the data volume and the aim the visualization should have on its viewers.

1.5.1 Icons

To represent one-dimensional data, simple icons can be used. They combine one single information with one object. This data is consistent over time.

1.5.2 Graphs and diagrams

To visualize two or three dimensional data mostly graphs are used. Graphs are best in showing the change of a value within time. There are different possibilities to convert actual values into graphical objects.
General types of graphs are columns, pictorial representations, pies, globes or simple lines.

2. Past Study

Shortest Path Bridging (SPB) has been developed to overcome the limitations of legacy Ethernet protocols while allowing the application of advanced networking concepts on the Ethernet. As the Ethernet makes expansion of its application from LAN to provider networks, SPB also is designed to operate in large scale networks such as providers' networks, back-haul networks, and metro Ethernet. This chapter reviews literature connected to SPB. In starting, this episode defines the back ground of why SPB developed. The problems and limitations of legacy Ethernet protocols are reviewed. Legacy Ethernet protocols are the major Ethernet protocols currently deployed in real world networks. This chapter does not explain about problems of older Ethernet protocol urbanized in 90's. After that, this chapter examines the superior network techniques adopted by SPB. Finally, an open source network simulator is introduced.

2.1 Problems of legacy Ethernet protocols

Shortest Path Bridging (SPB) addresses the problems of legacy Ethernet network. There are problems in both control plane and data plane. In control plane, minimizing convergence time after topology changes is a major challenge. In data plane, inefficiency of link utilization is the main problem.

Switches in an Ethernet network share the same physical medium. The routing system of an Ethernet network was still based on the flood-and-learning mechanism. Massive amounts of frames flooding in the network may cause broadcast storms, effectively melting in the network. To avoid transmit storms while permitting such a flood-and-learning mechanism; the Spanning Tree Protocol (STP) was designed [5].

Redundant paths in an STP network are disabled to suppress loop formation. A spanning tree topology is activated in the network and the topology appears as replica of the shared medium. The STP locates each end point on a spanning tree hence is a single point of failure. This simple connectivity is not on the optimal path between two ends. To assurance the round free status at all times, any topology change shuts down all connectivity on the spanning tree until the new tree has converged [7]. Though this shutdown period is no longer than a few seconds, it is still not acceptable.
The main problem of STP's data plane is inefficient use of links in a network.

First, to avoid loop formation in STP region, some of the links in a network must be disengaged. An additional difficulty is that traffic within an STP region does not follow the optimal path recurrently. Seeing as an STP region is a replica of shared medium, it does not provide a sophisticate routing mechanism. Figure 2.1 demonstrates the problems of STP data plane.

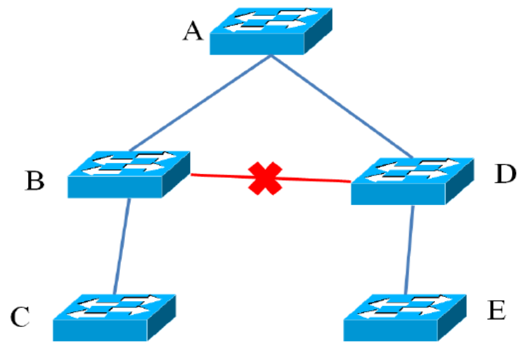

Figure 2: STP Problems

3. Problem Statement

In the real world, an SPB switch may have its own SPB link state database. A link state database in a SPB node is synchronized with other databases installed in other nodes unless there is a non-synchronization in these nodes. Therefore, in most of the cases link state databases in the same SPB domain are indistinguishable. Current simulation design does not consider miss-synchronization because the objective of our design is to offer a base SPB model for future extension. In contrast to a typical Internet node, SPB node's protocol handler communicates with Spb Net Device directly since that SPB is a layer 2 protocol. The following is the description of each module.

3.1.1 Spb Link State Data-Base

This is the hash table where all references to each Spb Interface are saved. Its key is a nodal B-MAC address which identifies a node. Therefore, a query with a nodal B-MAC returns the Spb Interface on the node. This is a singleton object which has only one instance in the simulator. It is a global database shared by every node in the simulator. In a real world situation, a SPB enabled switch has its own Link State Database (LSDB) and the LSDB is synchronized with other LSDBs installed in other switches.

3.1.2 Spb Interface

This is the interface to the control plane module for a node. It offers APIs for accessing Intermediate System to Intermediate System (IS-IS) sub-TLVs for SPB. By this module, we can set or get sub-TLV's values and manage the sub-TLVs. For example, we can allocate an I-SID on a node in order to register the node in the specific group represented by the I-SID.
In a real world situation each SPB switch (represented by a node in simulation) constructs IS-IS sub-TLVs and

exchanges the digests2 of them with neighboring switches. After interchanging the digests, a node in a SPB network can build LSDB.

