Title:	**Document Version:**
Deliverable D5.1 **Initial evaluation of state of the art Mobile IPv6 alternatives**	2.1

Project Number:	**Project Acronym:**	**Project Title:**
027002	ENABLE	Enabling efficient and operational mobility in large heterogeneous IP networks

Contractual Delivery Date:	**Actual Delivery Date:**	**Deliverable Type* - Security**:**
31/12/2006	22/12/2006	R – PU

* Type: P – Prototype, R – Report, D – Demonstrator, O – Other
** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author:	**Organization:**	**Contributing WP:**
Antonio F. Gómez Skarmeta	UMU	WP5

Authors (organizations):

Fritsche, Wolfgang (IABG), Fu, Xiaoming (UGOE), García Segura, Pedro (UMU), Gómez Skarmeta, Antonio F. (UMU), Kiani, Adnan (Brunel), Le, Deguang (UGOE), Lei, Jun (UGOE), Mayer, Karl (IABG), Pérez, Alejandro (UMU), Yang, Xiaodong (UGOE)

Abstract:

This report provides a state of the art analysis of Mobile IPv6 alternatives under study within different standardization forums and how these proposals could affect the future deployment of mobility and security as a service in operational environments. Recommendations on the most promising technologies to be deployed are also given.

Keywords:

IPv6, Mobile IPv6, FARA, HIP, I3, Si3, Shim6, Proxy Mobile IPv6

Revision History

The following table describes the main changes done in the document since created.

Revision	Date	Description	Author (Organization)
v1.0	29/08/2006	Document creation	Antonio F. Gómez Skarmeta and Pedro García Segura (UMU)
V1.1	25/09/06	Added section 4.1, 4.2, 4.3, 4.4.1, 4.4.2, 4.4.3 and 4.4.4 Added Section 4.5, 4.6. Section 7	Karl Mayer (IABG) Xiaodong Yang and Deguang Le (UGOE)
V1.2	05/10/06	Added section 8.1, 8.2, 8.3 and 8.5	Wolfgang Fritsche (IABG)
V1.3	09/10/2006	Added section 8.4	Jun Lei (UGOE)
V1.5	25/10/06	Updated section 2	Antonio F. Gómez Skarmeta and Pedro García Segura (UMU)
V1.6	6/11/06	Added section 2.1	Antonio F. Gómez Skarmeta and Pedro García Segura (UMU)
V1.7	07/11/06	Updated Section 2.1	Xiaodong Yang (UGOE)
V1.8	08/11/06	Updated section 8 on NetLMM reflecting the agreements taken in Shanghai	Wolfgang Fritsche (IABG)
V1.9	9/11/2006	Restructured and update section 4.6	Karl Mayer (IABG)
V1.10	11/11/2006	Updated section 7.3.1.1	Adnan Kiani (Brunel)
V1.11	12/11/2006	Updated Section 4.6	Xiaodong Yang (UGOE)
V1.12	14/11/2006	Some suggestions/comments to Section 2.1	Xiaoming Fu (UGOE)
V1.13	15/11/2006	Updated Section 7	Deguang Le (UGOE) Adnan Kiani (Brunel)
V1.14	15/11/2006	Added section 4.7	Karl Mayer (IABG)
V1.15	18/11/2006	Updated sections 4.1 and 4.6	Xiaodong Yang (UGOE)
V1.17	20/11/2006	Updated section 5	Antonio F. Gómez Skarmeta and Pedro García Segura (UMU)
V1.18	21/11/2006	Minor modifications in section 4.6	Karl Mayer (IABG)
V1.19	28/11/2006	Updated section 2	Pedro García Segua (UMU)
V1.20	28/11/2006	Minor additions to section 4.6	Karl Mayer (IABG)
V1.21	28/11/2006	Updated section 7.3	Deguang Le (UGOE)
V1.22	29/11/2006	Added relativity modifiers for NETLMM	Wolfgang Fritsche (IABG)

Revision	Date	Description	Author (Organization)
V1.23	29/11/2006	Added initial summary table Modifications to “Support for legacy hosts” criterion	Pedro García and Alejandro Pérez (UMU)
V1.24	29/11/2006	Added section 9. PMIPv6	Jun Lei (UGOE)
V1.25	30/11/2006	Revised section 2.1 Added PMIPv6 to the summary table Updated evaluation criteria	Pedro García and Alejandro Pérez (UMU)
V 1.26	30/11/2006	Revised Section 7 Added the assessment of PMIPv6 into the summary table	Jun Lei (UGOE)
V 1.27	01/12/2006	Added introduction Added conclusions Updated summary table Style revision	Pedro García and Alejandro Pérez (UMU)
V 1.28	03/12/2006	Minor modifications on the Introduction and Conclusions Revised Section 6.4 and 6.5 Updated Section 7	Jun Lei (UGOE)
V1.29	03/12/2006	Updated Introduction and Conclusions Document edits	Pedro García (UMU)
V1.30	04/12/2006	Grouped all the references in one section Updated figures of PMIPv6	Alejandro Pérez (UMU)
V1.31	04/12/2006	Updated Figure 3.7 and Figure 3.8	Xiaodong Yang (UGOE)
V1.32	05/12/2006	Updated Executive Summary Updated figures of SHIM6	Alejandro Pérez (UMU)
V1.33	15/12/2006	Updated styles, text and figures Initial review and preparation for the final review	Pedro García Segura (UMU)
V1.34	19/12/2006	Included the updated figures in PPT format Fixed some figure references Fixed some issues in I3 section Included changes from IABG Included changes from TSSG	Alejandro Pérez Méndez (UMU)
V2.0	21/12/2006	Document Review	Pedro García Segura, Antonio F. Gómez Skarmeta, Alejandro Pérez Méndez (UMU)
V2.1	22/12/2006	Minor fixes to shim6 section	Pedro García Segura (UMU)

Executive Summary

This deliverable provides a state of the art analysis of the Mobile IPv6 alternatives under study within different standardization forums, e.g., IETF, and other experimental approaches currently published in the scientific literature, describing how these proposals could affect the future deployment of mobility and security as a service in operational environments.

In order to assess the identified technologies, a set of evaluation criteria is defined. This ensures that all the relevant features of the mobility management systems are evaluated in a consistent way, and enables a straightforward comparison of the different solutions.

The analyzed technologies are the following:

- **Host Identity Protocol (HIP)**. HIP is a network protocol intended to maintain shared IP-layer state between end hosts. HIP provides decoupling between the IP network address and the host identifier, and hence communication continues even on IP address changes.

- **Internet Indirection Infrastructure (*i3*) and related technologies such as FARA**. *i3* proposes an overlay-based indirection infrastructure that offers a rendezvous-based communication abstraction, decoupling the act of sending a packet from the act of receiving it. FARA is a more experimental approach, defining a new organization of network architecture concepts, but it is based on the same indirection principle.

- **Site Multihoming by IPv6 Intermediation (SHIM6)**. SHIM6 is a multihoming solution for IPv6 based on the addition of a new network sub-layer. This new layer allows for separating the well known IP location-identifier association by managing a group of assigned IP address and providing to the upper layers a single fixed address.

- **Network-based Localized Mobility Management (NetLMM) and Proxy Mobile IPv6 (PMIPv6)**. NetLMM and PMIPv6 are two alternative technologies that perform localized mobility management, allowing a MN to move from one access router to another inside the same organization in a transparent way. This kind of localized management allows for the reduction in mobility signaling traffic and the improvement of handover performance. One of the most remarkable features is that they work with unmodified (legacy) MNs.

After the assessment of each mobility management solution, the most interesting and deployable technologies are selected for further study and improvements within the second year of the ENABLE project.

Table of Contents

1. Introduction ***8***

2. Evaluation Criteria ***10***

2.1 Reference Mobile IPv6 Assessment **15**

2.1.1 Functional assessment 15

2.1.2 Deployment assessment 16

2.1.3 Security assessment 17

2.1.4 Performance assessment 17

2.1.5 Additional properties 18

3. Host Identity Protocol ***19***

3.1 Overview and Objectives **19**

3.1.1 Current situation in the Internet 19

3.1.2 HIP approach 20

3.1.3 Host Identity namespace 20

3.1.4 Host Identity Protocol itself 20

3.1.5 Interoperability with legacy networks 21

3.1.6 Status 21

3.2 HIP Base Exchange **22**

3.3 ESP Transport **23**

3.4 Mobility **24**

3.4.1 UPDATE message exchange 25

3.4.2 Rendezvous and registration mechanism 26

3.4.3 Domain Name Service (DNS) extension 27

3.4.4 HIP initialization process 28

3.4.5 Simultaneous movement of both HIP peers 30

3.5 Multihoming **31**

3.5.1 Host multihoming 31

3.5.2 Site multihoming 31

3.6 HIP Assessment **32**

3.6.1 Functional assessment 32

3.6.2 Deployment assessment 33

3.6.3 Security assessment 35

3.6.4 Performance assessment 35

3.6.5 Additional properties 36

3.7 Conclusions 36

4. I3 and Related Approaches 37

4.1 Internet Indirection Infrastructure 37

4.1.1 Overview 37

4.1.2 Robust Overlay Architecture for Mobility 41

4.1.3 Secure-*i3* 43

4.1.3.1 Hiding IP addresses 43

4.1.3.2 Defending against attacks 44

4.1.3.3 Avoiding new vulnerabilities 45

4.1.4 i3 assessment 47

4.1.4.1 Functional assessment 47

4.1.4.2 Deployment assessment 48

4.1.4.3 Security assessment 49

4.1.4.4 Performance assessment 50

4.1.5 Conclusions 50

4.2 FARA 51

4.2.1 Overview 51

4.2.2 Security 53

4.2.3 Performance 53

4.2.4 Conclusions 54

5. SHIM6 56

5.1 SHIM6 Overview 56

5.2 Potential of SHIM6 Mobility Support 57

5.2.1 Dynamic addition and deletion of locators 57

5.2.2 Mobility with single locator and ULID-Pair 58

5.2.3 Seamless mobility with multihoming support 59

5.3 SHIM6 Mobility Assessment 61

5.3.1 Functional assessment 61

5.3.2 Deployment assessment 62

5.3.3 Security assessment 63

5.3.4 Performance assessment 63

5.3.5 Additional properties 64

5.4 Conclusions 64

6. NetLMM 65

6.1 Motivation 65

6.2 Protocol Overview 66

6.3 Protocol Procedures 68

6.4 NetLMM Assessment 72

6.4.1 Functional assessment 72

6.4.2 Deployment assessment 74

6.4.3 Security assessment 75

6.4.4 Performance assessment 75

6.4.5 Additional properties 76

6.5 Conclusions 77

7. Proxy Mobile IPv6 (PMIPv6) 79

7.1 PMIPv6 Overview 79

7.2 PMIPv6 Procedure 79

7.3 PMIPv6 Operation 82

7.3.1 Home agent operation 82

7.3.2 Proxy mobile agent operation 83

7.3.3 Mobile node operation 83

7.4 PMIPv6 Assessment 83

7.4.1 Functional assessment 84

7.4.2 Deployment assessment 85

7.4.3 Security assessment 86

7.4.4 Performance assessment 86

7.4.5 Additional properties 86

7.5 Conclusions 87

8. Summary Table 88

9. Conclusions 90

10. References 91

1. INTRODUCTION

This deliverable provides a state of the art analysis of the Mobile IPv6 alternatives under study within different standardization forums, e.g., IETF, and other experimental approaches currently published in the scientific literature, describing how these proposals could affect the future deployment of mobility and security as a service in operational environments.

In order to assess the identified technologies, a set of evaluation criteria is defined in Section 2. This ensures that all the relevant features of the mobility management systems described in this document are evaluated in a consistent way, and enables a straightforward comparison of the different solutions. A summary table is provided in Section 8 in order to sum up the evaluations and to allow the reader to easily compare the technologies.

Section 3 describes and evaluates the Host Identity Protocol (HIP), a network protocol intended to maintain shared IP-layer state between end hosts. HIP provides decoupling between the IP network address and the host identifier, and hence communication continues even on IP address changes. It also provides integrity protection and optional encryption for upper-layer protocols (i.e. TCP and UDP).

The Internet Indirection Infrastructure (*i3*) and other related technologies, such as FARA, are evaluated in Section 4. *i3* proposes an overlay-based indirection infrastructure that offers a rendezvous-based communication abstraction, decoupling the act of sending a packet from the act of receiving it. FARA is a more experimental approach, defining a new organization of network architecture concepts, but it is based on the same indirection principle.

Section 5 assesses SHIM6, a multi-homing solution for IPv6 based on the addition of a new network sub-layer (shim). This new layer allows for separating the well known IP location-identifier association by managing a group of assigned IP address and providing to the upper layers a fixed address called a ULID. Since new locations can be added and deleted dynamically, mobility is also allowed.

Finally, sections 6 and 7 evaluate NetLMM and PMIPv6, two alternative technologies that perform localized mobility management, allowing a MN to move from one access router to another inside the same organization in a transparent way. This kind of localized management allows for the reduction in mobility signaling traffic and the improvement of handover performance. One of the most remarkable features is that they work with unmodified (legacy) MNs. The main difference between them is that PMIPv6 leverages the MIPv6 design and tries to re-use as much as possible what is already there, in particular the home agent (HA), while NetLMM is a new design that has the potential for being a more optimized solution, especially

with reference to the security architecture, but it requires the deployment of new functionality (i.e. a local mobility anchor) in addition to the MIPv6 HA.

In the conclusions section, the most interesting and deployable technologies are selected for further study and improvements within the second year of the ENABLE project.

2. EVALUATION CRITERIA

In order to compare different mobility management systems and their associated protocols, a set of evaluation criteria is specified in this document. Each mobility management system is evaluated against these criteria. It should be noted that some of these criteria are absolute (A), while others are relative (R). An example of an absolute criterion is “support for simultaneous movement of both endpoints of a communication session”. Such a criterion can either be met, or not be met by a given mobility management system. By contrast, a relative criterion such as “scalability” can be met with respect to a certain degree, e.g. to a “medium” or “high” degree.

Whether or not a particular mobility management system meets a given criterion depends on the properties of the system. This section shows the criteria and a brief description of the properties on which they depend. Note that the relationship between criteria and properties in n:m, i.e. a single property can influence whether or not multiple criteria are met, and vice versa. Also note that the list of properties in this table is not exhaustive, i.e. a given mobility management system may possess critical properties that are not listed in the table.

The evaluation criteria are classified in the following categories:

- **Functional criteria.** Evaluate features of the mobility management system, such as support for multihoming, flexible placement of service elements, scalability, and other functional aspects of the system.

- **Deployment criteria.** These criteria take into account the possible issues that would be encountered during the deployment of the mobility management system, from estimated deployment and operation efforts to more specific aspects such as support for legacy hosts and transparency for legacy applications.

- **Security criteria.** Criteria related to the availability of mechanisms that enable the mobility management system to defend itself against misuses of the mobility features, such as stealing of legitimate addresses, Denial of Service (DoS) attacks and eavesdropping attacks.

- **Performance criteria**. Include aspects relevant to the performance of the mobility management system, such as support for routing and signaling optimizations, and minimization of packet loss during the handover process.

- **Additional properties.** Individual systems may have some unique properties that may be important when one must choose one out of many systems. Due to fact that such properties are only important in specific circumstances, it may not be reasonable to regard them as major criteria.

Table 2-1 enumerates each criterion, including its type (absolute or relative) and a brief description. Relativity modifiers are defined for every relative criterion.

Table 2-1: Evaluation criteria

Functional Criteria	
Criterion	**Description**
Support for simultaneous movement of both endpoints (**A**)	The system supports simultaneous movement of both endpoints by using a rendezvous service, DNS-based techniques or any other techniques.
Support for simultaneous use of multiple interfaces (multihoming) (**A**)	This criterion evaluates the availability of multihoming support, by which the MN can gain access to the Internet through multiple links simultaneously and dynamically switch links while moving. Multihoming support enables load sharing techniques, coupling of streams to interfaces, reachability tests on multiple address pairs, etc.
Support for flexible placement of service elements (**R**)	The system supports flexible placement of service elements if the involved mobility anchor can be deployed in the home network (i.e. the network to which the IP address used by the MN to communicate belongs to), visited network, and/or third party network. Relativity modifiers: **Home only**, **Home or visited**, **Third-party only**, **Any**
Robustness level and failover support (**R**)	This criterion evaluates the system's support for failure recovery and its overall robustness level. Relativity modifiers: **None** If some component fails, all MNs lose their connections. **Partial** System automatically recovers after failures. **Full** System functions are redundant, i.e. even in the presence of failures, MNs continue to get service.

