

## Overhead in Multicast Guillermo Rigotti

UNICEN – Fac. de Ciencias Exactas  
ISISTAN – Grupo de Objetos y Visualización  
Pje. Arroyo Seco, (7000) Tandil, Bs. As. Argentina  
TE: +54-2293-440363 FAX: +54-2293-440362  
Email: grigotti@exa.unicen.edu.ar

### Abstract

*A class of multicast applications, such as reliable multicast, consists of a sender and a set of receivers sending feedback information to the sender and the rest of the receivers.*

*If multicast support is based on source based trees, the above feature can cause an excessive overhead, specially when number of receivers increases (in the order of thousands). The reason is the need to maintain a distribution tree per receiver.*

*This paper presents an extended use of source based trees. It enables distribution of information originated by members of the group through the tree rooted at the sender. This aims to reduce the amount of memory in routers as a consequence of maintaining only one distribution tree per application.*

Keywords: Multicast Routing, Multicast support.

### 1. Introduction

Over the last decade, IP multicast has experienced a notable growth. The deployment of the MBONE has lead to the development of both, a variety of multicast applications each with its proper requirements and a set of multicast routing protocols based on different paradigms.

The first and most deployed multicast routing protocol is DVMRP [1][2], widely tested over the MBONE and other multicast-capable networks. Currently, several multicast protocols exists, each designed to cope with certain characteristics of applications and groups, aiming to offer better quality of service and diminish network resources consumed.

The aspect to be considered in this paper is the type of distribution tree established by the protocol.

Distribution of multicast packets is achieved by creating and maintaining distribution trees, composed of links and routers, and spanning all subnetworks with group members present. These trees enable distribution of multicast packets to all members of the group.

Source based protocols build a multicast delivery tree for each sender to a group. This tree is rooted at the source and spans all members of the group. Flow of packets is unidirectional, from root to leaves.

Shared tree based protocols build a multicast delivery tree shared by all senders to a group. This tree is rooted at a node called “core” or “rendevouz point” and spans all members of the group.

There are two kinds of shared trees: unidirectional and bidirectional. In the former, senders encapsulate multicast packets addressed to the group and send them (unicast) to the core. The core then decapsulates the packet and injects it into the multicast delivery tree. Multicast packets travel from the root towards the leaves.

In bidirectional shared trees, a sender<sup>1</sup> injects multicast packets directly into the distribution tree. Then the packet is distributed in all branches of the tree.

To be deployed in the Internet, a multicast routing protocol would satisfy different types of requirements.

From the application viewpoint, a routing protocol must provide quality of service, such as assured bandwidth, end-to-end delay, etc.

---

<sup>1</sup> This is the case of a sender in a network belonging to the distribution tree. In other case, the sender proceeds as described in shared unidirectional trees.

From the network viewpoint, a routing protocol must try to reduce the amount of network resources involved in its operation.

These requirements, in particular network resources consumed, topic of this paper, can be better satisfied through collaboration between applications<sup>2</sup> and the network level.

These interactions between network and application level don't imply loose of transparency of the multicast infrastructure with respect to the application level.

The interactions must be considered in a situation where a general API is provided and the network level maps application requirements, if possible, to operational characteristics of multicast routing protocols, transparently to the application.

### Solution Overview

With the support provided by this new multicast functionality at network layer and a stable API, a receiver application (M) member of a group (G), can solicit to the network level to send control (or feedback) information addressed to the sender of the group (S) and the rest of members.

Multicast packets generated by the receiver application must have addresses S and G, it would be generated by source S. These packets are encapsulated in special packets at the originating host (where the receiver application resides).

Those packets will traverse the routers belonging to the distribution tree in direction M to (leaf to root from the point of view of the distribution tree rooted at S).

Each special packet (identified by a new option) received by an in-tree router<sup>3</sup> is sent to the next (upstream) in-tree router towards the source. In addition, the original multicast packet is decapsulated and distributed in the subtree rooted at the router as any other packet originated by the source.

---

The word 'application' refers to multicast support at higher level (over UDP), such as reliable multicast transport protocols, for example.

An in-tree router is a router that belongs to the distribution tree.

Applications must be able to distinguish multicast packets originated by S from those

100	NEW_OPT (5)	LENGTH (8)	T (2)	TTL (2)	RESERVED (12)
DOWNSTREAM-ROUTER (32)					

Figure 1. New IP Option

originated by members of the group. Since both classes of packets carries identical addresses (S and G), applications must have some other mean to identify the source of a packet, for example use a port for data packets and another for feedback information.

The implementation of the new functionality does not require significant changes in router's code. As in mtrace [3], a packet is sent unicast for each router to its upstream, which is determined using any information available in the router.

The principal advantage of this approach can be appreciated in applications with a source and many receivers sending feedback information intended for the source and the rest of the receivers.

