Design and Evaluation of Router-Supported and End-to-End Multicast Receiver-Based Scoping Protocols*

Lenitra M. Clay
Mostafa H. Ammar

Networking and Telecommunications Group
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332
{lenitra,ammar}@cc.gatech.edu
(404)894-3292, Fax: (404)894-0272

May 28, 1999

Technical Report: GIT-CC-99-17

Abstract

IP multicasting allows a source to define a multicast group address and receivers can dynamically join and leave this group. Currently the propagation of multicast packets is controlled by two scoping methods: TTL scoping and administrative scoping. Both of these approaches require the source to control the scope of the multicast. In this paper we explore two receiver-based scoping protocols. Receiver-based scoping allows a receiver to place conditions on its multicast join that must be met in order to join successfully and remain a member of the multicast group. Such a scoping mechanism would be useful in environments where the receiver incurs a cost for its membership of the multicast group. In this paper we discuss receiver-based scoping and its design goals. In order to perform receiver-based scoping a receiver needs information about the shape of the multicast tree. We describe two receiver-based scoping approaches to obtaining this information: router-supported and end-to-end. The router-supported approaches requires state to be maintained in routers while the end-to-end approach uses the multicast traceroute tool and does not require state maintenance by the routers. We evaluate the performance of these approaches in terms of their accuracy, bandwidth overhead and state requirements.

*This work is supported in part by a research grant from NSF under contract number NCR96-28379 and the AT&T Labs Fellowship program*
1 Introduction

With IP multicasting, a multicast address defines a multicast group. Receivers can dynamically join and leave this group using IGMP (Internet Group Management Protocol) [1, 2, 3]. Packets are forwarded to the multicast group by a multicast routing protocol, e.g. PIM, DVMRP [4, 5, 6]. The propagation of these multicast packets can be controlled by multicast scoping. The two methods currently being used to perform multicast scoping are: TTL scoping and administrative scoping. With TTL scoping, the source specifies a TTL (Time-To-Live) to be used in the TTL field of the IP packets. The TTL field is decremented each time the packet passes through a router. Once the value reaches a defined threshold value, the packets are dropped. With administrative scoping, a range of multicast addresses are assigned to particular boundaries, and packets destined to these multicast addresses will not be forwarded outside the boundary [7].

Both of the multicast scoping techniques discussed require that the source control the scope for the multicast. We have defined a receiver-based scoping (RBS) mechanism to allow a receiver to specify certain scope conditions with their join request [8, 9]. We call this type of join a conditional or scoped join and it indicates that the receiver wishes to join the multicast group and remain a member of the group as long as the specified scope conditions are met. Motivation and specific applications making use of RBS include: multicast cost allocation, cost-controlled monitoring, and layer addition in layered video multicast. These have been described in detail in [8].

For the purposes of this paper, we define the receiver scope in terms of hop counts. More general definitions are possible but we relegate their consideration to future research. For each receiver of the multicast group we define a marginal hop count (MHC) to be the number of hops in the multicast tree that are being used exclusively for the delivery of multicast packets to this receiver. A conditional joiner will specify a join threshold, $J$, and a leave threshold, $L$, at the time of join. The join threshold indicates that the receiver wants to join the group if the receiver’s MHC is less than or equal to $J$. The leave threshold states the receiver’s willingness to remain in the group as long as the receiver’s MHC is less than or equal to $L$. Clearly, $J$ must be less than or equal to $L$. If $L$ is greater than $J$ a receiver could successfully join the group with an MHC equal to $J$ but would need to leave immediately since the leave threshold $L$ would be exceeded. The scoping mechanism could be enhanced later to include QoS measures such as loss, jitter, delay as part of the receiver scoping conditions.

In order to perform receiver-based scoping, information about the shape of the multicast tree is needed. In previous work [8, 9], we devised a receiver-based multicast scoping protocol that obtained this information by maintaining state in the routers. In this paper we develop a mechanism that allows receiver-based scoping to be performed end-to-end, without storing state in the routers. This mechanism uses the multicast traceroute tool, mtrace [10] to obtain information about the shape of the multicast tree. We also develop a more efficient version of our original router-supported RBS scheme. Finally we focus on an evaluation of the performance of the two schemes and their overhead.

This paper is organized as follows. The next section will give an overview of receiver-based scoping. Section 3 discusses the RBS design goals while section 4 details both of our approaches to receiver-based scoping. Section 5 contains an evaluation of our two approaches and section 6 gives conclusions and a discussion of future work.

