Open Shortest Path First (OSPF): A Routing Protocol Based on the Link-State Algorithm

Group 4

due November 14^{th}, 2019

1 Explain why the **Areas** and **Balanced** scenarios result in different routes than those observed in the **No_Areas** scenario, for the same pair of routers.

The routes created by the OSPF link-state algorithm are created by each router gathering information on all network link costs through broadcasted Link State Advertisements, then running Dijkstra's algorithm to calculate the minimum-cost distance between every pair of routers. In the **No_Areas** scenario there is only a backbone area (0.0.0.0), so the minimum-cost from **Router A** to **Router C** is the minimum cost over the entire network, which has the path *Router D* → *Router E* → *Router C*, with a cost of 15, as shown in Figure 1. In the **Areas** scenario, the autonomous system is configured into hierarchical areas, with **Router A**, **Router B** and **Router C** in area 0.0.0.1, while **Router D** and **Router E** are still in the backbone area 0.0.0.0. When hierarchical areas are established, each router in an area broadcasts it's own link-state to all other routers in that area, and routes are calculated from there. If the destination router were in a different area, the router would route the message to the nearest Area Boundary Router (ABR), then the backbone area would've helped route the message to the destination router, however since **Router A** and **Router C** are within the same area, the shortest path is shown to be a direct path to *Router C*, as seen in Figure 2.

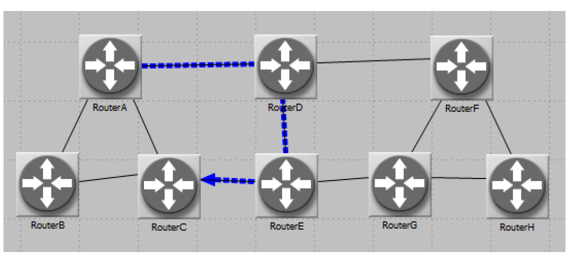

Figure 1: **No_Areas** path from A to C

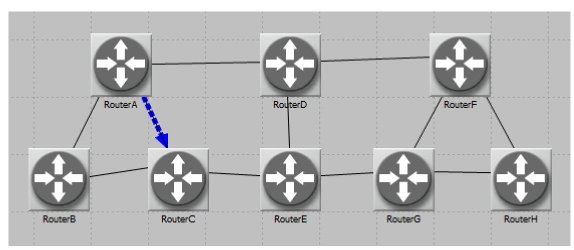

Figure 2: **Areas** path from A to C

An issue arises in network routing when the minimum-cost path to a router is discovered and utilized heavily, overloading the minimum-cost path and ignoring other paths that are either the same or negligibly different cost to the minimum. Load balancing avoids this problem, by allowing multiple minimum-cost (or low-cost) paths to be used between routers, thus to balance the load along each path. This was shown in the Riverbed simulation along the path from **Router B** to **Router H** in the **Balanced** scenario (Figure 4), when packets could be sent along the route *Router A* → *Router D* → *Router G* → *Router H*, or along the route *Router C* → *Router E* → *Router F* → *Router H*, as both paths share a total cost of 40. This is in oppisition to the **No_Areas** scenario which chose a single minimum-cost path for all messages to be sent (Figure 3).

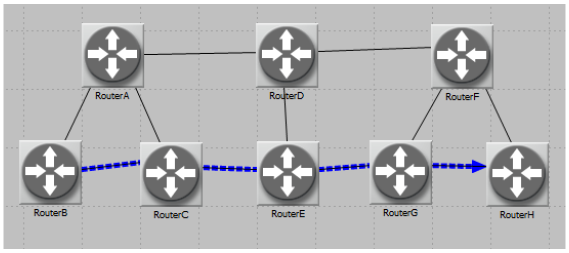

Figure 3: **No_Areas** path from B to H

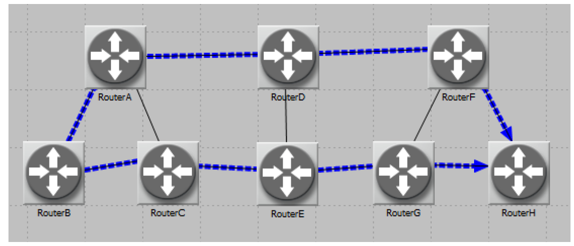

Figure 4: **Balanced** path from B to H

2 Using the simulation log, examine the generated routing table in **RouterA** for each of the three scenarios. Explain the values assigned to the *Metric* column of each route.

