

IGMPv3-based method for avoiding DoS attacks in Multicast-enabled networks

Antonio F. Gómez-Skarmeta,
Angel L. Mateo Martínez
Dept. of Computer Science, Artificial Intelligence and Electronics,
University of Murcia.
Campus de Espinardo. E-30071. Murcia, Spain.
skarmeta@dif.um.es
amateo@dif.um.es

Pedro M. Ruiz Martínez
Area of Telematics Engineering,
Univ. Carlos III of Madrid.
Campus de Leganés. E-28911.
Leganés (Madrid), Spain.
pedrom@it.uc3m.es

18th May 2000

Abstract

IP Multicast has proven to be very good for many-to-many multimedia communications like audio and video-conferencing. However, there are only few Internet Service Providers (ISPs) offering it as a true Internet service. Nowadays, IP Multicast has various issues that are not solved yet and that are making ISPs to think twice before offering IP Multicast to their customers. Some of these issues are billing, authentication, security, protection against malicious people and so on. In fact, nowadays it is very easy for a malicious user to get an IP Multicast-enabled network clogged. In this paper we will describe how to solve these problems.

Keywords: IP Multicast, IGMPv3, Multicast Authentication, Multicast Security

1 Introduction

Internet has undergone lots of improvements during the last years, making possible the development of new interactive services like videoconferencing. However, these services are bandwidth intensive and if we do not make a good network engineering, our network could be clogged. The key when dealing with multimedia traffic is to replicate the flows as few times as possible.

For avoiding having various flows carrying the same content over the same link, recent popular applications like *NetMeeting* or *CU-SeeMe* use the concept of *reflector*. A reflector provides for the replication of streams, that is to say, after receiving a datagram from a participant, it sends

a copy of that datagram to the rest of the people taking part in the session. In this manner, if we place several *reflectors* in cascade, we can save bandwidth.

As an alternative to the use of this kind of replication, it is possible to use IP Multicast[2]. IP Multicast uses group addresses instead of host addresses. So, in order to be IP Multicast compliant, an Internet host must be able to receive packets addressed to its IP unicast address as well as packets addressed to some of the groups it has joined. So, the IP Multicast model works as follows:

- IP Multicast senders only need to address IP Multicast datagrams to the proper IP Multicast address in the range 224.0.0.0 to 239.255.255.255.
- IP Multicast receivers need to join the IP Multicast group they want to receive. To inform their local attached multicast-enabled router which groups it is interested on, a host uses the Internet Group Management Protocol[4] (IGMP).
- Routers will conspire to deliver IP Multicast traffic from senders to receivers. For this task to be accomplished, routers will use IP Multicast routing protocols to interact between them.

These new multicast routing equipments, act like a normal router (in fact most of the routers in the market can act as a multicast router). The main difference is that they use a multicast routing algorithm in order to learn IP multicast routing information. When such a router receives a packet addressed to a multicast group, it will select the output interfaces that must be used to forward that datagram based

on the routing information that the router knows. So, IP Multicast routing protocols differ on what information the router uses for routing decisions and how the router learns that information.

During the last years, a lot of work has been done in the IP Multicast routing area. In fact, IP Multicast routing protocols have evolved a lot since the old IP-in-IP tunnels[3] building a virtual IP Multicast Backbone called *MBone*[1]. New intradomain multicast routing protocols have come up and nowadays IP Multicast works very well within the same Autonomous System (AS). The problems arising now are related with the deployment of IP Multicast natively in the whole Internet.

New protocols for interdomain multicast routing have also come up. In fact, Multicast Source Discovery Protocol [5](MSDP) and Multiprotocol Extensions to BGP [9](MBGP) are working fine these days. However, there are some scalability issues that are making the IETF to think in Border Gateway Multicast Protocol[10] (BGMP) to be the Interdomain multicast routing protocol used in a near future.

So, if we look at a picture of the current IP Multicast state, we could think that this technology is mature enough. However, if we look deeply, we can see that not enough attention have been paid to some key issues like security and authentication.

Nowadays, security authentication and so on, are necessary in most of the services offered over the current Internet. Almost every Internet service involve money transactions taking place. The killer example is the boom that E-Commerce has experienced. So, if you intend to make IP Multicast to be used for serious matters, like pay-per-view content, teleteaching, etc, you cannot permit anyone to interrupt the service to your customers because it could cost you a lot of money.

