**Tìm hiểu giao thức định tuyến OSPF**

Giao thức định tuyến OSPF là giao thức định tuyến động thuộc nhóm Link State. Trên mỗi Router đều có bản đồ mạng của cả vùng (bảng định tuyến) thông qua việc đồng nhất bảng cơ sở dữ liệu trạng thái đường link (LSDP - Link State Database). Từ bản đồ mạng này Router sẽ tự tính toán ra đường đi ngắn nhất và xây dựng bảng định tuyến cho nó.  
Giao thức OSPF được sử dụng rộng rãi trong các công ty cho hệ thống mạng lớn.

**1. overview OSPF Protocol**

**- OSPF**(Open shortest path first) là chuẩn mở do IEEE đưa ra

* Hoạt động ở nhóm classless
* **AD**= 110
* **Metric**phụ thuộc vào Bandwith
* Sử dụng thuật toán Dijkstra để tìm đường đi ngắn nhất.

**2. Router-id trên OSPF**

- Để chạy OSPF nó phải tạo ra 1 định danh để chạy gọi là Router-id( giống như CMND)

* có định dạng A.B.C.D(vd: IPv4:192.168.1.1)
* để tạo ra Router-id có 2 cách
  + Cách 1: Router tự động tạo ra
  + Cách 2: Do mình tạo bằng cách config

**Cách 1: tự động tạo**

* Dựa vào interface nào có địa chỉ *IP cao nhất* thì nó lấy IP đó làm Router-id  
  **Vd**: Router có f0/0=10.0.0.1 f0/1=172.16.1.1 s0/0/0=192.168.1.1  
  => Router-id = 192.168.1.1
* Nếu Router có Loopback tồn tại và cho tham gia định tuyến thì Router-id *ưu tiên cho Loopback*trước  
  **Vd:** lookback 0=4.1.1.1 ;lookback1=4.2.2.2; f0/0=172.16.1.1; f0/1=192.168.1.1  
  => Router-id = 4.2.2.2

**Cách 2 : admin tạo**

* Router-id không nhất thiết là phải chọn IP có trên interface  
  **Vd**: lookback 0=4.1.1.1 ;lookback1=4.2.2.2; f0/0=172.16.1.1; f0/1=192.168.1.1  
  Ta có thể cấu hình để chọn **Router-id = 100.100.100.100**. Ip này không thuộc interface nào của router cả

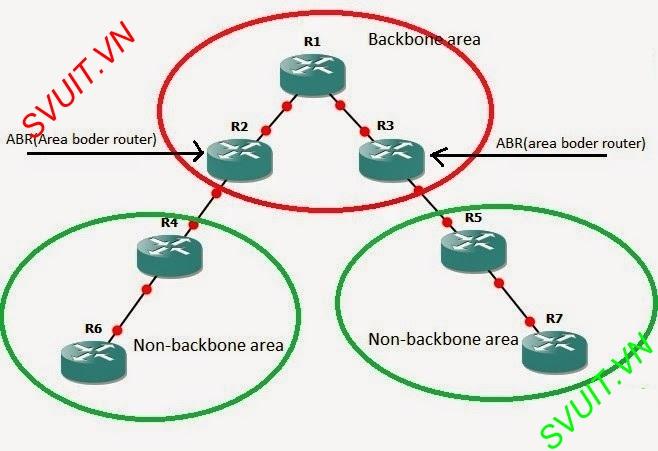
**3. Neighber trong OSPF**

- Neighbor trong giao thức định tuyến OSPF

* Khi cả 2 Router đã chạy OSPF thì chúng bắt đầu gửi gói tin Hello để thiết lập neighbor.
* Đinh kì của gói tin Hello là 10s/1 lần
* OSPF sử dụng địa chỉ multilcast : 224.0.0.5 khác với RIPv2 224.0.0.9
* Gói tin hello của OSPF chỉ dùng để định neighbor còn RIPv2 thì gói tin hello của nó có cả bảng định tuyến bên trong.
* Để làm neighbor của nhau thì gói tin hello của 2 router phải giống nhau 1 số thông số

**Điều kiện 1:** *Area\_id*: 

* **Backbone area**: Phải có ít nhất 1 vùng. Kí hiệu : 0
* **Non-backbone are**: phải kết nối trực tiếp với vùng Backbone area. Kí hiệu: 1-2^52. => sẽ có 1 còn Router đứng giữa 2 vùng
* **Backbone Router**: là Router nằm trong vùng backbone area.
* **Internal Router**: là Router nằm trong vùng non-backbone area.
* **Area Boder Router(ABR)**: Router nằm giữa ranh giới backbone area và non-backbone area .
* **Autonomous System Boder Router(ASBR)**: là Router biên giới giữa định tuyến OSPF và 1 giao thức định tuyên khác( nghĩa là nó vừa chạy OSPF vừa chạy RIPv2 chẳng hạn)

