A Novel Overlay Network for a Secure Global Home Agent Dynamic Discovery

Ángel Cuevas[1], Rubén Cuevas[1], Manuel Urueña[1], and Carmen Guerrero[1].

Departamento Ingeniería Telemática. Universidad Carlos III de Madrid, 28911 Leganés (Spain) *
{rcuevas,acrumin,muruenya,guerrero}@it.uc3m.es

Abstract. Mobile IP and Network Mobility are the IETF proposals to obtain mobility. However, both of them have routing limitations, due to the presence of an entity (Home Agent) in the communication path. Those problems have been tried to be solved in different ways. A family of solutions tries to improve the routing by locating closer Home Agents making shorter the communication path. These techniques require a method to discover a close Home Agent from the Mobile Device. We proposed peer-to-peer based solution, *Peer-to-Peer Home Agent Network*, in order to discover a close Home Agent. This paper defines the necessary mechanisms to make this solution secure based on a mechanism named Secure Join Procedure.

1 Introduction

Mobile IP (MIP) is the mechanism proposed by the Internet Engineering Task Force (IETF) to enable host mobility, in IPv4 (MIPv4) [1] and IPv6 (MIPv6) [2]. However, mobility is also required in networks (planes, trains, etc). Hence, support for network mobility is required. The Network Mobility (NEMO) Basic Support Protocol [3] [4] is the IETF proposal to provide network mobility support. The basic solution of Mobile IP and NEMO presents the so-called triangular-routing as the main performance limitation: mobile nodes' communications must pass through an entity, called the Home Agent (HA). It is possible that some communications suffer from higher delays than those required by some kind of applications (e.g. real time applications like voice or video) in order to obtain an acceptable performance. Several solutions have been proposed in order to solve these routing problems. One family of solutions proposes (so as to improve the routing) to reduce the distance between the HA and the mobile devices as much as possible, minimizing the total path length. This paper is based on [9] which proposes the use of an overlay peer to peer network (Peer-to-Peer HA Network), formed by HAs, in order to discover a close HA to a certain mobile device. It

* This work was supported by the European Commission through NoE CONTENT FP6-CONTENT-038423, the Spanish government through the Project IMPROVISA TSI2005-07384-C03-027 and the Madrid regional government through the Project BIOGRIDNET CAM-S-0505/TIC-0101

is simple, fully global, dynamic and it can be deployed in IPv4 and IPv6. But [9] does not consider the security aspects, thus, this paper describes the main security mechanisms needed to make the Peer-to-Peer HA Network a secure solution.
Peer-to-Peer Home Agent Network (P2PHAN) is an architecture focused on a structured DHT (Distributed Hash Table) based Peer-to-Peer network. This kind of Peer-to-Peer (p2p) networks have been extensively investigated and several approaches have been defined (e.g. Chord [10] or Kademlia [11])[1]. An important effort has been done in security aspects for p2p networks and the main problems have been identified, specially for file sharing p2p networks [13] [14]. Moreover, security becomes a primary issue when p2p is applied to scenarios as the one considered in this paper, the Home Agent discovery.
This paper focuses on the security of a specific application, the P2PHAN. It has some specific features different from the file sharing scenario but also common problems. Security issues can be solved because of some specific features of this architecture, as verifiable data based on Border Gateway Protocol (BGP) information [15] and a reduced number of peers in comparison with a file sharing p2p networks.

In addition, a mechanism that secures the communications between MRs and HAs (i.e to guarantee the trust between HAs and MRs) must be used. It is IKEv2 [16] and its application to mobile environment can be done as it is proposed in [17].

All this guarantees a practical high security level for the P2PHAN approach. The paper proposes a main mechanism, *The Secure Join Procedure*, and some others associated to this one (as redundancy or parallel queries) to guarantee the security on the P2PHAN. The Secure Join Procedure is based on the use of a central bootstrapping server. The presence of bootstrapping nodes is used in commercial p2p networks since it is an efficient method for the peers to join the network and find other peers (e.g. Emule [12]). The Secure Join Procedure contains a secure Peer-ID assignment based on random assignment. This solves the main cause of possible attacks in the structured DHT p2p Networks which is that peers can choose its own Peer-Id. In addition, the paper evaluates the complexity of the possible attacks concluding that the proposed mechanisms introduce a practical level of security.
The structure of this document is as follows. In section 2, the Peer-to-Peer Home Agent Network will be more accurately defined. Section 3 exposes the security problems of the Peer-to-Peer Home Agent Network and the mechanisms which solve them and Section 4 shows the conclusions extracted from this work and introduces the further work to be developed.

