JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 27, 1747-1759 (2011)

Short Paper

Load Balancing in Metro Ethernet Networks by Selecting the Best Spanning Tree*

GHASEM MIRJALILY, MOHAMMAD HADI KARIMI TAFTI
AND SHAHRAM RAJAEE
Faculty of Electrical and Computer Engineering
Yazd University
Yazd, 89195-741 Iran

Ethernet networks rely on the so-called spanning tree protocol (STP) in order to prevent loops. This protocol imposes a severe penalty on the performance and scalability of metro Ethernet networks, since it makes inefficient use of the links and switches. This paper presents a new scheme that ranks all of the possible spanning trees and finds the best spanning tree not only based on shortest path selection but also based on load balancing on links and switches. Actually, we define three major criterions: load balancing over links, load balancing on switches and shortest path selection. We can weight the importance of each criterion based on our goal. Our solution is not a replacement for STP; it is actually a complement for it. In our approach, after finding the best spanning tree, we can force STP to select it by assigning proper values to switch IDs and link costs. In a failure, STP can immediately select another spanning tree based on its rank by changing switch IDs and link costs. Through some simulations on a typical network, we show effectiveness of our approach.

Keywords: Ethernet, spanning tree protocol, load balancing, metro Ethernet network, best spanning tree

1. INTRODUCTION

Ethernet technology has so far been widely accepted in enterprise deployments, and millions of Ethernet ports have already been deployed. The significant advancement of Ethernet technology is pushing Ethernet from the local area network environment to metropolitan and wide area network environments [2]. However, metro networks have very different requirements from the early LANs. Therefore Ethernet needs to be enhanced if it is to meet these requirements. Among the requirements, Ethernet needs to provide a good Quality of Service (QoS) as well as admission control and traffic engineering functionalities. One of the main tasks of traffic engineering is load balancing, the idea of which is uniform resource utilization by moving traffic from congested links to other parts of the network in a well-controlled way.

Current Ethernet technology relies on IEEE spanning tree protocol (STP), which

---

Received December 21, 2009; revised May 6 & July 14, 2010; accepted August 11, 2010.
Communicated by Ren-Hung Hwang.

[*] This work was partially supported by Iran Telecommunication Research Center (ITRC), and parts of this paper have been presented at the 8th IEEE International Conference on Computer and Information Technology, July 2008, Sydney, Australia [1].

provides a loop-free connectivity across various network nodes. STP does this task by reducing the topology of a switched network to a tree topology where redundant links are pruned. This action ensures there is a unique path from any node to every other node.

STP is a well-known and standard algorithm that selects the root and designated ports of the tree from switch IDs, port IDs and link costs. For switches, a unique 64-bit switch identifier is assigned by appending a 16-bit priority field in front of a unique 48-bit MAC address. The MAC address comes from the switch vendor, but the priority field can be set by the user. The switch with the lowest numerical identifier will become the root switch. Similarly, a 16-bit port identifier exists consisting of an 8-bit port address preceded by an 8-bit priority field [3]. In the IEEE standard 802.1D, it is recommended to assign the link costs based on link speeds as shown in Table 1.

Table 1. Link cost recommendations.

Bandwidth	Link Cost Value
4 Mb/s	250
10 Mb/s	100
16 Mb/s	62
100 Mb/s	19
1 Gb/s	4
10 Gb/s	2

In STP, the Disabled nodes are kept in a standby mode of operation until a network failure occurs. At that time, the STP will attempt to construct a new tree using any of the previously disabled links. The main disadvantages of STP are slow failure recovery and lack of load balancing. The re-convergence time of STP in the case of a failure is typically 30 seconds. Time critical applications on Ethernet can only afford a reconfiguration time less than few tens of milliseconds. To improve upon reconstruction time, IEEE standardized its Rapid Spanning Tree Protocol (RSTP).

In both STP and RSTP, all customers need to use the same spanning tree and there isn’t any traffic engineering mechanism for load balancing. This results in uneven load distribution and bottlenecks, especially close to the root, and as pointed out by the Metro Ethernet Forum, leads to inefficient utilization of expensive fiber links in metropolitan area networks. Currently, one of the main approaches to address this issue is to overlay the physical network with logical networks, referred to as virtual LANs. A spanning tree instance is then run separately for each virtual LAN (or group of virtual LANs). This approach of maintaining multiple spanning trees can add significant complexity to network management. Further, having too many spanning trees may cause extremely high overhead due to control frames and could have compatibility problems with the legacy tree [4].

In this paper, we present a new graph theoretic approach for ranking spanning trees in metro Ethernet networks and for finding the best tree, not only based on shortest path selection but also based on load balancing on links and switches. Actually, we define three major criterions: load balancing over links, load balancing on switches and shortest path selection.

