TD(06)010

5th COST 290 Management Committee and Technical Meeting

Delft, 9-10 February, 2006

Evaluating and Improving Alternative Multicast Solutions: CastGate and CastGate with PIM-SM

Tudor Mihai Blaga, Virgil Dobrota
Technical University of Cluj-Napoca, Baritiu 26-28, Cluj-Napoca,
{tudor.blaga, virgil.dobrota}@com.utcluj.ro

Abstract

One solution to the lack of native multicast is CastGate. It makes use of tunneling to transmit data to end hosts. We describe a solution to improve this technology with the aid of PIM-SM. Further we evaluate the different CastGate based solutions, and compare them to native multicast. Some of the metrics (stress, resource usage, stretch) can be evaluated whilst other metrics (control overhead, join latency) can be determined only through measurement in a real testbed.

Keywords

Multicast, AGCS

Working Group 3: “Network architecture and planning”

Evaluating and Improving Alternative Multicast Solutions: CastGate and CastGate with PIM-SM

Tudor Mihai Blaga[1], Virgil Dobrota[1]

Abstract

One solution to the lack of native multicast is CastGate. It makes use of tunneling to transmit data to end hosts. We describe a solution to improve this technology with the aid of PIM-SM. Further we evaluate the different CastGate based solutions, and compare them to native multicast. Some of the metrics (stress, resource usage, stretch) can be evaluated whilst other metrics (control overhead, join latency) can be determined only through measurement in a real testbed.

1. Introduction

Nowadays numerous applications make intensive use of network resources. The distribution of multimedia content over the Internet would be better served by multicast. Traditionally streaming media content is offered using unicast. In this case, the bandwidth requirements increase linearly with the number of receivers. Also the load on the servers increases. The use of multicast provides a better solution. The media content needs to be sent only once, so the bandwidth is independent of the number of users.

Native multicast makes use of specialized routing protocols to create distribution trees for the data from the source to the receivers. 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 (PIM Dense Mode) and PIM-SM (PIM Sparse Mode). PIM-DM builds source-based trees using flood-and-prune, and is intended for large multicast groups where most networks have a group member. PIM-SM builds core-based trees as well as source-based trees with explicit joins, and it is intended for environments where group members are distributed across many regions of the network.

The issues regarding multicast deployment or rather the lack of multicast deployment are discussed in [1, 2] and [3]. Basically there are technical reasons and marketing reasons. The complexity of the protocols involved, the limitations and lack of customer demand have led to the development of several proposals for alternative group communication services (AGCS). These can be used to bypass the multicast communication problems [1]. Some make use of tunneling, overlay multicasting or group specific routing services.

[1] Technical University of Cluj-Napoca, Communications Department, 26-28 Baritiu Street, 400027 Cluj-Napoca, Romania,
Tel/Fax: +40-264-597083, E-mail: {tudor.blaga, virgil.dobrota}@com.utcluj.ro

The remainder of this article is organized as follows. We describe the CastGate technology, we evaluate its performances using metrics defined for AGSC and compare it to native multicast and we present the details regarding the implementation of CastGate with PIM-SM.

2. 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 provides seamless access to multicast content through the use of auto-tunneling [4]. 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) called Enhanced UMTP is used [5]. 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 (Figure 1) 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.

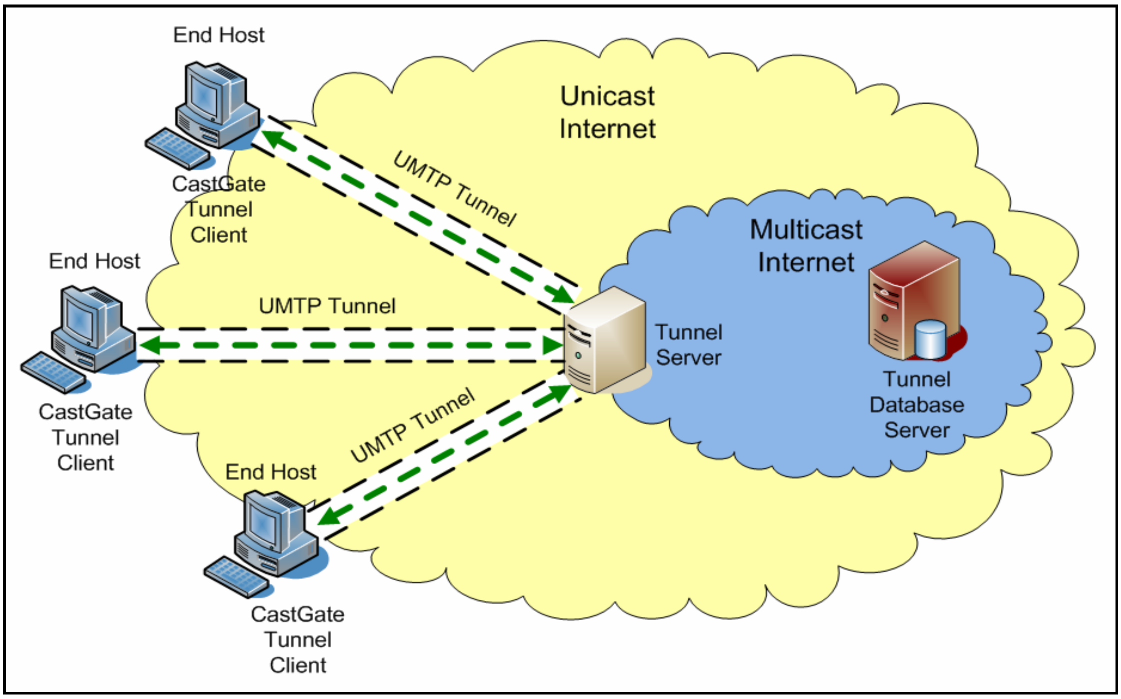

Figure 1. CastGate Client

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.

