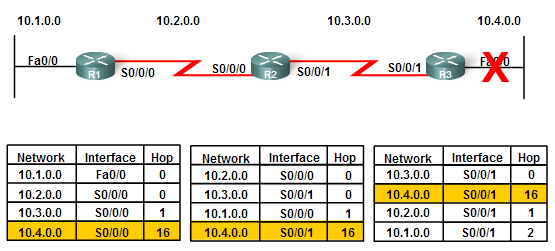
* 1. **Bước 7 –** Lặp lại các bước 4, 5 6,…

Ta nhận ra vấn đề đó là metric (hop count) cứ tăng dần đến vô cùng. Router R1, R2 và R3 đều cập nhật sai thông tin định tuyến và cả 3 Router đều lưu 1 tuyến đường “ma” tới mạng 10.4.0.0 với metric càng lúc càng tăng dần. Giả sử 1 PC từ mạng LAN của Router R1 gửi tới mạng 10.4.0.0 sẽ được đẩy sang Router R2, rồi Router R3, rồi Router R2 và rồi lại Router R3,… cho đến khi kết thúc TTL (TTL bằng 0)

1. **Phương pháp giới hạn hop-count**

Để giải quyết bài toán trên, các giao thức Distance-vector đưa ra 1 giới hạn cho hop-count. Khi 1 tuyến đường có hop-count chạm tới giới hạn, tuyến đường đó sẽ được gán mác “unreachable” Khi 1 tuyến đường bị xem là unreachable, tuyến đường đó sẽ không được cập nhật vào bảng định tuyến

Đối với trường hợp trên, với giao thức RIP sử dụng hop-count là metric, thì phương pháp giới hạn hop-count không chỉ là cách thức chống loop trong bài toán trên, mà còn được sử dụng cho các phương pháp khác như Route poisoning (đề cập trong phần 4). RIP sử dụng hop-count băng 16 là giới hạn metric



*Hình 4.10 – metric tối đa với RIP, giới hạn hop-count bằng 16*

Không phải giao thức Distance-vector nào cũng sử dụng hop-count là metric, chẳng hạn IGRP hay EIGRP sử dụng rất nhiều thông số như băng thông, độ tải, độ trễ, độ tin cậy,… để tính ra metric cho 1 tuyến đường. Tuy nhiên các giao thức khác vẫn sử dụng hop-count như 1 thông số tham khảo để giới hạn quy mô của hạ tầng định tuyến. **Ý nghĩa thứ 2 của phương pháp giới hạn hop-count chính là xác định quy mô cho phép của 1 mạng định tuyến.** Với giao thức RIP, đường kính tối đa của miền định tuyến cho phép bằng 16 Router. Với giao thức EIGRP, đường kinh tối đa mặc định cho phép của miền định tuyến bằng 100 Router, tuy nhiên có thể điều chỉnh để tăng lên.

### Giải pháp thứ hai – Holddown timer

1. **Bài toán**

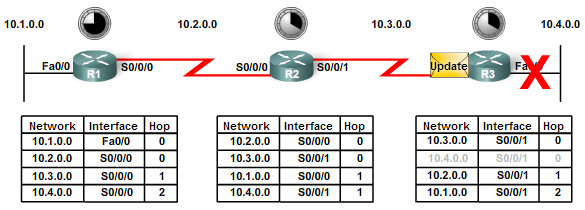
Trong phần trước, chúng ta đã đề cập tới 1 vấn đề của kỹ thuật cập nhật kích hoạt đó là: Nếu 1 interface trên 1 Router “nhảy” trạng thái up/down liên tục do 1 vấn đề gì đó, Router sẽ gửi ra cập nhật kích hoạt tới tất cả Router hàng xóm về trạng thái tuyến đường. Ta thấy vấn đề đó là sự thiếu ổn định của định tuyến, và sự thiếu ổn định này có thể dẫn tới những vấn đề như loop. Đặc biệt khi cập nhật định kỳ kết hợp với cập nhật kích hoạt, chúng ta thấy loop vẫn xảy ra như trường hợp trên cho tới giới hạn hop-count.

1. **Phương pháp Holddown Timer**

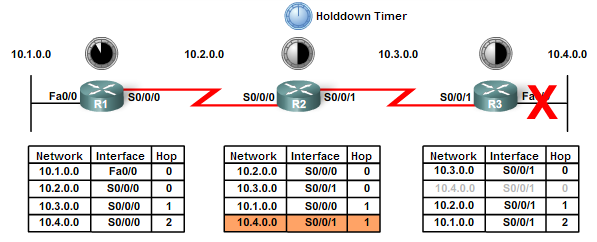
Để giải quyết bài toán trên, chúng ta có phương pháp thứ 2 là sử dụng một bộ đếm được gọi là holddown timer. Holddown timer được sử dụng để tránh việc định tuyến không ổn định do cập nhật kích hoạt diễn ra quá nhanh và thiếu tin cậy. Đồng thời holddown timer cũng tránh việc loop “dần dần” cho tới giá trị giới hạn hop-count.

**Nguyên tắc của Holddown timer** Khi 1 mạng đích bị xác định là mất kết nối dẫn đến 1 tuyến đường bị xác định là không tồn tại hoặc có thể không còn tồn tại, bất kỳ thông tin định tuyến nào về mạng đích đó có metric kém hơn thông tin cũ, thì thông tin cập nhật mới đó sẽ bị loại bỏ trong 1 khoảng thời gian gọi là holddown timer. Trong khoảng thời đó, tuyến đường vẫn được lưu trong bảng định tuyến và vẫn được sử dụng để định tuyến, nhưng được gán mác “unreachable” Khi đó Router sẽ không gửi cập nhật về mạng đó nữa, đồng thời cũng không nhận cập nhật có metric kém hơn. Điều này để đảm bảo thông tin định tuyến mới nhất được cập nhật trên tất cả Router trong mạng

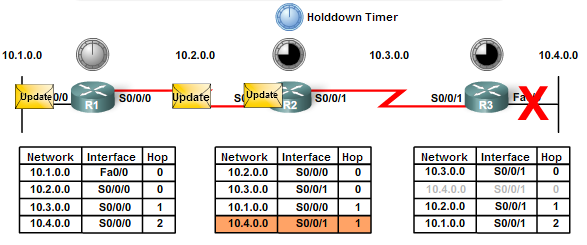
* 1. **Bước 1 –** Router R3 xác định mất kết nối tới mạng 10.4.0.0 trên interface Serial0/0/1. Router R3 xóa tuyến đường tới mạng 10.4.0.0 khỏi bảng định tuyến và cố gắng gửi cập nhật kích hoạt sang cho Router R2



* 1. **Bước 2 –** Router R2 nhận bản tin cập nhật từ R3 và cập nhật lại tuyến đường tới mạng 10.4.0.0 trong bảng định tuyến là unreachable Đồng thời kích hoạt Holddown timer



* 1. **Bước 3 –** Router R2 nhận thấy có sự thay đổi nên gửi cập nhật kích hoạt sang cho R1 có mang trạng thái mới của tuyến đường tới mạng 10.4.0.0. Đồng thời lúc đó R1 gửi cập nhật định kỳ sang cho R2 trong đó có thông tin tuyến đường về mạng 10.4.0.0 cũ với metric bằng 3



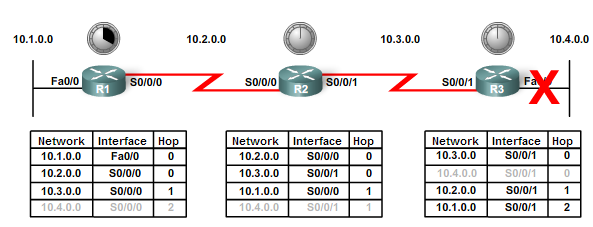
* 1. **Bước 4 –** Router R2 nhận thấy cập nhật về mạng 10.4.0.0 có metric bằng 3 kém hơn thông tin hiện tại, đồng thời thấy tuyến 10.4.0.0 đang ở trạng thái unreachable nên bỏ thông tin cập nhật về mạng 10.4.0.0 từ R1. R1 nhận thấy cập nhật kích hoạt về mạng 10.4.0.0 từ R2, nên tương tự đặt tuyến đường tới mạng 10.4.0.0 vào trạng thái unreachable kết hợp với kích hoạt bộ đếm holddown timer



* 1. **Bước 5 –** Dữ liệu được gửi từ R1 hoặc R2 tới mạng 10.4.0.0 sẽ vẫn được định tuyến tới Router R3, tuy nhiên lúc này R3 đã xác định mât kết nối tới mạng 10.4.0.0 nên sẽ bỏ các gói tin đó đi. Hành động này gián tiếp tiêu tốn băng thông không cần thiết trên kết nối giữa R1-R2 và R2-R3



* 1. **Bước 6 –** Kết thúc Holddown Timer trên lần lượt Router R2 và R1. 2 Router này xóa thông tin về tuyến đường tới mạng 10.4.0.0 trong bảng định tuyến



Như vậy, với Holddown timer, sẽ không bao giờ xảy ra trường hợp 1 tuyến đường cập nhật metric về 1 tuyến đường chạm tới giới hạn hop-count. Đồng thời tránh được các vấn đề định tuyến không ổn định do cập nhật kích hoạt liên tục.

Nhược điểm lớn nhất của phương pháp này chính là gián tiếp tạo ra 1 “hố đen” trong mạng định tuyến. Khi 1 Router xác định 1 tuyến đường là “unreachable” Router đó nghi ngờ mạng đích có thể không tồn tại, nhưng không chắc chắn về điều đó, nên Router vẫn tiếp tục định tuyến cho mạng đó. Như vậy, nếu mạng đích chính xác đã mất kết nối, các Router vẫn gửi gói tin theo tuyến đường cũ và tiêu tốn 1 lượng băng thông vô ích trong khoảng thời gian Holddown timer

### Giải pháp thứ ba – Split horizon kết hợp Route poisoning

1. **Bài toán**

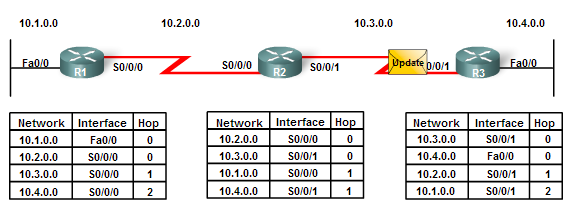
Quay lại vấn đề của bài toán đầu tiên, chúng ta thấy Router R3 quảng bá thông tin về mạng đích 10.4.0.0 cho Router R2, sau đó Router R2 gửi cập nhật cho Router R1 và R3 thông qua cập nhật định kỳ. Do đặc điểm của giao thức Distance-vector dựa trên sự “tin tưởng” giữa các Router hàng xóm, vì thế nếu Router R2 tin tưởng bản tin cập nhật từ R3, và ngược lại, cho cùng 1 tuyến đường tới mạng đích, thì sẽ thật là phi logic, và thực tế điều đó dẫn tới vấn đề loop chúng ta đã gặp

1. **Nguyên tắc Split Horizon**

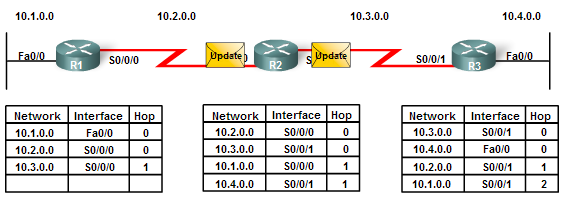
Một kỹ thuật được đưa vào để giúp tối ưu quá trình cập nhật định tuyến cho các giao thức Distance-vector, đó là Split Horizon. **Luật Split Horizon nói rằng 1 Router không nên quảng bá tuyến đường tới 1 mạng đích ra 1 interface mà Router nhận đã cập nhật tuyến đường từ interface đó** và ngược lại, 1 Router không nên nhận bản tin cập nhật tuyến đường tới 1 mạng đích từ 1 interface mà Router đã sử dụng để quảng bá cập nhật cho tuyến đường đó. Từ “nên” ở đây được sử dụng vì có những trường hợp không tránh khỏi việc Router phải gửi cập nhật và nhận bản tin cập nhật về cùng 1 mạng đích trên cùng 1 interface, chúng ta sẽ đề cập tới vấn đề đó sau

Trong ví dụ trên, nếu áp dụng luật Split Horizon, chúng ta xem xét cách thức các Router cập nhật định tuyến tới mạng 10.4.0.0

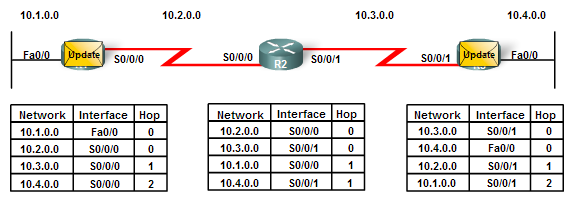
* 1. **Bước 1 –** Router R3 quảng bá mạng 10.4.0.0 sang cho R2 từ interface Serial0/0/1



* 1. **Bước 3** – Router R2 quảng bá mạng 10.3.0.0 và 10.4.0.0 sang cho R1 từ interface Serial0/0/0, quảng bá mạng 10.1.0.0 và 10.2.0.0 sang cho R3 từ interface Serial0/0/1.. Router R2 sẽ tương ứng không quảng bá ngược lại cho R3 và R1 những tuyến đường đã học từ các Router này



* 1. **Bước 4 –** Router R1 nhận bản tin cập nhật và thêm tuyến đường tới mạng 10.4.0.0 qua interface Serial0/0/0 vào bảng định tuyến



* 1. **Bước 5 –** Do luật Split-horizon, Router R1 sẽ không quảng bá ngược lại mạng 10.4.0.0 cho Router R2

Tương tự với mạng 10.4.0.0, chúng ta xem xét quá trình quảng bá tất cả các mạng trên các Router:

* R2 quảng bá mạng 10.3.0.0 và 10.4.0.0 sang cho R1.
* R2 quảng bá mạng 10.1.0.0 and 10.2.0.0 sang cho R3.
* R1 quảng bá mạng 10.1.0.0 sang cho R2.
* R3 quảng bá mạng 10.4.0.0 sang cho R2.

Kỹ thuật Split-horizon mặc định được bật trên các giao thức Distance-vector, tuy nhiên trong những trường hợp nhất định có thể dẫn tới cập nhật thiếu thông tin định tuyến. Do đó người quản trị có thể bật hoặc tắt kỹ thuật này trên các giao thức định tuyến

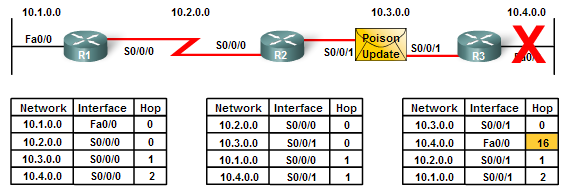
1. **Phương pháp Split horizon kết hợp Route Poisoning**

Một kỹ thuật chống loop sử dụng đồng thời ký thuật Split-horizon kết hợp với giới hạn hop-count được đưa ra để giải quyết bài toán chống loop, tăng tốc độ hội tụ nhanh hơn rất nhiều so với Holddown timer, và hiển nhiên, kỹ thuật này cũng tránh được bài toán metric tăng “dần dần” tới giới hạn hop-count

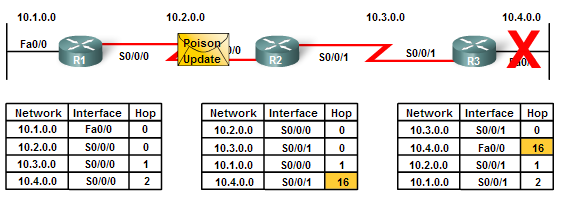
**Ý tưởng của Split horizon kết hợp Route Poisoning –** Khi 1 Router xác định mất kết nối tới 1 mạng đích, Router sẽ cập nhật lại thông tin tuyến đường tới mạng đích đó trong bảng định tuyến với giá trị metric mới bằng giới hạn hop-count hay metric unreachable (bằng 16 với giao thức RIP). Sau đó thông qua cập nhật kích hoạt, Router sẽ gửi cập nhật định tuyến về tuyến đường đó với metric unreachable cho Router hàng xóm. Như vậy đồng loạt tất cả Router sẽ cập nhật lại thông tin bảng định tuyến để không sử dụng tuyến đường cũ nữa.

Chúng ta xem xét với ví dụ cũ:

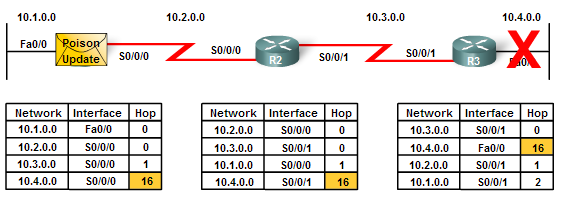
* 1. **Bước 1 –** Router R3 mất kết nối tới mạng 10.4.0.0, ngay lập tưc cập nhật lại trong bảng định tuyến tuyến đường tới mạng 10.4.0.0 với metric bằng 16. Metric này là metric unreachable, sau đó gửi cập nhật sang cho R2 thông qua cập nhật kích hoạt. Bản tin cập nhật này về mạng 10.4.0.0 có metric bằng 16, nên gọi là poison update



* 1. **Bước 2 –** Router R2 nhận bản tin update, ngay lập tức cập nhật lại thông tin về tuyến đường tới mạng 10.4.0.0 có metric bằng 16. Rồi gửi kích hoạt poison update sang cho R1



* 1. **Bước 3 –** Router R1 cập nhật tuyến đường tới mạng 10.4.0.0 với metric bằng 16

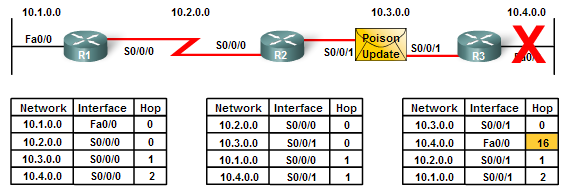


### Giải pháp thứ tư – Poison Reverse

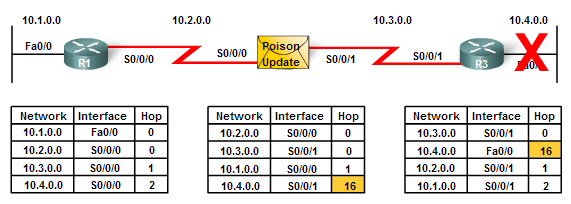
Ngoài kỹ thuật Split-horizon kết hợp Route poisoning, các giao thức Distance-vector còn 1 kỹ thuật với mục đích tương tự, đó là Poison Reverse. Như đã đề cập, Split-horizon không phải luôn được sử dụng 100%, có những trường hợp để đảm bảo định tuyến thông suốt, cần phải tắt chức năng Split-horizon, khi không còn Split-horizon thì Route poisoning cũng không thể triển khai. Để giải quyết các vấn đề trên, 1 kỹ thuật được sử dụng là Poison Reverse

**Ý tưởng của Poison Reverse –** Khi 1 Router nhận được 1 bản tin cập nhật “poisoned” từ hàng xóm, Router đó sẽ cập nhật lại metric của tuyến đường tương ứng với giá trị unreachable. Sau khi cập nhật, Router đó sẽ quảng bá lại bản tin cập nhật “poisoned” tương tự với metric là unreachable, ra interface mà nó vừa nhận bản tin cập nhật. Chúng ta xem xét ví dụ ban đầu:

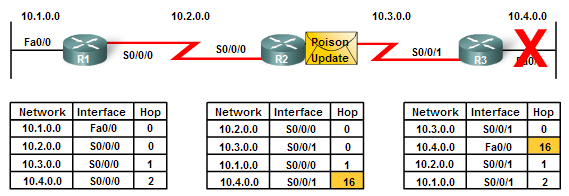
1. **Bước 1 –** Router R3 mất kết nối tới mạng 10.4.0.0, ngay lập tưc cập nhật lại trong bảng định tuyến tuyến đường tới mạng 10.4.0.0 với metric bằng 16. Metric này là metric unreachable, sau đó gửi cập nhật sang cho R2 thông qua cập nhật kích hoạt. Bản tin cập nhật này về mạng 10.4.0.0 có metric bằng 16, nên gọi là poisond update



1. **Bước 2 –** Router R2 nhận bản tin update, ngay lập tức cập nhật lại thông tin về tuyến đường tới mạng 10.4.0.0 có metric bằng 16.



1. **Bước 3 –** Sau đó Router R2 gửi lại bản tin cập nhật “poison” ngược lại cho Router R3



**Mục đích của Poison Reverse –** Hẳn bây giờ chúng ta đang thắc mắc, tại sao Router R2 lại cần gửi lại ngược về cho Router R3 ? Có 2 mục đích chính của Poison Reverse:

* Tránh trường hợp 1 lúc nào đó, vì 1 lý do nào đó Router R2 lại học được 1 tuyến đường 10.4.0.0 có metric tốt hơn metric unreachable, Router R2 sẽ gửi lại cho Router R3, và Router R3 sẽ cập nhật lại. Rõ ràng nếu mạng 10.4.0.0 là trực tiếp kết nối với Router R3 thì không lý nào Router R3 lại chấp nhận cập nhật từ Router R2. Hay nói cách khác, Router R2 chủ động thông báo cho R3 rằng hãy bỏ qua những cập nhật từ R2 về mạng 10.4.0.0
* Trường hợp thứ 2 được đề cập tới trong môi trường đa truy cập không quảng bá – NBMA (Non broadcast MultiAccess), đây là môi trường mà bắt buộc Split Horizon phải tắt đề định tuyến có thể thông suốt. Chúng ta sẽ đề cập tới NBMA trong những kỳ sau

## Tổng kết

Trong chương này, chúng ta có khái quát về nhóm giao thức IGP đâu tiên là Distance Vector:

* Các đặc điểm của giao thức Distance-Vector
* Quá trình giao thức Distance Vector trao đổi thông tin định tuyến
* Cách giao thức Distance Vector duy trì bảng định tuyến
* Lý do và ảnh hưởng của loop
* Các phương pháp chống loop của giao thức Distance Vector

# ****CHƯƠNG 5 - RIPv1****

Trong chương này, chúng ta sẽ xem xét các nội dung sau:

* Mô tả chức năng, đặc điểm và vận hành của RIP
* Cấu hình cơ bản RIP
* Kiểm tra và gỡ rối RIP

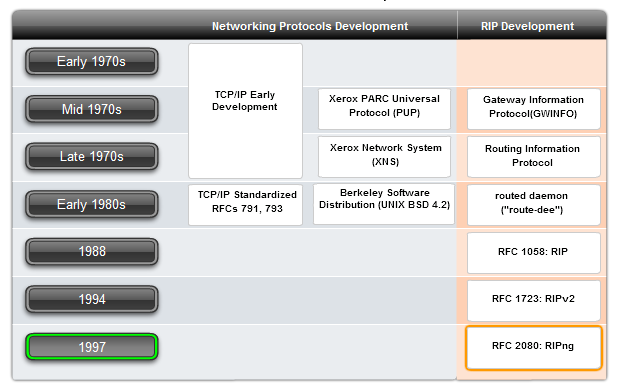
Theo dõi một vài bài toán đặc thù liên quan đến RIP

## Giới thiệu giao thức RIP

RIP là một trong những giao thức định tuyến động đầu tiên và vẫn còn được sử dụng đến ngày nay vì đặc điểm đơn giản, dễ triển khai, dễ gỡ rối, phù hợp với mô hình mạng nhỏ. Chúng ta đã học về các đặc điểm của giao thức Distance-vector và ta sẽ gặp lại hầu hết các đặc điểm đó trong việc xem xét quá trình vận hành của RIP. Nắm về RIP là nắm được nền tảng của họ giao thức Distance-vector, từ đó dễ dàng so sánh và đánh giá với các giao thức họ Link-State. Nắm về RIP là nắm được nền tảng của định tuyến động với những khái niệm cơ bản nhưng xuyên suốt trong mọi giao thức định tuyến

### Lịch sử phát triển

Nền tảng của RIP là một giao thức được phát triển bởi Xerox những năm 1970 tên là Gateway Information Protocol (GWINFO) sau đó được đổi tên thành RIP. Trong thời kỳ đầu của TCP/IP, RIP được triển khai bởi nhiều nhà cung cấp phần cứng mạng khác nhau với các cải tiến nhỏ so với RIP ban đầu. RIP được sử dụng rộng rãi tại thời điểm đó dẫn đến yêu cầu chuẩn hóa. RIP hay RIPv1 được chuẩn hóa vào năm 1988 bởi Charles Hedrick trong RFC1058, và dần dần phát triển để phù hợp với nhu cầu thực tế. Từ Classful Routing trở thành Classless Routing trong bản chuẩn hóa sau đó – RIPv2 – mô tả trong RFC1723. Tới năm 1997, RIPng chính thức được chuẩn hóa để hỗ trợ Ipv6



*Hình 5.1 – Lịch sử phát triển RIP*

### Đặc điểm của RIPv1

Chúng ta đã xem xét rất nhiều đặc điểm của Distance-vector trong chương trước, theo đó :

* RIP là 1 giao thức Distance-vector
* RIP xác định thước đo tuyến đường bằng số lượng Router (hop) cần đi qua để tới mạng đích – Metric của RIP được gọi là hop-count
* Hop-count bằng 15 bị xem là “unreachable”
* Sử dụng UDP cổng 520 để đóng gói bản tin update
* Sử dụng địa chỉ đích Broadcast 255.255.255.255 để gửi bản tin update
* RIP sử dụng Classful routing – Chúng ta sẽ phân tích nhược điểm của Classful Routing trong phần dưới
* RIP sử dụng AD cố định bằng 120

### Các thông số thời gian của RIP

RIP sử dụng nhiều thông số thời gian để quản lý thông tin cập nhật

**Update Timer**: Mô tả chu kỳ RIP gửi ra bản tin cập nhật định kỳ

**Invalid Timer**: Thời gian “sống” tối đa của 1 tuyến đường cập nhật, hay *invalid timer* bằng 180 giây. Nếu trong phạm vi 180 giây, Router không nhận được cập nhật về tuyến đường thì tuyến đường đó sẽ bị coi là unreachable và bước vào Holddown timer

**Holddown Timer:** Thời gian “hoãn” của 1 tuyến đường, hay **holddown timer** bằng 180 giây. Trong phạm vi Holddown timer, tuyến đường bị xem là unreachable, Router sẽ không chấp nhận bất cứ thông tin cập nhật nào về tuyến đường đó mà có metric bằng hoặc kém hơn

**Flush Timer**: Thời gian “tồn tại” tối đa của 1 tuyến đường kể từ lần cuối cùng nhận thông tin cập nhật cho tới khi tuyến đường đó bị xóa khỏi bảng định tuyến, **flush timer** bằng 240 giây

Thông tin thời gian có thể kiểm tra với show ip protocol

R2#show ip protocols

Routing Protocol is "rip"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Sending updates every 30 seconds, next due in 23 seconds

Invalid after 40 seconds, hold down 10, flushed after 90

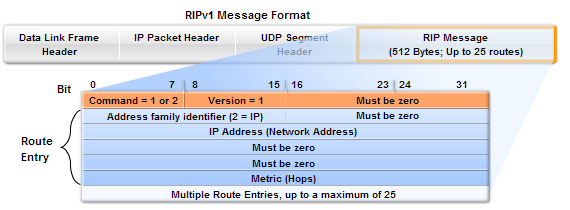
Redistributing: rip

Default version control: send version 2, receive version 2

<output omitted>

## Vận hành của RIPv1

### Cấu trúc bản tin RIP



*Hình 5.2 – Cấu trúc bản tin RIPv1*

Các trường quan trọng trong header của bản tin RIP bao gồm:

**Command –** Trường xác định loại bản tin, có thể nhận 2 giá trị 1 hoặc 2. Với Command bằng 1 chỉ bản tin yêu cầu, command bằng 2 chỉ bản tin phản hồi

**Version –** Trường xác định phiên bản RIP đang sử dụng. Đây là bản tin RIPv1 nên trường version bằng 1

**Must be zero –** Là các trường để trống, các trường này được sử dụng để mang thông tin nâng cao hơn trong các phiên bản cao hơn của RIP

**Address family Identifier (AFI) –** RIP có thể hỗ trợ mang thông tin định tuyến cho nhiều giao thức khác nhau, AFI bằng 2 được sử dụng cho IP. AFI bằng 0 khi yêu cầu gửi ra toàn bộ bảng định tuyến

**IP address –** Địa chỉ IP mạng đích, có thể là địa chỉ mạng hoặc địa chỉ host

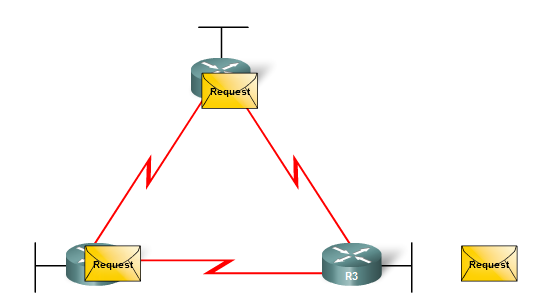
**Metric –** Mang thông tin metric cho tuyến đường tới mạng đích

Một bản tin RIP update có thể mang thông tin cho 25 tuyến đường tới mạng đích, cho phép chiều dài tối đa cho dữ liệu của bản tin RIP lên tới 512 byte.

### Vận hành RIPv1

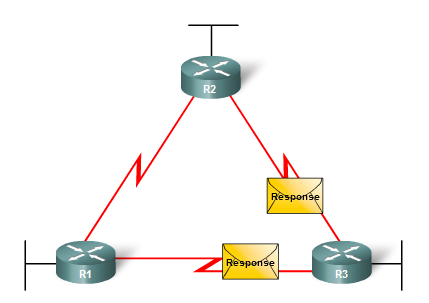
Vận hành của RIP rất đơn giản, chỉ gồm 3 bước trao đổi bản tin, tương ứng với 2 loại bản tin là Yêu cầu (request) và phản hồi (Respond)

**Bước 1 -**  1 Router chạy RIP bắt đầu quá trình trao đổi bản tin update, vì Router không biết được có những Router hàng xóm nào xung quanh và cũng không có cơ chế nào thiết lập mối quan hệ hàng xóm thực sự nên lúc này, RIP sẽ gửi ra bản tin Yêu cầu. Bản tin này yêu cầu các Router hàng xóm hãy gửi ra tất cả thông tin định tuyến.



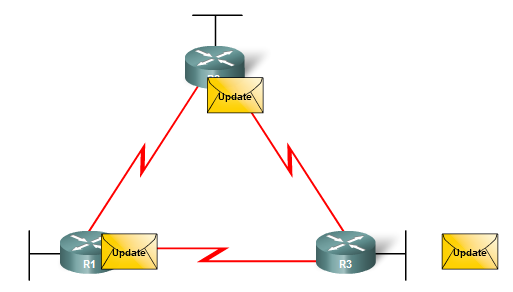
*Hình 5.4 – R3 gửi bản tin request ra tất cả interface*

**Bước 2 -** Bản tin phản hồi được gửi lại bởi Router hàng xóm (Router có kết nối trực tiếp và cũng chạy RIPv1), đây là phản hồi là bản tin Unicast, chỉ gửi tới Router đã gửi ra Broadcast. Router nhận được phản hồi của Router hàng xóm sẽ đọc thông tin tuyến đường và so sánh với bảng định tuyến hiện tại. Router chỉ cập nhật những tuyến đường tới mạng đích hiện tại chưa có hoặc tuyến đường tới cùng mạng đích nhưng có metric tối ưu hơn (tất nhiên nếu đây là lần trao đổi update đầu tiên sẽ không có trường hợp nhận được tuyến đường có metric tối ưu hơn, vì không có tuyến đường nào tối ưu hơn kết nối trực tiếp), các tuyến đường còn lại sẽ bị loại bỏ.



*Hình 5.4 – R1 và R2 gửi lại bản tin phản hồi mang thông tin định tuyến*

**Bước 3 -** Sau đó Router khởi tạo sẽ gửi bản tin cập nhật kích hoạt ra tất cả interface chạy RIP để quảng bá thông tin tuyến đường RIP của mình.



*Hình 5.5 R3 gửi cập nhật cho các Router hàng xóm*

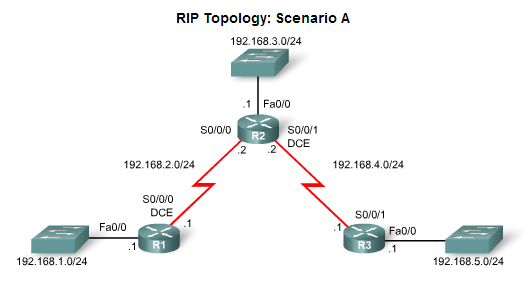
## Cấu hình RIPv1

RIPv1 là 1 giao thức cơ bản, trong chương này chúng ta sẽ làm quen với cấu hình một vài chức năng của giao thức RIPv1. Các chức năng bao gồm:

* Quảng bá 1 mạng vào định tuyến
* Điều khiển quá trình gửi/nhận cập nhật trên 1 interface
* Cơ chế tự động gộp tuyến – Auto-summary
* Quảng bá tuyến mặc định dèfault-route

Toàn bộ phần tiếp theo sẽ được trình bày dựa trên topology dưới đây

**Cho topology:**



**Yêu cầu:**

Triển khai định tuyến động trong mạng, sử dụng RIPv1, sao cho tất cả thiết bị có thể kết nối với nhau

Chúng ta sẽ sử dụng topology trên để minh họa cho toàn bộ cấu hình của phần này. Chúng ta có bảng định tuyến hiện tại trên các Router:

Trên R1:

R1#show ip route

<output omitted>

C 192.168.1.0/24 is directly connected, FastEthernet0/0

C 192.168.2.0/24 is directly connected, Serial0/0/0

Trên R2

R2#show ip route

<output omitted>

C 192.168.2.0/24 is directly connected, Serial0/0/0

C 192.168.3.0/24 is directly connected, FastEthernet0/0

C 192.168.4.0/24 is directly connected, Serial0/0/1

Trên R3

R2#show ip route

<output omitted>

C 192.168.4.0/24 is directly connected, Serial0/0/1

C 192.168.5.0/24 is directly connected, FastEthernet0/0/0

### Quảng bá 1 mạng vào RIP

**Cú pháp**

Để bắt đầu cấu hình 1 giao thức định tuyến, chúng ta cần vào sub-mode Router của CLI. Từ mode Configure Terminal, ta gõ router rip

Router#config t

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#**router ?**

bgp Border Gateway Protocol (BGP)

egp Exterior Gateway Protocol (EGP)

eigrp Enhanced Interior Gateway Routing Protocol (EIGRP)

igrp Interior Gateway Routing Protocol (IGRP)

is-is ISO IS-IS

mobile Mobile routes

odr On demand stubs Routes

ospf Open Shortest Path First (OSPF)

rip Routing Information Protocol (RIP)

Router(config)#**router rip**

Router(config-router)#

Để dừng một tiến trình định tuyến, ví dụ RIP ta sử dụng no router rip, khi sử dụng no router rip, mọi cấu hình trước đó về RIP sẽ bị xóa

Để quảng bá 1 mạng vào RIP, chúng ta sử dụng cú pháp network trong sub-mode Router. Cú pháp hoàn chỉnh

Router(config-router)#network {địa chỉ mạng classful có kết nối trực tiếp}

Trong đó:

**Network –** Kích hoạt tiến trình RIP trên 1 interface nào đó, lúc này interface có thể gửi và nhận các bản tin RIP, định kỳ gửi ra update mang thông tin mạng gắn với interface đó

**{Địa chỉ mạng classful có kết nối trực tiếp} –** Địa chỉ mạng gắn với interface muốn tham gia định tuyến được viết dưới dạng classful. Chẳng hạn 1 mạng con 192.168.1.32 chỉ cần viết thành 192.168.1.0, nếu người dùng nhập vào 1 subnet thì Router cũng sẽ tự chuyển về mạng classful tương ứng

Quay lại với ví dụ, chúng ta bắt đầu chạy RIP trên tất cả interface của Router

**Cấu hình**

Trên R1

R1(config)#router rip

R1(config-router)#network 192.168.1.0

R1(config-router)#network 192.168.2.0

Trên R2

R2(config)#router rip

R2(config-router)#network 192.168.2.0

R2(config-router)#network 192.168.3.0

R2(config-router)#network 192.168.4.0

Trên R3

R3(config)#router rip

R3(config-router)#network 192.168.4.0

R3(config-router)#network 192.168.5.0

Để kiểm tra chúng ta có thể sử dụng câu lệnh show running-config

R1#show running-config

!

Router rip

Network 192.168.1.0

Network 192.168.2.0

!

R2#show running-config

!

Router rip

Network 192.168.2.0

Network 192.168.3.0

Network 192.168.4.0

!

R3#show running-config

!

Router rip

Network 192.168.4.0

Network 192.168.5.0

!

**Kiểm tra định tuyến**

Trên mỗi Router, chúng ta kiểm tra bảng định tuyến bằng câu lệnh show ip route

Trên R1

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

R 192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:08, Serial0/0/0

R 192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:08, Serial0/0/0

R 192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:08, Serial0/0/0

C 192.168.1.0/24 is directly connected, FastEthernet0/0

C 192.168.2.0/24 is directly connected, Serial0/0/0

Trên R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

C 192.168.3.0/24 is directly connected, FastEthernet0/0

C 192.168.4.0/24 is directly connected, Serial0/0/1

R 192.168.5.0/24 [120/1] via 192.168.4.1, 00:00:12, Serial0/0/1

R 192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:24, Serial0/0/0

C 192.168.2.0/24 is directly connected, Serial0/0/0

Trên R3

R3#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

R 192.168.1.0/24 [120/2] via 192.168.4.2, 00:00:08, Serial0/0/1

R 192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:08, Serial0/0/1

R 192.168.3.0/24 [120/2] via 192.168.4.2, 00:00:08, Serial0/0/1

C 192.168.4.0/24 is directly connected, Serial0/0/1

C 192.168.5.0/24 is directly connected, FastEthernet0/0

Ta thấy lúc này mạng đã hội tụ, mỗi Router đều đã có đầy đủ tuyến đường tới các mạng khác. Xét trên R1 ta thấy metric tương ứng với tuyến kết nối trực tiếp (bằng 0), các tuyến quảng bá từ R2 có metric bằng 1 (1 hop là R2), và quảng bá từ R3 (mạng 192.168.5.0) có metric bằng 2 (2 hop là R2 và R3). Ngoài ra ta cũng thấy AD bằng 120, Router next-hop là 192.168.2.2 chính là R2. Những thông tin tương tự trên R2 và R3

|  |  |
| --- | --- |
| Show | Mô tả |
| R | Xác định đây là tuyến đường học qua RIP |
| 192.168.5.0 | Địa chỉ IP mạng đích |
| /24 | Mặt nạ mạng của mạng đích |
| [120/2] | Giá trị AD (12) và metric (2) |
| Via 192.168.2.2 | Xác định Router hàng xóm là next-hop hay chính là Router gửi update |
| 00:00:23 | Khoảng thời gian kể từ khi Router nhận update, lần update tiếp theo sẽ trong 7 giây tiếp theo |
| Serial0/0/0 | Outgoing interface để đẩy gói tin tới mạng đích |

Ngoài câu lệnh show ip route để kiểm tra bảng định tuyến, chúng ta cũng có thể sử dụng show ip protocol để xác định cấu hình của RIP

R1#show ip protocol

Routing Protocol is "rip" (1)

Sending updates every 30 seconds, next due in 12 seconds (2)

Invalid after 180 seconds, hold down 180, flushed after 240

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Redistributing: rip

Default version control: send version 1, receive any version (3)

Interface Send Recv Triggered RIP Key-chain (4)

Loopback0 1 2 1

FastEthernet0/0 1 2 1

Loopback1 1 2 1

Automatic network summarization is in effect

Maximum path: 4

Routing for Networks: (5)

192.168.1.0

192.168.2.0

Passive Interface(s):

Routing Information Sources: (6)

Gateway Distance Last Update

192.168.2.2 120 00:00:23

Distance: (default is 120) (7)

Các thông tin cần quan tâm:

* (1) Giao thức định tuyến là RIP
* (2) Các khoảng thời gian Update / Invalid / Holddown / Flush timer
* (3) Phiên bản gửi nhận bản tin RIP
* (4) Số lượng tuyến đường học từ RIP
* (5) Cac mạng được định tuyến
* (6) Các địa chỉ IP nguồn gửi bản tin cập nhật RIP
* (7) AD

Nếu với show ip route và show ip protocol ta vẫn thấy mạng chưa hội tụ, hãy kiểm tra với câu lệnh debug ip rip để thấy quá trình trao đổi các bản tin update.

