International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 8 Issue 1 –APRIL 2014.

IMPROVING LINK UTILIZATION IN DATA CENTER NETWORK USING NEAR OPTIMAL TRAFFIC ENGINEERING TECHNIQUES

L. Priyadharshini [1], S. *Rajanarayanan, M.E (Ph.D)* [2]
[1]*Final Year M.E-CSE,* [2]*Assistant Professor*
[1&2]*Selvam College of Technology, Namakkal*

Abstract- Data center networks are driven by intensive application like data mining, web searching and cloud computing. Nowadays, topologies in data center networks applied multi-rooted tree like Canonical Tree, and Fat-Tree. In multipath routing Equal cost multiple path (ECMP) forwarding is used in most of the current data center networks. However, it fails to increased path diversity. Modified Penalizing Exponential Flow-spliTing (PEFT) routing algorithm in a data center environment based on the topologies such as canonical and fat-tree to increase the path diversity over unequal cost link. In proposed new algorithm to increase the link utilization ,efficient load balancing and the effects of increased packet reordering on application performance with MTCP and Packet scatter, can further reduce MLU, increase link utilization through PEFT routing and better load balancing, finally increase the overall network performances.

Index Terms-Data center routing, data center topology, multipath routing, traffic engineering

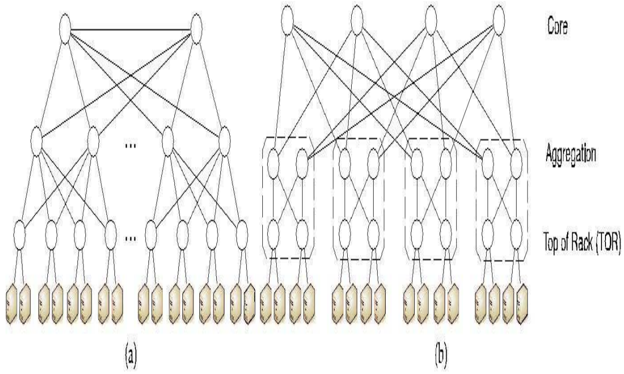
1. INTRODUCTION

Today's largest data centers have hundreds of thousands of servers to provide services across the internet and storage across many thousands of machines. Nowadays, topologies in data center networks applied multi-rooted Fat-Tree topology, Canonical tree as shown in Fig.1. (a) Canonical Tree topology with expensive switches in the higher layers and lower end edge switches that connect to thousands of servers [15] In Fig.1. (b) as shows that can horizontally expand DC architecture for increasing aggregate bandwidth among all communicating hosts by interconnecting a large number of inexpensive commodity switches [9]. In TE techniques based on a simple link state routing protocol able to provide path diversity in DC networks. The Penalizing Exponential Flow-spliTing (PEFT) routing algorithm is just such a protocol split traffic along all paths, but penalizes longer paths (i.e., paths with larger sums of link weights) exponentially [12]. It is a traffic engineering (TE) technique with hop-by-hop forwarding, i.e., routers running PEFT make forwarding and traffic splitting decisions locally and independently of each other. Packets can be forwarded through a set of unequal cost paths but the longer paths are penalized based on total link weights along the paths.

In this paper implemented new modified PEFT that improved the overall performance of network infrastructure. The network can operate efficiently without congestion and without adding extra networking components and better load balancing on other ways bottlenecked links.

The PEFT routing reduces MLU and increase the overall network capacity [1].In this paper implemented and evaluated new algorithm for a DC network environment. The main contributions are:

- A practical implementation of new algorithm for DC networks.
- Evaluated fat-tree and canonical DC network topologies. The result indicates that modified PEFT algorithm falls only small percent short of optimal TE in DC. At the same time, modified PEFT provides performance gain of at least 10 percent over PEFT in such topologies.
- Result shows that modified PEFT algorithm in this topology increase the DC network link utilization and load balancing, which in turn



improves DC capacity.