CastGate Router is a result of the further development of CastGate technology [6]. 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. 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.

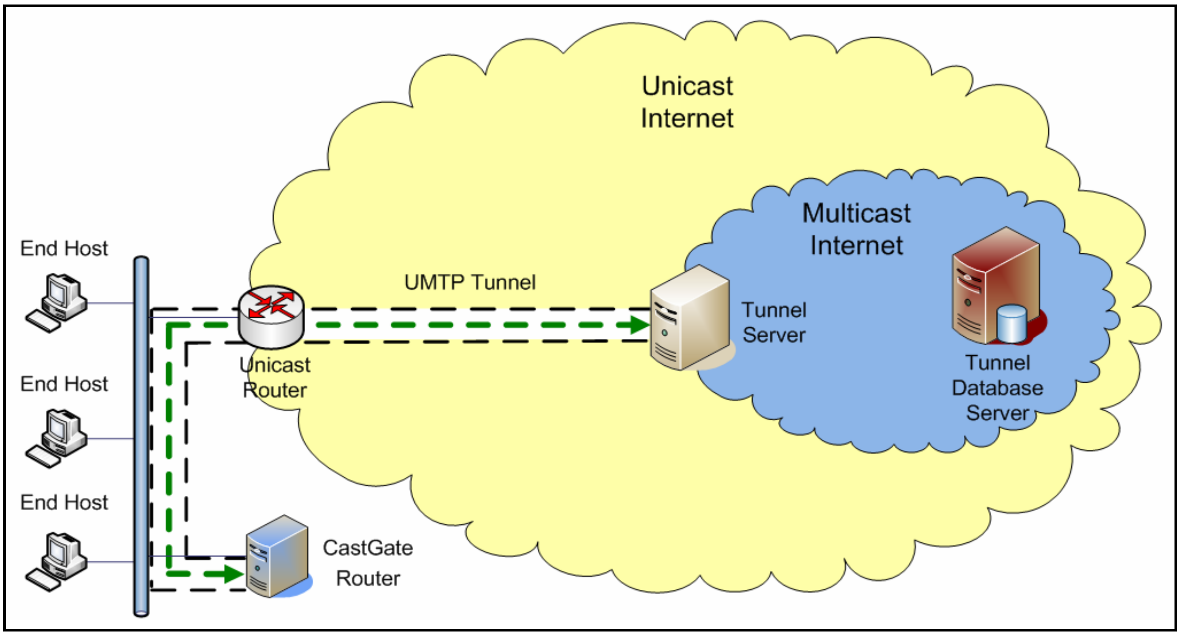

Figure 2. CastGate Router

CastGate allows to address some issues which are not solved by the current IP multicast service. One of them is that native multicast lacks AAA (Authentication Authority Accounting). By adding support for some of the AAA features to Enhanced UMTP, CastGate provides a temporary solution. The Tunnel Server can be combined with RADIUS (Remote Authentication Dial-In User Service) server to enable the authentication of the user, to allow for selective authorization of sessions and to provide accounting of the use of the multicast service .

To the CastGate project belong CastGuide and CastContent [6]. CastGuide is a session directory tool that allows you to obtain a list of available and upcoming IP Multicast content. It is the equivalent of a TV-guide, but then for Multicast Content. CastContent deals with tools for the content provider, to address certain issues about access control and accounting. Two versions are being developed: CastLive for "live" distribution of content and CastCOD for Content-On-Demand distribution using multicast.

3. Improving CastGate

Our idea is to extend, and thus improve, the functionality of the CastGate Router, so that it can provide multicast access 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. Tunneling traffic to the local domain with the use of an extended CastGate Router and then distributing that traffic through native multicast (Figure 3.) would prove a great benefit.

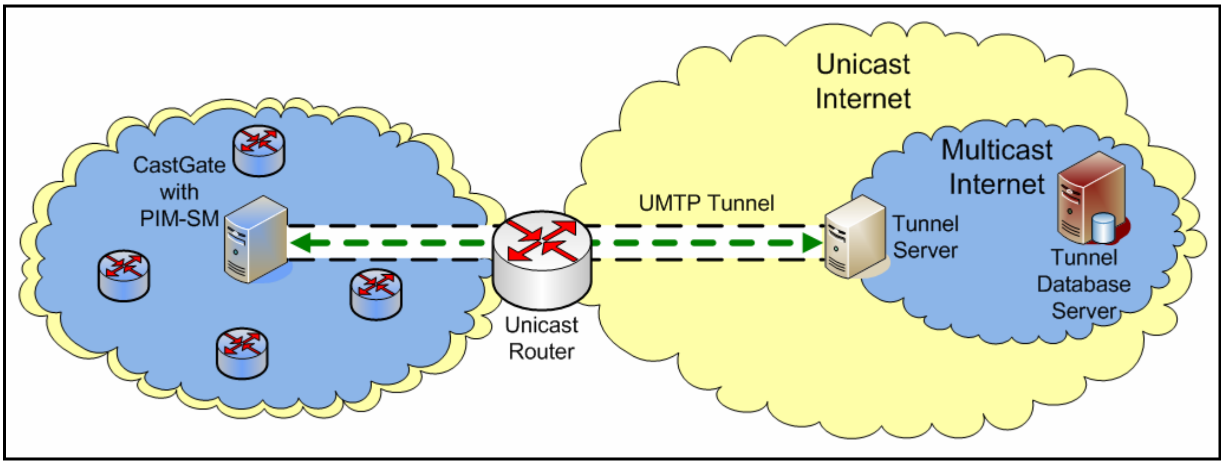

Figure 3. PIM-SM CastGate Router

One of the issues is to the choice of multicast routing protocol that would operate in the local domain. PIM-SM [7] routing protocol is best fitted for the job because it creates multicast delivery trees with a single common root. This shared root is called Rendezvous Point (RP). Information about multicast activity in the domain is gathered by the RP. Placing a modified CastGate Router on the same link 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 this link will be delivered to all the receivers by the PIM-SM routers without need for further intervention. Also multicast traffic from a source located anywhere in the local domain will reach the RP. Due to implementation complexity it was decided not to embed a PIM-SM router with the CastGate Router, but rather to extract the minimum functionality from the PIM-SM standard [7].

