

Enhanced Forwarding Strategy in Metro Ethernet Networks

Farhad Faghani*, Abbas Rastgou*, Ghasem Mirjalily**

*Department of Electrical Engineering, Najafabad Branch, Islamic Azad University, Isfahan, Iran.

** Faculty of Electrical and Computer Engineering, Computer and Communication Networks Research Group, Yazd University, Yazd, Iran.

faghani@iaun.ac.ir, abbasrastgou@gmail.com, mirjalily@yazduni.ac.ir

Abstract— Spanning Tree Protocol (STP) is an approved standard protocol that provides a tree forwarding topology in Ethernet networks. Tree topology ensures a unique path from any node to every other node in the network. One of the main disadvantages of STP is its poor load balancing capabilities that results in uneven load distribution and bottlenecks. In our previous work, we have suggested Shortcut Switching Strategy (SSS) to overcome this problem. SSS improves load balancing by using some blocked links to forward traffic. In this paper, we propose a novel forwarding strategy named Enhanced Forwarding Strategy (EFS) that enhances the performance of SSS. Simulation results show that this new strategy decreases the traffic load on links and switches and has a good load balancing performance.

Keywords— Ethernet Networks, Forwarding Strategy, Load Balancing, Traffic Engineering, Spanning Tree.

I. INTRODUCTION

Ethernet which is known by IEEE 802.3 is the most common LAN technology. In recent years, Ethernet is pushing from local area network environment to metropolitan and wide area network environments [1].

One of the most important protocols in Ethernet networks is Spanning Tree Protocol (STP) known as IEEE 802.1d standard. STP is a link management protocol that provides path redundancy while preventing undesirable loops in the network. If a loop exists in the network topology, it confuses the forwarding algorithm and allows duplicate frames to be forwarded. STP forces certain redundant data paths into a standby (blocked) state. In the case of failure or if link costs change, the STP reconfigures the spanning tree topology and reestablishes the paths by activating the standby links in 30-60 seconds. IEEE solved the slow recovery problem of STP by introducing Rapid STP (RSTP), as defined in IEEE 802.1w standard [2].

In both STP and RSTP, all users 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.

Traffic engineering of Ethernet network using spanning tree is a widely researched topic because of performance issues. In literature, there are many forwarding strategies proposed by researches to overcome the challenge described

above [3,4]. Some well-known solutions are SmartBridge [5], GOE [6], AMSTP [7] and Viking [8].

In our previous works, we have considered the problem of traffic engineering in metro Ethernet networks and we have proposed some solutions [9-11].

In [9], we proposed a strategy to select the best spanning tree (BST) in a given network based on load balancing on links and switches. The main problem of BST is its high computational complexity in large-scale networks. In [10], we proposed a new simple forwarding strategy named Shortcut Switching Strategy (SSS) that uses some of the blocked links to forward some traffic. We analytically proved that SSS has a better load balancing on links and switches versus STP.

In this paper, we propose a new forwarding strategy based on SSS named Enhanced Forwarding Strategy (EFS). EFS uses more blocked links in comparison to SSS to forward traffic loads; therefore decreases the traffic load on links and switches, especially close to the root and has a good load balancing performance. It also reduces the average number of hops.

The reminder of this paper is organized as follows: Section II describes the main features of the STP and SSS. The proposed algorithm is explained in Section III. In Section IV, some simulation results are presented and conclusions are drawn in the last section.

II. SPANNING TREE PROTOCOL AND SHORTCUT SWITCHING STRATEGY

In this section, we briefly explain IEEE 802.1d (STP) and our previous proposed algorithm named Shortcut Switching Strategy (SSS).

A. Spanning Tree Protocol

Ethernet traditionally relies on spanning tree protocol defined in IEEE 802.1d, which inherently provides a loop-free communication among all the nodes. In this protocol, the construction of spanning tree is through exchanges of switch IDs and link costs carried by BPDUs. BPDUs are control messages that are exchanged across the switches within the network. These packets carry information about switches, ports, costs, and priorities. The STP uses the BPDU information to select the Root switch and root ports for the switched network [12].

