Reducing the Reservation Establishment Time in IP Tunnels by Using Staged Refresh Timers *

Paulo Cesar S. Vidal
Departamento de Engenharia de Sistemas
Instituto Militar de Engenharia
Praça General Tibúrcio, 80
CEP 22290-270 - Rio de Janeiro, RJ, Brasil
vidal@de9.ime.eb.br

Otto Carlos M. B. Duarte
Grupo de Teleinformática e Automação
Universidade Federal do Rio de Janeiro
COPPE/EE - Programa de Engenharia Elétrica
CEP 21945-970 - Rio de Janeiro, RJ, Brasil
otto@gta.ufrj.br

Abstract

This paper presents a state aggregation mechanism of the RSVP protocol for multicast flows using IP-in-IP tunneling. The mechanism relies on periodic refresh messages between routers to maintain the reservation states (soft-state approach). Because of the loss of tunnel RSVP messages, the mechanism presents a very long latency at reservation establishment. The problem reduces the performance of multimedia applications. We report the use of staged refresh timers in the aggregation mechanism at the set-up procedure of the resource reservation. The simulation results show a significant decrease of the establishment time of multicast sessions.

1 Introduction

The Internet Engineering Task Force (IETF) defined the Integrated Services Architecture [1] for providing service classes with Quality of Service (QoS) guarantees related to bandwidth, packet queuing delay, and packet loss for real-time multimedia applications. The QoS level provided by these classes is programmable per flow, in accordance with applications requirements. These requirements are passed to routers using Resource Reservation Protocol (RSVP) [2]. Although RSVP protocol was originally designed for resource reservation, some works use RSVP protocol to carry other types of control information in the network, as security QoS [3], and label-switched paths in MPLS (Multi-Protocol Label Switching) [4]. Another example is the use of RSVP protocol within the Differentiated Services [5].

* This work was supported by IME, UFRJ, FUJB, PROTEM-CC, CNPq and CAPES.

A flow is identified by its unicast or multicast address and its state is stored in the routers between the source and the destination. A greater usage of integrated services means a greater amount of stored states in the routers and a large processing is needed to manage them, thus overloading the routers. Hence, the flow aggregation mechanisms has been proposed to reduce the amount of stored states [6, 7, 8, 9].

The work of Terzis et al. [6] implements the RSVP protocol over IP-in-IP tunnels where several end-to-end unicast sessions are mapped into a tunnel session between two tunnel end points. The state reduction is achieved because the interior routers do not process end-to-end RSVP messages. Interior routers are placed between two tunnel end points, and handle only tunnel messages that carry aggregated data of the end-to-end flows. In this way, the interior routers store only the state of the tunnel sessions.

Jens et al. [7] present a mathematical analysis of the use of aggregation for identical flows of guaranteed service, that is, the flows must have the same traffic characteristic and follow the same route up to their destination. The analysis shows that an amount of bandwidth needed to reserve a flow group is smaller than the addition of individual reservations.

Davie et al. [8] describes a mechanism that uses a single RSVP reservation to aggregate other RSVP reservations across a transit routing region, in a manner conceptually similar to the use of virtual paths in a ATM network. It proposes a way to dynamically create the aggregate reservation, classify the traffic for which the reservation applies and to determine how much bandwidth is needed to achieve the requirement.

Fisher et al. [9] presents a design of the Beagle signalling protocol to support of the temporal resource sharing. The protocol provides the sharing for the same set of resources among related flows over time leading to significant resource gains and a decrease in the number of reservation states.

This article presents a state aggregation mechanism of RSVP protocol that identifies multicast flows in transit in a determined aggregation region, and groups its states among boundary routers, using IP-in-IP tunneling. The aggregation region can be an Internet Services Provider (ISP), or a Virtual Private Network (VPN). The mechanism is an extension of the model of Terzis et al. [6], with additional functions to aggregate multicast flows.

The multicast aggregation mechanism relies on periodic refresh messages between routers to maintain the reservation states (soft-state approach). The mechanism can present a very long latency at reservation establishment when tunnel RSVP messages are lost during transmission. For example, when a user tries to make a reservation over the aggregation region, if the first tunnel RESV message from the exit router is lost due to network link noise or congestion, the message will not be retransmitted until the next refresh cycle arrives. The default refresh interval is 30s.

