

# REPORT

---

## Computer Network Assignment #2

Understanding OSPF Routing protocol



학과	컴퓨터학과
학번	2015410056
이름	김지윤
제출 일자	2018/06/16
담당 교수	민성기 교수님

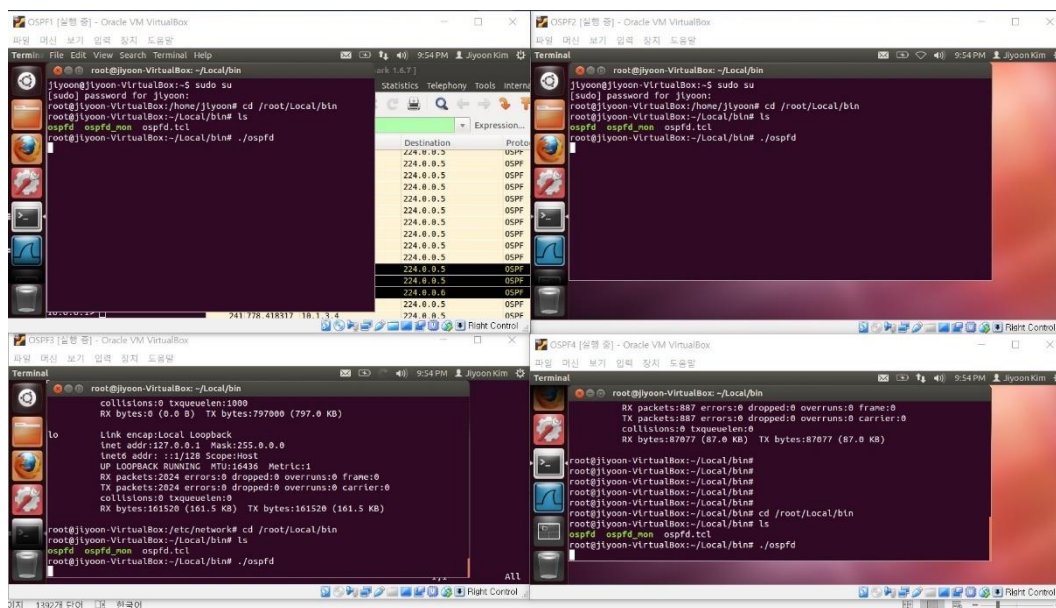
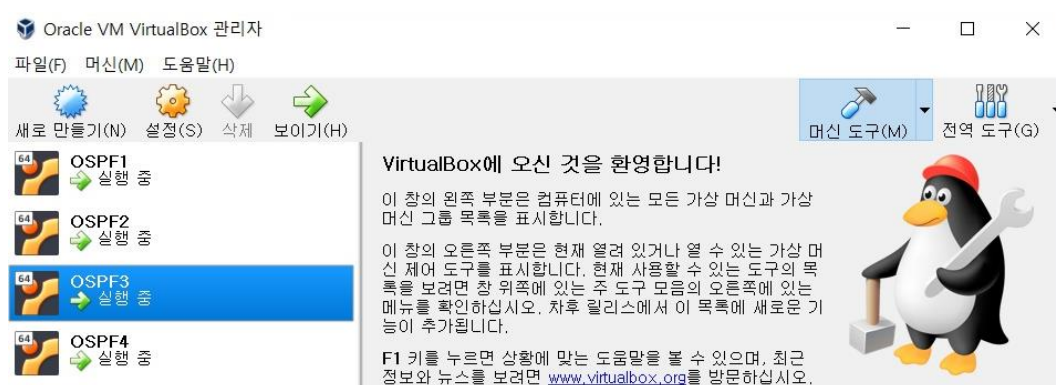
---

## Assignment #2: OSPF Packet Analysis

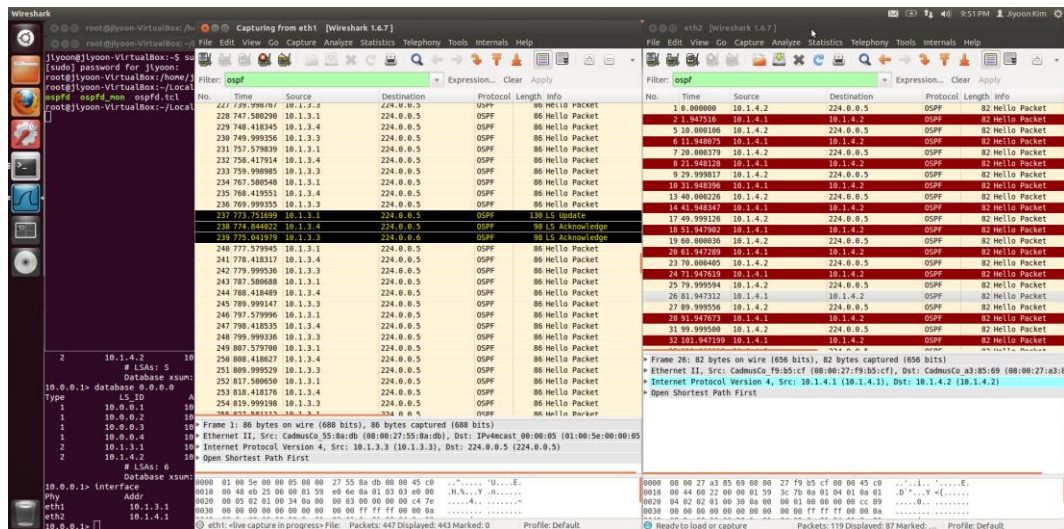
Understand how OSPF routing protocol works and analysis packets with Wireshark.

### ■ Virtual Machine 설정 및 실행 화면

- ✓ Virtual Machines: OSPF1, OSPF2, OSPF3, OSPF4
- ✓ Network Topology의 VM1, VM2, VM3, VM4에 맞춰 각 virtual machine의 네트워크를 설정
- ✓ ospfd는 4개의 virtual machine에서 모두 실행
- ✓ ospfd\_mon과 두 개의 Wireshark는 OSPF1(VM1)에서 실행



왼쪽 위부터 차례대로 OSPF1, OSPF2, OSPF3, OSPF4의 virtual machine이고 ospfd를 동시에 실행시키고 있는 모습이다.



Virtual machine OSPF1의 실행화면이다. 설정한 두개의 내부 네트워크 eth1, eth2의 packet을 각각 캡처 하고 있는 Wireshark를 실행하고 있고 왼쪽의 터미널에서는 ospfd와 ospfd\_mon를 터미널 당 하나씩 동시에 실행시키고 있다.

## ■ OSPF (Open Shortest Path First)

OSPF는 인터넷 프로토콜 네트워크를 위한 Link State 라우팅 프로토콜이다. Network topology의 변화에 대해 빠르게 대응하고 최소한의 라우팅 프로토콜 traffic을 이용한다.

