Study of IEEE 802.1p GARP/GMRP Timer Values*

Fouad A. Tobagi
Pablo Molinero-Fernández
Mansour J. Karam

Computer Systems Laboratory
Department of Electrical Engineering
Stanford University
Stanford, CA 94305

tobagi@bodega.stanford.edu, molinero@stanford.edu, mans@leland.stanford.edu

Abstract

As new services requiring multicast communications become more common, the deployment of selective multicasting in bridged LANs becomes essential. The GARP/GMRP protocol, defined in the IEEE 802.1p standard, addresses this issue. The performance of this protocol depends heavily on the values of the various timers that control it. In this paper we study the performance of GARP/GMRP in terms of response time, traffic overhead created by the protocol's control messages, and other artifacts which arise from the particular operation of the protocol. We analyze the sensitivity of the various performance measures to various timers used in the protocol, and find the appropriate values to use for these timers. This performance assessment is done by means of computer simulation focusing on 10Base-T and 100Base-T network configurations. Usage scenarios are designed appropriately to bring out the specific effects of interest.

I. Introduction

Bridges are used to interconnect several LAN segments so as to form larger LANs (also referred to as extended LANs), maintaining all the features of a single LAN segment for both unicast and multicast traffic. The bridges organize themselves into a single spanning tree along which all traffic among stations flow. Bridges learn about stations' locations transparently by examining the source address in all frames, and forward unicast traffic to their destinations along the spanning tree. If the destination has not been learned, the unicast frame is broadcast

---

*. This work has been supported by 3com corporation. Pablo Molinero Fernández has been supported by a fellowship from the Fundación Pedro Barrié de la Maza, Conde de Fenosa.

throughout the extended LAN. Multicast traffic is treated as broadcast traffic; that is, it is flooded along the spanning tree so as to reach all stations. Stations receive frames with multicast addresses they are interested in and filter out all other multicast frames. The operation of bridges is described in the IEEE 802.1D standard [802.1D].

The handling of multicast traffic as broadcast traffic is acceptable as long as the volume of such traffic is low, which has been the case with traditional data applications; conversely, this way of handling multicast traffic places a limit on the total volume of such traffic that must be low enough to be accommodated by the slowest link in the topology, and still leave sufficient bandwidth to carry other traffic. Take for example a switching hub with 124 10 Mbps ports and an internal back plane bus capacity of 1.24 Gbps. Assume that the multicast traffic consists of 1.5 Mbps video channels. If one were to broadcast all multicast traffic, only 6 video channels would be sufficient to use up the 10 Mbps bandwidth on all segments leaving no bandwidth for other applications, while the backplane bus capacity can handle over 800 such channels, and generally speaking, not all users would be interested in exactly the same channels. If one were to selectively forward on each port only the multicast traffic that is desired by the users attached to that port, then it would be possible to handle a much larger volume of multicast traffic and give the user the flexibility to choose dynamically which multicast traffic to receive. This requires a protocol which allows the users to express dynamically their interest in receiving specific multicast traffic. Such a protocol has been designed and is currently being standardized by the IEEE, and its description can be found in IEEE 802.1p [802.1p], a supplement to IEEE 802.1D. The name given to this protocol is GARP/GMRP. As multicast based applications (such as video conferencing, video multicasting, virtual LANs) are expected to become more widely used, and to involve higher data rates than has been the case with traditional data applications, this protocol is expected to play a key role in achieving good performance in extended LANs. This extends to the data link layer functionality which has already been introduced at the network layer in such protocols as IGMP and the various IP multicast routing protocols [JSAC97a].

In this paper we study the performance of GARP/GMRP in terms of response time, traffic overhead created by the protocol’s control messages, and other artifacts which arise from the particular operation of the protocol. We analyze the sensitivity of the various performance measures to various timers used in the protocol, and find the appropriate values to use for these timers. This performance assessment is done by means of computer simulation focusing on 10Base-T and 100Base-T network configurations [802.3]. Usage scenarios are designed appropriately to bring out the specific effects of interest.

The organization of this paper is as follows. In section II, we give a description of the protocol. In section III, we describe the scenarios simulated and the numerical results obtained. In section IV, we conclude with a few remarks.

II. Description of GARP/GMRP

GARP/GMRP is a scheme defined in IEEE802.1p which allows stations to dynamically register/deregister for a multicast group. The design objectives behind GARP/GMRP include simplicity, robustness and scalability, consistently with the design objectives of transparent bridges; furthermore, its design takes advantage of the broadcast nature of LANs whenever possible, and guarantees the propagation of information about the whereabouts of destinations throughout the entire network along the spanning tree.

The basic goal of the scheme is to determine at any point in time if a port in a bridge is to forward the traffic for a particular multicast group out on that port; this would be the case if some stations reachable by that port are interested in receiving that traffic; by following this rule for all ports in all bridges, the interested stations would be in a position to receive the appropriate multicast traffic, regardless of where the sources for these multicast groups happen to be. To achieve this goal, the scheme makes use of two control messages; namely:

1.the JOIN message which is sent by a station to express its interest in receiving traffic destined to a group;
2. the LEAVE message which is sent by a station to express its interest in leaving a group.

In a multihop network, these messages are also used by bridges to propagate information about the stations' interest throughout the entire extended LAN.

For the sake of simplicity and scalability, no explicit information about the stations' identifications is kept in the bridges, nor is there any explicit confirmation that information about stations' interests has been received by the bridges or propagated throughout the network. Accordingly, some redundancy is built into the protocol to overcome the possibility of packet loss in the network. Particularly, resiliency to the loss of a single control packet is incorporated into the scheme by ensuring that stations (and bridges) send collectively two such messages before considering that the intent has been registered. Note that the two copies of the same message need not be sent by the same device. That is, if several devices (stations or bridges) are attached to the same LAN segment, and wish to issue a given control message (e.g., a JOIN for a particular multicast group), whenever a device hears another device issue the control message, this message counts as one of the two copies that the device is to transmit.

Since no station identification nor station counts are kept at the bridges, whenever a bridge receives a LEAVE message on a port (sent by a device leaving that group), the bridge is bound to cease forwarding packets for that group on that port; however it does so after waiting for a certain time-out period, called the LeaveTime. This time-out period is there to allow other devices which are still interested in receiving that multicast traffic to reaffirm their interest by sending JOIN messages as if they were joining the group for the first time.

To limit the number of control messages generated by a device, a control message may have multiple arguments associated with it (e.g., a JOIN message for a multiplicity of multicast groups). Furthermore, each control message need not be transmitted as an independent protocol data unit (PDU). Instead, it is possible to group multiple control messages into a single GARP/GMRP data unit. To limit the number of PDUs needing to be transmitted by a device, the generation of PDUs is regulated to take place at intervals of time determined by a parameter called the JoinTime (More specifically, according to IEEE 802.1p, these intervals of time are generated uniformly between 0 and JoinTime, independently of each other). This "gating" process reduces the number of PDUs by allowing a device (particularly a bridge) to eliminate duplicates of control messages and aggregate together multiple requests received on different ports to be forwarded on a given port.

This gating process also allows the reduction of similar control messages needing to be transmitted by different devices connected to the same LAN segment (e.g., the transmission of JOIN messages in response to a LEAVE message). Indeed, the commitment to send the JOIN message at any given device is not finalized until the time the gating process allows the formation of the PDU; thus, if such a message is heard on the LAN segment due to some other device having transmitted it prior to that time, such a control message is canceled. To prevent the possibility of synchronization of the gating processes at the various devices, the intervals separating the generation of PDUs need to be randomized. IEEE 802.1p specifies these intervals to be uniformly distributed between 0 and the parameter JoinTime.

A third control message is the LEAVE-ALL message which is sent by a bridge on each of its ports periodically to check that no multicast traffic is being forwarded unnecessarily, as would be the case if stations leave multicast groups without explicitly sending the LEAVE message (e.g., a user suddenly powers off its computer). The LEAVE-ALL message pertains to all multicast groups. Stations react to it similarly to the way they react to the reception of a LEAVE message for each group; the bridge will cease forwarding packets on a given registered group if no JOIN message is received for that group within the LeaveTime time-out.

There are several effects to take note of:

Consider the scenario whereby a user decides to join a multicast group. The time it takes for the JOIN message to be propagated along the spanning tree is function of the gating interval and of the delay incurred in the transmission of the PDUs throughout the network. In any given hop, the delay is the sum of the gating delay and of the network delay. The average gating delay is equal to (13/48)·JoinTime (the average residual life of a gating interval) which is approximately equal to JoinTime/4. Now if the source or the first bridge between the interested user and the source already receiving this multicast traffic is a certain number of hops away, the response time is equal to the number of hops times JoinTime/4 plus the cumulative network delay. While the network delay component is totally independent of the GARP protocol and its parameters, the gating delay component increases with the value of JoinTime; thus, there is an incentive to have the JoinTime parameter as small as possible.

Consider the scenario whereby a certain number of users are subscribed to a given multicast channel. Every time a LEAVE message for that channel is transmitted, all subscribed users get “anxious” and plan to transmit the two JOIN messages. The number of such control messages that get transmitted is function of the gating interval: the larger JoinTime is, the more spread are the potential control messages, and the more reduction occurs due to the ability for stations to learn of the other transmissions and to cancel the transmission of their own control messages. Conversely, the smaller JoinTime is, the more PDUs are committed and the higher is the network delay incurred, and consequently the lower is the reduction. Thus there is an incentive to have the JoinTime parameter be large enough to achieve a good enough reduction in the number of control messages.

