Performance Evaluation of the Preemptive Bandwidth Allocation Multicast Protocol

Nawel Chefaï, Gregor V. Bochmann, Nicolas D. Georganas

SITE, University of Ottawa
800 King Edward Ave. Ottawa, On. K1N 6N5, Canada
{chefai, bochmann, georganas}@site.uottawa.ca

Abstract

We present the Preemptive Bandwidth Allocation Multicast Protocol; a distributed multicast QoS-aware signaling protocol that adapts the users' bandwidth requirements to the limited resources available in the network by preempting bandwidth of less prioritized streams from existing multicast groups. We assume that each multicast group will have different multicast streams with predefined quality requirements and each stream will have a priority level assigned to it. When a join request comes to the network and there is a lack of bandwidth, the communication service will try to preempt some streams of existing multicast groups to satisfy the new request without disconnecting the basic stream of any of these multicast groups. The aim is to accommodate the maximum number of users within the network with at least their minimum requirements (e.g. the I frames of an MPEG video). We also present a performance evaluation that compares two versions of the distributed multicast preemptive approach with the traditional non-preemptive one.

1 Introduction

Multicast routing protocols can be classified into two categories: source-based and shared-based multicast trees. A *source-based multicast tree* constructs the multicast tree starting at the root to reach all the destinations. Distance Vector Multicast Routing Protocol (DVMRP) [1] and Multicast Open Shortest Path First (MOSPF) [2] are examples of such schemes. A *shared multicast tree* is a mode where a "meeting place", called a *core* or *Rendezvous Point*, is advertised for each multicast group, toward which sources send their packets and receivers send explicit join messages. The Core Based Tree (CBT) [3] and the Protocol Independent Multicast-Sparse Mode (PIM-SM) [4] are examples of such multicast trees. These multicast routing protocols, which construct only the shortest paths between the source/core and the receivers for a given multicast group, do not consider the users' QoS requirements. However, QoS routing, which is the process of finding a path from the destination to the source with a specific reservation of resources, is necessary for multimedia applications, such as videoconferencing, that have stringent QoS requirements, particularly when limited resources are available in the system. In the common approach, when there is a lack of bandwidth, the system will either refuse a new connection (in a non-prioritized system) or will disconnect less prioritized connections in a prioritized system in order to offer the preempted bandwidth to the new connection requests. Algorithms proposed in [5] and [6] are examples of such schemes where preemption of lower priority connections is considered to admit higher priority requests in peer-to-peer scenarios. Another approach is to degrade a given connection, instead of disconnecting it completely, when a new request enters the system and there is lack of bandwidth. Sakate et al. [7] have proposed a centralized algorithm that responds to the above requirement. In [8] a distributed version of Sakate's algorithm, in peer-to-peer scenarios, has been proposed. In this paper, we present a distributed version of the same protocol for multicast scenarios. We will therefore look for the missing bandwidth among existing multicast groups by reducing their bandwidth without disconnecting them. For instance, instead of refusing a new user to join a multicast group when the branch that connects the user to the multicast tree does not have enough bandwidth, we allow for bandwidth degradation of existing multicast groups in order to admit the new user. This operation requires the introduction of priority levels to the multicast groups. In fact, we assume that a multicast data can be divided into multiple streams; for each multicast group, the i^{th} stream will be characterized by an amount of bandwidth, Bandth (i), and a priority level, PLi. The sum of all the bandwidths of the streams belonging to the same multicast group is the maximum bandwidth of that multicast group. We consider that each multicast group has a basic multicast stream that should never be preempted. This stream, to which we assign the minimum bandwidth requirement of the multicast group and the highest priority value, will never be preempted. However, the other streams of a multicast group, which have lower priority values, may be subject to preemption from new join requests, if not enough bandwidth is found along the path that permits these joins to graft to their respective multicast tree. The *Preemptive Bandwidth Allocation Multicast Protocol* (PBA-M Protocol), which has been designed to choose streams for preemption among the non-basic ones to satisfy the minimum bandwidth of a new join request, maximizes the number of users admitted to the network with at least their minimum bandwidth requirements, and minimizes the loss of priority among the degraded multicast groups.

The rest of the paper is organized as follows. In Section 2, we will give an overview of the PBA-M Protocol. In Section 3, we will describe the different algorithms of the PBA-M protocol and the different selection criteria that can be used to select the multicast streams to be preempted. In Section 4, we will compare this protocol to a non-preemptive protocol and evaluate its performance when using two selection criteria. Section 5, contains concluding remarks.