R1#debug ip rip

RIP protocol debugging is on

RIP: sending v1 request to 255.255.255.255 via Serial0/0/0 (192.168.2.1)

RIP: received v1 request from 192.168.2.2 on Serial0/0/0

RIP: sending v1 update to 192.168.2.2 via Serial0/0/0 (192.168.2.1)

RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0 (192.168.1.1)

RIP: build update entries

network 192.168.1.0 metric 1

network 192.168.2.0 metric 1

RIP: sending v1 update to 255.255.255.255 via FastEthernet (192.168.1.1)

RIP: received v1 update from 192.168.2.2 on Serial0/0/0

192.168.1.0 in 2 hops

192.168.2.0 in 1 hops

192.168.3.0 in 1 hops

192.168.4.0 in 1 hops

Nếu thấy 1 Router gửi ra update nhưng không nhận được update, hãy kiểm tra trạng thái interface của 2 Router với show ip interface brief và cấu hình RIP trên Router đầu bên kia

R1#show ip interface brief

Interface IP-Address OK? Method Status Protocol

FastEthernet0/0 192.168.1.0 YES manual up up

FastEthernet0/1 unassigned YES unset administratively down down

Serial0/0/0 192.168.2.0 YES manual up up

Serial0/0/1 unassigned YES manual administratively down down

**Chú ý:** Mặc dù vẫn thường nói cú pháp network thực hiện quảng bá 1 mạng vào định tuyến, tuy nhiên bản chất câu lệnh này kích hoạt 1 interface tham gia vào định tuyến. Chúng ta suy ra 2 đặc điểm:

* Router sẽ không thể quảng bá 1 mạng mà Router đó không kết nối trực tiếp, nếu địa chỉ IP đằng sau cú pháp network là 1 địa chỉ khác các mạng kết nối trực tiếp của Router thì chẳng có chuyện gì xảy ra cả.
* Chỉ cần địa chỉ IP đằng sau cú pháp network khớp địa chỉ IP của 1 interface thì interface đó xem như đã kích hoạt định tuyến. Như vậy, nếu 1 địa chỉ IP đằng sau cú pháp network khớp nhiều địa chỉ IP của interface thì đồng thời các interface đó sẽ chạy định tuyến

Chẳng hạn Router có 4 interface tương ứng với 4 mạng con 172.17.0.0/24, 172.17.1.0/24, 172.17.2.0/24 và 172.17.3.0/24. Router này chạy RIP và chỉ muốn interface nối tới mạng 172.17.1.0/24 chạy định tuyến còn các interface khác thì không. Chúng ta có cú pháp:

RouterX(config-router)#Network 172.17.1.0

Tuy nhiên thực tế RIPv1 là giao thức Classful do đó nó sẽ tự hiểu mạng 172.17.1.0 chúng ta đưa vào là 172.17.0.0

RouterX#

R2#show ip pro

Routing Protocol is "rip"

Sending updates every 30 seconds, next due in 6 seconds

<output omitted>

Routing for Networks:

172.17.0.0

Passive Interface(s):

<output omitted>

Điều này dẫn đến việc là đồng thời cả 4 interface sẽ bị kích hoạt định tuyến vì 4 mạng của 4 interface đều là mạng con thuộc dải 172.17.0.0/16. Như vậy khi đó Router sẽ gửi các bản tin RIP ra các interface khác không mong muốn, chúng ta nhớ RIPv1 cập nhật bằng bản tin Broadcast, hãy nghĩ xem chuyện gì xảy ra khi bản tin Broadcast được gửi ra trên tất cả interface.

Để giải quyết vấn đề đó, với RIP, chúng ta sẽ phải dùng 1 chức năng gọi là passive-interface

### Cấu hình passive-interface

**Bài toán –** Cấu hình để R1 ngừng gửi bản tin RIP cập nhật định tuyến ra interface fastEthernet 0/0

Trong ví dụ debug ip rip ở trên, ta thấy R1 gửi bản tin cập nhật ra các interface FastEthernet0/0 mặc dù không có thiết bị chạy RIP nào trong miền LAN đó. RIPv1 là giao thức đơn giản, do đó không thể xác định được Router hàng xóm nằm trên interface nào và interface nào không có, do đó, Router chạy RIP sẽ gửi bản tin cập nhật theo chu kỳ 30 giây ra tất cả các interface bật RIP - Ảnh hưởng của quá trình này với hiệu năng mạng:

* Băng thông tiêu tốn lãng phí bởi những bản tin cập nhật không cần thiết. Đặc biệt do bản tin cập nhật của RIP là bản tin quảng bá (Broadcast) nên các thiết bị chuyển mạch như Switch hoặc Hub sẽ gửi bản tin update ra tất cả các cổng.
* Việc xử lý bản tin cập nhật không đơn giản, các thiết bị trong mạng LAN sẽ phải xử lý gói tin đến tận tầng chuyển vận, tuy nhiên sau đó gói tin sẽ bị loại bỏ. Điêu này ảnh hưởng hiệu năng của các thiết bị trong mạng LAN
* Vấn đề thứ 3 đó là vấn đề bảo mật, vì bản tin được gửi ra dưới dạng quảng bá, do đó bất kỳ thiết bị nào trong mạng LAN cũng đều có thể bắt được gói tin. Gói tin cập nhật định tuyến sau đó có thể bị thay đổi rồi gửi lại để phá hoại hạ tầng định tuyến trong mạng, hoặc bị sử dụng để theo dõi hạ tầng định tuyến của 1 mạng

Để giải quyết các bài toán đó, người ta đưa ra tính năng passive interface

**Cú pháp**

Để RIP không gửi thông tin cập nhật định tuyến ra 1 interface, chúng ta có 2 cách. Giả sử R1 không muốn gửi bản tin RIP ra interface fastEthernet 0/0, cách thứ nhất đó là sử dụng câu lệnh no network 192.168.1.0 tuy nhiên cách này gián tiếp làm cho RIP không quảng bá thông tin về mạng LAN 192.168.1.0 ra các interface khác. Như vậy gián tiếp chúng ta đã làm gián đoạn định tuyến

Cách thứ 2 là sử dụng tính năng passive interface, cấu hình trong sub-mode router rip, cú pháp hoàn ceu chỉnh:

Router(config-router)#passive-interface {loại interface} {sổ thứ tự cổng}

Trong đó, {loại interface} {số thứ tự cổng} ví dụ fastEthernet 0/0, hoặc Serial0/0/0

Câu lệnh này sẽ buộc Router ngừng gửi bản tin cập nhật định tuyến ra interface cụ thể, nhưng mạng kết nối trực tiếp gắn với interface đó vẫn được quảng bá ra những interface khác

**Triển khai**

Bước đầu tiên chúng ta đi vào sub-mode Router RIP, sau đó cấu hình tính năng passive-interface trên cổng fastEthernet 0/0

R1#configure terminal

R1(config)#router rip

R1(config)#passive-interface fastEthernet 0/0

**Kiểm tra**

Chúng ta kiểm tra trạng thái 1 interface được cấu hình passive-interface, thông qua câu lệnh show ip protocol

R1#show ip protocol

Routing Protocol is "rip"

Sending updates every 30 seconds, next due in 12 seconds

Invalid after 180 seconds, hold down 180, flushed after 240

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Redistributing: rip

Default version control: send version 1, receive any version

Interface Send Recv Triggered RIP Key-chain

Loopback0 1 2 1

FastEthernet0/0 1 2 1

Loopback1 1 2 1

Automatic network summarization is in effect

Maximum path: 4

Routing for Networks:

192.168.1.0

192.168.2.0

Passive Interface(s):

FastEthernet 0/0

Routing Information Sources:

Gateway Distance Last Update

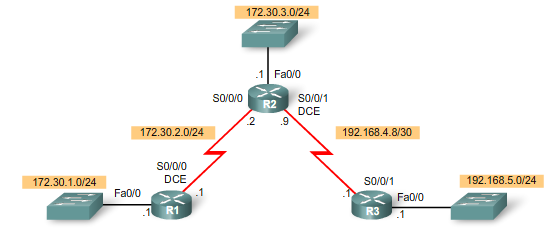
192.168.2.2 120 00:00:23

Distance: (default is 120)

### Quá trình auto-summary

RIP là giao thức định tuyến classful, do đó RIP sẽ tự động gộp (summary) các mạng classful khi thông tin định tuyến đi qua giữa 2 miền mạng major. Trong ví dụ tiếp theo, chúng ta sẽ có kịch bản Router R2 có các interface thuộc 2 miền mạng major khác nhau, điều này khiến R2 trở thành Router biên giữa 2 miền mạng 172.16.0.0 (lớp B) và miền mạng 192.168.4.0 (lớp C) với interface Serial0/0/0 và fastEthernet 0/0 thuộc miền 172.16.0.0 còn interface Serial0/0/1 thuộc miền mạng 192.168.4.0. Trong ví dụ tiếp theo, giao thức RIP trên Router R2 sẽ tự động summary các mạng con của miền 172.16.0.0 thành mạng major rồi mới đẩy sang Router hàng xóm bên miền 192.168.4.0.

Để kiểm tra tính năng auto-summary trên giao thức RIP, chúng ta sửa lại cấu trúc địa chỉ IP của topology như sau:



**Cấu trúc địa chỉ IP**

Chúng ta sử dụng 3 dải mạng Classful sau

* 172.30.0.0/16
* 192.168.4.0/24
* 192.168.5.0/24

Trong đo mạng 172.30.0.0/16 được chia thành 3 mạng con /24

* 172.30.1.0/24 trên interface f0/0 của Router R1
* 172.30.2.0/24 trên Serial link giữa Router R1-R2
* 172.30.3.0/24 trên interface f0/0 của Router R2

Mạng classful 192.168.4.0/24 được chia thành mạng con duy nhât 192.168.4.8/30 cho Serial link giữa R2-R3

Mạng 192.168.5.0/24 được sử dụng cho mạng kết nối trực tiếp trên interface f0/0 của Router R3

**Cấu hình khởi tạo**

Chúng ta cấu hình tương tự 2 bài trước. Triển khai đưa các mạng vào RIP để quảng bá, và triển khai tính năng passive-interface trên các interface LAN của Router R1, R2 va R3. Cấu hình khởi tạo như sau (bao gồm địa chỉ IP)

Trên R1

R1(config)#interface serial0/0/0

R1(config-if)#ip address 172.30.2.1 255.255.255.0

R1(config)#interface fastethernet0/0

R1(config-if)#ip address 172.30.1.1 255.255.255.0

!

R1(config)#router rip

R1(config-router)#network 172.30.1.0

R1(config-router)#network 172.30.2.0

R1(config-router)#passive-interface f0/0

R1(config-router)#end

Trên R2

R2(config)#interface serial0/0/0

R2(config-if)#ip address 172.30.2.2 255.255.255.0

R2(config)#interface fastethernet0/0

R2(config-if)#ip address 172.30.3.2 255.255.255.0

R2(config)#interface serial0/0/1

R2(config-if)#ip address 192.168.4.9 255.255.255.252

!

R2(config)#router rip

R2(config-router)#network 172.30.0.0

R2(config-router)#network 192.168.4.8

R2(config-router)#passive-interface f0/0

R2(config-router)#end

Trên R3

R3(config)#interface serial0/0/1

R3(config-if)#ip address 192.168.4.10 255.255.255.252

R3(config)#interface fastethernet0/0

R3(config-if)#ip address 192.168.5.1 255.255.255.0

!

R3(config)#router rip

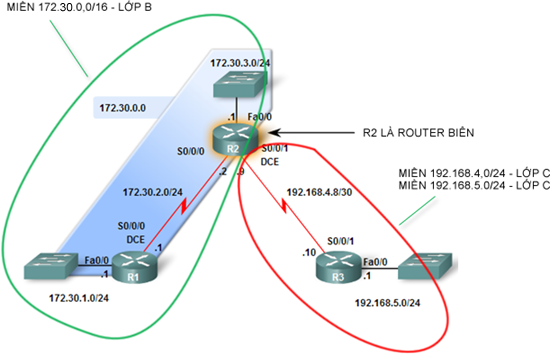
R3(config-router)#network 192.168.5.0

R3(config-router)#network 192.168.4.0

R3(config-router)#passive-interface f0/0

R3(config-router)#end

**Tiến trình Auto-Summary**



*Hình 5.6 – Các miền mạng major, Router R2 là Router biên giữa 2 miền*

Các nguyên tắc trao đổi thông tin cập nhật lớp mạng trên RIPv1

* Nếu mạng đích trong 1 bản tin cập nhật định tuyến và outgoing interface thuộc cùng 1 mạng major, mặt nạ mạng của interface sẽ được sử dụng để gán cho mạng đích trong bản tin cập nhật
* Nếu mạng đích trong 1 bản tin cập nhật định tuyến và outgoing interface được gửi ra không thuộc cùng 1 mạng major, mặt nạ mạng *gốc* (mặt nạ của mạng classful tương ứng) của mạng đích sẽ được dùng để gán cho mạng đích trong bản tin cập nhật đó

Theo đó, chúng ta kiểm tra bảng định tuyến trên các Router, R2:

R2(config-router)#do show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

172.30.0.0/24 is subnetted, 3 subnets

R 172.30.1.0 [120/1] via 172.30.2.1, 00:00:17, Serial0/0/0

C 172.30.2.0 is directly connected, Serial0/0/0

C 172.30.3.0 is directly connected, fastEthernet0/0

192.168.4.0/30 is subnetted, 1 subnets

C 192.168.4.8 is directly connected, Serial0/0/1

R 192.168.5.0/24 [120/1] via 192.168.4.10, 00:00:22, Serial0/0/1

Chúng ta thấy, Router R2 nhận bản tin update từ Router R1 và cập nhật mạng 172.30.1.0/24 vào bảng định tuyến, câu hỏi là: vì sao Router R2 xác định được subnet mask của mạng 172.30.1.0 là 255.255.255.0 (~ /24) – Câu trả lời chi tiết cho vấn đề này sẽ được đề cập trong [chương 6 – VLSM và CIDR](#_CHƯƠNG_6_-). R2 sẽ áp dụng subnet mask của interface Serial0/0/0 cho mạng 172.30.1.0, do mạng 172.30.1.0 và mạng 172.30.2.0 trên interface Serial0/0/0 đều thuộc cùng 1 mạng major (172.16.0.0/16). Như vậy mạng 172.16.1.0 được cập nhật vào bảng định tuyến 1 cách chính xác. Tương tự với mạng lớp C: 192.168.5.0/24, Router R2 là Router biên do đó nó sẽ có thông tin cập nhật định tuyến chính xác nhất

Chúng ta sử dụng debug ip rip để kiểm tra việc trao đổi cập nhật trên Router R2 - Khi gửi 1 bản tin update, Router biên R2 sẽ gửi địa chỉ mạng cùng với metric đi kèm, không có subnet mask.

R2(config-router)#RIP: sending v1 update to 255.255.255.255 via Serial0/0/0 (172.30.2.2)

RIP: build update entries

network 172.30.3.0 metric 1

network 192.168.4.0 metric 1

network 192.168.5.0 metric 2

RIP: sending v1 update to 255.255.255.255 via fastEthernet0/0 (172.30.3.2)

RIP: build update entries

network 172.30.1.0 metric 2

network 172.30.2.0 metric 1

network 192.168.4.0 metric 1

network 192.168.5.0 metric 2

RIP: sending v1 update to 255.255.255.255 via Serial0/0/1 (192.168.4.9)

network 172.30.0.0

Router R1 cần nhận được thông tin tuyến đường từ các mạng 172.30.3.0, 192.168.4.8 và mạng 192.168.5.0. Chúng ta hãy xem R1 cập nhật định tuyến như thế nào:

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

172.30.0.0/24 is subnetted, 3 subnets

C 172.30.1.0 is directly connected, Loopback0

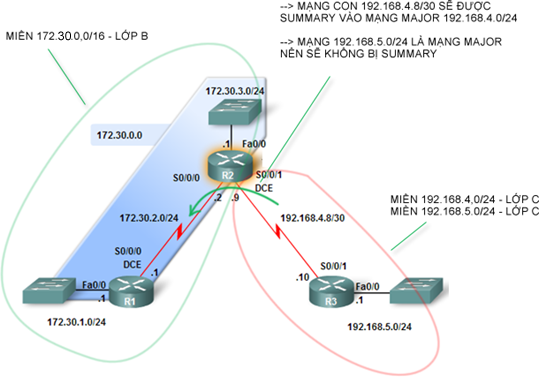
C 172.30.2.0 is directly connected, FastEthernet0/0

R 172.30.3.0 [120/1] via 172.30.2.2, 00:00:24, FastEthernet0/0

R 192.168.4.0/24 [120/1] via 172.30.2.2, 00:00:24, FastEthernet0/0

R 192.168.5.0/24 [120/2] via 172.30.2.2, 00:00:24, FastEthernet0/0

Ta thấy Router R1 nhận được cập nhật về mạng 172.30.3.0 từ interface Serial0/0/0, nhận thấy mạng đích và mạng trên interface thuộc cùng 1 mạng major (172.16.0.0), nên Router R1 áp dụng subnet mask /24 của interface Serial0/0/0 cho mạng 172.30.3.0. Tuy nhiên chúng ta thấy mạng 192.168.4.8 đã trở thành mạng 192.168.4.0/24. Khi bản tin update được gửi qua giữa 2 miền major, các mạng con sẽ được *tự động* gộp tuyến (summary) vào mạng major tương ứng. Trong trường hợp này mạng 192.168.4.8/30 là mạng con bị summary vè 192.168.4.0/24, còn mạng 192.168.5.0/24 là mạng major nên không có vấn đề



*Hình 5.7 – Auto-summary diễn ra khi R2 quảng bá sang R1*

Tương tự đối với R3, Router R2 có các tuyến đường tới mạng con 172.30.1.0/24, 172.30.2.0/24 và 172.30.3.0/24 cần được gửi. Chúng ta kiểm tra bảng định tuyến trên R3

R3#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

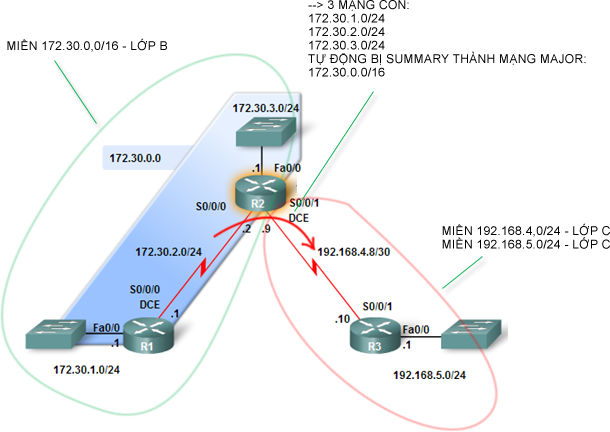
R 172.30.0.0/16 [120/1] via 192.168.4.9, 00:00:04, FastEthernet0/0

192.168.4.0/30 is subnetted, 1 subnets

C 192.168.4.8 is directly connected, FastEthernet0/0

C 192.168.5.0/24 is directly connected, Loopback0

Tuy nhiên trong bản tin định tuyến của R3 chỉ có duy nhất 1 tuyến tới mạng major lớp B: 172.30.0.0/16, các mạng con 172.30.1.0, 172.30.2.0 và 172.30.30 đã được summary vào mạng major. Quay lại phần debug phía trên của R2, ta thấy trong bản tin cập nhật gửi bởi R2, chỉ có duy nhất mạng major 172.30.0.0 được gửi đi ra interface Serial0/0/1



*Hình 5.8 – Auto-summary diễn ra khi R2 gửi quảng bá sang R3*

Vậy là R2 thực hiện tự động (auto) gộp tuyến (summary) khi gửi bản tin cập nhật giữa 2 mạng major. Như vậy:

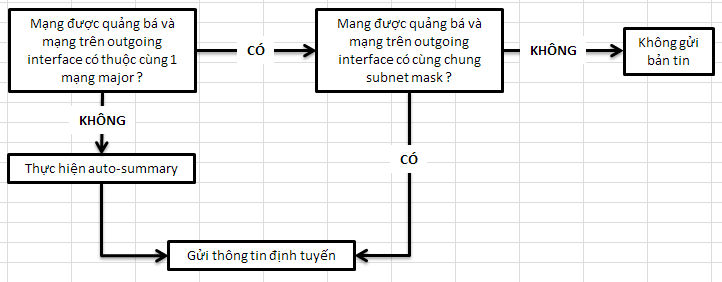
Auto-summary là quá trình trên Router biên giữa các miền mạng major tự động thực hiện gộp các mạng con khi quảng bá tư 1 mạng major này sang 1 mạng major khác

Cơ chế Auto-summary là cơ chế mặc định chạy trên RIPv1

**Câu hỏi:** Vậy liệu R3 có thể kết nối tới tất cả các mạng con thuộc 172.30.0.0/16 không ?

Chúng ta dễ dàng thấy trong bảng định tuyến rằng R3 sẽ vẫn có thể kết nối tới tất cấ các mạng con thuộc 172.30.0.0, vì để tới mạng con 172.30.1.0/24 hay 172.30.2.0/24 hay 172.30.3.0/24 thì R3 vẫn đẩy gói tin ra interface fastEthernet 0/0 tới router R2. **Như vậy chắc chắn Router R3 vẫn kết nối tới các mạng con được**

Tổng kết lại, ta có sơ đồ mô tả điều kiện gửi cập nhật của RIPv1:



*Hình 5.9 – điều kiện gửi bản tin cập nhật của RIPv1*

**Ưu điểm của cơ chế Auto-Summarization**

Lợi ích của cơ chế Auto-summaization chúng ta đã thấy khi triển khai định tuyến tĩnh ứng dụng gộp tuyến. Đối với Router R3, thay vì lưu 3 tuyến đường tới 3 mạng con, R3 chỉ lưu 1 tuyến đường duy nhất. Điều này giúp tiết kiệm không gian bảng định tuyến. Tương tự trên Router R2, thay vì gửi cập nhật mang 3 tuyến đường, R2 chỉ gửi đi cập nhật cho 1 tuyến đường duy nhất.

Lợi ích thứ 2 của Auto-summarization, đó là cho dù có bao nhiêu mạng con, phân cấp địa chỉ như thế nào chẳng nữa, miễn thuộc 1 mạng major thì sẽ đều chỉ xuất hiện dưới 1 mạng major duy nhất. Theo đó quá trình tra cứu tuyến đường trong bảng định tuyến cũng diễn ra nhanh hơn

**Nhược điểm của cơ chế Auto-Summarization**

Ngoài những ưu điểm “nhỏ” trên, Auto-summarization lại đặt ra các vấn đề to đùng liên quan đến triển khai hạ tầng địa chỉ IP. Chúng ta dễ thấy vấn đề là :

**Vấn đề 1 -** Trong 1 miền mạng major, các mạng con chỉ có 1 subnet mask duy nhất, hay, không hỗ trợ VLSM, chúng ta đã bàn vấn đề này tệ thế nào trong kỳ 1

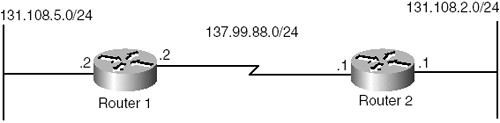
**Vấn đề 2 –** Vấn đề “hố đen” khi chỉ có 1 số ít mạng con bị summary – Hãy xét trường hợp trên, Router R3 chỉ có 1 tuyến đường 172.30.0.0/16, và nó đại diện cho 3 mạng con của miền bên kia R2, tuy nhiên mạng 172.30.0.0/16 còn đại diện cho 252 mạng con /24 khác nữa. Chuyện gì sẽ xảy ra các thiết bị trong LAN của R3 cứ tiếp tục gửi bản tin tới 1 mạng con không có thực ở phía bên kia, chẳng hạn 172.30.4.1 ?

**Vấn đề 3** – Nghiệm trọng nhất, là không thể triển khai mạng không liên tục / mạng gián đoạn (discontigous network)

### Vấn đề mạng không liên tục – Discontigous network

Mạng không liên tục là hiện tượng 2 mạng con thuộc cùng 1 mạng major bị ngăn cách bởi 1 mạng major khác

**Ví dụ 1**



*Hình 5.10 – Mạng không liên tục*

Cách thức cập nhật định tuyến của RIPv1 biến mạng không liên tục trở thành 1 vấn đề.

Hình 5.10 là 1 ví dụ về mạng không liên tục, miền mạng major 131.108.0.0/16 có 2 mạng con 131.108.5.0/24 và 131.108.2.0/24 là 2 mạng LAN trên Router 1 và Router 2, hai mạng con này bị ngăn cách bởi 1 miền mạng major khác 137.99.0.0/16. Giả sử Router 2 và Router 1 chạy RIPv1, và 2 Router quảng bá tất cả tuyến đường mà chúng biết cho nhau, hãy xem chuyện gì xảy ra, chúng ta kiểm tra bảng định tuyến trước

Trên Router 1

R1# show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

131.108.0.0/24 is subnetted, 1 subnets

C 131.108.5.0 is directly connected, fastEthernet0/0

137.99.0.0/24 is subnetted, 1 subnets

C 137.99.88.0 is directly connected, Serial0/0/0

Và trên Router 2

R2# show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

131.108.0.0/24 is subnetted, 1 subnets

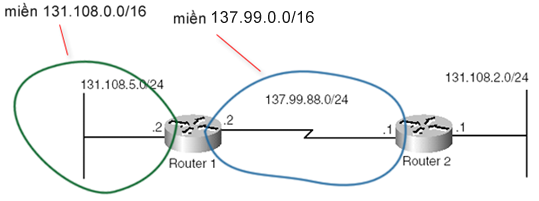
C 131.108.5.0 is directly connected, fastEthernet0/0

137.99.0.0/24 is subnetted, 1 subnets

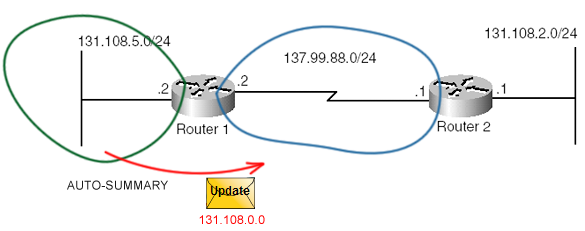
C 137.99.88.0 is directly connected, Serial0/0/0

Ta thấy 2 Router không cập nhật được thông tin định tuyến tới mạng xa, chúng ta theo dõi các tiến trình

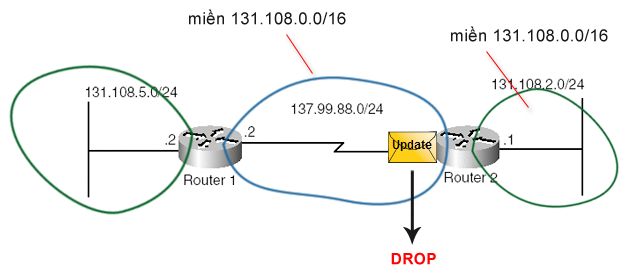
1. Router 1 gửi quảng bá mạng 131.108.5.0/24 sang Router 2, và thấy rằng mình là Router biên giữa 2 mạng major là 131.108.0.0/16 và 137.99.0.0/16



1. Router 1 thực hiện auto-summary biến mạng con 131.108.5.0/24 thành 131.108.0.0 rồi ném sang R2

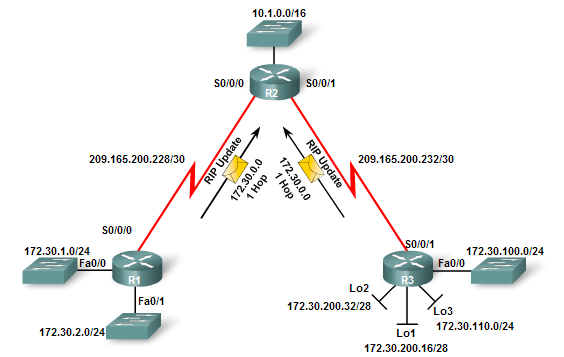


1. Router 2 nhận cập nhật về mạng đích 131.108.0.0, nhưng nhận ra rằng mình là Router biên giữa 2 mạng major 131.108.0.0/16 và 137.99.0.0/16
2. Router 2 không chấp nhận bản tin cập nhật về mạng major mà nó đang có kết nối trực tiếp.

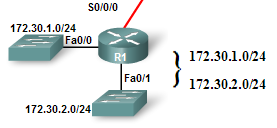
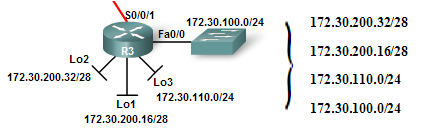


1. Quá trình ngược lại diễn ra khi Router 2 gửi cập nhật sang cho Router 1

**Ví dụ 2,** cho topology như sau:



Phía trên là 1 ví dụ nữa về mạng không liên tục, mạng major 172.30.0.0/16 gồm các mạng con:

và 

2 miền mạng con này bị chia bởi miền mạng major 209.165.200.0/24. Chúng ta xem cách các Router R1, R2, R3 trao đổi thông tin cập nhật:

1. R1 xem xét gửi thông tin về mạng 172.30.1.0 và mạng 172.30.2.0. Do R1 là Router biên giữa mạng major 172.30.0.0 và mạng 209.165.200.0, do đó R1 sẽ thực hiện auto-summary và gửi ra update về mạng major 172.30.0.0
2. R2 nhận thông tin về mạng 172.30.0.0, do đó Router R2 sẽ cập nhật vào bảng định tuyến thông tin về mạng 172.30.0.0 với metric bằng 1

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/16 is subnetted, 1 subnets

C 10.1.0.0 is directly connected, FastEthernet0/0

R 172.30.0.0/16 [120/1] via 209.165.200.234, 00:00:23, Serial0/0/0

209.165.200.0/30 is subnetted, 2 subnets

C 209.165.200.228 is directly connected, Serial0/0/0

C 209.165.200.232 is directly connected, Serial0/0/1

1. R3 tương tự, là Router biên giữa 2 miền 172.30.0.0 và miền 209.165.200.0, do đó R3 sẽ thực hiện auto-summary và gửi cập nhật về mạng major 172.30.0.0
2. R2 lại nhận được tuyến đường tới mạng 172.30.0.0 và cập nhật vào bảng định tuyến với metric bằng 1. Như vậy Router R2 nhận 2 cập nhật về mạng 172.30.0.0 tới từ 2 nguồn khác nhau (Router R1 và Router R3) với 2 metric bằng nhau. R2 sẽ thực hiện cân bằng tải

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/16 is subnetted, 1 subnets

C 10.1.0.0 is directly connected, FastEthernet0/0

R 172.30.0.0/16 [120/1] via 209.165.200.234, 00:00:23, Serial0/0/1

[120/1] via 209.165.200.230, 00:00:21, Serial0/0/0

209.165.200.0/30 is subnetted, 2 subnets

<output omitted>

Tới đây ta thấy mạng “hội tụ”, vì bảng định tuyến ổn định, không có sự thay đổi trạng thái tuyến đường. Chúng ta thử kiểm tra kết nối từ R2 tới 1 trong 2 miền mạng giữa R1 và R3, chẳng hạn R2 gửi gói tin tới mạng con 172.30.2.0/24 của R1

R2#ping 172.30.2.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.30.2.1, timeout is 2 seconds:

U!.!U

Success rate is 40 percent (2/5), round-trip min/avg/max = 1/6/4 ms

Chúng ta thấy cứ 1 gói tin ping thành công sẽ lại có 1 gói unreachable [U]

Chúng ta kiểm tra kết nối từ R2 tới 1 mạng con phía R3, chẳng hạn mạng 172.30.200.16/28

R2#ping 172.30.200.17

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 172.30.200.17, timeout is 2 seconds:

U!.!U

Success rate is 40 percent (2/5), round-trip min/avg/max = 1/7/5 ms



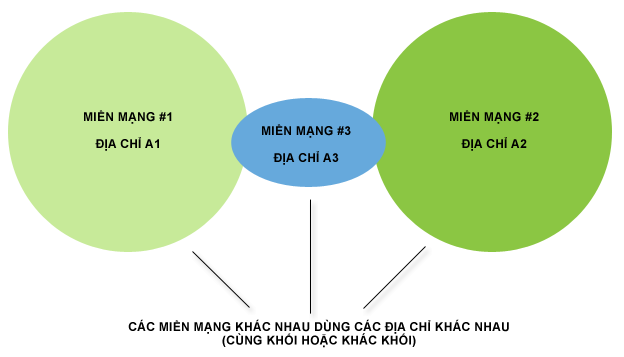
Tình trạng tương tự xảy ra, chúng ta tự hỏi chuyện gì đang diễn ra với định tuyến. Khi Router R2 có 1 gói tin muốn gửi tới mạng 172.30.2.0/24, R2 sẽ tra bảng định tuyến, và tuyến đường được sử dụng là 172.30.0.0/16. Tuy nhiên tuyến đường này cân bằng tải trên 2 interface serial0/0/0 và Serial0/0/1, do đó cứ 1 gói tin được gửi ra R1, sẽ có 1 gói tin gửi ra R3.

Khi cân bằng tải diễn ra, các gói tin tới Router R3 sẽ lần lượt bị loại bỏ (drop) vì Router R3 không hề biết đường tới mạng con 172.30.2.0/24 (do mạng con này không thể được quảng bá tới Router R3). Lý giải tương tự khi R2 muốn gửi dữ liệu tới các mạng con phía R3, lúc đó R2 sẽ cân bằng tải với Router R1, và R1 sẽ drop bất kỳ gói tin nào nó nhận được.

2 ví dụ trên cho ta thấy rằng, không thể triển khai mạng không liên tục với giao thức Classful routing như RIPv1

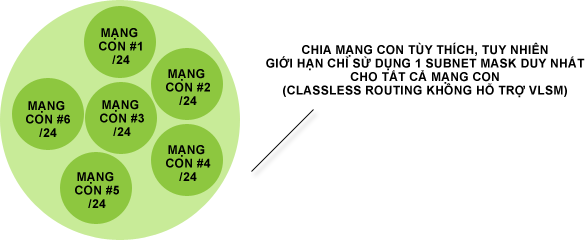
**Thiết kế địa chỉ IP với RIPv1**

RIPv1 ra đời trong giai đoạn của classful addressing, tại thời điểm đó, 1 doanh nghiệp hay 1 tổ chức được cấp 1 địa chỉ thuộc 1 trong 3 dải A, B, C. Tại thời điểm 1 tổ chức sở hữu 1 địa chỉ A1 chẳng hạn, thì địa chỉ đó sẽ không được sử dụng 1 tổ chức khác nữa. Tư duy thiết kế tại thời điểm đó là 1 đia chỉ classful không được sử dụng cho 2 miền mạng khác nhau, do đó với RIPv1 sẽ không bao giờ gặp tình trạng mạng gián đoạn



*Hình 5.11*

VLSM chưa xuất hiện tại thời điểm này, do đó việc chia mạng con chỉ được thực hiện 1 lần, hay nói cách khác, 1 mạng con không thể được chia nhỏ 1 lần nữa. Vậy thì trong mạng có thể có bao nhiêu mạng con tùy ý, nhưng tất cả mạng con đều phải có chung subnet mask. RIPv1 chưa có khái niệm về VLSM, do đó với thiết kế địa chỉ kiểu truyền thống trên là hoàn toàn phù hợp. Do đó bài toán chia địa chỉ kiểu “vét cạn” triển khai VLSM là “khắc tinh” của giao thức này.