Consider now the scenario whereby there are several multicast channels, with very few subscribers each (ultimately, a single subscriber per channel). It is not very likely for LEAVE messages corresponding to the various channels to be synchronized in time to be a problem. The LEAVE-ALL messages however cause all subscribers to have to respond with their two JOIN messages to avoid getting disconnected. No reduction of control messages is possible here due to the fact that we are considering a single subscriber per channel. The main issue here is the possibility of subscribers getting disconnected. It is important that the LeaveTime time-out be appropriately selected, so as to guarantee with a very high probability that a JOIN message for each subscriber reaches the bridge prior to the time-out period elapsing. The choice of LeaveTime is linked to the value of JoinTime, since the delay incurred by the JOIN messages is itself directly linked to the value of JoinTime. Here again, the delay incurred by a JOIN message from when a LEAVE-ALL is heard until it is received by the bridge is the sum of the gating delay and the network delay. The former increases with JoinTime, while the latter decreases, due to the fact that the transmission of JOIN messages will be spread over a larger period of time, and will incur fewer collisions with each others. Given a choice of JoinTime, the value of LeaveTime should be selected large enough so as to guarantee a desired threshold on the probability of disconnection. The value of LeaveTime, on the other hand, should not be arbitrarily large so as to limit the amount of excess (unnecessary) forwarding, which would result from keeping a multicast channel alive while no users are interested in it. The amount of excess forwarding is in fact the sum of both the gating delay and the LeaveTime value. Indeed, given that a subscriber has decided to leave a multicast group (and it is the only subscriber for that group), its LEAVE message is going to incur the gating delay before a PDU is generated, then the network delay before the bridge sees the LEAVE message. The bridge then will continue forwarding the traffic for a period of time equal to LeaveTime. The sum of these three components constitutes the excess forwarding.

III. Simulation Results

There are two main issues that are addressed in this work: (i) the reduction of control messages, and (ii) the possibility of disconnection. For each of these issues, an appropriate scenario that emphasizes the effects underlying the issue is constructed and simulated. The goal of these experiments is to find appropriate values to use for the scheme's timing parameters, namely, JoinTime and LeaveTime.

The most general scenario consists of a number of multicast channels, say K, and a number of devices, say N, connected to a port on an Ethernet switch. This port may be 10Base-T (with bandwidth $W = 10$ Mbps) or 100Base-T ($W = 100$ Mbps). These devices may be bridges or user stations, and are possible subscribers to the various multicast channels. It is not particularly important to consider multihop operation in our simulation study because different hops operate independently of each other. Thus we focus on single hop scenarios. The particular choice of the parameters K and N and the particular mapping of devices to multicast channels depend on various factors and are chosen differently for different issues. We consider an upper limit on N of 100 (i.e., no more than 100 devices attached to the same port of a GARP/GMRP capable bridge); this is based on the assumption that any larger value for N would begin to defeat the purpose of having selective multicasting.

In all scenarios studied, we consider the network to be also supporting background traffic, with an average load denoted by G_S ranging from 5% to 75% of the Ethernet segment's bandwidth. This background traffic is considered to be generated by a number of devices (specifically 6) according to the following simple model: the traffic is generated in the form of constant size bursts of M_S bytes each, where M_S ranges from 1,500 bytes to 64,000 bytes, (transmitted in maximum size packets of 1,500 bytes). The time separating the generation times of consecutive bursts is uniformly distributed, with the mean of the uniform distribution appropriately selected to match the background load specified by G_S. With such a model, it is possible for us to study the effect of background traffic load G_S and burstiness M_S on the selection of timing parameters.

A. Reduction of control messages

We are concerned here with the reduction of similar JOIN messages generated by different devices in response to events such as the reception of LEAVE or LEAVE-ALL messages. The appropriate scenario to consider for this issue is that of a number of users N subscribed to a multicast channel. We generate a LEAVE message every 1.5 s and examine the number of JOIN messages that get transmitted by the population of subscribers as a function of JoinTime. We examine the mean number, the standard deviation, the 90th and 99th percentiles. We plot these in Figure 1 for a baseline scenario consisting of a 10Base-T segment (W = 10 Mbps), N = 100, G_S = 7.5 Mbps, and M_S = 1,500 bytes. The results show that for small values of JoinTime, the number of control messages transmitted is extremely high, but that this number decreases rapidly with increasing values of JoinTime. This “knee” behavior to almost all measures considered indicates that an appropriate value to use for JoinTime is one close to the knee of the curve, as larger values give relatively small improvements. Based on the mean number of control messages, a good choice for JoinTime for the baseline scenario is in the neighborhood of 75 ms, in which case the mean number is about 10 and the 99th percentile is about 60. In an attempt to keep the number of control messages low in most cases, it is preferable to consider the 99th percentile curve to determine an appropriate value for JoinTime; based on this curve, a value for JoinTime of 150 ms guarantees that 99 percent of the time the number of control messages is below 40; while a value of 300 ms would guarantee this number to be as low as 20. The mean number then is about 6. The standard deviation for this case is fairly close to the mean, suggesting that to a first approximation the distribution of the number of control messages in this case may be a geometric one. As we can see, the 90th percentile curve is fairly close to that of the mean, so in most of the cases the number of control messages generated is going to be much smaller than the value obtained by the 99th percentile curve. To be on the conservative side, however, we consider in the sequel only the 99th percentile for the number of control messages transmitted.

To see the effect that N has on the selection of JoinTime, we plot in Figure 2 the 99th percentile for the number of control messages as a function of JoinTime for various values of N, namely, 10, 50, and 100, everything else remaining the same as for the baseline scenario. From these results, it is clear that the smaller N is, the smaller JoinTime can be. In fact, we suggest that for N = 50 a value for JoinTime in the range of 75 to 150 ms be used, and for N = 10, a value in the range of 50 to 75 ms be used. Another interesting effect was that the standard deviation decreased significantly with smaller values for N.

To examine the effect of background traffic load, we consider the extreme case of G_S = 0.5 Mbps, and plot in Figure 3 the 99th percentile for the various values of N. We note that a decrease in background load leads to smaller values of JoinTime. The reason for this is that the network delay has become smaller. For N = 100, we suggest a value in the range of 50 to 100 ms with a guaranteed number of messages of less than 20; for N = 50, we suggest a value in the range of 40 to 75 ms, and for N = 10, we suggest a value of 25 ms. For loads in between 0.5 and 7.5 Mbps, the appropriate choices would lie between the two cases examined above.

To study the effect of burstiness in the background traffic, we vary M_S from 1,500 to 64,000 for the baseline case as well as for smaller values of N and G_S. The case of $N = 100$ is shown in Figure 4 for $G_S = 7.5$ Mbps and M_S = 1,500, 12,000, and 64,000. The results show that the difference between the various curves is relatively small and does not lead to different choices for JoinTime. This result is expected since the successful transmission of the first few JOIN messages that help reduce the total number of JOIN messages is not dependent on the burstiness of the background traffic. The JOIN messages to be transmitted by the stations are all identical and serve the same purpose, and therefore are not prone to capture effects created by the burstiness in the background traffic.

We now address the fact that in the results above, when JoinTime is arbitrarily small and almost no reduction is possible, the total number of JOIN messages that are transmitted exceed considerably the number $2N$ that would be expected given that each station has to transmit at most 2 JOIN messages to achieve resiliency to the loss of a single packet. The explanation to this phenomenon lies in the specific finite state machine used to implement the scheme, one instance of which exists for each multicast group that is active. To simplify the presentation, the finite state machine may in fact be decomposed into two finite state machines, the first referred to as the applicant, and the other called the registrar.

The applicant, which resides in all devices, is in charge of the generation of GARP messages. (See Figure 5.) One role of the applicant is to guarantee that two JOIN messages are collectively transmitted; this part has three states: Very Anxious (VX), Anxious (AX), and Quiet (QT); when in the VX state, 2 JOIN messages are yet to be transmitted (or seen), when in the AX state, only one JOIN message is yet to be transmitted (or seen), and when in the QT state, the device is satisfied, as it believes that two such messages have been sent. The other role of the applicant has to do with when a device wants to leave a group; it also has three states, and it guarantees that two LEAVE messages are transmitted or seen, unless either a LEAVE-ALL message is seen or a JOIN in response to a first LEAVE is also seen.

The registrar on the other hand is responsible to track whether or not other devices on the same segment are subscribers to the multicast group. (See Figure 6.) The registrar exists at every port of every bridge, as it is used to control the forwarding of packets out of the port. It has three states: In (IN), Empty (MT) and Leaving (LV); the MT state indicates that the registrar is not aware of any other device in the same segment being member of the group; the IN state indicates that there are other devices that are members of the group or interested in becoming members, and is entered from the other two states upon the reception of a JOIN message; the LV state is entered from the IN state upon the receipt of a LEAVE or LEAVE-ALL message; the transition from the LV state to the MT state is upon the expiration of the LeaveTime timer under the condition that no JOIN message has been heard. Traffic for a multicast group is blocked at a given port whenever the registrar at that port for that group is in the Empty state.