[](https://lh6.googleusercontent.com/HSSEI1i8odo1m_VtxJMte_lqZw_ixK6oRfaKPZ8UTpk83_hO4Srt7dB90S_F83StSQOOzxDFyMbn9sQqaEm-IRmniUUoNnHVeFocq26GidWxKO0T1PGZe-Xr7oldmt_t7L_HgRSC4H9HqLC2Kg)

**Điều kiện 2 :** *2 router phải cùng subnet và subnet-mask*

* **Vd1**:

R1=192.168.1.110/25  
192.168.130/25  
=> Major-network = 192.168.1.0/24   
=> mượn 1 bít --> bước nhảy = 128  
=> Có 2 mạng 192.168.1.0/25 và 192.168.1.128/25  
==> 2 router không thể là neighbor của nhau được *vì 2 IP trên khác mạng*​

* **Vd2**:

192.168.1.110/25 và 192.168.1.11/26  
2 router cùng mạng nhưng không thể làm neighbor của nhau *vì không cùng subnet-mask*​

**Điều kiện 3:** *Có cùng hello-timer/die-timer(10s/40s)*  
  
**Điều kiện 4:** *Cùng loại xác thực.*

* cùng là plain-text
* cùng xác thực MD5

***\*\*\* khi cả 4 điều kiện trên giống nhau thì 2 Router có thể làm neighbor của nhau.***

**4. Quá trình tìm đường đi tối ưu**

* B1: thiết lập được neighbor của nhau. Sau đó liệt kê các neighbor vào trong neighbor của mình. Lúc này, mối quan hệ giưa các neighbor gọi là 2-way
* B2: Bắt đầu gửi thông tin trạng thái đường link để dựng lên 1 bảng database(bảng topology).
* B3: từ bảng topology nó bắt đầu dùng thuật toán Dijkstra để tìm ra đường đi tối ưu để đưa ra bảng định tuyến.
* B4: bảng LSDB chứa các LSA. Để có LSA thì nó phải trải qua các giai đoạn sau :

Sau khi ở trạng thái 2-way thì nó bắt đầu gửi thông tin cho nha để hình thành lên 1 bảng database gọi là LSDB(Link-state database)​

* **Gđ1:**Router sẽ gửi thông tin trạng thái đường link của nó cho các neighbor gọi là LSA(Link state Advertisement).
* **Gđ2**:Trước khi gửi LSA nó sẽ gửi 1 bản tin *DBD(Database Description)* để mô tả những thông tin mà nó có được cho router neighbor.
* **Gđ3:** Khi neighbor nhận được DBD nếu nó thấy thông tin nào trong DBD mà nó không có thì nó sẽ gửi *LSR(link state request)*để xin thông tin thiếu.
* **Gđ4:** Khi router nhận được request LSR thì nó phải cho những thông tin thiếu cho router xin bằng LSA nằm bên trong*LSU(Link state update).* LSU giống như là 1 phương tiện để chở LSA trả về cho.
* **Gđ5:** Khi router xin nhận được LSU thì nó bỏ phần LSU lấy phần LSA. Khi nhận xong nó phải trả lời lại là đã nhận được bằng*LSACK(link state acknowledgment)*

==>*Sau khi có LSDB thì router có thể tự chọn được đường đi tốt nhất dựa vào thuật toán Dijkstra*  
  
**5. Môi trường mạng**

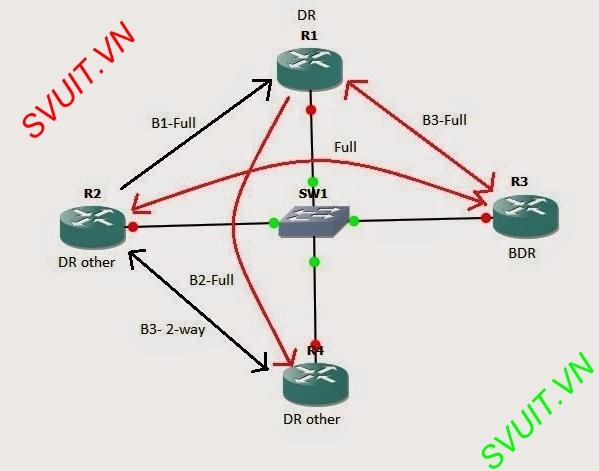
* Tùy thuộc vào mỗi 1 môi trương mạng thì nó có 1 cạc trao đổi khác nhau để nó tìm được đường đi tốt nhất.
* Trong môi trường mạng có 2 môi trường chính:

**a.point to point:**

* là môi trường mà 2 router kết nối với nhau bằng cổng serial(WAN).
* Khi ở môi trường này thì các router gửi SLDB trực tiếp qua nhau thì từ mối quan hệ 2-way chuyển sang mối quan hệ FULL (nếu không quan hệ trực tiếp với nhau thì ko cần chuyển quan hệ 2-way)