For example, in the case of an RTP [4] source sending information to thousands of receivers, each receiver must send periodic reports to the source. These reports must be received also by each receiver, in order they be able to adapt its sending rate based on the volume of feedback information.

In case of a source based multicast routing protocol, a distribution tree for each receiver would be needed, causing a significant overhead in routers and links.

If the new functionality is provided, only one distribution tree is needed, rooted at the source and spanning all the receivers. This tree is used to send both, data from the source and feedback originated at receivers.

In addition, feedback packets travel through the distribution tree following the inverse path of information packets. This topological feature can be exploited by multicast transport protocols that need to reduce exposure or localize

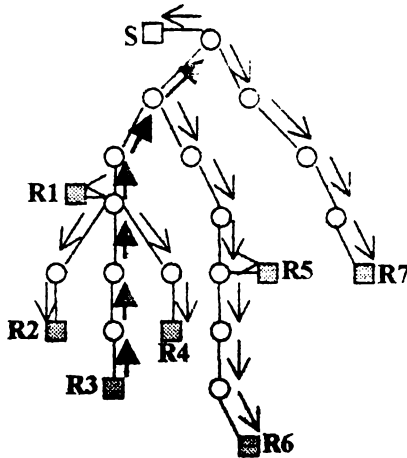


Figure 2. A "reversed" packet originated for receiver R3 is sent through distribution tree for group G, rooted at S. Routers are indicated as circles. Source S and receivers Ri by squares. Thick arrows indicate "reversed" packets, traveling through (S,G) tree in direction leaf-root towards S. Thin arrows represent decapsulated multicast (S,G) packets, originated at R3 and injected in the distribution tree for the corresponding router.

processes to subtrees, such as reliable multicast and RTCP translators.

### 3. The New IP Option

Packets that encapsulates feedback information must receive special treatment in the routers they traverse and need to carry certain control information. The way this may be done is defining a new IP option [5] and specify that values into appropriate fields.

Figure 1 illustrates the new IP option. Its fields are the following:

- Option Type:
  - Copied Flag = 1
  - Option Class = 0
- Option number = To be defined: Indicates the packet needs special process in the router.
- Option Length: 12
- T (Type): Type of packet:
  - 00: (reversed): Indicates the packet is sent in the direction leaf-root in the distribution tree corresponding to the addresses of the encapsulated multicast packet...
  - 01: (normal): Indicates the packet is sent in the direction root-leaf in the distribution tree

corresponding to the encapsulated multicast packet

10: Reserved.

11: Reserved.

TTL: Indicates the value that a decapsulating router will give to TTL field of the multicast packet before injects it into the distribution tree:

00: Decapsulate multicast packet and preserve its original TTL value.

01: Set TTL value of multicast decapsulated packet to the remaining value of original packet's TTL.

10: Reserved.

11: Reserved.

- Downstream router: Address of router that has sent the packet. This address correspond to the interface through the packet was sent.

### 4. Operation

Figure 2 illustrates the operation of multicast routers. In the figure, a source-based tree rooted at S is shown. Routers are indicated as circles, and group members (receivers) as squares.

Each receiver sends feedback or control packets that must be received by source S and the rest of receivers.

Information originated by the source (not shown in the figure) consists of multicast packets with source and destination addresses S and G respectively. These packets are delivered normally in the distribution tree.

Thick arrows represent "reversed" packets originated by the IP level at a receiver (in this case R3). These packets are generated in response to application requirements and encapsulate a multicast (S, G) packet with application level feedback information that must be distributed to the source and the rest of receivers.

The packets are delivered unicast from an in-tree router to another in-tree router towards the root (S). Any information available in the router is used to determine the previous hop towards the source.

Each "reversed" packet finally arrives at S or is silently discarded by a router in case its TTL value reaches 0.

Upon receiving a "reversed" packet containing the new IP option, in addition to propagate it towards the source, a router must decapsulate the original (S, G) addressed

unicast packet generated by the application, and distribute it in the corresponding sub-tree.

Since these packets carry addresses S and G, they can't be differentiated from multicast packets originated by the source S, and are distributed normally through the subtree rooted at node. They are represented as thin arrows in figure 2.

Note that decapsulated packets are actively distributed for the decapsulator router, and avoids sending them over the interface associated to the downstream sender of the encapsulating packet.

A host member of the group receives the decapsulated packet from its designated router, in the same way as another router does. It can be observed in figure 2, in the case of host R1 and, in particular in the case of S.

The interaction between the host where the receiver application resides (in this case host R1) and its designated router requires further consideration. Since the host hasn't necessarily knowledge of the unicast address of the designated router in the LAN, packets must be addressed to "all-multicast-routers-in-this-net" (224.0.0.2). This way, packets will be received by all routers in the local subnet, but addressed only by the designated router. Other routers in the subnet discard these packets to avoid generation of duplicates.