In the above analysis, it is assumed that the registrar is present at all stations. One particular benefit for having the registrar in users' stations is source pruning: the source of a multicast could become quiet if it has knowledge that there is no member in the network, and, therefore, save bandwidth in the segment on which it resides. If a registrar is present at a station, then the JOIN and LEAVE messages are qualified by I or E depending on whether the registrar at that station is in the IN state or not. The implication is that the finite state machine for the applicant includes transitions among states that depend on the qualification (I or E) as shown in Figure 5; particularly, upon receipt of a JOIN-E, a station that may have transmitted already a JOIN message and thus has moved to the Anxious state is forced to move back to the Very Anxious state, and proceeds as if it had not send a JOIN message nor seen a JOIN-I message. Thus, whenever a LEAVE message is heard, the registrar's state machine transitions out of the IN state; noting that when the JoinTime value is small, JOIN messages are committed prior to the station being able to hear JOIN messages from other stations, the JOIN messages are tagged with the denomination Empty (JOIN-E) and cause other devices that have advanced to the AX or QT states to transition back to the VX state. This is the reason for having more than 2*N* messages sent in response to a LEAVE.

It is possible to run the algorithm without a registrar in users stations and limit the presence of a registrar only to switches and potential sources. In this case the majority of devices (particularly when *N* is large) would be without a registrar and would always send JOIN-I messages. This approach is expected to limit the number of control messages and to allow JoinTime to be smaller. To see the effect of removing the registrar, we plot in Figure 7 the 99th percentile for the various values of *N* with the registrar removed. Indeed, the removal of the registrar decreases the number of control messages for all values of JoinTime. From the point of view of selecting a value of JoinTime, we note that while with the registrar the suggested value for JoinTime for $N = 100$ was in the range of 150 to 300 ms to reach a number of control messages below 20, with the registrar removed, a value of JoinTime equal to 50 ms would reach the same maximum. Conversely, with a JoinTime value of 150 ms, the 99th percentile is over 35 when the registrar is present, while without the registrar, this number is as low as 15. Similar results are seen for all other values of *N*.

We now consider the case of 100Base-T. Due to the improvement in network delay in 100Base-T as compared to 10Base-T, it is expected that a smaller degree of spreading would be required to achieve the same degree of reduction. Indeed, this is the case. We plot in Figure 8 and Figure 9 the 99th percentile for the various values of *N* with background traffic loads of 75 and 5 Mbps and M_S = 1,500 bytes. For G_S = 75Mbps, a value for the JoinTime equal to 75 ms is sufficient to decrease the number of control messages to a value between 10 and 15 for $N = 50$ and $N = 100$, respectively, although if one is willing to accept a lower reduction to achieve a number of control messages similar to that attained in 10Base-T, then smaller values may be used (e.g., 40 for $N = 100$, 20 for $N = 50$). However, the knee observed in these graphs suggest that the use of the slightly higher values will allow a significant reduction and is thus suggested. For $N = 10$, a JoinTime equal to 10 ms is sufficient to reduce the number of control messages to 15. For G_S = 5 Mbps, appropriate values for JoinTime are 40, 15 to 30, and 10 ms for $N = 100$, 50, and 10, respectively, resulting in a number of control messages between 5 and 10.

We summarize the results obtained above in the following table which shows the suggested values for the JoinTime depending on the number of GARP participants and the average background load:

Table 1: Suggested values for the JoinTime parameter

	10 Base-T		100 Base-T	
N	G_s = 7.5 Mbps	G_s = 0.5 Mbps	G_s = 75 Mbps	G_s = 5 Mbps
10	50 - 75	25	10 - 40	10
50	75 - 150	40 - 75	50 - 75	15 - 30
100	150 - 300	50 - 100	75	40

B. Possibility of disconnection

We are concerned here with the situation where a JOIN message for a multicast channel transmitted by a device (or set of devices) in response to a LEAVE or LEAVE-ALL message is received by the bridge after the Leave Timer has expired, and thus an unwanted disconnection has resulted. The issue is to select an appropriate value for the parameter LeaveTime to guarantee a probability of disconnection that is sufficiently low. We recall that the delay incurred by a JOIN message has two components: the gating delay and the network delay; the latter is function of the number of control messages having to be transmitted and the likely collisions that is to be incurred (which in turn is function of the gating delay), as well as the background traffic on the network. The scenario considered to study this issue (believed to be the worst case scenario) is a number of multicast channels with a single subscriber per channel, since in this case there can be no reduction of JOIN messages among the stations, and thus there will always be 2 JOIN messages per channel. Furthermore, if there were more than one subscriber per channel, then the delay for the first JOIN message to reach the bridge would be (statistically) smaller than that incurred in the case of a single subscriber. We are concerned with the reception of LEAVE-ALL messages which triggers all devices subscribed to multicast traffic to respond with JOIN messages. We consider probabilities of disconnection P_d equal to 10^{-2}, 10^{-3} and 10^{-4}. Given that it is suggested in IEEE 802.1p that LEAVE-ALL messages be transmitted every 10 s, these probabilities translate to a disconnection per channel occurring on the average once every 16 minutes, 2.7 hours, and 27 hours, respectively.

We consider first 10Base-T with a background load of 7.5 Mbps and M_s = 1,500 bytes. We plot in Figure 10 the required LeaveTime as a function of JoinTime to achieve 10^{-2}, 10^{-3} and 10^{-4} probabilities of disconnection for N = 10, 25 and 50. In this figure it is assumed that the probability of loss of a control packet (at the bridge) is 0; this means that it is the delay for the first JOIN message from each subscriber that determines whether a disconnection has occurred or not, the second message is superfluous and contributes to network load. It is observed that for the range of JoinTime of interest (0 to 300 ms, as determined by the need for reduction of control messages in the scenario studied above), the required LeaveTime depends on both N and P_d, although the dependence on N is much weaker than that on P_d. However, the required value for LeaveTime is relatively independent of JoinTime. With a value for N equal to 50, the suggested values for LeaveTime are 400 ms for $P_d = 10^{-4}$, 300 ms for $P_d = 10^{-3}$, and 200 ms for $P_d = 10^{-2}$.

To understand the effect of packet loss, we consider an independent loss model whereby a control message is lost with a probability P_l independently of all other control packets. This would be reasonable if JoinTime is not too small, and thus different control messages are uncorrelated. Under these conditions there is a nonzero probability (specifically, $P_l^{\,2}$) that both JOIN messages from the same subscriber get lost, leading also to a disconnection; (in this case, the duration of the disconnection is at least equal to the interval separating two LEAVE-ALL messages). This also means that to meet a certain probability of disconnect P_d, the probability of packet loss could not be higher than the square-root of P_d. In Figure 11, we plot the required value for LeaveTime for $N = 50$, $P_d = 10^{-4}$, and various values of P_l ranging from 0 to 0.0045. The results show that P_l has very little effect, if any, on the choice of LeaveTime, which for this case remains 400 ms. Clearly, as P_l approaches the square-root of P_d, then the required LeaveTime becomes extremely large.

The effect of background load is seen in Figure 12 in which we plot for $N = 10$ and 25, the required LeaveTime for both $G_s = 7.5$ Mbps and $G_s = 0.5$ Mbps. It is seen that if the load is reduced, the required LeaveTime can be reduced as well. Also it is observed that for $G_s = 0.5$ Mbps (a very light background load,) the network delay component becomes insignificant, leaving only the gating delay component as the determining factor; the required LeaveTime is then more strongly dependent on JoinTime, and is about JoinTime/2.

The effect of burstiness in the background traffic also has a strong effect on the network delay component and thus the selection of LeaveTime. We plot in Figure 13 the required LeaveTime as a function of JoinTime for $N = 50$, $G_s = 7.5$Mbps, $P_d = 10^{-4}$, and various values of M_s, namely, 1,500, 12,000, and 64,000 bytes. It is seen that the required value for LeaveTime has to increase from 400 ms for $M_s = 1{,}500$ bytes to 1.2 s and 1.6 s for $M_s = 12{,}000$ and 64,000 bytes, respectively.

It is interesting now to examine the duration that a channel remains disconnected, that is the time from when the bridge has blocked the forwarding of packets until a JOIN message is received. We consider the case of $N = 50$, $G_s = 7.5$ Mbps, $M_s = 1{,}500$ bytes, $P_l = 0$, JoinTime = 100 ms, and the required LeaveTime to achieve $P_d = 10^{-2}$, 10^{-3}, and 10^{-4}, and plot in Figure 14 the histograms of disconnection time, given that a disconnection has occurred. It is observed that for all cases the disconnection time is below 200ms, with the majority of them below 100ms. As P_d is decreased to 10^{-4} (requiring LeaveTime to be increased to 328 ms), the tail of the distribution for the duration of disconnection itself gets shorter, and the disconnection time decreases to below 100 ms. We plot the same in Figure 15, but for $M_s = 64{,}000$ bytes. There, it is seen that the tail of the distribution spreads out to 2 s for $P_d = 0.01$ (with LeaveTime = 536 ms) and to 1 s for $P_d = 10^{-4}$ (with LeaveTime = 1,490 ms). This is indicative of the fact that, with bursty background traffic, capture effects could delay the JOIN messages quite significantly leading to long periods of disconnection.

We consider now 100Base-T, and plot in Figure 16 the required LeaveTime to achieve $P_d = 10^{-2}$ and 10^{-4} for $N = 10$, 25, and 50, with a background load of 75 Mbps and $M_s = 1{,}500$. The range of JoinTime of interest here is from 0 to 75 ms. As with the 10Base-T case, the required LeaveTime is relatively insensitive to the choice of JoinTime. While for $P_d = 10^{-2}$ LeaveTime is quite small (below 50 ms) and independent of N, for $P_d = 10^{-4}$, the required value of LeaveTime is quite dependent on N; it is suggested that LeaveTime be 50ms for $N = 10$, 125 ms for $N = 25$ and 350 ms for $N = 50$ (almost the same value as for the 10Base-T). This is indicative of the fact that with the larger population of users, the tail of the network delay distribution in 100Base-T is quite long, so that as P_d gets smaller, the required LeaveTime gets larger.