The figure below shows a diagram of the system, with red labels indicating the loopback interface router ID set by the simulator. Blue IP addresses indicate interface ports leaving each router, and are not important for answering these questions. Figure 5 will make it easier to describe the contents in the three routing tables below.

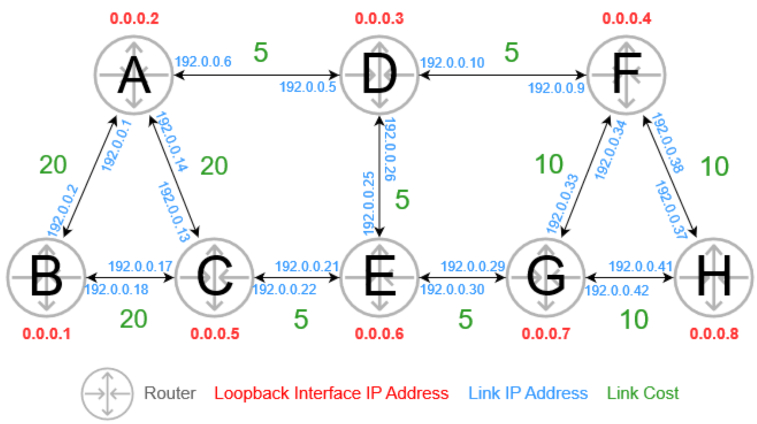

Figure 5: Diagram of the OSPF Simulated System for Questions 1, 2, and 3

In the routing tables I removed two columns to make them fit in this format, the Process Tag and an empty column.

Starting with the **No_Areas** scenario, the routing table for **Router A** at the end of the 10 minute simulation is in the table below. The rows which show the destination to be one of the 8 Loopback interface router ID's indicate the path determined by the OSPF algorithm for **Router A** to reach all other nodes. We can notice that there is only intra-area routing, since there is only the backbone area (0.0.0.0) in this scenario. The *Metric* column denotes the total cost along the minimum-cost path to reach each destination. This is calculated by **Router A** by running Dijkstra's algorithm once it has accumulated all Link State Advertisements from other nodes, giving it a bird's-eye view of the network topology. Every value in the metric column for Loopback Interface router destinations is equal to the minimum cost plus 1. This added cost of 1 seems to be associated with the overhead of sending a packet to an OSPF Loopback interface. This is most apparent for the second entry (highlighted below), when both the sender and destination are **Router A**. The outgoing interface is LB0, a Loopback interface that has a cost of 1 in an operation that travels on 0 edges.

Destination	Metric	Next Hop Address	Next Hop Node	Outgoing Interface	Area ID	Route Type
0.0.0.1/32	21	192.0.0.2	B	IF0	0.0.0.0	Intra-Area Network
0.0.0.2/32	1	0.0.0.2	A	LB0	0.0.0.0	Intra-Area Network
0.0.0.3/32	6	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.4/32	11	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.5/32	16	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.6/32	11	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.7/32	16	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.8/32	21	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.0/30	20	192.0.0.0	A	IF0	0.0.0.0	Intra-Area Network
192.0.0.4/30	5	192.0.0.4	A	IF2	0.0.0.0	Intra-Area Network
192.0.0.8/30	10	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.12/30	20	192.0.0.12	A	IF1	0.0.0.0	Intra-Area Network
192.0.0.16/30	35	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.20/30	15	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.24/30	10	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.28/30	15	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.32/30	20	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.36/30	20	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.40/30	25	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network

