

**OSPF – Open Shortest Path First**, là một giao thức định tuyến link – state điển hình. Đây là một giao thức được sử dụng rộng rãi trong các mạng doanh nghiệp có kích thước lớn. Trong chương trình CCNA, đây cũng là một chủ đề chính được đề cập nhiều. Do đó, nắm vững những nguyên tắc hoạt động của OSPF sẽ giúp các bạn đang theo học chương trình CCNA hoàn thành tốt kỳ thi lấy chứng chỉ quốc tế CCNA cũng như đáp ứng tốt nhu cầu công việc trong thực tế.

### Một số đặc điểm chính của giao thức OSPF:

1. OSPF là một giao thức link – state điển hình. Mỗi router khi chạy giao thức sẽ gửi các trạng thái đường link của nó cho tất cả các router trong vùng (area). Sau một thời gian trao đổi, các router sẽ đồng nhất được bảng cơ sở dữ liệu trạng thái đường link (Link State Database – LSDB) với nhau, mỗi router đều có được “bản đồ mạng” của cả vùng. Từ đó mỗi router sẽ chạy giải thuật Dijkstra tính toán ra một cây đường đi ngắn nhất (Shortest Path Tree) và dựa vào cây này để xây dựng nên bảng định tuyến.

2. OSPF có AD = 110.

3. Metric của OSPF còn gọi là cost, được tính theo bandwidth trên cổng chạy OSPF.

4. OSPF chạy trực tiếp trên nền IP, có protocol – id là 89.

5. OSPF là một giao thức chuẩn quốc tế, được định nghĩa trong RFC – 2328.

Ta cùng review hoạt động của OSPF thông qua các bước hoạt động như sau:

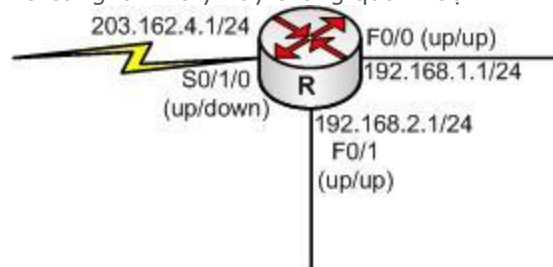
1. Bầu chọn Router – id.
2. Thiết lập quan hệ láng giềng (neighbor).
3. Trao đổi LSDB.
4. Tính toán xây dựng bảng định tuyến.

#### Router – id:

Đầu tiên, khi một router chạy OSPF, nó phải chỉ ra một giá trị dùng để định danh duy nhất cho nó trong cộng đồng các router chạy OSPF. Giá trị này được gọi là Router – id.

Router – id trên router chạy OSPF có định dạng của một địa chỉ IP. Mặc định, tiến trình OSPF trên mỗi router sẽ tự động bầu chọn giá trị router – id là địa chỉ IP cao nhất trong các interface đang active, ưu tiên cổng loopback.

Ta cùng làm rõ ý này thông qua ví dụ:

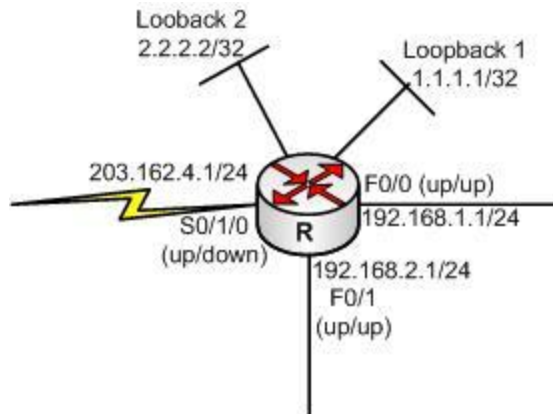


Hình 1 – Bầu chọn router – id (1).

Khi cho router R tham gia OSPF (xem hình 1), router R phải bầu chọn ra một ‘nick name’ để định danh R khi R chạy OSPF. Vì ‘nick name’ này có định dạng của một địa chỉ IP nên R sẽ lấy một trong các địa chỉ IP trên nó để làm Router – id. Như đã nói ở trên, chỉ địa chỉ của các interface đang active, tức là ở trạng thái up/up (status up, line protocol up) mới được tham gia bầu chọn. Ta thấy trên hình, chỉ có hai cổng F0/0 và F0/1 của R là up/up nên router R sẽ chỉ xem xét hai địa chỉ trên hai cổng này là 192.168.1.1 và 192.168.2.1. Để xác định trong hai địa chỉ này, địa chỉ nào là cao hơn, R tiến hành so sánh hai địa chỉ này theo từng octet từ trái sang phải, địa chỉ nào có octet đầu tiên lớn hơn được xem là lớn hơn. Ta thấy, với cách so sánh này, địa chỉ 192.168.2.1 được xem là lớn hơn địa chỉ 192.168.1.1 nên nó sẽ được sử dụng để làm router – id. Vậy R sẽ tham gia OSPF với giá trị ‘nick name’ – router id là 192.168.2.1.

Ta cũng thấy trong 03 địa chỉ xuất hiện ở trên hình 1, địa chỉ 203.162.4.1 của cổng serial S0/1/0 trên router R là lớn nhất nhưng vì cổng này down nên không được tham gia bầu chọn.

Cũng ví dụ trên nhưng lần này trên router R có thêm các interface loopback:



Hình 2 – Bầu chọn router – id (2).

Khi ta bật OSPF trên router R, R xúc tiến việc bầu chọn router – id. Vì lần này có các interface loopback nên R sẽ bỏ qua, không xem xét các địa chỉ của các interface vật lý. Hai địa chỉ của hai interface loopback 1 và 2 sẽ được so sánh để chọn ra router – id cho router R, và ta thấy rõ ràng  $2.2.2.2 > 1.1.1.1$  nên router R sẽ chọn 2.2.2.2 làm router – id khi tham gia OSPF. Từ hình 2, ta thấy, 2.2.2.2 không phải là địa chỉ IP cao nhất nhưng vì tiến trình ưu tiên cổng loopback nên các địa chỉ trên các cổng loopback sẽ được xem xét trước. Điều này được giải thích là sẽ đem lại sự ổn định cho tiến trình OSPF vì interface loopback là loại interface luận lý không bao giờ down trừ khi người quản trị shutdown interface này.