[1] Detailed information about peer to peer networks can be found in [21] and [22] which are surveys about this technology.

2 Peer-to-Peer Home Agent Network Architecture

Peer-to-Peer Home Agent Network is a structured DHT p2p network with ring structure formed by HAs. It is similar to Chord [10]. In our scheme the search key will be: *hash(AS number)*.
When one node joins the P2PHAN, it chooses an identifier (*Peer-ID*) from the ids pool. Its position in the ring is determined by the chosen id (it is placed between the two nodes with the closest higher and the closest lower id *Peer-ID* than its own id). Each peer has direct references to its two neighbors and also with other peers (crossing the ring) so as to make faster the routing within P2PHAN. These references are called *fingers*. Each peer uses the fingers to create its P2PHAN routing table.
On the other hand, each peer must store its Autonomous System (AS)[2] number within P2PHAN. The peer obtains a key by computing the *hash(AS number)*. Then, it looks for the peer with the most similar Peer-ID to that key and sends to this peer the key and its IP address. The destination peer stores the pair *<key, IP address>*.
Eventually, an MR connected to a HA_1 detects that the distance to this HA is higher than the desired (e.g. it measures RTT with HA_1 higher than a threshold). Then, it launches the procedure to discover a closer HA. The MR sends its current CoA to HA_1. At this point, HA_1 discovers (using BGP) the CoA's AS number. Afterwards, it computes the *hash(AS number)* which is the *search key*. With this *search key* the HA launches a search within the P2PHAN and obtains the list of the HAs placed at the same AS as the MR. The list is sent to the MR which decides its preferred HA.

Fig. 1 shows the P2PHAN functionality explained above. A more detailed explanation can be found in [9]. It must be noticed that the solution has been explained for NEMO but it also works on MIP.

3 Securing the Peer-to-Peer Home Agent Network

The previous section introduces the standard P2PHAN functionality to solve the HA discovery problem. However, this architecture suffers from some security problems due to the use of a p2p scheme. Hence, this section presents the necessary mechanisms to give to the P2PHAN a high security level. Firstly, the security threats existing in the scenario will be analyzed. Then, the main security mechanism proposed (the *Secure Join Procedure*) will be explained. Finally, the solution to each of the posed threats will be detailed.

3.1 Peer-to-Peer Home Agent Network Security Threats Analysis

Since P2PHAN is based on a structured DHT p2p network, it has the problems of this kind of p2p networks. These problems are fully detailed in [13] and [14].

[2] In the Internet, an autonomous system (AS) is a collection of IP networks and routers under the control of one entity (or sometimes more) that presents a common routing policy to the Internet [5]

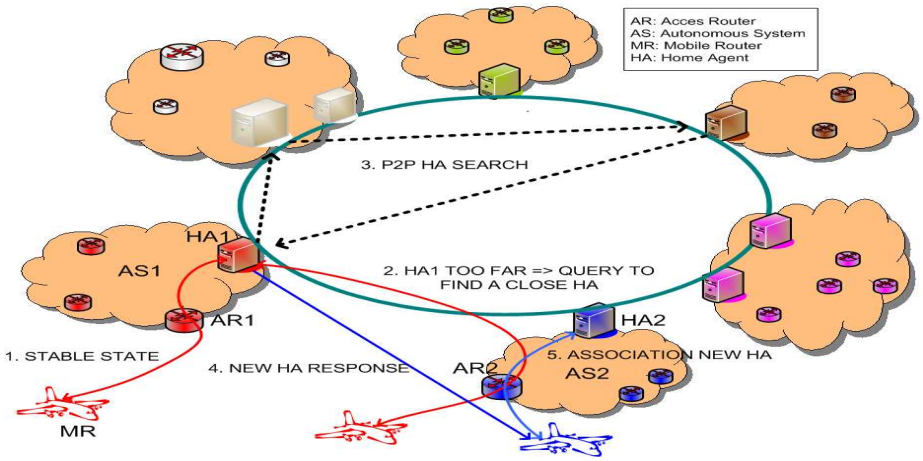

Fig. 1. P2P Home Agent Network Scheme

Due to the specific features of our scenario (i.e. verifiable BGP information, a reduced number of users compared with a file-sharing application as Emule [12] or Kademlia [11]), it allows to define a number of security and trust mechanisms which offer a high security level. It must be also considered that when we refer to a malicious node it can be both a node of a malicious user or a non-malicious node with a bad behaviour (e.g. due to an hardware or software failure). HAN approach.

