

QoS-aware Multiple Spanning Tree Mechanism over a Bridged LAN Environment

Yujin Lim, Heeyeol Yu, Shirshanka Das, Scott Seongwook Lee, Mario Gerla
Computer Science Department
University of California Los Angeles
Los Angeles, CA 90095-1596
{yujin, heeyeoly, shanky, sslee, gerla}@cs.ucla.edu

Abstract—Today’s emerging traffic is far removed from the traffic trends seen during the early days of Ethernet technology. As a result, the current IEEE 802.1 standards and its extensions to the Spanning Tree protocol fall short of providing satisfactory quality of service for traffic which has a significant amount of QoS-sensitive multimedia and VOIP traffic. In the current and near-future scenario of campus-wide networks with significantly large layer-2 clusters and numerous VLANs, we show significant shortcomings of the basic Spanning Tree and the Multiple Spanning Tree protocols with regard to QoS. We propose a novel, simple and yet highly effective enhancement to the Multiple Spanning Tree protocol to achieve high degree of QoS by keeping in perspective the different characteristics of the various traffic types in the Diffserv framework. We discuss the problems of the current standards and present in detail our proposed extension to overcome them. Our simulation results show up good improvement in throughput and significant benefits in delay for all classes of traffic to conclusively prove our claims.

I. INTRODUCTION

The campus network has always been at the forefront of networking technology and research; through the birth of the Internet to the current explosion of peer-to-peer traffic. Currently, most campus design models consist either of the flat campus-wide VLANs model or the hierarchical multi-layer model [1]. Both these models involve significantly large layer-2 networks, either throughout the campus for flat models or in the access layer for hierarchical models. Such layer-2 models are attractive not only for their simplicity and plug-and-play style of operation, but also for their better performance and lower cost compared to layer-3 alternatives.

In both flat and hierarchical campus models, there is a lot of reliance on the basic loop free Spanning Tree Protocol for the correct and desirable operation of the network. The legacy IEEE 802.1D/W standard [2], [3] adopts a spanning tree structure to avoid forwarding loops, because a loop in a data link layer is more critical than in a network layer. This approach seemed plausible for the last few decades. However, recently multimedia applications have attracted a lot of attention from the network community, and various data types with different service requirements have emerged. In this environment, the legacy standard approach relying on a single spanning tree brings about the following problems. First of all, because all the traffic is funneled into the branches of the single tree, network congestion takes place although blocked links have plenty of available bandwidth. It brings

about the degradation of service quality for resource intensive and QoS(Quality of Service)-sensitive flows. Thus the overall network becomes underutilized, and the traffic engineering and load balancing issues become very difficult to be solved in such conditions. Secondly, the scheduling discipline in the legacy standard deploys simple priority queuing. In this discipline, because the multimedia traffic is prior to the non-multimedia traffic, non-multimedia flows of lower priority may be starved by multimedia flows of higher priority if multimedia/higher priority flows are a significant fraction of the traffic. This scenario is very true today and typical solutions involve network administrators manually deciding on the spanning tree configuration of different VLANs to perform load balancing. Clearly there is a lot of time and effort that will be saved, if the spanning tree protocol can converge to a good topology for the traffic in consideration without human intervention and this paper addresses this issue.

II. RELATED WORKS

To cope with the inefficiency of the legacy IEEE 802.1D/W standard, the IEEE 802.1S [4] working group investigated the overlay mechanism of multiple trees instead of a single tree. This supplementary approach adds the facility for IEEE 802.1Q [5] VLAN(Virtual LAN) bridges to use multiple spanning trees, providing for traffic belonging to different VLANs to flow over potentially different paths within the virtual bridged LAN. A VLAN is a group of end stations with a common set of requirements and it segments a broadcast domain which is the extent that a broadcast frame propagates through a layer 2 network, independent of physical location. VLANs have the same attributes as a physical LAN but allow to group end stations even if they are not located physically on the same LAN segment.

The other representative approach for coping with network congestion is the STAR(Spanning Tree Alternate Routing) [6] protocol which adopted a forwarding graph in the place of a forwarding tree and improved the performance by using the shortest path. Because the QoS of a flow can be affected by the length of an end-to-end forwarding path, this protocol finds and forwards frames over alternate paths that are provably shorter than the legacy spanning tree paths.

However, in the both of the above approaches, all the different types of traffic (such as voice, video, and so on)

in the same group are still forwarded in the same tree or path. Thus the degradation of service quality and starvation of low prioritized flows still remain.

