Performance evaluation of an enhanced bridging algorithm in RPR networks

Amund Kvalbein, Stein Gjessing and Fredrik Davik
Networks and Distributed Systems Group
Simula Research Laboratory
Oslo, Norway
Email: {amundk, steing, bjornfd}@simula.no

Abstract

Support for bridging is one of the requirements in the standardisation of Resilient Packet Ring (IEEE P802.17) currently conducted by the IEEE. The bridging algorithm in the present draft standard, known as *basic bridging*, has obvious weaknesses regarding bandwidth efficiency and scaling properties. An improved bridging strategy, termed *enhanced bridging* has been presented to the RPR working group. We have detailed out and implemented this enhanced algorithm. Basic and enhanced bridging are compared analytically and through simulations. We show that the enhanced bridging algorithm more than doubles the bandwidth efficiency for bridged traffic compared to basic bridging. Furthermore, we show that the efficiency improvements give significantly better latency characteristics in a network with typical Internet background traffic.

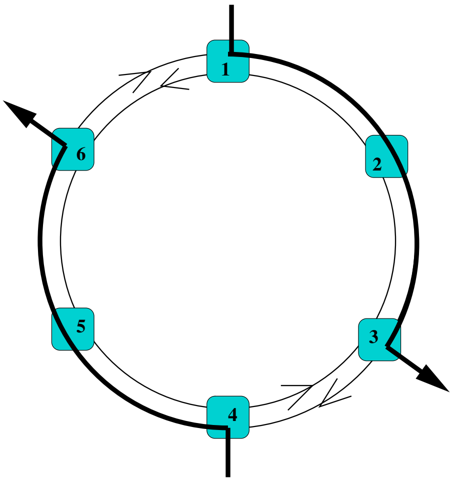


Figure 1: Spatial reuse allows several packets to be sent on different spans of the same ringlet at the same time.

1 Introduction

Resilient Packet Ring (RPR) is an architecture for a packet based dual ring network. An RPR ring consists of two counter-rotating ringlets, giving each station on the ring a full duplex connection to each of its neighbours. RPR is primarily intended to be a technology for the metro area environment, supporting link capacities up to multiple gigabits per second.

RPR uses destination stripping, meaning that packets are stripped from the ring upon reaching the destination station. Destination stripping allows *spatial reuse*, which is an important characteristic of RPR. Spatial reuse gives better bandwidth utilization, by allowing each packet to travel only part of the circumference of the ring. This way, several packets can be in transit on different spans of the same ringlet at the same time. Spatial reuse is illustrated in figure 1. Each station on an RPR ring maintains a topology image, with information on the cost of reaching each other station on the ring through either of the ringlets.

Several RPR rings can be interconnected by *transparent bridges* [1]. Transparent bridges are defined by the IEEE 802.1D standard [2]. Bridges are devices with two or more network interfaces, called ports, that forward data packets from one port to one or more of the others. The presence of transparent bridges are traditionally invisible to the stations in a network, allowing them to communicate without paying attention to whether the network is bridged or not. As will be explained in the sequel, this is not possible in bridged RPR networks. Transparent bridging demands some adaptations in the RPR standard, due to the spatial reuse properties of RPR. Two main approaches have been proposed, termed *basic* and *enhanced* bridging. The major shortcoming of basic bridging, is that it leads to loss of the spatial reuse properties for bridged traffic. Enhanced bridging was developed as a response to this. In this paper, we discuss the two different strategies for RPR bridging, and present simulation results that show their differences with respect to network utilisation.

The remainder of this paper is organised as follows. Section 2 explains the need for special treatment of bridged traffic in RPR, and discusses basic bridging as a solution to this. In section 3, we introduce our detailed version of enhanced bridging, which maintains spatial reuse also for bridged traffic. Then, section 4 presents two generic simulation scenarios that show important differences between the two bridging algorithms, and illustrate the performance gain achieved by enhanced bridging. Finally, section 5 sums up and concludes this work, and points out some ideas for further research.

2 Basic bridging in RPR

Transparent bridging was originally developed for an Ethernet environment. Ethernet is a *shared medium* network, in which all packets transmitted on the transmission medium are available to all stations. In a shared medium network, no extra measures need to be taken to make sure all packets reach the intended bridge. A bridge in such a network can operate in a *promiscuous mode*, where it picks up all passing packets and forwards them according to a filtering database. The filtering database prevents waste of network resources by only forwarding packets on the correct port. The entries in the filtering database can be learnt dynamically by inspecting the source address of the received packets, and associating that address with the port it arrived on. Transparent bridging relies on the Spanning Tree Algorithm [3] to make sure there are no cycles in the network graph.

