

IP Multicasting for Point-to-Point Local Distribution

George Xylomenos and George C. Polyzos
 {xgeorge, polyzos}@cs.ucsd.edu

Computer Systems Laboratory
 Department of Computer Science and Engineering
 University of California, San Diego
 La Jolla, CA 92093-0114, U.S.A.

Abstract

While support for IP multicasting continues to spread enabling new applications, an increasing number of hosts connects to the worldwide Internet via low bandwidth Point-to-Point links, such as wireline or wireless telephone lines. In this paper we discuss existing proposals for local and wide area IP multicasting and their implications for Point-to-Point links, identify problems with their integration in this environment, and propose alternative special purpose mechanisms to solve these problems. The main problems are overhead due to IGMP leave latency and unnecessary continuous probing of potentially power constrained hosts. Our solution is an alternative to IGMP mechanisms based on **join/leave** messages for tracking group membership over PtP networks. After presenting the implementation requirements of our proposed and the existing mechanisms, we compare them with respect to performance, interoperability, robustness and implementation complexity, demonstrating that our **join/leave** protocol is uniformly superior.

I. INTRODUCTION

The traditional modes of communication in computer networks are *unicasting* and *broadcasting*, where data are sent to one and all hosts in a network, respectively. *Multicasting*, where data are sent to an arbitrary set of hosts referred to by a *single* identifier, can be viewed either as an intermediate case, or as a generalized mode encompassing both unicasting and broadcasting as extreme cases. Multicasting eases communication with a *logical* set of hosts that may implement distributed or replicated services or may participate in group communication applications. Broadcasting also uses a single identifier for communication with multiple entities, but is wasteful of host and network resources. Alternatively, multicasting delivery semantics may be achieved by multiple unicasts, but the sender must keep track of intended recipients, while independent unicasts cause data duplication whenever paths to separate recipients share links. As multicasting offers a useful addressing abstraction and the potential for bandwidth savings, it is a desirable service for both local and wide area networks.

This research was supported in part by a National Semiconductor Corporation Graduate Fellowship, a DDR&E Focused Research Initiative under ARO Grant No. DAAH 04-95-1-0248, and the UC MICRO program.

On the Internet, the *Internet Protocol* (IP) has been used for many years to achieve interoperability among heterogeneous technologies at the network layer, forming a single wide area internetwork that presents a common service interface to end-to-end layers. IP extensions that support multicasting have been proposed and implemented, leading to the development of the *Mbone*, a wide area multicasting testbed [8]. The Mbone has been used for audio and video conferencing, following the general trend towards integration of various modes of digital communications in a single network. For these applications multicasting is used due to its potential for bandwidth savings. On the other hand, its increased flexibility compared to broadcasting is important for resource discovery and automatic host configuration applications, which are becoming more important as host mobility is introduced in IP. The next version of IP will integrate support for multicasting from the beginning with the goal of replacing broadcast based with multicast based services.

Although IP multicasting has evolved from mechanisms available in shared medium broadcast LANs, IP itself is an internetworking protocol that achieves interoperability among different technologies. In recent years the Internet has been expanded by incorporating emerging or existing *Point-to-Point* (PtP) local distribution mechanisms, such as *Asynchronous Transfer Mode* (ATM) networks and serial lines. Many end hosts are connected to Internet service providers via telephone links, either analog or digital, wireline or wireless, effectively forming PtP link LANs with the provider's equipment as the hub of a star topology. These PtP networks are sufficiently different from shared medium LANs to warrant closer examination of the implications of transplanting the existing IP multicasting model to them. By separating *models* from *mechanisms*, we examine in this paper the problems that emerge when existing mechanisms are used in PtP networks and propose alternative optimized mechanisms that remain compatible with the IP multicasting model.

In Section II we describe the IP multicasting model, its supporting mechanisms and their origins, as well as their potential problems when used for PtP local distribution. In Section III we examine alternative approaches to solving these problems and identify a *join/leave* mechanism as the most promising one. In Section IV we describe the modifications to existing mechanisms required to implement this approach. In Section V we

evaluate our approach by comparing it to existing ones with respect to performance, interoperability, robustness and implementation complexity. In Section VI we examine quantitatively the performance of all proposals discussed in the paper under two sets of assumptions. We present our conclusions in Section VII.

II. IP MULTICASTING AND POINT-TO-POINT LOCAL DISTRIBUTION

A. IP multicasting model

The basic concept in IP multicasting is the *host group*, an arbitrary set of network hosts identified by a single, class D, IP address [3]. Hosts can *join* or *leave* a group at any time. Group members receive all datagrams addressed to the group, while *any* host can send datagrams to a group, regardless of its membership status. Multicast IP datagrams are distinguished from unicast ones by their destination address only. To achieve multicasting in a wide area network, we need a mechanism to keep track of the dynamic membership of each group and another mechanism to route the multicast datagrams from a sender to these group members without unnecessary duplication of traffic. IP multicasting implements these mechanisms in two parts: *local* mechanisms track group membership and deliver multicasts to the correct hosts within a local network, and *global* mechanisms route datagrams between local networks. Distinguishing local from global mechanisms is appropriate for IP since it is an internetworking protocol: each local network can use mechanisms appropriate to its technology, while co-operation among networks is achieved by hiding local differences behind a common interface.

