Approaches to Multicast over Firewalls: an Analysis

Loïc Oria
Loico@hplp.hpl.hp.com

August 1999

1 Introduction

Most commercial organisations, and increasingly even universities, use firewalls to constrain Internet packets passing between the outside and their internal networks. A firewall is a security gateway that provides numerous advantages to sites by helping to increase overall host security. But firewalls plague the free deployment of multicast on these Intranets, since they do not normally allow the free flow of the UDP packets, which are fundamental to the multicast concept.

Taking this simple statement as a starting point, new solutions must be defined. These solutions should securely let multicast traffic cross firewalls without undermining the security policy. The objective of this paper is to introduce two possible approaches that could provide high security features, to study and compare them.

Section 2 gives a brief introduction to the concepts of multicast and firewalls, as well as a description of the main issues between multicast and firewall. Section 3 looks at few preliminary assumptions, especially about the firewall. Section 4 presents two possible approaches to solve these problems. Section 5 details a comparative study of these solutions across a range of criteria, such as user authentication and logging facilities, compatibility with IPsec… And finally section 6 concludes this report.

2 Background

2.1 Multicast

Multicast was born from the need to efficiently deliver information to multiple recipients. Its aim is to provide a service that allows group communication. A communication group consists of several members receiving all the data sent to the group.

Multicast communication has two major advantages:

- *Simple Addressing*: Data to a multicast group is sent to a single address identifying the multicast group. Every group member receives the data.
- *Lower Resource consumption*: The delivery path in the network forms a tree, so called multicast tree that connects all group members. Data delivered over the multicast tree are not transmitted multiple times on the same link anywhere; they are only copied and transmitted once at appropriate points.

An IP multicast enabled network requires two essential protocol components:

- An IP router-based protocol to allow any routers to communicate with other routers, in order to establish the multicast tree.
- An IP host-based protocol to allow a receiver application to notify immediately neighbouring multicast routers that it has joined the group. This is achieved by the Internet Group Management Protocol (IGMP). Mrouters (routers that support IP Multicast[)] periodically transmit group membership queries in order to determine which groups have members on their directly connected subnets. A host then sends a group membership report for each group it belongs to after a random amount of time to avoid flooding the subnet

The MBone was set-up as a test bed to support and promote the development of multicast applications. It makes use of special routers (mrouters) that are able to distinguish a unicast packet from a multicast packet. Unfortunately, at the moment, most commercial products don't support IP multicasting. In order to link multicast capable networks, the approach is to connect them via tunnel over the unicast network. This mechanism is called IP tunneling.

2.2 Firewall

A firewall is an approach to security. The main purpose of a firewall system is to control access to or from a protected network. A firewall system can be a router, a personal computer, a host, or a collection of hosts, set up specifically to implement a network access policy by forcing connections to pass through the firewall, where they can be examined and evaluated.

2.2.1 What a firewall can do?

- *Protection from Vulnerable Services.* A firewall can greatly improve network security and reduce risks to hosts on the subnet by filtering inherently insecure services.

- *Controlled Access to Site Systems.* A firewall also provides the ability to control access to site systems. For example, some hosts can be made reachable from outside networks, whereas others can be effectively protected against unwanted access.

- *Concentrated Security*. A firewall can actually be less expensive for an organisation. For most modified software and additional security software could be located on the firewall systems. They are then easier to update or manage whereas distributed software on many hosts would be more difficult to control.

- *Enhanced Privacy*. A firewall can hide relevant information that would be useful to an attacker, such as local IP addresses, user identifications…

- *Logging and Statistics on Network Use.* If all access to and from the Internet passes through a firewall, the firewall can log accesses and provide valuable statistics about network usage. When suspicious activity occurs, a firewall can also provide details on whether the firewall and network are being probed or attacked. The more information, the better.

2.2.2 What a firewall can not do?

- *Restricted Access to Desirable Services.* The most obvious disadvantage of a firewall is that it may likely block certain services that users would like to access.

- *Large Potential for Back Doors.* Firewalls do not protect against back doors into the site, like unrestricted modem access or printers.

- *Little Protection from Insider Attacks.* Firewalls generally do not provide protection from insider threats.

- *No Protections against Problems with Higher Level Protocols and Viruses.* There is little protection against data-driven attacks, in which data processed by the clients can contain dangerous instructions to the clients. Firewalls do not protect against users downloading virus-infected personal computer programs from Internet archives or transferring such programs in attachments to e-mail.

- *Throughput.* Firewalls represent a potential bottleneck, since all connections must pass through the firewall and, in some cases, be examined by the firewall.

- *Secure firewall? A firewall system concentrates security in one spot. Then a compromise of the firewall could give access to other less-protected systems on the subnet and be disastrous to them. The firewall itself must then be really well protected against all kind of attacks.*

2.2.3 Primary components

A firewall is essentially composed of a Network Policy, packet filters and application gateways.

a) Network Policy
A Network Policy consists of a Service Access Policy that defines those services that will be allowed or explicitly denied from the restricted network, how these services will be used, and the conditions for exceptions to this policy. The Firewall Design Policy describes how the firewall will actually restrict the access and filter the services that were defined in the service access policy.

b) Packet Filter
A packet filter is a router designed for filtering packets as they pass between the router's interfaces. A packet filtering router usually can filter IP packets based on the destination and/or source IP addresses and port numbers. This router looks in its configuration files to see if the datagrams it has received are allowed to be forwarded or not.

c) Application Gateway
To counter some of the weaknesses associated with packet filtering routers, firewalls need to use software applications to forward and filter connections for services. Such an application is referred to as a proxy service, while the host running the proxy service is referred to as an application gateway.

The idea is simple. The user's client program talks to this application gateway instead of directly to the "real" server out on the Internet. If the request is approved, the proxy server talks to the real server on behalf of the client, relays requests from the client to the real server and relays the real server's answers back to the client.

Proxy architecture has several advantages:
- *Proxy services and protocol filtering.* Proxy services allow only those services through for which there is a proxy. For some sites, this degree of security is important, as it guarantees that only those services that are considered safe are allowed through the firewall. Another benefit to using proxy services is that the protocol can be filtered (e.g. for ftp, one could allow mget function and no mput function).

- *Robust authentication and logging.* The application traffic can be pre-authenticated before it reaches internal hosts and can be logged more effectively than if logged with standard host logging.

- *Less-complex filtering rules.* The rules at the packet filtering router will be less complex than they would if the router needed to filter application traffic and direct it to a number of specific systems. The router need only allow application traffic destined for the application gateway and reject the rest.

2.3 Multicast and firewall: issues

2.3.1 Multicast communication, an ability to access all UDP ports

Multicast addressing raises a specific security issue. For it creates an “address alias” that can be used for malicious port probing. Let's assume that a client runs Rat on the multicast address MA and the port P. It joins the group session by sending a IGMP join request message to the mrouter. Because the IGMP message does not include the port number, the mrouter will forward all the datagrams addressed to the multicast address (MA), regardless of the port number. So the client can receive datagrams with a multicast address that it is really interested in but with another port number. The problem is then that the client host (IP and TCP levels) could deliver the data to the application as if it was a unicast datagram. This can depend on the operating system's multicast code. Multicast addressing can then create a kind of "address alias". Thus the problem is that *"when a host joins a multicast group, it gives outsiders the ability to direct traffic to any of its UDP ports, including ports that should be accessible only to the insiders"* [3].

2.3.2 RPC-based services

Thus the multicast model allows an attacker to access any port on a host as soon as this host has joined a multicast group. This fact implies that the port number of the datagrams must be checked and possibly filtered. The question is now how a security administrator can decide which port numbers he can let pass through and which he must filter. The default configuration must be to filter all port number on UDP. This fact is due to the UDP protocol itself. Indeed the problem with UDP is the services that use it, like NFS (Network File System) or NIS (Network Information System, previously Yellow Pages). Blocking access to those service is further complicated by the fact that they are RPC-based, which means that they don't run on a fixed port number on every machine (NFS normally uses port 2049 but it is not a requirement). That's why most firewalls block UDP: this is the only effective way to block access to RPC-based services.

3 Assumption on the firewall architecture

Before exploring solutions to allow multicast traffic securely cross firewalls, some assumptions need to be made about this firewall.

At the moment, most well defined firewall architecture implements a screened subnet firewall.

