Steps towards Native IPv6 Multicast: CastGate Router with PIM-SM Support

Tudor Blaga, *Student Member, IEEE,* Virgil Dobrota, *Member, IEEE* , Kris Steenhaut, *Member, IEEE*
Ionut Trestian, Gabriel Lazar

***Abstract*—The paper presents a testbed for native IPv6 multicast and tries to evaluate the available routing protocols such as PIM-DM and PIM-SM. From an objective perspective we can measure the join latency, as well as multicast forwarding delay, its jitter and the control overhead. Having a calibrated software tool, we can evaluate later the existing alternative solution towards IPv6 multicast: CastGate and CastGate Router. The proposed enhancements are referring to PIM-SM support and to the possibility of migrating from IPv4 to IPv6 on CastGate architectures.**

Index Terms— **CastGate, Communication system routing, Protocol-Independent-Multicast**

I. INTRODUCTION

THERE are three versions of IGMP (Internet Group Management Protocol) used in multicast networks. The first one (IGMPv1) is working in IPv4 only and uses two types of messages: membership query and membership report. A host sends IGMP membership report corresponding to a particular multicast group to indicate that it is interested in joining it. On the other hand a router periodically sends IGMP membership query messages to verify that at least one host of the given subnet is interested in multicast traffic. IGMPv2 (its IPv6 version is called MLDv1 (Multicast Listener Discovery)) is an enhanced version and it includes a few extensions. Among them there is a procedure for the election of the multicast query device for each LAN, explicit leave messages for faster pruning and group specific query messages. IGMPv3 (its IPv6 version is called MLDv2) permits a host to join a group and to specify the sources of that group from which it wants to receive multicast. This feature is called source filtering.

To deliver traffic to all receivers, multicast-capable routers create distribution trees. The simplest way of providing multicast routing is by flooding. If a multicast router receives a multicast packet for the first time, it forwards this packet to all the outgoing interfaces except the one from which it receives it. This solution is very inefficient in terms of network bandwidth utilization. A better solution is to build a spanning tree where a multicast router could forward multicast packets to all the interfaces that are part of a multicast tree except the source. Multicast forwarding algorithms can be classified in two categories: source-based (shortest path tree) and core-based (shared tree).

A source-based tree has its root at the source and branches forming a spanning tree through the network to the receivers. The source-based tree has the drawback that it is dependent on the source of the multicast tree; it must be computed separately for each source.

Core-based tree algorithms need a single common root placed at a chosen point in the network, but the root is not at the source. The disadvantage of core-based trees is that under certain circumstances the paths between the source and receivers might not be the optimal. Therefore the placement of the core router must be carefully considered.

Source-based trees, as well as core-based trees can be constructed using RPF (Reverse Path Forwarding). The idea of RPF is the following: by giving the address of the tree's root, a router selects its upstream neighbor, i.e. the next-hop router for forwarding unicast packets to the root. The network interface used to reach this upstream neighbor is called the RPF interface.

RPF tells each router the upstream neighbor in the distribution tree, but not the downstream neighbors, so additional protocol mechanisms are needed to determine the outgoing interfaces. One method to achieve this is *flood-and-prune,* which starts by forwarding multicast packets on all its interfaces, and then deletes interfaces which are not part of the distribution tree. Another method, called *explicit join,* requires that multicast receivers initiate the process of

Manuscript received May 16, 2005 and reviewed August 8, 2005. This work was supported in part by the Technical University of Cluj-Napoca under Socrates-Erasmus 2004-2005 mobility Grant (Coordinator Professor Virgil Dobrota) at Vrije Universiteit Brussel (Promotor Professor Kris Steenhaut).

Tudor Blaga is with the Technical University of Cluj-Napoca, 400020 Constantin Daicoviciu 15, Cluj-Napoca, Romania (phone: +40-264-401816; fax: +40-264-597083; e-mail: Tudor.Blaga@ com.utcluj.ro).

