IP Multicast Group Management 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, USA

***Abstract*—We examine the applicability of existing IP multicast *mechanisms* for Point-to-Point links such as wired and wireless telephone lines. We identify problems such as overhead due to IGMP *leave latency* and unnecessary probing of hosts, both important issues for power constrained mobile hosts and low bandwidth wireless links. We propose alternative mechanisms that preserve the IP multicasting *model* but employ *join/leave* messages to track group membership. We describe the implementation requirements of our mechanisms and compare them to existing ones with respect to performance, mobile power efficiency, interoperability, robustness and implementation complexity, demonstrating that the *join/leave* approach is uniformly superior for this environment.**

I. Introduction

The traditional modes of communication in computer networks are *unicast* and *broadcast*, where messages are sent to one and all hosts in a network, respectively. *Multicast*, where messages 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 unicast and broadcast as special cases. Multicast provides the ability to address *logical* sets of hosts as single entities, thus easing the implementation of distributed and replicated services. Compared to broadcast, multicast economizes on host and network resources by only delivering data to required recipients rather than to all hosts. If multiple unicasts were used to achieve multicast delivery semantics, the sender would need to track a potentially huge set of intended recipients, while duplicate data transmissions would occur wherever paths to separate recipients shared network links.

On the Internet, the *Internet Protocol* (IP) joins local area networks that employ heterogeneous technologies into a single wide area internetwork, providing a common network layer service interface to end-to-end layers. IP originally only supported unicast and broadcast, which are either natively supported or easily implemented on top of each other on any type of network link. Although multicast is harder to implement, IP extensions that support it have been developed, leading to the deployment of the *Mbone*, a wide area testbed [6] used for experimentation with multicast enabled applications. Audio and video conferencing and distribution applications were among the early adopters of the MBone, due to multicast's potential for bandwidth savings for multipoint communications. On the other hand, the single logical address concept is useful for resource discovery and automatic host configuration, which are becoming more important as host mobility is introduced in IP [9]. All servers that provide the same type of service can share a well-known multicast address so that they can be easily located. Finally, group communication based applications, such as dynamic distributed routing, can substitute multicast for broadcast or multiple unicasts for efficiency.

This research was supported in part by a National Semiconductor Corporation Graduate Fellowship and the UC MICRO program.

The IP multicast extensions basically consist of the *native* multicast mechanisms available in broadcast based LANs, plus additional mechanisms for wide area multicast distribution over backbone *Point-to-Point* (PtP) links. Recently however, the use of *local* PtP links has been increasing steadily on the Internet. Many hosts are connected to their *Internet Service Provider* (ISP) via a telephone link, that may be either analog or digital, wireline or wireless. The *Asynchronous Transfer Mode* (ATM) LANs are an example of a PtP link based LAN. Finally, the mobility extensions for IP [9] support local multicast distribution using *virtual* PtP links, or *tunnels*. All these local PtP links, whether physical or virtual, are sufficiently different from shared medium LANs to warrant closer examination of the implications of transplanting the existing IP multicast mechanisms to them.

In this paper, we separate *models* from *mechanisms* in order to identify the problems that emerge when existing mechanisms are used with local PtP links, and propose alternative optimized mechanisms that remain compatible with the IP multicast model. The main problems that we try to solve are overhead due to IGMP *leave latency* and unnecessary continuous probing of hosts, issues especially important for mobile hosts and wireless links. In Section II we describe the IP multicast model and its supporting mechanisms, as well as their potential problems when used with PtP local distribution. In Section III we examine alternative approaches to these problems and identify a *join/leave* mechanism as the most promising one. In Section IV we show how our mechanisms can be implemented by modifying existing multicast implementations. In Section V we evaluate the proposed mechanisms by first examining the performance of existing and proposed mechanisms, and then discussing how they compare with respect to mobile power efficiency, interoperability, robustness and implementation complexity. We present our conclusions in Section VI.

II. IP Multicast and Point-to-Point Local Distribution

A. IP Multicast Model

The IP multicast model is based on the *host group*, an arbitrary set of hosts identified by a single, class D, IP address [5]. Group membership is dynamic, i.e. hosts can *join* or *leave* a group at any time. Group members receive all datagrams addressed to the group, while *any* host can send to a group, regardless of its membership status. Multicast IP datagrams are distinguished from unicast ones only by their Class D destination address. To deliver multicasts to a host group in a wide area internetwork, we need a mechanism to track group membership and a mechanism to route multicast datagrams from a sender to all group members, without duplicating traffic. In IP multicast, these mechanisms are split in two parts: *local* mechanisms track group membership within a LAN and deliver multicasts to the correct hosts in that network, and *global* mechanisms route datagrams between LANs. Local mechanisms can be customized to the particular properties of each LAN, as long as they support the common interface over which the global mechanisms operate, although such customization is not currently done.

In each LAN at least one host acts as a *multicast router* that keeps track of local membership in multicast groups and forwards multicasts between the local network and external sources and destinations. Multicast delivery to local receivers and capture of local multicasts for subsequent forwarding by the router depend on the capabilities of the underlying network technology, therefore the information needed within a LAN regarding group membership in order to carry out local delivery tasks 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 LAN and the outside world. The *only* information needed in order to decide if a multicast should be delivered to a target LAN is whether at least one member for its destination group is present there. Wide area multicast routing and delivery is based on the exchange of this information between routers. Thus, irrespective of the group membership information tracked by a multicast router for local purposes, *the interface between local state and global routing is a list of groups present at each local network*. Based on this common interface, alternative mechanisms can be used for wide area multicast routing, without affecting local mechanisms, and vice versa.