Thực chất, việc up/down của các interface không ảnh hưởng nhiều lắm đến router – id của các router chạy OSPF.

Thật vậy, giả sử trong ví dụ trên, router R đã chọn xong router – id là 192.168.2.1 là IP của cổng F0/1 (xét trường hợp chưa có các interface loopback) và tham gia vào OSPF với router – id 192.168.2.1. Lúc này, nếu ta có bổ sung thêm các interface loopback trên router thì router cũng sẽ không đổi lại router – id thành IP của các interface loopback. Hơn nữa, cho dù lúc này cổng F0/1 có down, thì router vẫn giữ giá trị router – id mà nó đã chọn. Có nghĩa là, router – id đơn thuần chỉ là một cái tên. Khi tên đã được chọn thì tiến trình OSPF sẽ làm việc với cái tên này và không thay đổi lại nữa. Cổng có IP được trích xuất làm tên của router lúc này có up/down cũng không ảnh hưởng gì cả. Vậy nếu chúng ta muốn đổi lại router – id của tiến trình thì sao? Ta phải thực hiện khởi động lại router hoặc gỡ bỏ tiến trình OSPF rồi cấu hình lại, khi đó tiến trình bầu chọn router – id sẽ được thực hiện lại với các interface đang hiện hữu trên router.

Và như vậy, ta thấy việc ưu tiên sử dụng IP trên loopback mang nhiều ý nghĩa về mặt quản trị hơn là tính ổn định của tiến trình. Nó cho phép người quản trị kiểm soát hiệu quả hơn các router – id của các router.

Có một cách khác để thiết lập lại giá trị router – id cho router mà không cần phải khởi động lại router hoặc cấu hình lại OSPF là sử dụng câu lệnh 'router-id' để thiết lập bằng tay giá trị này trên router:

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

```
Router(config-router)#router-id A.B.C.D
```

Lúc này giá trị của router – id có thể không cần phải là một địa chỉ IP có sẵn trên router. Bên cạnh đó, nếu tiến trình OSPF đã chạy và router – id đã được thiết lập trước đó, ta phải khởi động lại tiến trình OSPF thì mới áp dụng được giá trị router – id mới được chỉ ra trong câu lệnh 'router-id'. Câu lệnh khởi động lại tiến trình OSPF:

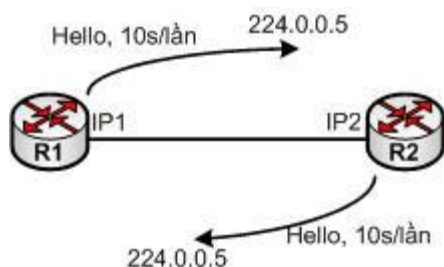
```
Router#clear ip ospf process
```

```
Reset ALL OSPF processes? [no]: yes <- Ta chọn 'Yes'
```

Sau khi đã chọn xong router – id để hoạt động, router chạy OSPF sẽ chuyển qua bước tiếp theo là thiết lập quan hệ láng giềng với các router kết nối trực tiếp với nó.

### Thiết lập quan hệ láng giềng

Bước tiếp theo, sau khi đã chọn xong router – id, router chạy OSPF sẽ gửi ra tất cả các cổng chạy OSPF một loại gói tin được gọi là gói tin hello. Gói tin này được gửi đến địa chỉ multicast dành riêng cho OSPF là 224.0.0.5, đến tất cả các router chạy OSPF khác trên cùng phân đoạn mạng. Mục đích của gói tin hello là giúp cho router tìm kiếm láng giềng, thiết lập và duy trì mối quan hệ này. Gói tin hello được gửi theo định kỳ mặc định 10s/lần.



Hình 3 – Các router gửi gói tin hello.

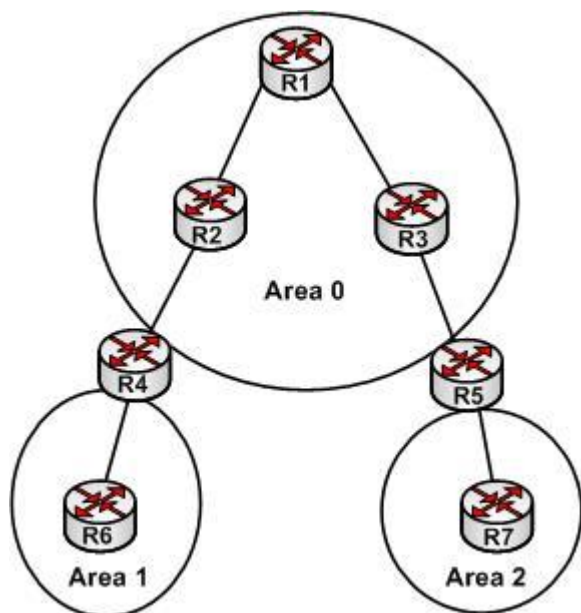
Có nhiều thông tin được hai router kết nối trực tiếp trao đổi với nhau qua gói tin hello. Trong các loại thông tin được trao đổi, có năm loại thông tin sau bắt buộc phải match với nhau trên hai router để chúng có thể thiết lập được quan hệ láng giềng với nhau:

- 1) Area – id.
- 2) Hello timer và Dead timer.
- 3) Hai địa chỉ IP đầu nối phải cùng subnet (một vài trường hợp còn yêu cầu cùng cả subnet – mask).
- 4) Thỏa mãn các điều kiện xác thực.
- 5) Cùng bật hoặc cùng tắt cờ stub.

Ta cùng phân tích từng thông số đã nêu ở trên.