The effect of M_s in 100Base-T is seen in Figure 17 where we consider $N = 25$, $G_s = 75$ Mbps, and $M_s =$ 1,500, 12,000, and 64,000 bytes. It is seen that with 100Base-T, the increase in the required LeaveTime with M_s is not as severe as it is with 10 Mbps, due to the fact that 100Base-T has a smaller slot time and the transmission time of a packet that has acquired the channel is in absolute time 10 times smaller. It takes only values for LeaveTime of 150 ms and 200 ms for $M_s = 12{,}000$ and $M_s = 64{,}000$ bytes, respectively, as compared to the much higher values observed in 10Base-T.

IV. Conclusion

The selection of JoinTime is primarily driven by the interest to reduce the number of similar control messages needing to be transmitted by different devices on the same segment in response to an event (e.g., a reception of a LEAVE message). The appropriate value for JoinTime is dependent on the number of participants and the background load. The JoinTime values can thus be appropriately chosen depending on the situation. For instance, consider a switch in the backbone of a LAN. In general these switches share their segments with a few other bridges, and they may even have point-to-point links to other backbone switches. The amount of traffic is potentially very large, and the number of multicast channels, too. However, because there is a small number of GARP participants, most of the GARP requests are aggregated into a few PDUs. One should be using small values for JoinTime such as those shown in Table 1 for $N = 10$ and $G_s = 75\%$ of the total bandwidth for 10Base-T and 100Base-T.

The case of point-to-point links is a degenerate one, where there is no need for a LeaveTime, as a bridge does not have to wait for responses from other potential bridges (as there are none). Also the gating is only useful to do internal aggregation of control messages, as there is no possibility of doing any other reduction in the number of GARP PDUs. This requires a fairly small value for JoinTime.

On the other hand stub segments, may have larger numbers of GARP participants. There is a strong possibility of reduction of the number of control messages across the different stations. A larger spreading in the generation of GARP PDUs is therefore needed, which requires larger values for the JoinTime, as show the cases for $N = 100$. It is also possible that most of the devices in such leaf segments do not implement the registrar state machine, which will in turn limit the number of control messages that may be sent.

Once the JoinTime has been adequately chosen, it is essential to choose a value for the LeaveTime that is large enough to avoid disconnections. In networks that are designed to support multimedia traffic, burstiness should not be particularly an issue; this is because other studies have shown that the effect of bursty data traffic on multimedia streams is very destructive, as it quickly increases the latency experienced by the multimedia packets to unacceptable levels; thus, for networks supporting multimedia traffic, ways must be used in order to separate the two, including the installation of full duplex links and the implementation of prioritization. Such networks would then need a LeaveTime of the order of 600 ms or even lower. On the other hand, if burstiness is high as is the case in networks supporting data applications which could be producing bursts up to 64,000 bytes, then higher values for the LeaveTime, on the order of 1 s, need to be selected in order for GARP to work properly. The use of GARP/GVRP in IEEE 802.1Q [802.1Q] for the dynamic registration and deregistration of stations to Virtual LANs should consider values for LeaveTime obtained with high degrees of burstiness to minimize the possibility of disconnection.

V. References

[802.1p] IEEE, 802.1p, Standard for Local and Metropolitan Area Networks - Supplement to Media Access Control (MAC) Bridges: Traffic Class Expediting and Multicast Filtering (Draft 5).

[802.1D] IEEE, 802.1D, Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges, 1990.

[802.1Q] IEEE, 802.1Q, Standard for Virtual Bridged Local Area Networks, Draft 4, 1996.

[802.3] ISO/IEC 8802-3: 1996 [ANSI/IEEE Std 802.3, 1996 Edition], Information technology--Local and metropolitan area networks--Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

[JSAC97a] C. Diot, W. Dabbous, and J. Crowcroft, Multipoint Communications: A Survey of Protocols, Functions and Mechanisms, IEEE Journal on Selected Areas In Communications, April 1997.

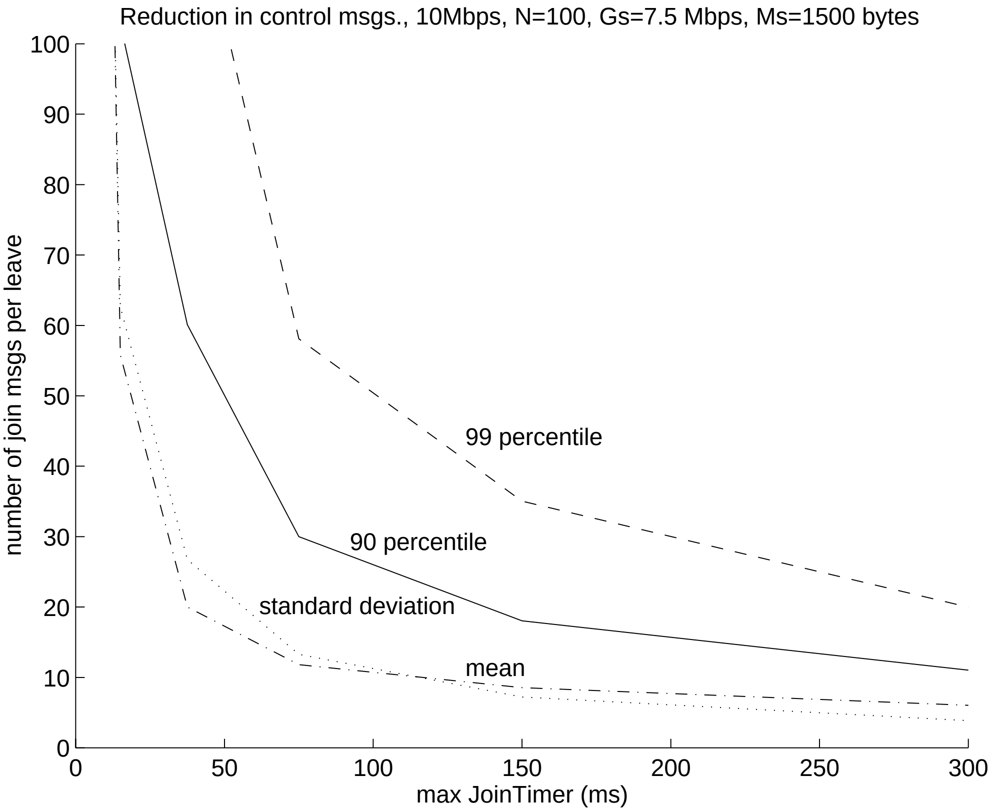

Figure 1. Number of control messages vs. JoinTime for the baseline scenario (10Base-T, N = 100, G_s = 7.5 Mbps, M_s = 1500 bytes).

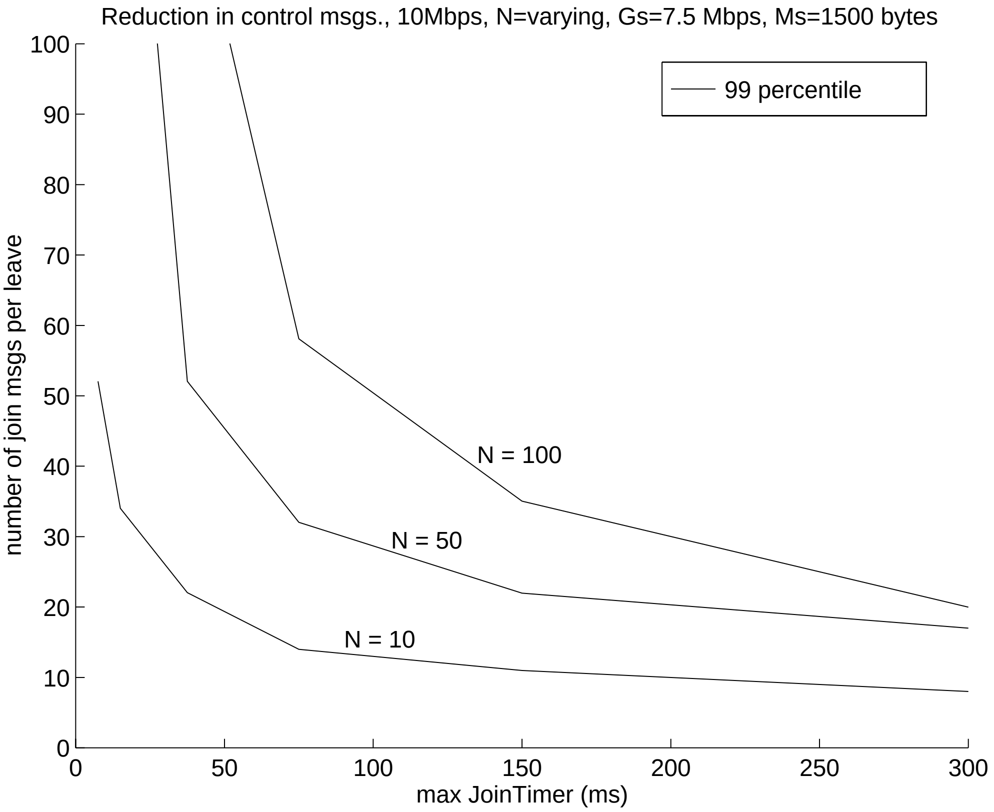

Figure 2. Number of control messages vs. JoinTime for various values of N (10Base-T, G_S = 7.5 Mbps, M_S = 1,500 bytes).

