Simulation-assisted Analysis and Design of STP-based Networks

Paolo Medagliani, Gianluigi Ferrari
University of Parma
viale G.P. Usberti 181/A
Parma, Italy
paolo.medagliani@unipr.it,
gianluigi.ferrari@unipr.it

Gianpietro Germi, Fabio Cappelletti
Selta spa
Via Emilia 231
Cadeo, Piacenza, Italy
g.germi@selta.it, f.cappelletti@selta.it

ABSTRACT

In this paper we analyze, through simulations, the performance of *Spanning Tree Protocol* (STP)-based Ethernet networks with ring and double ring topologies. In particular, we consider both the presence and the absence of *Virtual Local Area Networks* (VLANs), and we derive the optimized STP parameters which minimize the STP convergence time and maximize the network stability. Two possible techniques for STP internal timers management are evaluated. The presence of failures (either broken links or nodes) is also taken into account, in order to determine the proper STP parameters which guarantee connectivity recovery and convergence in all possible network scenarios. Some of the simulation results are also verified through an experimental testbed. Finally, the use of "transparent" switches is proposed as a solution to (i) speed convergence, (ii) increase the reaction capability to failures, and (iii) overcome the limitations, on the maximum number of sustainable nodes, imposed by the STP. In particular, this approach allows to extend the number of nodes in the network, still guaranteeing the possibility of incorporating VLANs.

Categories and Subject Descriptors

C.4 [**PERFORMANCE OF SYSTEMS**]: Modeling techniques; C.2.2 [**COMPUTER-COMMUNICATION NETWORKS**]: Network Protocols—*protocol verification*; I.6.4 [**COMPUTING METHODOLOGIES**]: Simulation and Modeling—*Model Validation and Analysis*

1. INTRODUCTION

Ethernet is a widely used connection technology for Local Area Networks (LANs) [1]. Its simplicity, low cost, and high data transfer capacity have pushed its use in several application scenarios. For example, Ethernet is the technology of choice to interconnect racks of servers with low latency and high reliability, or to store data on remote hard disks (i.e., Storage Area Networks, SANs). In all cases, the ability of the network to react against failures is of paramount importance.

In order to improve the network robustness against failures, a first solution consists in guaranteeing redundancy of paths between a source and a destination. For example, in most process automation plants a long connectivity loss cannot be tolerated. Therefore, exploiting path redundancy to prevent from data loss and react against possible failures is highly desirable. However, the use of redundant paths is not allowed because it leads to the creation of loops in the network, which may quickly saturate its transport capacity. A solution to this problem is given by the adoption of the *Spanning Tree Protocol* (STP) [7], which eliminates the presence of loops in the network and provides alternative paths when the active one fails.

In [12], the author presents a framework for the enhancement of Ethernet networks, considering the bandwidth limitations introduced by the STP. Due to the wide diffusion of Ethernet networks in many application fields, the security aspects of this technology must be also taken into account. In [8], the authors analyze the stability of STP and develop a spanning tree port cost-based approach to resist to possible external attacks. In [13], the authors propose the division of an STP-based network into two tiers in order to increase security and hide network infrastructure operations.

Even if STP-based Ethernet networks have a capillary diffusion, in the literature there are a few papers analyzing the performance of these networks. Moreover, the choice of the optimized STP parameters, which guarantee network convergence, i.e., absence of loops, is typically left to heuristic trials. In this paper, we present a simulation-based performance analysis of STP-based Ethernet networks and, on the basis of the obtained results, we derive some guidelines for optimized configuration of STP parameters. In particular, network behavior is analyzed through the Opnet simulator [6]. Since no suitable models are provided by Opnet, a custom model, through which the convergence of the network is evaluated and the optimized parameters that allow fastest network convergence are derived, is implemented. The optimized values of the STP parameters are obtained both in the presence and in the absence of VLANs connected to the switches. In addition, the robustness of the STP-based networks against node failures is evaluated and a set of optimized configuration rules for the STP parameters is derived. Finally, the use of "transparent" switches is proposed as a possible approach to overcome the limitation,

SIMUTools '09, Rome, Italy

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5656
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5656

on the maximum sustainable number of nodes, imposed by the STP.

The structure of this paper is the following. In Section 2, an overview of the STP is provided and the logical subdivision of a network in VLANs is introduced. In Section 3, the performance of STP-based networks is first evaluated in the absence of failures, considering two different internal timer management strategies. In particular, the maximum network dimension (in terms of nodes) is derived as a function of the main STP parameters. Then, this analysis is extended to account for possible failures. In order to overcome the limitations (in terms of network dimension and convergence speed) imposed by the STP, in Section 4 we propose the use of "transparent" switches, which are properly characterized. In Section 5, on the basis of the previous results we summarize simple design guidelines for configuring the STP parameters. Finally, Section 6 concludes the paper.

2. SPANNING TREE PROTOCOL

2.1 Basics

The segments of a Local Area Network (LAN) are connected through switches, which operate at the Layer 2 of the ISO/OSI stack [9]. These devices forward the packets received from an input port towards one or more output ports. In order to have correct network operation, logical (or layer 2) path loops between the nodes must be avoided, i.e., there must be a unique active path between any pair of switches. In the opposite case, packets would be endlessly forwarded by the nodes, with catastrophic effects on the network performance. These problems are not relevant when transmitted packets have a Medium Access Control (MAC) address field with information known by the switches. In this case, each switch is aware of the devices connected to its ports. On the other hand, a broadcast message or a message directed to a node with an unknown MAC address is forwarded by a switch to all the active ports, except for the input one. In this case, in the presence of a loop, the packets will be replicated by all the switches in the network, thus quickly saturating the network.