2 Receiver-Based Scoping

Receiver-based scoping (RBS) will allow receivers to specify certain conditions to be met at the time of join. This type of join is called a conditional join or scoped join, and its success depends upon whether or
not the scope conditions are met. These scope conditions can include measures such as: join threshold, leave threshold, max delay, max loss. As mentioned earlier, the two scope conditions used in both of our receiver-based scoping approaches are a **join threshold**, $J$, and a **leave threshold**, $L$. These thresholds indicate that a receiver wishes to join as long as its marginal hop count (MHC) at the time of the join is less than or equal to $J$, and to remain a member of the group as long as its MHC is less than or equal to its specified $L$ value. A joiner’s MHC is defined as the number of additional hops that would be added to the multicast tree in order for the joiner to receive the multicast packets. For group members, the MHC is the number of unshared hops being used to reach the receiver. Thus a receiver’s MHC is the number of hops in the multicast tree that are being used exclusively for the delivery of multicast packets to this receiver. So a receiver’s MHC will change as the topology of the multicast tree changes.

There are two ways a receiver can leave a multicast group after a successful conditional join. The receiver can leave via:

1. A **voluntary leave** which occurs when the receiver quits the multicast application.
2. A **forced-leave** which occurs when the receiver’s leave threshold has been exceeded.

A receiver can avoid being forced to leave the group by setting $L$ to $\infty$ or a number greater than the number of hops to needed to reach the source. The following example will illustrate the workings of receiver-based scoping.

We start with Figure 1 where we have a multicast sender at host $H0$ and one receiver at host $H4$ who has specified $J = 5$ and $L = 7$ as the scope conditions. The links over which the multicast data is being transmitted are indicated with bold lines. At this time host $H4$ has an MHC of 4 since it is the only receiver. Note that we assume that the leaf network for the source will always contain the multicast data and do not count that network in our MHC calculations. Now assume that host $H1$ attempts to join.

![Sample Multicast Routing Tree](image-url)
the group by specifying $J = L = 2$. This join will not be successful because three links, Networks 2A, 3AA, and 4AA, would need to be added to the multicast tree for the data to reach $H1$. Next, $H2$ tries a join with $J = L = 4$. This join will be successful because exactly four links need to be added for data to reach $H2$. Now host $H1$ retries to join the group with the same threshold values, $J = L = 2$. This time the join is successful. Finally, host $H3$ successful joins the group with scope values $J = L = 7$. Note, that after another host, either $H1$, $H2$, or $H3$, joins the group, the MHC for host $H4$ changes from 4 to 3 because Network 1A is now shared with the other host.

Figure 2 shows the multicast routing tree after the joins described in the last paragraph. Assume that host $H4$ decides to leave the group voluntarily. The MHCs for the other hosts does not change since these hosts all still share Network 1A. Next, host $H3$ voluntarily leaves the group and again the MHCs for the other hosts are not affected. Now however the $L$ values for hosts $H1$ and $H2$ are not individually large enough to keep them in the multicast group. These hosts can remain members of the multicast group only because they both share the cost of Network 1A. Finally, host $H2$ decides to leave the group voluntarily. This causes the MHC for host $H1$ to change from two to four because networks 1A and 2A become unshared. Host $H1$ would be notified that its leave threshold, $L$, has been exceeded.

### 3 RBS Design Goals

In general, a receiver-based scoping protocol should meet the following criteria:

The scoping protocol should be independent of the multicast routing protocol that is used in the network. The protocol should only require that the network routers have the capability to forward multicast data.
The amount of overhead created by the scoping protocol should be a minimum. The network bandwidth consumed and the state required to determine the marginal hop counts should be minimal.

The protocol is timely in that a conditional join decision is made quickly. A conditional joiner should not have to wait for an extended period of time to determine if its join is successful.

The protocol should be accurate. A conditional join is successful if and only if the receiver is $J$ or less hops away from a current part of the multicast tree, and the receiver is notified to leave if and only if the closest part of the multicast tree becomes more than $L$ hops away from the receiver.

Figure 3 shows a general receiver-based scoping mechanism. Multicast data is transferred between the receiver and source as usual. A receiver and sender RBS module will handle the generation and processing of the protocol messages. The protocol may also require state to be stored at the source, receivers, and/or routers.