B. Global IP Multicast Mechanisms

The original global multicast mechanism used on the MBone was the *Distance Vector Multicast Routing Protocol* (DVMRP). DVMRP v.1 [4] is a variation of *Truncated Reverse Path Broadcast* [5]. Each router uses a separate distribution tree for each source sending to a group, so that datagrams from the source (root) are duplicated only when paths towards destination networks (leaves) diverge. The router identifies the first link on the shortest path from itself to the source (a *reverse* path) using a distance vector algorithm. Datagrams arriving from this link are forwarded towards downstream (with respect to the tree) routers. As a result, a *broadcast* tree is formed and datagrams reach all routers. Since each router knows which groups are present in its LAN, redundant datagrams are not forwarded there, thus *truncating* the tree. The latest incarnation of this mechanism, DVMRP v.3 [10], uses the improved *Reverse Path Multicast* [5] algorithm, which *prunes* tree branches leading to networks that have no members for a group, and *grafts* them back when such members appear. Another mechanism in use is the *Multicast Open Shortest Path First* (MOSPF) [8] protocol, based on a link state algorithm. In MOSPF routers flood their membership lists among them, so that each one has complete topological information concerning group membership. Using this state, shortest path multicast distribution trees from each source to all destination networks can be computed on demand using Dijkstra's algorithm.

A radically different approach to multicast routing is the *Core Based Trees* [1] (CBT) protocol, which employs a *single* tree for each group, shared among all sources. The tree is rooted at an arbitrarily chosen multicast router (the *core*) and extends towards all LANs containing group members. It is explicitly constructed from leaf network routers towards the core as group members appear. Sending to the group is accomplished by sending towards the core; when the datagram reaches the tree it is relayed towards the leaves. Routing is thus a two stage process, and usually sub-optimal as the first stage may propagate datagrams away from their destinations. Traffic with CBT tends to concentrate on the single tree rather than being spread throughout the network. Since no routing mechanism is perfect, the *Protocol Independent Multicast* (PIM) [3] protocol, employs either shared or per source trees, depending on application requirements. Note that all global schemes route multicasts between *networks* rather than *hosts*, so as to keep routing calculations and tables within reasonable limits. Multicasting within each LAN is a task for the local mechanisms. Protocols that construct per source trees, (DVMRP, MOSPF and PIM in *dense mode* [3]), also deal with source networks.

Each *Autonomous System* (AS) on the Internet, may employ its own multicast routing scheme, necessitating the use of an inter-area routing protocol [11]. In addition, networks supporting IP multicast may only be connected via multicast unaware routers. In this case, *tunnels* are used, i.e. virtual links between two multicast routers that are composed of a sequence of physical links. Multicast datagrams are relayed between the tunnel endpoints by being encapsulated within unicast datagrams at the sending end and decapsulated at the receiving end. The MBone [6] is a virtual network composed of multicast aware networks bridged by tunnels. Multicast routers may limit multicast propagation, or *scope*, by only forwarding through tunnels datagrams that have *Time to Live* values above a threshold.

C. Local IP Multicast Mechanisms

In contrast to global mechanisms, there exists only a single choice for local mechanisms. These mechanisms were developed for shared medium broadcast networks, where multicast delivery is simple as all hosts can listen to all datagrams and select the correct ones. For LANs that support multicast as a native service (e.g. Ethernet), class D IP addresses can be mapped to LAN multicast addresses to filter out redundant multicasts in hardware. Software filtering may be employed to substitute or complement hardware filtering. Multicasts with local scope do not require any intervention by the multicast router, while externally originating multicasts are relayed to the LAN by the router if the destination group is present there. The router also monitors all locally originating multicasts in order to forward to the outside world those for which receivers exist elsewhere. Since the medium is broadcast in nature, the exact number of members for each group does not matter, only the presence or absence of members counts. As a result, the router keeps internally only a local group membership list, the same information on which global multicast routing is based.

The *Internet Group Management Protocol* (IGMP) provides a mechanism for local group management targeted to broadcast LANs, i.e. only group presence or absence is tracked. In IGMP v.1 [2] the multicast router periodically sends a local *query* message to a multicast group to which all local multicast receivers listen. Each host, on reception of the query, schedules a reply, or *report*, to be sent after a random delay (smaller than the query interval), for each group in which it participates. Reports are sent to the address of the reported group, so that the first report sent will suppress the rest. The random delays before replying spread the scheduled report transmissions so that usually only one report is sent. The multicast router monitors all multicast addresses and updates its membership list after receiving each report. If no reports are received for a previously reported group after a number of queries, the group is removed from the membership list. Each host when joining a group sends a number of unsolicited reports to reduce *join latency*, i.e. the time between a host joining the group and the router starting multicast propagation towards the host. Hosts do not take any explicit action when leaving a group, as group presence will time out eventually.

In IGMP v.1 after the last member of a group leaves it, datagrams for the group are still forwarded to the LAN until its membership times out, thus wasting bandwidth. The time between these two events is called the *leave latency*. To reduce it, in IGMP v.2 [7] a host sends a *leave* message when abandoning a group if it was the *last* host to send a membership report for that group. This implies that each host must maintain per group state showing if it was the last group member to send a report for that group or not. If this host was the last member of the group, it must have sent the last report, so the leave message means that the group is absent. However, since this last report may have suppressed others, the multicast router must explicitly probe for group members by sending a *group specific* query to trigger membership reports only for the group in question. It can then assume the group absent only if no reports arrive after a number of such queries. IGMP v.2 queries include a time interval within which reports must be sent; general queries use a long interval for effective randomization of periodic reports, while group specific queries may use a short interval to speed up group status detection.

The basic periodic query/report mechanism has two drawbacks: it transmits the same information periodically, and it wastes bandwidth due to leave latency. While join latency is avoided by unsolicited reports, the IGMP v.2 leave messages reduce but do not eliminate leave latency, since they are only *hints* rather than authoritative information. Leave messages are insufficient to establish group absence and have to be supplemented by periodic queries and membership presence timeouts due to the fact that only a group presence list is kept by the multicast router. This list is sufficient for supporting delivery decisions for broadcast LANs, trading off bandwidth overhead due to repeated query/report cycles and leave latency for *soft state* group management at the router. This is an appropriate choice for a LAN with plenty of bandwidth and hosts that may reboot without the router noticing. The duration of the query interval is a compromise between management overhead, arguing for long intervals, and delivery overhead after a group disappears, arguing for short intervals. Hence the decision in IGMP v.2 to support distinct intervals for each query at the price of complicating router operations. This scheme is also complex for end hosts, which have to continuously listen for queries, set up random timers, and suppress or send reports.