In Internet Multicast Backbone (MBone), the average packet loss rate can vary between 1.4% and 11% [10]. Packet loss is a factor determining the quality seen by delay-sensitive multimedia applications such as video conferencing and Internet telephony. In this case, the existing delivery method of tunnel RSVP messages may not be acceptable for these applications.

Therefore, we report the use of staged refresh timers [11] to decrease the latency. The staged refresh timers control the retransmission interval between tunnel messages at reservation establishment. The staged refresh timer starts at a small interval which increase exponentially until it reaches a threshold. The reduction in retransmission interval of RSVP messages decrease the reservation establishment time. The simulation results show the performance gain in reservation set-up delay of the aggregation mechanism when the staged refresh timers is used.

The article is organized in the following way: Section 2 describes the model of RSVP tunnels for unicast flows. Section 3 describes the extended mechanism to aggregate multicast flows. Section 4 presents the results of aggregation mechanism simulation and Section 5 shows final considerations and future works.

2 RSVP Tunnels

The technique of IP-in-IP tunneling [12] can be used to create an flow aggregation region. This way, the flows of multiple sources that reaches the entry router of the aggregation region follow to the exit router of the aggregation region inside of the IP-in-IP tunnel in an aggregated way. Therefore, all interior routers in the aggregation region process an aggregated flow, instead of processing individual flows.

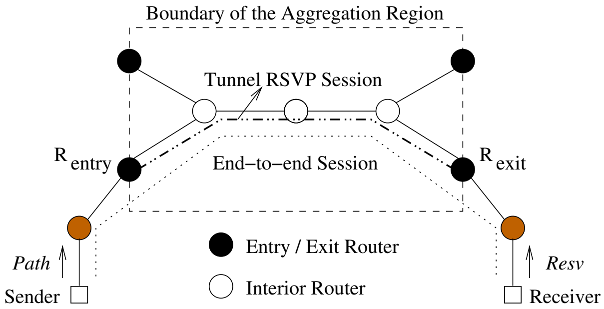

Figure 1. Unicast RSVP Tunnel.

Based on this technique, the work of Terzis et al. [6] implements an aggregation mechanism of RSVP protocol states over tunnels for unicast flows.

The mechanism of RSVP tunnels is shown in Figure 1, where the sender and the receiver of an end-to-end RSVP session are connected by the tunnel. The R_{entry} router is a tunnel entry router which encapsulates and sends the datagrams to the R_{exit} router through the interior routers. The R_{exit} router is the end point of the tunnel that receives, decapsulates and sends the datagrams based upon the original inner IP header. In this mechanism there are two different sessions: a tunnel RSVP session and an end-to-end RSVP session. The end-to-end RSVP session is a individual session with resource reservation between sender and receiver. The tunnel RSVP session placed between the R_{entry} and the R_{exit} points provides resource reservations for a group of individual RSVP sessions set up in the tunnel.

The *Path* and *Resv* messages of an end-to-end RSVP session that reach the tunnel ends are mapped into the tunnel session associated with an aggregated traffic class. These messages are represented in lower case. They are encapsulated and decapsulated in the same way as IP packets. The outer header specifying the tunnel entry point as source and the tunnel exit point as destination is added. Figure 1 illustrates the mapping of an end-to-end RSVP session, but there can be other end-to-end sessions associated with the tunnel RSVP session.

There is two possibilities to associate end-to-end sessions with corresponding tunnel sessions: to map all end-to-end sessions of same traffic class to a single tunnel session, or to allow multiple end-to-end sessions for each traffic class to be associated with a tunnel session. The association between end-to-end sessions and a tunnel session is made through the object SESSION_ASSOC inserted in the end-to-end *Path* message in the R_{entry} router. This object associates an end-to-end session with a tunnel session. The SESSION_ASSOC object has information about the end-to-end session (destination address, transport protocol and

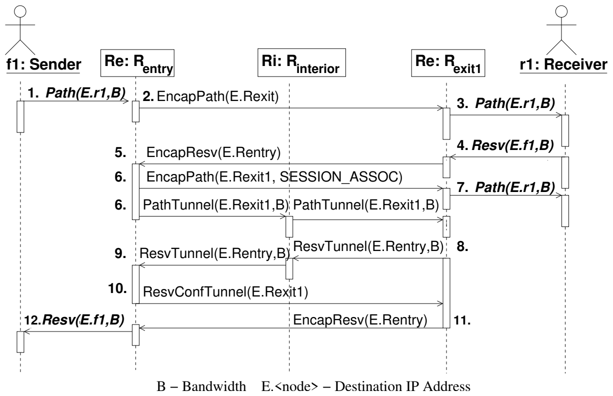

Figure 2. Message Sequence Diagram.

destination port number) and information about the tunnel session (address and port of the tunnel entry router). If there is only the simplest mapping of all the end-to-end sessions of the same traffic class to a single tunnel session, we eliminate the need for the SESSION_ASSOC object.

