# Performance Analysis of Shortest Path Bridging Control Protocols

János Farkas and Zoltán Arató
Traffic Analysis and Network Performance Laboratory
Ericsson Research
Budapest, Hungary
{Janos.Farkas, Zoltan.Karoly.Arato}@ericsson.com

*Abstract* — **Ethernet networks are evolving from the enterprise to the carrier thus facing new demands and requirements. Therefore, a set of new control approaches are being defined for Ethernet. Amongst these, IEEE 802.1aq Shortest Path Bridging (SPB) aims to assure the transmission of unicast and multicast frames on the shortest path in a bridged network. Three protocols have been proposed for the control of SPB. This paper provides an extensive analysis of these proposals based on simulations focusing on protocol convergence time, which is a key attribute of carrier grade networks. The simulation model was created based on recent measurements. Furthermore, the analysis presented in the paper provides reference for further development of Ethernet control protocols.**

***Keywords - Carrier Ethernet; convergence time; protocol performance***

## I. Introduction

Simplicity and the high bandwidth available at low cost made Ethernet an attractive choice in various networking deployments. Ethernet was originally developed to provide connectivity in Local Area Networks (LANs), as such, it did not have the characteristics required in carrier grade networks to provide quality of service guarantees. Therefore, the past few years have seen significant innovations around Ethernet standards in order to meet recent requirements.

IEEE 802.1aq Shortest Path Bridging (SPB) [1] is one of the ongoing developments, which aims to improve network utilization and minimize latency by implementing shortest path forwarding in Ethernet. Furthermore, fast failover is also aimed to be provided by SPB. In order to increase network throughput, automatically controlled multiple Shortest Path Trees aimed to be maintained in support of a VLAN in an SPB network thus allowing both for unicast and multicast frames to reach their destination on the shortest path.

Multiple Spanning Tree Protocol (MSTP) [2] is the currently applied protocol for the control of forwarding in Ethernet networks, which typically implies roundabouts in frame transmission. Optimizations can be performed on the trees of MSTP, nonetheless, it requires manual configuration.

Three control protocols have been proposed for SPB networks. One of them is a distance-vector protocol, which extends MSTP, whereas, the two other proposals rely on link-state operation. Former comparisons of link-state and distance-vector routing protocols provided e.g. in [3], are not really relevant for SPB controlled Ethernet as the extensions needed for link-state protocols affect their performance significantly. In addition, MSTP implements such enhancements that were not part of the analyzed distance-vector routing protocols, e.g. a fast re-route kind of switching to a safe alternate path.

The goal of this paper is to provide a performance analysis of the control protocols proposed for SPB in order to make them comparable. Therefore, we implemented the protocols in the same simulator platform. We modeled network delays and set the corresponding simulator parameters carefully based on recent measurement results. We invoked extensive simulations for various parameter settings on different network topologies of diverse size in order to present a wide comparison. Protocol convergence time was in the focus of our investigation as it is a key reliability characteristic of carrier grade networks providing transport for traffic with QoS requirements.

The rest of the paper is structured as follows. SPB architecture is explained in the next section where the investigated control protocols are also described. The performance evaluation method and the results are presented in Section III. Finally, we conclude the paper in Section IV.

## II. Shortest Path Bridging

### A. Architecture Overview

IEEE 802.1aq Shortest Path Bridging (SPB) [1] implements frame forwarding on the shortest path between any two bridges of an Ethernet network. In order to achieve shortest path forwarding, each bridge maintains its own Shortest Path Tree (SPT) where the owner bridge is the Root. Shortest Path Bridges form an SPT Region and edge bridges of the Region forward the frames that are incoming to the Region on their own tree. As the Root Bridge is the only source in a tree, the SPTs applied in SPB are source rooted. Multicast trees are also source rooted, thus the SPTs support multicast frame forwarding, which is a key feature of Ethernet networks.