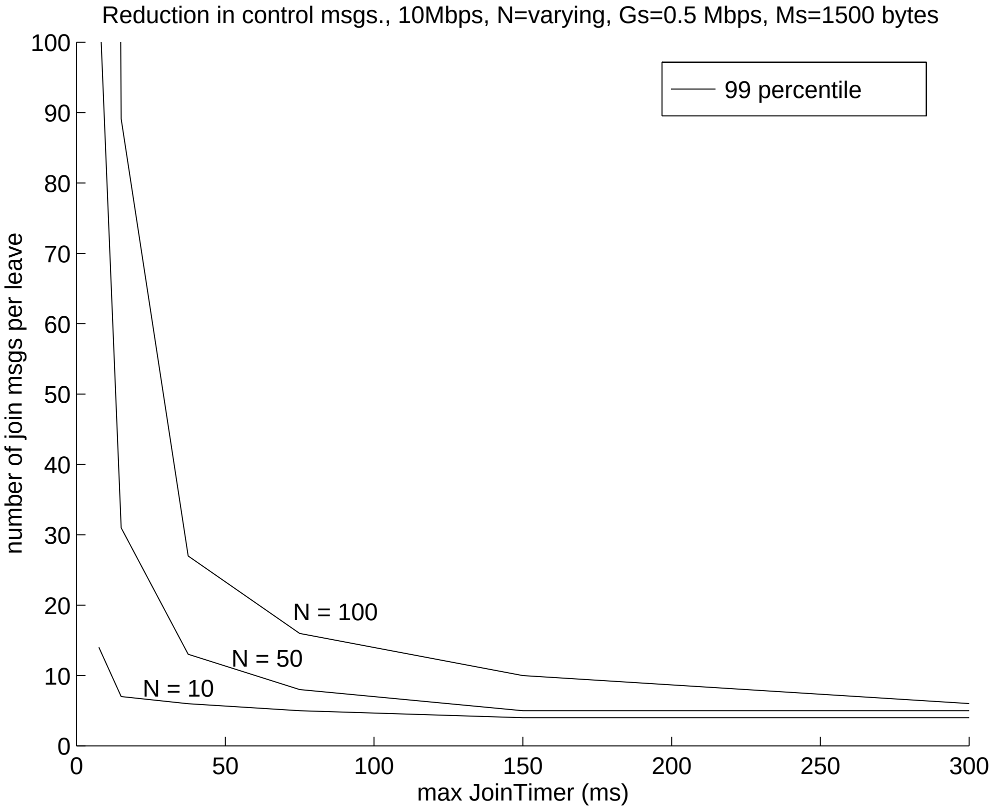

Figure 3. Number of control messages vs. JoinTime for various values of N (10Base-T, G_s = 0.5 Mbps, M_s = 1,500 bytes).

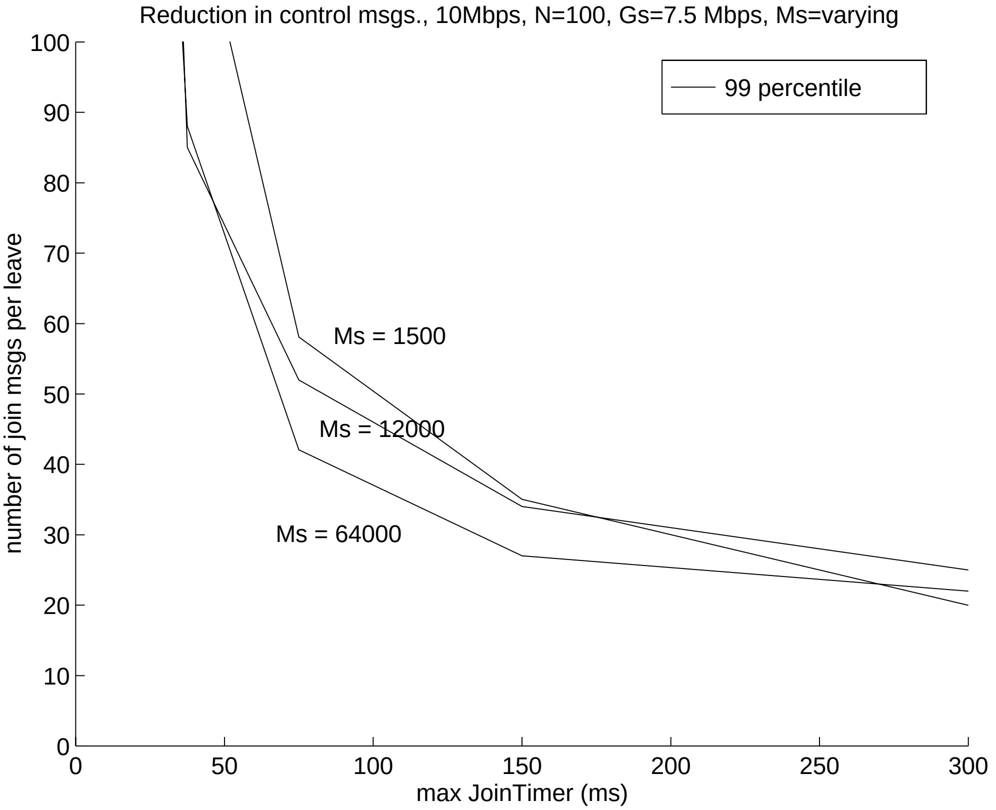

Figure 4. Number of control messages vs. JoinTime for various values of M_s (10Base-T, N = 100, G_s = 7.5 Mbps).

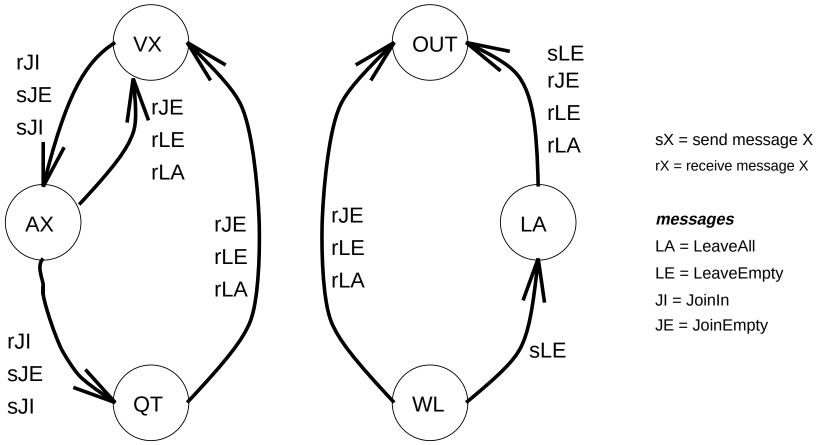

Figure 5. GARP applicant state machine (simplified).

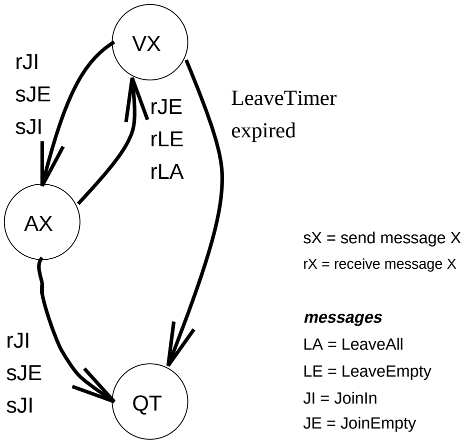

Figure 6. GARP registrar state machine.

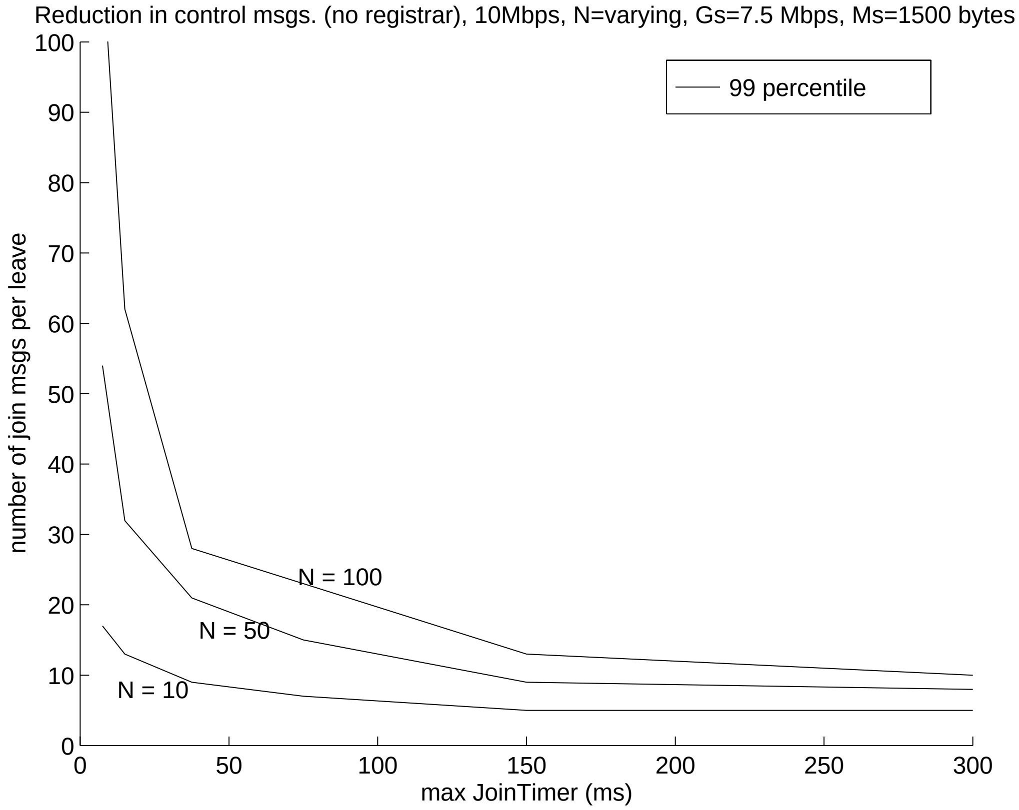

Figure 7. Number of control messages vs. JoinTime for the case where the registrar at the stations is removed (10Base-T, various values of N, G_S = 7.5 Mbps, N= 1500 bytes).