An earlier ring technology, Token Ring [4], also relies on shared medium properties to do bridging, even if the bridging strategy itself, named Source Route Bridging [5], is completely different from transparent bridging. In Token Ring, all packets that are transmitted by a station continue around the entire ring, and are stripped by the source. All the packets are thus available to all stations on the ring, just like in an Ethernet. Bridges on a Token Ring check each packet for the presence of a special bit that is used to mark packets eligible for bridge forwarding. With Source Route Bridging, the source station is responsible for calculating the whole path of a bridged packet, and to include it in the packet header.

RPR is *not* a shared medium network. On the contrary, spatial reuse is one of the important features in RPR, as explained above. A packet normally travels only the shortest path between source and destination, hence it is not seen by all stations on the ring. This prevents the learning process in a bridge from working correctly. Recall that the learning process in a bridge relay depends on inspecting the source address in the packet headers. If these packets are not available to all bridges, the filtering databases will not be built to a complete state. This can lead to persisting forwarding of frames to the wrong networks. This wastes network resources, and is not consistent with the IEEE 802.1D standard.

The main idea of basic bridging is to make the RPR ring behave like a shared medium network for packets that traverse more than one ring, while keeping the spatial reuse properties for local traffic. Basic bridging makes sure that packets that are either sourced in or destined for a remote network, termed *remote* packets, are always seen by all stations on all rings they pass through. By doing this, the basic bridging algorithm makes RPR emulate shared medium behaviour for remote traffic.

Basic bridging makes remote packets available to all stations by using a mechanism called *flooding*, which can be looked upon as a special kind of broadcast. Flooded packets are marked with a flooding marker in the packet header, and are not stripped from the ring at the destination like normal packets. Instead, a time-to-live field in the packet header is used to limit the scope of the packet.

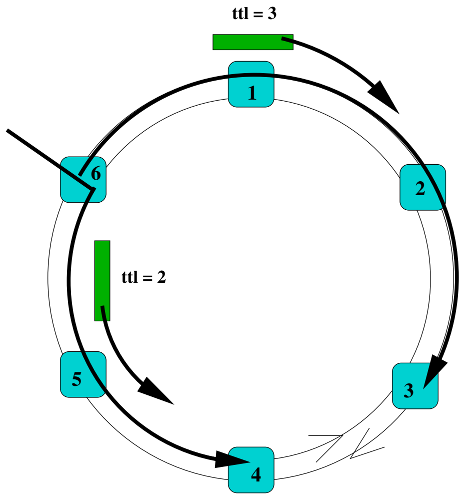


Figure 2: The ttl of a flooded frame is set so that it reaches each station exactly once. With bidirectional flooding, a separate copy of the frame is transmitted on each ringlet.

With the basic bridging approach, bridges no longer operate in a promiscuous mode. Only packets with the flooding marker set are presented to the bridge relay by the RPR MAC, and considered for forwarding. This eases the demand for processing power in the bridges, since they no longer have to make a lookup in the filtering database at line rate. The idea of marking the frames to be considered for forwarding with a special marker, resembles the strategy used in Source Route Bridging found in Token Ring networks.

Basic bridging demands some special treatment of remote packets by the sending stations. Remote traffic is not transmitted shortest path on one of the ringlets like local traffic; instead it is flooded on all the rings it passes. This means taking one step away from the original transparent bridging principles in an Ethernet environment, where remote and local traffic are treated alike by the end stations. This way, some of the bridging logic has been distributed in the RPR stations, and the bridging can no longer be called transparent in the traditional sense.

The main achievement of basic bridging is to make the RPR ring behave like a shared medium network for remote packets, while keeping the spatial reuse properties for local traffic. By always flooding remote traffic, basic bridging allows correct learning in the bridge relays. The major problem with this approach, is the poor scaling properties, as all spatial reuse properties are lost for remote traffic. As we will show in section 4, the extra load put on a network by the flooding can be considerable.

3 An enhanced bridging algorithm

Several methods have been proposed to make bridging in RPR more efficient. Such proposals have been outlined in presentations in the RPR working group [6, 7, 8]. We have detailed out the ideas in these presentations with respect to packet format, table formats and reception rules. Based on this, we have implemented an enhanced bridging algorithm for RPR.

The purpose of enhanced bridging is to allow spatial reuse also for remote traffic. This is achieved by letting the RPR stations address remote frames directly to the bridge responsible for forwarding traffic to the specific receiver. To achieve this, every station must keep and maintain a table that maps a remote address to a local (bridge) address. The functionality that builds, maintains and uses these tables is known as the Spatial Reuse Control Sublayer (SRCS)[1]. The remote-to-local mapping tables are therefore termed SRCS tables in the following.

