DEEP - A Generic Name Resolution Protocol for Heterogeneous Networks

Petteri Pöyhönen
Nokia
petteri.poyhonen@nokia.com

Rui Campos
INESC Porto
rcampos@inescporto.pt

Pekka Pääkkönen
VTT Technical Research Centre of Finland
pekka.paakkonen@vtt.fi

Nadeem Akhtar
University of Surrey
n.akhtar@surrey.ac.uk

Cornelia Kappler
Siemens Communications
cornelia.kappler@siemens.com

Di Zhou
PSE/Siemens AG Austria
di.zhou@siemens.com

Abstract

The current trend towards convergence of telecommunication and data networks considering all emerging access technologies is leading to more heterogeneous and dynamic environments. Networking visions are evolving towards ubiquitous computing and it is foreseeable that once users start to establish their own networks formed by their personal devices, the number of both mobile networks and mobile nodes will increase. As the peer-to-peer communication paradigm is becoming more popular, the number of resolvable service endpoints is increasing.

The paper focuses on introduction of a multi-stage name resolution mechanism for future heterogeneous and dynamic environments called Destination Endpoint Exploration Protocol. It presents the design principles and the specification of the protocol and depicts how the protocol relates to existing name resolution systems by means of concrete examples.

Keywords: Name Resolution, Heterogeneous Networks, DNS, DEEP.

1. Introduction

Future networking encompasses more heterogeneous and dynamic aspects when compared to the current networking environments. In addition, the number of mobile nodes is increasing with each passing day. As the peer-to-peer communication paradigm becomes more popular, each mobile node is also potentially hosting one or more service endpoints whose names have to be resolved by a name resolution service.

One challenge for the future networking environments is to maintain node and service reachability. Typically, this requires a name resolution system to map service names into routable addresses or other references. Also, this introduces new challenges for name resolution, since the infrastructure support is not always available and name bindings are becoming more dynamic.

The networking research community is currently studying future networking technologies and possibilities to enhance the existing Internet architecture. One interesting finding is the concept of interconnecting separate networks and domains at higher logical levels to form a single global network [14]. These (potentially moving) networks and domains may use different technologies including name resolution mechanisms which results in more diversity in the name resolution services and their realizations. For instance, a moving network maintains its own contact address information in an external name resolution system, so that it becomes reachable by external nodes but does not reveal its local configuration and interconnections to the outside. Instead, it utilizes its own name resolution system, which is not coordinated with the external systems and may be technologically different from the latter. Therefore, it is not reasonable to assume that a single name resolution mechanism, such as the Domain Name System (DNS) is always available or applicable in the future heterogeneous and dynamic environment.

This paper introduces the Destination Endpoint Exploration Protocol (DEEP), a generic name resolution protocol for heterogeneous environments. DEEP resolves the communication endpoints such as control signaling endpoints specified by symbolic, descriptive names, relying on existing name resolution systems, such as DNS. This process may involve the use of multiple independent and possibly incompatible name resolution systems. Additionally, DEEP provides name resolution services for other protocols and applications via a unified interface, which hides the implementation details of name resolution systems. This eases the deployment of new control signaling application and protocols in heterogeneous environments. DEEP does not replace existing name resolution systems; rather, DEEP relies on such systems to accomplish its task.

The rest of this paper is organized as follows. In Section 2, we discuss the related work. A set of technical use cases that motivate the need for DEEP is described in Section 3 to motivate the need for DEEP. The design of DEEP is presented in Section 4, followed by the specification of the protocol in Section 5. In Section 6, the operational aspects of DEEP are further discussed using an example from the real world. Finally, we draw the conclusions and mention the future directions.