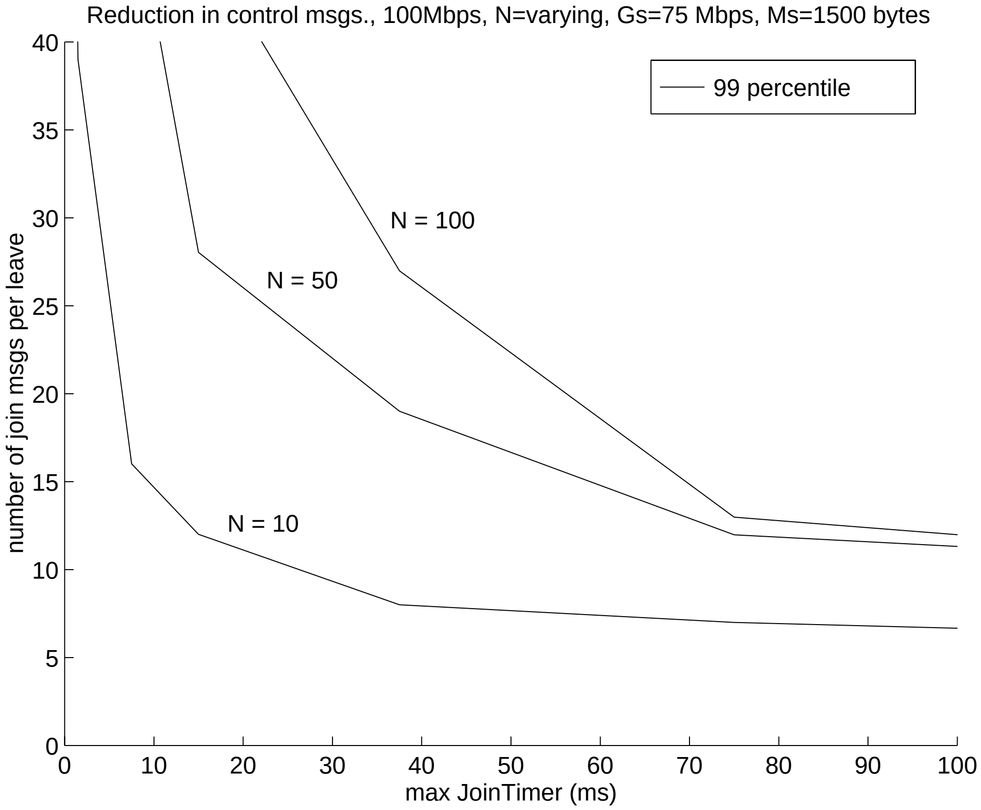

Figure 8. Number of control messages vs. JoinTime for various values of N (100Base-T, G_s = 75 Mbps, M_s = 1,500 bytes).

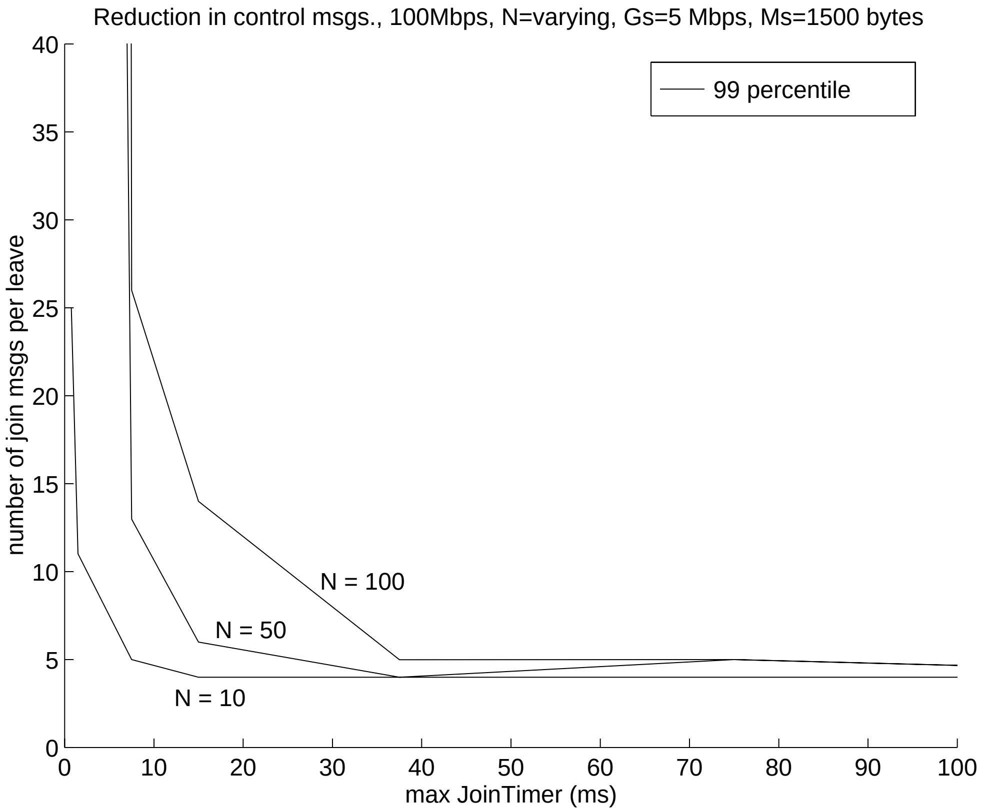

Figure 9. Number of control messages vs. JoinTime for various values of N (100Base-T, G_S = 5 Mbps, M_S = 1,500 bytes).

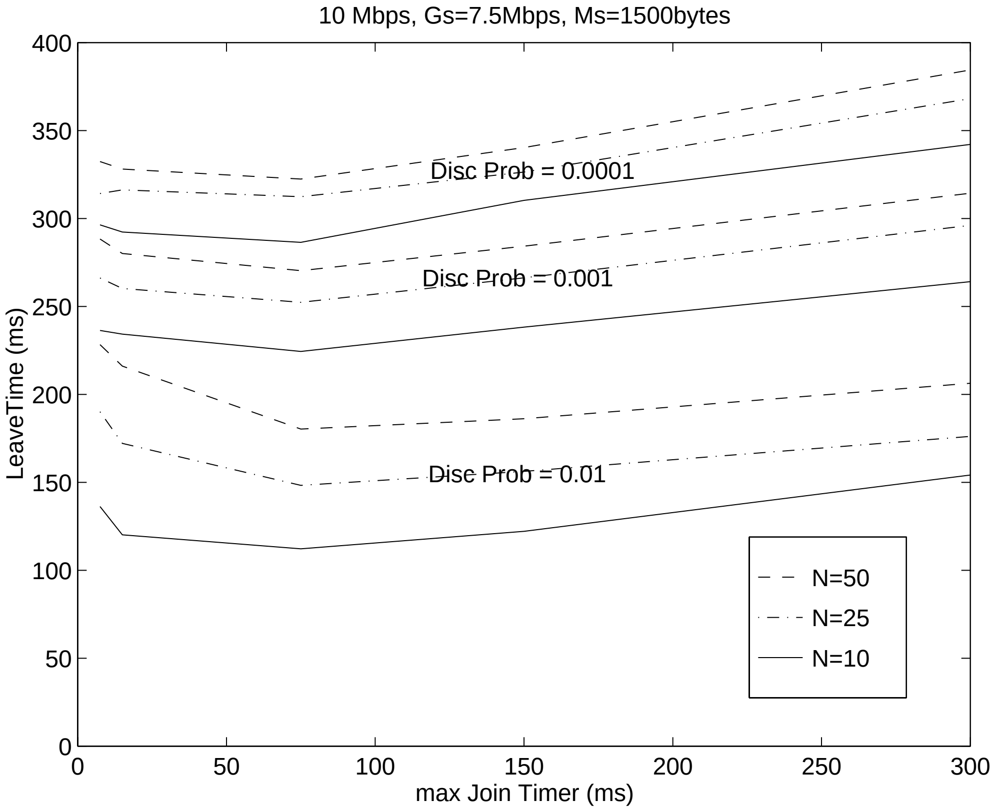

Figure 10. Required LeaveTime vs. JoinTime to achieve various values of P_d (10^{-2}, 10^{-3}, 10^{-4}) for various values of N (10, 25, 50) and a zero probability of packet loss (10Base-T, G_s = 7.5 Mbps, M_s = 1,500 bytes).