In each local network, at least one host or router acts as a *multicast router*. A multicast router keeps track of local group membership and is responsible for forwarding multicasts originating from its network towards other networks, and for delivering multicasts originating elsewhere to the local network. Multicast delivery of either externally or locally originated datagrams to local receivers, as well as reception of local multicasts by the router for subsequent propagation to other networks, depend on the underlying network technology. Accordingly, the information needed within the local network regarding group membership in order to achieve local multicast delivery may vary. In contrast, co-operation among multicast routers with the purpose of delivering multicast datagrams between networks is based on a network independent interface between each local network and the outside world. The information needed in order to decide if multicasts should be delivered to target networks is whether at least one group member for a destination group is present there. A multicast router uses the information for each of its attached local networks along with information exchanged with its neighboring routers to support wide area multicasting. Irrespective of the group membership information tracked by a multicast router for local purposes, *the interface between local information and global routing is a list of groups present at each attached network*. Based on this interface, alternative algorithms can be used for routing among networks, without affecting local mechanisms. Conversely, as long as this interface is provided by the local mechanisms, they can be modified without affecting routing.

B. IP multicasting mechanisms

Since in IP multicasting global and local mechanisms are decoupled, any combination of mechanisms supporting the group membership list interface described above can be employed. A variety of global, wide area multicast routing, mechanisms exist, with the earliest and most widespread being the *Distance Vector Multicast Routing Protocol (DVMRP)*. DVMRP v.1 [6] is a variant of *Truncated Reverse Path Broadcasting* [7]. Routers construct distribution trees for each source sending to a group, so that datagrams from the source (root) are duplicated only when tree branches diverge towards destination networks (leaves). Each router identifies the first link on the shortest path from itself to the source, i.e. on the shortest *reverse* path, using a distance vector algorithm. Datagrams arriving from this link are forwarded towards downstream multicast routers, i.e. those routers that depend on the present one for multicasts from that source. A *broadcast* distribution tree is thus formed with datagrams reaching all routers. Since each router knows which groups are present in its local networks, redundant datagrams are not forwarded by *truncating* the tree. DVMRP v.3 [11] implements the improved *Reverse Path Multicasting* [7] mechanism, which *prunes* tree branches leading to networks that have no members, and *grafts* them back when members appear, thus turning the group distribution tree to a real multicasting one.

Another proposal is the *Multicast Open Shortest Path First (MOSPF)* [10] protocol, which uses a link state algorithm: routers flood their membership lists among them, so that each one has complete topological information concerning group membership. Shortest path multicast distribution trees from a source to all destinations are computed on demand as datagrams arrive. These trees are real multicast ones, but the flooding algorithm introduces considerable overhead. A radically different proposal for multicast routing is the *Core Based Trees* [2] (CBT) protocol, which employs a *single* tree for each group, shared among all sources. The tree is rooted on at least one arbitrarily chosen router, called the *core*, and extends towards all networks containing group members. It is constructed starting from leaf network routers towards the core as group members appear, thus it is a multicast tree composed of shortest reverse paths. Sending to the group is accomplished by sending towards the core; when the datagram reaches any router on the tree, it is relayed towards tree leaves. Routing is thus a two stage process which can be sub-optimal. The first stage may propagate datagrams away from their destinations until the tree is reached, thus increasing delay, and in addition, traffic tends to concentrate on the single tree rather than being spread throughout the network. Another proposal, the *Protocol Independent Multicast (PIM)* [5] protocol, employs either shared or per source trees, depending on application requirements.

Networks supporting IP multicasting may be separated by multicast unaware routers. To connect such networks, *tunnels* are used: tunnels are *virtual* links between two endpoints, that are composed of a, possibly varying, sequence of physical links. Multicasts are relayed between routers by encapsulating multicast datagrams within unicast datagrams at the sending end of the tunnel and decapsulating them at the other end. The MBone [8] is a virtual network composed of multicast

aware networks bridged by such tunnels. Multicast routers may choose to forward through the tunnels only datagrams that have *Time to Live* (TTL) values above a threshold, to limit multicast propagation.

Routing mechanisms are concerned with multicast propagation on a mesh of PtP links, the typical topology for an inter-network backbone. The end points of this mesh are local networks served by their local multicast routers. Global routing mechanisms are able to function correctly as long as a group membership list for each local network is maintained and made available by local routers. This interface enables global routing to deal with networks as single entities, thus reducing routing calculations and tables to manageable magnitudes, while hiding local details from the outside world. Multicast routing protocols that employ multicast sources in addition to destinations (DVMRP and MOSPF), also view sources as source *networks* rather than as source *hosts*.