Virgil Dobrota is with the Technical University of Cluj-Napoca, 400020 Constantin Daicoviciu 15, Cluj-Napoca, Romania (e-mail: Virgil.Dobrota@com.utcluj.ro).

Kris Steenhaut is with the Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium (e-mail: ksteenha@info.vub.ac.be)

Ionut Trestian is with the Technical University of Cluj-Napoca, 400020 Constantin Daicoviciu 15, Cluj-Napoca, Romania (e-mail: trestianionut@yahoo.com).

Gabriel Lazar is with the Technical University of Cluj-Napoca, 400020 Constantin Daicoviciu 15, Cluj-Napoca, Romania (e-mail: Gabriel.Lazar@com.utcluj.ro).

getting connected to the distribution tree.

Multicast routing protocols minimize the paths from the receivers to the source, as opposed to minimizing the path from the source to the receiver.

II. PROTOCOL INDEPENDENT MULTICAST

PIM is a multicast routing protocol that is independent of the mechanisms provided by any unicast routing protocol. It requires some unicast routing protocols (such as RIP or OSPF) to determine the network topology and the topology changes. PIM is not a single multicast routing protocol, it has two different modes: PIM-DM (Dense Mode) and PIM-SM (Sparse Mode).

PIM-DM was designed for large multicast groups where most networks have a group member. It is quite similar to DVMRP because it builds source-based trees using flood-and-prune. However there are some important differences between these two protocols. The first is that PIM-DM involves an existing unicast routing to adapt to the topology changes, but at the same time is independent of the mechanisms of this unicast routing protocol.

The operation of PIM-DM is similar to DVMRP without the route exchange. To avoid duplicate multicast packets forwarding in multi-access networks, PIM-DM uses assert messages to determine a designated forwarder for the network.

Multicast forwarding is performed for the interfaces from the *oilist* (output interface list). The oilist is populated with those interfaces on which neighbors were discovered or on which multicast receivers have indicated their desire to receive traffic.

PIM-SM builds core-based trees as well as source-based trees with explicit joins. It was designed for environments where group members are distributed across many regions of the network. PIM-SM assumes that each receiver has to explicitly join a multicast tree if it wants to receive multicast packets. It creates a core-based tree with a shared root called RP (Rendez-vous Point). The RP is responsible for forwarding all packets destined for the multicast group. Each-group has a single RP at any given time. PIM-SM operation consists in three processes:

- Neighbor discovery, which uses router query messages
- RP registering, accomplished with register and register-stop messages
- RP joining/pruning, with join/prune messages

During neighbor discovery for a multi-access network a query message is sent to the all-routers multicast address, 224.0.0.2, which serves as the DR (Designated Router) election mechanism. When a source sends a multicast packet to a certain group, the DR of that source encapsulates the first message in a register message and sends it to the RP of that group as a unicast message. After receiving this message, the RP sends back a join message to the DR of the source. This way a distribution tree is created from the DR to the RP so the next multicast message of this source can be forwarded to the RP. Until the distribution tree is created, all multicast messages will be forwarded as encapsulated unicast messages. When the RP detects that multicast packets from the source are received as normal IP multicast packets, the RP sends a register-stop message to the DR. Upon reception of register-stop message the DR will stop encapsulating the multicast traffic from the source. Using the shared tree is not the best option in all cases. PIM-SM provides a method for using shortest-path trees for some or all of the receivers. When a threshold on a leaf router is exceeded, the router will switch from the shared tree through the RP to the source tree. In these situations, the leaf router sends a join message to the source node, thus creating a shortest-path tree.

PIM messages are encapsulated in IP packets with protocol number 103 and are sent to the multicast group 224.0.0.13, ALL-PIM-ROUTERS. The type field from the PIM packet header identifies the operation mode (dense/sparse) and the message type. The operation mode of PIM for IPv6 does not imply any major changes, except the type of addresses used within the header. The ALL-PIM-ROUTERS address ff02::d is the destination address for most messages.