2. Related Work

Traditionally DNS has provided global name resolution services to network nodes participating in IP-based communication. DNS resolves names into locators, (typically IP addresses). Additionally, DNS also has other functions such as name based service discovery, certification and public key distribution, etc. For the static Internet, DNS has proved to be a feasible and working solution. However, with the tremendous increase in the number of mobile devices and a growing trend towards small, self-contained and often private, mobile wireless networks, the ability to operate with limited and even no infrastructure support is increasingly important. Also, the highly dynamic nature of mobile networks affects the applicability of DNS due to frequent updates, which are not well supported by DNS due to the use of caching. A number of approaches have been discussed in literature to provide name and service resolution in such scenarios. More recent work on Link Local Multicast Name Resolution (LLMNR) [2] and multicast DNS (mDNS) [3] proposals aim to enable link local name resolution scenarios in which DNS infrastructure is not present or available. In [4], a multicast-based secure DNS system architecture is proposed to provide mobile nodes in IPv6 mobile ad hoc network with secure name-to-address resolution and service discovery.

The schemes mentioned above are mainly based on the current Internet naming scheme, namely the DNS, which assumes that applications typically are aware of their intended form and scope of the communication and based on this, a locator that determines a network attachment point (location) in the network topology is searched. In [5], the authors argue that inherent rigidity of the naming systems used currently have proven to be a big hindrance towards efforts to efficiently enable new services such as mobility, group communication, resource discovery, service location, caching, etc., and they propose an intentional network naming architecture, where applications describe what they are looking for (i.e., their intent), not where to find it. [6] proposes a resource location and discovery system based on a Peer-to-Peer protocol in a dynamic networking environment as an alternative to current solutions that rely on static, hierarchical name resolution such as DNS that is not generally well-suited for fast updates and changes in topology.

Based on the research reported above, it is safe to say that future networking environments will not only have diverse naming and addressing architectures but also different schemes to provide name resolution services for different network domains. However, since all these domains eventually have to interwork for the Internet to work, it is absolutely essential to bridge the gap between these diverse architectures and protocols. [1] proposes a new inter-networking architecture which subsumes existing architectures but at the same time exposes the heterogeneity between the networks. It introduces the concepts of interstitial functions as a mechanism to bridge the differences between the interworking heterogeneous networks. On the other hand, the Ambient Networks concept [7] tries to hide the heterogeneity of networking technologies by using a technology-independent control space. In either case, the presence of potentially diverse address and name space requires a mechanism that allows resolution of names across different networking domains that may have different name resolution systems. DEEP is designed to provide this service as it will be clear in the following sections.

3. Technical Use Cases

The basic motivations and special requirements for DEEP design and development are presented in this section by using of 2 different use cases.

In general, any form of signaling between two network entities requires that contact addresses (locators, protocols and port numbers) of the two communication endpoints are known beforehand so that the signaling messages can be routed between the source and destination. However, in dynamic networking environments, contact addresses may change for a variety of reasons, such as mobility of network nodes or the network itself, internal reorganization of entities within a network for administrative reasons etc. Therefore, for the sake of flexibility, transparency and easy of operation, it makes sense to allow signaling applications use symbolic names for addressing destinations when sending messages; this ensures that any change in locator is transparent to the signaling endpoints. Since these symbolic names are typically independent on related locators, they also provide a unified way to address communication endpoints. It is obvious that the symbolic names have to be resolved to locators before messages can be routed. The locator typically happens to be an IP address and we use them interchangeable in the discussion that follows.

It is one of the important assumptions for DEEP design that, in most cases, signaling applications use symbolic names instead of locators to address signaling destinations, and that there is no single name resolution system containing all information of both communication endpoints. The following simplified use cases illustrate this motivation.

A signaling application starts a new session and tries to send a signaling message to a remote counterpart for the first time. The source DEEP has to ask available name resolution systems for help. In this case a local DNS server is contacted via a local DNS client, which maps the symbolic name of a signaling destination to the corresponding IP address. After getting reply from the source DNS, DEEP provides the resolution results back to the external entity that has invoked it. Except for the message exchange with the local DNS and since the whole symbolic name has been successfully resolved, the source DEEP does not need to contact any remote

DEEP instances. This use case is represented in Figure 1 and requires a well-defined interface in the DEEP design to enable DEEP to cooperate with other conventional name resolution systems like DNS.