Also we define three coefficients α, β, γ corresponding to above criterions respectively. This allows us to weight the importance of each criterion based on our goal. We name this approach “Best Spanning Tree” (BST).

BST algorithm is application-independent and can be used to avoid under-utilization of network resources (links and switches). It is purposefully designed to be backward-compatible, as it relies on the IEEE STP protocol. In fact, it is not a replacement for STP; it is actually a complement for it.

In STP, the construction of spanning tree is based on switch IDs and link costs. In our approach, after finding the best spanning tree online or offline, we can force the standard STP to select it by assigning proper values to link costs and the priority field of the switch IDs. For example, assign zero cost to links you want to be a part of the spanning tree and assign a very large (infinity) cost to other links. In a failure, STP can immediately select another spanning tree based on its rank by changing the link costs and the priority field of the switch IDs.

This paper is organized as follows: Section 2 gives a brief review of the previous related works. The proposed Best Spanning Tree algorithm is explained in section 3. In section 4, some simulation results are presented and some conclusions are drawn in section 5.

2. RELATED WORK

Traffic engineering of Ethernet using spanning tree is a widely researched topic because of performance issues. Several enhancements have been proposed in the literature in order to solve this problem by mitigating congestion near the root.

In [5], Edge-disjoint trees have been investigated for fault tolerance. However, it is not always possible to build two edge-disjoint trees from a given physical topology unless it is 4-edge-connected.

The Distributed Load Sharing (DLS) technique enables use of some of the links not belonging to the tree [6]. Under specific topological constraints, this technique can provide alternate paths to the spanning tree, thus helping to relieve local congestion. In general, however, no guarantee is provided on the overall performance improvement of the network.

SmartBridge [7] is a new bridge architecture proposed to address the problems associated with spanning trees in LANs. Packet forwarding in smartBridge architecture is done along the shortest paths. Although shortest path switching may provide a low latency path, it does not address the load balancing issue in the network and requires all bridges in the network to be smartBridge compliant.

King-Shan Lui *et al.* [8] discuss a novel approach named STAR (Spanning Tree Alternate Routing) to find and forward frames over alternate paths that are probably shorter than their corresponding tree paths. Although the approach reduces latency between most of the source and destination pairs, it risks overloading of critical links.

Another approach to load balancing is Tree-Based Turn-Prohibition (TBTP) [9]. TBTP constructs a less restrictive spanning tree by blocking a small number of pairs of links around nodes, called turn, so that all cycles in a network can be broken. However, TBTP is complex and did not consider the best spanning tree and switch load balancing.

MSTP or Multiple Spanning Tree Protocol [10] is defined in IEEE 802.1s. MSTP uses a common spanning tree that connects all of the regions in the topology. The regions in MSTP are multiple instances of the spanning tree. An instance of RSTP governs a region, where each region has its own regional root. The regional roots are in turn connected to the common root that belongs to the common spanning tree. Since MSTP runs pure RSTP as the underlying protocol, it inherits some drawbacks of RSTP as well. However, a failure in MSTP can be isolated into a separate region leaving the traffic flows in other regions untouched. In addition, the administrators can perform light load balancing manually by assigning certain flows to a specific spanning tree.

As mentioned before, the approach of maintaining multiple spanning trees can add significant complexity to network management and may cause extremely high overhead due to control frames. An approach named STEP (Spanning Tree Elevation Protocol) goes a step further in resilience but it doesn't provide a good load balancing [11].

A fault tolerant multiple spanning tree protocol was proposed in Viking [12]. Viking relies on per-VLAN-spanning tree implementation of Cisco where there is a separate spanning tree running on every switch for every VLAN. This has the limitation on the number of VLANs the Metro Ethernet can support due to the maximum VLAN tag size and the number of spanning trees a switch can handle without compromising performance.

The article in [13] looks into the basic problems of traffic engineering in Metro Ethernet, but its solution is again based on grouping customer VLANs and using MSTP.

In our previous works, we have considered the problem of traffic engineering in Ethernet networks and we have proposed some solutions [14-17]. In [14], we proposed a new forwarding strategy named Shortcut Switching Strategy (SSS) that uses some of the blocked links to forward some traffic. In [15] we analytically proved that SSS has a better load balance on links and switches versus STP. In [16], we proposed a Genetic Algorithm approach to find the best spanning tree. In [17], we introduced a new switching strategy that uses some blocked links as E-links, to connect nodes in the same level. In some situations, we can use these E-links to reduce the load on links and switches.