III. TESTBED FOR NATIVE IPv6 MULTICAST

Initially we used the Nexthop's free software version of a routing tool for academic and research purposes called GateD Enterprise 2.0. Its operation and configuration is similar to Cisco routers but unfortunately it implements the IPv4 unicast/ multicast routing protocols only: RIPv1/v2, OSPF, BGP, DVMRP, PIM-DM, PIM-SM, PIM-SSM (Source Specific Multicast). Later the IPv6 solution was offered by the KAME Project. The code is merged in several BSD platforms (FreeBSD 4.0, OpenBSD 2.7, NetBSD 1.5, BSD/OS 4.2 and beyond). A separate KAME kit is also available and includes more experimental protocols but is not as stable as the included one.

Additionally, we used two pairs of programs (one for IPv4 and one for IPv6) for generating and receiving multicast traffic. The *m4sender/m6sender* programs transmit periodically (every second) UDP datagrams to a given multicast group. They do not join the multicast group. On the other hand the *m4receive/m6receive* programs join the multicast group and display the payload of received multicast messages to the standard output. In the experiments presented in this paragraph the sender is generating traffic for three different multicast groups:

```
#./m6send ff15::1 1111 "IPv6 multicast" 4
#./m6send ff15::2 2222 "IPv6 multicast" 4
#./m6send ff15::3 3333 "IPv6 multicast" 4
```

Note that the testing datagram's payload contains the text given by the user ("IPv6 multicast"), whilst the added parameter (in our case 4) represents the hop limit (within the header). The receiver is started with the command:

```
#./m6receive ff15::1 1111
#./m6receive ff15::2 2222
#./m6receive ff15::3 3333
```

This program sends Multicast Listener Report messages to notify the DR that the host wants to receive traffic for that multicast group. The topology presented in Fig. 1 is not very complex, but it is an excellent testbed for the performance evaluation of a native IPv6 multicast environment.

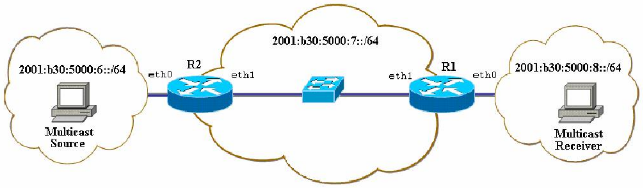

Fig. 1 Testbed for native IPv6 multicast

We involved two multicast routers, R1 and R2, operating under FreeBSD, and two machines acting as source and respectively as multicast receiver. Details regarding the configuration and the setup of our testbed were presented in [1] and are not covered by this paper. The IPv6 unicast routing daemon implementing RIPng protocol is called *route6d*. No configurations are needed for this daemon; the defaults assure the route exchange between R1 and R2. The configuration files for the two multicast routing daemons *pim6dd* or *pim6sd* are */etc/pim6dd.conf* and */etc/pim6sd.conf*. These files describe how the corresponding daemon treats each interface on the system. When using *pim6dd* the defaults (or no configuration file present) activate PIM-DM on all interfaces, thus enabling multicast routing. Operating *pim6sd* requires some configuration, given by the special operation of the protocol. The experiment was initially considered successful if the *m6receiver* program receives the packets from the source. However we captured the packets with ethereal 0.10.7 and implemented an XML parser to interpret the messages related to PIM sessions. The idea was to calculate the join latency as the receiving time of the first data packet since the Multicast Listener Report was sent to the closest router belonging to the distribution tree. Actually due to the robustness variable which is by default 2, there is more than one MLD message to tune for expecting packet loss on the link. However our software tool calculates the join latency based on the first Multicast Listener Report that was followed by multicast data traffic. The measurements were useful to calibrate the tool in the most favorable network condition and followed a scenario of about 10 minutes presented in Table I. A permanent ping6 from the receiving host (2001:b30:5000:8::3) to the source host (2001:b30:5000:6::6) was needed to keep track of the packets successfully received. Also the range of RTTs (Round Trip Times) was obtained for comparison. Two separate tests were carried out, one for PIM-DM (as in Fig.2) and one for PIM-SM (as in Fig.3). We tried to detect the influence of the number of multicast groups with/without members that joined (3 in our trial).