Following, the security problems affecting the P2PHAN are described:

1. *Starvation Attack:* A malicious node does not answer (or return false information) to the queries which are soliciting information that this node is storing. Then, the nodes which have registered their information in this malicious node cannot be contacted. In the P2PHAN if a malicious peer selects its Peer-ID as close as possible to a given key = hash(AS), it becomes the closest HA to that key and thus, the responsible of storing the information of the given AS. In this situation, this malicious peer can starve the given AS by not answering (or by giving false answers) to the queries soliciting the information of this AS. We call this attack *targeted starvation attack*. There is a less sophisticated version of this attack. It occurs if the malicious peer does not have any specific target AS, thus, it obtains any Peer-ID and if this Peer-ID has associated any AS key (i.e. hash(AS number)) it could starves that AS. Finally, the attack can be performed by one single attacker or by several attackers. When several attackers works together, the attack is called *Collaborative Attack* and usually it is more harmful because the resources to perform the attack increase with the number of attackers collaborating.
2. *Routing Attack:* A malicious node does not route the messages or select bad routes for the queries. If we focus on the P2PHAN, this attack can have different objectives. The first one is affect the performance of the P2PHAN

without any other more specific objective, this can be interpreted as a non-targeted routing attack. On the other hand, the objective of the routing attack could be starve a victim node. That is, the malicious peer selects the Peer-IDs so as to obtain all the fingers of the victim node. In this situation, all the queries sent by the victim node must be routed by the malicious peer(s) which does not route (or selects bad routes for) the queries of the victim node. We call to this attack the *targeted routing attack* and it is a good example where the collaborative attack is more effective.

3. *High Rate of Joins and Leaves:* A malicious node joins and leaves the P2PHAN continuously in order to make the topology unstable and generate a huge amount of signalling traffic. It could be also a collaborative attack.
4. *Register False Information:* A malicious node registers a false AS different from the AS where it is located.
5. *Multiple Registers:* A malicious node joins the network several times with the same IP Address in order to obtain as many Peer-ID as possible.

3.2 Secure Join Procedure

In order to define the *Secure Join Procedure* (SJP) in the proposed scenario the re-use of a Bootstrapping Server as security point is proposed. The main security function of this Bootstrapping Server is to assign a random identifier for the new HA which wants to become a member within the P2PHAN. However, this bootstrapping server can not guarantee the Secure Join Procedure itself. Therefore, the next method will be applied in order to get a secure access to the P2P network.

First of all, if an organisation managing an AS wants to introduce HAs in the P2PHAN, it has to create a pair public key-private key (AS_pu_key - AS_pr_key). Therefore, if a HA wants to register itself within the P2PHAN, it must own the AS_pr_key to be able to register its information within the P2PHAN. The list of HAs of an AS is stored for a node in the P2PHAN (which is another HA). This node is called *Responsible HA*. When a HA tries to register its information, its *Responsible HA* will use the AS_pu_key as it will be described later to check that the new HA trying to join the P2PHAN knows the AS_pr_key. This implies that the new HA is an authorised node of that AS. The *Responsible HA* can obtain the AS_pu_key from a repository or it could be included in the registration message of the first HA of an AS which is registered within the P2PHAN. Following the SJP is described.

A new HA which wants to join the P2PHAN sends a *Join Request* to the Bootstrapping Server (See step 1 in fig. 2) with its IP address, the AS number and a cheksum of all this information ciphered with the private key (AS_pr_key) of the AS where it is located, that is, its signature.

After that, the Bootstrapping Server generates a random peer-ID for the new HA and launches a search in the p2p network to find the HA which has the most similar ID to the peer-ID generated, which is the Responsible HA (See step 2 in fig. 2). Then, the bootstrapping server forwards the message received from the

