Design of a new architecture for audioconferences based on SIP Multicast and Genetic Algorithms

Carlos M. MORENO
Universidad Central de Venezuela
Jefe del Departamento de Comunicaciones. Caracas, Postal Office 1051 , Venezuela

and

Maribel ALVAREZ
Universidad Simón Bolívar
Decanato de Estudios Tecnológicos. Sartenejas, Postal Office 1080 , Venezuela

ABSTRACT

Group services like the audioconference require a minimum level of quality of service for the adequate transmission of multimedia packets, and the improvement of signaling processes for the establishment, maintenance and release of multicast group sessions. Traditionally, the SIP (Session Initiation Protocol) servers have dealt with unicast signaling and complementary support IP (Internet Protocol) multicast. The distribution patterns of the session are described on the SDP (Session Description Protocol) payload. As a consequence, the same existing drawbacks of IP multicast affect the performance of SIP-based requested audioconferences. This work proposes a new multicast architecture based on SIP extensions and genetic algorithms in a multicast manager. Such architecture supports overlay multicast. The group management functions are executed inside the multicast manager. Two testbeds were mounted to measure the joining and leaving time for both IP multicast and SIP multicast with several levels of extra load traffic from a potential group member to the source of multimedia traffic without a genetic algorithm as a first approach. The results show that the difference between the minimum and maximum standard deviation value for both joining and leaving time on the architecture proposed with SIP multicast is lower than with IP multicast for the studied scenarios. Finally, we propose a genetic algorithm and describe a case study.

Keywords: multimedia, multicast, open, source, overlay.

1. INTRODUCTIÓN

Nowadays, voice, video and data traffic are getting integrated in the same networking platform. The design of such architecture has to take into account the applications to be deployed considering the highest layers of the TCP / IP model. One of the most demanded applications is the audioconference between groups, and its improvement is being done over the applicative layer **[1]**. An adequate level of quality of service is required for these applications, like a low response time for joining and leaving a group member, and this can be achieved through the use of multicast. There are three types of multicast: IP multicast **[2]**, Overlay Multicast and Application Layer Multicast. IP multicast is used in IPTV with the use of protocols IGMP and any routing one like PIM-SM. The second approach uses the end systems and intermediate proxies to form the trees. In the third approach the end systems are completely responsible for the creation and destruction of trees.

One architecture for audioconferences with several clients can be implemented with open source projects like SER, OPENSIPS or ASTERISK; or propietary software like 3CX, by using the SIP protocol based on unicast together with some Overlay Multicast complementary implementations. The proposed architecture is based on SIP extensions that include Overlay Multicast. That permits to decrease the joining time and leaving time in some scenarios where an additional traffic is injected towards the source of media. At great scale, it is possible that the response time depends not only on the SIP timers and delays of the network but also on the proces to find the best multicast trees inside a module named Conference Manager. That is the reason why we propose at the end the inclusion of a genetic algorithm.
In chapter 2 we explain the SIP protocol, its relationship with multicast and some research done about genetic algorithms and multicast. In chapter 3 we explain the modules of the proposed architecture. In chapter 4 we describe the experimentation donde and results. In the chapter 5, we study a case study, construct a model and propose a genetic algorithm. Finally, in chapter 6 we explain the conclusions of our study.

2. SIP PROTOCOL AND MULTICAST

SIP PROTOCOL

The SIP protocol permits establish, maintain and liberate a multimedia session **[3]**. It is situated on the applicative layer of the TCP / IP model and is based on a client-server interaction between user agents**[4]**. Two types of messages are defined: requests and responses. It is point-to point oriented basically and additional specifications have been created regarding instant messaging, reliability of provisional responses, updates of description sessions and preconditions for quality of service. Although there exists a specification for the management of multiconferences, the original definition presents strong limitations regarding the use of IP multicast.

MULTICAST WITH SIP

IP multicast can be used with SIP as a discovery-like service for a simple host, where it sends a simple requests to a group of homogeneous servers, and processes the response of only one of them. This functionality is mainly used for registrations with multiple servers (**Fig.1**).
Based on processing rules of transactions, the client accepts the first answer, and will see the others as retransmissions as they have the same identifier Via. Another limitation is that the client

that sends the request needs to be included inside the multicast group it sends the request to.