The control protocol of an SPB network has to be designed such that it complies with the requirements of an Ethernet network. First of all, it is essential to avoid loops as the multiplication of broadcast and multicast frames in a loop may cause a network melt-down. Furthermore, it may cause out of order delivery and multiple reception of the same frame at the receiver's side.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

The forwarding paths of an Ethernet network have to meet two congruency requirements in order to support MAC address learning and Ethernet OAM. That is, the very same path has to be used in the forward and backward directions between any bridge pairs; furthermore, unicast and multicast frames have to be forwarded on the same path. Therefore, the control protocol of an SPB network has to maintain SPTs that fulfill both congruency requirements.

The three different control protocol approaches proposed for SPB are briefly described in the following.

### *B. Distance-vector SPB*

Multiple Spanning Tree Protocol (MSTP) [2] could be applied for the control of SPB as it maintains multiple spanning trees, which are referred to as MSTP Instances (MSTI). MSTP is an extension to the Rapid Spanning Tree Protocol (RSTP) hence it operates along the same principles. The protocol elects a Root Bridge for a spanning tree and the distance from the Root Bridge is then propagated in Bridge Protocol Data Units (BPDU) in the network thus each bridge can determine the shortest path to the Root Bridge hence an SPT is formed in a distance-vector based manner.

As proposed in [4], MSTP can be applied for the control of SPB if an MSTI is assigned to each bridge on which the owner bridge is the Root Bridge, thus each bridge has its own SPT. In order to meet the congruency requirements, BPDUs have to be extended with a so called Path Vector [4], which reflects the list of bridges comprising a path. The Path Vector is then included into the end-to-end path cost, thus different bridges select the same path among equal cost multiple paths. Loop-free operation is assured in this approach as MSTP provides loop prevention.

MSTP is a distance-vector protocol thus the well known count-to-infinity problem [5] may increase its convergence time to the range of seconds. Count-to-infinity may appear on an MSTI in case of its Root Bridge failure, which is not a problem for SPB as the affected MSTI becomes unused after the break down of its solely source bridge.

### *C. Link-state SPB*

Link-state protocols, such as Open Shortest Path First (OSPF) or Intermediate System to Intermediate System (IS-IS) [6],[7] are designed for computing and maintaining the shortest path between node pairs. A link-state protocol is also suitable for the control of SPB as each bridge is able to compute the SPT of all bridges based on its link-state topology database. IS-IS is preferred for Ethernet networks as it supports the handling of MAC addresses and it is able to run directly over Ethernet as it is not tight to IP.

Application of IS-IS in SPB has further advantages besides eliminating the drawbacks of MSTP as the introduction of IS-IS provides new features for Ethernet networks. The topology discovery mechanism can be easily extended to also provide multicast service membership discovery hence no additional protocol is needed besides IS-IS to manage multicast trees. Furthermore, load sharing can be implemented in SPB as IS-IS is able to take advantage of equal cost multiple paths.

However, IS-IS has to be extended to meet the requirements of Ethernet networks. Transient loops may appear in an IS-IS controlled network due to inconsistent link-state topology databases in different nodes during a topology change. Therefore, a loop prevention mechanism has to be implemented in link-state SPB. In order to provide path congruency, the end-to-end path cost is extended with the ordered list of the IDs of bridges comprising the path [1]. Applying different ordering role, equal cost multiple paths can be incorporated into distinct SPTs, among which load sharing can be then implemented. A further challenge of link-state SPB is that each bridge has to maintain the SPTs of other bridges besides its own one, which increases the computational complexity compared to former applications of IS-IS and may also influence the convergence time after a topology change.

Two approaches have been proposed for the application of IS-IS in SPB, which apply different extensions to the standard IS-IS. A common extension is the tie-breaking for equal cost multiple paths. Transient loops are avoided in a different manner and the computation complexity is also addressed differently. The two approaches are presented in the following

#### *1) Basic IS-IS*

IS-IS is applied without any major change in its operation in this approach, which is referred to as Basic IS-IS in the following. The main extension is that Dijkstra algorithm with the tie-breaking described above is used for the computation of SPTs and each bridge computes the SPTs of every other bridge besides its own SPT in order to set frame forwarding properly. Thus, maintenance of SPTs is computationally intensive in this approach, which was proposed in [8] and [1] uses the same operational principles.