*Hình 5.12*

Vậy thì ta thấy vấn đề khó nhất khi triển khai RIP không nằm ở việc cấu hình định tuyến, mà chính là nằm ở khâu quy hoạch địa chỉ IP

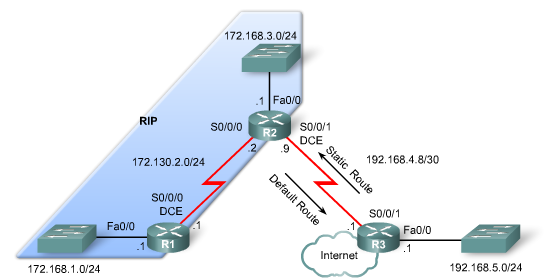
### Cấu hình default route

ĐỊnh tuyến tĩnh cho phép chúng ta cấu hình tuyến tĩnh với phần địa chỉ có thể là:

* 1 mạng con
* 1 mạng gộp
* Default route

Tương tư, hầu hết giao thức định tuyến cho phép chúng ta quảng bá ra 1 trong 3 loại mạng trên. RIPv1 không phải là ngoại lệ, chúng ta đã xem xét cách RIPv1 quảng bá ra 1 mạng con với câu lệnh network và 1 mạng gộp thông qua cơ chế auto-summary. Nội dung cuối cùng trong phần cấu hình sẽ là chức năng quảng bá ra 1 tuyến đường default route

**Topology**



**Yêu cầu –** Doanh nghiệp thuê nhà cung cấp dịch vụ 1 đường kết nối trực tiếp ra Internet. Cấu hình sao cho tất cả thiết bị trong mạng đều có thể ra Internet

**Phân tích**

Trong mô hình chúng ta thấy sự xuất hiện của 2 Router trong miền RIP là Router R1 và R2, trong ví dụ này chúng ta cần triển khai tuyến mặc định trên tất cả Router trong mạng để trỏ gói tin chưa rõ mạng đích ra Router R3. Để thực hiện yêu cầu trên, chúng ta có thể triển khai câu lệnh cấu hình tuyến tĩnh (ứng dụng default route) trên tất cả Router, như vậy chúng ta cần 2 câu lệnh định tuyến tĩnh:

R1(config)#ip route 0.0.0.0 0.0.0.0 serial0/0/0

R2(config)#ip route 0.0.0.0 0.0.0.0 serial0/0/1

Trong phần này, chúng ta sẽ sử dụng 1 cách khác đó là cấu hình sao cho Router biên, R2, sẽ quảng bá tuyến mặc định vào cho các Router trong miền RIP

**Cấu hình khởi tạo**

Chúng ta triển khai định tuyến động giữa R2 và R3, trên Router R3 chúng ta cấu hình tuyến đường trở về mạng 172.30.0.0/16

Trên Router R1

R1(config)#router rip

R1(config-router)#net 172.30.0.0

Trên Router R2

R2(config)#router rip

R2(config-router)#net 172.30.0.0

Trên Router R3

R3(config)#ip route 172.30.0.0 255.255.0.0 serial0/0/1

**Cú pháp**

RouterX(config)#router rip

RouterX(config-router)#default-information originate

**Triển khai**

Yêu cầu duy nhất khi triển khai quảng bá default route đó là Router quảng bá (Router biên) phải có sẵn tuyến default route trong bảng định tuyến. Do đó, trước khi cấu hình quảng bá default route, chúng ta cấu hình tuyến mặc định trên Router R2 trỏ ra Router R3

R2(config)#ip route 0.0.0.0 0.0.0.0 serial0/0/1

Chúng ta kiểm tra bảng định tuyến trên Router R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

C 192.168.4.8 is directly connected, Serial0/0/1

S\* 0.0.0.0/0 is directly connected, Serial0/0/1

Ta thấy đa xuât hiện tuyến tĩnh mặc định trong bảng định tuyến, bây giờ chúng ta sẽ quảng bá tuyến đường mặc định này ra cac Router chạy RIP trong mạng:

R2(config)#router rip

R2(config-router)#default-information originate

Triển khai rât đơn giản, chúng ta sẽ kiểm tra trạng thái định tuyến trên Router R1

**Xác nhận**

Chúng ta bắt đầu với bảng định tuyến trên Router R1

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

R 172.30.3.0 [120/1] via 172.30.2.2, 00:00:09, Serial0/0/0

R\* 0.0.0.0/0 [120/1] via 172.30.2.2, 00:00:09, Serial0/0/0

Ta thấy Router R1 cập nhật tuyến đường mặc định từ Router R2 (via 172.30.2.2), tuyến đường mặc định (dấu \*) được học từ RIP (R) sử dụng outgoing interface là Serial 0/0/0

Chúng ta kiểm tra kết nối từ mạng LAN của R1 tới mạng LAN của R3. Chúng ta chú ý trong trường hợp này, bảng định tuyến của R1 không hề có tuyến đường tới mạng 192.168.5.0/24

R1#ping 192.168.5.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.5.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/7/10 ms

Ta thấy kết nối thành công

### Tổng kết

Trong chương này, chúng ta đã có cái nhìn khái quát về:

* Các đặc điểm định tuyến của giao thức RIP
* Các thông số thời gian của RIP
* Vận hành của RIPv1
* Cách thức cấu hình quảng bá, và default route
* Cơ chế auto-summary của RIP

# ****CHƯƠNG 6 - GIAO THỨC ĐỊNH TUYẾN KHÔNG PHÂN LỚP. VLSM VÀ CIDR.****

Mạng ARPANET, tiền thân của Internet, được triển khai trong 1 trường đại học với cấu trúc mạng rất nhỏ tại thời điểm đó, vì dưới dạng nghiên cứu vẫn chưa có các ứng dụng mạng hỗ trợ người dùng. Vào những năm 1981, địa chỉ IP chỉ được sử dụng 8 bit của octet đầu để xác định phần mạng, cho phép 256 mạng. Gần như rất nhanh ngay sau đó, người ta nhận ra con số đó là không đủ. RFC 791 đã đưa vào khái niệm lớp địa chỉ (Classful addressing) với 3 lớp A, B, C tương ứng 3 lớp mặt nạ mạng cho dải địa chỉ 32 bit. Chúng ta đã đề cập tới khái niệm Classful addressing trong bài trước, với địa chỉ lớp A sử dụng 8 bit mạng, địa chỉ lớp B sử dụng 16 bit mạng và 1 địa chỉ lớp C có 24 bit mạng.

Classful addressing ra đời để giải quyết bài toán thiếu tài nguyên địa chỉ IP, nhưng không thể giải quyết triệt để, vì không ai lường trước được tốc độ tăng chóng mặt nhu cầu về mạng và sự bùng nổ của Internet, công việc đánh địa chỉ không chỉ phụ thuộc vào tài nguyên địa chỉ dồi dào hơn, mà còn là bài toán quản lý, người ta nhận ra rằng nếu không có một kỹ thuật quản lý và phân cấp địa chỉ tối ưu hơn thì 256 địa chỉ mạng lớp A, 16,000 địa chỉ mạng lớp B và 2,000,000 địa chỉ mạng lớp C cũng chỉ như muối bỏ bể. Ngay lâp tức, IETF đưa ra CIDR là kỹ thuật Classless routing sử dụng cùng với VLSM là kỹ thuật Classless addressing để giải quyết triệt để bài toán không gian địa chỉ

Kiến thức về VLSM và CIDR là kiến thức nền tảng của bất kỳ 1 quản trị viên nào, vì nếu không có anh sẽ không thể quy hoạch địa chỉ IP trong mạng 1 cách hiệu quả được. Với VLSM, các mạng được chia nhỏ thành các mạng con tùy thuộc vào chính sách hoặc quy mô của mỗi mạng con. Với CIDR, các mạng con có thể được định tuyến, đảm kết nối giữa các mạng không liên tục (discontigous network – đề cập trong phần sau) đồng thời tối ưu bảng định tuyến với kỹ thuật gộp tuyến đường (Aggregation – đề cập trong phần sau)

## Classful vs Classless

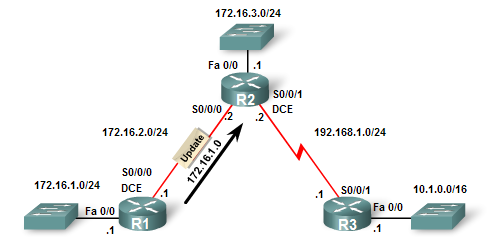
### Classful và Classless Addressing

Classful hay Classless Addressing là 2 cách tiếp cận trong vấn đề quy hoạch địa chỉ IP, với Classful Addressing, tiền tố mạng được xác định bởi các lớp địa chỉ, lớp của 1 địa chỉ được xác định dựa vào octet đầu tiên của địa chỉ IP đó. *Như vậy với Classful Addressing, 1 địa chỉ sẽ không cần subnet mask đi kèm*, chẳng hạn 1 địa chỉ lớp A như 11.1.1.1 chắc chắn sẽ có subnet mask bằng 255.0.0.0 vì đó là địa chỉ lớp A. Ngược lại với Classless Addressing, dựa trên VLSM cho phép 1 địa chỉ có thể có nhiều subnet mask, chẳng hạn 11.1.1.1 có thể có subnet mask tương ứng /8 hoặc /16 hoặc /24 hay bất kỳ giá trị nào tùy thuộc vào cách chia của người quản trị. Như vậy, với Classless Addressing, 1 địa chỉ IP buộc phải đi kèm với subnet mask để xác định mạng của địa chỉ IP đó.

### Classful Routing

Tương ứng với Classful Addressing thì trong định tuyến, chúng ta có khái niệm Classful Routing. Tương tự Classful Addressing, thì Classful Routing sẽ không gửi subnet mask đi kèm địa chỉ mạng trong bản tin cập nhật định tuyến. Do đó, Classful Routing hoạt động tốt nhất với Classful Addressing, hay nói cách khác, Classful Routing không hỗ trợ VLSM. Tuy nhiên, không có nghĩa rằng không thể triển khai Classful Routing với Classless Addressing.

Các đại diện của Classful Routing là RIPv1 và IGRP. Ta xét 1 ví dụ sau:



Cho topology:

* R1 kết nối với 2 mạng 172.16.1.0/24 và 172.16.2.0/24
* R2 kết nối trực tiếp với 3 mạng 172.16.2.0/24, 172.16.3.0/24, 192.168.1.0/24
* R3 kết nối trực tiếp tới 2 mạng 192.168.1.0 và 10.1.0.0/16
* R1, R2, và R3 chạy Classful Routing, trong trường hợp này là RIPv1

Yêu cầu:

* Định tuyến sao cho host từ 172.16.1.0/24, 172.16.3.0/24 và 10.1.0.0/16 có thể giao tiếp với nhau

Phân tích:

**[?]Câu hỏi thứ nhất**: *Khi R1 quảng bá các mạng 172.16.1.0/24 và 172.16.2.0/24 vào RIP, do đặc điểm RIPv1 nên bản tin quảng bá sẽ chỉ mang theo địa chỉ IP chứ không có subnet mask. Khi R2 nhận được bản tin update, R2 sẽ cố gắng cập nhật tuyến đường để định tuyến, nhưng R2 không biết subnet mask của 172.168.1.0. Vậy làm sao R2 xác định được subnet mask cho mạng 172.168.1.0 để cập nhật vào bảng định tuyến ?*

Vì R2 không xác định được subnet mask của mạng 172.168.1.0 được gửi từ R1, nên R2 sẽ dựa vào subnet mask của địa chỉ IP được gán cho interface nối tới cổng nhận được bản tin update. Trong trường hợp này là subnet mask của địa chỉ IP trên interface s0/0/0 của R2, và bằng 255.255.255.0 (~ /24). Vậy R2 sẽ cập nhật mạng 172.16.1.0/24 vào bảng định tuyến ! Chúng ta có hành vi thứ nhất: **1 Router chạy Classful Routing sẽ gán subnet mask của địa chỉ IP trên interface nhận bản tin update cho địa chỉ mạng đích nhận từ bản tin update**

Vấn đề xảy ra là, nếu thực sự mạng 172.16.1.0 trên R1 không phải có subnet mask bằng 255.255.255.0 mà có thể bằng 255.255.0.0 hoặc 255.255.255.128 hoặc 1 giá trị khác. Như vậy đồng nghĩa R2 đã cập nhật sai thông tin định tuyến. Vậy hành vi thứ 2 của Classful Routing đó là: **1 Router chỉ gửi quảng bá về 1 subnet khi subnet đó thuộc cùng mạng major và có subnet mask trùng với địa chỉ IP của interface gởi ra bản tin update**. Trong trường hợp này, mạng 172.168.1.0 và mạng 172.168.2.0 thuộc cùng mạng major 172.16.0.0/16 (lớp B) và có chung subnet mask (/24). Như vậy R1 sẽ có thể gửi update về mạng 172.168.1.0 cho R2. Nếu địa chỉ IP của interface f0/0 của R1 có subnet mask khác subnet mask của interface nối tới R2, thì R1 sẽ không gửi update về mạng 172.168.1.0 cho R2 dù 2 Router này đang chạy cùng giao thức.

Đặc điểm Classful routing của giao thức RIPv1, đó là các Router chạy giao thức định tuyến này bị giới hạn phải dùng 1 subnet mask **duy nhất** cho tất cả mạng con thuộc cùng 1 mạng major classful. Đặc điểm này hoàn toàn không thể tối ưu được trong mạng triển khai VLSM (chia mạng con của mạng con, dẫn đến trong mạng có nhiều hơn 1 subnet mask cho tất cả mạng con) – Để giải quyết bài toán đó chúng ta sẽ có giao thức RIPv2 (chương 7) hoặc nâng cao hơn, cho phép sử dụng nhiều subnet mask khác nhau cho các mạng con thuộc cùng 1 mạng major

**[?]Câu hỏi thứ hai:** *Khi R2 gửi bản tin sang cho R3, do đặc điểm Classful Routing nên R2 sẽ không gửi subnet mask kèm theo các mạng đích của tuyến đường update. Như vậy làm sao R3 biết được các mạng 172.16.1.0 và 172.16.2.0 và 172.16.3.0 có subnet mask bằng bao nhiêu?*



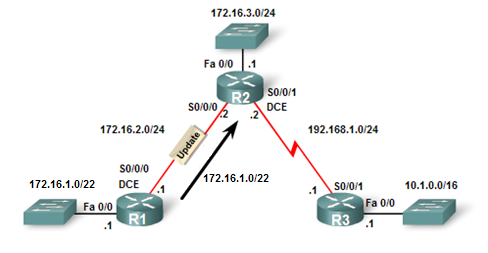
R3 sẽ không thể xác định được các subnet trong bản tin quảng bá từ R2, do đó, chúng ta có hành vi thứ 3: **1 Router sẽ tự động summary (auto summary) các subnet về mạng major tương ứng khi quảng bá giữa các miền mạng major**. Trong ví dụ trên, R2 sẽ tự động summary và gộp các subnet 172.16.1.0, 172.16.2.0 và 172.16.3.0 vào mạng major tương ứng là 172.16.0.0/16 (Class B) để gửi sang R3, vì R3 thuộc miền mạng 192.168.1.0/24 (Class C) khác với mạng major của các subnet của R2. Tuy nhiên R3 sẽ vẫn chỉ nhận mạng 172.16.0.0, hoàn toàn không có subnet mask đi cùng. Khi đó, hành vi thứ 4 – hành vi cuối cùng của Classful Routing : **Khi 1 Router nhận được bản tin update về 1 mạng thuộc mạng major khác với mạng major của địa chỉ IP của interface nhận bản tin update, Router sẽ tự động gán cho tuyến đường đó subnet mask mặc định tương ứng với lớp địa chỉ đó**. Trong ví dụ trên, R3 nhận được update cho mạng 172.16.0.0, R3 thấy mạng 172.16.0.0 là 1 địa chỉ lớp B, khác với lớp của mạng major của interface s0/0/1 (mạng lớp C 192.168.1.0), vì thế R3 sẽ tự động cập nhật mạng 172.16.0.0 với subnet mask bằng 255.255.0.0 (~ hay /16) vào bảng định tuyến

### Classless Routing

Tương ứng với Classless Addressing, với định tuyến chúng ta có khái niệm Classless Routing. Do đặc điểm của Classless Addressing dẫn tới phần mạng của 1 địa chỉ IP không được xác định bởi lớp nữa, mà thông qua subnet mask, vì thế số lượng subnet mask không bị giới hạn bởi /8, /16 và /24. 1 mạng đích không thể được xác định dựa vào octet đầu tiên của địa chỉ mạng đích đó, vì thế để Router có thể cập nhật tuyến đường về 1 mạng đích vào bảng định tuyến, bản tin định tuyến mang thông tin địa chỉ mạng đích cần phải được gửi kèm với subnet mask tương ứng.

Các giao thức định tuyến còn lại đều là đại diện của Classless Routing, bao gồm: RIPv2, EIGRP, OSPF, IS-IS,… Chúng ta xét ví dụ sau:

Cho topology:



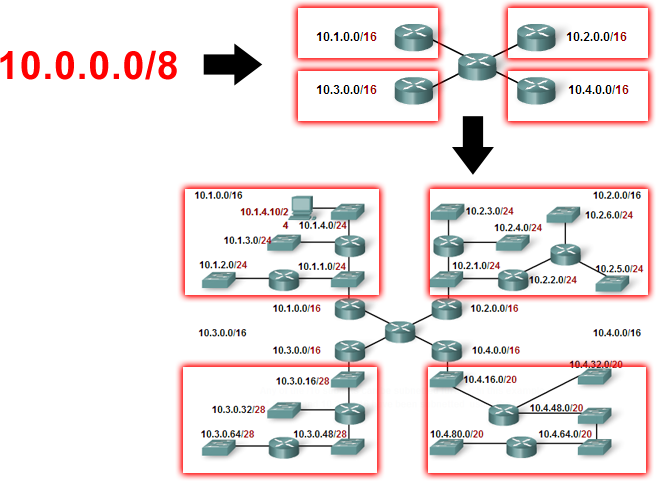
Classless Routing sẽ gửi subnet mask kèm theo địa chỉ mạng đích, vì thế trong trường hợp này, với câu hỏi thứ nhất như phía trên ***làm sao R2 xác định được subnet mask cho mạng 172.168.1.0 để cập nhật vào bảng định tuyến?*** chúng ta dễ dàng nhận ra R2 sẽ luôn xác định được subnet mask cho mạng 172.168.1.0 vì subnet mask đã được gởi kèm bởi R1. Do đó, R1 cũng không cần phải có hành vi thứ 2 (- **1 Router chỉ gửi quảng bá về 1 subnet khi subnet đó thuộc cùng mạng major và có subnet mask trùng với địa chỉ IP của interface gởi ra bản tin update**)

Tương tự như vậy với câu hỏi thứ 2 - ***làm sao R3 biết được các mạng 172.16.1.0 và 172.16.2.0 và 172.16.3.0 có subnet mask bằng bao nhiêu****?*  Cũng rất dễ dàng, vì R3 luôn có thể biết được các mạng con 172.16.1.0, 172.16.2.0 và 172.16.3.0 có subnet mask bằng bao nhiêu, vì khi quảng bá thông tin mạng con, R2 đã gởi kèm thông tin đó rồi

## VLSM và CIDR

### VLSM

Chúng ta đã đề cập tới trong kỳ 1 CCNA, VLSM là 1 cách tiếp cận, ý tưởng rất cơ bản: cho phép chia mạng con của mạng con (subneting the subnet)



*Hình 6.1 – VLSM hỗ trợ chia địa chỉ “vét cạn” –mạng con của mạng con*

### CIDR

CIDR (Classless Inter-Domain Routing) là một khái niệm được đưa ra từ năm 1993 để giải quyết bài toán gộp tuyến đường. CIDR – đúng như tên gọi, là cách tiếp cận dành cho Classless Routing

Bắt đầu từ Classful routing, chúng ta thấy giữa các miền mạng major, chỉ có các subnet mask gốc của các lớp tương ứng A, B, và C:

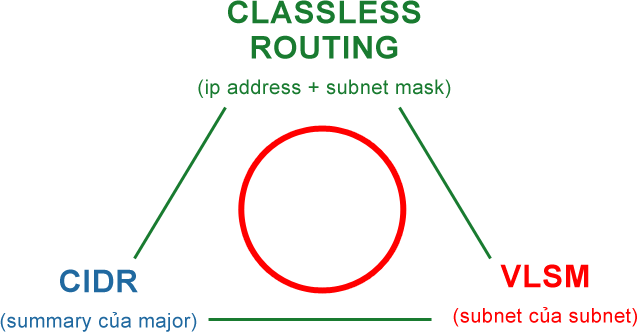
* **Nếu triển khai trong Doanh nghiệp:** Classful Routing không hỗ trợ VLSM, do đó không hỗ trợ nhiều hơn 1 subnet mask cho 1 miền mạng major, và nhất là không hỗ trợ mạng không liên tục
* **Nếu triển khai trên Internet:** Classful Routing dựa vào Classful Addressing đặt ra giới hạn về subnet mask với 3 giá trị cố định. Làm sao chúng ta có tuyến đường tới các mạng nhỏ hơn mạng major tương ứng, chẳng hạn 1 tuyến đường tới các mạng /22 (nhỏ hơn lớp C), /14 (nhỏ hơn lớp B) và thậm chí /6 (nhỏ hơn lớp A), có 2 cách: Hoặc là kết nối trực tiếp, hoặc là định tuyến tĩnh, nhưng như thế không phải Classless routing. Chúng ta cần 1 kỹ thuật *cho phép các giao thức định tuyến động trao đổi thông tin về các mạng tổng quát hơn các mạng major*

Ý tưởng của CIDR là cách nói ngược lại so với VLSM:

* VLSM cho phép chia mạng con của mạng con (subnet of subnet) từ 1 địa chỉ mạng major
* CIDR sẽ cho phép gộp các mạng major thành các mạng tổng quát hơn, được gọi là *supernet*

Supernet là mạng gộp của các mạng major. Như vậy nói Supernet là 1 mạng gộp (summary route) là chính xác, tuy nhiên 1 mạng gộp thì không phải luôn là 1 supernet. 1 Mạng gộp có thể chỉ là 1 mạng con 1 mạng major, có thể là 1 mạng major, và có thể là 1 mạng supernet

Cơ chế gộp tuyến đường không khác gì khi chúng ta triển khai gộp tuyến với định tuyến tĩnh, tuy nhiên CIDR là khái niệm đi kèm với định tuyến động, CIDR cho phép Router quảng bá các mạng supernet giữa các miền mạng major khác nhau. Như vậy rõ ràng là, VLSM và CIDR luôn đi kèm với nhau, và luôn đi kèm với Classless routing, và 3 khái niệm này là 3 cột trụ định hình nên cấu trúc địa chỉ IP của toàn bộ mạng Internet hiện nay

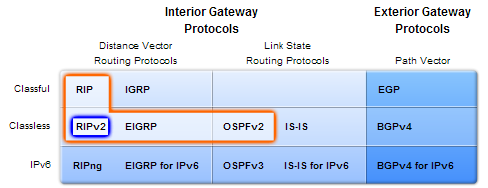


*Hình 6.2 – Mối tương quan giữa Classless routing, VLSM và CIDR*

# ****CHƯƠNG 7 - RIPv2****

RIPv2 được mô tả chi tiêt trong RFC 1723, là phiên bản nâng cao hơn, hỗ trợ định tuyến không phân lớp so với RIPv1. Mặc dù được đưa vào rất nhiều cải tiến so với RIPv1, tuy nhiên RIPv2 vẫn chưa trở thành lựa chọn tối ưu khi so sánh với các giao thức khác như EIGRP, OSPF hay IS-IS. Vấn đề nằm ở cách thức hoạt động của RIP, khiến cho nó trở thành 1 giao thức thiếu khả năng mở rộng cũng như kém tối ưu trong việc lựa chọn tuyến đường và tốc độ hội tụ. Tuy vậy, RIP vẫn là giao thức được hỗ trợ bởi hầu hết hệ điều hành IOS cũng như tất cả các nhà sản xuất phần cứng hiện nay, vì đặc điểm đơn giản và dễ triển khai, nên RIP lại trở thành lựa chọn tối ưu với những mạng có quy mô vừa và nhỏ

Chương 7 sẽ tập trung phân tích những khác biệt giữa RIPv1 và RIPv2, những hạn chế của RIPv1 và cách giải quyết các hạn chế đó trên RIPv2. Giới hạn lớn của RIPv1 nằm ở việc định tuyến theo lớp (Classful Routing) dẫn tới không hỗ trợ VLSM và CIDR. Tương ứng, thay đổi lớn nhất tại RIPv2 đó là việc chuyển sang định tuyến không theo lớp (Classless Routing) nhờ đó hỗ trợ VLSM và CIDR, cũng như giải quyết được bài toán mạng gián đoạn



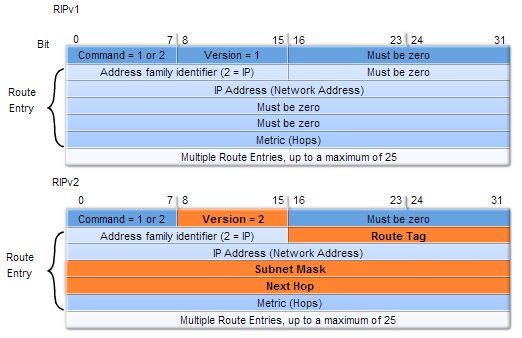
## Các tính năng RIPv2

### Đặc điểm chung

Tương tự RIPv1, RIPv2 là giao thức Distance-Vector, do đó cả 2 phiên bản đều chia sẻ chung các đặc điểm và giới hạn như

* Sự tồn tại của Loop dẫn tới các cơ chế chống loop như Holddown-timer và Split horizon, kết hợp với poison-reverse
* Sử dụng 4 trạng thái thời gian bao gồm: Update (chu kỳ cập nhật) Invalid (thời gian chờ update), Holddown (thời gian hoãn), và Flush (thời gian tồn tại)
* Sử dụng cập nhật kích hoạt khi mạng có thay đổi để tăng tốc độ hội tụ
* Sử dụng metric giới hạn bằng 15 hop

### Bản tin RIPv2



*Hình 7.1 – so sánh sự khác nhau giữa bản tin RIPv1 và RIPv2*

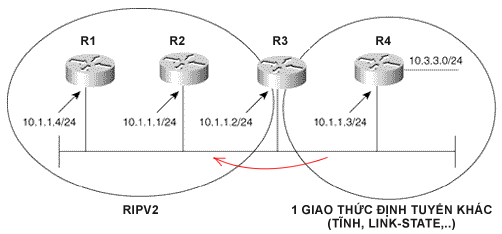
Ta thấy RIPv2 có 1 chút thay đổi về cấu trúc bản tin so với RIPv1:

**Version = 2 :** Để chỉ phiên bản RIP này là phiên bản 2

**Route Tag :** Là 1 trường đặc biệt để đánh dấu tuyến đường, giúp việc nhận biết các gói tin trở nên dễ dàng hơn. Chẳng hạn chúng ta có thể ngừng cập nhật các thông tin tuyến đường tới từ 1 Router nào đó bằng cách đánh dấu các bản tin cập nhật của Router đó với 1 giá trị cụ thể, sau đó lọc tất cả bản tin có giá trị đánh dấu đó.

**Subnet mask :** Trường mang thông tin về mặt nạ mạng, là thông tin quý giá nhất cho phép biến RIP trở thành 1 giao thức Classless routing, hỗ trợ đồng thời cả VLSM và CIDR

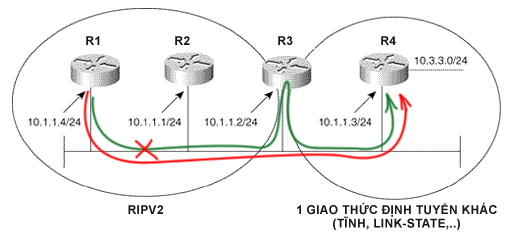
**Next-hop:** Là 1 trường đặc biệt cho phép Router xác định địa chỉ IP nào được sử dụng bởi Router nhận update sử dụng như địa chỉ IP next-hop. Nếu trường này 0.0.0.0 đồng nghĩa với Router gửi update chính là Router next-hop. Vậy còn trường hợp nào trường next-hop được sử dụng?



*HÌnh 7.2 Trường next-hop được sử dụng tron môi trường đa truy cập*

Trong ví dụ, chúng ta giả sử có 2 miền định tuyến và Router R3 là Router biên của 2 miền định tuyến, R3 chạy RIPv2 với R1 và R2, còn chạy 1 giao thức khác với R4, do đó Router R3 có đầy đủ thông tin về các mạng kể cả mạng 10.3.3.0.

Do các Router R1, R2 chỉ chạy RIPv2, nên sẽ không học được thông tin về mạng 10.3.3.0 trực tiếp từ R4, thông tin về mạng đích 10.3.3.0/24 sẽ được quảng bá bởi R3 cho R1 và R2. Router R3 nhận ra rằng, mặc dù nó là Router gửi ra quảng bá, nhưng sẽ tối ưu hơn nếu R1 và R2 gửi gói tin tới mạng 10.3.3.0/24 trực tiếp tới R4. Lúc này, trường next-hop sẽ là địa chỉ IP của R4 là 10.1.1.3



*Hình 7.3 – Tối ưu quá trình chuyển tiếp gói tin*

### Các cải tiến so với RIPv1

**Định tuyến Classless**

RIPv1 là giao thức Classful routing, bản thân điều này hạn chế vô số chức năng của RIP, vì classful routing không thể triển khai VLSM và CIDR. RIPv2 đã sửa lại điều đó và trở thành 1 giao thức Classless routing. Hay nói cách khác RIPv2 gửi subnet mask kèm theo địa chỉ IP trong bản tin cập nhật

**Hỗ trợ VLSM**

RIPv1 không hỗ trợ VLSM, vì Classful routing không gửi subnet mask kèm theo thông tin địa chỉ IP mạng trong bản tin cập nhật. Điều đó dẫn tới 1 điều kiện khi thực hiện quy hoạch địa chỉ IP: Trong 1 miền mạng major, chúng ta có thể chia bao nhiêu mạng con tùy thích (phụ thuộc vào khả năng đáp ứng của địa chỉ mạng major đó) nhưng *tất cả mạng con phải có cùng chiều dài subnet mask.* Chúng ta đã phân tích rát nhiều về vấn đề này, đó là cách thức chia địa chỉ không tối ưu, quá lãng phí không gian địa chỉ. Để giải quyết bài toán lãng phí, ta phải dùng VLSM, nhưng dùng VLSM thì không dùng Classful routing được.

RIPv2 là giao thức Classless routing, do đó RIPv2 hỗ trợ VLSM.

**Hỗ trợ CIDR**

Tương tự VLSM, CIDR cũng được hỗ trợ bởi RIPv2. Với CIDR, RIPv2 cho phép triển khai 1 kiểu gộp tuyến đường mới hiệu quả hơn được gọi là *gộp tuyến thủ công*. Với CIDR, RIPv2 cho phép mạng đích trong bản tin cập nhật liên mạng major có subnet mask khác /8, /16 và /24. Hay nói cách khác RIPv2 cho phép bản tin cập nhật la thông tin về 1 mạng con, 1 mạng major, hoặc thậm chí la 1 mạng supernet

**Multicast thay vì Broadcast**

Kiểu truyền bằng cách gửi ra bản tin Broadcast theo kiểu RIPv1 là không tối ưu.

* Thứ nhất, bản tin Broadcast sẽ chạy dọc toàn bộ miền quảng bá, việc gửi ra bản tin Broadcast gặp các vấn đề liên quan tới lãng phí băng thông trên link
* Thứ hai, bản tin Broadcast được gửi tới tất cả thiết bị, kể cả PC (các thiết bị không định tuyến), và kể cả tới các Router khác nhưng không tham gia định tuyến.

Để giải quyết 2 bài toán trên, bản tin Multicast được sử dụng để thay thế bản tin Broadcast. RIPv2 sử dụng bản tin Multicast 224.0.0.9 để gửi tới tất cả Router đang chạy RIP, khi đó các thiết bị không chạy RIP thì sẽ không phải nhận các bản tin RIP nữa

### Gộp tuyến thủ cong – Manual Summarization

Kỹ thuật gộp tuyến là 1 kỹ thuật tối ưu tuyến đường được sử dụng bởi tất cả giao thức định tuyến. Về cơ bản sẽ có 2 kiểu gộp tuyến:

* Tự động gộp tuyến (Auto-Summarization)
* Gộp tuyến thủ công (Manual-Summarization)

Gộp tuyến tự động là tính năng Router tự động thực hiện gộp các mạng con thuộc cùng 1 miền mạng major khi gửi update sang 1 miền mạng major khác. Gộp tuyến tự động là “di tích lịch sử” để lại của Classful Addressing, do đó các giao thức định tuyến Classful Routing đều hỗ trợ gộp tuyến tự động. Đại diện các giao thức này là RIPv1, RIPv2, IGRP và EIGRP. 1 Giao thức hỗ trợ gộp tuyến tự động, đồng nghĩa với hỗ trợ Classful Addressing, đồng nghĩa với 1 phương thức triển khai định tuyến phi VLSM và CIDR.

Gộp tuyến thủ công là tính năng cho phép giao thức định tuyến thực hiện gộp tuyến theo yêu cầu của người quản trị. Hay nói cách khác, gộp tuyến thủ công cho phép thực hiện gộp tuyến với bất kỳ nhóm mạng con nào không quan tâm thuộc cùng miền mạng major hay không, và cũng không giới hạn chiều dài subnet mask tuyến đường gộp. Như vậy gộp tuyến thủ công chỉ có thể triển khai khi giao thức hỗ trợ CIDR, các mạng gộp có thể là 1 supernet, có thể là 1 mạng major, hay cũng có thể là 1 mạng con. Khi đó, giao thức cho phép gộp tuyến thủ công phải là 1 giao thức Classless routing. Các giao thức hỗ trợ gộp tuyến thủ công bao gồm: RIPv2, EIGRP, và các giao thức Link-State như OSPF.

Ý nghĩa của phương pháp gộp tuyến đường đó là làm tăng tốc độ tra cứu tuyến đường và giảm kích cỡ của bảng định tuyến, chúng ta đã đề cập tới điều này trong các phần trước

Từ giờ chúng ta sẽ sử dụng manual-summarization, manual-summary hoặc ngắn gọn là summary khi đề cập tới gộp tuyến thủ công

### Hỗ trợ xác thực

RIPv2 hỗ trợ xác thực giữa các Router chạy định tuyến trước khi bản tin định tuyến thực sự được trao đổi. Nếu Router chạy RIP cập nhật mọi thông tin định tuyến mà nó nhận được, hiển nhiên là kém an toàn vì đối tượng tấn công hoàn toàn có thể gửi ra thông tin định tuyến không chính xác khiến cho việc tính toán định tuyến không còn chính xác nữa. Xác thực trên RIPv2 yêu cầu 2 Router phải có cùng chìa khóa mới được trao đổi bản tin định tuyến với nhau

## Cấu hình RIPv2

RIPv2 là phiên bản cải tiến của RIPv1, nên những tính năng cơ bản trên RIPv1, thì RIPv2 đều kế thừa:

Quảng bá tuyến đường:

RouterX(config-router)#network {địa chỉ IP mạng major}

Passive-interface

RouterX(config-router)#passive-interface {loại/slot}

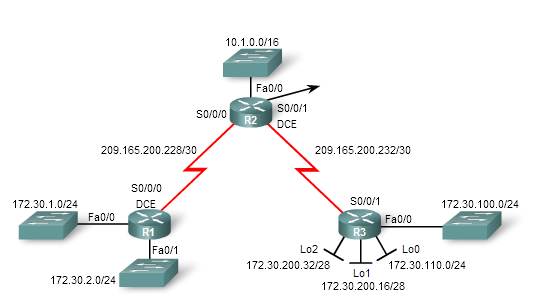
Auto-summary

RouterX(config-router)#auto-summary

Quảng bá default route

RouterX(config-router)#default-information originate

Để theo dõi ví dụ triển khai các tính năng của RIPv2, chúng ta sử dụng topology dưới đây:



**Cấu hình khởi tạo:**

Chúng ta cấu hình khởi tạo 3 Router R1, R2, R3 định tuyến quảng bá các mạng với RIPv1

Trên Router R1

R1(config-router)#network 172.30.0.0

R1(config-router)#network 209.165.200.0

Trên Router R2

R1(config-router)#network 10.0.0.0

R1(config-router)#network 209.165.200.0

Trên Router 3

R1(config-router)#network 172.30.0.0

R1(config-router)#network 209.165.200.0

**Kiểm tra bảng định tuyến:** Giao thức là RIPv1, topology là mạng không liên tục, nên chúng ta sẽ thấy vấn đề định tuyến không chính xác

Trên R1

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

R 10.0.0.0/8 [120/1] via 209.165.200.229, 00:00:13, Serial0/0/0

172.30.0.0/24 is subnetted, 2 subnets

C 172.30.1.0 is directly connected, FastEthernet0/0

C 172.30.2.0 is directly connected, FastEthernet0/1

209.165.200.0/30 is subnetted, 2 subnets

C 209.165.200.228 is directly connected, Serial0/0/0

R 209.165.200.232 [120/1] via 209.165.200.229, 00:00:13, Serial0/0/0

Trên R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/16 is subnetted, 1 subnets

C 10.1.0.0 is directly connected, FastEthernet0/0

R 172.30.0.0/16 [120/1] via 209.165.200.230, 00:00:02, Serial0/0/0

[120/1] via 209.165.200.234, 00:00:06, Serial0/0/1

209.165.200.0/30 is subnetted, 2 subnets

C 209.165.200.228 is directly connected, Serial0/0/0

C 209.165.200.232 is directly connected, Serial0/0/1

Trên R3

R3#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

R 10.0.0.0/8 [120/1] via 209.165.200.233, 00:00:06, Serial0/0/1

172.30.0.0/16 is variably subnetted, 4 subnets, 2 masks

C 172.30.100.0/24 is directly connected, FastEthernet0/0

C 172.30.110.0/24 is directly connected, Loopback0

C 172.30.200.16/28 is directly connected, Loopback1

C 172.30.200.32/28 is directly connected, Loopback2

209.165.200.0/30 is subnetted, 2 subnets

R 209.165.200.228 [120/1] via 209.165.200.233, 00:00:06, Serial0/0/1

C 209.165.200.232 is directly connected, Serial0/0/1

### Chạy RIPv2

**Cú pháp**

Mặc định khi chúng ta từ đi vào sub-mode router rip, chúng ta đang cấu hình RIPv1, để triển khai RIPv2, chúng ta sử dụng cú pháp version 2

RouterX(config)#router rip

RouterX(config-router)#version 2

Lúc này, tất cả các tính năng của RIPv2 sẽ được sử dụng, kiểu định tuyến sẽ là Classless routing, hỗ trợ VLSM, CIDR, truyền thông multicast, và các tính năng khác.

