Applying Traffic Merging to Torus Datacenter Networks

Saurabh Verma[1], Ravi Shankar Shukla[2], Mohd. Fraz[3], Hina Saxena[4]
[1]M.Tech. Scholar, Invertis University, Bareilly,
[2]Head- Deptt. Of Information Technology, Invertis University, Bareilly,
[3]M.Tech. Scholar, Invertis University, Bareilly,
[4]M.Tech. Scholar, Invertis University, Bareilly,

ABSTRACT
Numerous studies have shown that datacenter networks typically see loads between 5% to 25% but the energy draw of these networks is equal to operating them at maximum load. In this paper, we propose a novel way to make these networks more energy proportional i.e. the energy draw scales with the network load. We propose the idea of traffic aggregation in which low traffic from N links is combined together to create K < N streams of high traffic. These streams are fed into K switch interfaces which run at maximum rate while the remaining interfaces are switched to the lowest possible one. We show that this merging can be accomplished with minimal latency and energy costs while simultaneously allowing us a deterministic way of switching link rates between maximum and minimum. Hence, for as much as the packet losses are statistically insignificant, the results show that energy-proportional datacenter networks are indeed possible.

KEY WORDS: Traffic Merging and Torus Topology.

1. INTRODUCTION
The electricity consumption of datacenters is a significant contributor to the total cost of operation over the lifetime of these canters and as a result, there have been several studies that aim to reduce this cost. Since the cooling costs scale as 1.3x the total energy consumption of the datacenter hardware, reducing the energy consumption of the hardware will simultaneously lead to a linear reduction in cooling costs as well [1].

In this paper, we present an innovative approach to adapt energy consumption to load for datacenter networks. The key idea is to merge traffic from multiple links prior feeding it to the switch. This simple strategy allows more switch interfaces to remain in a low power mode while having a minimal impact on latency [1, 2, 3]. Other general approaches attempt to reduce network-wide energy consumption by dynamically adapting the rate and speed of links, routers and switches as well as by selecting routes in a way that reduces total cost [4, 5,6].

We have explored the idea of traffic merging in depth in the context of enterprise networks in [7, 8, 9]. Indeed, the big advantage of the merge network is that, unlike the most other approaches, it works in the analog domain, so it does not introduce delays for store-and-forward Layer 2 (L2) frames, rather it redirects such frames at Layer 1 (L1) between external and internal links of the merge network itself. In addition, the merge network allows reducing frequent link speed transitions due to the use of the low power mode. In our approach, such transitions happen only infrequently thus allowing us to minimize the delay due to the negotiation of the new link rate and the additional energy required for the rate transition. Concept of merge network has been applied on mesh topology already.

2. MERGE NETWORK

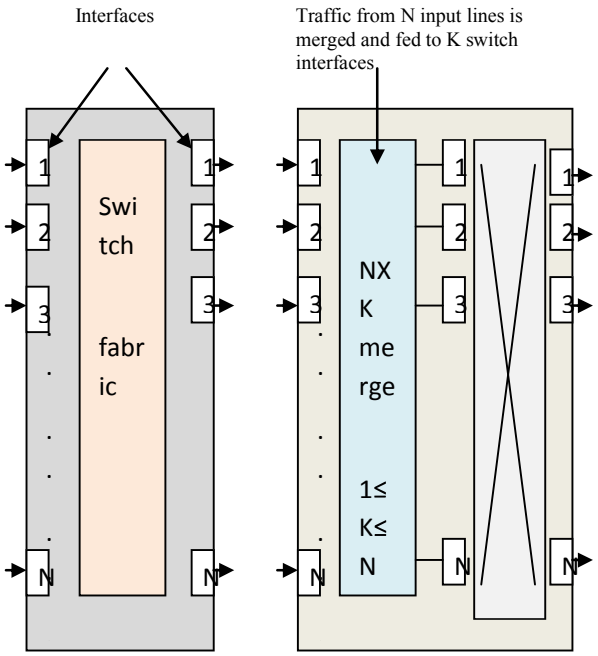
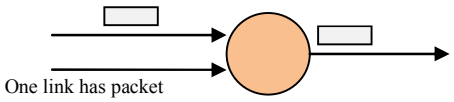


Figure 1: Switch without and with merge network.