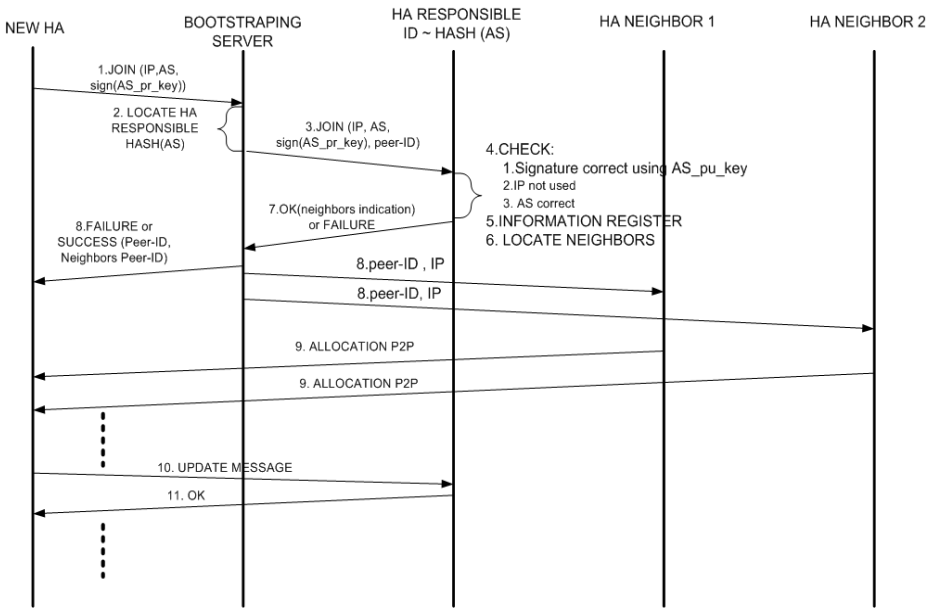

Fig. 2. Secure Join Procedure Message Exchange

new HA adding the peer-ID generated to the Responsible HA (See step 3 in fig. 2).

The function of this *Responsible HA* is to make several tests in order to check if the new HA is a malicious node. If one of this test is not successful the *Responsible HA* returns a *Check Failure* to the *Bootstrapping Server*. Otherwise, the new HA information is stored by the *Responsible HA*. The following three tests are executed (See step 4 in fig. 2):

1. The *Responsible HA* uses the AS_pu_key which has stored in order to check if the checksum obtained is the same than the checksum ciphered with the AS_pr_key for the new HA. If it is not, it returns a *Check Failure*, otherwise it runs the second test.
2. The *Responsible HA* checks whether it has information stored for the IP which appears in the *Check Request* or not. If it has information stored for that IP, it returns a *Check Failure*, otherwise it runs the third test.
3. The *Responsible HA* checks if the IP within the *Check Request* belongs to the AS present in the request. In order to get this information the *Responsible HA* obtains the AS path for the IP address using BGP. Then, it checks if the AS number given in the *Check Request* matches with the last AS number returned in the AS path.

If some of these tests are not successful the *Responsible HA* returns a *Check Failure*. Otherwise, since all test were successful it registers the new HA information (See step 5 in fig. 2) and sends a query to the P2PHAN in order to find

the neighbors for the new HA, i.e. the the two nodes with the closest higher and the closest lower Peer-ID (See step 6 in fig. 2). After locating the neighbors, the *Responsible HA* sends to the Bootstrapping Server a *Check Success* adding the neighbors IP addresses and peer-IDs (See step 7 in fig. 2). Next, the Bootstrapping Server sends the neighbors peer-IDs and the random peer ID to the new HA. In parallel the Bootstrapping server sends the peer-ID and the IP of the new HA to the neighbors (See step 8 in fig. 2). When the neighbors receive the message from the Bootstrapping Server, they allocate the new HA using standard P2P techniques (See step 9 in fig. 2).

From this moment, the new HA will send periodically update messages in order to indicate that it is alive to the *Responsible HA*. If the Responsble HA does not receive these update messages during a pre-configured time out, it removes the entry for that HA (See steps 10 and 11 in fig.2).

It must be noticed that a P2PHAN member has to sign the update messages sent to the *Responsible HA* with its AS_pr_key.

3.3 Security Problems Resolution

The SJP is the basic mechanism used to solve the problems introduced in Section 3.1. In this section we analyze how the proposed solution solve those problems. For this purpose we divide the attacks into three categories: targeted attacks, non-targeted attacks and other attacks.

Targeted attacks The key point in our solution is that the Peer-ID assignment is performed by the network instead of each HA can choose its position in the P2PHAN. Hence, in order to perform any of the targeted attacks defined in 3.1 the attacker should solicit as many Peer-ID as necessary until obtain one valid Peer-ID for its purpose.

Based on the study developed in [24], table 1 shows a realistic example of the results that would be obtained in a targeted *Starvation Attack* with a probability of success equal to 0.7. This scenario has 10000 HAs where the time spent to obtain one Peer-ID is 1 second. It is shown the number of Peer-IDs and time needed for 1, 5 and 10 replicas available within the P2PHAN.