Figure 3 shows an example network with SRCS tables in some of the stations. Note that the bridge ports can be seen as stations in their respective networks. Hence, each bridge port that belongs to an enhanced bridging network contains a SRCS table, just like any other station in the network. In addition, the bridges contain the normal filtering database in their relay unit.

In addition to the SRCS tables in the stations, the enhanced bridging algorithm depends on some way to identify a packet's local sender and destination. Several different schemes have been proposed to the RPR working group to achieve this [9, 10, 11]. Most of the proposals introduce an extra pair of local source/destination addresses in the RPR header, in addition to the global addresses. The size and placement of these local identifiers vary. A bridge that relays a packet onto an RPR ring, inserts its own address as the local source in the packet header. Similarly, the bridge address is used as the local destination address for packets destined beyond that bridge. Note here that a way to identify the local source of a packet is also needed with basic bridging, to prevent duplication or reordering of packets. Hence, enhanced bridging does not introduce any extra packet overhead compared to basic bridging.

The enhanced bridging algorithm relies on flooding in the initial phase of a bridged traffic session, before the SRCS tables are built to a complete state. As the packets make their way through the network, the bridges and end stations involved in the packet transport learn the association between local and remote addresses by comparing the global and local source addresses. Once the learning phase is complete, no more flooding is needed during normal operation. The source and the bridges on the packet's path will use their SRCS tables to decide the next hop, and can send the packets directly using their topology image.

As will be confirmed in the sequel, the enhanced bridging algorithm allows spatial reuse for remote traffic after an initial learning phase. The primary price to pay is the need for some amount of memory in the RPR stations to keep the SRCS tables. The SRCS tables can probably be kept relatively small and simple, so the need for memory can be limited. The size of these tables will be O(n), where n is the number of stations in the bridged network.

Some techniques can be imagined that would reduce the amount of flooding in the learning phase. First, it is reasonable to believe that many rings only contain one or at least very few active bridges. An active bridge here means a bridge whose port to this ring has not been blocked by the spanning tree algorithm. The SRCS tables could then be equipped with a default bridge address, decided during topology discovery. Frames destined for remote nodes that did not match any of the other entries in the SRCS table, would be sent to this default bridge. The bridge would then be responsible for the further handling of the frame. If the bridge knew the location of the destination node, the frame would be forwarded as normal. Else, the bridge would flood the frame on all ports, including the one it arrived on. Another possibility is to run a "global" topology discovery algorithm when several RPR rings are connected by bridges. This way, the SRCS tables in the end nodes could be built complete before any traffic was sent in the network. This strategy would, however, generate more traffic in the network when building and maintaining the global topology image.

The enhanced bridging algorithm goes even further than the basic in adding bridging logic to the RPR stations on the ring. The source stations are now responsible for calculating the first hop on the remote packet's path. This way, enhanced bridging has a similarity with Source Route Bridging, where the source is responsible for including the packet's whole path in the header.

Note that enhanced briding capable RPR stations can be introduced gradually in a network where basic bridging is used. In such a network, basic bridging stations will flood all packets as normal. All bridge ports on a network segment containing stations with enhanced briding capabilities must also support enhanced bridging. On an RPR ring with mixed bridging strategies, spatial reuse will be achieved for traffic between two stations (bridge-bridge or end station-bridge) that support enhanced bridging.

4 Performance evaluation

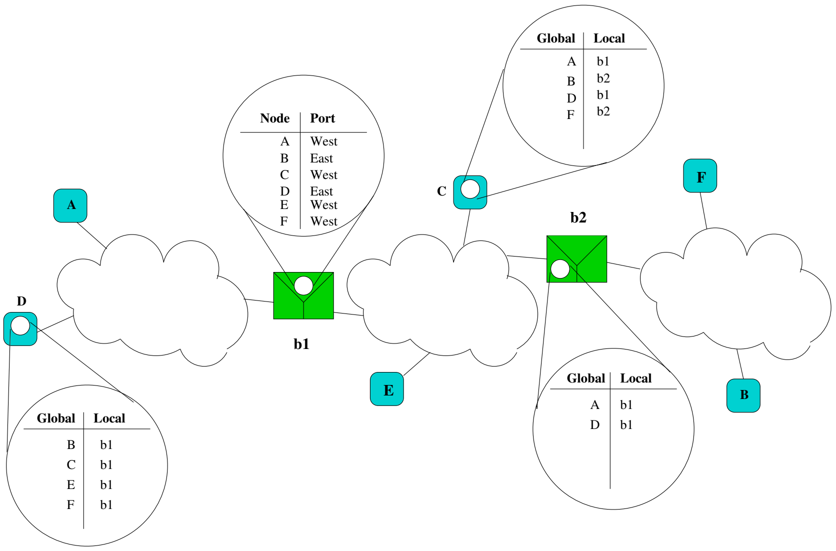
Simulations have been performed as a basis for comparing the basic and enhanced bridging algorithms. The simulations are performed using a simulation model of an RPR network developed at the Simula Research Laboratory in Oslo. The model is written in the Java programming language, and it allows simulation of scenarios with different link speeds, ring sizes, technical solutions, placement of rings and bridges etc. [2]