A further extension is that Basic IS-IS uses the neighbor handshake mechanism specified in [1] for loop prevention, which does not depend on the SPT computation.

#### *2) Root Controlled Bridging*

Root Controlled Bridging (RCB) has been proposed as an extension to IS-IS for the control of SPB networks in [9]. In RCB, bridges maintain the link-state database as specified in IS-IS, nonetheless, each bridge only computes its own tree and controls the set-up or update of its own tree, i.e. each tree is controlled by its Root Bridge. Thus, the architecture is distributed and robust as each bridge controls a different tree independently of each other. Nonetheless, the control of each tree is centralized, which has significant advantages in reducing computation. Note that if a bridge goes down, then the tree where it was the Root Bridge becomes unnecessary.

After invoking SPT computation using the extended Dijkstra, the Root Bridge advertises its SPT to the rest of the bridges in Tree Advertisement (TA) messages. TA can be implemented in a new TLV (Type, Length, Value) defined according to [10] thus it is a standard compliant extension to IS-IS. A compressed description of SPTs is specified in [9] for TA messages in order to minimize the size of control messages. Furthermore, TA messages only contain the relevant subtree when they are forwarded thus the size of the TA messages decreases as they are sent towards the leaf bridges, which further decreases control traffic. An important feature of TA message propagation is that a TA message is only forwarded along the tree that it describes from the root towards the leaves, i.e. it is not flooded. Another key feature of the SPT update process is that link blocking is always set before link activation and TA forwarding. That is, the links not taking part in the new SPT are blocked after the reception of the TA message before any other action thus loop-free operation is ensured as shown in [9]. Therefore, no additional mechanism is needed for loop prevention in RCB as it is involved in the SPT update process.

RCB significantly decreases the computational complexity compared to Basic IS-IS as bridges do not need to compute the SPTs of other bridges but each bridge only computes its own tree. Nevertheless, RCB applies more control messages than Basic IS-IS.

## III. Performance Analysis

We provide a performance analysis of the three control protocol alternatives proposed for SPB in the following. The protocol convergence time is in the focus of our investigations as it is a key characteristic of packet networks carrying real-time traffic, furthermore, fast failover is also addressed by IEEE 802.1.

### A. Analysis Method

We evaluated protocol performance through simulation analysis. We implemented the bridge architecture as specified by [2] in our event driven packet simulator platform, which follows the ISO/OSI layers. We then implemented MSTP including its state machines as specified in [2] and the extensions needed for SPB. On top of the same bridge architecture we also implemented IS-IS according to [6] and [7]. In addition, the Basic IS-IS and the RCB extensions were also added to the IS-IS implementation. That is all three protocols are implemented on the same platform thus helping objective comparison by eliminating implementation differences as much as possible.

We investigated the convergence time, which is the time elapsed from a failure event realized by the control protocol until the forwarding has been recovered, i.e. all control protocol messages affecting the forwarding have been processed and the new forwarding topology has been set. That is, the end of processing of the last BPDU causing any change in an MSTI is considered as the end of the convergence time for MSTP based SPB (MSTP-SPB). The end of convergence is the end of the processing of the last Link State Protocol Data Unit (LSP) causing any change in an SPT for IS-IS based approaches. We are interested in the performance of the protocol, therefore, the time needed for failure detection is not taken into account. That is, we assume that physical layer upcall is implemented for the immediate notification of the MSTP or the IS-IS entity if a link goes down.

Three different topologies were used for the evaluation of protocol performance as the convergence time depends on network topology. The size of a network also affects the convergence time especially in larger networks; hence we investigated the different topologies for various network sizes form 50 to 280 bridges.