So, for IP Multicast to be totally deployed in the Internet, these problems need to be solved. In this paper we will present the current research taking place to solve these problems and we will propose some methods for avoiding these problems. Once these problems get solved, there won't be excuses for not to use IP Multicast and it will become widely used in the whole Internet.

The paper proceeds as follows: Section 2 describes the current work that is being done these days related to these issues. Section 3 identifies the problems not covered yet and describes our proposals to solve these problems. And finally, Section 4 presents the conclusions of the work and describes some future work that we are planning in order to improve these mechanisms.

2 Related Work

As we have previously said, not enough attention has been paid to security problems in the recent IP Multicast-related researches. Everybody has been looking for better compression methods, better scalability, reduction in the state maintained by multicast-enabled routers,... But, they have not taken into account that these issues could break down their researches.

Note that we are not saying that IP Multicast evolution followed wrong steps. In fact, the evolution of the Internet was very similar: people started to think in how to provide an e-mail service before thinking in how to avoid SPAM or how to secure Post Office Protocol (POP) connections. We're only saying that IP Multicast is mature enough right now and it's time to start working on its security issues.

Although there haven't been much researching in this area, there are various works that we will comment on in this section.

2.1 IGMP extensions for IP Multicast Sender and Receiver Authentication

In August 1998, Norihiro Ishikawa et al. proposed an Internet-Draft called *IGMP Extension for Authentication of IP Multicast Senders and Receivers*[8]. This draft, proposed (as its name suggests) some additions to the IGMPv2[4] protocol for preventing unauthorized users from sending and receiving IP Multicast datagrams.

The proposed schema met the following requirements:

- It allowed for IP Multicast sender authentication.
- It allowed for IP Multicast receiver authentication.
- The authentication mechanism was independent of the IP Multicast routing protocols.

The proposed solution was based on the concept of *ingress routers* which were responsible for senders authentication and *egress routers* that were responsible for receivers authentication. For authenticating, a Challenge-Response method was used in a similar way as CHAP[11]. So, the local attached multicast router was responsible for authentication matters.

Although this proposal is an excellent work and a very good starting point, it has many drawbacks:

- It does not provide for a *strong* authentication method. Authentication is done based on the IP address of the sender/receiver that can be easily spoofed.

- You are supposing that every other network administrators are good guys and they are using this mechanism.
- It does not allow us to authenticate individual persons but only a host.
- It does not check the scope of the sent packets. That is, if somebody is allowed to send ip multicast packets, it can send them to everywhere. It would be nice to allow certain people to only send them to certain regions. (This is also applicable to Administratively Scoped Regions)
- It cannot avoid Denial of Service (DoS) attacks when two or more people conspire to clog a network. Think in somebodys in your network spoofing the IP address of an authorized host.
- In shared media networks, receiver authentication does not work well because when an authorized host joins a session non-authorized hosts can also receive the data. This problem needs to be solved at the application layer by ciphering the session and distributing group-keys to the participants.

2.2 Access control in Multicast Environments

In July 1998 we, the authors of this paper, started to work on solving the problem of access control in multicast environments. Our goal was to offer IP Multicast to all the students in our university without being worried about the damages they could cause by flooding the Spanish Research Network with multimedia data or by disturbing some other sessions taking place in the same time slot.

Our work [12] was based on the same ideas proposed by Ishikawa's draft. But, as we didn't want to modify all the IGMP stacks in the students computers, we decided to use a different approach for informing the local attached multicast router whether a user was allowed to send multicast traffic to a certain multicast group or not. For that task we used a Call/Response protocol (which we called *mcontrol*). This protocol was used to carry information about the filters that the local multicast router had to apply.

We developed a Web-based interface for allowing the administrator to grant or deny permissions for specific users and specific multicast sessions. This interface was able to parse SAP/SDP[6, 7] packets and show a list of active multicast sessions in real time. It also allowed the

administrator to limit the scope of the packets that a particular user could use. You can see this architecture in the Fig. 1.

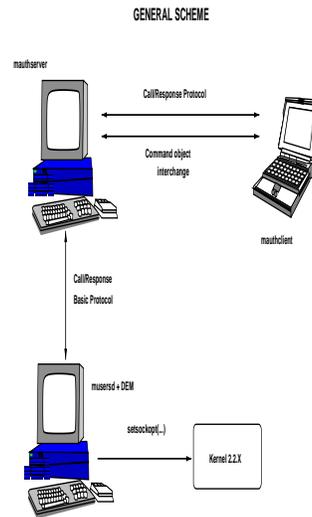


Figure 1: System architecture of our previous work.