In contrast to global mechanisms, only a single set of local mechanisms exists. These local multicasting and group management mechanisms were based on shared medium broadcast networks such as Ethernet, and this is evident on some of the design decisions made. Delivery is straightforward on these LANs, as all hosts can listen to all datagrams and select the correct ones. If a LAN supports multicasting as a native service, class D IP addresses may be mapped to LAN multicast addresses to filter datagrams in hardware rather than in software. Multicasts with local scope do not require any intervention by the multicast router, while externally originated multicasts are delivered to the LAN by the router. The router also monitors all multicasts so that it can forward to the outside world those for which receivers exist elsewhere. Both unicasts and multicasts are physically broadcast on these LANs, so the only issue for the router when delivering externally originated multicasts is whether at least one member for the destination group exists in the network. The router only has to keep internally a local group membership list, which coincides exactly with the information on which global multicast routing is based.

Both versions of the *Internet Group Management Protocol* (IGMP) provide a mechanism for group management well suited to broadcast LANs, since only group presence or absence is tracked for each group. In IGMP v.1 [4] the multicast router periodically sends a *query* message to a multicast address to which all local receivers listen to. Each host, on reception of the query, schedules a *reply*, or report, to be sent after a random delay, for each group in which it participates. Replies are sent to the address for the group being reported, so that the first reply will be heard by all group members and suppress their own transmissions. The router monitors all multicast addresses, so that it can update its membership list after receiving each reply. If no reply is received for a previously present group for a number of queries, the group is assumed absent. In steady state, in each query interval the router sends one query and receives one reply for each present group. When a host joins a group it sends a number of unsolicited reports to reduce *join latency* for the case where it is the first local member of the group. No explicit action is required when a host leaves a group, as group presence times out when appropriate.

In IGMP v.2 [9] a host must send a *leave* message when aban-

doning a group, but only if it was the *last* host to send a report for that group. However, since this last report may have suppressed other reports, the router must explicitly probe for group members by sending a *group specific* query to trigger membership reports for the group in question. It can only assume the group absent if no reports arrive after a number of queries. All IGMP v.2 queries include a time interval within which replies must be sent: general queries may use a long interval to avoid concentrating reports for all groups, while group specific queries may use a short interval to speed up group status detection. The time between the last host leaving a group and the router stopping multicast delivery for that group is called the *leave latency*.

C. Point-to-Point vs. broadcast LANs

The query/reply mechanism has two peculiarities: first, it periodically repeats the same cycle of queries and replies, and second, joining and leaving a group normally takes effect after one or more cycles are completed. Join latency is reduced by the, possibly redundant, unsolicited reports. In contrast, after all hosts leave a group, it will not be assumed absent for a number of cycles. During this period redundant multicasts will be delivered to the network. Even if *every* host did multicast a leave message when abandoning a group, the router would still need to decide whether other members of the group still exist locally, by further probing for group status. IGMP v.2 requires members to send leave messages only when they were the last to send a membership report: in all other cases, at least one other group member existed locally a while ago. The reason that leave messages are not enough to determine group status is that routers keep a list of present groups rather than a list of member hosts for each group. The simple group list is sufficient for making delivery decisions in broadcast LANs, but then group absence can only be determined by the absence of membership reports and not by leave messages alone. In a LAN with plenty of bandwidth, this approach trades off bandwidth overhead due to query/reply cycles and leave latency, for simplified group management at the router. The duration of the query interval is a compromise between management overhead and delivery overhead after a group disappears, hence the decision in IGMP v.2 to support distinct intervals for each query at the price of complicating somewhat the router protocol. Although the periodic queries add robustness to group management, they also add complexity to hosts, which have to continuously set up timers for replies and suppress or send periodic reports.

When end hosts are connected to a router via Point-to-Point links, some of these assumptions do not hold. While bandwidth may be plentiful in ATM LANs, hosts connected to the Internet via telephone links are currently quite bandwidth constrained, and will continue to be so if these links become wireless. Wireless mobile hosts have additional battery power and processing constraints that urge for simplified local mechanisms. Thus, local mechanisms that reduce the leave latency and periodic group management overhead of existing mechanisms would be preferable. IGMP v.2 can reduce overhead compared to IGMP v.1 by using different intervals for general and group specific queries, but still some query/reply cycles are needed to detect group absence, thus management and leave latency

overhead cannot be completely eliminated. In addition, the periodic queries disturb mobile hosts that could otherwise employ *sleep mode* to conserve battery power. Fortunately, we can take advantage of the fact that only one host is at the end of a PtP link. Group presence on such a link is equivalent to the host being a member of this group, solving the problem of distinguishing between individual and collective group membership. This means that both join and leave messages can be interpreted unambiguously as indications of group presence and absence, respectively. Therefore, periodic queries are not needed to detect group absence. To take advantage of PtP links, the router needs a group presence list per link, the same information kept by existing mechanisms. We will show how these lists may be simply aggregated to present all local PtP links as one local network for routing purposes. Multicast delivery from the router to end hosts is trivial, as is reception by the router of local multicasts that need to be forwarded to the external world. Local multicasts are *not* automatically received by other local receivers; they must be relayed to them via their PtP links by the router.