Scalability (**R**)	This criterion evaluates how many MNs per domain the system supports, and how many movements per unit time. Relativity modifiers: **Low**, **Medium**, **High**
Deployment Criteria	
Criterion	**Description**
Transparency to legacy applications (**A**)	This criterion evaluates if the mobility management mechanism is transparent and doesn't require changes to current services and applications. When evaluating this criterion the following points should be considered: ▪ If applications on both MN and Correspondent Node (CN) continue to identify communication partner based on a socket. ▪ The layer where the mobility solution is located. ▪ If new (global or not) namespaces are introduced by the solution.
Support for legacy hosts (**R**)	This criterion evaluates whether the mobility management system offers support for legacy hosts. This support implies that no modifications to the legacy host operating system, network stack, etc. are needed. Relativity modifiers: **MN only** Support for legacy MNs only. **CN only** Support for legacy CNs only. **Both** Support for both legacy MNs and CNs. **None** No support for legacy hosts.

Deployment Effort (**R**)	This criterion evaluates how much effort is required to deploy the system. When evaluating this criterion one should consider the number of new architectural entities that must be deployed, as well as the need for different providers to sign agreements. Relativity modifiers: **Low**, **Medium**, **High**
Operational Effort (**R**)	This criterion evaluates the number of entities that must be regularly monitored. This includes the number of times that the entities have to be re-configured, and how often manual re-keying is necessary. Relativity modifiers: **Low**, **Medium**, **High**
Need to deploy new security infrastructure (**R**)	This criterion evaluates the system's need for the deployment of a new security infrastructure. This may include: ▪ Deployment of new MN-specific long term credentials. ▪ Deployment of new long term credentials to enable roaming partners to communicate securely. ▪ Possibility of using the system with existing infrastructure, such as AAA or web PKI. Relativity modifiers: **No need** System can completely rely on existing infrastructure. **Partial need** System only needs partial deployment of new security infrastructure components. **Full need** MNs need new long-term credentials.
Maturity (**R**)	This criterion evaluates whether the mobility management system is being implemented and, if yes, to what extent. Relativity modifiers: **Low**, **Medium**, **High**

Security Criteria	
Criterion	**Description**
DoS resistance (**R**)	This criterion evaluates the capability of the mobility management solution to protect itself against DoS attacks. Relativity modifiers: **Low**, **Medium**, **High**
Support for location privacy (**A)**	This criterion evaluates whether the mobility system offers support for location privacy.
Performance Criteria	
Criterion	**Description**
Support for packet loss minimization (**R**)	This criterion evaluates whether the system can handle routing of packets when the point of attachment of the MN changes, preserving on-going sessions at transport layer or above. For this, the system might support the identifier/locator split paradigm. This could also be coupled with techniques that reduce packet loss during handover. Relativity modifiers: **Low** Applications insensitive to packet loss/reordering are supported. **Medium** Applications sensitive to loss/reordering are supported. **High** Real-time applications (e.g. VoIP) are supported.
Support for routing optimization (**A**)	This criterion evaluates whether the system uses triangular routing or routing optimizations to enable direct MN/CN routing.

Support for signaling optimizations (**R**)	This criterion evaluates the existence of multiple, hierarchically deployed mobility anchors. Relativity modifiers: **None** Every handover involves signaling back to home domain. **Some** Handover signaling goes back to intermediate anchor. **Full** Signaling may stay within current domain.

2.1 Reference Mobile IPv6 Assessment

This section contains an assessment of Mobile IPv6 based on the evaluation criteria defined in the previous section. This assessment will be used as a reference for the reader, and will make very simple to verify if a particular technology improves some aspect of MIPv6. The version of MIPv6 evaluated in this section includes the base protocol defined in [RFC3775] and [RFC3776], as well as the extensions defined in [ENA-D1.2] (i.e. AAA interfaces with the HA, dynamic bootstrapping and HA load sharing).

2.1.1 Functional assessment

- **Support for simultaneous movement of both endpoints**

  MIPv6 uses binding updates through the HAs associated to each endpoint to support simultaneous movement of end hosts.

- **Support for simultaneous use of multiple interfaces (multihoming)**

  MIPv6 alone does not support end host multi-homing.

- **Support for flexible placement of service elements**

  The mobility anchor in MIPv6 is the HA located in the home network. The relativity modifier for this criterion is **Home only**.

- **Robustness level and failover support**

  The mobility anchor in MIPv6 is the HA. If this anchor fails, all MNs lose their connections. The relativity modifier for this criterion is **None**.

- **Scalability**

  The mobility anchor in MIPv6 is the HA. Since a group of MNs share the same HA, the amount of work to be done by that HA will be increased for each new MN that joins the group. However, if route optimization is used, the work of this HA becomes significantly lower, since the HA does not need to route all the MN traffic. Note that the use of route optimization is not a decision on the part of the HA, so the number of MNs that a HA can support varies with the amount that support routing optimization.

  The extensions to the MIPv6 base protocol proposed in [ENA-D1.2] include HA load balancing schemas and dynamic bootstrapping of the mobility service that may be used to distribute the load among a number of different HAs, and improve the scalability of the system, so the relativity modifier for this criterion is **High**.

2.1.2 Deployment assessment

- **Transparency to legacy applications**

  MIPv6 is a network layer protocol, so it enables the mobile node to maintain the combination between the network layer IPv6 address and the transport layer socket address in mobility. MIPv6 does not introduce new global namespaces thus has legacy applications compatibility.

- **Support for legacy hosts**

  MIPv6 supports communication with a legacy IPv6 correspondent node when route optimization mode is not used. The relativity modifier for this criterion is **CN only** as the MN must support MIPv6.

- **Deployment Effort**

  MIPv6 needs at least a HA entity in each home network supporting mobility. When all the mobility parameters are statically provisioned, the deployment effort would be limited to the deployment of the HAs. However, the more flexible architecture defined in [ENA-D1.2] (with dynamic configuration of mobility via bootstrapping solutions) implies a higher number of architectural entities (e.g. AAA clients, proxies, servers, etc.) that would have to be deployed, so the relativity modifier for this criterion is **Medium**.

- **Operational Effort**

  The entities to be monitored are the HAs and the nodes of the AAA infrastructure. Although IPsec SAs between MN and HA are often rekeyed automatically by a keying

protocol (i.e. IKEv2), it may be necessary to change/refresh the pre-shared keys or certificates used in the MN authentication.

The relativity modifier for this criterion is **Medium**.

- **Need to deploy new security infrastructure**

The security in MIPv6 is given by protecting MIPv6 signaling (and optionally all) traffic using IPsec. Usually, MIPv6 IPsec SAs are created using a keying protocol like IKEv2. Since this kind of protocol allows for the utilization of existing AAA infrastructure or certificates based on a PKI with little or no change, the relativity modifier for this criterion is **No need**.

- **Maturity**

MIPv6 has not been deployed in an operational environment yet. However, the technology has been developed by the IETF since middle of 1990s and the basic specification was standardized in 2004, with a number of available implementations and ongoing extensions to security enhancements, load sharing and possible multihoming support. In this sense it could be regarded as a highly mature technology, so the relativity modifier is **High**.

2.1.3 Security assessment

- **DoS resistance**

When MIPv6 signaling is authenticated (using IPsec or RFC 4285), most DoS attacks against MIPv6 can be avoided. However, some kinds of DoS attacks are inevitable. Therefore the relativity modifier for this criterion is **Medium**.

- **Support for location privacy**

As described in [MIP6LPPS], MIPv6 provides location privacy when no routing optimization is used and when ESP encryption of inner IP packet is used between MN and HA.

2.1.4 Performance assessment

- **Support for packet loss minimization**

In MIPv6, the changes of CoA must be notified to the HA (and to the CN if route optimization is being used). Since the CoA change notifications are reactive, the HA (or CN) may send some packets to the old CoA until they update the Binding Update Cache. This may lead to a significant loss of packets, particularly in the case where the HA is

assigned in a network situated far away from the current location of the roaming user. This problem can be mitigated by bootstrapping a new HA situated closer to the visited network.

Hence, applications sensitive to loss are not supported, as there will always be a significant packet loss, so the relativity modifier for this criterion is **Low**.

- **Support for routing optimization**

  MIPv6 supports route optimization, allowing the CN to directly route packets to the MN's CoA, rather than through the home network. This improves latency, robustness and reduces home network congestion.

- **Support for signaling optimization**

  When using MIPv6, each handover is always performed with the HA located in the home domain. Therefore, the relativity modifier for this criterion is **None**.

2.1.5 Additional properties

No additional properties have been identified.

3. HOST IDENTITY PROTOCOL

3.1 Overview and Objectives

The Host Identity Protocol (HIP) framework is currently developed and standardized in the IETF HIP working group. One of its objectives - which is the most relevant for this study - is the provision of mobility support for end hosts; however, HIP supports not only mobility, but also other features like multihoming and security.

The HIP framework consists of:

- A new HIP namespace.
- The Host Identity Protocol itself.
- Mechanisms to transport user data (so far only ESP transport is defined).
- Mechanisms to support end-host mobility and multihoming.
- A new infrastructure component, the rendezvous server, that is used for mobility purpose.
- A new DNS resource record that can store Host Identities and Host Identity Tags together with IP addresses under a certain domain name.
- A service registration mechanism that can be used, for example, by mobile nodes to register at a rendezvous server.

3.1.1 Current situation in the Internet

Currently there are just two important namespaces in the Internet: IP addresses and DNS names. Thereby, an IP address has two roles. From the network point of view an IP address is used as *locator* for an interface in the Internet, since it is used by routing protocols to find the path to the interface. From the application point of view an IP address is used as host *identifier* and applications expect that the host identifier is stable during sessions.

This situation has some deficiencies. First of all, the network layer is coupled with the transport layer, e.g. the TCP checksum calculation has source and destination IP address as parameters. This means that both layers cannot evolve independently. Furthermore, an IP address couples the locator of an interface with an end point name (identifier), which means that in case of a re-addressing (e.g. due to a mobile node changing its point of attachment to the Internet) the identifier changes as well. As a result, application sessions, which are bound to identifiers, have to be terminated and re-established for the new address. Moreover, authentication for systems

and datagrams is not provided by default; however, depending on the scenario, this may or may not be a requirement.

3.1.2 HIP approach

The HIP approach is based on the introduction of a new namespace and a new kind of shim layer between network and transport layer in all HIP-enabled hosts. The HIP framework removes the identifier role from IP addresses and uses host identifiers instead. Transport layer sockets are no longer bound to IP addresses but to host identifiers (more precisely to the host identity representations, e.g. HITs and LSIs, see below) that do not change in case of a re-addressing event. This enables seamless mobility, i.e. application sessions are not interrupted in case of re-addressing. Additionally, the HIP framework provides support for host authentication prior to any communication as well as mandatory support for secure communication by deploying ESP.

Furthermore, since transport layer sockets are not bound to network layer addresses, transport sessions can span different network address realms without losing end-to-end transparency, e.g. session can span private and global address realms or IPv4 and IPv6 realms.

3.1.3 Host Identity namespace

Additional to the DNS namespace and the IP namespace, the HIP architecture relies on a third namespace: the host identity namespace. Each host gets assigned at least one Host Identifier (HI) that identifies the host uniquely. However, more than one HI can be assigned to a single entity. In order to support security, the HI represents the public key of an asymmetric public/private key pair. The HIP layer manages the dynamic binding of HIs to IP addresses.

In protocols and processes it is beneficial to have fixed length parameters. Therefore, a Host Identity Tag (HIT) or a Local Scope Identifier (LSI) is used instead to represent the HI. The Host Identity Tag (HIT) is a 128 bit hash value of the HI that can be used in IPv6-sized address structures. Due to its length (128 bit) it is unlikely that two HITs of different HIs are equal. Nevertheless, in such a case the HI (the public key) would make the difference. The Local Scope Identifier (LSI) is a 32 bit datum of the HI that can be used in IPv4-sized address structures in IPv4-based protocols and APIs. Due to its short length (32 bit) uniqueness is not assured and therefore its scope is only local.

3.1.4 Host Identity Protocol itself

As mentioned above, the host identity protocol is only one component of the HIP framework. HIP consists of a frame format for the HIP header, several specified message types, and the HIP base exchange that is used to establish a secure HIP association between two HIP peers prior to any communication. So far, the HIP header format is only defined for HITs, not for LSIs. HIP packets start with a HIP header next to the IP header that contains e.g. the HIT of the packet

sender, the HIT of the packet receiver, and HIP parameters. So far 8 different HIP packet types are defined, as shown in Table 3-1.

Table 3-1: HIP packet types

I1	The HIP Initiator packet
R1	The HIP Responder packet
I2	The second HIP Initiator packet
R2	The second HIP Responder packet
UPDATE	The HIP Update packet, that is used to inform a peer about new IP addresses or to trigger re-keying of an ESP security association
NOTIFY	The HIP Notify packet, which is optional and may be used to provide information to a peer
CLOSE	The HIP Association Closing packet
CLOSE_ACK	The HIP Association Closing Acknowledgment packet

The first four packets are used in the HIP base exchange protocol which is used for mutual peer authentication (see section 3.2).

It is important to note that currently there is no user data transported in HIP packets. User data is transported via other mechanisms like the ESP transport format (see below). Therefore, the HIP header itself means no overhead to user data transport.

3.1.5 Interoperability with legacy networks

Although some proposals exist in individual drafts, an important aspect when evaluating HIP is that so far no HIP proxy has been specified in an IETF standard or working group draft. A HIP proxy is required to connect HIP-enabled networks with non-HIP-enabled networks. This means that it is up to now not clear how non-HIP nodes can establish communication with HIP nodes and vice versa.

3.1.6 Status

So far the IETF HIP working group has produced one informational RFC (RFC4423) [HIP Arch] that describes the HIP architecture and six working group drafts that specify:

- HIP itself [HIP Base].
- Support for mobility and multihoming [HIP MM].

- ESP transport of user data [HIP ESP].
- A rendezvous mechanism [HIP RVS].
- A DNS extension [HIP DNS].
- And a registration mechanism [HIP Reg].

Although the HIP specification is not yet standardized, there are already some HIP implementations available. The BSD implementation has been done by Ericsson Research in Finland, NomadicLab, in the "HIP for inter.net" project [HIP4Internet]. The Linux implementation has been done by both the InfraHIP project [InfraHIP] and the OpenHIP project [OpenHIP]. The windows and OSX implementations are developed by the OpenHIP project [OpenHIP].

3.2 HIP Base Exchange

The objectives of the HIP base exchange are to create a HIP association between two HIP-enabled hosts, between the *Initiator (I)* and the *Responder (R)*, prior to any data exchange between both peers. It is a two-party cryptographic protocol for creation of a shared secret via a Diffie-Hellman exchange. The message flow is given in Figure 3-1.

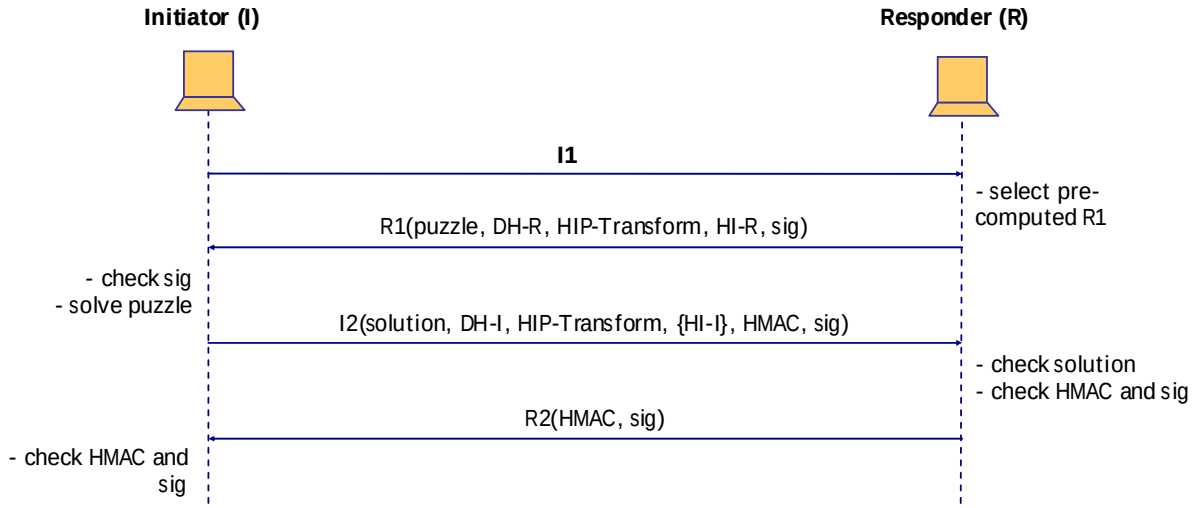

Figure 3-1: HIP base exchange

The Initiator (I) triggers the HIP base exchange by sending an I1 message to the Responder (R). In [HIP Base] it is mentioned that other triggers are possible as well but not specified so far. I1 contains the HIT of itself and the HIT of the Responder. Although the HIT of the Responder may be omitted for HIP opportunistic mode, this has security issues (e.g. vulnerability to man-in-the-middle attacks) and is not recommendable.