### Area – id

Nguyên tắc hoạt động của OSPF là mỗi router phải ghi nhớ bảng cơ sở dữ liệu trạng thái đường link của toàn bộ hệ thống mạng chạy OSPF rồi từ đó thực hiện tính toán định tuyến dựa trên bảng cơ sở dữ liệu này. Để giảm tải bộ nhớ cũng như tài tính toán cho mỗi router và giảm thiểu lượng thông tin định tuyến cần trao đổi, các router chạy OSPF được chia thành nhiều vùng (area), mỗi router lúc này chỉ cần phải ghi nhớ thông tin cho một vùng mà nó ở trong đó (hình 4).



Hình 4 – Kiến trúc phân vùng trong OSPF.

Cách tổ chức như vậy rõ ràng tiết kiệm tài nguyên mạng và tài nguyên trên mỗi router. Ngoài ra, cách tổ chức này còn cô lập được những bất ổn vào trong một vùng: khi có một link nào đó trên một router up/down, sự kiện này chỉ lan truyền trong nội bộ một vùng và gây ra sự tính toán lại định tuyến của các router trong vùng ấy chứ không ảnh hưởng đến các router thuộc vùng khác.

Mỗi vùng được chỉ ra sẽ có một giá trị định danh cho vùng gọi là Area – id. Area – id có thể được hiển thị dưới dạng một số tự nhiên hoặc dưới dạng của một địa chỉ IP. Ví dụ Area 0 có thể được biểu diễn là Area 0.0.0.0. Một nguyên tắc bắt buộc trong phân vùng OSPF là nếu chia thành nhiều vùng thì bắt buộc phải tồn tại một vùng mang số hiệu 0 – Area 0, Area 0 còn được gọi là Backbone Area và mọi vùng khác bắt buộc phải có kết nối nối về vùng 0.

Khi thực hiện cấu hình phân vùng cho router, ta không gán cả router vào một vùng mà thực hiện gán link trên router vào một vùng. Area – id được gán cho link của router chứ không phải gán cho bản thân router. Ví dụ: trên hình 4, ta thấy toàn bộ router R2 nằm trong vùng 0 là vì khi cấu hình ta đã gán hai link trên R2 vào vùng 0. Những router mà có tất cả các link đều được gán vào một vùng thì sẽ lọt hẳn vào vùng đó và được gọi là các Internal router, các Internal router chỉ phải ghi nhớ trạng thái đường link của vùng mà nó nằm bên trong. Ta cũng xét tiếp router R4. Router này có một link thuộc vùng 0, lại có một link thuộc vùng 1, như vậy nó thuộc về cả hai vùng và phải ghi nhớ trạng thái đường link của cả hai vùng. Những router như vậy được gọi là các router ABR – Area Border Router – router biên giới giữa hai vùng.

Khi hai router láng giềng kết nối với nhau qua một link, chúng phải thống nhất với nhau về area – id của link này. Cả hai router phải gán cùng một số area – id cho link kết nối giữa chúng với nhau. Nếu điều này bị vi phạm, chúng sẽ không thể thiết lập được quan hệ láng giềng thông qua link này và do đó không bao giờ có thể trao đổi được thông tin định tuyến qua link. Đó là điều kiện thứ nhất trong việc thiết lập quan hệ láng giềng: thống nhất về area – id trên link kết nối.

Chương trình CCNA không đề cập đến OSPF đa vùng mà chỉ nói về OSPF đơn vùng, trong đó mọi router đều được gán vào một vùng. Kiến trúc đa vùng và các vấn đề chi tiết của nó sẽ được đề cập chi tiết trong course Route của chương trình CCNP.

### Hello timer và Dead timer

Hello timer là khoảng thời gian định kỳ gửi gói tin hello ra khỏi một cổng chạy OSPF. Khi một router nhận được hello từ láng giềng, nó sẽ khởi động Dead timer. Nếu sau khoảng thời gian được chỉ ra trong Dead timer mà router không nhận được gói tin hello từ láng giềng, nó sẽ coi như láng giềng này không còn và sẽ xóa mọi thông tin mà nó học được từ láng giềng. Ngược lại, cứ mỗi lần nhận được gói tin hello từ láng giềng, Dead timer lại được reset. Giá trị mặc định của hello – timer và dead – timer là 10s và 40s. Ta có thể hiệu chỉnh các giá trị này trên cổng chạy OSPF bằng cách sử dụng câu lệnh:

```
R(config-if)#ip ospf {hello-interval | dead-interval} seconds
```

Để hai router thiết lập được quan hệ láng giềng với nhau, cặp giá trị này bắt buộc phải khớp nhau trên hai router ở hai đầu của đường link.

### Cùng subnet

Hai địa chỉ IP1 và IP2 đấu nối nhau giữa hai router bắt buộc phải cùng subnet thì hai router này mới có thể thiết lập quan hệ láng giềng với nhau (xem hình 3). Một số trường hợp còn bắt buộc hai địa chỉ này phải cùng cả subnet – mask để có thể thiết lập neighbor.

### Thỏa mãn authentication

Trong trường hợp để tăng cường tính bảo mật của hoạt động trao đổi thông tin định tuyến, chúng ta thực hiện cài đặt các password trên hai router hai đầu đường link. Yêu cầu bắt buộc là hai password này phải khớp nhau ở hai đầu để hai router có thể thiết lập neighbor (tất nhiên!). Cấu hình xác thực sai có thể dẫn đến không thiết lập neighbor được giữa hai router từ đó dẫn đến không trao đổi được thông tin định tuyến.

### Cờ stub

Trong kiến trúc đa vùng của OSPF có một loại vùng gọi là vùng stub. Vùng stub là vùng không tiếp nhận LSA type – 5. Khi ta đã cho một link của một router thuộc vùng stub thì bắt buộc đầu kia của link cũng phải gán link này thuộc vùng stub. Khi đó các gói tin định tuyến trao đổi nhau giữa hai đầu sẽ có cờ stub được bật lên. Chi tiết về vùng stub được đề cập cụ thể trong course Route của chương trình CCNP, chương trình CCNA không cover vấn đề này.