Fig.1. (a) Canonical tree and (b) Fat-tree DC network topology

2. RELATED WORK

By Greenberg et al. [9], the enterprise networking architectures were originally developed for much smaller data centers, in contrast to the ones existing today. The limitations of the traditional architecture have resulted in several workarounds and patches for the protocols to keep up with the new demands on data centers.

In [2], authors analyze the end-to-end traffic patterns in data center networks, and examine temporal and spatial variations in link loads and losses.

Delivering full bandwidth between arbitrary hosts in larger clusters requires a "multi-rooted" tree with multiple core switches requires a multi-path routing technique, such as ECMP [4].

Several systems have been proposed for energy-aware DCNs using the strategy of powering off unneeded switches and links [5] [8].

2.1 (PEFT) Penalizing Exponential Flow- splitting:

PEFT, packet forwarding is hop-by-hop forwarding and it splits the traffic along all paths but penalizes longer paths exponentially, i.e., routers running make forwarding and traffic splitting decisions locally and independently of each other.in addition packets can be forwarded through a set of unequal cost paths but penalizes longer paths exponentially based on the total link weight, In this protocol algorithm simplicity in path diversity and optimality in minimizing maximum link utilization is nontrivial. So that PEFT exploit path diversity (link utilization) and load balance traffic in the traffic is not efficient one through this reduced overall network performances.

3 TE FOR DC NETWORKS

TE is a necessary tool used on the internet to select route that make efficient use of network resources. In comparisons, ECMP is the multipath routing used in DC networks, but it fails to increase the path diversity and it is static mapping of flow to paths does not consider current network utilization or flow size and degrading overall switch utilization [13]. Centralized TE techniques, schedules in which track the network matrix and global optimal route with the least utilization will be assigned to the flow using ECMP [13]. In this schedules elephant flow can be exactly detected if the buffer of flow crosses a pre-defined threshold values, but does not consider below threshold values,

these flows never be scheduled. In comparison, virtual layer-2 [9] uses Valiant Load Balancing to randomize packet forwarding on a per-flow basis, which is essentially an ECMP mechanism over a virtual layer-2 infrastructure. The PEFT is a simple and link state protocol that can achieved optimal TE by splitting traffic not only the shortest paths, but also splitting traffic over longer path with exponential penalization [1], PEFT exploit link utilization and load balancing is non-trivial.

Based on the above observations, argue that the performance of current DC networks can be significantly managed to avoid congestion on bottleneck links. Among a large number of available TE techniques, such as [4], [12], [14], we have modified the PEFT routing algorithm to provide close to optimal TE for a variety of DC topologies [12]. Modified PEFT is a simple and link-state protocol that can achieve optimal TE by using not only shortest paths, but also splitting traffic over longer path with exponential penalization.

3.1 Self-stabilizing algorithms (Link Utilization)

Self-stabilizing distributed systems are guaranteed to con- verge to a desired state or behavior in finite time, regardless of the initial state. Convergence is guaranteed, i.e., after the system is affected by transient faults of unknown scale or nature, it will return to the desired behavior.

Hence, self- stabilization is a powerful approach for non-masking fault- tolerance. The actions of each individual node of a self- stabilizing system lead to a global behavior possibly not known to each entity. A node can only evaluate its local view, i.e., its own state and that of its neighbors.

3.2 weighted_roSund_robin (Load Balancing)

A round robin algorithm, but with different proportions of traffic being directed to the back-end nodes. Weights must be defined as part of the load balancer's node configuration.

The weighted round robin algorithm maintains a weighted list of servers and forwards new connections in proportion to the weight (or preference) of each server.

This algorithm uses more computation times than the round robin algorithm. However, the additional computation results in distributing the traffic more efficiently to the server that is most capable of handling the request.