Upon receiving an I1 message, the Responder replies with an R1 message. Most parameters in R1 can be pre-computed, which is important to be resistant against DoS attacks. For instance, an attacker may send a high number of I1 messages to a single Responder; however, being pre-computed, creating R1 and responding means similar effort than creating I1, so an attacker has to commit similar resources.

R1 contains a puzzle that has to be solved by the Initiator, the Diffie-Hellman key of R (DH-R), the HIP-Transform parameters that signal the cryptographic algorithms supported by R, the public key (HI) of R, and the signature of the message using the private key of R corresponding to HI (some fields are set to zero to enable pre-computing).

The Initiator checks the signature by using the public key (HI-R) received via R1, solves the puzzle, and replies with an I2 message containing the solution of the puzzle, the Diffie-Hellmann key of I (DH-I), information about the cryptographic algorithms selected by I (via the HIP-Transform parameter), the public key of I (HI-I), which may be encrypted by applying the already created shared secret, an HMAC computed over the HIP packet excluding the HMAC and subsequent parameters, and a signature of the HIP packets deploying the private key of I.

R verifies the signature and the solution and, in case both checks are successful, sends an R2 message to I, containing an HMAC and a signature, which are verified by I after reception.

After this process, both peers have successfully authenticated each other and have created a shared secret that can be used afterwards for securing the communication between them.

3.3 ESP Transport

User data exchanged between two HIP-enabled peers is not transported in HIP packets but via other transport mechanisms. So far, only the ESP transport is specified and its support is mandatory in HIP compliant implementations. ESP is used similar as specified in [RFC2406]; more precisely it is stated that "The HIP ESP packet looks exactly the same as the IPsec ESP transport format packet" with reference to [ESP]. However, this draft has expired already.

During the HIP base exchange a pair of Security Associations (SAs) are created between the peer hosts (Initiator and Responder). The SAs are bound to HITs, not to IP addresses. This has the advantage that re-addressing of IP addresses (e.g. due to a mobility event) have no effect on the SAs and require no SA renegotiation. For ESP processing, the peer hosts use the keying material generated during the base exchange.

Note that for processing TCP and UDP checksums headers, the IPv6 pseudo header format is used, even in case of IPv4 addresses, and HITs are used in place of IPv6 addresses.

For outgoing packets, first the ESP header is calculated (using HITs), second, HITs are replaced by IP addresses where present, and, third, the packets are sent out. For incoming packets, first, IP addresses are replaced by HITs, afterwards ESP is processed further, e.g. integrity verification and authentication is performed, and the packet is forwarded to upper layers. For identifying the right association between IP addresses and HITs, the ESP Security Parameter Indexes (SPIs) is used.

3.4 Mobility

A key aspect of HIP for the investigation in this study is its support for mobility. As described above, one objective of the HIP framework is to facilitate mobility by decoupling IP addresses and transport sockets, i.e. transport sockets are not bound to IP addresses. As a result, a re-addressing of a network interface does not require a change of the transport layer socket. A requirement, of course, is that any peer node of the mobile node gets informed about the new IP address of the mobile node. In the HIP framework this is performed by an UPDATE message exchange.

The protocol stack of a HIP-enabled host in case the ESP transport is deployed is given in Figure 3-2.

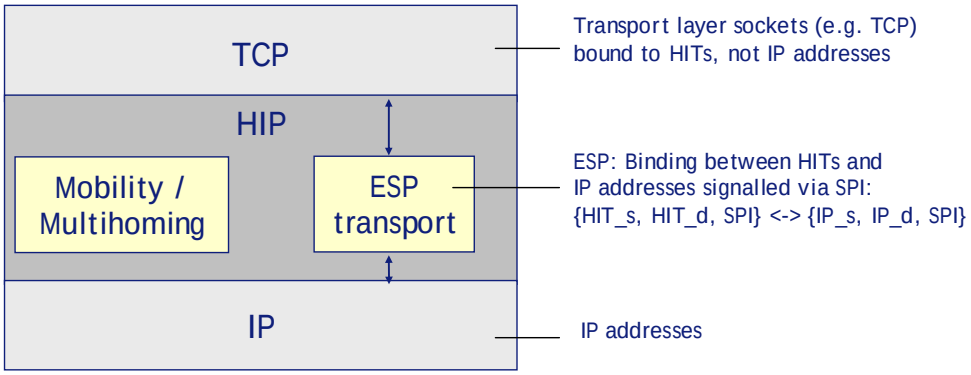

Figure 3-2: ESP transport

Transport layer sockets are bound to HITs. The IP layer, of course, uses IP addresses. Mapping between the two namespaces is performed by the HIP layer, e.g. the HIP layer keeps track of the association between source and destination HIT and source and destination IP address. The association is indexed by the SPI value. The mobility and multihoming mechanism of the HIP layer assures updating of these associations in case of a mobility event.

3.4.1 UPDATE message exchange

An UPDATE message contains a LOCATOR parameter that includes zero or more Locators. Each Locator include a traditional network addresses such as an IPv4 or IPv6 address and may include further information, e.g. ESP SPI, transport port numbers, or IPv6 Flow Labels.

The UPDATE message contains the following parameters:

- HIP message header:
    - SRC HIT: the HIT of the sender of the UPDATE message.
    - DST HIT: the receiver of the UPDATE message.
- UPDATE message specific parameters present in every UPDATE packet:
    - HMAC: HMAC computed over the HIP packet excluding the HMAC itself and any following parameters.
    - HIP_SIGNATURE: signature over the HIP packet (using the private key corresponding to the respective HI) excluding the signature itself and any subsequent parameters.
- UPDATE message parameters that are used in some UPDATE packets:
    - SEQ: gives the sequence number of UPDATE message.
    - ACK: is present in UPDATE messages that acknowledge the reception of another UPDATE message; it gives the sequence number of the UPDATE message it acknowledges.
    - ESP_INFO: beside some other parameters, the ESP_INFO field contains the old and the new SPI value of the ESP SA. In case old and new SPIs are different, this signals to perform re-keying.
    - LOCATOR: contains zero or more Locators that give information about alternate IP addresses of the sender (and some additional information like SPI, ports, etc).
    - ECHO_REQUEST/ECHO_RESPONSE: contains a nonce that is used for address reachability verification.

The address update process consists of a three way handshake as given in Figure 3-3. For the sake of simplicity, parameters that are present in every UPDATE message are omitted in the diagram.

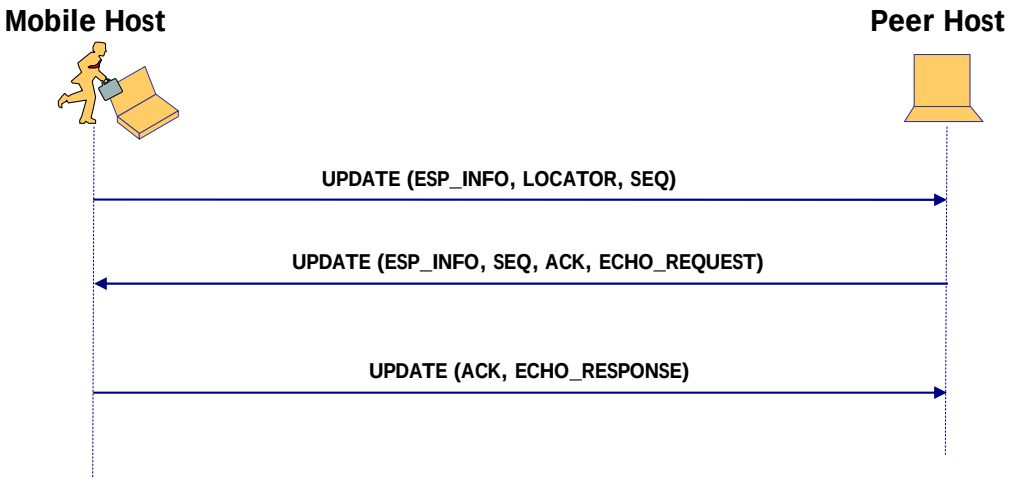

Figure 3-3: UPDATE message exchange

The update process starts in case the mobile host recognizes a mobility (or multihoming) event and sends an UPDATE message to its peer host, containing a LOCATOR parameter that informs the peer host about the current IP address(es) of the mobile node. The peer host acknowledges the reception of the UPDATE message by replying with another UPDATE message, which is once again acknowledged by the mobile node via another UPDATE message.

The parameters ECHO_REQUEST and ECHO_RESPONSE are used in the second and third UPDATE message for address reachability verification, i.e. the receiver of an UPDATE message verifies that the mobile node is really reachable under its claimed new address by comparing the values of the ECHO_REQUEST and ECHO_RESPONSE values.

Please note that all UPDATE messages of this exchange are protected via an HMAC and a signature and the receiver of an UPDATE packet can authenticate the sender and check the packet's integrity by evaluating the HMAC and signature.

After successful completion of the update process, the involved HIP peers must change local bindings between HITs and IP addresses. Furthermore, the ESP_INFO parameter in the UPDATE message can be optionally used to set new SPI values and trigger re-keying.

3.4.2 Rendezvous and registration mechanism

In order to improve reachability of frequently moving mobile HIP nodes, the HIP framework specifies a rendezvous service [HIP RVS]. A rendezvous client registers its HIT-IP address mapping via a registration mechanism [HIP Reg] at a rendezvous server (RVS). Afterwards, HIP nodes that intend to contact the mobile HIP node but do not know the current IP address can initialize the HIP base exchange by sending the I1 message to RVS, which, in case it has a registration of a node with the respective HIT in its database, relays the I1 message to the final destination, the mobile HIP node. For sending the I1 message to the RVS, the Initiator needs to

know the address of RVS. For this purpose, the HIP framework specifies a new DNS resource record (see section 3.4.3) that stores for a mobile HIP node the FQDN of its RVS (e.g. rvs.example.com). When the initiator sends a DNS query for the mobile HIP node with a QTYPE of "HIP", the DNS response should contain the HIT, the HI, and the domain name of RVS (e.g. rvs.example.com). Via a second DNS query for the IP address of the RVS, the initiator obtains the IP address of RVS. This mechanism is described and illustrated in more detail in sections 3.4.3 and 3.4.4. The HIP rendezvous mechanism also supports reachability in case both communication partners (the Initiator and the Responder) are mobile.

The registration service of the HIP framework allows HIP nodes to register with third parties like rendezvous servers or middleboxes. Although middleboxes are within scope as well, most interesting here is the registration at a rendezvous server so we will focus on this in the following. Registration could be performed either:

a) during the standard HIP base exchange process in case the Requester (here the mobile node) has not already established a HIP association with the rendezvous server, or

b) via an UPDATE message exchange in case the mobile node has already an association with RVS and just wants to update the registration.

In case a) the HIP base exchange is slightly modified. The Registrar (here the rendezvous server) sends the R1 message to the Requester (the mobile node) with an additional REG_INFO parameter, which includes Reg Type parameters that signal the service it provides (the rendezvous service has Reg Type 1). In case the Requester is interested in a specific service it appends the I2 message with a REG_REQUEST parameter. In case of a positive answer the Registrar appends the R2 message with a REG_RESPONSE parameter.

In case b) the mechanism is similar to case a), however, the parameters REG_INFO, REG_REQUEST, and REG_RESPONSE are transported in UPDATE messages.

The HIP framework so far does not specify how a mobile HIP node gets the IP address (or FQDN) of a rendezvous server, i.e. how it bootstraps mobility.

3.4.3 Domain Name Service (DNS) extension

The HIP framework proposes a new DNS resource record (RR) type that can store under a specific HIP node Domain Name:

- The Host Identity Tag (HIT) of the HIP node.
- The Host Identifier (public key) of the HIP node.
- The Domain Names of the rendezvous servers the HIP node has registered with.

This new RR is important for name resolution. It is assumed that in most cases a HIP node just knows the domain name of a peer it would like to communicate with but not its HIT. So it queries a DNS server for a resolution.

Additionally, for supporting the rendezvous service (see section above), a mobile HIP node that has registered with an RVS stores the domain name of the RVS in this new RR. In case an Initiator sends a DNS query to the DNS server with the domain name of the mobile node as parameter, the DNS server responds with the domain name of RVS. This signals the Initiator that it should send the I1 message of the HIP base exchange via RVS. RVS' HI, HIT, and IP address are obtained by the Initiator via another two DNS query/response rounds.

3.4.4 HIP initialization process

Having investigated and described the components of the HIP framework, in this section we will investigate and illustrate the HIP initialization process, i.e. the processes and protocols used and performed to establish a HIP session in case the Initiator only knows the domain name of a mobile peer.

The HIP architecture (see Figure 3-4) consists of the Initiator (I) (I may be mobile as well; however, this scenario is not described in the HIP specification so far), the Responder (R) which is mobile, a DNS server that supports the new HIP DNS RR type, and a Rendezvous Server (RVS)

Being mobile, the Responder uses the registration mechanism to register its current HIT-IP address mapping at the Rendezvous Server (RVS) (see Figure 3-4). Furthermore, R stores in a HIP resource record its HIT, HI, and domain name of RVS.

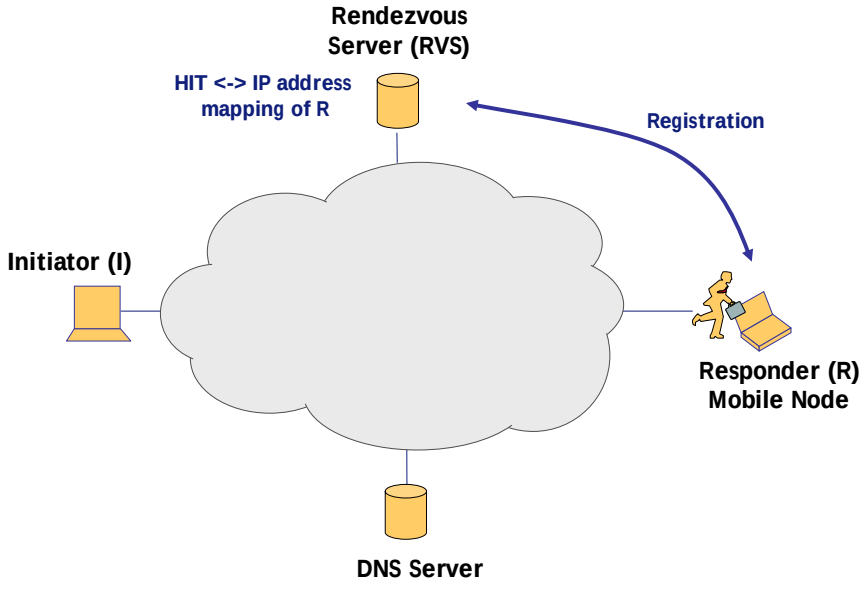

Figure 3-4: HIP architecture - registration

The HIP initialization process starts in case I would like to establish a session with R. I only knows the domain name of R (e.g. R.domain.com) so it sends a DNS query to one of its DNS servers. The query has as parameters the FQDN of R and a QTYPE=HIP that signals that the requester wants to get HI and HIT values back. Having an entry for an RVS domain name in the HIP RR (e.g. RVS.domain.com), the DNS server response contains additionally to the HIT and HI of R (HIT-R and HI-R) the FQDN of the RVS server that serves R (see Figure 3-5). Receiving this DNS response, I will query the DNS server for the IP address of RVS (IP-RVS).