**Triển khai**

Cấu hình RIPv2 trên tất cả Router

Trên Router R1

R1(config-router)#version 2

Trên Router R2

R2(config-router)#version 2

Trên Router R3

R3(config-router)#version 2

**Xác nhận:**

Để kiểm tra, chúng ta sử dung câu lệnh show ip protocol

R3#show ip protocol

Routing Protocol is "rip"

Sending updates every 30 seconds, next due in 5 seconds

Invalid after 180 seconds, hold down 180, flushed after 240

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Redistributing: rip

Default version control: send version 2, receive 2

Interface Send Recv Triggered RIP Key-chain

Loopback0 2 2

Loopback1 2 2

Loopback2 2 2

Serial0/0/1 2 2

Automatic network summarization is in effect

<output omitted>

Ta thấy Router gửi ra các bản tin RIPv2. Tuy nhiên mặc định cơ chế auto-summary vẫn được bật

Chúng ta kiểm tra kiểu truyền thông trong RIPv2 với câu lệnh debug ip rip

R3#debug ip rip

RIP: sending v2 update to 224.0.0.9 via Serial0/0/1 (209.165.200.234)

RIP: build update entries

172.30.0.0/16 via 0.0.0.0, metric 1, tag 0

RIP: received v2 update from 209.165.200.233 on Serial0/0/1

10.0.0.0/8 via 0.0.0.0 in 1 hops

209.165.200.228/30 via 0.0.0.0 in 1 hops

RIP: sending v2 update to 224.0.0.9 via Loopback0 (172.30.110.1)

RIP: build update entries

10.0.0.0/8 via 0.0.0.0, metric 2, tag 0

172.30.100.0/24 via 0.0.0.0, metric 1, tag 0

172.30.200.16/28 via 0.0.0.0, metric 1, tag 0

172.30.200.32/28 via 0.0.0.0, metric 1, tag 0

209.165.200.0/24 via 0.0.0.0, metric 1, tag 0

Chúng ta thấy RIPv2 gửi bản cập nhật ra địa chỉ Multicast 224.0.0.9 thay vì Broadcast. Ngoài ra, thông tin định tuyến có sự xuất hiện thêm của thông số next-hop (via 0.0.0.0) và thông tin tag (tag 0). Đây là 2 trường mới xuất hiện trong bản tin của RIPv2

Tuy nhiên vấn đề quan trọng nhất chúng ta thấy đó là thông tin cập nhật được gửi ra vẫn là 172.30.0.0, mặc dù đã có subnet mask đi kèm với thông tin địa chỉ IP (/16) Do mặc định trên RIPv2 vẫn triên khai auto-summary. Cơ chế auto-summary sẽ tự động gộp tuyến khi bản tin cập nhật đi qua giữa 2 miền mạng major. Để tối ưu RIPv2, chúng ta tắt tính năng auto-summary. Như vậy vấn đề mạng không liên tục lúc ban đầu vẫn còn, chúng ta kiểm tra bảng định tuyến Router R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/16 is subnetted, 1 subnets

C 10.1.0.0 is directly connected, FastEthernet0/0

R 172.30.0.0/16 [120/1] via 209.165.200.234, 00:00:07, Serial0/0/1

[120/1] via 209.165.200.230, 00:00:16, Serial0/0/0

209.165.200.0/30 is subnetted, 2 subnets

C 209.165.200.228 is directly connected, Serial0/0/0

C 209.165.200.232 is directly connected, Serial0/0/1

và 2 miền LAN của R1 và R3 vẫn chưa thể kết nối được:

R3# show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

R 10.0.0.0/8 [120/1] via 209.165.200.233, 00:00:17, Serial0/0/1

172.30.0.0/16 is variably subnetted, 4 subnets, 2 masks

C 172.30.100.0/24 is directly connected, FastEthernet0/0

C 172.30.110.0/24 is directly connected, Loopback0

C 172.30.200.16/28 is directly connected, Loopback1

C 172.30.200.32/28 is directly connected, Loopback2

209.165.200.0/30 is subnetted, 2 subnets

R 209.165.200.228 [120/1] via 209.165.200.233, 00:00:17, Serial0/0/1

C 209.165.200.232 is directly connected, Serial0/0/1

### Tắt auto-summary

**Cú pháp:**

Tương tự những câu lệnh xóa tính năng khác, chúng ta thêm tiền tố no trước câu lệnh gốc

RouterX(config)#router rip

RouterX(Config)#ver 2

RouterX(config)#no auto-summary

Chú ý: Mặc định có thể tắt auto-summary trên RIPv1, nhưng câu lệnh sẽ không có tác dụng gì

**Triển khai:**

Tắt auto-summary trên tất cả 3 Router

Trên R1

R1(config-router)#no auto-summary

Trên R2

R2(config-router)#no auto-summary

Trên R3

R3(config-router)#no auto-summary

**Xác nhận**

Để kiểm tra chúng ta có thể sử dụng show ip protocol, tuy nhiên kiểm chứng rõ ràng nhất là theo dõi bảng định tuyến:

Trên R1

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/16 is subnetted, 1 subnets

R 10.1.0.0 [120/1] via 209.165.200.229, 00:00:00, Serial0/0/0

172.30.0.0/16 is variably subnetted, 6 subnets, 2 masks

C 172.30.1.0/24 is directly connected, FastEthernet0/0

C 172.30.2.0/24 is directly connected, FastEthernet0/1

R 172.30.100.0/24 [120/2] via 209.165.200.229, 00:00:00, Serial0/0/0

R 172.30.110.0/24 [120/2] via 209.165.200.229, 00:00:00, Serial0/0/0

R 172.30.200.16/28 [120/2] via 209.165.200.229, 00:00:00, Serial0/0/0

R 172.30.200.32/28 [120/2] via 209.165.200.229, 00:00:00, Serial0/0/0

209.165.200.0/30 is subnetted, 2 subnets

C 209.165.200.228 is directly connected, Serial0/0/0

R 209.165.200.232 [120/1] via 209.165.200.229, 00:00:00, Serial0/0/0

Trên R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/16 is subnetted, 1 subnets

C 10.1.0.0 is directly connected, FastEthernet0/0

172.30.0.0/16 is variably subnetted, 6 subnets, 2 masks

R 172.30.1.0/24 [120/1] via 209.165.200.230, 00:00:19, Serial0/0/0

R 172.30.2.0/24 [120/1] via 209.165.200.230, 00:00:19, Serial0/0/0

R 172.30.100.0/24 [120/1] via 209.165.200.234, 00:00:06, Serial0/0/1

R 172.30.110.0/24 [120/1] via 209.165.200.234, 00:00:06, Serial0/0/1

R 172.30.200.16/28 [120/1] via 209.165.200.234, 00:00:06, Serial0/0/1

R 172.30.200.32/28 [120/1] via 209.165.200.234, 00:00:06, Serial0/0/1

209.165.200.0/30 is subnetted, 2 subnets

C 209.165.200.228 is directly connected, Serial0/0/0

C 209.165.200.232 is directly connected, Serial0/0/1

Trên R3

R3(config-router)#do show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/16 is subnetted, 1 subnets

R 10.1.0.0 [120/1] via 209.165.200.233, 00:00:15, Serial0/0/1

172.30.0.0/16 is variably subnetted, 6 subnets, 2 masks

R 172.30.1.0/24 [120/2] via 209.165.200.233, 00:00:15, Serial0/0/1

R 172.30.2.0/24 [120/2] via 209.165.200.233, 00:00:15, Serial0/0/1

C 172.30.100.0/24 is directly connected, FastEthernet0/0

C 172.30.110.0/24 is directly connected, Loopback0

C 172.30.200.16/28 is directly connected, Loopback1

C 172.30.200.32/28 is directly connected, Loopback2

209.165.200.0/30 is subnetted, 2 subnets

R 209.165.200.228 [120/1] via 209.165.200.233, 00:00:15, Serial0/0/1

C 209.165.200.232 is directly connected, Serial0/0/1

Chúng ta thấy tất cả mạng con đều đã được cập nhật bởi các Router. RIPv2 đã thực hiện thành công Classless routing, để chi tiết chúng ta kiểm tra quá trình cập nhật qua debug

R3#debug ip rip

RIP: build update entries

10.1.0.0/16 via 0.0.0.0, metric 2, tag 0

172.30.1.0/24 via 0.0.0.0, metric 3, tag 0

172.30.2.0/24 via 0.0.0.0, metric 3, tag 0

172.30.100.0/24 via 0.0.0.0, metric 1, tag 0

172.30.110.0/24 via 0.0.0.0, metric 1, tag 0

172.30.200.16/28 via 0.0.0.0, metric 1, tag 0

209.165.200.228/30 via 0.0.0.0, metric 2, tag 0

209.165.200.232/30 via 0.0.0.0, metric 1, tag 0

RIP: sending v2 update to 224.0.0.9 via Serial0/0/1 (209.165.200.234)

RIP: build update entries

172.30.100.0/24 via 0.0.0.0, metric 1, tag 0

172.30.110.0/24 via 0.0.0.0, metric 1, tag 0

172.30.200.16/28 via 0.0.0.0, metric 1, tag 0

172.30.200.32/28 via 0.0.0.0, metric 1, tag 0

RIP: received v2 update from 209.165.200.233 on Serial0/0/1

10.1.0.0/16 via 0.0.0.0 in 1 hops

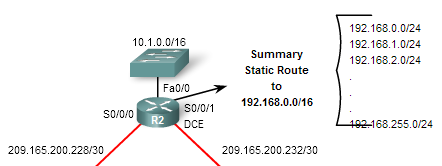
172.30.1.0/24 via 0.0.0.0 in 2 hops

172.30.2.0/24 via 0.0.0.0 in 2 hops

209.165.200.228/30 via 0.0.0.0 in 1 hops

### RIPv2 và CIDR – redistribution

RIPv2 hỗ trợ CIDR là 1 khác biệt đáng kể, chúng ta sẽ mô tả đặc điểm đó qua ví dụ tiếp theo:



Vẫn topology ban đầu, lúc này giả sử Router R2 có thêm kết nối trực tiếp tới cac mạng 192.168.0.0/24, 192.168.1.0/24 và 192.168.2.0/24. R2 muốn quảng bá các mạng trên vào định tuyến cho R1 và R3, để mô phỏng các mạng kết nối trực tiếp mới, chúng ta sử dụng interface loopback

R2(config)#interface loopback0

R2(config-if)#ip address 192.168.0.1 255.255.255.0

R2(config)#interface loopback1

R2(config-if)#ip address 192.168.1.1 255.255.255.0

R2(config)#interface loopback2

R2(config-if)#ip address 192.168.2.1 255.255.255.0

Interface loopback là 1 loại interface logic được sử dụng trên Router để giả lập 1 mạng kết nối trực tiếp. CHúng ta có thể tạo ra số lượng interface loopback thoải mái, và mỗi interface loopback, tương tự các interface vật lý khác, có thể đặt địa chỉ IP và các tính năng như định tuyến

Tuy nhiên người quản trị không muốn quảng bá tất cả cá mạng 192.168.0.0, 192.168.1.0, và 192.168.2.0, mà muốn quảng bá 1 mạng gộp supernet 192.168.0.0/16. Theo đó, với RIPv2, chúng ta có 3 cách:

* Cách 1 – Quảng bá với cú pháp network, sau đó thực hiện manual-summary
* Cách 2 – Triển khai 1 tuyến tĩnh gộp các mạng lại rồi phân phối lại tuyến tĩnh vào RIP

Chúng ta chưa đề cập tới tính năng manual-summary trong tài liệu này, do đó chúng ta sẽ sử dụng cách thứ 2

**Cú pháp kỹ thuật phân phối lại – redistribution:**

Kỹ thuật redistribution là kỹ thuật cho phép phân phối thông tin định tuyến giữa các nguồn định tuyến khác nhau. Chẳng hạn các thông tin định tuyến từ 1 nguồn OSPF quảng bá vào miền định tuyến RIPv2. Các nguồn định tuyến có thể là:

* Một giao thức định tuyến động
* Định tuyến tĩnh (static)
* Kết nối trực tiếp (connected)

Cú pháp được sử dụng là redistribution

RouterX(config)#router rip

RouterX(config-router)#redistribution {nguồn định tuyến}

**Triển khai:**

Trước khi phân phối lại, chúng ta cần tạo ra tuyến tĩnh gộp cho các mạng 192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24.

R2(config)#ip route 192.168.0.0 255.255.0.0 null0

null tương tự loopback, là 1 interface logic. Interface null được sử dụng để dẩy các gói tin vào “thùng rác” hay nói cách khác, các gói tin đẩy vào null sẽ bị drop. Ý nghĩa quan trọng nhât của tuyến tĩnh trên là tạo ra 1 tuyến tĩnh gộp được 1 mạng supernet 192.168.0.0/16

Chúng ta kiểm tra bảng đinh tuyến của R2

S 192.168.0.0/16 is directly connected, Null0

C 192.168.0.0/24 is directly connected, Loopback0

C 192.168.1.0/24 is directly connected, Loopback1

C 192.168.2.0/24 is directly connected, Loopback2

Bây giờ chúng ta sẽ triển khai phân phôi lại để quảng bá mạng 192.168.0.0/16 sang cho R1 và R3. Nguồn lúc này là tĩnh (static)

R2(config)#router rip

R2(config)#redistribution static

**Xác nhận:**

Để xác nhận chúng ta kiểm tra bảng định tuyến của R1 và R3

Trên R1

R1#show ip route

<output omitted>

R 172.30.200.16/28 [120/2] via 209.165.200.229, 00:00:12, Serial0/0/0

R 172.30.200.32/28 [120/2] via 209.165.200.229, 00:00:12, Serial0/0/0

R 192.168.0.0/16 [120/1] via 209.165.200.229, 00:00:12, Serial0/0/0

209.165.200.0/30 is subnetted, 2 subnets

<output omitted>

Trên R3

R3>show ip route

<output omitted>

C 172.30.200.16/28 is directly connected, Loopback1

C 172.30.200.32/28 is directly connected, Loopback2

R 192.168.0.0/16 [120/1] via 209.165.200.233, 00:00:05, Serial0/0/1

209.165.200.0/30 is subnetted, 2 subnets

<output omitted>

Như vậy ta thấy R1 và R3 đã học được tuyến đường tới các mạng 192.168.0.0/24, 192.168.1.0/24 và 192.168.2.0/24 dưới dạng 1 tuyến gộp 192.168.0.0/16. Ta thấy, 192.168.0.0/16 là 1 tuyến supernet, và RIPv2 thực sự hỗ trợ CIDR

# ****CHƯƠNG 8 - CHI TIẾT BẢNG ĐỊNH TUYẾN****

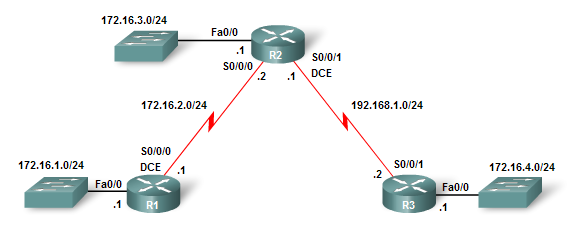
Qua các bài trước, chúng ta đã có 1 phương pháp để kiểm tra bảng định tuyến, bằng cách sử dụng câu lệnh show ip route . show ip route cho chúng ta biết thông tin về các tuyến đường tới mạng đích bất kỳ, mỗi tuyến đường có thể có độ ưu tiên khác nhau dựa trên giao thức định tuyến được sử dụng. Nắm được cụ thể cấu trúc bảng định tuyến sẽ hỗ trợ cho quá trình gỡ rối rất nhiều, trong việc:

* Phân loại và đánh giá các tuyến đường khác nhau đang được sử dụng
* Mô tả tiến trình tra cứu đệ quy xác định outgoing interface
* Mô tả quá trình định tuyến trong mạng IP

## Cấu trúc bảng định tuyến chạy Classful Addressing

Để xem xét cụ thể cách hoạt động và thay đổi của bảng định tuyến, chúng ta sẽ xem xét một mạng định tuyến:

Cho topology:



Các Router được cấu hình địa chỉ IP cho interface tương ứng topology

* Router R1 có kết nối trực tiếp tới mang 172.16.1.0/24 và 172.16.2.0/24
* Router R2 có kết nối trực tiếp tới mang 172.16.2.0/24, 172.16.3.0/24 và 192.168.1.0/24
* Router R3 có kết nối trực tiếp tới mang 192.168.1.0/24 và 172.16.4.0/24

Như vậy với topology trên, mạng 172.16.0.0 là không liên tục, do bị chia cách bởi mạng major 192.168.1.0/24. Chúng ta sẽ xem xét ảnh hưởng của việc này đối với bảng định tuyến trong các mục dưới

### Các bản ghi (entry)

1 Bản ghi trong bảng định tuyến mang toàn bộ thông tin về 1 tuyến đường tới một mạng đích. Các thông tin bao gồm:

* Nguồn học tuyến đường (kết nối trực tiếp, tuyến tĩnh, các loại tuyến động)
* Địa chỉ mạng đích
* Cổng đẩy gói tin ra (outgoing interface) hoặc địa chỉ IP next-hop
* Các thông tin hỗ trợ đưa ra quyết định chọn tuyến đường (AD, metric)

Dưới đây là 1 ví dụ về bảng định tuyến:

Router# show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area

\* - candidate default, U - per-user static route, o - ODR

P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 3 subnets

S 172.16.1.0 is directly connected, FastEthernet0/0

C 172.16.2.0 is directly connected, FastEthernet0/0

R 172.16.3.0 [120/1] via 172.16.2.1, 00:00:09, FastEthernet0/0

Ta thấy nguồn học 1 tuyến đường không ảnh hưởng tới cấu trúc của bảng định tuyến. Cả 3 tuyến đường trên được học thông qua [C]-kết nối trực tiếp, [S]-học tĩnh, [R]-học động (qua RIP) nhưng đều là mạng con của mạng major 172.16.0.0 và đều có subnet mask bằng 255.255.255.0 (~ tương ứng /24)

Ta cũng có nhận xét thứ 2 về cấu trúc bảng định tuyến của các HĐH Cisco IOS: Bảng định tuyến được thể hiện phân cấp dựa trên định tuyến theo lớp (Classful Routing), mặc dù hỗ trợ đồng thời cả Classful và Classless Addressing nhưng dạng trình bày vẫn dựa theo cấu trúc Classful

### Cấp độ tuyến đường

Bảng định tuyến của Cisco không phải bảng cơ sở dữ liệu phẳng, mà có cấu trúc phân cấp. Mục đích là để tăng tốc độ tiến trình tra cứu khi xác định cổng ra để chuyển tiếp gói tin cho 1 tuyến đường. Do đặc điểm của quá trình phân cấp nên sẽ có 2 loại cấp độ tuyến đường:

* Tuyến đường cấp 1
* Tuyến đường cấp 2

1. **Tuyến đường cấp 1 [Level 1 route]**

Tuyến đường cấp 1 là các tuyến đường tới mạng đích có chiều dài subnet mask nhỏ hơn hoặc bằng chiều dài subnet mask gốc của lớp địa chỉ tương ứng – Chẳng hạn tuyến đường tới mạng 192.168.1.0/24 là tuyến đường cấp 1 vì:

192.168.1.0/24 có subnet mask bằng /24

192.168.1.0 là địa chỉ lớp C nên có subnet mask gốc cũng bằng /24

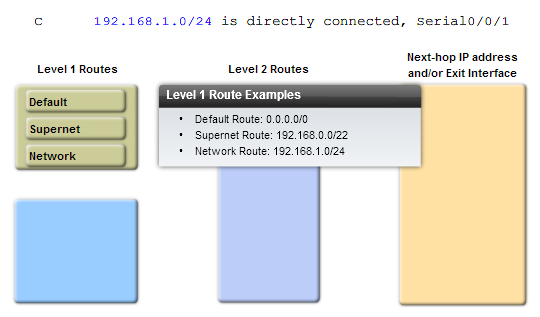
Hoặc 1 ví dụ khác, tuyến tới mạng 10.0.0.0/5 là tuyến đường cấp 1 vì:

10.0.0.0/5 có subnet mask bằng /5

10.0.0.0 là địa chỉ lớp A nên có subnet mask /8, nhỏ hơn /5

Theo định nghĩa trên, 1 tuyến đường cấp 1 có thể là:

* **Tuyến mặc định (default route) –** Default route có thể học tĩnh hoặc học động, địa chỉ mạng của tuyến default route trong bảng định tuyến là 0.0.0.0/0
* **Tuyến Supernet –** Tuyến Supernet là tuyến tới mạng đích có chiều dài subnet mask nhỏ hơn subnet mask mặc định của lớp địa chỉ tương ứng – Một tuyến Supernet được tạo ra thông qua kỹ thuật gộp tuyến đường, có thể học tĩnh hoặc học động – Chẳng hạn 172.16.0.0/14 là 1 tuyến Supernet
* **Tuyến mạng (Network route) –** Tuyến mạng là tuyến tới mạng có chiều dài subnet bằng chiều dài subnet mask mặc định của lớp địa chỉ tương ứng – Chẳng hạn 192.168.1.0/24 là 1 tuyến mạng. Tuyến mạng có thể học qua kết nối trực tiếp, học tĩnh hoặc động

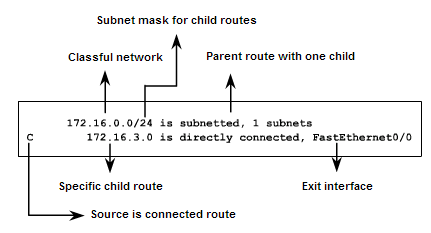


*Hình 8.1 – Tuyến đường cấp 1*

1. **Parent và Child route**

Nhu cầu quy hoạch địa chỉ IP trong mạng dẫn tới kỹ thuật subneting (chia mạng con), từ 1 dải địa chỉ cho trước đó chúng ta tạo ra các subnet (mạng con), theo đó chúng ta có định nghĩa parent và child route:

* **Parent route –** Là các tuyến mạng (tuyến cấp 1) nhưng không có thông tin giao diện đẩy gói ra hoặc IP next-hop đi kèm
* **Child route –** Là các tuyến có địa chỉ đích tới subnet của parent route



*Hình 8.2 – không phai mọi tuyến cấp 1 đều là parent route !  
Chỉ các tuyến không có outgoing interface mới là parent rouet*

Như vậy, trên bảng định tuyến, Parent route được sử dụng để “gom” các child route thành từng nhóm mạng cùng thuộc 1 mạng major. Như vậy ta thấy, mỗi khi 1 subnet được thêm vào bảng định tuyến sẽ tạo ra tương ứng parent route cấp 1 cho subnet đó – Chúng ta xem bảng định tuyến trên R2:

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area

\* - candidate default, U - per-user static route, o - ODR

P - periodic downloaded static route

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 2 subnets 🡪 Parent route

C 172.16.3.0 is directly connected, FastEthernet0/0 🡪 child route

C 172.16.2.0 is directly connected, Serial0/0/0 🡪 child route

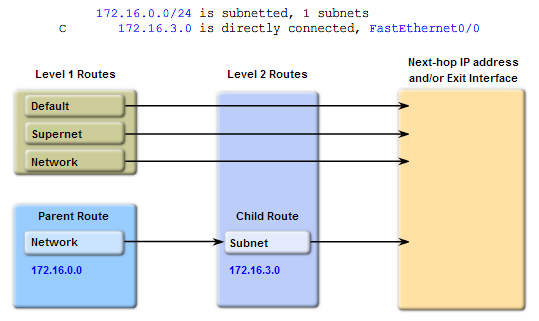
C 192.168.1.0/24 is directly connected, Serial0/0/1

Theo đó, Parent route *luôn là tuyến cấp 1*

1. **Tuyến đường cấp 2**

Tuyến đường cấp 2 là tuyến tới mạng con của 1 mạng Classful, tập hợp các tuyến đường cấp 2 trong bảng định tuyến là tập hợp các child route của tất cả parent route

Trong ví dụ trên, trên R2, các tuyến 172.16.3.0/24, 172.16.1.0/24 và 172.16.2.0/24 đều là tuyến đường cấp 2 vì đều là subnet của mạng Classful 172.16.0.0. Một tuyến đường cấp 2 cũng có thể được học qua 1 trong 3 cách: kết nối trực tiếp, học tĩnh và học động



*Hình 8.3 – tập hợp tất cả child route đều là tuyến đường cấp 2*

1. **Ultimate route**

Ultimate route là các tuyến mà có bản ghi trong bảng định tuyến mang theo thông tin:

* IP Next-hop
* Giao diện chuyển tiếp gói tin (outgoing interface)

Như vậy các Ultimate route có thể là tuyến cấp 1 hoặc tuyến cấp 2. Không phải mọi tuyến cấp 1 đều là Ultimate route. *Nhưng chỉ Ultimate route mới được định tuyến*. Chúng ta quay lại với ví dụ bảng định tuyến hiện tại trên R2 trong ví dụ ban đầu:

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP

i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area

\* - candidate default, U - per-user static route, o - ODR

P - periodic downloaded static route

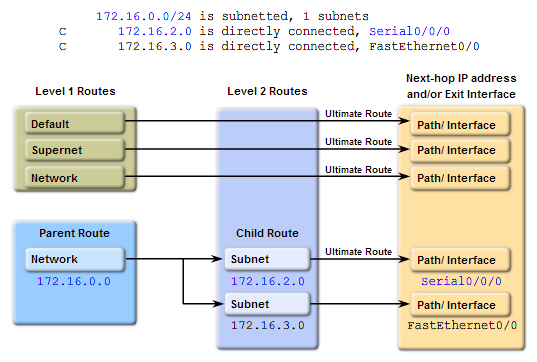
Gateway of last resort is not set

172.16.0.0/24 is subnetted, 3 subnets

C 172.16.3.0 is directly connected, FastEthernet0/0 🡪 Ultimate route

C 172.16.2.0 is directly connected, Serial0/0/0 🡪 Ultimate route

C 192.168.1.0/24 is directly connected, Serial0/0/1 🡪 Ultimate route



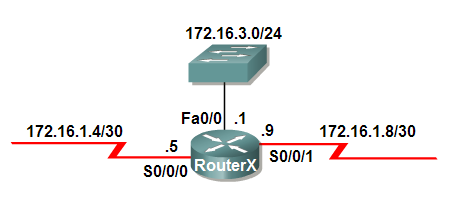
*Hình 8.4 – Chỉ ultimate route mới được định tuyến*

## Cấu trúc bảng định tuyến chạy Classless Addressing

Tương tự với Bảng định tuyến chạy Classful Routing, chúng ta vẫn giữ các định nghĩa về tuyến đường cấp 1, tuyến đường cấp 2, tuyến Ultimate, Parent route và Child route. Tuy nhiên chúng ta sẽ xem xét ảnh hưởng của VLSM đối với định tuyến không theo lớp (Classless Routing)

Để demo cho cấu trúc bảng định tuyến chạy Classless Addressing, chúng ta sử dụng 1 ví dụ khác, ví dụ 2:

Cho topology:



Cấu hình khởi tạo:

RouterX#configure terminal

RouterX(config)#interface s0/0/0

RouterX(config)#ip address 172.16.1.4 255.255.255.252

RouterX(config)#interface s0/0/1

RouterX(config)#ip address 172.16.1.9 255.255.255.252

RouterX(config)#interface f0/0

RouterX(config)#ip address 172.16.3.1 255.255.255.0

Chúng ta kiểm tra bảng định tuyến để thấy sự khác biệt:

RouterX#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks

C 172.16.1.4/30 is directly connected, Serial0/0/0

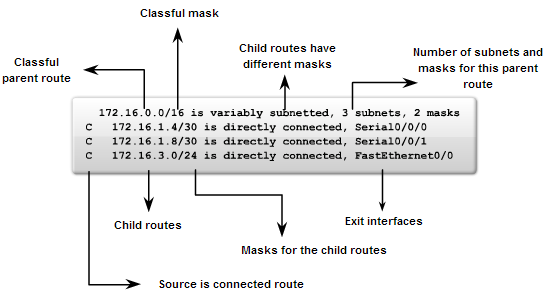
C 172.16.1.8/30 is directly connected, Serial0/0/1

C 172.16.3.0/24 is directly connected, FastEthernet0/0

Chúng ta thấy RouterX có 3 mạng kết nối trực tiếp, cả 3 mạng đều là subnet thuộc mạng Classful tương ưng 172.16.0.0/16 vì thế đều là Child route

Trong trường hợp triển khai VLSM, mỗi child route sẽ có 1 subnet mask đi kèm và khác nhau. Thay đổi thứ 2 đó là thông tin VLSM được thể hiện tại bản ghi parent route, variably subnetted và chỉ rõ số lượng subnet mask, trong trường hợp trên là 2 (2 masks)

Bất cứ khi nào xuất hiện ít nhất 2 tuyến đường có mạng đích là 2 subnet với 2 Subnet mask khác nhau, thuộc cùng mạng Classful thì 1 Parent route tương ứng với VLSM được tạo ra. Lúc này Parent route sẽ sử dụng subnet mask của mạng Classful chứ không phải Subnet mask của các subnet như trường hợp Classful Addressing

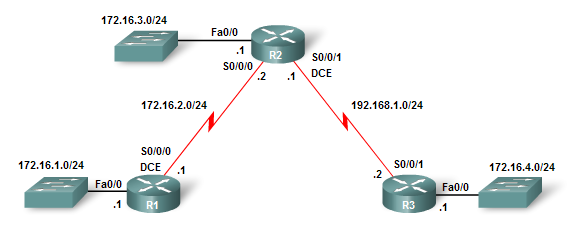


*Hình 8.5 – Classless routing cho phép Subnet mask khác nhau  
do đó parent route chỉ xác định subnet mask mặc định (/16)*

Tương tự, chúng ta cũng có 3 tuyến child route trong trường hợp này là tuyến cấp 2, và đồng thời là tuyến Ultimate

## Quá trình tra cứu bảng định tuyến

Quay lại với topology của ví dụ thứ nhất, với 3 Router R1, R2 và R3. Trong ví dụ trước, với bảng định tuyến tương ứng với Classful Addressing, chúng ta đã kiểm tra cấu trúc bảng định tuyến trên R2. Bây giờ chúng ta sẽ triển khai 1 giao thức định tuyến Classful Routing để kiểm tra việc trao đổi bản tin tuyến đường:



**Cấu hình khởi tạo:**

Trên R1:

R1(config)#router rip

R1(config-router)#network 172.16.0.0

Trên R2

R2(config)#router rip

R2(config-router)#network 172.16.0.0

R2(config-router)#network 192.168.1.0

Trên R3

R3(config)#router rip

R3(config-router)#network 192.168.1.0

R3(config-router)#network 172.16.0.0

**Kiểm tra bảng định tuyến hiện tại:**

Router R1

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 3 subnets

C 172.16.1.0 is directly connected, FastEthernet0/0

C 172.16.2.0 is directly connected, Serial0/0/0

R 172.16.3.0 [120/1] via 172.16.2.2, 00:00:01, Serial0/0/0

R 192.168.1.0/24 [120/1] via 172.16.2.2, 00:00:01, Serial0/0/0

Router R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 3 subnets

R 172.16.1.0 [120/1] via 172.16.2.1, 00:00:10, Serial0/0/0

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.3.0 is directly connected, FastEthernet0/0

C 192.168.1.0/24 is directly connected, Serial0/0/1

Router R3

R3#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

172.16.0.0/24 is subnetted, 1 subnets

C 172.16.4.0 is directly connected, FastEthernet

C 192.168.1.0/24 is directly connected, Serial0/0/1

**Đánh giá**

Topology trên cố gắng triển khai Classful Routing với thiết kế mạng không liên tục (Discontigous). Mạng Discontigous là mạng trong đó 2 subnet của 1 mạng major bị ngăn cách bởi 1 mạng major khác. Trong ví dụ trên, subnet 172.16.4.0/24 bị ngăn cách với các subnet 172.16.1.0/24, 172.16.2.0/24 và 172.16.3.0/24 bởi mạng 192.168.1.0/24 là 1 mạng major khác. Xem xét bảng định tuyến trên 3 Router, chúng ta thấy các vấn đề:

* Các Router R1 và R2 không có tuyến đường tới mạng 172.16.4.0/24 trên R3
* Router R3 không có tuyến đường tới các mạng 172.16.1.0/24, 172.16.2.0/24 và 172.16.3.0/24 trên R1 và R2

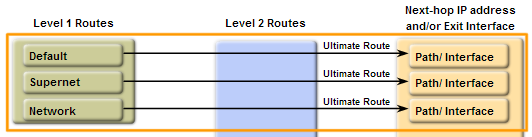
Tìm hiểu cách Router tra cứu bảng định tuyến để xác định lý do vì sao Classful Routing không thể triển khai trong mạng không liên tục. Chúng ta xem xét các câu hỏi sau:

* 1. Router sẽ làm gì khi nhận 1 gói tin IP, đọc thông tin địa chỉ đích và tra cứu bảng định tuyến ?
  2. Router sẽ lựa chọn 1 tuyến đường là tuyến đường tốt nhất như thế nào ?
  3. Ý nghĩa của Subnet mask trong tiến trình tra cứu bảng định tuyến
  4. Router sẽ quyết định sử dụng tuyến Supernet và tuyến mặc định (default route) như thế nào khi không tìm được tuyến đường chi tiết (tuyến đường cấp 2)

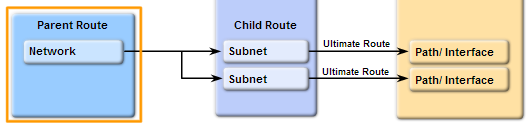
### Thứ tự tra cứu bảng định tuyến:

Khi 1 Router nhận 1 bản tin với địa chỉ IP đích xác định, Router sẽ tra cứu bảng định tuyến theo thứ tự sau:

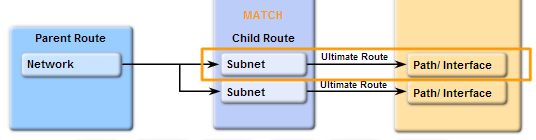
* 1. **Bước 1 -** Router xác định tuyến đường cấp 1 phù hợp nhất địa chỉ IP đích. Các tuyến đường cấp 1 bao gồm Default route, Supernet và tuyến mạng (Network route)
     + Nếu tuyến đường phù hợp nhất là 1 tuyến Ultimate, có thể là 1 tuyến tới 1 mạng Classful, Supernet hoặc default route, thì tuyến cấp 1 phù hợp nhất trong số đó sẽ được sử dụng để chuyển tiếp gói tin



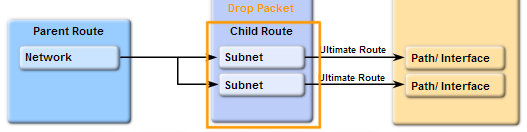
* + - Nếu tuyến đường phù hợp nhất là 1 tuyến Parent route cấp 1, Router chuyển sang bước 2



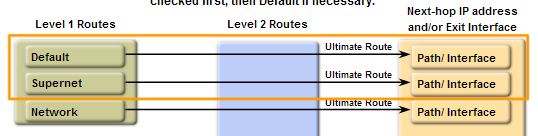
* 1. **Bước 2 –** Router sẽ tra cứu tuyến Child route, hay mạng con của tuyến Parent route phù hợp nhất để tìm ra tuyến Child route phù hợp nhất
     + Nếu có tuyến Child route phù hợp nhất, tuyến đó sẽ được sử dụng để chuyển tiếp gói



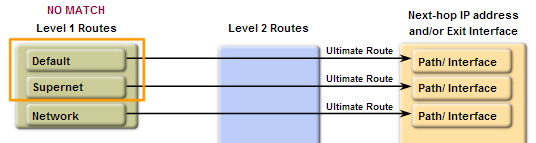
* + - Nếu không có tuyến Child route cấp 2 phù hợp, Router sẽ chuyển sang bước tra cứu thứ 3
  1. **Bước 3 –** Router lúc này sẽ xác định hành vi định tuyến dựa trên Classful Routing hoặc Classless Routing
     + Nếu là Classful Routing, Router sẽ ngừng tra cứu và loại bỏ (drop) gói tin



* + - Nếu là Classless Routing, Router sẽ quay lại với các tuyến cấp 1 – Nếu có tuyến Supernet phù hợp thì tuyến Supernet đó sẽ được dùng để chuyển tiếp gói. Nếu không có tuyến Supernet nhưng có default route, thì tuyến mặc định sẽ được sử dụng



* 1. **Bước 4 -** Trong trường hợp không có tuyến Supernet phù hợp hoặc không có default route, Router sẽ drop gói tin đó

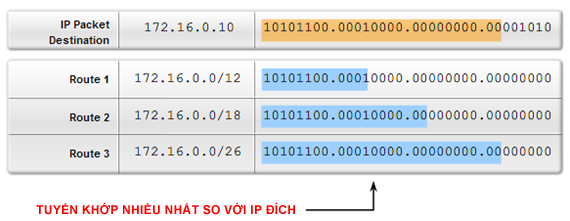


Quá trình tra cứu ở trên chỉ bao gồm việc xác định tuyến đường tối ưu khớp với địa chỉ mạng đích trong gói tin IP. Để có thể chuyển tiếp gói, Router sẽ từ tuyến đường tốt nhất thực hiện tra cứu đệ quy (có thể 1 lần hoặc nhiều lần) để xác định ra giao diện chuyển tiếp gói. Tra cứu đệ quy đã được đề cập trong baI định tuyến tĩnh

### Tuyến phù hợp nhất – So sánh chiều dài khớp (longest match)

Trong các bước tra cứu phía trên, chúng ta đã đề cập nhiều lần tới “tuyến phù hợp nhất” – Giữa tuyến Supernet và default route, tuyến nào được xem là phù hợp nhất. Giữa các subnet của 1 Parent route, tuyến nào được xem là phù hợp nhất ?

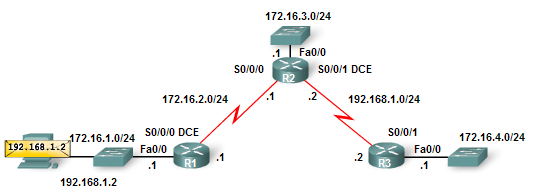
Để xác định 1 tuyến đường phù hợp nhất đối với địa chỉ IP đích, chúng ta sẽ so sánh các bit giống nhau giữa địa chỉ IP đích đó và địa chỉ mạng của mỗi tuyến đường. **Nếu tuyến đường nào có mạng đích có số lượng bit trùng nhiều nhất (so sánh từ trái qua) với địa chỉ IP đích thì tuyến đường đó được xem là phù hợp nhất**. Tuyến đường đó được là *Longest match route* (tuyến khớp dài nhất)



*Hình 8.6 – Số bit trùng nhiều nhất, và nhớ là đếm từ trái qua*

Trong ví dụ trên, 1 gói tin có địa chỉ IP đích 172.16.0.10, Router sẽ thực hiện tra cứu bảng định tuyến. Trong tất cả các tuyến Ultimate bao gồm cả tuyến cấp 1 và cấp 2, Router tìm thấy 3 tuyến đường (route 1 – tuyến Supernet, route 2 – Child route, route 3 – Child route) tương ứng có bit phù hợp với địa chỉ IP đích. Router sẽ thực hiện so sánh số bit khớp. Trong đó route 1 khớp 12 bit đầu tiên, route 2 khớp 18 bit đầu tiên còn route 3 khớp 26 bit đầu tiên. Vậy trong trường hợp này route 3 được lựa chọn là tuyến phù hợp nhất (longest match route) để chuyển tiếp gói

Ví dụ 2, chúng ta quay lại với topology ban đầu, xem xét việc tra cứu bảng định tuyến khi PC1 muốn gửi 1 gói tin tới địa chỉ 192.168.1.2. PC1 đóng gói gói tin IP.