D. Point-to-Point Local Link Issues

When hosts are connected to multicast routers via Point-to-Point links, some IGMP assumptions do not hold any more. While bandwidth may be plentiful in ATM LANs, hosts connected to the Internet via telephone lines are bandwidth limited, especially when these links are wireless. Mobile hosts have additional battery power and processing constraints that urge for simplified local mechanisms and minimal transmission overhead. Another case of a PtP link is the tunnel used by Mobile IP [9] to connect a mobile host to the Internet via tunneling of datagrams from an *agent* in its *home* network. When a multicast router is not present on the network visited by a mobile host, local multicast mechanisms are employed over the tunnel by having the home agent act as the multicast router [12]. Thus, IGMP operation and multicast delivery is carried out over this virtual PtP link, that includes at least one wireless link, either broadcast or unicast based. Scarce bandwidth and battery power favor local mechanisms that minimize or eliminate leave latency and group management overhead. IGMP v.2 reduces leave latency overhead compared to IGMP v.1, but does not eliminate it, while it introduces additional group management overhead. In addition, the periodic queries prohibit mobile hosts from using *sleep mode* to conserve battery power when no multicast traffic is present.

We can fortunately avoid these problems of the query/report mechanism on PtP links, based on the key observation that since only one host resides at each endpoint of the link, group presence or absence on a PtP link is equivalent to this single host being or not being a member of a group. This means that both join and leave messages can be interpreted unambiguously as indications of group presence and absence, respectively, and periodic queries are not needed to detect group absence. The multicast router does not need to maintain additional data structures to exploit this property of PtP links, and it can easily aggregate membership lists across many PtP links in order to present the image of a single LAN to the outside world. Multicast delivery from the router to end hosts is trivial, and the same holds for the capture of multicasts from end hosts that the router may need to forward towards other networks. To complete the emulation of a broadcast LAN by a PtP LAN, the only additional task for the multicast router is to deliver locally originating multicasts to all other local PtP links where the destination group is present.

III. Supporting Point-to-Point Networks

A. Extending Existing Mechanisms

Existing IP multicast implementations treat each interface of a multicast router, including each attached local PtP link, as an entry point to a LAN (note the distinction between *backbone* and *local* PtP links). The router executes a separate instance of the IGMP protocol for each local PtP link, complete with timers and membership lists. In DVMRP v.1 each multicast reaches all routers, which use their membership lists to determine which local links should receive the datagram. Routing protocols that limit multicast propagation to routers that actually have group members in their LANs, such as MOSPF and DVMRP, require routers to summarize membership information across their local links to avoid routing explosion. This is especially important for routers serving large numbers of slow PtP telephone links. As the number of PtP links grows, executing multiple IGMP instances and periodically aggregating information for wide area routing imposes a significant processing burden. Since per PtP link membership lists are required to avoid relaying redundant multicasts over *all* links, per link IGMP operation and membership list aggregation are unavoidable.

One approach to mitigating these problems is to review the design assumptions of the broadcast LAN based IGMP and try to make PtP specific improvements. IGMP v.2 leave messages are not authoritative indications of group absence in broadcast LANs, so the router has to send group specific queries to determine group status, and these queries must be repeated to guard against IP datagram losses. In a broadcast based LAN, if the group is not absent these messages are wasted. Increasing general query intervals to balance this potential overhead has its limits, as group absence *cannot* be noticed via leave messages in some (admittedly rare) cases, where general queries are the only way to determine group status. In contrast to broadcast LANs, leave messages over PtP links *are* authoritative: the single host connected to the link determines group status. Furthermore, this single host is always the one that sent the last report for the group, and therefore it will always send a leave message on abandoning the group. Even though this means that leave messages are always sufficient for determining group absence on PtP links, IGMP still goes through its periodical general and group specific query/report cycles.

An alternative approach would be to treat local PtP links the same as backbone PtP links by turning each end host into a multicast router. Although multicast routing was designed for PtP links, extending wide area routing to end hosts is impractical due to scalability problems. DVMRP v.1 would broadcast all multicast datagrams to each end host due to the lack of truncation. DVMRP v.3 starts in broadcast mode and periodically reverts to it, before pruning the distribution trees, so it would encounter similar problems. MOSPF would face routing table explosion since each router would have to keep track of every end host's group membership. Only protocols employing shared trees (such as CBT or PIM in *sparse mode* [3]) could work with this approach, since they only employ local information for routing and they never use broadcasts. Regardless of routing protocol however, 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 possible optimizations can be taken advantage of, rather than forcing internetwork mechanisms over local networks that are ill suited to them.

B. The Join/Leave Mechanism

Group membership reports are always sufficient to indicate group presence on both broadcast and PtP links, while leave messages are always sufficient to indicate group absence on PtP links only. Thus, periodic group membership queries and their accompanying reports can be replaced on PtP links by *join* and *leave* messages: the leave message obviates the need to periodically reconfirm membership, while the join message immediately indicates group presence. General queries, which serve as synchronization points for the random timers in broadcast LANs, are redundant for PtP links, and so are the group specific queries of IGMP v.2. Since IP only promises best effort delivery, existing IGMP implementations maintain their membership lists as *soft state* that is refreshed by the periodic query/report cycles. The *hard state* provided by the join/leave messages can be made equally robust by having the router acknowledge them and the end hosts retransmit them if an acknowledgment does not appear before a timer expires. Mobile IP tunnels which are even more unreliable due to their wireless links, provide additional motivation for using confirmed join/leave messages. The join/leave mechanism is similar to how multicast group membership is tracked in IP multicast over ATM, but rather than a necessity, here it is an optimization. It is based on the observation that in PtP links the periodic queries result in end hosts repeatedly transmitting their *complete state*. The join/leave messages instead transmit only a *state difference*.