TABLE I
SCENARIO FOR NATIVE IPv6 MULTICAST TRIAL

Step No.	Members of receiver's group ff15::1	Members of receiver's group ff15::2	Members of receiver's group ff15::3	Join No.
1	Not joined	Not joined	Not joined	
2	*Newly Joined*	Not joined	Not joined	1
3	Joined	*Newly Joined*	Not joined	2
4	Joined	Joined	*Newly Joined*	3
5	Not joined	Not joined	Not joined	
6	*Newly Joined*	Not joined	Not joined	4
7	Joined	*Newly Joined*	Not joined	5
8	Joined	Joined	*Newly Joined*	6
9	Not joined	Not joined	Not joined	
10	*Newly Joined*	*Newly Joined*	*Newly Joined*	7,8,9
11	Not joined	Not joined	Not joined	

Fig. 2. Join Latency for PIM-DM: 586 packets transmitted, 586 received, 0% packet loss, time 584988ms, rtt min/avg/max/mdev = 0.315/0.371/0.796/0.040 ms, pipe2. The average join latency measured by our program was about 0.541 ms

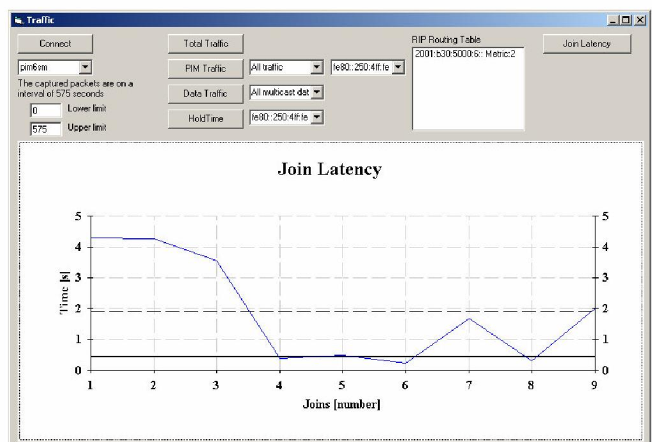

Fig. 3 Join Latency for PIM-SM: 574 packets transmitted, 574 received, 0% packet loss, time 573019ms, rtt min/avg/max/mdev = 0.325/0.437/1.097/0.154 ms, pipe 2. The average join latency measured by our program was about 1.91 ms

Although in [11] it was considered that the join latencies will not be an issue when choosing between dense and sparse mode, being not far from RTT to the closest router belonging to the distribution tree, the preliminary measurements revealed that in average is with 45% higher for PIM-DM and could be even more than four times higher in case of PIM-SM. However in case of packet losses on the distribution tree, the join latency is at least two times higher for both modes.

The importance of this parameter could be better understood in conjunction with the average bandwidth requested by the messages belonging to the routing protocol involved. Sometimes this is referred to as control overhead [2]. For the given scenario we obtained the average control traffic rate over total traffic rate as 19.01/341 Bps (PIM-DM) and 20.41/342 Bps (PIM-SM).