Ring topologies are widely deployed in real networks, e.g. in access networks. Therefore, one of the investigated topologies, which is referred to as Rings, consists of an inner and multiple outer rings as depicted in Fig. 1-a. More redundant topologies are used in core networks, therefore, we also investigated two types of mesh. The Heavy-mesh, which is illustrated in Fig. 1-b, comprises an almost full-mesh inner part and a bunch of outer bridges connected to the same inner bridge pair in a dual-homing manner. Fig. 2 shows the Light-mesh topology, which consists of inner bridges better connected than in the Rings topology but less connected than in the Heavy-mesh topology and single outer bridges dual-homing to the inner part.

Convergence time has been investigated for failure events occurring in the inner part of the network topologies. That is either a link between two inner bridges went down, e.g. between $I_1$ and $I_2$ shown in the figures, or an inner bridge broke down, e.g. $I_1$.

### B. Parameter Settings

We intended to compare the core of the protocols, therefore, we set the protocol parameters affecting convergence time such that they do not put any limit on it. For the same reason, we assumed immediate messaging in reaction to an event without any random delay before message transmission. The effects of IS-IS timers were eliminated as proposed in [11]. We used the standard default value of the protocol parameters that do not affect convergence. Thus, the simulation results only show computation and messaging times.

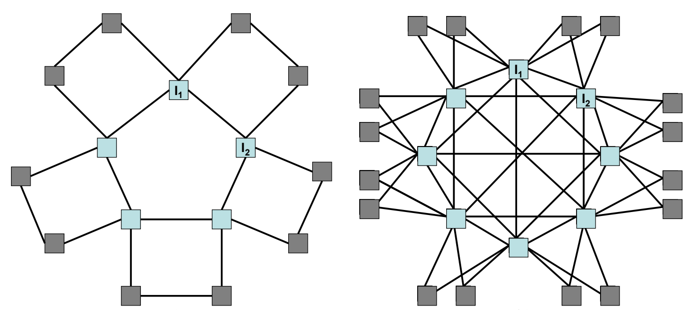


Figure 1. Rings and Heavy-mesh physical topologies

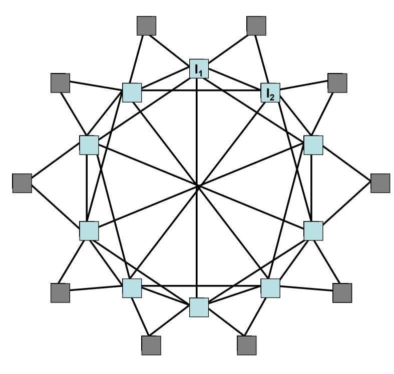


Figure 2. Light-mesh physical topology

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

The messaging and processing delays were modeled and implemented in our simulator based on recent measurement results published e.g. in [12], which we used for the validation of our simulation results. The transmission delay of a message on a link is proportional to the linkrate and the message size. We assumed Fast-Ethernet links during the simulations and the size of the control messages was determined based on the standards. Furthermore, we assumed that control traffic has high priority thus it does not suffer queuing delay due to background traffic. The message processing time is another delay factor appearing in all approaches, nonetheless, heavily implementation dependent. Therefore, we evaluated the convergence time for various control message processing time values from 0.1 ms to 10 ms. We applied deterministic values as probability distribution is not available for message processing time. The same message processing time was applied within a scenario for all the three protocols in order to provide comparable results.

As none of the parameters have probability distribution, the results presented in the paper are deterministic, which is useful for unbiased comparison of protocol performance in such a large parameter space.

### *C. Simulation Results*

The convergence time of the three control protocols have been analyzed with various parameter settings for the topologies described above. The effects of different processing parameter settings are evaluated first and the dependency on network topology and size is investigated afterwards.