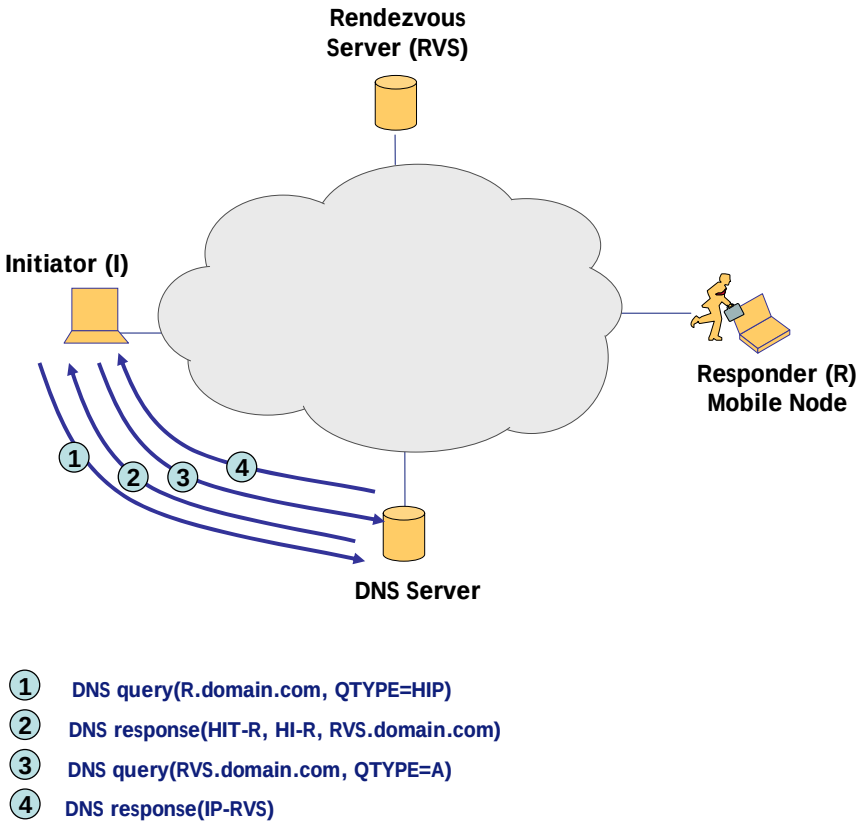

Figure 3-5: HIP architecture - DNS query

I will send afterwards the I1 message with R's HIT as Receiver HIT (HIT-R) but with RVS' IP address as destination address (IP-RVS). After reception, RVS will recognize that the Receiver HIT does not belong to himself but to a mobile HIP node that has registered; therefore RVS will replace the source IP address of the I1 message with its own (to cope with ingress filtering) and the destination IP address with R's IP address (IP-R) (see Figure 3-6). Afterwards the HIP base exchange will continue as usual without involving the RVS any more (see section 3.2), except some additional parameters for security and debugging purpose (see [HIP RVS] for details).

It has to be noted that the rendezvous server is only involved in relaying the I1 message of the HIP base exchange towards the responder HIP node. All other HIP base exchange messages and all data packets are exchanged directly between the communication peers without involving the

RVS. The RVS may also be involved in relaying the UPDATE message in order to support simultaneous movement of both HIP nodes (see section 3.4.5).

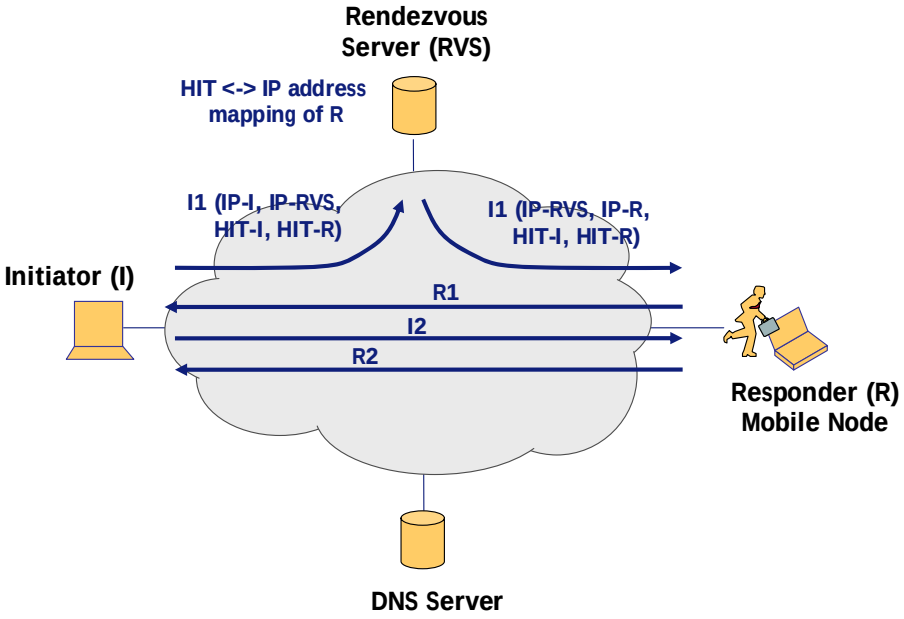

Figure 3-6: HIP base exchange via RVS

3.4.5 Simultaneous movement of both HIP peers

The document addressing "HIP Mobilility and Multihoming" [HIP MM] does not describe the simultaneous movement of both HIP nodes and it states that the "simultaneous movement of both hosts … are not covered by this document." [HIP MM] states that for "simultaneous mobility of both hosts … there is a need for some helper functionality in the network, such as a HIP Rendezvous server", which is described in [HIP RVS].

Also in [HIP RVS] the scenario where both HIP nodes are mobile is not described explicitly. Nevertheless, the RVS mechanism could be used to support this mechanism. In case its communication partner is mobile as well, the HIP node could send the respective UPDATE message to the IP address of RVS (serving the HIP peer) instead of sending it directly to the IP address of the HIP peer (as described in section 3.4.1). Upon reception of the UPDATE message, containing RVS's IP address as destination IP address but the HIT of the peer as receiver's HIT, the RVS would recognize that the destination HIT does not belong to itself but to a HIP node it serves and forward the message, similar as done with the I1 message.

The HIP node would obtain knowledge about its peer being mobile during the initialization phase when it gets a domain name of a rendezvous server in the DNS response (see section 3.4.4).

Of course, the process of both HIP peers being mobile simultaneously needs further investigation and specification in the HIP working group.

3.5 Multihoming

Multihoming refers to a situation where an end host has several parallel communication paths that it can use. It means a mobile or stationary host is having more than one interface or global address. The split between identifiers and locators introduced by HIP provides the possibility of multi-homed host having single identifier, where the multiple IP addresses are independent. There are two categories of multi-homing: end host multi-homing (the host having several network interfaces) and site multi-homing (the network where the host is located having redundant paths towards the external Internet).

3.5.1 Host multihoming

In host multihoming, the host will notify the peer host of the additional interface(s) or address(es) by using the LOCATOR parameter. When more than one locator is provided to the peer host, the host will indicate which locator is preferred. Figure 3-7 illustrates the host multihoming scenario, where HIP host1 has two independent IP addresses.

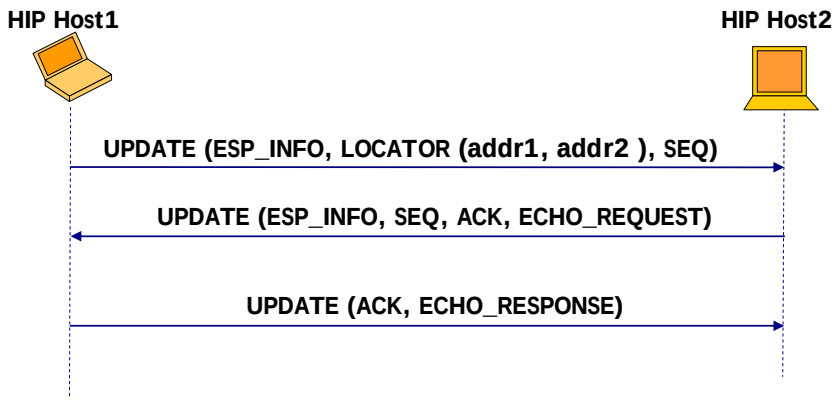

Figure 3-7: Host multihoming illustration

3.5.2 Site multihoming

A host may have an interface that has multiple globally reachable IP addresses. Such a situation may be a result of the site having multiple upper ISPs or just because the site provides all hosts with both IPv4 and IPv6 addresses. Site multihoming makes the host stay reachable with all or any subset of the currently available globally routable addresses, independent on how they are provided. The site multihoming is shown in Figure 3-8.

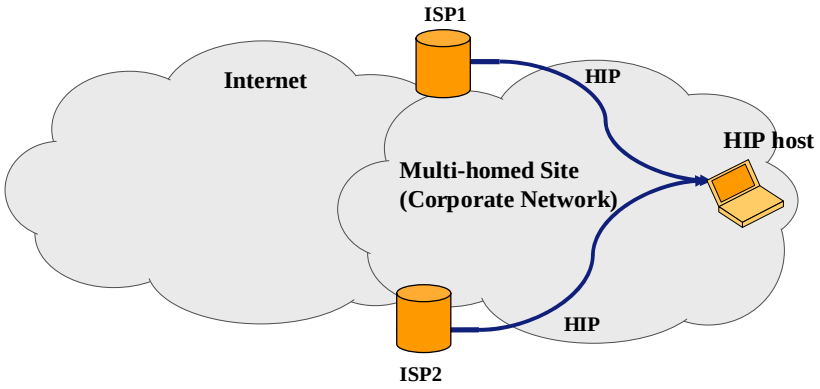

Figure 3-8: Site multihoming

3.6 HIP Assessment

3.6.1 Functional assessment

- **Support for simultaneous movement of both endpoints**

  As explained in section 3.4, HIP uses DNS and RVS server to support movement of end hosts. Mobile HIP hosts register at an RVS server which contains current HI – IP address mapping. In the case where the current HI – IP address mapping in unknown, a communication partner sends the I1 message of the HIP base exchange via the RVS server, which forwards the message to the right receiver. The receiver learns from the I1 message the HI of the sender and its current IP address via a FROM parameter. The receiver responds with an R1 message containing its current IP address. Therefore, in case both HIP peers are mobile, both can commence communication.

  As explained in section 3.4.5, the rendezvous mechanism could be used to provide mobility management after session setup has been completed. It has to be noted, however, that the HIP reference documents [HIP MM] and [HIP RVS] do not specify this scenario so far.

- **Support for simultaneous use of multiple interfaces (multihoming)**

  The split of identifier and locator in HIP provides natural support for end host multi-homing. More than one IP address can be bound to a single HI. It has to be noted, however, that the specification [HIP MM] currently does not specify detailed policies and procedures for multihoming, i.e. it is stated: "Although this document defines a basic mechanism for multihoming, it does not define detailed policies and procedures such as which locators to choose when more than one pair is available, the operation of simultaneous mobility and multihoming, source address selection policies (beyond those

specified in [RFC3484]), and the implications of multihoming on transport protocols and ESP anti-replay windows.".

- **Support for flexible placement of service elements**

  The rendezvous point in HIP is the RVS server where a mobile HIP host registers and updates its HI - IP address mapping while publishing the IP address of the RVS via a new DNS resource record. The RVS can be placed anywhere so this criterion can be classified as **Any**.

- **Robustness level and failover support**

  Currently failover mechanisms for the rendezvous server (RVS) are not specified which means that in case the RVS fails all mobile HIP nodes that have registered at this RVS are no longer reachable by HIP nodes that do not know the current HI – IP address mapping. Since HIP establishes a direct route between the HIP peers, such a failure has no impact on existing connections. A failover solution would be that a mobile HIP node registers at more than one RVS and also publishes more than one RVS records in the DNS resource record. For opportunistic HIP mode, where the Initiator sends an opportunistic I1 with a NULL destination HIT in case the Initiator does not know the HIT of the responder, the RVS server will not be necessary. In this way, the failure of RVS has no effect to opportunistic HIP mode. However, after all the robustness level and failover support can be classified as **Partial**.

- **Scalability**

  So far there are no experiences about the scalability of HIP. In case we just consider the scalability of the mobility management in HIP, it is **High** scalable since only the initial packet of a session establishment (I1) is routed via the RVS. Of course, movement has to be slower than the speed that a mobile HIP host needs to update all communication peers. The update mechanism consists of an exchange of three messages and its duration is therefore 1.5 times the round trip time plus processing time in each HIP peer; however, updating of several peers can be done simultaneously.

3.6.2 Deployment assessment

- **Transparency to legacy applications**

  HIP introduces a new host identity layer between network layer and transport layer and a new Host Identity namespace. In HIP, transport layer sockets are bound to host identities instead of IP addresses. Thus HIP does not provide compatibility for legacy applications.

- **Support for legacy hosts**

  In HIP both communication partners have to be HIP-enabled. HIP currently does not support communication between HIP hosts and legacy hosts. Potentially HIP proxies can be used to enable HIP between legacy hosts and HIP hosts and already some proposals exist in individual drafts; however, this needs further standardization effort and therefore we have to state that currently there is no support for legacy hosts (**None**).

- **Deployment effort**

  DNS and RVS servers are necessary for HIP deployment and communication with a AAA infrastructure for authorizing the rendezvous service, for example. Furthermore, a new DNS resource record has to be introduced in DNS servers that are used for HIP mobility support. The major deployment effort, however, is to deploy HIP itself which means that nodes have to be prepared for HIP communication. Although HIP opportunistic mode does not require DNS and RVS deployment, taking into account the installation of both DNS and RVS for a normal HIP communication, the deployment effort can be marked **High**.

- **Operational effort**

  The entities to be monitored are the rendezvous servers and the AAA servers. Reconfiguration of entities and re-keying is done automatically, e.g. reconfiguration of the HI – IP address mapping in the RVS. Hence, the operational effort of using HIP is similar to that of MIPv6 and this criterion is marked **Medium**.

- **Need to deploy new security infrastructure**

  HIP is designed to provide secure authentication of hosts. In HIP, ESP Security Associations are setup between the HIP nodes during the Base Exchange. HIP is based on a public/private key mechanism since a Host Identifier is the public keys of a public/private key pair. Therefore HIP can be based on PKI infrastructures. Since every HIP node needs a public/private key pair, the performance of existing PKI infrastructures may not be enough. Therefore, there is a **Partial need** to deploy a new security infrastructure.

- **Maturity**

  HIP has not been deployed yet. However several implementations are available already, e.g. "HIP for inter.net" project [HIP4Internet], OpenHIP [OpenHIP], and .InfraHIP [InfraHIP]. The maturity of HIP is therefore **Medium**.

3.6.3 Security assessment

- **DoS resistance**

  The HIP base exchange has built-in DoS resistance. The R1 message can be mainly pre-computed and the initiator of a session has to commit resources by solving a puzzle contained in R1 before a session is established successfully. Since any communication setup involves the HIP base exchange, DoS resistance is always given. Of course, if an attacker floods the network link that connects the RVS towards the Internet, no new sessions with a mobile node that the RVS is serving can be established. However, no current sessions are affected and mobile HIP nodes could have registered by multiple RVSs. Therefore, this criterion can be marked as **High**.

- **Support for location privacy**

  HIP uses a global identifier - the Host Identifier. Furthermore, a HIP node has to inform its peers about its locators. Therefore, location privacy is not given.

3.6.4 Performance assessment

- **Support for packet loss minimization**

  HIP provides locator and identifier split, thus provides seamless mobility. Because the IP addresses are decoupled from the transport layer sockets, the applications are not interrupted. HIP also supports multihoming. In case a HIP node attaches to a new point of attachment while the connection to the old point of attachment is still alive, it can still use the old locator while informing its peers about a new preferred locator. In case a HIP node attaches to a new point of attachment but the connection to the old point of attachment is lost, packets sent by the HIP peer towards the mobile HIP node are lost until the update process is completed. This situation is similar to MIPv6 and this criterion is marked as **Low.**

- **Support for routing optimization**

  HIP is a host-based end to end approach and inherently has direct routing mechanisms. Already during the HIP base exchange a direct route between the HIP peers is established and application data is routed directly to the peer node right from the beginning. In this way, besides in-network IP routing optimizations, additional routing optimization for HIP is not required.