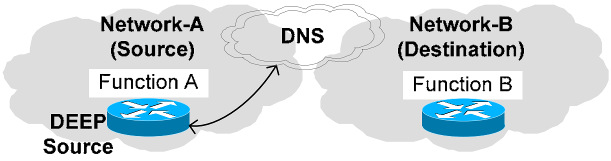

Figure 1: DEEP name resolution illustration.

The second use case describes a simplified situation where the symbolic name of a signaling destination cannot be fully resolved without remote help. For the sake of security, privacy and efficiency it is not always possible for a network to publish all its name resolution information. For instance, a network may provide to the public only a mapping of all its names to the locator of one of its well protected gateways. But it would not provide name resolution information for its accounting center to the public both for security and manageability reasons, since the locator of the accounting center may change. This requires a multi-layer resolution system and DEEP is designed to provide optimal support for such a system.

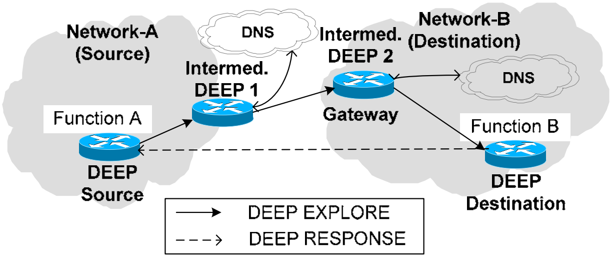

Figure 2: DEEP name resolution illustration.

In Figure 2, a signaling application for “Function A” in Network A starts a session and wishes to send a signaling message to its remote counterpart with the symbolic name functionB@networkB. The source DEEP at the same node as Function A asks the central DEEP entity in network A (Intermed. DEEP 1) to find out the locator of the remote endpoint. DEEP contacts the local DNS to resolve the name. Since Function B can be resolved only in Network B, the locally available DNS, which has access only to public information, can resolve only a part of the symbolic name, and maps “Network B” to the IP address of a gateway of network B.

After receiving the response from the local DNS, DEEP sends the DEEP request to the remote DEEP (Intermed. DEEP 2) located at the gateway. The remote DEEP finds the locator of function B with the help of the DNS of network-B and forwards the DEEP information there.

The destination DEEP at the receiving Function B returns this information back to the source DEEP in network A. This way the full symbolic is resolved in two steps. This use case requires DEEP to support multi-layered naming schemes and to resolve a multi-layer symbolic name in multiple steps in different locations.

4. DEEP Design

DEEP is a protocol for resolving symbolic names into contact addresses like locators, port numbers and protocol types. DEEP does not define a resolution infra-structure. Rather, it employs existing infrastructures such as DNS. From the use cases presented in the previous section the following basic requirements can be derived for DEEP:

- DEEP must resolve symbolic names into contact addresses like locators, port numbers and protocol types using a heterogeneous, existing name resolution infrastructure.
- DEEP must allow a multi-stage, sequential name resolution via one or more intermediary DEEP nodes.
- DEEP must provide a name resolution interface to applications/protocols to accept symbolic names and returns contact addresses.
- DEEP nodes shall be stateless.

In the current DEEP design, mobility is not covered, and therefore no corresponding requirement is formulated.

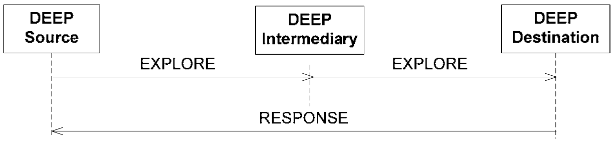

Figure 3: DEEP messaging.

Our proposal for DEEP follows a simple query-response pattern as illustrated in Figures 2 and 3. The DEEP instance at the source application sends an EXPLORE message to the next DEEP node, a DEEP Intermediary node, known to it e.g. by pre-configuration. The DEEP source includes its own contact address. The DEEP Intermediary node may use available name resolution systems, e.g. DNS to fully resolve the destination symbolic name or to find the contact address of the next DEEP node it should contact (e.g. another DEEP Intermediary node, such as the gateway of another network). Name resolution state that indicates to what extent the symbolic name has already been resolved is carried in the EXPLORE message. Thus, this information doesn’t need to be stored in intermediary nodes and DEEP nodes can be stateless. The EXPLORE message is eventually delivered to the node implementing the function corresponding to destination symbolic name, the DEEP Destination. The RESPONSE message is directly sent from the DEEP destination back to the DEEP Source.