A possible approach would be physically avoiding loops during the network creation phase. However, this choice can be unreliable in the case of a link failure, after which some areas of the network could become unreachable and isolated. A better approach would exploit the redundancy of paths from a source to a destination. After network start-up, only one of the redundant paths becomes active, whereas the remaining paths are left inactive. In the presence of a failure, the original path will be replaced with one of the inactive paths, guaranteeing correct network operations.

The algorithm which manages the activation of the links is known as *Spanning Tree Algorithm* (STA) and is used into the STP, which is a part of the IEEE 802.1D standard [2]. Since the operations, needed to manage the STP, are performed by all the switches in the network, the STA is totally distributed. The goal of the STP is the creation of a tree which allows to route data packets to any segment of the network, avoiding loops and leaving only one active path between any pair of source-destination nodes.

The limitations of this protocol can be summarized as follows: (i) the convergence time increases when the number N of switches in the network increases; (ii) the control traffic introduced by the STP degrades the network performance; (iii) the inactive paths do not increase the overall capacity of the network; and (iv) the maximum number of switches in the network is limited. Solutions to these problems are provided through possible enhancements of the basic STP, such as the Rapid STP [3] and the Multiple STP [4].

The main phases of the convergence process of the STP are: (i) election of a root node (i.e., a *root bridge*, RB, according to the STP reference names), (ii) determination of the least cost paths, (iii) disactivation of the remaining paths, and (iv) resolution of the paths with equivalent costs. The last phase occurs only when there is more than one path with the same characteristics, whereas the other three steps occur at the network start-up.

Every switch has a unique identifier and an associated priority. In the case of different priorities, the switch with the lowest priority becomes the RB. On the other hand, if all the priorities are equal, the node with the lowest identifier will become the RB [11]. Once the tree is created, every node has a minimum-cost path towards the RB. Note that the STP does not guarantee that the path between any pair of source-destination nodes is the one with minimum cost, because the minimization of the path cost is not the goal of the STP. The use of minimum-cost paths to the RB is guaranteed by the following rules:

- after the election of the RB, every switch computes the cost of every path from itself to the RB and chooses the one with least cost (the associated port is referred to as root port, RP);
- the switches in the same network segment cooperatively select the switch with least cost (the port which connects a switch to the network segment is referred to as designated port, DP).

Every switch needs to have a complete knowledge of the network (i.e., of the priorities and the identifiers of the other nodes). To this end, a periodical "special" packet, referred to as *Bridge Protocol Data Unit* (BPDU), is transmitted. A BPDU contains information about the transmitting node, i.e., the states of its ports, its priority, the cost of the path from the switch which originates the BPDU to the RB, and the identifier of the RB. The BPDUs are not forwarded by the receiving switch. According to the STP, there are three kinds of BPDU: (i) Configuration BPDU (CBPDU or simply BPDU), used by the switches to create and maintain the tree in the network; (ii) Topology Change Notification (TCN), used to notify topology changes in the network or the addition of a new switch; and (iii) Topology Change Acknowledgment (TCA), used to confirm a topology change in the network.

In order to prevent the formation of loops in the network, the STP defines five possible states for the ports of a switch: (i) *disabled*, (ii) *listening*, (iii) *learning*, (iv) *forwarding*, and (v) *blocking*. When the network is created, all ports connected to valid links are in listening state, while the others are in disabled state. In the former state, the switches receives the BPDUs from the other nodes. In the presence of a loop, a port of a switch, which becomes aware of the presence of the loop, is turned into the blocking state, in order to prevent the transit of BPDUs. After a time interval, referred to as forward delay (FD), equal to 15 s by default, the ports in listening state switch to learning state [10]. In this

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5656
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5656

phase, the nodes become aware of the surrounding switches. After another FD interval, the ports in learning state are switched into forwarding state and data packets can then circulate throughout the network.

According to the STP, the RB transmits a BPDU with a period denoted as "hello time" (equal to 2 s by default). At the network start-up, each switch elects itself as the RB and transmits a BPDU. If the priority information conveyed by a BPDU from a given switch is higher than the one stored in the receiving switch, the latter updates its status, electing as RB the transmitting switch, and stops transmitting its own BPDUs. In fact, from this moment on it will retransmit only the BPDUs received from the elected RB.

The conditions to be satisfied to guarantee convergence of the STP are the following: (i) all the switches elect the same RB; (ii) one of the switches which is part of the loop has a port in blocking state; (iii) the remaining ports of that switch and of all the other switches are in forwarding state; and (iv) the previous conditions are stable during the time. Once all previous conditions are met, only the RB will broadcast the BPDUs every 2 s and the other switches, upon the reception of a BPDU, will retransmit it and refresh their internal information.