The interior routers do not recognize end-to-end RSVP messages, they handle only the RSVP messages generated by the end points of the tunnel. Therefore, there is a reduction in the number of states in the interior routers.

The sequence of the process of reservation set up in the tunnel is represented in Message Sequence Diagram [13] in Figure 2 and it is detailed here:

1. The R_{entry} router receives an individual *Path* message and creates the path state of the end-to-end RSVP session.

2. The R_{entry} router sends the encapsulated end-to-end *Path* message over the tunnel to the R_{exit} router.

3. The R_{exit} router receives the message, decapsulates it, creates the path state of the end-to-end RSVP session and forwards the original message to the destination.

4. When an end-to-end *Resv* message associated to an end-to-end RSVP session reaches the R_{exit} router, the router creates the reservation state, encapsulates and sends the message to the R_{entry} router.

5. The entry router receives the encapsulated *Resv* message, decapsulates it and creates the reservation state mapping the end-to-end RSVP session to a tunnel RSVP session.

6. The R_{entry} router sends the end-to-end *Path* message encapsulated with the SESSION_ASSOC object, associating the end-to-end RSVP session with the tunnel RSVP session to exit router. The entry router also sends the tunnel PATH message associated with the tunnel RSVP session to interior routers. The tunnel RSVP messages are represented in capital letter.

7. The R_{exit} router receives the *Path* message with the SESSION_ASSOC object, registers the association, removes the object and forwards the message to the destination.

8. The exit router receives the tunnel PATH message and sends a tunnel RESV message with the RESV_ CONFIRM object to the interior of the aggregation region, requesting resource reservation. The RESV_ CONFIRM object carries the exit router IP address that requested a confirmation.

9. If some interior router does not accept the reservation request, then it sends a tunnel error message to the exit router. Thus, when the tunnel RESV message reaches the R_{entry} router, it means that the request was accepted by the interior routers.

10. After receiving the tunnel RESV message, the entry router sends a tunnel RESV_CONFIRM message to the R_{exit} router confirming the reservation in the tunnel.

11. The R_{exit} router receives the message and sends the encapsulated end-to-end *Resv* message to the R_{entry}.

12. The R_{entry} router receives the message, decapsulates it and sends the original end-to-end *Resv* message to the sender.

After the reservation set up, the exit router receives periodically *Resv* messages. However, an end-to-end *Resv* message can cause a change in individual reservation state parameters and in the associated tunnel session. If the change requires to increase the reservation, then R_{exit} router sends a tunnel RESV message including the RESV_CONFIRM object. If a RESV_CONFIRM message is received, the original *Resv* message is encapsulated and sent through the tunnel. If the tunnel reservation update fails, the R_{exit} router must send an end-to-end *Resv_Error* message to the originator of the end-to-end Resv message. If the change decreases the reservation, then R_{exit} router sends a tunnel RESV message without the RESV_CONFIRM object to update the reservation state in interior and entry routers.

3 Multicast Aggregation Mechanism

The multicast flow aggregation mechanism provides bandwidth guarantees and aggregates the traffic in transit in the aggregation region by using the tunneling technique.

The mechanism is an extension of the model proposed by Terzis et al. [6] and the changes introduced are:

- Diffusion of *Path* messages to several exit routers in the mapping of the end-to-end RSVP multicast session in tunnel RSVP sessions.

- Marking the reserved data packets through the *Type of Service* (TOS) field or *Differentiated Services* (DS) field.