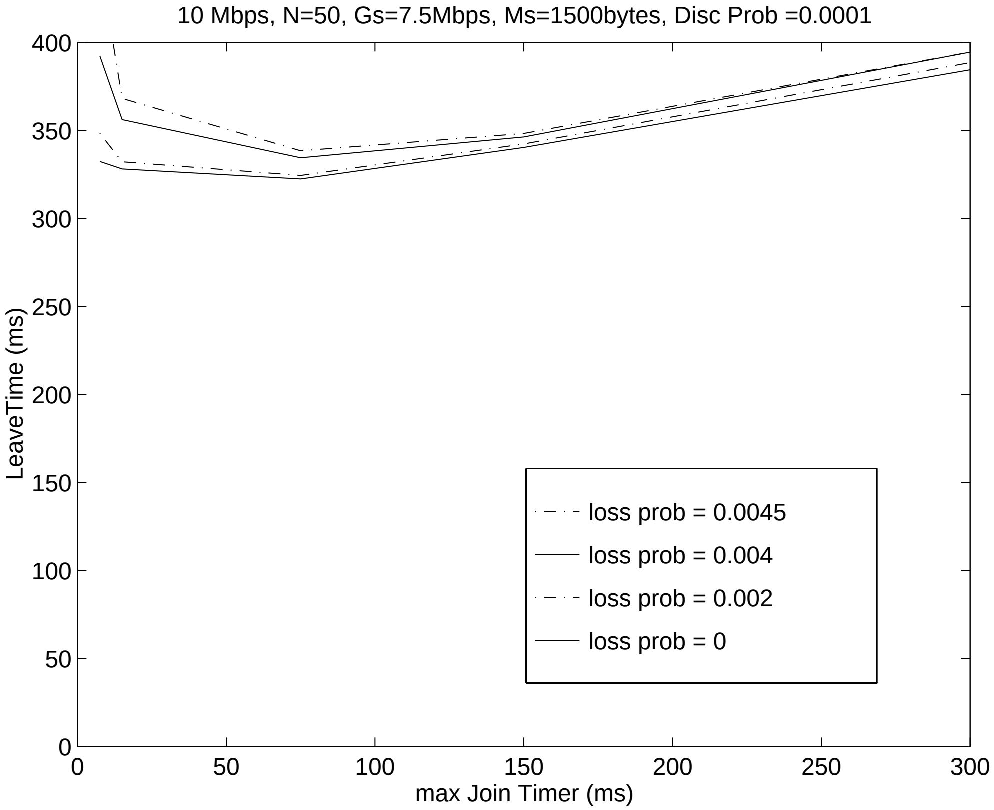

Figure 11. Required LeaveTime vs. JoinTime to achieve $P_d = 10^{-4}$ with various values of the packet loss probability P_l (10Base-T, $N = 50$, $G_s = 7.5$ Mbps, $M_s = 1,500$ bytes).

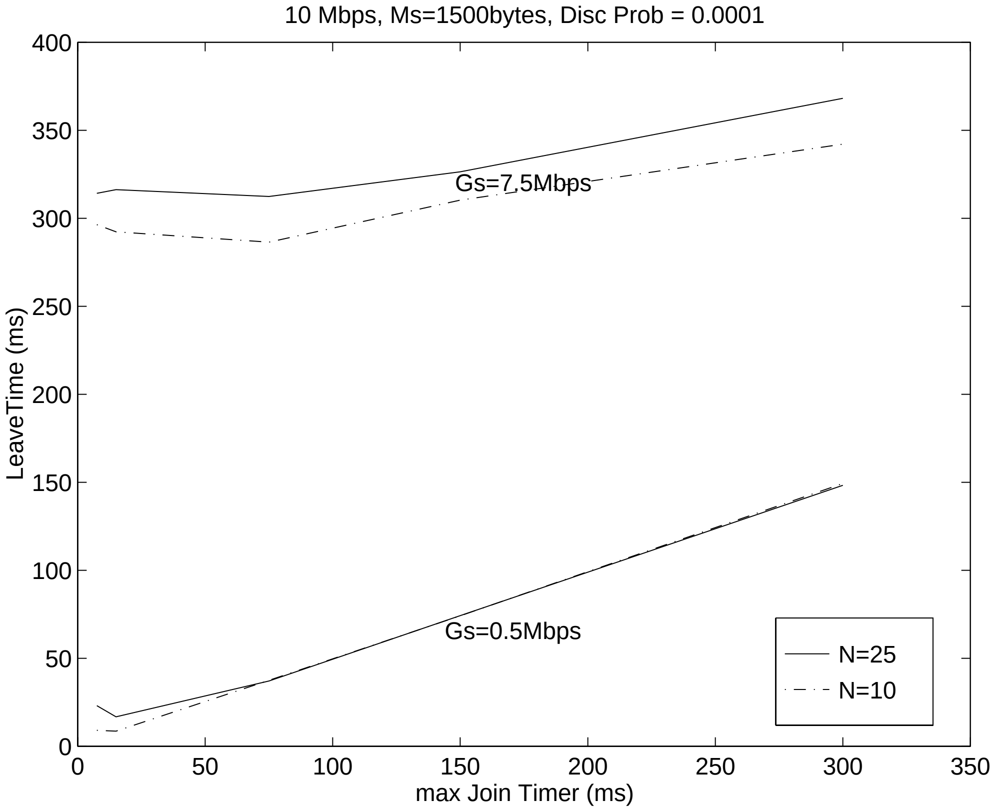

Figure 12. Required LeaveTime vs. JoinTime to achieve $P_d = 10^{-4}$ for different values of background load (10Base-T, M_s = 1,500 bytes).

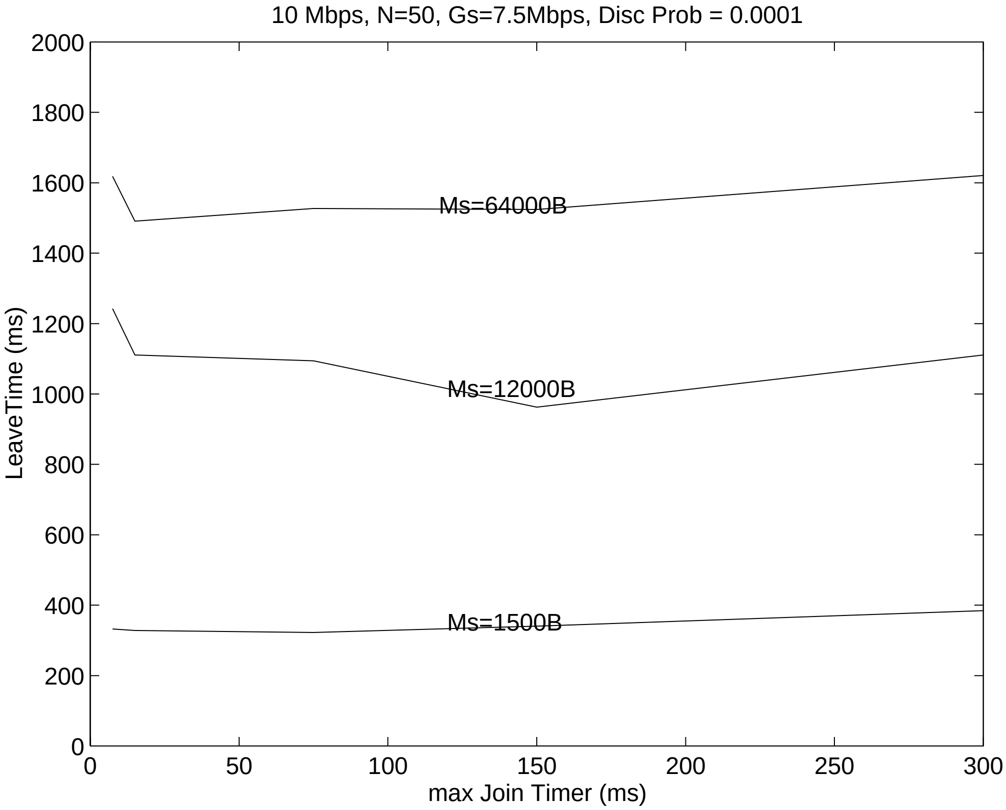

Figure 13. Required LeaveTime vs. JoinTime to achieve $P_d = 10^{-4}$ for various values of M_s (10Base-T, N = 50, G_s = 7.5 Mbps).

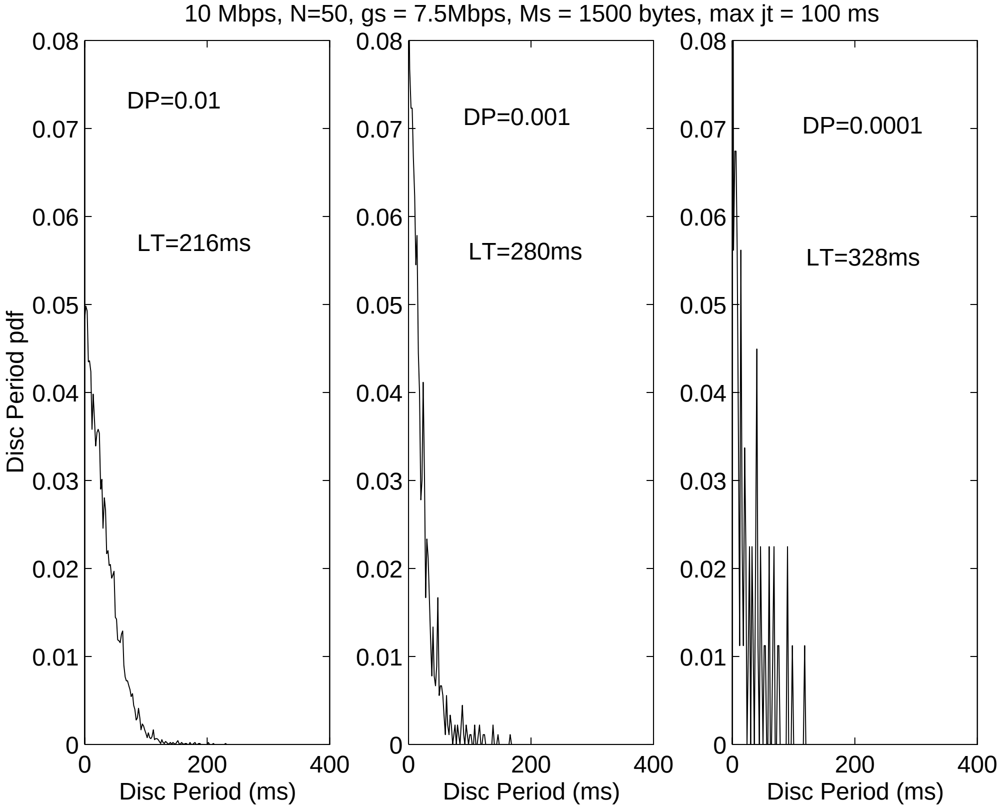

Figure 14. Histogram of the duration of disconnection for JoinTime = 100ms and the required values for LeaveTime to achieve various values of P_d (10Base-T, N = 50, G_s = 7.5 Mbps, M_s = 1,500 bytes).