The join/leave mechanism completely eliminates local join and leave latencies. Since there are no waiting periods for group timeouts, overhead from redundant local multicast transmissions is avoided. Similarly, periodic queries and reports are also eliminated so that group management overhead is also reduced. 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 only have to wake up when there are multicasts for them or their membership status changes, not periodically due to router queries. There is no need for periodic per group timer management to handle membership reports, although per join/leave event timers are now required for the retransmission of lost join/leave messages. The multicast router, apart from avoiding periodic queries, is basically unaffected. It must update its membership lists when join and leave messages arrive rather than periodically. It can also immediately aggregate group information across all local PtP links (see below), in order to present to other routers the image of a single local network. More timely aggregation of group membership information contributes to more efficient wide area multicast routing: groups new to a network may be propagated faster to their local recipients, while groups that disappear from a network may stop being forwarded there earlier.

IV. Implementation Issues for the Join/Leave Mechanism

In order to facilitate comparisons between our approach and existing mechanisms, we will describe here a possible implementation of the proposed join/leave local multicast mechanisms. To clarify both differences and similarities between the join/leave and query/report schemes, we present a sample implementation as a set of modifications to existing query/report implementations. We assume that the network layer multicast process is notified whenever the state of the PtP link changes, either due to the peer rebooting or due to a *handoff* in a cellular network. Since there are only two endpoints on a PtP link, changes on the peer's state will normally be noticed somewhere between the physical and the network layer, and a notification should be sent to the multicast process. In the case of false alarms, i.e. when the state change does not really influence the network layer, our mechanisms still operate correctly, at the expense of some group management overhead.

Multicast transmission between hosts and the router uses the same primitives as unicasts, so the router automatically receives all multicasts originating at the host. The router must maintain per link group membership lists in order to deliver both locally and externally originated multicasts only to group members, but this state is already kept by the existing mechanisms which treat each PtP link as a separate interface with its own group membership list. At an end host, as processes join and leave multicast groups, the host updates a reference count for membership in each group, so that it can notify the multicast process when a group is initially joined or finally abandoned. In contrast to existing mechanisms, these notifications cause the transmission of acknowledged join and leave messages, i.e. the host retransmits the messages if no reply arrives after a timeout, and therefore a timer must be kept for each as yet unacknowledged message. These acknowledgments and retransmissions compensate for the automatic recovery from lost messages provided by the periodic queries and reports in existing mechanisms. Since a leave message cancels a previous join message for the same group, we only need to keep track of one timer per group (when required) and an indication of whether a join or a leave message is to be retransmitted. With existing mechanisms hosts instead have to periodically set and reset one random timer for each group that they belong to. Hosts also do not need to remember if they sent the last membership report to the router, as in IGMP v.2. Since with join/leave group management consists of isolated rather than periodic actions, in periods of network inactivity battery powered hosts can employ sleep mode without interruptions by periodic IGMP queries.

One issue for the join/leave mechanism is choosing timeout values for retransmissions. For physical PtP links, the round trip transmission delay is known and relatively static, although queuing delays within the router and the end host influence total delay. The problem is more complicated with virtual PtP links such as Mobile IP tunnels, where delay is unpredictable due to the arbitrary tunnel length, and also varies due to dynamic routing and host mobility. However, estimation of tunnel round trip delay is also an issue with Mobile IP itself [9], thus the mechanism employed there can be reused. Note that timer value estimation with join/leave is a host issue that does not influence the multicast router, in contrast to the query/report case where timer values are selected by the router, which faces similar problems to the ones discussed above when tunneling is involved.

On the router side, query timers, general queries and group specific queries, are all eliminated. Actually, with join/leave the router *never* initiates any message exchanges, it only acknowledges join/leave messages as they arrive. Updating per link membership lists is simple: add the group after a join message, delete it after a leave message. When acknowledgments are lost, duplicate join/leave messages may be received by the router, which are distinguished by the fact that they do not have any effect on the link membership list. Only when new join/leave messages arrive that cause the router to modify the corresponding link membership list, is there a need to update the aggregate router membership list. One simple method of aggregate list management is to use reference counts for each group entry, updating them based on the join and leave actions that modify the per link lists. If a new group on a link is not on the aggregate list, it is added and its reference count is set to one, while new joins on other links cause the reference count to be increased. Conversely, leave actions that modify a per link list cause the reference count to be decreased, and the group to be deleted when the reference count drops to zero. With this scheme both per list and aggregate group membership lists are always updated immediately.

The join/leave messages can employ the same format as existing IGMP messages, with new type numbers, since all existing IGMP messages contain the group address field that join/leave messages use. For static networks, both multicast router and hosts would use the same IGMP variant, so existing messages could be reused (*report* and *leave*) unambiguously. In mobile wireless networks however, visiting hosts and routers may not all employ the join/leave mechanism, so some negotiations are needed to establish a common mode of operation. All systems should minimally implement the standard query/report mechanism. Since with join/leave routers do not initiate any transactions, a mobile host could initiate operations by sending a new type join message for an arbitrary group: if the router acknowledges the message, the join/leave mechanism is supported, and the host can then leave the group, else, if after a number of retransmissions there is no response, the mobile host should fall back to query/report. The router should start in query/report mode, and switch to join/leave mode for a *specific* PtP link only after receiving a new type join.

When an end host reboots, the router should clear its link membership list, and the host re-establish its state as its applications are restarted. When the router reboots, the host should retransmit a join for each of the groups it participates in to re-establish the correct link membership list on the router. For mobile hosts, when a handoff is detected, the host should act as if its router was rebooted, and re-establish its state on its new router, while the router previously serving the mobile should empty its membership list as if the end host was rebooted. Thus, link state changes are treated the same whether they result from a reboot or a handoff. If a mobile host employs Mobile IP tunnels for multicast delivery, handoffs are ignored as long as the host uses its home agent as its multicast router, with handoff/reboot processing triggered only when the mobile switches between home agent to local multicast routing. The home agent would notice such a change and also act as if the mobile host was rebooted, during the Mobile IP *registration* phase. Note that we assumed earlier that the multicast processes on both ends are somehow notified of reboot/handoff events. When the notifications are false alarms, the state is re-established, thus wasting some bandwidth.