- **Support for signaling optimization**

  When moving, a HIP node has to inform all its communication peers and rendezvous servers. There is no support for signaling optimization hence the qualifier is **None**.

3.6.5 Additional properties

- **Spanning different addressing realms**

  HIP sessions can span different address realms transparent to the applications. Addressing realms could be domains with local and global scope addresses, IPv4 and IPv6 domains, or even new network address domains that may appear in the future

3.7 Conclusions

HIP is clearly a disruptive technology with the objective to change the way nodes in the Internet are communicating. HIP introduces a new namespace, the Host Identity namespace, and applications are bound to Host Identifiers that are used to identify each node in the Internet uniquely while removing the identifier role from IP addresses. Besides other features, the HIP framework provides mechanisms to support mobility of HIP nodes, e.g. a rendezvous mechanism, a new DNS resource record, and mobility and multihoming management mechanisms. The session initialization sequence of a communication between two HIP peers in case one of them is mobile has been illustrated above.

An advantage of HIP is the separation of locator and identifier that enables to bind applications to a fixed identifier while the locator (the IP address) may change, which supports for mobility, multihoming, and sessions across addressing realms. Data communication is routed directly between the communication peers right from the beginning and not routed via a mobility anchor point as in MIPv6, for example. Furthermore, HIP has built-in security, e.g. DoS resistance and mutual authentication prior to any communication.

A clear drawback of HIP is its disruptive nature that requires modifications to applications for deployment. This makes lots of legacy applications unusable in a HIP environment, which also means the cost of fully deploying HIP is high. Moreover, although some implementations exist already, standardization is not mature yet and requires further time and effort. Some problems are not solved at all so far, e.g. the connection of HIP networks to non-HIP networks, which is required for a smooth transition. Because of these deficiencies it is not clear when and whether HIP will be deployed and whether it could replace MIPv6.

In conclusion, HIP is clearly a disruptive technology and is not meant to complement other mobility management solutions like MIPv6 but to replace them. It is an interesting and beneficial technology for the future but is not ready to be deployed in short term.

4. I3 AND RELATED APPROACHES

This section describes and evaluates the Internet Indirection Infrastructure (*i3*) [I3] and FARA [FARA]. Since both technologies are based on the same approach, only *i3*, the most mature and promising of the two of them, will be fully evaluated and assessed against the evaluation criteria. FARA will be described and briefly evaluated for the sake of completeness of this deliverable.

4.1 Internet Indirection Infrastructure

4.1.1 Overview

i3 proposes a single overlay network that serves as a general-purpose indirection infrastructure for the Internet, offering a rendezvous-based communication abstraction. *i3* decouples the act of sending a packet from the act of receiving the packet, so instead of explicitly sending a packet to a destination, each packet is associated with an identifier, which is then used by the receiver to obtain delivery of the packet. The delivery is best-effort, so no guarantees are made about packet delivery.

Figure 4-1 illustrates how two nodes communicate within the *i3* service model. When host *R* wants to receive packets sent to the identifier *id*, it inserts a **trigger** (*id, R*) in the *i3* infrastructure. Triggers are pairs *(id, addr)*, where *id* is the identifier of the trigger, and *addr* is the IP address and port number of a node. The trigger (*id, R*) indicates that all packets with an identifier *id* should be forwarded by the *i3* infrastructure to the node with the address *R*.

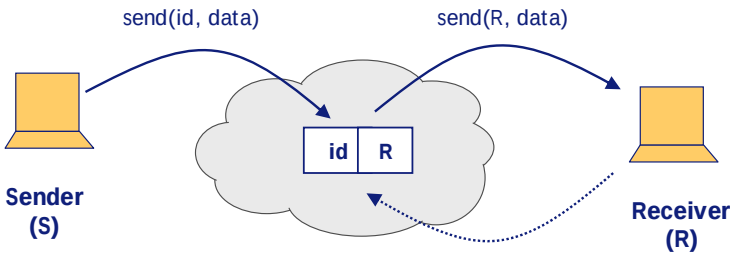

Figure 4-1: Communication between two nodes

The identifier *id* represents a logical rendezvous between the sender's packet and the receiver's trigger. This level of indirection decouples the act of sending from the act of receiving, and allows *i3* to efficiently achieve the more general communication abstractions of mobility, multicast and anycast:

1. **Mobility**. *i3* provides natural support for mobility. Figure 4-2 illustrates how a mobile host *R* that moves from one subnet to another, thus changing its address from *R1* to *R2*, can update its trigger from (*id*, *R1*) to (*id*, *R2*) to preserve end-to-end connectivity with the sender *S* in a completely transparent way. *i3* can maintain the connectivity even if both *S* and *R* move simultaneously.

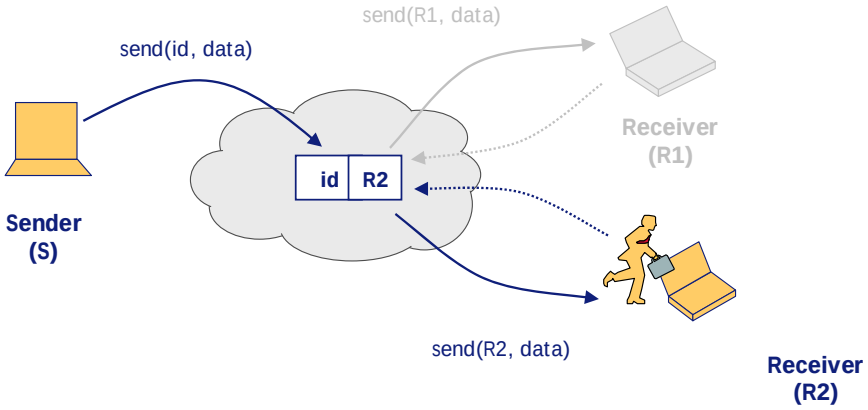

Figure 4-2: Mobility in *i3*

2. ***Multicast***. A multicast tree can be built by using a hierarchy of triggers, where each member R_i of a multicast group id_g replaces its trigger (id_g, R_i) by a chain of triggers (id_g, x_1), (x_1, x_2), …, (x_i, R_i). This is completely transparent to the sender, and a packet (id_g, *data*) will still reach R_i via the triggers chain. An example of an *i3* multicast tree is illustrated in Figure 4-3. The number of triggers with the same identifier represents the replication factor of a packet at an infrastructure node. Note than in *i3* there is no difference between unicast or multicast packets, so an application can switch between unicast and multicast by having hosts insert triggers with the same identifier.

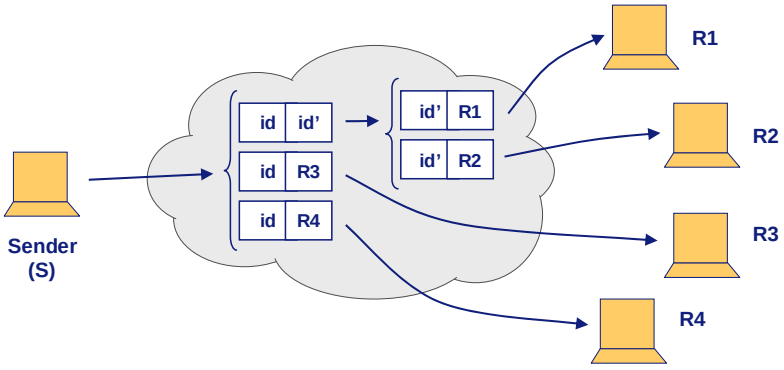

Figure 4-3: Multicast tree

3. ***Anycast***. *i3* provides support for anycast by allowing inexact matching between identifiers. The matching is done according to the longest prefix rule. In Figure 4-4,

receivers *R1*, *R2*, *R3* insert triggers whose identifiers share a common prefix *p*. A packet with an identifier *p** is matched according to the longest prefix rule and forwarded to the corresponding receiver *R1*. Multiple anycast policies can be implemented by choosing different suffixes. For example, load balancing can be provided by randomly choosing the suffixes of all triggers and request packets.

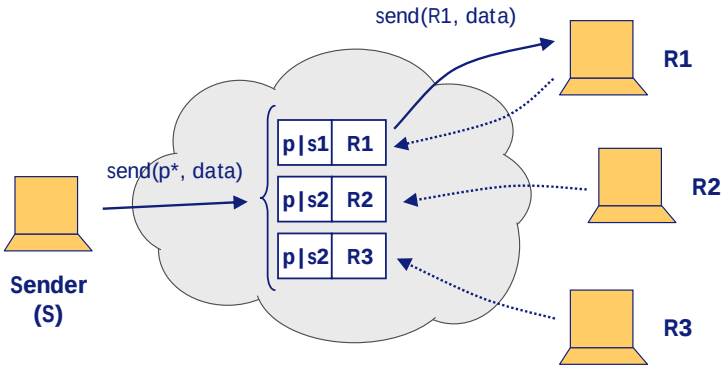

Figure 4-4: Anycast n *i3*

i3 allows applications to replace an identifier with an identifier stack, which is a list of identifiers (id_1, id_2, …, id_k). This generalized form allows a source to send packets to a series of identifiers, allowing for great flexibility and enabling service composition, heterogeneous multicast and increased robustness of the infrastructure. Figure 4-5 shows an example of how a sender *S* can request that all its packets are forwarded through a transcoder *T* before they are delivered to their destination *R*. The transcoder inserts a trigger id_T, while sender *S* sends packets with the stack (id_T, *id)*, so each packet is forwarded according to the first identifier id_T to the transcoder, processed and then forwarded to *R*.

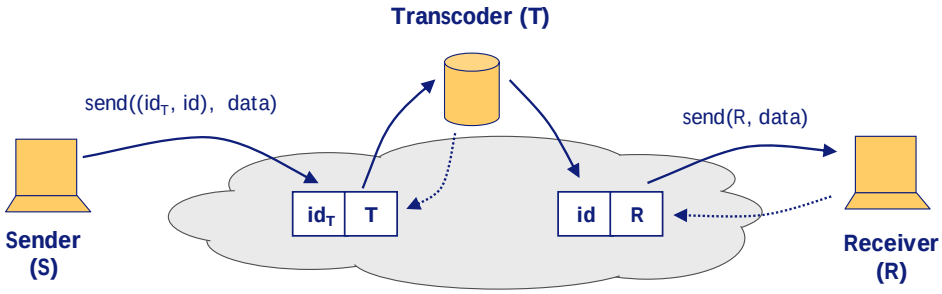

Figure 4-5: Forwarding packets through a transcoder

i3 is organized as an overlay network in which every node stores a subset of triggers. Each trigger is stored at only one server. Each host knows about at least one server, so when it wants to send a packet it forwards the packet to one of the servers it knows. If the server does not store the trigger matching that packet, it forwards the packet via IP to another server and so on, until the packet reaches the server that stores the matching trigger. The packet is then sent via IP to its

destination. *i3* is currently implemented on top of the Chord lookup protocol [CHORD], which according to [I3] satisfies the properties of *robustness*, *efficiency* and *stability* required by the overlay network, that will determine *i3*'s performance. Other lookup protocols could be used, as long as they satisfy the aforementioned properties. We will now look into how *i3* fulfils these requirements, and then we will take a look at some actual numbers from simulations to evaluate *i3*'s performance:

1. ***Robustness***. Routing of packets within *i3* is fairly robust against node failures, and the soft-state approach allows for a simple and efficient implementation, freeing the infrastructure from having to recover lost state when nodes fail. When a server goes down all of its triggers are lost, and the hosts will have to reinsert them at another server after the next refresh. This is a potential problem because for some applications the latency between refresh times will be too large and the triggers will be unavailable for a large amount of time. A possible solution is to maintain backup triggers in addition to the primary triggers; another solution is using the replication policy of Chord (or the equivalent on other lookup protocols) to replicate the triggers and manage the replicas.

2. ***Efficiency.*** Routing in an overlay network is less efficient than routing the packet directly via IP. An approach that can mitigate this problem is to make the sender cache the *i3* server's IP address. This can be accomplished by using a flag in every packet that is activated by the sender to indicate the server that it must return its IP address back to the original sender. This address is cached by the sender and used to send subsequent packets with the same identifier. The robustness of the system is not undermined by this optimization: if the trigger moves to another server, *i3* will route the packets from the server where the trigger was originally located to the new location of the trigger, and the cache of the sender will be updated as soon as the first packet is received at the new location.

   Triangular routing is a problem that is still not solved by caching the server storing triggers. The routing can be very inefficient if the sender and receiver are very close but when the server storing the trigger is located far away from both. A possible solution is to have the receivers choose their private triggers by sampling the identifier space to find ranges of identifiers that are stored at nearby servers.

3. ***Scalability.*** The number of triggers stored in *i3* is of the order of the number or flows plus the number of end hosts (note that each flow requires two triggers, one for each end-point). Since each trigger is stored at only one node at a time, if there are *n* triggers and *N* servers, each server will store an average of *n/N* triggers. *i3* can be upgraded by adding more servers to the network, in no specific location.

A balance should be found between these properties, since some of them appear to be in conflict with each other. For example, maintaining backups triggers my increase robustness and stability, but will probably hurt efficiency. [I3] does not address the issue of which criteria are most important and which is less.

4.1.2 Robust Overlay Architecture for Mobility

In [ROAM], the *i3* authors introduce the Robust Overlay Architecture for Mobility (ROAM), a solution built on top of *i3* that has been designed to provide mobility support for TCP-based legacy applications. Support for host mobility is already a part of *i3*, since a mobile host that changes its address can preserve end-to-end connectivity by simply updating its triggers. ROAM, however, is designed to improve mobility in *i3* by featuring the following properties:

- **Efficient routing**

  ROAM presents a new technique that extends the already existing solutions to improve routing efficiency in *i3* (i.e., trigger server caching and trigger sampling). This extension takes into account the movement pattern of mobile hosts, which consists of many short movements and occasional movements to networks located very far away, and is called *Mobility-Aware Trigger Catching*. This technique involves caching sampled triggers and creating diversity in the cache, so that a trigger in the cache is near each of the remote locations that a mobile node (occasionally) visits, while preventing the more frequent local movements from polluting the cache.

  According to [ROAM], this solution greatly reduces routing latency: in an *i3* system with 2^{16} servers, taking 32 samples result in a 90^{th} percentile latency stretch (the ratio of the latency of the optimized route to the latency of the shortest IP path) of 1.5.

- **Efficient handoff**

  ROAM supports fast handoff by making end-hosts choose triggers that map onto nearby *i3* severs. Additionally, another solution proposed by [ROAM] to reduce the loss of packets during handoffs is the use of multicast-based soft handoff: when a MN moves to another network, the MN inserts a trigger with the same identifier as its existing trigger, but associated with the new address. This causes the same packet to be delivered to both the old and new addresses, thus allowing the MN to take advantage of the best available connectivity.

- **Fault tolerance**

  ROAM can recover gracefully from server failure thanks to the fact that triggers are periodically refreshed. Simulations in [ROAM] show that when nodes have a 15%

change of failing, MIP likelihood of successful connectivity drops to only 50%, while ROAM's likelihood of successful connectivity is more than 70%.

- **Simultaneous mobility**

  Both the sender and the receiver can move simultaneously in ROAM, since the *i3* overlay network servers as an anchor point for the two sides of the communication channel.

- **Location privacy**

  ROAM allows end-hosts to choose triggers in order to hide their location. Although this can have an impact in routing and handoff, *i3* is flexible enough to allow end-hosts to make the tradeoffs between location privacy and routing efficiency. For instance, a MN can choose an *id* so that the trigger is stored at an *i3* server close to the CN instead of itself, which would reduce the latency while preserving the privacy of the MN. Preserving fast handoff is also possible if the MN uses two triggers, one inserted near the CN of the form (*id*, *id'*) and one inserted close to itself of the form (*id'*, *MN*), as illustrated in Figure 4-6. This choice ensures low latency and enables the MN to do fast handoff by simply updating the trigger (*id'*, *MN*). In the case where both end-points require location privacy, they can always choose random *i3* servers.

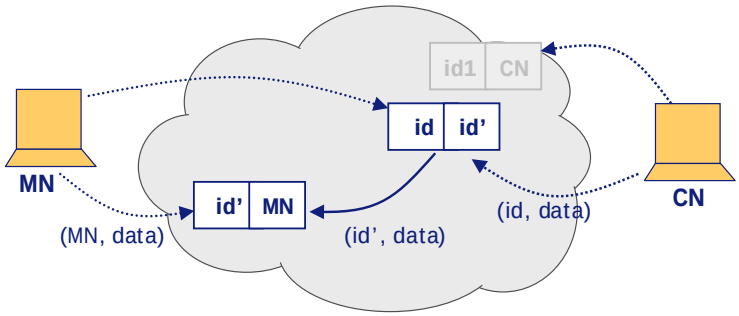

Figure 4-6: Location privacy

- **Personal/session mobility**

  ROAM allows a user to redirect a new session or migrate an active one from one application or device to another when a better choice becomes available.