Replicas	*Peer-IDs Required*	*Time (hr)* ($)
1	23333	6.48
5	135420	37.62
10	275540	76.5

Table 1. Example Scenario for a Targeted Starvation Attack with $P_s = 0.7$

The analysis in [24] focuses on the targeted starvation attack, thus, it can be used as a method to analyze the complexity of this kind of attack introduced

in 3.1. Therefore, it seems that the attack is quite feasible for one attacker and it becomes easier in the case of a collaborative attack. Thus, an attacker must obtain 23333 Peer-IDs in order to perform and attack with P_s equal to 0,7 and only one replica. That is, the attacker should receive more than the double of the Peer-IDs in the P2PHAN, thus, the Bootstrapping server can detect easily the attack by evaluating the rate of solicited Peer-IDs. This rate would be in a normal situation 10000 Peer-IDs in the time of live of the P2PHAN, whereas in an attack scenario this rate would be hundred or even thousand of Peer-IDs per day. Therefore, it would be really easy to detect attack attempts.

In the targeted routing attacks, again a high number Peer-IDs is needed by the attacker. In this case the attacker needs to become all the fingers of the target HA. This implies the solicitation of many peer-IDs ,based on the analysis made in [24]. Therefore, it would be easy to detect attack attempts due to the high rate of Peer-IDs solicitation received in the Bootstrapping Server.

In a nutshell, a targeted attack against the P2PHAN could be viable in terms of time, but it is easily detectable by applying an access control policy based on the high rate of joins attempts.

Non-Targeted attacks In this subsection the non-targeted starvation and routing attacks are analyzed. The non-targeted starvation attack is solved with the use of replicas. That is, in an scenario where r replicas are being used in order to store the information of each AS, the malicious node would be responsible of 1 of r replicas. Then, the victim AS would never be starved. In this situation, if the peers only send one query in order to obtain the information about the desired AS, the malicious peer would affect to the $\frac{1}{N} * 100\%$ of the queries for the victim AS, because 1 of each r queries would arrive to the malicious node. In order to avoid this loss of performance the peers sends in paralell at less three queries for different paths (i.e. using different fingers). Statistically, each query arrives to different peers storing different replicas. By doing so, the correct result would be the most common among the responses. Obviously, this mechanism is more efficient with higher r and number of paralell replicas.

On the other hand, the non-targeted routing attack is characterized by a node which does not route the queries or it uses bad routes for them. In this case, the number of replicas is not a critical point. However, the solution is also the utilization of parallel queries in order to reach the destination. Again, the correct result is the most common among the responses and the mechanism is also more efficient with a high number of parallel queries.

Furthermore, it must be highlighted that all these proposed security methods are also useful if non-malicious nodes have non-standard behaviour which may affect the performance of the P2PHAN.

Other attacks The rest of the security problems described in Section 3.1 are solved by the SJP as is following described:

– *High Rate of Joins and Leaves:* The Bootstrapping Server will have a list with the IP addresses of the recent joins. Based on this list it is easy to check

if a node is continuously joining the network. In this case, this node will be introduced in a *Black List* and its join requests will not be accepted during an established time. The fact of preventing continuous joins inherently avoids continuous leaves. This mechanism prevents the inconsistent behaviour of non-malicious nodes with an unstable network connection.
- *Register False Information:* It is checked by the *Responsible(s) Node(s)* during the SJP in the second and third performed tests.
- *Multiple Registers:* The SJP prevents that one node joins the network twice with the same IP address twice with the same IP address by using the second test performed by the *Responsible(s) Node(s)*.

4 Conclusion and Further work

This paper is focused on adding security to the Peer-to-Peer Home Agent Network [9]. This architecture is used to discover HAs geographically distributed in a simple, dynamic, fully global and distributed way. Besides, it works over IPv4 and IPv6. The main security mechanism proposed is the *Secure Join Procedure*. This method reuse the Bootstrapping Server (present in commercial p2p networks) so as to assure a secure Peer-ID assignment procedure. The conclusion after the analysis of the SJP is that an attacker must have the following characteristics to perform an attack to the P2PHAN:

- The attacker must know a private key which is controlled for the organisation which manages the AS.
- BGP capabilities, it is hard because it is not necessary only supports BGP but have any relationship with other AS in order to obtain BGP information.
- HA capabilities, it is feasible.
- Thousand of IP addresses, if the P2PHAN is formed by thousand of HAs.