V. Evaluation of the Join/Leave Mechanism

A. Performance Analysis

We can compare quantitatively the performance of the various group membership management mechanisms described in this paper by calculating their overhead, i.e. the bandwidth that is used beyond that for the multicast data that group members wish to receive. The first source of overhead is redundant multicasts transmitted by the router during the *leave latency* period of each mechanism, that is, after a host has left a group and before the router stops forwarding its datagrams to it. The second source of overhead is the group management messages that are required for the mechanism's operation. In IGMP v.1 the balance between group management and leave latency incurred overhead is determined by the query interval: small intervals waste more bandwidth for control, but reduce leave latency, and vice versa. In IGMP v.2 leave latency is reduced due to the leave messages, but additional group specific queries are employed. In PtP links leave messages never cause unnecessary overhead as they are always authoritative, even though IGMP v.2 does not exploit that fact. Leave latency is also reduced due to the short group specific query interval of IGMP v.2. Note that the same query interval values are used for *all* groups, i.e. it is not possible to adjust the intervals based on a specific group's traffic. The join/leave mechanism completely eliminates leave latency, so in terms of overhead due to redundant multicast transmissions it is *always* superior to both query/report mechanisms, when no messages are lost. When messages are lost, all mechanisms recover after additional messages, but while join/leave can use tight timers that only account for round trip and processing delay, query intervals must be large enough to make randomization effective in suppressing duplicate reports, even though this is not needed for PtP links. As a result, the join/leave mechanism recovers faster from losses, remaining superior to the query/report mechanisms in terms of leave latency overhead.

Group management overhead, as measured by the number of control messages exchanged, is somewhat more complicated. Query/report mechanisms require exactly one report for each group that the host belongs to (for PtP links), but leaving *any* number of groups during the same interval leads to *all* groups being deleted when their membership timers expire. Assuming that control messages are never lost and that a host belongs to many groups (so that we can ignore the cost of the single query per interval), the join/leave mechanism is superior to IGMP v.1 when *membership to any group lasts for more than four consecutive intervals*. The reason 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, equal to the number of reports that are sent for a group after four queries are received. There is no cost for leaving a group in IGMP v.1 as detection of group absence is automatic. In practice, IGMP v.1 hosts send a few unsolicited reports on joining a group to reduce join latency, thus tipping the balance in favor of join/leave even for shorter time intervals. IGMP v.2 also supports leave messages and group specific queries, that normally have to be repeated for robustness. Leave messages always appear on PtP links since the single host is always the last sender of membership reports for *all* groups. As a result, IGMP v.2 (which improves on IGMP v.1 in terms of leave latency overhead) balances the four control messages of join/leave *after only two query intervals*, since the host sends two periodic reports, one leave message, and the router sends one group specific query. Thus, protocol message overhead for join/leave is usually lower than that for IGMP, even under the most favorable (for IGMP) parameters, while leave latency overhead is always lower.

For a concrete performance analysis, we will show the exact overhead incurred when the two IGMP versions and our join/leave mechanism are employed in a very general setting. For simplicity, we assume that no protocol messages are lost, and that all IGMP and join/leave messages are 256 bits long on the link (20 bytes for the IP header, 8 bytes for the IGMP payload, and 4 bytes for link layer overhead). To compute the overhead for each mechanism, we need a set of values for the timers and counters of the IGMP variants; we use the values specified in the draft specification for IGMP v.2 [7] for both protocols. The specification requires all IGMP hosts to send two unsolicited reports on joining a group, and then reply to periodic queries that are sent every 125 seconds. In IGMP v.1 a group is assumed absent after 350 seconds (two query intervals plus 100 seconds), so assuming that a host leaves a group exactly halfway between two reports (spaced on average every 125 seconds), the leave latency for IGMP v.1 is 287.5 seconds. In IGMP v.2 a leave message is always sent and the router sends two group specific queries (every 1 second) and waits for 1 more second before assuming the group absent, so the leave latency is 3 seconds. For the join/leave mechanism we always need one acknowledged join and one acknowledged leave message per group.

Figures 1, 2 and 3 show the combined protocol and leave latency overhead for one multicast group, as a percentage of the (constant) data rate of the group. The overhead as a percentage is computed by dividing overhead by the data rate only. Overhead is shown for group membership durations between 1 minute and 100 minutes to capture the effect of different group membership dynamics. Since the unsolicited IGMP reports are spaced 10 seconds apart, and any leave latency overhead is incurred regardless of membership duration, the overhead calculations are valid for the entire membership duration range. The multicast data rates under examination range from 1 bps to 100 Kbps: the lower limit could be the data rate of a messaging or very low rate information service, while the upper limit is within the reach of digital wireline telephone links. We have ignored the impact of query messages on overhead, since it is only about 2 bps which would be significant only for a tiny area