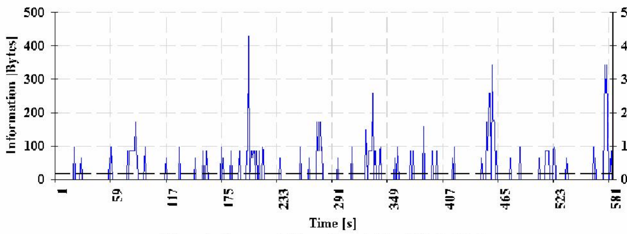

Fig. 4 Control Overhead for PIM-DM

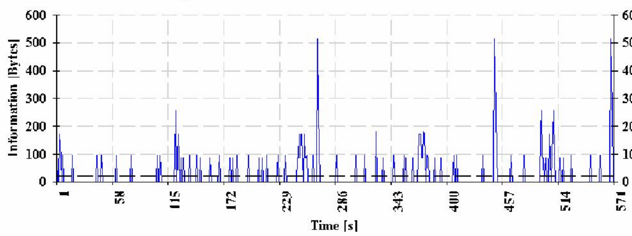

Fig. 5 Control Overhead for PIM-SM

Other parameters evaluated were the multicast forwarding delay (depending on RPF check, throughput and fanout) and its jitter. These tests were important for calibrating the software tool. The main idea is to find the proper configuration of the IPv6 multicast environment in order to have better performances in delivering the packets.

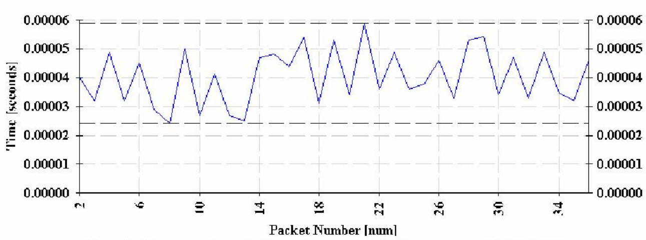

Fig. 6 Forwarding delay for router R2 running PIM-DM

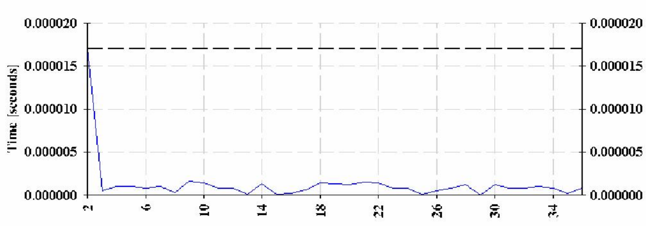

Fig. 7 Jitter of the forwarding delay for router R2 running PIM-DM

IV. EXISTING ALTERNATIVE TECHNOLOGIES TOWARDS MULTICAST ENVIRONMENT

Multicast access is not available to the regular Internet user. According to [7], the main reason is not the lack of multicast capable equipment; nowadays all commercially available routers have built-in support, but the so called "three-fold" deadlock. Three parties are involved in this situation, ISPs, content providers and customers. The first parties, ISPs are not eager to deploy it because the setup and management of a multicast architecture is more complex than for unicast. Also their clients (content providers and end users) have a low demand for it, so the deployment cost are not justified. The second party, the content providers, do not use multicast because end users do not have access to it. The last party, the customers, is only interested in the quality of the service. They are not aware of the benefits of multicast.

A number of proposals for alternative group communication services can be used to bypass the multicast routing deployment problems [2]. CastGate is one of them. It provides seamless access to multicast content through the use of auto-tunneling [7].

A. CastGate

The CastGate technology is the result of work by the Digital Telecommunications (TELE) research group of the ETRO department at the Vrije Universiteit Brussel. It is intended as a transition technology that will lead to an increase in the number of multicast users, thus ISPs will consider deploying native multicast.

Multicast is transmitted through a unicast tunnel, from a tunnel server which is located on the multicast part of the Internet and a tunnel client at the user side. A modified version of the UMPT (UDP Multicast Tunneling Protocol) is used. Two modes of tunneling, channel tunneling and raw tunneling are supported. In the first mode only datagrams destined for a specific multicast group and port number will be tunneled. Raw tunneling operation will tunnel all datagrams for a given multicast address. Due to some issues besides transport over UDP, HTTP tunneling transport is also included.

The basic CastGate architecture consists of three parts: TC (CastGate Tunnel Client), TS (CastGate Tunnel Server) and TDS (CastGate Tunnel Database Server). The database contains information about all the available TSs. Multiple TDSs form what is called a Hierarchical Tunnel Database.

