1 Understand the working of OSPF (Level 3)

1.1 Objective

To understand the working of OSPF and Shortest Path First (SPF) tree creation.

1.2 Theory

OSPF

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) standardized by the Internet Engineering Task Force (IETF) and commonly used in large Enterprise networks. OSPF is a link-state routing protocol providing fast convergence and excellent scalability. Like all link-state protocols, OSPF is very efficient in its use of network bandwidth.

Shortest path First Algorithm

OSPF uses a shorted path first algorithm to build and calculate the shortest path to all known destinations. The shortest path is calculated with the use of the Dijkstra algorithm. The algorithm by itself is quite complicated. This is a very high level, simplified way of looking at the various steps of the algorithm:

- Upon initialization or due to any change in routing information, a router generates a link-state advertisement. This advertisement represents the collection of all link-states on that router.

- All routers exchange link-states by means of flooding. Each router that receives a link-state update should store a copy in its link-state database and then propagate the update to other routers.

- After the database of each router is completed, the router calculates a Shortest Path Tree to all destinations. The router uses the Dijkstra algorithm in order to calculate the shortest path tree. The destinations, the associated cost and the next hop to reach those destinations form the IP routing table.

- In case no changes in the OSPF network occur, such as cost of a link or a network being added or deleted, OSPF should be very quiet. Any changes that occur are communicated through link-state packets, and the Dijkstra algorithm is recalculated in order to find the shortest path.

The algorithm places each router at the root of a tree and calculates the shortest path to each destination based on the cumulative cost required to reach that destination. Each router will have its own view of the topology even though all the routers will build a shortest path tree using the same link-state database.

Example

Refer from OSPF RFC 2328 (https://tools.ietf.org/html/rfc2328#section-2.3)

The below network shows a sample map of an Autonomous System.

```
        +
        |  3+---+                          N12        N14
      N1|--|RT1|\ 1                          \ N13 /
        |   +---+ \                          8\ |8/8
        +          \  ___                      \|/
                    /    \    1+---+8      8+---+6
                   *  N3   *---|RT4|------|RT5|---------+
                    \___ /     +---+        +---+       |
          +          /   |                   |7         |
          |  3+---+  /    |                   |          |
        N2|--|RT2|/1     |1                  |6         |
          |   +---+    +---+8            6+---+          |
          +           |RT3|--------------|RT6|          |
                      +---+               +---+          |
                       |2                 Ia|7          |
                       |                    |           |
                +---------+                 |           |
                    N4                      |           |
                                            |           |
                                            |           |
                                            |           |
            N11                             |           |
        +---------+                         |           |
            |                               |           |  N12
            |3                              |          |6 2/
         +---+                              |         +---+/
         |RT9|                              |         |RT7|---N15
         +---+                              |         +---+ 9
          |1                                |          |1
         _|__                          +    |         __|_
        /    \     1+----+2            |  Ib|5        /    \
       *  N9   *------|RT11|----   |---|RT10|---*  N6   *
        \____/        +----+   |    3+----+1   \____/
          |                    |                 |
          |1                   +                 |1
    +--+   10+----+            N8              +---+
    |H1|-----|RT12|                           |RT8|
    +--+SLIP +----+                           +---+
          |2                                   |4
          |                                    |
      +---------+                          +---------+
          N10                                  N7
```
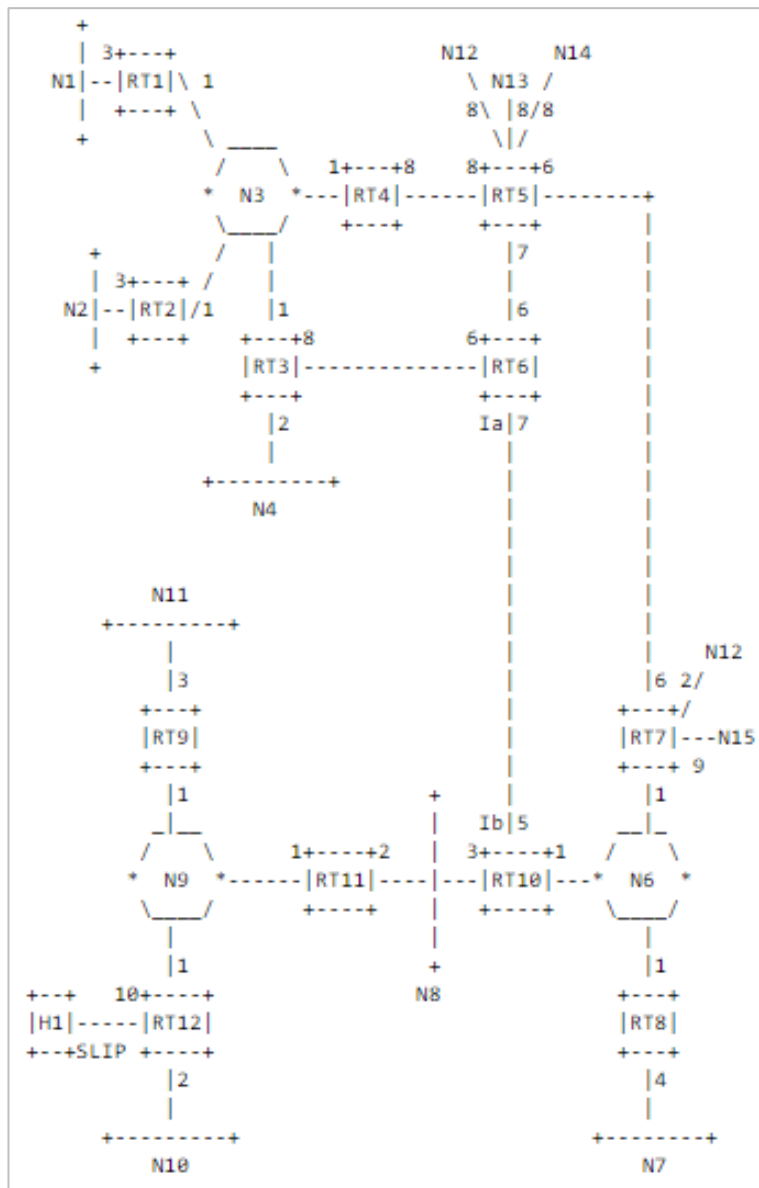
Figure 1-1: Sample maps of an Autonomous system