In the application/protocol calling DEEP, it is essential that name binding state is cached for each signaling session for some time to avoid excessive use of

DEEP for each signaling message to be sent. This name binding state maps the symbolic name to the contact address of each signaling destination. Whenever a signaling application sends a message destined to a symbolic name, a local process is initiated to find whether a mapping to a contact address already exists. In some sense, our proposal is thus a generalization of the Host Identity Layer ("Layer 3.5") introduced by Host Identity Protocol (HIP) [9]. In HIP, cryptographic identifiers rather than symbolic names are resolved into IP addresses. The name binding state has a lifetime, which may also depend on information received by the DEEP source.

5. DEEP Protocol

All DEEP messages consist of a common header and Type-Length-Value encoded objects (see Figure 4) [12]. The common header identifies the type of message, and has a flags field reserved for further use. Both EXPLORE and RESPONSE messages contain a Name Resolution State (NRS) object. In addition, the EXPLORE message has a DEEP Source Info (DSI) object and the RESPONSE has a Remote Contact Info (RCI) object.

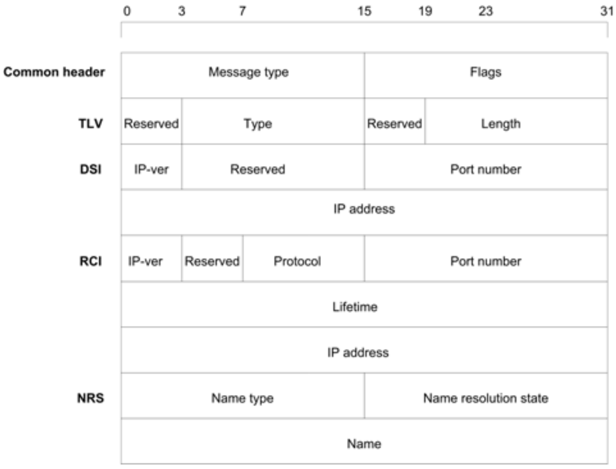

Figure 4: DEEP protocol objects.

The purpose of the NRS object is to contain the symbolic name to be resolved. The name type indicates the syntax and semantics of the name. The name resolution state expresses the state of the name resolution process (which parts of the name have been resolved). DSI object is used to include IP-version, port and IP address information of the source to the EXPLORE-message. RCI object is used for expressing the same information as the DSI object, but in addition lifetime of the name binding state is included in the object. The lifetime is expressed in milliseconds.

DEEP is a stateless Query-Response protocol, which consists of the source, intermediary and destination DEEP-nodes processing the messages. The EXPLORE message (query) is forwarded potentially via multiple intermediary DEEP-nodes to the destination, while the RESPONSE is sent directly back to the IP address contained in the DSI-object of the EXPLORE.

The DEEP messages in a successful name resolution process at the different nodes are processed as follows:

DEEP-source:

When a DEEP-source receives a request to resolve a symbolic name, firstly the local services are used for name resolution. If the whole name cannot be resolved, an EXPLORE message is created, which contains the current state of the name resolution process in the NRS-object, and the contact information of the source in the DSI-object. The message is sent to the next intermediary DEEP-node towards the destination based on the information received from the local name resolution service.

When a DEEP-source receives a RESPONSE message, it checks the presence of the RCI-object. If found, the name resolution state is retrieved from the NRS-object and delivered to the external entity which initially requested resolution of the symbolic name.