The key idea we study is that of merging traffic arriving at a switch from multiple links and feeding that to few interfaces. The motivation for doing so is the observation made by various authors that per-link loading in datacenter networks tends to be well below 25% all the time and is frequently below 10% as well. Thus, by merging traffic we are allowing several of the switch interfaces to operate in low power modes. Indeed, as we discuss in [8] it is also possible to replace high port density switches with lower port density switches without affecting network performance in any way. Figure1 illustrates the traffic to/from N links are merged and fed to K interfaces. Setting the parameter K according to the incoming traffic load allows us to reduce the number of active interfaces to K and enables N - K interfaces to be in low power modes. As an example, if the average traffic load on 8 links coming in to a switch is 10%, we could merge all the traffic onto one link and feed it to one switch port running at maximum rate, thus allowing the remaining ports to enter low power mode [8, 9].

In order to understand how traffic merging can help in datacenter networks, we need to examine the details of the merge network itself [10,11]. A generic N*K merge (with K ≤ N) is defined with the property that if at most K packets arrive on the N uplinks (i.e. from N links into the switch) then the K packets are sent on to K sequential ports (using some arbitrary numbering system). For example, consider a 4x4 merge network as in Figure 2 denotes the incoming links and 1 - 4 denote the switch ports. The traffic coming in from these links is merged such that traffic is first sent to interface 1 but, if that is busy, it is sent to interface 2, and so on. In other words, we load interfaces sequentially. This packing of packets ensures that many of the higher numbered interfaces will see no traffic at all, thus allowing them to go to the lowest rate all the time [8, 10, 11].

The key hardware component needed to implement this type of network is called selector, whose logical operation is described in Figure 2. There are 2 incoming links and 2 outgoing links. If a packet arrives only at one of the two incoming links, then it is always forwarded to the top out going link.



Both links have packets. The earlier one is sent along the upper (default) output of the selector.

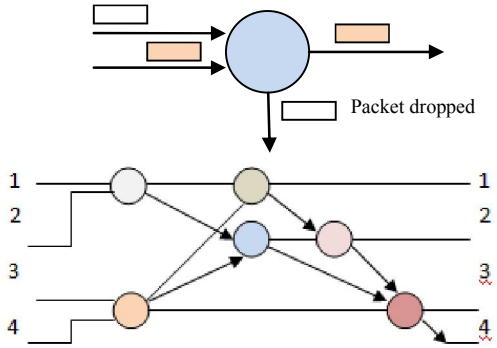


Figure 2: A 4x4 uplink merge network.

However, if packets arrive along both incoming links, then the earlier arriving packet is sent out along the top outgoing link and the latter packet along the other one. The hardware implementation, described in [7], is done entirely in the analog domain. Thus, a packet is not received and transmitted in the digital sense, rather it is switched along different selectors in the network much as a train is switched on the railroad. This ensures that the latency seen by a packet through the merge is minimal and the energy consumption is very small as well. We have also shown previously [9] that the minimum depth of an NxK merge network is $\log_2 N + K - 1$ with the number of selectors needed equal to

$$\sum_{i=1}^{k} N-i$$

On the downlink (i.e. from the switch to the N links) the merge network has to be able to forward packets from any of the switch ports (connected to the K outputs of an N x K merge network) to any of the N downlinks and be able to forward up to N packets simultaneously. This network uses a simple implementation consisting of multiplexers since we have to send packets from any of the K interfaces to any one of N links. However, in order for this part to work correctly, we need to embed the control logic inside the switch because the packet header has to be parsed to determine which of the N links they must be send out on [7]. In addition to this hardware, the merge network requires a software layer within the switch to ensure that the wide variety of LAN protocols continue working correctly (protocols such as VLANs IEEE 802.1P and 802.1H, access control IEEE 802.1X and many others). The needed software is essentially a port virtualization layer that maps K physical ports to N virtual ports in the switch. Thus, the protocol functionality is unaffected.

3. DATACENTER NETWORK TOPOLOGY

We study the application of our merge network to torus datacenter network topology. This concept has been applied on mesh topology already. A torus interconnect is a network topology for connecting processing nodes in a parallel computer system. It can be visualized as a mesh interconnect with nodes arranged in a rectilinear array of N = 3, 3, or more dimensions, with processors connected to their nearest neighbours, and corresponding processors on opposite edges of the array connected.