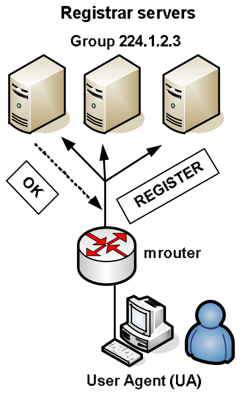

Fig. 1 Registration with multiple servers

Some work has been done since 1998 to allow multicast and unicast conference sessions with SIP **[7]**. In that year, an article published by both Dr. Schulzrinne and Dr. Rosenberg show how IP multicast conferences could be established.

They suggest that a new IP multicast participant needs to get an IP address class D of such group as a first step. Then, it must send an INVITE message to every group member of the conference with a SDP description which indicates that, it wishes to receive media on the multicast group. Other endpoints which are multicast capable reply with a 200-class response, others that are not, reply with a 600-class response. This would allow the new IP multicast participant to know from which users it can expect to receive data through multicast, and which through unicast.

This approach presents the drawbacks of IP multicast in a LAN or WAN. It forces the network administrator to install mrouters, and switches that support dynamic group management protocols like IGMP snooping or CGMP.

In the cases of centralized audioconferences, a simple agent refered as "Focus" maintains a dialog with each participant and coordinates signaling processes. The "Focus" is in charge of manage the SIP signaling (control plane), and the "Mixer" coordinates the transmission of information (data plane). This procedure is defined by IETF**[8]** (**Fig.2**).

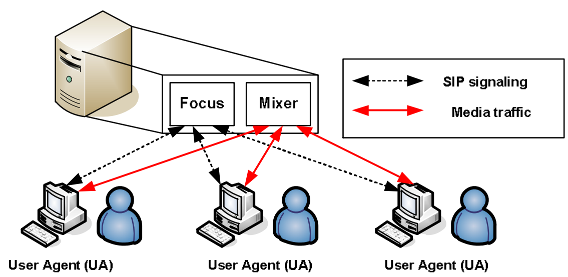

Fig. 2 Use of focus in centralized audioconference

GENETIC ALGORITHMS AND MULTICAST

As a consequence of the use of multimedia applications in high velocity networks, multicast routing techniques have become very important **[2]**. Some work has been done in several areas like finding criteria for multicast routing with quality of service, or its use in NGN with IP multicast, or application layer multicast.

Due to the complexity in the process of formation of trees, which are calculated with high computational demand, thus sacrifying the performance of the routing process we propose the use of a genetic algorithm **[5]**. They use as fitness parameters the end-to-end delay, bandwidth, packet loss rate, jitter or combinations of them. The architecture was firstly evaluated in some scenarios without the use of genetic algorithms, where an extra traffic was injected from a receiver to the source.

3. SIP MULTICAST ARCHITECTURE

We propose a new architecture for fixed networks based on the SIP protocol **[6]**, with Overlay Multicast and groups management (**Fig.3**.) that can be extended to wireless communications on the future. Initially, the use of genetic algorithms was not considered. Such architecture consists of: (1) SIP Multicast extender, (2) A special module named MGA (Multicast Gateway agent) and (3) A multicast manager.

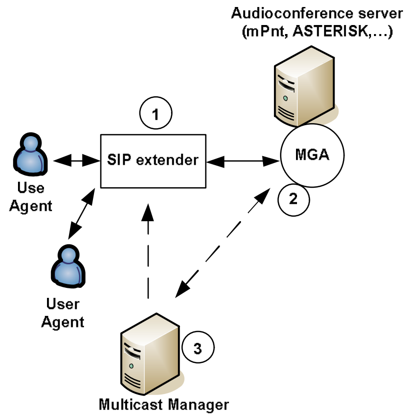

Fig 3. Proposed architecture

SIP MULTICAST EXTENSOR

This component permits to work with SIP extended signaling. It adds an additional header that indicates the use of Overlay Multicast. It also converts extended messages with SIP multicast into SIP conventional ones.

MULTICAST GATEWAY AGENT(MGA)

It interprets the extended SIP multicast messages and responds to the user agent. It fills out and reads information from a Applicative Multicast Table that contains information about the user agents associated with their respective SIP extenders. The URI and an IP address identify an user agent; the ID of the audioconference is the new URI that is found on the new field added and contains the status registered, included or excluded.