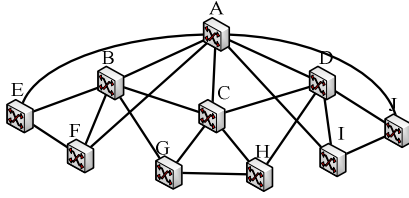


Figure 1. A typical network topology

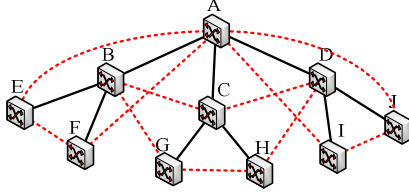


Figure 2. Typical network topology after running STP

In STP, each of the ports of a switch assumes one of the five states, namely: Disabled, Blocked, Listening, Learning, and Forwarding. In addition to port states, STP assigns port roles or functions to each port namely: Root port, Designated port, and Blocked port.

The most important problem of the standard STP is its poor traffic engineering. First of all, we have usually a huge traffic on the links close to the Root. This results in uneven load distribution and bottlenecks, especially close to the root.

The second problem is the big number of hops in some cases, because if two nodes located in the leaf nodes want to communicate with each other, their traffic should pass a long way through the Root [10]. Furthermore, in a typical network, by using standard STP, many of the links will be blocked. This leads to inefficient utilization of expensive fiber links in metropolitan and wide area networks.

For example, consider a full mesh network with N nodes. It has $\frac{N(N-1)}{2}$ Links, but its spanning tree has only $(N-1)$ links. Therefore, the number of blocked (unused) links is:

$$N_{Blocked} = \frac{(N-1)(N-2)}{2} \quad (1)$$

Here, we describe the above mentioned problems by considering the typical network topology shown in Figure 1. Figure 2 shows this topology after running STP. In this figure, the black solid lines are active links and the red dashed lines are the blocked links.

In the network shown in Figure 2, the total number of links is 20, nine of them are active and 11 links are disabled. This means that more than half of the links will be blocked by using STP in this network.

Now suppose node J wants to send data to node F along the spanning tree. By using the active spanning tree, traffic must travel 4 hops through the root node A; while by using the blocked links, the routing path from J to F has only 2 hops.

As a conclusion, using STP is not a good solution in metro Ethernet networks and there is an implicit talent to improve it.

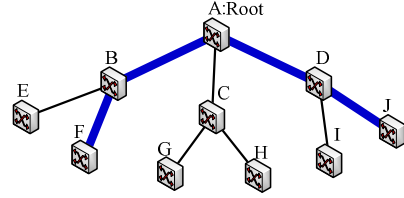


Figure 3. Routing path from J to F by using STP

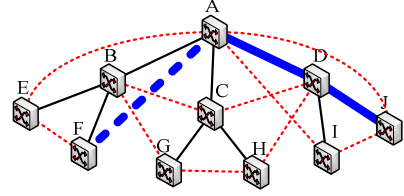


Figure 4. Routing path from J to F by using SSS

In next sub-section, we will describe one of the proposed solutions to improve the performance of STP by using some blocked links.

B. Shortcut switching Strategy

In Shortcut Switching Strategy (SSS), we use the traditional spanning tree constructed by STP to forward frames in a modified manner.

In summary, the address learning mechanism and forwarding table construction of SSS is the same as STP; but, in this new approach, each port learns the addresses of its Blocked Neighbor Nodes (BN-Nodes) by using the source address field in each received BPDU. Blocked Neighbor Nodes are the neighbors that are connected via blocked links. Each node registers its BN-Nodes in a BN-Nodes list.

In SSS, when a switch receives a data frame, it searches its BN-Nodes list and applies these forwarding rules:

- If the destination address is in its BN-Nodes list, it forwards the frame directly to that BN-Node via its blocked port.
- If the address is not available in its BN-Nodes list, it forwards the frame as usual by using its traditional STP forwarding table.