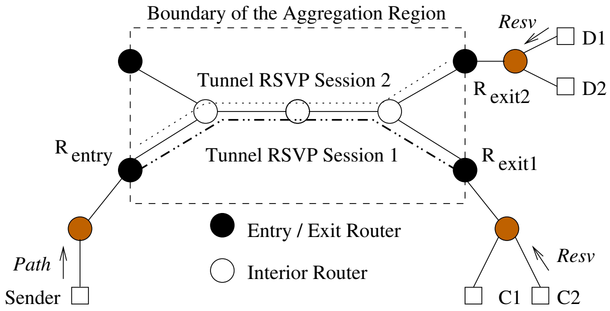

Figure 3. Mechanism of multicast RSVP tunnels.

- Use of the destination multicast address in the IP reserved data packet outer header to decrease the bandwidth in aggregation region.
- Use of the staged refresh timers in the tunnel RSVP message retransmission.

The association of the multicast RSVP session with several tunnel RSVP sessions is illustrated in Figure 3. The multicast tree is composed by a source and four receivers (C1 and C2 in C domain, and D1 and D2 in D domain), all of them belonging to a multicast session. In the example, two tunnel RSVP sessions were created. The first session is placed between the R_{entry} router and the R_{exit1} router and the second session is placed between the R_{entry} router and R_{exit2} router. Each tunnel session provides resource reservation to the end-to-end multicast RSVP session.

When the R_{entry} router receives a *Path* message, it creates the path state of the end-to-end RSVP session. It then verifies through which exit router or exit routers the encapsulated *Path* message will be sent by using the Border Gateway Protocol (BGP) [14]. After choosing the exit router(s), the R_{entry} router sends an encapsulated end-to-end *Path* message to the R_{exit1} router and a copy of the encapsulated end-to-end *Path* message to R_{exit2} router.

The process of reservation set-up continues for each tunnel RSVP session through of steps 3-12 describe in Section 2.

Reserved data packets are encapsulated in IP and in (User Datagram Protocol) UDP protocol in entry router [6]. The use of the UDP encapsulation increases the processing time in routers. The packet marking executed directly in network layer is faster than in transport layer. Therefore, we propose marking the data packets of the reserved multicast session at the entry router with the traffic class of the TOS field or of the DS field of the IP packet outer header. The DS field is obtained by renaming the TOS field, in the case of the IPv4 protocol, or *Traffic Class* field in the case of the IPv6 protocol [15]. The utilization of the DS field is to maintain compatibility with QoS requests that use Differentiated Services. After the marking, the entry router generates copies (one for each tunnel), and every packet is encapsulated and sent to the exit router address. Each exit router receives the packets, decapsulates and sends them to their destination.

As the number of exit routers to where the reserved packet must be sent increases, it also increases the number of packet copies in the region. Hence, the bandwidth consumed by the network is increased. A solution to this problem is to use the multicast IP address in the destination address field of the outer header of the reserved packet. The reserved multicast address range would be used in aggregation region (e.g., a VPN or a ISP) and each reserved multicast address would represent a group of exit routers. Each router in the region must have a table (multicast address, list of exit routers). The state of each tunnel session would be maintained. The entry router would send a reserved packet and its duplicating would happen only in the router that has two interfaces (Figure 3).

The aggregation mechanism uses the staged refresh timers in the tunnel RSVP message retransmission to decrease the set-up procedure in occurrence of message losses.

The staged refresh timers is a feedback mechanism to support reliable delivery of the RSVP messages and also to reduce the time needed in the resource reservation establishment of flows [11]. After sending the initial *Path* message, an RSVP node will wait for a lapse of time called initial retransmission interval (default value is 3s). If by that time a *PathAck* message has not been received, the node retransmits the *Path* message. The retransmission interval is controled by a staged refresh timer and it increases exponentially until it reaches a threshold (refresh interval R). The value is multiplied by a factor $(1 + \Delta)$ at each retransmission of a *Path* message. The default value used for Δ is 0.3. The retransmission will continue until a *PathAck* message arrives. Therefore, the period of the flow establishment is reduced to the initial retransmission interval plus the round-trip-time (RTT) of messages when an only *Path* message is lost. The process of *Resv* message retransmission occurs of the same way that in *Path* messages. Staged refresh timers also provides confiability through of acknowledgment messages *PathAck* and *ResvAck*.