**Bước 1 –** PC1 cố gắng gửi bản tin tới R1, do địa chỉ đích thuộc 1 mạng khác

**Bước 2 –** Router R1 đọc bản tin IP, lấy địa chỉ IP đích và tra cứu bảng định tuyến:

172.16.0.0/24 is subnetted, 3 subnets

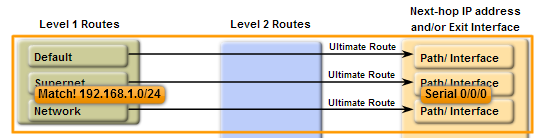
C 172.16.1.0 is directly connected, FastEthernet0/0

C 172.16.2.0 is directly connected, Serial0/0/0

R 172.16.3.0 [120/1] via 172.16.2.2, 00:00:01, Serial0/0/0

R 192.168.1.0/24 [120/1] via 172.16.2.2, 00:00:01, Serial0/0/0

Router R1 sẽ bắt đầu với các tuyến đường cấp 1, trong đó tuyến [R]: 192.168.1.0/24 được xác định là tuyến phù hợp nhất (đây là 1 tuyến mạng cấp 1 – Ultimate route). Vì thế tuyến [R]: 192.168.1.0/24 được sử dụng để định tuyến. Router thực hiện tra cứu đệ quy để xác định nguồn học qua giao thức RIP, IP next-hop của hàng xóm là 172.16.2.2 và giao diện chuyển tiếp gói là Serial0/0/0



*Hình 8.7 – So sánh khớp bắt đầu từ tuyến cấp 1*

**Bước 3 –** Router R1 đóng gói và gửi gói tin IP với địa chỉ IP đích giữ nguyên sang cho Router R2

**Bước 4 –** Router R2 đọc bản tin IP nhận từ interface Serial0/0/0, lấy địa chỉ IP đích và tra cứu bảng định tuyến

172.16.0.0/24 is subnetted, 3 subnets

R 172.16.1.0 [120/1] via 172.16.2.1, 00:00:10, Serial0/0/0

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.3.0 is directly connected, FastEthernet0/0

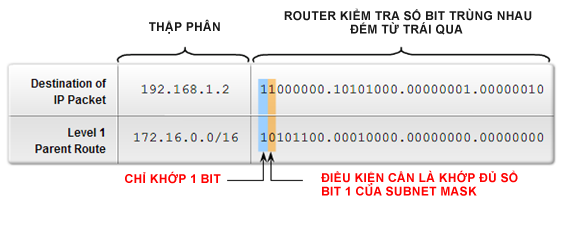
C 192.168.1.0/24 is directly connected, Serial0/0/1

Tương tự Router R1, R2 bắt đầu với các tuyến cấp 1 và xác định tuyến phù hợp nhất là [C] : 192.168.1.0/24. Router R2 thực hiện tra cứu đệ quy và xác định giao diện chuyển tiếp gói là Serial0/0/1. Chính là địa chỉ IP của giao diện Serial0/0/1 của R2

**Bước 5 –** Router R2 đóng gói tin và gửi gói tin phàn hồi về cho PC1 tại giao diện Serial0/0/0

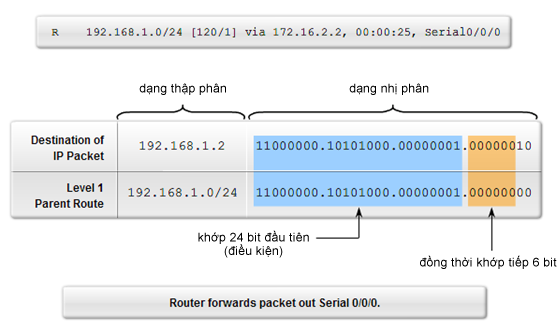
Chúng ta thấy Router R1 và R2 xác định tuyến đường tới mạng 192.168.1.0/24 là phù hợp nhất (tốt nhất) để chuyển tiếp gói tin. Chúng ta thấy Router so sánh bằng cách chuyển địa chỉ IP sang chuỗi 32 bit nhị phân, đặt thành hàng và bắt đầu so sánh số bit khớp từ trái qua. Vậy subnet mask có ý nghĩa gì trong việc lựa chọn tuyến đường tốt nhất ? Nếu subnet mask không có ý nghĩa gì, vậy thì tại sao cần đưa ra cấu trúc bảng định tuyến khác nhau cho Classful routing và Classless Routing! Thực tế, điều kiện số bit khớp nhiều nhất là *điều kiện đủ* để 1 tuyến đường được lựa chọn là phù hợp nhất, nhưng Subnet mask đóng vai trò là *điều kiện cần* để đảm bảo quá trình tra cứu diễn ra đúng logic và hợp lý. Hãy so sánh tuyến 172.16.1.0/24 và tuyến default route, tuyến nào sẽ là tối ưu hơn trong trường hợp trên khi địa chỉ IP đích trỏ 192.168.1.0, nhưng trên Router R1 không có mạng 192.168.1.0/24.

Giả sử, chúng ta so sánh sô bit trùng nhiều nhất của default route và tuyến 172.16.1.0 khi so sánh với địa chỉ IP đích 192.168.1.2. Tuyến default route không khớp bit nào, trong khi tuyến 172.16.1.0 khớp 1 bit đầu tiên



*Hình 8.8 – Subnet mask đưa ra số bit tối thiểu phải khớp  
vậy mạng càng tổng quát càng dễ qua vòng điều kiện cần*

Vậy thì gói tin sẽ được định tuyến vào mạng 172.16.1.0 ? Sẽ thật là không hợp lý khi 1 gói tin được gửi tới 1 mạng khác mạng của địa chỉ IP đích trong gói tin. Subnet mask đưa ra 1 điều kiện cần để 1 tuyến đường được xem xét trước khi so sánh: **Subnet mask xác định số bit tối thiểu phải khớp của 1 tuyến đường đối với 1 địa chỉ IP đích, mà tuyến đường đó cần phải khớp trước khi có thể so sánh số bit trùng nhiều nhất với các tuyến đường khác.**

Như vậy, tuyến 172.16.1.0/24 nhất định phải khớp 24 bit đầu tiên trước. Lúc đó tuyến 172.16.1.0 chỉ khớp bit đầu tiên nên sẽ không được xem xét định tuyến. Ngược lại, tuyến default route với subnet mask bằng 0.0.0.0 do đó không cần phải khớp bất kỳ bit nào trước khi được so sánh. Vậy trong trường hợp trên, tuyến default route sẽ được lựa chọn là tốt hơn so với tuyến 172.16.1.0/24

*Hình 8.9 – Tuyến đường cần phải khớp số bit tối thiểu (bằng subnet mask) trước khi so sánh*

### Tra cứu bảng định tuyến, khớp hay không khớp

Tuy nhiên, do cấu trúc của bảng định tuyến hơi khác 1 chút giữa Classful Addressing và Classless Addressing nên sẽ có 1 chút khác nhau trong điều kiện cần:

1. **Với Classful Addressing**

Subnet mask của child route được mang theo trong bản ghi của parent route. Do đó subnet mask gốc tương ứng với lớp của Parent route sẽ được sử dụng để xác định điều kiện cần trên. Đó là điều kiện cần. Sau đó Router sẽ sử dụng subnet mask đi kèm trong bản ghi của parent route để xác định điều kiện cần thứ 2 cho các child route. Sẽ không có điều kiện đủ với bảng Classful, vì chúng ta không có nhiêu subnet mask cho các mạng con (Classful Routing không hỗ trợ VLSM)

Ví dụ:

Địa chỉ IP đích: 172.16.2.1

172.16.0.0/24 is subnetted, 3 subnets

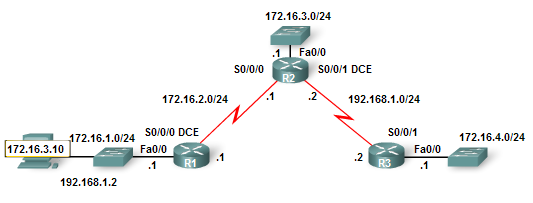
C 172.16.1.0 is directly connected, FastEthernet0/0

C 172.16.2.0 is directly connected, FastEthernet0/1

🡪 Xét điều kiện cần – Subnet mask gốc của tuyến parent bằng 16, vì 172.16.0.0 là địa chỉ lớp B. Như vậy cả 2 tuyến 172.16.1.0 và 172.16.2.0 đều thỏa mãn

🡪 Xét điều kiện cần thứ 2 - Subnet mask đi kèm trong bản ghi của parent route là /24, lúc này chỉ có tuyến 172.16.2.0 mới thỏa mãn (khớp 24 bit), trong khi tuyến 172.16.1.0 chỉ khớp 22 bit. Do đó tuyến 172.16.2.0 sẽ được dùng để chuyển tiếp cho gói tin trên

Để kiêm tra các nguyên tắc trên, chúng ta xét ví dụ thứ 3, chúng ta quay lại với topology đầu tiên, trường hợp này PC1 muốn gửi 1 gói tin tới địa chỉ IP đích 172.16.3.10



**Bước 1 -** Tương tự như ví dụ 2, PC1 sẽ gửi gói tin ra default Gateway là R1

**Bước 2 –** Router R1 sẽ đọc gói tin, xác định địa chỉ IP đích và tra bảng định tuyến

172.16.0.0/24 is subnetted, 3 subnets

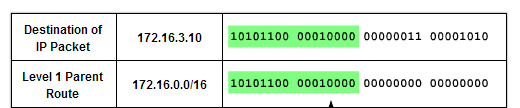
C 172.16.1.0 is directly connected, FastEthernet0/0

C 172.16.2.0 is directly connected, Serial0/0/0

R 172.16.3.0 [120/1] via 172.16.2.2, 00:00:01, Serial0/0/0

R 192.168.1.0/24 [120/1] via 172.16.2.2, 00:00:01, Serial0/0/0

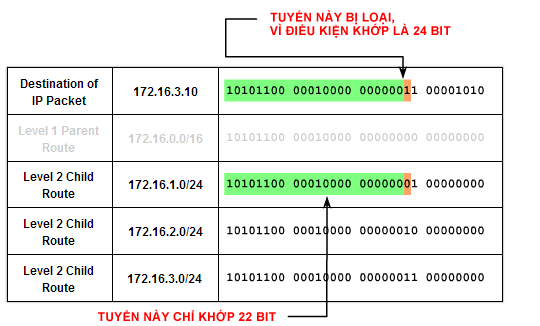
Router R1 sẽ thực hiện kiểm tra điều kiện cần bằng cách kiểm tra số bit khớp giữa đia chỉ IP đích và địa chỉ parent route 172.16.0.0. Parent route 172.16.0.0 là địa chỉ lớp B, do đó 16 bit đầu cần phải khớp giữa địa chỉ IP đích và parent route. Điều kiện cần thứ nhất vượt qua vì parent route khớp 22 bit (so với 16 bit yêu cầu)



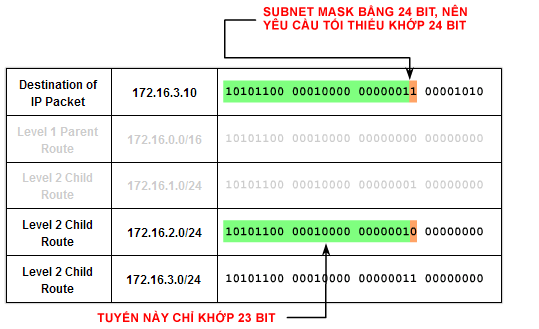
*Hình – điều kiện cần đủ, chỉ cần 16 bit khớp*

Router R1 sẽ thực hiện kiểm tra điều kiện cần thứ 2 để chọn lựa tuyến Child/Ultimate route thuộc tuyến parent 172.16.0.0 sẽ được dùng để chuyển tiếp gói. Lúc này subnet mask tương ứng /24 sẽ được sử dụng để đánh giá:

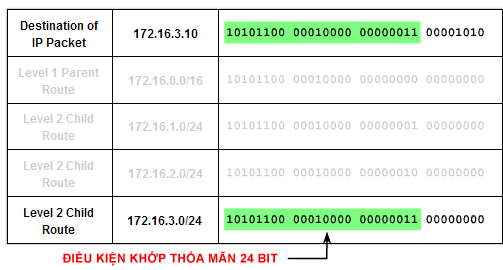
* So với tuyến 172.16.1.0 – không thỏa mãn



* So với tuyến 172.16.2.0 – không thỏa mãn

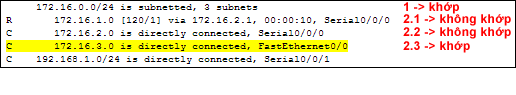


* So với tuyến 172.16.3.0 – thỏa mãn



Như vậy tuyến child route 172.16.3.0/24 sẽ được tham gia so sánh khớp số bit dài nhất để lựa chọn tuyến đường tốt nhất. Trong trường hợp này, không có tuyến đường nào khác “cạnh tranh” với 172.16.3.0/24 vì thế tuyến [R] : 172.16.3.0 sẽ được sử dụng để định tuyến. Tương tự Router R1 thực hiện tra cứu dệ quy để xác định giao diện chuyển tiếp gói. Trong trường hợp trên là đẩy tới next-hop 172.16.2.2 hay đi ra interface Serial0/0/0

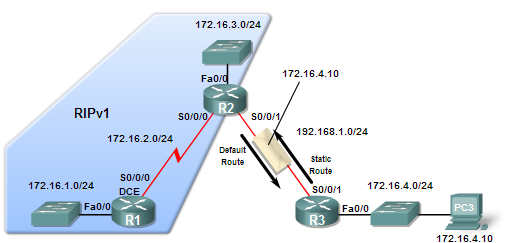
**Bước 3 –** Router R2 đọc gói tin, làm tương tự như R1 trong bước 2, và xác định ra tuyến [C] : 172.16.3.0 sẽ được sử dụng để định tuyến. Đây là mạng kết nối trực tiếp tại interface FastEthernet0/0 của Router R2 nên R2 sẽ gửi gói tin IP này ra interface FastEthernet0/0



1. **Với Classless Addressing**

Tương tự Classful Addressing, Classless Addressing cũng trải qua 2 lần kiểm tra điều kiện cần, lần thứ nhất sử dụng subnet mask mặc định (tương ứng với lớp) của địa chỉ parent (classful) route. Vì subnet mask của child route mang kèm theo trong bản ghi của từng child route. Do đó các child route sẽ được so sánh trực tiếp Subnet mask để kiểm tra điều kiện cần thứ 2. Sau lượt kiểm tra thứ 2 này, các tuyến thỏa mãn mới được so sánh với nhau *điều kiện đủ* (khớp nhiều bit nhất) để chọn ra tuyến đường tối ưu.

Để kiểm tra nguyên tắc trên, chúng ta sử dụng ví dụ dưới đây, cho topology:



**Bài toán:** Xem xét quá trình lựa chọn tuyến đường trong bảng định tuyến trên Router R2 và R3 khi có 1 gói tin gửi từ mạng LAN với địa chỉ IP: 

**Hiện trạng bảng định tuyến:**

Trên Router R2

172.16.0.0/24 is subnetted, 3 subnets

R 172.16.1.0 [120/1] via 172.16.2.1, 00:00:12, Serial0/0/0

C 172.16.2.0 is directly connected, Serial0/0/0

C 172.16.3.0 is directly connected, FastEthernet0/1

C 192.168.1.0/24 is directly connected, Serial0/0/1

S\* 0.0.0.0/0 is directly connected, Serial0/0/1

Trên Router R3

172.16.0.0/16 is variably subnetted, 3 subnets, 2 masks

C 172.16.4.0/24 is directly connected, FastEthernet0/0

S 172.16.0.0/16 is directly connected, Serial0/0/1

S 172.16.3.0/24 is directly connected, Serial0/0/1

C 192.168.1.0/24 is directly connected, Serial0/0/1

Chúng ta sẽ xem xét cách Router tra cứu bảng định tuyến theo 2 chiều, chiều gửi đi từ R2, và chiều phản hồi từ R3

**Bước 1 –** Router R2 thực hiện tra cứu để chọn đường đi, địa chỉ IP đích 172.16.4.10. Chúng ta sẽ xem cách R2 tra cứu từng tuyến đường:

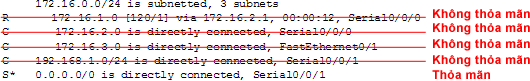
Điều kiện cần 1 – kiểm tra từng tuyến cấp 1 (trên cùng là IP đích sẽ so sánh)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| thập phân | Dạng nhị phân | Khớp | Cần khớp | Trạng thái |
| 172.16.4.10 | 10101100.00010000.00000100.00001010 |  |  |  |
| 172.16.0.0 | 10101100.00010000.00000000.00000000 | 21 | 16 | Thỏa mãn |
| 192.168.1.0 | 11000000.10101000.00000001.00000000 | 1 | 24 | Loại |
| 0.0.0.0 | 00000000.00000000.00000000.00000000 | 0 | 0 | Thỏa mãn |

Điều kiện cần 2 - Như vậy sau lần kiểm tra thứ nhất, ta có tuyến cấp 1: 172.16.0.0 thỏa mãn và tuyến default 0.0.0.0 thỏa mãn. Chúng ta bắt đầu bước kiểm tra thứ 2, kiểm tra tuyến cấp 2/ultimate:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| thập phân | Dạng nhị phân | Khớp | Cần khớp | Trạng thái |
| 172.16.4.10 | 10101100.00010000.00000100.00001010 |  |  |  |
| 172.16.1.0 | 10101100.00010000.00000001.00000000 | 21 | 24 | Loại |
| 172.16.2.0 | 10101100.00010000.00000010.00000000 | 21 | 24 | Loại |
| 172.16.3.0 | 10101100.00010000.00000011.00000000 | 21 | 24 | Loại |
| 0.0.0.0 | 00000000.00000000.00000000.00000000 | 0 | 0 | Thỏa mãn |

Vậy, sau lần kiểm tra thứ 2, chỉ còn tuyến mặc định 0.0.0.0/0 còn lại. Vậy Router R2 sẽ sử dụng tuyến mặc định để chuyển tiếp gói tin



*Hình 8.10 – Chỉ còn tuyến default thỏa mãn*

**Bước 2 –** Router R2 đóng gói gói tin và chuyển sang Router R3

**Bước 3 –** Router R3 nhận bản tin từ R2 , Router R3 tra bảng định tuyến và đẩy gói tin ra PC3

**Bước 4 –** PC3 đọc và xử lý gói tin, sau đó đóng gói lại và gửi bản tin phản hồi tới Gateway là Router R3 

**Bước 5 –** Router R3 nhận gói tin, thực hiện tra bảng định tuyến. Tương tự như trên, chúng ta sẽ xét từng điều kiện với từng tuyến đường

Điều kiện thứ nhất – Chúng ta kiểm tra các tuyến cấp 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| thập phân | Dạng nhị phân | Khớp | Cần khớp | Trạng thái |
| 172.16.3.1 | 10101100.00010000.00000011.00000001 |  |  |  |
| 172.16.0.0 | 10101100.00010000.00000000.00000000 | 22 | 16 | Thỏa mãn |
| 192.168.1.0 | 11000000.10101000.00000001.00000000 | 1 | 24 | Loại |

Sau lần kiểm tra thứ nhất, chỉ có tuyến cấp 1 172.16.0.0 thỏa mãn, chúng ta bươc sang bước kiểm tra thứ 2, đó là kiểm tra các tuyến cấp 2/ultimate/child route của tuyến cấp 1 trên:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| thập phân | Dạng nhị phân | Khớp | Cần khớp | Trạng thái |
| 172.16.3.1 | 10101100.00010000.00000011.00000001 |  |  |  |
| 172.16.4.0 | 10101100.00010000.00000100.00000000 | 21 | 24 | Loại |
| 172.16.0.0 | 10101100.00010000.00000000.00000000 | 22 | 16 | Thỏa mãn |
| 172.16.3.0 | 10101100.00010000.00000011.00000000 | 24 | 24 | Thỏa mãn |

Sau lần kiểm tra thứ 2, chúng ta thấy có 2 tuyến thỏa mãn là tuyến tĩnh 172.16.0.0/16 và tuyến tĩnh 172.16.3.0/24. 2 Điều kiện cần đã đều thỏa mãn, 2 tuyến đường này sẽ bước vào pha kiểm tra điều kiện đủ, hay kiểm tra số bit khớp dai nhất

|  |  |  |  |
| --- | --- | --- | --- |
| thập phân | Dạng nhị phân | Khớp | Trạng thái |
| 172.16.3.1 | 10101100.00010000.00000011.00000001 |  |  |
| 172.16.0.0 | 10101100.00010000.00000000.00000000 | 22 |  |
| 172.16.3.0 | 10101100.00010000.00000011.00000000 | 24 | Khớp nhiều hơn |

Ta thấy, tuyến đường 172.16.3.0/24 khớp 24 bit, trong khi tuyến 172.16.0.0/16 chỉ khớp 22 bit. Do đó tuyến tĩnh 172.16.3.0/24 sẽ được lựa chọn để chuyển tiếp gói

**Bước 6 –** Router R3 đóng gói lại bản tin tương ứng với tuyến tĩnh 172.16.3.0/24 và đẩy ra outgoing interface (trường hợp này là Serial 0/0/1)



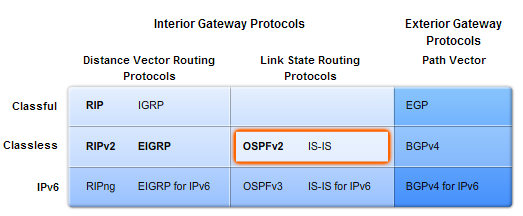
# ****CHƯƠNG 9 - GIAO THỨC ĐỊNH TUYẾN LINK-STATE****

Định tuyến động (Dynamic Routing) được chia làm 2 nhóm dựa vào cách tiếp cận cập nhật thông tin định tuyến. Chúng ta đã đề cập tới nhóm thứ nhất – Distance-vector trong chương 4 với đại diện RIPv1 và RIPv2, chương 9 đề cập tới nhóm giao thức còn lại – Link-state với giao thức đại diện là OSPF

Các giao thức Link-State thường được gọi là giao thức SPF (Shortest Path First), vì các giao thức này được phát triển dựa trên thuật toán SPF của Edsger Dijkstra. Cac giao thức định tuyến thuộc họ Link-State phổ biến nhất:

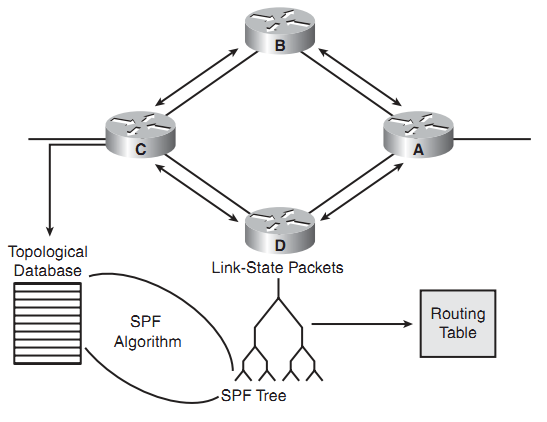
* Open Shortest Path First (OSPF)
* Intermediate System-to-Intermediate System (IS-IS)

Ngoài các giao thức tren được thiết kế dành cho hệ thống mạng IP. Với những mạng chạy giao thức khác ngoài IP, một vài giao thức định tuyến Link-State khác có thê kể tới như DNS Phase V của DEC và NLSP của Novell



## Giới thiệu giao thức Link-State

Các giao thức định tuyến Link-State sử dụng thuật toán định tuyến Link-State được biết đến với cái tên SPF (Shortest Path First) làm nhiệm vụ duy trì 1 bảng cơ sở dữ liệu phức tạp mang thông tin topology trong mạng. Khác biệt hoàn toàn về cách tiếp cận so với các giao thức Distance-vector, các Router chạy Link-State đều biết về tất cả các node (các Router khác trong cùng miền định tuyến) và cách các Router khác kết nối với nhau cũng như các interface kết nối trực tiếp trên mỗi node



*Hình 9.1 – thuật toan SPF*

Định tuyến Link-state sử dụng các bản tin quảng bá Link-State LSA (Link-state Advertisement) để dựng nên một cơ sở dữ liệu về topology logic, sử dụng thuật toán SPF để xây dựng nên cây SPF – tập hợp các nhánh để xác định các tuyến đường tới tất cả các nhanh khác trong mạng và “chiều dài” của mỗi nhanh. Và cuối cùng, sử dụng bảng định tuyến để thực hiện nhiệm vụ định tuyến

Các giao thức Link-state được thiết kế để giải quyết những hạn chế của giao thức Distance-vector. Khi mạng mở rộng với quy mô lớn dần, các giao thức Link-State trở nên phù hợp hơn với 1 vài lý do sau:

* Giao thức Link-State phản hồi nhanh với thay đổi trong mạng, sử dụng cập nhật kích hoạt để đảm bảo hội tụ nhanh chóng mỗi khi có thay đổi xảy ra
* Sử dụng cập nhật định kỳ để làm tươi định tuyến trong toàn mạng sau 1 chu kỳ đủ dài (không làm giảm hiệu năng mạng)
* Sử dụng cơ chế Hello để duy trì kết nối và làm nhiệm vụ keepalive giữa các Router chạy Link-State.
* Mạng chạy giao thức Link-state có thể được chia nhỏ dưới dạng các miền (area) với cấu trúc phân cấp để tăng khả năng quản lý và mở rộng
* Giao thức Link-State luôn hỗ trợ Classless Addressing, hay VLSM và CIDR
* Giao thưc Link-State hỗ trợ nhiều kỹ nâng cao hiệu quả hơn như lọc tuyến đường, gộp tuyến đường,…

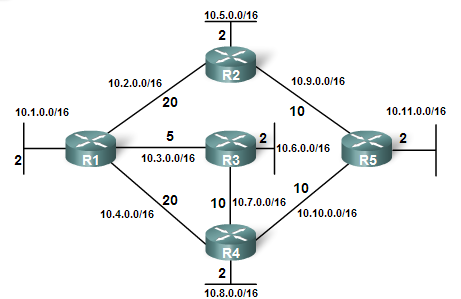
## Các tiến trình được sử dụng bởi giao thức Link-State

Các tiến trình cơ bản được thực hiện bởi giao thức Link-State để hội tụ mạng bao gồm:

* 1. **Học từ kết nối trực tiếp** – Xác định các interface (link) tham gia định tuyến đang UP và trạng thái của các link đó
  2. **Thiết lập các mối quan hệ cần thiết** với cac Router chia sẻ cùng môi trường truyền (cùng nằm trên 1 link) – Trao đổi bản tin Hello với các Router chạy cùng giao thức Link-State và thiết lập các mối quan hệ kết nối trực tiếp
  3. **Xây dựng các bản tin Link-State** (LSP – Link-state Packet) – Ghi các thông tin cần thiết vào từng loại bản tin cập nhật định tuyến. Các thông tin này là thông tin trạng thái của link
  4. **Gửi bản tin update sang cho tất cả Router hàng xóm** – Router gửi tất cả bản tin cập nhật định tuyến sang tất cả Router hàng xóm chạy cùng giao thức, đảm bảo tất cả node trong miền định tuyến đều có 1 bản sao của bản tin LSP để xây dựng bảng cơ sở dữ liệu
  5. **Xây dựng bảng cơ sở dữ liệu và tính toán đường đi tốt nhất** – Dựa trên các bản tin LSP, Router xây dựng lại topology logic về toàn mạng, thông qua thuật toán SPF, Router xác định đường tới tất cả các mạng và xác định tuyến đường tối ưu đến từng mạng để cập nhật vào bảng định tuyến

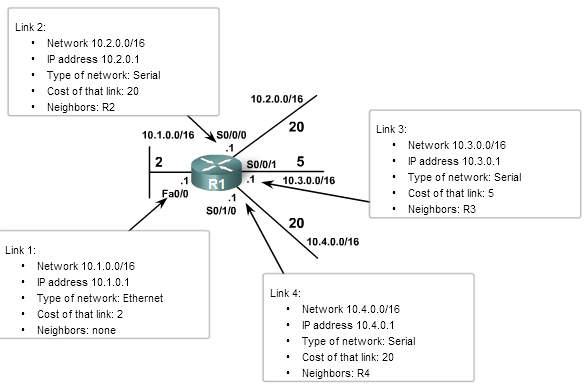
Chúng ta sẽ xem xét từng bước 1 trong 5 bước trên để hiểu về cách thức trao đổi bản tin và hội tụ mạng của của giao thức định tuyến Link-state

Để dễ tiếp cận vấn đề, chúng ta sẽ xem xét topology mạng dưới đây và cách các Router trong mạng triển khai 1 giao thức Link-State:



### Học từ kết nối trực tiếp

Khi một interface được kích hoạt tham gia vào định tuyến động, Router sẽ ngay lập tức đưa mạng kết nối trực tiếp gắn với interface đó vào giao thức định tuyến. Đối với giao thức Link-State, mỗi interface được xem là 1 link và trạng thái của Link thì không chỉ có thông tin về mạng kết nối trực tiếp gắn với link đó. Giả sử với topology trên chúng ta kích hoạt định tuyến Link-State trên tất cả interface của Router R1



*Hình 9.1 – Router tự câp nhật trạng thái trên các link kết nối trực tiếp*

Trong ví dụ trên, R1 đưa 5 link tương ứng với 5 interface của mình vào định tuyến

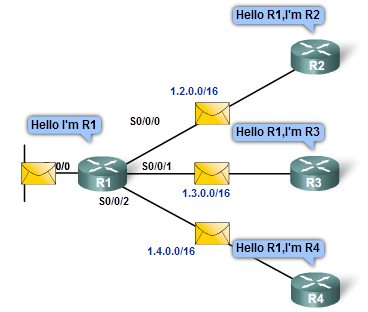
* Interface FastEthernet 0/0 nối tới mạng 10.1.0.0/16
* Interface Serial 0/0/0 nối tới mạng 10.2.0.0/16
* Interface Serial 0/0/1 nối tới mạng 10.3.0.0/16
* Interface Serial 0/0/2 nối tới mạng 10.4.0.0/16

Đối với mỗi Link sẽ có trạng thái link bao gồm các thông tin:

* Địa chỉ IP và Subnet mask của từng link
* Loại mạng trên link, chẳng hạn link Ethernet là loại mạng Broadcast, link Serial là loại mạng Point-to-Point (điểm – điểm) hoặc Point-to-Multipoint (điểm – đa điểm)
* Cost trên từng link, cost được sử dụng để tính metric các tuyến đường
* Các Router hàng xóm chia sẻ cùng link kết nối

### Thiết lập các quan hệ cần thiết

Các Router chạy giao thức Link-State sẽ sử dụng giao thức Hello để khám phá Router hàng xóm trên link, thiết lập mối quan hệ để sẵn sàng trao đổi thông tin định tuyến, thực hiện cơ chế keepalive để đảm bảo kết nối giữa các Router neighbor



*Hình 9.2 Giao thức Link-State sử dụng Hello để keepalive*

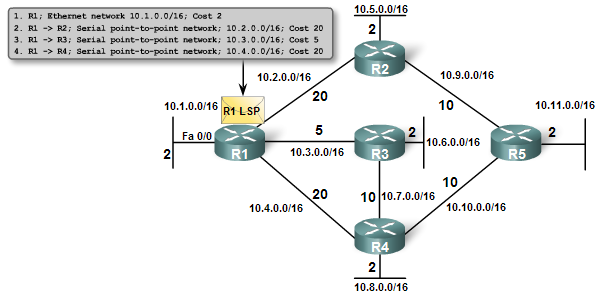
Khi 1 interface được kích hoạt định tuyến Link-State, Router sẽ gửi ra interface đó bản tin Multicast Hello để khám phá Router hàng xóm chạy cùng giao thức. Nếu link có Router hàng xóm chạy cùng giao thức định tuyến, nhận được bản tin Hello, Router hàng xóm đó sẽ gửi lại bản tin Hello để xác nhận. Nếu interface nào không nhận lại bản tin Hello phản hồi, Router sẽ không tiếp tục gửi bản tin Hello ra interface đó nữa. **Chức năng khám phá hàng xóm là chức năng thứ nhât của giao thức Hello**

Sau khi 2 Router xác nhận mối quan hệ hàng xóm, các Router chạy link-State sẽ tiếp tục trao đổi các bản tin Hello, các bản tin hello mang thông tin thuộc tính mà các Router cần phải khớp để có thể thiết lập mối quan hệ phụ cân – Adjacency. Mối quan hệ Adjacency là mối quan hệ điểm-điểm bắt buộc phải thiết lập trước khi các Router chạy Link-State có thể trao đổi bản tin định tuyến. Các thông tin thuộc tính có thể là thời gian định kỳ gửi ra Hello, loại mạng trên link, hoặc thông tin xác thực, chúng ta sẽ xem xét các thuộc tính này trong phần sau. **Chức năng thiết lập Adjacency là chức năng thứ 2 của giao thức Hello**

Sau khi thiết lập mối quan hệ Adjacencu, các Router chạy Link-State vẫn tiếp tục gửi ra bản tin Hello theo chu kỳ được thống nhất giữa 2 các hàng xóm trong lúc trao đổi thiết lập Adjacency. Bản tin hello lúc này để đảm bảo kết nối giữa các Router hàng xóm, nếu một interface không còn nhận được bản tin Hello, đồng nghĩa với Router hàng xóm trên interface đó đã rời khỏi miền định tuyến. Lúc này, **cơ chế keepalive là chức năng thứ 3 của giao thức Hello**

### Xây dựng bản tin Link-State (Link-state packet)

Sau khi thiết lập mối quan hệ Adjacency giứa các Router hàng xóm, Router chạy giao thức định tuyến Link-State sẽ chuẩn bị các bản tin Link-State (LSP) mang thông tin về các link kết nối trực tiếp để gửi sang hàng xóm



*Hình 9.3 – Giao thức link state tự tạo ra các bản tin cập nhật LSP*

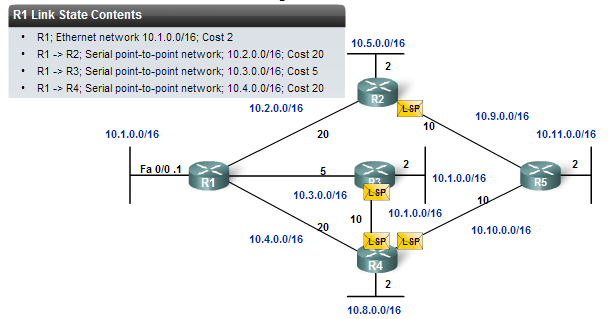
Tùy thuộc vào mỗi giao thức định tuyến Link-State mà cấu trúc và có nhiều hơn 1 loại bản tin trạng thái link LSP. Trong ví dụ trên chúng ta chỉ mô tả bản tin Link-State cơ bản nhất, mang thông tin trạng thái các link kết nối trực tiếp trên Router R1. Thông tin trạng thái trong bản tin LSP bao gồm:

* Router hàng xóm trên Link (nếu có) trong ví dụ trên link Ethernet của R1 không có Router hàng xóm
* Loại mạng
* Mạng kết nối trực tiếp trên từng link
* Cost của từng link

Thông tin này sẽ được truyền tới tất cả Router trong miền định tuyến, từ đó các Router khác (có thể hàng xóm (R2, R3, R4) hoặc không phải, chẳng hạn R5) sẽ có thể xác định được vị trí của Router R1 trong mạng, cách thức Router R1 kết nối tới các Router khác và cost để xây dựng cây SPF

### Flood bản tin update cho tất cả hàng xóm

Khái niệm “flood” – lụt bản tin, phản ánh đúng cách thức Router chạy Link-State trao đổi bản tin LSP. So với bản tin update của giao thức Distance-vector (bản tin update chỉ được gửi cho Router hàng xóm, sau đó Router hàng xóm sẽ cập nhật lại ra bản tin update mới rồi đẩy sang các Router xa hơn) bản tin LSP mang thông tin của cơ sở dữ liệu, vì thế để cấu trúc mạng được xây dựng lại đồng nhât giữa các Router, bản tin LSP được khởi tạo từ 1 Router phải được gửi tới tất cả Router khác trong mạng, mà không được có sự thay đổi hay chỉnh sửa nào.