3. BEST SPANNING TREE (BST)

As mentioned before, both the STP and RSTP use the minimum spanning tree topology over the network, and forward packets along this tree. They construct it through exchanges of switch IDs and link costs. In practice, link costs are assigned based on link speeds. The minimum spanning tree problem involves determining the links which can join all the nodes of a network together such that the sum of the costs of the chosen links is minimized. This doesn't mean the shortest path between source and destination nodes. Furthermore, STP doesn't have any traffic engineering mechanism for load balancing, causes overloading of some resources. These problems result in suboptimal path selection and increase network congestion of some links and switches, while the utilization of some other links and switches is forbidden.

In a network, depending on its topology, a large number of spanning trees can be found that some of them can function better than those chosen by STP in terms of shortest path selection and load balancing. In order to find these trees, we propose the Best Spanning Tree approach that can find the best spanning tree based on the user preferred criteria by using a simple graph theory algorithm.

Here, we define three major criterions: links load balancing (LLB), switches load balancing (SLB) and the shortest path selection (SPS). Here, the shortest path between two nodes means a path with maximum aggregated bandwidth and minimum hop. This guarantees the traffic demands travel along the shortest paths and minimizes latency. In this work, we define three normalized coefficients α, β, γ corresponding to above criterions respectively to indicate the importance of each. Note that $0 \le \alpha, \beta, \gamma \le 1$ and $\alpha + \beta + \gamma = 1$. For example, if your main criterion is LLB, assign $\alpha = 1$, $\beta = \gamma = 0$, but if your goal is to find the best spanning tree based on LLB and SPS criterions but not SLB, assign $\alpha = \gamma = 0.5$, $\beta = 0$.

Now, let us explain our approach in details. We model a metro Ethernet network by a graph $G(V; E)$ where V is the set of N nodes (vertices) representing switches and E is the set of M links (edges) connecting nodes. For simplicity, we show the network graph by a connection matrix A that is N-by-N. Nonzero entries in matrix indicate the presence of an edge. Assume $b_k(k = 1, 2, \ldots, M)$ denotes the bandwidth of kth link, $c_i(i = 1, 2, \ldots, N)$ denotes the switching capacity of ith node (switch) and $d_{ij}(i, j = 1, 2, \ldots, N, i \neq j)$ denotes the traffic demand, that is, the mean rate of customer traffic between nodes i and j. Furthermore, $\alpha, \beta, \gamma\,(0 \le \alpha, \beta, \gamma \le 1, \alpha + \beta + \gamma = 1)$ are user-defined parameters indicate the importance of the criterions: LLB, SLB and SPS, respectively. For each spanning tree, after distributing traffic demands, we can define $l_k(k = 1, 2, \ldots, M)$ as the traffic load on the kth link and $s_i(i = 1, 2, \ldots, N)$ as the traffic load crosses ith switch. In this work, we assume symmetric full duplex links and demands ($d_{ij} = d_{ji}$).

Now, let us define some notations. For a given graph with N nodes, each spanning tree has exactly $(N - 1)$ links. Define the variance of normalized link loads (σ_l^2) as

$$\sigma_l^2 = \frac{1}{N-1}\sum_{k=1}^{N-1}\left(\frac{l_k}{b_k} - \bar{l}\right)^2, \tag{1}$$

where

$$\bar{l} = \frac{1}{N-1}\sum_{k=1}^{N-1}\frac{l_k}{b_k} \tag{2}$$

is the average of normalized link loads. For each spanning tree, σ_l^2 is a useful parameter that indicates the degree of link load balancing. To clarify this, note that l_k/b_k indicates the utilization of kth link. In LLB criterion, we need uniform utilization of links; therefore, the goal is to find a spanning tree with minimum σ_l^2.

Similar to Eq. (1), for a spanning tree we can define the variance of normalized switch loads (σ_s^2) as

$$\sigma_s^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{s_i}{c_i} - \bar{s}\right)^2, \tag{3}$$

where

$$\bar{s} = \frac{1}{N}\sum_{i=1}^{N}\frac{s_i}{c_i} \tag{4}$$

is the average of normalized switch loads. For each spanning tree, σ_s^2 is a useful parameter that indicates the degree of switch load balancing. To clarify this, note that s_i/c_i indicates the utilization of ith switch. In SLB criterion, we need uniform utilization of switches; therefore, the goal is to find a spanning tree with minimum σ_s^2.