3.1.3 Spb App

The model of a packet source and sink. It generates a packet and forwards it to Spb Protocol Handler to send a packet to other nodes. It also consumes the packets whose destination is matched by the node on which it is installed.

3.1.4 Spb Protocol Handler

This module extracts the SPB headers of received frames from Spb Net Device. It queries the forwarding (filtering) database (FDB) of the node to select the proper output ports.

3.1.5 Spb Net Device:

The software/hardware model of the network driver of SPB and the input/output interface of a frame to/from outside of the node. This is the same as output port of a bridge. It is inherited from NS-3 base class Net-Device. It encapsulates the packet received from layer above the current layer. When it receives a frame from the Spb Channel, it forwards the frame to Spb ProtocolHandler.

3.1.6 Spb Channel:

The model of a SPB link. It has two references of Spb NetDevice as endpoints of the link. The weight of the link is user configured.

3.1.7 Spb Forwarding DB

The model of a forwarding database of the SPB enabled through the bridge. When SpbProtocol-Handeller and SpbRouting objects query SpbForwardingDB, SpbForwardingdDB returns the instance of the table entry or the output port referencing the Spb NetDevice object.

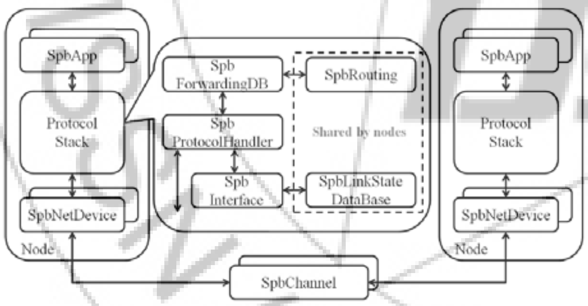

Figure 3: Architure of SPB

3.1.8 Spb Routing

This module is in charge of computing the path and populating the forwarding databases on nodes according to the result of the computation.

3.2 Actions of a SPB Node

The following is the brief illustration of how a node handles the received frame:

1) When a node receives a frame from outside, the frame arrives at Spb Net Device. A frame is a pointer value referencing the Packet object.
2) Spb Net Device calls the callback function registered at the node
3) The node looks for the correct handler for the packet type.
4) If it is a SPB packet, the node calls Spb Protocol Handler.
5) Spb Protocol Handler extracts SPB header from the received packet.
6) Spb Protocol Handler queries Spb Forwarding DB to know the output port number. Every Spb Net Device has corresponding output port number. If the output port number is '0', Spb Protocol Handler moves the packet to Spb App. Otherwise, move the packet to corresponding Spb Net Device.

When a node send a frame:

1) SpbApp generates a packet and forwards it to Spb Protocol Handler with I-SID and traffic type. There are two types of traffic, namely unicast and multicast.
2) Spb Protocol Handler queries Spb Forwarding DB and Spb Interface. First, Spb Protocol Handler queries Spb Interface to get BVID corresponding to I-SID received from Spb App. Second, it queries Spb Forwarding DB with BVID and Nodal MAC address of the node.
3) Spb Protocol Handler checks the out port referencing one of the Spb Net Device installed in the node. Spb Protocol Handler adds SPB header to the packet and moves it to Spb Net Device.
4) Spb Net Device corresponding to the output port sends the frame to connected Spb Channel.

4. Implementation

In general, the term control plane (In routing, the control plane is the part of the router architecture that is concerned with drawing the network map and building routing table. The system is refers to a part of the network architecture that collects the information of a network topology, and performs the routing calculations required to direct traffic. The information which is collected from the control plane is used to build a Forwarding Database (FDB).

A. Since Shortest Path Bridging (SPB) has evolved from the latest Ethernet technology the SPB control plane not only deals with the traditional LAN topology information, but also the service and virtual LAN identifiers. The important thing to consider is that an SPB switch encapsulates a customer frame into its frame when it receives the customer frame, and thus the SPB control plane is isolated from a customer network.

