Optimizing QoS Aware Ethernet Spanning Trees

Tibor Cinkler, István Moldován, András Kern, Csaba Lukovszki, Gyula Sallai
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Magyar tudósok krt. 2, Budapest, Hungary H-1117
email: {cinkler, moldovan, kern, lukovszki, sallai}@tmit.bme.hu

Abstract—Ethernet is gaining on its importance in both access and metro networks. As a layer 2 technology, Ethernet gives a basic framework for routing, QoS and Traffic Engineering (TE), as well as a protocol for building up trees. IEEE 802.1 standards define default configuration parameters considering the topology only.

In this paper we propose methods for resource management in Ethernet networks through spanning tree optimization for both STP (IEEE 802.1D) and MSTP (IEEE 802.1s). As a result of optimization we assign costs to the bridge ports in the network to build trees based on these costs via STP and MSTP. These trees yield optimized routing, TE and support for different QoS classes.

We show on typical metro-access networks, that through optimization the total network throughput can be significantly increased for both cases: when enforcing fairness or when allowing starvation of some demands. This gain can be realized by simultaneously assigning demands to trees and routing these trees.

I. INTRODUCTION

The decreasing price trend for Ethernet equipment has made this technology a very attractive alternative when building access networks. In fact, Ethernet is the most cost effective choice for aggregating customer traffic when focusing on cost per bandwidth units. However, the basic Ethernet has been developed for LAN environment, and some important features are missing for carrier grade service. Several vendors and service providers realized the opportunities laying in the Ethernet technology, and now standardization bodies and forums are constantly extending its limits.

The typical access network calls for quality of service and better traffic control. A simple class-based quality of service is provided by the IEEE 802.1Q [1] standard, which adds 8 QoS classes to the Ethernet. This standard also increases the scalability by segmenting the network into independent Virtual LANs (VLANs), each representing a different broadcast domain.

Resilience and loop protection is also required. The IEEE 802.1D Spanning-Tree Protocol (STP) [1] is developed to address this problem. STP is responsible for building a loop-free logical forwarding topology from a meshed physical topology, while providing connectivity to all nodes. However, in case of link failure or topology change, STP requires 30 to 60 seconds to detect the changes and reconfigure trees, which significantly reduces the network performance. Rapid STP (RSTP) [2] achieves faster convergence by relying on an active bridge-to-bridge handshaking mechanism instead of depending on timers specified by the root bridge as in 802.1D.

A real extension made by Multiple Spanning-Tree Protocol (MSTP) [3] allows several independent spanning tree instances for different groups of VLANs in the network. MSTP also introduces the concept of regions. Each region has its own VLAN assignment, and in each region multiple spanning tree (MST) instances can be defined. The role of MST instances is to optimize the network utilization, while MST regions can be used to increase the scalability of the network and also to improve the reaction time in case of failures.

In this study we address the design of an Ethernet access network focusing on QoS and optimal network utilization. First, the operation of the Spanning Tree Protocol is briefly reviewed in Section II. After that we overview the enhancements of the Rapid Spanning Tree Protocol and of the Multiple Spanning Tree Protocol. In Section III we formulate the problem and propose the optimization method. In Section IV we show the evaluation of the methods and the numerical results. Finally we conclude the paper in Section V.

II. BACKGROUND

The Spanning Tree Protocol [1] is the basic bridge protocol developed originally to avoid loops in the bridged network. The protocol builds a tree that spans all bridges in the network and data is propagated along this tree. The links that are not part of the tree are blocked. In the case of failure, the blocked links are activated providing a self-healing restoration mechanism for the bridged network. The main drawbacks of this algorithm are the long recovery times in case of failure and the inability to use the backup links for forwarding in case of no failures.

The long recovery times are reduced by the Rapid Spanning Tree Protocol [2][4]. It is based on STP and it has some major improvements: it uses proposal-agreement based handshaking mechanism to repair the connectivity in case of failures. After a port receives the agreement in reply of a proposal sent earlier it will be used immediately. This handshake propagates quickly towards the edge of the network and restores connectivity after a topology change.