Two routers are used to create an inner, screened subnet. This subnet (sometimes referred to as the ``DMZ'': Demilitarised Zone or “Open Network”) houses the application gateway, however it could also house information servers, and other systems that require carefully controlled access.

Especially, only specific communications are allowed between them by the following rules. All intra-network communications are allowed. Inter-networks communications between Open Ethernet and External Ethernet, between Open Ethernet and Internal Ethernet are also allowed. But there is no direct connection between the internal and the external Ethernet. Such connection must pass through the Open Ethernet and especially the application gateways.

The screened subnet firewall can be used to locate each component of the firewall on a separate system, thereby achieving greater throughput and flexibility, although at some cost to simplicity. But, each component system of the firewall needs to implement only a specific task, making the systems less complex to configure.

This architecture has several advantages:

- No site system is directly reachable from the Internet and vice versa, as with the dual-homed gateway firewall.
- The two routers provide redundancy in that an attacker would have to subvert both routers to reach site systems directly.

There are at least two different ways to control these communications based on either routing rules or filtering rules.
The Open Network can be delimited by packet filters that also force the requested rules of communication (See fig 1).

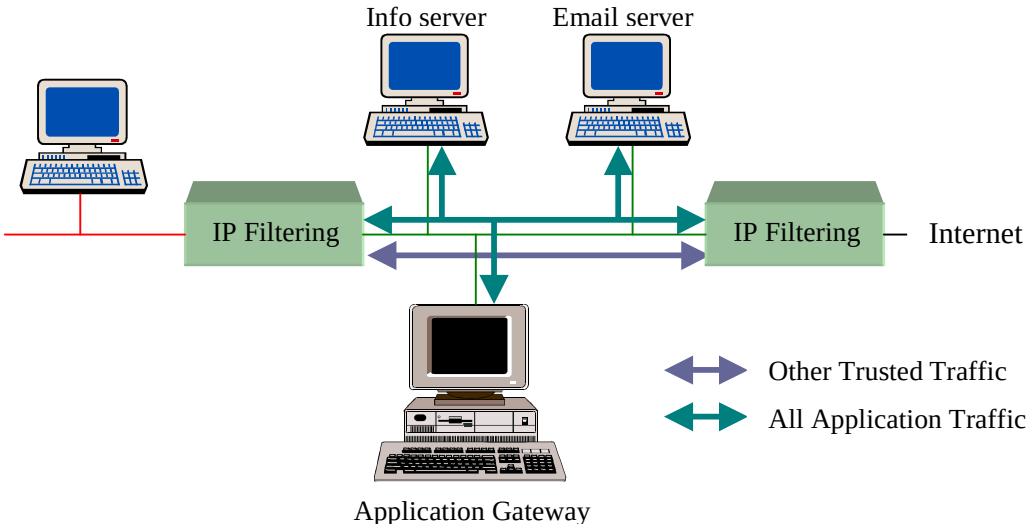

Fig 1: Screened Subnet Firewall with packet filter

The Open Network can be designed via static routing rules (See fig 2). The three networks are connected to the same router but the routing rules of the router prevent the direct communications between the External Network and the Internal Network.

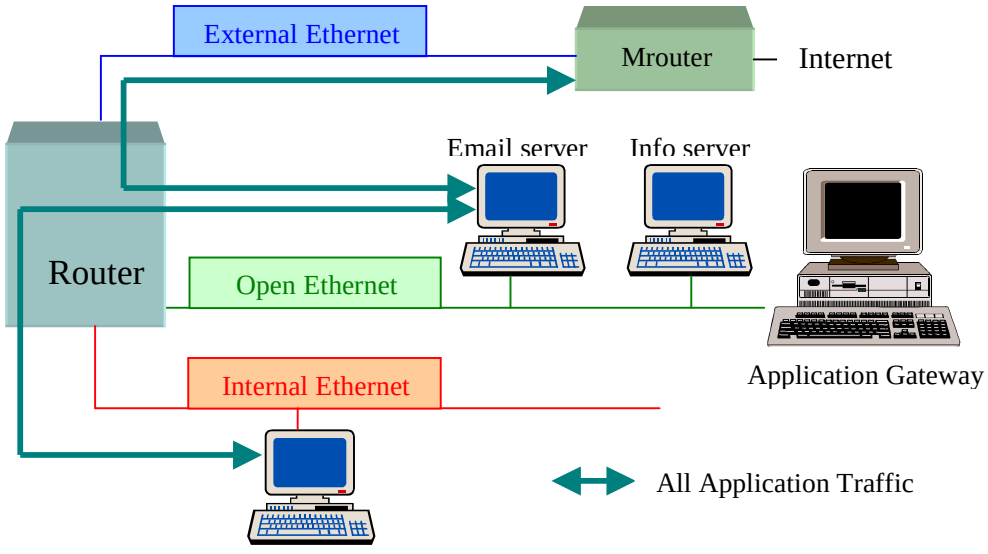

Fig 2: Screened Subnet Firewall with routing rules

In the next parts of this report, the firewall will be assumed to implement a screened subnet architecture in its last version that uses routing rules. Furthermore, we will assume that the firewall implements the following policy. It denies all services by default, but passes those that have been identified as allowed by the administrator.

4 Presentation of two different approaches

This report stems from this simple statement: multicast features require that a possible solution to let multicast traffic cross a firewall provides highly dynamic address and port number filtering facilities. A multicast security policy consists of specifying the set of allowed multicast group addresses and UDP ports that are candidates to be relayed across the firewall. There are two different ways to support such a policy: an “explicit dynamic configuration” of the firewall or an “implicit dynamic configuration”. With an “implicit dynamic configuration”, the set of candidate addresses/ports is implicitly determined, based upon the contents of session announcements. This solution is hereafter referred as to the packet filter solution. In the case of an “explicit dynamic configuration”, an approach called an MBone proxy, this set of candidates could be dynamically set, based upon an explicit request from an internal trusted client.

4.1 Packet filter with dynamic configuration rules

This approach is based on the interpretation of the Sdr announcements and on the IGMP protocol (Internet Group Management Protocol). It uses a mrouter (Multicast capable router) and a packet filter that is part of the firewall.

Sdr (Session Directory) is used to carry and advertise these descriptions. It mainly implements a SAP protocol (Session Announcement Protocol). This SAP protocol plans to periodically broadcast an announcement packet to a well-known address and port (respectively `224.2.127.254` and `9875`).

4.1.1 Mechanism

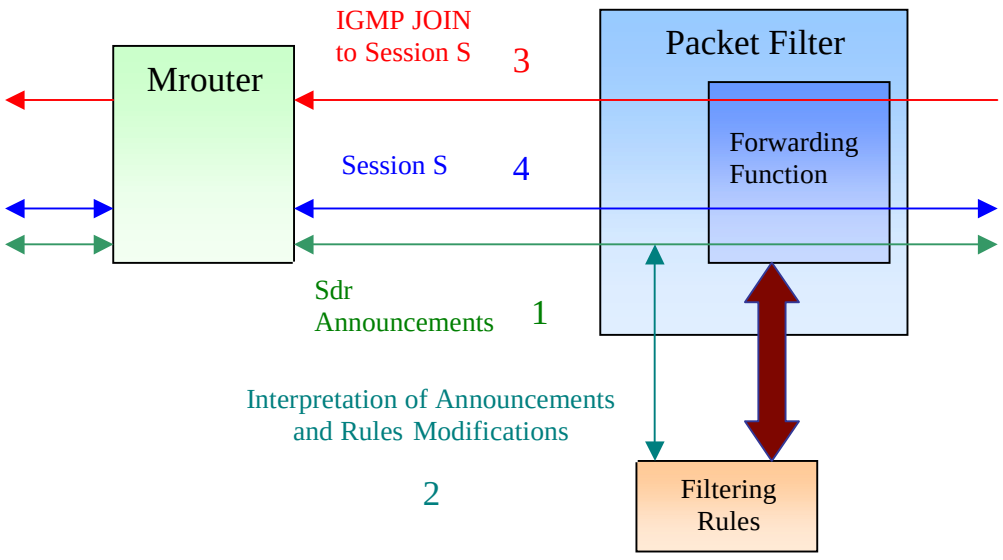

Fig 3: Mechanism of Dynamic Configuration Files Packet Filter

The mechanism has two phases:

- The packet filter runs a small program that listens to the session announcements. Actually, it only listens to the well-known address and port number of the Sdr tool. This program then reads these announcements and interprets them (See fig 3: 2). It identifies the different advertised sessions by pointing out their group address, port number and other relevant features of the announcement. Then it checks that these addresses and ports are within a set of allowed addresses and ports specified by the administrator. If so, this program writes in the configuration files the new rules, i.e. the rules that allow to forward the inbound and outbound packets addressed to a pair of multicast address and port that has been advertised and successfully checked.

