Router Level Filtering for Receiver Interest Delivery

Manuel Oliveira Jon Crowcroft
University College London
Gower Street
WC1E 6BT London, UK
+ 44 207 679 7214
{m.oliveira | jon}@cs.ucl.ac.uk

Christophe Diot
Sprint ATL
1 Adrian Court
Burlingame 94010 CA, USA
+ 1 650 375 4539
cdiot@sprintlabs.com

ABSTRACT

Delivering data to on-line game participants requires the game data to be "customized" in real-time to each participant's characteristics. Using multicast in such an environment might sound contradictory. But multicast is a very efficient communication paradigm to minimize the transmission delays. Also, multicast reduces the workload at the sender. Content delivery according to receiver interest can be achieved by group management in multicast. But the natural dynamics of the application results in numerous delays because of join/leave latencies. In this paper, we propose the Router Level Filtering as a solution to the above problem.

RLF relies on an extension to the current IP multicast service model. It introduces "filters" in the router forwarding process thereby providing a simple effective mechanism to customize the data delivered to a multicast session receiver while minimizing the number of groups and the related management cost. Contrary to other router filtering proposals, the filter semantics is determined by the application. The paper discusses protocol specification and implementation details of RLF, and shows how it may be implemented in routers.

Categories and Subject Descriptors

C.2.1 [**Computer System Organization**]: Computer Communication Network – *Network Architecture and Design.* C.2.2 [**Computer System Organization**]: Computer Communication Network – *Network Protocols.*

General Terms

Algorithms, Design.

Keywords

Keywords are your own designated keywords.

1. INTRODUCTION

The IP multicast service model was defined by Steve Deering [9]. In the past 10 years, the protocol architecture has been refined thanks to experiments led on the Mbone [14]. Applications were prototyped, from multicast data delivery [4] to audio and video-conferencing [20,33] thorough more interactive applications such as shared work environments and distributed games [12]. Despite all these experiments, multicast has not yet been successful as a commercial service. Reasons for this difficult deployment have been discussed in [13]. In summary, [13] invokes the complexity of the protocol architecture and the absence of some functionality, such as access control and address allocation.

Applications of large scale in terms of content and infrastructure are expected to see a wide deployment in the near future, namely Large Scale Virtual Environments (LSVE) and on-line games. However, multicast does not currently accommodate the requirements of their communication model. In fact, in LSVE the usage of multicast is quite limited considering that the underlying network infrastructure may be:

- A small-scale proprietary WAN of high bandwidth put together exclusively for the purpose of a particular military simulation based upon the Distributed Interactive Simulation (DIS) [24,25].
- A small set of multicast islands that are interconnected via proprietary solution based on TCP or UDP [16], thus foregoing any need for multicast routing protocols.

In the case of on-line games, even if multicast were ubiquitously available, it is not considered as an option by multicast game designers who claim that multicast is not an appropriate technology for the following reasons:

- Multicast does not provide mechanisms that may be considered efficient in content delivery based on receiver characteristics (or their interest).
- The process of transitioning between several multicast groups may be detrimental to both network and computer resources and performance, ultimately affecting the satisfaction of the end-user.
- There is an absence of access control along with other group management mechanisms.

To address some of the above problems, Source Specific Multicast (SSM) [22] is emerging as a new simplified multicast architecture. It relies on PIM-SM [15] and IGMPv3 [7], and provides straightforward address allocation, access control. It is based upon a single sender model (which proves to be appropriate for server-based architectures). However, some issues remain unsolved:

- SSM does not provide a simple way to tailor the content delivered to each receiver;
- The latency associated to join and leave remains unchanged, even if the absence of specific inter-domain routing protocol (i.e. Multicast Source Delivery Protocol (MSDP) [29]) significantly reduces join and transmission latency for a receiver in remote domains.
- SSM has been designed to support streaming application (it is source specific). The trend of current on-line game architectures is to rely on one or multiple servers and SSM may support the communication between a server and multiple game participants connected to it

This paper proposes a protocol extension called Router Level Filtering (or RLF) that allows receivers to customize the data they receive to their own needs without the drawback of joining a new group. In RLF, we assume that a LSVE session relies on few stable groups and that a participant will have to execute fewer joins and leaves, these operations being not performed under critical time constraints. Instead, the receiver will ask the routers to "filter" data. This approach is faster than the classic join procedure. It is also independent of the multicast protocol architecture and works in both multiple senders and single sender environments. RLF also provides an elegant solution to flow and congestion control, as filter semantics belongs to the application domain being setup by the receivers themselves, while processed by the routers.

In the next section, the problem of multicast usage in on-line games and LSVE is discussed, providing insight into related work. In section 3, the RLF proposal is discussed along with specification and implementation details involved in the extension of the Linux IP stack. Section 4, evaluates RLF by means of a model for cost analysis of the router state in comparison to non-RLF approach. Finally, in section 5, some conclusions are drawn, along with future work.

2. MOTIVATION

An inherent requirement prompted by the multicast model is that receivers must have overlapping interest in the data that is sent to a particular multicast group, otherwise there is no advantage of sharing the same group membership. In multimedia streaming applications, the clustering of user interest is relatively easy, considering that sessions are time bounded and the interaction amongst participants is done in small groups with common goals. Consequently, SSM provides a simple and efficient multicast protocol architecture to satisfy the transmission requirements of streaming applications. In fact, most application architectures are based on a single sender model. However with LSVE and on-line games, where every participant is potentially a sender and receiver, the scale of the problem requires a less constrained model than the sender based.

LSVE and on-line games have different requirements in terms of data communication since the content is:

- Not systematically of interest to all session members (e.g. an avatar located in a specific room in a Quake [37]-like game does not need to receive information from avatars in other rooms).
- Continuously changing in time as the properties of avatars change in time (an avatar can lose his sword and replace it by a gun in a further phase of the game session).