A cost is associated with the output side of each router interface. This cost is configurable by the system administrator. The lower the cost, the more likely the interface is to be used to forward data traffic. Costs are also associated with the externally derived routing data (e.g., the BGP-learned routes).

The directed graph resulting from the above network is depicted in the following table. Arcs are labelled with the cost of the corresponding router output interface. Arcs having no labelled cost have a cost of 0. Note that arcs leading from networks to routers always have cost 0.

		FROM															
		RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT 10	RT 11	RT 12	N3	N6	N8	N9
	RT1													0			
	RT2													0			
	RT3						6							0			
	RT4					8								0			
	RT5				8		6	6									
	RT6			8		7											
	RT7					6									0		
	RT8														0		
	RT9																0
	RT10							7							0	0	
	RT11															0	0
	RT12																0
	N1	3															
	N2		3														
	N3	1	1	1	1												
	N4			2													
	N5																
TO	N6							1	1		1						
	N7								4								
	N8										3	2					
	N9									1		1	1				
	N10												2				
	N11									3							
	N12					8		2									
	N13					8											
	N14					8											
	N15							9									
	H1												10				

Table 1-1: Directed graph

A router generates its routing table from the above directed graph by calculating a tree of shortest paths with the router itself as root. Obviously, the shortest-path tree depends on the