- The second phase deals with the client. If a client wants to join a session with GA as a group address, he sends an IGMP join message to the mrouter (See fig 3: 3). This message tells the mrouter that the client is interested by the data addressed to that multicast group and asks it to forward them. As such IGMP message only contains the group address and not the port number, the mrouter will normally forward all the datagrams addressed to the group address (GA), regardless of the port number. In that case, the mrouter will forward too much information on the internal network.

The packet filter device is used to filter datagrams on IP addresses and port numbers. In particular, it will filter this unnecessary information that the mrouter has forwarded. The packet filter will block all packets that are not allowed by its configuration rules. As explained in the first phase, these rules are dynamically created by interpreting SDP announcements. Thus, they only allow the sessions advertised by Sdr. So the packet filter is likely to allow the session that the insider wants to join as well as the other allowed sessions with the same group address (GA). But it will block all other packets addressed to the same group address (GA) but not to unauthorised port numbers.

4.1.2 Restrictions

This solution is highly based on the interpretation of the SDP announcements. This implies several restrictions on the use and implementation of that solution.

- *SDP Security.* SDP announcements are interpreted by the packet filter in order to determine the group address and port number of the announced sessions. As a result, these announcements are supposed to be safe. An attacker must not be able to damage the packet filter by sending fake and dangerous announcements.

- *Private announcements.* As this solution bases its filtering policy on the interpretation of announcements, this information must be available. Now private announcements are possible, by encrypting them. In that case, the packet filter can not access the multicast address and port number of the session. And it can not modify its filtering rules to allow this private session and will then filter it. The most straightforward solution would be to give the packet filter the encryption keys. However, users would be reluctant to give their private keys to the packet filter that can not guarantee the privacy of these keys.

4.2 MBone proxy

In the case of an “explicit dynamic configuration”, the set of group addresses/ports candidates is dynamically set, based upon an explicit request from an internal trusted client.

This scheme is really close to an application gateway architecture. The idea is simple. The user’s client program talks to this proxy server instead of directly to the “real” server out on the Internet. If the request is approved, the proxy server talks to the real server on behalf of

the client, relays requests from the client to the real server and relays the real server's answers back to the client.

4.2.1 Mechanism

This mechanism has three phases.

- Initialisation.

  When a user wants to run a MBone application, the proxy client program opens a TCP connection to the proxy server on the firewall on a well-known port. If user authentication is needed, an authentication dialogue is initiated. Every time that the client wants to open a connection to a specific group address and port number, it uses this TCP control connection to ask the proxy server to open a connection to this address and port. To do so, it sends a special message to the proxy server with the address and port number that it wants to join and the port that the server will have to use to send datagrams to it. Then the server selects one of its unused UDP ports and sends it to the client that will use it to send UDP datagram to the group address through the proxy server. The proxy server then joins the requested multicast group via an IGMP JOIN message.

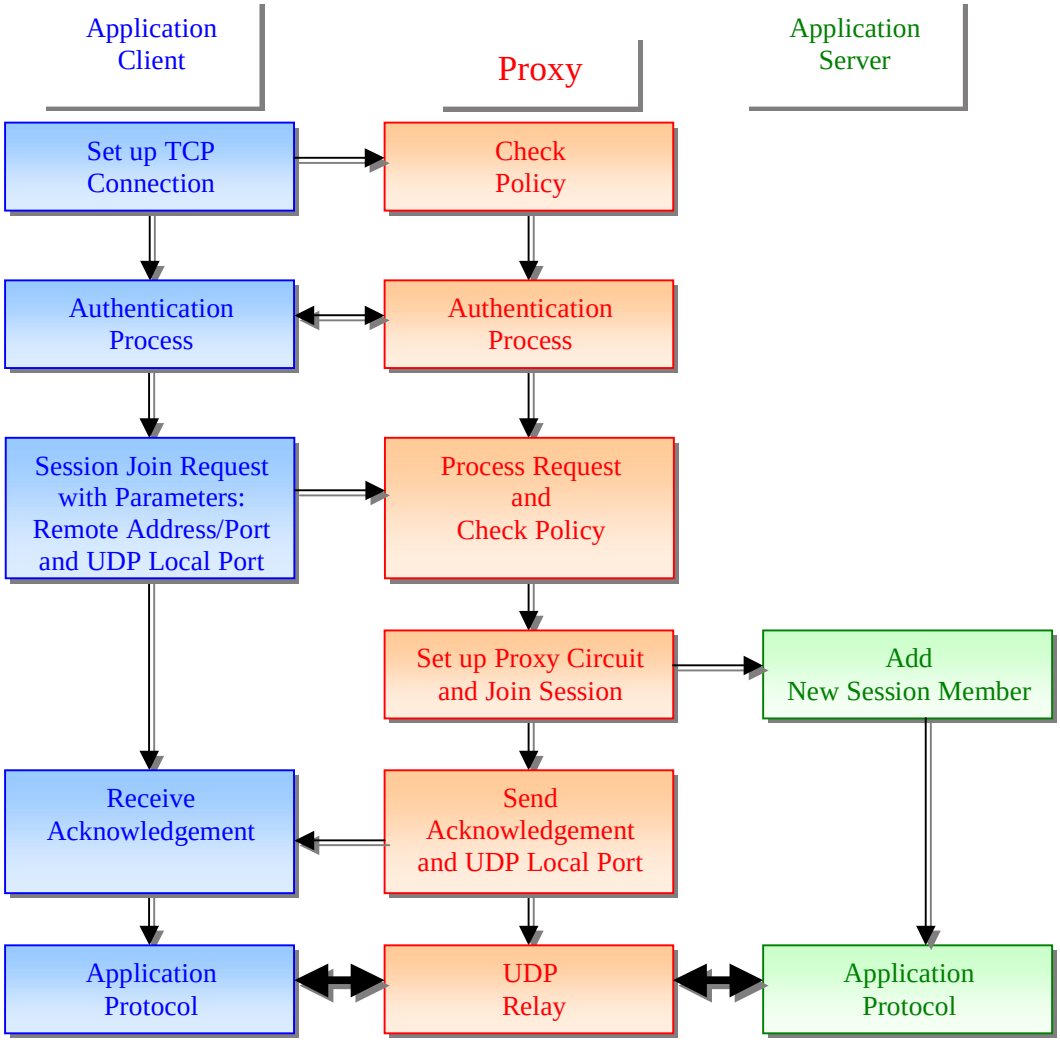

Fig 5: UDP Proxy Mechanism, *"Socksified"* Version

Before opening such connection, the proxy server examines the multicast address and port number and checks that they are within a set of allowed addresses and port numbers specified by the administrator. The server also adds the client characteristics in a client list that keeps the address, port, user name for each client that currently participates in a given conference. This is used for the distribution of outbound datagram.

- Datagram forwarding
  Then the proxy server waits for the arrival of inbound or outbound datagrams.
  When it receives an inbound multicast datagram, it looks in the client list of this conference and forwards the packet to each client of this list. When it receives an inbound datagram addressed to a conference port for which there is no client, it discards it. This provides protection against external probing of client system port.

  When the proxy server receives an outbound unicast datagram. It multicasts it to the external conference participants. And it also unicasts it to all other internal conference participants that are registered in the client list.

- Termination
  When the user terminates the application, the proxy client side notifies the proxy server that removes its name from the client list. When there is no more clients in the client list, the proxy server sends an IGMP LEAVE message to the mrouter. This message notifies the mrouter that the server is no more interested by the group data. As a result, the mrouter doesn't need to forward them to this subnet if no host wants them.

5 Comparison

In this section, these two solutions are compared across a range of criteria, such as the ability to support a unicast or multicast mode on the Intranet, user authentication and logging facilities, compatibility with IPsec and finally effects on network performance.