2 PBA-M Protocol Overview

Our approach to the QoS routing problem consists of the link-constrained problem, which corresponds to consider the bandwidth availability on the path from the source to the destination. In our design, a controller, named the *Controller of Bandwidth Preemption* (CBP), which resides at each node, will check for the admission of a join request when it arrives at the node. The CBP controllers involved in the admission of a given join request convey messages between one another in order to (1) coordinate the decisions of stream preemptions and (2) update the allocated bandwidth of the degraded multicast groups. When treating a join request, the PBA-M protocol goes through two phases: the *Feasibility Phase* and the *Confirmation or* the *Release Phas*e. The feasibility phase consists of looking at the path feasibility: starting from the destination and on each node of the path towards the in-tree node, a *path_feasibility* packet will arrive and the CBP controller will check if the requirements of the new request can be satisfied. During this phase, the controller may find out that the available bandwidth on the link is not sufficient and the bandwidth of existing multicast streams must be preempted to satisfy the minimum bandwidth requirement of this new request. As soon as a path_feasibility packet arrives at the in-tree node, the CBP controller will be certain that the path is feasible. In this case, it will generate a packet, called *path_confirmation* packet, which will be sent towards the receiver that generated the request in order to confirm the establishment of the path. This phase is called the confirmation phase. However, as soon as a CBP controller cannot find the minimum bandwidth required by a new join request on a given link, it will enter the release phase by generating a *path-release* packet, which will deallocate the reserved bandwidth.

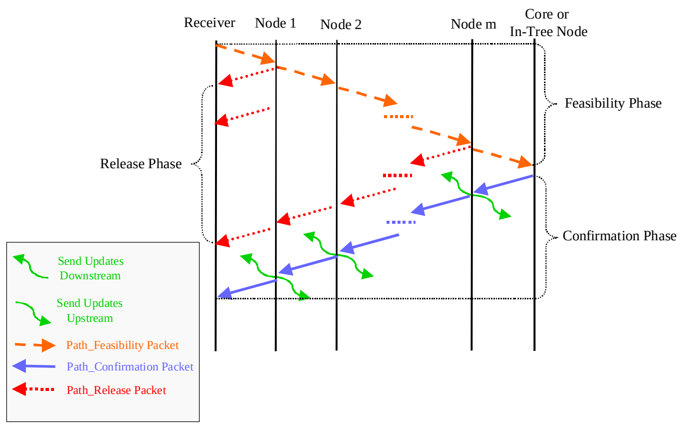

Fig. 1: The Late Preemption Update Protocol (LPU Protocol)

To preempt a stream, we have considered the pessimistic approach that waits to know whether the path is feasible before preempting any streams. Therefore, during the feasibility phase, the CBP controller will only mark the selected multicast streams for preemption. It will preempt the streams only during the confirmation phase when it knows for sure that the path is feasible. Thus, the update packets, which are needed to update the bandwidth allocated to the degraded multicast groups, are sent during this phase. We call this approach the *Late Preemption Update Protocol (LPU Protocol)* (Fig. 1). Another approach is to preempt the streams during the feasibility phase without knowing if the rest of the path is feasible. Thus, the update packets are sent during this phase to update the bandwidth allocated to the degraded multicast groups. We call this approach the *Early Preemption Update Protocol (EPU Protocol)*. In this paper, we present only the pessimistic approach. A comparative study of these two approaches in peer-to-peer scenarios can be found in [8].

3 The Controller of Bandwidth Preemption (CBP Controller)

In this section, we will describe the three phases of the protocol: the feasibility phase, the confirmation phase and the release phase. It should be noted that the peer-to-peer version of this protocol is given in [8]. In the following we will emphasize the issues related to multicast scenarios.

3.1 Feasibility Phase

During the feasibility phase, the CBP controllers use the path_feasibility packet to check for the admission of a given join request and to reserve the bandwidth along its path. To admit a new request on a given interface, the controller takes different actions depending on whether it finds the requested multicast group already established or not yet established on this interface.

a) Multicast Group Already Established: When a request enters a node and the controller finds that the multicast group that this request wants to join is already established on the interface, the controller will have to go through several steps. First, it will have to check whether any of the multicast streams that this request wants to join are confirmed or not. When the controller finds that one or more of the multicast streams are not yet confirmed, it will have to add this new join request identification to the list *members_of_mult_stream* to indicate the willingness of this join request to get this not yet confirmed multicast stream. The list *members_of_mult_stream* permits to manage multiple reservations on a not yet confirmed multicast stream. In fact, if any of the requests that have put their identifications within this list cannot get this multicast stream because of the non-feasibility of the path, the controller will not release this multicast stream bandwidth reservation until the list of *members_of_mult_stream* becomes empty. Having multiple requests on a given node that are all at their feasibility phase can occur when the number of requests is very high (high request rate) or when the control packets, the path_feasibility, the path_confirmation or the path_release packets, got lost. However, if we suppose that the control packets will never get lost, an alternative to the systematic sending of the path-feasibility packet from subsequent requests, which probe for the feasibility of the same path, is possible. In fact, in order to diminish the number of packets that probe for the feasibility of the same path on the same multicast group, the controller could decide to send only the path-feasibility packet of the first request that arrives to the node. The path-feasibility packets of the subsequent requests that arrive to this node are discarded and the controller will wait for either the path_confirmation packet or the path_release-packet to be able to respond to the pending requests.