router doing the calculation. The shortest-path tree for Router RT6 in our example is depicted in the following figure.
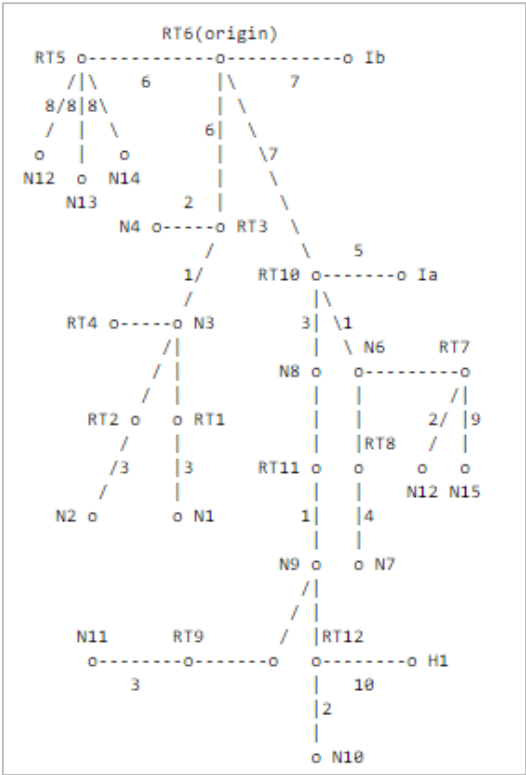
```
                    RT6(origin)
        RT5 o-------------o-----------o Ib
            /|\    6       |\      7
          8/8|8\           | \
          /  |  \        6|  \
         o   |   o        |   \7
        N12  o  N14       |    \
            N13      2    |     \
               N4 o-----o RT3   \
                     /       \    5
                   1/      RT10 o-------o Ia
                   /          |\
           RT4 o-----o N3    3| \1
                /|           |  \ N6      RT7
               / |          N8 o   o---------o
              /  |           |   |        /|
          RT2 o  o RT1       |   |      2/ |9
             /   |           |   |RT8  /   |
            /3   |3      RT11 o   o    o   o
           /     |           |   |    N12 N15
         N2 o    o N1       1|   |4
                             |   |
                           N9 o   o N7
                             /|
                            / |
          N11     RT9      /  |RT12
           o--------o-------o   o--------o H1
               3               |    10
                               |2
                               |
                               o N10
```

Figure 1-2: SPF tree for Router 6

Routing Table

The IP forwarding table formed in the routers and nodes can be accessed from the IP_Forwarding_Table list present in the Simulation Results window as shown below:

Node-8: As shown in the below screenshot, Node-8 has only one interface with IP 11.7.1.2 and its network address are 11.7.0.0 since its network mask is 255.255.0.0. The first entry represents the router forwards packets intended to the subnet   11.7.0.0 to the interface with the IP 11.7.1.2. 239.12.14.5 is the multicast Group address and 224.0.0.1 is the address for all multicast routers The IP forwarding table formed in the routers and nodes can be accessed from the IP_Forwarding_Table list present in the Simulation Results window as shown below:

The tree gives the entire path to any destination network or host. However, only the next hop to the destination is used in the forwarding process. Note also that the best route to any router has also been calculated. For the processing of external data, we note the next hop and distance to

Ver 13.3   Page **4** of **10**

any router advertising external routes. The resulting routing table for Router RT6 is shown in the following table.

Destination	Next hop	Distance
N1	RT3	10
N2	RT3	10
N3	RT3	7
N4	RT3	8
N6	RT10	8
N7	RT10	12
N8	RT10	10
N9	RT10	11
N10	RT10	13
N11	RT10	14
H1	RT10	21
RT5	RT5	6
RT7	RT10	8
N12	RT10	10
N13	RT5	14
N14	RT5	14
N15	RT10	17

Table 1-2: Routing Table for RT6

Distance calculation

Router6 has 3 interfaces i.e., RT3, RT5 and RT10. The distance obtained is 10 for destination N1 via RT3 interface. The packets from Router6 would reach N1 via RT3, N3 and RT1. The cost assigned to routers in this path is 6+1+3 = 10 (cost can be seen in SPF tree for Router6). This is how distance is calculated.