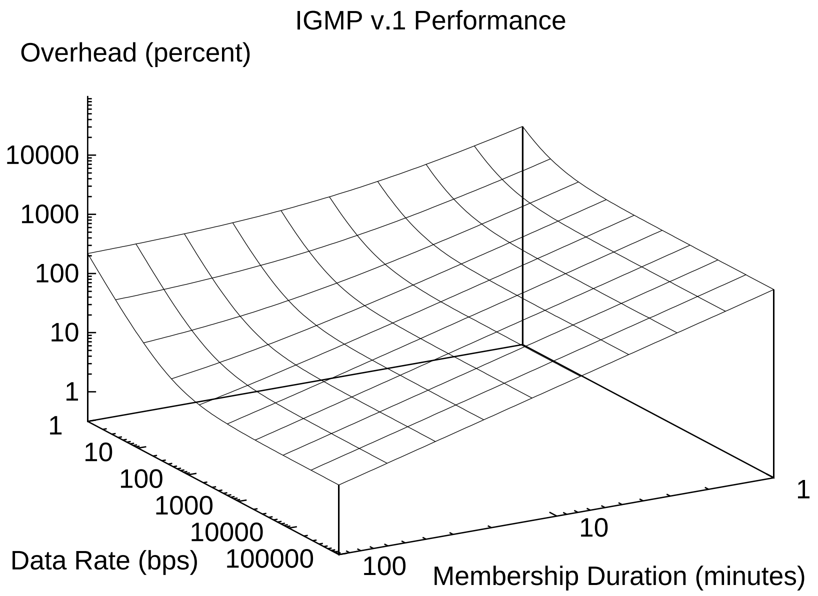

Fig. 1. Overhead as a percentage of data rate for IGMP v.1

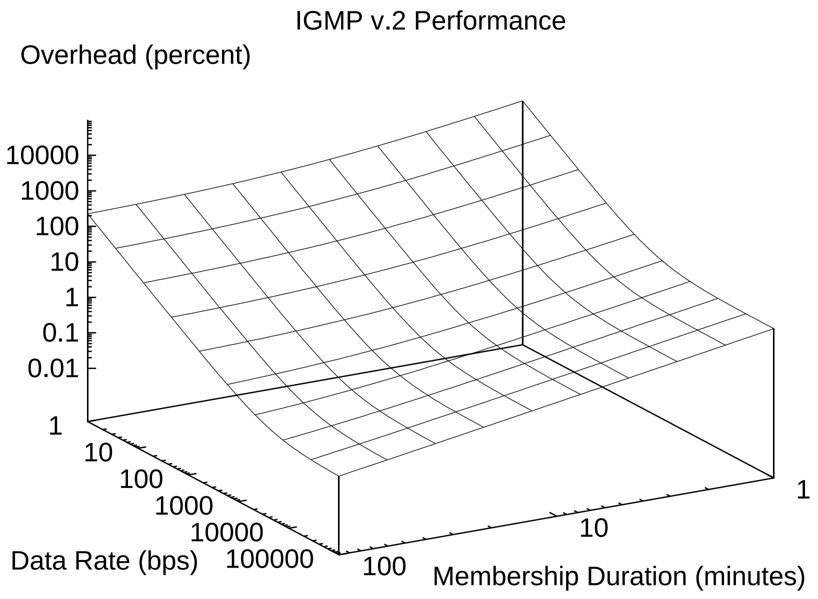

Fig. 2. Overhead as a percentage of data rate for IGMP v.2

of our data rate range and it is amortized among all groups. All scales are logarithmic in order to expose details at both ends of each parameter range. The graphs and the discussion that follows actually characterize the behavior of the mechanisms for wider parameter ranges, but we believe that these ranges adequately exhibit all relevant performance issues.

From Figure 1 it is clear that in IGMP v.1 leave latency overhead is the dominant factor, due to its long duration of 287.5 seconds, which is only somewhat amortized over large membership periods. On the other hand, group management overhead is only important for very low data rates, and has a marked effect on total overhead only when the membership duration is large enough to reduce the effects of leave latency. Apart from the unsolicited reports sent on joining a group, the protocol overhead of IGMP v.1 is static over any membership duration, at one report per query interval. IGMP v.2 adds to this overhead its three messages on leaving a group (one leave and two group specific queries), so its management overhead follows a similar curve to that of IGMP v.1, as shown in Figure 2. Since its leave latency is only 3 seconds though, this part of the overhead is quite small even for short membership durations, making the overhead curves flatter than those of IGMP v.1 as the data rate is increased, and the effect of control overhead more pronounced for small data rates. Since join/leave incurs no leave latency overhead at all, the surface depicted in Figure 3 is completely flat. The fixed control overhead of 4 messages is amortized very fast, either due to longer membership duration or due to higher data rates. For the operating parameters of the IGMP mechanisms used in the graphs, join/leave outperforms IGMP v.2 for *any* membership duration and data rate, as IGMP v.2 sends at least 5 messages (2 unsolicited reports, one leave, two group specific queries), regardless of leave latency. IGMP v.1 hosts always send 2 unsolicited reports on joining a group, so join/leave with its fixed cost (4 messages or 1024 bits) outperforms it in two general cases: first, when membership lasts for at least *one* query interval (one report and at least 287 redundant bits during the leave latency period are sent), and second, when the data rate is 2 bps or more (i.e. 574 or more bits are sent during the leave latency period).

As the preceding analysis shows, join/leave outperforms both versions of IGMP for the vast majority of membership durations and data rates considered. Figures 4 and 5 show the mechanisms compared to each other when we fix the data rate and the membership duration, respectively, at a representative value. Essentially, these are cross sections of the surfaces shown in the previous figures, with one of the two parameters held constant while the other one varies as before, and all other assumptions remaining the same. In Figure 4 we have fixed the data rate at 100 bps, while the membership duration interval varies from 1 to 100 minutes. Even for this low data rate, the leave latency overhead incurred by IGMP v.1 is dominant at short membership durations. By reducing this overhead, IGMP v.2 is closer to, but worse than, join/leave at this end of the range. As membership duration grows, leave latency overhead is amortized, and the dominant cost for either IGMP mechanism is the periodic control overhead, so they tend to converge towards the same overhead rate. In contrast, join/leave only causes a fixed number of control messages to be sent, so its overhead rate decreases indefinitely.