III. QoS-AWARE MULTIPLE SPANNING TREES

We propose a new QoS-aware multiple spanning tree mechanism for campus-wide LANs which builds spanning trees based on the traffic types and improves the service quality and load-balancing efficacy. To ensure backward compatibility, the mechanism does not require any modification to the legacy standard, but introduces just an extension.

The granularity for multiple spanning trees is one of the important performance factors, because it comes with tradeoff between control traffic overhead and load-balancing. When a single spanning tree is shared with all VLANs as in the legacy standard, we experience network congestion, starvation of traffic with lower priority, and degradation of service quality. On the contrary, if a different spanning tree is built for each VLAN, it may cause extremely high overhead due to control frames for tree construction and could have compatibility problems with the legacy tree. To solve the control frame explosion problem, the IEEE 802.1S standard adopts an aggregated control frame for all spanning trees instead of separate control frames, and provides the capability to build a common spanning tree for a group of VLANs. To solve the compatibility problem it introduces one Internal Spanning Tree (IST) and one or more Multiple Spanning Tree Instances (MSTIs) in a region. The IST receives and sends control frames from/to the outside of region. It is capable of representing the entire region as a virtual bridge to the outside world. The MSTIs are simple spanning tree instances that only exist inside a region and MSTIs never interact with the outside of the region.

The IEEE 802.1S standard does not specify the details of mapping of a group of VLANs to one or more MSTIs. In the QoS-aware mechanism, we propose a mapping policy based on the traffic characteristics. For non-multimedia traffic (e.g. best-effort traffic), a spanning tree for each traffic type is shared between all VLANs to avoid the control frame explosion. In the case of multimedia traffic (e.g. voice and video traffic), a spanning tree is built for each $\langle \text{VLAN}, \text{Traffic type} \rangle$ pair, because both throughput and end-to-end delay are equally important factors in service quality. In the legacy standard, a VLAN represents a set of end stations and a VLAN ID is allocated and advertised over the network by a network administrator or GVRP-like mechanism [5]. In our new QoS-aware mechanism, when a VLAN is created, its traffic type is also advertised along with its VLAN ID.

The standard [2] defines the correspondence between traffic types and user priority values; user priority 0 - best effort traffic, 1 - background traffic, 2 - spare, 3 - excellent effort traffic, 4 - controlled load traffic, 5 - video traffic, 6 - voice traffic, and 7 - network control traffic. For different traffic types, their most relevant cost metrics can be different. For instance, traffic 7 of the highest priority requires that packets be not lost. On the other hand, traffic 5 and 6 are about voice and video traffic and they ask for bounded delay and jitter.

TABLE I
PATH COST PARAMETER VALUES

Link Speed	Recommended Value
$\leq 100\text{Kb/s}$	200000000
1Mb/s	20000000
10Mb/s	2000000
100Mb/s	200000
1Gb/s	20000
10Gb/s	2000
100Gb/s	200
1Tb/s	20
10Tb/s	2

Traffic types 0 to 4 (i.e. non-multimedia traffic) should avoid starvation as much as possible. The legacy standard uses only the interface speed as the path cost metric to build a spanning tree. This static cost metric does not correctly reflect the network status, and the resulting spanning tree is always same unless a link failure happens. To overcome this shortcoming, we adopt the use of additional metrics based on the traffic type as well as interface speed to tune the path cost.

In the QoS-aware multiple spanning tree mechanism, initially a path cost is determined by the interface speed like the legacy standard as shown in Table I. Then the path cost is tuned by the additional metrics according to the traffic type being serviced.

At first, the spanning tree for non-multimedia traffic (e.g. traffic type 0 to 4) is built according to the legacy standard, in other words, it is built based on the interface speed. As mentioned earlier, IEEE 802.1S have one IST and one or more MSTIs. The spanning tree for best-effort traffic (traffic type 0) in the new QoS-aware mechanism plays a role of IST in the legacy standard, in other words it connects with an outside-campus spanning tree, and the other spanning trees for the traffic of the higher priority are related to the MSTIs.

