

Packet Reordering To Improve Data Center Network Using Near Optimal Traffic Engineering

Preetha M Kurup

PG Scholar
Anna University
Regional Center
Coimbatore

J Preethi

Assistant Professor
Anna University
Regional Center
Coimbatore

ABSTRACT

Data Centers are large collection of computers owned and operated by single or group of organizations. It is necessary to handle huge amount of data, there by traffic is complicated. To reduce the traffic volume and to diverse traffic in an effective way, I would like to introduce PEFT (Penalizing Exponential Flow spliTing), which penalize the longer path and diverse traffic in unequal cost path to reach its destination with in short time. An end-to-end transport layer protocol called migratory *Transmission Control Protocol*, which aggregate the available bandwidth of those redundant paths in parallel and also become more robust under path failures. Whenever some path fails, migratory Transmission Control Protocol will send the packets on other living paths and within few seconds it will be recovered. A shared detection mechanism is integrated into this system which dynamically detects and suppress paths with shared congestion so as to avoid the aggressiveness problem. A new scheduling algorithm can be used here to reorder the packet in an effective ways, that scheduling algorithm is PET (Packet Pair based earliest Delivery Path First) which effectively utilize full bandwidth and minimize the reordering together with Buffer Management Policy.

General Terms

Design, Performance

Keywords

Data Center network, Optimal Traffic Engineering, Buffer Management Policy, Migratory TCP

1. INTRODUCTION

Current DC networks employ Equal cost multipath forwarding (ECMP) to leverage the path diversity provided by network topology. Here traffic is spitted across multiple paths through hashing packets' headers. Recent research on DC network measurement has

confirmed that congestion happens when average link utilization is low. Therefore DC networks are often over provisioned, a small but significant fraction of link congestion can largely deteriorate the overall network performance, which demands the DC operators to expand or upgrade the networks. It seems the overall performance of network infrastructure can be improved through the performance of mitigation of congestion.

2. PROBLEM STATEMENT

There are many problems in the DC network. When packets are reordered, Transmission Control Protocol misinterprets the duplicate acknowledgements received as an indication of packet loss and invokes congestion. This can significantly lower Transmission Control Protocol throughput. Effective utilization of available bandwidth is another problem.

3. EXISTING SYSTEM

Existing system deals with PEFT (Penalizing Exponential Flow SpliTing). PEFT routing algorithm is such a protocol. It is a TE technique with hop-by-hop forwarding, i.e., routers running in PEFT makes forwarding and traffic splitting decisions locally and independently. The packets can be diverted through paths which carrying unequal cost, but the longer paths are avoided based on total link weights along the paths. PEFT consists of two separate components, namely link-state routing, including traffic splitting and link weight optimization. Despite PEFT [1] having been proven to achieve optimal TE[3] for wide-area ISP networks, its applicability for DC networks[2] remains largely unanswered because both traffic patterns and network topologies are enormously different in many ways due to the nature of cloud DC applications. We have implemented and evaluated a reactive online version of PEFT for a DC network environment. Contributions are threefold:

i) It provided a practical implementation of PEFT for DC networks.

ii) PEFT provides performance gain of at least 20 percent over ECMP in canonical and fat tree topologies in DC.
 iii) This algorithm has an advantage of PEFT's ability to route packets over unequal-cost paths.
 The use of online unequal cost TE is an efficient mechanism to improve load balancing and performance [5] over DC topologies. It offers large amount of improvements over the commonly employed ECMP and other TE technique for DC. Protocol forwards packets over multiple unequal cost paths in which traffic splitting decisions are independently made based on the total link weight over all reachable paths, and exponentially penalize longer path by means of PEFT algorithm. PEFT achieves near-optimal TE and outperforms ECMP in many ways, which included fairer network traffic load-

balancing, minimizing Maximum Link Utilization, and increasing network capacity. I have proposed a well suitable architecture that interlinks edge switches to further increase physical path diversity between any communicating server pairs. This modification also provides significant performance gains in reducing Maximum Link Utilization as well as in increasing network capacity, without requiring any further investment from DC network operators.

4. PROPOSED SYSTEM

Proposed system architecture shown in the following figure.