Consequently, the use of multicast, as the supporting communication model, is forfeited if the receivers' interest is disjoint to the point of having groups with a singular membership. Therefore, multicast is an interesting communication paradigm in that it minimizes the transmission delay to all group members and it reduces the workload of the server when sending "customized data" to a group of participants that potentially share the same interest.

We have consequently designed an extension to the multicast forwarding model that provides receiver interest based content delivery using the multicast transmission paradigm. RLF was designed to allow the receiver to:

- Avoid the management of multiple multicast groups.
- Minimize the role of the sender in content "customization".
- Improve the customization process in order to maximize the real-timeliness of the application.

Our original motivation was LSVE and on-line games, but RLF can be used with any interactive content delivery application. We will consequently use either on-line games or LSVE to describe the addressed application range in the remainder of this paper.

2.1 Grouping Strategies

In [31] a framework is proposed, which provides a clear analysis of the problem regarding clustering receiver interest using multicast, as illustrated Figure 1.

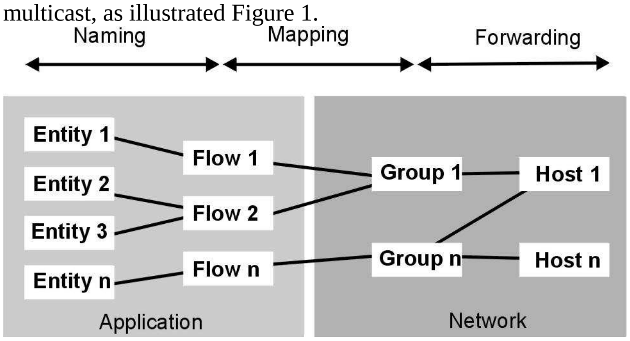

Figure 1 - Framework for clustering receiver's interest

This framework is composed of the following three components:

- **Naming** is exclusively of the application domain and defines the relation between the application data model (semantic) and the data flows identification (syntax).
- **Mapping** defines the correspondence between data flows and a multicast group.
- **Forwarding** is a network level component that aggregates the IP addresses of the receivers in one or more logical multicast addresses.

Traditionally the grouping mechanisms in LSVEs have been implemented at the naming level (host filtering) or at the mapping level (address filtering). These grouping mechanisms are commonly known as "*Area Of Interest Management*" (AOIM) and make LSVEs scalable in term of environment size and number of participants.

The existing AOIM mechanisms provide a technique to cluster participants based on various functional and spatial properties. These mechanisms generally fall into one of the following two categories (in both cases the environment is divided into a grid for indexing purposes):

- **Grid Based**. These clustering mechanisms are tightly coupled with spatial positioning of which grid cell is the entity in, such as [1,2,28,32,43]
- **Aura Based**. The clustering mechanisms rely more on entity spatial or functional proximity, such as [18,41].

The efficiency of either of these 2 approaches is difficult to determine as it depends on complex imbricated application and network metrics [38,49]. However, aura based enjoys extensive usage in small group interaction while grid based is most often found in DIS systems. In other clustering mechanisms, such as the matchmaker protocol [48], a more abstract approach is taken where no semantic specific context is attributed to the mapping. However, [48] requires the periodic re-organization of the session participants in clusters, which happens to be a NP-complete problem. However, all the mechanisms are constrained by the limitation of the address space [19,27], along with the join/leave latency times [17].

2.2 Related Work

We are not aware of many router filtering protocols or mechanisms, since router vendors are reluctant to increase the complexity of the routers with no clear application benefit. However, the PraGmatic Multicast (PGM) [43] protocol has been proposed by a major vendor to support reliable multicast protocol at the router level. Unfortunately, PGM is tightly coupled to the semantics of reliable multicast transmission. Parallel to PGM, a similar protocol in terms of functional objectives has been proposed by [36]. PGM scope has been further extended in BreadCrumb Forwarding Service (BCFS) [45] and in Generic Multicast Transport Services (GTMS) [5].

BCFS combines the source specific multicast service model (as in EXPRESS [21]) with PGM in order to provide a configurable subcasting mechanism. The receivers setup state at the routers by sending request messages towards a particular source containing a label that is semantic free. Unlike PGM, this approach allows the application to determine the usage of labels and their meaning. In addition, BCFS includes the notion of level numbers to support regeneration of the soft state at routers and to suppress unnecessary requests. Subcasting is then possible if receivers send periodically null requests towards a particular source with a specific label and a level number equal to one. The source, in turn, sends its data packets with the same label with level number equal to zero.

The GTMS proposal is a routing level mechanism that aims to provide a small number of fixed and simple service that requires minimal additional state to be handled in the routers. GTMS is based on a sender-based model, independently of the one used by the underlying routing protocol. The proposal requires the existence of GTMS objects along the forwarding path of a particular session (source, group address), which contain filter state that is modifiable by the set of operations exposed.

Another proposal is the Addressable Internet Multicast (AIM) [30] architecture which provides labeling capabilities to the routing tables, consequently embedding more information regarding to the forwarding process. The architecture provides a set of services based on different types of labels: distance, position and stream. AIM is the closer to RLF than the other protocols. However AIM is more complex to deploy in the routers than RLF as its scope is broader than the one of RLF.

3. ROUTER LEVEL FILTERING

The core assumption in our filtering approach is that LSVE users maintain interest in certain category (or group) of data for the duration of the session, or at least for a significant amount of time. This assumption is corroborated by the time expenditure necessary for users to complete goals while engaged in an application, as well as by the "nature" of the avatars (a human hardly becomes a vehicle, so it maintains an interest in the same type of data, e.g. voice, vision).

So, even if the participant's interest may vary considerably in the session, there will exist a common interest among users of the same "type".

