Decentralised, Dynamic Network Path Selection in High Performance Computing

John Anderson, Matt Piazza, Aspen Olmsted
Department of Computer Science
College of Charleston, Charleston, SC 29401
andersonjd@g.cofc.edu, piazzamp@g.cofc.edu, olmsteda@cofc.edu

***Abstract*— In this paper, we investigate the problem of providing highly available, decentralized, dynamic path selection in high performance computing networking. We look at a use case for dynamic path selection that better utilizes bandwidth available in the network. The network architecture we propose is a partial mesh grid whereby each host is directly connected to four forwarding devices. We propose an approach that decouples artificial network information and state from the devices making forwarding decisions. This method constructs a path one hop at a time while at each hop, selecting the next link based on the status of the link. The goal of this approach is to create a network topology in which new transmissions are routed around congestion. It also allows us to build a network with low-cost, lower bandwidth links instead of a few high cost, high bandwidth links. The goal of this work is to take the ideas of a controller based SDN solution and examine the merits of distributing that functionary to the actual nodes in the network.**

Keywords-networking; path slection; decentralised; SDN

I. INTRODUCTION

Most network routing protocols today rely on some amount of artificial information in order to make a forwarding decision. We refer to this as artificial information because it is information ascertained from outside the protocol itself. For instance, BGP prefers the path with the highest "weight" which is an arbitrary value that a network engineer assigns to a link [1]. The outside information causes the routing protocol to become more deterministic, making it less able to dynamically adapt to changing network load. This means that in a trivial case, the traditional protocol will use the same path no matter the amount of saturation on that path. The old approach causes network engineers to construct networks with high cost, high bandwidth backbone links that most traffic traverses at some point [2]. While traffic flow may be predictable in this model, once the backbone reaches capacity, it is costly to increase bandwidth and redundancy.

Our design views a high performance compute cluster connected in a grid fashion (Fig. 1). In this design, each forwarding node is connected to each of its closest neighbors, and each compute node (generalized as a server) is connected to all four of the closest forwarding nodes. This partial mesh topology builds in a great deal of redundancy while allowing each individual link to remain small in terms of bandwidth. Now that we have decentralized and removed the backbone from the architecture, we can allow the network to construct a vast number of unique paths between two compute nodes.

Our solution removes a great deal of complexity from the standard network protocols used to making forwarding decisions. Our protocol dynamically constructs the best path between two compute nodes before the two nodes begin transmitting. This is the basic operation of an SDN controller [3]. However, path selection is preformed hop by hop and is thus decentralized, unlike the basic principles of the OpenFlow standard [4]. By decentralizing the flow construction, we eliminate the need for a controller. The standard method that we compare to is a typical 3-tier star topology in which traffic traverses to a core forwarding node and then makes its way to the destination. We compare the two methods by looking at the average link saturation over the entire path of a transmission and how this value is affected by the number of concurrent transmissions.

II. EXAMPLE TRANSMISSION

We demonstrate our work using a High Performance Computing Cluster. In such a cluster, nodes utilize the network to transmit large amounts of data between one another. This topology consists of a partial mesh, grid of forwarding nodes and compute nodes connected throughout. **Error! Reference source not found.** exemplifies such a topology consisting of nine compute nodes in which we wish to construct a path from node A to node B. This path

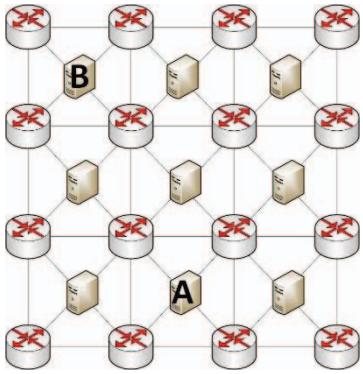

Figure 1. Example of our physical architecture

978-1-908320/61/2/$31.00 © 2016 IEEE

will be negotiated before the transmission begins and will exist only as long as the transmission is active. The method we propose is able to be adapted to operate at either OSI layers two or three. In this example, we have abstracted the saturation state of the various network links to say that because of already in progress transmissions, each link has some current saturation value, x.

III. PROTOCOL

Based on the example transmission above, the protocol is initiated by node A sending a Path Construction Request (PCR) to a directly connected forwarding node of its choosing. The PCR contains metadata about the transmission such as the source and destination nodes, the estimated saturation cost and the estimated life time of the transmission; known as the Transmission Metadata Packet (TMP). This metadata is sent to each hop as the path is constructed so that each forwarding node can make the best link selection. At this point, the directly connected forwarding node acknowledges the PCR with either an acceptance or denial. A denial of PCR tells the compute node to select another forwarding node. On the other hand, an acceptance PCR tells the compute node that its path is now being constructed.