We have developed several metrics to evaluate the receiver-based scoping mechanisms. These metrics are:

1. *Average Message Hops* - This is the average number of links traversed by the scoping protocol messages per join. The average message hops metric is one way to measure the network overhead of the protocol.

2. *Average Stored State* - Another measure of the network overhead. This is the average amount of state required by the protocol per join.

3. *Join Decision Time* - The join decision time is the average amount of time it takes a scoped joiner to determine if the join is successful. The time is measured from scoped join initiation until the first join decision is made. This metric will be used to evaluate the timeliness of the protocol.

4. *Lost Joiner* - A lost joiner is a joiner who gives up on the join attempt. This metric is closely related to the join decision since a longer join decision time will cause more scoped joiners to give
up on the join attempt. Since there will certainly be some number of scoped joiners who will give up quickly we are more interested in what protocol changes cause an increase or decrease in lost joiners.

5. Incorrect Join - An incorrect join is a scoped join that succeeded when it should have failed. An incorrect join occurs when the MHCj perceived by the scoped joiner is lower than the actual MHCj.

6. Incorrect Leave - An incorrect leave is a forced leave that should not have occurred. The scoped joiner has perceived an MHCj that is higher than the actual one.

7. Prevented Join - A prevented join is a scoped join that failed when it should have succeeded. This occurs when the scoped joiner perceives an MHCj that is higher than the actual one.

8. Prevented Leave - A prevented leave occurs when a scoped joiner perceives an MHCj that is lower than the actual one, so the joiner remains in the group.

The last four metrics are used to measure the accuracy of the scoping protocol. We would prefer the protocol err in favor of allowing scoped joiners to enter the group or remain members of the group. Thus a high number of incorrect joins or prevented leaves is preferred to a high number of incorrect leaves or prevented joins.

4 RBS Designs

This section discusses our two receiver-based scoping designs. The first design is a router-supported approach. An initial version was described in [8, 9], here we report on a modification that requires less state in the routers and fewer protocol messages. Our second receiver-based scoping design is an end-to-end approach that does not require router support and can be implemented within the context of the Internet’s near-term architecture and capabilities.

4.1 Router-Supported RBS

We report on our updated router-supported RBS design in this section. The idea behind the router-supported design is to have the multicast source of the group periodically send out messages to notify receivers and conditional joiners of their MHC. First a separate multicast control group Gc is setup for each multicast data group Gd. (Any simple algorithm for determining the address of Gc from the address of Gd will suffice.) Then the source of the multicast data group Gd will periodically send an advertisement message to the control group Gc. The advertisement messages sent on group Gc contain the same sender scope (TTL) as the actual data message sent on group Gd. Now a receiver who wishes to conditionally join the data group Gd will first unconditionally join group Gc. The advertisement messages sent on group Gc are similar to the PATH messages defined by RSVP [11, 12] in that the they traverse a path identical to the data of interest. The advertisement messages are much smaller and less frequent than the multicast data, and so the overhead created by the control group is minimal.

The advertisement messages contain the following fields: the source address S, the data group address Gd, the marginal hop count for conditional joiners (MHCj), the marginal hop count for members of the data group (MHCi), and the parent node address (PNA) which identifies the parent node in the routing tree. In addition to the advertisement messages flowing down the tree, this design also requires the

---

1For those familiar with our initial design [8, 9], we have removed the forced-leave notification message, eliminated the physical hop count and total hop count fields from the leave threshold report state, and from the advertisement and leave threshold report messages respectively.
sending of leave threshold reports (LTR) that flow up the multicast tree towards the source. These LTR messages are needed to correctly determine the MHC for current members of the multicast group. A leave threshold report, LTR, contains the following fields: source address $S$, data group address $G_d$, control group address $G_c$, current hop count CHIC, maximum leave threshold MLT and a shared link flag. The current hop count, CHIC, indicates the height of the subtree rooted at the router, the MLT field indicates the largest $L$ value of any downstream host minus the number of hops from that host to this router, and the shared link flag SLF indicates if there a single or multiple hosts downstream. Thus a LTR message allows a router to determine the sharing level of its links and whether there is a downstream host with an $L$ that is large enough to reach it. To reduce the number of LTR messages generated the messages are aggregated at each router.