The scenario used is the RP-on-a-stick scenario [8]. This happens when the incoming interface of an (S, G) entry at the RP is also the only outgoing interface on the shared tree for group G. It is important to understand that multicast traffic is never forwarded on the same interface it was received on.

3.1. *Receiving multicast from the Internet*

The PIM-SM module has to listen to all the messages destined for the RP and it must decide whether to join or leave a group through the tunnel. The module captures PIM-SM messages from which it extracts information about multicast groups that have members in the domain. From the different PIM-SM message types, only two are of interest, Hello and Join/Prune messages. The first type of messages contains information about the neighboring PIM-SM routers on the link, information that will be recorded in a neighbor list. Information about group membership across the domain is contained in the Join/Prune messages (actually (*, G) Join/Prune). For each group, a state machine keeps track of state information and of corresponding timers. Based on the information from the state machine the group will be joined through the tunnel and in this situation multicast traffic from the tunnel is forwarded to the domain.

3.1.1. Neighbor list

Neighbors will not accept Join/Prune messages from a router unless they have first heard a Hello message from that router. The information from these messages is kept in a list of neighbors on a per interface basis. This list contains the following data: IP address, Holdtime, GenID, LAN Prune Delay (Propagation_Delay (I) and Override_Interval (I)).

Holdtime is the amount of time a receiver must keep the neighbor reachable, in seconds. The default value is 210 seconds. If the Holdtime is set to `0xffff', the receiver of this message never times out the neighbor. Hello messages with a Holdtime value set to `0' are also sent by a router on an interface about to go down or changing IP address. These are effectively goodbye messages and the receiving routers should immediately time out the neighbor information for the sender.

The GenID option contains a randomly generated 32-bit value that is regenerated each time PIM forwarding is started or restarted on the interface, including when the router itself restarts. When a Hello message with a new GenID is received from a neighbor, any old Hello information about that neighbor should be discarded and superseded by the information from the new Hello message.The values of Propagation_Delay (I) and Override_Interval (I) from the LAN Prune Delay are used only if there are more than two neighbors in the list and all the neighbors in the list advertise it. The default values are: 0.5 secs and 2.5 secs [7].

3.1.2. State machine

The PIM-SM module uses a modified version of the downstream per-interface (*, G) state machine from the protocol specifications. Modifications were necessary because the PIM-SM extension module does not implement the entire functionality of PIM-SM. For example we do not send a (*, G) PruneEcho message when a transition occurs from Prune-Pending to NoInfo state.

Join/Prune messages carry information about the active groups in the domain. The module will listen only to (*, G) Join/Prunes which are used to create core-based trees. These messages specify that group G must be joined or pruned from any source (*).

The listener module should check whether the Upstream Neighbor Address and the Joined/Pruned Source Address in the incoming (*, G) Join/Prune message matches the address of the RP (RP address should be configured on the listener module for security reasons).

Figure 4 presents the state machine for PIM-SM and the PIM-SM module. The differences are marked using blue. Notice the send Prune-Echo (*, G) is crossed out, because it is not used. The transition to Join state determines the creation of the tunnel.