Sau khi cả 05 điều kiện nêu trên đã được thỏa mãn, hai router thiết lập với nhau một mối quan hệ gọi là quan hệ láng giềng và được ký hiệu là 2 – WAY. Khi các router đã thiết lập được quan hệ 2 – WAY với nhau, chúng bắt đầu thực hiện trao đổi bảng cơ sở dữ liệu trạng thái đường link (LSDB – Link State Database) cho nhau. Việc trao đổi này được lan ra toàn mạng và cuối cùng mỗi router đều có được trạng thái đường link của mọi router trong mạng, từ đó chúng thực hiện tính toán trên cơ sở dữ liệu trạng thái đường link này và xây dựng bảng định tuyến.

### Trao đổi LSDB

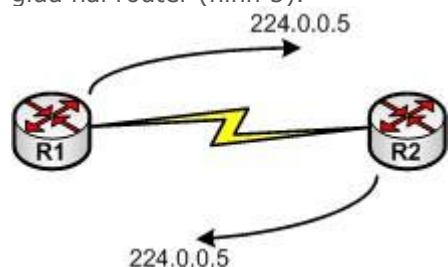
LSDB – Link State Database – Bảng cơ sở dữ liệu trạng thái đường link là một bảng trên router ghi nhớ mọi trạng thái đường link của mọi router trong vùng. Ta có thể coi LSDB là một “tấm bản đồ mạng” mà router sẽ căn cứ vào đó để tính toán định tuyến. LSDB phải hoàn toàn giống nhau giữa các router

cùng vùng. Các router sẽ không trao đổi với nhau cả một bảng LSDB mà sẽ trao đổi với nhau từng đơn vị thông tin gọi là LSA – Link State Advertisement. Các đơn vị thông tin này lại được chứa trong các gói tin cụ thể gọi là LSU – Link State Update mà các router thực sự trao đổi với nhau. Lưu ý: LSA không phải là một loại gói tin mà chỉ là một bản tin. LSU mới thực sự là gói tin và nó chứa đựng các bản tin này.

Việc trao đổi thông tin diễn ra rất khác nhau tùy theo từng loại network – type gán cho link giữa hai router. Trong khuôn khổ chương trình CCNA, chúng ta chỉ xét đến hai loại network – type là Point – to – Point và Broadcast Multiaccess.

### Point – to – point

Loại link point – to – point điển hình là kết nối serial điểm – điểm chạy giao thức HDLC hoặc PPP nối giữa hai router (hình 5).

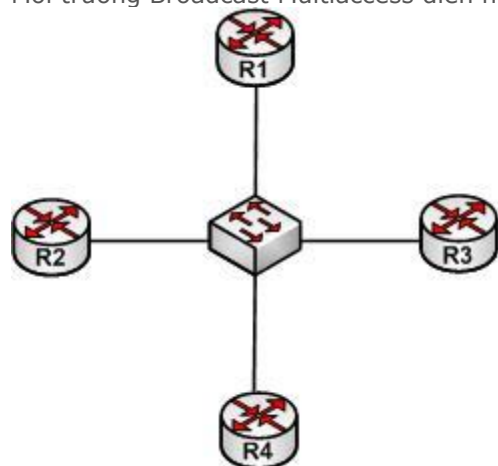


Hình 5 – Trao đổi LSDB với kết nối point – to – point.

Trong trường hợp này, hai router láng giềng sẽ ngay lập tức gửi toàn bộ bảng LSDB cho nhau qua kết nối point – to – point và chuyển trạng thái quan hệ từ 2 – WAY sang một mức độ mới gọi là quan hệ dạng FULL. Quan hệ Full qua một kết nối serial point – to – point được ký hiệu là FULL/ – .

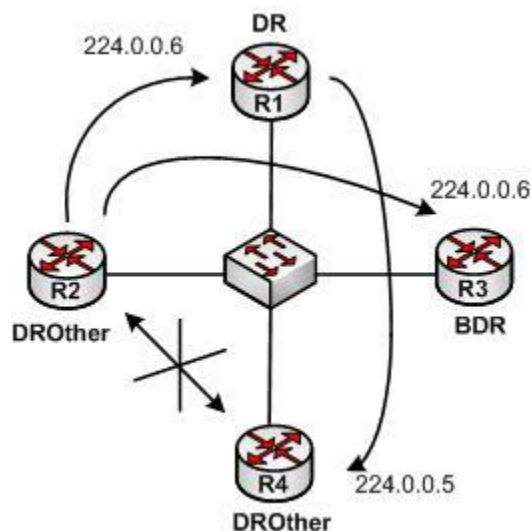
### Broadcast Multiaccess

Môi trường Broadcast Multiaccess điển hình chính là môi trường Ethernet LAN (hình 6).



Hình 6 – Broadcast MultiAccess.

Việc trao đổi LSDB diễn ra hoàn toàn khác trong môi trường này. Với môi trường này, mỗi router đều kết nối trực tiếp với nhau và đều thiết lập quan hệ 2 – WAY với nhau. Tuy nhiên, các router sẽ không trao đổi trực tiếp với nhau mà sẽ tiến hành trao đổi thông tin thông qua một router đầu mối gọi là DR – Designated Router. Trên mỗi kết nối Multi – access, một DR router được bầu ra. Một router khác sẽ được bầu làm Backup DR (BDR) để dự phòng cho DR trong trường hợp DR down. Các router còn lại đóng vai trò là DROther. Nguyên tắc đặt ra như sau: các router DROther khi trao đổi thông tin định tuyến sẽ không gửi trực tiếp cho nhau mà sẽ gửi lên cho DR và BDR. Sau đó router DR này sẽ forward lại thông tin xuống cho các router DROther khác. Khi các router gửi thông tin lên cho DR và BDR, chúng sẽ sử dụng địa chỉ multicast 224.0.0.6 còn khi DR forward lại thông tin xuống các router khác, nó sử dụng địa chỉ 224.0.0.5. Nhắc lại, các DROther không trao đổi trực tiếp với nhau.