For example, consider Figure 1 and suppose node J wants to send a frame of data to node F along the spanning tree constructed by STP. The routing path is shown in Figure 3 by blue lines. As you can see, the number of hops is 4 hops. Now, Consider Shortcut Switching Strategy. At first J searches its own BN-Node list :{A,I}. Node F is not in the list; therefore, SSS uses traditional STP and forwards the frame to node D. Node D searches its own BN-Node list: {C, H}. Node F is not in the list; therefore, SSS use traditional STP and forwards the frame to node A. Node A searches its own BN-Node list: {E,F,I,J}. Node F is in the list; therefore, SSS forwards the frame to node F directly via blocked link. So, the total number of hops is equal to 3 hops. The forwarding path is shown in Figure 4.

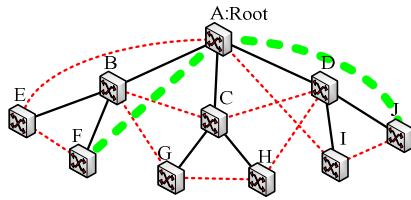


Figure 5. Routing path from J to F by using EFS

As a conclusion, using blocked links can reduce the number of hops. In our proposed forwarding strategy (EFS), we are going to use more blocked links than SSS to reduce the number of hops. In other words, by using SSS we only use one blocked link in routing path; but in EFS, we can use many blocked links in routing path from source to destination. The details will be explained in the next section.

III. ENHANCED FORWARDING STRATEGY

The basic idea in Enhanced Forwarding Strategy (EFS) is identifying the potential blocked links in the path from source to the destination. To do this, we obtain some information about potential shortcuts along the path from delivering the previous frame sent from source to the destination; then we use this information hop-by-hop to deliver current frame on shorter paths by using shortcuts.

In order to explain this process clearly, we divide our new forwarding strategy to two steps: Initialization step and Running step.

In Initialization step, we use the STP or its improved versions such as BST protocol to construct the traditional minimum-cost spanning tree by selecting the proper root and designated ports as usual.

In running step of EFS, for delivering the first frame of a flow¹, we use Shortcut Switching Strategy (SSS). For other frames of the flow, we use a more sophisticated forwarding strategy that uses the information about potential shortcuts.

In EFS, in order to find the potential shortcuts along a path, each node after forwarding the received frame, informs the sender node that which node is selected as the next hop. We call the next hop node as *Potential Next Hop (PNH)* for sender node if it is in the BN-Nodes list of the sender. *PNH* will be saved at the sender node and provides useful information because if there is a blocked link between sender node and next hop node, it can be used as a shortcut to reduce the number of hops.

As an example, consider Figure 2. Suppose node J sends a frame to D. When node D forwards the frame to A, it informs to node J that the next hop is A. A is at the BN-nodes list of D; therefore, node D registers A as its *PNH* and will use this information in next steps.

As explained before, to forward a frame belongs to specific flow, we use the information obtained from delivering previous frame of that flow. In this regard, when a switch receives a data frame:

¹ Flow is a stream of packets from a source node to its corresponding destination node.

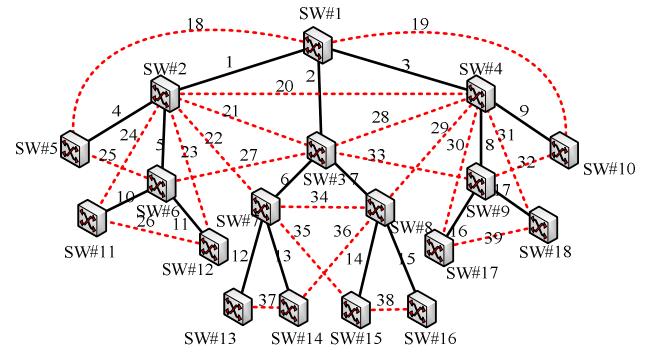


Figure 6. Network topology used in simulations

- searches its BN-Nodes list:
 - If the destination address is in its BN-Nodes list, it forwards the frame directly to that BN-Node via its blocked port.
 - Otherwise, if there is a registered Potential Next Hop, it forwards the frame directly to PNH node via its blocked port. Otherwise, it forwards the frame as usual by using its traditional STP forwarding table.
- Informs the sender node about the selected next hop. Sender will save this address as PNH if it is in its BN-Nodes list.