III. SUPPORTING POINT-TO-POINT NETWORKS

A. Extending local mechanisms

Existing implementations of the IP multicasting extensions treat each physical interface of a multicast router as an entry point to a distinct network, including local PtP links. Note the distinction between backbone and local PtP links; global routing mechanisms are used in the former, while local delivery and group management mechanisms are used in the latter. The router executes an instance of the IGMP protocol for each local PtP link, treating it exactly the same as a broadcast LAN. In DVMRP v.1 each multicast reaches all routers, so the only local decision is whether to deliver it locally. This decision is based on the individual membership lists. If however the routing protocol limits multicast propagation to only those areas containing group members, information must be summarized. MOSPF routers have complete knowledge about membership in all networks, so summarizing this information across all local links is the only way to avoid routing explosion. DVMRP v.3 prunes and grafts distribution trees based on aggregate membership lists at each router [11]. It is advantageous for last hop routers to keep aggregate group membership lists spanning all their local PtP links, since routers serving telephone links can be expected to have numerous PtP interfaces. IGMP processing and group membership aggregation across these links places a considerable overhead to the router, but per link group membership information is necessary to avoid delivering multicasts that have only one local receiver over *all* links.

Routers can periodically update their aggregate group membership list by combining individual lists across all PtP links. The problem is that existing IGMP mechanisms are not very well suited to PtP links as discussed earlier. IGMP v.2 leave messages attempt to reduce leave latency but they are not authoritative indications of group absence in broadcast LANs, so the router must send group specific queries to determine group status. To guarantee robustness, multiple such queries are required. In a broadcast LAN, if the group is not absent, unnecessary messages are exchanged, but if it is absent leave latency

is reduced. To compensate for this overhead, general query intervals may be increased, but not too much, as group absence *cannot* be noticed via this mechanism in some cases. For example, when the last host to send a membership report crashes and all other hosts leave the group before the next query, no leave messages will appear, making the general queries the only way to determine group absence. In contrast, leave messages over PtP links are authoritative; since only one host is connected to the link, they are definite proof that the group is absent. Since only one host uses the link, it will always be the one that sent the last reply for the group, thus IGMP v.2 reduces leave latency without requiring any changes for PtP links. Even though this solves the problem of detecting group absence, the protocol still goes through periodical general and group specific query/reply cycles that are largely redundant, as we will show.

An alternative treatment for PtP links would be to view them as backbone PtP links, by turning each end host into a multicast router and extending the area covered by global routing. Even though multicast routing works well over a mesh of PtP links, moving all end hosts into the backbone is impractical, due to scalability problems. DVMRP v.1 broadcasts each multicast to all routers, so multicasting would effectively turn into broadcasting all the way to the end hosts. DVMRP v.3 starts in a broadcasting mode until the distribution tree is pruned, and periodically reverts to broadcasts, so it would have similar problems at a lesser scale. MOSPF would face routing table explosion, since each router would have to keep information on all end hosts. Only core based protocols (such as CBT or PIM) could work with this approach, as they explicitly construct distribution trees without ever employing broadcasts and routers make forwarding decisions based on local information only. Regardless of the routing protocol used, extending global mechanisms to local PtP links violates the IP distinction between internetworking and local networking. Local mechanisms should be used to address issues at a network characterized by common physical attributes so that optimizations can be taken advantage of, rather than forcing global internetwork mechanisms on dissimilar networks. Thus, extending local mechanisms to best support PtP links is the only viable approach.

B. The Join/Leave mechanism

As already discussed, on both PtP and broadcast links one group membership report is enough to indicate membership presence on that link. Furthermore, for PtP links one leave group message indicates group absence from the link. Thus, periodic group membership queries and replies can be eliminated completely from a PtP link by employing simple *join* and *leave* messages to unambiguously determine the membership status of the end host. The leave message obviates the need to periodically reconfirm membership, while the (unsolicited) join message immediately indicates group presence. General group queries, which serve as synchronization points for the randomized timers in broadcast networks, as well as their corresponding replies, are redundant. The group specific queries of IGMP v.2 are also redundant on a PtP link. Since the periodic queries and the group presence timeouts enhance robustness with IP, which only offers unreliable delivery, join/leave messages should be *confirmed*, that is, they should be explicitly

acknowledged by the router and retransmitted if an acknowledgment does not arrive before a timer expires. The join/leave mechanism is similar to the proposal for tracking multicast group membership in the IP multicast over ATM case [1], although there connectivity between the *multicast server* and end hosts is over virtual circuits rather than physical PtP links, and the delivery mechanisms differ. The join/leave mechanism is based on the observation that in PtP links the periodic queries result in end hosts sending repeatedly their *complete* state. By only transmitting the state *difference*, we end up with the join/leave messages, since it is these actions only that result in state changes.