Router A No_Areas Routing Table

Next, the **Areas** scenario **Router A** routing table is shown, for which the simulation had hierarchical autonomous system areas assigned. Again only looking at entries associated with the Loopback Interface IP addresses of routers, the main difference is that there are more than one entry for many destinations. This is attributed to the different routing types introduced by the hierarchical structure. For example, looking at the possible routes to go from **Router A** to **Router C** has three possible options (all three highlighed below). The first is by Intra-Area Area Border Router (ABR) routing, which sends the packet to *Router D* → *Router E* → *Router C* with a cost of 15. The ABR is **Router A** in this case. The second is by Intra-Area Network routing, which sends it directly to **Router C** with a cost of 21. The last again uses Intra-Area ABR routing sending the packet directly to **Router C** with a cost of 20, the ABR being **Router C** in this case. Sending to an ABR does not invoke the Loopback interface cost of 1, which I speculate to be because the ABR is associated with a path to leave the area, not necessarily a destination.

Destination	Metric	Next Hop Address	Next Hop Node	Outgoing Interface	Area ID	Route Type
0.0.0.1/32	21	192.0.0.2	B	IF0	0.0.0.1	Intra-Area Network
0.0.0.2/32	1	0.0.0.2	A	LB0	0.0.0.1	Intra-Area Network
0.0.0.3/32	6	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.4/32	11	192.0.0.5	D	IF2	0.0.0.0	Inter-Area Network
0.0.0.4/32	10	192.0.0.5	D	IF2	0.0.0.0	Intra-Area ABR
0.0.0.5/32	15	192.0.0.5	D	IF2	0.0.0.0	Intra-Area ABR
0.0.0.5/32	21	192.0.0.13	C	IF1	0.0.0.1	Intra-Area Network
0.0.0.5/32	20	192.0.0.13	C	IF1	0.0.0.1	Intra-Area ABR
0.0.0.6/32	11	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.7/32	16	192.0.0.5	D	IF2	0.0.0.0	Inter-Area Network
0.0.0.7/32	15	192.0.0.5	D	IF2	0.0.0.0	Intra-Area ABR
0.0.0.8/32	21	192.0.0.5	D	IF2	0.0.0.0	Inter-Area Network
192.0.0.0/30	20	192.0.0.0	A	IF0	0.0.0.1	Intra-Area Network
192.0.0.4/30	5	192.0.0.4	A	IF2	0.0.0.0	Intra-Area Network
192.0.0.8/30	10	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.12/30	20	192.0.0.12	A	IF1	0.0.0.1	Intra-Area Network
192.0.0.16/30	40	192.0.0.13	C	IF1	0.0.0.1	Intra-Area Network
		192.0.0.2	B	IF0	0.0.0.1	Intra-Area Network
192.0.0.20/30	15	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.24/30	10	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.28/30	15	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.32/30	20	192.0.0.5	D	IF2	0.0.0.0	Inter-Area Network
192.0.0.36/30	20	192.0.0.5	D	IF2	0.0.0.0	Inter-Area Network
192.0.0.40/30	25	192.0.0.5	D	IF2	0.0.0.0	Inter-Area Network

Router A Areas Routing Table

Lastly, the **Balanced** scenario **Router A** routing table is shown. This scenario had load balancing running over the backbone area. It's routing table looks identical to the routing table in the **No_Areas** scenario, because the routing table only holds minimum-cost information. However, **Router A** still maintains a Link State Database, and if the current path becomes slow due to high usage it can rerun the Link-state algorithm (Dijkstra's) to find another low-cost path.

Destination	Metric	Next Hop Address	Next Hop Node	Outgoing Interface	Area ID	Route Type
0.0.0.1/32	21	192.0.0.2	B	IF0	0.0.0.0	Intra-Area Network
0.0.0.2/32	1	0.0.0.2	A	LB0	0.0.0.0	Intra-Area Network
0.0.0.3/32	6	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.4/32	11	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.5/32	16	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.6/32	11	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.7/32	16	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
0.0.0.8/32	21	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.0/30	20	192.0.0.0	A	IF0	0.0.0.0	Intra-Area Network
192.0.0.4/30	5	192.0.0.4	A	IF2	0.0.0.0	Intra-Area Network
192.0.0.8/30	10	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.12/30	20	192.0.0.12	A	IF1	0.0.0.0	Intra-Area Network
192.0.0.16/30	35	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.20/30	15	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.24/30	10	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.28/30	15	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.32/30	20	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.36/30	20	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network
192.0.0.40/30	25	192.0.0.5	D	IF2	0.0.0.0	Intra-Area Network

