This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2010 proceedings.

Simple Protocol Enhancements of Rapid Spanning Tree Protocol over Ring Topologies

M. Marchese, M. Mongelli, G. Portomauro

DIST – Department of Communications, Computer and Systems Science
CNIT – Italian National Consortium for Telecommunications
University of Genova, Via Opera Pia 13, 16145, Genova, Italy
{ mario.marchese, maurizio.mongelli, giancarlo.portomauro }@cnit.it

***Abstract*—The paper addresses resilience over Ethernet networks using the *Rapid Spanning Tree Protocol* (RSTP). The topic constitutes an open issue of debate as clear indications on the real RSTP performance can hardly be found from the literature. Actually, the complicated protocol structure makes the analysis intricate and unsuitable for generalization. Moreover, the presence of other resilience algorithms, whose mechanisms and rules are explicitly designed for resilience, solves the problem beyond the application of RSTP. Even though those solutions are actually more efficient than RSTP, they are more expensive. In this perspective, the purposes of this paper are to critically evaluate the intrinsic limitations of RSTP and propose some simple protocol modifications to speed up its performance. The analysis validates the achievable performance as a trade-off between fast reactions and bandwidth overhead.**

Keywords—Network resilience, RSTP, ring topology

I. Introduction

ETHERNET is widely considered as the ultimate solution for deployment of *Metropolitan Area Networks* (MANs). One of the most important issues is *resilience*, namely, the capability to sustain user's traffic in the presence of network faults. The *Spanning Tree Protocol* (STP) and its "*Rapid*" variant (RSTP, [1], chapter 17) are *Logical Link Control* (LLC) modules of Ethernet devices, also known as *bridges* (Brs), used to: **1)** generate a logical tree topology among Brs and **2)** support resilience. They employ dedicated control packets, called *Bridge Protocol Data Units* (BPDUs). Action **1)** is needed to avoid traffic loops over redundant Ethernet network paths. As far as issue **2)** is concerned, the *topology change* (TC) flag of a BPDU is set in order to signal a TC when some alteration of the network is detected (e.g., after a fault).

The rationale of this work relies on the fact that some controversy exists on the real RSTP performance. An unresolved debate is still open on the topic. Some results of the literature outline good performance (e.g., [2, 3]), some others do not [4, 5]. The general flavour of network engineers on RSTP is negative (e.g., [6]), even though recent industrial reports express good performance [7, 8]. In this perspective, the paper investigates how to achieve the best performance by introducing light protocol modifications. The ring topology is taken into account due to its wide diffusion and to simplify the analysis. The performance metric of interest is the time necessary to recover user traffic after a (link or node) failure in a ring topology. The ideal target (denoted by T_c^* in the paper) is $T_c^* = \sum_{l=1}^{n} d_l$, being n the number of Brs in the ring and d_l the l-th link delay plus the Br' processing delay when it generates BPDUs on the port adjacent to link l.

A huge amount of other specialized protocols have been designed to guarantee resilience over Ethernet networks: *Ring RSTP* (RRSTP), Viking, *Ethernet Automatic Protection Switching* (EAPS), *Resilient Packet Ring* (RPR), RSTP with Epochs [2], and, more recently, the ITU-T G.8032 standard, including the GMPLS [3] as well. Specialized protocols are expected to have the mentioned ideal performance (T_c^*). However, as a drawback, they are implemented in the MAN core, while regular STPs are in any case present over the *Local Areas Networks* (LANs). For this reason, the joint adoption of a unique protocol (the RSTP) for both the LAN and the MAN would be recommended. This would reduce costs and simplifies the network management, thus eliminating the need of interworking among different protocols.

The remainder of the paper is organized as follows. In the next section the features of the RSTP protocol are summarized. In section III, the inherent limitations for resilience and the proposed protocol enhancements are detailed. In section IV, a performance evaluation is investigated and, in section V, conclusions and future work are finally outlined.

II. RSTP over Ring: Protocol Features

In the following, the term *regular working condition* defines a period of time in which no fault occurs and configuration BPDUs are sent without any TC indication. Configuration BPDUs maintain the virtual tree connectivity computed by Brs using STP or RSTP. The first element of the tree is called *root* Br. It has an important role for the protocol as described in the following. The fundamental states of a port are: *forwarding* (i.e., the port enables forwarding of traffic frames) and *discarding* (i.e., the port does not forward traffic frames). The *learning procedure* of a Br consists of

This work has been funded by Selex Communications (Selex) S.p.A, Italy and it is part of a scientific project between Selex and the Department of Communication, Computer and System Sciences (DIST) of the University of Genova. Parts of the work have been registered under patent at the 'Camera di Commercio' of Turin, 03 Dec. 2009 (TO2009A000947).