In SPS criterion, the goal is to find a spanning tree with maximum aggregated bandwidth and minimum hop count paths. This is equal to select a tree with minimum L defined as

$$L = \frac{\sum_{k=1}^{N-1} l_k}{\sum_{k=1}^{N-1} b_k}. \tag{5}$$

To clarify this, note that $\sum_{k=1}^{N-1} l_k = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq i}}^{N} d_{ij} h_{ij}$ indicates the aggregated load on links. Here, h_{ij} is the hop count from i to j and d_{ij} is the traffic demand between them that is a fixed given value. It is clear that if traffic demands travel on shortest (minimum hop count) paths, $\sum_{k=1}^{N-1} l_k$ will be minimum. On other hand, $\sum_{k=1}^{N-1} b_k$ indicates the aggregated bandwidth of links. Therefore, in SPS criterion, the goal is to find a spanning tree with minimum L. This guarantees each traffic demand travels on shortest path with minimum hop count.

The input parameters of our algorithm for a network with N nodes and M links are connection matrix A, $b_k(k = 1, 2, \ldots, M)$, $c_i(i = 1, 2, \ldots, N)$, $d_{ij}(i, j = 1, \ldots, N, i \neq j)$ and α, β, γ $(0 \leq \alpha, \beta, \gamma \leq 1, \alpha + \beta + \gamma = 1)$.

Our proposed algorithm is as follows:

1. Find all spanning trees of the given network graph. A spanning tree of a graph is a subgraph that connects all nodes and represents a tree. In order to find all spanning trees, we use the enumeration methods introduced by other researchers. A classical well-known method is based on the Cayley's theorem [18]. In this method, we establish a one-to-one correspondence between the set of labeled trees of order N and the set of all ordered symbols $(a_1, a_2, \ldots, a_{N-2})$, where each a_i is an integer satisfying $1 \leq a_i \leq N$. With this method, for a full mesh network, we can find all spanning trees, but if the given network isn't full mesh, don't consider each spanning tree which is not a subset of the given graph. Recently, some other more efficient algorithms have been developed to make the enumeration faster. Although there are numerous algorithms and numerous ways to improve enumeration algorithms, no general technique or framework for their improvements has been proposed. Most enumeration algorithms have been taking advantage of data structures that speed their iterations. The approach proposed by Shioura, *et al.* [19], finds a new spanning tree by exchanging one edge of a current one. This technique has the merit of enabling us to compress the whole output of all spanning trees by outputting only relative changes of edges. This algorithm is optimal in the sense of both time and space complexity [19].
2. Distribute traffic demands on each spanning tree separately and calculate $l_k(k = 1, 2,$

…, M) which represent the data traffic volume on each link and $s_i(i = 1, 2, …, N)$ which represent the data traffic volume on each switch. In a spanning tree, there is only one path between each node pair. By finding these single paths between nodes, we can distribute the traffic on active links of the spanning tree.

3. For a spanning tree, if some links or switches are overloaded, discard that spanning tree. In other words, the following conditions must be valid for each spanning tree:

$$l_k \leq b_k,\ k = 1, 2, \dots, M \text{ and } s_i \leq c_i,\ i = 1, 2, \dots, N.$$

4. Assign three costs C_1, C_2 and C_3 to each spanning tree based on the defined criterions LLB, SLB and SPS respectively by using,

$$C_1 = \sigma_l^2,\ C_2 = \sigma_s^2,\ C_3 = L. \tag{6}$$

Normalize these costs and represent the normalized costs by $\tilde{C}_1$, $\tilde{C}_2$ and $\tilde{C}_3$ respectively;

$$\tilde{C}_i = \frac{C_i}{\underset{j}{\mathrm{Max}}(C_{ij})},\ i = 1, 2, 3 \tag{7}$$

where C_{ij} is the ith cost assigned to jth spanning tree and $\underset{j}{\mathrm{Max}}(C_{ij})$ represents the maximum value among all the C_is assigned to spanning trees.

5. Find the best spanning tree based on the user goals. To do this, multiply each normalized cost obtained in step 4 in its corresponding coefficient and add the results,

$$\mathit{Total\ cost} = (\alpha \times \tilde{C}_1) + (\beta \times \tilde{C}_2) + (\gamma \times \tilde{C}_3). \tag{8}$$

This will rank all spanning trees. Now, select the spanning tree with lowest total cost as the Best Spanning Tree.

In our approach, we evaluate all of the possible spanning trees and then we select the best one with minimum total cost. For a full mesh network with N nodes, there is $N^{(N-2)}$ possible spanning trees that must be evaluated. We can show that the computational complexity of the BST algorithm using Cayley's algorithm is $O(N^N \log N)$. By using shioura's algorithm, for a graph with N nodes, M edges and K spanning trees, we reduced the complexity to $O(N + M + K)$ [19]. As a future work, we are trying to reduce the computational complexity of the BST algorithm by using hierarchical and iterative methods.