DEEP-intermediary:

When an intermediary DEEP-node receives an EXPLORE, the symbolic name contained in the NRS-object is checked whether or not it has been completely resolved. If not, then it is checked from local configuration whether the purpose of this node is to perform local name resolution for the symbolic name, or to forward the EXPLORE to the next intermediary DEEP-node. If local name resolution has to be performed, the unresolved parts of the symbolic name are resolved. As a result the name resolution state is updated in the NRS-object, which is included in the EXPLORE message to be sent to the next intermediary DEEP-node towards the destination DEEP-node.

DEEP-destination:

Once the destination has received an EXPLORE message, the name resolution state is updated for the symbolic name stored in the NRS-object. If the whole symbolic name can be resolved, a RESPONSE is created, which includes the updated NRS-object and a RCI-object, which consists of the contact information of the destination and lifetime of the name resolution state. The message is sent to the address indicated in the DSI-object of the EXPLORE message.

7. Example

In the following, we provide an example scenario that illustrates the major DEEP advantages, namely its capability to interwork with different name resolution technologies, e.g., DNS, LLMNR and mDNS, invoked for actual symbolic name resolution at each DEEP hop, and its ability to hide these different frameworks from the entities using DEEP. As mentioned above, DEEP accomplishes this by providing a unified name resolution interface to such entities. This example is divided into two phases (one with and another without infrastructure support) and represents the envisioned dynamic communication scenarios in the context of

Beyond 3G networks, where small moving networks, such as PANs, connect/disconnect to/from other neighbor devices/networks dynamically while moving around. The example assumes that IP connectivity is pre-established between networks and therefore routing aspects are omitted.

Figure 5 illustrates the phase 1 of the example, where PAN_1 is connected to an Infrastructure Network (IN) in order to reach the destination network (Network-B). In phase 2, PAN_1 moves and loses its connectivity to IN, as represented in Figure 6. It then searches for other networks in the neighborhood, detects PAN_2, and connects to it.

In phase 1, PAN_1 is interworking with Network-B through IN. The signaling application (QoS) running within PAN_1 (in R1) specifies its destination signaling endpoints using a symbolic name; the local DEEP is then invoked to resolve the symbolic name into the contact address like locator, port number and protocol type as represented in Figure 5; R1 node is acting as DEEP source, R2 and R3 are DEEP intermediaries and R4 is DEEP destination. Once the signaling application has requested the DEEP to resolve "qos.net2.org" Fully Qualified Domain Name (FQDN), it sends a DEEP EXPLORE message towards R2 (name resolution gateway) based on its local configuration.

R2 then uses public DNS to resolve the received FQDN. The DEEP instance in R2 is configured to first try to perform the HIP type of DNS lookup (QTYPE=HIP). The DNS has only CNAME Resource Record (RR) for this FQDN pointing to r3.net2.org, which is returned. Nevertheless, it should be noted that the exact DNS protocol message exchange sequence depends on the involved DNS servers and their configuration, but for the sake of simplicity, we assume that R2 directly communicates with the authoritative DNS of "net2.org". Then, R2 performs another DNS lookup for "r3.net2.org" FQDN and this lookup results Host Identity Tag (HIT), Host Identity (HI) and the FQDN of RVS (RendezVous Server) [9] according to [10]. Now, R2 has sufficient contact information of the next DEEP node (R3) and after it has resolved the IP address of RVS by performing the A type of DNS lookup (QTYPE=A) for "rvs1.org" FQDN. After this, R2 and R3 are performing HIP Base Exchange [9] and R2 sends a DEEP EXPLORE message carrying the original FQDN ("qos.net2.org") to R3.

R3 uses a local DNS to resolve the received FQDN ("qos.net2.org"), which points to another FQDN ("r4.net2.org") based on CNAME RR stored in the local DNS. After that, R3 resolves an IP address of R4 node and sends a DEEP EXPLORE message.

Once R4 has received the message, it checks local configuration to select port number and protocol type parameters according its current configuration for the response. After this, R4 sends a DEEP RESPONSE message back to R1. This message carries the IP address of R4, port number and protocol type that can be used for the requested QoS service.