Hình 7 – Hoạt động trao đổi thông tin thông qua DR.  
Về quan hệ giữa các cặp router lúc này, ta thấy như sau:

- Các DROther không bao giờ trao đổi thông tin với nhau nên quan hệ giữa chúng mãi mãi chỉ dừng lại ở mức độ 2 – WAY. Thực hiện show bảng neighbor trên các router DROther sẽ thấy rằng các router này hiển thị tình trạng quan hệ với nhau là 2-WAY/DROther.
- Các DROther có trao đổi dữ liệu với DR và BDR nên trong bảng neighbor của các router DROther, các router DR và BDR sẽ hiện ra với quan hệ dạng full: FULL/DR và FULL/BDR. Ngược lại, các router DR và BDR cũng thấy tình trạng quan hệ của các router DROther với chúng là FULL/DROther.

Như vậy, router DR đóng một vai trò rất quan trọng trên môi trường Multiaccess: đó là router điều phối thông tin trên môi trường này. Vậy router nào sẽ được chọn làm DR? Ta có nguyên tắc bầu chọn DR và BDR cho một môi trường multi – access như sau:

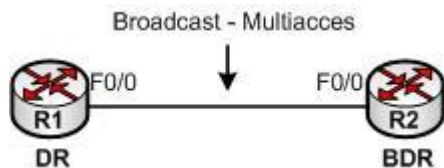
- Trên mỗi cổng đầu nối multi – access của mỗi router đều có một giá trị gọi là priority. Giá trị priority này nằm trong dải từ 0 đến 255 và được trao đổi giữa các router trong các gói tin hello. Router nào nắm giữ giá trị priority cao nhất sẽ được bầu chọn làm DR, priority cao nhì làm BDR, còn lại sẽ là DROther. Giá trị priority mặc định trên các cổng router là bằng 1. Lưu ý rằng nếu router mang giá trị priority bằng 0 sẽ không tham gia vào tiến trình bầu chọn DR và BDR, nó luôn luôn đảm nhận vai trò là DROther.

- Trong trường hợp giá trị priority bằng nhau (ví dụ để mặc định bằng 1 hết, không cấu hình gì thêm), router nào có Router – id cao nhất sẽ làm DR, Router – id cao nhì sẽ làm BDR, còn lại làm DROther. Ta nói Router – id là tie – breaker của Priority.

Có một số lưu ý cho việc bầu chọn DR và BDR như sau:

- Nếu ta cấu hình một router nhận giá trị priority bằng 0, router này sẽ không tham gia vào tiến trình bầu chọn DR và BDR, nó luôn luôn là DROther. Chúng ta phải lưu ý điều này vì nếu chúng ta cấu hình cho tất cả các router đầu nối vào môi trường multi – access giá trị priority = 0 thì sẽ không có router nào chịu làm DR cho môi trường này! Lỗi này dẫn đến lỗi hỏng trong việc trao đổi thông tin định tuyến.
- Luật bầu chọn DR là non – preempt: khi một DR đã được bầu chọn xong, nếu router mới tham gia vào môi trường multi – access có priority hay router – id cao hơn router DR nó cũng không thể chiếm quyền của DR hiện tại. Chỉ khi nào DR hiện tại down, router khác mới có cơ hội tranh quyền DR.
- Một router có thể đóng nhiều vai trò khác nhau trên nhiều cổng multi – acces khác nhau. Ví dụ: nó có thể là DR trên môi trường Multi – access đầu nối vào cổng F0/0 nhưng lại là DROther trên môi trường Multi – acces đầu nối vào cổng F0/1.
- Chúng ta không được nhầm lẫn kết nối Ethernet nối 02 router là một kết nối point – to – point, kết nối này vẫn được xem là Multi – access. Trong trường hợp này, một router sẽ làm DR, một làm BDR, không có DROther (hình 8).





Hình 8 – Đây là môi trường Multi – access dù chỉ có 02 router.

Sau khi hoàn thành xong thao tác trao đổi LSDB, mỗi router trong vùng đều đã có được bảng cơ sở dữ liệu trạng thái đường link của mọi router trong vùng, hay nói một cách khác, mỗi router đã có được “tấm bản đồ mạng” của cả vùng. Dựa trên LSDB này, các router sẽ chạy giải thuật Dijkstra để xây dựng một cây đường đi ngắn nhất đến mọi đích đến trong mạng với gốc cây chính là router ấy. Từ cây này, router xây dựng lên bảng định tuyến của mình. Chi tiết về giải thuật Dijkstra xin không đề cập ở đây. Các bạn quan tâm có thể tìm hiểu thông qua các giáo trình về Toán rời rạc hoặc Lý thuyết đồ thị của các trường Đại học. Bài viết này sẽ giới thiệu cách OSPF tính toán metric cho các đường đi và cách người quản trị nhìn vào sơ đồ mạng để xác định đường đi mà OSPF đã chọn mà không phải “chạy” thuật toán Dijkstra trong đầu ^^:

### Tính toán metric với OSPF

Metric trong OSPF được gọi là cost, được xác định dựa vào bandwidth danh định của đường truyền theo công thức như sau:

$\text{Metric} = \text{cost} = 10^8 / \text{Bandwidth}$  (đơn vị bps).

Ta phân biệt giữa bandwidth danh định trên cổng và tốc độ thật của cổng ấy. Hai giá trị này không nhất thiết phải trùng nhau và giá trị danh định mới chính là giá trị được tham gia vào tính toán định tuyến. Giá trị danh định được thiết lập trên cổng bằng câu lệnh:

`R(config-if)#bandwidth BW` (đơn vị là kbps)

Ta phải chỉnh giá trị danh định này trùng với tốc độ thật của cổng để tránh việc tính toán sai lầm trong định tuyến. Ví dụ: một đường leased – line kết nối vào cổng serial chỉ có tốc độ thật là 512kbps nhưng giá trị bandwidth danh định trên cổng serial luôn là 1.544Mbps ở mặc định. Điều đó dẫn đến OSPF xem một cổng 512 kbps như một cổng 1.544 Mbps! Ta phải chỉnh lại bằng thông danh định trên cổng trong trường hợp này để phản ánh đúng tốc độ thật:

`R(config-if)#bandwidth 512`

Dựa vào công thức metric đã nêu ở trên, ta có giá trị cost default của một số loại cổng:

Ethernet (BW = 10Mbps) -> cost = 10.