MULTICAST MANAGER

Consists of a program that works only on the control plane. A human operator can choose whether a specific user agent will solicite its inclusion or exclusion of an audioconference if a static configuration is done. For that, the multicast manager activates the SIP extender through a TCP mechanism. In the automatic modality, the multicast manager can obtain information from all the audioconference servers that play one roll in the session by using a TCP socket with each MGA module. If there are various servers in the WAN, it can calculate the most optimal topology. Every period, the multicast manager asks the MGA modules inside the audioconference servers about the status of their group members (**Fig. 4**.) by using UPDATE methods of SIP and responses.

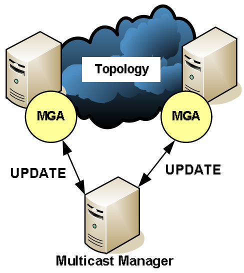

Fig. 4 Interaction between the multicast manager and the MGAs

The exchange messages between the multicast manager and the SIP extenders, and between the first, and the MGAs permits to send the information about the most optimal topology calculated with a getetic algorithm. We propose the use of UPDATE method, with a new header.

4. EXPERIMENTATION AND RESULTS

We present two testbeds. One of them contains an IP multicast architecture with VideoLAN and the other has a subset of the proposed architecture for its evaluation using SJphone. Both testbeds were implemented in the laboratory in order to perform a comparison and to have a preliminary evaluation of our proposal. We did two experiments for each testbed: in the first one we configured IP multicast and measured joining and leaving duration in a scenario, in the second one we implemented a subset of our architecture and we also measured joining and leaving duration in one scenario.
The first testbed (**Fig. 5**.) contains 3 routers model CISCO 3640, 2 LAN switches model ENTERASYS 3820 and 2 PCs. The routers were configured for unicast routing (RIPv2), multicast routing (PIM-SM) and IGMPv2. R2 was chosen as the Rendez-vous-point (RP). An IGMP message is sent by the VideoLAN client (joining or leaving message).

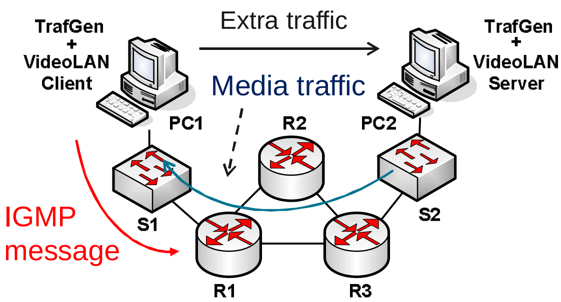

Fig. 5 Testbed 1 for IP multicast scenarios

The PC1 contains the open source application VideoLAN (client side) and traffic generator named TRAFGEN created at Telecom Sud-Paris. The PC2 contains the application VideoLAN (server side) and TRAFGEN.
An extra traffic from TRAFGEN in PC1 towards PC2 was injected. We established a continuous flow of IP packets with payload 1024 bytes. The load was increased progressively by decreasing the interarrival periods T as follows:
No extra traffic, T=28 ms , T=21 ms , T=14 ms , T=7 ms
The second testbed (**Fig. 6**.) contains 3 routers model CISCO 3640, 3 LAN switches model ENTERASYS 3820 and 3 PCs. The routers were configured for unicast routing (RIPv2) only. A SIP multicast user agent (SIP extender module is embedded) manifests its interest to participate or leave a conference. A conference server with its respective MGA module forms a logical topology that can be optimized by any algorithm. On this testbed a genetic algorithm was not configured initially, but a static topology was defined. As a first try, the MGA and the multicast manager were located on the same PC (PC3).

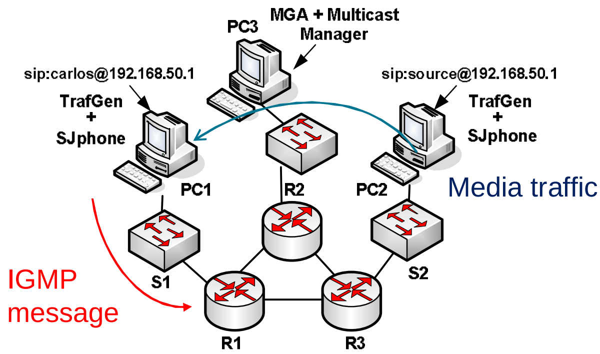

Fig. 6 Testbed 2 for SIP multicast scenarios

The extra traffic was similarly injected from TRAFGEN in PC1 towards TRAFGEN in PC2.