네트워크상의 라우터들은 neighbor라우터를 인식하고 통신하여 네트워크 내의 다른 라우터들의 정보를 직접 수집한다. 각 라우터들은 수집된 정보를 통해 Link State Database를 구축하는데 주어진 영역 내의 모든 라우터는 통신을 마친 후엔 모두 동일한 database를 가져야한다.

### 1. Terminology

OSPF 라우팅 프로토콜 관련 용어는 아래와 같은 것들이 있다.

- ✓ Neighbor: 공통의 네트워크상에 인터페이스를 가지고 있는 라우터. (인접 라우터)
- ✓ Adjacency: 일반 neighbor 라우터들 중 point-to-point 또는 virtual link로 연결된 라우터를 Adjacency 라우터라고 하는데, OSPF의 목적은 neighbor 라우터들을 모두 adjacency라우터로 만들어 완전 인접 상태 (Full adjacency state)를 만드는 것이다.
- ✓ DR(Designated Router): 한 area를 대표하는 router이다. OSPF의 경우 Multi Access환경에서 라우터들 간 정보 교환을 1:1로 하지 않고 DR을 선출해서 DR을 통해 정보교환을 하도록 하여 중복 데이터 교환을 최소화한다.

- ✓ LSA (Link State Advertisement): 각 라우터들은 자신의 라우팅 기초 정보를 LSA에 담아 알린다. 이때 라우팅 기초 정보란 OSPF 라우터에 연결된 링크의 link state 정보, 인접 관계 형성, 요약 정보 등 다양한 링크 상태에 관한 정보를 말한다.
- ✓ LSDB (Link State Database): 위에 각 라우터들은 링크 상태 정보를 수집한 후 자신의 데이터베이스를 구축한다고 설명했는데 그 데이터베이스가 LSDB이다.

## 2. OSPF Packet

OSPF 패킷은 총 다섯가지이다.

- 1) **Hello**: neighbor 라우터들과 인접 정보를 수립하고 유지하는 역할. OSPF로 네트워크 topology를 구축할 때 가장 우선적으로 보내는 패킷이다. Multi Access 네트워크에선 DR과 BDR 선출에도 사용된다. 224.0.0.5의 멀티캐스트 주소를 사용하지만 라우터들 간 연결이 virtual link인 경우에는 유니캐스트 IP 주소를 직접 사용하기도 한다.
- 2) **DBD (Database Description)**: Hello 패킷으로 neighbor를 확인했으면 서로 간 LSA를 교환하게 되는데, 이때 주고받는 것이 DBD 패킷이다. OSPF 패킷의 payload 부분에 LSA가 포함된다. (보통은 LSA가 포함되는데, 비어 있는 경우도 있다. 아래에서 설명)
- 3) **LSR (Link State Request)**: OSPF 라우터가 자신의 LSDB의 업데이트가 필요하다고 판단할 때 상대 라우터에게 보내는 패킷이다. 상대 라우터가 보낸 DBD 패킷에 자신에게 없는 LSA가 있게 되면 자신의 LSDB를 최신 상태로 업데이트 시키기 위해 보다 상세한 LSA정보를 요청한다.
- 4) **LSU (Link State Update)**: OSPF 동작에 있어 핵심적인 요소라고 할 수 있는 패킷이다. 각 라우터가 자신의 링크상태 정보(LSA 목록)을 광고할 때 사용하는 패킷인데, 이 과정을 통해 같은 영역의 라우터는 결국 모두 동일한 LSDB를 가지게 된다. 이 패킷은 LSR 패킷의 response로서 보내질 수도 있다.
- 5) **LSA (Link State Acknowledgement)**: LSU 패킷에 대한 응답용 패킷이다. 여러 LSA 헤더들의 리스트가 payload로서 담긴다.

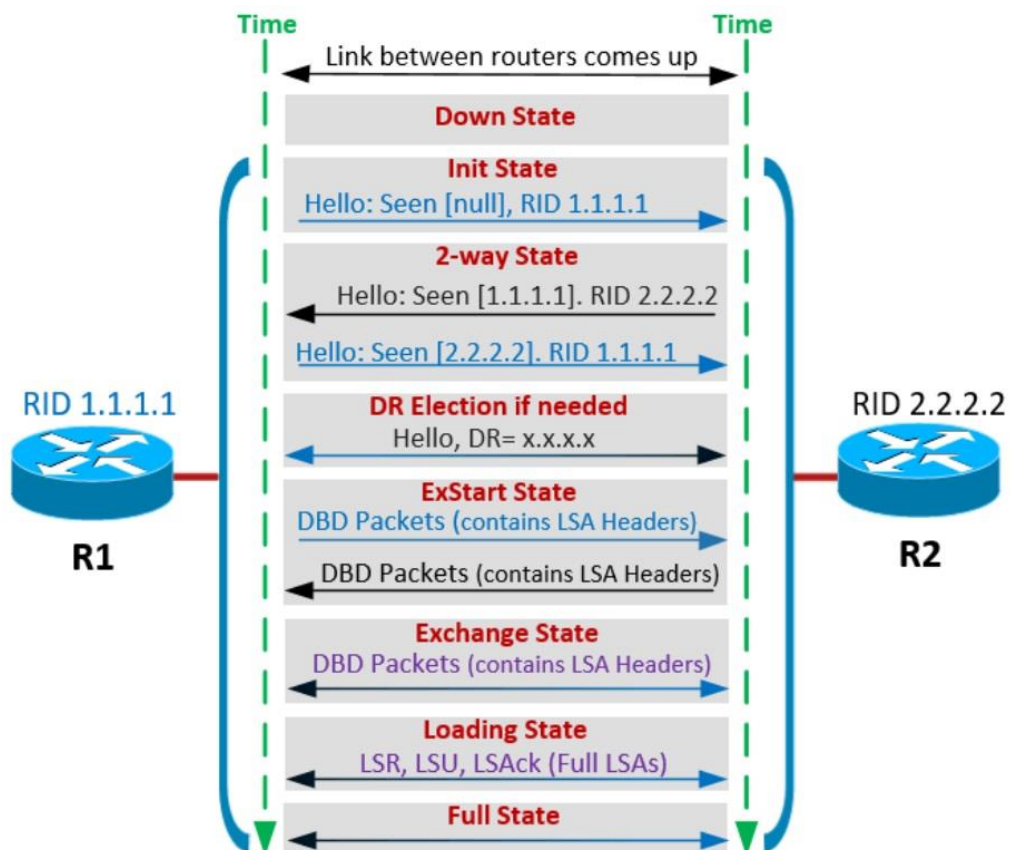
## 3. OSPF routers' state

OSPF router는 neighbor를 discovering할 때 7단계의 state를 거친다.