### 4.1.1 Message Processing

For each interface forwarding data for $G_c$, the router stores the following information: the source address $S$, data group address $G_d$, control group address $G_c$, parent node address PNA, the height of the routing subtree rooted at this router HEIGHT, and a shared link flag SLF. The shared-link flag indicates whether there are zero, one, or more receivers downstream. So when an advertisement message is received, for each interface the router determines if it is forwarding data for group $G_d$ over that interface and if so it sets MHC$_j$ to zero, otherwise it increments MHC$_j$. If the router is forwarding data for group $G_d$ over more than one interface MHC$_i$ is set to one since this means the link the advertisement message crossed to get to this router is being shared by at least two downstream hosts. Otherwise, if the router is maintaining leave threshold state for that interface and the SLF is set to shared, MHC$_i$ is set to zero, otherwise MHC$_i$ is incremented. Once the advertisement message has been processed, it is forwarded to group $G_c$ as normal. The router then starts a leave threshold report timer which determines the LTR message aggregation period. The LTR timer is set to $(\text{HEIGHT} + 1) \times T_{LTR}$, where $T_{LTR}$ is the maximum time for a given router to prepare and forward the aggregated LTR message.

When a host $H$ trying a conditional join receives an advertisement message, it simply compares the value of the join threshold $J$ with the MHC$_j$, and if $J \leq \text{MHC}_j$ $H$ can join $G_d$ using standard IGMP messages, otherwise the conditional join has failed. If the join is successful the host prepares a LTR message, and stores: the source address $S$, the data group address $G_d$, and both marginal hop counts (MHC$_i$ and MHC$_j$), so when another process attempts a conditional join for this group the cached information can be used. If no advertisement messages are received within a reasonable time period $H$ can assume that either there are no senders or the TTL specified by the sender is not large enough to reach it. When a member host receives an advertisement message, if $L \leq \text{MHC}_i$ the host can remain in the group so it prepares a leave threshold report and multicasts it to group $G_c$, otherwise the leave threshold has been exceeded and the host should leave the multicast group.

Finally, routers receiving LTR messages will aggregate these messages until the expiration of the leave threshold report timer. During aggregation, for each interface a count of the messages received is started and the largest MLT value of the received LTR messages is stored. Also, the largest CHIC value of the received LTR messages is stored in the router state for that interface in the HEIGHT field. When the LTR timer expires, the count of received messages is used to determine the correct value of the shared link flag. The shared link flag is set to single if one and only one child router reports a link in single state, none if no reports are received, and shared if more than one child router reports a single state or if at least one child router reports a shared state. Now, if the largest $L$ value reported by the router's children minus one is greater than or equal to zero, the LTR is forwarded upstream with this new value.

### 4.1.2 Example
Another quick example should illustrate the workings of this approach and the need for the two type of messages. Referring to Figure 4 which shows the tree after all hosts have joined $G_d$ and reported their leave thresholds. Routers $R1$-$R4$ will have the shared link flag set to shared for the interfaces over which they forward data for group $G_c$. Routers $R7$ and $R9$ are not forwarding data for $G_c$ and thus are not maintaining state. All other routers have the shared link set to single. Now say host $H2$ voluntarily leaves $G_d$ and $G_c$. The next advertisement message received by host $H2$, $H3$, and $H4$ will have values reflecting the state of the tree before $H2$ left since the state in the routers is not updated until after the leave threshold report aggregation. So router $R2$ will still believe that Network 2A is being shared and will set the MHC$_l$ field of the advertisement to zero. The reception of the advertisement message will trigger leave threshold reports by hosts $H1$, $H3$, and $H4$ since they will see an MHC$_l$ of 2, 2, and 3 respectively, which do not exceed their specified leave thresholds. Now router $R2$ will receive a single LTR message that indicates a single host downstream. $R2$ then sets its shared link flag for that interface to single. Now, when the next advertisement message is sent $H1$ will see an MHC$_l$ of 3 which now exceeds its leave threshold of 2.

4.1.3 Cost

The overhead in this approach is the state that is maintained in the routers and the bandwidth consumed by the advertisement and LTR messages. The router state overhead has two parts: the processing cost for updating the state and the storage cost for maintaining the state for the different router interfaces. Since the advertisement and LTR messages are small, the network load they generate should also be small as long as they are not sent at a high rate. However the advertisement rate will affect the accuracy and timeliness of the protocol. Thus a high advertisement can lead to increased network load but with accurate and timely joins while a low advertisement rate would mean less protocol overhead at the sacrifice of accuracy and quick scoped join decision times.
4.2 End-to-End RBS