978-1-4244-5638-3/10/$26.00 ©2010 IEEE

memorizing users' addresses within the internal *Forwarding Database* (FDB) by "sniffing" forwarded frames. When a frame arrives on a port, the FDB tells through which port the destination of that frame is reachable. When a destination is not present in the FDB, the packet is "flooded" on all ports in forwarding state. The virtual tree topology, induced by STPs, assures no loops are generated in case of flooding.

In RSTP, some new port roles are defined: *alternate* (as an alternative way to reach the root) and *backup* (as an alternative way to give connectivity to a given LAN). A port with these roles immediately wakes up if a TC is detected. A Br infers a TC if information coming from received BPDUs is not consistent with the port roles of the Br or a fault is signaled directly by the physical layer. After a TC, if the role is 'backup', no other changes must be made on the tree configuration of the network. If the role is 'alternate', a proposal-agreement mechanism is triggered to let Brs build a new tree configuration.

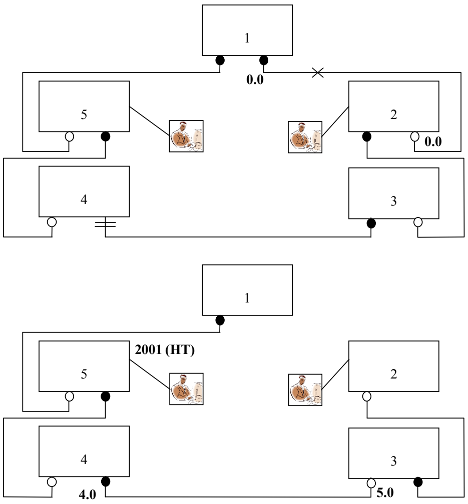


Figure 1. RSTP over 5-nodes ring; (a)-top before, (b)-bottom after the fault.

RSTP ports roles over a 5-nodes ring is depicted in the top case of Fig. 1 (Fig. 1(a)) (the **bold** values of Fig. 1 are explained later in subsection III.C). For the sake of simplicity, the MAC addresses are visualized as the number identifiers of the Brs in all the pictures of the paper. At beginning (when the network is "switched on"), each Br proposes itself as the root; after that, it compares priority of messages coming from other Brs and compares its MAC address with address of other Brs and finally discovers as the root the Br with the higher priority (lowest value) address in the network. For the ring case, the ports role convergence time at beginning (*Tt*) is (*n*/2)+1 (being *n* the number of Brs). After *Tt*, the ports in the direction of the root become "root port" (empty circles, as to the standard notation), all other ports are "designated ports" (ports forwarding traffic, depicted as filled circles), except the port of the Br in the left corner at the bottom, which becomes "alternate" (Br 4 in Fig. 1(a)).

A. The ports role handshake after a fault

Fig. 1(b) represents the new state of the ports after the fault of the link between Br 1 and Br 2. This is the worst case for convergence (i.e., loosing connectivity to the root) [8]. The fault produces a reaction that re-configures some port roles of the Brs; the reaction is called here *ports role handshake* and acts as follows. When Br 2 detects that its right port is disabled, it declares itself as the root; this information is propagated by other bridges until Br 4 is reached, then Br 4 informs that original root is still alive and starts the port role re-assignment on the right side of the ring via an acknowledge mechanism (the mechanism is similar to the one used for the inizialization of the tree); a step-by-step description of the port role handshake is reported in [8]. The final result is depicted in Fig. 1(b). The duration of the ports role handshake (i.e., the convergence time for setting all new ports roles after the fault) is denoted by *Tc* in the paper.

B. The clearFDB operation

It is worth noting that *Tc* is not the time of completion of all operations necessary to forward traffic correctly after the fault. After the *Tc* period the RSTP *Topology Change* (TC) *State Machine* (SM) (section 17.31 of [1]) becomes "active" (i.e., it enters in the "propagating" state) and informs the Port Transmit SM to forward regular BPDUs with the TC flag set to 1. This active flag produces on the receiving Br the flush, or *clearFDB*, operation, whose result is deleting all the addresses learned by the Br in previous working operations. This assures traffic will be correctly forwarded after the fault (an example is reported in subsection III.C).