- 1) **Down**: neighbor 라우터와 정보 교환을 하지 않은 상태
- 2) **Init**: 라우터의 인터페이스가 첫 번째 Hello 패킷을 받으면 Init상태에 들어간다. Init상

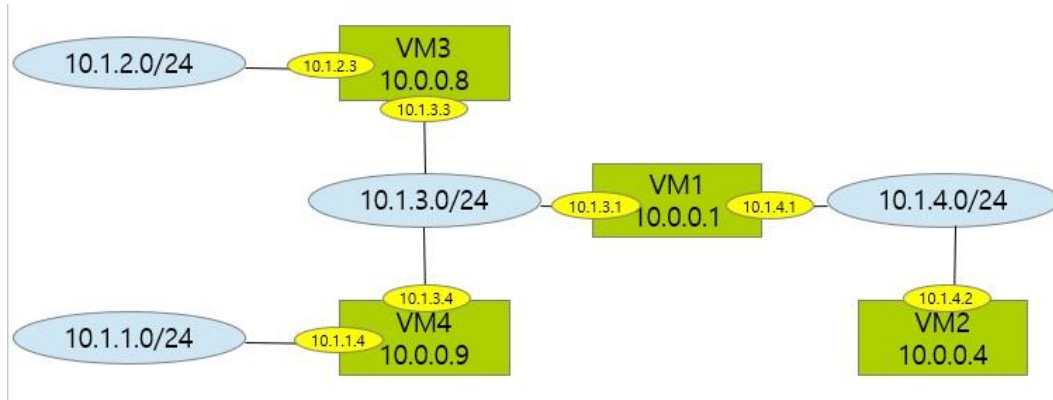
태의 라우터는 hello패킷의 source 라우터가 다음 단계로의 관계를 얻기 위해 기다리고 있음을 알게 된다.

- 3) **2-way**: hello 패킷을 사용해서 같은 영역의 라우터들은 모든 neighbor 라우터들과 양방향 (Bidirectional) 통신을 수립하려고 시도한다. Hello 패킷에는 자신이 알고 있는 neighbor라우터의 리스트를 담아 보내는데, 송신 라우터가 이 리스트에서 자신을 발견하면 2-way 상태에 진입하게 된다.
- 4) **ExStart**: 2-way 상태의 두 라우터는 서로 간의 DBD를 교환하고 master/slave관계를 협상한다.
- 5) **Exchange**: 라우터들은 LSA header가 담긴 DBD 패킷을 교환하게 된다. 라우터는 이 패킷의 정보를 보고 자신의 LSDB에 추가 또는 수정할 정보가 있는지 체크한다.
- 6) **Loading**: 이 단계에서 라우터들은 LSR, LSU 패킷을 주고받으며 전체 LSA 정보를 얻는다. 이 과정에서 라우터는 자신이 알게 된 새로운 정보를 상대와 공유하고, LSDB를 업데이트 시킨다.
- 7) **Full Adjacency**: 모든 라우터의 LSDB가 완전히 synchronized된, 완전 인접 상태이다. OSPF를 통해 Network Topology를 완성한 단계이다.



## ■ OSPF Packet Analysis

과제에서 주어진 Network Topology는 아래와 같다.



(VM1으로 설정한) OSPF1에서 사용하는 내부 네트워크 10.1.3.0은 OSPF3 (VM3), OSPF4 (VM4)와 연결되어있고, 10.1.4.0은 OSPF2 (VM2)와 연결되어 있으므로 각각의 패킷들을 캡처 하는 Wireshark 프로그램 두 개를 OSPF1에서 실행시킨다. (**eth1**: 10.1.3.0 (OSPF3, OSPF4)/**eth2**: 10.1.4.0 (OSPF2))

### 1. Packet capture [eth1]

eth1\_new.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ospf

No.	Time	Source	Destination	Protocol	Length	Info
5	19.999304	10.1.3.1	224.0.0.5	OSPF	78	Hello Packet
6	29.999238	10.1.3.1	224.0.0.5	OSPF	78	Hello Packet
7	39.998654	10.1.3.1	224.0.0.5	OSPF	78	Hello Packet
8	49.998517	10.1.3.1	224.0.0.5	OSPF	78	Hello Packet
9	59.998961	10.1.3.1	224.0.0.5	OSPF	78	Hello Packet
10	69.999364	10.1.3.1	224.0.0.5	OSPF	78	Hello Packet
12	77.116021	10.1.3.3	224.0.0.5	OSPF	78	Hello Packet
14	78.719969	10.1.3.4	224.0.0.5	OSPF	78	Hello Packet
16	79.998774	10.1.3.1	224.0.0.5	OSPF	86	Hello Packet
19	80.000394	10.1.3.3	10.1.3.1	OSPF	66	DB Description
20	80.000677	10.1.3.1	10.1.3.3	OSPF	66	DB Description
21	80.000866	10.1.3.1	10.1.3.3	OSPF	166	DB Description
22	80.001387	10.1.3.3	10.1.3.1	OSPF	118	LS Request
23	80.001645	10.1.3.1	10.1.3.3	OSPF	258	LS Update
24	80.002064	10.1.3.3	10.1.3.1	OSPF	86	DB Description
25	80.002256	10.1.3.1	10.1.3.3	OSPF	70	LS Request
26	80.002665	10.1.3.3	10.1.3.1	OSPF	98	LS Update
27	80.002812	10.1.3.1	10.1.3.3	OSPF	66	DB Description
28	80.003828	10.1.3.1	224.0.0.5	OSPF	142	LS Update
32	80.015494	10.1.3.4	10.1.3.1	OSPF	66	DB Description

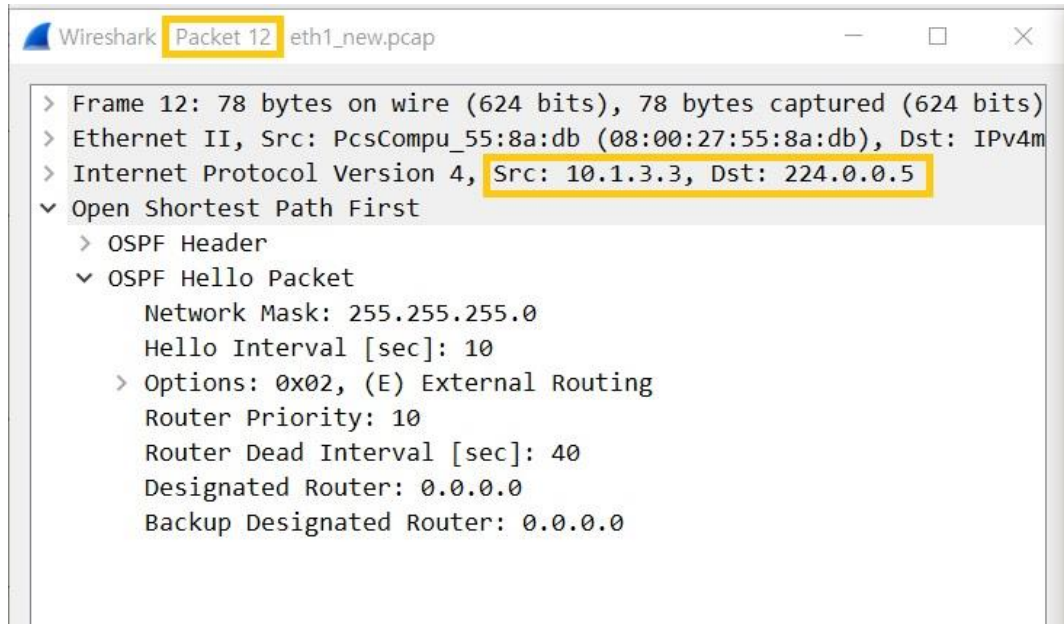
> Frame 1: 78 bytes on wire (624 bits), 78 bytes captured (624 bits)  
> Ethernet II, Src: PcsCompu\_a1:bb:9f (08:00:27:a1:bb:9f), Dst: IPv4mcast\_05 (01:00:5e:00:00:05)  
> Internet Protocol Version 4, Src: 10.1.3.1, Dst: 224.0.0.5  
> Open Shortest Path First