Another important timer of the STP is the max age (MA) timer, which defines the time interval after which a reset of the switch is required if no refreshing BPDU is received. When a BPDU is retransmitted by a switch, the latter modifies only the cost of the path to the RB and a timer, referred to as message age (m_{age}), used to measure the "distance" of a node from the RB. In particular, this value is generally increased by 1 s or 2 s, depending on the state of internal timers.[1] This value is used in combination with the MA in order to guarantee the reliability of the information conveyed in a BPDU. More precisely, the RB generates a BPDU with $m_{\text{age}} = 0$. Then, the switches which receive this BPDU assume that the information transported by the BPDU is valid for a time interval equal to MA. When the BPDU is relayed, m_{age} is incremented and the receiving switch assumes that the information conveyed in the BPDU is valid for $MA - m_{\text{age}}$ s. When the message age of a switch becomes larger than the max age, the switch sends a TCN BPDU to the other switches in order to notify them of the occurred problem. In this case, the STP does not converge.

2.2 Incorporation of VLANs

VLANs are instrumental to logically segment the network into areas. This solution is less expensive and more flexible than the traditional approach based on the use of dedicated switches. This technology, included into the IEEE 802.1Q standard [5], allows to interconnect the switches which share the same VLAN, even if they belong to geographically separated networks. This operation, referred to as trunking, is based on the use of tags which identify which packets belong to which VLAN. A port which conveys tagged traffic is named trunk port, whereas a port, through which untagged packets enter inside the VLAN, is referred to as access port. In the case of mixed traffic, instead, the port is referred to as hybrid.

There are two possible ways of assigning a device to one or more VLANs: (i) dynamic and (ii) static. In the former case, database-based software packages are used to associate a switch with a specific VLAN. In the latter case, instead, assignments are based on a strict association between a port and the corresponding VLANs. This approach corresponds to the use of *port-based* VLANs. With this mechanism, all users connected to a port are automatically associated to the assigned VLAN. The STP can be applied to every created VLAN. In this case, the mechanism of the extension of the STP, referred to as *per-VLAN Spanning Tree* (PVST), remains the same, except for the fact that the tree is computed also for each VLAN (i.e., for the switches belonging to the same VLAN).

3. RING NETWORKS

3.1 Scenarios without failures

In order to analyze the STP and characterize its performance, a proper Opnet simulator model has been developed. This model, which allows to periodically extract (or log) the states of the ports of each switch, has been derived from the model of a Cisco CS2948 switch and presents 4 layer-2 ports. The links which connect any pair of switches are Ethernet 100 Mbps connections. The configurations considered in this paper refer to (a) ring and (b) double ring topologies, as shown in Figure 1. In particular, in the double ring topology, the switches are configured so that the RB is the node in the center of the double ring. Since the goal of this paper is to provide some guidelines for the configuration of the parameters of the switches in STP-based networks, we derive, for given values of the max age and forward delay, the network convergence time.

We also validate some of the simulation results through an experimental testbed formed by 9 Cisco 2811 switches[2] equipped with a 4-port HWIC-4ESW interface which operates at 100 Mbps, configured with a VLAN, referred to as VLAN 1, on all the ports.

The first indicator considered for STP convergence analysis, according to the definition provided in Subsection 2.1, is the number of exchanged packets in the network. This number, as a function of time, is shown in Figure 2. The network parameters are $N = 30$ switches, $MA = 20$ s and $FD = 15$ s. A ring network topology is considered. From the results in Figure 2, one might be tempted to conclude that the network has converged after approximately 50 s. However, observing the simulator log files and the states of the ports of each switch—not shown here for lack of space—it can be concluded that the network does not meet the convergence conditions described in Subsection 2.1, even if BPDUs are regularly transmitted. In fact, this analysis technique does not take into account the different types of BPDUs transmitted. In the results presented in Figure 2, the traffic is due to the presence of TCN BPDUs and not to CBPDUs transmitted after network convergence. Since the value of MA is not correctly selected for the considered network, the most distant nodes receive BPDUs with m_{age} equal to MA. According to the STP, as soon as a switch experiences this situation, it starts sending a TCN message to the other switches, which will eventually reply with TCA messages, upon acknowledgement of the topology change. This means

[1] Generally, a 1 s increment is associated with STP convergence, whereas a 2 s increment denotes no STP convergence in the network.

[2] The Cisco switches, i.e., CS2948, used in the simulator have the same functionalities of the switches in the experimental testbed.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5656
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5656

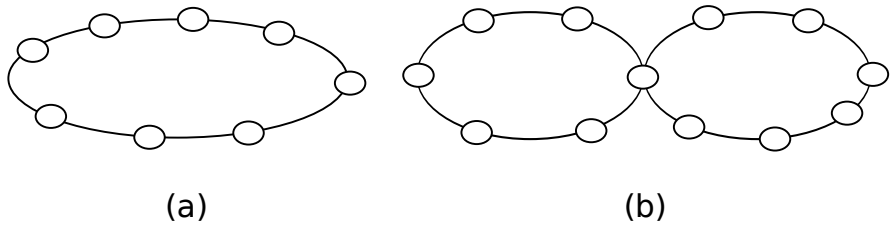

Figure 1: Topologies considered for simulation-based analysis: (a) ring topology and (b) double ring topology.