Because only one end host is attached to the PtP link, there is no need to synchronize random timers to avoid multiple reports, in turn making periodic queries redundant. This mechanism eliminates join and leave latencies between the local router and end hosts since there are no waiting periods for group timeouts, potentially eliminating overhead from redundant multicast transmissions. Similarly, periodic queries and replies are also eliminated, reducing group management overhead, but not always minimizing it (see Section V). The join/leave mechanism is end host initiated, with changes to membership state being transmitted as they occur rather than periodically. As a result, mobile wireless hosts would have to wake up only when absolutely necessary, to process either join/leave messages that they initiate, or multicasts that they want to receive. There is no need for periodic per group timer management to handle replies to queries, although per join/leave event timers are still required for retransmissions of the (acknowledged) join/leave messages. The multicast router has always updated information on local membership by modifying its tables whenever join and leave messages arrive rather than periodically. It can also immediately aggregate group information across all local PtP links, as we will see below, in order to present to other routers the image of a single local network, by using a single group list.

IV. IMPLEMENTING THE JOIN/LEAVE MECHANISM

In this section we describe the implementation of the join/leave local multicasting mechanisms in order to facilitate comparisons with existing mechanisms (see Section V). To clarify both differences and similarities between the join/leave and query/reply mechanisms, we describe a proposed implementation as a set of modifications to existing query/reply ones. We assume that the link layer notifies the network layer multicast module when the state of the PtP link changes either due to the peer rebooting or due to a *cell handoff* in a cellular network, where the end host changes its point of attachment to the network. This may be achieved by an upcall to the network layer whenever the link layer re-establishes its PtP connection.

Multicast transmission between end hosts and the router uses the same primitives as normal unicasts, so the multicast router automatically receives all multicasts that may have to be forwarded to other networks. The router maintains per link group membership lists in order to deliver both local and external multicasts only to group members, treating each PtP link as a separate interface and using the same data structures employed by existing mechanisms.

From the viewpoint of end hosts, group management is considerably simplified. Processes join and leave multicast groups, and the host keeps a reference count for membership in each group so that it can notify the multicast module when a group is initially joined or finally abandoned, similar to existing mechanisms. These notifications however lead directly to acknowledged join and leave IGMP messages; the host must retransmit these messages after a timeout if no acknowledgment arrives. Acknowledgments and retransmissions are used as a substitute to the automatic recovery from lost messages provided by the periodic queries and replies in existing mechanisms. The timer bookkeeping overhead is reduced since instead of one timer per group that has to be set and reset continuously, only one timer per join/leave operation is required. The host does not need to remember if it sent the (nonexistent) last membership report, so eliminating the local leave latency using leave group messages is achieved considerably easier. Since with join/leave group management consists of isolated rather than continuous periodic actions, in periods of network inactivity battery powered hosts can employ sleep mode.

From the viewpoint of the multicast router, query timers, general queries and group specific queries, are completely eliminated. In the join/leave approach, the router *never* initiates any messages, its only responsibility being to acknowledge join/leave messages from end hosts. Queries and their associated timers are redundant because in PtP links both join and leave messages are authoritative. Updating per link membership lists consists of adding the group when a join arrives and deleting it when a leave arrives. For the aggregate group membership list, join actions cause adding elements to the list, if needed, but leave actions in one link do not necessarily mean that there are no members in other links. One approach for simple aggregate list management is to use reference counts for each membership entry, updating them based on join and leave actions for each per link list. Reference counts, if used, should be updated only the *first* time that a join/leave is received, and not for its retransmissions, by ignoring, for reference counting purposes only, join/leave messages that do not modify the per link list. Using these mechanisms, both per list and aggregate group membership lists are immediately updated, eliminating leave latency incurred overhead.

The proposed join and leave messages can employ the same format as existing ones, with new type numbers. IGMP messages already contain a group address field which is the only data that join/leave messages need. For fixed networks, router and end hosts could be expected to use the same IGMP variant, so existing messages could be used (*report* and *leave*) without causing interoperability problems. For mobile wireless hosts however, it is not guaranteed that all systems (routers and hosts) would agree on IGMP operations, so an initial negotiation phase is needed. As a minimum, all systems should implement, and be able to fall back to, the standard query/reply mechanism. Since in the join/leave approach routers do not initiate any transactions, the mobile host could initiate operations by sending a join message for an arbitrary group: if the router acknowledges the message, the join/leave mechanism is supported, and the host can send a leave for the group, else, if after a number of retransmissions there is no response, the mobile

should use the standard mechanisms. The router should start in standard mode to cater for hosts using standard mechanisms, and switch to join/leave mode for a *specific* PtP link only after receiving a join.