5.1 Multicast and/or unicast supporting on the Intranet

Until now, I have implicitly supposed that the Intranet only supports unicast communications. A client then receives data via a unicast connection. But using unicast mode on the Intranet raises scalability issues. Indeed, unicast forwarding does not scale well and can imply poor performance when the number of participants in a conference increases on the internal Ethernet. Since a copy of each packet must be sent to each participant in the session (See fig 4), it can cause excessive amount of datagrams in the Intranet.

- *Why multicast communication on the Intranet is better?*
  Multicast has been designed to solve this scalability issue. Only one packet is sent on the internal Ethernet (See fig 5). And as explained in [6], if multicast routing exists, there should rarely be a compelling reason to replace multicast by multiple unicast.

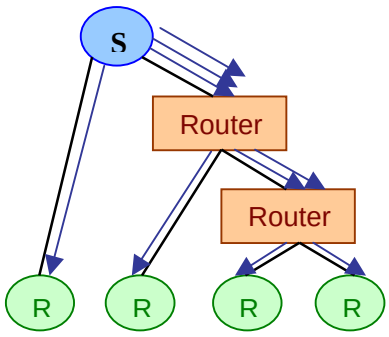

Fig 4: Unicast Communication

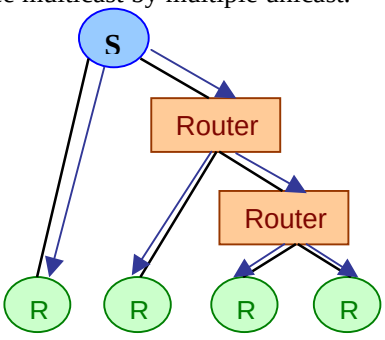

Fig 5: Multicast Communication

- *When can unicast communication be useful?*
  At the moment not every company supports multicast on the Intranet. When multicast is not supported, a unicast mode on the internal network is useful. There is another situation for which unicast is interesting. “Road Warriors” are becoming more and more common. A “Road Warrior” is “*a roaming user - outside the firewall - who wishes to access a private internal multicast session, using a virtual private network*” (See [6]). Using a unicast mode on the Intranet just happens to be a way to allow a remote user to participate in a purely internal multicast session.

Both multicast and unicast communication can be interesting, even if multicast seems to be more natural. That is why for some people unicast communications on the Intranet should only be considered when multicast is not supported. In any case, a network administrator would like to know if the architecture of the considered solutions could support the multicast and/or unicast communication mode.

The UDP proxy solution is able to support both solutions. We have seen that a unicast mode is supported on the Intranet. This is in fact the default option. Few modifications need to be done to change this unicast mode into a multicast mode. For example, we could allow the proxy server to multicast the inbound multicast datagrams. After a user has successfully authenticated and after the session he wants to join has been accepted, the proxy server multicasts inbound data instead of using unicast connections.

On the other hand, the packet filter solution can not support unicast on the Intranet. A translation from multicast to unicast must be done at a special node. The latter will have to run specific software. Indeed, because host IP addresses are needed to handle unicast communications, it must know the IP address of the systems to which it must forward the datagrams. Knowing and storing this information is beyond the capabilities of a packet filter.

5.2 User authentication facility

One of the first services that a system administrator requires is the ability to authenticate users that wish to participate in a multicast session. It is possible to authenticate systems (i.e. IP address) that want to do so. But according to most security policies, host IP address is not sufficient information. User authentication is then necessary in order to control who can send and/or receive data. This is particularly useful when security policy distinguishes between the insider users. For example, different rights will be given to a temporary student or a full project manager.

5.2.1 Localisation and definition of the user authentication device

Before looking at the ways that the different solutions can be used to provide user authentication service, one needs to locate where this authentication must take place, i.e. which device will authenticate the user. One requirement is that this device and authentication scheme must be controlled by the security administrator.

- *UDP proxy*
  In the case of a UDP proxy, the authentication device is easy to find. Indeed, since a communication protocol is already defined between the client and the proxy server, it is quite easy to add an authentication scheme to that protocol. The client opens a TCP connection to the proxy server and starts to exchange messages with it. Once the client and the server have agreed on a unique authentication method, they can run the authentication program, like a “Challenge-Response” mechanism, a “One-Time Password” scheme or more complicated mechanisms.

- *Packet filter with dynamic configuration files*
  In the case of the packet filter with dynamic configuration files, one can make use of the IGMP message exchanges. Indeed, IGMP extensions to allow IP multicast senders and receivers authentication have been defined in an Internet draft (See [8]).

  - When an IP multicast sender wants to send IP multicast datagrams, it sends a message to a mrouter that may need to authenticate it. When this authentication is successful, the mrouter can forward IP multicast datagrams sent by this IP multicast sender. But if the result of the authentication is not successful, the mrouter silently discards IP multicast datagrams sent by the IP multicast sender. This mechanism prevents an unauthorised user from sending IP multicast datagrams to the Internet.

  - When an IP multicast receiver starts to receive IP multicast datagrams, it must join the multicast group address by sending a join message to a mrouter that may want to authenticate it. When the result of the authentication is successful, the mrouter starts to transmit IP multicast datagrams to the IP multicast receiver. If the authentication is not successful, the mrouter does not transmit IP multicast datagrams to the IP multicast receiver. This mechanism prevents an unauthorised user from receiving IP multicast datagrams from the Internet.

  At the moment, this draft forecasts the use of a "Challenge-Response" mechanism but other systems could also be used or developed. The authentication device is then a mrouter. This mrouter must be on the Intranet, on the firewall itself or on the external network of the company in order to be controlled by the administrator. The localisation of this mrouter compared with the packet filter localisation will be studied in section II.3 in this chapter.

5.2.2 User authentication to control who can send or receive multicast data

User authentication mainly aims at controlling who can send or receive multicast data. Can this objective be achieved and how? The security administrator determines a set of possible senders and/or receivers. This information is stored at the UDP proxy server level or at the mrouter level (in case of the packet filter solution). Once a user has authenticated himself, this device can decide whether to forward the data, depending on this set of authorised senders and receivers.

Another issue appears when many different receivers are present on the Intranet and especially if multicast is supported on this Intranet. Indeed, as soon as one IP multicast receiver on this shared media network is authenticated, a multicast router (or a UDP proxy server in the case of the UDP proxy solution) starts to send multicast datagrams to the internal network. As a result, other IP multicast receivers on the network can receive IP multicast datagrams, even if they are not authenticated. And then the administrator can not control who receives multicast data on the Intranet. The most straightforward solution on this issue is the use of encryption. This encryption between the device and the user is independent of the possible encryption of the data itself between end users that provide confidentiality on the Internet. The encryption between the firewall and the user only allows the security administrator to control who, on the internal network, receives and sends multicast data. One possible scenario for the solution is as follows.

- When the authentication of an IP multicast sender is successful, an ingress mrouter (or a UDP proxy server in the case of the UDP proxy solution) sends a group key (i.e. symmetric key) to the IP multicast sender. The key is encrypted with the public key of the IP multicast sender. The IP multicast sender encrypts IP multicast datagrams with the group key and sends them to the ingress mrouter.

- Similarly, when the authentication of an IP multicast receiver is successful, an egress mrouter sends a group key to the IP multicast receiver. The key is encrypted with the public key of the IP multicast receiver. The egress mrouter transmits IP multicast datagrams encrypted with the group key to the IP multicast receiver. The IP multicast receiver decrypts IP multicast datagrams received, using the group key.

This is efficient but a trade-off between security and performance must be studied. Indeed, this encryption between the authentication device and the user, which needs many expensive computations, can decrease the quality of service (Latency…).

5.3 Localisation of the components

Well-designed firewalls distinguish three different networks: the internal network, the external network (Internet) and the open network that is a network without users but only proxies and main servers (Mail server…). Furthermore, all communications between the internal and external networks must pass via the open network. All considered solutions need to install new components on the network. But administration policy differs from the internal or external network to the open network. Indeed company policy generally dictates that the central security manager must agree every decision concerning the Open Ethernet. Such an agreement can be difficult to obtain. On the other hand, it is easier to access and install new elements on the internal or external networks. That's why a discussion about the localisation of the new components is interesting.

5.3.1 UDP proxy solution

The best localisation of the UDP proxy server would be on the Open Ethernet like the other proxies (Mail and Socks for the TCP connections). This solution would permit to implement a more generic proxy. However, typical company policies dictate that every decision concerning the Open Ethernet must be agreed by the central policy manager. Moreover, in order that the proxy is accepted on the Open Ethernet, it must be extremely well implemented because it is exposed to the external attacks; that is the code must be short, very secured… All these facts make this localisation more difficult to apply.