Router A Balanced Routing Table

3 Modeler allows you to examine the link-state database that is used by each router to build the directed graph of the network. Examine this database for **RouterA** in the **No_Areas** scenario. Show how **RouterA** utilizes this database to create a map for the topology of the network and draw this map (This is the map that will be used later by the router to create its routing table.)

The Link State database for the **No_Areas** scenario obtained by **Router A** is shown below. I deleted two columns (timestamp and age) to make the table fit on this page. They weren't needed for answering this question.

This database is filled by routers broadcasting Link-State Advertisements (LSA, the number of links) across the area, or in in the case of the **No_Areas** scenario, the entire network. It provides a network topology to **Router A** so that **Router A** has all necessary information for running the centralized Dijkstra's algorithm and determining its routing table. What we see below is an entry for each router that has sent out a LSA that has been routed through **Router A**. Each router has information regarding Stub Network links and Point to Point links. The Point to Point links are what we care most about, as it contains information regarding the cost and existance of a path from one router to another. Stub networks are included in OSPF because technically there don't have to be IP addresses assigned to the routers themselves (such as before our instructions for this problem involved setting loopback interface IP addresses). In the case of the router not having a designated address, the stub network link would be used. This comes down to us not needing to worry about the stub network links. Picking one to look at further, we can see that for **Router G** (0.0.0.7, highlighted below) there are 3 Point to Point link connections. The first is **Router F** (0.0.0.4), **Router H** (0.0.0.8), and **Router E** (0.0.0.6). These have associated costs 5, 10, and 10.

Process Tag (Area ID)	Link State ID	Adv Router ID	Sequence Number	Link Type	Link ID	Link Data	Link Cost
1 (0.0.0.0)	0.0.0.4	0.0.0.4	111	Stub Network	0.0.0.4	255.255.255.255	1
				Point-To-Point	0.0.0.7	192.0.0.34	10
				Stub Network	192.0.0.32	255.255.255.252	10
				Point-To-Point	0.0.0.8	192.0.0.38	10
				Stub Network	192.0.0.36	255.255.255.252	10
				Point-To-Point	0.0.0.3	192.0.0.9	5
				Stub Network	192.0.0.8	255.255.255.252	5
	0.0.0.5	0.0.0.5	112	Stub Network	0.0.0.5	255.255.255.255	1
				Point-To-Point	0.0.0.1	192.0.0.17	20
				Stub Network	192.0.0.16	255.255.255.252	20
				Point-To-Point	0.0.0.2	192.0.0.13	20
				Stub Network	192.0.0.12	255.255.255.252	20
				Point-To-Point	0.0.0.6	192.0.0.22	5
				Stub Network	192.0.0.20	255.255.255.252	5
	0.0.0.2	0.0.0.2	132	Stub Network	0.0.0.2	255.255.255.255	1
				Point-To-Point	0.0.0.1	192.0.0.1	20
				Stub Network	192.0.0.0	255.255.255.252	20
				Point-To-Point	0.0.0.5	192.0.0.14	20
				Stub Network	192.0.0.12	255.255.255.252	20
				Point-To-Point	0.0.0.3	192.0.0.6	5
				Stub Network	192.0.0.4	255.255.255.252	5
	0.0.0.1	0.0.0.1	133	Stub Network	0.0.0.1	255.255.255.255	1
				Point-To-Point	0.0.0.5	192.0.0.18	20
				Stub Network	192.0.0.16	255.255.255.252	20
				Point-To-Point	0.0.0.2	192.0.0.2	20
				Stub Network	192.0.0.0	255.255.255.252	20
	0.0.0.3	0.0.0.3	153	Stub Network	0.0.0.3	255.255.255.255	1
				Point-To-Point	0.0.0.4	192.0.0.10	5
				Stub Network	192.0.0.8	255.255.255.252	5
				Point-To-Point	0.0.0.2	192.0.0.5	5
				Stub Network	192.0.0.4	255.255.255.252	5
				Point-To-Point	0.0.0.6	192.0.0.26	5
				Stub Network	192.0.0.24	255.255.255.252	5
	0.0.0.6	0.0.0.6	192	Stub Network	0.0.0.6	255.255.255.255	1
				Point-To-Point	0.0.0.3	192.0.0.25	5
				Stub Network	192.0.0.24	255.255.255.252	5
				Point-To-Point	0.0.0.5	192.0.0.21	5
				Stub Network	192.0.0.20	255.255.255.252	5
				Point-To-Point	0.0.0.7	192.0.0.30	5
				Stub Network	192.0.0.28	255.255.255.252	5
	0.0.0.8	0.0.0.8	212	Stub Network	0.0.0.8	255.255.255.255	1
				Point-To-Point	0.0.0.7	192.0.0.41	10
				Stub Network	192.0.0.40	255.255.255.252	10
				Point-To-Point	0.0.0.4	192.0.0.37	10
				Stub Network	192.0.0.36	255.255.255.252	10
	0.0.0.7	0.0.0.7	213	Stub Network	0.0.0.7	255.255.255.255	1
				Point-To-Point	0.0.0.4	192.0.0.33	10
				Stub Network	192.0.0.32	255.255.255.252	10
				Point-To-Point	0.0.0.8	192.0.0.42	10
				Stub Network	192.0.0.40	255.255.255.252	10
				Point-To-Point	0.0.0.6	192.0.0.29	5
				Stub Network	192.0.0.28	255.255.255.252	5