Both the STP and RSTP use a single spanning tree topology over the network, and forward packets of all VLANs along
new Service VLAN (S-VLAN) tag. The S-VLAN tag is used in the metro network to identify services at the edge nodes, and also to provide Traffic Engineering (TE). Customer traffic is assigned to S-VLANs based on service class and destination edge node.

In contrast to the topology driven approach, here we propose a Traffic Engineering method to build up the trees and assign the VLANs in “traffic driven” manner where the goal is to optimize VLAN assignments and trees ensuring QoS requirements and network utilizations. After an optimization step weights are assigned to the links of the Ethernet network: the same low link weights to edges to be used by the considered tree while the same high weights to edges not to be used.

A. Notation, Definitions, Assumptions and the Proposed Graph Model

Let us assume that the network is modeled as a directed graph $D(N,E,B)$, where $N$ is the set of nodes (vertices) where the Ethernet bridges are placed, $E$ is the set of edges (links) connecting the nodes, and $B$ is the bandwidth (capacity) of these links. A link $l \in E$ is defined as $l = (i,j)$ where $i,j \in N$.

The C-VLANs or demands $o \in O$ are assumed to originate from access nodes (sources: $s(o)$) that are e.g., Ethernet segments or Digital Subscriber Line Access Multiplexers (DSLAMs). The destinations ($d(o)$) of demands $o$ are the edge nodes that connect the access to the backbone. They are also the roots of the trees. Implicitly, we have made here two assumptions. First, without loss of generality we assume that these demands can be connected to each other only through root nodes. Second, we assume demands always directed from the access to the edge nodes, however we will allocate symmetrical capacity (bandwidth $b(o)$) in both directions.

The demands belong to different traffic (or QoS) classes $c \in C$ having different requirements. Since there are 3 bits in the Ethernet header for this purpose, up to $|C| = 8$ traffic classes can be differentiated. For simplicity reasons we use only four classes as shown in Table I. We assign the highest weight ($w^3 = 4$ in our case) to the highest priority traffic class, however, it will have the lowest volume, therefore the lowest ratio of link resources ($\beta^3 = 10\%$ in our example) is assigned to it. Each demand $o \in O$ belongs to one traffic class $c \in C$, therefore, set $O$ consists of $|C|$ subsets $O = O^1, O^2, \ldots, O^{|C|}$.

S-VLANs carry the traffic from the access to the edge node, thus, they correspond to a demand or a set of demands that have the same destination and traffic class. Under VLAN assignment we implicitly assume assigning a C-VLAN to a S-VLAN. Thus, we refer to the S-VLANs simply as VLANs.

The trees in an STP or MSTP based Ethernet network define how to route traffic. Each tree has only one root residing in the edge nodes. There can be up to 64 trees $t_i \in T$ in a region ($|T| \leq 64$) of MST. All the nodes that are neither roots nor leaves are referred to as intermediate nodes.
TABLE I
THE TRAFFIC (QoS) CLASSES

<table>
<thead>
<tr>
<th>Number</th>
<th>Name</th>
<th>Resource Ratio</th>
<th>Priority Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Platinum</td>
<td>β1 = 15%</td>
<td>w1 = 4</td>
</tr>
<tr>
<td>2</td>
<td>Gold</td>
<td>β2 = 20%</td>
<td>w2 = 3</td>
</tr>
<tr>
<td>3</td>
<td>Silver</td>
<td>β3 = 30%</td>
<td>w3 = 2</td>
</tr>
<tr>
<td>4</td>
<td>Best Effort</td>
<td>β4 = 40%</td>
<td>w4 = 1</td>
</tr>
</tbody>
</table>

In case of STP all VLANs follow the same spanning tree (|T| = 1). MSTP requires unique VLAN assignment to the MST instances within regions. In our case, we consider only one region for the whole access network. The assignment of trees and VLANs is based on the destination of the demands: a VLAN belongs to a tree if the destination of the demands forming the VLAN is the root node of the tree. Thus, let \( t(o) \subseteq T \) be the available trees for demand \( o \in O \). If only one tree belongs to each VLAN (\(|t(o)| = 1\)) the assignment of the VLANs to the trees is straightforward; otherwise distributing the VLANs among the trees is needed. When making the assignment both, the traffic engineering and the QoS requirements will be considered.