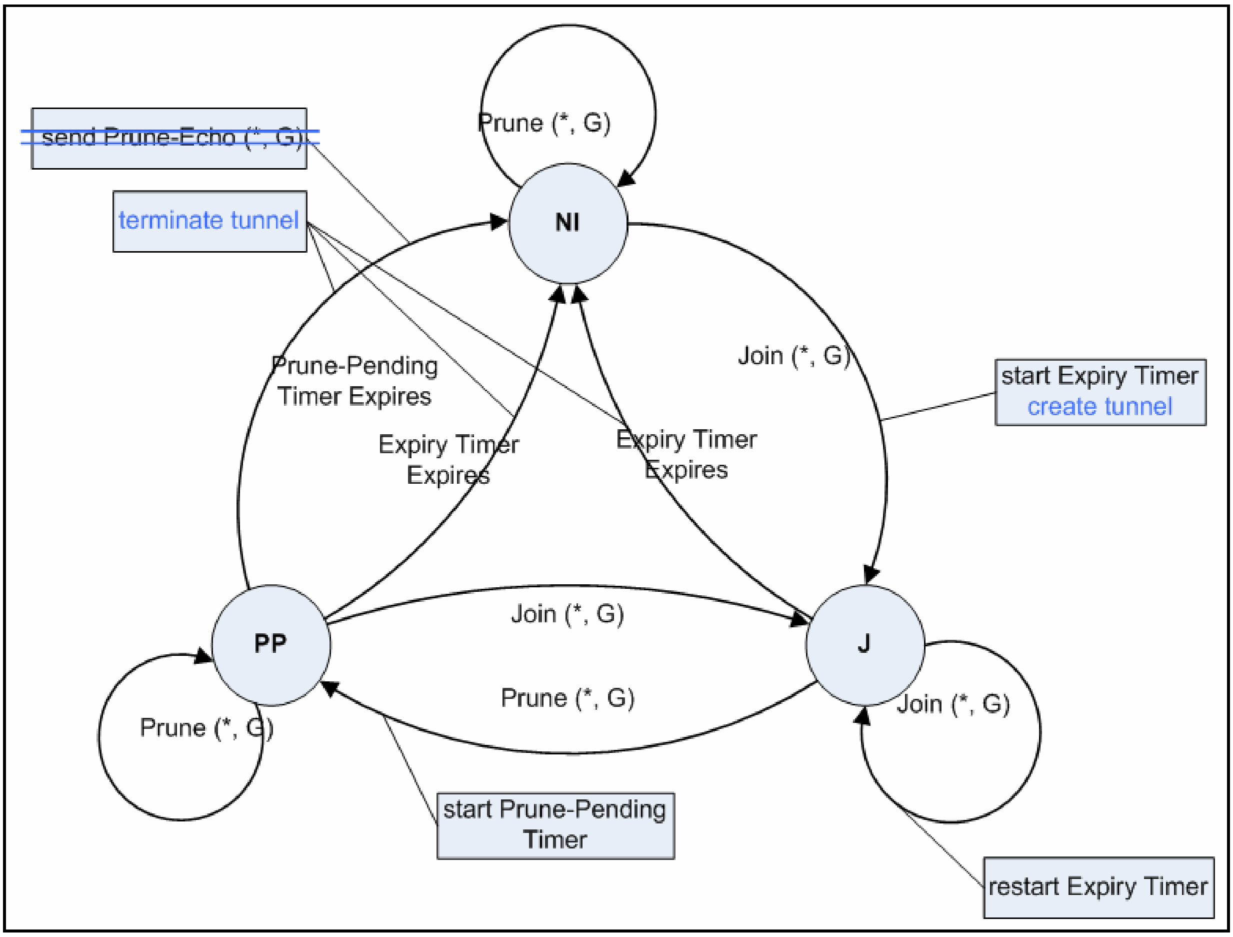

Figure 4. State machine for PIM-SM module and PIM-SM

The state machine has three states:

- NoInfo (NI) - the interface has no (*, G) Join state and no timers running.

- Join (J) - the interface has (*, G) Join state. The module will forward packets destined for G from this interface, packets received from the tunnel.
- Prune-Pending (PP) - the module has received a (*, G) Prune on this interface from a downstream neighbor and is waiting to see whether the prune will be overridden by another downstream router. For forwarding purposes, the PP state functions exactly like the Join state.

The state machine uses two timers:

- Expiry Timer (ET) - causes the interface state to revert to NoInfo state on expiry. ET is restarted when a (*, G) Join is received.
- Prune-Pending Timer (PPT) - is set when a (*, G) Prune is received. Expiry of the PPT causes the interface state to revert to NoInfo state for this group.

The transition events "Receive (*, G) Join" and "Receive (*, G) Prune" imply receiving a Join/Prune targeted to this router's primary IP address (Upstream Neighbor Address) on the receiving interface.

1. Transitions from NoInfo state:
   - Receive (*, G) Join on interface - transition to Join state, group G is joined through the tunnel and ET is started, ET = Holdtime.

2. Transitions from Join state:
   - Receive (*, G) Join on interface - ET is restarted and set to ET = max (current value, Holdtime from received message).
   - Receive (*, G) Prune on interface - transition to Prune-Pending state and PPT is stared, PPT = J/P_Override_Interval (I) if the module has more than two neighbors on that interface, otherwise it is set to zero causing it to expire immediately.
   - ET expires - transition to NoInfo state and tunnel is closed.

3. Transitions from Prune-Pending state:
   - Receive (*, G) Join on interface - transition to Join state. PPT is canceled and ET is started, ET = max (current value, Holdtime from received message).
   - ET expires on interface - transition to NoInfo state and tunnel is terminated. PPT expires on interface - transition to NoInfo state and tunnel is terminated

3.2. *Sending multicast from the local multicast domain*

In order to be able to send multicast traffic from the sources within the local domain, first we must make sure that it will reach the RP. The RP is notified of existing multicast sources through the Source Registration process. When a multicast source begins to transmit, the DR (directly connected to the source) receives the multicast packets sent by the source and encapsulates each one in PIM Register messages. These messages are unicast to the RP, which de-encapsulates them. The RP will forward the multicast packet down the shared tree and will join the SPT for source S so that it can receive (S, G) traffic natively. If there is no active shared tree for the group, the RP discards the multicast packets and does not send a Join toward the source. The RP unicasts PIM Register-Stop messages to the DR, to instruct it to stop sending PIM Register messages, if it receives the multicast traffic via the (S, G) SPT or if it has no need for the traffic because there is no active shared tree for the group.

The PIM-SM module must intervene during the Source Registration process. It must intercept the PIM Register messages, extract information about the group and de-encapsulate the multicast data which will be tunneled to the Tunnel Server. Join (*, G) messages must be sent to make sure that the RP will join the SPT for the source. Also, in order for the Join message to be accepted by the RP we must send Hello messages [7].

3.2.1. Sending Hello messages

Hello messages are sent periodically. The value for the default Hello Period is 30 seconds. We must take into consideration that these messages are used for the DR election on that link. If the DR_Priority Option is used, the router with the highest value will be the DR. If this option is not present, then the values of the IP address is used to compute the DR [7]. In this case the machine with the highest value is elected. Because our module does not implement the full functionality of a PIM-SM router we must make sure that it will not be elected DR on the link. This can be accomplished by the use of DR_Priority Option with the value set to zero in the Hello messages sent by the module.

3.2.2. Analyzing Register messages

The captured PIM Register messages must be analyzed. First the Null-Register bit will be checked. If this bit is set to 1, then the message will be discarded because it contains in the multicast data packet portion a dummy header [7]. If the value is 0, then this Register message contains a real multicast data packet. This packet will be sent over the tunnel, and also the information regarding the existence of a source for multicast group G will be extracted.

3.2.3. Sending Join (*, G) messages

Once the presence of a source for group G is detected, we must “convince” the RP to join the SPT for source S. This can be accomplished by sending a Join (*, G) message. This message must be sent periodically every 60 seconds. According to standard specification if a PIM router sees a Join (*, G) message on the interface it must suppress its own Join (*, G). Also if it sees a Prune (*, G) it must override it by sending a Join (*, G) almost immediately.