**b.Broadcast multi access:**

* các router kết nối với nhau = interface LAN.
* Trước khi trao đổi thông tin thì các router sẽ bầu chọn ra 1 router đóng vai trò làm chủ đạo gọi là **DR(designated router)** có nhiệm vụ tiếp nhận các thông tin trao đổi và gửi qua cho các router khác.
* **BDA(backup designated router)**là router dự phòng cho DR.
* **DA other:** Những router còn lại. Những router không nói chuyện trực tiếp với nhau (vẫn giữ trạng thái 2-way) mà phải thông qua DR. Đồng thời DR gửi thông tin copy cho BDA để backup

[[](https://lh5.googleusercontent.com/VNF3guJPJOFBaqPAJednQ47UaSWmKLKZeZnRjXYOadfCgSdbTrhk7A2ufXy4P9HRrW3YYK8ia_DhBM1khXoE9zcUdx7Zpv4UGsBbOGFXBOfJGvI5GNinmx_otKIt3MmJqSPqtcos5ovZtPXN9Q)](https://lh5.googleusercontent.com/VNF3guJPJOFBaqPAJednQ47UaSWmKLKZeZnRjXYOadfCgSdbTrhk7A2ufXy4P9HRrW3YYK8ia_DhBM1khXoE9zcUdx7Zpv4UGsBbOGFXBOfJGvI5GNinmx_otKIt3MmJqSPqtcos5ovZtPXN9Q)   
​

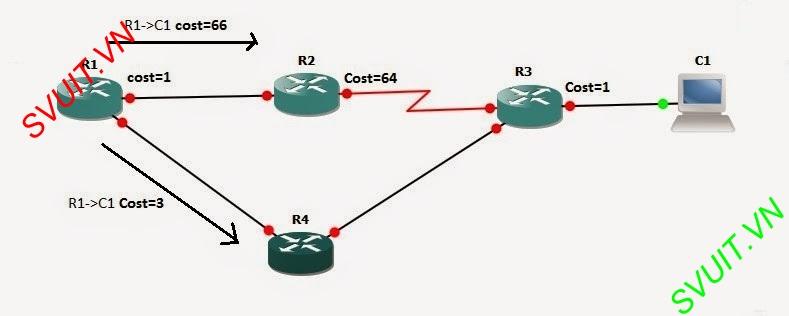
- Việc bầu dựa vào DR,BDA:

* **priority**: chỉ số ưu tiên của cổng. Có giá trị từ 0 – 255(default=1)
  + Router có *priority cao nhất làm DR*, cao thứ 2 làm BDR, còn lại DA other.
  + Khi đặt priority = 0 cho 1 interface router thì router đó không bao giờ được làm DR,BDR.
* **Router-id:** Khi xét quá trình chọn DR thì router-id sẽ không xét loopback
  + Riêng DR và BDR nói chuyện với nhau = 224.0.0.6 còn lại nói chuyện với nhau = 224.0.0.5
  + trong OSPF có 1 quy luật Non-preemptive: nghĩa là không bầu chọn lại.Khi DR chết BDR thay nhưng sau đó DR sống lại thì nó sẽ ko được bầu chọn lại làm DR lại như cũ.

**6. Tính Cost trong OSPF**

- Trong OSPF không còn gọi là Metrict mà gọi là Cost (Cost trên interface)  
*- Cost được tính khi đi vào 1 cổng và đi ra không tính*  
- Công thức tính cost như sau :  
**Cost= 10[SUP]8[/SUP][SUB]/bandwith**  
**[/SUB]**

* Ethernet(10Mbps) --> Cost = 10
* FastEthernet(100Mbps) --> cost=1
* Serial(1,544Mbps) --> cost=64
* **Ví dụ** :

[](https://lh6.googleusercontent.com/6whJ6oa_LYa9gpadnU2Aa3U9pfhJs-cMcW6kqEDIMTUshxfVI7yJdiGfB8_banBHfHWMkkG3o1_Al1g-oCKFcEahTLU7QPVxXq7OXT6Q95gEHNgxoRtAXAWARK0tvNUo2dyES8ZA1RHTqPCPdw)

Để tính cost từ R1--> C1 ta tính ngược lại đi là   
Từ R1 --> đi vào f0/1 R3(+cost=1) --> đi ra s0/0/0 R3(+0) --> đi vào f0/1 R2(+64) --> đi ra f0/0 R2(+0) --> đi vào f0/0 R1(+1).  
​

* Ở đây để Router sẽ chọn đường có cost = 3 . Để có thể loadbalancing trong sơ đồ này ta cần thay đổi
  + **Bandwith**: việc thay đổi bandwith nào chỉ nhằm mục đích thay đổi giá trị cost nó không ảnh hưởng gì đến traffic của interface
  + **Cost**: ở đường dưới ta thấy tại f0/0 của R3 cost=1 ta không thể hạ cost được nữa vì cost = 1 là giá trị minimum. Vì vầy trong sơ đồ trên ta nên tăng cost ở f0/1 lên sao cho tổng cost của 2 đường đều = 66