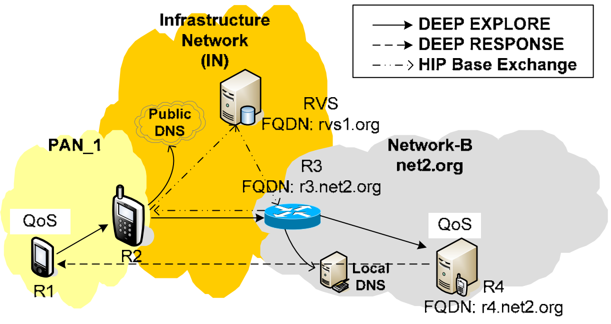

Figure 5: Example Scenario – Phase 1.

Once R1 receives the DEEP RESPONSE message, it provides the name resolution results for the signaling application requesting the name resolution service.

HIP is only used between R2 and R3 for transferring DEEP EXPLORE messages. Additionally, HIP also provides host reachability for "net2.org"; i.e. when the network is moving only R3's contact information is updated in the RVS. It should be noted that the way how the DNS was used in this example is only one alternative and that there are also other ways such as NAPTR (Naming Authority Pointer) RRs [11].

In phase 2, after PAN_1 has established connectivity with PAN_2, the QoS signaling application at PAN_1 starts a new session. DEEP is invoked to resolve the symbolic name of the destination signaling endpoint ("qos.pan2.") specified by the signaling application. R1 sends a DEEP EXPLORE message towards R2 – PAN_1's name resolution gateway – in order to get the contact address, port number, and protocol type corresponding to the symbolic name passed by the QoS signaling application. R2 receives this message, and by interacting with the local name resolution service (multicast DNS in this case), it is able to acquire the locator corresponding to PAN_2's name resolution gateway (R3); at the mDNS level R2 performs a DNS QUERY (QTYPE=A) for "pan2.local." (the suffix "local." is appended due to a mDNS peculiarity, which explicitly specifies this domain name for all nodes connected to a local link), and sends it to the mDNS multicast address according to [3].

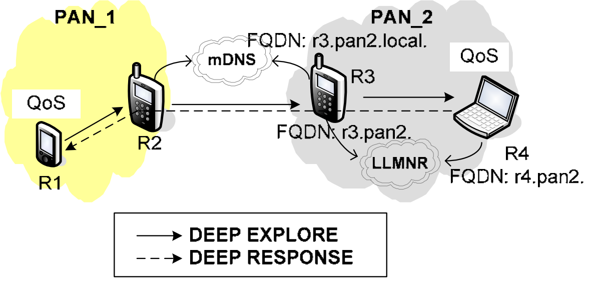

Figure 6: Example Scenario – Phase 2.

R3 recognizes that this is a request for it and returns a DNS QUERY RESPONSE; this message includes its canonical name ("r3.pan2.local.") and IP address; DEEP only takes the IP address. Afterwards, R2 sends a DEEP EXPLORE message towards R3; this message still includes the original symbolic name “qos.pan2.”. Upon receiving the DEEP EXPLORE message, DEEP in R3 interacts with local name resolution (LLMNR) in order to acquire the locator of the node running the QoS signaling application in PAN_2; at the LLMNR, the resolver makes a DNS QUERY (QTYPE=A) for “qos.pan2.” (the suffix “local.” is not required in this case, since the LLMNR mechanism does not impose any rigid local domain name).

This query is sent to the LLMNR multicast address according to [2]. R4 recognizes that this is a query for it and replies with a DNS QUERY RESPONSE including its canonical name (“r4.pan2.”) and IP address; again, DEEP only takes the IP address. At this point, R3 sends a DEEP EXPLORE towards R4. R4 then checks its local configuration concerning the port number and protocol type parameters that should be included in the DEEP RESPONSE to be sent back towards R1. Subsequently, it creates the DEEP RESPONSE message and sends it directly to the source IP address included in the DEEP EXPLORE message (in this case R1’s IP address); in Figure 6, from DEEP viewpoint this message is directly exchanged between R4 and R1 based on IP routing. Finally, the signaling application is able to contact R4.