Weighted Round Robin - Pseudo code

```
// calculate number of packets to be served each
round by connections
for each flow f and servers in this architecture are 1
Gb/s. The host-to-switch links are
f.normalized_weight = f.weight / f.mean_packet_size
min = findSmallestNormalizedWeight
for each flow f
f.packets_to_be_served = f.normalized_weight / min
// main loop
loop
  for each non-empty flow queue f
    min(f.packets_to_be_served,
f.packets_waiting).times do
      servePacket f.getPacket
```

Self-stabilizing algorithms - Pseudo code

```
Predicate :
inNeighbor(v) _ 9 w 2 N(v) : w.status = IN
waitNeighborWithLowerId(v) _ 9 w 2 N(v) : w.status
= WAIT ^ w.id < v.id
inNeighborWithLowerId(v) _ 9 w 2 N(v) : w.status =
IN ^ w.id < v.id
Actions:
R1 :: [status = OUT ^ : inNeighbor(v)]
□! status := WAIT
R2 :: [status = WAIT ^ inNeighbor(v)]
□! status := OUT
R3 :: [status = WAIT ^ : inNeighbor(v) ^ :
waitNeighborWithLowerId(v)]
□! status := IN
R4 :: [status = IN ^ inNeighbor(v)]
□! status := OUT
```

4. NUMERICAL RESULTS

In this part, we evaluate the proposed algorithms of the Modified PEFT is analyzed in terms of load balancing and Minimum Link Utilization.

MINIMIZATION OF MAXIMUM LINK UTILIZATION

Fig. 2 shows modified PEFT MLU performance close optimal to PEFT. The protocol deviates only by a small percentage from PEFT. The reason for such significant improvement is that modified PEFT optimizes distribution of flows such that they are unevenly split over all outgoing paths.

For example, if a server wants to transmit to another server in a neighboring rack, it physically has two equal-length shortest paths of three hops. But with PEFT the two paths may become unequal (Reflected in the sum of link weights along the path) after optimization. Then, the traffic is exponentially split over the outgoing interfaces.

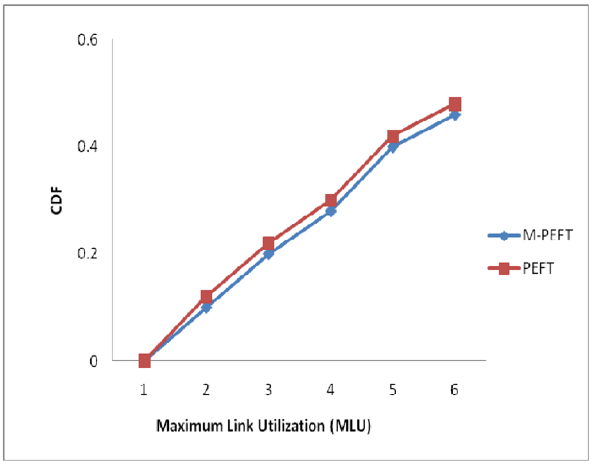


Fig. 2. Comparison of MLU for Modifed PEFT versus PEFT.

Similar to other TE techniques, PEFT needs to measure TM regularly and then update link weight and compute new traffic splitting ratios accordingly. Reactive and sparse TM updates prevent PEFT from reacting to changes in a timely manner [1].

LOAD BALANCING

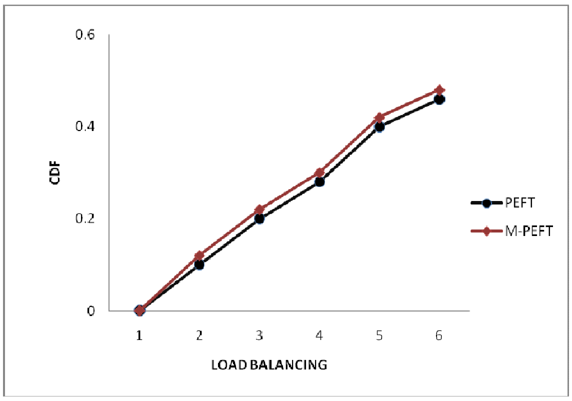


Fig. 3. Comparison of Load Balancing for Modifed PEFT versus PEFT

Fig.3 shows Load Balancing for modified PEFT versus PEFT. Performance of the DC network highly depends on the degree of path diversity. PEFT, a path based traffic splitting technique, schedules and splits traffic over longer paths to leverage path diversity, and, thus, better load balancing of the distribution of traffic [1].