Although this approach solves some of the problems mentioned previously, it has some drawbacks:

- It does not solve the problem with multicast receivers.
- It requires the presence of an administrator.
- It avoids clogging somebodys network. However it does not protect our network from DoS attacks caused by people conspiracy.
- The *mcontrol* protocol is not standardized. It would be better to use an standard protocol.

2.3 IGMPv3

When IGMPv3[13] first appeared, our problem related to using an standard protocol for informing the local attached routers about the filters to apply disappeared. In fact, the new version of IGMP now allows for source filtering. That's to say, when a host joins a multicast group, it can tell the router to forward only the traffic addressed to that group coming from certain hosts. Being more specific you can use two kinds of filters:

Include Filters. Inform the router to forward all the data addressed to the group we are joining but only if it comes from the sources listed in the IGMPv3 message.

Exclude Filters. Inform the router to forward all the data addressed to the group we are joining coming from whichever host but those listed in the IGMPv3 message.

Although IGMPv3 helps us when establishing the filters to apply that could help us to protect ourselves from DoS attacks, it does not cover various issues:

- It does not say anything about authentication.
- It does not protect us from DoS attacks caused by users conspiracy¹.

2.4 SDP Source Filters

In May 2000, B. Quinn proposed a new Internet draft called *SDP Source Filters*¹⁴. This new draft describes how to adapt the Session Description Protocol (SDP) to express one or more source addresses as a source filter for one or more destination addresses. Receiver applications will use the SDP source-filter information to identify traffic coming from legitimate senders and discard all other traffic to that specific group.

The idea is that once the application knows the filter to use, it can inform the local attached router (for example using IGMPv3) about the filters to establish. So, we can protect ourselves against DoS attacks.

This approach does not intend to provide authentication, so it does not solve some of the problems we have previously commented on. For example, using the source IP for authentication is very weak and this address could be easily spoofed. So, we need a more robust mechanism.

3 Problems not solved yet

With the picture of the work that has been done in mind, one could think that there is no reason for being worried about a DoS attacks in our network. But, the reality is quite different.

In this section, we will make a broad discussion about the current lacks and how to solve the problems that we have been commenting on.

¹We will explain this kind of attacks in the following section

3.1 Help provided by the RPF-check mechanism

Most of the current IP Multicast routing protocols, use a mechanism called RPF-check^[15] in order to decide whether a multicast datagram needs to be forwarded or not. The mechanism works as follows:

- The router examines the source address of the multicast datagram
- If the datagram arrived on the same interface that the router would have used for sending packets to the source, the RPF-check succeeds and the datagram is forwarded.
- Otherwise the RPF-check fails and the datagram is dropped.

The importance of this mechanism is that it protects our network from DoS attacks coming from out of our network via IP spoofing. This is a natural protection because if a router receives a packet on an interface other than the one towards the *real* source, it will drop the packet. However, this still does not provide a full guarantee of protection. In fact, it is very likely for an interface to be used for accessing both the *real* source and the source doing the spoofing.

In addition, all the IP Multicast routing protocols based on Source Base Trees (SBT) offer a natural protection. This is because the Join Messages will be sent only on the interfaces used for sending datagrams to the *real* source².

3.2 Why IGMPv3 does not suffice?

In Fig. 2, we can observe an example of DoS attack. Using this example we will see how none of the approaches we have previously described suffice for avoiding this type of attack.

²Note that although PIM-SM uses Core Based Trees (CBT), it implements a mechanism for switching to a SBT when a limit in the received traffic is exceeded. In fact, some implementors use as this threshold 0 bps

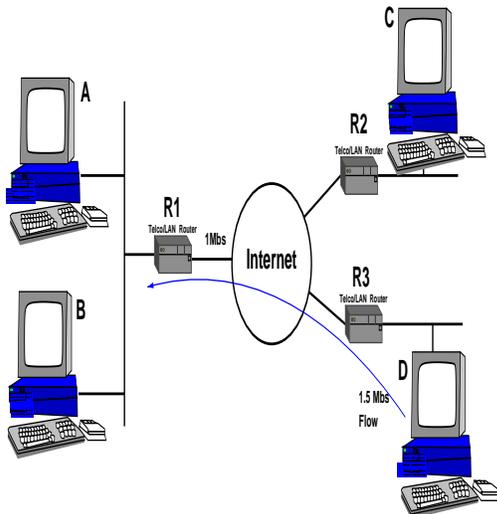


Figure 2: Example of internal DoS attack.