Simulation results and a deeper analysis of these properties can be found in [ROAM]. While the detailed results will not be reproduced in this deliverable, the technology assessment takes all results into account when evaluating each criterion.

4.1.3 Secure-*i3*

i3's security problems are described and analyzed by the authors of Secure-*i3* [SI3]. This work proposes a re-design of *i3* in order to protect against DoS attacks better than the current Internet, by following three simple design principles:

1. The overlay architecture should enable end-hosts to communicate without revealing their IP addresses, thus leaving an attacker with no direct way to attack a host.

2. The infrastructure should give end-hosts the ability to defend against attacks, by possibly stopping the attack in the infrastructure. This would dramatically improve the current situation of the Internet, where an end-host can't do almost anything to defend from flooding attacks directed to its IP address.

3. Make sure that the new infrastructure does not introduce new security holes that are not present in the Internet.

The following subsections describe the proposed solutions to the three design principles.

4.1.3.1 Hiding IP addresses

An overlay communications infrastructure should enable end-hosts to communicate without revealing their IP addresses in order to provide protection against flooding attacks. In *i3*, an attacker can learn the IP address of the node that stores the public trigger of a node by simply inserting a trigger pointing to itself with the same prefix as the victim's public trigger. When the attacker sends a packet to that node, the packet will be sent back to him via IP, revealing the source IP of the node. Figure 4-7 illustrates this attack. Since in *i3* triggers with the same prefix are stored at the same *i3* node, the attacker can then use an IP flooding attack to attack the *i3* node that stores the victim's ID.

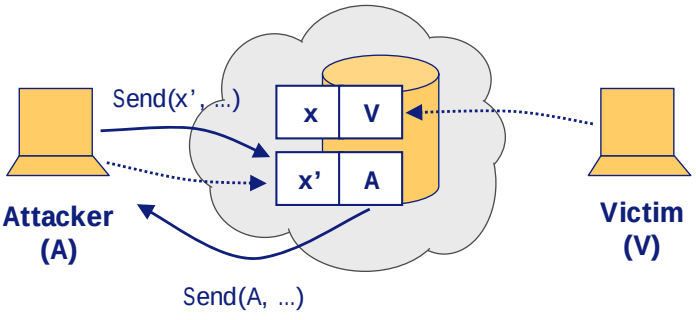

Figure 4-7: The attacker reveals the victim's IP address

A possible solution to this problem is to make sure that nodes that store public IDs will never allow triggers pointing to end-hosts. Instead, these nodes will store triggers that point to private

IDs, which will then point to the actual end-host. In this way, the attacker can learn the IP address of a node storing private triggers, but will not be able to find out where the private triggers are located since they are secret.

4.1.3.2 Defending against attacks

Now that attacks at the IP level have been addressed, the overlay infrastructure must provide mechanisms to avoid attacks at the *i3* level, which is still possible. [SI3] states that the best defense is giving end-hosts the ability to defend themselves against the attacks. This section explain how end-hosts can stop attacks on private triggers, dilute or slow down attacks on public triggers, evade attacks on *i3* servers and provide multicast access control:

1. ***Stopping attacks on private triggers***. In current IP networks, defending against flooding attacks is theoretically possible but requires and increased network complexity and time to detect and create defenses for the attack. In *i3*, however, end-host can simply stop the attack by removing the trigger where the attacking traffic is coming from. Since each receiver communicates with different senders by private triggers that are kept secret by the senders, removing that trigger would only stop traffic form the attacker, without affecting legitimate traffic. Figure 4-8 shows how the victim can drop the private (x_3, V) and (x_4, V) triggers in order to reduce by half the traffic sent by the attacker.

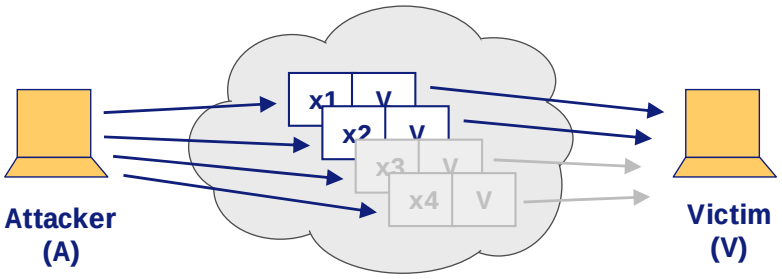

Figure 4-8: Defending against attacks on private triggers

   This defense would not work well with public triggers, since dropping a public trigger would stop the attacker but also make the server unreachable for new clients. Current clients would still be able to maintain the communication with the server via their private triggers.

2. ***Diluting attacks***. The proposed solution to alleviate the problem of flood attacks to public triggers is giving the server the ability to drop a fraction of the traffic destined to public triggers. Although this would also hurt legitimate users, at least it gives the server the chance to continue providing the service. Since the attacked triggers are probably the ones receiving a disproportionate amount of traffic, the server should gradually drop the

triggers that receive the most traffic. This solution in the IP world would require the router of the ISP to implement packet classification and bandwidth management.

3. ***Slowing down attacks***. A possible solution to slow down attacks on public triggers is using a DoS-filter server that protects the original server form the attack. This server must be powerful enough to sustain the DoS attack, and implement a defense based on cryptographic puzzles. Clients must solve these puzzles before contacting the defended server, thus slowing down the rate of the flooding attack. In order to set up this defense, the original server stores a private trigger pointing to itself with an identity only known by the DoS-filter server. In turn, the DoS-filter server stores a public trigger advertising the public identity of the original server, so all packets delivered to the original server are now delivered to the filter server, which then sends a cryptographic puzzle to the client. If the client replies with a correct solution to the puzzle, the DoS-filter server forwards the packet to the new private identity of the original server, which then allocates a private trigger for the client. Of course, servers would only use this defense when under attack, so under normal operation clients will not have the burden of solving the puzzles.

4. ***Evading attacks***. When the attack is directed to a particular *i3* node, the end-hosts can use a different trigger to find an alternative route that does not involve the attacked node. Note that, if a router is attacked in the Internet, the end-host does not have the capacity to route around the attack.

5. ***Multicast access control***. Secure-*i3* addresses the inherent insecurity of IP multicast, where any receiver can potentially send multicast traffic to an entire multicast group, since only one multicast address is used by both senders (to send traffic) and receivers (to subscribe to the group). In Secure-*i3* this problem can be avoided by having different IDs for senders and receivers of the multicast group, delegating the management of the multicast group to a third party. The manager would construct a tree of triggers where each sender would be provided with a private id to send traffic to the group, and receivers would only be able to receive traffic, unaware of the tree topology.

   The proposed solution to multicast access control is non-cryptographic, and according to [SI3] it allows simple member addition and deletion, and achieves forward and backward secrecy (assuming that eavesdropping within the Secure-*i3* infrastructure is hard).

4.1.3.3 Avoiding new vulnerabilities

[SI3] also analyzes new potential vulnerabilities that the *i3* infrastructure introduces that are not present in the Internet, and propose countermeasures to all of them. A brief overview of the new vulnerabilities is provided in the following lines:

1. **Attacks using triggers pointing to end-hosts**. In an eavesdropping attack, the attacker sets the legitimate trigger's destination address of an end-host to its own address, effectively eavesdropping all traffic to that end-host. This is much easier than eavesdropping in the Internet. A variation of this attack is making the end-host to drop its public trigger by flooding it, and then inserting a new trigger to not only eavesdrop on the host's traffic, but also responding to it. This problem is inherent in *i3*'s architecture. The attacker could also abuse the end-host by inserting a new trigger to sign-up the host for high-bandwidth streams

2. **Attacks using triggers pointing to IDs**. The attacker could use triggers pointing to ids to construct topologies to multiply the attack traffic and direct it to the end host. Examples of such topologies are loops, where packets sent to any of the IDs of the loop would indefinitely cycle around consuming resources; confluence, where packets are first replicated as they would be in a multicast tree, and then delivered to the same host via its public trigger; and dead-ends, where an attacker constructs a chain of triggers which ultimately does not point to a valid end-host, thus making packets routed trough that dead-end topology consume network resources. Also, by inserting arbitrary triggers, the attacker can channel traffic between different applications.

3. **Attacks by sending data to arbitrary topologies.** An attacker can build a large multicast tree and store all its leaf triggers at the target *i3* node, so that for every packet sent by the attacker to the group the target *i3* node would be flooded with a number of duplicates equal to the number of leaf triggers of the tree.

[SI3] proposes a solution based on enforcing constraints on the structure of triggers and constraints on trigger insertion. For this, the 256-bit identifier in Secure-*i3* is divided into three fields: a 64-bit prefix, a 128-bit key, and a 64-bit suffix, and only triggers of the form (x, y) where $x.key = h_1(y)$ or $y.key = h_r(x)$ are allowed, where h_l and h_r are one-way cryptographic hash functions. If $y.key = h_r(x)$ the trigger is right-constrained, otherwise it is left-constrained.

If a trigger points to an end-host, the fields *y.prefix* and *y.suffix* are used to encode the end-host address. Additionally, if a constrained trigger points to an end-host, only *y.key* is used to constrain *x.key*, ignoring *y.prefix* and *y.suffix* when computing $h_1(y)$ to preserve support for anycast and mobility. Publicnodes allow only constrained triggers, which avoids eavesdropping and impersonation of end-hosts. It also does not allow attackers to attack public nodes by leveraging multicast functionality. In addition, public nodes do not allow triggers pointing to end-hosts.

To avoid eavesdropping and impersonation, an end-host inserts only left-constrained triggers. The fields *y.prefix* and *y.suffix* are used to store the address, and *y.key* constains a key that is

known only by the end-host, thus making impossible for an attacker to insert a trigger pointing to itself in order to eavesdrop the traffic.

4.1.4 i3 assessment

4.1.4.1 Functional assessment

- **Support for simultaneous movement of both endpoints**

  i3 offers a rendezvous-based communication abstraction, decoupling the act of sending a packet from the act of receiving the packet, so it provides natural support for simultaneous movement of both endpoints. When a MN moves from a subnet to another, it updates its trigger to preserve end-to-end connectivity with the sender in a completely transparent way. *i3* can maintain the connectivity even if both the sender and the receiver move simultaneously, acting as an anchor point for the two sides.

- **Support for simultaneous use of multiple interfaces (multihoming)**

  The indirection approach of *i3* provides a straightforward implementation for multi-address multihoming.

- **Support for flexible placement of service elements**

  Since is rendezvous-based, the *i3* architecture itself serves as an anchor point for the two sides of the communication channel. There's no need to deploy mobility-specific anchor points in the network, but since the triggers can be placed in any server of the overlay network, the relativity modifier for this criterion is **Any**.

- **Robustness level and failover support**

  Routing of packets within *i3* is fairly robust against node failures, and the soft-state approach allows for a simple and efficient implementation, freeing the infrastructure from having to recover lost state when nodes fail. When a server goes down all of its triggers are lost, and the hosts will have to reinsert them at another server after the next refresh. This is a potential problem because for some applications the latency between refresh times will be too large and the triggers will be unavailable for too long. A possible solution is to maintain backup triggers in addition to the primary triggers; another solution is using the replication policy of Chord (or the equivalent on other lookup protocols) to replicate the triggers and manage the replicas. Simulation results presented in [ROAM] show significant improvement in fault tolerance over MIPv6.

  The relativity modifier is **Full**, since the overlay network is robust enough by design to give mobility service to the MN in the case of node failures.

- **Scalability**

  Support for mobility in *i3* comes in a natural way, without the need of an additional infrastructure, so the scalability of the mobility solution depends on the general scalability of *i3*. According to [I3], since a trigger is stored at only one node at a time, the overlay network can be easily upgraded by simply adding more servers to the network, in no specific locations of the network. However, depending on the peer to peer protocol employed, the time needed to recover a trigger may increase as the overlay network grows, with the effect of compromising the usability of the solution, so the proposed relativity modifier for this criterion is **Medium**.

4.1.4.2 Deployment assessment

- **Transparency to legacy applications**

  The current distribution of *i3* includes a proxy-based solution that allows unmodified applications to run on top o *i3*. This solution is called OCALA proxy [OCALA] and is implemented as a proxy that captures IP packets sent/received by legacy applications and relays them over the *i3* overlay network. The implementation currently works on Windows 2000/XP, Linux and Mac OS X, requiring no changes to the operating system.

  In this solution, legacy applications identify overlay end-hosts using DNS-like names. These names consist of a suffix that specifies the overlay type, a middle part that specifies the overlay instance, and a prefix which specifies the overlay specific-name. Different OCALA modules process the name and resolve it to a specific *i3* identifier/address.

  In addition, [ROAM] also introduces a similar concept, the user-level ROAM proxy, that encapsulates and decapsulates IP packets within *i3* packets, determines the triggers of remote hosts, and sends the local private triggers to remote hosts.

- **Support for legacy hosts**

  In both [OCALA] and [ROAM] it's clear that with the use of the proxies no changes to the legacy applications, operating systems, NATs or firewalls are required. However, these proxies must run in both the home and the remote machine in order to support mobility, so hosts must be modified in order to communicate using the overlay network. The relativity modifier for this criterion is **None**.

- **Deployment effort**

  Deployment in end-hosts is difficult because each host runs a ROAM/OCALA proxy. While this provides significant robustness and efficiency, the management and

development costs will be too high, so it's clear that deployment of the proxies must be gradual. [ROAM] suggests the possible alternative of deploying a home proxy for a MN that implements the functionality of a ROAM proxy for all its non-ROAM CNs, acting in an analogous way to a HA in MIPv6.

As for the overlay infrastructure, it could start as a single server provided by a third party, but it is not clear how to involve ISPs for an efficient and robust deployment. An advantage of *i3*, as stated in [I3], is that it is incrementally deployable, as it is designed as an overlay network. Adding more servers to the system (which could initially consist of a single server that stores all the triggers) does not require any configuration other that joining the Chord protocol, which automatically makes the new node become responsible for a subset of the identifier space.

The proposed relativity modifier for this criterion is **High**, as complete support of *i3* implies deploying a proxy in every end host.

- **Operational effort**

The efforts of monitoring the *i3* infrastructure will be focused on the monitoring of the overlay network, which is based on the Chord lookup protocol [CHORD]. This should not be a high effort thanks to the characteristics of the overlay network, which is peer-to-peer and has very high fault tolerance. As a consequence of this, the relativity modifier for this criterion is **Medium**.

- **Need to deploy new security infrastructure**

At least a partial deployment of new security infrastructure components would be needed in order to implement all the countermeasures to DoS attacks and security improvements described in [SI3], so the relativity modifier for this criterion is **Partial need**.

- **Maturity**

Prototype implementations of *i3* (using the Chord protocol) and the user-level ROAM exist, although they haven't been widely tested and/or deployed. Therefore, the proposed relativity modifier for this criterion is L**ow**.

4.1.4.3 Security assessment

- **DoS resistance**

[SI3] has a very good analysis of potential DoS attacks and countermeasures to prevent, stop or mitigate the effects of the attacks. If these defenses are properly implemented in a

i3 network, the solution clearly has a very high resistance to this kind of attacks. The proposed relativity modifier for this criterion is **High**.

- **Support for location privacy**

  [SI3] and [ROAM] propose solutions to support location privacy. These solutions are very flexible and allow end-hosts to make the tradeoffs between location privacy and routing efficiency.

4.1.4.4 Performance assessment

- **Support for packet loss minimization**

  i3 end-hosts can alleviate the problem of packet loss minimization by choosing triggers that map onto nearby *i3* server. Since the number of packets that are lost during the handover are proportional to the delay between the mobile node and the server were the trigger is stored, this choice will reduce packet loss. Since it's impossible to verify how this solution will perform in an advanced deployment of *i3*, the relativity modifier for this criterion is **Low**.