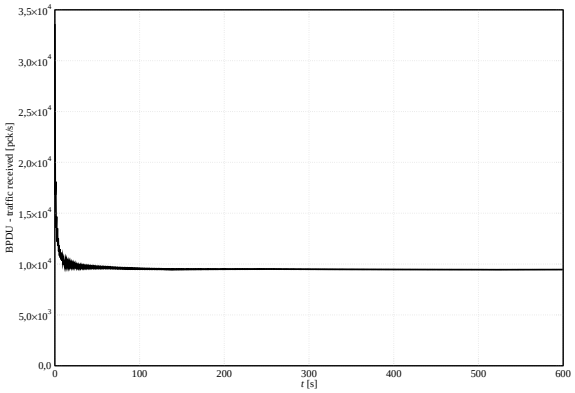

Figure 2: Number of exchanged packets in a scenario with $N = 30$ switches, $MA = 20$ s and $FD = 15$ s.

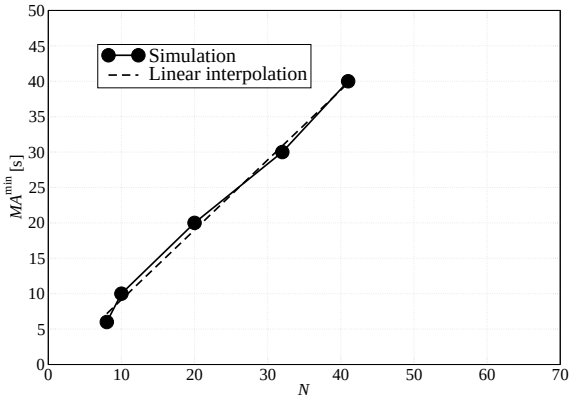

Figure 3: Minimum MA considering ring topology with Cisco switches.

that the chosen values of MA and FD are too small for the considered scenario and the STP does not converge. In this case, the network will be divided into two areas and the RB in the network will not be unique.

The convergence of the network can be verified by careful examination of a properly generated (through our simulator) log file at each node. In particular, we have considered scenarios with different numbers of switches, varying the max age and forward delay, in order to obtain the minimum values of MA and FD which guarantee network convergence. In Figure 3, the minimum values of MA required for convergence is shown as a function of the number of switches in the network. The FD is not shown, since it can be extracted according to the relation

$$2(FD - 1) \geq MA$$

required by the IEEE 802. 1D standard. Therefore,

$$FD^{\min} = \frac{MA^{\min}}{2} + 1.$$

By linearly interpolating (as a function of N) the MA values in Figure 3, one finds that

$$MA^{\min} = 0.9876 \cdot N - 0.7242 \simeq N - 0.7. \qquad (1)$$

Selecting the max age to the value in equation (1) guarantees fastest convergence. However, equation (1) is no longer valid when there is a link or a node failure (this scenario will be analyzed in Subsection 3.2). The first point of the curve shown in Figure 3 has been also verified experimentally (without the VLAN 1). Experimental results show that there is convergence with $N = 8$, $MA = 6$ s, and $FD = 4$ s, whereas with $N = 9$ and the same STP parameters the network does not converge.

In Figure 4, the minimum MA required for convergence, in a scenario with the double ring topology in Figure 1 (b), is shown as a function of N. The considerations carried out for the scenario with ring topology still hold in this scenario. The minimum value of MA is given, as a function of N, by the following expression:

$$MA^{\min} = 0.5292 \cdot N - 0.9643 \simeq \frac{N}{2} - 1. \qquad (2)$$

Note that expression (2) holds in networks where the RB is the central node of the double ring topology. In other scenarios, this expression is not valid and the convergence is no longer guaranteed.

As we have seen, the main limitation to the formation of large STP-based networks is given by the absence of convergence, caused by the fact that the m_{age} becomes larger than the MA at the nodes which are distant from the RB. A possible solution would be the modification of the increment which the m_{age} undergoes when the BPDU crosses a node. In particular, this increment can be expressed as follows:

$$\begin{aligned} m_{\text{age}} &= \lfloor BPDU_{\text{cross}} + m_{\text{a-io}} + D_{\text{ma}} \rfloor \\ &= \lfloor BPDU_{\text{cross}} + 1 + 0.5 \rfloor \end{aligned} \qquad (3)$$

where $BPDU_{\text{cross}}$ is the BPDU crossing time at switch, $m_{\text{a-io}}$ is the message age increment overestimate, and D_{ma} is the medium access delay. The first term is given by the

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5656
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5656

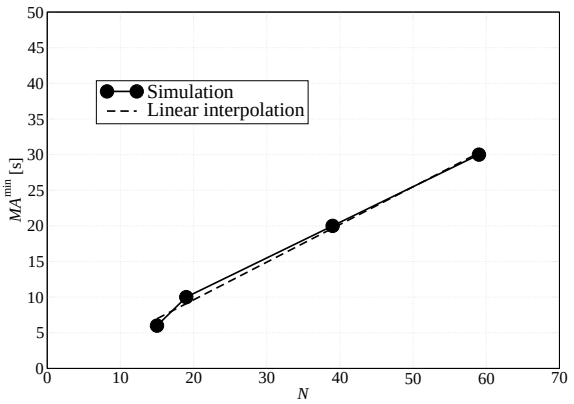

Figure 4: Minimum MA considering double ring topology with Cisco switches.