Another solution (See fig 6) would be more specific and would make use of UDP tunneling. The idea would be to divide the proxy server in two parts, one on the internal Ethernet (internal server part) and the other on the external Ethernet (external server part). In order to pass multicast datagrams from the internal Ethernet to the external Ethernet (which is normally forbidden by the router's rules), a UDP tunnel (UDP datagrams in TCP packets) is opened between these two components. This tunnel will use the existing solution based on a TCP proxy situated on the Open Ethernet. The client communicates with the internal server part and exchanges with it the same messages defined in the communication protocol between the proxy server and the client. The proxy client program opens a TCP control connection to the internal proxy server on the firewall. When it wants to open a connection to a specific group address and port number, it sends a special message to that internal proxy server, via this TCP control connection, with the address and port number that it wants to join. Then this internal server opens a UDP tunnel (UDP in TCP) between itself and the external server part by opening a TCP connection dedicated to that session. Then the external server part joins effectively the multicast session. When the external server part receives a datagram addressed to that session, it wraps it, sends it to the internal server part. The latter unwraps it and sends it to the client via the UDP connection previously opened during the communication protocol between the client and the UDP proxy. When the internal server part receives data from a client via the communication protocol, it wraps it, sends it to the external server part that unwraps it and forwards it on the Internet. This last solution is an easier way to set a UDP proxy, regarding the difficulties about the policy of the Open network.

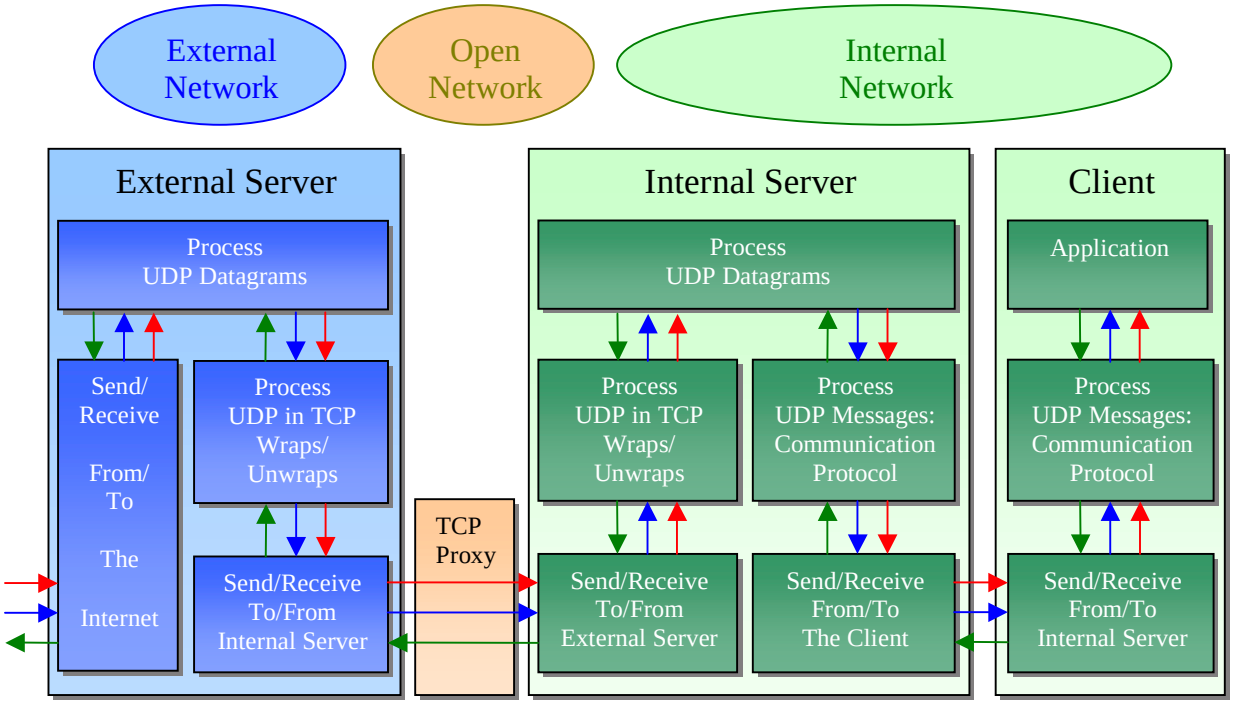

Fig 6: UDP proxy with UDP Tunneling over TCP

5.3.2 Packet filter with dynamic configuration rules

This solution needs two devices: a mrouter and a packet filter. Where can these devices be set? The mrouter must be on the Intranet, on the firewall or on the external network, that the administrator can control in order to allow user authentication via extensions of the protocol IGMP. Three solutions are at least possible. First, the mrouter can be set on the firewall, i.e. on the packet filter itself. Then it could be set on the Intranet before the packet filter and finally after the packet filter on the external network. Let's consider these three solutions.

- *On the internal network:*
  Let's now assume that the mrouter is on the internal network, whereas the packet filter is on the firewall (See fig 7). Since this relevant information is collected by these two separate devices, this implies that relationships between users and sessions are difficult to maintain. Moreover, it implies some new rules to add to the filtering rules of the firewall. Indeed, the mrouter inside the network knows who wants to join a group thanks to the IGMP JOIN message. It has then to transmit this information to the outside in order to ask the other mrouters on the Internet to forward the data towards its network. These exchanges of routing information are achieved via different routing protocols, like DVMRP (Distance vector Multicast Routing Protocol) or MOSPF (Multicast extensions to the Open Shortest Path First). The packet filter must then allow these messages to get through. Actually these protocols use IGMP format header to send their messages. The packet filter then only needs to permit IGMP protocol to get through. IGMP protocol is a protocol over IP with the protocol number 2. IGMP header contains a type field that distinguishes the different protocols; for example DVMRP has a type value equals to Ox13.

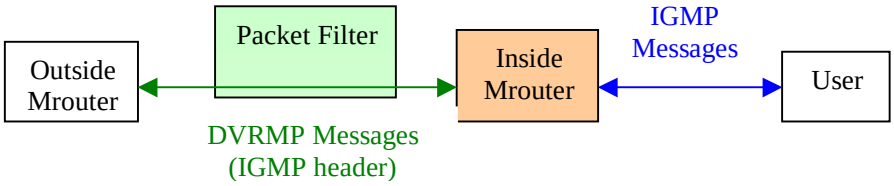

Fig 7: Mrouter on the Internal Network and Configuration Rules

- *On the external network:*
  Let's now assume that the mrouter is on the external network, whereas the packet filter is on the firewall (See fig 8). This implies that relationships between users and sessions are difficult to maintain and that some new rules must be added to the filtering rules of the firewall. Indeed, when a client wants to join a group, he sends an IGMP JOIN message to the mrouter. Then this router will contact other routers to tell them, via routing protocols that someone on the internal network is interested by data from this multicast group in order that these other mrouters forward the data towards its sub-network. As the mrouter is on the external network, IGMP messages must cross the firewall. As a result, the filtering rules must allow IGMP traffic to pass through the firewall. IGMP protocol is a protocol over IP with the protocol number 2.

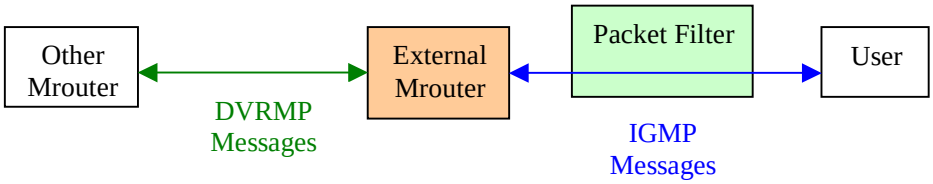

Fig 8: Mrouter on the Eternal Network and Configuration Rules

- On the firewall itself:
  The administrator can easily monitor and control this system. As the mrouter is on the firewall, no special filtering rules are requires. Indeed the firewall has nothing to forward since packets are directly addressed to it. It has not to forward the IGMP message to the outside and has not to forward the multicast routing information to the inside (See fig 9). Actually, this architecture would look like a UDP proxy system: the client has to contact the firewall to authenticate himself and ask to join a multicast session and then the firewall forwards the data after checking the policy. The packet filter solution uses IGMP message whereas the UDP proxy needs its own protocol.