Currently, because of the cost of group management, grouping techniques use a limited number of groups that consequently increases the amount of data unnecessarily delivered to group members. In that case, any change in the interest of a participant requires them to join and/or leave a multicast group. Even though from the application perspective, the host is no longer part of a particular group, it will continue to receive data packets until the multicast routing protocol effectively takes the leave into account. However, leaving a group is not a critical operation from a latency standpoint (IGMPv2 introduced explicit leaves). On the other hand, the joining a group introduces a significant latency due to the various group management and routing protocols involved. In PIM-SM, a receiver initially joins the Rendezvous Point (RP) and later joins the routing tree along the shortest path, so effectively two joins (and one prune) are made. The latency is even increased in an inter-domain scenario where the MSDP uses TCP to carry data to participants located in distinct domains.

During the time the join is processed, relevant data may be lost to the receiver. It is possible to limit these problems by implementing anticipation mechanisms (a participant joins a group in advance in order not to lose data at the time he will need them). Such an anticipation mechanism results in the reception of undesired data, additional state information, as well as complex synchronization problems.

3.1 Approach

RLF relies on the application to define flows of data among all data carried in a multicast group. Each sub-flow is allocated a flow identifier, which receivers express their interest by means of a filter. These filters are then pushed to the routers and propagated across the multicast routing structure. The Routers aggregate the receivers interest by combining into a single filter the aggregated interest of the receivers along a particular route. The forwarding of data at a particular interface is based on the associated filter, thus customizing the content delivered to the receiver interest. The application controls the semantics associated to the filter, thus embracing Application Layer Framing (ALF) [8] principles, rather than confining the subcasting capabilities to packet retransmission [44]. In this way, RLF introduces a filtering mechanism to the forwarding component of the framework presented in section 2.1.

The RLF proposal is not a multicast routing protocol, but an extension to any existing multicast routing protocol [3,10,11,35,46]. The simplicity of RLF allows it to scale independently of the topological arrangement of the receivers and to be easily integrated into an existing protocol architecture. RLF does not totally solve the problems encountered in classic multiple group management, however its main benefits are:

- RLF allows the application to utilize more stable groups, thereby reducing the number of multicast groups required to support a LSVE. This property increases the scalability of the system.
- RLF minimizes the join and leave latency, thus limiting problems related to participant fast change in content interest.
- RLF limits the amount of unnecessary data received by a session participant.
- RLF allows a receiver to have a refined selective mechanism to the existing data within a multicast group.

3.2 Protocol Overview

Unlike other router filtering approaches [44,46], which introduce a new multicast routing protocol, RLF is designed to work with any multicast routing protocol. RLF assumes that the multicast routing tree already exists and is significantly stable.

RLF requires an extension of the forwarding process at the routers to include packet filtering. The RLF subscribe/unsubscribe operation does not affect the multicast routing structure; consequently the operation is much faster than the join/leave mechanisms. In addition it provides the possibility of keeping multicast group memberships even though no traffic is forwarded to the receivers, thus foregoing the cost of systematically joining/leaving several times the same multicast group as is the case in LSVE. The illustration in Figure 2 shows RLF working with a tree based routing protocol, using a filter with only 2 bits corresponding to flows A and B. At each router, there is a filter per route indicating the aggregated interest of the receivers along a particular path. Consequently, there are two receivers with interest in flow A and one in flow B.

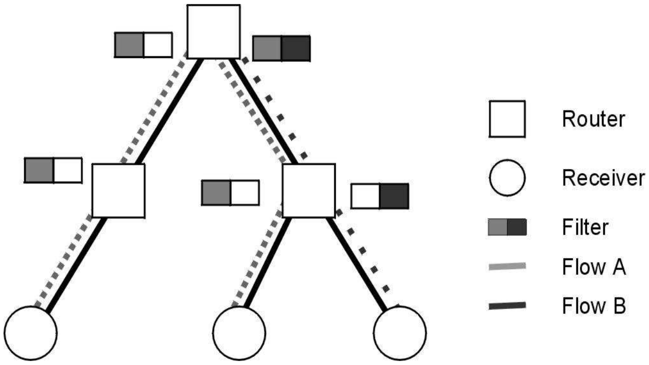

Figure 2 - Overview of RLF

As mentioned before, multicast group join and leave operations are not affected by RLF. Initially, the filter is considered null; therefore, although the receiver is part of the multicast group no packets will be received because the receiver is not subscribed to any flows. Note that this is a design choice and that the initial filter could be configured as “forward all”, which is the normal case of multicast, with no added complexity.

3.2.1 Filter and Flow Identifiers

In RLF, the adopted approach was to consider exclusive flow identifiers. This avoided the need of having a unique filter to identify each flow; rather, it was possible to share a filter amongst several flows, where each bit identified a separate flow. Another advantage of this design choice was the possibility of aggregating flows together without additional complexity to the searching and maintenance of the supporting structures at the routers.

It is necessary for every packet needs to carry a filter indicating the flows it belongs to. At each network element, host and router, a filter of aggregated interest is associated to each route. Based on this information incoming packets are either forwarded (or passed to transport layer) or discarded:

- In the case of a host, the filter represents the local interest of the application in particular flows. It is necessary to incorporate filters at the hosts because not all receivers, within the same LAN, are interested in the same flows.
- In the case of the router, filters are maintained per route, where each filter represents the aggregated interest of the receivers of a multicast group along that route.

The design properties of the filter provide the following advantages:

- The RLF mechanisms at the routers are based upon boolean operations, which incur low cost performance wise.
- Minimal state is required at the routers, considering that only one filter per route is necessary for a particular multicast group.
- The filtering process is highly simplified and efficient since no search mechanism is necessary to find if there is a receiver interested in the flow.
- It is possible to simultaneously send a packet to multiple flows by setting the appropriate bits of the packet filter.
- To add congestion control capabilities, filters can be ordered from highest priority to lowest priority so that a router can decide, with no knowledge of the application semantic, what filter to discard in case of congestion [45].