The end-to-end design transfers the burden of state maintenance from the routers to the source and receivers of the multicast session. As in the router-supported design a separate control group is used to send information between the source and the conditional joiners and receivers. To implement end-to-end receiver-based scoping a receiver needs to be able to determine its MHC which means the receiver needs information about the topology of the tree. Specifically he needs to know the current tree topology (tree view) and where he would fit onto the tree. We use the multicast traceroute tool, mtrace [10], to gain this information. Mtrace can be used to determine the physical hop count from a receiver to a source. In the end-to-end RBS design, each member of the multicast group periodically sends (via unicast) the result of its mtrace to the multicast source. The multicast source uses this information to create a view of the multicast tree and stores it in a database. Then this multicast tree view is periodically sent out over the multicast control group. Potential joiners join the control group to receive the multicast tree view and perform an mtrace to the multicast source to determine their place in the tree. With the tree view and the multicast path from the source to themselves, a potential joiner or current member can calculate their MHC value. Note, to further reduce network load created by the control group, conditional data group members can periodically join the control group to check if their MHC has been exceeded.

There are three types messages passed between the source and the receivers: 1) mtrace information, 2) the tree view, and 3) the actual multicast data. The mtrace information is periodically sent from the receivers to the source to keep the multicast tree view up to date. The source uses this information to create the current view of the multicast tree and sends out this information over the control channel. The multicast data is not affected by the scoped join mechanism and is sent by the source as normal. Thus, this protocol only adds the mtrace information which is unicast to the multicast source, and the multicast tree view that is sent over the multicast control channel. The rest of this section will discuss some of the design details further.

4.2.1 Message Formats

When a receiver performs an mtrace to the source, the address for each hop is stored in a buffer. For hops whose IP address the mtrace is not able to determine, the address “0.0.0.0” is used. This buffer of addresses is prepended with the length of the buffer and placed in a UDP packet. So the first two bytes of the mtrace message will be the length of the data portion and the remaining bytes contains the IP address of each hop. Since mtrace finds the reverse path from the receiver to the source, the first address in the data portion will be the address of the receiver, and if the trace is successful, the last address will be the address of the multicast source.

The format of the multicast tree view message is a buffer containing node information for each node in the tree view. The node information is: the IP address for the host or router, a count of the number of children this node has, and a list of addresses corresponding to the children. The order of the nodes in the message is the order obtained by doing a breadth-first traversal of the tree view database. Note that there will be some repetition of addresses for the internal nodes. The header of the tree view message contains a sequence number, a type field and a length field specifying the length of the data portion of the message. (Currently the type field is not used). From a tree view message a receiver can create a multicast tree view.
4.2.2 The Tree View

The tree view database is a linked set of nodes with each node containing a pointer to child links and a pointer to a sibling. An example tree and database structure is shown in Figure 5. Each node in the tree view database contains the following information: 1) the IP address of the router or host, 2) a linked list of children links, 3) a pointer to a sibling of this node, and 4) a timestamp marking the last time the node information was updated. The children links are used to separate the different router interfaces and connect the node to its children nodes. The IP address of the host or router is used to determine to which child link it is added. The tree view database is initialized by creating a node for the multicast source. The node timestamp is used when the tree is update to remove old nodes.

4.2.3 Cost

The main overhead of this approach is the storage of the tree view at the source and the periodic tree view and mtrace messages. The mtrace messages should be fairly small but if these messages are sent by a large number of receivers the we could have an implosion at the source. Each receiver needs to determine the rate at which to send their mtrace information to the multicast source. This rate needs to be frequent enough so that tree is fairly up to date but not to often so as to bombard the source. We will call the time to wait between sending mtrace information to the source, $\tau_m$. It is easy to see that a high $\tau_m$ means a less accurate tree view and a low $\tau_m$ leads to a more accurate tree view. Thus to have the source send out an up to date tree view, the interval between sending an entire tree view message, $\tau_t$ would be close to $\tau_m$. If new mtrace information is not received from a receiver after a set update interval time, $\tau_u$, the source assumes that receiver is no longer a member of the group and removes his address (and any other unnecessary nodes) from the latest multicast tree view. To further reduce the multicast traffic on the control channel, changes to the multicast tree can be sent between the messages containing the entire tree view. This will be useful for conditional members to determine if their MHC exceeds their $L$ (leave threshold). This additional feature will be considered as future work for now.
5 Evaluation

In order to evaluate the performance of the router-supported and end-to-end RBS designs we created an event-driven simulation. We used data collected by the Mlisten tool [13] over several days for the MBone multicast of the Space Shuttle Mission STS-80 in November 1996. We randomly classify a receiver to be either conditional or unconditional. For each conditional joiner the \( J \) value was randomly chosen equally likely from 1 to 20. Then the \( L \) was set to be equally likely between \( J \) and \( J + 10 \). For both RBS approaches we wanted to see how the timing of messages would affect the success of the conditional joiners, as well as the number of forced leaves. The rate of messages also affects the accuracy of a receiver’s perceived MHC so the prevented join, prevented leave, incorrect join, and incorrect leave metrics will be used to measure the accuracy of our scoping approaches and the cost of improving this accuracy.