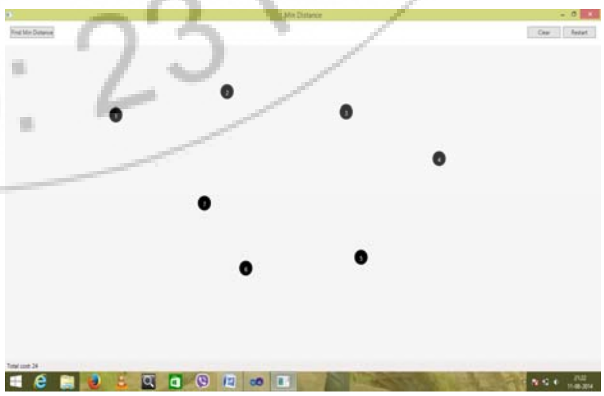

Figure 4: Implementation of Network usnig SPB

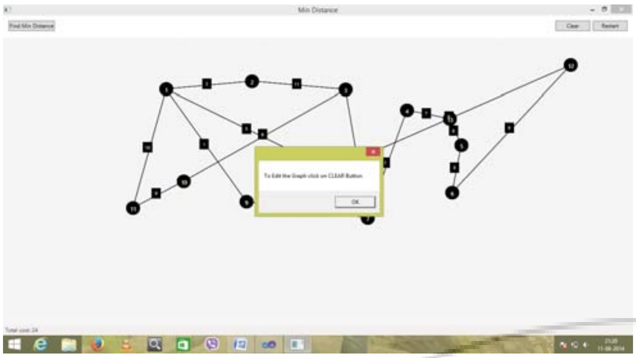

Figure 5: Implementation of IS-IS extension for SPB

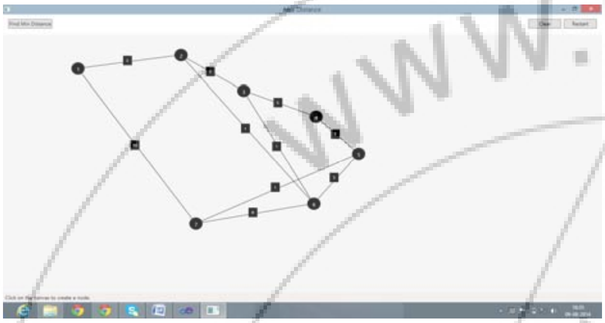

Figure 6: SpbInterface Class Diagram

5. Result

To calculate the paths, we use two n by n matrices. The matrix distance[n][n] contains the distances between nodes in the simulation. For example, if distance[i][j] is 10, the weight of the path from i to j is equal to 10. Another matrix predecessor[n][n] contains the predecessor to node j on a shortest path from i to j. In other words, predecessor is the intermediate node. A predecessor matrix value is a 64-bit Bridge Identifier which is the concatenation of the bridge priority and the bridge system id. A bridge system id is numerical form of a MAC address of a node. The algorithm is provided below:
Lines number 9 and 10 both create path from i to j. A Path is a list of 64-bit Bridge Identifiers. The difference in lines 9 and 10 is that line 10 creates the path passing through k. The 10th line joins two paths, one is from i to k and another one is from k to j. If path from i to k is on a shortest path and path k to j is, then path, i to j, is also on a shortest path. Computing a shortest path has optimal substructure. Hence we guarantee that path i to j passing through k also on a shortest path [19]. Line 11 compares two paths using ECT-Algorithm. There are 16 different ECT-Algorithm. The least most significant 1 byte of an ECT-Algorithm is used to XOR on each path.

Shortest Path Bridging (SPB) has the tie-breaking mechanism to prioritize the equal cost path. Each node of the path advertises the costs of the attached links. These costs are presented in SPB Link Metric sub-TLV. The addition of the link costs on the path is equal to the cost of the path. If cost of the paths is same between two end points, the path with smaller hop counts has the priority. If there exist more than two paths with the same link cost and hop counts, the tie-breaking mechanism by default picks up the path traversing the intermediate node with the lower Bridge Identifier [17]. Mesh network such as a data center may have multiple paths with the same link cost and the hope counts. This SPB tie-breaking mechanism guarantees diversity.

11^{th} Line compares two equal cost paths. A path with lower hop counts has the higher priority. As a path is a list of bridge identifiers, the number of elements in the list is equal to the hop counts including the source and the destination. Thus, the smaller size of this list, the path, has the higher priority. If two paths have the same hop counts then, XOR of ECT-Algorithm value with the paths is taken. The path with the smaller result has the higher priority. For example, we assume that path1 has sequence of bridge identifiers 0,1,4 and 5, path2 have sequence of 0, 2, 4 and 5. If path1 and path2 are XORed with 0xFF, path 2 will produce smaller number. In that case, path2 has higher priority hence it would be selected.