Router A No_Areas Link State Database

We could draw each component with it's associated links to other routers, then compile them into one picture showing the network topology. This is already shown in Figure 5, with all node Loopback interface IP addresses and link costs.

4 Create another scenario as a duplicate of the **No_Areas** scenario. Name the new scenario **Q4_No_Areas_Failure**. In this new scenario simulate a failure of the link connecting **RouterD** and **RotuerE**. Have this failure start after 100 seconds. Rerun the simulation. Show how that link failure affects the content of the linkstate database and routing table of **RouterA**. (You will need to disable the global attribute **OSPF Sim Efficiency**. This will allow OSPF to update the routing table if there is any change in the network.)

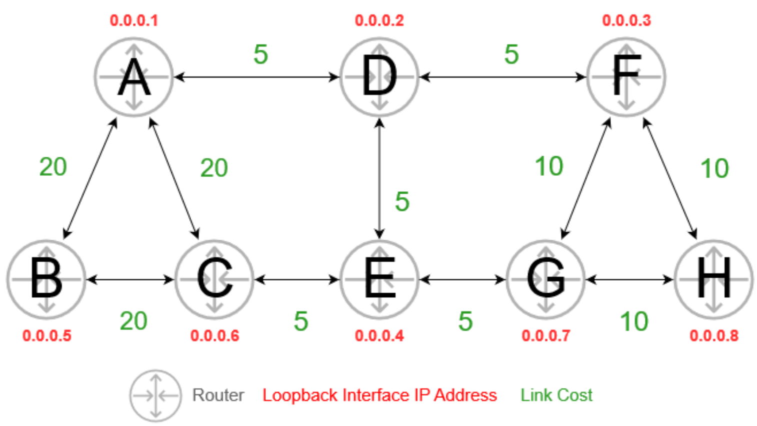

Figure 6: Diagram of the OSPF Simulated System for Q4 and Q5

Following the instructions, we created a Failure scenario using the Failure Recovery object palette. The object was configured to fail the link between Router D and Router E after 100 seconds of the start of the simulation. According to the previous **No_Area** setup which we duplicated, for OSPF routing protocol the profitable path of sending the packets from A to C was A-D-E-C as shown in Figure 1. But according to our current scenario, the path is going to get updated with the shortest path as edge A-C in the network. This is shown below by comparing the link-state database and routing table of **RouterA** for both **No_Areas** and **Q4_No_Areas_Failure** scenario.