Two simulated scenarios are presented. The first one illustrates the learning phase with enhanced bridging, and shows the efficiency gain achieved with respect to bandwidth. The traffic in this scenario has deliberately been

---

[1] The term SRCS is adapted from a presentation given by the RPR working group's bridging ad-hoc subcommittee [7]

[2] The simulation model can be downloaded at http://software.simula.no/nd/



Node	Port
A	West
B	East
C	West
D	East
E	West
F	West

Global	Local
A	b1
B	b2
D	b1
F	b2

Global	Local
A	b1
D	b1

Global	Local
B	b1
C	b1
E	b1
F	b1

Figure 3: With enhanced bridging, all stations are equipped with tables that map global remote addresses to a local bridge.

kept very simple, as the purpose is to investigate the load posed on the network by a single data exchange. This approach allows us to easily compare the performance of the two bridging algorithms. The second scenario illustrates how the choice of bridging strategy affects the local ring traffic. This scenario offers a more complex traffic pattern, where the effect of more realistic LAN/Internet background traffic upon a data flow is investigated.

4.1 Simple flow scenario

Enhanced bridging makes spatial reuse possible after an initial learning phase, when the SRCS tables in the bridges and end stations have been built. Before the learning phase is completed, packets are flooded through the network. This learning phase must take place when a station first starts communicating on the network. The purpose of this scenario is to illustrate this learning phase, and to show how the amount of flooding develops with the basic and enhanced bridging strategies respectively.

In the general case, two stations in a bridged RPR network will have zero or more rings between them, and zero or more rings in the network that lies outside the path between the stations. A simple and reasonable special case of such a network is used in this scenario, with one intermediate ring, and one additional ring outside the source-destination path.

The topology used is shown in figure 4. Four RPR rings, each with eight stations, are connected by three bridges. The links between the stations are 5km long, and the link capacity is about 2.5 Gbit/s (OC-48). The simulation has been repeated with different link lengths and capacities, with very similar results.

The traffic pattern in this scenario is very simple. Station s1 sends a continuous flow of 1kB data packets to station s2 at one third of the link capacity. For each packet received, station s2 returns a small (80 byte) packet to station s1. Thus, we have two packet flows in this simulation: a flow of 1kB “data frames” from s1 to s2, and a flow of 80B “acks” in the opposite direction.

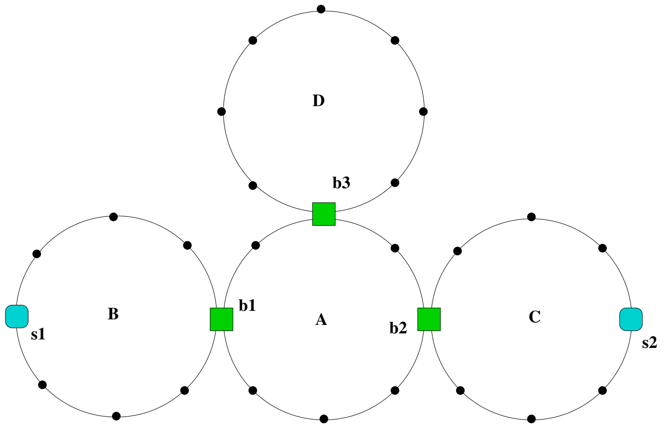


Figure 4: Two stations send traffic through a network of four RPR rings connected by three bridges.

The same traffic scenario is simulated twice, using the basic and the enhanced bridging algorithms respectively. The packets in the simulations are sent at equal intervals. Hence, the results are exact, without any differences caused by random variations.

The first performance metric chosen for this scenario, is the number of *floods*. A flood is performed either by the source station, or by a bridge forwarding the frame onto the next ring. That is, the flooding count tells how many times *any* of the stations, bridge or end station alike, floods a frame on *any* of the rings.