Additionally, it may happen that the bandwidth requirements of this join request is still not completely fulfilled by the already confirmed or reserved bandwidth of the multicast group, which means: (1) that the new request has a higher bandwidth requirement than what already exists within this multicast group and (2) that there is at least one multicast stream within the multicast group that still does not have a bandwidth allocated or reserved to it. In this case, the controller will have to check whether the interface can offer the missing bandwidth required by this join request. If it finds enough bandwidth within the interface that permits to add one or more multicast streams to the multicast group, it will have to mark this bandwidth locked for these multicast streams and will add this join request identification to each list *members_of_mult_stream* of these newly not yet confirmed multicast streams. The path_feasibility packet is then sent to the next node to verify if the rest of the path can offer the previously reserved bandwidth. It should be noted that the path_feasibility packet is discarded and the protocol enters the confirmation phase when the multicast group bandwidth allocation corresponds to the requested bandwidth requirements or when the link cannot provide more bandwidth to the request than what already exists in the multicast group.

b) Multicast Group Not Yet Established: When a join request enters a node and the controller does not find the requested multicast group, the controller will first check if enough bandwidth is found on the interface. If it finds at least the minimum bandwidth required by the request, which corresponds to the multicast group minimum bandwidth, the controller will not make any preemption to admit the join request. It will add this multicast group identification on the node with its status equal to 0, which indicates that this multicast group is still not confirmed, and will add the join request identification to the list *members_of_mult_group*. The join request will then have passed successfully the First Admission Control at this node and the path_feasibility packet is sent to the next node towards the core of the multicast group.

However, when the requested minimum bandwidth cannot be fulfilled with what the interface can offer, the controller will have to go through a second admission control. Within this second step, the controller will have to select certainly multicast stream(s) for preemption. In the following, we will describe two paradigms for selecting multicast streams for preemption: one-level criterion and two-level criterion.

The One-Level Multicast Stream Selection Criterion: We propose two possible multicast stream selection criteria: the *LP* criterion and the *LMD* criterion. The LP criterion corresponds to the Lowest Priority multicast stream criterion and selects the lowest priority multicast stream among those available for preemption. The LMD criterion corresponds to the Lowest number of Members Degraded criterion and selects the multicast stream (among those available for preemption) that has the lowest number of members to degrade.

The Two-Level Multicast Stream Selection Criterion: In the case that there are several streams that fulfill one of the above criteria, a second criterion could be applied to make a final selection. This two-level criterion paradigm can be expressed as follows: given two criteria *X* and *Y*, the combination of the criterion *X* with the criterion *Y*, represented by *X*-*Y*, means that we select first the streams that fulfill the criterion *X* and among these streams select one that fulfills the criterion *Y*. Thus with this paradigm, we can get two other policies: the *LP-LMD* and the *LMD-LP*.

In the case we have more than one stream that fulfills the one-level criterion or the two-level criteria, we select a stream randomly.

After having selected the multicast streams for preemption in order to fulfill the minimum bandwidth requirement of the join request, the controller will send the path_feasibility packet to the next node until the multicast group is encountered or the core node is reached. If not enough bandwidth is found among the other multicast groups to satisfy the minimum requirement of the request the protocol will enter the release phase.

It should be noted that the LMD criterion adds more overhead to the state information to be saved on each node. In fact, this criterion requires the knowledge of the number of members attached to the non-basic multicast streams and needs this information on each node on the multicast tree.

Unselecting the Previously Selected Multicast Streams: When the controller finds that it needs to consider preemption to fulfill the minimum bandwidth of the new join request and that some streams have been selected for preemption on the previous link, an additional test needs to be performed to maintain the streams' priority precedence within the multicast group. In fact, for every multicast stream already selected for preemption on the previous link and which also traverses this link, the controller needs to test whether this multicast stream is effectively the lowest priority multicast stream within the multicast group on the current link. If the considered multicast stream previously selected for preemption is of a higher priority than the other multicast streams within the same multicast group not yet preempted, then the controller will not select this multicast stream for preemption in order to respect the precedence that exists among the multicast streams on the same multicast group. Instead, it will have to select another stream among those available for preemption following one of the criteria described above. This situation happens when the up-multicast tree has one or more multicast streams that the down branches of the same multicast tree do not have.