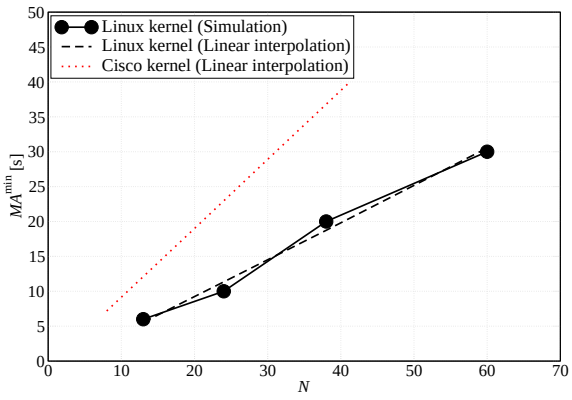

Figure 5: Minimum MA considering ring topology with switches with Linux kernel.

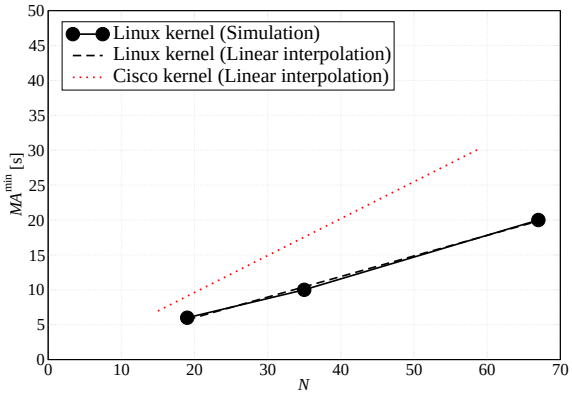

Figure 6: Minimum MA considering double ring topology with switches with Linux kernel.

difference between the transmission instant and the reception instant of the BPDU, i.e., the physical time interval required by the BPDU to cross a switch. The second and third terms, instead, are derived from statistical considerations and are equal to 1 s and 0.5 s, respectively, as indicated by Cisco [10]. In particular, $m_{\mathrm{a-io}}$ is the minimum increment necessary to avoid an underestimation of the BPDU age. D_{ma}, instead, is the time necessary for a device to gain the access to the medium for initial transmission. In other words, D_{ma} corresponds to the time between the instant at which the switch decides to retransmit the BPDU and the instant at which the BPDU effectively begins to leave the switch.

Expression (3) for the message age applies to Cisco switches (i.e., nodes with switch capabilities running the Cisco kernel). A Linux kernel is also publicly available. The Linux kernel neglects the contribution of the message age increment overestimate. Therefore, the message age can be given by the following expression:

$$m_{\mathrm{age}}^{\mathrm{Linux}} = \lfloor BPDU_{\mathrm{cross}} + D_{\mathrm{ma}} \rfloor = \lfloor BPDU_{\mathrm{cross}} + 0.5 \rfloor.$$

In Figure 5, the minimum MA of a network is shown, as a function of N, in scenarios with ring topology and switches running the Linux kernel. Since the crossing time is lower in nodes with the Linux kernel, the maximum number of switch, for which convergence is still guaranteed, is larger than in the case with Cisco kernel. In particular, given a value of MA, the number of switches which can be supported using the Linux kernel is twice that supported using the Cisco kernel. More precisely, the minimum message age depends on N as follows:

$$MA^{\mathrm{min}} = 0.5277 \cdot N - 1.309 \simeq \frac{N}{2} - 1.3. \qquad (4)$$

In Figure 5, for comparison purposes, the performance with the Cisco kernel (dotted line) is also shown—this curve is the linear interpolation curve in Figure 3.

The use of the Linux kernel allows to reduce the convergence time of the STP. In fact, since the convergence occurs after a time interval equal to $2FD$, the possibility of using a lower value of MA and, consequently, a lower value of FD reduces the convergence time. On the other hand, as already mentioned, a switch can detect a failure of a link or of another node if it does not receive any refreshing BPDU for a period of time longer than $MA - m_{\mathrm{age}}$. In scenarios with the Linux kernel, since the value of m_{age} increases more slowly (when crossing switches) than in scenarios with the Cisco kernel, the information stored in a node is valid for a longer time interval. Therefore, the failure reaction capability is reduced.

Similar considerations can be made in a scenario with double ring topology and Linux kernel. The minimum required value of message age is shown in Figure 6. As assumed in scenarios with Cisco kernel at the switches, the RB is forced to be the switch in the center of the double ring, and the obtained results are valid only for this case. The relation between the number of switches and the minimum max age can be approximated as follows:

$$MA^{\mathrm{min}} = 0.2948 \cdot N - 0.1161 \simeq 0.3N - 0.1. \qquad (5)$$

Similarly to the results presented in Figure 5, the minimum max age allowed with the Linux kernel is almost twice that allowed with the Cisco kernel.

In this Subsection, the convergence performance has been so far evaluated in the absence of VLANs connected to the switches. We have then extended our analysis to account

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5656
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5656

for the presence of VLANs, in the case of both single ring and double ring topologies. In particular, we have connected two VLANs (VLAN 1 and VLAN 2) to each node. From the analysis of the log files generated by the switches (not presented in this paper), it can be concluded that the convergence performance remains the same. In particular, the PVST allows to create both a common tree for all the switches and particular trees for each VLAN. Since in our simulations the VLANs have been connected to each switch using the previously described parameters, the common tree, the tree for VLAN 1, and the tree for VLAN 2 coincide. The convergence instant is still equal to $2FD$.