5. CONCLUSION AND FUTURE WORK

The use of online unequal cost TE as an efficient and viable mechanism to improve load balancing and performance over DC environment. Based on PEFT algorithm, protocol forwards packets over multiple unequal cost paths, whereas traffic splitting decisions are independently made based on the total link weight overall reachable paths, and exponentially penalize longer ones. Reliable and in-order packet transmission can be readily achieved through multipath congestion aware protocols, such as multipath TCP and packet scatter .packet scatter is already in use in today’s switch. In modified PEFT using self-stability and weighted round robin algorithm, including fairer network-wide traffic load balancing, minimizing MLU, and increasing network capacity and efficient link utilization,

Finally improves the overall network performances. In Future work the effects of increased packet reordering on application performance with MTCP and Packet scatter.

REFERENCES

[1] Fung Po Tso, Dimitrios P. Pezaros, Pezaros,”improving data center network utilization using near- optimal traffic engineering,” IEEE Transactions on Parallel and Distributed Systems, 2013.

[2] A. Greenberg, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta. Towards a next generation data center architecture: scalability and commoditization. In PRESTO ’08: Proceedings of the ACM workshop on Programmable routers for extensible services of tomorrow,pages 57–62, New York, NY, USA, 2008. ACM.

[3] T. Benson, A. Anand, A. Akella, and M. Zhang, "Understanding data center traffic characteristics," in WREN '09: Proceedings of the 1st ACM workshop on Research on

enterprise networking. New York, NY, USA: ACM, 2009, pp. 65-72.

[4] C. Hopps. Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992, Internet Engineering Task Force, 2000.

[5] B. Heller, S. Seetharaman, and P. Mahadevan, "ElasticTree: saving energy in data center networks," *2010 NSDI*.

[6] Y. Shang, D. Li, and M. Xu, "Energy-aware routing in data center network," *2010 SIGCOMM*.

[7] IBM ILOG CPLEX Optimizer, www.ibm.com/software/integration/optimization/cplex-optimizer/, 2013

[8] M. Al-Fares, A. Loukissas, and A. Vahdat, "A Scalable Commodity Data Center Network Architecture," Proc. ACM SIGCOMM,2008.

[9] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri,D.A. Maltz, P. Patel, and S. Sengupta, "VL2: A Scalable and Flexible Data Center Network," Proc. ACM SIGCOMM, 2009

[10] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine Grained Traffic Engineering for Data Centers," Proc. ACM CoNEXT, 2011.

[11] U. Hoelzle and L.A. Barroso, The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines. Morgan and Claypool Publishers, 2009.

[12] D. Xu, M. Chiang, and J. Rexford, "Link-State Routing with Hop-by-Hop Forwarding Can Achieve Optimal Traffic Engineering," IEEE/ACM Trans. Networking, vol. 19, no. 6 pp. 1717- 1730, Apr. 2011.

[13] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: Dynamic Flow Scheduling for Data Center Networks," Proc. Seventh USENIX Symp. Networked Systems Design and Implementation (NSDI '10), 2010.

[14] A. Elwalid, C. Jin, S. Low, and I. Widjaja,"MATE: MPLS Adaptive Traffic Engineering," Proc. IEEE INFOCOM, 2001.

[15] Cisco Systems, "Data Center: Load Balancing Data Center Services Solutions Reference Network Design," Mar. 2004.