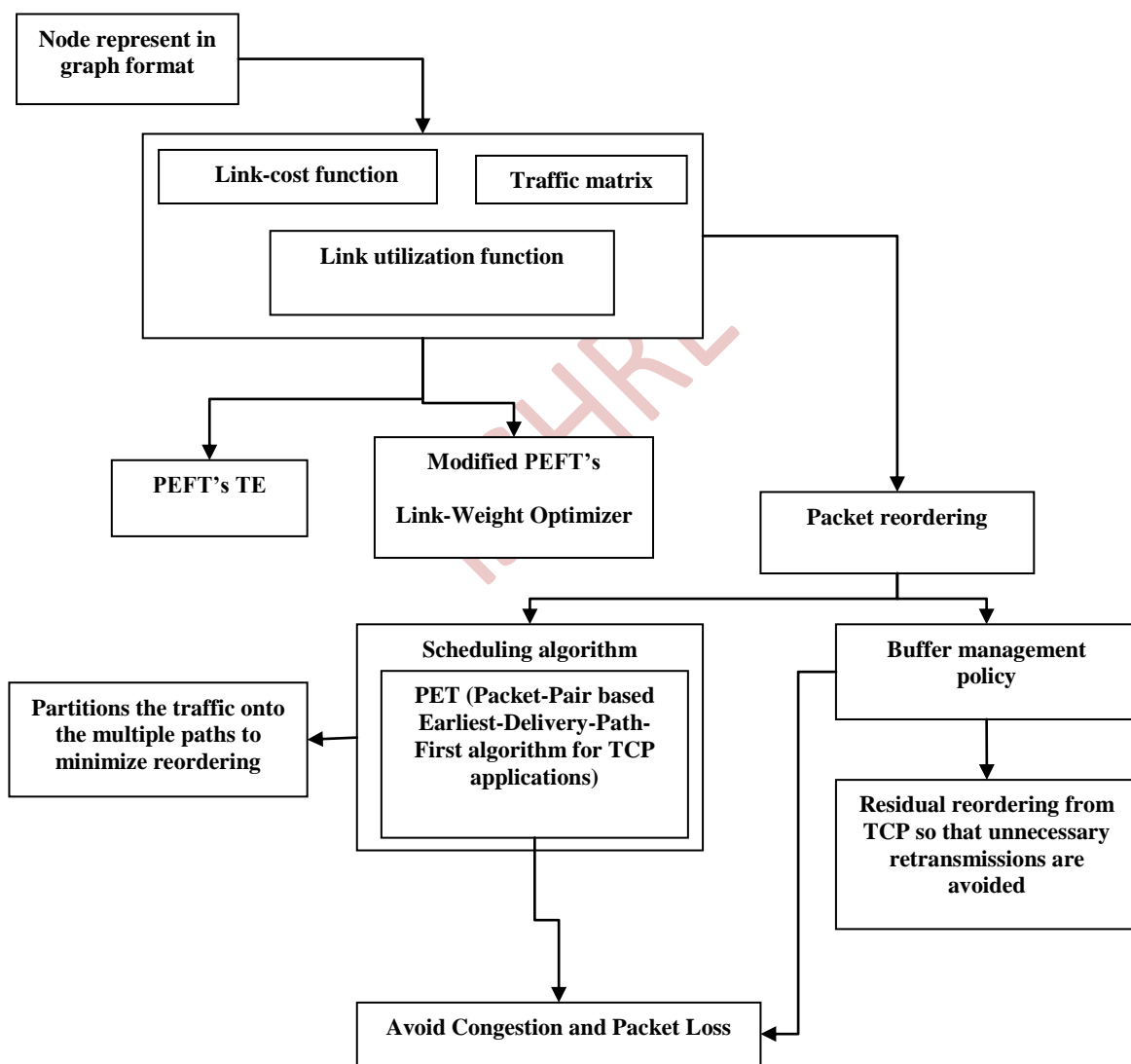


Fig 1: Architecture Diagram

Two approaches can be considered for improving overall performance of TCP. i) It propose a scheduling algorithm

that partitions traffic into different path (corresponding to each interface), so that reordering is minimized. This

algorithm estimates its available bandwidth and also minimizes reordering by sending packet pairs on the path.
 ii) A buffer management policy is another concept, which will hide any residual reordering from TCP. Through simulations in network-layer approach will achieve good bandwidth aggregation under a variety of network conditions. This approach proposed a scheduling algorithm - PET (Packet-Pair based Earliest-Delivery-Path-First algorithm for TCP applications) that partitions the traffic into the multiple paths to minimize reordering while utilizing bandwidths of the interfaces effectively. PET minimizes reordering by estimating the delivery time of packets on each Internet path and scheduling the packets on the path that delivers it at earliest. Proposed modules are follows.

4.1. Creation of Network Model

A network is simulated, with minimum of 30 nodes moving in a defined area. Each node moves randomly in this area, with a speed selected in a range $[0, v_{max}]$ with no pause time. The number of node are created using ns2. Consider a network as a directed graph $G = (W, E)$ where V is the set of nodes (where $N = |V|$), $E = E$ is the set of links (where E is the edge), and link $(u; v)$ has capacity $c_{u,v}$.