With this topology and sending pattern, the flooding characteristics with basic and enhanced bridging should

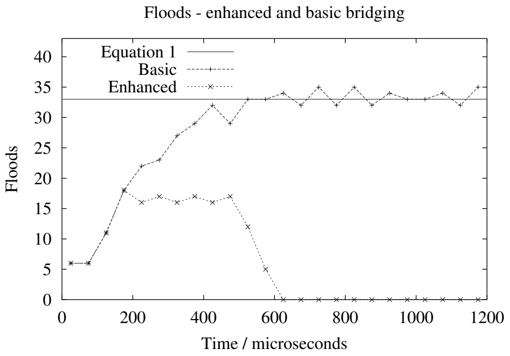


Figure 5: Number of floods per 50 μs. The horisontal line marks the value calculated in (1)

be predictable. With basic bridging, the data frames will be flooded over all rings in the network. The flooding on ring D will stop once the first ack reaches bridge b3, and the position of s2 is learned. When the s1-s2-s1 pipe is filled with packets, each data frame and ack will be flooded three times; once on each ring. With s1 sending at 1/3 of the line rate, and s2 responding with an equal amount of acks, the basic bridging flooding count for the two flows should stabilize at a level as shown by equation (1). Enhanced bridging will never flood the data packets on ring C, and the acks are never flooded. The maximum flooding count for enhanced bridging should therefore be reached when packets are flooded on rings A, B and D, and then gradually be reduced to zero as the acks propagate through the network.

$$\begin{aligned} 2*3*\frac{1}{3}*\frac{330MB/s}{1kB/flood} &= 6.6*10^5 floods/s \\ &= 33 floods/50\mu s \qquad (1) \end{aligned}$$

Figure 5 shows the total number of floods on the four rings, per 50 μs interval. As expected, the figure shows that the enhanced bridging algorithm reduces and eventually eliminates the amount of flooding in the network, after the initial period of learning. With the basic algorithm, the flooding count reaches a stable state when the s1-s2-s1 pipe is filled with traffic.

Some key events decide the shape of the graphs in figure 5. Station s1 starts flooding packets on ring B at time zero. The first packet reaches bridge b1 after 100 μs, and b1 starts flooding packets on ring A. The flooding count increases even faster when the first frames are flooded on ring D by bridge b3 after 150 μs. At time 200, the first packet reaches bridge b2. From this time the difference between the two algorithms becomes clear. The basic algorithm floods packets on ring C, while the enhanced algorithm forwards them directly to station s2 using the topology image. The position of s1 is now known to all the stations involved in the traffic session. Hence, the acks returned from s2 never have to be flooded with the enhanced bridging algorithm, and the flooding count does not increase further. The acks from s2 reach b1 after 500 μs. Bridge b1 then learns that traffic to s2 goes through b2, and can stop flooding those frames. After 600 μs, the first ack from s2 reaches s1, and the learning process is complete. No more flooding occurs from this time with the enhanced bridging algorithm. With the basic algorithm, the acks from s2 must be flooded on all rings. This causes the flooding count to keep increasing after 200 μs. The increase is temporarily reversed when bridge b3 learns the location of s2 after 450 μs, but it continues when the acks are flooded onto ring B after 500 μs. With the basic bridging algorithm, the flooding count stabilises on a high level when the whole s1-s2-s1 pipe is filled with packets in transit.

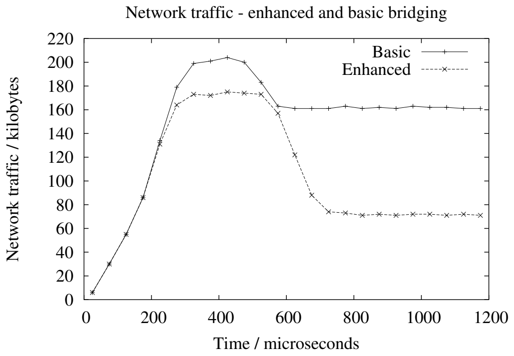


Figure 6: Total network traffic per 50 μs

The elimination of flooding with the enhanced bridging algorithm, gives a more efficient use of the bandwidth resources. Figure 6 shows the total load imposed on the network by the exchange of data between stations s1 and s2. The total load is calculated as the number of bytes that passes through all the links in the network per 50 μs interval, that is the total throughput for all the links.

The graphs in figures 5 and 6 show that the potential gain in bandwidth efficiency for remote traffic is significant with enhanced bridging. In this scenario, enhanced bridging gives a 56% reduction in the total load posed on the network, after the initial learning period.

A period of learning is needed before the enhanced bridging algorithm achieves spatial reuse for remote traffic. This learning period ends after one round-trip-time in the communication between stations s1 and s2. When a frame from s1 has reached s2, and the first reply message has returned to s1, all the necessary learning in bridges as well as in end stations is finished.

Note that a topology with more intermediate rings on the path between s1 and s2 would give a longer round-trip-time, and thus a longer learning period for the enhanced bridging algorithm. But the eventual result would be the same - no flooding on any of the rings. With basic bridging, the packets would have to be flooded on each additional intermediate ring. This way, enhanced bridging helps on the scaling properties of a bridged RPR network. Adding more rings to the network that are not on the s1-s2 path (like ring D), would amplify the load peak seen in figure 6. But when the learning in the bridge relay has finished, such extra rings will not affect the amount of flooding with either of the two bridging strategies.