In Figure 5 we have fixed the membership duration at 10 minutes, while the data rate varies from 1 bps to 100 Kbps. For this moderate membership duration interval, IGMP v.1 overhead is considerable over all data rates, since its significant leave latency overhead is not amortized. Interestingly, IGMP v.2 performs worse than IGMP v.1 for very low data rates, since its leave messages and group specific queries consume much more bandwidth than multicast traffic, dominating even the leave latency overhead of IGMP v.1. As the data rate is increased, the control overhead is dominated by leave latency overhead, which is proportional to the data rate (for a fixed membership duration), so both IGMP versions converge to constant overhead rates (a high rate for IGMP v.1 and a low rate for IGMP v.2). Join/leave only causes a fixed number of control messages to be sent, so its overhead percentage again decreases indefinitely.

B. Qualitative Analysis

Since the join/leave mechanism is host initiated, as opposed to the router initiated query/report mechanisms, battery powered mobile hosts can employ sleep mode when there is no traffic of interest to them. They only have to wake up in order to process join/leave messages generated by their applications and multicasts that their applications have asked for, not to reply to queries by the group management protocol. Since the multicast router has more accurate information about their membership status, no power is wasted receiving redundant multicasts (equivalently, the leave latency period is eliminated). Furthermore, since the join/leave mechanism is considerably simpler in operation than the standard mechanisms, due to a lack of random timers and a simplified protocol state machine, its processing requirements are lower, which is more important for battery powered hosts. As an aside, added accuracy in tracking local group membership can also be beneficial by reducing multicast traffic among routers.

Interoperability among different IGMP versions can be achieved by using the simple procedures described earlier that provide for automatic discovery and use of the best mechanism supported by both peers at every link, with a fallback to suboptimal mechanisms for compatibility. Routers using join/leave mechanisms for their local PtP links can seamlessly participate in wide area multicast routing based on the information maintained on their present groups list, the standard interface between local and global multicast mechanisms. An improvement over existing IGMP mechanisms is the simple and fast aggregation of all PtP link information into a single list, as if a single LAN interface was supported by the router. Deployment of the join/leave mechanism should be direct in networks where all hosts and routers are under the same administrative control, as all systems can be upgraded transparently to the rest of the Internet. For wireless mobile networks where hosts and routers employing different IGMP versions may be encountered, both versions should be implemented in a common module to ensure backwards compatibility. Since all mechanisms share the same data structures, dual implementations should not be considerably larger than existing ones, placing only minor additional storage requirements on hosts. Finally, the join/leave mecha-

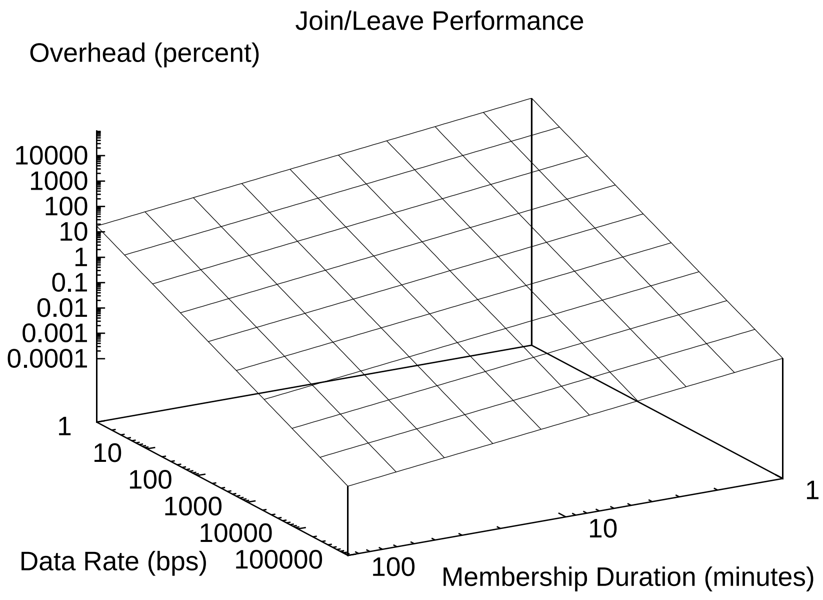

Fig. 3. Overhead as a percentage of data rate for Join/Leave

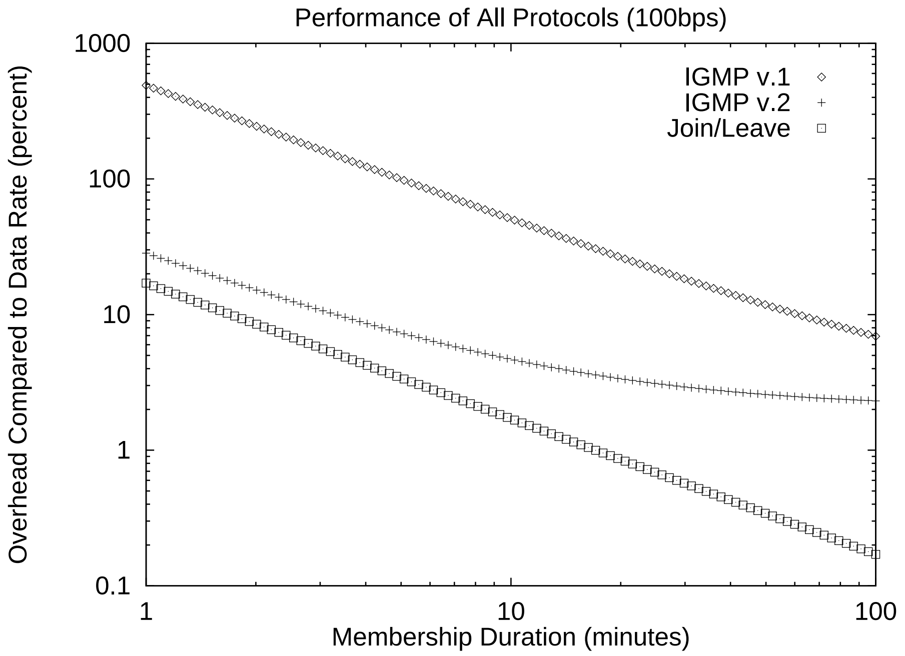

Fig. 4. Comparison of overhead as a percentage of data rate when data rate is 100 bps

nism explicitly caters for and co-operates better with Mobile IP tunnels and in general with host mobility and handoffs.