4. SIMULATION RESULTS

In this section, we compare the performance of our approach based on different criterions with standard IEEE STP. We implemented our algorithm in MATLAB. Its input parameters for a network with N nodes and M links are connection matrix A, bandwidth vector B which elements are $b_k(k = 1, 2, …, M)$, switching capacity vector C which elements are $c_i(i = 1, 2, …, N)$, traffic demands matrix D which elements are d_{ij}, $i, j = 1, 2, …, N$, $(d_{ii=0})$ and $\alpha, \beta, \gamma\ (0 \leq \alpha, \beta, \gamma \leq 1,\ \alpha + \beta + \gamma = 1)$. The outputs are ranked spanning trees.

For simulation of the standard STP, we use OPNET simulator tool, was chosen because of its comprehensive implementation of Ethernet protocols. In this simulator, for generating and seeking traffic demands, we attach a workstation and a server to each node.

Now we consider a realistic topology for Metro Ethernet Networks (MENs). MENs usually comprise a metro core network and several aggregation and access networks. The task of the core part is to transmit the traffic of the aggregation parts toward the edge nodes. The shape of the core part is usually one or more interconnected rings that are formed by high capacity switches and high speed links. The aggregation part concentrates the traffic of several dozens of Access Nodes to several internal switches; those are connected to the core rings. Traditionally, the access parts are trees, since the cost of building the interconnections are high. For aggregation part, to introduce resilience and traffic engineering capabilities, denser topologies such as rings or dual homing structures are required [2, 20].

Fig. 1 shows a typical topology representative of the Metro Ethernet Networks. It is used also as an evaluation framework in [11, 20]. The core part of this network is formed of four switches with switching capacity of 8 Gbps interconnected into a high-speed ring, where 2 Gbps Ethernet connections are considered. The IEEE 802.3ad Link Aggregation allows the combination of the GbE links into a single virtual link having 2 Gbps of bandwidth.

In this topology, the two Edge Nodes with switching capacity of 8 Gbps are connected to core switches with 2 GbE links and four aggregation nodes with switching capacity of 4 Gbps are connected to two core nodes using dual homing with 1 GbE links. At the bottom layer there are eight Access Nodes with switching capacity of 2 Gbps; each of them is connected to an aggregation switch with 1 Gbps link.

As you can see from Fig. 1, the physical structure of the access part is tree. Therefore, we only consider the core, aggregation and edge parts of the network in our load balancing algorithm as shown in Fig. 2. The labels indicated on links and switches are link numbers and switch names; where, the notation ‘‘Ed”, ‘‘Co”, and ‘‘Ag” stands for Edge, Core and Aggregation, respectively. In this scenario, assume sixteen constant bit rate bidirectional traffic flows between Access nodes and Edge nodes as shown in Table 2; where, the notation “Ac” stands for Access.

The default spanning tree created by standard STP is shown in Fig. 3. This tree is created by OPNET with default values of link costs and switch IDs. On the other hand, the best spanning trees selected by our approach based on LLB, SLB and SPS are shown in Figs. 4-6, respectively.

We can show the effectiveness of the BST approach in terms of link load balancing and switch load balancing by drawing traffic distribution charts as shown in Figs. 7 and 8. As you can see from these figures, the load distribution on links and switches in standard STP is non-uniform, whereas our algorithm based on LLB and SLB criterions distributes traffic more uniformly on links and switches respectively.

As a numerical comparison, we can calculate the defined performance metrics in each case. The results are shown in Table 3. As you can see from this table, the minimum value of σ_l^2 is for LLB, the minimum value of σ_s^2 is for SLB and the minimum value of L is for SPS.

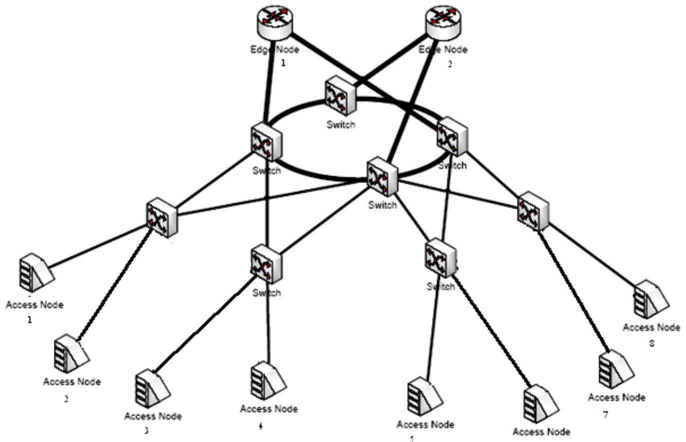

Fig. 1. A typical metro Ethernet network [20].