Fast Ethernet (BW = 100Mbps) -> cost = 1.

Serial (BW = 1.544Mbps) -> cost = 64 (chặt bỏ phần thập phân trong phép chia).

Ta cùng xem xét một ví dụ để khảo sát cách tính toán path – cost cho một đường đi:



Hình 9 – Sơ đồ ví dụ tính path – cost.

Yêu cầu đặt ra với sơ đồ hình 9 là tính path – cost (metric) cho đường đi từ R1 đến mạng [192.168.3.0/24](#) của R3.

Ta thấy một cách dễ dàng: từ R1 đi đến mạng [192.168.3.0/24](#) của R3 sẽ đi qua các đường link Fast Ethernet có cost = 1, serial có cost là 64 và link Fast Ethernet có cost bằng 1. Vậy tổng cost tích lũy sẽ là 1 + 64 + 1 là 66. Metric từ R1 đến mạng [192.168.3.0/24](#) là 66.

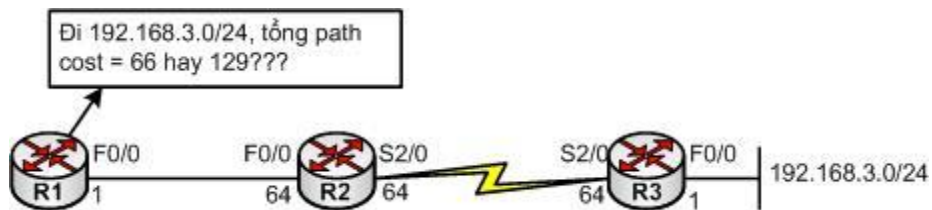
Tuy nhiên việc tính toán sẽ trở nên phức tạp hơn nếu hai cổng router ở hai đầu link không đồng nhất về giá trị cost. Ví dụ, ta vào cổng F0/0 của R2 đổi lại giá trị cost thành 64 bằng cách đánh lệnh sau đây trên cổng F0/0 của R2:

`R2(config)#interface f0/0`

`R2(config-if)#ip ospf cost 64`

Vậy câu hỏi đặt ra là với link Fast Ethernet nối giữa R1 và R2 ta chọn cost của link này là 1 hay 64?

Nếu chọn là 1, tổng cost toàn tuyến vẫn giữ giá trị như cũ là 66, nhưng nếu chọn là 64, tổng cost toàn tuyến sẽ là 64 + 64 + 1 là 129, hai giá trị rất khác nhau!

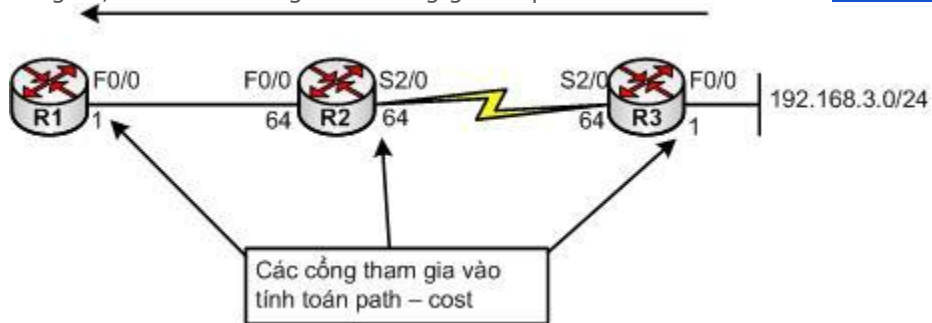


Hình 10 – Tổng path – cost là 66 hay 129?

Để trả lời được câu hỏi này, ta cần nắm nguyên tắc sau đây trong việc tính tổng cost với OSPF:

Để tính tổng cost từ một router đến một mạng đích theo một đường (path) nào đó, ta thực hiện lần ngược từ đích lần về và cộng dồn cost theo quy tắc đi vào thì cộng, đi ra thì không cộng.

Áp dụng quy tắc này cho ví dụ ở hình 10: để tính tổng cost từ R1 đến mạng [192.168.3.0/24](#), ta đi ngược từ mạng [192.168.3.0/24](#) đi về. Khi đi về ta đi vào cổng F0/0 của R3, cộng giá trị cổng này (tổng cost lúc này là 1); đi ra khỏi cổng S2/0 của R3, bỏ qua không cộng (tổng cost vẫn là 1); đi tiếp vào cổng S2/0 của R2, cộng giá trị cổng này (lúc này tổng cost là  $1 + 64 = 65$ ); đi ra khỏi cổng F0/0 của R2, bỏ qua không cộng (tổng cost vẫn là 65); đi tiếp vào cổng F0/0 của R1, cộng giá trị cổng này (tổng cost là  $65 + 1 = 66$ ), kết thúc hành trình. Vậy tổng cost vẫn là 66, việc thay đổi giá trị cost trên cổng F0/0 của R2 không ảnh hưởng gì đến path – cost từ R1 đi đến [192.168.3.0/24](#).



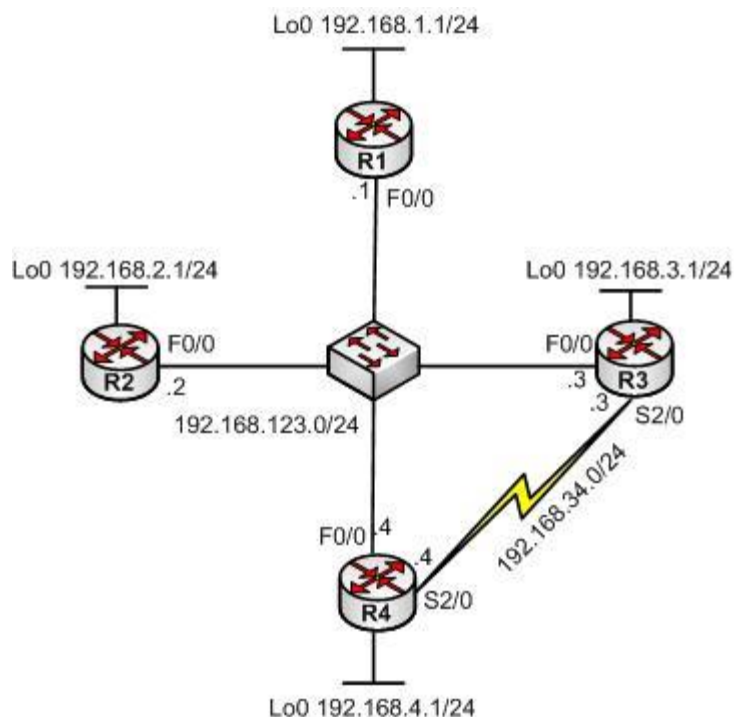
Hình 11 – Các cổng tham gia vào tiến trình tính toán path – cost với OSPF.