In the case of multimedia traffic (traffic type 5 and 6), the end-to-end delay is one of the most significant factors. But queue information is very dynamic and unpredictable, especially in the simple priority queuing discipline. So we introduce a new metric, the number of active VLANs which indicates the number of actual VLAN flows passing through a bridge. Because the spanning tree spans all the bridges in a network and the actual traffic may flow over only a minor part of the spanning tree, we use the up-to-date entries in the filtering database to know how many VLANs are active through a particular port. Using this information, when a bridge receives a control frame for spanning tree construction (a BPDU frame) for a $\langle \text{VLAN}, \text{Traffic type} \rangle$ pair, the bridge calculates the feasible bandwidth, $F_{bw}(tc)$ and effective bandwidth, $E_{bw}(tc)$ based on the traffic class tc specified in the BPDU.

For each queue i of the same or higher priority than the traffic type in the BPDU,

$$F_{bw}(i) = C - \sum_j (Rate_j \cdot Num_j) \text{ foreach } j \geq i \quad (1)$$

$$E_{bw}(i) = \alpha \cdot C - \sum_j (Rate_j \cdot Num_j) \text{ for all } j \quad (2)$$

$Rate_i$ indicates the average or expected data rate of traffic type i . Usually simple bridges do not have any facilities of throughput calculation. But, because a voice or video traffic is encoded or compressed by the common and general few schemes, we can deduce the average data rate of the voice or video traffic. For instance, the average data rate of a voice traffic is about 90Kbps [7] when using the general encoding scheme such as G.711, G.726, G.729A, or G.723.1. In a video traffic using the general compression scheme such as H.261, H.263, or MPEG1/2, the average rate is about 730Kbps/640Kbps for VBR/CBR, respectively. Num_i and C mean the number of pending VLANs of traffic type i and the interface speed respectively. α is a parameter to limit the bandwidth utilization of the traffic of the higher priority, like Diffserv [8]. In other words, it prevents the multimedia traffic from hogging all the link capacity and starving out the non-multimedia traffic.

As mentioned above, the existing standard adopts a simple priority queuing scheme. So, the service quality of a flow can be affected by the flows of the same or higher priority. The feasible bandwidth, $F_{bw}(i)$ indicates a amount of bandwidth which a flow can occupy even though the flows of the lower priority may be sacrificed. On the other hand, the effective bandwidth, $E_{bw}(i)$ means, the amount of bandwidth that a flow can occupy without the starvation of the best effort traffic or the sacrifice of the existing flows of lower priority. Using these two bandwidth values, the path cost is tuned as follows:

*If $Rate_{BPDU} \leq F_{bw}(tc)$ (and $Rate_{BPDU} > E_{bw}(tc)$),
increase the path cost to the next higher value
in the recommended table .* (3)

*if $Rate_{BPDU} > F_{bw}(tc)$ and $Rate_{BPDU} > E_{bw}(tc)$,
increase the path cost two levels higher
in the recommended table .* (4)

$Rate_{BPDU}$ refers to the average data rate of the traffic type specified in the received BPDU. In the first case, we increase the cost to avoid the sacrifice of the existing flows of lower priority as much as possible. The latter case implies that there is extreme congestion on the link, and it justifies jacking up the path cost by one more level to avoid poor service quality or unlimited end-to-end delay. Increment to the next higher value for instance means that if the interface speed in a link is 10Gbps, the recommended path cost is 2000 as in Table I. Then if a single increment of the path cost is needed, the path cost is increased to the next higher value, 20000. In the case we want to increase it to two levels higher, the path cost is increased to 200000. Because the root path in a spanning tree is selected based on the minimum cost path, through this cost increasing step the spanning tree can avoid links with heavy traffic and reduce the starvation or sacrifice of the lower prioritized traffic.

Our new QoS-aware mechanism builds a spanning tree based on the network status. Since the path cost updates are sent to and from switches every HELLO time (typically 2 seconds), the tree may be changed over time. However this is undesirable until absolutely necessary, since the switching overhead in terms of packets dropped during the convergence of the tree is prohibitive. However doing away with changing the spanning tree may also lead to problems. This situation can come about because best effort traffic is consigned to use the same spanning tree. So, there could be situations where class 5/6 traffic comes in at time t , takes up links which were empty at that time, but part of the dormant BE spanning tree. These links could get overloaded when more BE vlans are created say at time $t + \theta$ and more traffic pours into those links. Since BE traffic is consigned to follow the pre-created common spanning tree, while multimedia traffic has already chosen and pinned down its spanning tree, both classes of traffic suffer. Thus in such a situation it might be viable to change the spanning trees for the class 5/6 vlans, so that both classes benefit in spite of the temporary burst of packet loss due to the change overhead.