*Hình 9.4 – Bản tin cập nhật LSP được lụt khắp mạng*

Tuy nhiên, do đặc điểm của lụt bản tin là chiếm dụng tài nguyên mạng rất nhanh trong 1 khoảng thời gian ngắn, nên các giao thức Link-State không gửi ra LSP định kỳ trong thời gian ngắn như Distance-vector. Bản tin LSP chỉ được gửi ra (flood) ngay tức thì trong các trường hợp:

* Quá trình khởi tạo định tuyến, các Router cập nhật cơ sở dữ liệu lân đầu tiên
* Bất cứ khi nào có sự thay đổi về link và trạng thái của link bất kỳ trong mạng, hoặc sự thiết lập mới hoặc ngắt kết nối neighbor
* Bắt đầu giai đoạn làm tươi bảng cơ sở dữ liệu, chu kỳ làm tươi với giao thức Link-State thường rất lâu, nên không thực sự làm ảnh hưởng hiệu năng trong mạng

Đề điều khiển tiến trình lụt bản tin, các trường dữ liệu khác được sử dụng, chẳng hạn chuỗi sequence-number được dùng để xác định 1 thông tin cập nhật cơ sở dữ liệu là cũ hay mới, hoặc thời gian sống – aging time – để xác định chu kỳ làm tươi

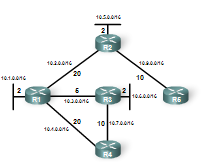
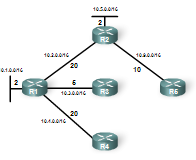
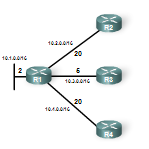
### Xây dựng bảng cơ sở dữ liệu và tính toán tuyến đường tốt nhất

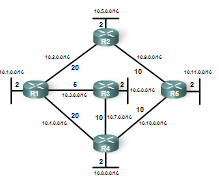
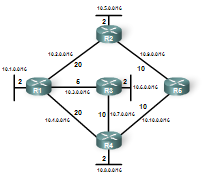
Bước cuối cùng sau khi 1 Router đã thu thập đủ các bản tin LSP từ tất cả Router chạy cùng giao thức định tuyến, Router đó sẽ dựa trên cơ sở dữ liệu đó để xây dựng nên cây SPF, cây SPF phản ảnh topology logic của mạng, vị trí các node trong mạng, link kết nối giữa các node, trạng thái link và cost trên từng link. Dựa vào đó Router luôn xác định được các tuyến đường tới 1 mạng đích (có thể có nhiêu hơn 1 tuyến đường tới 1 mạng đích) chiều dài của mỗi tuyến đường, và dễ dàng xác định được tuyến đường nào là tối ưu hơn. Đặc biệt quan trọng, dựa vào cây SPF, Router chạy Link-State không bao giờ phải lo lắng vấn đề loop, vì Router luôn biết được 1 bản tin khi được gửi ra theo 1 tuyến đường, thì sẽ đi qua những Router nào để tới được mạng đích. Đó chính là đặc điểm ưu việt nhất của các giao thức Link-State

Đối với ví dụ trên, chúng ta xem xét bảng cơ sở dữ liệu trên R1 sau khi thu thập đủ các bản tin LSP (mô tả phía dưới chỉ mang tính giới thiệu, không phải cách hiển thị của 1 giao thức định tuyến Link-State cụ thể)

|  |
| --- |
| **Bảng cơ sở dữ liệu Link-State trên R1** |
| * **LSP của R2** |
| * Link nối tới R1, mạng 10.2.0.0/16, cost = 20 * Link nối tới R5, mạng 10.9.0.0/16, cost = 10 * Link tới mạng 10.5.0.0/16, cost = 5 |
| * **LSP của R3** |
| * Link nối tới R1, mạng 10.3.0.0/16, cost = 5 * Link nối tới R4, mạng 10.7.0.0/16, cost = 10 * Link tới mạng 10.6.0.0/16, cost = 2 |
| * **LSP của R4** |
| * Link nối tới R1, mạng 10.4.0.0/16, cost = 20 * Link nối tới R3, mạng 10.7.0.0/16, cost = 10 * Link nối tới R5, mạng 10.10.0.0/16, cost = 10 * Link tới mạng 10.8.0.0/16, cost = 2 |
| * **LSP của R5** |
| * Link nối tới R2, mạng 10.9.0.0/16, cost = 10 * Link nối tới R4, mạng 10.10.0.0/16, cost = 10 * Link tới mạng 10.11.0.0/16, cost = 2 |
| * **Trạng thái link của R1** |
| * Link nối tới R2, mạng 10.2.0.0/16, cost = 20 * Link nối tới R3, mạng 10.3.0.0/16, cost = 5 * Link nối tới R4, mạng 10.4.0.0/16, cost = 10 * Link tới mạng 10.1.0.0/16, cost = 2 |

Ta thấy R1 lưu trữ tất cả LSP của các Router khác, cả hàng xóm (R2, R3, R4) lẫn không có kết nối trực tiếp (R5), thông tin lưu trữ là link và trạng thái các link kết nối trên mỗi Router, từ đó R1 dựng lại được topology logic của mạng, chúng ta xem R1 xây dựng topology dựa trên LSP như thế nào:





*Hình 9.5 – Các bước Router tái cấu truc topology*

Dựa trên topology logic của mạng, Router R1 xây dựng cây SPF (ví dụ dưới đây mô tả cây SPF, Router chạy Link-State thực sự không sử dụng 1 bảng riêng để lưu trữ cây SPF)

|  |  |  |
| --- | --- | --- |
| Mạng đích | SPF [Tuyến đường ngắn nhất] | Cost |
| R2 LAN | R1 🡪 R2 | 22 |
| R3 LAN | R1 🡪 R3 | 7 |
| R4 LAN | R1 🡪 R3 🡪 R4 | 17 |
| R5 LAN | R1🡪 R3 🡪 R4 🡪 R5 | 27 |

Với Link-State, số lượng hop không phải tiêu chí quan trọng, tối ưu tốc độ truyền dữ liệu trên từng link mới quan trọng, do đó cost được sử dụng để đánh giá độ hiệu quả của từng link. Với tuyến đường đến mạng LAN của R5, R1 có thể lựa chọn:

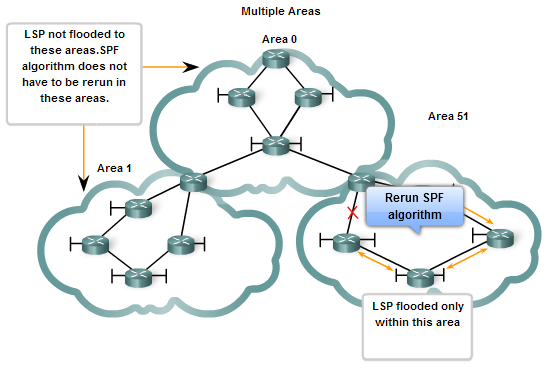
* Tuyến 1 : R1 🡪 R2 🡪 R5: 3 hop, cost = 32
* Tuyến 2 : R1 🡪 R4 🡪 R5 : 3 hop, cost = 32
* Tuyến 3 : R1 🡪 R3 🡪 R4 🡪 R5 : 4 hop, cost = 27

Tuy nhiên SPF sẽ lựa chọn 1 tuyến đường tối ưu dựa trên cost tốt nhất. Cost được tính dựa trên các thông số QoS như Băng thông, tuyến đường có băng thông tốt hơn sẽ được ưu tiên hơn. Việc tính cost là “end-to-end”, hay cost của 1 tuyến đường được tính dựa trên cost của tất cả các link gộp thành tuyến đường đó. Sau khi xây dựng được cây SPF hoàn chỉnh tới tất cả mạng đích, Router sẽ cập nhật các tuyến đường tốt nhất dựa trên cây SPF vào bảng định tuyến. Cost được sử dụng như metric

### Tổng kết

Dựa trên các tiến trình hoạt động của giao thức link-state, về mặt triển khai chúng ta thấy các ưu điểm vượt trội của giao thức Link-State so với các họ các giao thức Distance-vector

* **Xây dựng topology logic của mạng dựa trên cơ sở dữ liệu -** Với cơ sở dư liệu, Router chạy Link-state sẽ luôn chủ động xác định được tuyến đường tối ưu nhất, xác định được kết nối end-to-end 1 tuyến đường (các node dọc tuyến và cost trên link giữa 1 cặp node) và đảm bảo không bao giờ xảy ra loop trong mạng
* **Hội tụ nhanh –** Khác với Distance-vector, các giao thức Link-State sẽ tốn nhiều thời gian hơn khi thiết lập mối quan hệ Adjacency lúc đầu, và thời điểm cập nhật LSP đầu tiên. Tuy nhiên sau khi mạng hội tụ lần đầu tiên, các Router chạy Link-State luôn đảm bảo 1 hạ tầng định tuyến ổn định, phản ứng nhanh khi có thay đổi, tốc độ tự hội tụ nhanh chóng và phương án lựa chọn tuyến đường tối ưu linh hoạt.
* **Cập nhật kích hoạt –** Cập nhật kích hoạt trên các giao thức Link-State thỏa mãn 2 điều kiện: Thứ nhất, bản tin cập kích hoạt chỉ mang thông tin về link có sự thay đổi trạng thái, không phải toàn bộ cơ sở dữ liệu, không phải toàn bộ LSP của Router phát hiện thấy thay đổi. Thứ 2, bản tin cập nhật chỉ được gửi khi có sự thay đổi. Như vậy cập nhất kích hoạt của giao thức Link-State vừa đảm bảo tốc độ hội tụ nhanh, vừa không ảnh hưởng lớn tới hiệu năng mạng. Cập nhật kích hoạt của định tuyến Link-State cũng không có vấn đề với quá trình làm tươi bảng cơ sở dữ liệu.
* **Thiết kế phân cấp (Hierarchical design) –** Khác với kiến trúc phẳng của miền định tuyến chạy Distance-vector, thiết kế miền định tuyến của Link-State là thiết kế phân cấp sử dụng các miền SPF riêng biệt gọi là Area. Với Area, việc thiết kế hạ tầng định tuyến hiệu quả hơn vì chúng ta luôn có thể điều khiển được quá trình trao đổi LSP giữa cac miền Area khác nhau, nhưng vẫn thuộc 1 miền định tuyến Link-State. Lợi ích thứ 2 của kiến trúc phân cấp đó là việc triển khai kỹ thuật nâng cao như lọc tuyến đường hoặc gộp tuyến đường sẽ trở nên linh hoạt hơn nhiều (hỗ trợ nhiều kỹ thuật lọc tuyến hơn) và dễ dàng hơn (chỉ cần triển khai trên các Router biên đứng giữa các miền Area).



*Hình 9.6 – Mô hình phân vùng/cấp với giao thức Link-State*

## Triển khai giao thức Link-State

### Yêu cầu phần cứng

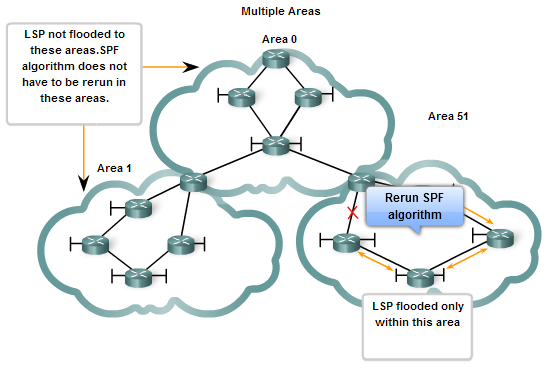
Chúng ta đã đi qua các tiến trình vận hành của 1 giao thức Link-State, để đạt được những lợi ích vượt trội của mình, cac giao thức Link-State yêu cầu mỗi Router phải duy trì 1 bảng cơ sở dữ liệu riêng. Mỗi Router phải chạy thuật toán SPF 1 cách độc lập để xây dựng topology và tự xác định đường đi tối ưu. Mỗi Router phải thực hiện 1 quá trình lụt bản tin lúc khởi tạo và định kỳ để làm tươi bảng cơ sở dữ liệu. Như vậy yêu cầu phần cứng khi triển khai mạng chạy định tuyến Link-State cần phải xem xét:

* **Bộ nhớ [Memory] –** Bộ nhớ RAM của Router phải đủ lớn để duy trì bảng cơ sở dữ liệu. Mạng càng mở rộng, càng lớn, bộ nhớ yêu cầu càng nhiều vì số lượng bản tin LSP 1 Router phải duy trì là rất nhiều
* **Tốc độ xử lý –** Thuật toán SPF phức tạp hơn các thuật toán của Distance-vector, nhất là quá trình xây dựng topology với cơ sở dữ liệu và xây dựng cây SPF cho bảng định tuyến, đặc biệt trong 1 vài thời điểm việc trao đổi dữ liệu xảy ra dồn dập. Do đó vấn đề về tốc độ xử lý thực sự là vấn đề cần xem xét
* **Băng thông –** Khác với các giao thức Distance-vector, Router chạy Link-State không có việc trao đổi cập nhật định kỳ. Tuy nhiên có 3 tiến trình yêu cầu băng thông mà khi triển khai Link-State chúng ta cân quan tâm – Thứ nhất là băng thông cho việc trao đổi Hello định kỳ (chu kỳ tương đối ngắn) thứ 2 là băng thông cho việc lụt bản tin khi có thay đổi xay ra với cập nhật kích hoạt, và thứ 3 là băng thông cho tiến trình lam tươi bảng cở sở dữ liệu. Kết hợp 3 yêu cầu băng thông trên lại thì thực sự đây là vấn đề 1 giao thức Link-State cần phải để ý, nhất là trên những kết nối WAN có tốc độ truyền dữ liệu thấp.

### Yêu cầu thiết kế

Vấn đề lụt gói tin luôn là 1 vấn đề lớn cho dù chúng ta triển khai giao thức gì hay bất kỳ 1 tiến trình nào. Với các giao thức Link-State, việc lụt bản tin LSP là vấn đề không dễ dàng, nhât với với topology mạng có quy mô lớn, số lượng bản tin LSP được đẩy ra là rất lớn, và phần nhiều trong đó là thừa thãi, chẳng hạn 1 Router có thể nhận LSP của 1 Router đầu xa từ nhiều Router hàng xóm, nhưng chỉ dùng 1 bản tin LSP, những bản tin còn lại đều bị loại bỏ (drop). Để hạn chế điều đó xảy ra, cộng với việc yêu cầu 1 hệ thống có khả năng mở rộng mạnh mẽ, nhưng vẫn phải đảm bảo hội tụ nhanh và khả năng chống loop tự nhiên, yêu cầu thiết kế là yêu cầu bắt buộc khi triển khai 1 giao thức Link-State

* Với mạng có quy mô vừa và nhỏ, triển khai mô hình Link-State “Phẳng” với 1 Area duy nhất
* Với mạng có quy mô lớn, triển khai mô hình Link-State phân cấp với nhiều Area kết nối với nhau, trong đó có những Area làm những chức năng đặc biệt dựa vào các kết nối của Area đó với các Area khác (mà chúng ta sẽ đi chi tiết trong các tài liệu khác)



Đối với định tuyến Link-State, việc tối ưu hệ thống luôn là 1 yêu cầu tất yếu mà ít khi được quan tâm. Có 2 hướng mà người quản trị cần quan tâm khi xây dựng hạ tầng định tuyến với Link-State, đó là hướng tối ưu thiết kế và hướng tối ưu triển khai.

* Tối ưu thiết kế sẽ tập trung phân cấp các Area 1 cách có cấu trúc và hệ thống dựa trên đặc điểm, yêu cầu, quy mô của mỗi Area
* Tối ưu triển khai tập trung vào các kỹ thuật nâng cao như lọc và gộp tuyến đường; điều khiển các khoảng thời gian update, timer; điều khiển các tiến trình đặc biệt trong những trường hợp cụ thể

Sẽ là tối ưu nhất nếu người quản trị thực hiện đồng thời 2 tiến trình tối ưu trên ngay từ đầu.

### Lựa chọn 1 giao thức Link-State

Hai giao thức Link-State phổ biến được sử dụng rộng rãi hiện nay:

* OSPF – Open Shortest Path First
* IS-IS – Intermediate System to Intermediate System
  1. **OSPF**

OSPF được phát triển bởi IETF từ những năm 1987, tới nay OSPF là giao thức định tuyến nội được sử dụng rất phổ biến. Các phiên bản của OSPF được chuẩn hóa bởi các RFC

* OSPFv2: OSPF dành cho mạng IPv4 (RFC 1247 and RFC 2328)
* OSPFv3: OSPF dành cho mạng IPv6 (RFC 2740)

OSPF được sử dụng rộng rãi vì tính mở và hỗ trợ nhiều tính năng.

* 1. **IS-IS**

IS-IS được thiết kế bởi tổ chuẩn hóa ISO (International Organization for Standarlization) và được mô tả trong ISO-10589. Tiền thân của IS-IS là giao thức DECnet Phase V được phát triển bởi DEC. IS-IS được phát triển trên nền tảng mô hình OSI, không thuộc bộ giao thức TCP/IP, hay nói cách khác, không triển khai trên mạng IP. Sau đó, bản IS-IS tích hợp (integrated IS-IS) được đưa ra để hỗ trợ mạng IP. IS-IS được triển khai nhiều bởi các ISP, tuy nhiên ngày càng nhiều các doanh nghiệp bắt đâu chuyển sang dùng IS-IS

## Tổng kết

Trong chương này chúng ta đã có cái nhìn khái quát về giao thức Link-State bao gồm:

* Đặc điểm chung các giao thức họ Link-State
* Các tiến trình để cập nhật thông tin định tuyến
* Yêu cầu khi thiết kế định tuyến với giao thức Link-State

# ****CHƯƠNG 10 - OSPF****

Đại diện của Distance-vector chúng ta đã tìm hiểu là RIPv1 và RIPv2. Chương 10 sẽ giới thiệu và hướng dẫn triển khai 1 đại diện của giao thức định tuyến họ Link-State là OSPFv2

## Giới thiệu giao thức OSPF

OSPF là 1 giao thức định tuyến thuộc họ link-State được phát triển để thay thế các giao thức họ Distance-vector như RIP. Khi yêu cầu mạng càng mở rộng, RIP ngày càng trở nên kém tối ưu, vì thứ nhất: vận hành dễ xảy ra sự cố khi mạng đủ lớn, những vấn đề liên quan đến loop, tốc độ hội tụ, độ ổn định của định tuyến trong mạng khiên cho RIP càng lúc càng khó đáp ứng, thứ 2 là nghèo nàn chức năng, thiết kế topology của RIP là dạng phẳng, không hỗ trợ nhiều kỹ thuật điều khiển tuyến đường.

OSPF là giao thức định tuyến Classless sử dụng ý tưởng về các miền SPF được gọi là Area để tăng cường mạnh mẽ khả năng mở rộng. OSPF sử dụng cost là metric, giá trị của cost phụ thuộc vào băng thông trên link, vì thế tuyến đường tối ưu được lựa chọn bởi OSPF sẽ đáng tin cậy hơn và đảm bảo hiệu năng cao hơn. Lợi thế lớn nhất của OSPF so với RIP đó là tốc độ hội tụ nhanh và khả năng mở rộng rất tốt trong quá trình triển khai. Chương 10 OSPF của CCNA kỳ 2, chúng ta sẽ chỉ tiếp cận những vấn đề cơ bản nhất của OSPF.

### Lịch sử phát triển

Giao thức OSPF được phát triển từ những năm 1987 bởi nhóm nghiên cứu OSPF của IETF.

Phiên bản OSPFv1 được mô tả trong RFC 1131 hỗ trợ triển khai trên Router hoặc máy trạm UNIX. Tuy nhiên OSPFv1 chỉ chỉ giao thức định tuyến thử nghiệm và chưa từng được triển khai.

Tới năm 1991, OSPFv2 được mô tả trong RFC 1247 bởi John Moy, OSPFv2 hỗ trợ rât nhiều tính năng nâng cao so với OSPFv1. OSPFv2 được phát triển song song với IS-IS, và bắt đầu xuất hiện rộng rãi như 1 giao thức IGP cho mạng IP ổn định, mạnh mẽ nhưng không quá phức tạp. Bản nâng cấp của OSPFv2 được mô tả bởi RFC 2328 và là phiên bản OSPF chuẩn được triển khai hiện nay

Năm 1999, với xu hướng của thế hệ giao thức IP nextgen, OSPFv3 hỗ trợ mạng Ipv6 được mô tả trong RFC 2740, bởi John Moy, cùng với Rob Coltun và Dennis Ferguson.



*Hình 10.1 – Lịch sử phát triển OSPF*

### Các đặc điểm cơ bản của OSPF

Chúng ta liệt kê dưới đây các đặc điểm cơ bản của giao thức OSPF, ngoài các đặc điểm của 1 giao thức Link-State:

* Trường protocol trong bản tin IP của gói tin OSPF bằng 89
* Sử dụng địa chỉ multicast 224.0.0.5 để gửi cập nhật giữa các Router hàng xóm (224.0.0.5 là bản tin gửi tới tất cả Rouer chạy OSPF)\
* Sử dụng 1 trường RouterID để định danh 1 Router là duy nhất trong mạng
* Hỗ trợ nhiều loại mạng khác nhau với các môi trường truyền khác nhau
* Các Router có nhiều vai trò khác nhau để thực hiện các mục đích khác nhau
* Định nghĩa nhiều loại tuyến đường khác nhau để xác định mức độ ưu tiên khi lựa chọn tuyến đường
* Sử dụng khái niệm {vùng} để chia cách các miền SPF, hỗ trợ định tuyến đơn vùng hoặc đa vùng
* Sử dụng nhiều loại vùng khác nhau có các đặc điểm và vai trò khác nhau đối với định tuyến
* Hỗ trợ nhiều kỹ thuật nâng cao như lọc tuyến, gộp tuyến, xác thực Router hàng xóm

## Vận hành của giao thức OSPF

### Các tiến trình vận hành của OSPF

Trước khi đi sâu vào các thành phần cụ thể của OSPF, chúng ta tóm tắt các tiến trình vận hành của giao thức này, là 1 giao thức Link-State, về cơ bản OSPF cũng bao gồm các tiến trình đã được đề cập ở chương 9

1. **Hello -** Các Router gửi bản tin Hello ra các interface chạy OSPF để khám phá hàng xóm. Khi 2 Router chia sẻ cùng liên kết trực tiếp, cùng rchạy OSPF và thỏa mãn 1 số thông số mang theo trong bản tin Hello thì 2 Router đó trở thành hàng xóm (neighbor)
2. **Adjacency -** Sau khi có mối quan hệ hàng xóm, từng cặp Router sẽ thiết lập mối quan hệ Adjacency, quan hệ Adjacency là quan hệ điểm-điểm cho dù môi trường truyền dữ liệu là điểm-điểm hay đa truy cập. Các Router phải thống nhất với nhau nhiều thông số hơn liên quan tới loại mạng, loại Router và các thông số thời gian để có thể thiết lập Adjacency
3. **LSA (Link-State Advertisement) –** Các Router sau khi thiết lập Adjacency sẽ bắt đầu trao đổi các bản tin cập nhật, mang theo thông tin LSA. Do đặc điểm OSPF nên sẽ có nhiều loại LSA mang theo các loại thông tin về link và trạng thái link khác nhau
4. **Xây dựng cơ sở dữ liệu -** Mỗi Router sau khi nhận được LSA sẽ lưu lại vào bảng cơ sở dữ liệu Link-State. Sau đó các Router sẽ tạo 1 bản sao và gửi bản sao sang cho các Router hàng xóm. Đặc điểm của quá trình này được mô tả là “lụt” gói tin, để đảm bảo tất cả Router trong miền định tuyến đều nhận được đầy đủ LSA
5. **Xây dưng cây SPF -** Sau khi bảng cơ sở dữ liệu hoàn chỉnh, mỗi Router sẽ sử dụng thuật toán SPF để xây dựng 1 đồ thị dạng cây mô tả tất cả tuyến đường ngắn nhất (tốt nhất) tới tất cả các mạng đích. Đồ thị dạng cây này được gọi là cây SPF
6. **Cập nhật bảng định tuyến -** Cuối cùng, các Router dựa vào cây SPF để cập nhật các tuyến đường vào bảng định tuyến

Khi tất cả thông tin Link-State được cập nhật trên tất cả Router, xem như OSPF đã đồng bộ thành công bảng cơ sở dữ liệu, khi đó mạng hội tụ. Sau khi mạng hội tụ, OSPF sẽ chỉ tiếp tục trao đổi bản tin Hello với mục đích duy trì kết nối giữa các hàng xóm, toàn bộ LSA sẽ được gửi lại sau mỗi 30 phút, đây là giai đoạn làm tươi. Nếu định tuyến trong mạng ổn định, sẽ không có bất cứ tiến trình nào khác diễn ra

### Các bản tin OSPF

1. **Cấu trúc bản tin OSPF**

Bản tin OSPF tương tự như như những ứng dụng khác, cũng được đóng gói vào packet. Giao thức OSPF hỗ trợ 5 loại bản tin, chúng ta sẽ đề cập tới từng loại bản tin trong phần sau. Cấu trúc cơ bản của 1 bản tin OSPF như sau:



*Hình 10.2 – cáu trúc bản tin OSPF*

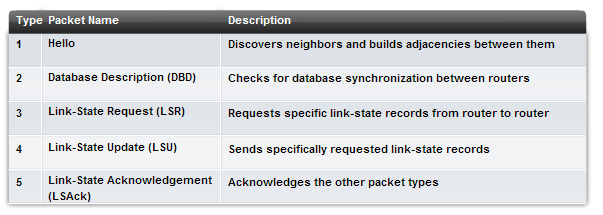
Tất cả cac bản tin OSPF đều có OSPF header, phần nội dung và header của bản tin OSPF được đóng gói thành IP packet với IP header mang các thông tin xác định bản tin OSPF:

* Trường protocol bằng 89 để chỉ dữ liệu được đóng gói xuống là OSPF
* Địa chỉ IP đích sẽ là 1 trong 2 địa chỉ Multicast: 224.0.0.5 hoặc 224.0.0.6

Nếu bản tin OSPF được gửi ra môi trường Ethernet, địa chỉ MAC đích trong Frame header sẽ dạng tương ứng với 2 địa chỉ IP multicast: 01-00-5E-00-00-05 và 01-00-5E-00-00-06

1. **Các loại bản tin OSPF**

Trong chương trước giới thiệu về giao thức định tuyến Link-State, chúng ta đã đề cập tới bản tin LSP (Link-state Packet), OSPF sử dụng 5 loại bản tin LSP, và mỗi loại được sử dụng cho những mục đích khác nhau trong quá trình thực hiện định tuyến



*Hình 10.3 – các loại bản tin OSPF khác nhau*

**Bản tin OSPF loại 1 - Hello -** Bản tin Hello được sử dụng bởi giao thức Hello, làm nhiệm vụ khám phá hàng xóm, thiết lập và duy trì mối quan hệ Adjacency với các Router chạy OSPF khác

**Bản tin OSPF loại 2- DBD –** Bản tin mô tả cơ sở dữ liệu (Database Description) mang thông tin danh sách header các bản tin cơ sở dữ liệu (Database Link-State) sẽ được gửi bởi Router. Bản tin DBD được gửi trước khi các bản tin mang thông tin cơ sở dữ liệu đầy đủ được gửi. Bản tin DBD được sử dụng để đồng bộ danh sách các LSP giữa các Router chạy OSPF

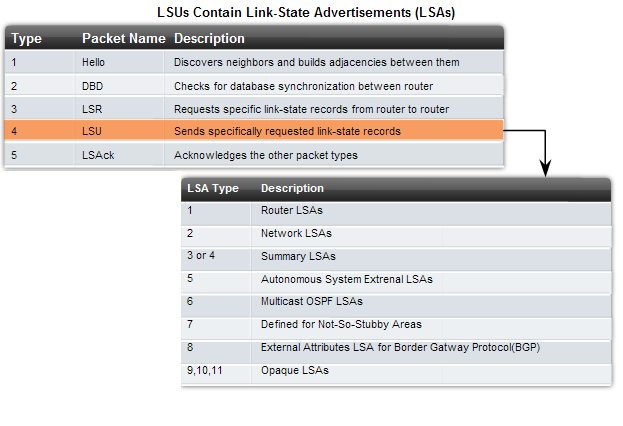
**Bản tin OSPF loại 3 - LSR** – Bản tin yêu cầu Link-State LSR (Link-State Request) được gửi ra khi 1 Router muốn nhận thông tin về link và trạng thái link cụ thể. LSR được sử dụng khi 1 Router nhận bản tin DBD và thấy rằng cơ sở dữ liệu hiện tại chưa đầy đủ, Router sẽ gửi ra LSR để yêu cầu mảnh dữ liệu còn thiếu

**Bản tin OSPF loại 4 - LSU –** Bản tin cập nhật Link-State LSU – Link-state Update, bản tin LSU có thể xem như tương tự như bản tin LSP chúng ta đã đề cập rất nhiều trong chương về giao thức Link-State. Bản tin LSU được sử dụng để phản hồi bản tin LSR hoặc quảng bá thêm rthông tin khi trong mạng có sự thay đổi. Bản tin LSU mang thông tin cơ sở dữ liệu về link và trạng thái link, nhưng không phải luôn luôn cố định. Thông tin về link và trạng thái link bên trong LSU được gọi là LSA – Link-State Adverisement. OSPF hỗ trợ 7 loại LSA khác nhau. Chúng ta sẽ đề cập tới các loại LSA trong các phần sau

**Bản tin OSPF loại 5 - LSAck –** Khi 1 Router nhận 1 bản tin LSU, Router đó sẽ phản hồi bằng bản tin LSAck để xác nhận

1. **Bản tin OSPF Link-State update**

LSU là các bản tin được trao đổi trong quá trình trao đổi cập nhật thông tin định tuyến. LSU mang các thông tin cơ sở dữ liệu trên các Router gửi ra. Thông tin cơ sở dữ liệu trong LSU được gọi là LSA (Link-State Advertisement) – Có nhiều loại LSA khác nhau mang các thông tin cơ sở dữ liệu khác nhau phục vụ các tiến trình cập nhật khác nhau. Có 10 loại LSA được mô tả, tuy nhiên chỉ có 6 trong số đó là được sử dụng phổ biến, thiết bị Cisco cũng không hỗ trợ một vài LSA.



*Hình 10.4 – Các bản tin cập nhật LSA*

**LSA loại 1 – Router LSA –** Mang thông tin về các link và trạng thái link kết nối trực tiếp của từng Router. Mỗi Router chỉ gửi ra 1 LSA loại 1 duy nhất. Router LSA được gửi ra trong phạm vi 1 vùng (area)

**LSA loại 2 – Network LSA -** Mang thông tin về mạng trong môi trường đa truy cập. Network LSA liệt kê ra các Router tham gia định tuyến OSPF trong 1 mạng đa truy cập. LSA loại 2 được gửi ra trong phạm vi 1 vùng

**LSA loại 3 – Network Summary LSA** – Mang thông tin về mạng bao gồm địa chỉ mạng và địa chỉ Router OSPF quảng bá giữa các vùng khác nhau. Chúng ta sẽ đề câp trong các giao trình khác.

**LSA loại 4 – ASBR Summary LSA –** Mang thông tin về Router chạy OSPF được sử dụng để định tuyến liên giao thức (Router chạy đồng thời OSPF và 1 giao thức định tuyến khác). Chúng ta sẽ đề cập trong các giáo trình khác

**LSA loại 5 – AS External LSA –** Mang thông tin về 1 mạng ở ngoài miền định tuyến OSPF, hay nói cách khác, 1 mạng được định tuyến bởi giao thức khác. Chúng ta sẽ đề cập trong giáo trình khác

### Giao thức Hello

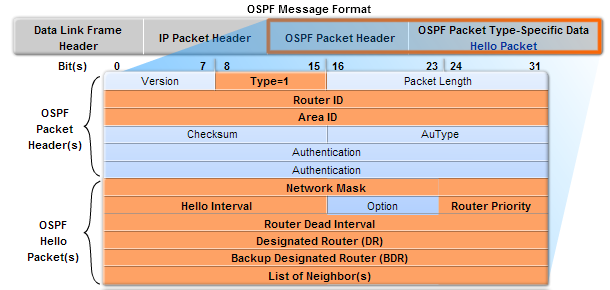
Giao thức Hello là giao thức đặc biệt quan trọng trong cấu trúc OSPF, kích cỡ nhỏ, gửi định kỳ, nhưng giao thức Hello không chỉ thực hiện chức năng khám phá hàng xóm và cơ chế keepalive, quan trọng hơn OSPF Hello được sử dụng để thiết lập mối quan hệ Adjacency, và triển khai 1 số tiến trình đặc biệt

* 1. **Các chức năng của bản tin Hello**

Bản tin OSPF được sử dụng bởi giao thức OSPF, đây là bản tin loại 1 trong số các loại bản tin OSPF. Chức năng của gói tin Hello:

* Khám phá Router hàng xóm
* Trao đổi thông tin các tham số mà 2 thiết bị phải khớp mới có thể thiết lập mối quan hệ Adjacency
* Cơ chế keepalive để đảm bảo kết nối giữa 2 Router hàng xóm
* Đảm bảo kết nối là 2 chiều khi 1 Router nhìn thấy trường RouterID của mình trong bản tin Hello nhận từ Router hàng xóm
* Bình bầu DR/BDR (Designated Router/ Backup Designated Router) trong môi trường mạng đa truy cập như Ethernet trong LAN và Frame-relay trong WAN
  1. **Các trường trong bản tin Hello**

Chúng ta xem xét các trường thông tin bên trong 1 bản tin Hello:



*Hình 10.5 Cấu trúc bản tin Hello*

**Type –** Bằng 1 vì đây là bản tin Hello

**RouterID** **-** Định danh 1 node là duy nhất trong cả miền định tuyến OSPF. RouteID được viết dưới dạng 1 địa chỉ IP 32 bit với 4 octet.

**Area ID -** Xác định Area (vùng) mà bản tin Hello được sinh ra

**Mặt nạ mạng - Network Mask -** Là Subnet mask của địa chỉ IP gán cho interface gửi ra bản tin Hello. Trông loại mạng đa truy cập như Ethernet, mặc định thông tin Network mask phải khớp giứa các bản tin Hello để 2 Router có thể thiết lập Adjacency, còn trong môi trường dạng điểm-điểm, Network mask không yêu cầu phải khớp

**Hello Interval -** Hello interval (thời gian gửi Hello) là một trong những thông số cần phải khớp để 2 Router thiết lập Adjacency, 2 thông số còn lại bắt buộc phai khớp khác là: Dead Interval (thời gian Dead) và loại mạng. Hello interval chỉ ra chu kỳ 1 Router OSPF gửi ra bản tin Hello, mặc định với loại mạng đa truy cập/quảng bá như Ethernet, Hello interval bằng 10 giây, còn với loại mạng đa truy cập/ không quảng bá (Non-broadcast MultiAccess) (như Frame-relay, X.25 hoặc ATM) thì Hello interval bằng 30 giây

Trong phần lớn trường hợp, bản tin Hello OSPF được gửi dưới dạng Multicast tới địa chỉ 224.0.0.5 (nhóm tất cả Router chạy OSPF) Sử dụng địa chỉ Multicast cho phép 1 Router từ chối bản tin Hello nếu interface của Router đó chưa kích hoạt OSPF.

**Router Priority -**  Được sử dụng trong tiến trình bình bầu DR/BDR

**Router Dead Interval -** Dead interval là khoảng thời gian tối đa 1 Router cho phép chờ 1 bản tin Hello trước khi tuyến bố mất kết nối Router hàng xóm. Mặc định Cisco sử dụng Dead interval bằng 4 lần Hello interval. Đối với loại mạng quảng bá/đa truy cập với hello interval bằng 10 giây thì Dead interval bằng 40 giây, tương tự bằng 120 giây với loại mạng đa truy cập/không quảng bá (NBMA). Nếu hết Dead interval mà Router vẫn không nhận được bản tin Hello, OSPF sẽ xóa bản ghi hàng xóm tương ứng từ cơ sở dữ liệu. Router sẽ gửi thông tin Link-State để thông báo cho các Router khác về việc mất kết nối hàng xóm

**Designated Router (DR) -** Thông tin RouterID của DR (sau khi bình bầu), nếu hiện đã có

**Backup Designated Router (BDR) -** Thông tin RouterID của BDR, (sau khi bầu) nếu hiện đã có

**Neighbor(s)-** Danh sách RouterID của các Router hàng xóm trên link gửi ra bản tin Hello đó.

### Hàng xóm và quan hệ Adjacency

Trước khi các bản tin Link-State có thể được gửi, các Router phải khám phá ra Router hàng xóm trên các link kết nối trực tiếp và thiết lập quan hệ Adjacency với các Router hàng xóm. OSPF sử dụng bản tin Hello gửi ra địa chỉ Multicast 224.0.0.5 tới tất cả Router chạy OSPF để khám phá hàng xóm. Danh sách các Router hàng xóm được lưu lại trên mỗi Router chạy OSPF vào trong 1 bảng riêng (neighbor table), tất cả thông tin liên quan đến Router hàng xóm đó đều được lưu lại, chúng ta xem bảng hàng xóm của OSPF:

Router#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

172.16.1.1 1 FULL/BDR 00:00:32 172.16.2.1 FastEthernet0/0

172.16.4.1 1 EXSTART/DR 00:00:32 192.168.1.1 FastEthernet0/1

Router#show ip ospf neighbor detail

Neighbor 172.16.1.1, interface address 172.16.2.1

In the area 0 via interface FastEthernet0/0

Neighbor priority is 1, State is FULL, 5 state changes

DR is 172.16.2.2 BDR is 172.16.2.1

Options is 0x00

Dead timer due in 00:00:37

Neighbor is up for 01:24:09

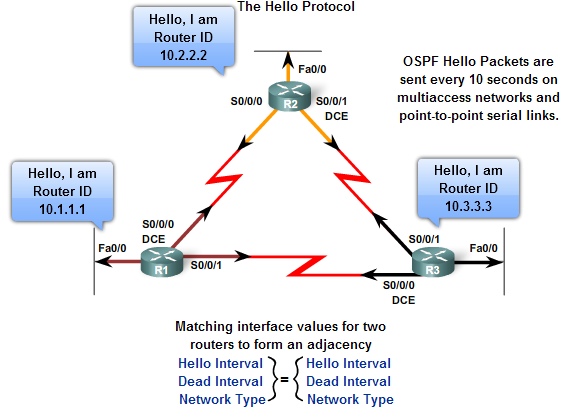
Index 1/1, retransmission queue length 0, number of retransmission 0

First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)

Last retransmission scan length is 0, maximum is 1

Last retransmission scan time is 0 msec, maximum is 0 msec

<output omitted>



*Hình 10.6 – các thông tin phải khớp để 2 Router thiết lập ADjacency*

**Router ID**

Là trường định danh 1 Router là duy nhất trong miền định tuyến OSPF. Khi 1 Router bật OSPF trên 1 interface bất kỳ, Router đó sẽ tự tạo ra thông tin RouterID và quảng bá thông tin này cho các Router hàng xóm. Việc xác định RouterID diễn ra như sau:

* Router lựa chọn địa chỉ IP của cổng loopback (nếu có) có giá trị lớn nhất
* Nếu không có cổng Loopback, Router lựa chọn địa chỉ IP của 1 cổng vật lý có giá trị lớn nhất.

Khi 1 Router đã xác định xong RouterID thì RouterID đó sẽ duy trì mãi mãi trên Router đó cho đến khi Router khởi động lại, hoặc các tiến trình OSPF bị khởi động lại. Ngoài phương pháp tự xác định RouterID như trên, OSPF cũng cho phép người quản trị tự xác định RouterID cho Router. Chúng ta cũng nhớ rằng RouterID không phải 1 địa chỉ IP, chỉ có dạng giống 1 địa chỉ IP, là 1 chuỗi 32 bit chia làm 4 octet. Vì thế chúng ta có thể gán cho Router chạy OSPF bất kỳ giá trị RouterID nào, có thể 1 địa chỉ host, 1 địa chỉ mạng, 1 địa chỉ Broadcast, 1 địa chỉ Multicast,…

Các thông tin được lưu trong bảng hàng xóm bao gồm:

* RouterID của Router hàng xóm
* Interface nối ra Router hàng xóm và địa chỉ IP của interface đó
* Mạng kết nối trực tiếp với Router hàng xóm
* Chỉ số ưu tiên [priority] được sử dụng trong tiến trình bình bầu DR/BDR sẽ đề cập trong phần sau

Quan hệ Adjacency chỉ được thiết lập sau khi 2 Router hàng xóm thống nhất các thông số thuộc tính. Các thông số cần phải khớp:

* **Area ID –** Trong nội dung CCNA chúng ta chỉ đề cập tới OSPF đơn vùng nên trường Area ID luôn khớp giữa các Router. Khi triển khai OSPF đa vùng thì các Router chỉ có thiết lập Adjacency khi các interface của các Router đó thuộc cùng 1 vùng
* **Mật khẩu xác thực và loại xác thực -** OSPF hỗ trợ xác thực để có thế thiết lập Adjacency. Loại xác thực có thể là MD5 hoặc clear-text hoặc không có xác thực, thông tin mật khẩu dĩ nhiên phải khớp giữa các Router
* **Subnet mask (trong môi trường đa truy cập) –** Mặt nạ mạng để xác định các Router thuộc cùng 1 miền broadcast domain, nêu là môi trường điểm-điểm thì thông tin này không cần thiết phải khớp
* **Hello interval và Dead interval –** Chu kỳ gửi ra bản tin Hello và thời gian cho phép chờ tối đa trước khi tuyên bố đứt hàng xóm
* **Giá trị MTU –** Maximum Transmission Unit, OSPF yêu cầu MTU trên interface giữa 2 Router phải khớp nhau

### Bình bầu DR/BDR

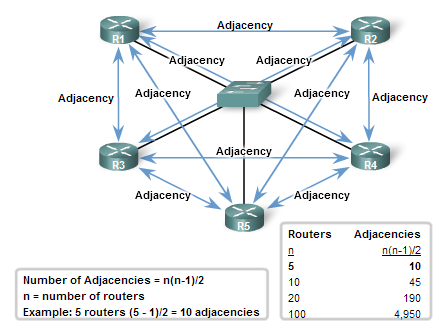
1. **Bài toán OSPF trong môi trường đa truy cập**

Chúng ta đã biết trong tài liệu CCNA kỳ 1 rằng, một mạng đa truy cập là một mạng có nhiều hơn 2 thiết bị cùng chia sẻ một môi trường truyền dẫn chung. Đaị diện của 1 mạng đa truy cập là Ethernet LAN, có thể tạo thành từ 1 hoặc nhiều miền đụng độ, nhưng luôn luôn là 1 miền quảng bá. Đặc điểm của miền quảng bá đó là tất cả thiết bị đều nhìn thấy frame, hay miền quảng bá cho phép 1 bản tin Broadcast “lụt” bên trong.