The active stations and relaying bridges in this scenario are placed at the opposite ends of the respective rings. This means that a frame that travels the shortest path from s1 to s2 still has to traverse half of ring B, then half of ring A, and finally half of ring C. This is a worst-case scenario for enhanced bridging, giving a total station hop count of twelve for each frame. If the active stations were placed otherwise on the rings, the difference in throughput between basic and enhanced bridging would further increase.

A drawback with the enhanced algorithm in this scenario is that bridge b3 never sees any of the acks from s2. Hence, no entry for s2 is ever recorded in the filtering database. If a new RPR station starts transmitting to s2, this traffic would again be forwarded by b3, until the necessary learning is completed. This problem can easily be overcome, by always letting an enhanced bridge flood the first packet it receives from a previously unknown station. This way, all bridges on ring A will learn that packets from station s2 goes through b2.

Note that in case of a bridge failure, the entries in the SRCS tables becomes incorrect. The spanning tree protocol will calculate a new spanning tree, and the association between remote host and local bridge might no longer be correct. Hence, the learning process must be repeated for all nodes in case of a bridge failure, leading to a period of potentially massive flooding in the network.

4.2 Complex traffic scenario

The main objective of the second scenario is to illustrate how the choice of bridging strategy influences the local traffic on an RPR ring. This is done by letting a local best-effort traffic flow compete for bandwidth with remote traffic. Traffic sources sending remote traffic, are distributed around the ring. It is not important whether the traffic sources are bridges or end stations. The amount of this remote traffic competing for resources with the local flow, depends on whether the traffic is flooded or not, i.e. whether basic or enhanced bridging is used.

Studies of traffic traces in an Ethernet environment show that this traffic is statistically *self-similar* [12]. The same is true for wide-area/Internet traffic [13]. The intuitive characteristic of such traffic is that there is no natural length of a “burst”. Independent of the time scale of the measures, similar-looking traffic bursts in the traffic traces are observed.

Self-similar traffic can be modelled by superpositioning several ON/OFF packet sources, if the length of the ON and OFF periods of these sources have infinite variance [14]. Probability distribution functions with high or infinite variance are often called *heavy-tailed* distributions, due to the shape of their graph. One such heavy-tailed distribution is the Pareto distribution given in (2).

An active station in this scenario contains 10 ON/OFF packet sources, with strictly alternating ON and OFF periods. The length of the ON and OFF periods are drawn from a Pareto distribution with $\alpha = 1.2$. The b parameter is used to regulate the traffic intensity, through varying the lengths of the OFF periods of the packet sources.

$$P(x) = \frac{\alpha b^{\alpha}}{x^{\alpha+1}} \qquad x \geq b \tag{2}$$

The topology in this scenario is illustrated in figure 7. The active stations in this configuration are placed on ring A. These are connected to the outside world through station R, which is placed on ring B. The two rings are connected by a bridge, named b. This way, all traffic between the stations on ring A and the Internet must pass through bridge b.

Each of the rings consists of eight stations. These are linked with cables of about 1000 metres, giving a total ring span of about 8 km. The links have a capacity of 8 Gbit/sec.

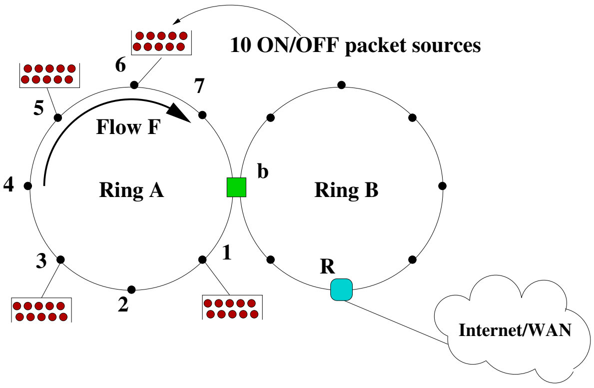


Figure 7: The active stations in this scenario are connected to the outside world through a router placed on the other side of a bridge.

Stations 1, 3, 5 and 6 on ring A produce self-similar traffic destined for station R on ring B. As mentioned above, these stations may or may not be bridges. At the same time, station 4 sends a continuous flow of 500 byte frames at 20% of the link capacity to station 7 on the outer ringlet. This constant stream of frames is named flow F. The traffic load in this scenario is such that packets in flow F always get through, they are never discarded at the ingress of the ring due to buffer overflow.