When an end host reboots, the router should empty its link membership list and the host will re-establish this state as its applications are restarted, while when the router reboots, the host should retransmit a join for each of the groups that it participates in to re-establish its link membership list. For mobile hosts, when a handoff is detected, the host acts as if the router on the other end is rebooted in order to re-establish its link state, while the old router serving the mobile empties its link membership list, as if the end host had rebooted. These events are communicated to the multicast module by the link layer whenever the link state is changed from connected to disconnected and back.

V. EVALUATION OF THE JOIN/LEAVE MECHANISM

As discussed in Section III, the join/leave mechanism is a simplified form of the query/reply mechanism that takes advantage of PtP link properties to optimize operations. In this section we will present a more detailed comparison between the two models with respect to performance, interoperability, robustness, and implementation complexity. The generic performance analysis presented in this section is taken further in Section VI with two examples showing the tradeoffs involved under specific assumptions on protocol parameters and application behavior.

Concerning performance, in IGMP v.1 the parameter that determines the balance between group management and leave latency incurred transmission overhead is the query interval. It is impossible to eliminate both types of overhead with a single query interval value and it is also impossible to make reasonable tradeoffs based on individual group behavior as periodic queries trigger reports for *all* groups. In IGMP v.2 leave latency is reduced due to the leave messages, but subsequent group specific queries are needed to guarantee that a group is absent (ignoring the case discussed in Section III-A where group absence is determined by general queries) and to guarantee robustness multiple queries are needed. In PtP links leave messages do not cause unnecessary overhead as they are always authoritative, even though the protocol does not know it. By adjusting the group specific query interval the leave latency can be reduced. Thus, with respect to transmission overhead the zero local leave latency of the join/leave mechanism is superior to the query/reply mechanisms. When messages are lost, all mechanisms recover after additional messages, but while join/leave timers can be tight as they only have to account for round trip and processing delay, query intervals should be large enough to make randomization effective in suppressing duplicate reports, even though they are impossible in PtP links. Thus, even when messages are lost, the join/leave mechanism remains superior. The added accuracy in tracking group membership may also reduce transmissions among routers when a non broadcast based routing protocol is employed.

Group management overhead, as measured by IGMP messages, is more complicated. Membership in a group in the query/reply model requires one report per interval, but leaving

any number of groups in the same interval leads to *all* groups being detected as absent after some queries. Assuming messages are not lost and a large number of groups (so that the cost of queries can be ignored) the join/leave mechanism is superior to the IGMP v.1 when *membership to any group lasts for at least four consecutive intervals*. The justification is that membership to a group in the join/leave model for *any* period of time costs four messages: a join, a leave, and their acknowledgments, which is the cost of four periodic replies, as a separate report is sent for each group. As the number of groups that a host participates in at the same time is reduced, the join/leave mechanism becomes more favorable since the group queries are amortized among fewer reports. The cost for leaving a group in IGMP v.1 is small as its detection is automatic and it does not depend on the number of groups present. In the presence of lost messages the costs are harder to compare as they depend on the exact messages lost. To reduce join latency, IGMP requires hosts to immediately send a number of reports upon joining a new group. In IGMP v.2 we also have leave messages sent on PtP links with only one receiver. Routers respond to each such message with a number of group specific queries. When these mechanisms are taken into account, even when each message is only sent once, the four mandatory messages of the join/leave mechanism are balanced in IGMP v.2 *after only one query interval*: one initial report, one report after the first query, one leave message, and one group specific query. Thus, protocol message overhead for the join/leave model is usually lower than any version of IGMP, while leave latency transmission overhead is always lower. For battery powered mobile wireless hosts, in addition to the reduced transmission and protocol overhead, we should also take into account the advantages arising from the receiver initiated nature of the join/leave mechanism. As the join/leave mechanism is simpler in operation than existing mechanisms, due to fewer timers, a simpler protocol state machine and easier aggregation of membership state, its processing requirements are lower.

Interoperability among different versions of IGMP can be achieved using the procedures described in Section IV. Routers using join/leave mechanisms locally can participate in multicast routing based on the present groups list, which remains the interface between local group membership information and multicast routing. An optimization over existing IGMP versions is the automatic aggregation of all PtP link information as if a single LAN interface was present. Deployment of the join/leave mechanism should be direct for fixed networks, while for mobile wireless networks both variants should be implemented in a common module for backward compatibility. Since all mechanisms share the same data structures, dual implementations should not be considerably larger than existing ones.

Robustness in join/leave is achieved by acknowledged messages, while with query/reply recovery is provided by the periodic queries. An advantage of the join/leave mechanism is that the explicit acknowledgments to joins are a positive indication to end hosts that they are members of the group, while in the query/reply model a host cannot distinguish between group inactivity and lost reports. Conversely, absence of replies to periodic queries may mean either group absence or lost reports, while in the join/leave case leave actions are explicit and au-

thoritative, and the protocol knows it.