1.3 Network Setup

Open NetSim and click on **Experiments> Internetworks> Routing and Switching> Understand the working of OSPF** then click on the tile in the middle panel to load the example as shown in below Figure 1-3.
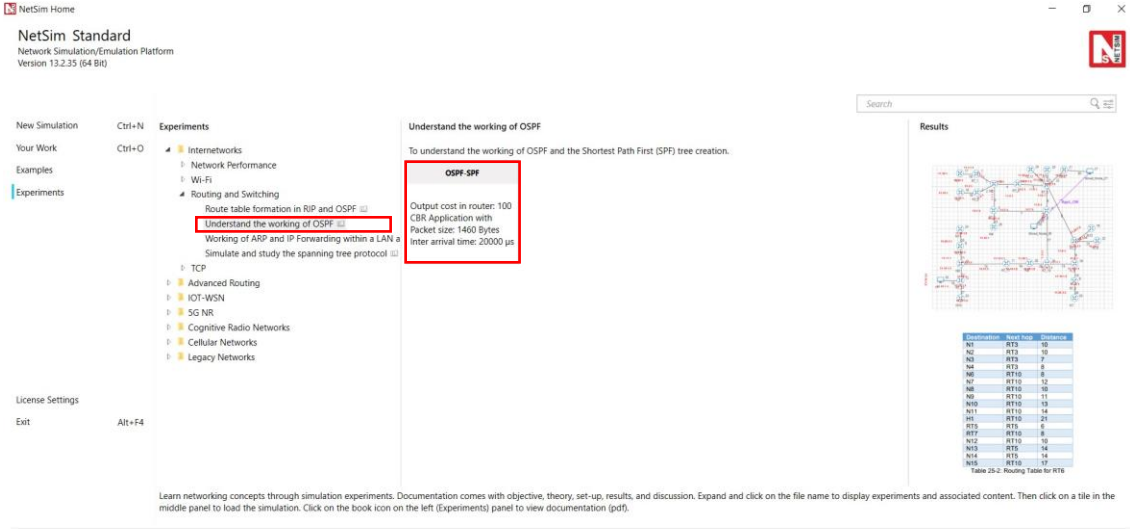
Figure 1-3: List of scenarios for the example of Understand the working of OSPF

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 1-4.
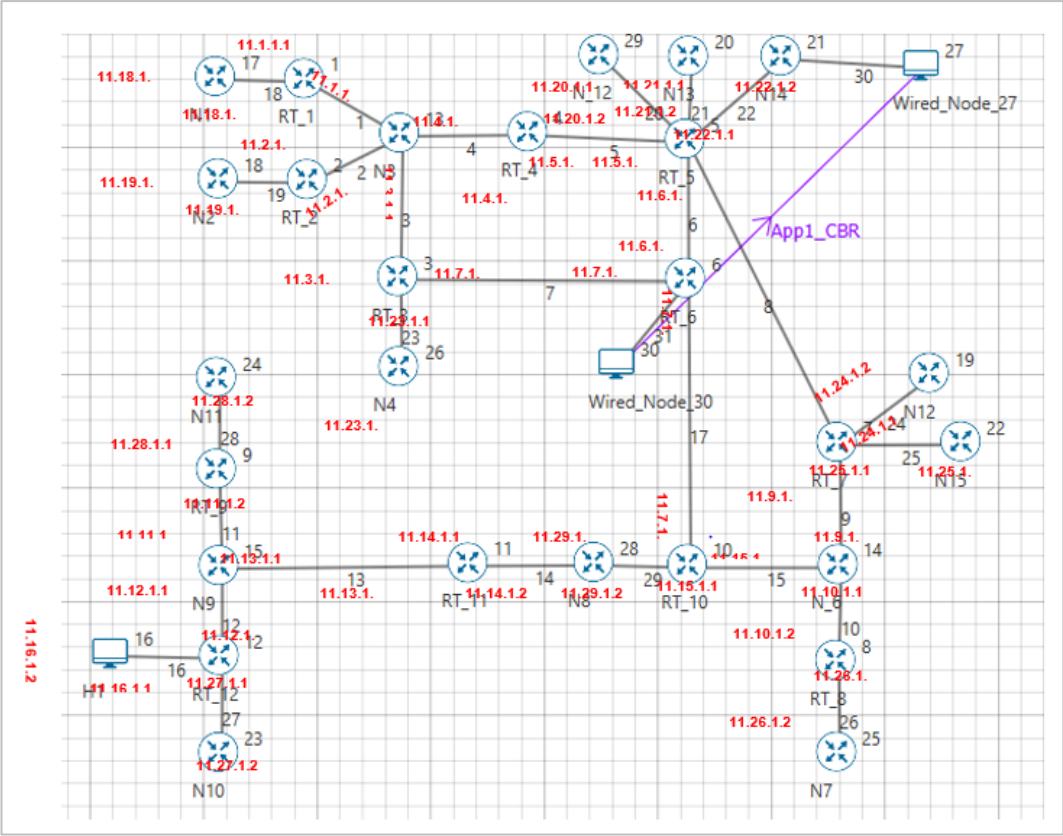
Figure 1-4: Network topology showing IP Addresses in each Router interface and in end nodes