3.2 Scenarios with Failures

The results presented in the previous section have been obtained considering a network without failures. However, in order to test the validity of the STP, one must take into account also the reaction capability in the presence of a failure in the network. According to the STP, when a switch does not receive a refreshing BPDU for a period of time longer than $MA - m_{\text{age}}$, it sends a TCN BPDU in order to notify the neighbouring switches of the lack of convergence in the network. After the transmission of the TCN, the switch which has notified the change of topology, starts sending BPDUs assuming to be the RB of the network. If the neighbouring nodes have information about a better RB, they start replying with BPDUs in order to notify the node of their information. However, since the node which originates the TCN receives an information which is still "too old," it starts sending another TCN, thus originating the message exchange just described. This exchange of messages is a symptom of the fact that the STP parameters are too small for the considered network. On the other hand, when a neighbouring switch accepts the TCN, it sends back a TCA and stores information about the new RB. In this case, the network separates into two segments with different RBs. In particular, the switch, which originated the TCN, is characterized by an unstable state, since it oscillates between two possible values of the RB.

The latter case is exactly the scenario which occurs when a link or a node fails. More precisely, referring to the scenario with ring topology, a failure creates an open-chain network. If the STP parameters are too small for a network, the switch which verifies that the message age is larger than the max age, will start broadcasting a TCN. Since the nodes after that switch will not receive any other BPDU from the real RB, they will acknowledge the topology change and the new RB in the network. A solution to this problem is given by the use of more "relaxed" STP parameters, which let the BPDUs propagate into the open-chain to the most distant switch from the RB. The determination of the STP parameters through the analysis of an open-chain network with N switches has a peculiar importance in network design. In fact, the STP parameters optimized for this scenario guarantee that every network with loops, where the distance between any couple of (source-destination) nodes is smaller than N hops (i.e., $N-1$ switches must be crossed), converges.

According to the STP, a switch realizes that a reset is required when the value of m_{age} of a received BPDU is equal to the value of MA. Assume that there is a node or a link failure at a generic instant T^*. After this failure, a switch which does not receive any BPDU for a period of time equal to $MA - m_{\text{age}}$, starts sending a TCN message.[3] After the TCN message has propagated through the network and some switches have reset their information, their ports are first put into listening state. Then, after a time interval equal to FD, these ports are switched into learning state and, after another interval of length FD, into forwarding state. On the other hand, when the failure is recovered and the loop is restored, as soon as a BPDU propagates through the network and updates the information stored in each node, the switch which must put a port in blocking state, changes its port state and convergence is reached again.

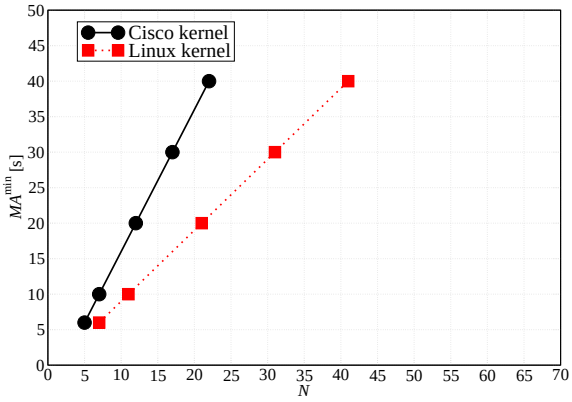

Figure 7: Minimum MA considering an open-chain scenario with a single failure.

In Figure 7, the minimum max age required to guarantee convergence in a open-chain network is shown as a function of the number N of switches in the network. Note that an open-chain network, obtained from a ring network with N nodes, upon a single node failure contains $N-1$ nodes. The performance in scenarios with Cisco and Linux kernels is evaluated. When the Cisco kernel is used, the nodes introduce $m_{\text{a-io}}$ equal to 1 s, and the MA rapidly reaches the value of m_{age}. On the other hand, when the Linux kernel is used, as mentioned above, given a value of MA, the number of admitted switches is larger. This consideration is confirmed by the equations which characterize the minimum values of MA as functions of the number of switches in the network. For the scenario with Cisco kernel one obtains

$$MA^{\min} = 2N - 4$$

whereas for the scenario with Linux kernel it holds that

$$MA^{\min} = N - 1.$$

This performance analysis can be extended to the case of a link failure, after which there are still N active nodes, unlike the previous scenario, where $N-1$ switches are active after a node failure. In the case of Cisco kernel, it holds that

$$MA^{\min} = 2N - 2$$

[3]Considering $T^* = 30$ s, $MA = 6$ s and a ring-topology with a failure of the switch neighbouring the RB, the network becomes aware of the failure at 33 s. In fact, the BPDU at $t = 30$ s is not received by the switch, therefore the last refreshing BPDU was received at $t = 28$ s and the information conveyed is valid for $6 - 1 = 5$ s upon the reception of the last BPDU.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5656
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5656

Table 1: Minimum value of MA predicted by the experimental testbed with switches with Cisco kernel.