A keepalive timer on a group basis will be used to decide when to stop sending Join messages. This represents the period after the last data packet for group G was received during which we keep on sending Join messages, and has a value of 210 seconds.

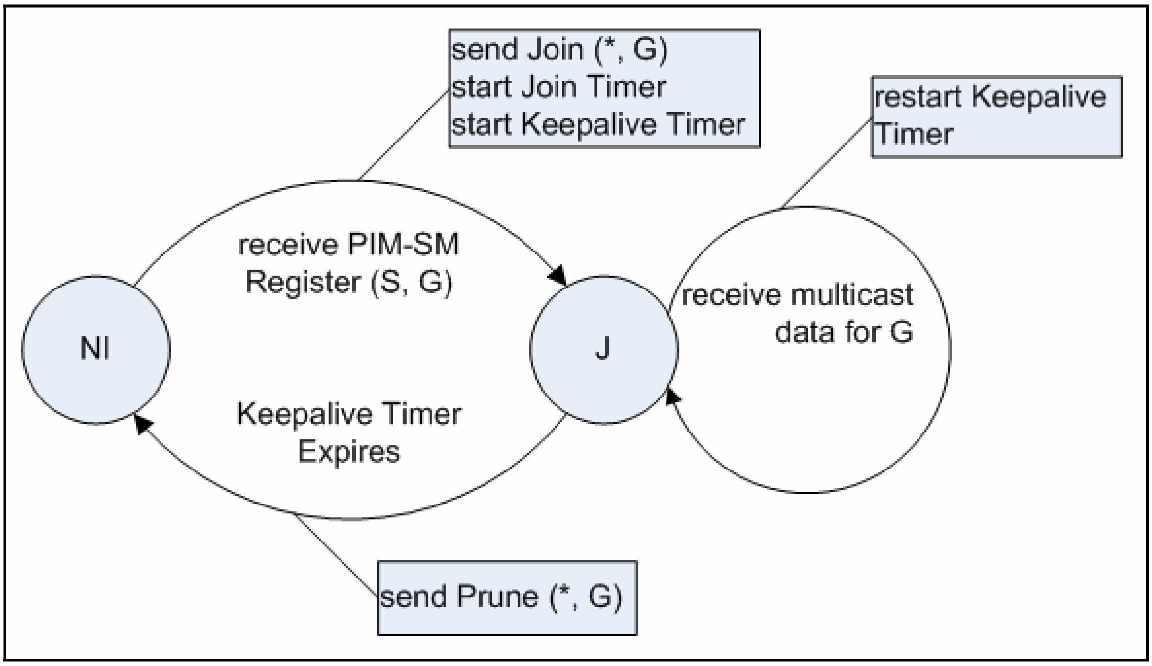

Figure 5. PIM-SM send multicast state machine

The state machine has two states:

- NoInfo (NI) – there is no (*, G) Join state, no timers are running;
- Join (J) – the module is in (*, G) Join state. Multicast is forwarded from the domain through the tunnel.

The state machine uses two timer:

- Join Timer – period for sending (*, G) Join messages with default value 60 seconds.
- Keepalive Timer – is set when a PIM Register message is received. Expiry causes the module to send (*, G) Prune and to revert to NoInfo state. Default value is 210 seconds.

We have two transition events:

1. Transition from NoInfo state:
    - Receive PIM Register (S, G) destined for the RP – transition to Join state and both timer are set.
2. Transition from Join state:
    - Keepalive timer expires – transition to NoInfo state.

4. Evaluating CastGate

Several performance metrics have been defined in [1] and [3] to characterize AGCS performance and impact on the network. The goal of our evaluation is to establish the downside of using the CastGate technology compared to native multicast. Some of these metrics can be determined using a simulated network architecture and some can only be determined in an operational network.

The metrics considered are:

- *Stress*: the number of identical copies of a packet carried by a physical link as the *stress of a physical link*. For example, if on a link the packet arrives tunneled from the source and then it is distributed through multicast, the stress on that link has a value of 2. In general, we would like to keep the stress on all links as low as possible.
- *Resource usage*: is defined as $\sum_{i=1}^{L} d_i * s_i$, where L is the number of links active in data transmission, d_i is the delay of link i, and s_i is the stress of link i. The resource usage is a metric of the network resources consumed in the process of data delivery to all receivers. There is the assumption that links with higher delay tend to be associated with higher cost.
- *Stretch*: also called *Relative Delay Penalty* represents the ratio of the delay between the source and the receivers along the AGCS route to the delay of the unicast path.
- *Control overhead*: quantifies the cost of maintaining the AGCS topology, in terms of control information exchanged (number of messages and bandwidth).
- *Join latency*: also known as *Time to First Packet*, defines the time required for a newly joined member to start receiving the data flow.

In a previous paper [9] we determined through experiments the values for join latency and control overhead for IPv6 PIM-DM and PIM-SM. In the case of native multicast join latency refers to the time passed from the first Multicast Listener Report sent by the end host to the first multicast packet received by the host. Control overhead is obtained by measuring the bandwidth occupied by the PIM messages, that are used to create and maintain the multicast distribution trees.

Using the network topology from figure 6 we evaluate the following metrics: stress, resource usage and stretch. Our analysis focuses on what happens in the local domain, so the results presented refer only to these links. In order to determine join latency and control overhead measurements will be carried out in a operational network.