Fig. 2, shows an scenario in which there are two receivers (noted as A and B) in the same network whereas there are also two senders: C is the sender whose IP address is going to be spoofed and D is the host that will spoof C's address. Note also, that the link connecting A and B to the Internet has a bandwidth of 1Mbs.

In this case, the natural RPF-check protection does not help at all. That's because the router R1 has only one interface for reaching whatever host out of its local network. So, all multicast datagrams coming from the Internet will succeed in the RPF check. Thus, it is very easy for host D to send datagrams packets as if he were host C. As host A has joined the group (using IGMPv3 and a filter for only allow packets coming from C), all the traffic coming from D will congest the network at router R1.

But, the problem is not as simple as avoiding spoofing in our network. Let's suppose that the router R1 has some active rules avoiding host D to spoof C's address. A bigger and more difficult problem to solve could come up. Let's suppose that host A has just joined the conference using IGMPv3 and using a filter only allowing datagrams coming from C. As D cannot spoof C's address, that 1.5 Mbps flow will not clog A's network. However, if B and D conspire, B could join the same group (or session) with a filter of allowing datagrams coming from whichever host. So, that traffic would clog the network again³

As you have seen, the only way of being really pro-

³Note that routers usually offer the possibility to limit the bandwidth used by multicast traffic. This can be used for avoiding to get the whole link clogged. However, the DoS attack will cause all active sessions to get a very poor quality. So, it continues being a DoS attack.

ected, is establishing certain policies about who can join what group, and who is doing what thing in your multicast network. Of course, you need to make the defined policy to be accomplished. So, we need to authenticate the people and make the router to only pay attention to commands (like IGMP joins) coming from authorized people. This implies adding some authentication functionality to IGMPv3.

3.3 Why IPSec does not offer us the required functionality?

If one intends to avoid IP spoofing, IPSec¹⁶ could be very useful. In fact, it offers different possibilities that range from integrity to authentication (using the *authentication header* (AH) field). However, it has various problems that made us to look for another option:

- The authentication is basically made based on IP addresses. If authentication is made via user information, that information about the user is not delivered to the upper layer protocol (in our case IGMP). So, it is not possible to check if the user is authorized or not from the received IGMP packet.
- The overload of using the key exchange (via IKE¹⁷ or ISAKMP¹⁸ or whatever) could be so much for such a dynamic protocol as IGMP.

3.4 Adding functionality to IGMPv3

Then, the key idea is to make IGMP to carry authentication information so that the local attached router could decide whether to attend the IGMPv3 Report or not. Although the idea is simple, the problems arise when trying to merge all the concepts: authentication, scalability, etc.

When thinking in this mechanism, we had to take various decisions. The first of them was related to the method for extending IGMPv3. On the one hand, we were thinking in adding a new field carrying all the authentication data. On the other hand, we could use the *Auxiliary Data Field* that had no specific use according to the IGMPv3 draft. At the end, we decided to use the *Auxiliary Data Field*. So, no changes to the IGMPv3 draft needed to be proposed. Fig. 3, shows the format of the new IGMPv3 messages.

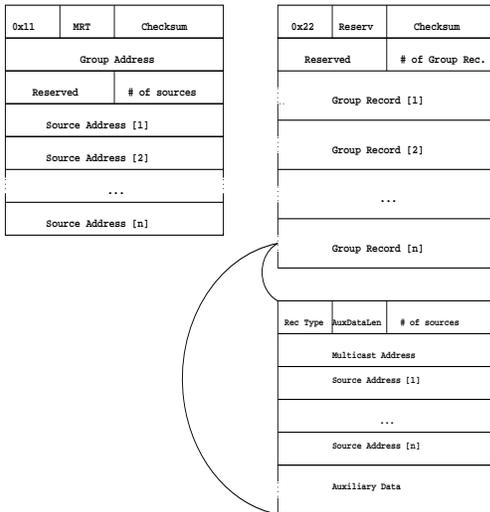


Figure 3: IGMPv3 packets formats. Left IGMPv3 Query. Right IGMPv3 Report.

Once we have decided where to carry the authentication information, many other questions come up. What authentication info should we carry? How will this information be used in order to achieve our goal?. The answer to these questions is very simple. You can carry whatever authentication information you want. The only requirement is to make the router know how can it check if the packet comes from an authorized user.