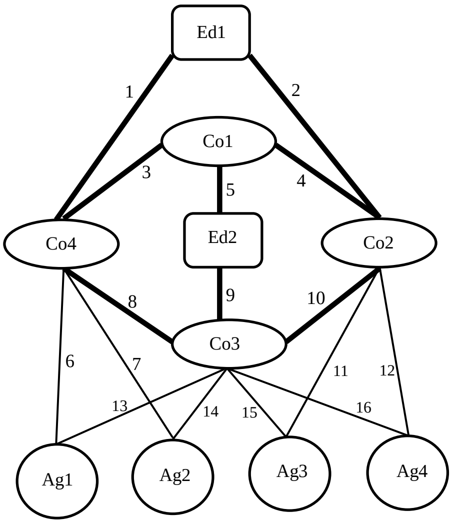

Fig. 2. Metro Ethernet network graph.

Table 2. Traffic demands (Mbps).

	Ac1	Ac2	Ac3	Ac4	Ac5	Ac6	Ac7	Ac8
Edge1	50	50	100	200	200	200	200	100
Edge2	100	100	300	300	200	100	100	200

Table 3. Performance metrics.

	STP	LLB	SLB	SPS
σ_l^2	0.049	0.013	0.025	0.058
σ_s^2	0.023	0.037	0.006	0.017
L	0.53	0.75	0.61	0.46

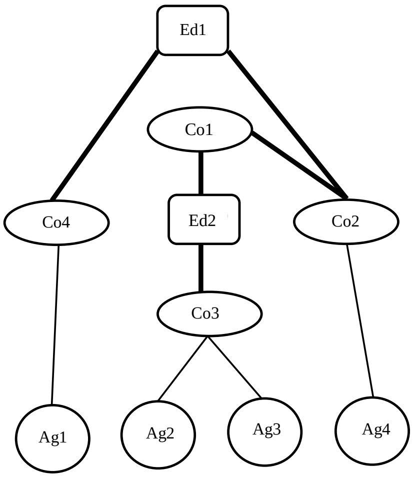

Fig. 3. Spanning tree selected by standard STP.

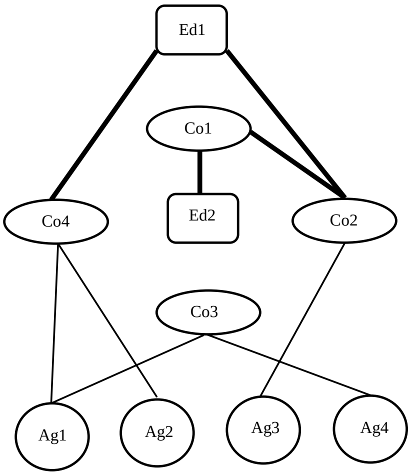

Fig. 4. Best spanning tree selected based on LLB criterion.

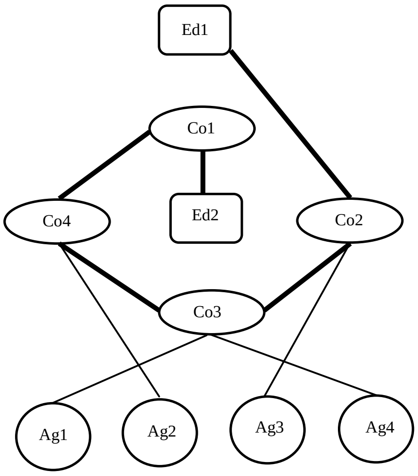

Fig. 5. Best spanning tree selected based on SLB criterion.

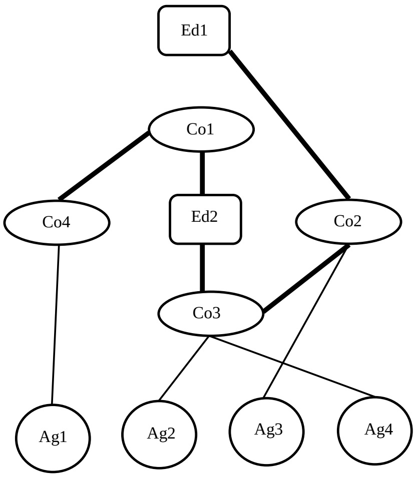

Fig. 6. Best spanning tree selected based on SPS criterion.

5. CONCLUSION

In this paper, we introduced a new graph theoretic approach for ranking all of the possible spanning trees in a metro Ethernet network and for finding the best tree not only based on shortest path selection but also based on load balancing on links and switches. We defined three criterions: load balancing over links, load balancing on switches and shortest path selection and three coefficients corresponding to above criterions respecttively. This allows us to weight the importance of each criterion based on our goal. This algorithm can be used to avoid under-utilization of network resources (links and switches). It is designed to be backward-compatible, as it relies on the IEEE STP protocol.