$MA^{\min}$	N
6	4
8	5
14	8

and in the case of Linux kernel one has

$$MA^{\min} = N.$$

The results with Cisco kernel and a link failure have been confirmed through the experimental results presented in Table 1.

According to this configuration rule, the parameters are tighter than those presented in Section 3.1. However, even if this configuration leads to a longer convergence time, the stability of the network is guaranteed in all the scenarios where the paths between any couple of nodes are shorter than N hops. In addition, according to [2, 10], the recommended network diameter (i.e., the maximum number of switches that a packet crosses in order to link any two switches in the network) should be equal to 7. However, our results show that the maximum number of switches which guarantees convergence is 11 in the case of Cisco kernel, and 20 in the case of Linux kernel.

4. EXTENDED NETWORKS

In the previous section, we have presented the performance of the STP both in the absence and in the presence of failures in nodes or links. However, with large networks it is necessary to configure the STP with large values of MA and FD. This solution, which refers to the configurations presented in Subsection 3.2, is feasible and reliable, but leads to a longer convergence time and a slow reaction to a failure.

Generally, the network dimension can be extended through hubs, which relay received packets but do not participate to the operations of the STP. This solution is limited, since a hub cannot manage a VLAN. In this paper, we propose the use of "transparent" switches with disabled STP. Transparent switches, unlike hubs, can still manage VLANs. In real networks, the transparent switches are implemented by disabling the STP for both VLANs and the common tree. Since the models available in the Opnet simulator do not support this functionality, we have derived a transparent switch model from the model with enabled STP. A transparent switch receives BPDUs from a port and forwards them to all the other active ports. In addition, this switch can manage the VLANs, so that it can act as access port for the VLAN activated on each port and as a trunk for the links which connect the other switches. Since the STP increases the m_{age} as soon as a BPDU crosses a switch, the switches without STP must relay the received BPDU without increasing this value.

We have analyzed a network with double ring topology and three nodes with enabled STP: one is placed in the center of the double ring and the other two at the extremes of the double ring. In the middle of the STP-enabled nodes, a variable number of transparent switches can be placed. In particular, in our simulations a topology with 8 transparent switches, as shown in Figure 8, is considered. The

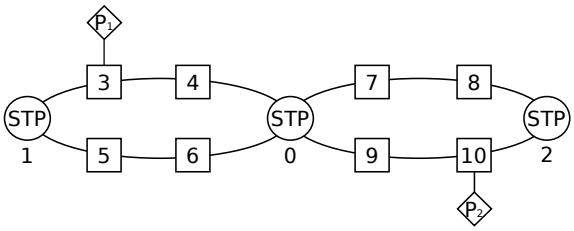

Figure 8: Extended network topology.

nodes with STP enabled (e.g., switch 0, 1, and 2) have been configured with $MA = 6$ s and $FD = 4$ s. Recalling the performance presented in Figure 3, in a scenario with 11 switches, the minimum value of MA should be equal to 10 s and, consequently, the minimum value of FD should be 6 s. In this scenario, we have also introduced two VLANs, named VLAN 1 and VLAN 2, on every node in the newtwork, both for those with STP enabled and for those with STP disabled. We have then used two nodes, referred to as P_1 and P_2, which periodically send a ping message in order to trace the active path between them. In addition, we have introduce a link failure at $T^* = 30$ s and a link recovery at $T^{**} = 90$ s in order to evaluate the reaction capability of this network. These instants have been chosen to let the network converge and, subsequently, after an alteration of the state of a node or a link, analyze its reaction speed.

The considered network converges at $t = 8$ s, as soon as the ports of the nodes running the STP switch into forwarding state. When the ping messages start to flow, the preferred route from P_1 to P_2 is 3 - 4 - 0 - 9 - 10. When a failure occurs at node 9 (we remark that the BPDU at $t = 30$ s is not delivered), the ping messages do not reach P_2 and BPDUs do not reach switch 2 on that side of the loop for 6 s. Then, at $t = 34$ s switch 2 sends a TCN message.[4] This message forces a change of state in the ports of the switch, so that for a period of time equal to $2FD$ switch 2 is no longer able to relay the received data. As soon as its ports are turned into forwarding state at $t = 42$ s, the new route for the ping message becomes 3 - 4 - 0 - 7 - 8 - 2 - 10. Once the failure is recovered, as soon as a BPDU propagates through the previously failed link, the information stored in the switch 2 is changed and the network converges again since switch 2 turns a port into blocking state. Referring to the failure recovery at $T^{**} = 90$ s (we remark that the BPDU at $t = 90$ s is delivered correctly), the ping message at $t = 91$ s is routed through the path 3 - 4 - 0 - 9 - 10.

5. DESIGN GUIDELINES

The performance analysis presented in the previous sections is useful to derive a set of considerations which help to design STP-based networks. Recalling Figures 3-6, one can note that the maximum number of switches that a network can tolerate, while implementing STP successfully, is limited. In particular, this number depends on the kernel used by the switches in the network. In fact, the use of the

[4]In this case, the failure affects a transparent switch, which does not take part to the STP. Therefore, the m_{age} of a BPDU received by the switch 2 is equal to 0 and its stored information are valid for MA seconds.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5656
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5656

Linux kernel, given a specific value of the MA allows to create a network with largest number of switches. For example, considering $MA = 20$ s, the maximum number of nodes using the Linux kernel is 40, whereas the maximum number of nodes using the Cisco kernel is 20. However, the use of the same MA has a different impact on the reaction capability. In fact, as explained in Section 3.1, the Linux kernel introduces a lower m_{age}, so that the information stored in the switches is valid for a longer period of time from the reception of the last BPDU.