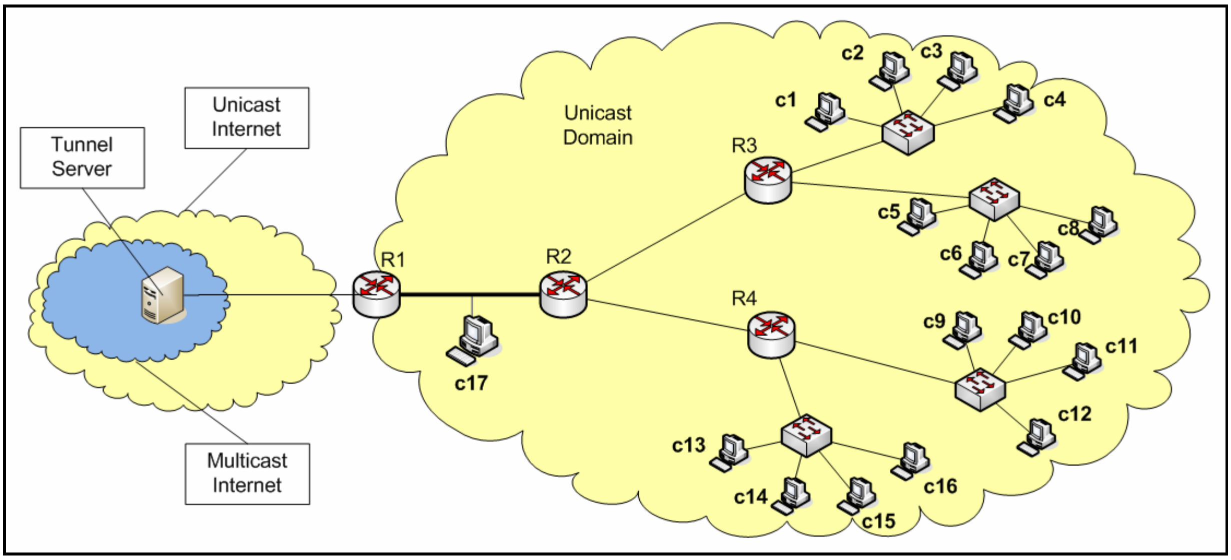

Figure 6. Network architecture

The test network has a total of 7 links and only one connection to the Internet through router R1. Inside the domain we have three routers R2, R3 and R4. Routers R3 and R4 each have two LAN segments with multicast receivers connected. The number of end hosts is 17, from c1 to c17. We consider each LAN segment to be only one link because it only one broadcast domain. We assume that the delay on all the links is equal and has a relative value of 1.

Four scenarios were analyzed:

- CastGate Client
- CastGate Router
- CastGate with PIM-SM
- Native multicast

The first scenario taken into consideration (figure 7) is the use of the CastGate Client. Each host in the domain runs the Java applet to receive multicast content. They are all receiving the same data. The stress values for each link can be observed in the figure. This scenario is similar to unicast from the determined metrics point of view, because the CastGate technology uses tunneling over unicast. A separate analysis for unicast was not carried out.

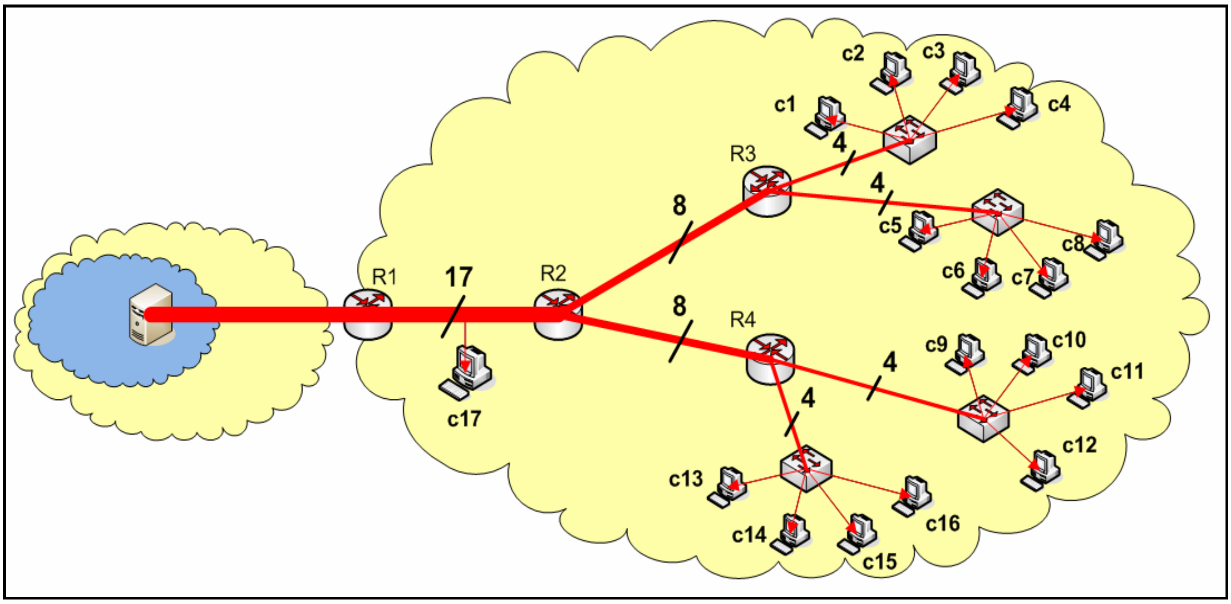

Figure 7. CastGate Client scenario

In the second case (figure 8) one of the hosts in each LAN acts as a CastGate Router. We have four multicast enabled LAN segments, where the host use IGMP (Internet Group Management Protocol). The distribution of data through native multicast is represented with a different color that the tunneled data. Multicast traffic must be tunneled to the local domain only 5 times, for the following hosts c1, c5, c9, c16 and c17.