TESTBED 1 - EXPERIMENT 1: JOINING DURATION WITH IP MULTICAST

In this scenario, VideoLAN (server side) begins to transmit a multicast flow (file MPEG1) to the private group 239.255.12.42 and port 1234 continously. Such flow is originated from PC2 (the source does not belong to the group 239.255.12.42).VideoLAN (client side) receives the multicast flow by listening to the port 1234 for the traffic sent to 239.255.12.42.
The joining duration has been measured by substracting two instances: the first one (T_1) is when the first IGMP message (join) is sent by PC1 to the group 239.255.12.42, and the second (T_2) is when the first media packet (UDP/RTP) is received by the IP multicast host (PC1).

$$T_{IN} = T_2 - T_1 \qquad \text{Eq. (1)}$$

We have run 10 times this experiment for each value of T, and the results show that the values of joining duration are generally lower than 100 ms. There are two exceptions, in the case of T=14 ms where it is higher than 400 ms. The table1 shows the average and standard deviation.

Table 1 – Average and Standard Deviation for T_{IN}

T (ms)	Average (ms)	Standard Deviation (ms)
7	47	11
14	139	189
21	45	12
28	56	14
No extra	51	11

We can notice that for T=14 ms the average and standard deviation are the highest. However, we can not conclude that it is due to the extra traffic injected.

TESTBED 1 – EXPERIMENT 2: LEAVING DURATION WITH IP MULTICAST

In this scenario VideoLAN (server side) transmits a multicast flow (file MPEG1) to the private group 239.255.12.42 and port 1234 continously. VideoLAN (client side) receives the multicast flow by listening to the port 1234 for the traffic sent to 239.255.12.42. Then, it stops receiving such multicast flow by sending an IGMP leave group message.

The leaving duration has been measured by substracting two instants: the first one (T_1) is when the first IGMP message (leave) is sent by PC1 to the group 239.255.12.42, and the second (T_2) is when the last media packet (UDP/RTP) is received by the IP multicast host (PC1).

$$T_{OUT} = T_2 - T_1 \qquad \text{Eq. (2)}$$

We have run 10 times this experiment for each value of T, and the results show that the values of leaving duration are a bit higher or lower than 2000 ms. In only one case (T=28 ms of extra traffic) the leaving duration is much higher than 200 ms (2499 ms). It is due to the configuration of the routers with Last Query Interval Count = 2 , Last Query Interval =1 s and the uncertainty produced by the extra traffic.

Table 2 – Average and Standard Deviation for T_{OUT}

T (ms)	Average (ms)	Standard Deviation (ms)
7	1999	4
14	2000	2
21	1998	5
28	1949	369
No extra	2000	1

We can notice that the standard deviation for T=28 ms is the highest. We can not conclude that it is due to the effect produced by the extra traffic injected.

TESTBED 2 – EXPERIMENT 3: JOINING DURATION WITH SIP MULTICAST

In this scenario, the joining duration has been measured in a similar manner as in Testbed 1. It was calculated by substracting two instants: the first one (T_1) is when the first SIP INVITE extended message is sent by the PC1 to the MGA module in PC3, and the second (T_2) is when the first voice packet (UDP/RTP) is received by SJPhone in PC1.

$$T_{IN} = T_2 - T_1 \qquad \text{Eq. (3)}$$

MGA takes the information provided by the multicast manager, just to choose which is the best topology for the group of MGAs. In this case, the transmission of the topology information between the multicast manager and it depends only on the internal processing at PC3.

We ran 10 times this experiment for each value of T, and the results show that the values of joining duration in this experiment are usually situated between 200 and 500 ms. They vary with a higher range than the values of the results on the experiment 1. There are two peaks for T=21 ms and without extra traffic; we can not conclude that this is due to the presence of extra traffic.

Table 3 – Average and Standard Deviation for T_{IN}

T (ms)	Average (ms)	Standard Deviation (ms)
7	416	128
14	311	79
21	400	102
28	386	159
No extra	422	66

In Table 3 we can observe that the highest standard deviation occurs with T=28 ms. We can not conclude that this is a consequence of the injection of extra traffic.