#### *1) MSTP Parameter Settings*

The convergence time of MSTP maybe influenced by the upper limit set for the BPDU transmission rate on a port. The *Transmit Hold Count (TxHoldCount)* parameter limits the number of BPDUs that can be transmitted in a second, which can be set in the range of 1-10 and its default value is 6. We investigated the effect of *TxHoldCount*, and summarized the results in Table I. The convergence time of a single MSTI is shown for different topologies and failure events. Besides the maximum value allowed by the standard, *TxHoldCount* = 100 was also investigated, which ensures that it has no effect on convergence. The processing delay of BPDUs was set to 1 ms during the investigations described in this section. As the table shows, the standard *TxHoldCount* value influences the convergence time in sparsely connected and larger topologies. That is, it significantly slows the convergence in the Rings topology and the limitation effect also appears in larger Light-mesh topologies but it has no effect on the Heavy-mesh. The reason for this is that in a well connected network it is more likely that there are alternative paths to the Root Bridge and the failure is handled by switching to a formerly blocked safe alternate path. As opposed to this, a major reconfiguration of the tree is typically needed in a sparsely connected topology, which may be affected by a limited BPDU transmission rate.

TABLE I. THE EFFECT OF TXHOLDCOUNT PARAMETER ON THE CONVERGENCE TIME (S) OF MSTP

| | | Heavy-mesh | | Light-mesh | | Rings | |
|---|---|---|---|---|---|---|---|
| | N | *TxHC=10* | *TxHC=100* | *TxHC=10* | *TxHC=100* | *TxHC=10* | *TxHC=100* |
| Link failure | *50* | 0.0043 | 0.0043 | 0.00431 | 0.00431 | 2.00107 | 0.01074 |
| | *100* | 0.00456 | 0.00456 | 0.00456 | 0.00456 | 12.0023 | 0.01593 |
| | *150* | 0.00483 | 0.00483 | 2.00121 | 0.00482 | 14.0012 | 0.02055 |
| | *200* | 0.00507 | 0.00507 | 3.00127 | 0.00507 | 18.0013 | 0.02535 |
| | *280* | 0.00548 | 0.00548 | 6.0014 | 0.00548 | 28.0014 | 0.03562 |
| Node failure | *50* | 0.00538 | 0.00538 | 0.00755 | 0.00755 | 5.0011 | 0.01504 |
| | *100* | 0.00684 | 0.00684 | 2.00114 | 0.01026 | 25.0023 | 0.02504 |
| | *150* | 0.00846 | 0.00846 | 10.0012 | 0.01568 | 24 | 0.03384 |
| | *200* | 0.00887 | 0.00887 | 13.0013 | 0.01901 | 26.0013 | 0.04943 |

TABLE II. COMPARISON OF MEASURED AND SIMULATED CONVERGENCE TIMES (S) FOR BASIC IS-IS

| | Heavy mesh | | Light mesh | | Rings | |
|---|---|---|---|---|---|---|
| N | *Measured* | *Simulated* | *Measured* | *Simulated* | *Measured* | *Simulated* |
| 50 | 0.0007 | 0.000963 | 0.0008 | 0.001285 | 0.0004 | 0.001284 |
| 100 | 0.003 | 0.003478 | 0.0041 | 0.00412 | 0.0016 | 0.00292 |
| 200 | 0.0136 | 0.014727 | 0.0245 | 0.016581 | 0.0068 | 0.009375 |

We set *TxHoldCount* to 100 in the rest of the simulations in order to show the performance of the protocol itself without this limitation.

#### *2) Dijkstra Computation Time*

The computation of the SPTs plays a key role in the convergence time of IS-IS based SPB protocols hence we analyzed the corresponding parameter settings.

The computational time of a binary heap implementation of the Dijkstra algorithm, which is common in commercial routers [11], is

$$DijkstraDelay = C \cdot L \cdot \log N \qquad (1)$$

where $N$ and $L$ denote the number of nodes and links in the network, respectively. Note that the computational complexity of the RCB approach is the same as that of the Dijkstra algorithm. As opposed to this a multiplier of $N$ has to be applied besides the Dijkstra in case of Basic IS-IS because each bridge computes the trees of all bridges.