The IP addresses are assigned differently for different routers in the two scenarios as they were created by different people. For reference please refer to Figure 5 for IP addresses of **No_Area** scenario and Figure 6 for IP addresses of **Q4_No_Areas_Failure**.

4.0.1 Linkstate database

From the **Router A No_Areas** Link State Database table as shown in Question 3 can see that **Router D** (0.0.0.3) and **Router E** (0.0.0.6) are connected as they have an entry for each others Link ID listed as a Point to Point network connection. But for the **Router A Q4_No_Areas_Failure** Link State Database table shown below we can see that they are not connected as there was no entry for a connection to **Router D** (0.0.0.2) in **Router E**'s (0.0.0.4) Link State Advertisement, and vice versa (highlighed below). This is due to the link failure.

Process Tag (Area ID)	Link State ID	Adv Router ID	Sequence Number	Link Type	Link ID	Link Data	Link Cost
1 (0.0.0.0)	0.0.0.5	0.0.0.5	115	Stub Network	0.0.0.5	255.255.255.255	1
				Point-To-Point	0.0.0.6	192.0.0.26	20
				Stub Network	192.0.0.24	255.255.255.252	20
				Point-To-Point	0.0.0.1	192.0.0.13	20
				Stub Network	192.0.0.12	255.255.255.252	20
	0.0.0.1	0.0.0.1	135	Stub Network	0.0.0.1	255.255.255.255	1
				Point-To-Point	0.0.0.2	192.0.0.2	5
				Stub Network	192.0.0.0	255.255.255.252	5
				Point-To-Point	0.0.0.5	192.0.0.14	20
				Stub Network	192.0.0.12	255.255.255.252	20
				Point-To-Point	0.0.0.6	192.0.0.18	20
				Stub Network	192.0.0.16	255.255.255.252	20
	0.0.0.6	0.0.0.6	156	Stub Network	0.0.0.6	255.255.255.255	1
				Point-To-Point	0.0.0.5	192.0.0.25	20
				Stub Network	192.0.0.24	255.255.255.252	20
				Point-To-Point	0.0.0.4	192.0.0.21	5
				Stub Network	192.0.0.20	255.255.255.252	5
				Point-To-Point	0.0.0.1	192.0.0.17	20
				Stub Network	192.0.0.16	255.255.255.252	20
	0.0.0.7	0.0.0.7	176	Stub Network	0.0.0.7	255.255.255.255	1
				Point-To-Point	0.0.0.4	192.0.0.33	5
				Stub Network	192.0.0.32	255.255.255.252	5
				Point-To-Point	0.0.0.8	192.0.0.42	10
				Stub Network	192.0.0.40	255.255.255.252	10
				Point-To-Point	0.0.0.3	192.0.0.29	10
				Stub Network	192.0.0.28	255.255.255.252	10
	0.0.0.3	0.0.0.3	215	Stub Network	0.0.0.3	255.255.255.255	1
				Point-To-Point	0.0.0.2	192.0.0.5	5
				Stub Network	192.0.0.4	255.255.255.252	5
				Point-To-Point	0.0.0.7	192.0.0.30	10
				Stub Network	192.0.0.28	255.255.255.252	10
				Point-To-Point	0.0.0.8	192.0.0.38	10
				Stub Network	192.0.0.36	255.255.255.252	10
	0.0.0.8	0.0.0.8	216	Stub Network	0.0.0.8	255.255.255.255	1
				Point-To-Point	0.0.0.7	192.0.0.41	10
				Stub Network	192.0.0.40	255.255.255.252	10
				Point-To-Point	0.0.0.3	192.0.0.37	10
				Stub Network	192.0.0.36	255.255.255.252	10
	0.0.0.2	0.0.0.2	369	Stub Network	0.0.0.2	255.255.255.255	1
				Point-To-Point	0.0.0.1	192.0.0.1	5
				Stub Network	192.0.0.0	255.255.255.252	5
				Point-To-Point	0.0.0.3	192.0.0.6	5
				Stub Network	192.0.0.4	255.255.255.252	5
	0.0.0.4	0.0.0.4	389	Stub Network	0.0.0.4	255.255.255.255	1
				Point-To-Point	0.0.0.6	192.0.0.22	5
				Stub Network	192.0.0.20	255.255.255.252	5
				Point-To-Point	0.0.0.7	192.0.0.34	5
				Stub Network	192.0.0.32	255.255.255.252	5