A sample implementation has been described in Section IV. Since most structures and operations are based on existing implementations, join/leave mechanisms are easy to implement, while shared data structures mean that dual implementations will be compact. The only significant addition is the initial handshaking used to discover the version of IGMP supported by the peer. In most aspects, including timer management, mapping process actions to messages, and mapping messages to state updates, the join/leave mechanism either simplifies existing mechanisms, as is the case with timers, or eliminates them, as is the case with group specific queries. A pure join/leave implementation could be developed by deleting code from existing implementations and adding join/leave handling and retransmission timers, resulting not only in easier maintenance, but also in simpler protocol operation, which in turn affects performance. Even when multiple protocol variants are implemented for compatibility, only join/leave will be executed when supported by both peers.

VI. QUANTITATIVE PERFORMANCE EXAMPLES

To supplement the analysis of Section V, we examine the exact overhead involved when the two IGMP versions and our join/leave mechanism are used in two scenarios. Note that on a PtP link, when an IGMP v.2 host leaves a group it always sends leave messages. We use the timer, query interval and repetition values specified in the draft specification of IGMP v.2 for both IGMP versions [9]. We assume that IGMP messages, regardless of mechanism, are 256 bits long on the link, which includes 20 bytes for the IP header, 8 bytes for the IGMP payload, and 4 bytes for link layer overhead. We also assume that no messages are lost. Under these assumptions, the overhead can be analytically computed.

The first scenario is a video conference which uses a total of 128 Kbps for one audio and two video streams, each using a separate group. This is the bandwidth of two ISDN B channels, although in practice we would use the 128 Kbps to accommodate both data and overhead rather than only data. In this example group membership lasts as long as the host participates in the conference, so the host joins and leaves each group exactly once. We compute protocol and leave latency overhead as a percentage of data rate, as membership duration varies from 0.1 to 100 minutes. IGMP sends two unsolicited reports when joining a group and queries are sent every 125 seconds. In IGMP v.1 two query intervals plus 10 seconds are needed to notice group absence: assuming that the host leaves the group halfway between two queries, it continues to receive data for about 200 seconds. In IGMP v.2 when a host leaves a group it sends one leave message and the router replies with two group specific queries, so datagrams for the group stop being forwarded after only 3 seconds. Join/leave only sends one acknowledged join and one acknowledged leave message per group, regardless of membership duration. Figure 1 shows total protocol and transmission overhead as a percentage of data rate for the videoconference example. We ignored the impact of query messages, which is negligible on this busy high speed link. Scales are logarithmic, to show the differences in overhead despite the wide

range of numbers involved and to clarify the effects of membership duration. IGMP v.2 and join/leave are close, although for short membership intervals IGMP v.2 has significant overhead, while join/leave drops immediately below 1%. IGMP v.1 causes considerable overhead, with very high values for short conferences and significant values for longer ones. Overhead in this link is mainly due to leave latency, as the high data rates dwarf the group management overhead. IGMP v.1 is slow in detecting leave events, so it performs very poorly until the conference duration is long enough to amortize the 200 seconds of wasted transmissions. IGMP v.2 is much faster in detecting group absence due to leave messages and their short timeout interval of 3 seconds. Over longer intervals, IGMP v.2 is only marginally worse than join/leave due to the periodic reports. For join/leave the cost is constant and small, so overhead percentage quickly approaches zero. Thus, when membership duration is extremely short, join/leave has a definite advantage over IGMP v.2, which vanishes for longer membership durations. Similar results hold when the aggregate bandwidth is varied from 16 Kbps to 1 Mbps: join/leave performance is slightly affected since its fixed cost is amortized slower or faster.

In the second scenario, the host participates in an arbitrary set of groups that emit 64 bps data streams, e.g., an 8 byte message every 4 seconds, including IP and link layer overhead. This could be a wireless host receiving low data rate streams, such as messaging services. Since we assume the same data rate for each group, we analyze them separately. The formulation is as in the first scenario, the only difference being that each group has a 64 bps data rate and we need the overhead as a percentage of that rate. Membership duration varies independently for each group. We ignore the query overhead as it is again much less than the data rate, although periodic queries prohibit wireless hosts from entering sleep mode. The results are shown in Figure 2, again using logarithmic scales. Here the periodic group management overhead is much more important than before, due to the low data rates. For the same reason, leave latency has a smaller effect. As membership duration increases, join/leave amortizes its fixed total cost with the overhead eventually approaching zero, while for short membership periods the overhead is significant, due to the low data rate. Similarly, both IGMP versions start with a significant overhead, and amortize it as membership duration increases. However, periodic IGMP costs, which are the same for both versions, make both curves eventually approach the same asymptote, away from join/leave. Thus, even though IGMP v.2 and join/leave start close, as membership duration increases join/leave overhead approaches zero while IGMP v.2 overhead never falls below a certain level, making join/leave a superior choice. Similar results hold for data rates between 8 bps and 512 bps, i.e. the relative positions of the curves do not change.