Furthermore, in order to coordinate the different multicast stream preemptions, the controller would need to send the path-feasibility packet towards the core node as long as the bandwidth of the multicast group is not confirmed. In fact, even if the multicast group were encountered before the request reaches the core, it would happen that some or all of the bandwidth of the multicast group is still not confirmed. This is the case when the multicast group and/or when one or more of its multicast streams are not confirmed yet. The path-feasibility packet will be discarded as soon as the request bandwidth requirement corresponds to the already confirmed multicast group bandwidth.

3.2 Confirmation Phase

During the confirmation phase, the CBP controllers use the path_confirmation packet, which is conveyed from the core or from an in-tree node to the receiver, to coordinate the bandwidth allocation of a given connection request along its path.

It should be noted that the confirmation phase starts whenever the path_feasibility packet arrives (1) at an intermediate node and the controller finds that the requested bandwidth of the join request is already confirmed, which means that the multicast group is already established and the multicast streams that the new request wants to get are also established or (2) at the multicast group core, which means that the multicast group or/and the multicast streams are not yet confirmed.

A join request that enters the confirmation phase can have either been admitted without or with preemption. When a join request did not necessitate bandwidth preemption, it could happen that this request negotiated a better bandwidth than what already existed within the multicast group. In this case, the members that are attached to the same LAN as this request could benefit from this additional bandwidth and could therefore get the newly established multicast streams. The path_confirmation packet could be used to announce this upgrade to the members when it arrives at the LAN.

However, when a join request necessitates bandwidth preemption to satisfy its minimum bandwidth requirement, the controllers along the path will need to preempt the selected multicast streams and send upstream and downstream update packets. The downstream packets are sent towards the members without further tests to update the bandwidth allocated to the degraded multicast group and to inform the members about the change of their current bandwidth allocation. However, the controller will need to make further tests before making the upstream update.

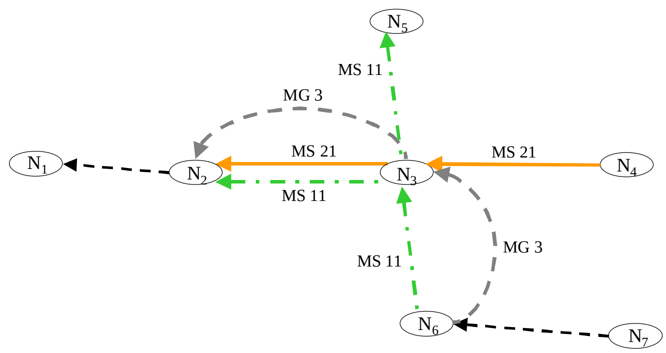

Fig. 2: Case of Discard of the Upstream Update Packet

An illustration of this case is given on Fig. 2, where we have the already established multicast group MG 1 with two multicast streams: MS 11 and MS 12 on the links (N6, N3) and (N3, N5) and one multicast stream MS 11 on link (N3, N2) and where we suppose that a new join request, whose path includes the links (N2, N3) and (N3, N6), wants to establish the multicast group MG 3. In this example, we suppose that no preemption is required on the kink (N6, N3). However, we suppose that there is a lack of bandwidth on link (N3, N2) and that the controller will preempt MS 11 to satisfy the minimum bandwidth of this request. The controller will therefore send an upstream packet as well as a downstream packet from node N3. However, and as mentioned earlier, the upstream update will be made only if the upstream branches do not use the preempted multicast stream. Therefore, and because the sub-tree (N6, N3) and (N3, N5) of the multicast group MG 1 makes use of the multicast stream MS 11, the controller will not make the update on the link (N6, N3).

More generally, before preempting any multicast upstream, the controller will need to test if this very multicast stream is used on another interface. If it is used, the update of the previously degraded multicast is not done and the upstream update packet is discarded. However if the multicast tree keeps the identification of its members, in the case we use the LMD criterion, the controller will still need to propagate the update packet to remove the identification of the members that have been preempted on the previous links. It should be noted that a counter representing the number of members attached to a given multicast stream would be a better alternative to keeping the identification of the members. Thus, whenever a stream is preempted or a member joins the multicast stream, this counter will be decremented by the number of members degraded or incremented by one, respectively.

Additionally, we may be in the situation where a multicast stream is preempted but some requests, which are still in their feasibility phase, want to join this stream. In that case, the reserved bandwidth of these requests, on these nodes, would diminish since the bandwidth of the preempted stream would not be anymore available to these requests. Therefore, we may have the latest amount of bandwidth negotiated during the feasibility phase higher than what the nodes are still reserving for these requests. Thus, to allow bandwidth coordination between all the nodes along these requests, these requests confirmation packets should update the allocated bandwidth of these requests to the minimum between the two values: the local reservation found on the node and the amount of bandwidth that the last node on the feasibility path has agreed on.

In the case of the downstream update, no additional test is needed at the intermediate nodes. However, to be able to notify the receiver about the update, the designated router of the LAN will have to send the update information to each of the confirmed members that have joined this specific multicast stream.