III. RSTP over ring: Protocol Enhancements

The fundamental timers of the protocol are: *HelloTime* (HT), *Forward Delay* (FD) and *Max Age* (MA). HT defines the periodic retransmissions of BPDUs, default range is [1, 10]s, defaultvalue is 2s. Each Br periodically sends BPDUs every HT from its ports towards its neighbors to maintain the tree connectivity. If the TC SM is active, BPDUs with TC flag set are propagated every HT from root ports to neighbors, until TC SM transits to the inactive state. FD is a delay imposed to some port state transitions, default range is [4, 30]s defaultvalue is 15s; MA is the maximum acceptable period between the receiving of a BPDU and the wake up of a port towards a new state; default range is [6, 40]s defaultvalue is 20s. In the proposed variation of the RSTP, the fundamental timers of the protocol are left to their respective defaultvalues. The rationale of this choice relies on the following circumstances: **1)** refining the timers' values has an impact on STP's convergence, not on RSTP's one; **2)** the main RSTP performance limitation over rings is not due to specific values of these timers [5]; **3)** when dealing with RSTP protocol modifications, these timers are usually left untouched [2]. It is useful to recall that the time granularity of RSTP's action is 1s. It means the RSTP protocol wakes up every 1s and activates the inherent SMes in that moment. The Br's waking up protocol action is known as *stpm_one_second*(·) [9]. The study reported in the following subsections is derived from an extensive inspection made through the "BridgeSim" simulator [9]. In this perspective, some simulations are anticipated to show the most critical features of the protocol.

A. The TxHoldCount effect

As to the standard, TxHoldCount (TxHC) is "*the value used by the Port Transmit state machine to limit the maximum transmission rate*" of BPDUs; default range is [1, 10]s, defaultvalue 6s. When a BPDU is ready for transmission, but the number of previous transmissions in the last second has reached TxHC, the current transmission is delayed of 1s. Some examples help explain its effect on *Tc*.

Without loss of generality, the case of 17 Brs is now considered to highlight all TxHC implications. Br identification numbers and port roles are analogous to the ones of Fig. 1. Table 1 reports *Tt* (the ports role convergence time at the beginning, when the network can be considered "switched on"), together with the time of the last registered clearFDB (i.e., after that time the TC SM of all Brs becomes inactive) as functions of TxHC (no fault is produced in Table 1). Every fault introduced before the inactivity of the TC SM (Tables 2 and 3) makes the *Tc* performance very sensitive to the TxHC. Note that, as already said, a period of TC activity is necessary also after a fault. Thus, every successive fault (or network element recovery) during TC SM activity leads to higher values of *Tc* than a fault without TC SM activity. In this perspective, numerical examples in Tables 2-4 show that the number of BPDUs propagation during TC SM activity decreases with time and so does the negative effect of TxHC.

TxHC		*Tt* [ms]		Last clearFDB [s]
2		8001		40
3		7001		38
4		5002		38
5		5001		36
6		4001		36
7		3001		34
8		1002		34
9		9		32
10		9		32
disabled		9		32

Table 1. TxHC effect – 5-nodes ring: *Tt* performance and last clearFDB after completion of the tree at beginning.

TxHC		*Tc* [ms]
2		5009
3		5009
4		3009
5		3001
6		2009
7		1009
8		1009
9		18
10		18
disabled		18

Table 2. TxHC effect – 5-nodes ring: *Tc* performance with fault at 4s.

TxHC		*Tc* [ms]
2		2009
3		2008
4		1009
5		18
6		18
7		18
disabled		18

Table 3. TxHC effect – 5-nodes ring: *Tc* performance with fault at 8s.

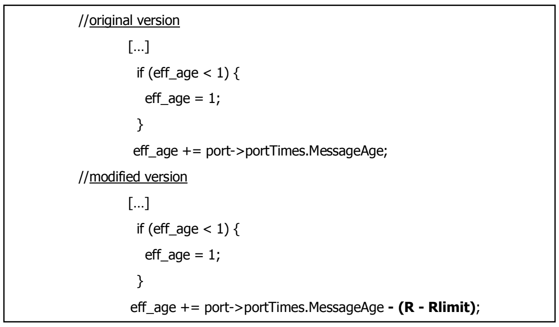
TxHC		*Tc* [ms]
2		1003
3		18
4		18
7		18
disabled		18

Table 4. TxHC effect – 5-nodes ring: *Tc* performance with fault at 200s.