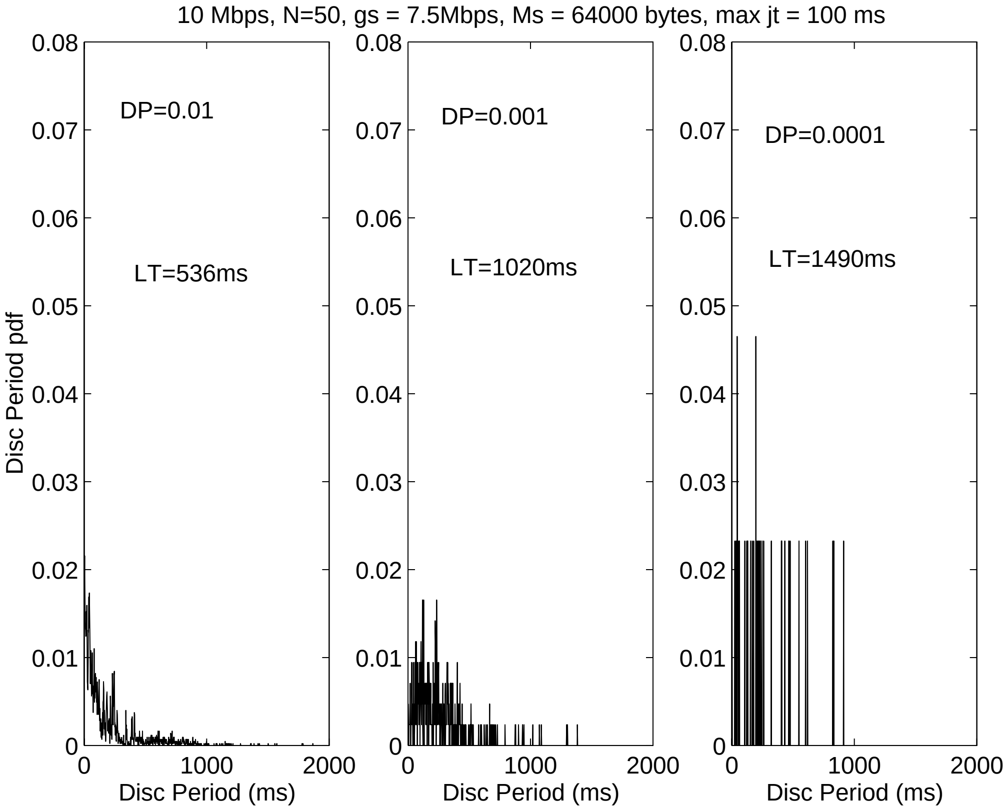

Figure 15. Histogram of the duration of disconnection for JoinTime = 100ms and the required values of LeaveTime to achieve various values of P_d (10Base-T, N = 50, G_s = 7.5 Mbps, M_s = 64,000 bytes).

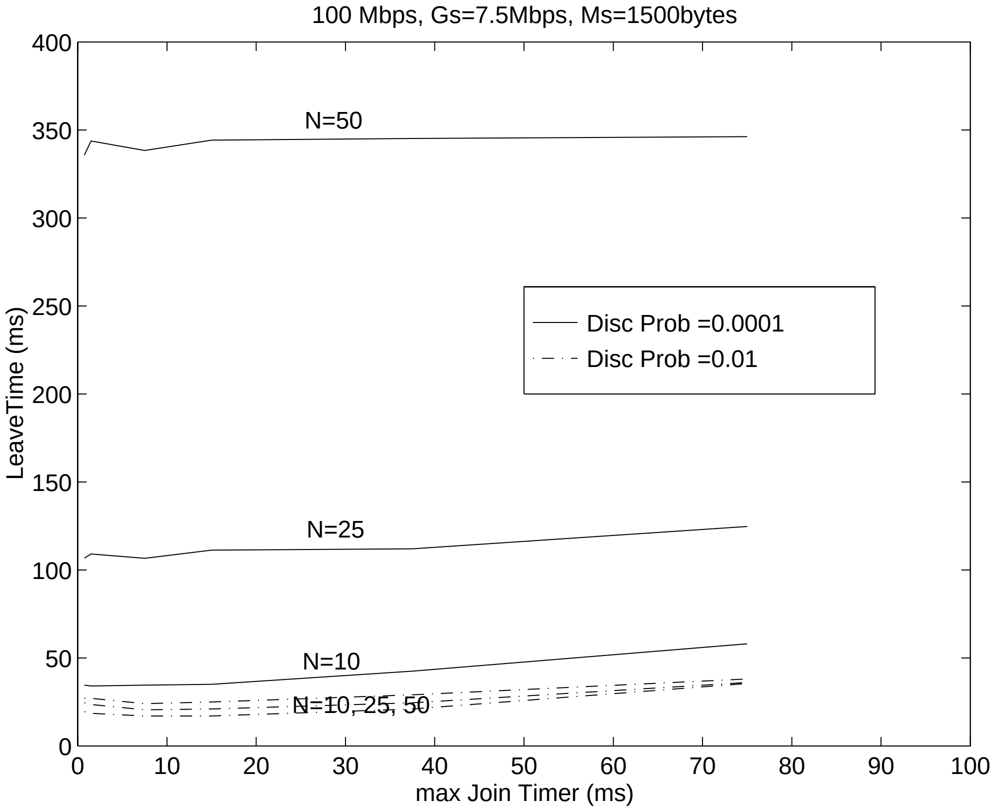

Figure 16. Required LeaveTime vs. JoinTime to achieve various values of P_d (10^{-2} and 10^{-4}) for various values of N (10, 25, 50) (100Base-T, G_s = 75 Mbps and M_s = 1,500 bytes, P_l = 0).

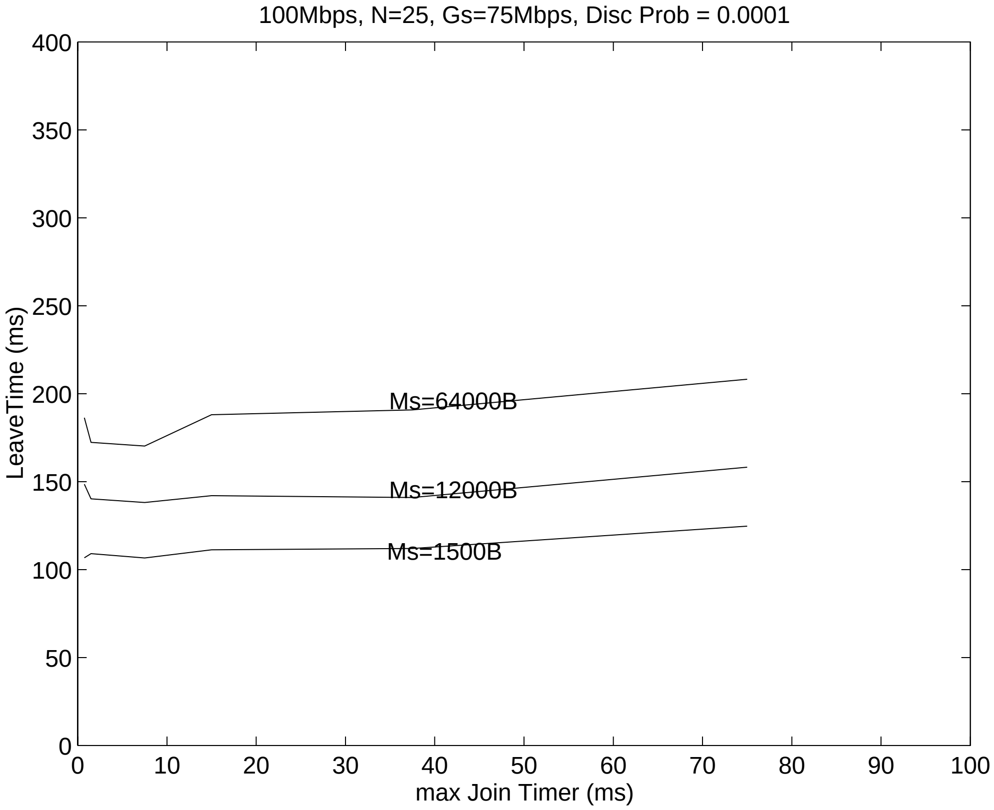

Figure 17. Required LeaveTime vs. JoinTime to achieve $P_d = 10^{-4}$ for various values of M_s (100Base-T, N = 25, G_s = 75 Mbps, $P_l = 0$).