Thus we experiment with two QoS-aware schemes, QoS-MST and QoS-DYN-MST. QoS-MST pins down the spanning trees once they are constructed, by never modifying their path cost later. QoS-DYN-MST however keeps a tab on the number of vlans going through the ports even after the tree is constructed. To reduce the switching overhead of the tree, QoS-DYN-MST only changes the path cost when congestion becomes severe. According to the standard, a root sends a BPDU over a network periodically and the bridge which received the BPDU can know whether the BPDU is for a new spanning tree or the existing one. If the BPDU is for the creation of a new spanning tree, the bridge calculates the path cost based on the (3, 4) and if it is for the existing one, it tunes the path cost depending on whether its running QoS-MST or QoS-DYN-MST as follows to keep the existing spanning tree unchanged as much as possible.

*If $Rate_{BPDU} \leq F(bw)$ and $Rate_{BPDU} > E(bw)$,
do nothing.* (5)

*If $Rate_{BPDU} > F_{bw}(tc)$ and $Rate_{BPDU} > E_{bw}(tc)$
and Scheme = QoS – DYN – MST,
Increase the path cost as in (4).* (6)

Finally, traffic type 7, the network control traffic is sensitive to the packet loss. If there is any packet loss in the queue of the same or higher priority, the path cost is increased to the next higher value. Thus, the spanning tree for traffic type 7 can avoid links with packet loss.

IV. PERFORMANCE EVALUATION

To evaluate the performance of our new mechanism, we adopted the Qualnet 3.5 network simulator [9] which has the IEEE 802.1D/W and 802.1Q modules for the spanning tree and vlan mechanism. We implemented the IEEE 802.1S for multiple spanning tree (MST) and our QoS-aware MST

TABLE II
THROUGHPUT FOR EACH TRAFFIC TYPE (KBPS)

10 VLANs				
	Best Effort		Video	
Dist.	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST
20 / 70 / 10	0.019	0.003	5100	335
30 / 60 / 10	1570	1570	941	941
45 / 45 / 10	0.01	0.01	12	12
20 VLANs				
	Best Effort		Video	
Dist.	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST
20 / 70 / 10	-4400	-64	5260	483
30 / 60 / 10	-3260	322	1570	611
45 / 45 / 10	693	693	0.157	0.157
30 VLANs				
	Best Effort		Video	
Dist.	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST
20 / 70 / 10	-2700	-404	6900	554
30 / 60 / 10	-3800	-1800	1000	757
45 / 45 / 10	180	909	40	-1000

modules, and compared the performance between both of them in terms of the throughput and end-to-end delay.

In the grid topology with 16 bridges and 10Mbps links, we adopted four types of traffic (best effort, video, voice, and network control), which are generated by CBR and VOIP traffic generator. In the variance of the distribution between the four types of traffic over a whole network, we measured the performance for each type of traffic.

The table shows the improvement experienced by the different traffic distributions for each traffic class when compared with the regular single Spanning Tree ST (802.1D/W) implementation. Its obvious that both MST (802.1S), QoS-MST and QoS-DYN-MST do much better than ST. Thus the schemes that are really interesting in terms of comparison are MST (802.1S), QoS-MST and QoS-DYN-MST.

The remaining tables therefore are devoted to comparing the improvement experienced by the different traffic distributions for the QoS-MST and the QoS-DYN-MST scheme, when compared with simple MST (802.1S). We used 10, 20 and 30 vlans, to saturate the network completely. The Table II and III show the throughput and end-to-end delay for each traffic class in variance of the traffic distribution. In the case of network control traffic of the highest priority, it is never sacrificed by the other traffic and it is rarely dropped except when a link is congested by the other network control traffic. Thus, we focused on the performance of best effort, video, and voice traffic. In the case of voice traffic (VOIP), because its throughput is small and not a significant factor, we devoted to comparing the throughput of best effort and video traffic. The traffic distribution indicates the portion of the each type of traffic over the whole network traffic. For instance, 20/70/10 means 20% for best effort traffic, 70% for multimedia traffic (voice and video traffic), and 10% for network control traffic. If the portion of the multimedia traffic is so small compared with the non-multimedia traffic, it cannot affect the starvation

of best effort traffic, so we kept the amount of multimedia traffic not so small to clarify the difference of performance.