4.2. PEFT

The traffic is represented by a traffic matrix (TM) $D(s; t)$ for source-destination pairs indexed by $(s; t)$. $\phi(f_{u,v}, c_{u,v})$ is a link cost function that is an increasing function of $f_{u,v}$. When we consider the link utilization function $f_{u,v} = c_{u,v}$, then the PEFT's TE objective is to minimize $\max_{u,v \in E} \phi(f_{u,v}, c_{u,v})$. Optimal TE requires solving the following flow conservation and link capacity constraints

$$\min \phi(\{f_{u,v}, c_{u,v}\}) \quad (1a)$$

$$\sum_{v: (s,v) \in E} f^t_{s,v} - \sum_{u: (u,v) \in E} f^t_{u,s} = D(s, t) \forall s \neq t \quad (1b)$$

PEFT allows traffic splitting exponentially over unequal-cost path, as shown in below, where $p_{u,t}$ is the set of paths from u to t and $x^i_{u,t}$ is the fraction of packet forwarded to the i^{th} path, i.e., $p^i_{u,t}$

$$x^i_{u,t} = e^{-p^i_{u,t}} / \sum_j e^{-p^j_{u,t}} \quad (1c)$$

4.3. Modified PEFT

Modified PEFT uses TM and link cost function together with link the link-weight optimization module for computing the "best-fit" link weights for the best possible

traffic distribution. Resulting link weights to the PEFT-enabled switches used to compute the desirable traffic distribution in the network. Implementing PEFT requires solving for optimal traffic flow distribution followed by executing the link-weight optimization procedure to produce the link weights that will achieve this optimal traffic distribution. Modified PEFT will propagate the TM over the network and each switch independently compute link weights locally, which is reported on a DC network. A negligible fraction of traffic flows are destined outside the rack. On the other hand, the link utilization is a reactive mechanism that detects abnormalities in link utilization. Link weight computation and TM exchange are triggered immediately. Locally measured TMs are exchanged over the network through a link-state advertisement (LSA). On this front we have also implemented a lightweight Hello protocol (based on OSPF's LSA) to facilitate LSA among switches running PEFT.

4.4. PET scheduling algorithm

- Criterion 1: Utilize bandwidth of all interfaces
- Criterion 2: Minimize reordering
- Criterion 3: Hide reordering from TCP
- Criterion 4: Detect packet losses and react to them in a timely fashion
- Criterion 5: Avoid Burstiness of Traffic
- Criterion 6: Isolate losses

4.5. Buffer Management Policy (BMP)

Due to the use of packet-pairs, and also due to errors in bandwidth estimation, PET scheduling would result in some amount of reordering.

4.6 Performance Evaluation

Here evaluate the performance of PEFT and PET together with BMP with respect to improved load-balancing derived from path diversity, reduced maximum link utilization, throughput, congestion control and the overall network capacity gain.

5. ACKNOWLEDGMENTS

First and foremost I place this project work on the feet of GOD ALMIGHTY who is the power of strength in each steps of progress towards a successful completion of project. I would like to express my sense of proud gratitude and indebtedness to those who guided me for her valuable guidance, suggestions, timely supervision for successful completion of project. Above all I would like to thanks all members of my family and friends for their constructive criticism and construct support in making this project a grand success.

6. REFERENCES

- [1] Fung Po Tso, Dimitrios P Pezaros, "Improving Data Center Network Utilization Using Near Optimal Traffic Engineering, Vol 24, No 6, June 2013 "

- [2] 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.
- [3] T. Benson, A. Akella, and D. Maltz, "Network Traffic Characteristics of Data Centers in the Wild," Proc. Internet Measurement Conf. (IMC), 2010.
- [4] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine Grained Traffic Engineering for Data Centers," Proc. ACM CoNEXT, 2011. Cisco Systems, "Data Center: Load Balancing Data Center Services Solutions Reference Network Design," Mar. 2004.
- [5] A. Curtis, J. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, and B.S, "Devoflow: Scaling Flow Management for High-Performance Networks," Proc. ACM SIGCOMM, 2011.
- [6] M. Schlansker, Y. Turner, J. Tourrilhes, and A. Karp, "Ensemble Routing for Datacenter Networks," Proc. ACM/IEEE Sixth Symp. Architectures for Networking and Comm. Systems (ANCS), 2010
- [7] B. Fortz and M. Thorup, "Increasing Internet Capacity using Local Search," Computational Optimization and Applications, vol. 29, no. 1, pp. 13-48, 2004.
- [8] P. Gill, N. Jain, and N. Nagappan, "Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications," Proc. ACM SIGCOMM, 2011.

IJSHRE