B. ILP Formulation

Here we give the objective and the constraints of the Integer Linear Programming (ILP) formulation.

a) Variables: The input parameters are: the network \((D,N,E,B)\), the sets of demands \( O \), the trees \( T \) and the demand classes \( C \). The solution consists of edges forming the paths of the VLANs and of the trees. Both are described with binary indicator variables:

\[ x_{ij}^{o,t} \] is 1 iff demand \( o \in O \) uses tree \( t \in T \) on link \( l \in E \).

\[ y^t_i \] is 1 iff tree \( t \in T \) uses link \( l \in E \).

b) The objective: (see Eq. 1) is to minimize the weighted sum of used resources for all links in the network. The weighting parameter \( \alpha \), where \( 0 < \alpha \leq 1 \) is used to minimize the number of links used by a tree (for larger \( \alpha \)) or to minimize the total capacity used (for smaller \( \alpha \)). \( w^o \) is used to make even shorter paths for demands of higher priority.

\[
\text{min} \sum_{(i,j) \in E} \left[ \alpha \sum_{v \in T} y^t_i + (1-\alpha) \frac{1}{B_{ij}} \sum_{o \in O, v \in T} w^o x_{ij}^{o,t} \right]
\]

Subject to:

c) Capacity Constraints \((\forall (i,j) \in E)\): We have similar capacity constraints for each class (Eqs. 2-4) guaranteeing that each class uses only its dedicated resources. However, the best effort (BE) traffic may use any resources that are not used by higher classes, therefore we sum up all the demands including BE traffic and force them to fit into the link capacity (Equation 5).

\[
\sum_{v \in O} \left( x_{ij}^{o,t} + x_{ji}^{o,t} \right) b^o \leq \beta^1 \cdot B_{ij} \quad \forall (i,j) \in E
\]

\[
\sum_{v \in O^2} \left( x_{ij}^{o,t} + x_{ji}^{o,t} \right) b^o \leq \beta^2 \cdot B_{ij} \quad \forall (i,j) \in E
\]

d) Flow Conservation \((\forall o \in O, \forall i \in N)\): The flow conservation constraints guarantee that a demand goes from its source to its destination, and if it enters any other node it has to leave it as well. However, VLAN assignment has to be given for all trees that are assigned to a certain demand (Eq. 8). The selection of the used tree is performed at the source and destination nodes (Eqs. 6 and 7).

\[
\sum_{v \in T(o)} \left( \sum_{j \in N^{i\rightarrow}} x_{ij}^{o,t} - \sum_{j \in N^{i\leftarrow}} x_{ji}^{o,t} \right) = 1, \quad \forall i \in N \quad \text{if } i = s(o),
\]

\[
\sum_{v \in T(o)} \left( \sum_{j \in N^{i\rightarrow}} x_{ij}^{o,t} - \sum_{j \in N^{i\leftarrow}} x_{ji}^{o,t} \right) = -1, \quad \forall i \in N \quad \text{if } i = d(o),
\]

\[
\sum_{j \in N^{i\rightarrow}} x_{ij}^{o,t} - \sum_{j \in N^{i\leftarrow}} x_{ji}^{o,t} = 0, \quad \forall t \in T(o) \quad \forall i \in N, \text{otherwise},
\]

where \( N^{i\rightarrow} \) is the set of nodes \( j \in N \) for which \((i,j) \in E\), and \( N^{i\leftarrow} \) is the set of nodes \( j \in N \) for which \((j,i) \in E\).

e) Demand – tree mapping constraints: A demand can be carried over a link only if the tree that carries that demand is set up over that link.

\[
x_{ij}^{o,t} \leq y^t_i \quad \forall l \in E, \forall o \in O, \forall t \in T(o).
\]

f) Tree conservation constraints: The root of a tree may only be at the edge node and the leaves of a tree may be only at access nodes:

\[
\sum_{v \in E^{N^t \rightarrow}} y^t_i = 0 \quad \text{if } N \text{ is root of } t \\
\sum_{v \in E^{N^t \rightarrow}} y^t_i \leq 1 \quad \text{otherwise}
\]

\[
y^t_i \leq \sum_{v \in E^{N^t \rightarrow}} y^t_i \quad \forall i \in N \setminus \{EN_t\}, \forall k \in N^{i\rightarrow}, \forall t \in T
\]

\[
y^t_i \leq \sum_{v \in E^{N^t \leftarrow}} y^t_i \quad \forall i \in N \setminus \{AN_t\}, \forall k \in N^{i\leftarrow}, \forall t \in T
\]

EN and AN are the sets of the edge and access nodes assigned to the tree \( t \).