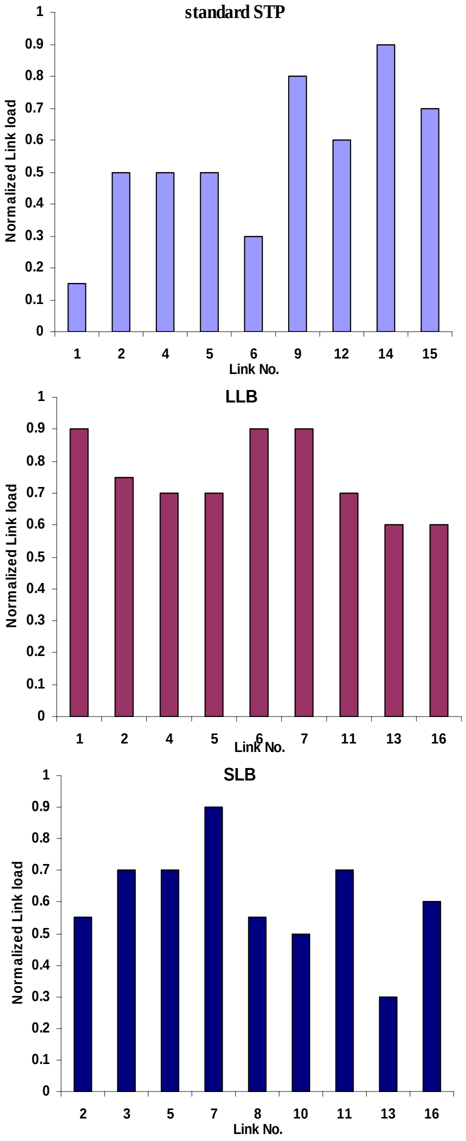

Fig. 7. Normalized traffic distribution on active links.

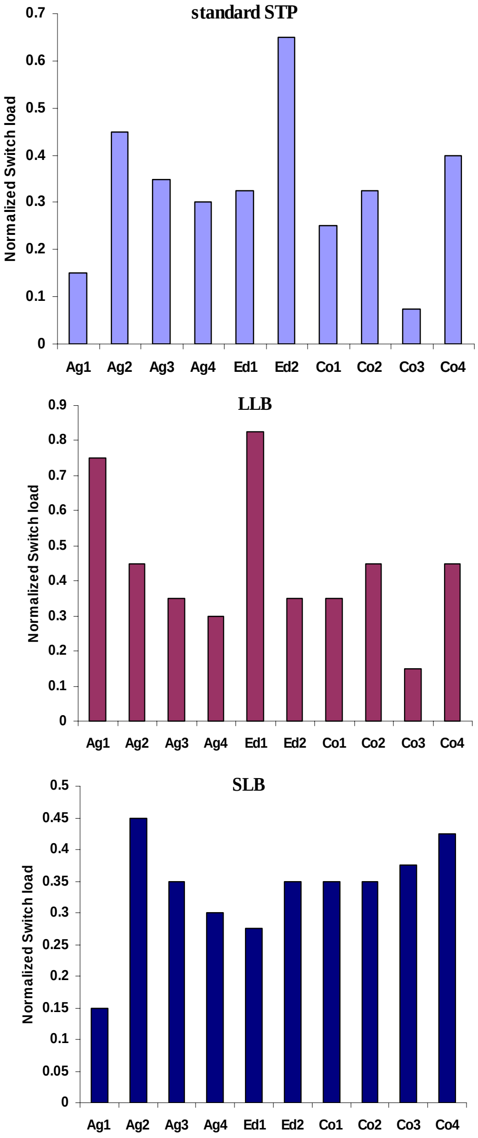

Fig. 8. Normalized traffic distribution on switches.

As a drawback, the proposed approach is basically an exhaustive search algorithm, therefore it is very time consuming, especially when our network is large-scale. For a full mesh network with N nodes, there is $N^{(N-2)}$ possible spanning trees that must be evaluated. We will try to reduce the computational complexity of the BST algorithm by using hierarchical and iterative methods in our future works.