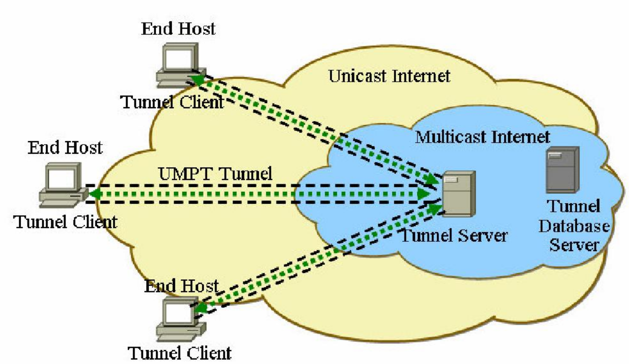

Fig. 8 The original CastGate solution [5]

The TS is to be found in the multicast part of the Internet, where it terminates one end of the tunnel. The TC is located at the client side, where it terminates the other end of the tunnel. It will ask the Hierarchical Tunnel Database for a list of Tunnel Servers. The TC signals the chosen TS the multicast group it wants to receive traffic for, and the TS will tunnel it to the client side. The TC can be integrated in a multicast application or it can be a Java applet which runs in a web browser. In either situation, the operation is transparent to the end user. From its point of view it is as good as native multicast.

B. *CastGate Router*

CastGate Router is a result of the further development of CastGate technology. It integrates the functionality of an IGMP querier with the Tunnel Client. Thus it provides multicast access to all the hosts joined to the same LAN segment.

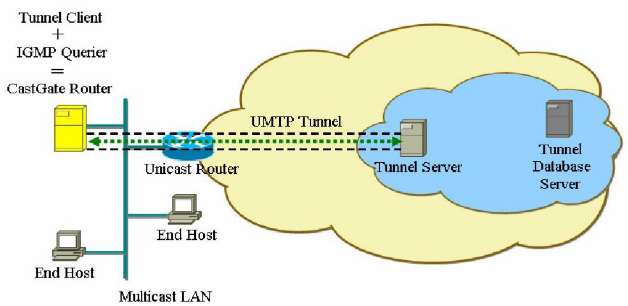

Fig.9 CastGate Router [6]

The IGMP querier from the CastGate Router keeps track of the group membership for that LAN segment. Based on this information the Tunnel Client will join or leave the multicast group through the tunnel. The advantage of using a CastGate Router is that multicast traffic is tunneled only once for all the receivers on that LAN [6]. The use of the initial technology requires each end host to run a Tunnel Client, thus several unicast packets with identical multicast data are transmitted on the same link.

V. PROPOSED IMPROVEMENTS: CASTGATE ROUTER WITH PIM-SM SUPPORT

A. *CastGate Router with PIM-SM Support*

The functionality of the CastGate Router was extended so that multicast access is provided to an entire local domain. By domain we understand a group of networks under local administration, where any multicast protocol can be used, but without global multicast access.

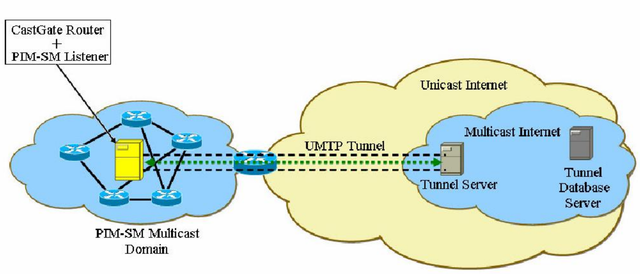

Fig. 10 CastGate Router with PIM-SM support

The PIM-SM routing protocol is best fitted for the job because it creates multicast delivery trees with a single common root RP. The information about multicast activity in the domain is gathered by the rendez-vous point. Placing a modified CastGate Router on the same links as the RP would give us access to information regarding multicast receivers and sources in the domain. The multicast traffic tunneled (by the CastGate Router) to these links will be delivered to all the receivers by the PIM-SM routers. These functions are performed by a new module in the CastGate Router, called PIM-SM Listener. Fig.10 presents the modified CastGate Router, the work being realized by T. Blaga during his work at Vrije Universiteit Brussel.