3.3 Release Phase

During the release phase, the CBP controllers use the path_release packet, which is conveyed from the node, which failed to satisfy the two-admission control steps during the feasibility phase, towards the receiver in order to deallocate the reserved bandwidth and to inform the receiver of the reject of its join request. The reserved bandwidth could come either from the links or from the multicast streams selected for preemption. The controller will then have to give back to the corresponding links the reserved link bandwidth if the latter is not held by other requests. In fact, because of the possibility of having multiple requests holding onto the same reserved bandwidth, the controller should not release the reservation as long as there are still requests in the lists *members_of_mult_group* and/or *members_of_mult_stream*.

Furthermore, the multicast streams selected for preemption will not be unselected systematically. The controller will have to check first whether the selected multicast streams have been preempted. When a stream is preempted and the controller finds that the multicast stream is reserved for a given join request, it will set a flag associated to this multicast stream to indicate that this stream should be preempted whether or not the request that wants to preempt it, preempts it effectively.

4 Performance Evaluation of the PBA-M Protocol

The performance evaluation of the protocol is intended to (1) compare it to a non-preemptive protocol and (2) show the impact of choosing the LP criterion over the LMD criterion. For this purpose, we evaluated the PBA-M protocol when applying the LP, and LMD-LP policies using the OPNET simulator. It should be noted that we have preferred to evaluate the LMD-LP policy instead of the LMD policy because the objective of our work is to minimize the loss of priorities.

For the evaluation, we have used an intra-domain network topology of 15 LANs and 15 routers generated randomly using a version of the Tiers program [9] adapted to the OPNET environment. The data rate of the links is 33kunits of bandwidth. We have defined 45 multicast groups where each LAN has three different multicast senders. Each of these LANs has one designated core, which is the router connected to the designated router of the LAN. Each of these 45 multicast senders belongs to one of the three traffic categories described in Table 1 (no quality value is assigned to *Stream (1)* since it is the basic stream and it will never be considered for preemption). More precisely, to each LAN, which has 3 multicast senders, we have assigned the following traffic categories: the multicast group 1 has been assigned the traffic category 1; the multicast group 2 has been assigned the traffic category 2 and the multicast group 3 has been assigned the traffic category 3.

Table 1. The Used Traffic Categories

Tra-ffic	Min_ Bandth	Max_ Bandth	Stream (2)		Stream (3)	
			Band.	Prior.	Band.	Prior.
1	300	1500	1200	18	None	None
2	200	1200	300	16	700	2
3	400	1500	300	6	800	1

These three different traffic categories give us 5 different priority-levels: (1) Priority 18 with the associated bandwidth of 1200units, (2) Priority 16 with the associated bandwidth of 300units, (3) Priority 6 with the associated bandwidth of 300units, (4) Priority 2 with the associated bandwidth of 700units, and (5) Priority 1 with the associated bandwidth of 800units.

We also assume that each LAN has an unlimited number of receivers. The hosts at each LAN generate a join request to a randomly chosen multicast group following a Poisson distribution.

At the starting point of the simulation, there is no traffic in the network till the receivers start sending join requests to randomly chosen cores (each LAN has one designated core) and to randomly chosen multicast groups, using a uniform distribution. When a join request gets admitted to the network, it will have a connection lifetime that follows an exponential distribution. The join and leave scenario will permit us to evaluate the two approaches in a stable system.

4.1 Analysis

The collected data during these simulations are: (1) the average number of admitted members, (2) the average number of non-basic streams at the LANs (each admitted member will have at least the non-preemptable basic stream and, if the system is not overloaded, other streams that are subject to preemption to which priorities are inherited from the multicast group that the member is attached to), (3) the average number of degraded members, (4) the average number of preempted multicast streams, (5) the percentage of non-basic streams per priority and (6) the percentage of preempted multicast streams per priority.

The average inter-arrival time of 70 units of time between consecutive requests has been chosen to simulate a low requests rate in the system. This rate permits a given join request to be treated by the controllers along its path without having to deal with other join requests. In these performance studies we do not consider the case of concurrent requests.

To be able to analyse the LP and LMD-LP policies under different network loads, we made simulations with different connection lifetimes of 15mn (900 units of time), 25mn (1500 units of time), 35mn (2100 units of time), 45mn (2700 units of time), 55mn (3300 units of time), 65mn (3900 units of time), 75mn (3900 units of time), 85mn (5100 units of time), 95mn (5700 units of time) and 105mn (6300 units of time). For each on these connections lifetime durations, we have run 11 simulations with different seed numbers. The confidence interval, which is shown at the top of each bar on each of the figures below, is calculated with a confidence level of 95%.