The above network was created in NetSim and it is similar to the network as per the OSPF RFC 2328 (Refer - https://tools.ietf.org/html/rfc2328#page-23)

1.4 Procedure

The following set of procedures were done to generate this sample:

Step 1: A network scenario is designed in NetSim GUI comprising of 3 Wired Nodes and 27 Routers in the **"Internetworks"** Network Library.

Step 2: The Output Cost for all the Routers in the network is set as per the network shown in Figure 1-5.
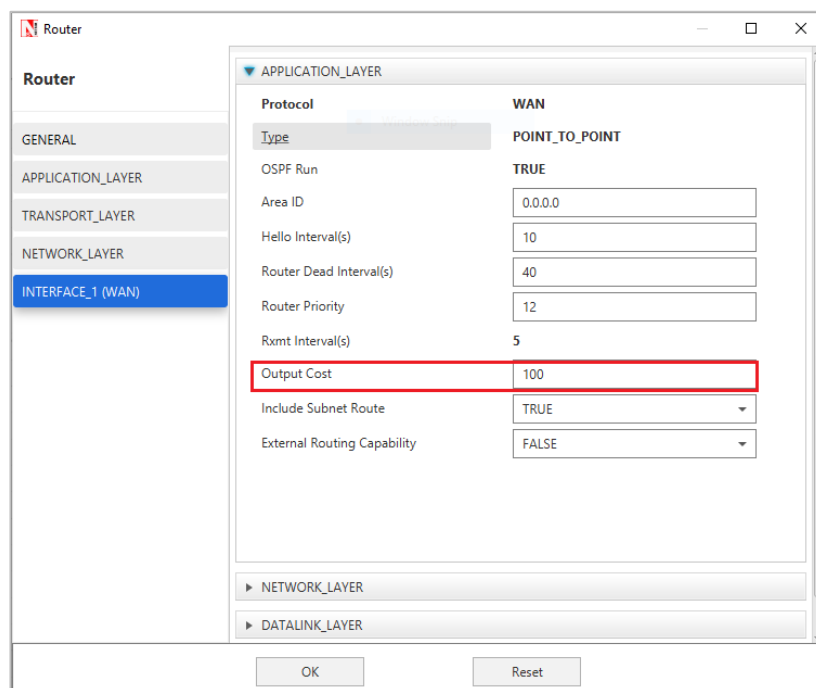
Figure 1-5: Output Cost is set to 100 in WAN interface

Step 3: Packet Trace is enabled in the NetSim GUI, and hence we are able to track the route which the packets have chosen to reach the destination based on the Output Cost that is set.

Step 4: Right click on the Application Flow App1 CBR and select Properties or click on the Application icon present in the top ribbon/toolbar.

A CBR Application is generated from Wired Node 30 i.e. Source to Wired Node 27 i.e. Destination with Packet Size remaining 1460Bytes and Inter Arrival Time remaining 20000μs. Additionally, the "Start Time(s)" parameter is set to 30, while configuring the application. This time is usually set to be greater than the time taken for OSPF Convergence (i.e. Exchange of

OSPF information between all the routers), and it increases as the size of the network increases.