The latency and jitter characteristics of flow F are then measured with increasing background traffic intensity. Latency in this scenario is measured as the time it takes from a frame is at the head of the MAC client add queue in the source station, to the frame is received by the MAC client at the destination. This time will vary depending on how long the frames have to wait for transmission in the ingress buffer in station 4, and the transit buffers in stations 5 and 6. These waiting times are dependent on

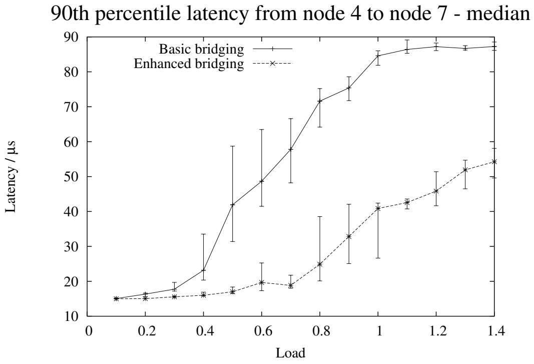


Figure 8: Median latency for flow F with 95% confidence interval

the amount of traffic on the outer ringlet between nodes 4 and 7. As seen in the first scenario, enhanced bridging reduces the traffic load on the ring produced by remote traffic. It is therefore reasonable to expect the experienced latency to be lower with enhanced bridging than with basic.

The sending of packets in flow F goes on for about 4 ms, which gives us a sufficient number of samples to see a pattern. Figure 9 shows how the latency varies for the frames with different traffic loads and bridging strategies. Most of the frames are gathered in the low end of the scale, but the number of frames with a significantly higher latency is not negligible. The arrow in figure 9 points out the 90th percentile (also called the 0.9 quantile) for the data set. The 90th percentile denotes the value on the x axis so that 90% of the measures are smaller than this value. This way we can get a picture of the latency for the bulk of the traffic. As the collected data shows, the latency can never drop below 15 μs. This is the aggregate propagation delay from station 4 to 7. Frames that are not held back in any buffers along the way will achieve this minimum latency.

The six plots in figure 9 show that the latency and jitter characteristics of the traffic depends on the traffic load in the network, and the choice of bridging strategy. A higher traffic load results in more periods with congestion and thus higher latency for some of the traffic. In figure 8, the 90th percentile for the latency is plotted for different network loads, with both basic and enhanced bridging. Each point on the graphs represents the median value for the 90th percentile found by repeating the simulation 20 times with a different seed in the random generator. The vertical bars show a 95% confidence interval.

The simulations here described, show that enhanced bridging allows more traffic to be sent over the ring before the increase in latency and jitter becomes significant. With this configuration, enhanced bridging allows roughly twice as much traffic to be sent through the network before the 90th percentile latency becomes the same as for basic bridging. This way, enhanced bridging helps improving the quality of service properties for local best effort traffic.

5 Conclusions and further work

The basic bridging algorithm used in the existing RPR draft standard, prevents spatial reuse for remote traffic. This leads to poor bandwidth utilization, and gives bad scaling properties. The simulation results presented in this paper show that enhanced bridging allows spatial reuse also for remote traffic, after an initial learning period. By letting packets be sent shortest path from source to destination without flooding, enhanced bridging reduces the load imposed on the network by a single TCP-like traffic session by over 50%.

The reduction in traffic load achieved by enhanced bridging has a positive effect on the latency characteristics of local traffic on the ring. Simulations with self-similar traffic, show that substantially more remote traffic can be sent through the network before the basic bridging latency levels are reached.

The results presented in this work show that the enhanced bridging algorithm clearly gives better performance than the basic algorithm in the RPR draft standard. We believe that the inefficiency of the current bridging algorithm effectively rules out bridging as an interconnection method for RPR networks. It is the clear view of the authors that an algorithm more in the direction of enhanced bridging should be adapted in a later revision of the RPR standard.

In the follow-up to this work, we would like to investigate the effect of using a caching strategy in the SRCS tables in large bridged networks, combined with techniques for shortening the initial learning period with enhanced bridging. In a larger bridged network, the SRCS tables can be used in a cache-like manner, to reduce the amount of memory needed to keep the SCRS tables. With such a scheme, the least recently used entry would be discarded to make room for a new record when packets are received from an unknown remote host.

Acknowledgements

The authors would like to thank Olav Lysne for his help and insightful comments during the work with this paper.

References

[1] F Backes. Transparent bridges for interconnection of ieee 802 lans. *Network, IEEE*, 2(1):5–9, Jan 1988.

[2] IEEE. *ISO/IEC 15802-3: 1998 [ANSI/IEEE Std. 802.1D, 1998 Edition], Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Common specifications - Part 3: Media Acess Control (MAC) bridges*, 1998. IEEE standard for transparent bridges.