Robustness in join/leave is achieved by using acknowledgements, while in query/report mechanisms it is provided by the periodic queries. An advantage of the join/leave mechanism is that explicit acknowledgments to join messages are a positive indication to hosts that they are indeed members of a group, while in the query/report case a host cannot distinguish between normal group inactivity and lost reports. Conversely, absence of reports to periodic queries may mean either group absence or lost reports, while in the join/leave case leave actions are authoritative. Since the protocol caters for peer reboots and handoffs, group membership state recovery in these cases is faster as join/leave messages are initiated by the end host as soon as possible, rather than as replies to router queries.

The sample implementation that was described earlier is based on what is already implemented for existing mechanisms, showing that join/leave mechanisms are easy to add and deploy. The only significant addition over IGMP is the initial handshaking phase used to discover the protocol mechanisms sup-

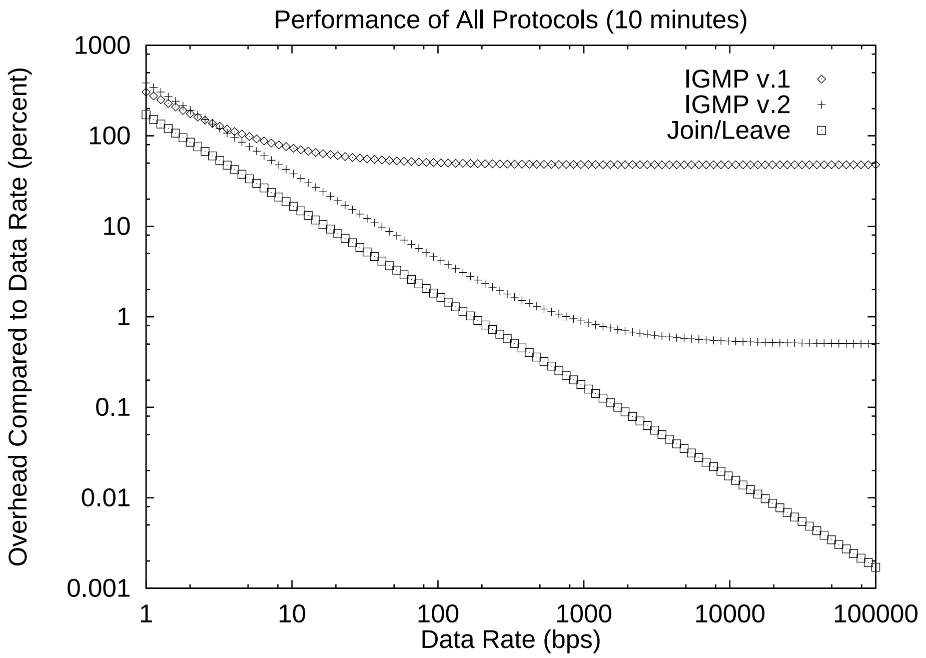

Fig. 5. Comparison of overhead as a percentage of data rate when membership duration is 10 minutes

ported by the peer. In most aspects, including timer management, mapping process events to protocol messages, and mapping protocol messages to state updates, the join/leave mechanism either considerably simplifies existing mechanisms, as with timers, or completely eliminates them, as with group specific queries. A pure join/leave implementation could be developed by deleting code from existing query/report implementations and adding join/leave handling and retransmission timers, resulting not only in easier code maintenance and further development, but also in improved performance. Even when multiple protocol variants are implemented for compatibility, only the faster join/leave code would be executed when supported by both peers.

VI. Conclusion

We have presented some problems that existing implementations of the IP multicast 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 multicast group management mechanisms towards broadcast LANs. After considering various alternatives, we proposed a *join/leave* mechanism for tracking group membership over PtP networks. We then presented an implementation outline for this mechanism based on modifications to existing mechanism implementations, showing that join/leave considerably simplifies operations. We also compared the join/leave approach with the standard query/report approach with regard to protocol performance, as measured by transmission overhead, mobile power efficiency, interoperability with existing protocols, robustness and implementation complexity. Based on this comparison we draw the conclusion 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 and processing power are limited, as in battery powered mobiles employing cellular telephone links, our proposal offers clear and significant advantages.

References

[1] A. Ballardie, J. Crowcroft and P. Francis, Core based trees (CBT) – An architecture for scalable inter-domain multicast routing, in: *Proc. of the ACM SIGCOMM '93* (San Fransisco, CA, 1993) 85–95.
[2] S. Deering, Host extensions for IP multicasting, *RFC 1112* (1989).
[3] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, C. Liu and L. Wei, An architecture for wide-area multicast routing, in: *Proc. of the ACM SIGCOMM '94* (London, UK, 1994) 126–135.
[4] S. Deering, C. Partridge and D. Waitzman, Distance vector multicast routing protocol, *RFC 1075* (1988).
[5] S.E. Deering and D.R. Cheriton, Multicast routing in internetworks and extended LANs, *ACM Transactions on Computer Systems* **8** (1990) 85–110.
[6] H. Eriksson, MBONE: The multicast backbone, *Communications of the ACM* **37** (1994) 54–60.
[7] W. Fenner, Internet group management protocol, version 2, *Internet Draft – Work in Progress* (1997).
[8] J. Moy, Multicast routing extensions for OSPF, *Communications of the ACM* **37** (1994) 61–66.
[9] C. Perkins, IP mobility support, *RFC 2002* (1996).
[10] T. Pusateri, Distance vector multicast routing protocol, *Internet Draft – Work in Progress* (1997).
[11] A. Thyagarajan and S. Deering, Hierarchical distance-vector multicast routing for the MBone, in: *Proc. of the ACM SIGCOMM '95* (Cambridge, MA, 1995) 60–66.
[12] G. Xylomenos and G.C. Polyzos, IP multicast for mobile hosts, *IEEE Communications Magazine* **35** (1997) 54–58.