As a result of this analysis, disabling TxHC on the ports interconnecting the ring is the first protocol modification needed. TxHC increments during time can be disabled with a small modification of the Port Transmit SM. Eliminating the 'txCount+=1' instruction (see Fig. 17-17 at page 172 of [1]) within the "Transmit_config", "Transmit_TCN", "Transmit_RSTP" states of the SM is sufficient.

B. The message_age effect

As to the standard: "[…] *each* BPDU *includes a message age and a maximum age. The message age is incremented on receipt, and the information discarded if it exceeds the maximum. Thus the number of bridges the information can traverse is limited*". This is necessary to avoid that old BPDUs do not endlessly circulate through redundant paths and prevent propagation of new BPDUs. In practice, using defaultvalues of the HT, FD and MA timers and incrementing the message_age (of 1s) at each Br limits the dimension of the ring to no more than 18 Brs. The consistency check equation on the message_age (subsection 17.21.23 of [1]) and its rationale are rather complicated (they maintain compatibility with STP) and are not detailed here. However, to avoid the message_age limitation, a simple modification to the Port Info SM is sufficient. This means modifying an internal timer, the "rcvdInfoWhile", in that SM. The rcvdInfoWhile is the time remaining before the information held on a port expires if no other BPDUs are received on that port. Intuitively, if rcvdInfoWhile=HT on a port, that port is not capable to maintain consistent relationship with the other port of the adjacent Br (i.e, the BPDU does not arrive from the adjacent Br before the rcvdInfoWhile timer expires); this happens over a given ring size (the "**Rlimit**" threshold below). The updating equation of the rcvdInfoWhile in the Port Info SM can be slightly modified, as in Fig. 2, in order to set the ring size.



```
//original version
        [...]
         if (eff_age < 1) {
           eff_age = 1;
         }
        eff_age += port->portTimes.MessageAge;
//modified version
        [...]
         if (eff_age < 1) {
           eff_age = 1;
         }
        eff_age += port->portTimes.MessageAge - (R - Rlimit);
```

Figure 2. UpdtRcvdInfoWhile() to set the maximum ring size.

The **R** quantity is the desired ring size and **Rlimit** is the ring limit, which depends on the HT (**Rlimit**=18 with HT=2, **Rlimit**=19 with HT=1). Such a modification assures that rcvdInfoWhile (whose entire updating equation is not reported in Fig. 2) is always above HT when ring size is **R**.

C. The clearFDB effect

As mentioned above, after a topology change is detected after a fault, the TC SM enables BPDUs transmission with the TC flag set to trigger clearFDB operations throughout the network. Traffic is not correctly forwarded until these operations are made by all Brs. An example is shown in Fig. 1, where the times (in ms) of clearFDB are in **bold**. Note that Br 5 flushes its ports only after 2001ms (one HT, bottom case of Fig. 1). The rationale of this behavior relies on the TC SM' time granularity that depends on the HT. As a consequence, indications of clearFDBs are propagated throughout the network with that granularity. The same period of time is seen by applications depicted in Fig. 1 as a period of lost frames because Br 5 needs two seconds (one HT) to stop forwarding traffic on the wrong direction (left port) after the fault. On the other hand, since the right side of the network converges in 5ms, if the application located at Br 2 forwards traffic in the right direction just after *Tc*, Br 5 is immediately trained to forward traffic in the right direction. To summarize, the period of lost traffic depends on the real behavior of the application. The same concept holds true for the 11 Brs example in Fig. 3: even if Br 11 receives indication of clearFDB after 7001ms, the right side is ready to correctly forward traffic in 12ms.

A simple solution is the activation of the protocol (by its main function, which regulates the internal SMes, called '*stpm_one_second*(·)') with a finer time granularity. This means letting clearFDB propagation be much faster than in regular situations, by drastically decreasing the time granularity of RSTP. In this view, let Δ_{ST} be the repetition frequency of the stpm_one_second(·) over time. As to the RSTP protocol, the HT parameter defines how many Δ_{ST} a port waits before transmitting a BPDU. If HT=1, BPDUs are transmitted every Δ_{ST}. Note that this solution recalls to some extent the "heartbeat" continuity check (sending control packets with high frequency, e.g., every 1ms), used by EAPS or ITU-T G.8032 to supervise ring connectivity and react to faults.