Topologically, Torus is arrangement of computer nodes in circle. In this topology all node are connected to adjacent nodes and nodes at the end are connected directly or in wrap around connections. Torus topology is like a mesh topology, the only difference between torus and mesh topology is that the switches on the edges are connected to the switches on the opposite edges through wrap-around channels. Every switch has five active ports: one is connected to the local resource while the others are connected to the closest neighbouring switches. A Torus topology is a multi-dimensional direct networks. Although the torus architecture reduces the network diameter, the long wrap-around connections may result in excessive delay. However this problem can be avoided by folding the torus [12].

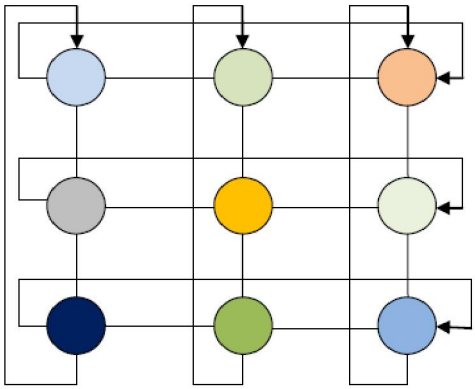


Fig 3 : 3x3 Torus Topology

The main problem with the mesh topology is its long diameter that has negative effect on communication latency. Torus topology was proposed to reduce the latency of mesh and keep its simplicity also.

Parameter	Symbol	Torus Topology	Mesh Topology
Number of switches connected to other switches	m	4 at each junction in 2-d torus or 6 at each junction in 3-d torus.	2d
Diameter	h_{max}	$\sqrt{\mathbf{k}}$	d√s

Table 1: Summary of key parameters. s = n/c - number of switches, d = dimension, n = number of hosts, c = number of hosts/switch, k=no of nodes.

4. Results

The results of traffic merging on Torus Topology are obtained with help of node analysis. Node analysis is accomplished by obtaining throughput, end to end delay and packet fraction. Simulation of topology is completed on NS2 networking tool.

Throughput is amount of data transferred from source to destination or processed in a specified amount of time. Data Transfer rates for disk drives and networks are measured in terms of throughput. Typically, throughputs are measured in Kbps, Mbps and Gbps. Greater value of throughput means the better performance of the protocol. In Fig 4 throughput of Torus is greater than Mesh topology. Throughput of Torus and Mesh is 8832.1kbps and 5838.0 kbps respectively.

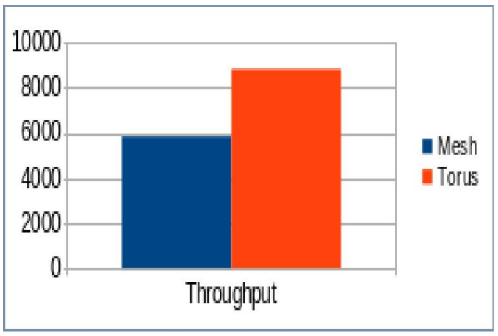


Fig 4: Throughput

End to End Delay is average time taken for a packet to be transmitted across a network from source to destination. It also includes the delay caused by route discovery process and the queue in data packet transmission. Only the data packets that successfully delivered to destinations that counted. The lower value of end to end delay means the better performance of the protocol. In Fig 5 end to end delay of Mesh is greater than Torus topology. End to End Delay of Torus and Mesh is 0.022 and 0.033 respectively.

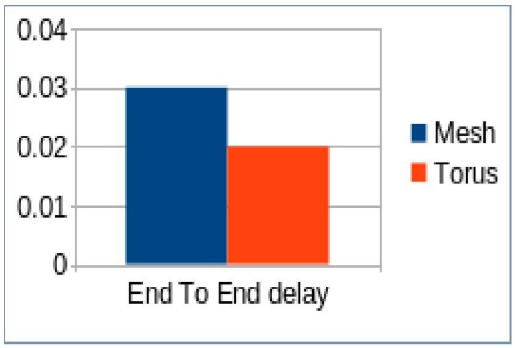


Fig 5: End to End Delay

Packet Fraction is ratio of the number of delivered data packet to the destination. This illustrates the level of delivered data to the destination. The greater value of packet delivery ratio means the better performance of the protocol. In Fig 6 Packet Fraction of Torus and Mesh is 2.0 and 1.25 respectively.