REFERENCES

1. G. Mirjalily, M. H. Karimi Tafti, F. Adibnia, and S. Rajaee, “A new approach to select best spanning tree in metro Ethernet networks,” in *Proceedings of IEEE International Conference on Computer and Information Technology*, 2008, pp. 634-639.
2. M. Huynh and P. Mohapatra, “Metropolitan Ethernet network: A move from LAN to MAN,” *Computer Networks*, Vol. 51, 2007, pp. 4867-4894.
3. R. Seifert and J. Edwards, *The All-New Switch Book: The Complete Guide to LAN Switching Technology*, Wiley Publishing, Inc., Indianapolis, 2008.
4. P. M. V. Nair, “Quality of service in metro Ethernet,” Ph.D. Thesis, Faculty of Computer Science and Engineering, Southern Methodist University, 2006.
5. C. S. J. A. Nash-Williams, “Edge disjoint spanning trees of finite graphs,” *Journal of London Mathematical Society*, Vol. 36, 1961, pp. 445-450.
6. J. Hart, “Extending the IEEE 802.1 MAC bridge standard to remote bridges,” *IEEE Network*, Vol. 2, 1988, pp. 10-15.
7. T. Rodeheffer, C. Thekkat, and D. Anderson, “Smart bridge: A scalable bridge architecture,” *ACM Computer Communication Review*, Vol. 30, 2000, pp. 205-218.
8. K. Lui, W. C. Lee, and K. Nahrstedt, “STAR: A transparent spanning tree bridge protocol with alternate routing,” *ACM Computer Communication Review*, Vol. 32, 2002, pp. 33-46.
9. F. D. Pellegrini, D. Starobinski, M. G. Karpovsky, and L. B. Levitin, “Scalable, distributed cycle-breaking algorithms for gigabit Ethernet backbones,” *Optical Networking*, Vol. 5, 2006, pp. 122-144.
10. IEEE 802.1s, Standards for Local and Metropolitan Area Networks, 2002.
11. M. Huynh, P. Mohapatra, and S. Goose, “Spanning tree elevation protocol: Enhancing metro Ethernet performance and QoS,” *Computer Communications*, Vol. 32, 2009, pp. 750-765.
12. S. Sharma, K. Gopalan, S. Nanda, and T. Chiueh, “Viking: A multi-spanning-tree Ethernet architecture for metropolitan area and cluster networks,” in *Proceedings of IEEE International Conference on Computer Communications*, 2004, pp. 2283-2294.
13. M. Ali, G. Chiruvolu, and A. Ge, “Traffic engineering in metro Ethernet,” *IEEE Network*, Vol. 19, 2005, pp. 10-17.
14. F. Faghani and G. Mirjalily, “Traffic engineering in Ethernet networks by using shortcut switching strategy,” in *Proceedings of International Conference on Future Computer and Communications*, 2009, pp. 371-375.
15. F. Faghani and G. Mirjalily, “An analytic comparison of shortcut switching strategy and spanning tree protocol,” in *Proceedings of International Conference on Communication Systems, Networks and Applications*, 2009, pp. 619-623.
16. F. Faghani and G. Mirjalily, “Selecting the best spanning tree in metro Ethernet networks using genetic algorithm,” *International Journal of Computer Science and Network Security*, Vol. 8, 2008, pp. 106-112.
17. F. Faghani and G. Mirjalily, “A new Ethernet switching method based on extended forwarding topology,” in *Proceedings of International Conference on Future Computer and Communications*, 2009, pp. 365-370.
18. B. Y. Wu and K. Chao, *Spanning Trees and Optimization Problems*, Chapman and Hall/CRC, Florida, 2004.

19. A. Shioura, A. Tamura, and T. Uno, "An optimal algorithm for scanning all spanning trees of undirected graphs," *SIAM Journal on Computing*, Vol. 26, 1997, pp. 678-692.
20. A. Kern, "Traffic-driven optimization of resilient QoS-aware metro Ethernet networks," Ph.D. Thesis, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, 2007.

Ghasem Mirjalily was born in Iran in 1969. He received the B.Eng. degree from Mashhad University, Iran, in 1991 and the M.S. degree from K. N. Toosi University of Technology, Iran, in 1994, both in Electrical Engineering and with honors. In 2000 he completed the Ph.D. program at Tarbiat Modarres University, Iran. He has been a visiting researcher at the Communications Research Laboratory, McMaster University, Canada in 1998. Since fall 2001, he has been with Yazd University, Iran, where he is an Associate Professor. His current research interests include traffic engineering, metro Ethernet networks and wireless sensor networks. Dr. Mirjalily is a senior member of IEEE.

Mohammad Hadi Karimi Tafti was born in Iran in 1983. He received the B.Eng. and M.S. degree from Yazd University, Iran, in 2005 and 2008 respectively, both in Electrical Engineering. He is currently an instructor and faculty member in Department of Electrical and Computer Engineering, Islamic Azad University, Mehriz Branch. Mr. Karimi Tafti is a member of International Iranian Innovators elites Institute (IIIEI). His current research interests include networking, traffic engineering and metro Ethernet networks.

Shahram Rajaee was born in Iran in 1982. He received the B.Eng. and M.S. degree from Yazd University, Iran, in 2005 and 2008 respectively, both in Electrical Engineering. He has worked on wireless communication networks, image processing and neural networks.