The retransmission scheme used in aggregation mechanism is equivalent to the staged refresh timers. However, the aggregation mechanism do not use the additional messages for acknowledgments. The role of an initial tunnel PATH message is to prepare for a subsequent tunnel RESV message. A tunnel RESV message can be considered as an acknowledgment for a tunnel PATH message. A tunnel RESV_CONFIRM message represents an acknowledgment for a tunnel RESV message. The staged refresh timer controls the retransmission interval between tunnel PATH messages and tunnel RESV messages. When the exit router sends the tunnel RESV message first, it includes the RESV_CONFIRM object in message and it includes the object in a waiting list. The RESV_CONFIRM object is composed of the entry router address and of the object number. There is a staged refresh timer associated with every object and the timer controls the retransmission of messages. If the exit router receives the RESV_CONFIRM corresponding, it deletes the object in list, else it retransmites the tunnel RESV message.

4 Simulation Results

This section presents the simulation results of the process of reservation set-up executed by multicast aggregation mechanism when there are tunnel RSVP message losses.

The goal of simulations is to compare the performance between the utilization of staged refresh timers and fixed refresh timers in the aggregation mechanism in presence of tunnel RSVP message losses. The fixed refresh timer refers to classical soft-state mechanism of the RSVP protocol which the retransmission interval is a fixed value (30 seconds) during the flow establishment.

We use the *ns* (Network Simulator) [16] to obtain the measurements of the establishment time of multicast sessions. An extension to the *ns* basic packet was implemented to support the aggregation mechanism being formed by two modules: the classic RSVP and the aggregated RSVP. The classic RSVP module was obtained from [17]. The two modules were implemented using the C++ and the Otcl languages.

The aggregated RSVP module developed executes the following functions:

- Processing of tunnel PATH, tunnel RESV and tunnel RESV_CONFIRM messages.
- Mapping of end-to-end sessions to tunnel sessions.
- Packets encapsulating and decapsulating.
- Aggregated states tables maintenance.
- Retransmission of messages within staged refresh timers.

The simulations consisted of repeated reservation establishments for multicast groups (a group is composed of a source and n receivers) in the aggregation region, over multicast trees of various lengths and under different loss conditions.

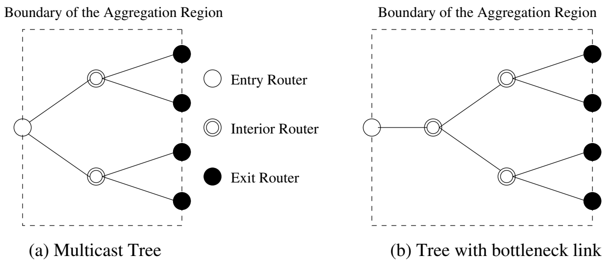

Figure 4. Simulation Topologies.

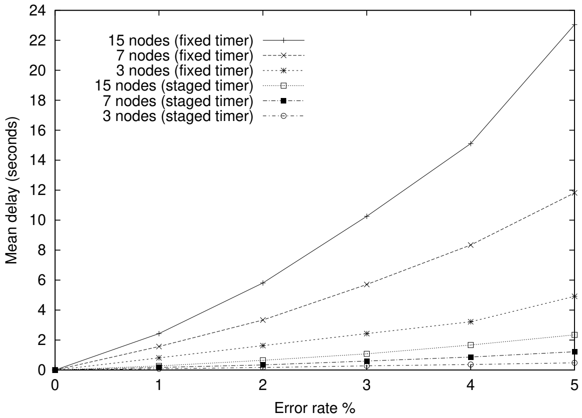

Figure 5. Mean establishment delay in full multicast tree within 1-5% of packet loss.

Figure 4 presents two types of topologies used in the simulations. A full binary multicast tree and binary multicast tree with bottleneck link. The root node is the entry router and the leaf nodes are the exit routers. The root node receives the end-to-end RSVP messages of the source and the leaf nodes receive the end-to-end RSVP messages of the receivers. The configurations of the full binary multicast tree comprise 3, 7 and 15 nodes. The configurations of the binary multicast tree with bottleneck link comprise 4, 8 and 16 nodes.

For every configuration, 3000 flows (multicast groups) were established and no delay was introduced in links to isolate the time overhead introduced only by the operation of the mechanism. The mean duration of each multicast session and of the transmission between tunnel PATH messages are 180s and 100ms respectively with exponential distributions. A tunnel RESV message is sent immediately after the arrive of the tunnel PATH message.