The tables show clearly that both QoS-MST and QoS-DYN-MST show clear improvements over simple MST. A few results are quite interesting and validate our intuition. For example, for 20 vlans, traffic distribution 20/70/10 and 30/60/10, we see that BE traffic actually suffers much more in case of QoS-MST than in the case of QoS-DYN-MST, while Multimedia traffic gains much more in QoS-MST compared to QoS-DYN-MST. This is because, as we earlier said, QoS-MST pins down the spanning tree, thus traffic class 5 and 6, get a good tree, and hog it, leading to starvation of BE traffic that comes in later. On the other hand, QoS-DYN-MST, continuously monitors and changes the spanning tree if needed for traffic classes 5 and 6. This leads to better sharing of bandwidth between the various classes. In fact, in the 30/60/10 distribution for 20 vlans, we see that both BE and multimedia traffic gain, compared to MST, while for QoS-MST, multimedia traffic gains a lot, but BE traffic is sacrificed.

V. CONCLUSION AND FUTURE WORK

We explained the significant shortcomings of the basic Spanning Tree and the Multiple Spanning Tree protocols with regard to QoS. The Multiple spanning tree protocol achieves only geographical load balancing, and doesn't take into account the varied characteristics of the traffic passing through it apart from the simple priority queuing technique. We detail a novel, simple and yet highly effective enhancement to the Multiple Spanning Tree protocol to achieve high degree of QoS. We achieve this by keeping in perspective the different characteristics of the various traffic types in the Diffserv framework. The details of the scheme, make it clear that its very practical and can be put into implementation and practice with minimal effort.

TABLE III
END-TO-END DELAY FOR EACH TRAFFIC TYPE (MS)

10 VLANs						
	Best Effort		Video		Voice	
Dist.	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST
20 / 70 / 10	-0.02	-0.07	22	11	28	23
30 / 60 / 10	-0.03	-0.03	17	17	0.61	0.61
45 / 45 / 10	0.38	0.38	0.33	0.33	0.39	0.39
20 VLANs						
	Best Effort		Video		Voice	
Dist.	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST
20 / 70 / 10	-60	-5	23	11	75	-1.7
30 / 60 / 10	-172	-27	11	0.2	29	12
45 / 45 / 10	36	36	0.01	0.01	9.6	9.6
30 VLANs						
	Best Effort		Video		Voice	
Dist.	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST	QoS-MST	QoS-DYN-MST
20 / 70 / 10	135	142	58	15	126	72
30 / 60 / 10	-430	-107	4.4	1.5	15	-34
45 / 45 / 10	52	77	0.26	-2.0	91	53

We look at two variations of the scheme to validate our intuition, that the spanning tree should be changed when there is extreme congestion. Our simulation results show much improvement in throughput and significant benefits in delay for all classes of traffic to conclusively prove our claims. Using two variations of the tree updating scheme, we show the tradeoff between the gains of multimedia traffic and the loss of the non-multimedia traffic. we plan to emulate our scheme over a testbed [10] to further prove our point about its practical feasibility.

REFERENCES

- [1] Kennedy Clark and Kevin Hamilton, CCIE professional development: Cisco LAN switching - The most complete guide to Cisco Catalyst switch network design, operation, and configuration. Cisco Press.
- [2] IEEE 802.1D, Standard for local and metropolitan area networks: media access control (MAC) bridges, 1998.
- [3] IEEE 802.1W, Standard for local and metropolitan area networks: media access control (MAC) bridges - Amendment 2: Rapid configuration, 2001.
- [4] IEEE 802.1S/D15, Draft standard for local and metropolitan area networks: amendment 3 to 802.1Q virtual bridged local area networks: multiple spanning trees, 2002.
- [5] IEEE 802.1Q, Standard for local and metropolitan area networks: virtual bridged local area networks, 1998.
- [6] K. S. Lui, W. C. Lee, and K. Nahrstedt. STAR: a transport spanning tree bridge protocol with alternate routing. ACM SIGCOMM Computer Communications Review, vol.32(3), pp22-46, July 2002.
- [7] M. J. Karam and F. A. Tobagi. On traffic types and service classes in the Internet. IEEE GLOBECOM, pp548-554, Nov. 2000.
- [8] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An architecture for differentiated services. IETF RFC 2475, Dec. 1998.
- [9] Scalable Network Technologies (SNT). <http://www.scalable-networks.com>.
- [10] A Framework of QoS emulator. <http://www.cs.ucla.edu/NRL/hpi/qmulator/index.html>.