Router A Q4_No_Areas_Failure Link State Database

4.0.2 Router Table

Here we are going to compare the cost of the path taken by the packet which is sent from **Router A** to **Router C** using **Router A** Routing table for **No_Areas** and **Q4_No_Areas_Failure** scenarios. From the **Router A No_Areas** Routing Table seen in Question 2 we can see the minimal cost for sending a packet from **Router A** to **Router C**(0.0.0.5) is 16. This is from the addition of the cost from A to D, D to E and E to C which is minimal. An addition of cost 1 is there due to the Loopback mechanism of the links. Contrary to this cost the Routing table of **Router A** is updated in **Q4_No_Areas_Failure** with a new cost which is larger than the previous scenario cost and travels directly from A to C. This is because the link between D and E failed, so the same minimum-cost path as before is no longer an option. Therefore, OSPF calculated a new minimum-cost path, which is a direct link from A to C with a cost of 21. As shown below in the **Router A Q4_No_Areas_Failure** Routing Table we can see that the cost now from **Router A**(0.0.0.1) to **Router C**(0.0.0.6) is 21 (highlighted below). Also we can notice the *Next Hop Node* from A to C which was previously **Router D** is updated to **Router C**.

Destination	Metric	Next Hop Address	Next Hop Node	Outgoing Interface	Area ID	Route Type
0.0.0.1/32	1	0.0.0.1	A	LB0	0.0.0.0	Intra-Area Network
0.0.0.2/32	6	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
0.0.0.3/32	11	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
0.0.0.4/32	26	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
		192.0.0.17	C	IF2	0.0.0.0	Intra-Area Network
0.0.0.5/32	21	192.0.0.13	B	IF1	0.0.0.0	Intra-Area Network
0.0.0.6/32	21	192.0.0.17	C	IF2	0.0.0.0	Intra-Area Network
0.0.0.7/32	21	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
0.0.0.8/32	21	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
192.0.0.0/30	5	192.0.0.0	A	IF0	0.0.0.0	Intra-Area Network
192.0.0.4/30	10	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
192.0.0.12/30	20	192.0.0.12	A	IF1	0.0.0.0	Intra-Area Network
192.0.0.16/30	20	192.0.0.16	A	IF2	0.0.0.0	Intra-Area Network
192.0.0.20/30	25	192.0.0.17	C	IF2	0.0.0.0	Intra-Area Network
192.0.0.24/30	40	192.0.0.17	C	IF2	0.0.0.0	Intra-Area Network
		192.0.0.13	B	IF1	0.0.0.0	Intra-Area Network
192.0.0.28/30	20	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
192.0.0.32/30	25	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
192.0.0.36/30	20	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network
192.0.0.40/30	30	192.0.0.1	D	IF0	0.0.0.0	Intra-Area Network

5 For both **No_Areas** and **Q4_No_Areas_Failure** scenario, collect the **Traffic Sent (bits/sec)** statistic (one of the **Global Statistics** under **OSPF**). Rerun the simulation for those two scenarios and obtain the graph that compares the OSPF's **Traffic Sent (bits/sec)** in both scenarios. Comment on the obtained graph.