In the following situations, "normal" packets travelling in direction root to leaf must be treated:

- 1- A router has no means to determine over which interfaces propagate the decapsulated packet<sup>4</sup> or,
- 2- The reversed packet has arrived through a multiaccess interface.

In both cases, the router generates a "normal" packet and distributes it instead the original (decapsulated) multicast packet, delegating the responsibility for the distribution of the packet to each downstream router.

A "normal" packet is sent in the opposite direction (root to leaf) to the "reversed" packet.

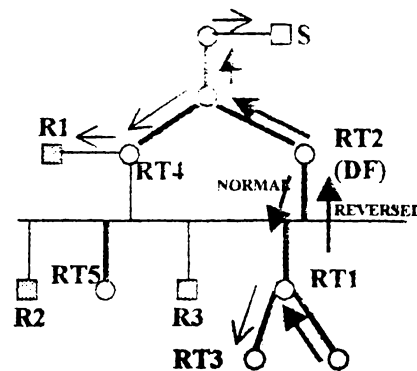


Figure 3 Multiaccess operation. Links belonging to the distribution tree (only those between routers) are represented by thick lines. Reversed packets containing encapsulated multicast ones carrying feedback are represented by thick arrows. Multicast (decapsulated) packets are represented using thin arrows.

Figure 3 illustrates a multiaccess network with two hosts members of group G: R2 and R3, and four routers: RT5, RT4, RT2 (designated router) and RT1. RT1 receives a "reversed" packet from its downstream router. Its responsibilities consist in decapsulate the multicast packet, and distribute it in the corresponding subtree (to RT3) and propagate the "reversed" packet towards the source S.

RT1 sends a "reversed" packet addressed to RT2, who has the responsibility of injecting the encapsulated packet into the multiaccess network.<sup>5</sup>

If RT2 decapsulates the multicast packet and injects it in the distribution tree (with addresses S, G), it will be accepted by R2, R3 and RT5 and RT1, who has no means to know it is a duplicate.

To avoid this situation, RT2 must change the "reversed" packet into a "normal" packet and multicasts it with addresses S, G. This packet will be accepted by hosts R2 and R3, and routers RT1 and RT5. RT1 knows that the packet is a duplicate because its address matches with "downstream router" field of the IP option, and discards it.

RT5, R2 and R3, decapsulate the multicast packet and process it as a normal S, G packet.

<sup>4</sup> This situation cannot occur in PIM-DM or DVMRP, because of exchanging of neighbor information.

<sup>5</sup> Note that a "reverse" packet originated by a local host, would be sent to a multicast address "all routers", but processed only by RT2.

## 5. Conclusions and Future Work

The extension to IP multicast presented enable applications with many receivers sending feedback information to interact with the network level to diminish the amount of network resources consumed in case of source based distribution tree multicast support.

This feature does not mean loose of transparency of multicast infrastructure to the applications. It must be viewed in a context of a set of multicast support oriented functions that, through a general multicast API, enables cooperation of applications and multicast support. The goals of this cooperation are to offer multicast quality of service and reduce the use of resources based on the combination of application and multicast support characteristics.

~~The main advantage of the proposed approach is to reduce network resources consumed, is clear. This must be contrasted with the overhead in routers due to decapsulation and modification of "reversed" packets to be sent upstream. This aspect is currently under consideration.~~

Additional advantages, for example its use to localize process to subtrees and avoid exposure, are currently considered and will be evaluated through simulation [6].

Another aspect to analyze is how the presented approach fits in reliable multicast approaches such as PGM [7] or OTERS [8].

## 6. References

- [1] S. Deering, C. Partridge, D. Waitzman, "Distance Vector Multicast Routing Protocol", RFC 1075, November 1988. <http://www.isi.edu>
- [2] T. Pusateri, "Distance Vector Multicast Routing Protocol, version 3", Internet draft, March 1998. <draft-ietf-idmr-dvmp-v3-06.txt>.
- [3] W. Fenner, S. Casner, "A 'traceroute' facility for IP Multicast", Internet Draft, August 1998. <draft-ietf-idmr-ipm-03.txt>.
- [4] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.
- [5] J. Postel, "Internet protocol", RFC 791. September 1981
- [6] K. Varadhan, K. Fall, "Ns Notes and Documentation", (VINT Project), UC Berkeley, May 1998. <http://www.isi.edu/~kannan/nsDoc.ps.gz>
- [7] T. Speakman, D. Farinacci, S. Lin, A. Tweedly, "PGM Reliable Transport Protocol Specification", Internet draft, <draft-speakman-pgm-spec-02.txt>, August 1998.
- [8] D. Li, D. Cheriton, "OTERS (On-Tree Efficient Recovery using Subcasting): A Reliable Multicast Protocol", Department of Computer Science, Stanford University, 1998.