Each entry in the Mlisten data contains the time the receiver joined the group as well as the duration of their join (in seconds), so that the time the receiver left the group can be calculated. In all simulations, this leave time is used as the time the conditional joiner whose \( L \) will never be exceeded, or the unconditional member voluntarily leaves the data group after joining. In the scoped join cases where this leave time would occur before a join decision could be made, a lost joiner is counted. The remainder of this section discusses the results of our experiments for both of the receiver-based scoping approaches.

5.1 Router-Supported RBS Results

The main parameter for the router-supported simulation was the rate at which the advertisement messages are sent. The timeliness and accuracy of joins for this protocol depends solely on this rate. The accuracy of the leaving of the multicast depends on how often advertisement messages are sent but also on how often the host performs a leave check. We varied the advertisement message interval to see its effect on the join decision time, our four accuracy measures, the percentage of lost joiners and the protocol overhead. The advertisement message interval was varied from 30 seconds to 300 seconds with a 30 second step.

The percentage of incorrect joins, and prevented joins was zero for these experiments. This is because
MHC_3 is included in the advertisement messages and updated by routers as described earlier, so that its value should always be correct. Figure 6 shows the effect of the advertisement interval on the percentage of incorrect and prevented leaves. The percentage of incorrect leaves and prevented leaves was less than 10% regardless of the advertisement message rate. The percentage of incorrect leaves is the lowest when the advertisement message interval is below two minutes. The number of incorrect leaves generated by the protocol is due to the fact that the state maintained by the routers is only from the conditional joiners. If there is a conditional and an unconditional member on the same network the multicast router on that network will receive the single LTR from the conditional member and will not know that the network link is being shared by two multicast receivers. As alluded to earlier, this protocol errs more on the side of keeping joiners in the multicast group. This is because the state in the router is not updated until after the LTR aggregation period and this only affects the value of MHC_l. So that joiners see an MHC_l that is less than the actual one. This means that scoped joiners are not notified to leave the group until after an advertisement message interval.

5.2 End-to-End RBS Results

For the end-to-end approach we varied the rate at which tree view messages were sent to see the effect on our accuracy measures. The mtrace message rate and tree view update rate were fixed at once every minute and once every two minutes respectively.

The percentage of incorrect joins was less than 1% and the percentage of incorrect leaves was less than 1.5% for all experiments. Figure 7 shows the results for the percentage of prevented joins and prevented leaves. The percentage of prevented joins was less than 10% and decreased as the tree view message interval increased. The percentage of prevented leaves increases as the tree view message interval increases. This is expected since a larger message interval means that the receivers’ view of the multicast tree is not update and so they are more likely to calculate an MHC_l that is lower than their actual one.

5.3 Comparison of the Two Approaches

In order to measure the effect of a receiver-based scoping mechanism on multicast sessions we created two synthetic traces using the receiver set from the Mlisten data. The arrival time for the first receiver
was set to be within five minutes of the simulation start time and the time between arrivals was randomly chosen, equally likely from 0 to 3600 seconds. For each receiver the duration of its join was set randomly to be between 1 and 10800 seconds for the first trace, and between 3600 and 43200 seconds. So, for the short stay trace the receivers could remain in the group for a maximum of three hours and for the long stay trace, receivers could remain in the group for a maximum of twelve hours. Finally, 75% of the receiver set was chosen to re-enter the multicast group for a second time. The re-entry time for the receivers was randomly chosen to be between 1 second and 48 hours after the receiver left the group the first time. The results for these traces will be used to help compare the two approaches when the membership time is short and long.

5.3.1 Timeliness

Figure 8 shows the effect of the tree view message interval on the join decision time and the percentage of lost joiners. The graphs are the same for varying the advertisement message interval since advertisement messages allow receivers to determine if their join is successful or not. As expected as the message interval increases the join decision time increases and the percentage of lost joiners also increases. To keep the percentage of lost joiners below 50% the message interval should not exceeded 120 seconds. This would also keep the average join decision time below one minute.