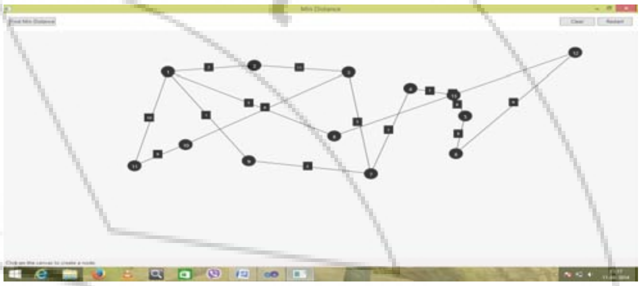

Figure 7: Build the Network

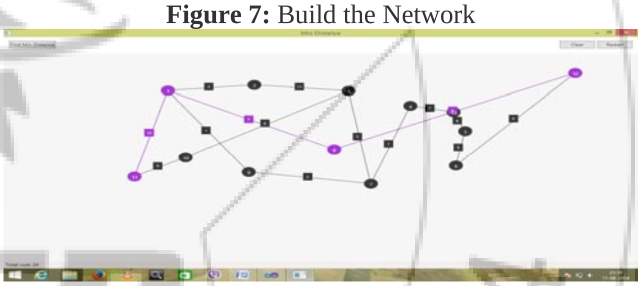

Figure 8: Find the Shortest Path

6. Conclusion

The cloud data center network requires certain properties. The first property is virtualization. The infrastructure of the cloud data center is shared with multiple customers and each customer requires different levels of services. Without network virtualization, it is hard to offer different classes of services. Individual customer traffic has to be identified in the cloud data center for delivering customized services. The second property is acceptable convergence time. The cloud data center network has to quickly response to network events. A cloud data center runs almost all type of application. Some of the application may need real time stream of data. In order to meet real time constraints, the convergence time after network changes should be reduced. The third property is higher utilization. Network traffic in a cloud data center should take diverse paths to achieve higher utilization. Fourth property is a minimum arrangement. Traffic in a cloud data center changes dynamically. Human intervention to the dynamicity may not be practical. Finally, cost of deploying network should be economical.

Ethernet meets all the requirements described above. It was developed to reduce a network arrangement. Its plug-and-play feature can migrate hosts to different locations without arrangement. Virtual machine instance in cloud data center may require frequent migrations. The plug-and-play mechanism would reduce the efforts following virtual machine migrations.

References

[1] N. B. David Allan, "IEEE 802.1aq in a Nutshell: Antecedents and Technology, Exploiting Multiple Paths in SPB," in *802.1aq shortest Path Bridging: Design and Evolution The Architect's Perspective*, New York, IEEE Press, 2012, p. 29.
[2] N. B. David Allan, "The Data Center and General Enterprise Application," in *802.1aq Shortest Path Bridging Design and Evolution*, IEEE Press, 2012, pp. 152-155.
[3] P. Draft Standard, *802.1 Q/D10, IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks, Copyright by the Institute of Electrical and Electronics Engineers*, Inc, 1997.
[4] IEEE Std 802.1ah, IEEE Standard for Local and metropolitan area networks -- Virtual Bridged Local Area Networks Amendment 7: Giver Backbone Bridges, IEEE, 2008.
[5] IEEE Standard, *802.1D IEEE Standard for Local and Metropolitan Area Networks. Media Access Control (MAC) Bridges*, IEEE, 2004.
[6] P. Draft Standard, *IEEE 802.1aq/D4.6 Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks-Amendment XX: Shortest Path Bridging*, IEEE, 2012.
[7] N. B. David Allan, "802.1aq Shortest Path Bridging Design and Evolution," IEEE Press, 2012, p. 3.
[8] N. B. David Allan, "Why the SPB Control Plane Looks as it does," in *802.1aq shortest Path Bridging: Design and Evolution The Architect's Perspective*, New York, IEEE Press, 2012, pp. 110-115.
[9] ISO/IEC, *Information technology-Telecommunications and information exchange between systems - Intermediate System to Intermediate System intra-domain routing information exchange protocol(ISO 8473)*, ISO/IEC, 2002.
[10] P. Ashwood-Smith, "Shortest Path Bridging IEEE 802.1aq Tutorial and Demo," in *NANOG 50*, 2010.
[11] N. B. David Allan, "Why SPB Looks as it does, History," in *802.1aq shortest Path Bridging: Design and Evolution The Architect's Perspective*, New York, IEEE Press, 2012, pp. 52-60.
[12] "ns-3," 2013. [Online]. Available: http://www.nsnam.org.
[13] "Nsnam," 5 november 2011. [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/Main_Page.
[14] M. L. F. R. D. J. K. Thomas R. Henderson, "Network simulations with the ns-3 simulator," in *SIGCOMM demonstration*, 2008.
[15] R. S. S. G. F. Thomas R. Henderson, "NS-3 project goals," in *Proceeding from the 2006 workshop on ns-2: the IP network simulator*, New York, NY: ACM, 2006.
[16] H. E. Issariyakul teerawat, Introduction to network simulator NS2, Springer, 2011.
[17] P.-S. D. N. B. P. U. D. Fedyk, *IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging*, 2012.
[18] C. E. L. R. L. R. C. S. Thomas H. Cormen, Introduction To Algorithms, Third Edition, Cambridge: MIT press, 2009.
[19] C. E. L. R. L. R. C. S. Thomas H. Cormen, "25 All Pairs Shortest Paths," in *Introduction to algorithms, Third Edition*, MIT Press, 2009, p. 686.
[20] N. B. David Allan, 802.1aq shortest Path Bridging: Design and Evolution The Architect's Perspective, New York: IEEE, 2012.