Here, we consider an example to clarify the process of our proposed forwarding strategy. In Figure 2, suppose that node J wants to send some frames of data to node F. For the first frame, the forwarding is based on SSS. First, J forwards the frame to node D, then node D forwards the frame to node A and informs to node J that the next hop is A. Node J registers A as its PNH. Finally, node A forwards the frame to node F directly via blocked link and informs to node D that the next hop is F. F is not in the BN-Nodes list of D; therefore, F is not registered as the PNH of node D.

For sending second frame from J to F, node J forwards the frame directly to its PNH that is node A. Now, node A forwards the frame to node F directly and informs to node J that the next hop is F. F is not in the BN-Nodes list of J; therefore, F is not registered as the PNH of node J. As a result, the routing path of the other frames will be the same as the routing path of second frame (shown in Figure 5).

It is clear that EFS has all positive features of the Shortcut Switching Strategy (SSS); therefore, we can name the EFS as an extended version of SSS. EFS uses more blocked links than SSS; therefore, we have better load balancing on links and switches.

In next section, through some simulations, we will show some advantages of EFS in comparison to STP and SSS.

IV. SIMULATION RESULTS

We implemented EFS, SSS and STP in MATLAB. The typical topology used in simulations is shown in Figure 6. In this figure, the active links of STP are shown with black solid lines and blocked links are shown with red dashed lines. In this network, the bandwidths of links are assumed the same and equal to 1Gbps and the switching capacity of switches are

also assumed the same and equal to 100Gbps. In our simulations, we set the link costs all the same, and the traffic between each node pair is assumed to be 1Mbps constant bit rate traffic. Note that in Figure 6, the links are identified using number labels (not weights or cost variables).

To compare the results, we use four parameters: average and variance of the normalized link loads, average and variance of the normalized switch loads. The variance of the normalized link loads is defined as [9]:

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

where L is the number of all available links in the network (includes both active and blocked links), l_k is the traffic load on the k th link, b_k denotes the bandwidth of the k th link and:

$$\bar{l} = \frac{1}{L} \sum_{k=1}^L \left(\frac{l_k}{b_k} \right) \quad (3)$$

is the average of the normalized link loads. σ_l^2 is a useful parameter that indicates the degree of link load balancing in the network.

In a similar manner, the variance of normalized switch loads is defined as [9]:

$$\sigma_s^2 = \frac{1}{n} \sum_{i=1}^n \left(\frac{s_i}{c_i} - \bar{s} \right)^2 \quad (4)$$

where n is the number of nodes, s_i is the traffic load on the i th switch, c_i denotes the switching capacity of the i th switch and:

$$\bar{s} = \frac{1}{n} \sum_{i=1}^n \left(\frac{s_i}{c_i} \right) \quad (5)$$

is the average of the normalized switch loads. σ_s^2 is a useful parameter that indicates the degree of switch load balancing.

The simulation results are shown in Figures 7 and 8 and Table 1. As you can see from these figures, the load balancing on links and switches in EFS is better than both SSS and STP. Also, numerical results shown in Table 1 show that EFS can achieve better performance in comparison to STP and SSS. SSS has better performance than STP, because it uses a blocked link as the last hop if possible. In comparison, EFS by using more blocked links than SSS can achieve better results. In EFS, we can use several blocked links on the routing path. This reduces average number of hops and brings us better load balancing on links and switches.

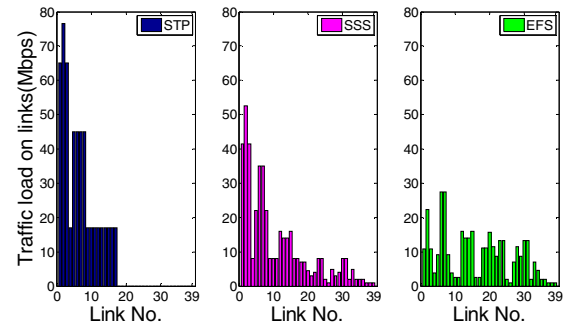


Figure 7. Traffic loads on links.