In the following, before describing the similarities and the differences between the LP and LMD-LP policies, we will describe the advantage that the preemptive approach has over the non-preemptive one.

a) The Preemptive and Non-Preemptive Approach: Fig. 3-Left shows that when the system gets more loaded we have more members admitted to the system than in a non-preemptive approach. For instance we have a gain of respectively 3, 8, 11, 14, 15, 16 17, 18, 23 and 26% (for the different network loads) over the non-preemptive approaches. Moreover and as shown on Fig. 3-Right, the preemptive approaches, particularly the LMD-LP policy, have also more non-basic streams at end systems than the non-preemptive one, even though the latter approach never preempts the admitted members. For instance, for the connection lifetime of 75mn and in the case of the LMD-LP policy, we have a gain of 3% more non-basic streams.

b) The Two Preemptive Approaches: Fig. 3-Left shows that the number of admitted members is the same for the two preemptive schemes. This result is to be expected since the two schemes select the multicast streams to preempt on the feasibility phase making only bandwidth reservations for the join request and preempt at the confirmation phase when the join request has passed successfully the admission control along the path from the receiver to the in-tree node or to the Core. It is the Late Preemption Update version of the protocol. However, these two schemes present slightly different values regarding: (1) the number of non-basic streams admitted at the LANs (see Fig. 3-Right), (2) the number of preempted multicast streams (see Fig. 4-Right) and (3) the number of members degraded (see Fig. 4-Left). This can be explained by the different policies in use when selecting the multicast streams to be preempted. As described in Section 3.1, the LP policy chooses the lowest priority streams first. If there is more than one candidate, it chooses one at random. The LMD-LP policy does the opposite; it starts by choosing the multicast streams with the lowest number of members degraded and, if there is more than one candidate, chooses the one that has the lowest priority level. In the following, we will discuss in more details these differences.

b-1) The number of non-Basic streams at the LANs: As we can see from Fig. 3-Right, the total number of non-basic streams at the LANs is higher with the LMD-LP policy than with the LP policy. This advantage increases, as the network gets more loaded. For instance, for a connection lifetime of 65mn, we have about 1.5% more basic streams with the LMD-LP policy than with the LP policy. Thus considering the number of participating members when preempting multicast streams increases the numbers of non-basic streams at the LANs. However, this increase comes with a cost; information about the number of members attached to each non-basic stream must be added to the nodes.

It should be noted that this result supposes that the members of the same LANs benefit for the latest bandwidth negotiated by one of the LAN new multicast member. For instance, we have computed a gain of 10% more non-basic streams at the LANs for connection lifetimes of 55mn and 65mn (figures are not shown).

When we look at the percentage of the non-basic streams per priority, we notice that the LP policy has more non-basic streams with higher priorities than the LMD-LP policy (see Fig. 5 and Fig. 6). For instance, Fig. 5-Left shows that for the connection lifetime of 75mn, we have about 2% more non-basic streams of priority 18 in the LP policy than in the LMD-LP policy. In the contrary, the percentage of non-basic streams of lowest priority is higher in the LMD-LP policy. As shown on Fig. 7, we have about 6 percent more streams of priority 1 than in the LP policy for the connection lifetime of 75mn. This can be explained by the fact that the LMD-LP policy takes first the number of admitted members as the first criterion. Thus the non-basic streams that stay in the system are not necessary those with higher priority. On the opposite, even if the number of the non-basic streams is slightly lower in the LP policy, these streams are of higher priority because these schemes do not disturb the multicast streams of higher priority but only those that have the lowest priority level.

b-2) The number of preempted multicast streams in the network : Fig. 4-Right shows that the number of preempted multicast streams is slightly higher with the LMD-LP policy than with the LP policy. For instance, we have 3 more preempted multicast streams when the connection lifetime is 75mn. An explanation of this small difference can be found in the way these schemes select the streams to be preempted. As already mentioned, the LP policy chooses always the lowest priority streams; it happens that the priorities in use in these experiments have different amounts of bandwidth assigned to them. For instance, the lowest priorities, namely priority 1 and priority 2, are assigned, respectively, 800units and 700units of bandwidth and the intermediate priorities, namely priority 6 and priority 16, are assigned 300units. When the LP policy preempts the lowest priority multicast streams, it also preempts the larger amount of bandwidth. However, as described in Table 1, the minimum bandwidth that the multicast groups require ranges between 200units and 400units; thus when preempting the lowest priority multicast stream, it happens that not all the bandwidth is used by the new join request and the excess bandwidth is given back to the links, which will be available to the subsequent join requests. Therefore, these subsequent requests will not need to preempt existing multicast streams to get admitted since sufficient bandwidth would be available on the link. However, since the LMD-LP policy selects first the multicast streams with fewer members, it could happen that these streams are not necessary the lowest in terms of priority, which will let the request preempts multicast streams of intermediate priorities such as streams of priority 6 and priority 16 and, as already mentioned above, these multicast streams have 300units of bandwidth. In this case, because the minimum required bandwidth ranges from 200units and 400units, less bandwidth or no bandwidth at all is given back to the links. Thus, subsequent join requests would have to preempt other multicast streams, since less bandwidth would be available on the links. In fact, Fig. 8 shows effectively that the LP policy chooses almost 100% of the streams to preempt from the streams of priority 2 and 1. However, the LMD-LP policy chooses respectively 97, 99, 99, 97, 94, 87, 82, 73, 67 and 63% of these streams for the different network loads. The rest of the streams are of higher priorities.