#### a) Hello packet

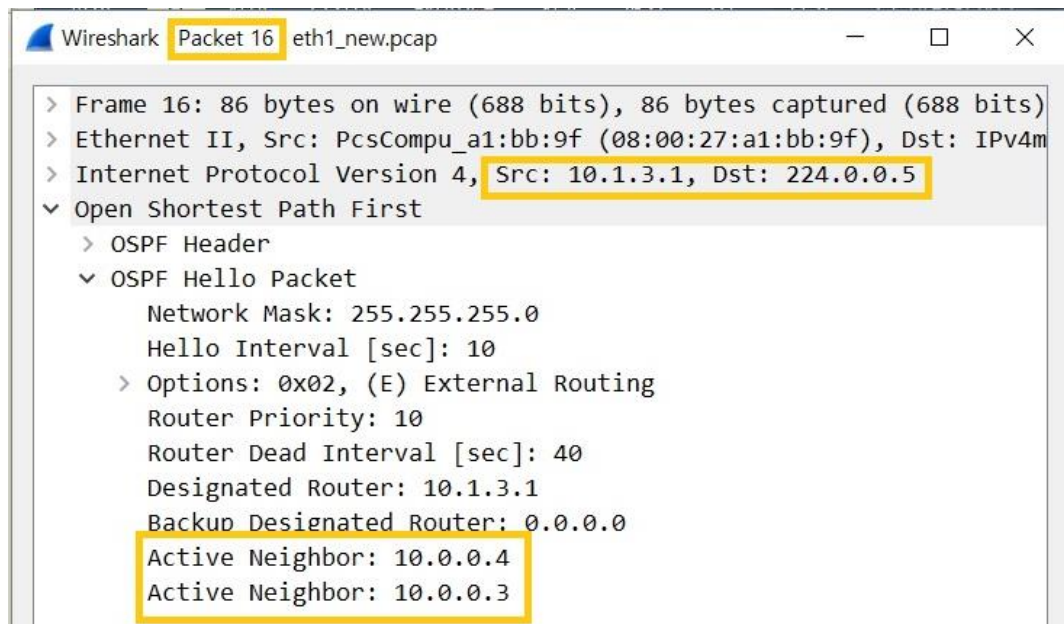
eth1의 모든 라우터들은 멀티캐스트로 hello 패킷을 보내고 있다. 라우터들은 서로의



존재를 알리고 인식한다. 라우터들은 hello 패킷을 받게 되면 **Init**상태에 돌입하게 되는데, **2-way** 상태로 들어가기 위해 hello 패킷에 자신이 알고 있는 neighbor 라우터의 정보를 담아서 보낸다.



이 패킷은 OSPF3에서 보낸 hello 패킷이다. OSPF3는 아직 어떤 neighbor도 확인하지 못한 상태이다.

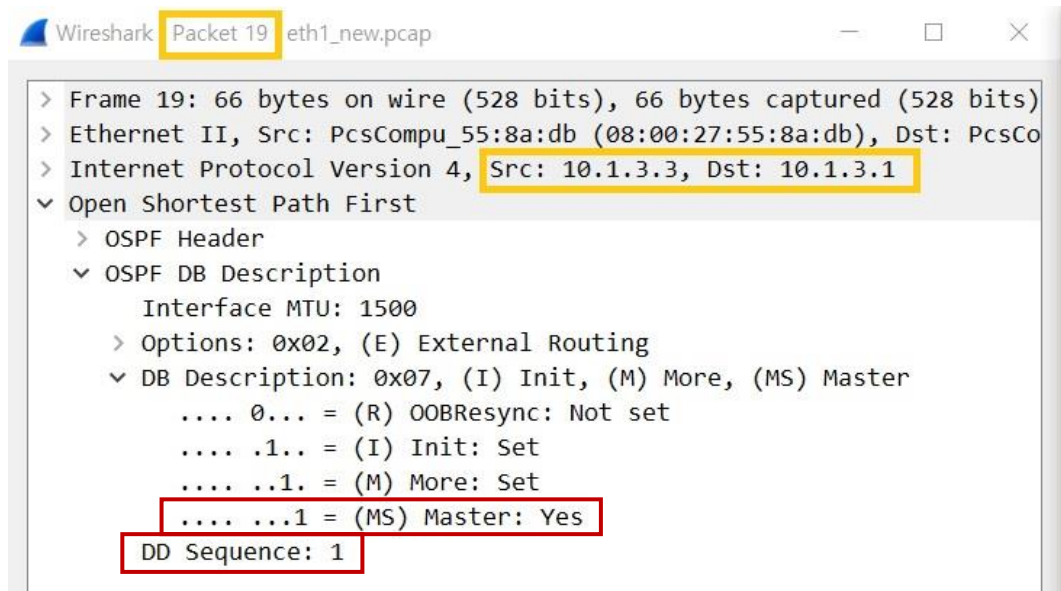


위의 패킷 이후에 OSPF1에서 보낸 hello 패킷이다. Active Neighbor를 보면 OSPF1은 OSPF4와 OSPF3의 존재를 알고 있다는 것을 알 수 있다. 이 패킷이 OSPF3와 OSPF4에 도착하게 되면 각 라우터는 패킷 내에 자신의 정보가 존재함을 알고, OSPF1과 **2-way** 상태에 돌입하게 된다. 이제 OSPF3와 OSPF1, OSPF4와 OSPF1은 양방향 통신이