Observe, that this method is not intended for defining how to authenticate multicast users, but for offering a method for doing the authentication. That's to say, if your organization has a *Public Key Infrastructure* (PKI), it could be likely for you to include the result of hashing the IGMP packet with the user's private key in the IGMPv3 packet, and tell the router to ask a *Lightweight Directory Access Protocol* (LDAP) server for the public key of that user and so, decide whether the user is authorized for joining the multicast group or not. It could also be possible use PGP and tell the router to ask the PGP Key server for the public key of that user or a RADIUS server or whatever. However, for a better performance it would be better for the router to have the information locally. Note that the amount of information is not very big because a router is only responsible for their locally attached users.

As you would probably have noticed, carrying a hash of the IGMP packet is not enough information for the router. What's the problem?. The problem is that the router has to obtain somebody's public key but he has no idea about who he is. So, the IGMP packet needs to include some user information. Fig. 4 shows how to add both the user

information and the hash value into the first *Group Record Auxiliary Data* field.

A new question arises. How can the router know which of the bytes in that field represents the user information and which ones represents the hash value?. The key here is that a hash value has a fixed length in bytes. So, the *Auth-DataLen* field in the First Group Record of the IGMPv3 Report will contain the number of 32-bits words in the Auxiliary Data field. The first bytes in that field will be those corresponding to the hash value, and the rest will be used by the router for identifying the user.

He has to take into account that the *AuxDataLen* field is only 4 bit long. So, we can only use up to 60 bytes for carrying both the user identification and the hash value. Thus, the use of a low-length hash value is encouraged.

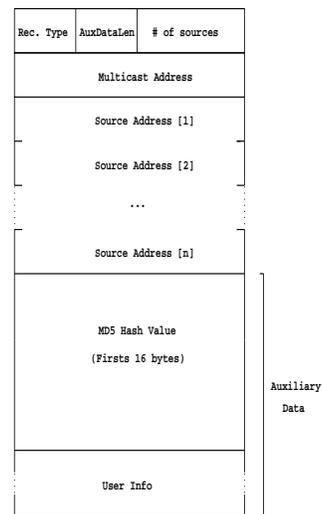


Figure 4: Method for carrying Authentication information in the Auxiliary Data field.

For example, a mechanism that fits very well in these requirements is based on the use of a hash algorithm like MD5. MD5 generates a 16 bytes hash value that would left 44 bytes for the user information. So, the authentication will be made via a shared secret. That is, some value that only is known by the user and the router (for example a password). So, if we think in a user/password authentication, the process could be as follows:

1. The user passes its login and password to the application being used. Then, when the application opens the socket, it will pass the login and password to the IGMP level. Thus, this host will compose the IGMP packet, and will calculate the hash value by applying

the MD5 algorithm to the concatenation of the IGMP packet (with the authentication fields set to zeros) and the *shared secret* (in our case the password). Then, it fills the authentication fields with the calculated hash and login value and sends the packet.

2. When the router receives the packet, he can extract a copy of the IGMP packet and then fill with zeros the authentication fields of the copy. Thus, as long as the router also knows the *shared secret* based on the login information, he can calculate the hash value and check if both hash values are identical. If they are not identical, it drops the packet. In other case, the router is sure that the packet comes from the user and so, he can decide what to do with that packet depending on the policy that has been defined for that user.

There is also a little issue to consider. The behaviour of an IGMP host needs to be slightly different. IGMP defines that after receiving an IGMP report from the designated router, if one of the hosts detects that some other host in the network has sent an IGMP Report to the same group that it was interested on, then it does not need to send the IGMP Report because the router will forward the datagrams addressed to that group. However, with the current approach a host cannot be sure about the validity of the IGMP Report sent by another host in its network because the router could ignore it if it is not coming from an authorized user. So, the host needs to always send the IGMP Report. Note that this is a minor drawback because the number of hosts in the same LAN interested on the same IP multicast group is not expected to be very high.

3.5 Improvements introduced by this method

As you would have noticed, none of the methods proposed in Section 2 solves completely the problems we are talking about. As we saw in Fig. 2, there are several issues that could clog our multicast-enabled network. Some of the key benefits of this work are:

- It allows us to apply user-based policies in multicast networks.
- It allows us to protect our network from DoS attacks coming both from outside and inside our network even when people inside our network conspire with people outside.
- The mechanism works even when the malicious people use spoofing.