The module captures PIM-SM messages from which it extracts information about multicast groups that have members in the domain. Only two types of messages are of interest, Hello and Join/Prune messages. The first type of messages contains information about the PIM-SM routers on the link. Information about group membership across the domain is contained in the Join/Prune messages (actually (*, G) Join/Prune). For each group, on interface basis a state machine keeps track of state information and of corresponding timers. Based on the information from the state machine a group will be joined or leaved through the tunnel and multicast traffic from the tunnel is forwarded.

In order to reduce the resources utilized in the domain, only one RP can be used for all the multicast groups. Thus only one CastGate Router with PIM-SM listener module will serve the entire domain.

One feature of PIM-SM, SPT (Shortest Path Tree) switchover must be disabled. SPT switchover permits the leaf router to switch to the source-based tree after receiving multicast traffic from that source on the core-based tree. The local domain does not have global multicast access, so receiving multicast along the source-based tree is impossible.

B. *IPv6 CastGate*

UMPT tunnels multicast data over unicast UDP datagrams between a pair of nodes, which form the endpoints of the tunnel. One node will act as a master (CastGate Tunnel Client) while the other node will be the slave (CastGate Tunnel Server). Each UMPT datagram contains a 12/16-octet trailer, which is a command. CastGate makes use of an enhanced version of UMPT. One issue of UMPT is the use of large trailers, 12/16 octets, that can lead to fragmentation [12]. Another issue of the IP Multicast Service model, the lack of support for Authentication, Authorization and Accounting (AAA) was addressed by extending the UMTP protocol. The enhancement consists of option and parameter negotiation.

The next step towards native IPv6 multicast environment is to have an IPv6-enabled CastGate architecture. This requires the addition of new functionality to it. The communication between the TC, TS and TDS can be performed using IPv4. Also IPv6 multicast traffic can be tunneled over IPv4. Thus IPv6 unicast connectivity is not required between them. The TDS does not need any modification, but the TC and the TS must be IPv6 capable.

CastGate Router would support IPv6 end hosts only if a MLD querier is installed. Also the Tunnel Client must implement an IPv6 Enhanced UMTP. This means the 12-octet trailer should be replaced by a 24-octet trailer and the 16-octet one should have 40 bytes, as in Fig.11. Also the TTL field is changed to Hop Limit which performs similar functions in IPv6 as TTL in IPv4.

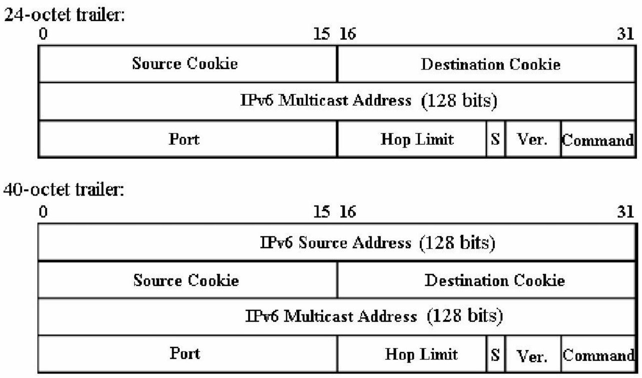

Fig. 11 IPv6 Enhanced UMPT Trailer

We propose a new type of command IPv6_DATA with code 13, that will indicate the tunneling of IPv6 multicast data. Using the value of the S bit for this new type of command we can differentiate between the 24 and 40 octet trailers. The negotiation of IPv6 support between the client and the server will be performed through the use of a new Enhanced UMPT option, called OPT_IPV6 with code 20 and length 0.

IPv6 CastGate Router can listen to MLD messages on the link, and get information about IPv6 group membership on that LAN. From this point on, the operation of the router is similar to IPv4; the group will be joined through the tunnel. Regarding the PIM-SM CastGate Router, as in Fig.12, the changes imply that the PIM-SM listener module must also capture IPv6 routing messages destined to the RP and extract the needed information similar to the IPv4 version.