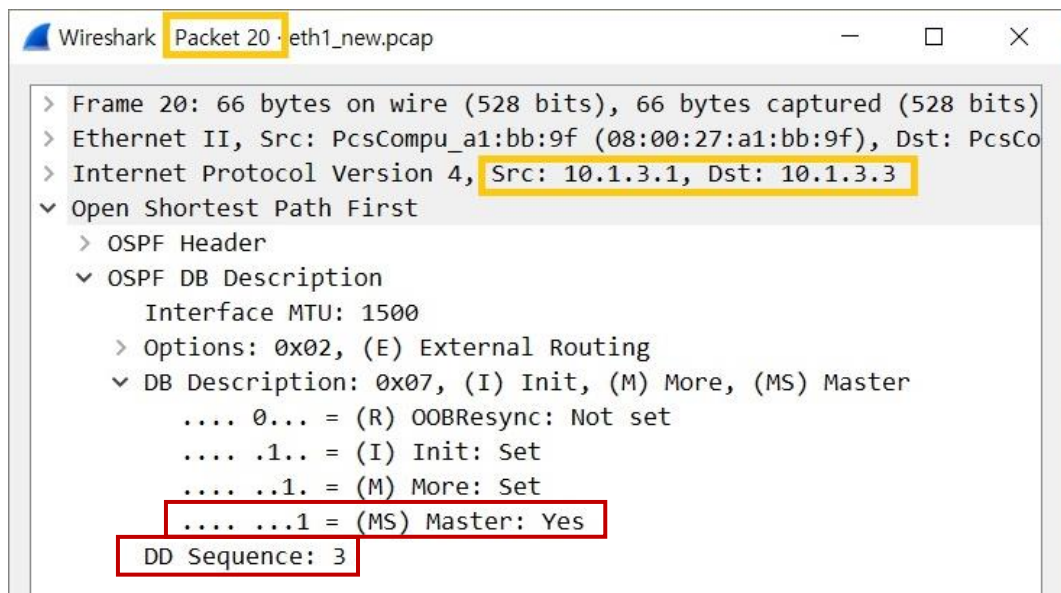
가능하다.

## b) DBD packet

2-way상태에 돌입한 두 라우터는 DBD 패킷을 서로 교환하게 되고 master/slave관계를 정한다.

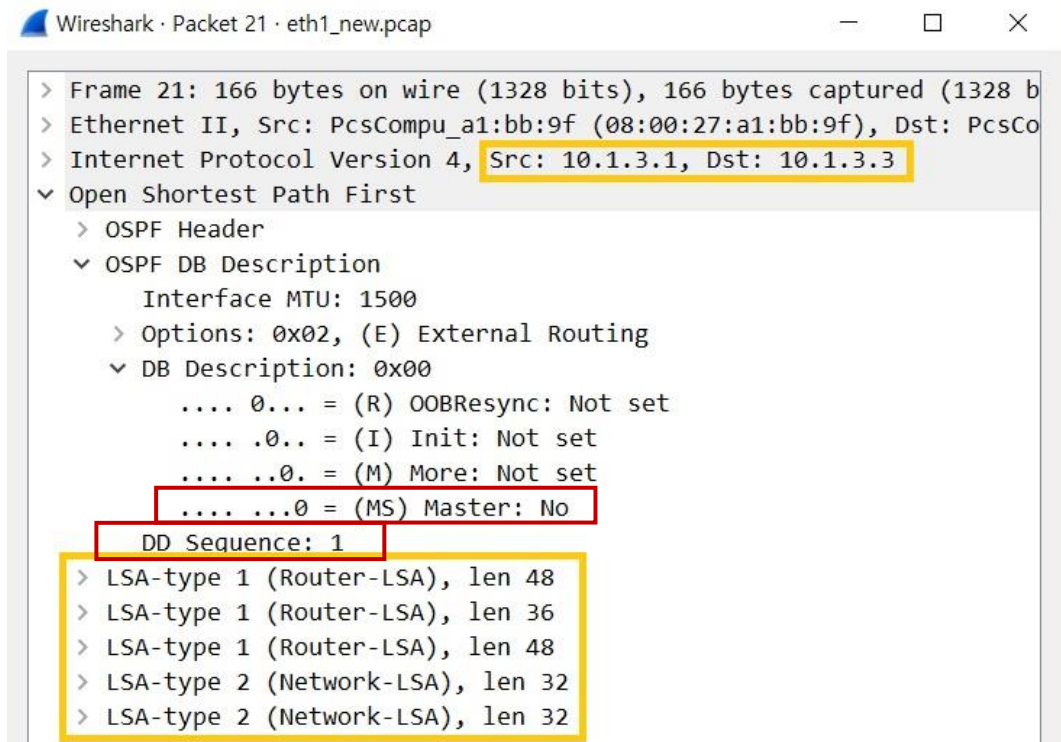


OSPF3에서 OSPF1으로 보내는 DBD 패킷이다. 아직 두 라우터는 master/slave 관계를 정하는 중이므로 master를 요청하는 빈 DBD패킷을 보낸다.



OSPF1에서 OSPF3으로 보내는 DBD 패킷이다. 마찬가지로 master를 요청하는 빈 DBD패킷이기 때문에 MS가 1로 되어있다.