TESTBED 2 - EXPERIMENT 4: LEAVING DURATION WITH SIP MULTICAST

In this scenario, the leaving duration has been measured in a similar manner as in Testbed 2. It was calculated by substracting two instants: the first one (T_1) is when the first SIP BYE extended message is sent by the PC1 to the MGA module in PC3, and the second (T_2) is when the last voice packet (UDP/RTP) is received by SJPhone in PC1.

$$T_{OUT} = T_2 - T_1 \qquad \text{Eq. (4)}$$

We have run 5 times this experiment for each value of T, and the results show that in some cases the leaving duration values are negatives (Table 4). It is explained by the fact that the last voice packet arrived earlier than when the first IGMP leave message was sent. The CODEC used was GSM 6.10 (lossy speech compression).

The results show that two peaks occur with T=7 ms and T= 21 ms of extra load injected. The values were 248 ms and 379 ms respectively.

We can not conclude that the peaks are due to the injection of extra traffic. However, we can conclude that the compression algorithm is correlated with the leaving duration in this case.

The highest standard deviation values occur with T=7 ms and T= 21 ms.

Table 4 – Average and Standard Deviation for T_{OUT}

T (ms)	Average (ms)	Standard Deviation (ms)
7	45	113
14	-6	6
21	65	176
28	-18	7
No extra	-19	7

VALIDATION

Although we can not conclude that the highest values of standard deviation are due to the presence of extra traffic, we can analyze in table 5 the difference between the maximum and minimum value of standard deviation for T_{IN} considering all the measurements.

Table 5 – Standard Deviation for T_{IN}

IP Multicast	SIP Multicast
MAX: 189	MAX: 159
MIN: 11	MIN: 66
Δ=178	Δ=93

We can see that the difference is lowest for our proposal than for IP multicast in our scenarios studied.
We can also analyze in table 6 the difference between the maximum and minimum value of standard deviation for T_{OUT} considering all the measurements.

Table 6 – Standard Deviation for T_{OUT}

IP Multicast	SIP Multicast
MAX: 388	MIN: 176
MIN: 1	MAX: 6
Δ=387	Δ=170

These tables show that our proposal is more stable for the scenarios studied. However, we propose to include a genetic algorithm inside the multicast manager, so that the existence of several clients, and the calculus of the best trees affect the least the response time.

5. INCLUSION OF A GENETIC ALGORITHM IN THE ARCHITECTURE

We propose to use a genetic algorithm as a metaheuristic method as this is the most common one applied to multicast architectures **[9]**. At first we need to represent a general scenario with one SIP client extended trying to connect to the best MGA, as it received the result of calculus inside the multicast manager.

MODELING OF ARCHITECTURE WITH GRAPH THEORY

One simple case study of a SIP extender, a MGA and a multicast manager can be represented with a graph. Each vertice sends and receives information from any of the others. The multicast manager sends UPDATE messages to the other two modules with the best association, and receives confirmation (**Fig. 7**.).

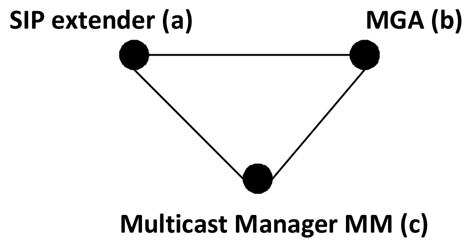

Fig. 7 Graph for the basic architecture

This graph can be defined in the following manner:

$$G = (V, E, \phi) \qquad \text{Eq. (5)}$$

Where:
G = graph.
V = a set of components of the architecture (vertices)
E = an association between any pair of components and is bidirectional
ϕ = the function that assigns to each association of elements of the architecture, the components that interact.

We can consider a case study where two SIP extended clients interact with any of two MGAs according to the information distributed from the multicast manager to both (**Fig. 8**)

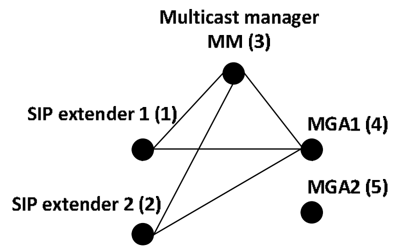

Fig. 8 Testbed 1 for IP multicast scenarios

The representation of this possible topology can be expressed with a matrix as follows in (**Fig. 9**.).

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 9 Representation with a graph in matrix form

If we take only the digits that are situated over the diagonal filled with zeros, the graph can be represented as follows in (**Fig. 10**)
:

0110|110|10|0

Fig. 10 Bits representing a graph

We have represented one possible solution (topology) for the architecture. If we have many SIP extenders, and several MGAs, there can be a lot of solutions. We can define a chromosome in this way. We can also add zeros so that ASCII digits are used to represent the solution. In this example, with characters B2 and B1 (**Fig. 11**).