1.5 Output

The following image is a depiction of the shortest path first tree created in NetSim. This is for representational purposes and cannot be opened in NetSim. The blue color numbers are the "Output Cost" parameter of the link and is set by the user in Router > WAN Interface > Application layer. The red numbers are the IP addresses of the interfaces of the Routers.
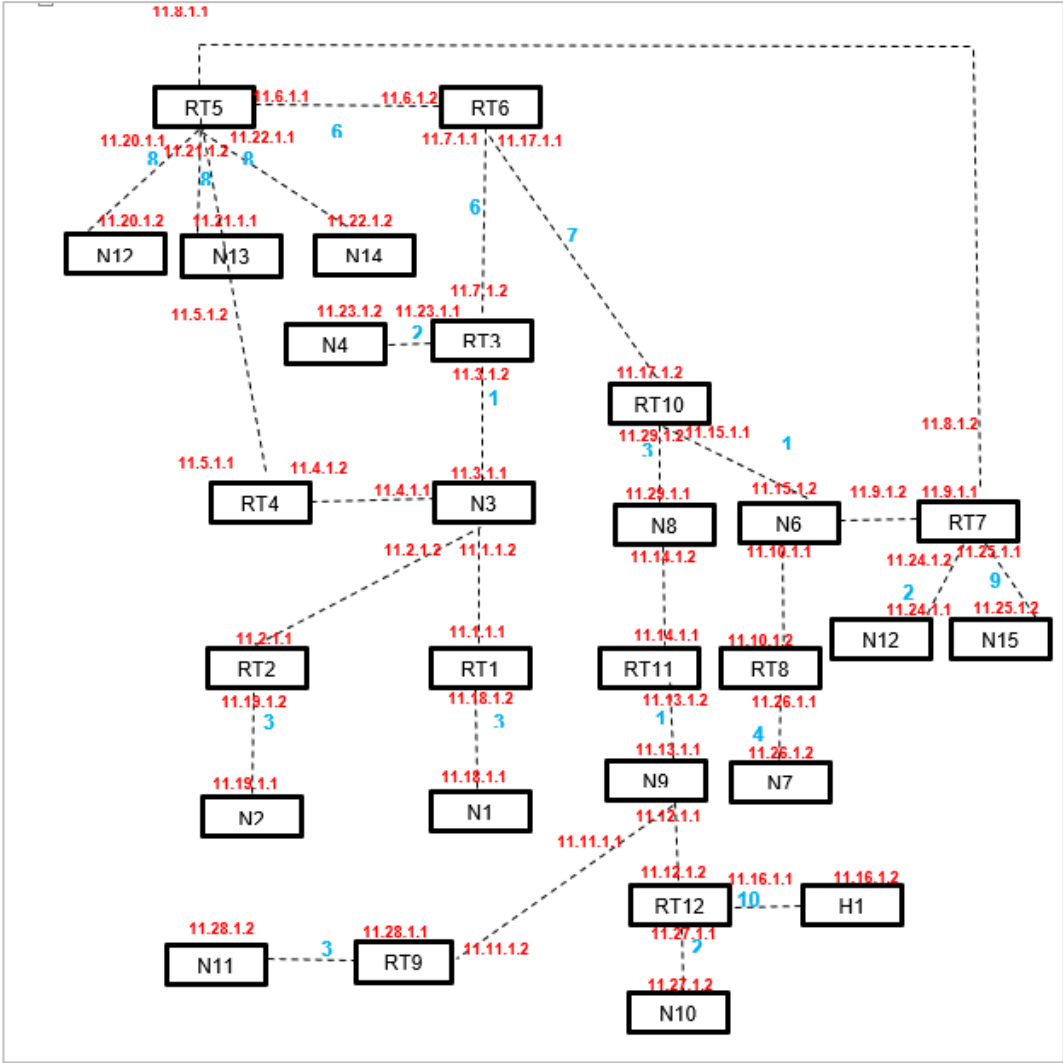

Figure 1-6: SPF tree for Router 6

NOTE: *NetSim, does not implement Link type3 (Link to Stub Network). Hence users would notice a slight difference between the SPF trees of RFC and NetSim.*

The IP forwarding table formed in the routers can be accessed from the IP_Forwarding_Table list present in the Simulation Results window as shown below Figure 1-7.