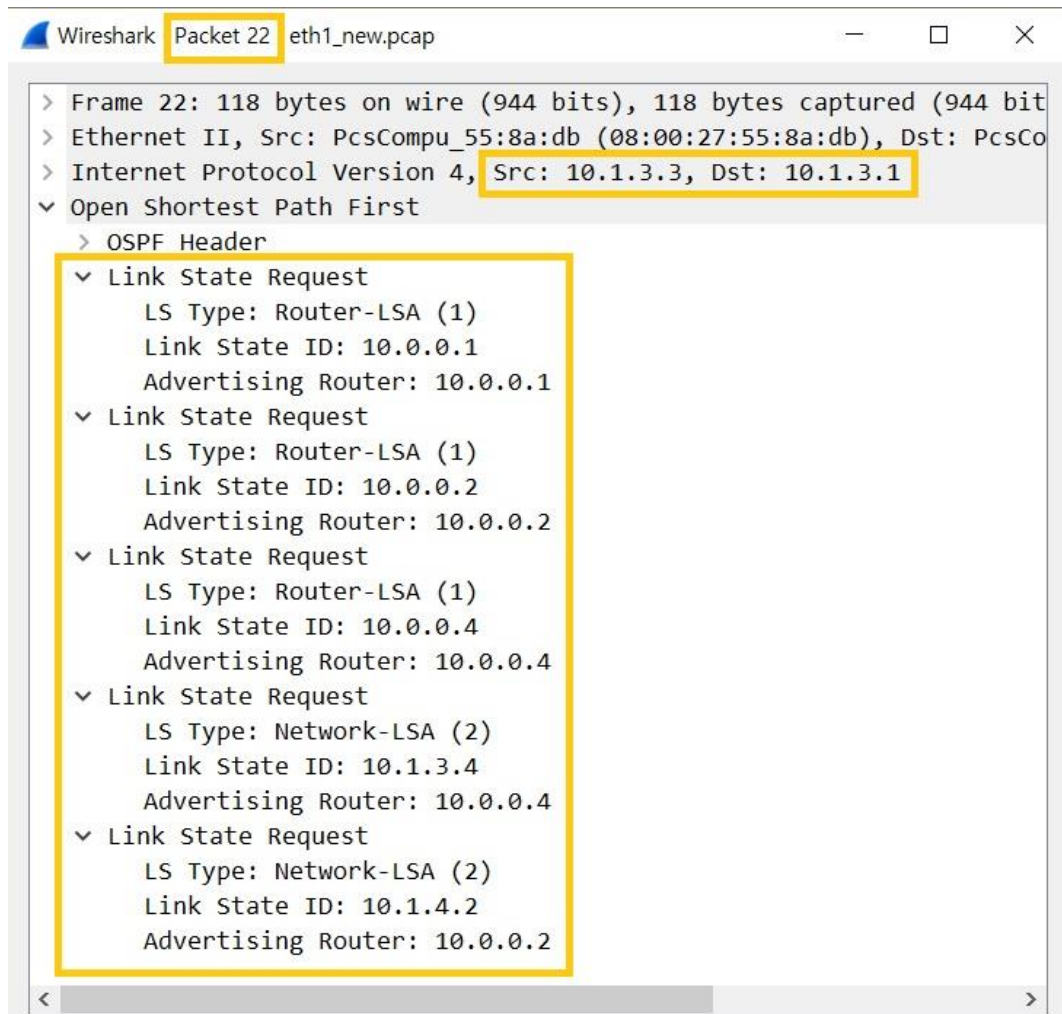




OSPF1과 OSPF3사이의 master/slave 관계가 협상되었다. Router ID가 높은 OSPF3이 master가 되었으며 따라서 이 패킷의 MS는 0으로 셋팅 되어있고 DD Sequence도 master인 OSPF3의 것을 따르게 된다.

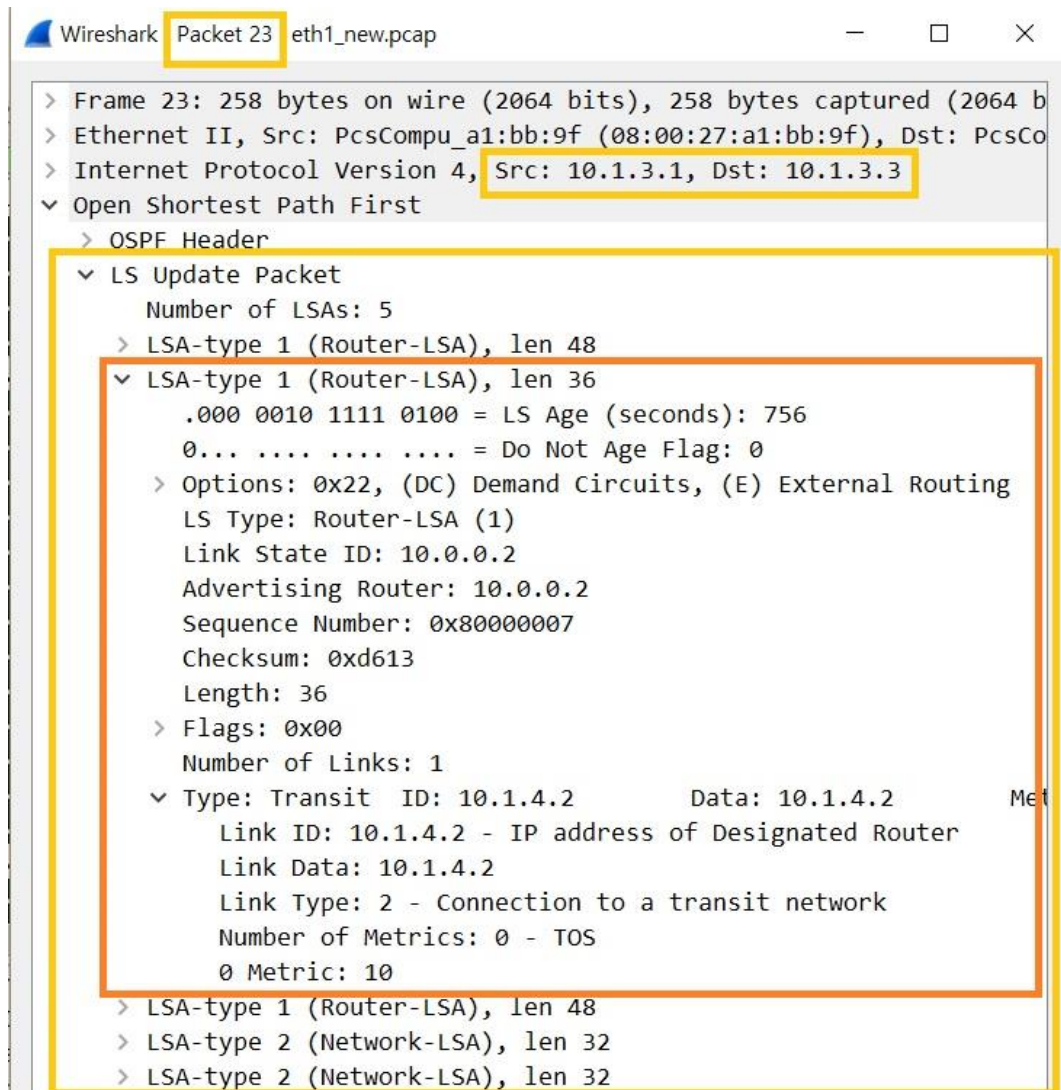
앞선 두 DBD 패킷은 **ExStart** 상태를, 이 패킷은 **Exchange** 상태를 나타낸다. OSPF3는 이 패킷을 받고 자신의 LSDB와 비교하여 추가 또는 수정할 사항이 있는지 확인할 것이다.

c) LSR packet



OSPF3은 OSPF1이 보낸 DBD 패킷(바로 직전 사진의 패킷)에 담겨있던 5개의 LSA에 대한 정보가 현재 자신의 LSDB에는 없기 때문에 (없거나 동일하지 않기 때문에) 업데이트를 위해 더 상세한 정보가 필요하다고 판단하였다. 따라서 정보 요청 대상인 LSA 목록을 담아 DBD패킷의 출처인 OSPF1에게 보낸다.