We consider two types of the loss model. The first is the average packet loss. This model shows the impact of losses

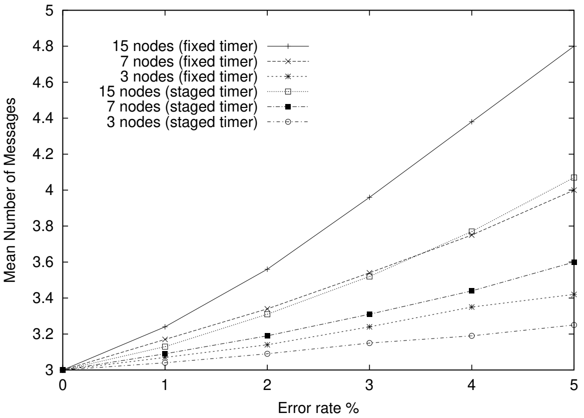

Figure 6. Mean number of control messages in full multicast tree within 1-5% of packet loss.

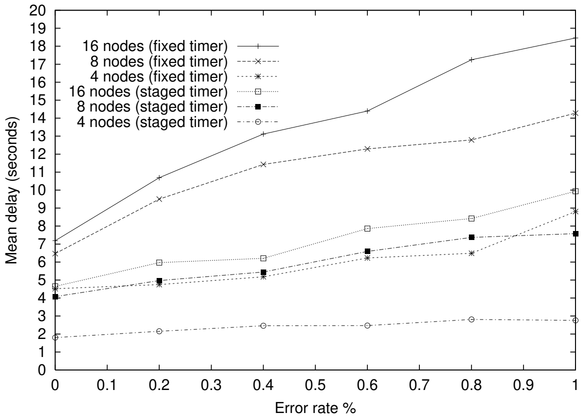

Figure 7. Mean establishment delay in multicast tree with losses ocurring each 30 seconds.

in long-term QoS provided by RSVP Tunnel in full binary multicast tree (Figure 4.(a)). The loss process on each direction of a link is represented by a packet loss rate with uniform distribution. The links used in simulations have an error rate (1-5%). These values can represent a variation of the loss rate during a hour in high speed networks [18] or a day in MBone [19].

Figure 5 shows the measured mean establishment delay and clearly demonstrate the gain in performance obtained with staged refresh timers. The error rate represents the packet loss in all links for each configuration of the multicast tree. Note the low establishment delay achieved by staged refresh timers over multicast trees with error rate between 3% and 5%.

The other measured quantity were the mean number of tunnel RSVP messages per reservation (multicast session). There is however a trend showing a better effectiveness of staged timers as error rate increase (Figure 6).

In second loss model, we simulate the mechanism in the multicast binary tree with periodic packet losses in bottleneck link (Figure 4.(b)). Periodic losses occur in congestion periods in routers each 30s or 60s, when a serie of consecutive packets were lost. This is due to the routing updates as report in [18, 20]. The loss process on each direction of a bottleneck link is represented by a two-state model. One of the states represents congestion (i.e. loss) periods while the other one represents no-loss periods. The loss process spends an exponetially distributed time in loss state with these exponential distributions set so that the mean congestion period is 250ms. The losses occur every 30s or 60s. The loss process of other links is represented by a packet loss rate with uniform distribution with a low error rate (0.1-1%).

The Figures 7 and 8 show the measured mean establishment delay for periodic losses. The error rate represents the packet loss in links with a low error rate (0.1-1%) for each configuration of the multicast tree. The zero value in x axis represents packet losses only in bottleneck link. Once more, the results show the gain in performance with staged refresh timers, even when only facing a single point of congestion. However, the gain in losses ocurring each 30s is bigger than in losses ocurring each 60s. The multicast tree of 4 nodes show a low delay variation because the two links connected in leaf nodes influence little in total mean delay of the topology.

5 Final Considerations

The flow aggregation mechanisms provide a decrease in the number of states necessary to store about the resources reservation of RSVP protocol, thus increasing the scalability of its use in the Internet.

The multicast aggregation mechanism presented in this paper reduces the number of stored states and the number of messages processed in the aggregation region because the interior routers handle only tunnel RSVP messages generated in tunnel end points.