There is the disadvantage that the filter needs to have a fixed size (in order to allow the processing of the filter on the fast path of the router). This constraint places a significant limitation to the number of flows supported. However, independently of the size of the filter, the purpose of RLF it not to replace the need of having multiple multicast groups within an application but provide a more refined mechanism of delivering content to receivers based on their interest.

3.2.2 Filter Updates

The operation of the subscribe/unsubscribe mechanisms remains essentially algorithmically identical, although the semantic result is opposite in nature. In addition, the mechanisms operate in similar fashion whether the network element is a host or router, with the origin of the update request and triggered actions being different as illustrated in Figure 3.

```
Filter_Update(filter)
  Local_var newFilter;
  UpdateNewFilter(newFilter, currentFilter, filter);
  If (change in currentFilter) then {
      currentFilter := newFilter;
      Send appropriate IGMP or routing packet;
  }
```

Figure 3 - Algorithm for updating a RLF filter, either at the host or router.

After a receiver joins a multicast group it must indicate interest to receive traffic from at least one data flow. Towards this purpose, the receiver sends a packet to the first hop router indicating its interest. We have chosen to implement this packet as a new type of IGMP packet. The IGMP extension will be discussed later in the section 3.3 that is focused on implementation. When the router receives the IGMP packet, it updates the filter of the route on which the packet was received. If there is a change to the previous filter, then the new filter is forwarded on the path to the sender, otherwise no action is taken.

Filter updates are propagated throughout the multicast routing structure. The method by which the packet is propagated depends entirely upon the routing protocol being used. An example of the update process is illustrated in Figure 4, where one of the receivers of Figure 2 updates its interest filter by unsubscribing from flow A. Action 1 corresponds to the IGMP_UNSUBSCRIBE and action 2 represents the routing specific protocol packet corresponding to unsubscribe.

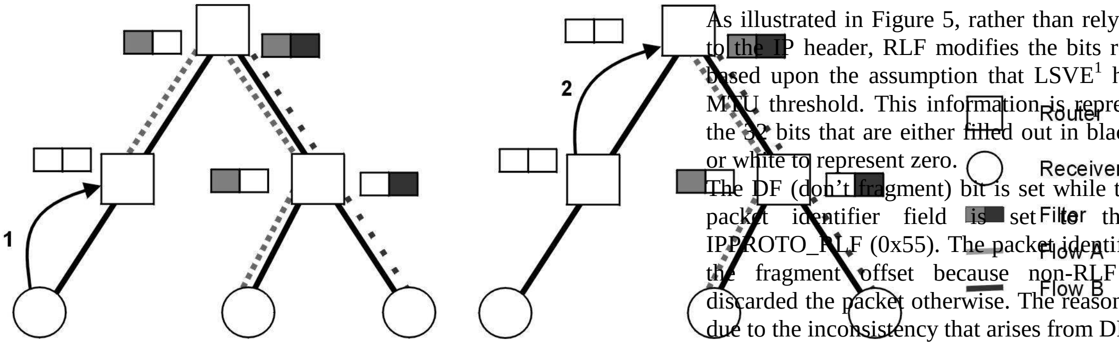

Figure 4 - Receiver unsubscribe from flow A

3.2.3 *Filtering*

Filtering happens at both the routers and hosts. When a router receives a packet identified as being RLF, it proceeds to do the usual forwarding based on the routing tables. If the forward process determines that the packet is to be dropped then no further action is taken. Otherwise, before the packet is forwarded along the relevant interfaces, the router validates the packet's flow identifier against the associated filter for each interface. The packet is forwarded only if validated against the filter, otherwise the packet is dropped.

In a host a similar filtering operation is done, not to determine if the packet should be forwarded but to decide if the packet belongs to one of the flows that are relevant for any local application.

3.3 Implementation

The driving aim of the implementation effort was to cause minimal impact on the existing IP stack of any operating system, which in this particular case turned out to be Linux. In the Linux operating system, as many others, there exists code to support filtering of IP packets, which is used by firewalls and packet filters [42]. This capability was not used in our implementation because the extensions required modification to the IP packet header.

We adopted a design approach that facilitates incremental deployment of RLF within a network, where it is irrelevant if the network elements are RLF aware or not. However, the first hop router is required to be RLF capable because RLF relies on a modification of IGMP.

3.3.1 *Packet Format*

Although AIM proposes a similar idea regarding the identification of streams within a multicast routing tree, the additional labeling functionality increases the complexity of the protocol. RLF proposes a very simple protocol that consists of a small addendum to the current multicast model. Thus, the IP header of an IP packet requires only minor modifications.

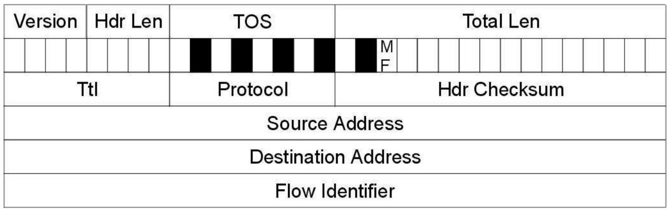

Figure 5- Modified IP header

As illustrated in Figure 5, rather than rely solely on an extension to the IP header, RLF modifies the bits related to fragmentation based upon the assumption that LSVE[1] have packets below the MTU threshold. This information is represented in the figure to the 32 bits that are either filled out in black to represent a bit set or white to represent zero.

The DF (don't fragment) bit is set while the lower byte from the packet identifier field is set to the protocol identifier IPPROTO_RLF (0x55). The packet identifier was used instead of the fragment offset because non-RLF routers would have discarded the packet otherwise. The reason for doing so would be due to the inconsistency that arises from DF being set.

In addition to these modifications, the IP header is extended to include 32 bits regarding the filter, which identifies all the flows to which the packet belongs. It is based on this information that the RLF routers filter the packets into flow.

Although RLF extends the IP header, the checksum is done excluding the information regarding the flow identifier so non-RLF routers do not reject the packet because of bad checksum.