8. Conclusions

The DEEP is a new protocol to support multi-stage name resolution over multiple separate name resolution systems. This paper presented the DEEP protocol; the rationale behind it, design principles and two concrete examples to show how it could be used with existing name resolution systems like DNS. The examples demonstrated how multi-stage name resolution works and how the name resolution information can be divided into two parts; public and private. This type of separation better supports dynamic network configuration, since network internal configuration changes do not require the update of the public part. Network reachability is achieved through the public name resolution systems over which service reachability is implemented based on locally stored configuration data.

The DEEP scalability is studied in [8] and therefore it was omitted herein. There is also a working prototype that was used as proof-of-concept [12]. The protocol is a candidate protocol in Ambient Networks EU project [14] to be used for name resolution between Ambient Networks. The development work with the DEEP is continuing in the project to study, for instance, the support of locator/identifier split such as HIP, security, mobility, and stateful DEEP.

9. Acknowledgment

This work is supported by the Ambient Networks Project, partially funded by the European Commission under its Sixth Framework Programme. It is provided “as is” and without any expressed or implied warranties, including, without limitation, the implied warranties of fitness for a particular purpose. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Ambient Networks Project or the European Commission.

10. References

[1] J. Crowcrfot, S. Hand, R. Mortier, and T. Roscoe, A. Warfield, “*Plutarch: An Argument for Network Pluralism*”, In ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA'03) (2003).

[2] L. Esibov, B. Aboba, and D. Thaler, ”*Linklocal multicast name resolution (LLMNR)*”, Internet Draft (Work in Progress) draft-ietf-dnsext-mdns-45, Internet Engineering Task Force, October 2005.

[3] S. Cheshire, and M. Krochmal, "*Multicast DNS*", draft-cheshire-dnsext-multicastdns, Internet Draft (Work in Progress), June 2005.

[4] J. Jeong, J. Park, and H. Kim, “*DNS Name Service based on Secure Multicast DNS for IPv6 Mobile Ad Hoc Networks*”, in the Proceedings of the International Conference on Advanced Communication Technology, Feb. 2004.

[5] W. Adjie-Winoto, E. Schwartz, and H. Balakrishnan,” *An Architecture for Intentional Name Resolution and Application-level Routing*”, MIT Technical Report MIT/LCS/TR-775, Feb. 1999.

[6] D. Doval, and D. O'Mahony, ”*Nom: Resource Location and Discovery for Ad Hoc Mobile Networks*”, in the Proceedings of the 1st Annual Mediterranean Ad Hoc Networking Workshop, Medhoc –Net, Sept. 2002.

[7] N. Niebert, A. Schieder, H. Abramowicz, G. Malmgren, J. Sachs, U. Horn, Ch. Prehofer, and H. Karl, “*Ambient Networks: An Architecture for Communication Networks beyond 3G*”, IEEE Wireless Communications, vol. 11, pp.14-22, April 2004.

[8] P. Pääkkönen, N. Akhtar, R. Campos, C. Kappler, P. Pöyhönen, and D. Zhou, “*Scalability of Name Resolution for Ambient Networks”*, to appear in the 4th International Conference on Wired/Wireless Internet Communications, Bern, May 9-12, 2006.

[9] R. Moskowitz, and P. Nikander, “*Host Identity Protocol Architecture*”, draft-ietf-hip-arch (at RFC Editor), August 2005.

[10] P. Nikander, and J. Laganier, “*Host Identity Protocol (HIP) Domain Name System (DNS) Extensions”*, draft-ietf-hip-dns-05, August 2006.

[11] M. Mealling, and R. Daniel, “*The Naming Authority Pointer (NAPTR) DNS Resource Record*”, RFC 2915, September 2000.

[12] “*Connecting Ambient Networks – Final Architecture, Protocol Design and Evaluation*”, AN public deliverable D3.3, http://www.ambient-networks.org/main/deliverables.html, December 2005.

[13] “AN Framework Architecture”, AN public deliverable D1-5, http://www.ambient-networks.org/main/deliverables.html, December 2005.

[14] The Ambient Networks Project, http://www.ambient-networks.org.