- It is almost compatible with the current IETF drafts and standards and only minor changes are needed.
- This mechanism is not intended for providing end-to-end encryption. It is only responsible for network matters. Media ciphering is interesting but it will be done by the application layer.

4 Conclusions and Future Work

This document describes how IGMPv3 can be used to provide some level of security in IP Multicast networks. Its main goal is to propose an extension to IGMPv3 making it to be able to deal with the problem of avoiding DoS attacks and access control in multicast networks.

The main reasons for doing this proposal are:

- Although there are few proposals related to IP Multicast security issues, none of them actually offer a complete solution.
- Besides this, most of these proposals are IGMPv2-based instead of using IGMPv3 that is the current work.
- A mechanism providing access control in IP Multicast networks is needed. Some people think that this problem is not different from access control in Unicast networks. However, multicast traffic has some requirements (real-time, bandwidth consumption, etc) that make this problem to be difficult to solve.

Although IGMPv3 does not directly solve our problems, it offers us a mechanism to:

1. Explicitly define filters. That is to say, it allows a host to inform to its local attached router about which sources he is interested on within a multicast group.
2. It also defines a field to provide IGMPv3 with a mechanism to define extensions to the protocol.

So, our proposal is based on the use of IGMPv3 with some extensions to deal with the problem of access control and DoS attacks. These extensions have been widely described in this paper.

Nevertheless, there are a lot things to be done:

- We have to work in a usable implementation of IGMPv3 while vendors do not implement it. We are thinking in using Linux because of its freely available

source code. However, when as soon as IGMP get standardized, all vendors will have to implementing it.

- IGMP defines an *Auxiliary Data* field. However, the proposed IGMPv3 API 19 does not define any way to specify a value for this field. So, we are going to propose an extension to this API in order to use this field.
- We have torically proved that our proposal solves the problems we have commented on. But, we do not have any testbed demonstrating the correct behavior of our system. So, we need to work in an environment for testing purposes.
- Finally, we proposed a solution, but we think that it will be useless unless we start discussing our proposal within some IETF groups and standardizing this work in the Internet Community.

References

- [1] Vinay Kumar. "MBone: Interactive Multimedia on the Internet". New Riders, 1996. ISBN 1-56205-397-3.
- [2] Steeve Deering. "IP Multicast Extensions for 4.3BSD Unix and related systems". June 1989. Stanford University
- [3] D. Waitzman, C. Partridge, S. Deering. "Distance Vector Multicast Routing Protocol". RFC 1075
- [4] W. Fenner. "Internet Group Management Protocol. Version 2". RFC 2236. November 1997.
- [5] D. Farinacci, Y. Rekhter, P. Lothberg and J. Kilmer. "Multicast Source Discovery Protocol (MSDP)". INTERNET-DRAFT. February 2000.
- [6] Mark Handley. "SAP: Session Announcement Protocol". INTERNET-DRAFT. November 1996.
- [7] M. Handley, V. Jacobson. "SDP: Session Description Protocol". RFC 2327. April 1998.
- [8] N. Ishikawa, N. Yamanouchi, O. Takahashi. "IGMP Extension for Authentication of IP Multicast Senders and Receivers". INTERNET-DRAFT. August 1998.
- [9] T. Bates, R. Chandra, D. Katz and Y. Rekhter. "Multi-protocol Extensions to BGP-4". RFC 2283, February 1998.
- [10] D. Thaler, D. Estrin and D. Meyer. "Border Gateway Multicast Protocol (BGMP): Protocol Specification". INTERNET-DRAFT, November 1998.
- [11] W. Simson. "PPP Challenge Handshake Authentication Protocol (CHAP), RFC 1994, August 1996.
- [12] A. Gómez Skarmeta, A.L. Mateo Martinez and P.M. Ruiz Martinez. "Access Control in Multicast Environments: an approach to senders authentication". Proceedings of the IEEE Lanoms'99, pp 1-13, 1999.
- [13] B. Cain, S. Deering, A. Thyagarajan. "Internet Group Management Protocol, Version 3". INTERNET-DRAFT, November 1999.
- [14] B. Quinn. "SDP Source-Filters". INTERNET-DRAFT. May 2000.
- [15] B. Williamson. "Developing IP Multicast Networks". ISBN 1-57870-077-9
- [16] S. Kent, and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [17] D. Harkins, and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [18] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.
- [19] D. Thaler, B. Fenner and B. Quinn. "Socket Interface Extensions for Multicast Source Filters". INTERNET-DRAFT. February 2000.