Ngược lại với mạng đa truy cập là mạng kết nối dạng điểm-điểm, mạng điểm điểm là mạng mà chỉ có 2 thiết bị nằm 2 đầu của 2 liên kết, các kết nối WAN thương thấy là kết nối điểm-điểm

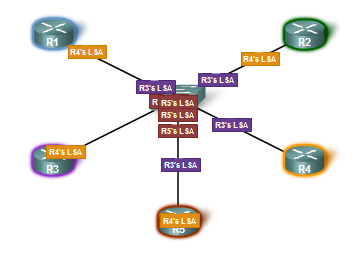
2 Router chạy OSPF chỉ có thể trao đổi thông tin định tuyến sau khi thiết lập mối quan hệ Adjacency(phụ cân). Chúng ta xét mối việc hình thành các mối quan hệ Adjacency và trao đổi thông tin cập nhật trong môi trường đa truy cập. Trong môi trường đa truy cập, có 2 vấn đề của OSPF liên quan tới quá trình lụt bản tin cập nhật và số lượng phiên Adjacency yêu cầu thiết lập:

* **Adjacency là mối liên kết điểm-điểm giữa 2 Router chạy OSPF**. Như vậy trong môi trường đa truy cập, với mỗi n Router chúng ta sẽ cần có n(n-1)/2 liên kết Adjacency – Việc này dường như tạo ra quá nhiều bản sao thông tin cập nhật định tuyến không cần thiết, nhất là trong trường hợp nếu thiết bị kết nối các Router là Hub. Trong ví dụ dưới, với n bằng 5 Router, chúng ta sẽ cần 10 liên kết Adjacency. Số lượng Router càng nhiều, số yêu cầu Adjakngugcency càng nhiều, số lượng bản tin trao đổi bao gồm: Hello định kỳ, các bản tin cập nhật LSU diễn ra càng nhiều, phần lớn trong đó là thừa thãi. Hãy xem trường hợp Router R1 có 1 cập nhật về trạng thái link, sẽ gửi ra bản tin LSU tương ứng để thông báo, thì R3 sẽ nhận được 4 bản tin LSU giống hệt nhau về trạng thái 1 link trên Router R1 từ Router R1, R2, R4 và R5. CHúng ta có nhận thấy đó chính là vấn đề



*Hình 10.7 – càng nhiều Router, số phiên thiết lập càng lớn*

* **Lụt bản tin luôn làm giảm hiệu năng mạng đáng kể**. Các giao thức định tuyến mà OSPF là 1 đại diện sẽ lụt bản tin khi Router nhận thấy có sự thay đổi trạng thái link cần cập nhật, Router đó sẽ lại lụt cho tất cả Router hàng xóm khác. Hoặc trong trường hợp làm tươi bảng cơ sở dữ liệu định kỳ, hãy xem cách tất cả Router “lụt” toàn bộ mạng với số lượng bản tin cập nhật trạng thái rất lớn. Ta thấy gián tiếp OSPF tạo ra bão “Broadcast” trong mạng, cách truyền dữ liệu này chiếm rất nhiều tài nguyên băng thông. Một lần nữa, ta lại thấy thông tin được nhân bản quá nhiều lần, mặc dù phần lớn là không cần thiết. Chúng ta có thể làm tình huống tồi tệ hơn với việc sử dụng Hub thay vì Switch làm thiết bị trung tâm



*Hình 10.8 – lụt bản tin cập nhật trong môi trường đa truy cập*

1. **Bình bầu DR/BDR**

Để giải quyết bài toán đó, người ta đưa ra khái niệm **DR** (designated Router) và **BDR** (Backup Designated Router)

DR hay BDR là các Router OSPF “trung tâm” nhận và gửi thông tin cập nhật LSA trong mạng đa truy cập. Ý tưởng của DR hay BDR đó là khi có thông tin cập nhật, các Router khác (được gọi là DROTHERS) sẽ chỉ gửi cập nhật lên DR và BDR thông qua địa chỉ đích 224.0.0.6 (địa chỉ xác định tất cả nRouter DR và BDR trong mạng). Sau đó Router làm DR sẽ nhân bản và gửi bản sao thông tin cập nhật đó ra địa chỉ 224.0.0.5 (địa chỉ xác định tất cả Router chạy OSPF trong mạng) sang cho tất cả Router hàng xòm. *Câu hỏi đặt ra là: Liệu dữ liệu tập trung qua 2 Router làm DR và BDR có làm giảm hiệu năng mạng vì tăng tải qua 2 Router này và định tuyến kém hiệu quả do phải transit qua 1 Router trung gian ?*

Câu trả lời là **Không,** vì Router DR và BDR chỉ làm nhiệm vụ tập trung các bản tin cập nhật định tuyến, dữ liệu thực sự sẽ không phải truyền qua Router DR và BDR

Để có thể làm được điều đó các Router sẽ không thiết lập Adjacency với tất cả Router hàng xóm khác nữa, mà chỉ thiết lập Adjacency với Router là DR và BDR mà thôi. Chúng ta kiểm tra bảng hàng xóm của OSPF thông thường trong mạng đa truy cập với 3 Router (Router\_1, Router\_2, và Router\_3 với số hiệu RouterID tương ứng 123.123.123.1, 123.123.123.2 và 123.123.123.3)

Router\_4# show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

123.123.123.3 1 2WAY/DROTHER 00:00:36 123.123.123.3 FastEthernet0/0

123.123.123.2 1 FULL/DR 00:00:37 123.123.123.2 FastEthernet0/0

123.123.123.1 1 FULL/BDR 00:00:33 123.123.123.1 FastEthernet0/0

Ta thấy Router\_4 thiết lập mối quan hệ Adjacency với Router\_1 (làm DR – RouterID 123.123.123.2) và Router\_2 (làm BDR- RouterID 123.123.123.1) trong khi chỉ xác nhận hàng xóm chứ không Adjacency với Router\_3 (RouterID bằng 123.123.123.3). Điều này cho phép tất cả Router DROther trong mạng đa truy cập vẫn nhận và gửi ra các bản tin Hello tới hàng xóm xung quanh. Theo đó mỗi Router vẫn nhận định được tình hình hàng xóm. Để kiểm tra trạng thái hàng xóm, ngoài phương pháp sử dụng câu lệnh show ip ospf neighbor, người quản trị cũng có thể sử dụng câu lệnh show ip ospf interface để có cái nhìn chi tiết hơn

Router#show ip ospf interface

FastEthernet0/0 is up, line protocol is up

Internet address is 192.168.1.2/24, Area 0

Process ID 1, Router ID 192.168.31.22, Network Type BROADCAST, Cost: 1

Transmit Delay is 1 sec, State BDR, Priority 1

Designated Router (ID) 192.168.31.33, Interface address 192.168.1.3

Backup Designated Router (ID) 192.168.31.22, Interface address 192.168.1.2

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

Hello due in 00:00:08

Index 1/1, flood queue length 0

Next 0x0(0)/0x0(0)

Last flood scan length is 1, maximum is 1

Last flood scan time is 0 msec, maximum is 0 msec

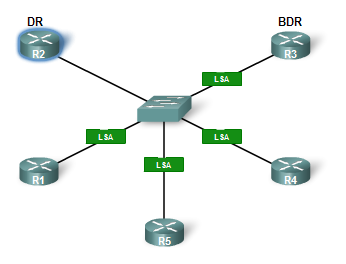
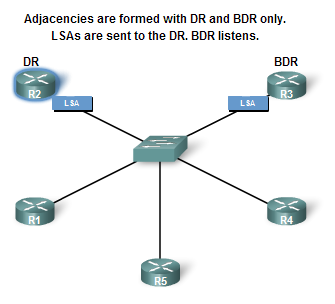
Neighbor Count is 2, Adjacent neighbor count is 2

Adjacent with neighbor 192.168.31.11

Adjacent with neighbor 192.168.31.33 (Designated Router)

Suppress hello for 0 neighbor(s)

RouterB#



*Hình 10.9 – Lúc này Adjacency chỉ thiết lập giữa DR, BDR với các Router DROTher*

Do đặc điểm trên nên tiến trình bình bầu DR/BDR chỉ diễn ra trong mạng đa truy cập kể cả trong trường hợp trong mạng chỉ có 2 Router, trong mạng liên kết điểm-điểm sẽ không diễn ra quá trình bình bầu này vì các vấn đề được đề cập ở trên không xảy ra trong mạng điểm-điểm. Như vậy ý nghĩa của DR là :

* Đại diện 1 miền đa truy cập sử dụng OSPF
* Quản lý tiến trình lụt bản tin cập nhật trong mạng

Để xác định Router là DR hay BDR, các Router sẽ trải qua 1 giai đoạn bình bầu DR/BDR - Giai đoạn này diễn ra ngay sau khi các Router xác định Router hàng xóm của mình. Tiêu chí bình bầu DR/BDR dựa trên:

* **Thứ nhất là Trường priority –** Interface trên Router nào có chỉ số priority cao nhất thì Router đó sẽ trở thành DR. Router có giá trị priority cao thứ 2 sẽ trở thành BDR. Giá trị mặc định của priority bằng 1. Router có giá trị priority bằng 0 thì không bao giờ được bầu chọn làm DR hay BDR
* **Thứ 2 là Giá trị RouterID –** Nếu 2 interface của 2 Router có priority bằng nhau, 2 Router đó sẽ so sánh giá trị của trường RouterID để xác định Router làm DR, Router nào có RouterId lớn hơn thì sẽ trở thành DR, Router có RouterID lớn thứ 2 thì sẽ trở thành BDR

Để tranh trường hợp các Router phải bình bầu DR lại khi trong mạng mất DR, các Router bầu ra luôn vị trí BDR. Router làm BDR sẽ trở thành DR khi trong mạng mât kết nối với Router DR

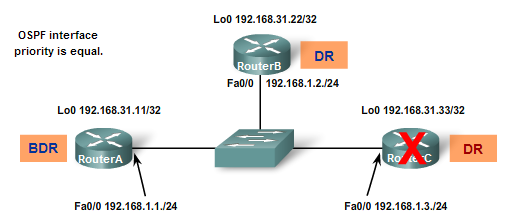
Mặc dù trong cách viết đề cập tới Router làm DR hay Router DR, nhưng thực chất DR/ hay BDR là thuộc tính của interface. Vì interface mới chính thức là đối tượng được đưa vào định tuyến (chứ không phải Router). Tương tự không phải interface nào của Router cũng làm DR và ngược lại, như vậy trong trường hợp cụ thể, chỉ interface của Router nối tới 1 mạng đa truy cập thì interface đó có khả năng trở thành DR hay BDR. Tuy nhiên việc xác định vị trí của 1 Router trong OSPF lại dựa vào RouterID. Vì thế trong 1 miền đa truy cập, có thể sử dụng RouterID như thông tin để xác định vị trí của DR và BDR. Thông tin này không có nghĩa rằng Router đó là DR trên tất cả interface đang chạy OSPF

1. **Các trường hợp thay đổi DR/BDR**

Tiến trình bình bầu DR/BDR diễn ra ngay lập tức khi 1 interface trên 1 Router bắt đầu tiến trình OSPF (tất nhiên chỉ trong môi trường đa truy cập). Quá trình bình bầu chỉ diễn ra trong vài giây, được thực hiện thông qua việc trao đổi các bản tin Hello. Khi 1 Router được xác định là DR, Router đó sẽ duy trì trạng thái và vai trò của mình trong suốt tiến trình cho đến khi có điều kiện đặc biệt dẫn tới mất kết nối DR. Các điều kiện có thể là:

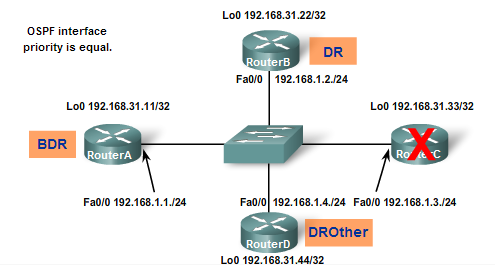
* Router làm DR ngừng tham gia trong hạ tầng định tuyến (Router tắt, khởi động lại, hoặc ngừng định tuyến OSPF)
* Tiến trình OSPF trên Router có vấn đề
* Interface nối tới miền mạng đa truy cập có vấn đề

**Router làm DR ngừng tham gia trong hạ tầng định tuyến –** Nếu Router DR không còn tồn tại trong mạng, ngay lập tức Router BDR sẽ trở thành Router DR, sau đó sẽ có quá trình bình bầu lại Router làm BDR

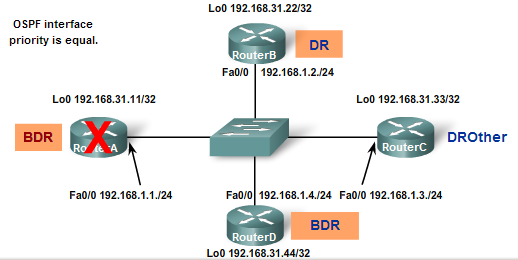


Sau khi mạng hội tụ, Router C đang là DR, Router B là BDR, còn Router A là DROther. Khi Router C rời khỏi mạng, vị trí của DR biến mất, và ngay lập tức Router B “nắm quyền” và trở thành DR – Ngay sau đó, các Router DROther sẽ tham gia bình bầu để xác định xem Router nào sẽ trở thành Router BDR mới. Trong trường hợp này, Router A là lựa chọn duy nhất còn sót lại

**Thêm 1 Router mới –** Khi trong mạng hội tụ với vị trí của Router DR và BDR đều xác định rõ ràng, sẽ không có sự xáo trộn nào các vị trí của DR và BDR trong mọi trường hợp trừ khi các Router DR và BDR hiện tại có vấn đề. Còn lại mọi Router mới tham gia, bất luận trường priority có tốt đến đâu hay RouterID cao nhất trong miền mạng đa truy cập đó, thì Router đó cũng chỉ có thể trở thành DROther, trong ví dụ dưới, Router D có RouterID tốt nhất, nhưng không thể thay thế vị trí của Router B. Khi 1 Router mất vị trí DR, trong trường hợp này là Router C, thì vai trò DR được giao cho 1 Router khác, tuy nhiên rất có thể 1 lúc Router C “quay trở lại”, vậy có ưu tiên nào dành cho “cựu” DR không? Câu trả lời là không, Router C cũng như Router D hay mọi Router mới, làm nhiệm vụ của DROther



**Thay thế vị trí BDR –** Vị trí BDR là dự phòng cho Router DR trong mạng. Việc xác định cặp đôi DR/BDR là bắt buộc trong môi trường mạng đa truy cập, vì thế cho dù không có xáo trộn Router DR trong mạng, nhưng vẫn cần vị trí Router BDR xác định. Trong ví dụ dưới, khi Router A (BDR hiện tại) mât kết nối tới các Router khác, vị trí BDR trống, các DROther còn lại sẽ tham gia trao đổi Hello để xác định Router BDR mới, tiêu chí vẫn như cũ: Priority và RouterID



Theo đó, chúng ta thấy, mặc dù cách bình bầu DR/BDR là bình đẳng, nhưng vị trí này rất nhạy cảm trong hạ tầng định tuyến. Router làm DR sẽ phải nhận và xử lý nhiều dữ liệu định tuyến hơn bất cứ Router nào khác, vì thế yêu cầu người quan trị cần xác định Router DR rõ ràng để đảm bảo yêu cầu tài nguyên cần thiết. Để 1 Router được lựa chọn là DR, chúng ta thấy có 2 cách:

* Khởi động Router đã quyết định làm DR lên đầu tiên, như vậy Router đó chắc chắn trở thành DR, vì hiện tại trong mạng không có Router nào khác.
* Cách thứ 2 là thay đổi trường Priority trong bản tin Hello.

### OSPF Metric

1. **Cost – OSPF metric**

Metric được sử dụng bởi OSPF được gọi là cost. Theo RFC 2328: "Cost được gắn với chiều ra của mỗi interface trên Router. Giá trị cost càng nhỏ, càng được ưu tiên đẩy dữ liệu”

Tuy nhiên RFC 2328 không chỉ ra yếu tố nào được sử dụng để xác định cost

Cisco IOS sử dụng băng thông tích lũy (cumulative bandwidth) để tính ra metric của tuyến đường. Metric của tuyến đường bằng tổng cost trên tất cả interface ra dọc theo tuyến đường tới mạng đích. Công thức tính cost trên Router chạy OSPF:

Cost = 108 / BW

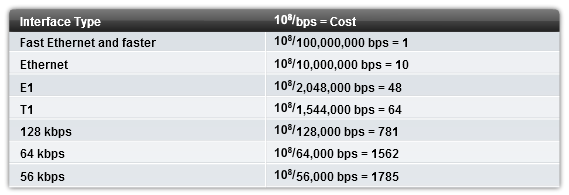
Trong đó

* 108 là băng thông tham khảo (reference bandwidth)
* BW là băng thông cấu hình trên interface

Như vậy interface có băng thông càng cao sẽ có cost càng thấp, và ngược lại cost càng cao đồng nghĩa với băng thông càng thấp. Trong định tuyến, tuyến đường có metric càng thấp càng được ưu tiên, chẳng hạn với RIP, tuyến đường có hop-count bằng 3 sẽ tốt hơn tuyến đường có hop-count bằng 10.

1. **Băng thông tham khảo - Reference Bandwidth**

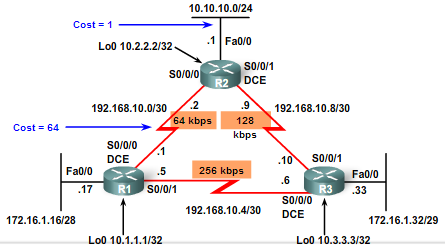
Giá trị băng thông tham khảo trong OSPF bằng 108, hay 100,000,000 bps (100 Mbps) The reference bandwidth defaults to 10 to the 8th power, 100,000,000 bps or 100 Mbps. Như vậy, giá trị cost tốt nhất cho interface có băng thông 100 Mbps hay lớn hơn sẽ có cost bằng 1. Băng thông tham khảo cũng có thể được điều chỉnh dựa theo yêu cầu hệ thống để tối ưu các liên kết có băng thông lớn hơn 100 Mbps. Băng thông tham khảo có thể được cấu hình với câu lệnh auto-cost reference-bandwidth



*Hình 10.10 – quy đổi metric*

1. **Metric tĩnh lũy trên OSPF**

Metric của tuyến đường là metric tĩnh lũy, hay nói cách khác, cost của cả tuyến đường được tính bằng tổng cost trên tất cả các link dọc tuyến đường đó. Xét ví dụ dưới:



Chúng ta kiểm tra cost trên link Serial0/0/0 của Router R1 với câu lệnh show ip ospf interface serial0/0/0

R1#show ip ospf interface

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.1/30, Area 0

Process ID 1, Router ID 10.1.1.1, Network Type BROADCAST, Cost: 64

<output omitted>

Link Serial0/0/0 của Router R1 có cost bằng 64, bằng 108 / 1544000 (Băng thông trên interface Serial0/0/0 bằng 1544 Kbps). Tương tự chúng ta kiểm tra cost trên link fastEthernet của Router R2:

R2#show ip ospf interface

FastEthernet0/0 is up, line protocol is up

Internet address is 10.10.10.0/24, Area 0

Process ID 1, Router ID 10.2.2.2, Network Type BROADCAST, Cost: 1

<output omitted>

Link FastEthernet 0/0 của Router R2 có cost bằng 1, băng 108 / 100,000,000 (băng thông trên interface fastEthernet bằng 100 Mbps) Như vậy cost của tuyến đường tới mạng 10.10.10.0/24 trên Router R1 phải bằng tổng cost trên link Serial0/0/0 của R1 (64) và cost trên link fastEthernet0/0 của R2 (1), hay bằng 65. Chúng ta kiểm tra bằng câu lệnh show ip route trên Router R1

R1#show ip route

<output omitted>

O  10.10.10.0/24 [110/65] via 192.168.10.2, 14:26:47, Serial0/0/0

<output omitted>

1. **Băng thông mặc định trên link Serial**

Băng thông là 1 thuộc tính của interface phản ánh khả năng truyền dữ liệu tối đa. Tuy nhiên chúng ta nhớ rằng giá trị băng thông trên interface Serial không phản ánh tốc độ truyền dữ liệu thực tế, thay vào đó được sử dụng bởi các giao thức định tuyến để tính toán ra metric

RouterX# show interface serial 0/3/0

Serial0/3/0 is up, line protocol is up (connected)

Hardware is HD64570

Internet address is 192.168.1.1/24

MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,

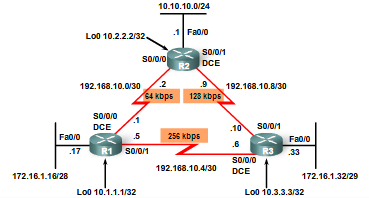
reliability 255/255, txload 1/255, rxload 1/255

<output omitted>

Tốc độ truyền dữ liệu trên interface Serial được điều khiển bởi xung đồng hồ (Clock rate) cung cấp bởi thiết bị DCE

Chúng ta cũng thấy rằng OSPF sử dụng băng thông như 1 giá trị tham khảo để xác định ra cost tương ứng, vậy 1 cách gián tiếp băng thông trên kết nối Serial lại là nhân tố quyết định metric của 1 link kết nối dù cho giá trị đó không thực sự phản ánh tốc độ truyền. Vấn đề của việc không tương đồng đó là OSPF sẽ đánh giá sai độ ưu tiên trên 1 đường truyền, dẫn đến có thể tính toán định tuyến không chính xác

Chẳng hạn, doanh nghiệp sử dụng đường kết nối T1 của nhà cung cấp dịch vụ, nhưng hợp đồng chỉ thuê ¼ tốc độ của gói T1 đầy đủ, tương ứng với tốc độ truyền 384 kbps. Tuy nhiên, OSPF sẽ nhìn nhận đây là 1 đường kết nối T1 hoàn chỉnh, và xem băng thông trên link, không phải 384 kbps, mà là 1544 kbps. Như vậy, thay vì cost bằng 256 (10,000,000 / 384,000) thì OSPF lại tính cost bằng 64 (100,000,000 / 1,544,000). Xét ví dụ bên dưới, định tuyến OSPF trên Router R1 sẽ đánh giá cost trên Serial 0/0/0 và Serial 0/0/1 bằng nhau và đều bằng 64. Tuy nhiên nhìn vào tốc độ thực tế bên dưới, ta thấy tốc độ truyền trên Serial 0/0/0 chỉ bằng 64 kbps, trong khi của Serial 0/0/1 bằng 256 kbps.



Hãy xem điều gì xảy ra khi OSPF trên Router R1 đánh giá sai cost trên link, chúng ta show ip route trên Router R1:

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<ouput omitted>

O  192.168.10.8 [110/128] via 192.168.10.6, 00:03:41, Serial0/0/1

                [110/128] via 192.168.10.2, 00:03:41, Serial0/0/0

<output omitted>

Ta thấy, để tới mạng 192.168.10.8/30, thay vì sử dụng đường qua Serial 0/0/1 có tốc độ cao hơn, Router R1 lại xem 2 đường Serial 0/0/0 (tới Router R2) và Serial 0/0/1 (tới Router R3) có tốc độ như nhau và triển khai load-balancing. Cả 2 tuyến đường đều có cost bằng 128 (64 + 64). Chúng ta kiểm tra cost trên 2 link này để thấy cách OSPF đánh giá bằng câu lệnh show ip ospf interface

R1#show ip ospf interface

Serial0/0/1 is up, line protocol is up

Internet address is 192.168.10.5/30, Area 0

Process ID 1, Router ID 172.16.1.16, Network Type POINT-TO-POINT, Cost: 64

Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0

<output omitted>

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.1/30, Area 0

Process ID 1, Router ID 172.16.1.16, Network Type POINT-TO-POINT, Cost: 64

Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0

<output omitted>

Như vậy Router R1 gán cost bằng 64 cho cả 2 interface Serial 0/0/0 và Serial 0/0/1, mặc dầu giá trị cost này không phản ánh được tốc độ truyền thực tế. OSPF mặc định có thể đánh giá sai băng thông trên kết nối, để giải quyết bài toán đó, yêu cầu người quản trị phải sửa lại sao cho cost trên link phản ánh đúng tốc độ truyền dữ liệu, chúng ta sẽ đi vào các giải pháp trong phần cấu hình

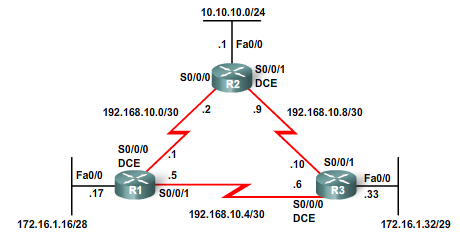
### OSPF và các loại mạng

OSPF là 1 giao thức thông minh, do đó giao thức này định nghĩa các cách ứng xử khác nhau trên các loại mạng khác nhau, chẳng hạn ứng xử trên 1 kết nối điểm-điểm thì khác ứng xử trên 1 môi trường đa truy cập. Chúng ta xem xét các loại mạng được định nghĩa bởi OSPF:

* Point-to-Point (Điểm-điểm)
* Broadcast Multi-access (Quảng bá đa truy cập)
* Non-Broadcast Multi-access (không quáng bá – Đa truy cập)
* Point-to-Multipoint (Điểm –đa điểm)
* Virtual-link (Liên kết ảo)

## Cấu hình OSPF

Để minh họa triển khai OSPF, chúng ta sử dụng topology dưới đây:



Chúng thấy mô hình mạng trên triển khai VLSM với mạng không liên tục (discontigous) – Điều này không có vấn đề gì với OSPF vì đây là giao thức định tuyến Classless. Giả sử các Router đã được cấu hinh khởi tạo với địa chỉ IP như trong topology

### Quảng bá một mạng vào OSPF

**Cú pháp**

GIống như bất kỳ 1 giao thức định tuyến khác, mục đích của OSPF là quảng bá mạng vào định tuyến. Tương tự cú pháp của mọi giao thức định tuyến, Router chạy OSPF quảng bá 1 tuyến đường với cú pháp network trong sub-mode Router. Chức năng của câu lệnh network là kích hoạt tất cả interface có địa chỉ IP thuộc dải mạng cấu hình trong câu lệnh thực hiện gửi và nhận các bản tin OSPF.

Router(config)#router ospf {process-ID}

Router(config-router)#network {địa chỉ IP mạng} {wildcard mask} area {số hiệu vùng}

Trong đó:

* **ProcessID –** xác định 1 phiên vân hành OSPF trên Router, mỗi phiên vận hành OSPF sẽ có các bảng định tuyến, bảng hàng xóm, bảng cơ sở dữ liêu link-state,… riêng biệt. Thông số này chỉ có ý nghĩa cục bộ, không nhất thiết phải khớp giữa các Router khác nhau. Giá trị Process ID có thể trải từ 1 - 65535
* **Địa chỉ IP mạng –** địa chỉ mạng cần được quảng bá
* **Wildcard mask –** Một loại mặt nạ đặc biệt được sử dụng để nhóm một loạt địa chỉ IP có cùng 1 tính chất nào đó. Chúng ta sẽ đề cập tới Wildcard mask trong CCNA kỳ 4. Về mặt hình thức hay được sử dụng, Wildcard mask có dạng “ngược lại” subnet mask. Chẳng hạn tiền tố mạng /24 có subnet mask tương ứng 255.255.255.0 thì wildcard mask sẽ biểu diễn dưới dạng 0.0.0.255

Chẳng hạn:

   255.255.255.255

 - 255.255.255.0  Trừ đi Subnet mask tương ứng tiền tố mạng /24

   ---------------

     0.  0.  0. 255  Giá trị Wildcard mask

* **Số hiệu vùng –** thông số xác định vùng mà mạng được quảng bá vào. Một vùng OSPF bao gồm 1 nhóm các Router chạy OSPF chia sẻ cùng thông tin Link-State. Tất cả Router trong cùng 1 vùng sẽ có bảng cơ sở dữ liệu giống hệt nhau, kết quả này được thực hiện bởi các tiến trình đã mô tả phía trên bao gồm lụt toàn bộ bản tin cập nhật Link-State tới tất cả Router trong vùng. Trong phạm vi CCNA chúng ta chỉ đề cập tới OSPF đơn vùng, Số hiệu vùng được mang theo bởi bản tin Hello và là 1 thông số cần phải khớp để 2 Router có thể thiết lập quan hệ Adjacency, do đó sô hiệu vùng trên các Router cần được cấu hình giống nhau.

**Triển khai**

Chúng ta triển khai quảng bá tất cả các liên kết trực tiếp trên tất cả Router vào OSPF

Trên Router R1

R1(config)#router ospf 1

R1(config-router)#network 172.16.1.16 0.0.0.15 area 0

R1(config-router)#network 192.168.10.0 0.0.0.3 area 0

R1(config-router)#network 192.168.10.4 0.0.0.3 area 0

Trên Router R2

R2(config)#router ospf 1

R2(config-router)#network 10.10.10.0 0.0.0.255 area 0

R2(config-router)#network 192.168.10.0 0.0.0.3 area 0

R2(config-router)#network 192.168.10.8 0.0.0.3 area 0

Trên Router R3

R3(config)#router ospf 1

R3(config-router)#network 172.16.1.32 0.0.0.15 area 0

R3(config-router)#network 192.168.10.8 0.0.0.3 area 0

R3(config-router)#network 192.168.10.4.0 0.0.0.3 area 0

**Chú ý:**

* Process ID không nhất thiết phải giống nhau trên 3 Router
* Area ID nhất định phải giống nhau, nhưng không nhất thiết phải cùng bằng 0
* Wildcard mask được tính tương ứng từ giá trị Subnet mask
* Thông tin {địa chỉ mạng IP} {wildcard mask} không nhất thiết phải tương ứng với mạng (con) kết nối trực tiếp trên các Router, mà có thể là mạng major (bao gồm cả subnet mask gốc) bao gồm mạng con, tuy nhiên OSPF vẫn sẽ hành xử theo Classless Routing

**Xác nhận**

Để xác nhận cấu hình OSPF, chúng ta sử dụng 3 câu lệnh show:

* Show ip ospf neighbor
* Show ip route
* Show ip ospf database

Ngoài ra có thê sử dụng các câu lệnh khác để kiểm tra cấu hình OSPF, độc giả tự thử nghiệm và kiểm tra:

* Show ip ospf interface
* Show ip protocol

1. **Kiểm tra quan hệ hàng xóm và Adjacency**

Sau khi cấu hình network, chúng ta có thể kiểm tra định tuyến OSPF. Trước khi kiểm tra bảng định tuyến, chúng ta chờ các Router thiết lập mối quan hệ hàng xóm và Adjacency tương ứng. Kiểm tra bảng hàng xóm trên các Router bằng câu lệnh show ip ospf neighbor

Trên Router R1

R1#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.10.10 0 FULL/ - 00:00:31 192.168.10.6 Serial0/0/1

192.168.10.9 0 FULL/ - 00:00:37 192.168.10.2 Serial0/0/0

Trên Router R2

R2#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.10.5 0 FULL/ - 00:00:36 192.168.10.1 Serial0/0/0

192.168.10.10 0 FULL/ - 00:00:37 192.168.10.10 Serial0/0/1

Trên Router R3

R3#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.10.5 0 FULL/ - 00:00:35 192.168.10.5 Serial0/0/0

192.168.10.9 0 FULL/ - 00:00:31 192.168.10.9 Serial0/0/1

Ta thấy 3 Router từng đôi một thiết lập quan hệ hàng xóm và quan hệ Adjacency (trạng thái FULL) – Chúng ta quan tâm các thông tin:

* Neighbor ID - Router ID của Router hàng xóm
* Pri – Giá trị priority
* State - Trạng thái giữa các Router hàng xóm, trạng thái này phản ánh quá trình thiết quan hệ giữa 2 Router chạy OSPF. Mối quan hệ Adjacency chỉ được xem là thiết lập thành công khi trạng thái là FULL. Các trạng thái khác (sẽ đề cập trong giáo trình khác) có thể kể đến:
  + **Down –** Chưa nghe thấy bản tin Hello từ hàng xóm
  + **Init –** Nghe thây bản tin Hello từ hàng xóm, nhưng vẫn chưa thấy giá trị RouterID của mình trong bản tin hello đó
  + **2Way –** 2 thiết bị trao đổi thành công bản tin Hello
  + **Exstart –** Quá trình xác định Router nào sẽ gửi ra bản tin cập nhật trước
  + **Exchange –** Quá trình Router bắt đầu gửi ra bản tin cập nhật đề nghị
  + **Loading –** Quá trình Router yêu cầu 1 bản tin cập nhật cụ thể và bắt đầu đồng bộ
  + **Full –** 2 Router đã đồng bộ thành công bảng cơ sở dữ liệu
* Dead Time – Khoảng thời gian tối đa còn lại cho phép nhận được bản tin Hello từ hàng xóm trước khi thông báo mất hàng xóm, giá trị này sẽ được reset sau khi Router nhận được 1 bản tin Hello.
* Address – Địa chỉ kết nối trực tiếp của Router hàng xóm, chú ý, địa chỉ này không phải luôn luôn bằng giá trị RouterID
* Interface - Interface ra mà Router thiết lập quan hệ Adjacency với Router hàng xóm

**Chú ý:** Trong môi trường đa truy cập, các Router sẽ chỉ thiết lập hàng xóm với Router làm DR và BDR, các Router làm DROther sẽ chỉ dừng lại ở quan hệ hàng xóm. Chúng ta xem bảng hàng xóm của 1 Router OSPF triển khai trong môi trường đa truy cập:

Router#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

123.123.123.3 1 2WAY/DROTHER 00:00:36 123.123.123.3 FastEthernet0/0

123.123.123.2 1 FULL/DR 00:00:37 123.123.123.2 FastEthernet0/0

123.123.123.1 1 FULL/BDR 00:00:33 123.123.123.1 FastEthernet0/0

1. **Kiểm tra bảng định tuyến**

Sau khi các Router thiết lập quan hệ Adjacency, chúng ta kiểm tra bảng định tuyến để theo dõi việc cập nhật thông tin tuyến đường, kiểm tra bảng định tuyến bằng câu lệnh show ip route

Trên Router R1

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

O 10.10.10.0 [110/65] via 192.168.10.2, 00:07:40, Serial0/0/0

172.16.0.0/28 is subnetted, 2 subnets

C 172.16.1.16 is directly connected, FastEthernet0/0

O 172.16.1.32 [110/65] via 192.168.10.6, 00:04:11, Serial0/0/1

192.168.10.0/30 is subnetted, 3 subnets

C 192.168.10.0 is directly connected, Serial0/0/0

C 192.168.10.4 is directly connected, Serial0/0/1

O 192.168.10.8 [110/128] via 192.168.10.2, 00:04:21, Serial0/0/0

[110/128] via 192.168.10.6, 00:04:21, Serial0/0/1

Trên Router R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

C 10.10.10.0 is directly connected, FastEthernet0/0

172.16.0.0/28 is subnetted, 1 subnets

O 172.16.1.16 [110/65] via 192.168.10.1, 00:00:57, Serial0/0/0

O 172.16.1.32 [110/65] via 192.168.10.10, 00:04:06, Serial0/0/1

192.168.10.0/30 is subnetted, 3 subnets

C 192.168.10.0 is directly connected, Serial0/0/0

O 192.168.10.4 [110/128] via 192.168.10.1, 00:04:06, Serial0/0/0

[110/128] via 192.168.10.10, 00:04:06, Serial0/0/1

C 192.168.10.8 is directly connected, Serial0/0/1

Trên Router R3

R3#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/24 is subnetted, 1 subnets

O 10.10.10.0 [110/65] via 192.168.10.9, 00:04:14, Serial0/0/1

172.16.0.0/28 is subnetted, 1 subnets

O 172.16.1.16 [110/65] via 192.168.10.5, 00:01:20, Serial0/0/0

C 172.16.1.32 is directly connected, FastEthernet0/0

192.168.10.0/30 is subnetted, 3 subnets

O 192.168.10.0 [110/128] via 192.168.10.5, 00:04:14, Serial0/0/0

[110/128] via 192.168.10.9, 00:04:14, Serial0/0/1

C 192.168.10.4 is directly connected, Serial0/0/0

C 192.168.10.8 is directly connected, Serial0/0/1

Ta thấy cả 3 Router đều cập nhật đủ thông tin tuyến đường tới tất cả các mạng xa. Chúng ta xem xét các thông tin trong bảng định tuyến:

|  |  |
| --- | --- |
| Show | Mô tả |
| O | Xác định đây là tuyến đường học qua RIP |
| 192.168.10.8 | Địa chỉ IP mạng đích |
| /24 | Tiền tố mạng của mạng đích |
| [110/65] | Giá trị AD (110) và metric (65) |
| Via 192.168.10.2 | Xác định Router hàng xóm là next-hop hay chính là Router gửi update |
| 00:00:23 | Khoảng thời gian kể từ khi Router nhận update, lần update tiếp theo sẽ trong 7 giây tiếp theo |
| Serial0/0/0 | Outgoing interface để đẩy gói tin tới mạng đích |

1. **Kiểm tra bảng cơ sở dữ liệu**

Kiểm tra bảng dữ liệu trên cả 3 Router, do R1, R2 và R3 đều thuộc 1 vùng (area 0) nên bảng cơ sở dữ liệu sẽ giống hệt nhau, nên chúng ta sẽ chỉ xem xét trên Router R1. Chúng ta kiểm tra với câu lệnh show ip ospf database

R1#show ip ospf database

OSPF Router with ID (192.168.10.5) (Process ID 1)

Router Link-states (Area 0)

Link ID ADV Router Age Seq# Checksum Link count

192.168.10.9 192.168.10.9 891 0x80000005 0x0031f9 5

192.168.10.10 192.168.10.10 891 0x80000005 0x00363a 5

192.168.10.5 192.168.10.5 417 0x80000006 0x001581 5

Ta thấy bảng cơ sở dữ liệu trên các Router bao gồm các thông tin:

* Router ID của Router và vùng OSPF
* 3 bản ghi tương ứng 3 bản tin LSA cho 3 Router
* Bản tin LSA là bản tin LSA loại 1 (Router LSA) mang thông tin về link và trạng thái link kết nối trực tiếp trên từng Router. Để xem thông tin về từng bản tin LSA, chúng ta sử dụng câu lệnh show ip ospf database router {Router ID}

### Router-ID

Mặc định Router sẽ sử dụng địa chỉ IP của interface loopback làm giá trị RouterID. Trong trường hợp không có interface loopback, thì địa chỉ IP lớn nhất của 1 cổng vật lý sẽ được chọn làm RouterID, chẳng hạn Router R1 có 3 interface vật lý tương ứng: fastEthernet0/0: 172.16.1.17, Serial0/0/0: 192.168.10.5, Serial0/0/1: 192.168.10.5, trong đó địa chỉ lớn nhất là 192.168.10.5 sẽ được lựa chọn làm RouterID (bạn xem lại các đoạn show ở phía trên)

Tuy nhiên trong nhiều trường hợp RouterID có ý nghĩa quan trọng trong việc quyết định 1 Router được bình bầu là DR/BDR hoặc các tính năng khác. Yêu cầu đưa ra là người quản trị có thể điều khiển được giá trị này, về mặt cấu trúc của RouterID chúng ta đã đề cập bên trên. Để điều khiển RouterID như vậy ta có 2 cách:

* **Cách 1 :** Cấu hình interface loopback với địa chỉ IP chúng ta mong muốn làm giá trị RouterID
* **Cách 2 :** Cú pháp thay đổi RouterID là router-id được cấu hình trong sub-mode router của OSPF.