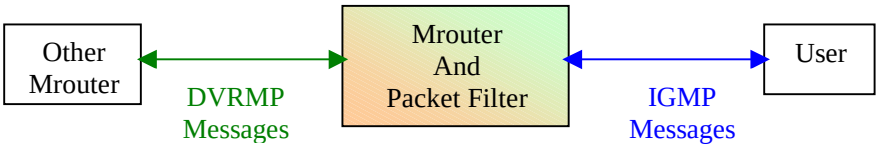

Fig 9: Mrouter on the firewall

5.3.3 Conclusion

Ultimately, the MBone proxy solution seems to be less flexible than the packet filter solution regarding the possible localisation of the different devices. The MBone proxy server is basically designed to be on the Open Network. However, in order to avoid some strict policy concerning this open Network, a practical solution could be developed, making use of the existing TCP proxy. The packet filter with dynamic filtering rules solution has more acceptable places to set the packet filter and the mrouter. All these possible positions seem to be equivalent. Perhaps, a mrouter on the firewall itself is easier to implement since it does not require changes of the filtering rules of the firewall (to allow IGMP traffic to get through).

Another aspect of the packet filter solution needs to be studied. Can firewall configuration safely let IGMP traffic get through?

5.4 Logging facilities

Another important facility that is required by security administrators is the ability to provide extensive audit logging capabilities. It is important for two main reasons:

- First it is one of the best methods of determining if your firewall is performing as it should be. If everything the firewall does is logged, an administrator is then able to examine the logs to determine exactly what it is doing and decide if that is what it is supposed to be doing.

- Secondly, the security administrators want to know who is bringing in multicast feeds, who is trying to send multicast outside of the organisation, user's name, IP addresses, port numbers… The more information, the better. They want to tie as much information as possible to a user. These audit logs are especially useful in post-attack analysis. So when someone does successfully break into the firewall, the system logs are one of the primary mechanisms to determine exactly what happened. By examining these logs and exploring what went wrong, an administrator should be able to keep such a break-in from happening again, and eventually find the intruder.

5.4.1 Packet filter with dynamic configuration files

This solution contains two distinct devices: a packet filter and a mrouter able to authenticate the users. These two devices can log different information.

- A packet filter usually gives the options of logging all of the packets it drops. Actually the administrator wants to know about any packets that are blocked by the packet filtering rules. For these rules reflect his security policy and he wants to know when someone tries to violate them. A packet filter also allows to log selected packets that were accepted. For example, it can be worth logging the start of each TCP connection. Of course, logging all accepted packets is not possible since it would generate too much data, but it can be useful for debugging and dealing with attacks in progress. In the case of a dynamic packet filtering, this packet filter filters packets depending on the addresses and port. These rules are defined by the interpretation of SDP announcements. So when this packet filter drops a packet, it was addressed to an unauthorised session. Someone could have tried to use multicast to maliciously deliver data to the system. That is why such drops are important to log.

- The mrouter is aware of the users because they must authenticate themselves to it via extensions of the IGMP protocol. The mrouter is then able to log relevant information concerning users' identity and other characteristics, like the group address they wanted to join. But as an IGMP JOIN message does not specify the destination port number. The router is not able to log the correspondences between a user and a session (address AND port).

A drawback of the packet filter with dynamic configuration rules is that two different devices are needed. The packet filter knows the authorised multicast sessions (i.e. multicast group address and port number) whereas the mrouter can authenticate the user and the group address he has joined. Relationships between users and sessions are difficult to determine since the information is stored in different places. By relationship, I mean that a user has joined a special session. However, this limitation can be partially avoided if the mrouter and the packet filter are the same machine. In that case, such relationships are easily determined.

On the other hand, one group address (GA) supports many sessions with different port number. In that case, an administrator is not able to determine which session a user has joined. Indeed the user tells the mrouter that he wants to receive all datagrams addressed to the group address (GA) (he sends an IGMP JOIN message). The mrouter will forward every session with group address (GA). The user will then receive all these sessions, even if just one interests him. The administrator can declare that this user receives all these sessions. He can not tell which session the user effectively handles. This is obviously an important drawback of the packet filter solution.

5.4.2 UDP proxy

A UDP proxy has not this disadvantage since it provides a central point where the relevant information is present. Moreover the proxy server is aware of the correspondences between a user and a session, not only the group address. For the communication protocol between the client and the proxy server forecasts the exchange of such information: group address, port number and user ID.

5.4.3 Conclusion

A proxy permits to provide logging at the session level whereas a packet filter can only provide logging at the transport level. Because a proxy server understands the underlying protocol, it allows logging to be performed in a particularly effective way. For example, instead of logging all of the data transferred, a proxy server only logs the commands issued and the server responses received. This results in a much smaller and more useful logging files.

5.5 Compatibility with IPsec

5.5.1 IPsec: definitions

IPsec focuses on the security that can be provided at the IP-layer of the network. The set of security services offered includes access control, integrity, data origin authentication, protection against replay (a form of partial sequence integrity) and confidentiality. As these services are provided at the IP level, they can be used by any higher layer protocol, such as TCP, UDP, and ICMP… IPsec security is based on the wide use of cryptography. IPsec is defined by the associations of security protocols, a concept of security associations (SA) and a mechanism of key and SA management.

a) Security protocols

IPsec relies on two protocols to provide those security services: Authentication Header (AH) and Encapsulating Security Payload (ESP). These protocols have been presented and first described in IETF RFC's ([10] and [11] respectively).

- The Authentication Header (AH) is a mechanism for providing strong integrity, authentication for IP datagrams and optionally anti-replay service.

- IPsec provides confidentiality services through Encapsulating Security Payload (ESP). ESP can also provide data origin authentication, connectionless integrity, and anti-reply service. Confidentiality can be selected independent of all other services. It is achieved by encrypting the upper layer data.

These two protocols add new headers to the IP datagram. They are calculated by taking into account as much of the IP header as possible, as well as the upper level protocol data. The complete IP datagram can not be used to compute this additional item because some fields may change in transit or their value, when arrived at the receiver, can not be predicted by the sender.

b) Security Associations

This section presents the concept of Security Association (SA). This model is fundamental to IPsec since both AH and ESP make use of SAs. A SA is a set of security information relating to a given network connection or set of connections, that affords security services to the traffic carried by it. The combination of a given Security Parameter Index (SPI), a security protocol (AH or ESP) and destination address uniquely identifies a particular SA. The destination address can be a unicast, broadcast or multicast IP address. A SA is unidirectional and must be associated to a single security protocol.

Two types of SAs are defined in IPsec architecture: “transport mode” and “tunnel mode”.

- A transport mode SA provides security services at the transport layer in the case of ESP, whereas the protection is extended to selected fields of the IP header in the case of AH.
- Tunnel mode encapsulates an entire IP datagram. For a tunnel mode SA, there is an “outer” IP header that specifies the IPsec destination, and an “inner” header that specifies the ultimate destination for the packet. In the case of AH, a part of the outer header is protected by AH header, as well as the total IP packet. In the case of ESP, the protection is only applied to the IP packet and not to the outer header.

IPsec is used to protect one or more “paths” between a pair of hosts, between a pair of security gateways (Intermediate system that implements IPsec protocol) or between a security gateway and a host. A Transport mode SA is necessary between two hosts. A Tunnel mode SA can be between whatever devices, security gateway or host.

c) Key and SA Management

IPsec is highly based on cryptographic technologies and the concept of Security Associations. This implies a cryptographic key management and the exchange of SA parameters (such as SPI: Security Parameter Index) and SA management.

This SA and key management can be achieved in two different ways.

- *Manual techniques.* This is the simplest form of management. A person manually configures each system with keys and SA data relevant to secure communications with other systems. This technique could be used to create a Virtual Private Network.
- *Automated SA and Key Management.* An administrator is not always able to determine in advance which SA would be needed. As a result, SA could be created on-demand. The default automated key management protocol is IKE (Internet Key Exchange [12]).

5.5.2 IPsec and Multicast

Unlike the unicast IPsec, multicast groups can have one or more senders, and one or more receivers. In the case of unicast IPsec, the destination system will normally select the SPI and other SA parameters. Some modifications of the concept of IPsec Security Association are then necessary to fit with multicast environment. For example, as there can be many receivers in multicast, this raises the issue of who selects the SPI and others SA parameters.