The parameter $C$ of (1) has to be carefully chosen in order to scale the Dijkstra computation realistically in the simulator tool. $C = 1.2 \cdot 10^{-5}$s according to the measurement results published in [11]. However, we measured $C = 2.2 \cdot 10^{-8}$s on a 2.2 GHz Intel Core2 Duo processor. Therefore, we compared our simulation results to recently published measurement results for Basic IS-IS in [13], where the processing delay for LSPs was set to 0.1 ms in the simulator and a link broke down. As the results summarized in Table II show, a value in the order of $10^{-8}$ seconds is more appropriate today. Therefore, $C = 2.2 \cdot 10^{-8}$s is used in the rest of the simulations.

#### *3) Message Processing Delay*

The processing time of control messages is a key component in protocol convergence time. We therefore investigated its effect on the performance of the three protocols.

The processing of Tree Advertisement messages introduces an extra message processing delay in the RCB approach. The TA messages suffer the same delays as any other LSP, but the parsing of the tree description that they carry takes additional time, which is implemented in our simulator as

$$ParseDelay = P \cdot n \qquad (2)$$

where $n$ nodes comprise the subtree carried in the TA. We measured 0.001 ms for $P$, nonetheless, we investigated various


This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

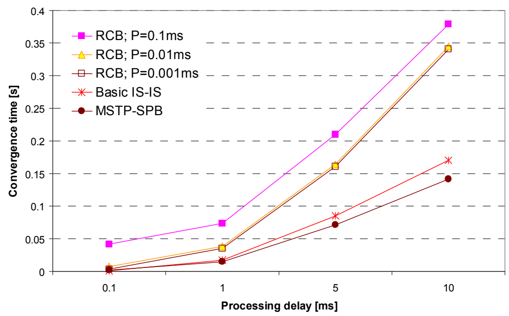


Figure 3. Effect of message processing delay on the convergence time after a bridge failure in a 50-node Rings topology

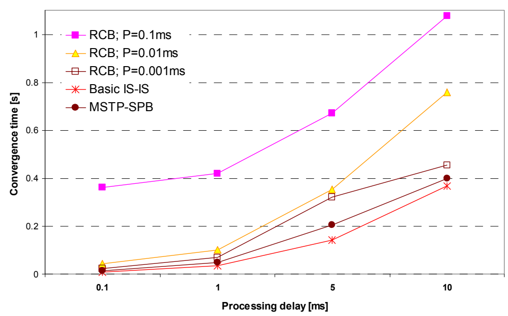


Figure 4. Effect of message processing delay on the convergence time after a bridge failure in a 200-node Rings topology

values for $P$ from 0.001 ms to 0.1 ms in order to provide a wider picture on the performance of RCB.

We analyzed the effect of message processing delay for a bridge failure in 50-node and 200-node Rings topologies; Fig. 3 and Fig. 4 show the results, respectively. All approaches are very sensitive to control message processing time, especially RCB, which is the most intense in messaging. Nonetheless, the computational intensity of Basic IS-IS increases with the size of the network, therefore, the difference between Basic IS-IS and RCB with the most realistic parsing delay ($P$ = 0.001 ms) decreases.

BPDU processing time is in the order of milliseconds according to [12], nevertheless, the measurement results published in [13] suggest LSP processing time in the order of 0.1 ms. We apply 1 ms processing time both for BPDUs and LSPs in the following simulations in order to provide comparable results.

### 4) *Network Topology and Size*

The convergence time of every control protocol largely depends on the network topology and on the size of the network. Therefore, we evaluated the convergence time of the SPB protocols for various network sizes of the topology types described in Section III.A. Fig. 5, 6 and 7 show the convergence time as a function of the network size in case of a single link failure.

As the figures show the size of the network has different impact on the protocols for different network types. In the sparse Rings topology, the convergence time of the different protocols changes similar to each other if we take into account the realistic $P$ = 0.001 ms for RCB. This topology is not favorable for control messaging as the control information has to travel long paths, thus distance-vectors of MSTP-SPB are also propagated on long paths. Furthermore, it is not likely in a sparse topology that MSTP-SPB is able to switch to a safe alternate path after a failure event.