VII. CONCLUSION

We have presented some problems that existing implementations of IP multicasting extensions face when deployed in networks with Point-to-Point (PtP) local links, such as wireline or wireless telephone lines, and identified as their cause the orientation of local IP multicasting mechanisms towards broadcast LANs. After considering the alternatives, we proposed

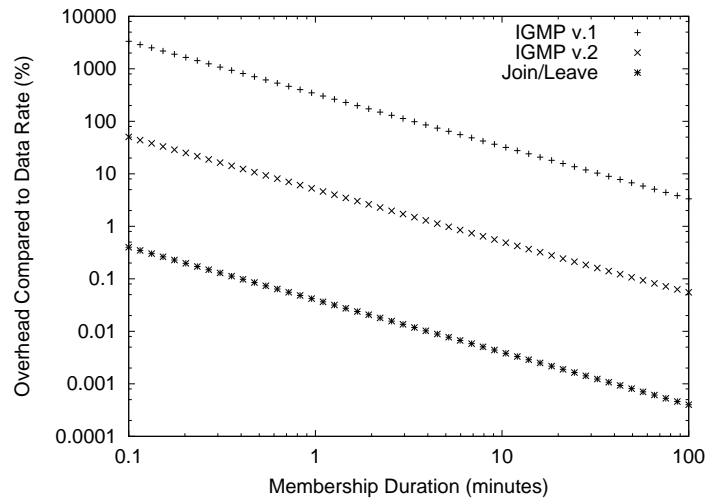


Fig. 1. Overhead as a percentage of data rate for a 128 Kbps videoconference as membership duration varies.

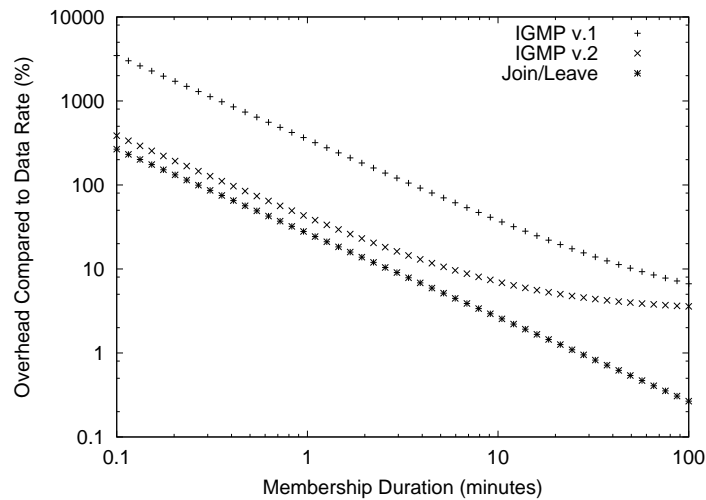


Fig. 2. Overhead as a percentage of data rate for a 64 bps group as membership duration varies.

a *join/leave* mechanism for tracking group membership over PtP networks. We then presented an implementation outline for this mechanism by modifying existing mechanisms, showing that the join/leave mechanism considerably simplifies operations. We also compared the join/leave approach with the standard query/reply mechanism with respect to protocol performance, as measured by transmission overhead, as well as interoperability with existing implementations, robustness and implementation complexity. Based on this comparison we conclude that the join/leave mechanism is superior to existing ones for any type of PtP link, while being easy to implement and deploy. When bandwidth or processing power are limited, as in battery powered mobiles using cellular telephone links, our proposal could lead to significant improvements.

REFERENCES

- [1] G.J. Armitage. Multicast and multiprotocol support for ATM based internets. *Computer Communications Review*, 25(2):34–46, April 1995.
- [2] A. Ballardie, J. Crowcroft, and P. Francis. Core based trees (CBT) — An architecture for scalable inter-domain multicast routing. In *Proceedings of the ACM SIGCOMM '93*, pages 85–95, October 1993.
- [3] D.R. Cheriton and S.E. Deering. Host groups: A multicast extension for datagram internetworks. In *Proceedings of the 9th Data Communications Symposium*, pages 172–179, September 1985.
- [4] S. Deering. Host extensions for IP multicasting. Internet Request For Comments, August 1989. RFC 1112.
- [5] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu, and L. Wei. An architecture for wide-area multicast routing. In *Proceedings of the ACM SIGCOMM '94*, pages 126–135, October 1994.
- [6] S. Deering, C. Partridge, and D. Waitzman. Distance vector multicast routing protocol. Internet Request For Comments, November 1988. RFC 1075.
- [7] S.E. Deering and D.R. Cheriton. Multicast routing in internetworks and extended LANs. *ACM Transactions on Computer Systems*, 8(2):85–110, May 1990.
- [8] H. Eriksson. MBONE: The multicast backbone. *Communications of the ACM*, 37(8):54–60, August 1994.
- [9] W. Fenner. Internet group management protocol, version 2. Internet Draft (Work in Progress), October 1996.
- [10] J. Moy. Multicast routing extensions for OSPF. *Communications of the ACM*, 37(8):61–66, August 1994.
- [11] T. Pusateri. Distance vector multicast routing protocol. Internet Draft (Work in Progress), September 1996.