b-3) The number of members degraded: Moreover, as depicted in Fig. 4-Left, the LP policy presents a higher number of members degraded when compared to the LMD-LP policy. For instance, it is about 42% higher for the connection lifetime of 75mn. This result is to be expected given preemption policies in use. In fact, the latter schemes selects always the multicast streams with the smallest participating members while the former schemes selects always the lowest priority multicast streams without taking into consideration the number of members degraded.

5 Conclusion

To the best of our knowledge a distributed preemptive multicast protocol has never been proposed or investigated before. We have shown that our proposed distributed preemptive bandwidth allocation multicast protocol permits to increase the number of multicast group members when there is lack of bandwidth (as compared to the non-preemptive approach) while minimizing the loss of multicast streams priorities. We have also shown how we can let more members benefit from an increase in their allocated bandwidth when allowing the multicast members within the same LAN to take advantage of the latest allocated bandwidth negotiated by one of the LAN new multicast member. Furthermore, the consideration of the number of members degraded when selecting the stream to be preempted permits to degrade fewer members than when using selection by priority. However, the first policy requires the knowledge of the number of members attached to each non-basic multicast stream within each network node, which adds more overhead to the protocol.

As future work, we intend to investigate the advantages of probing several paths in parallel. For this purpose, the PBA-M protocol would need some changes to let the receiver choose the path among the feasible paths that best satisfies its requirements.

Reference

1. T. Pusateri ,"Distance Vector Multicast Routing Protocol", IETF work in progess
2. J. Moy, "Multicast Extensions to OSPF", RFC 1584, March 1994
3. A. Ballardi, "Core Based Trees (CBT version 2) Multicast Routing", RFC 2189, Sept.1997
4. B. Fenner, M. Handley, H. Holbrook and I. Kouvelas, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", IETF work in progress
5. M. Peyravian, A. D. Kshemkalyani, "Decentralized Network Connection Preemption Algorithms", Computer Networks, June1998
6. F. Toutain, O. Huber, "A General Preemption-Based Admission Policy Using a Smart Market Approach", INFOCOM 1996
7. H. Sakate, H. Yamaguchi, K. Yasumoto, "Resource Management for Quality of Service Guarantees in Multi-party Multimedia Application", ICNP 1998

8 N. Chefaï, G. V. Bochmann, N. D. Georganas, “Performance Evaluation of the Preemptive Bandwidth Allocation Protocol”, MIPS, Nov. 2003

9. M. B. Doar, “A better model for generating test networks”, GLOBECOM 1996

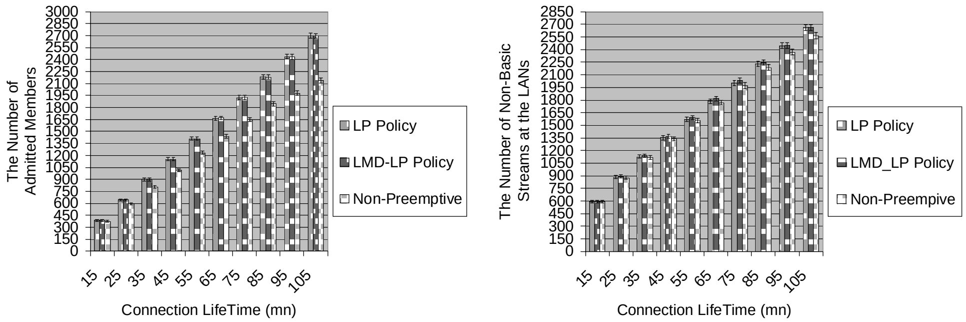

Fig. 3: left: Number of Admitted Members and right; Number of Non-Basic Streams at the LANs