Some solutions are currently under investigation, especially at the SMuG (Secure Multicast Research Group). SMuG is an IRTF (Internet Research Task Force) Research Group formed to discuss issues related to multicast security.

Actually, this interesting subject is out of the scope of this document, since it is specific to multicast communications and has nothing to do with firewall issues.

5.5.3 IPsec, packet filter, proxy and firewall

The use of IPsec with the MBone proxy or the packet filter with dynamic configuration files raises a specific issue. Two cases must be distinguished here, depending on the Security Association features.

a) Security Association from firewall-to-firewall

If a SA is set between two firewalls, a Security Gateway (SG) is present on the firewall. This Security Gateway could easily be the packet filter in the case of the packet filter solution, or the proxy server in the case of the MBone proxy solution. The SG knows every relevant information about the SA, and especially all the encryption keys. It is then able to decrypt inbound IP packets and access the destination multicast address and port number. The SG, either the packet filter or the proxy server, can now forward the datagrams to the client on the inside. On the other hand, it is obviously able to forward outbound traffic. It receives unsecured IP packets, computes the IPsec datagrams with the appropriate keys and sends it to the Internet, depending on the SA parameters.

b) Security Association from host-to-host

(i) Main issues

As explained in [13], this situation must face at least three main problems:

- The host will be responsible for verifying the Authentication Header or ESP. The firewall is then forced to trust him to check them. Many security administrators would be reluctant to make such assumption.

- In both solutions (Packet Filter or MBone proxy), the relay and filter policy is based on destination IP address and UDP or TCP port numbers. When a host uses ESP, the port numbers will be encrypting. This will deny the packet filter and the proxy server to access to this information. This is not the case if only AH is applied; if the device is able to parse IPsec headers, it can find the appropriate information.

- The packet filter and the proxy server can not distinguish between ESP transport mode and ESP tunnel mode, since this information is only available after decryption of the IPsec packet. The problem is then that malicious host can use ESP in tunnel mode to access hosts and services that are otherwise not accessible.

  Let’s assume that a Security Gateway (SG) is set on the internal side of a firewall (See fig 24). H1 is a host outside the firewall. H2 is a host on the Intranet and normally H1 is not allowed by the firewall to access H2. Let’s suppose that H1 is able to open an ESP Security Association in tunnel mode with SG. The “outer” IPsec header would contain SG IP address; but the “inner” IPsec header could contain H2 IP address. In that case, H1 can send a packet to SG through this ESP SA (this is allowed by the firewall). SG decrypts this packet and discovers in the “inner” IPsec header that the final receiver of this packet is actually H2. If nothing in SG rules prevents it from forwarding packets to H2, it will effectively send the IP packet to H2. H1 is then able to access H2 without the firewall authorisation.

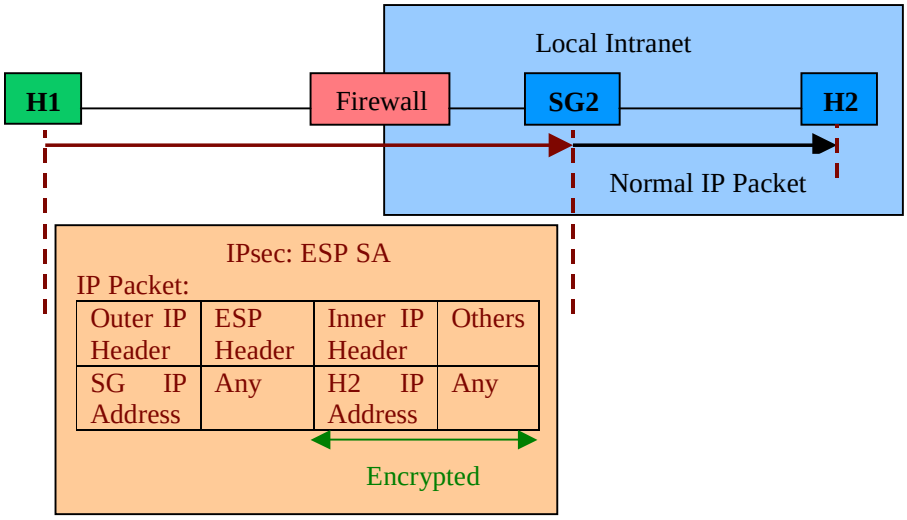

Fig 24: IP Connectivity to Forbidden Host via ESP SA.

- IPsec, NAT (Network Address Translation) and proxy
  NAT (Network Address Translation) is special technique that consists of hiding internal addresses of a private network behind one or several addresses that generally belong to a gateway machine. The address translation is performed by overwriting the source or destination address in the IP header. At the moment, unlike the packet filter solution, the MBone proxy is designed to achieve this address translation. This technique is supposed to improve the security of the Intranet by hiding the internal IP addresses, that could be used by attackers to access the internal network.

  There is strong potential for conflict between IPsec and NAT. The IPsec authentication mechanism has been explicitly designed to detect what NAT is good at. That is altering the header of the packet. This is for example the case in the following situation. A proxy server that implements that address translation receives a packet from an internal user. It will replace the IP address of that packet by its own external IP address before sending this new IP packet to the original destination host. The latter will receive it and digest the AH message. The AH process will then invalidate the packet since the IP header has been previously modified. The packet will then be discarded.

  The coexistence between NAT and IPsec is very uneasy. IPsec developers generally claim that NAT is useless and that it should be eliminated whereas the NAT defenders argue that IPsec should not break NAT.

  The conflict can sometimes be avoided by using tunnel mode. The IP header can sometimes be left unprotected. Indeed, in the ESP tunnel mode, the outer header is not included in the protected fields, either by the authentication header or by the encryption. The MBone proxy can then modify the outer protocol to hide the internal IP addresses.

(ii) “Give me the keys!”, possible solution for AH and ESP.

The previous section (i) has pointed out the main obstacle for the use of IPsec with one of the two considered solutions is that IPsec can deny the access to essential information like the final destination IP address and port number.

The most straightforward solution would be to give the packet filter or the proxy server access to the encryption keys. The security administrator can manually configure the appropriate device (packet filter or proxy server) with these different encryption keys. Of course, this static configuration needs that these keys are known in advance by the administrator. Another approach would consist in letting the host contact the device and tell this device the encryption keys. Such communication is not easy to implement between a host and a packet filter. But on the other hand, as the proxy solution already implements a communication protocol between the proxy and the client, the transmission of the encryption keys from the host to the server is easy to add to this existing protocol. This also solves the problem of the IP address translation. As the proxy server knows the encryption keys, it can manipulate IP addresses and recalculate the integrity check information.

However, giving the keys to the proxy and the server is not without danger. These devices are not always well protected against attacks. What would then happen if an outsider could access it and its private files? He would be able to decrypt the data and read it. He would also be able to send data on behalf of the inside user. Because of this danger, many users will be reluctant to give their private keys to the packet filter or the proxy server.

(iii) Special possible adaptations for AH only

ESP raises the main restricting issue. It encrypts important information, like port numbers and IP addresses. This fact makes ESP very difficult to apply in collaboration with either the packet filter solution or the proxy solution, in a host-to-host tunnel mode. Authentication Header is much less restrictive, since it does not hide this important information.

The main problem with AH in a host-to-host mode is that the authentication check must be done by the host system. Many security administrators would be reluctant to trust host to check the Authentication Header.

At the moment, the mandatory way to compute the authentication header is a symmetric cryptographic mechanism, so that only the endpoints of the association can check the authentication. A possible idea would be to use an asymmetric cryptographic mechanism like digital signature. In that case, the authentication header is encrypted with the private key of the sender. Anyone who can access the corresponding public key is able to check this authentication header. A packet filter or a proxy server would then be able to do so.

However, this asymmetric option needs a trade-off between security and performance. Asymmetric encryption and decryption consume a large amount of resources, especially CPU. This could then reduce the tool performance.

5.5.4 Conclusion

The use of IPsec inherently creates some conflicts with multicast that are currently under investigation. The packet filter solution and the MBone proxy solution also add some new issues. In particular, IPsec can deny the access to essential information like the final destination IP address and port number, whereas both solutions use this information to filter and relay datagrams.