3.3.2 *IGMP Extensions*

One of the core implementation goals was to make RLF as innocuous as possible, thus permitting co-existence with non-RLF routers. However, this is not possible with the first hop routers since the filter information would be lost when transitioning from IGMP to the routing protocol. Consequently first hop routers need to be RLF aware.

Two additional IGMP types were added to support the RLF subscribe (IGMP_SUBSCRIBE) and unsubscribe (IGMP_UNSUBSCRIBE) operations. In both cases, the IGMP packet includes the filter containing the flows. Whenever the receiver executes an operation, the corresponding packet is sent to the router. And the timer associated to the group is set for a random time. Once the timer expires, the packet is resent to the router. This redundancy ensures that the router receives the packet.

In the current multicast model, the router uses a soft-state [40] approach for maintaining information regarding group membership. Currently, the soft-state mechanisms are used solely for group membership based on membership queries and reports. A response to a query is delayed for a random time, with the first report suppressing the remaining ones.

The underlying premise of RLF is that the multicast groups are reasonably stable, not having to modify the associated routing structure. The soft-state mechanisms continue to exist, but are refocused to the maintenance of state concerning receiver interest. There is no need to have parallel mechanisms to keep group membership and flow subscription, since any packet of any flow indicates the existence of at least one receiver within the group. Consequently the existing query/report mechanism has been extended to accommodate filters.

The router refreshes the state of a filter associated to a particular group of an interface by sending a query for every flow that has registered interest. At most, the router sends as many queries as the number of bits that exist in the filter, which in our case corresponds to 32. A host only replies if they have an interest in

[1] In systems based on DIS protocol the packet size is less than 255 bytes, which happens to be the most common MTU on the Internet.

the flow. The reply is delayed by a random amount and the first response suppresses the other replies. If the router does not receive a reply for that particular flow then the filter is updated accordingly to reflect that no receiver is interested in the flow.

Although the router sends a query for every flow within a multicast group, the amount of traffic is not increased since the router does not make simultaneous queries. Therefore the periodicity of the queries is identical to IGMPv2. The query of the flows is done in round robin fashion.

3.3.3 RLF Host/Router Mechanisms

When a host joins a multicast group using a RLF socket, the convention adopted is that the host has no interest in any of the existing flows by having the filter set to zero. The host is required to use the subscribe mechanism to begin receiving packets from the relevant flows of interest. The fact that the filter has all flows inactive avoids the host receiving any data until it explicitly indicates their interest. To adopt any other convention would ultimately degenerate to the case of traditional multicast where all packets are received, even if the host was not interested in some of the flows. It is true, this approach incurs an additional step so the host becomes part of a multicast group. However, it is done very infrequently during the session, which reduces the impact of routing table manipulation, focusing on fast updates. These updates affect the filter information associated to each route and are carried out in identical fashion by both hosts and routers:

- **Subscribe** New_Filter := Current_Filter | Requested_Flow
- **Unsubscribe** New_Filter := Current_Filter & ~Requested_Flow

Before updating *Current_Filter* to the value of *New_Filter*, both the host and router do a XOR between both to validate if any bit has changed. In the case of the host, a new IGMP message is sent if there has been a change. While in the case of the router the process requires an additional check. For every outgoing interface[2] the router aggregates the receivers interest of the remainder interfaces, excluding the incoming. The inverse of the result is ANDed with the *Current_Filter* and if the final result is different from zero then a filter update is sent along the selected outgoing interface. It is possible to optimize the process by keeping an inbound and outbound filter per route, rather than just an outbound filter.

The filtering mechanism on the routers consists of a minor extension to the existing forwarding process of any multicast routing mechanism. When a RLF router receives a RLF packet, the normal forward routine is executed. However, before sending a packet off on a particular interface, the flow identifier is ANDed with the associated filter of the interface, therefore determining if the packet should be dropped or not.

3.3.4 Socket API Extensions

Although the socket API remains the same, the options available have been extended to support RLF. The information regarding the filter was passed along to the IP stack by replacing the padding bits of the addr_struct.

By default, every datagram socket will accept all multicast packets to which the host has group membership. This behavior ceases to occur, once the application subscribes to at least a single flow and hereafter the group is considered to be compliant to RLF. However, for a host to send packets according to flows, the socket must be RLF enabled by setting the socket option IP_SET_RLF.

Although the router may be made aware that potential interest in a multicast group exists along a particular route, all packets are discarded until at least one receiver explicitly subscribe to a flow.

3.4 Implementation Observations

Although the current implementation supports the ideas proposed by RLF, some details result from design idiosyncrasies that may have other alternatives.

3.4.1 IPv4 vs IPv6

The existing implementation adopted IPv4 as the IP stack, primarily due to its stability. However, RLF is an interesting application for the flow identifier field of the next generation of IP [26].

3.4.2 Filter Size

One of the main implementation decisions, which remains to be validated, is the size of the flow filter. Ideally, it would be convenient to have a variable sized filter, so each application would have a single multicast routing tree with as many data flows as necessary. However, this approach would compromise the deployment of RLF on the fast path of the router.

In reality, it is necessary to choose a fixed size for the flow filter. Unfortunately, there is not sufficient understanding of the application genre to stipulate what would be the optimal size. For the current implementation, a best estimation was made regarding the size of the flow filter, thus 32bits was adopted.

We claim that 32 flows is more than enough in the RLF context as RLF is not meant to eliminate the need for multiple multicast groups.

An alternative implementation would be to assume that each flow had a unique flow filter. While this approach would imply an increase in the number of flows (2^{32}), it would experience the same scalability problems experienced by AIM.

Yet another implementation approach would be to assume that only one flow existed per multicast group. In this case, the filter would act as a binary switch that the receivers used to start or stop receiving data from the multicast group. While this would certainly solve the issue of the most appropriate filter size, it presents drawbacks in terms of (state maintenance for all the trees), bandwidth consumption (IGMP traffic for all the trees) and abdicates functional benefits such as multiple flows.