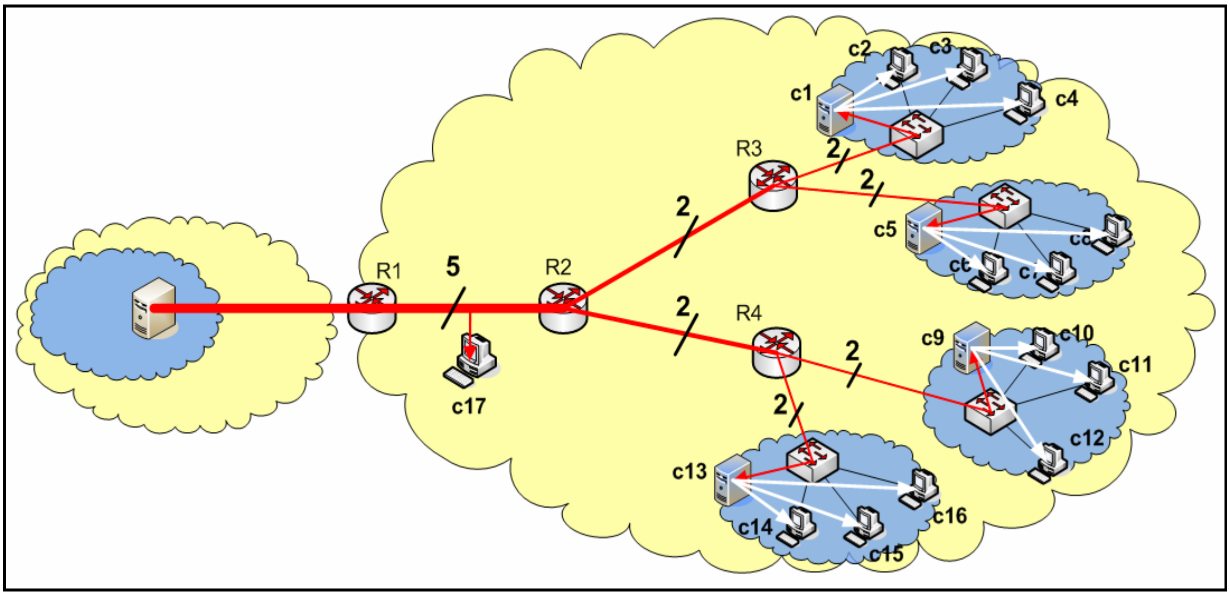

Figure 8. CastGate Router scenario

The use of CastGate with PIM-SM is considered in the third scenario (figure 9). The host c17 that is placed on the link between R1 and R2, acts as a CastGate with PIM-SM device thus providing multicast access for the entire domain. Its placement on the R1-R2 link offers the best performance.

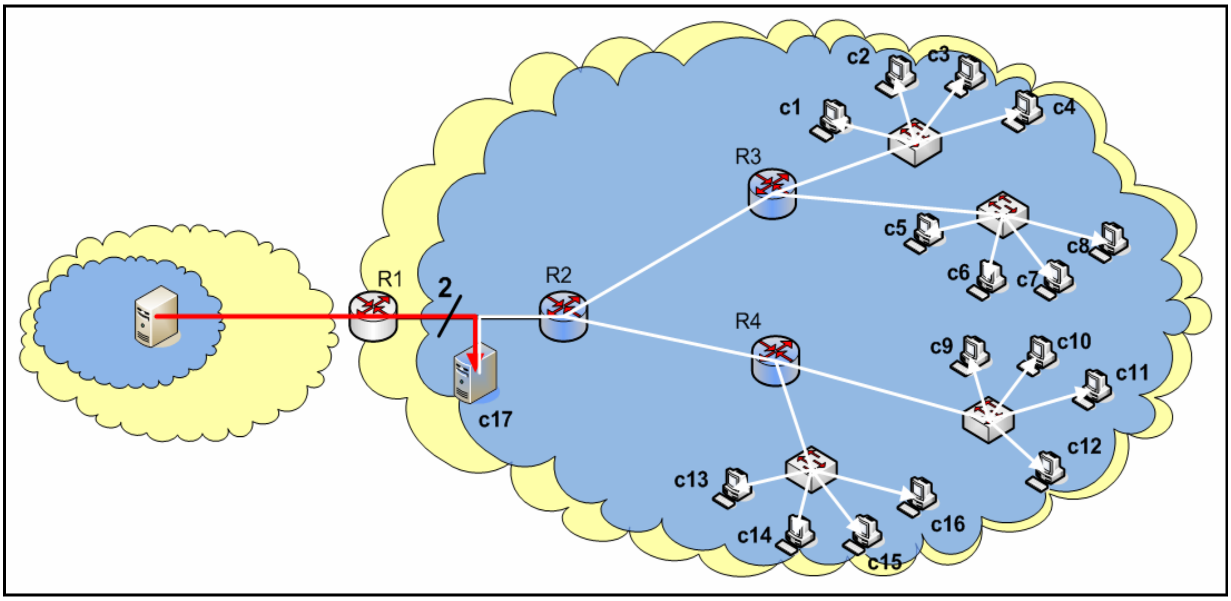

Figure 9. CastGate with PIM-SM scenario

The fourth scenario assumes the use of multicast in the entire Internet (figure 10). There is no further need to tunnel the multicast traffic to the local domain. The stress has the value 1 on all the links, as a result of native multicast routing.