In addition, if the attacker fulfils all the previous requirements, the attack can be easily detected in case of targeted attacks due to the increment in the Peer-ID solicitation rate, and easily avoidable in the case of non-targeted attacks due to the utilization of replication and multiples parallel queries.

Therefore, the security solution presented in the paper has a practical security level which makes any attempt of attack against the P2PHAN unaffordable.

The future work will be the implementation of the P2PHAN with all the security framework proposed in this paper for both scenarios: simulation environment and real testbed.

References

1. C. Perkins. "Mobility Support for IPv4". *RFC 3344*, August 2002.
2. D. Johnson, C. Perkins, J. Arkko. "Mobility Support in IPv6",*RFC 3775*. June 2004.
3. V. Devarapalli, R. Wakikawa, A. Petrescu, P. Thubert. "Network Mobility (NEMO) Basic Support Protocol" *RFC 3963*, January 2005.

4. K. Leung, G. Dommety, V. Narayanan, A. Petrescu "IPv4 Network Mobility (NEMO) Basic Support Protocol". *IETF Draft*, June 2006.
5. J. Hawkinson, T. Bates. "Guidelines for creation, selection, and registration of an Autonomous System (AS)", *RFC 1930*. March 1996.
6. K.Chowdhury, M.Khalil, H.Akhtar. "Home Subnet Prefix or the Home Agent discovery for Mobile IPv6',' *IETF Draft*. April 2004.
7. H.J.Jang, A.Yegin, J.Choi. "DHCP Option for Home Agent Discovery in MIPv6," *IETF Draft*. February 2006.
8. Y.S.Yen., C.C.Hsu, H.C.Chao."Global dynamic home agent discovery on mobile IPv6," in *Wireless Communications and Mobile Computing*. Volume 6, Issue 5, pp. 617–628.
9. R. Cuevas, C. Guerrero, A. Cuevas, M. Calderón, C.J. Bernardos. "P2P Based Architecture for Global Home Agent Dynamic Discovery in IP Mobility". *Proc. IEEE 65th Vehicular Technology Conference*, April 2007.
10. I.Stoica, R.Morris, D.Karger, M.F.Kaashoek, H.Balakrishnan. "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. ACM SIGCOMM'01*, 2001.
11. P.Maymounkov and D.Maziers, "Kademlia: A peer-to-peer information system based on the xor metric," in *LNCS*, 2002, pp. 53–65.
12. "Emule." [Online]. Available: `http://www.emule-project.net`
13. D. S. Wallach, "A survey of peer-to-peer security issues," in *Lecture Notes in Computer Science*, 2003, pp. 42–57.
14. E.Sit and R.Morris, "Security considerations for peer-to-peer distributed hash tables," in *IPTPS '02*.
15. Y. Rekhter, T. Li, S. Hares. "A Border Gateway Protocol 4 (BGP-4)". *RFC 4271*, January 2006.
16. C. Kaufman, "Internet Key Exchange (IKEv2) Protocol". *RFC 4306*, December 2005.
17. V. Devarapalli, F. Dupont. "Mobile IPv6 Operation with IKEv2 and the revised IPsec Architecture". *IETF Draft*. December 2006.
18. E. Perera, V. Sivaraman, A. Seneviratne. "Survey on network mobility support". *Mobile Computing and Communications Review*, Volume 8, Number 2, April 2004.
19. P. Thubert, R. Wakikawa, V. Devarapalli." Global HA to HA protocol," *IETF Draft.*October 2005.
20. M. Bagnulo, A. Garcï¿½-Martï¿½ez, C.J. Bernardos, A. Azcorra."Scalable Support for Globally Moving Networks".*ISWCS'06*. September 2006.
21. J. Risson and T. Moors. "Survey of Research towards Robust Peer-to-Peer Networks: Search Methods", Internet Draft, draft-irtf-p2prg-survey-search-00.txt
22. E.K. Lua, J. Crowcroft, M. Pias, R.Sharma, S. Lim, "A Survey and Comparison of Peer-to-Peer Overlay Networks Schemes" in *IEEE Communications Surveys*, 2nd Quarter, Volume 7, Number 2.
23. M. Castro, P. Drushel, A. Ganesh, A. Rowstron and D. Wallach, "Secure routing for structured peer-to-peer overlay networks", In Proc. of OSDI '02, Boston, MA, 2002.".
24. R. Cuevas, A. Cuevas, M. Urueña, A. Banchs and C. Guerrero, " Analysis of the Full Starvation Attack in structured p2p systems". Available: `http://www.it.uc3m.es/rcuevas/tech_report/p2p_full_starvation.pdf`