3.4.3 Hierarchical Filters

The current RLF implementation has each filter correspond to a direct set of exclusive flows. This design decision places limitation on the number of flows supported by a single multicast routing tree. However, it is possible to increase the number of flows by making the filters hierarchical in nature with the introduction of a tuple filter (flow, subflows)[3]. So for example, taking the 32 bit filter and assigning the initial two bytes to represent the flow it would be possible to have 16 flows, each with 16 subflows, totaling 272 exclusive subflows. The additional logic to support the hierarchical filtering would be minimal

[2] The incoming interface is the one through which the subscribe/unsubscribe request was received.

[3] This is similar to other indirection mechanisms such as virtual memory in computer memory architectures.

without compromising the scalability of RLF. Once again, it is not possible to determine if such a number is ideal for either gaming or LSVE applications.

4. PRELIMINARY ANALYSIS OF RLF

RLF permits the distinction of several flows within a single multicast group. It requires additional information in the router tables to indicate the aggregated interest of the receivers along a particular route. This implies that each multicast route entry has associated a filter.

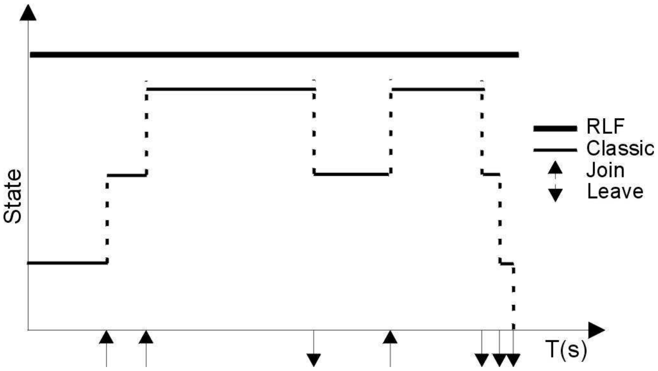

Figure 6 - Variation of the router state during a multicast session

In the classic approach, illustrated in Figure 6, the memory necessary to maintain the state of the multicast tree varies along time, as receivers join and leave the group. In the case of RLF, the memory consumption is assumed to be constant since the routing structure is expected to be more stable. Consequently, the classic approach will at most use the same amount of memory as RLF (detracted by the filter size), depending upon the nature of the application or particular situations. This is true when disregarding the fact that RLF comprises within a single multicast group several data flows (in our particular case 32). So it is possible that, depending upon the nature of the application, RLF saves memory when compared to classic multicast.

For the cost analysis of the router memory consumption in both RLF and classic multicast a simplified model was conceptualized with the following elements:

A Size of the IP address.

R Size of the structure corresponding to a route

F Size of the filter

G(t_i) Number of active multicast groups at t_i.

M_j(t_i) Number of routes that belong to a particular group j for t_i.

S(t_i) Total state router memory state at t_i

To compare RLF with the classic approach it is necessary to calculate the expected value of the total state in either case. The total state may be approximately found with (1).

(1) $S(t) = G(t_i)\big(A + M_j(t_i)\big)$

It is important to note, that both G(t_i) and M_j(t_i) are independent, so the resulting expected value in the classic approach is expressed as (2).

(2) $E[S_C] = E[G] \times (A + E[M_j] \times R)$

With RLF, it is important to remember that each multicast routing tree aggregates several multicast groups, thus (1) is slightly modified (3). Also, each route has associated to it a filter.

(3) $E[S_{RLF}] = \dfrac{E[G]}{F} \times (A + E[Mj] \times (R + F))$

Considering that in RLF all receivers are part of the tree, then all routes at a router belong to the routing tree. This implies that M_j(t_i) is always a constant "n" equal to the number of existing routes (4).

(4) $E[S_{RLF}] = \dfrac{E[G]}{F} \times (A + n \times (R + F))$

With both (4) and (2) it is possible to evaluate the cost between RLF and classic multicast in a relative manner, obtaining (6), which demonstrates that the number of groups is irrelevant for comparative analysis.

(5) $\dfrac{E[S_C]}{E[S_{RLF}]} > 1$

(6) $E[M_j] > \dfrac{A \times (1 - F) + n \times (R + F)}{R \times F}$

With (6) it is possible to plot a graph by varying the size of F and n, consequently obtaining the threshold of the average number of routes that are part of a multicast tree. The result is the line char illustrated in Figure 7, where the x-axis represents the total number of routes that are available in a router. The filter size ranges from 2 to 32 bits.

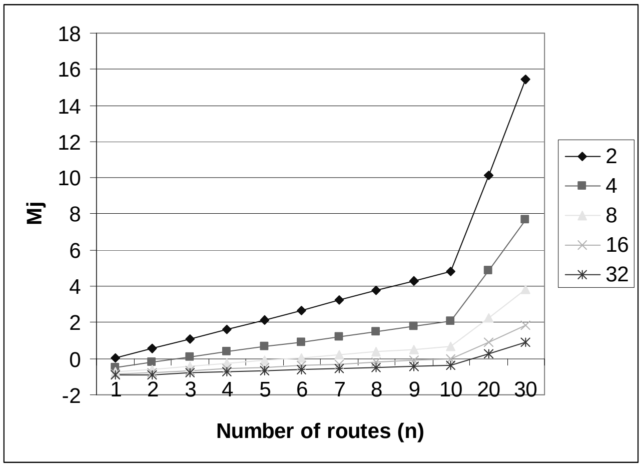

Figure 7 - Threshold Mj when the cost of RLF is less than classic multicast

The graph in Figure 7 indicates the threshold above which RLF saves state when compared to traditional multicast. In the case of a router where there exists ten interfaces (or routes) and a filter of four bits is used, RLF will save memory once the average number of routes that are part of the multicast group during a session is above two. This obviously depends upon the application, but considering that the target applications are real-time with high interaction amongst users the probability of M_j exceeding the threshold is very high.