RT_6_Table

RT_6 (a)

Network Destination	Netmask/Prefix len	Gateway	Interface	Metrics	Type
11.3.1.2	255.255.255.255	11.7.1.2	11.7.1.1	100	OSPF
11.7.1.2	255.255.255.255	11.7.1.2	11.7.1.1	100	OSPF
11.23.1.1	255.255.255.255	11.7.1.2	11.7.1.1	100	OSPF
11.5.1.2	255.255.255.255	11.6.1.1	11.6.1.2	100	OSPF
11.6.1.1	255.255.255.255	11.6.1.1	11.6.1.2	100	OSPF
11.8.1.1	255.255.255.255	11.6.1.1	11.6.1.2	100	OSPF
11.20.1.2	255.255.255.255	11.6.1.1	11.6.1.2	100	OSPF
11.21.1.2	255.255.255.255	11.6.1.1	11.6.1.2	100	OSPF
11.22.1.1	255.255.255.255	11.6.1.1	11.6.1.2	100	OSPF
11.29.1.2	255.255.255.255	11.17.1.2	11.17.1.1	100	OSPF
11.15.1.1	255.255.255.255	11.17.1.2	11.17.1.1	100	OSPF
11.17.1.2	255.255.255.255	11.17.1.2	11.17.1.1	100	OSPF
11.4.1.2	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF
11.5.1.1	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF
11.8.1.2	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF

a)

RT_6_Table

RT_6 (b)

Network Destination	Netmask/Prefix len	Gateway	Interface	Metrics	Type
11.9.1.1	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF
11.24.1.2	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF
11.25.1.1	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF
11.1.1.2	255.255.255.255	11.7.1.2	11.7.1.1	200	OSPF
11.2.1.2	255.255.255.255	11.7.1.2	11.7.1.1	200	OSPF
11.3.1.1	255.255.255.255	11.7.1.2	11.7.1.1	200	OSPF
11.4.1.1	255.255.255.255	11.7.1.2	11.7.1.1	200	OSPF
11.9.1.2	255.255.255.255	11.17.1.2	11.17.1.1	200	OSPF
11.10.1.1	255.255.255.255	11.17.1.2	11.17.1.1	200	OSPF
11.15.1.2	255.255.255.255	11.17.1.2	11.17.1.1	200	OSPF
11.21.1.1	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF
11.22.1.2	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF
11.23.1.2	255.255.255.255	11.7.1.2	11.7.1.1	200	OSPF
11.14.1.2	255.255.255.255	11.17.1.2	11.17.1.1	200	OSPF
11.29.1.1	255.255.255.255	11.17.1.2	11.17.1.1	200	OSPF

b)

RT_6_Table

RT_6 (c)

Network Destination	Netmask/Prefix len	Gateway	Interface	Metrics	Type
11.20.1.1	255.255.255.255	11.6.1.1	11.6.1.2	200	OSPF
11.31.0.0	255.255.0.0	on-link	11.31.1.1	300	LOCAL
11.17.0.0	255.255.0.0	on-link	11.17.1.1	300	LOCAL
11.7.0.0	255.255.0.0	on-link	11.7.1.1	300	LOCAL
11.6.0.0	255.255.0.0	on-link	11.6.1.2	300	LOCAL
11.1.1.1	255.255.255.255	11.7.1.2	11.7.1.1	300	OSPF
11.18.1.2	255.255.255.255	11.7.1.2	11.7.1.1	300	OSPF
11.2.1.1	255.255.255.255	11.7.1.2	11.7.1.1	300	OSPF
11.19.1.2	255.255.255.255	11.7.1.2	11.7.1.1	300	OSPF
11.10.1.2	255.255.255.255	11.17.1.2	11.17.1.1	300	OSPF
11.26.1.1	255.255.255.255	11.17.1.2	11.17.1.1	300	OSPF
11.13.1.2	255.255.255.255	11.17.1.2	11.17.1.1	300	OSPF
11.14.1.1	255.255.255.255	11.17.1.2	11.17.1.1	300	OSPF
11.24.1.1	255.255.255.255	11.6.1.1	11.6.1.2	300	OSPF
11.25.1.2	255.255.255.255	11.6.1.1	11.6.1.2	300	OSPF

c)

RT_6_Table

RT_6 (d)