C. The Three Considered Optimization Cases

The above ILP formulation we gave is a generic one that covers the following three cases.

1) Optimized Spanning Tree Protocol (STPopt): Here we still assume one tree per network analogously to STP, however, instead of the topology driven approach we assume a traffic driven one by setting \(|T| = 1\). Here we assume that all demands of all traffic classes have to be carried by this single tree.
2) Optimized MSTP with 1 tree per root (MSTP_{opt1}): Here we assume that there are multiple trees and there is one tree for each edge node. To each tree we assign S-VLANs for each class. C-VLANs are assigned to the S-VLANs based on the destination and class. Note, that a single leave can have multiple demands routed over different trees to different roots enabling that the traffic is better distributed between more links without the risk of having a loop in any of the trees.

3) Optimized MSTP with 2 trees per root (MSTP_{opt2}): In this case we assume that there may be two (or even more) trees per each edge node. To each tree we assign S-VLANs for each class, but the C-VLANs are assigned to the S-VLANs in an optimal way. This allows that a single link can carry both trees that can later on branch again (use different links). This approach supports TE better. Here we do not only route the trees, but we simultaneously assign the VLANs to these trees!

IV. NUMERICAL RESULTS

In this Section we study the performance of the three proposed optimization methods (STP_{opt}, MSTP_{opt1} and MSTP_{opt2}), and compare them to the two reference methods (STP and MSTP). For the reference methods the default settings are used (link capacity based port costs) as the standard proposes. The only restriction is that the root of the trees will be one of the edge nodes.

A. Evaluation Criteria

Most of the selected criteria focus on the quality of the given solutions. Among them the available throughput is the most important criterion, since the more demands can be transmitted at the same time, the higher income can the network provider realize. However, as the importance of value added applications grows, the requirement of not violating the QoS becomes more crucial. The maximal throughput, therefore, is determined in two ways:

- **Fair maximal throughput** where the ratios of the demand sizes are fixed, and
- **Greedy maximal throughput** where starvation of several demands is allowed to increase the total throughput.

Besides the throughput it is important to consider the question of how much capacity is allocated. Longer paths require more overall allocated capacity. To measure it total allocated capacity is evaluated as well.

B. Test Cases

Since we focus on the area of Ethernet based Metro Access Networks we make certain assumptions on the network topologies (see Figure 1). It consists of a core and several access parts. The core is usually realized by one or more interconnected rings. The edge nodes are connected to the ring as well. The access parts concentrate the traffic from several access nodes to one or more switches of the core part. It has either a tree or dual-homing structure. This latter one is capable of load balancing or improved fault tolerance.

Considering these assumptions three different networks are used: the small (12 nodes) and the medium-sized dual-homed networks (18 nodes), and the tree-like network (like Fig. 1). The link capacities in the core part are equal to 1 Gbps while in the access part to 100 Mbps.

Demands are defined between each access and edge node pair for every QoS class. The demand sizes follow the standard normal distribution where the mean value is the product of the global traffic level and of a QoS class traffic ratio parameter that expresses the average demand sizes.

C. Obtained Results

In each test case 12 independent problem instances are generated. The results presented are averaged of these 12 simulations. The relative variances of the statistics are below 8%.

a) **Fair and greedy throughput comparison:** Figures 2(a) and 2(b) depict the total fair and greedy throughputs for the small and medium-sized topology respectively. They clearly show the dramatically increased throughput produced by optimization. The difference between MSTP_{opt1} and MSTP is twofold for both results, while difference between (MSTP_{opt1}) and (MSTP_{opt2}) is negligible. However, negligible optimization gain is achieved when tree-like topology is investigated (Figure 2(c)) that comes from the fact that homogenous traffic is assumed thus the capacities of the bottleneck links limit the total available throughput.