4.1 Join Latency

In [17], join latency is defined in two ways. The first corresponds to the time it takes for the first packet to arrive at a receiver after sending an IGMP join packet. The second corresponds to the time it actually takes for a receiver to be join the multicast group via the nearest router. We adopt to analyze join latency according to the second definition.

With any multicast routing protocol, the join latency at any given network element (host or router) may be given as:

$$T_{join} = \sum_{i=1}^{n} \left(T_{prop\_i} + T_{process\_i} \right)$$

Where T_{prop} represents the time it takes for a packet to arrive at the next hop and $T_{process}$ represents the time it takes for the next hop router to process the join request. The latency increases, as more hops are necessary until reaching the nearest multicast router that is member of the multicast group. The Tprocess depends upon the processing time of the routing protocol and it is possible to verify that RLF improves it.

For our purposes we will consider the case of Core Based Trees (CBT) [3] routing protocol, thus (1) will yield:

$$T_{CBTjoin} = \sum_{i=1}^{n} \left(T_{propagation\_i} + \left(T_{lookup\_i} + T_{create\_i} \right) \right)$$

Where the T_{lookup} is the time it takes for the router to lookup the corresponding core associated to the multicast group and T_{create} represents the time it takes to create the necessary forwarding state. With RLF the receiver is already part of the CBT routing tree, but it is necessary to account for the latency involved in subscribing their interest, thus:

$$T_{RLFsubscribe} = \sum_{i=1}^{n} \left(T_{propagation\_i} + T_{update\_i} \right)$$

Intuitively, $T_{lookup} + T_{create} >> T_{update}$, thus the latency of the subscribe mechanism of RLF is less than the join mechanism of CBT. The difference becomes more evident as the number of hops increases.

4.2 RLF Traffic Overhead

The extensions proposed by RLF do not generate additional traffic to support its usage. In our implementation, neither the modifications to IGMP or the integration of RLF into the routing daemon proved not to require any additional control packets. Therefore no additional traffic overhead is generated.

In reality, in some cases there is a reduction of control traffic in the case of receivers leaving a group to join another. This operation involves two control packets (leave and join), unlike RLF that only requires a single packet to reflect the filter update.

4.3 Experimental evaluation

Initial experiments has been initiated to compare effectively the impact of RLF in terms of latency and consequently the superfluous data discarded by receivers. However, at the time of writing the experimental results were not available for analysis.

5. CONCLUSIONS AND FUTURE WORK

The RLF proposal requires a small modification to the current multicast model in the forwarding process. The filtering functionality is not novel, considering that packet filters, such as firewalls, are based on this operation. In fact, multicast routing filtering protocols do exist, albeit their constraints in terms of operational functionality. However, unlike other routing filtering approaches, RLF is based on the extension of what exists and delegates the application with the responsibility of choosing the filter semantics.

RLF enhances the forwarding process in multicast to improve the granularity of content delivery to receivers and the associated latency when joining/leaving a multicast group. Although RLF requires modifications to the routers, it is possible that some of the functionality may be available through General Router Assist [6], which has the support of the vendors.

We have demonstrated that RLF either benefits or is non-detrimental to any routing protocol in terms of router state, join latency and traffic overhead. Currently, experimental trials are being carried out to compare the timings of subscribe/unsubscribe with the traditional join/leave mechanism. In addition to latency, the amount of superfluous data, which receivers get during the transition phase of their change of interest, is another important result to be analyzed.

The source code will be made available on the web once the code has been fully debugged and the experimental analysis concluded.

6. ACKNOWLEDGMENTS

The authors would like to thank the insightful and detailed comments given by the reviewers.

7. REFERENCES

[1] H. Abrams, K. Watsen and M. Zyda, “Three Tiered Interest Management for Large Scale Virtual Environments”, Proc. VRST’98, September 1998

[2] D. Anderson, J. Barrus, J. Howard, C. Rich and R. Waters, “Building Multi-User Interactive Environments at MERL”, IEEE Multimedia, Winter 1995

[3] A. Ballardie, P. Francis and J. Crowcroft, “Core Based Trees (CBT): An Architecture for Scalable Inter-domain Multicast Routing”, Proc. ACM SIGCOMM’93, October 1993

[4] J. Byers, M. Luby, M. Mitzenmacher and A. Rege, “A Digital Fountain Approach to Reliable Distribution of Bulk Data”, Proc. ACM SIGCOMM’98, September 1998

[5] B. Cain and D. Towsley, “Generic Multicast Transport Services: Router Support for Multicast Applications”, CMPSCI Technical Report TR 99-74, October 1999

[6] B. Cain, T. Speakman and D. Towsley, “Generic Router Assist (GRA) Building Block – Motivation and Architecture”, IETF Internet Draft, March 2000

[7] B. Cain, S. Deering, I. Kouvelas and A. Thyagarajan, “Internet Group Management Protocol, version 3”, IETF Draft June 2000

[8] D. Clark and D. Tennenhouse, “Architectural Considerations for a New Generation of Protocols”, Proc. ACM SIGCOMM’90, September 1990, pp. 200-208

[9] S. Deering, “Multicast Routing in a Datagram Internetwork”, PhD Thesis, Stanford University, December 1991

[10] S. Deering et al, “An Architecture for Wide-Area Multicast Routing”, Proc. ACM SIGCOMM’94, 1994, pp. 126-135

[11] S. Deering, D. Estrin, D. Farinacci, V. Jacobson, G. Liu and L. Wei, “PIM Architecture for Wide Area Multicast Routing”, IEEE/ACM Transactions on Networking, April 1996

[12] C. Diot and L. Gautier, “A Distributed Architecture for Multiplayer Interactive Applications on the Internet”, IEEE Networks Magazine, vol. 13, N.4, July/August 1999