This consideration can be reversed in order to determine the STP parameters that, given a fixed number of nodes N, guarantee fastest convergence and reaction against network failures. For example, given a network with $N = 20$, in the case of Cisco kernel, the minimum value of the max age is $MA = 20$ s, whereas in the case of Linux kernel the minimum value is $MA = 10$ s. In the latter case, the network capability reaction is still under analysis. In fact, with Linux kernel the MA is half of that with Cisco kernel. However, in the former case the m_{age} increases slowlier than in the latter case, and its impact on the recovery capability needs to be investigated.

In order to have correct network operations, the best solution is configuring the parameters considering an open-chain network with N nodes. In this way, the convergence is guaranteed for every network where the distance between any couple of nodes is lower than N hops. This solution is less efficient in terms of convergence speed, but it assures a high network reliability.

In order to overcome the intrinsic limitations on the number of nodes of the STP, speed the convergence, and increase the failure reaction capability of a network, an appealing solution is the use of "transparent" switches. This solution, in fact, allows to extend the network dimension, still guaranteeing the use of VLAN tagging.

6. CONCLUDING REMARKS

In this paper, we have first analyzed, through the Opnet simulator, the performance of an Ethernet network running STP, with switches equipped with either Cisco or Linux kernels. For each type of device, optimizing rules for the STP parameters has been derived, in order to speed network convergence. In addition, the presence of VLANs has been taken into account and the STP parameters have been optimized also in this scenario. Our simulation results have also been verified through an experimental testbed, which has confirmed the performance obtained through simulations. In order to provide a complete set of rules for network configuration, the open-chain network has been considered. From our analysis, it turns out that a proper configuration of STP parameters should guarantee that the STP works with every possible network configuration where the distance between any couple of nodes is smaller than N hops, even if this leads to a longer convergence time and slower reaction capability. In addition, the extension of the network through "transparent" switches has been considered as a mean to overcome intrinsic limitations of the STP. The use of this type of switches allows to significantly extend the the maximum acceptable dimension (in terms of number of nodes) of an STP-based network.

Acknowledgments

We acknowledge useful discussions with and continuous support from A. Cavagna and A. Pasino (Selta spa).

7. REFERENCES

[1] IEEE standards for local area networks: supplements to carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. *ANSI/IEEE Std 802.3a,b,c, and e-1988*, 1987.

[2] IEEE Standard for Information technology-Telecommunications and information exchange between systems- Local and metropolitan area networks- Common specifications Part 3: Media Access Control (MAC) Bridges. *ANSI/IEEE Std 802.1D, 1998 Edition*, pages i – 355, 1998.

[3] IEEE standard for local and metropolitan area networks - common specification. Part 3: media access control (MAC) bridges - amendment 2: rapid reconfiguration. *IEEE Std 802.1W-2001*, 2001.

[4] IEEE Standards for Local and metropolitan area networks - Virtual Bridged Local Area Networks - Amendment 3: Multiple Spanning Trees. *IEEE Std 802.1S-2002 (Amendment to IEEE Std 802.1Q, 1998 Edition)*, 2002.

[5] IEEE standards for local and metropolitan area networks. Virtual bridged local area networks. *IEEE Std 802.1Q, 2003 Edition (Incorporates IEEE Std 802.1Q-1998, IEEE Std 802.1u-2001, IEEE Std 802.1v-2001, and IEEE Std 802.1s-2002)*, 2003.

[6] Opnet website. `http://www.opnet.com`.

[7] R. Perlman. An algorithm for distributed computation of a spanning tree in an extended LAN. *SIGCOMM Comput. Commun. Rev.*, 15(4):44–53, 1985.

[8] K. Segaric, P. Knezevic, and B. Blaskovic. An approach to build stable spanning tree topology. *Int. Conf. on Trends in Communications (EUROCON'01)*, 2:400 – 403, July 2001.

[9] W. Stallings. IEEE 802.11: wireless LANs from a to n. *IT Professional*, 6(5):32–37, 2004.

[10] Understanding and Tuning Spanning tree protocol, Cisco website. `http://www.cisco.com/en/US/tech/tk389/tk621/technologies_tech_note09186a0080094954.shtml`.

[11] Understanding Spanning tree protocol, Cisco website. `http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/sw_ntman/cwsimain/cwsi2/cwsiug2/vlan2/stpapp.htm`.

[12] M. Wadekar. Enhanced Ethernet for Data Center: Reliable, Channelized and Robust. *15th IEEE Workshop on Local & Metropolitan Area Networks (LANMAN'07)*., pages 65 – 71, June 2007.

[13] K. Yeung, F. Yan, and C. Leung. Improving Network Infrastructure Security by Partitioning Networks Running Spanning Tree Protocol. *Int. Conf. on Internet Surveillance and Protection (ICISP'06)*, August 2006. 4 pages.

Digital Object Identifier: 10.4108/ICST.SIMUTOOLS2009.5656
http://dx.doi.org/10.4108/ICST.SIMUTOOLS2009.5656