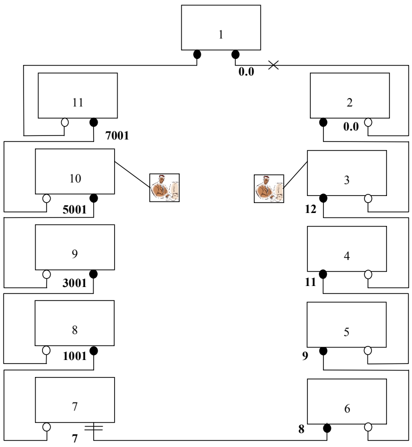


Figure 3. RSTP over 11-nodes ring: fast clearFDB action on the right side.

As a drawback of accelerating the RSTP, the ports role handshake produces a burst of BPDUs whose bandwidth overhead should be accurately evaluated to guarantee priority to RSTP over regular data packets or less critical signalling (e.g., Ethernet GVRP, GARP, GMRP signalling).

D. The ultimate performance

All the proposed modifications to the protocol are defined here as *RSTP over Ring* (RSTPoR). The ultimate performance to be optimized is the *Tc* (the durantion of the ports role handshake), plus the time necessary for the execution of the clearFDB operation in all Brs of the network. More specifically, let T_{cFDB} be the time of the first clearFDB of the last Br executing the clearFDB after the fault; this Br is the one adjacent to the root on the opposite side of the link fault (Br 5 in Fig. 1, Br 11 in Fig. 3). T_{cFDB} is the crucial time period corresponding to the duration of lost traffic seen at the application level.

IV. Performance Evaluation and Discussion

Table 5 outlines the RSTPoR performance as function of both Δ_{ST} and ring size. Link delay is 1ms for all links. HT is set to 1. Having in mind the need of speeding up RSTP performance up to T_c^*, results of Table 5 can be summarized as follows. If Δ_{ST} =5ms, T_{cFDB} is upper bounded by $2.4 \cdot T_c^*$ and, for the Δ_{ST} =1ms case, $T_{cFDB} \cong T_c^*$ below 40 Brs, and, in the worst case (70 Brs), 15ms should be added to T_c^* to obtain T_{cFDB}, thus leading to $T_{cFDB} \cong 1.2 \cdot T_c^*$.

Figs. 4 and 5 define a practical guide to drive network manager in tuning the RSTPoR solution, by also considering the BPDU rate metric. Fig. 5 represents the BPDU rate produced by the accelleration of the stpm_one_second(·) under the chosen Δ_{ST} values. The rate is computed as {(1/*Tc)*·(maxBPDU· DimFrame)}, where 'DimFrame' is the dimension of an Ethernet frame carrying a BPDU and 'maxBPDU' is the maximum number of BPDUs generated by a ring port during *Tc*; actually, *Tc* is the time horizontal with the largest number of generated BPDUs corresponding to the ports role handshake process. Values in Fig. 5 correspond to the highest rate registered on a specific port of a specific Br, as not all the Brs produce the same number of BPDUs during the handshake. RSTPoR performance must be therefore considered jointly with the signaling overhead. It is firstly necessary to find the desired fault recovery time with respect to the ring size. A value of Δ_{ST} can be found to match the desired performance (Fig. 4). Secondly, the inherent signaling overhead of that Δ_{ST} can be found in Fig. 5. With Δ_{ST} =1.0ms, the bandwidth overhead corresponds to an additional rate of 18% over the *continuity check* (CC) rate, in which one BPDU is sent every 1.0ms, corresponding to a rate of 424 kbps. The CC rate is common to RSTPoR, EAPS and G.8032. This limited amount of overhead leads to considering the RSTPoR solution acceptable for a wide range of practical cases where the CC is applied as well.

V. Conclusions and Future Work

The paper studies the fault recovery performance of the *Rapid Spanning Tree Protocol* (RSTP) over ring topologies. The analysis highlights insightful features of the protocol and explains how to speed up traffic recovery as seen by the application level. Future work mainly deals with the extension of RSTPoR to mesh topologies, by considering the RSTP ‘with epochs’ principle of [2], together with other tuning techniques.

References

[1] IEEE Std 802.1D™-2004, IEEE Standard for Local and metropolitan area networks, Media Access Control (MAC) Bridges.

[2] K. Elmeleegy, A.n L. Cox, T. S. Eugene Ng, “On Count-to-Infinity Induced Forwarding Loops in Ethernet Networks,” Proc. of IEEE *Infocom 2006*, vol. 25, no. 1, 23-29 Apr. 2006, pp. 1699-1711.