$\boxed{00}$0110|110|10|0

B2 B1

Fig. 11 Representation of graph with ASCII codes

GENETIC ALGORITHM

Now that we have represented a chromosome, which corresponds to a possible solution, we need to define the fitness of our genetic algorithm. Considering the simplest criteria to measure the behavior of best chromosome, we can define the fitness as follows in Eq. (6).

$$f = \frac{1}{\mathrm{Pr}} \qquad \text{Eq. (6)}$$

Where:

$\mathrm{P_r}$ = Average of the delays between the SIP extender clients and their respectives MGA assigned.

We also propose the use of mutation and crossover throughout the generations.

6. CONCLUSIONS.

1) The SIP protocol does not have native characteristics for multicast. However, it can be layed on a network infrastructure based on IP multicast with limitations, and it is possible to add support for Overlay Multicast with extensions.
2) The proposed architecture permits the use of basic methods of SIP protocol to support multicast signaling, so permitting the improvement of response time in some scenarios.
3) The standard deviation of the response time is directly proportional to the probability of retransmissions in the studied scenarios. The difference between the minimal and maximal standard deviation is lower in our proposal.
4) The multicast manager, as part of the architecture proposed, permits to optimize the topology formed by SIP extenders and MGAs in a static configuration. A genetic algorithm might be implemented inside the multicast manager just to reduce the effect of the calculus of the best topology in comparison with the effect of protocol SIP timers.
5) Future work aims to study the effect of middle-boxes (NAT and firewalls), our architecture and to simulate scenarios at great scale.

7. LIST OF ACRONYMS

CGMP CISCO Group Management Protocol

DM Dense Mode

IGMP Internet Group Management Protocol

MGA Multicast Gateway Agent

NGN Next Generation Network

PIM Protocol Independent Multicast

SDP Session Description Protocol

SER SIP Express Router

SIP Session Initiation Protocol

SM Sparse Mode

URI Uniform Resource Identifier

8. ACKNOWLEDGEMENTS

The authors thank to the Consejo de Desarrollo Científico y Humanístico (CDCH) of the Universidad Central de Venezuela for the budget assigned to the first part or the project: “Mejoramiento de arquitectura para servicios multimedia basaa en multidifusión SIP” under the identification PI-08-7820-2009/1.

9. REFERENCES

[1] O. Hersent, D. Gurle, and J-P. Petit, **"La Voix sur IP. Codecs, H.323, SIP, MGCP, deployment et dimensionnement"**, Ed. Paris: DUNOD, 2004, pp. 60-81.

[2] Beau Williamson, **"Developing IP Multicast Networks. Volume I"**, Ed. Indianapolis: CISCOPRESS, 2001, pp. 1-43.

[3] G. Camarillo, ***"SIP demystified"***. Ed. USA: McGraw Hill, 2002, pp. 37-45.

[4] P. Rogelio, **"Internet Multimedia Communications Using SIP"**. Ed. USA: Morgannn Kauffma, 2008, pp. 48-56.

[5] R, Bhattacharya ; P,Venkateswaran ; S, Sanyal, **“Genetic Algorithm Based Efficient Routing Scheme For Multicast Networks”** Proceedings of IEEE International Conference on Personal Wireless Communications ICPWC 2005, Colmar, Francia, 2005, pp. 500-504.

[6] J. R. H.Schulzrinne., G.Camarillo ., A., Johnston ., J. Peterson., R.Sparks., M. Handley., E.Schooler. (Consulta en Febrero 2012). **“SIP: Session Initiation Protocol”.** RFC 3261, 2002. Disponible en http://www.ietf.org/rfc/rfc3261.txt

[7] H. Schulzrinne ., J. Rosenberg, **“Signaling for Internet Telephony,”** *Proceedings. of IEEE Sixth International Conference on Network Protocols*, 2008, pp. 298-307.

[8] J. Rosenberg, (Consulta en Febrero 2012). **“A Framework for Conferencing with the Session Initiation Protocol (SIP)”.** RFC 4353, 2006. Disponible en http://www.ietf.org/rfc/rfc4353.txt

[9] N, Lin ; X, Chen ; X. Li, **“QoS Multicast Routing Algorithm Based on Immune Genetic Method in NGN”.** Proceedings of International Conference on Multimedia Technology 2010, Hangzhou, China, 2011, pp. 1-4.