The easiest way to solve that conflict is either to prevent users from using ESP in a host-to-host mode or to allow the use of IPsec from firewall to firewall. The packet filter or the MBone proxy server is then able to decrypt inbound IP packets and compute outbound packets. This solution also solves the conflict between NAT and IPsec. But the traffic between the host and the firewall is still unencrypted and could be sniffed. If a user also requires confidentiality on the Intranet, he can open a SA with firewall itself. His traffic will then be decrypted and re-encrypted again by the proxy server.

However, encryption and decryption are large CPU consumers. And if these computations are done at the firewall level, this could reduce the quality of service. The firewall could then constitute a bottleneck. A trade-off between security and quality of service is hence necessary.

5.6 Quality of Service and Bandwidth Consumption

This final requirement does not directly deal with security.
Each multimedia application requires a certain level of performance to ensure the information can be processed at the necessary throughput, end-to-end delay, jitter, and error rate. Real-time applications and thus multimedia applications have really strict requirements on these statistical properties, in particular delay and jitter.

The Internet provides a best-effort service to all of its applications. In other words, it makes its best effort to move each packet from sender to receiver as fast as possible, but does not make any promises about delay and jitter. Due to the lack of any special effort to deliver packets in a timely manner to receivers, it is extremely challenging to develop successful multimedia applications for the Internet.

Regarding these difficulties to achieve good performances for real-time applications, it is very important to know how security solutions for firewall traversal will behave and how severe a drain on resources such solutions are in reality, and whether it is satisfactory to organisations.

Of course, to study the implications of the security solutions for firewall traversal, one should undertake experimentation and tests, since such studies should measure the modifications on delays and jitters that were added by these solutions. Anyway few remarks could be done on that subject before any real experiments and tests.

5.6.1 Packet Filter Solution

This solution should not really influence the performances of the MBone tools. Indeed after the user authentication, the data are just forwarded by the packet filter. The packet filter is then like any other routers on the Internet except that it must look in its configuration files before forwarding the data. These checks should not drastically decrease the tool performance. Actually, the packet filter solution relays data at the network or transport layer, never at the upper layer. This implies that, concerning processing performances, it should behave better than the MBone proxy that acts at the application layer.

5.6.2 MBone proxy solution

With this solution, two aspects are concurrent. One of them would conclude that this solution could dramatically reduce the performance whereas the other presents it at an alternative to improve bandwidth allocation.

- *Limit Network Performance*
  While the current generation of firewall products is very effective at preventing network intrusions, they have introduced their own problems to enterprise. In particular, they limit performance and scalability.

  Because firewalls sit on the data path, they can limit network performance and scalability. All network traffic passing between the Internet to the Intranet must first traverse the firewall. Unfortunately, the processing architecture that works best for firewalls is not well suited to examining high volumes of data packets. Consequently, firewall can slow down communications having to process every packet. Scaling the performance of firewalls can be difficult because it generally involves an upgrade to a more powerful server.

The limitation of the network performances is not specific to the MBone proxy but is also true with any firewall architecture, even packet filters. However, these limitations are even more dramatic with a proxy architecture since it processes packets at the application layer or at least at the session layer. This is particularly the case with the MBone proxy solution, adding the fact that the MBone proxy must handle a huge amount of data, like audio or video streams.

- *Improve Bandwidth Allocation*
  Although bandwidth allocation does not directly deal with security, this was an important reason why many network administrators have rejected the introduction of multicast in their internal network. Indeed given that multicast typically carries multimedia contents (audio and video) that are an important bandwidth consumer, they may require capabilities to do resource metering, bandwidth control and allocation. Of course, this can be achieved by other means like the resource allocation protocol. But proxy architecture can ease this objective by providing a central point where bandwidth allocation and resources control facilities can also be provided.

Real investigations with experiments, tests and measurements must be undertaken in order to determine exactly the effects of the MBone proxy on the network performances. This aspect is of the first interest for the future of such solution. And the results of the measurements could balance the final decision of the network administrators.

6 Conclusion

Two different approaches to solve the problem of firewall and multicast have been presented. They both basically provide a way to dynamically allow or block some multicast traffic with respect to several possible rules defined by the administrator. The packet filter with dynamic filtering rules uses the SDP announcement to do so and IGMP traffic to provide user authentication facility. As for the MBone proxy, it relies its policy on an explicit request by a client for a specific session, via a special protocol implemented between the client and the proxy server. I have tried to compare these two solutions with respect to several criteria.

For many of these criteria, they seem to be equivalent, especially concerning the user authentication or the compatibility with IPsec. However, the MBone proxy provides more facilities. In particular, both approaches enhance security by dynamically defining the set of allowed sessions. But with the proxy solution, new port numbers are designed between the server and the client to accomplish the data forwarding. This means that the client can dynamically determine and control on which port he will receive data. This is important in the situation where a multicast session has the same port number that a local, eventually dangerous, functionality, like an RPC-based service. The client is able to change the port number and find an unused one to receive data. This extra facility is not provided by the packet filter solution.

The packet filter solution is less selective since it can forward too much information. Particularly, if two sessions use the same group address but different port numbers, it will transmit these two sessions on the Intranet though only one session was requested by a user.

Another small disadvantage of the packet filter solution is that it imposes some restrictions and assumptions. In particular, the session announcements are supposed to be safe and must be interpreted by the packet filter. So private announcements that are encrypted can not be handled by this solution.

A more embarrassing inconvenience of the packet filter firewall is that it provides less efficient logging facilities. An administrator is not able to determine which session a user has joined but only which group address he has requested, since the user can only tell the mrouter that he wants to receive all datagrams addressed to the group address via IGMP messages.

On the other hand, a packet filter solution does not require many lines of codes and seems to be much easier to implement. Moreover, with an application and multicast point of view, this solution could provide sufficient security facilities for a private corporation. It seems to be also more in the spirit of multicast since it makes use of existing facilities provided by multicast, such as the IGMP protocol.

Nevertheless, security administrators will not fail to notice that the MBone proxy solution provides additional services like logging, stronger differentiation and selection of the users with respect to their characteristics. These extra options should make security administrators prefer the MBone proxy solution.

This preference has led me to implement an MBone proxy, in order to study some of the possibilities and limits. In particular, I have defined a communication protocol between the MBone proxy and clients. This prototype has been proved to be compatible with the MBone applications.

While testing the prototype, I have got variable results on the audio quality, especially loss rate and delays. This important aspect of the MBone proxy solution still needs to be investigated. Indeed, because all network traffic passing between the Internet to the Intranet must first traverse the MBone proxy, network performance can be limited. This can even be more dramatic since it must handle a huge amount of data, like audio or video streams. Future investigations must then be undertaken to determine the real effects of the MBone proxy on network performance. This will constitute the main challenge for this solution to be approved and validated by users.

7 References

[1] W.R. Cheswick and S.M. Bellovin,
"Firewalls and Internet Security: Repelling the Wily Hacker"
Addison-Weslet Publishing, 1994

[2] D.B. Chapman and E.D. Zwicky
"Building Internet Firewalls"
O'reilly & Associates, 1995

[3] F. Djahandari and D.F. Sterne
"An MBone Proxy for an Application Gateway Firewall"
IEEE Symposium on Security and Privacy, 1997

[4] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas and L. Jones
"SOCKS Protocol Version 5"
RFC 1928, April 1996

[5] D. Chouinard
"SOCKS V5 UDP and Multicast Extensions"
Internet-Draft "draft-chouinard-aft-socksv5-mult-00.txt", July 1997

[6] R. Finlayson
"IP Multicast and Firewalls"
Internet-Draft "draft-ietf-mboned-mcast-firewall-02.txt", November 1998

[7] M. Handley and V. Jacobson
"SDP: Session Description Protocol"
RFC 2327, April 1998

[8] N. Ishikawa, N. Yamanouchi and O. Takahashi
"IGMP Extension for Authentication of IP Multicast senders and Receivers"
Internet-draft "draft-ishikawa-igmp-auth-01.txt", August 1998

[9] S. Kent
"Security Architecture for the Internet Protocol"
RFC 2401, November 1998

[10] S. Kent
"IP Authentication Header (AH)"
RFC 2402, November 1998

[11] S. Kent
"IP Encapsulation Security Payload (ESP)"
RFC 2406, November 1998

[12] D. Harkins and D. Carrel
"The Internet Key Exchange (IKE)"
RFC 2409, November 1998

[13] U. Ellermann (DFN-CERT)
"IPv6 and Firewalls"
Presented at "SECURICOM -- 14th International Congress on Computer and Communications Security Protection", Paris, June 1996.