These results show that the network topologies, the link capacities and the places of bottlenecks determine the available throughput according to the result obtained. To validate it two further test cases were derived from the medium-sized topology: the capacity restricted one (the core part links are set to 400 Mbps) and high bandwidth one (all links are set to 1Gbps).

The results are shown in Figure 3. As one can see the throughput of the simple and the optimized MSTPs compared to STP are twofold and fourfold again, only the performance of the optimized STP fluctuates between STP and MSTP. This shows that the performance of the STP_{opt} highly depends on the place of the bottlenecks.

b) **Capacity allocation and average path length per QoS class:** The allocated capacity in the network is influenced by the offered load and by the length of the paths used for transmission. To characterize the effectiveness of the methods, i.e., how much capacity is allocated to transmit the traffic, two further measures are introduced: average used link utilization and maximal link utilization.
trees assigned to the two edge nodes can use different paths to transmit the traffic in the access parts which increases the throughput.

The major drawback of the “topology-driven” STP and MSTP protocols have also turned out: all the access nodes in an access part are connected to only one internal bridge that has the smallest bridge ID. By having choice between two or more VLANs (MSTP$_{opt2}$) further gain can be achieved. Nevertheless, the traffic-driven methods have about the same performance as the topology-driven methods when the access parts have tree-like topologies.

V. CONCLUSION

In this paper we have suggested a novel optimization framework for setting up trees in STP and MSTP based Ethernet metro access networks. We have shown that for typical networks of practical interest the throughput gain of optimization is surprisingly high.

A further significance of our optimization framework is that we handle and differentiate multiple QoS classes within a single network, and we perform traffic engineering not only by routing trees but also by allowing VLAN assignment to different trees. For denser topologies (not for trees) this proposed traffic driven approach leads to better resource usage, and therefore higher throughput than the known topology driven approaches.

REFERENCES


Fig. 2. Fair and greedy throughput (TP) on the three investigated topology.

TABLE II
CAPACITY PARAMETERS OF THE MEDIUM-SIZED TOPOLOGY

<table>
<thead>
<tr>
<th>Methods</th>
<th>Total used capacity [Mbps]</th>
<th>Link util. [%]</th>
<th>Ave. path length per QoS class</th>
<th>Platinum</th>
<th>Gold</th>
<th>Silver</th>
<th>BE</th>
</tr>
</thead>
<tbody>
<tr>
<td>STP</td>
<td>898.806</td>
<td>32 100</td>
<td></td>
<td>4.50</td>
<td>4.50</td>
<td>4.50</td>
<td>4.50</td>
</tr>
<tr>
<td>MSTP$_{opt}$</td>
<td>1997.572</td>
<td>54 100</td>
<td></td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
</tr>
<tr>
<td>MSTP</td>
<td>1198.455</td>
<td>44 100</td>
<td></td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
<tr>
<td>MSTP$_{opt2}$</td>
<td>2397.862</td>
<td>73 100</td>
<td></td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
<td>3.00</td>
</tr>
</tbody>
</table>

Table II shows that if more spanning trees are used, the path length decreases, since there is no need to detour the traffic through the root node of the only tree. Moreover, decrease of average path lengths explains why the small increase of allocated capacity results higher throughput gain. Although more traffic is transmitted, it is routed on shorter paths, thus higher link utilization is achieved (average link utilization). It is important to notice that, the maximal link utilization is 100% in all cases, i.e. the traffic is scaled until a bottleneck comes up.

c) Complexity and scalability: The trees we search in all cases are Steiner trees. Therefore, the considered problems are NP-complete. The simulations show that tree optimization needs about 20 minutes to give solution for the medium-sized topology on an AMD Athlon 2000+, but MSTP$_{opt2}$ increases the time consumption dramatically up to about 3 hours. These facts raise the need for heuristic methods.

D. The Lessons Learned

The simulations indicate that the throughput depends mostly on the topology and on the type of tree optimization method. The advantage of the MSTP over STP is obvious: the two