[3] S. Ilyas, A. Nazir, F. S. Bokhari, Z .A. Uzmi, A. Farrel, “A Simulation Study of GELS for Ethernet over WAN,” Proc. of IEEE *Globecom 2007*, 26-30 Nov. 2007, pp. 2617 - 2622.

[4] J. Madsen, D. Tebben, A. Dwivedi, P. Harshavardhana, W. Turner, “Cross Layer Optimization in Assured Connectivity Tactical Mesh Networks,” Proceedings of the IEEE Military Communication Conference, *Milcom 2008*, San Diego (CA), 17-19 Nov. 2008.

[5] A. Myers, T. S. Eugene Ng, H. Zhang, “Rethinking the Service Model: Scaling Ethernet to a Million Nodes,” Proceedings of the *3rd HotNets Conference*, Nov. 2004.

[6] T. Gimpelson, “Metro vendors question Spanning Tree standard,” Network World White paper 2001, http://www.networkworld.com/archive/2001/123588_08-06-2001.html.

[7] D. DesRuisseaux, “Use of RSTP to Cost Effectively Address Ring Recovery Applications in Industrial Ethernet Networks,” Proceedings of the *13th ODVA Network Conference*, Howey-in-the-Hills, FL, USA, Feb. 2009.

[8] M. Galea, “Rapid Spanning Tree in Industrial Networks,” RuggedCom Inc. - Industrial Strength Networks, white paper, 2004, http://www.ruggedcom.com/pdfs/white_papers/rapid_spanning_tree_in_industrial_networks.pdf.

[9] BridgeSim: C++ Rapid Spanning Tree Protocol (RSTP) simulator, http://www.cs.cmu.edu/~acm/bridgesim/index.html.

Acknowledgments

The authors express their gratitude to Selex Communications, in the persons of M. Carniglia, A. Civardi, G. Fontana, A. Sogliani and L. Spinacci, for their helpful and constructive suggestions during the development of the work.

Ring size Δ_{ST} [ms]	10	20	30	40	50	60	70
1000	2001	8001	12001	18001	23001	27001	32001
500	1001	4001	6001	9001	11501	13501	16001
200	401	1601	2401	3601	4601	5401	6401
100	101	801	1201	1801	2301	2701	3201
50	101	401	601	901	1151	1351	1601
10	21	81	131	191	251	301	361
5	16	46	76	106	136	166	196
1	10	21	34	47	61	73	85

Table 5. T_{cFDB} [ms] as function of Δ_{ST} and ring size.

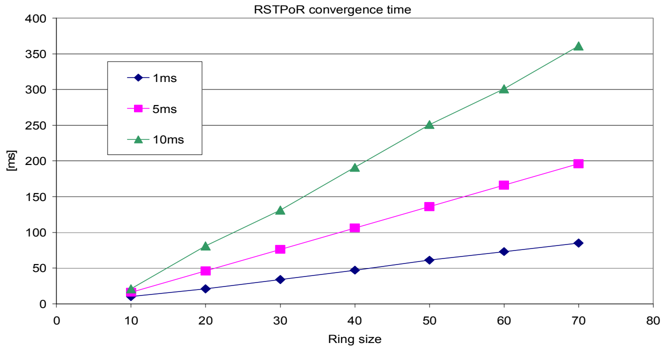


Figure 4. RSTPoR: T_{cFDB} as function of Δ_{ST} and ring size.

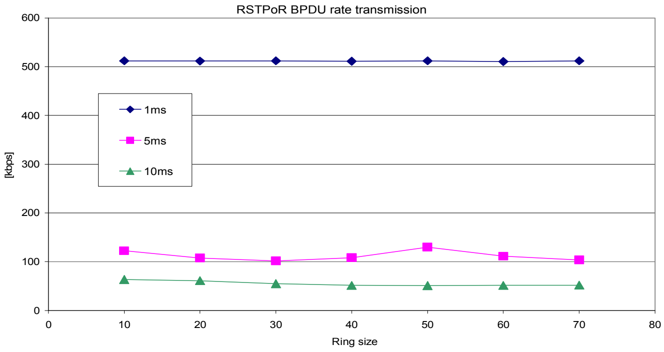


Figure 5. RSTPoR: signalling overhead as function of ring size and Δ_{ST} .