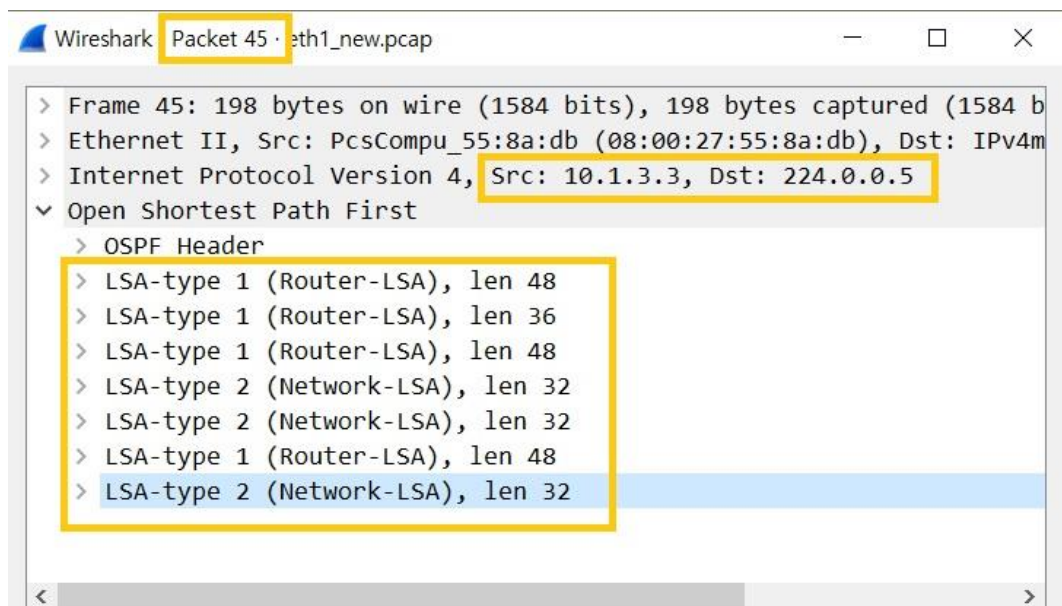
#### d) LSU packet



OSPF3에게 LSR 패킷을 받았던 OSPF1은 요청받은 LSA에 대한 상세한 정보를 LSU 패킷에 담아 response한다. DBD에 들어가는 LSA는 LSA의 header로 type link state ID, advertising router등의 기본적인 정보만 포함하지만 요청에 LSU 안의 LSA에는 link data, metric등 더 상세한 정보가 서술 되어있다.

OSPF1과 OSPF3처럼 서로 LSR, LSU 패킷을 주고받으며 자신의 LSDB를 업데이트 하는 상태를 **Loading** 상태라고 한다.

#### e) LSA packet



LSU패킷들을 통해 자신의 LSDB를 업데이트하고 그 정보 = LSA 목록을 패킷에 담아 보낸다. 이때 LSA는 header 부분만 담기게 되고 일반적으로 이 패킷은 멀티캐스트에 의해 전송된다.

라우터들은 LSR, LSU를 반복적으로 주고받으며 LSDB를 업데이트 시켜 나가고, 최종적으로 모든 라우터의 LSDB가 synchronized되면 각 라우터는 **full state** 상태가 된다. 이후에 각 라우터는 멀티캐스트로 살아있음을 인접 라우터들에 알리고 인접관계를 유지시킨다.

137	309.999370	10.1.3.1	224.0.0.5	OSPF	86 Hello Packet
138	310.836968	10.1.3.4	224.0.0.5	OSPF	86 Hello Packet
139	312.418321	10.1.3.3	224.0.0.5	OSPF	86 Hello Packet
140	319.998764	10.1.3.1	224.0.0.5	OSPF	86 Hello Packet
141	320.836898	10.1.3.4	224.0.0.5	OSPF	86 Hello Packet

(패킷 번호가 137~140으로 위의 패킷들의 것과 크게 차이 남) 모든 정보교환이 끝난 이후에도 지속적으로 hello packet을 보내며 인접관계를 유지한다.

## 2. Packet capture [eth2]

Neighbor를 발견하고 인접관계를 형성하는 전체적인 과정은 위와 같다. **Loading** 상태에서, LSR과 LSU를 교환하는 과정에서 eth1의 라우터들은 eth2의 OSPF2에 대한 LSA도 주고받으며, eth2의 라우터는 eth1의 OSPF3, OSPF4에 대한 LSA도 주고받게 된다.

eth1_capture.pcap						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
ospf						
No.	Time	Source	Destination	Protocol	Length	Info
34	241.495853	10.1.4.1	10.1.4.2	OSPF	82	Hello Packet
35	249.547512	10.1.4.2	224.0.0.5	OSPF	82	Hello Packet
36	251.495875	10.1.4.1	10.1.4.2	OSPF	82	Hello Packet
39	259.547137	10.1.4.2	224.0.0.5	OSPF	82	Hello Packet
40	261.266580	10.1.4.1	10.1.4.2	OSPF	66	DB Description
41	261.496464	10.1.4.1	10.1.4.2	OSPF	82	Hello Packet
42	265.871337	10.1.4.2	10.1.4.1	OSPF	66	DB Description
43	265.871683	10.1.4.1	10.1.4.2	OSPF	86	DB Description
44	265.873955	10.1.4.2	10.1.4.1	OSPF	70	LS Request
45	265.874249	10.1.4.1	10.1.4.2	OSPF	110	LS Update
46	265.875031	10.1.4.2	10.1.4.1	OSPF	86	DB Description
47	265.875278	10.1.4.1	10.1.4.2	OSPF	70	LS Request
48	265.875901	10.1.4.2	10.1.4.1	OSPF	98	LS Update
49	265.877304	10.1.4.1	10.1.4.2	OSPF	66	DB Description
50	265.877582	10.1.4.1	10.1.4.2	OSPF	110	LS Update
52	265.878517	10.1.4.2	224.0.0.5	OSPF	130	LS Update
53	266.422405	10.1.4.1	10.1.4.2	OSPF	118	LS Acknowledge
54	266.735810	10.1.4.2	224.0.0.5	OSPF	98	LS Acknowledge
55	269.548031	10.1.4.2	224.0.0.5	OSPF	82	Hello Packet
59	271.496082	10.1.4.1	10.1.4.2	OSPF	82	Hello Packet
61	273.356524	10.1.4.1	10.1.4.2	OSPF	158	LS Update
62	273.357550	10.1.4.1	10.1.4.2	OSPF	142	LS Update
63	274.216967	10.1.4.2	224.0.0.5	OSPF	138	LS Acknowledge
67	279.547732	10.1.4.2	224.0.0.5	OSPF	82	Hello Packet
68	281.494906	10.1.4.1	10.1.4.2	OSPF	82	Hello Packet
71	289.548021	10.1.4.2	224.0.0.5	OSPF	82	Hello Packet
72	291.495590	10.1.4.1	10.1.4.2	OSPF	82	Hello Packet
73	299.547683	10.1.4.2	224.0.0.5	OSPF	82	Hello Packet
74	301.495158	10.1.4.1	10.1.4.2	OSPF	82	Hello Packet

Init  
2-way  
ExStart  
Exchange  
Loading  
Full  
Adjacency

## ospfd\_mon Analysis

OSPF1에서 ospfd\_mon을 실행시켜 ospfd의 내부 상태 정보와 출력 값을 살펴본다.

```

root@jiyoon-VirtualBox:~/Local/bin# ./ospfd_mon
Connecting to jiyoon-VirtualBox:12767 ...connected

OSPF Router ID: 10.0.0.1      # AS-external-LSAs:      0
ASE checksum: 0x0             # ASEs originated:      0
ASEs allowed: 10              # Dijkstras:            8
# Areas: 1                     # Nbrs in Exchange:    0
MOSPF enabled: no             Inter-area multicast: yes
Inter-AS multicast: no        In overflow state: no
ospfd version: 0.1

```

ospfd\_mon을 실행했을 때의 기본 화면이다.