[13] C. Diot, B. Levine, B. Lyles, H. Kassem and D. Balensiefen. "Deployment Issues for the IP Multicast Service and Architecture". IEEE Network magazine special issue on Multicasting. January/February 2000

[14] H. Eriksson, "MBone: The Multicast Backbone", Communications of the ACM, Vol. 37, August 1994

[15] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification, June 1998, RFC-2362

[16] E. Frecon, C. Greenhalgh and M. Stenius, "The DIVEBONE – An Application-Level Network Arcgitecture for Internet-Based CVEs", Proc. VRST'99, December 1999

[17] A. Garg, S. Kasera, R. Kumar and D. Towsley, "Measurement of Join Latency on the MBone", CMPSCI Technical Report TR 99-47, August 1999

[18] C. Greenhalg and S. Benford, "MASSIVE: A Collaborative Virtual Environment for Teleconferencing", ACM Transactions on Computer Human Interfaces, Vol. 2, N. 3, September 1995

[19] M. Handle, D. Thaler and D. Estrin, "The Internet Multicast Address Allocation Architecture", IETF Internet Draft, December 1997.

[20] O. Hodson, S. Viarakliotas and V. Hardman, "A Software Platform for Multi-Way Audio Distribution over the Internet", Proc. Audio and Music Technology: The Challenge of Creative DSP, November 1998

[21] H. Holbrook and D. Cheriton, "IP Multicast Channels: EXPRESS Support for Large-Scale Single-Source Applications", Proc. ACM SIGCOMM'99, September 1999

[22] H. Holbrook and B. Cain, "Source Specific Multicast", IETF Internet Draft, March 2000

[23] D. Hook, S. Rak and J. Calvin, "Approaches to Relevance Filtering", Proc. 11th DIS Workshop, September 1994

[24] IEEE Standard for Distributed Interactive Simulation – Application Protocols, IEEE std. 1278.1-1995, IEEE Computer Society, 1995

[25] IEEE Standard for Distributed Interactive Simulation – Communication Services and Profiles, IEEE std. 1278.2-1995, IEEE Computer Society, 1995

[26] S. King, R. Fax, D. Haskin, W. Ling, T. Meeham, R. Fink and C. Perkins, "The Case for IPv6", IETF Internet Draft, October 1999

[27] K. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin and M. Handley, "The MASC/BGMP Architecture for Inter-Domain Multicast Routing", Proc. SIGCOMM'98, September 1998

[28] E. Lety and T.Turletti, "Issues in Designing a Communication Architecture for Large Scale Virtual Environments", Proc. NGC'99, November 1999

[29] D. Farinacci, D. Meyer, P. Lothberg, H. Kilmer and J. Hal, "Multicast Source Delivery Protocol", IETF Internet Draft, February 2000

[30] B. Levine and J. Aceves, "Improving Internet Routing with Routing Labels", Proc. IEEE International Conference on Network Protocols, October 1997

[31] B. Levine, J. Crowcroft, C. Diot, J. J. Garcia-Luna-Aceves, and J. F. Kurose. "Consideration of Receiver Interest in Delivery of IP Multicast", Proc. Infocom 2000, March 2000

[32] M. Macedonia, M. Zyda, D. Pratt, D. Brutzman and P. Barham, "Exploiting Reality with Multicast Groups", IEEE Computer Graphics and Applications", Vol. 15, N. 5, September 1995

[33] S. McCanne and V. Jacobson, "VIC: A flexible Framework for Packet Video", Proc. ACM Multimedia, 1995

[34] S. McCanne, V. Jacobson and M. Vetterli, "Receiver-Driven Layered Multicast", Proc. ACM SIGCOMM, August 1996

[35] J. Moy, "Multicast Extensions to OSPF", IETF RFC 1584, March 1994

[36] C. Papadopoulos, G. Parulkar and G. Varghese, "An Error Control Scheme for Large-Scale Multicast Applications", Proc. IEEE INFOCOMM'98, 1998

[37] www.quakeworld.com

[38] S. Rak and D. Hook, "Evaluation of Grid-Based Relevance Filtering for Multicast Group Assignment", Proc. 14th Workshop, March 1996

[39] S. Raman and S. McCanne, "Generalized Data Naming and Scalable State Announcements for Reliable Multicast", Proc. ACM Multimedia'98, September 1998

[40] S. Raman and S. McCanne, "A Model, Analysis and Protocol Framework for Soft State-Based Communication", Proc. ACM SIGCOMM'99, September 1999

[41] D. Roberts, B. Worthington and P. Sharkey, "Influence of the Supporting Protocol on the Latencies Induced by Concurrency Control within a Large Scale Multi-User Distributed Virtual Reality System", Proc. VWSIM'99, January 1999

[42] S. Satchell and H. Clifford, "Linux IP Stacks: Commentary", Coriolis, 2000

[43] S. Singhal and D. Cheriton, "Using Projection Aggregations to Support Scalability in Distributed Simulation", Proc. ICDCS'96, 1996

[44] T. Speakman, D. Farinacci, S. Lin and A. Tweedly, "Pragmatic General Multicast", Tech. Report, Internet Draft, January 1998

[45] L. Vicisano, L. Rizzo and J. Crowcroft, "A TCP-like Congestion Control for Layered Multicast Data Transfer", Proc. INFOCOMM'98, March 1998

[46] K. Yano and S. McCanne, "The Breadcrumb Forwarding Service: A Synthesis of PGM and EXPRESS to Improve and Simplify Global IP Multicast", Computer Communications Review, Vol.30, N.2, April 2000

[47] D. Waitzman, C. Partridge and S. Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, November 1988

[48] T. Wong, R. Katz and S. McCanne, "A Preference Clustering Protocol for Large-Scale Multicast Applications", Proc. NGC'99, November 1999

[49] L. Zou, M. Ammar and C. Diot, "An Evaluation of Grouping Techniques for State Dissemination in Networked Multi-User Games", submitted for publication