However, MSTP-SPB is the less affected by the increase in network size in better connected topologies as the results for the mesh topologies show, because it is more likely that a safe alternate path exists. Switching to a safe alternate path can be invoked very fast, which explains the slight increase in the convergence time of MSTP-SPB despite of the increasing network size. As opposed to this, Basic IS-IS is considerably affected by the network size as both the number of nodes and links influences its computational complexity. Therefore, RCB with realistic parsing speed over-performs Basic IS-IS after a certain network size.

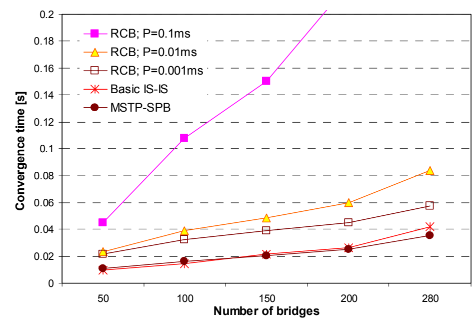


Figure 5. Link failure in the Rings topology

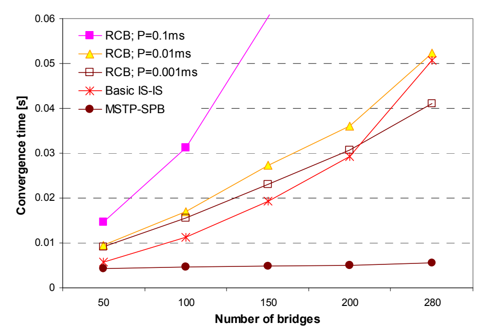


Figure 6. Link failure in the Light-mesh topology

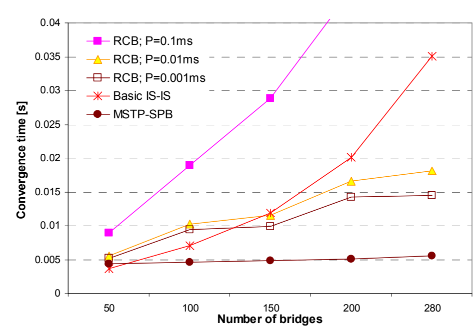


Figure 7. Link failure in the Heavy-mesh topology

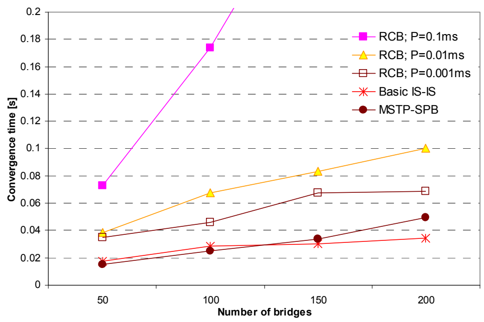


Figure 8. Bridge failure in the Rings topology

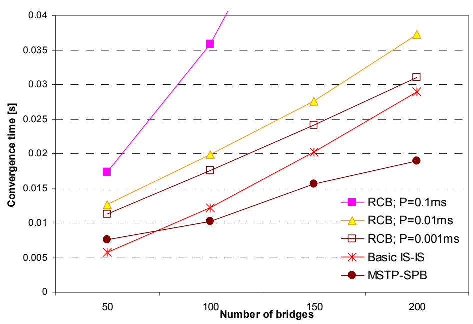


Figure 9. Bridge failure in the Light-mesh topology

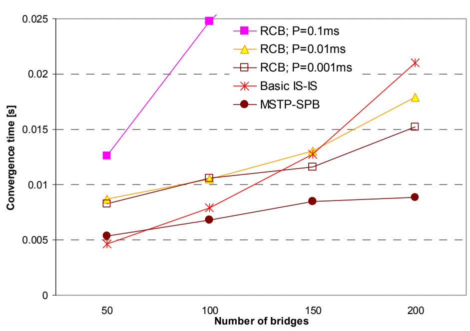


Figure 10. Bridge failure in the Heavy-mesh topology

We also investigated the performance of the protocols for a bridge break down event and present the results in Fig. 8, 9 and 10. A bridge failure initiates a major reconfiguration of SPTs. The sparser the topology, the more SPTs have to be reconfigured, which mostly affects MSTP-SPB and RCB as they involve messaging for SPT reconfiguration. A bridge failure decreases the chance for having a safe alternate path, therefore, the convergence time of MSTP-SPB increases compared to the link failure case. Furthermore, MSTP-SPB becomes more sensitive to network size even in mesh topologies for the same reasons. Basic IS-IS is the less affected by the different type of failure as its computational complexity remains the same and the processing time of additional LSPs is not significant. Nonetheless, RCB uses TA messages for the reconfiguration of SPTs hence its convergence time increases compared to the link failure scenario as a bridge failure affects more SPTs. Therefore, the intersection point of RCB and Basic IS-IS curves moves towards larger network sizes in case of mesh topologies compared to that of the link failure scenarios.

## IV. Conclusions

We extensively analyzed the performance of the control protocols proposed for IEEE 802.1aq SPB. We implemented these protocols over the same bridge architecture in our packet simulator in order to provide a just comparison. The protocol convergence time was investigated thoroughly because it is a key performance attribute of carrier grade networks.