The simulations evaluate the mechanism performance in process of reservation set-up when losses of the tunnel RSVP messages occur. The results shown the gain in mean establishment delay obtained with the use of staged refresh timers over fixed refresh timers. The results also revealed

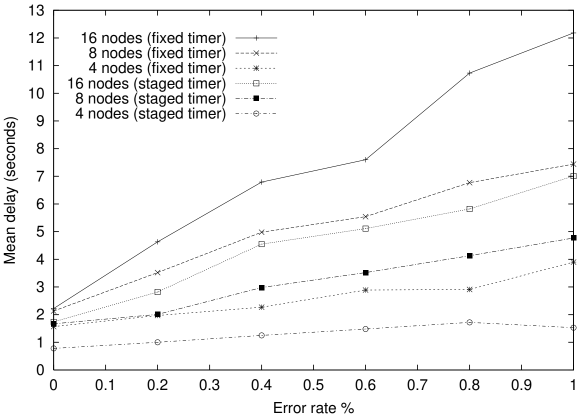

Figure 8. Mean establishment delay in multicast tree with losses ocurring each 60 seconds.

a low establishment delay in full multicast trees with error rate between 3% and 5%. In multicast tree with bottleneck link, the gain in losses ocurring each 30s is bigger than in losses ocurring each 60s. This result is important because it shows that the gain increases when the congestion augment.

For future works, we can emphasize, the execution of simulations in larger topologies to evaluate the processing overload associated with the packet encapsulating at tunnel end points.

References

[1] R. Braden, D. Clark, and S. Shenker, "Integrated Services in the Internet Architecture: An Overview", *RFC 1633*, July 1994.

[2] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: a new Resource ReserVation Protocol", *IEEE Network Magazine*, September 1993.

[3] U. Roedig, C. M. Gortz, M. Karsten, and R. Steinmetz, "RSVP as Firewall Signalling Protocol", *In Proceedings of the 6th Symposium on Computers and Communications*, July 2001.

[4] D. Awduche, L. Berger, T. Li, V. Srinivasan, and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", *RFC 3209*, July 2001.

[5] A. Ponnappan, L. Yang, R.Pillai, and P. Braun, "A Policy Based QoS Management Systems for the IntServ/DiffServ Based Internet", *In Proceedings of IEEE 3rd International Workshop on Policies for Distributed Systems and Networks*, June 2002.

[6] A. Terzis, L. Zhang, and E. L. Hahne, "Reservations for Aggregate Traffic: Experiences from an RSVP Tunnels Implementations", *In Proceedings of 6th International Workshop on Quality of Service*, May 1998.

[7] J. Schmitt, M. Karsten, L. Wolf, and R. Steinmetz, "Aggregation of Guaranteed Service Flows", *In Proceedings of 7th International Workshop on Quality of Service*, June 1999.

[8] F. Baker, C. Iturralde, F. Le Faucheur, and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", *RFC 3175*, September 2001.

[9] P. R. Chandra, P.Steenkiste, and A. Fisher, "Extensible Signaling for Temporal Resource Sharing", *IEEE Journal Selected Area in Communications*, vol. 19, no. 3, March 2001.

[10] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and Modeling of the Temporal Dependence in Packet Loss", *In Proceedings of IEEE INFOCOMM*, March 1999.

[11] P. Pan and H. Schulzrinne, "Staged Refresh Timers for RSVP", *In Proceedings of Global Internet Conference*, November 1997.

[12] C. Perkins, "IP Encapsulation within IP", *RFC 2003*, October 1996.

[13] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language - User Guide*, Addison-Wesley, 1999.

[14] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, "Multiprotocol Extensions for BGP-4", *Internet-Draft*, January 1998.

[15] K. Nichols, S. Blake, F. Baker, and D. L. Black, "Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers", *RFC 2474*, December 1998.

[16] NS, "Network Simulator - NS (version2)", *http://www-mash.cs.berkeley.edu/ns/*, 2000.

[17] M. Greis, "RSVP/ns: An Implementation of RSVP for the Network Simulator ns-2", *http://titan.cs.uni-bonn.de/ greis/rsvpns/rsvpns.ps.gz*, 1999.

[18] Y. Zhang, V. Paxson, and S. Shenker, "The Stationary of Internet Path Properties: Routing, Loss, and Throughput", *ACIRI Technical Report*, May 2000.

[19] M. Handley, "An Examination of MBone Performance", *UCS/ISI Technical Report ISI/RR-97-450*, January 1997.

[20] S. Floyd and V. Jacobson, "The Synchronization of Periodic Routing Messages", *IEEE/ACM Transactions on Networking*, vol. 2, no. 2, pp. 122–136, April 1994.