### a) interfaces

10.0.0.1> interfaces							
Phy	Addr	Area	Type	State	#Nbr	#Adj	Cost
eth1	10.1.3.1	0.0.0.0	BCast	DR	2	2	10
eth2	10.1.4.1	0.0.0.0	NBMA	Backup	1	1	10

OSPF1의 interfaces 정보를 보여준다. phy는 OSPF1에서 설정해줬던 내부 네트워크 정보를, Addr는 각 내부 네트워크에 연결 되어있는 interface 주소를 보여준다. 두 interface 모두 area 0.0.0.0에 존재하고, state를 통해 OSPF1이 eth1에선 DR역할을, eth2에선 Backup 역할을 한다는 것을 알 수 있다. #Nbr와 #Adj는 OSPF1의 neighbor와 adjacency 수를 나타내는데, 지금은 완전 인접 상태이기 때문에 #Nbr와 #Adj가 같은 값을 나타낸다.



## b) neighbors

```
10.0.0.1> neighbors
Phy      Addr      ID      State  #DD  #Rq  #Rxmt
eth1     10.1.3.3  10.0.0.3 Full   0    0    0
eth1     10.1.3.4  10.0.0.4 Full   0    0    0
eth2     10.1.4.2  10.0.0.2 Full   0    0    0
```

OSPF1의 neighbors 목록을 나타낸다. Neighbor가 존재하는 physical link와 interface의 IP 주소, Router ID, state를 나타내는데, 현재는 완전 인접 상태이기 때문에 "Full"이라는 값을 나타낸다.

## c) areas

```
10.0.0.1> areas
Area ID    #Ifcs  #Routers #LSAs  Xsum      Comments
0.0.0.0    2      4        6      0x2a55a   demand-capable
```

OSPF1이 존재하는 area에 대한 정보를 출력한다. 과제에서 주어진 topology의 네 라우터 모두 같은 area 0.0.0.0에 존재하므로 #Routers는 4를 나타낸다. #LSAs는 해당 area에 대한 LSA의 수를, #Ifcs는 OSPF1이 area에 대해 가진 interface수를 나타낸다.

## d) database

```
10.0.0.1> database 0.0.0.0
Type      LS_ID      ADV_RTR      Seqno      Xsum      Age
1         10.0.0.1   10.0.0.1 0x80000015 0xcadc    855
1         10.0.0.2   10.0.0.2 0x80000011 0xc21d    1610
1         10.0.0.3   10.0.0.3 0x8000000c 0xaa3a    857
1         10.0.0.4   10.0.0.4 0x80000014 0x0ea5    855
2         10.1.3.1   10.0.0.1 0x8000000c 0x3baf    851
2         10.1.4.2   10.0.0.2 0x80000010 0x23d3    1611
# LSAs: 6
Database xsum: 0x2a55a
```

database 키워드는 area를 인자로 함께 받는데, 과제의 topology 라우터들은 하나의 area 0.0.0.0에 존재하고, 이는 6개의 LSA를 갖는다. Type1은 router type, type2는 network type을 나타내는데, router type은 router마다 하나씩 생성되어 총 4개이고, network type은 transit network마다 생성이 되는데, transit network란 2개 이상의 router가 붙은 네트워크를 말한다. 따라서 OSPF1 기준으로 transit 네트워크는 eth1, eth2 두개이다. LSA마다 link state ID와 advertisement router도 같이 서술 되어있다.

## e) as-externals

```
10.0.0.1> as-externals
Type      LS_ID      ADV_RTR      Seqno      Xsum      Age
# LSAs: 0
Database xsum: 0x0
```

AS-external-LSA의 목록을 나타내는 옵션이다. AS-external-LSA란 Type이 5인 LSA로 AS 외부에 있는 모든 네트워크를 알리는 역할을 한다. 과제에서 주어진 network topology에 따르면 OSPF1을 포함한 4개의 라우터는 총 4개의 내부 네트워크를 사용하고 하나의 area를 형성하며 외부로의 연결은 생성하지 않았다. 따라서 AS-external-LSA는 존재하지

않는다.

f) **adv**

```
10.0.0.1> adv 1 10.0.0.2 10.0.0.2 0.0.0.0
LS age: 48
LS Options: 0x22
LS Type: 1
Link State ID: 10.0.0.2
Advert. Rtr.: 10.0.0.2
LS Seqno: 0x80000012
LS Xsum: 0xc01e
LS Length: 36
// Router-LSA body
Router type: 0x0
# links: 1
// Link #0
Link ID: 10.1.4.2
Link Data: 10.1.4.2
Link type: 2
# TOS metrics: 0
Link cost: 10
```

Adv는 type, link state ID, advertising router 등 LSA를 나타내는 기본 정보를 받고 해당 LSA에 대한 상세한 정보를 출력하는 기능을 한다. Type 1의 라우터 OSPF2 LSA를 요청하였고 Link Data, metric 등 링크 상태에 대한 정보를 보여주고 있다.

g) **routes**

```
10.0.0.1> routes
Prefix          Type    Cost    Ifc    Next-hop    Mpaths
10.1.1.0/24     SPF     20      eth1   10.1.3.4
10.1.3.0/24     SPF     10      eth1   n/a
10.1.4.0/24     SPF     10      eth2   n/a
```

OSPF에 의해 계산된 routing table을 출력한다. OSPF1의 두 interface 중 forwarding 시 어느 interface를 사용할 것인지는 IP address의 prefix에 따라 결정된다. ospfd setup할 때 라우터들의 모든 interface에 대해 cost (metric)을 10으로 설정해주었기 때문에 1-hop-away인 항목 (첫 번째, 세 번째 행)에 대해선 cost가 10이 되고, next hop이 다른 IP address로 설정 되어있는 항목 (첫 번째 행)에 대해서는 cost가 10을 초과한다. 이 행의 cost는 20이라고 나와있는데, next hop인 10.1.3.4를 거치면 destination까지 1-hop-away 상태라는 것을 유추할 수 있다. (모든 interface의 cost를 10으로 설정해주었기 때문)

## ■ References

- ✓ [http://www.ktword.co.kr/abbr\\_view.php?nav=2&id=440](http://www.ktword.co.kr/abbr_view.php?nav=2&id=440)
- ✓ <http://www.firewall.cx/networking-topics/routing/ospf-routing-protocol/1142-ospf-adjacency-neighbor-states-forming-process.html>
- ✓ <https://www.netmanias.com/ko/post/blog/5476/ip-routing-network-protocol-ospf-shortest-path-tree/ospf-basic-part-1-build-of-shortest-path-tree-topology>