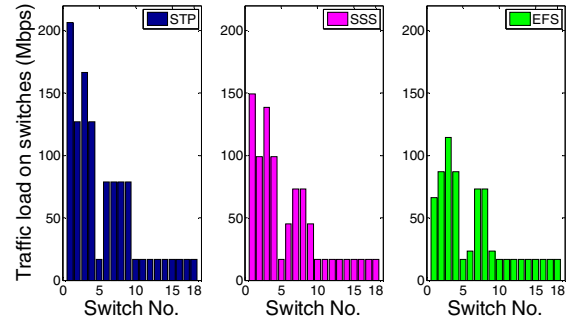


Figure 8. Traffic loads on switches

V. CONCLUSIONS

In this paper, we introduced a new simple and efficient forwarding strategy named *Enhanced Forwarding Strategy (EFS)* that uses some of the blocked links to forward traffic. Our proposed algorithm is designed to be backward-compatible, as it relies on the IEEE STP protocol.

REFERENCES

- [1] M. Huynh, and P. Mohapatra, "Metropolitan Ethernet Network: A move from LAN to MAN", *Computer Networks*, vol. 51, no. 17, pp 4867-4894, 2007.
- [2] CISCO SYSTEM, "Understanding rapid spanning tree protocol," *CISCO white paper*, www.cisco.com, 2006.
- [3] R. C. Sofia, "A survey of Advanced Ethernet Forwarding Approaches", *IEEE Communications surveys & tutorials*, vol. 11, No. 1, pp. 1-23, 2009.
- [4] F. Faghani, G. Mirjalily, R. Saadat, and F. Hendessi, "Toward Benchmarks for Evaluation of Forwarding Strategies in Metro Ethernet Networks," *Proc. of International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)*, pp. 114-119, Bali, Indonesia, 2011.
- [5] T. Rodeheffer, C. Thekkat, and D. Anderson, "Smartbridge: A scalable bridge architecture," *ACM Computer Communication Review*, vol. 30, no. 4, pp. 205-218, 2000.

TABLE 1. COMPARISON BETWEEN STP, SSS AND EFS

Parameter	STP	SSS	EFS	Improvement of EFS vs SSS	Improvement of EFS vs STP
Average of the Link Loads	14.2	11.43	9.19	19.60%	35.28%
Average of the Switch Loads	61.53	49.53	39.82	19.60%	35.28%
Variance of the Link Loads	44.5×10^{-5}	16.1×10^{-5}	4.7×10^{-5}	70.5%	89.3%
Variance of the Switch Loads	3.4×10^{-7}	1.9×10^{-7}	1.0×10^{-7}	46.1%	69.8%
Average of Hops	3.64	2.93	2.35	19.60%	35.28%

- [6] A. Iwata, Y. Hidaka, M. Umaybashi, N. Enimoto, and A. Arutaki, "Global Open Ethernet (GOE) System and its Performance Evaluation," *IEEE Journal Selected Areas Communication*, vol. 22, No. 8, 2004.
- [7] G. Ibanez, A. Garcia-Martinez, and A. Azcorra, "Alternative Multiple Spanning Tree Protocol (AMSTP) for Optical Ethernet Backbones," *Proc. of the 29th Annual IEEE International Conference on Local Computer Networks (LCN'04)*, 2004.
- [8] S. Sharma, K. Gopalan, S. Nanda, and T. Chiueh, "Viking: A Multi-Spanning-Tree Ethernet Architecture for Metropolitan Area and Cluster Networks," *Proc. of INFOCOM*, vol. 4, pp. 2283-2294, 2004.
- [9] G. Mirjalily, H. Karimi, and S. Rajaei, "Load Balancing in Metro Ethernet Networks by Selecting the Best Spanning Tree," *Journal of Information Science and Engineering*, vol. 27, pp. 1747-1759, 2011.
- [10] F. Faghani, G. Mirjalily, "Shortcut Switching Startegy in Metro Ethernet networks," *Computer Communications Journal*, Elsevier, vol. 34, no. 8, pp. 1022-1032, 2011.
- [11] F. Faghani, G. Mirjalily, "Isolated Forwarding Startegy in Ethernet Networks," *Proc. of IEEE 24th International Conference on Advanced Information Networking and Applications (WAINA 2010)*, pp. 152-156, 2010.
- [12] R. Seifert, and J. Edwards, *The All-New Switch Book*, Wiley Publication, 2008.