[3] Radia Perlman. A protocol for distributed computation of a spanning tree in an extended lan. *Ninth Data Communications Symposium*, 1985.

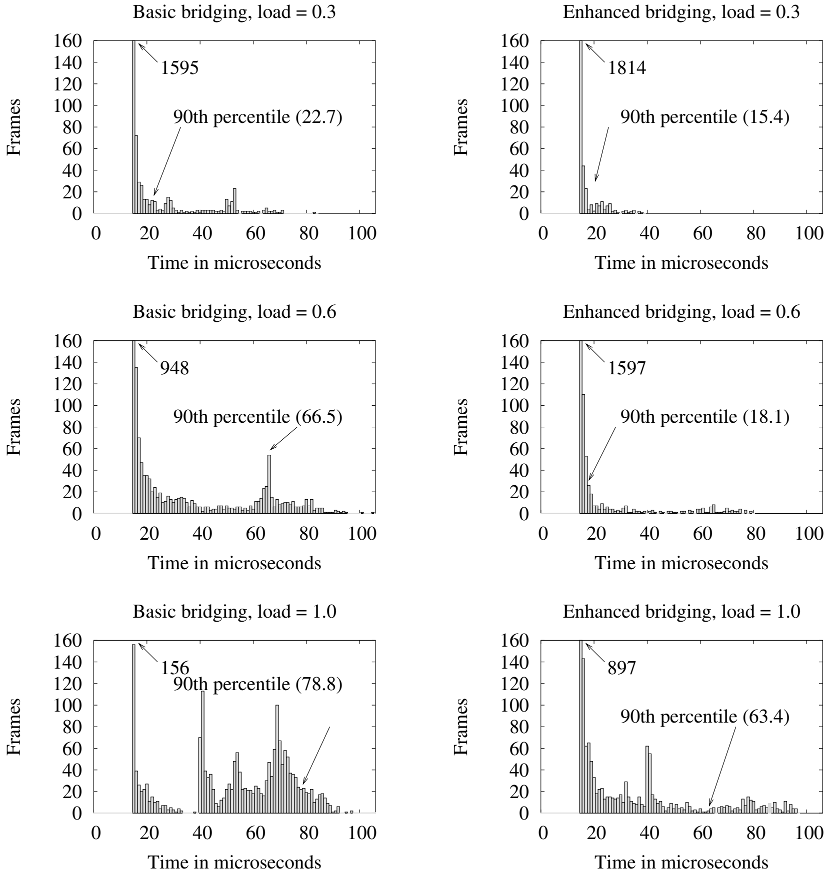


Figure 9: The fraction of frames that experience a low latency drops with increasing traffic load. Each bar in these plots represent a 100 ns time interval, and a total of 1960 frames was measured.

[4] W. Bux, F. Closs, K. Kuemmerle, H. Keller, and H. Mueller. Architecture and design of a reliable token-ring network. *IEEE Journal on Selected Areas in Communications*, 1(5):756–765, Nov 1983.

[5] M.C. Hamner and G.R. Samsen. Source routing bridge implementation [lans]. *Network, IEEE*, 2(1):33–36, Jan 1988.

[6] Marc Holness and Robert Castellano. Bridging ad-hoc overview. Presentation at RPR WG meeting in Ottawa, May 2002. Available at http://grouper.ieee.org/groups/802/17/.

[7] Robert Castellano. Enhanced bridging - spatial reuse of 802.17 bridge traffic. Presentation at RPR WG meeting in Vancouver, July 2002. Available at http://grouper.ieee.org/groups/802/17/.

[8] Marc Holness. Encapsulated bridging support of 802.17. Presentation at RPR WG meeting in Austin, TX, November 2001. Available at http://grouper.ieee.org/groups/802/17/.

[9] Robert Castellano. Frame control bit consolidation - brave proposal. Presentation at RPR WG meeting in Atlanta, January 2003. Available at http://grouper.ieee.org/groups/802/17/.

[10] RPR bridging ad-hoc committee. Bah 802.17 frame structure requirements. Presentation at RPR WG meeting in Vancouver, July 2002. Available at http://grouper.ieee.org/groups/802/17/.

[11] Marc Holness. Rpr frame transmission proposal. Presentation at RPR WG meeting in Hawaii, November 2002. Available at http://grouper.ieee.org/groups/802/17/.

[12] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of ethernet traffic. In Deepinder P. Sidhu, editor, *ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[13] Vern Paxson and Sally Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[14] Walter Willinger, Murad Taqqu, Robert Sherman, and Daniel Wilson. Self-similarity through high-variability: Statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, February 1997.