1. **Cú pháp**

Đối với **cách 1**, cú pháp tương tự gán địa chỉ IP cho 1 interface bất kỳ. Cisco IOS cho phép người dùng tạo ra số lượng interface loopback …. gần như không giới hạn

RouterX(config)#interface loopback {số hiệu}

RouterX(config-if)# ip address {địa chỉ IP host} {subnet mask}

Trong đó, **{số hiệu}** là 1 số thập phân bất kỳ từ 0 tới 2147483647 (tùy IOS)

Đối với cách 2, cú pháp router-id trong sub-mode Router của OSPF

RouterX(config)#router ospf {processID}

RouterX(config-router)#router-id {giá trị RouterID}

Trong đó **{giá trị RouterID}** được viết dưới dạng thập phân 1 địa chỉ IP với 4 octet. Địa chỉ này có thể là bất kỳ địa chỉ nào, kể cả địa chỉ mạng, địa chỉ broadcast hay 1 địa chỉ Multicast

1. **Triển khai**

Yêu cầu triển khai RouterID tương ứng cho Router R1, R2 và R3 là 10.1.1.1, 10.2.2.2, và 10.3.3.3

**Cách 1:**

Trên Router R1

R1(config)#interface loopback 0

R1(config-if)#ip address 10.1.1.1 255.255.255.255

Trên Router R2

R2(config)#interface loopback 0

R2(config-if)#ip address 10.2.2.2 255.255.255.255

Trên Router R3

R3(config)#interface loopback 0

R3(config-if)#ip address 10.2.2.2 255.255.255.255

Để tối ưu hóa dải địa chỉ, vì interface loopback dù đại diện 1 mạng con nhưng chỉ có 1 địa chỉ IP duy nhất gán vào interface loopback, nên mạng con sử dụng nên là mạng con host – mạng con có subnet mask bằng 255.255.255.255 (hay ~/32), mạng con đặc biệt có 1 địa chỉ duy nhất

**Cách 2**

Trên Router R1

R1(config)#router ospf 1

R1(config-router)#router-id 10.1.1.1

Trên Router R2

R2(config)#router ospf 1

R2(config-router)#router-id 10.2.2.2

Trên Router R3

R3(config)#router ospf 1

R3(config-router)#router-id 10.3.3.3

Cho dù triển khai theo cách nào, để RouterID mới có hiệu lực, yêu cầu tiến trình OSPF hiện tại phải được reset. Chúng ta reset tiến trình OSPF với câu lệnh clear ip ospf process tại mode privilege EXEC. Chúng ta thực hiện trên tất cả Router

R2#clear ip ospf process

Reset ALL OSPF processes? [no]: yes

01:01:44: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.5 on Serial0/0/0 from FULL to DOWN, Neighbor Down: Adjacency forced to reset

01:01:44: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.5 on Serial0/0/0 from FULL to DOWN, Neighbor Down: Interface down or detached

01:01:44: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.10 on Serial0/0/1 from FULL to DOWN, Neighbor Down: Adjacency forced to reset

01:01:44: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.10 on Serial0/0/1 from FULL to DOWN, Neighbor Down: Interface down or detached

01:01:55: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.5 on Serial0/0/0 from LOADING to FULL, Loading Done

01:01:55: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.10.10 on Serial0/0/1 from LOADING to FULL, Loading Done

Chúng ta thấy sau khi reset, Router R2 sẽ mất kết nối hàng xóm tạm thời, và sau đó thiết lập lại với giá trị RouterID mới

1. **Xác nhận**

Để xác nhận RouterID mới trên các Router chạy OSPF, chúng ta có thể sử dụng các câu lệnh:

* Show ip ospf interface
* Show ip ospf neighbor
* Show ip ospf database
* Show ip protocol

Độc giả xem kết quả với các câu lệnh show ip ospf neighbor và show ip ospf database trong ví dụ trước. Với ví dụ này chúng ta theo dõi việc cấu hình OSPF với các câu lệnh còn lại

**Kiểm tra với show ip protocol :**

Trên Router R1

R1#show ip protocol

Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Router ID 10.1.1.1

Number of areas in this router is 1. 1 normal 0 stub 0 nssa

Maximum path: 4

192.168.10.0 0.0.0.3 area 0

192.168.10.4 0.0.0.3 area 0

172.16.1.16 0.0.0.15 area 0

Routing Information Sources:

Gateway Distance Last Update

192.168.10.5 110 00:05:06

192.168.10.9 110 00:13:50

192.168.10.10 110 00:05:01

Distance: (default is 110)

Các thông số cần quan tâm từ kết quả show ip protocol

|  |  |
| --- | --- |
| Show | Ý nghĩa |
| OSPF 1 | giao thức định tuyến sử dụng là OSPF, processID là 1 |
| Router ID 10.1.1.1 | Router ID được Router R1 sử dụng sau khi thay đổi, giá trị mặc định trong ví dụ trước là 192.168.10.5 |
| Number of Areas in this Router is 1 | Trong topology triển khai OSPF đơn vùng nên chỉ có 1 vùng OSPF duy nhât là Area 0 |
| Routing for Networks: | Các mạng Router quảng bá |
| Routing information source | ĐỊa chỉ nguồn của các bản tin cập nhật OSPF nhận được. Chú ý đây là địa chỉ nguồn của bản tin cập nhật, không phải Router ID của Router |
| Distane: (default is 110) | Giá trị AD của OSPF bằng 110 |

Chú ý: Router ID là giá trị định danh 1 Router chạy OSPF trong toàn bộ miền định tuyến OSPF. Do đó không chấp nhận việc trùng lặp giá trị RouterID. Router sẽ phát hiện sự trùng lặp thông qua bản tin Hello. Khi xuất hiện lỗi trùng địa chỉ Router ID, yêu cầu người quản trị điều chỉnh lại giá trị Router ID bằng 1 trong 2 cách trên. Thông báo lỗi trùng router ID:

%OSPF-4-DUP\_RTRID1: Detected router with duplicate router ID

**Kiểm tra với show ip ospf interface {loại interface}**

R1#show ip ospf interface s0/0/0

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.1/30, Area 0

Process ID 1, Router ID 192.168.10.5, Network Type POINT-TO-POINT, Cost: 64

Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0

No designated router on this network

No backup designated router on this network

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

Hello due in 00:00:03

Index 3/3, flood queue length 0

Next 0x0(0)/0x0(0)

Last flood scan length is 1, maximum is 1

Last flood scan time is 0 msec, maximum is 0 msec

Neighbor Count is 1 , Adjacent neighbor count is 1

Adjacent with neighbor 1.1.1.1

Suppress hello for 0 neighbor(s)

Các thông tin cần quan tâm từ show ip ospf interface

|  |  |
| --- | --- |
| Show | Ý nghĩa |
| Area 0 | Chỉ số vùng |
| Router ID 192.168.10.5 | Là Router ID của R1 |
| Network type POINT-TO-POINT | Xác định loại môi trường là điểm-điểm |
| Cost : 64 | Giá trị metric của link Serial0/0/0 |
| Hello 10, Dead 40 | Chu kỳ gửi Hello và thời gian xác nhận mất hàng xóm |
| Neighbor count is 1 | Xác định số lượng Router chạy OSPF trên link |
| Adjacency with neighbor 1.1.1.1 | RouterID của Router hàng xóm đã thiết lập Adjacency |

### Điều chỉnh metric

Metric được sử dụng bởi OSPF được gọi là cost, cost được gán cho từng link kết nối trực tiếp và có ý nghĩa khi được nhìn là metric của outgoing interface trong tuyến đường tới 1 mạng đích, do đó trên cùng 1 link, cost của 2 Router có thể khác nhau. Công thức tính cost:

Cost =

Trong đó:

* **Băng thông tham khảo –** giá trị tham chiếu của OSPF, mặc định bằng 100Mbps hay 100,000,000 (bps)
* **Băng thông trên link –** Băng thông nhìn thấy khi show ip interface

Như vậy để điều chỉnh cost ta có 3 cách:

* **Cách 1 -** Điều chỉnh trực tiếp cost trên link
* **Cách 2 -** Điều chỉnh giá trị băng thông tham khảo
* **Cách 3 -** Điều chỉnh băng thông trên link

1. **Cú pháp**

**Cách 1 –** Điều chỉnh trực tiếp cost là 1 tính năng của OSPF được triển khai trên từng interface, cú pháp ip ospf cost. Cú pháp hoàn chỉnh như sau:

RouterX(config)#interface {số hiệu interface}

RouterX(config-if)#ip ospf cost {giá trị cost}

Trong đó, {giá trị cost} là cost mong muốn gán trực tiếp trên link. Khi sử dụng phương pháp này, thì cost sẽ không tuân theo công thức gốc nữa, nhưng sẽ được sử dụng bởi OSPF để tính toán metric tuyến đường. Phương pháp này sẽ luôn được ưu tiên so với 2 cách điều chỉnh cost còn lại

**Cách 2 –** Điều chỉnh giá trị băng thông tham khảo. Đây là 1 tính năng OSPF được triển khai trong sub-mode Router của OSPF, cú pháp auto-cost reference-bandwidth. Cú pháp hoàn chỉnh như sau:

RouterX(config)#router ospf {ProcessID}

RouterX(config-router)#auto-cost reference-bandwidth {giá trị băng thông tham khảo (Mbps)}

Trong đó, {Giá trị băng thông tham khảo (Mbps)} sẽ được sử dụng để áp dụng vào công thức gốc ban đầu. Mặc định giá trị này băng 100 (Mbps). Giá trị này sẽ được sử dụng khi tính toán cost cho tất cả Interface chạy OSPF. Băng thông tham khảo thường được điều chỉnh khi trên Router có các cổng vật lý có băng thông cao hơn 100Mbps, chẳng hạn cổng GigaEthernet (1000 Mbps) hoặc các chuẩn Ethernet cáp quang

**Cách 3 –** Điều chỉ băng thông trên link. Băng thông trên link là thuộc tính của 1 interface, do đó được cấu hình trong sub-mode interface. Cú pháp bandwidth. Cú pháp hoàn chỉnh:

RouterX(config)#interface {số hiệu interface}

RouterX(config-if)#bandwidth {giá trị băng thông trên link (kbps)}

Trong đó {giá trị băng thông trên link (kbps)} sẽ được sử dụng để áp dụng vào công thức gốc ban đầu. Giá trị băng thông trên link mặc định nhận các giá trị tùy thuộc vào chuẩn kết nối vật lý của link, chẳng hạn chuẩn 100BaseTX có băng thông trên link mặc định bằng 100Mbps, hoặc kết nối WAN T1 có băng thông trên link mặc định bằng 1544 Mbps

1. **Triển khai**

Yêu cầu cấu hình cost trên các link kết nối trực tiếp của Router thỏa mãn:

* Yêu cầu 1 - Cost trên link giữa R1 và R2 phản ánh tốc độ truyền 64kbps
* Yêu cầu 2 - Cost trên link giữa R2 và R3 phản ánh tốc độ truyền 128kbps
* Yêu cầu 3 - Cost trên link giữa R1 và R3 phản ánh tốc độ truyền 256 kbps

Chúng ta sẽ triển khai cả 3 cách tương ứng với 3 yêu cầu trên

Trên Router R1 (yêu cầu 1 với cách 1, yêu cầu 3 với cách 3)

R1(config)#interface s0/0/0

R1(config-if)#ip ospf cost 1562

R1(config-if)#exit

R1(config)#interface s0/0/1

R1(config-if)#bandwidth 256

Trên Router R2 (yêu cầu 1 với cách 1, yêu cầu 2 với cách 2)

R2(config)#interface s0/0/0

R2(config-if)#ip ospf cost 1562

R2(config-if)#exit

R2(config)#router ospf 1

R2(config)#auto-cost reference-bandwidth 120

Trên Router R3 (Sử dụng cách 3)

R3(config)#interface s0/0/0

R3(config-ìf)#bandwidth 256

R3(config-if)#exit

R3(config)#interface s0/0/1

R3(config-if)#bandwitdh 128

1. **Xác nhận**

Để xác nhận, chúng ta sử dụng câu lệnh show ip ospf interface

Trên Router R1

R1#show ip ospf interface s0/0/0

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.1/30, Area 0

Process ID 1, Router ID 10.1.1.1, Network Type POINT-TO-POINT, Cost: 1562

<output omitted>

Cost của interface serial 0/0/0 trên R1 bằng 1562, tương ứng 108 / 64,000 (hay 64 kbps – chính là băng thông tương ứng với tốc độ truyền)

R1#show ip ospf interface s0/0/0

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.1/30, Area 0

Process ID 1, Router ID 10.1.1.1, Network Type POINT-TO-POINT, Cost: 390

<output omitted>

Cost của interface serial 0/0/1 trên R1 bằng 390, tương ứng 108 / 256,000 (hay 256 kbps – chính là băng thông tương ứng với tốc độ truyền)

Trên Router R2

R2#show ip ospf interface s0/0/0

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.2/30, Area 0

Process ID 1, Router ID 10.2.2.2, Network Type POINT-TO-POINT, Cost: 1562

<output omitted>

Cost của interface serial 0/0/0 trên R2 bằng 1562, tương ứng 108 / 64,000 (hay 64 kbps – chính là băng thông tương ứng với tốc độ truyền)

R2#show ip ospf interface s0/0/1

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.9/30, Area 0

Process ID 1, Router ID 10.2.2.2, Network Type POINT-TO-POINT, Cost: 781

<output omitted>

Cost của interface serial 0/0/1 trên R2 bằng 781, tương ứng 120\*106 / 1544,000 hay 100\*106 / 128,000 (hay 128 kbps – chính là băng thông tương ứng với tốc độ truyền) – Trong trường hợp này, chúng ta đã điều chỉnh giá trị băng thông tham chiếu (120\*106), do đó với công thức mới chúng ta vẫn sử dụng băng thông trên link cũ (1544 kbps), kết quả này cũng tương ứng khi dùng băng thông tham chiếu cũ (100\*106), nhưng đổi băng thông trên link mới (từ 1544kbps thành 128kbs) nếu theo cách 3

Trên Router R3

R3#show ip ospf interface s0/0/0

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.6/30, Area 0

Process ID 1, Router ID 10.3.3.3, Network Type POINT-TO-POINT, Cost: 390

<output omitted>

Cost của interface serial 0/0/0 trên R3 bằng 390, tương ứng 108 / 256,000 (hay 256 kbps – chính là băng thông tương ứng với tốc độ truyền)

R3#show ip ospf interface s0/0/1

Serial0/0/0 is up, line protocol is up

Internet address is 192.168.10.10/30, Area 0

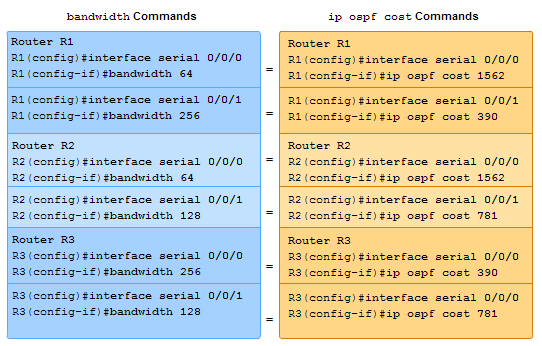
Process ID 1, Router ID 10.3.3.3, Network Type POINT-TO-POINT, Cost: 781

<output omitted>

Cost của interface serial 0/0/1 trên R3 bằng 781, tương ứng 108 / 128,000 (hay 128 kbps – chính là băng thông tương ứng với tốc độ truyền)

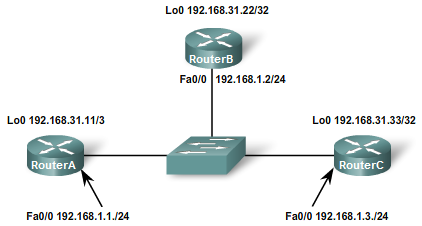
**Chú ý:**

* Khi sử dụng cách 1 (cấu hình cost trực tiếp), Router sẽ không sử dụng công thức tính cost từ băng thông tham khảo và băng thông trên link nữa.
* Có thể sử dụng 1 trong 3 cách trên cho tất cả trường hợp. Chúng ta có giải pháp tương ứng giữa cách 1 và cách 3 như sau:

  
*Hình – Quy đổi giữa điều chỉnh băng thông và điều chỉnh cost*

### Điều chỉnh bình bầu DR/BDR

Cho topology với môi trường đa truy cập:



Quá trình bình bầu DR/BDR là quá trình luôn diễn ra trong môi trường đa truy cập. Vài trò của DR và BDR là tập trung và phân phối lại bản tin Link-State, trong nhiều trường hợp, người quản trị phải chủ động xác định vị trí của Router làm DR và BDR

Chúng ta nhớ lại quá trình bình bầu DR/BDR dựa vào 2 yếu tố:

* So sánh thông số priority
* So sánh giá trị RouterID

Theo đó, chúng ta có thể điều chỉnh Router làm DR dựa vào 2 thông số trên, tương ứng có 2 cách làm là thay đổi priority và thay đổi RouterID

**Yêu cầu:** Cấu hình sao cho Router A là DR và Router B là BDR, Router C là DROther

**Cú pháp**

**Cách 1 –** ĐIều chỉnh priority trên interface chạy OSPF. Chúng ta cũng nhớ rằng khi 1 link của Router có Priority bằng 0 thì link đó không bao giờ trở thành DR hoặc BDR

RouterX(config)#interface {loại/slot}

RouterX(config-if)#ip ospf priority {giá trị}

**Cách 2 –** Điều chỉnh RouterID. Để điều chỉnh RouterID, chúng ta lại có 2 cách, tuy nhiên trong trường hợp này, chúng ta sẽ sử dụng chức năng gán tĩnh giá trị RouterID

RouterX(config)#router ospf {processID}

RouterX(config-router)#router-id {địa chỉ IP}

**Triển khai**

Các tiến trình triển khai:

1. Kiểm tra trạng thái DR/BDR hiện tại
2. Theo dõi quá trình thay đổi vai trò DR/BDR
3. Điều chỉnh Priority để thay đổi vai trò DR
4. Điều chỉnh RouterID để thay đổi vai trò BDR

**Tiến trình 1 –** Kiểm tra trạng thái DR/BDR hiện tại

Chúng ta nhớ rằng một khi OSPF được kích hoạt trên 1 interface trỏ ra 1 miền đa truy cập, quá trình bình bầu DR/BDR sẽ diễn ra ngay, chúng ta xem trạng thái DR/BDR của topology trên khi bắt đầu tiến trình định tuyến. Trước khi kiểm tra, chúng ta chờ mạng hội tụ

00:00:50: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.31.22 on FastEthernet0/0 from LOADING to FULL, Loading Done

!  
00:00:50: %OSPF-5-ADJCHG: Process 1, Nbr 192.168.31.11 on FastEthernet0/0 from LOADING to FULL, Loading Done

Đoạn log thể hiện Router RC đã thiết lập xong quan hệ Adjacency với Router RA và RB. Chúng ta kiểm tra trạng thái và vai trò của các Router hiện tại với câu lệnh show ip ospf neighbor

RA#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.22 1 FULL/BDR 00:00:35 192.168.1.2 FastEthernet0/0

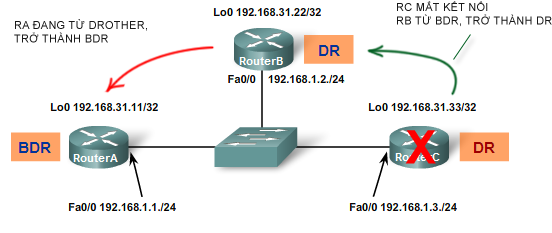
192.168.31.33 1 FULL/DR 00:00:35 192.168.1.3 FastEthernet0

Ta thấy Router RC hiện đang là DR và Router RB là BDR. Cả 3 Router RA, RB, RC có giá trị priority bằng nhau, vì thế RouterID sẽ được đem ra so sánh, trong đó Router RC có RouterID bằng 192.168.31.33 lớn nhất, do đó trở thành DR. Lý giải tương tự, Router RB là BDR, còn Router RA là DROther

**Tiến trình 2 –** Theo dõi quá trình thay đổi vai trò DR/BDR

Chúng ta sẽ xét 4 trường hợp:

**Router RC mất kết nối –** Router RC mất kết nối, các Router RA, RB phát hiện thấy mất kết nối tới DR, do đó ngay lập tức RB đang từ BDR sẽ trở thành DR. Router R1 triển khai bình bầu lại DR/BDR, nhưng do trong mạng không còn Router nào khác, không có “đối thủ” nên RA trở thành BDR



*Hình – DR mất kết nối, BDR lên thay*

Chúng ta kiểm tra trạng thái DR/BDR trong mạng với show ip ospf neighbor

RA#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.22 1 FULL/DR 00:00:24 192.168.1.2 FastEthernet0/0

RB#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.11 1 FULL/BDR 00:00:19 192.168.1.1 FastEthernet0/0

**1 Router mới RD được đưa vào mạng –** Router RD kết nối vào mạng và thiết lập quan hệ Adjacency với các Router khác. Interface fastEhternet của Router RD có Priority bằng 2 và RouterID bằng 192.168.31.44, tốt hơn cả RA và RB:

RA#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.22 1 FULL/DR 00:00:27 192.168.1.2 FastEthernet0/0

192.168.31.44 1 FULL/DROTHER 00:00:12 192.168.1.4 FastEthernet0/0

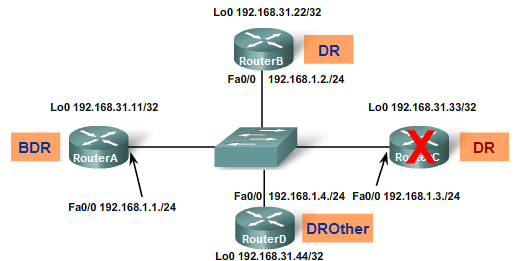
RD#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.22 1 FULL/DR 00:00:14 192.168.1.2 FastEthernet0/0

192.168.31.11 2 FULL/BDR 00:00:04 192.168.1.1 FastEthernet0/0

Ta thấy, khi 1 Router mới tham gia định tuyến, cho dù Router đó có priority hay RouterID tốt bao nhiêu, nhưng Router đó vẫn không thể trở thành DR và BDR. Quá trình bình bầu lại chỉ diễn ra khi tất cả Router bắt đầu lại tiến trình OSPF, hoặc khi trong mạng đang trống 1 trong 2 vị trí DR/BDR, lúc đó quá trình bình bầu chỉ diễn ra với các DROther



*Hình – Router mới thêm vào không thể thay đổi trạng thái DR/BDR hiện tai*

**Router RC kết nối lại –** Router RC là “cựu” DR, tuy nhiên, cũng như trường hợp của Router RD, cho dù là Router nào khi kết nối lại vào tiến trình OSPF cũng không làm thay đổi trạng thái DR/BDR hiện tại:

RD#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.22 1 FULL/DR 00:00:26 192.168.1.2 FastEthernet0/0

192.168.31.11 2 FULL/BDR 00:00:05 192.168.1.1 FastEthernet0/0

192.168.31.33 1 2WAY/DROTHER 00:00:17 192.168.1.3 FastEthernet0/0

Ngoài ra ta thấy, quan hệ Adjacency chỉ thiết lập giữa các DROther với DR và BDR. Giữa các Router làm DROther, trạng thái hàng xóm dừng lại là 2WAY, hay các Router nhìn thấy RouterID của nhau tron bản tin OSPF Hello

**Router RA mất kết nối –** Thời điểm hiện tại, Router Ra đang là BDR. Trong trường hợp RA mất kết nối, các Router DROther sẽ “thi đua” vào vị trí BDR đang trống của Router RA để lại. Các thông số so sánh với nhau vẫn tương tự, đó là sử dụng priority trước, sau đó nếu Priority bằng nhau thì RouterID sẽ được dùng để quyết định:

RB#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.44 2 FULL/BDR 00:00:22 192.168.1.4 FastEthernet0/0

192.168.31.33 1 FULL/DROTHER 00:00:19 192.168.1.3 FastEthernet0/0

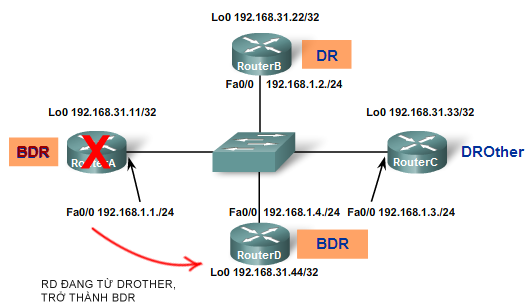
RD#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.22 1 FULL/DR 00:00:26 192.168.1.2 FastEthernet0/0

192.168.31.33 1 FULL/DROTHER 00:00:17 192.168.1.3 FastEthernet0/0

Trong trường hợp trên, Router RD đã giành chiến thắng với giá trị Priority cao hơn priority của Router RC



*Hình – Router RD trở thành BDR, khi Router RA mất kết nối*

**Router RB mất kết nối –** Router RB hiện tại đang là DR, và khi DR mất kết nối, chúng ta quay lại trường hợp đầu tiên, BDR sẽ trở thành DR, và các Router DROther còn lại sẽ trao đổi bản tin Hello để xác định xem ai sẽ trở thành BDR. Trong trường hợp này, mạng chỉ còn Router RC và RD, Router RD trở thành DR, và RC trở thành BDR

**Tiến trình 3 -** Điều chỉnh Priority để thay đổi vai trò DR:

Chúng ta quay lại topology ban đầu, với 3 Router RA, RB và RC trong đó

* RC là DR
* RB là BDR
* RA là DROther

Yêu cầu là cấu hình để Router RA trở thành DR và Router RC trở thành BDR

Để Router RA trở thành DR, chúng ta cấu hình sao cho interface fastEthernet của Router RA có priority cao nhất:

RA(config)#interface fastEthernet 0/0

RA(config-if)#ip ospf priority 100

RA#Reload or use "clear ip ospf process" command, for this to take effect

RA#clear ip ospf pr

Reset ALL OSPF processes? [no]: y

Để thông sô Priority mới áp dụng, chúng ta cần thiết lập lại tiến trình OSPF trên Router RA. Có 2 cách để thiết lập lại tuyến trình, đó là sử dụng câu lệnh clear ip ospf process hoặc thực hiện shutdown sau đó no shutdown interface fastEthernet 0/0

Chúng ta kiểm tra lại trạng thái DR/BDR trong mạng:

RA#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.22 1 FULL/BDR 00:00:22 192.168.1.2 FastEthernet0/0

192.168.31.33 1 FULL/DROTHER 00:00:19 192.168.1.3 FastEthernet0/0

RB#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

192.168.31.11 100 FULL/DR 00:00:14 192.168.1.1 FastEthernet0/0

192.168.31.33 1 FULL/DROTHER 00:00:03 192.168.1.3 FastEthernet0/0

Ta thấy, giá trị priority trên interface fastEthernet 0/0 của Router RA bằng 100, và lúc này, Router RA đóng vai trò DR. Chúng ta thấy mặc dù tiến trình được thực thi lại, tuy nhiên vai trò của BDR vẫn không đổi, cho dù Router RC có RouterID tối ưu hơn RB. Hay nói cách khác, vị trí của BDR không bị ảnh hưởng bởi câu lệnh thực thi lại tiến trình. Để BDR thay đổi vị trí trong mạng, thiết bị cần được thực thi lại tiến trình phải là BDR hiện tại, hay trong trường hợp này là Router RB

RB(config)#router ospf 1

RB(config-router)#router-id 100.100.100.10

RB(config)#

Reload or use "clear ip ospf process" command, for this to take effect

RB(config)#interface fastEthernet 0/0

RB(config-if)#shutdown

<output omitted>

RB(config-if)#no shutdown

Sau khi thực thi lại tiến trình, chúng ta kiểm tra bảng hàng xóm:

RA#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

100.100.100.10 1 FULL/DROTHER 00:00:11 192.168.1.2 FastEthernet0/0

192.168.31.33 1 FULL/BDR 00:00:16 192.168.1.3 FastEthernet0/0

RC#show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface

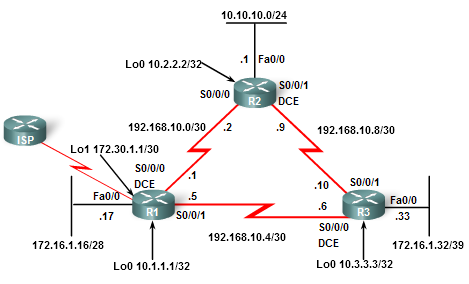
192.168.31.11 100 FULL/DR 00:00:08 192.168.1.1 FastEthernet0/0

100.100.100.10 1 FULL/DROTHER 00:00:12 192.168.1.2 FastEthernet0/0

Chúng ta thấy vai trò của BDR lúc này đã chuyển từ Router RB sang RC, đúng như yêu cầu.

### Default route

Cho topology



**Bài toán:** Nội mạng triển khai định tuyến nội với sao cho tất cả các Router đều có thể truy cập tới các mạng LAN. Doanh nghiệp triển khai kết nối tới ISP, Router biên là Router R1 kết nối trực tiếp với Router ISP.

**Yêu cầu:** Cấu hình định tuyến động với OSPF sau cho mạng hội tụ và các Router R1, R2 và R3. Router R1 quảng bá tuyến đường mặc định vào OSPF. Đây là bài toán định tuyến cơ bản.

**Cấu hình khởi tạo**

Cấu hình khởi tạo định tuyến động OSPF sau cho nội mạng hội tụ, chúng ta triển khai cấu hình quảng bá tất cả tuyến đường kết nối trực tiếp trừ loopback

Trên Router R1

R1(config)#router ospf 1

R1(config-router)#network 172.16.1.16 0.0.0.15 area 0

R1(config-router)#network 172.30.1.1 0.0.0.3 area 0

R1(config-router)#network 192.168.10.0 0.0.0.3 area 0

R1(config-router)#network 192.168.10.4 0.0.0.3 area 0

Trên Router R2

R2(config)#router ospf 1

R2(config-router)#network 10.10.10.0 0.0.0.255 area 0

R2(config-router)#network 192.168.10.0 0.0.0.3 area 0

R2(config-router)#network 192.168.10.8 0.0.0.3 area 0

Trên Router R3

R3(config)#router ospf 1

R3(config-router)#network 172.16.1.32 0.0.0.7 area 0

R3(config-router)#network 192.168.10.4 0.0.0.3 area 0

R3(config-router)#network 192.168.10.8 0.0.0.3 area 0

**Cú pháp:**

Tương tự RIP, cú pháp triển khai quảng bá tuyến mặc định của OSPF rất đơn giản

RouterX(config)#router ospf {processID}

RouterX(config-router)#default-information originate

**Triển khai:**

Câu lệnh default-information originate được cấu hình trên Router biên, mục đích để Router biên quảng bá 1 tuyến đường mặc định vào toàn bộ mạng, thay vì phải cấu hình định tuyến tĩnh trên tất cả Router. Trong OSPF, thuật ngữ để chỉ 1 Router biên đứng giữa 1 miền chạy OSPF và 1 miền định tuyến khác OSPF là ASBR (Autonomous System Boundary Router). Chúng ta cấu hình trên R1

R1(config)#router ospf 1

R1(config-router)#defautl-information originate

Tương tự RIP, để Router R1 quảng bá tuyến đường mặc định vào OSPF, bản thân Router R1 phải có tuyến default route trước đó. Chúng ta cấu hình định tuyến tĩnh trên Router R1 trỏ default- route ra Router ISP (trong ví dụ, interface trỏ ra ISP là loopback1)

R1(config)#ip route 0.0.0.0 0.0.0.0 loopback1

**Xác nhận:**

Chúng ta kiểm tra bảng định tuyến của tất cả Router:

Trên Router R1

R1#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

C 10.1.1.1/32 is directly connected, Loopback0

O 10.10.10.0/24 [110/1172] via 192.168.10.6, 00:03:33, Serial0/0/1

172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks

C 172.16.1.16/28 is directly connected, FastEthernet0/0

O 172.16.1.32/29 [110/391] via 192.168.10.6, 00:03:43, Serial0/0/1

172.30.0.0/30 is subnetted, 1 subnets

C 172.30.1.0 is directly connected, Serial0/1/0

192.168.10.0/30 is subnetted, 3 subnets

C 192.168.10.0 is directly connected, Serial0/0/0

C 192.168.10.4 is directly connected, Serial0/0/1

O 192.168.10.8 [110/1171] via 192.168.10.6, 00:03:43, Serial0/0/1

S 0.0.0.0/0 is directly connected, loopback1

Trên Router R2

R2#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

C 10.2.2.2/32 is directly connected, Loopback0

C 10.10.10.0/24 is directly connected, FastEthernet0/0

172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks

O 172.16.1.16/28 [110/1172] via 192.168.10.10, 00:07:13, Serial0/0/1

O 172.16.1.32/29 [110/782] via 192.168.10.10, 00:07:13, Serial0/0/1

192.168.10.0/30 is subnetted, 3 subnets

C 192.168.10.0 is directly connected, Serial0/0/0

O 192.168.10.4 [110/1171] via 192.168.10.10, 00:07:13, Serial0/0/1

C 192.168.10.8 is directly connected, Serial0/0/1

O\*E2 0.0.0.0/0 [110/20] via 192.168.10.10, 00:00:13, Serial0/0/1

Trên Router R3

R3#show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP

<output omitted>

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks

C 10.3.3.3/32 is directly connected, Loopback0

O 10.10.10.0/24 [110/782] via 192.168.10.9, 00:09:12, Serial0/0/1

172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks

O 172.16.1.16/28 [110/391] via 192.168.10.5, 00:09:12, Serial0/0/0

C 172.16.1.32/29 is directly connected, FastEthernet0/0

192.168.10.0/30 is subnetted, 3 subnets

O 192.168.10.0 [110/1952] via 192.168.10.5, 00:09:12, Serial0/0/0

C 192.168.10.4 is directly connected, Serial0/0/0

C 192.168.10.8 is directly connected, Serial0/0/1

O\*E2 0.0.0.0/0 [110/20] via 192.168.10.5, 00:00:27, Serial0/0/0

Các thông tin cần quan tâm:

|  |  |
| --- | --- |
| Show | Ý nghĩa |
| O | Tuyến học từ OSPF |
| \* | Tuyến mặc định (default route) |
| E2 | Loại tuyến đường là External 2 |
| 0.0.0.0/0 | Địa chỉ tuyến mặc định |
| [110/20] | AD của OSPF và cost mặc định cho tuyến E2 |
| Serial0/0/0 | Outgoing interface |

Các tuyến đường học từ 1 giao thức khác vào OSPF được gọi là các tuyến ngoại (external route). Đối với OSPF không sử dụng nhiều AD để định nghĩa ra tuyến nội / tuyến ngoại như EIGRP hoặc BGP, nhưng OSPF sử dụng nhiều loại tuyến đường khác nhau để chi ra cách học các tuyến đó. Tuyến E2 là tuyến mặc định được sử dụng khi quảng bá 1 mạng từ giao thức khác vào, metric mặc định của E2 bằng 20. Ngoài tuyến E2, OSPF hỗ trợ 1 kiểu tuyến ngoại khác là E1, chúng ta sẽ đề cập trong các tài liệu khác

Ta thấy, lúc này trên tất cả Router, trong bảng định tuyến đều đã có tuyến đường mặc định

### Điều chỉnh OSPF timer

OSPF sử dụng Hello-time để xác định chu kỳ gửi Hello và Dead-time để xác định thời gian chờ tối đa trước khi xác nhận mất kết nối tới hàng xóm. Hello-time và Dead-time có thể được sử dụng để tăng tốc độ hội tụ trong mạng

**Cú pháp:**

RouterX(config-if)#ip ospf hello-interval {thời gian - giây}

RouterX(config-if)#ip ospf dead-interval {thời gian - giây}

**Chú ý:** OSPF yêu cầu 2 Router phải khớp nhau về Hello-time và Dead-time để có thể thiết lập quan hệ Adjacency

**Triển khai:**

Triển khai sao cho R1 và R3 có thể phát hiện mất kết nối hàng xóm trong khoảng thời gian 20 giây, và chu kỳ Hello bằng 5 giây. Trước khi thay đổi, chúng ta kiểm tra giá trị Hello-time và Dead-time hiện tại trên các Router bằng câu lệnh show ip ospf interface

R2>show ip ospf interface s0/0/0

Serial0/0/0 is up, line protocol is up

<output omitted>

No backup designated router on this network

Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5

Hello due in 00:00:09

Index 3/3, flood queue length 0

<output omitted>

Chúng ta thấy hiện tại Hello-interval bằng 10 giây, còn Dead-interval bằng 40 giây

Trên Router R1

R1(config)#interface s0/0/0

R1(config-if)#ip ospf hello-interval 5

R1(config-if)#ip ospf dead-interval 20

Trên Router R3

R3(config)#interface s0/0/0

R3(config-if)#ip ospf hello-interval 5

R3(config-if)#ip ospf dead-interval 20

Ngay khi chúng ta vừa thay đổi Hello-time và Dead-time trên Router R1 thì R1 ngắt quan hệ Adjacency với R3

R1(config-if)#

00:03:14: %OSPF-5-ADJCHG: Process 1, Nbr 10.3.3.3 on Serial0/0/1 from FULL to DOWN, Neighbor Down: Dead timer expired

00:03:14: %OSPF-5-ADJCHG: Process 1, Nbr 10.3.3.3 on Serial0/0/1 from FULL to DOWN, Neighbor Down: Interface down or detached

Chúng ta chờ 1 chút để mạng hội tụ

**Xác nhận:**

Để kiểm tra khoảng thời gian Hello-time hoặc Dead-time mới, chúng ta sử dụng câu lệnh show ip ospf interface một lần nữa

## Tổng kết

Kết thúc chương 10, chúng ta đã có cái nhìn tương đối chi tiết về giao thức Link-State đầu tiên là OSPF. Đây là 1 giao thức mở, rât mạnh mẽ và phổ biến hiện nay, các vấn đề cần quan tâm:

* Các đặc điểm link-state của OSPF
* Các tiến trình vân hành và thuật toán SPF
* Bản tin Hello
* Thiết lập mối quan hệ hàng xóm và Adjacency
* Quá trình bình bầu DR và BDR
* Các loại bản tin OSPF, các loại môi trường OSPF
* Cấu hình quảng bá 1 mạng
* Điều chỉnh thông số RouterID và tiến trình bình bầu DR/BDR
* Quảng bá default-route
* Điều chỉnh metric OSPF