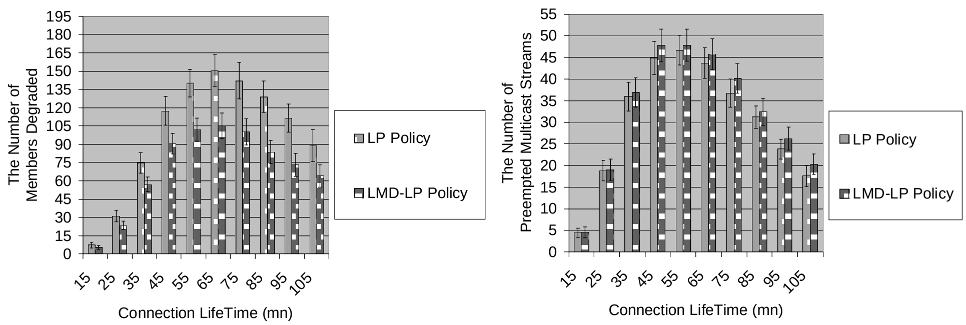

Fig. 4: Left: Number of Members Degraded and right: Number of Preempted Multicast Streams

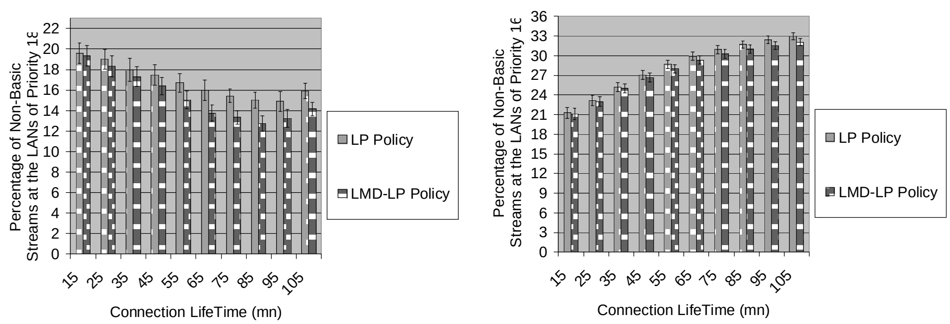

Fig. 5: Percentage of Non-Basic Streams at the LANs of Priority 18 (Left) and Priority 16 (Right)

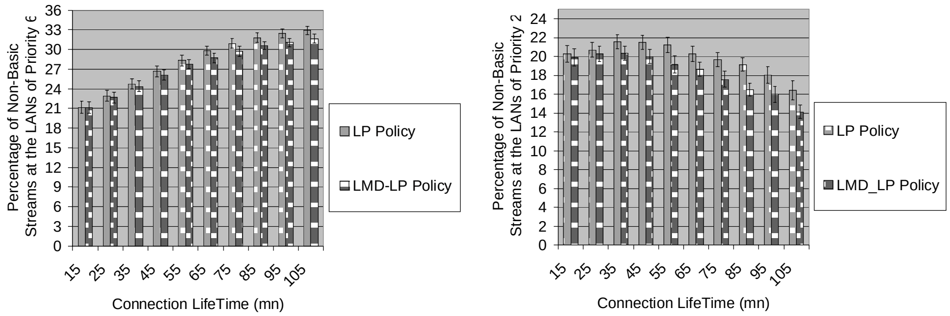

Fig. 6 Percentage of Non-Basic Streams at the LANs of Priority 6 (Left) and Priority 2 (Right)

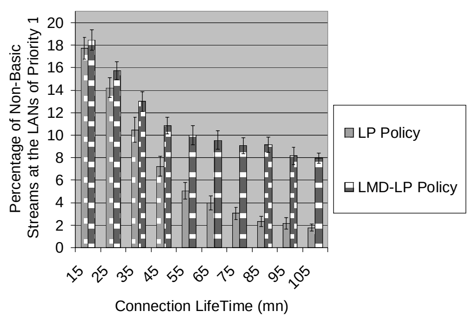

Fig. 7: Percentage of Non-Basic Streams at the LANs of Priority 1

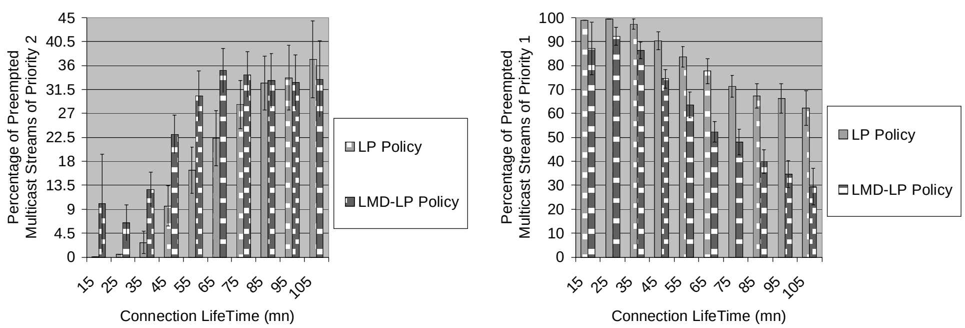

Fig. 8: Percentage of Preempted Multicast Streams of Priority 2 (Left) and Priority 1(Right)