Network Destination	Netmask/Prefix len	Gateway	Interface	Metrics	Type
224.0.0.1	255.255.255.255	on-link	11.6.1.2 1...	306	MUL...
224.0.0.0	240.0.0.0	on-link	11.6.1.2 1...	306	MUL...
11.11.1.1	255.255.255.255	11.17.1.2	11.17.1.1	400	OSPF
11.12.1.1	255.255.255.255	11.17.1.2	11.17.1.1	400	OSPF
11.13.1.1	255.255.255.255	11.17.1.2	11.17.1.1	400	OSPF
11.18.1.1	255.255.255.255	11.7.1.2	11.7.1.1	400	OSPF
11.19.1.1	255.255.255.255	11.7.1.2	11.7.1.1	400	OSPF
11.26.1.2	255.255.255.255	11.17.1.2	11.17.1.1	400	OSPF
11.11.1.2	255.255.255.255	11.17.1.2	11.17.1.1	500	OSPF
11.28.1.1	255.255.255.255	11.17.1.2	11.17.1.1	500	OSPF
11.12.1.2	255.255.255.255	11.17.1.2	11.17.1.1	500	OSPF
11.27.1.1	255.255.255.255	11.17.1.2	11.17.1.1	500	OSPF
11.27.1.2	255.255.255.255	11.17.1.2	11.17.1.1	600	OSPF
11.28.1.2	255.255.255.255	11.17.1.2	11.17.1.1	600	OSPF
255.255.255.255	255.255.255.255	on-link	11.31.1.1	999	BRO...

d)

Figure 1-7: IP Forwarding Tables in Result Dashboard for RT 6

In this network, Router6 has 3 interfaces with IP's 11.7.1.1, 11.6.1.2 and 11.17.1.1 and its network addresses are 11.7.0.0, 11.6.0.0 and 11.17.0.0 since its network mask is 255.255.0.0

From the above screenshot, the router forwards packets intended to the subnet:

- 11.1.1.2, 11.2.1.2, 11.3.1.2, 11.3.1.1, 11.4.1.1 via interface 11.7.1.1 with cost 7 (6+1)
- Similarly 11.23.1.1, 11.4.1.2, 11.1.1.1, 11.2.1.1, 11.23.1.2 via interface 11.7.1.1 with cost 8 (6+1+1)
- 11.15.1.1, 11.9.1.2, 11.10.1.1, 11.15.1.2 via interface 11.17.1.1 with cost 8 (7+1)
- 11.9.1.1, 11.10.1.2 via interface 11.17.1.1 with cost 9 (7+1+1)
- 11.29.1.2, 11.29.1.1 and 11.14.1.2 via interface 11.17.1.1 with cost 10 (7+3)
- 11.24.1.2, 11.24.1.1 via interface 11.17.1.1 with cost 10 (7+1+2)

- 11.18.1.2, 11.18.1.1, 11.19.1.2 and 11.19.1.1 via interface 11.7.1.1 with cost 10 (6+1+3)
- 11.13.1.2, 11.11.1.1, 11.12.1.1, and 11.13.1.1 via interface 11.17.1.1 with cost 11 (7+3+1)
- 11.8.1.1 via interface 11.6.1.2 with cost 12 (6+6)
- 11.11.1.2 and 11.12.1.2 via interface 11.17.1.1 with cost 12 (7+3+1+1)
- 11.17.1.2 via interface 11.17.1.1 with cost 12 (7+5)
- 11.26.1.1 and 11.26.1.2 via interface 11.17.1.1 with cost 12 (7+1+4)
- 11.14.1.1 via interface 11.17.1.1 with cost 12 (7+3+2)
- 11.6.1.1 via interface 11.6.1.2 with cost 13 (7+6)
- 11.27.1.1 and 11.27.1.2 via interface 11.17.1.1 with cost 13 (7+3+1+2)
- 11.7.1.2 via interface 11.7.1.1 with cost 14 (8+6)
- 11.5.1.2 via interface 11.6.1.2 with cost 14 (6+8)
- 11.20.1.2, 11.20.1.1, 11.21.1.1, 11.21.1.2, 11.22.1.1, 11.22.1.2 via interface 11.6.1.2 with cost 14 (8+6)
- 11.28.1.2 via interface 11.17.1.1 with cost 14 (7+1+6)
- 11.28.1.1 via interface 11.17.1.1 with cost 14 (7+3+1+3)
- 11.25.1.1, 11.25.1.2 via interface 11.17.1.1 with cost 17 (7+1+9)
- 11.5.1.1 via interface 11.7.1.1 with cost 15 (6+1+8)

We are thus able to simulate the exact example as provided in the RFC and report that SPF Tree obtained, and the routing costs match the analysis provided in the RFC.