Since the network topology is a grid and the TMP is available to each potential forwarding node in the path, the current code knows its relative location to the destination compute node. This means that at each hop, a forwarding node knows that the destination is in one quadrant relative to itself. This principal is illustrated in **Figure 2**, where by destination node B is quadrant two of the Current Forwarding Node (CFN).

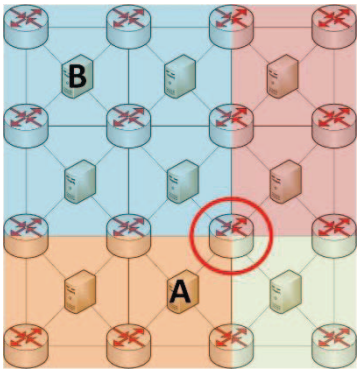

Figure 2. Showing that at each hop, the CFN knows in what quadrant the destination prosides

In most cases, this allows the CFN to prefer two links to select the next link in the path; a north/south and an east/west link. The CFN now makes its selection based on a few criteria. (1) Links that remain in the destination quadrant are preferred. (2) Links with the lowest saturation are preferred. (3) The selected link must not become oversubscribed when the TMP estimated saturation is applied to the current saturation. These criteria constitute a normal selection scenario. There are however a handful of non-standard scenarios we must account for in the selection process.

If the selected link results in the creation of a loop—returning to a forwarding node which is already in the candidate path—we check to see if another viable link is available on the returned node. If one is viable, backtrack to that node and select that link. If a secondary link is not available on the returned node, we must not select the link with results in a loop.

If the CFN is directly connected to the destination, compute node, select this link so long as the additional traffic will not oversubscribe the link, even if there is a link of lower saturation available. If, after exhausting the mentioned methods, oversubscription is the only means remaining to select a link, choose the link which will have a decreased saturation the soonest. This is where the TMP estimated transmission life time comes into play.

IV. RESULTS

Fig. 3 shows the results of our findings. Our method was found to have a significantly smaller average saturation across the transmission path.

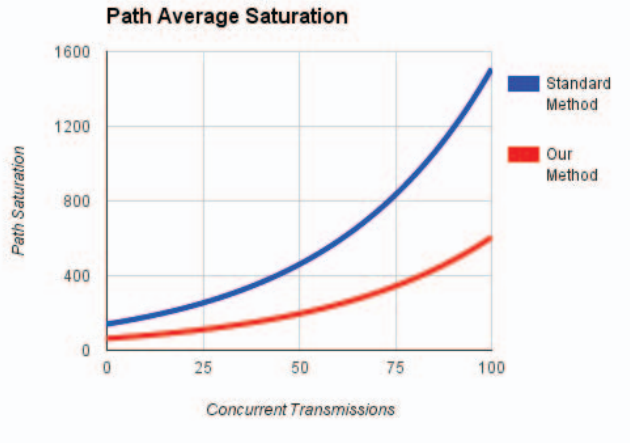

Figure 3. Our finding as compaired to the standard architecture and routing method

V. CONCLUSION

In this paper, we propose a new way to construct High Performance Computing network topology to allow for low cost, congestion avoidance, and a decentralized path selection method. In this way, we dynamically select a path which routes around congestion. We remove as much artificial influence from the protocol and allow it to select the most optimal link to use at each hop along the way. Our method reduces the saturation on individual links thereby reducing the need for high cost, high bandwidth links in the datacenter.

978-1-908320/61/2/$31.00 © 2016 IEEE

VI. REFERENCES

[1] I. John W. Stewart, BGP4: Inter-Domain Routing in the Internet, Boston, MA: Addison-Wesley Longman Publishing Co., Inc., 1998.

[2] Cisco Systems, Inc., *Cisco Data Center Infrastructure 2.5 Design Guide,* San Jose, CA: Cisco Systems, Inc., 2007.

[3] M. K. a. T. V. L. Sugam Agarwal, "Traffic engineering in software defined networks," in *INFOCOM*, 2013.

[4] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," 14 March 2008. (Online). Available: http://archive.openflow.org//documents/openflow-wp-latest.pdf. (Accessed 28 March 2016).

978-1-908320/61/2/$31.00 © 2016 IEEE