5.3.2 Accuracy

Table 1 shows the maximum percentages for our accuracy measures. By maximum percentages we mean that for all experiments where the advertisement and tree message intervals were varied, the percentages received were always less than or equal to the values in the table. As noted earlier, the percentage of incorrect and prevented joins is zero for the router-supported approach. For the end-to-end approach, the percentage of incorrect joins is also low remaining at less than one percent, however, the percentage of prevented joins is higher at less than 15.5%. The percentage of incorrect leaves is lower for the end-to-end approach with less than 2% compared to the 20% for the router-supported approach. Finally, the percentage of prevented leaves is much higher for the end-to-end approach than the router-supported approach. Thus, overall the router-supported approach is more accurate.
<table>
<thead>
<tr>
<th>Accuracy Measure</th>
<th>Router-Supported</th>
<th>End-to-End</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mlisten</td>
<td>Short Stay</td>
</tr>
<tr>
<td>Incorrect Joins</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Prevented Joins</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Incorrect Leaves</td>
<td>9%</td>
<td>18%</td>
</tr>
<tr>
<td>Prevented Leaves</td>
<td>7%</td>
<td>3%</td>
</tr>
</tbody>
</table>

Table 1: Accuracy Measures for the Router-Supported and End-to-End Approach

Figure 9: Percentage Successful Scoped Joins for the Router-Supported and End-to-End Approach

5.3.3 Successful Scoped Joins

Figure 9 shows the effect of varying the advertisement and tree view message intervals on the percentage of successful joiners for the Mlisten trace. We see that the percentage of successful scoped joiners decreases as the interval between the advertisement and tree view messages increases. We can also notice that the percentage of successful joiners is higher for the router-supported approach. This is due to the fact that the router-supported approach does not generate any incorrect or prevented joins while the end-to-end approach has less than one percent incorrect joins but about ten percent prevented joins. Once an advertisement message is received the MHC$_i$ value is always correct. This is not the case for the end-to-end approach, since the accuracy of the tree view depends on both the mtrace message rate and tree view update rate.

5.3.4 Forced Leaves

Figure 10 shows the effect of varying the advertisement and tree view message intervals on the percentage of forced leaves. The router-supported approach has a higher percentage of forced leaves. However, this may be due to the fact that the percentage of successful scoped joiners is higher for the router-supported approach. The oscillations in the graphs are due to the effect the interval length has on the percentage of successful scoped joiners and the accuracy of the MHC$_i$ calculations.
5.3.5 Bandwidth Overhead

In order to evaluate the bandwidth overhead for the two approaches, we measure the average number of links over which the protocol messages traverse and call this measure average message hops. For the router-supported approach the protocol messages are the advertisement messages and leave threshold reports. The advertisement messages flow down the control group multicast while the leave threshold reports travel back up the control group multicast tree. Both of these messages are small, less than 16 bytes, since they only contain a few fields. The protocol messages for the end-to-end approach are the tree view messages and the mtrace messages. The tree view message flow down the control group to the scoped joiners, while the mtrace messages are unicast to the multicast source by each receiver. The average size of the mtrace and tree view messages were calculated assuming that four bytes would be used for the IP addresses and that the format described in section 4.2.1 was used.

Figure 11 shows the average number of message hops for the advertisement and leave threshold
<table>
<thead>
<tr>
<th>Percentage of Scoped Joiners</th>
<th>Average Mtrace Message Size</th>
<th>Average Mtrace Message Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 %</td>
<td>61.76</td>
<td>13.94</td>
</tr>
<tr>
<td>40 %</td>
<td>63.42</td>
<td>14.36</td>
</tr>
<tr>
<td>70 %</td>
<td>62.49</td>
<td>14.12</td>
</tr>
<tr>
<td>80 %</td>
<td>63.77</td>
<td>14.44</td>
</tr>
</tbody>
</table>

Table 2: Average Advertisement and Leave Threshold Report Message Hops

![Graph](image)

Figure 12: Average Tree View Message Size and Message Hops

reports. We see that the advertisement messages and leave threshold reports traverse almost the same number of hops. This is because the count of hops corresponds to the size of the control group $G_c$ multicast tree, since the $LTR$s travel the reverse path up the tree for a number of hops equal to the maximum $L$ of the downstream hosts.