Như vậy với OSPF, để đánh giá đúng được cost của đường đi và có thể hiệu chỉnh cost trên cổng để bẻ đường đi của gói tin theo ý muốn, ta cần phải cẩn thận trong việc xác định xem cổng nào trên đường đi sẽ tham gia vào tính toán để hiệu chỉnh đúng cổng vì hiệu chỉnh không đúng cổng sẽ không mang lại bất kỳ thay đổi gì.

Trên đây, chúng ta đã cùng nhau review một số nét chính trong hoạt động của OSPF đơn vùng thông qua các bước hoạt động của tiến trình OSPF: chọn Router – id, thiết lập quan hệ láng giềng, trao đổi LSDB và tính toán xây dựng bảng định tuyến.

Tiếp theo chúng ta cùng review lại cách thức cấu hình OSPF thông qua ví dụ được trình bày dưới đây. Cấu hình





Hình 12 – Sơ đồ ví dụ cấu hình.

Trên hình 12 là 4 router đại diện cho bốn chi nhánh khác nhau của một doanh nghiệp: R1 cho chi nhánh 1, R2 cho chi nhánh 2, R3 cho chi nhánh 3 và R4 cho chi nhánh 4. Các interface loopback trên các router đại diện cho các mạng nội bộ của mỗi chi nhánh, sử dụng các subnet trên R1, R2, R3 và R4 lần lượt là [192.168.1.0/24](#), [192.168.2.0/24](#), [192.168.3.0/24](#) và [192.168.4.0/24](#). Bốn router này đều nối với nhau thông qua một kết nối multi – access (đại diện bằng một Ethernet Switch), sử dụng subnet là [192.168.123.0/24](#) (trên thực tế, đây có thể là một môi trường Metro net đầu nối giữa các chi nhánh). Router R3 và R4 còn thực hiện đầu nối riêng với nhau bằng một đường leased – line thông qua các cổng serial, sử dụng subnet [192.168.34.0/24](#).

Yêu cầu đặt ra là chạy định tuyến OSPF đảm bảo mọi địa chỉ trên sơ đồ này thấy nhau.

Để thực hiện chạy OSPF trên các router, chúng ta sử dụng câu lệnh sau:

```
R(config)#router ospf process-id
```

```
R(config-router)#network địa chỉ IP wildcard-mask area area-id
```

Trong đó:

Process – id: số hiệu của tiến trình OSPF chạy trên router, chỉ có ý nghĩa local trên router

Để cho một cổng tham gia OSPF, ta thực hiện “network” địa chỉ mạng của cổng đó. Với OSPF ta phải sử dụng thêm wildcard – mask để lấy chính xác subnet tham gia định tuyến. Ta cũng phải chỉ ra link thuộc area nào bằng tham số “area”.

Cấu hình trên các router:

Trên router R1:

```
R1(config)#router ospf 1
```

```
R1(config-router)#network 192.168.123.0 0.0.0.255 area 0
```

```
R1(config-router)#network 192.168.1.0 0.0.0.255 area 0
```

Trên router R2:

```
R2(config)#router ospf 1
```

```
R2(config-router)#network 192.168.123.0 0.0.0.255 area 0
```

```
R2(config-router)#network 192.168.2.0 0.0.0.255 area 0
```

Trên router R3:

```
R3(config)#router ospf 1
```

```
R3(config-router)#network 192.168.123.0 0.0.0.255 area 0
```

```
R3(config-router)#network 192.168.34.0 0.0.0.255 area 0
```

```
R3(config-router)#network 192.168.3.0 0.0.0.255 area 0
```

Trên router R4:

```
R4(config)#router ospf 1
```

```
R4(config-router)#network 192.168.123.0 0.0.0.255 area 0
```

```
R4(config-router)#network 192.168.34.0 0.0.0.255 area 0
```

```
R4(config-router)#network 192.168.4.0 0.0.0.255 area 0
```

Ta thấy khi muốn cho cổng F0/0 trên router tham gia OSPF, ta "network"

mạng [192.168.123.0/24](#) trên cổng này. Giá trị wildcard – mask được tính cho /24 sẽ là 0.0.0.255 (để tính được giá trị wildcard mask, ta lấy giá trị 255.255.255.255 trừ đi giá trị subnet – mask 255.255.255.0 từng octet một sẽ được kết quả cần tìm. Cách tính này chỉ đúng cho một dải IP liên tiếp, không phải đúng cho mọi trường hợp. Về wildcard – mask sẽ có một bài viết khác để cập chi tiết cho vấn đề này). Tương tự với các cổng khác của các router.