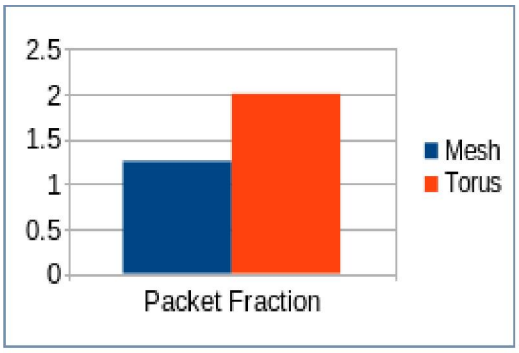


Fig 6: Packet Fraction

5. Conclusion and Future Work

At last, it has been concluded that Concept of Merging Traffic has been successfully applied on Torus topology. Results are better than Mesh topology. Merging Traffic technique is efficient than other existing techniques for energy conservation. So, we can say that energy is conserved on Torus Topology by Applying Traffic Merging.

Despite the positive results concerning energy saving, the proposed merge network solution is not proven to be optimal but we are studying that problem as part of future work. In addition, it would be interesting to test the merge network in other datacenter than the FBFLY and with real traffic traces. Merging Traffic concept can be applied on wireless Torus topology and other higher topologies.

References:

[1] D. Abts, M. Marty, P. Wells, P. Klausler, and H. Liu," "Energy Proportional Datacenter Networks",in Proceedings of the 37th International Symposium on Computer Architecture (ISCA). Saint Malo, France: ACM, June 2010, pp. 338-347.

[2] T. Benson, A. Akella, and D. Maltz, "Network Traffic Characteristics of Data Centers in the Wild," in Proceedings of the 10th Conference on Internet Measurement (IMC). Melbourne, Australia: ACM, November 2010, pp. 267-280.

[3] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "ElasticTree: Saving Energy in Data Center Networks," in Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI). San Jose, CA, USA: USENIX Association, April 2010, p. 17.

[4] R. Bolla, F. Davoli, R. Bruschi, K. Christensen, F. Cucchietti, and S. Singh, \The Potential Impact of Green Technologies in Next-Generation Wireline Networks: Is There Room for Energy Saving Optimization? ," IEEE Communications Magazine, vol. 49, no. 8, pp. 80-86, August 2011.

[5] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," IEEE Communications Surveys & Tutorials (COMST), vol. 13, no. 2, pp. 223-244, Second Quarter 2011.

[6] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing ISP Network Energy Cost: Formulation and Solutions," IEEE/ACM Transactions on Networking, vol. PP, no. 99, pp. 1-14, 2011.

[7] C. Yiu and S. Singh, "Energy-Conserving Switch Architecture for LANs," in Proceedings of the 47th IEEE International Conference on Communications (ICC). Kyoto, Japan: IEEE Press, June 2011, pp. 1-6.

[8] S. Singh and C. Yiu, "Putting the Cart Before the Horse: Merging Traffic for Energy Conservation," IEEE Communications Magazine, vol. 49, no. 6, pp. 78-82, June 2011.

[9] C. Yiu and S. Singh, "Merging Traffic to Save Energy in the Enterprise," in Proceedings of the 2nd International Conference on Energy-Efficient Computing and Networking (e Energy), New York, NY, USA, May-June 2011.

[10] Alessandro Carrega University of Genoa , Italy, Suresh Singh Portland State University Portland, OR 97207, Raffaele Bolla University of Genoa Genoa, Italy, Roberto Bruschi National Inter-University Consortium for Telecommunications (CNIT) Genoa, Italy "*Applying Traffic Merging to Datacenter Networks* " in 2012 IEEE.

[11] Alessandro Carrega, Roberto Bruschi and S. Singh, "*Traffic Merging for Energy-Efficient Datacenter Network*" in Proceedings of the 2nd International Conference on Energy-Efficient Computing and Networking (e-Energy), New York, NY, USA, May, June 2011.

[12] M. Mirza-Aghatabar, S.Koohi, S. Hessabi, M. Pedram "*An Empirical Investigation of Mesh and Torus NoC Topologies under Different Routing Algorithms and Traffic Models*"