- **Support for routing optimization**

  Routing in overlay networks in general is usually far less efficient that IP routing. *i3* tries to mitigate this problem by using two techniques. First, the address of the server storing the trigger is cached at the sender, so subsequent packets will be forwarded directly to that server via IP. Second, to mitigate the triangular routing efficiency problems if the trigger is stored at a server far away from the mobile node, end-hosts can use off-line heuristics to choose triggers that are stored at *i3* servers close to them.

- **Support for signaling optimization**

  The overlay network serves as the anchor point for both sides of the communication, so when a host changes its address all it does is update each of its existing triggers to point to the new address. There's no need to signal back to the home domain or to deploy intermediate anchors, so the relativity modifier for this criterion is **Full**.

4.1.5 Conclusions

Does *i3* work? Simulation results described in [I3], obtained with a bare-bones implementation of *i3* based on the Chord protocol, show that the system is feasible, but efficiency cannot be proved with such a simple implementation. The results show that routing efficiency and overall performance, particularly end-to-end latency and proximity routing, can be improved by simple

heuristics. The performance issues appear to be reasonably accounted for, but further refinement and experimentation is necessary before the *i3* overlay can be solidified.

4.2 FARA

FARA (Forwarding directive, Association, and Rendezvous Architecture) [FARA] is a general architectural model that proposes a new organization of the concepts of naming and binding in the Internet architecture. FARA is a high-level model that fulfils a set of assumptions, and is intended to be derived and instantiated by complete architectures that reduce the generality of the model, but still satisfy the assumptions of the parent model.

4.2.1 Overview

In FARA, host-to-host communication is replaced by communication between pairs of entities through logical communication links called associations. This communication is made by exchanging packets over a communication substrate that delivers data on behalf of the associations. These components are introduced below:

- **Entity**. It is a generalization of an application that is an end-point of network communication. An entity is the smallest unit that can be mobile and contains application and communication state. An entity might be a process, a computer, a cluster of computers, etc., since FARA is intended to encompass any of these forms of entities (depending on the particular FARA instantiation).

- **Association**. An association is an end-to-end (and thus invisible to the routers) communication link that entities use to communicate with each other. An association implies persistent communication state within the linked entities that evolves over the life of the association and is synchronized by the association. Each of the packets that are flowing between the entities belongs to exactly one association, and an entity can have multiple concurrent associations. Each packet carries a local **association ID** (**AId**) to enable the receiving entity to demultiplex the message to its association.

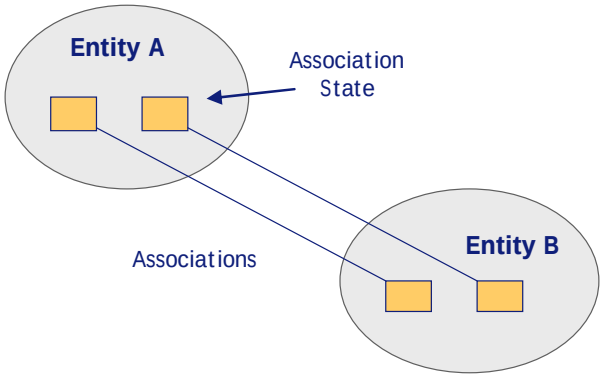

Figure 4-9: Associations between entities

- **Communication** Substrate. FARA assumes a connectionless packet delivery mechanism that is roughly equivalent to the network layer, but makes no assumptions about the delivery mechanism. A possibility would be hop-by-hop delivery with globally-unique topological addresses, as in the current IP. The delivery of the packets is all the way to the entity. Every FARA packet carries a destination Forwarding Directive (FD) that is used by the communication substrate to deliver the packet. The format and contents of the FD must be defined by the derived architectures, and could be simple global addresses, source routing or something more complicated. When an entity that anchors an association moves, the FD changes, but the AId does not, thus allowing the freedom to change the delivery path for packets belonging to a particular association. Figure 4-10 illustrates these concepts.

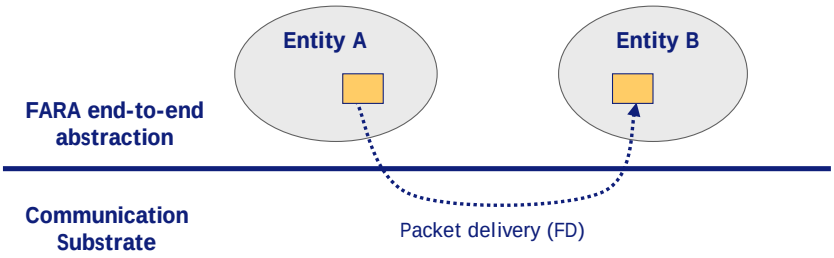

Figure 4-10: Communication substrate and FD

- **Slots**. The FD tells the network how to deliver a packet to a logical location in some system, yet the packet must still be delivered to the entity abstraction. To allow this, FARA introduced the concept of a slot, which is a logical point of attachment of the entity to the network topology.

The FARA model makes the following assumptions:

1. An entity is the unit of mobility for any kind of logical or physical mobility.

2. Associations do not have global names. The AId is local to the entity and does not change when the entity moves.

3. Entities do not have global names. The location of an entity is defined by the FD that will forward packets to that entity, but since entities do not have global names, there must be a higher-level mechanism to allow users to locate/construct FDs for target entities.

4. Globally-unique network addresses are not required. Some of the forwarding mechanisms may use a single global address space, but others may not.

When two entities want to establish an association, the entity that sends the initial message must have a FD to reach the other entity, and the AId for the association cannot be included in the first packet, since AIds are local to entities and can only be determined during the initial handshake. Two additional FARA components are needed in order to solve this bootstrapping problem.

The initial packet in an association must contain a rendezvous information (RI) string, which the destination entity can use to establish the association and assign an AId. If rendezvous takes place on the target (server) system in a client-server relationship, the RI would supply the parameters needed by the server daemon to start the entity and create the association. More general rendezvous mechanisms are admitted, so the rendezvous point might be a central agent that would rewrite the initial FD to point to another target entity or rendezvous agent.

Discovery may be accomplished by different high level mechanisms such as DNS, web sites or other programs. The FARA model Directory Service (fDS) subsumes these various discovery processes into a single generic discovery service that leaves the details to an instantiating architecture.

4.2.2 Security

FDs are not necessarily global and are not stable, because they may be rewritten or may change due to mobility. Therefore, an entity must implement some packet validation mechanism for the initial association establishment, and perhaps for all the subsequent packets in the association. FARA leaves this mechanism to the entities and to a derived architecture. The goal is to support a range of source verification mechanisms, ranging from a full cryptographic signature on each packet to no security.

4.2.3 Performance

To make sure that FARA is self-consistent and useful it is necessary to try deriving one or more specific architectures, complete with mechanisms. That's why the [FARA] describes the design and prototype of a derivative architecture: **M-FARA.** This architecture demonstrates

location/identity decoupling and explores mobility and addressing aspects of FARA. The main features of this particular architecture are described below:

- **Network addressing**. M-FARA assumes that there are multiple domains, each of which is a distinct addressing realm. The addresses are unique within each realm. A M-FARA FD contains a generalized source route of sub-FDs to traverse each realm along the path to the destination.

- **Packet delivery mechanism**. The packet delivery mechanism is hop-by-hop forwarding within realms and source routing across realms. The FD contains a realm-by-realm source route, and the reply FD will be transformed along the path so that it will be meaningful at the destination. To simplify route computations a distinguished core realm is assumed. Every entity knows a path FDup to reach the core realm, and the Directory Service contains the path FDdown to reach the target from the core realm. The sender can then easily compute an end-to-end FD as (FDup, FD down).

- **FD management**. **Mobility Agents (M-agents)** are introduced, acting as rendezvous points and as third parties to update FDs to handle mobility. When the destination entity moves it informs its M-agent that the FD has changed, so the M-agent can keep track of the entity's location. The entity sends packets with updated reply FDs to the remote entities for which it has associations, so these entities know where it is as it moves.

- **Security**. M-FARA supports authentication during the initial packet exchange that establishes the association, and also re-authentication after the entity moves. It uses a DCCP-style connection nonce.

As stated in [FARA], a prototype of M-FARA has been developed, with entities and M-agents implemented in C++ as Unix processes, and associations mapped onto Internet overlays. The prototype supports seamless migration of the endpoint of a reliable association to a new attachment point in the same or different addressing realms, as well as re-authentication using connection nonce whenever the FD changes. No simulation results are provided in [FARA].

4.2.4 Conclusions

Currently, the FARA model does not handle multicast, since associations are always between pairs of entities. Also, middleboxes and middlebox traversal is not discussed in the FARA article, so solutions for these issues are not provided, and further investigation is needed. Since FARA is, according to its designers, a work in progress, perhaps future iterations of the model will address these and other aspects.

As for the M-FARA instantiation, it does not define QoS or congestion control mechanisms. Also, it does not explore a range of rendezvous mechanisms. The current prototype is considered a “toy” implementation by its creators and does not implement the FARA Directory Service.

FARA is an interesting technology, but unfortunately it is still in a very initial stage of development and it’s a very immature technology, even when compared to alternatives like *i3*. This is why no detailed assessment is done in this section, although this description and brief analysis is included in this deliverable for the sake of completeness.

5. SHIM6

5.1 SHIM6 Overview

SHIM6 [SHIM6] is a multihoming solution in IPv6. It is a network layer approach for providing the split of locator/identifier of IP address [LOCSPLIT], so that multihoming can be provided for IPv6 with transport-layer survivability.

In SHIM6, a SHIM6 endpoint can use a constant IP address as an Upper Layer Identifier (ULID) for an association. Besides, it uses multiple IP addresses as locators (L) for routing packets. For each Upper Layer Protocol (ULP) connection, SHIM6 establishes a context state by using four signaling messages: I1, R1, I2 and R2, so the SHIM6 context, associating a ULID pair with a set of locators for endpoints, performs as a per-host header address mapping function. This functionality is indicated in Figure 5-1.

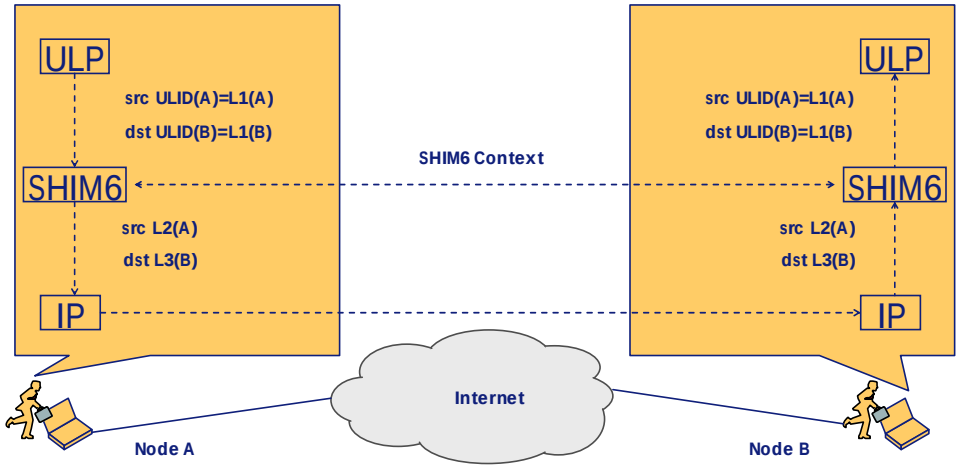

Figure 5-1: Shim6 mapping with changed locators

From the figure, we can see that, above the SHIM6 protocol of communicating endpoints (e.g. node A and node B), the ULP selects the initial locator pair (e.g. L1(A) and L1(B)) being the ULID pair [IPV6MHS], which avoids introducing a new identifier name space as well as the modification of ULP. The SHIM6 context provides a set of associations between endpoint identifier pairs (e.g. L1(A) and L1(B)) and locator sets (e.g. L2(A) and L3(B)). When packets are passed from the ULP to the IP, the endpoint identifiers of ULP are mapped to a current pair of locators. The reverse mapping is applied to incoming packets, where the incoming locator pair is stripped off the packet, and the packet header is rewritten with the mapped endpoint identifier pair. Packets are then passed to the ULP.

SHIM6 also specifies locator update messages (i.e. Update Request message and Update Acknowledgement message), a Locator Preference option and a Locator List option for changing the set of locators dynamically. For instance, SHIM6 might determine that there is a locally visible failure that implies that some locator is not longer usable. Therefore, an alternative locator can be set by SHIM6 through the Update Locator messages and a Locator List option, so that the new locator can later be used to preserve established communications through failures. This shows it has the potential of the mobility management on the Internet.

5.2 Potential of SHIM6 Mobility Support

The mobility enabled SHIM6 protocol allows for the change of the IP address of the network layer while keeping the end-to-end connection intact. In this section, we present the usage of SHIM6 in a mobility scenario. We assume that the communicating endpoints (e.g. the MN and the CN) support SHIM6 (called SHIM6 hosts) and may be mobile.

5.2.1 Dynamic addition and deletion of locators

The readdressing procedure is an operation through which a MN informs a CN about changes of IP addresses on affected locators. In SHIM6, the readdressing exchange is designed to be piggybacked on existing SHIM6 control messages. The main message on which the Locator List option and Locator Preference option are expected to be carried is the Update Request (UR) message (see Figure 5-2).

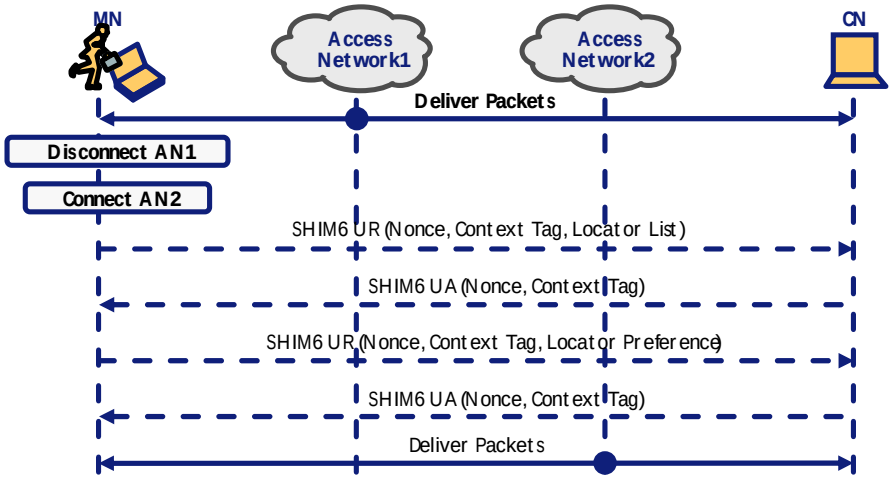

Figure 5-2: Dynamical locator update

From this figure, we can see that, in the mobility situation, when a SHIM6 host attaches to a new network (e.g. Access Network 2 (AN2) in this example) and obtains a new locator associated with the new network, it uses the Update Request (UR) message with a Locator List option. This option carries all the locators with a new added locator for the ongoing communication. This is

called “locator addition”. When a SHIM6 host detects that the current locator cannot be used because it moves out of the current location, it also uses the UR message with a Locator Preference option to indicate the current locator is dead and an alternative locator will be in use. This is called “locator deletion”.

The SHIM6 locator update messages make SHIM6 a mobility enabled protocol. This means that it allows a SHIM6 endpoint to change its locator (e.g. HoA or CoA). Furthermore, it is possible for a SHIM6 endpoint to signal to its peer which IP-address it should use as the active locator. This is very useful in case of multiple Internet accesses for seamless mobility support.

5.2.2 Mobility with single locator and ULID-Pair

This subsection considers the situations when coverage areas of access networks are distinctively apart, namely the MN has only one interface associated with an IP address at a given time and a single ULID-pair of SHIM6 context. This is the simplest scenario, as depicted in traditional MIPv6. Then, the basic operations of SHIM6 are as follows.

When a MN switches from one IP address to another, it is disconnected from the CN for a brief period of time. Upon obtaining a new IP address, the MN sends a Locator List option to the CN in an Update Request message. The Update Request message also contains a Receiver Context Tag parameter set to the value of the pre-established ULID-pair context. The Locator List option contains the new IP address and a locator lifetime. The MN waits for this locator update to be acknowledged, and retransmits if necessary, as specified in the base specification.

The CN receives the Update Request message, validates it, and extracts the context tag from the message. It then looks for a context which has a context tag that matches the one included in the Update Request message. In addition, the CN performs the address verification by the CGA Parameter Data Structure associated with the Update Request message [RFC3972, HBA]. Once any Locator List option in the Update Request has been verified, the correspondent generation number in the context is updated to be the one in the Locator List option. Then, the CN sends an Update Acknowledgment message to the MN at its new address, copying the nonce from the request, and using the context tag as the Receiver Context Tag.