Table 2 show the average mtrace message size and hops. These were the values obtained when the tree view message interval was fixed at 90 seconds, the mtrace message interval was fixed at 60 seconds and the update interval set at 120 seconds. On average the mtrace messages traverse 14 hops and their size is 64 bytes. This is fewer hops than the $LTR$ messages generated by the receivers in the router-supported approach, but the $LTR$ messages are smaller.

Figure 12 shows that the average message size of a tree view message is 1800 bytes but the messages only travel about 170 hops. Since the end-to-end approach had fewer successful scoped joins the size of the control group multicast tree is smaller than the router-supported one. Note that as the percentage of scoped joiners increases the tree view message size gets larger because the tree view reflects the entire multicast tree for the data group. So although the tree view messages traverse less hops than the advertisement messages, their size is much greater.

5.3.6 Stored State

As mentioned above, the main state overhead for the router-supported protocol is the state that the routers maintain for each interface forwarding data for the control group. State is also maintained by the host machines. As mentioned above, host may chose to cache the information obtained in advertisement to be used by other processes wishing to join the same multicast group. This cached information includes
Figure 13: Average State Size per Join for the Router-Supported Protocol for the Mlisten, Short Stay and Long Stay Traces

Figure 14: Average State Size per Join for the End-to-End Protocol for the Mlisten, Short Stay and Long Stay Traces
the IP address for the multicast source, data group and control group as well as the two MHC values. Thus the size of this stored state would be at least 14 bytes. The size of the state stored on the router interfaces would be at least 24 bytes for the four IP address, LTR timer, and the other fields. For the end-to-end protocol the state is the maintenance of a tree view at the source and the scoped receivers. A tree view link contains a pointer to a node and a pointer to the next link so its size is 8 bytes. A tree view node contains the node’s address, a timestamp, a link pointer and a pointer to its sibling node, thus the size of a node is 16 bytes. Looking at Figures 13 and 14 shows the amount of state stored per join for the two approaches for each of the traces. We see that the amount of state stored is lower for the router-supported protocol which is as expected since the receiver and every successful conditional joiner stores a tree view in the end-to-end protocol.

Table 3 shows how the average state size increases as the percent of scoped joiners increases. The table values were obtained when the advertisement and tree view message interval was fixed at 120 seconds, the mrace message interval set at 60 seconds and the tree view update interval set at 120 seconds. Again, the amount of state is much higher for the end-to-end protocol because of the maintenance of the data group tree view by the source and successful scoped joiners.

<table>
<thead>
<tr>
<th>Protocol Approach</th>
<th>Percent Scoped Joiners</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10%</td>
</tr>
<tr>
<td>Router-Supported</td>
<td>1.360</td>
</tr>
<tr>
<td>End-to-End</td>
<td>90,797</td>
</tr>
</tbody>
</table>

Table 3: Average State Sizes for the Router-Supported and End-to-End Approaches

6 Concluding Remarks

We have described receiver-based scoping and its design goals. Receiver-based scoping allows receivers to have control over their multicast membership. We discussed two approaches to receiver-based scoping: a router-supported approach and an end-to-end approach. Evaluation of these approaches indicated that the router-supported approach was more accurate with a lower percentage of incorrect joins, prevented joins and incorrect leaves. However, the percentage of incorrect leaves was higher for this approach which is not desirable since we would prefer for the protocol to err in favor of allowing receivers into the group rather than preventing their membership. In that aspect, the end-to-end protocol is more favorable. The bandwidth consumed by the RBS protocol was lower for router-supported approach, the message travel a large number of messages hops but were very small in size. The end-to-end messages traversed fewer hops but were much larger. Finally, the amount of state stored by the protocol was lower for the router-supported approach.

So the router-supported approach is more accurate and stores less state at the cost of having the routers processing protocol messages and maintaining the state. While the end-to-end approach requires more state at end systems and can be implemented with out router modifications. In fact, we have implemented the end-to-end protocol. The end-to-end approach can be improved by sending out the changes between entire tree view messages and reducing the size of the view maintained by the receivers. This would reduce the size of the protocol messages and the amount of state stored. Since receivers are only concerned with certain subtrees of the entire tree, they can maintain this view of the tree instead of an entire tree view.

We believe that receiver-based scoping is a powerful mechanism that can be used by receivers to
control their multicast membership. Future research includes improving the end-to-end approach, and using QoS measures as part of the scope conditions.

References