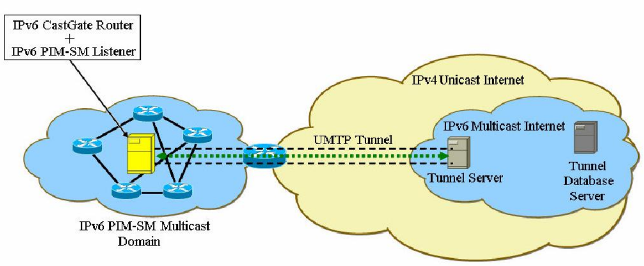

Fig. 12 IPv6 CastGate Router with PIM-SM support

VI. Conclusion

This paper determined the join latency in a native IPv6 multicast testbed, demonstrating that its average value could be higher than the round-trip time between a multicast receiver and the closest router belonging to the distribution tree of that group. The preliminary values for PIM-DM were with 45% higher or even at least two times higher in case of packet losses. Join latency was more than four times higher than RTT for PIM-SM. Other parameters measured were the control overhead, forwarding delay and its jitter. Due to so-called "three-fold" deadlock, the multicast access is not available to the regular Internet user. We proposed an enhancement to the existing alternative solutions (CastGate and CastGate router), i.e. PIM-SM support. This idea could be generalized to other multicast tunneling solutions. Furthermore, we investigated the possibility of using CastGate architectures in IPv6, with a remaining IPv4 tunnel. Obviously once the native IPv6 multicast in fully available, any CastGate-based solutions, no matter its version, will be replaced. However our approach to determine the network parameters could lead to proper selection of PIM-SM/DM configuration and better performances. This involves a co-operation with the telecom operators for deployment of IPv4/IPv6 multicast services.

Acknowledgment

The authors thank for the support provided by the staff from TELE Research Group of Vrije Universiteit Brussel, mainly to Marnix Goossens and Arnout Swinnen. Special thanks go to Pieter Liefooghe, the author of CastGate, for valuable discussions and technical support.

References

[1] T. Blaga, V. Dobrota, D.Zinca, G. Lazar, "Multicast Routing in IPv4/IPv6 Using Linux and FreeBSD", in *Scientific Bulletin of the "Politehnica" University of Timisoara, Transaction son Electronics and Communications, Tomul 49(63), Fascicola 1, 2004*, ISSN 1583-3380, pp. 383-387.

[2] Ayman El-Sayed, V. Roca, L. Mathy, "A Survey of Proposals for an Alternative Group Communication Service", *IEEE Network*, January-February 2003, pp 46-51

[3] Bill Fenner, M Handley, H. Holbrook, I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", Internet Engineering Task Force PIM WG INTERNET-DRAFT, *draft-ietf-pim-sm-v2-new-11.txt*, 25 October 2004

[4] Ian Brown, K. Dooley, *Cisco Cookbook*, O'Reilly Publishing, July 2003

[5] Pieter Liefooghe, "An Architecture for Seamless Access to Multicast Content", *PhD Thesis*, Vrije Universiteit Brussel, 2002

[6] Pieter Liefooghe, M. Goossens, A. Swinnen, B. Haagdorens, "The VUB Internet Multicast "CastGate" Project", *Technical Report 10/2004 v1.8*

[7] Pieter Liefooghe, "CastGate: An Auto-Tunneling Architecture for IP Multicast, *<draft-liefooghe-castgate-02.txt>*, October 2004

[8] William Parkhurst, *Cisco Multicast Routing & Switching,* McGraw-Hill, 1999

[9] http://www.xorp.net

[10] Beau Williamson, *Developing IP Multicast Networks*, Volume I, Cisco Press, July 2001

[11] Liming Wei, Deborah Estrin, "Multicast Routing in Dense and Sparse Mode: Simulation Study of Tradeoffs and Dynamics", University of Southern California, 1995

[12] Ross Finlayson, "The UDP Multcast Tunneling Protocol", <draft-finlayson-umtp-09.txt>, November 2003