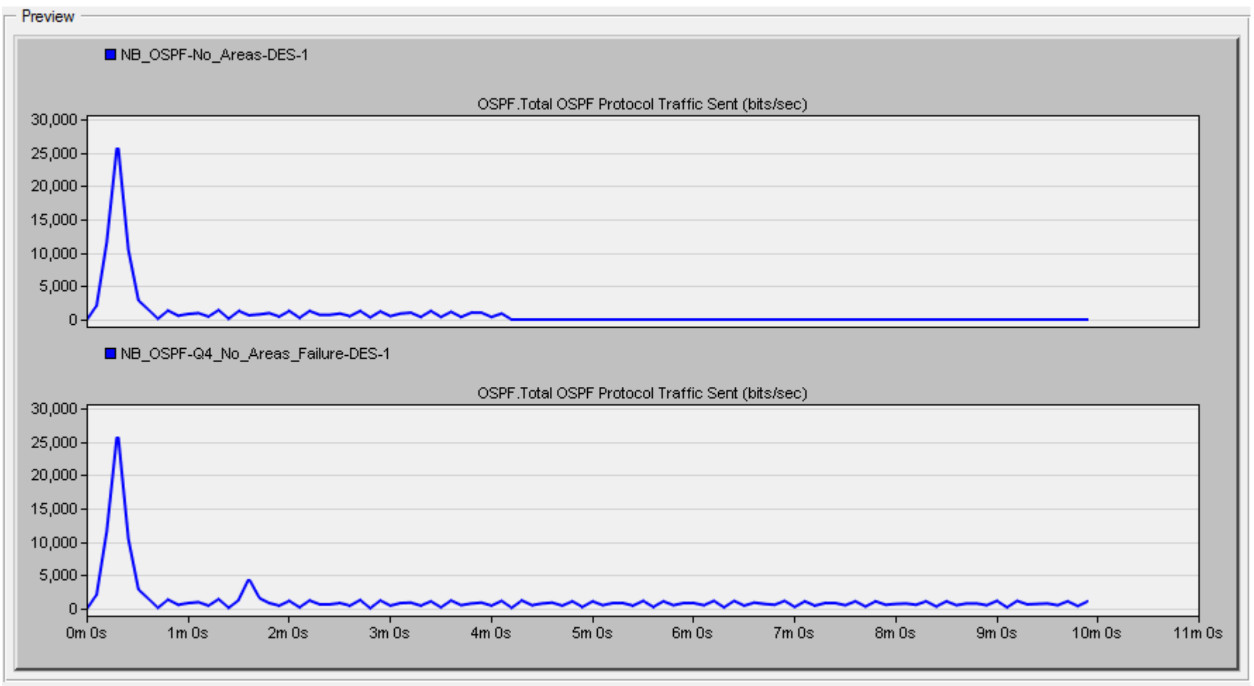

Figure 7: OSPF Simulation graph for No_Areas and Q4_No_Areas_Failure

In Figure 7 shown above we can see the graph comparison between the simulation of **No_Areas** (top) and **Q4_No_Areas_Failure** (bottom) for 10 min duration. We can see a spike in the number of packets transferred at 100 seconds after the simulation started for **Q4_No_Areas_Failure**. This is because we configured a failure node after 100 seconds from **Router D** to **Router E** which was on the shortest path for the communication between **Router A** and **Router C**. Due to this failure, the nodes it connected (D and E) had to resend Link State Advertisements across the entire network, so all routers could update their linkstate databases and rerun Dijkstra's algorithm to compute updated routing tables.

For the first graph (**No_Areas**) there is a peak of around 26,000 bits per second used, which accounts for roughly 3,200 bits per second for all 8 routers to run OSPF protocol and determine their linkstate database and routing tables. After 4 minutes the bits per second sent go to zero because of a parameter set under OSPF Simulation Efficiency that stops sending bits at 260 seconds. This way the network under OSPF assumes to have no failure and errors and successfully stops to configure its tables. In the second graph (**Q4_No_Areas_Failure**) the network's global attribute OSPF Simulation Efficiency has been disabled (as configured in the previous question), allowing the OSPF to updated the tables if there is any change in the network. That's why OSPF doesn't halt using the network. This seems correct as in the advent of any failure the protocol should be ready to update the tables.

As an experiment, we once again enabled the simulation efficiency and triggered the OSPF stop time after 400 seconds and the protocol stopped to use the network after the **Q4_No_Areas_Failure** scenario.