Kiểm tra bằng cách hiển thị bảng định tuyến của các router:

Trên R1:

```
R1#sh ip route ospf
```

```
192.168.4.0/32 is subnetted, 1 subnets
```

```
O 192.168.4.1 [110/2] via 192.168.123.4, 00:00:29, FastEthernet0/0
```

```
O 192.168.34.0/24 [110/65] via 192.168.123.4, 00:00:29, FastEthernet0/0  
[110/65] via 192.168.123.3, 00:00:29, FastEthernet0/0
```

```
192.168.2.0/32 is subnetted, 1 subnets
```

```
O 192.168.2.1 [110/2] via 192.168.123.2, 00:00:29, FastEthernet0/0
```

```
192.168.3.0/32 is subnetted, 1 subnets
```

```
O 192.168.3.1 [110/2] via 192.168.123.3, 00:00:29, FastEthernet0/0
```

Trên R2:

```
R2#show ip route ospf
```

```
192.168.4.0/32 is subnetted, 1 subnets
```

```
O 192.168.4.1 [110/2] via 192.168.123.4, 00:01:20, FastEthernet0/0
```

```
O 192.168.34.0/24 [110/65] via 192.168.123.4, 00:01:20, FastEthernet0/0  
[110/65] via 192.168.123.3, 00:01:20, FastEthernet0/0
```

```
192.168.1.0/32 is subnetted, 1 subnets
```

```
O 192.168.1.1 [110/2] via 192.168.123.1, 00:01:20, FastEthernet0/0
```

```
192.168.3.0/32 is subnetted, 1 subnets
```

```
O 192.168.3.1 [110/2] via 192.168.123.3, 00:01:20, FastEthernet0/0
```

Trên R3:

```
R3#show ip route ospf
```

```
192.168.4.0/32 is subnetted, 1 subnets
```

```
O 192.168.4.1 [110/2] via 192.168.123.4, 00:02:07, FastEthernet0/0
```

```
192.168.1.0/32 is subnetted, 1 subnets
```

```
O 192.168.1.1 [110/2] via 192.168.123.1, 00:02:07, FastEthernet0/0
```

```
192.168.2.0/32 is subnetted, 1 subnets
```

```
O 192.168.2.1 [110/2] via 192.168.123.2, 00:02:07, FastEthernet0/0
```

Trên R4:

```
R4#show ip route ospf
```

```
192.168.1.0/32 is subnetted, 1 subnets
```

```
O 192.168.1.1 [110/2] via 192.168.123.1, 00:21:57, FastEthernet0/0
```

```
192.168.2.0/32 is subnetted, 1 subnets
```

```
O 192.168.2.1 [110/2] via 192.168.123.2, 00:21:57, FastEthernet0/0
```

```
192.168.3.0/32 is subnetted, 1 subnets
```

```
O 192.168.3.1 [110/2] via 192.168.123.3, 00:21:57, FastEthernet0/0
```

Ta thấy rằng các subnet ở xa đã được học thông qua OSPF (các route OSPF được ký hiệu bằng ký tự "O").

Một điểm độc đáo chúng ta để ý là các mạng loopback khi hiển thị trong bảng định tuyến của các router đều được OSPF chuyển thành /32. Đây là một nét trong hoạt động của giao thức OSPF, một loại giao thức cung cấp cho mỗi router một cái nhìn tổng quan toàn mạng chứ không phải chỉ dựa vào "tin đồn được lan truyền" như với Distance – vector. Với cái nhìn như vậy, mỗi router xác định được mạng loopback nằm trên một cổng không đầu nối đi đâu cả và chỉ có một địa chỉ của cả một mạng được sử dụng. /32 được sử dụng để phản ánh điều này. Để khắc phục hiện tượng này và khiến cho các subnet loopback được hiển thị đúng giá trị prefix – length, ta thay đổi kiểu network – type trên interface loopback thành kiểu "point – to – point" bằng câu lệnh:

```
R(config)#interface loopback 0
```

```
R(config-if)#ip ospf network point-to-point
```

Chi tiết về các "network – type" được sử dụng trong OSPF sẽ được đề cập trong các bài viết khác trong khuôn khổ của chương trình CCNP.

Ta cùng khảo sát bảng neighbor trên các router:

Trên R4:

```
R4#show ip ospf neighbor
```

```
Neighbor ID Pri State Dead Time Address Interface
192.168.3.1 0 FULL/ - 00:00:34 192.168.34.3 Serial2/0
192.168.1.1 1 FULL/DROTHER 00:00:32 192.168.123.1 FastEthernet0/0
192.168.2.1 1 FULL/DROTHER 00:00:34 192.168.123.2 FastEthernet0/0
192.168.3.1 1 FULL/BDR 00:00:37 192.168.123.3 FastEthernet0/0
```

Ta thấy, R4 đã thiết lập quan hệ láng giềng với các router còn lại. Qua cổng F0/0, nó thấy 03 router láng giềng có router – id là 192.168.1.1, 192.168.2.1, 192.168.3.1. Tình trạng quan hệ cũng chỉ rõ quan hệ là dạng FULL và vai trò của các router láng giềng trong môi trường multi – access. Từ kết quả show ở trên, ta thấy rất rõ ràng router R1 (192.168.1.1), R2 (192.168.2.1) là các DROTHER router, R3 (192.168.3.1) là BDR router và dĩ nhiên, ta có thể suy ra được R4 chính là DR router. DR thiết lập quan hệ dạng FULL với tất cả các router trong môi trường multi – access.

Ta cũng thấy rằng R4 cũng thiết lập quan hệ láng giềng dạng FULL với R3 qua cổng serial 2/0. Vì đây là cổng point – to – point nên các router không bầu chọn DR và BDR và quan hệ giữa hai router được ký hiệu là FULL/-.

Trên R1:

```
R1#show ip ospf neighbor
```

```
Neighbor ID Pri State Dead Time Address Interface
192.168.2.1 1 2WAY/DROTHER 00:00:30 192.168.123.2 FastEthernet0/0
192.168.3.1 1 FULL/BDR 00:00:32 192.168.123.3 FastEthernet0/0
192.168.4.1 1 FULL/DR 00:00:31 192.168.123.4 FastEthernet0/0
```

Trên kết quả show của R1, ta cũng thấy các láng giềng R2 (192.168.2.1), R3(192.168.3.1) và R4 (192.168.4.1) và vai trò của từng con router trong môi trường multi – access. Ta để ý rằng R1 chỉ duy trì quan hệ ở mức 2WAY với R2, cũng là một DROTHER giống nó. Hai DROTHER không bao giờ trao đổi trực tiếp thông tin định tuyến với nhau nên không bao giờ thiết lập được quan hệ dạng FULL.

Thực hiện kiểm tra tương tự với các router R2 và R3.