The performance analysis showed that the convergence time of the Basic IS-IS approach is sensitive to the size of the network in case of mesh topologies due to its computational complexity. Root Controlled Bridging (RCB), which reduces the computational complexity but applies more control messages instead, converges faster than Basic IS-IS as the size of mesh topologies increases. However, RCB performs slightly worse than Basic IS-IS in sparse topologies as messages are transmitted on longer paths in these topologies. The MSTP based SPB approach performs well in mesh topologies because MSTP is able to quickly switch to a safe alternate path, which likely exists in a well-connected topology. The performance of MSTP based SPB is worse in sparse topologies, e.g. in rings, where alternate paths are typically not available.

Our future work includes the improvement of link-state approaches with respect to convergence time, e.g. by means of the adaptation of an incremental Dijkstra algorithm for the computation of Shortest Path Trees.

## References

[1] IEEE Std. 802.1aq D2.0, “IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks – Amendment 8: Shortest Path Bridging,” July 2009.
[2] IEEE Std. 802.1Q, “IEEE Standard for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks,” 2005.
[3] A. U. Shankar, C Alaettinoglu, I. Matta, and K. Dussa-Zieger, “Performance comparison of routing protocols using MaRS: Distance-vector versus linkstate” ACM SIGMETRICS Conference on Performance, June 1992.
[4] N. Finn, “Shortest Path Bridging,” September 2005, http://www.ieee802.org/1/files/public/docs2005/aq-nfinn-shortest-path-0905.pdf
[5] A. Meyers, E. Ng and H. Zhang, “Rethinking the service model: Scaling Ethernet to a million nodes,” HotNets'04, November 2004.
[6] ISO/IEC 10589:2002 Second Edition
[7] D. Oran, “IS-IS protocol specification,” IETF RFC 1142, February 1990.
[8] D. Allan, P. Ashwood-Smith, N. Bragg, and D. Fedyk, “Provider link state bridging,” IEEE Communications Magazine, September 2008.
[9] J. Farkas and R. Pallos, “Root Controlled Bridging: A Scalable Control Protocol for Shortest Path Bridging,” Networks 2008, Budapest, Hungary, September 2008.
[10] IETF RFC 3784, “Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE),” June 2004.
[11] P. Francois, C. Filsfils, J. Evans and O. Bonaventure, “Achieving sub-second IGP convergence in large IP networks,” ACM SIGCOMM Computer Communication Review, July 2005.
[12] R. Pallos, J. Farkas, I. Moldován and C. Lukovszki, “Performance of Rapid Spanning Tree Protocol in Access and Metro Networks,” AccessNets 2007, Ottawa, Canada, August 22-24, 2007.
[13] J. Chiabaut, “All pairs shortest path performance measurements,“ March 2008, http://www.ieee802.org/1/files/public/docs2008/aq-chiabaut-all_pairs_shortest_path-0308-v01.pdf

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.