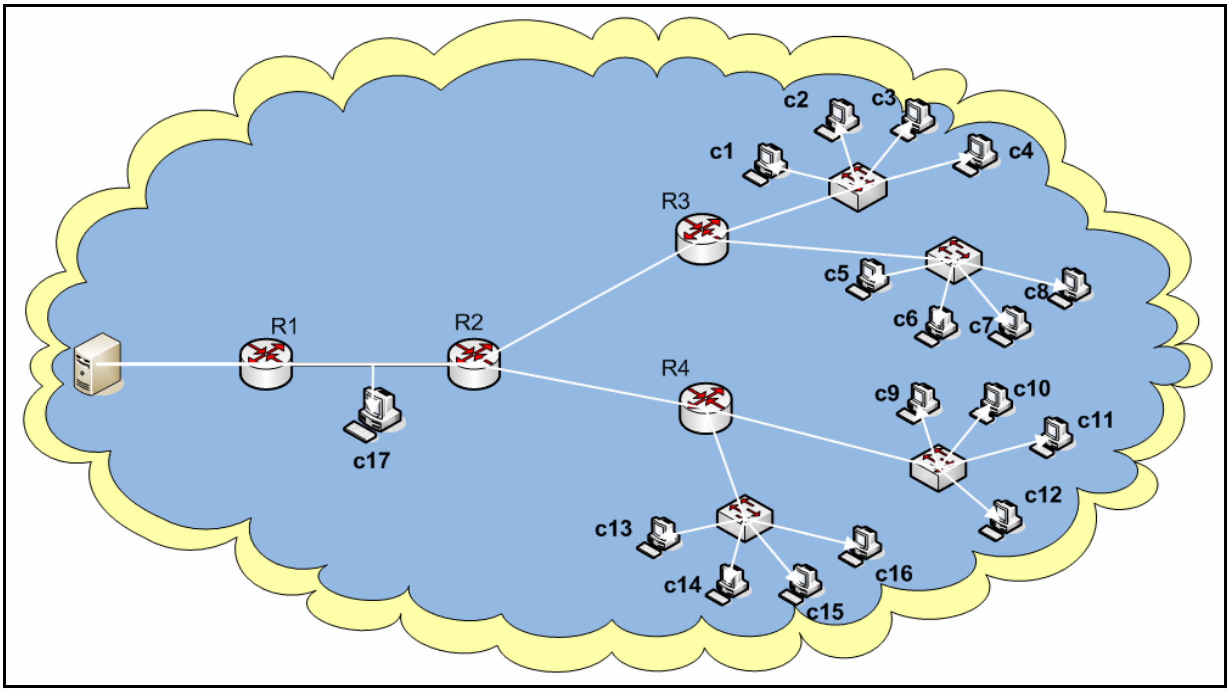

Figure 10. Native multicast scenario

In our scenarios the resource usage $R = \sum_{i=1}^{7} d_i * s_i$, because we have a number of links L = 7. The delay has a relative value of 1, so d_i = 1 for i = 1...7. This gives the following formula for resource usage in our scenarios $R = \sum_{i=1}^{7} s_i$.

Table 1 presents the determined metrics for each scenario. The results for the stress metric are presented in the first column on a per link basis. We observe that the access link (R1-R2) has the highest values (s1) in all of the CastGate based scenarios. This is due to the fact that all traffic for the local domain has to cross this link. The use of CastGate with PIM-SM reduces the stress on the links by 8 to 4 times depending on the scenario.

	STRESS			RESOURCE USAGE	STRETCH for **c7**
	s1	s2 / s3	s4 / s5 / s6 / s7		
CastGate Client	17	8 / 8	4 / 4 / 4 / 4	49	1
CastGate Router	5	2 / 2	2 / 2 / 2 / 2	17	1.33
CastGate with PIM-SM	2	1 / 1	1 / 1 / 1 / 1	8	1.33
Native Multicast	1	1 / 1	1 / 1 / 1 / 1	7	1

Table 1. Evaluated metrics

In order to get a better measure regarding the whole local domain we must look at the resource usage metric. If we compare the CastGate with PIM-SM case to the other CastGate based scenarios we note that it is more efficient 5 times more than the CastGate Client scenario and 2 times more than the CastGate Router scenario. Comparing it with native multicast shows that the resource usage is higher with 15%.

The stretch was determined considering R1 the source of the data. We made this assumption because of our interests in these metrics only from the point of view of the local domain. The delay from the actual source to router R1 has the same value, independent of the scenarios we considered. The reference delay for unicast in our local domain from router R1 to any of the hosts c1...c16 has the value 3. Note the increase in the CastGate Router and CastGate with PIM-SM scenario. This increase in delay compared to the unicast delay is due to the fact that in these scenarios traffic is first tunneled to a device in the network and then this devices forwards it through native multicast. This means that data has to pass the same link twice.

5. Conclusion

The focus of this paper is the evaluation of the different versions of CastGate. The solution proposed in [9] is further extended to allow multicast data to be sent from the local domain, not only received. The following metrics: stress, resource usage and stretch were determined on a given network topology. As results show, stress and resource usage decreases significantly when CastGate with PIM-SM is used. The values are very to close to the native multicast scenario. The resource usage for CastGate with PIM-SM is only 15% higher that for native multicast, while it is 2 to 5 times less than the other CastGate scenarios. We must also notice the increase in stretch for the CastGate Router and CastGate with PIM-SM scenarios.

However these results must be confirmed by practical experiments. Metrics like control overhead and join latency can only be determined this way. As further work we want to determine these metrics and compare them with existing results for IPv6 PIM.

At this point we can state that CastGate is a possible solution until native multicast is fully available, and we recommend the use of CastGate with PIM-SM where possible due to better efficiency.

References

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

[2] C. Diot, et al., "Deployment Issues for the IP Multicast Service and Architecture", *IEEE Network*, 2000, pp. 78–88.

[3] Y. Chu, S. Rao, and H. Zhang, "A Case for End System Multicast", *Proceedings of the ACM SIGMETRICS*, 2000.

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

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

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

[7] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*", draft-ietf-pim-sm-v2-new-11.txt*, 25 October 2004.

[8] Beau Williamson, *Developing IP Multicast Networks*, Volume 1, Cisco Press, 2001.

[9] Tudor Mihai Blaga, et al., "Steps towards Native IPv6 Multicast: CastGate Router with PIM-SM Support", *Proceedings of the 14th IEEE Workshop on Local and Metropolitan Area Networks - LANMAN 2005*, Chania, Greece, 2005.