Upon the reception of an Update Acknowledgement message, the MN extracts the Context Tag and the Request Nonce from the message. It then looks for a context which has a context tag that matches the one included in the Update Acknowledgement message. If the nonce matches, then the MN completes the Locator (e.g. CoA) update, considers the new address to be verified and can put it into full use.

After locator updating, packets can be routed through the new locator. When the MN sends the packets to the CN, it will set the source address field in the IPv6 header to the MN's locator and will set the destination address field in the IPv6 header to the CN's ULID. Then, a SHIM6

Payload Extension header will be added. The inserted extension header includes the CN's context tag. When packets are received by the CN, the CN will parse the extension headers in order as the normal IPv6 packet processing. If a SHIM6 Payload Extension header is found, it will extract the context tag from the Payload Extension header, and will use this to find a ULID-pair context. With the context in hand, the receiver can now replace the IP address fields with the ULIDs kept in the context. Finally, the Payload Extension header is removed from the packet, yielding the original IP datagram, which is then delivered to the upper layer protocols of the CN, finally processed by the upper layer protocols as if it had been routed to the CN's ULID.

Similarly, in order to send the packet to the MN, the CN sets the source address field in the IPv6 header to the CN's ULID and sets the destination address field in the IPv6 header to the MN's locator. When received by the MN, the MN must also process the received packet in the manner defined for IPv6 packet processing, which will result in the recovery of original packet, and then the original packet will be processed normally by upper-layer protocols within the MN as if it had been addressed only to the MN's ULID.

In such cases, there would be periods of no connectivity for a MN. This can result in considerable packet loss. We propose two methods to overcome this:

a) A buffer can be introduced in the CN. When the MN detects an impeding loss of connectivity, it can send a message to CN asking to stop packet transmission. The CN would then buffer the packets addressed to MN. When the MN reconnects to another access network, it would send an Update Request message containing the context tag. In addition to updating the locator set, this Update Request message would serve as a trigger for CN to send the buffered packets to MN.

b) A buffer can also be introduced in the Access Routers. A mechanism similar to FMIPv6 can be introduced where a tunnel is established between MN’s new and previous Access Routers. When MN connects to a new Access Router, the previous Access Router forwards all its packets through the tunnel.

5.2.3 Seamless mobility with multihoming support

It looks likely that in the future many nodes will be simultaneously mobile and multi-homed, i.e., have multiple mobile interfaces. Furthermore, if the MN is multi-homed, it is fairly likely that one of the interfaces may retain its current IP address active while some others may experience mobility and undergo IP address change, which shows a reasonable way to achieve the seamless mobility. To cover the whole paradigm of mobility, we also consider the case where access networks are overlaid or when their boundaries meet.

Figure 5-3 illustrates the Internet mobility architecture based on a mobility enabled SHIM6 supporting multihoming, called Multi-homed Mobility Architecture (MMA).

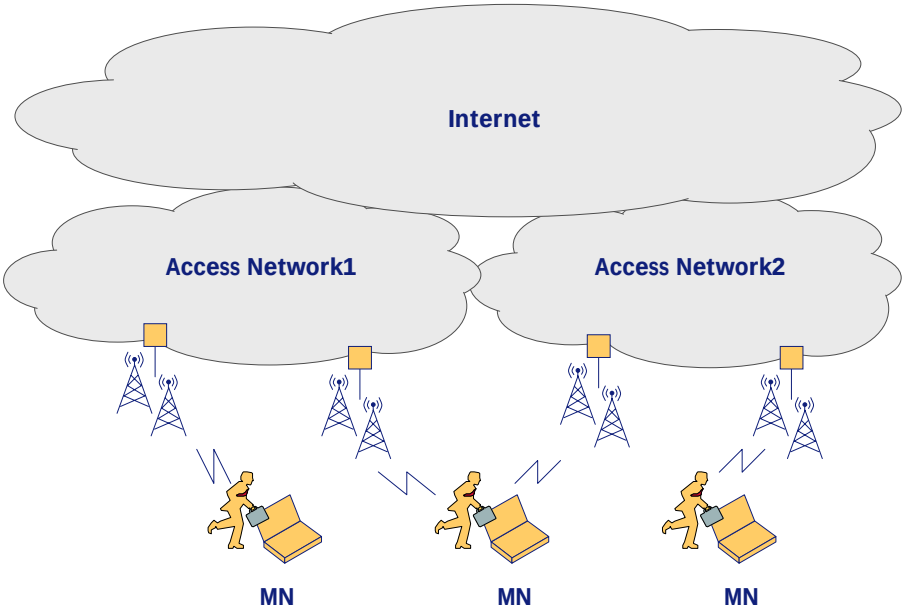

Figure 5-3: SHIM6-based Internet Mobility Architecture with Multihoming Support

In MMA, a MN connects to the Internet by some wireless technology and gets assigned an IP address from the local address space at access network 1. This can be accomplished by any of the techniques currently known for dynamic address assignment, like DHCPv6 [RFC3315] or IPv6 stateless address configuration [RFC2462]. With the MN now being reachable over the Internet, it establishes a transport layer connection to a CN and starts communicating.

When the MN moves from access network 1 towards access network 2 and gets acknowledgement of reaching the coverage of access network 2 by other information (e.g. from the physical layer of its NIC), it then establishes a link to access network 2 in addition to the already existing link and gets assigned an IP address of access network 2 on its second network interface. Thus the MN becomes multi-homed and is now reachable by two different networks.

In SHIM6, when a new prefix is advertised, a host generates an interface ID through Cryptographic Generated Address (CGA) or Hash Based Address (HBA) mechanisms. In the MMA architecture of Figure 5-3, it is appropriate to use CGA to generate an interface ID when the MN comes into the coverage area of access network 2. This method ensures that when the MN updates its list of locators through Update Request message, the new locator address can be verified.

The MN tells the CN using the established transport layer connection that it is now reachable by a second IP address. Technically speaking, it adds the newly assigned IP address as the locator to the SHIM6 context identifying the connection to the CN.

On reaching access network 2, the MN may leave the coverage of the access point at the access network 1 and may loose the link for its first IP address. The SHIM6 Payload extension header ensures that all data traffic between the MN and CN is sent over the second link in the case of permanent failure of the first link. If the MN has access to information about the strength of the wireless signal the handover to the second link will be initiated before severe packet loss occurs, ensuring seamless mobility.

5.3 SHIM6 Mobility Assessment

This Section analyzes and assesses SHIM6 mobility based on the evaluation criteria set forth in Section 2.

5.3.1 Functional assessment

- **Support for simultaneous movement of both endpoints**

  SHIM6 is not originally dedicated to mobility, so it can not currently trace the endpoint's network location by itself. Therefore, SHIM6 does not support simultaneous movement of both communicating endpoints. However, this function could be extended through dynamic DNS or rendezvous service.

- **Support for simultaneous use of multiple interfaces (multihoming)**

  SHIM6 is a multihoming solution for IPv6 that has support for failover and load sharing. This feature can be integrated into mobility seamlessly, and significantly improves mobility management in multi-homed mobile environments.

- **Support for flexible placement of service elements**

  SHIM6 is an end-to-end solution for mobility support, and no third devices are actually needed. Therefore, the service elements are only those placed on the mobile endpoints, whose location can clearly be anywhere in the Internet. Therefore, Hence the relativity modifier for this criterion is **Any**.

- **Robustness level and failover support**

  Since SHIM6 is a layer 3 solution for IPv6 networks, it can employ the neighbor discovery mechanism for detecting the movement of MN. Moreover, SHIM6 extends path failure detection and locator pair exploration mechanisms, which enables SHIM6 full robustness and failover redundancy. The relativity modifier for this criterion is **Full**.

- **Scalability**

  SHIM6 is an end-to-end system, and all operations of mobility management are performed directly between endpoints, so there is no limitation on the number of supported MNs. Moreover, SHIM6 could use the load sharing feature to avoid the network overload, which means that SHIM6 is inherently highly scalable. The relativity modifier for this criterion is **High**.

5.3.2 Deployment assessment

- **Transparency to legacy applications**

  SHIM6 is transparent to higher layer protocols. SHIM6 splits the double roles (i.e. locator and identifier) of the IP address, using one of its locators (i.e. IP addresses) as the identifier of upper layer protocols. Therefore, it doesn't require changes to current services and applications.

- **Support for legacy hosts**

  SHIM6 is an end-to-end system. All communicating endpoints (e.g. the MN and CN) need to support the SHIM6 protocol. The relativity modifier for this criterion is **None**.

- **Deployment effort**

  The deployment of SHIM6 as a mobility protocol would require the update of the communication endpoints. However, no applications need to be changed, nor any additional entities are required, so the relativity modifier for this criterion is **Medium**.

- **Operational effort**

  SHIM6 employs Locator Update (LU) messages for updating the set of locators directly between the communicating endpoints. Besides, SHIM6 uses an additional 8-byte Payload Extension header is accomplished with data traffic for direct routing. Therefore, there are no servers to be maintained for this solution to work because it is in fact a truly end-to-end approach. This advantage comes with the limitation that, since there is no rendezvous service, contemporary movement of both communicating peers is not supported. Therefore, the relativity modifier for this criterion is **Low**.

- **Need to deploy new security infrastructure**

  For SHIM6, no new security infrastructure is needed, so the relativity modifier for criterion is **No need.**

- **Maturity**

  SHIM6 is a novel technique. Its mobility support is still at an early stage. Therefore, the relativity modifier for this criterion is **Low**. However, its easy development and high mobility management performance show that SHIM6 mobility is a promising candidate for mobility management of future mobile Networks.

5.3.3 Security assessment

- **DoS resistance**

  SHIM6 employs four-way exchanges as suggested in HIP [HIP Base] for secure SHIM6 context establishment. The four-way exchanges help to protect SHIM6 against DoS attacks. Moreover, the context establishment messages use nonces to prevent replay attacks, so the SHIM6 context establishment is protected against off-path attackers from interfering with the establishment. For locator verification, SHIM6 uses the CGA technique [RFC3972] to verify that the locator is tied to the correspondent ULID, so redirection attacks are prevented. Therefore, its capability to protect itself against DoS attacks is **High**.

- **Support for location privacy**

  SHIM6 is an end-to-end solution. Both communicating endpoints support SHIM6. Whenever a locator changes at either endpoint, it will inform the other endpoint of the new location. Hence, SHIM6-based mobility management does not support location privacy.

5.3.4 Performance assessment

- **Support for packet loss minimization**

  With multihoming support, SHIM6 mobility makes the seamless handover, the optimal transport latency and the minimization of packet loss possible. Therefore, the support for packet loss minimization are **High**.

- **Support for routing optimization**

  SHIM6 supports inherent route optimization.

- **Support for signaling optimization**

  No hierarchical architecture is available in SHIM6. For each handover, singling messages are exchanged between endpoints directly. Therefore, the relativity modifier of this criterion is **None**.

5.3.5 Additional properties

- **Resiliency to Path Failures**

  SHIM6 mobility solution can provide a way of detecting path failures and recovering from them. This would ensure session continuity and least disruption to real-time applications.

5.4 Conclusions

In this section we have seen that locator/identifier splitting is needed to redirect flows from a failed path to a new path in mobile Internet environments. Like MIPv6, the separation between the locator and identifier information in SHIM6 makes clear that the packet identification and routing can be separated from each other. Thus, when communication between the initially chosen address couple for a transport connection is no longer possible, a SHIM6 layer makes it possible to switch to a different set of addresses without breaking on-going transport protocol sessions. We have shown that SHIM6 can be a new Internet mobility solution. As analyzed in previous section, we can summarize that the SHIM6 Internet mobility has the following advantages. First, SHIM6 mobility supports association of multiple IP addresses during the lifetime of any transport connections. It operates only in the end systems and affects only participating nodes. SHIM6 does not require the modification to the Internet infrastructure and does not require the modification to any node IPs or transport modules, although improved functionality can be obtained with small changes. Second, SHIM6 mobility works with existing IPv6 transport services and it does not define any new naming or addressing structure. It has a minimal additional packet-processing overhead. It employs existing administrative structures. Hence, SHIM6 mobility has a low barrier to adoption and use, while permitting more advanced functions with more extensive adoption. Finally, for a multihomed MN, it would be reasonable to associate multiple IP address with a transport connection at the time when the SHIM6 context between communicating endpoints is initiated. For a MN, the addresses may be added or removed as the node moves across the Internet, so that the MN can acquire or obsolete the use of different IP addresses. Over the lifetime of a mobile transport connection, different addresses may be active at different time. Therefore, seamless mobility support is provided for continuation of services across the movement of the MN. The main disadvantage is the lack of the location management by itself currently, which needs to be considered seriously in the future. Future work will make a continuous analysis of our proposal to be able to adapt to the future mobility architecture, and study the extension of the location management by combining with MIPv6 or Dynamical DNS (DDNS). Besides, we will study a possible transition path for its smooth deployment starting from the architecture based on MIPv6.

6. NETLMM

6.1 Motivation

If a mobility scenario doesn't involve roaming on a global scale, e.g. as the moving entities roam only within a certain administrative domain, the usage of a localized mobility management approach has many benefits. For example the mobility anchor point is placed within the localized mobility management domain, which decreases the overhead of the signaling information sent during a handover. In parallel the exchange of signaling information with a local anchor point also decreases the handover delay. Finally location privacy can be provided more easily, as the mobile node isn't required to use its address for signaling purposes outside the localized mobility management domain. This concept of local mobility management is exploited by protocols such as Hierarchical Mobile IP or Fast Mobile IP.

However, most of the currently specified localized mobility management protocols involve the mobile node itself, that is, they require certain additional functionality on the node, which makes deployment more complex and expensive. Involving the mobile node in the signaling process also mean exchanging signaling information over the air interface, which represents a consumption of costly resources.

To avoid this, network-based localized mobility management protocols are an alternative for handling the mobility without involving the mobile node itself. This approach is also very attractive for network operators, as the complete control of mobility management resides within the network. More information about the problem scope leading to network-based localized mobility management (NetLMM) can be found in [NLMMNH].

Consequently the following design goals have been identified for a possible NetLMM protocol [NLMMGL]:

- Handover performance improvement: the time required for a handover should be minimized.
- Reduction in handover-related signaling volume: the amount of signaling information to be exchanged during a handover should be minimized.
- Location privacy: corresponding nodes should not be able to recognize the mobility of their communication partner (mobile nodes).
- Efficient use of wireless resources: the introduction of new signaling on the wireless interface of a mobile node should be avoided.

- Reduction of signaling overhead: also the signaling information exchanged on a wired network behind the wireless access points should be minimized.

- No extra security between MN and network: extra security mechanisms between the mobile node and the network should be avoided.

- Support for heterogeneous wireless link technologies: the network-based localized mobility management approach should work for different wireless link technologies.

- Support for unmodified MNs: the network-based localized mobility approach shouldn't require any modifications on mobile nodes.

- Support for IPv4 and IPv6: the network-based localized mobility approach should work for IPv4 and IPv6.

Looking at the currently available host mobility management protocols, such as Mobile IP, Hierarchical Mobile IP, Fast Mobile IP or Cellular IP, none of them addresses all of the above mentioned design goals in a satisfying way. For example, the majority of these protocols is not transparent to MNs and consequently requires MN modification. Also these protocols send signaling information up to the MN, and therefore via the wireless link.

For this reason the NetLMM WG of the IETF decided to establish a design team, which has been tasked to architect a new NetLMM protocol.

6.2 Protocol Overview

This section gives an overview of a first protocol design for NetLMM [NETLMM]. This design is mainly based on two entities handling the mobility within a localized mobility management domain.

The Mobile Access Gateway (MAG) is a kind of first-hop access router within the network-based localized mobility domain, to which mobile nodes (MNs) can connect to. This means that a MAG represents the interface between a specific link layer technology used by the MN and the NetLMM-based domain behind. Usually multiple MAGs are deployed within a single domain.

The Local Mobility Anchor (LMA) is a router in the NetLMM infrastructure, which maintains the reachability of MNs within the localized mobility domain. For this purpose the LMA will communicate with one or multiple MAGs using the NetLMM protocol. One or multiple LMAs are deployed per localized mobility domain.

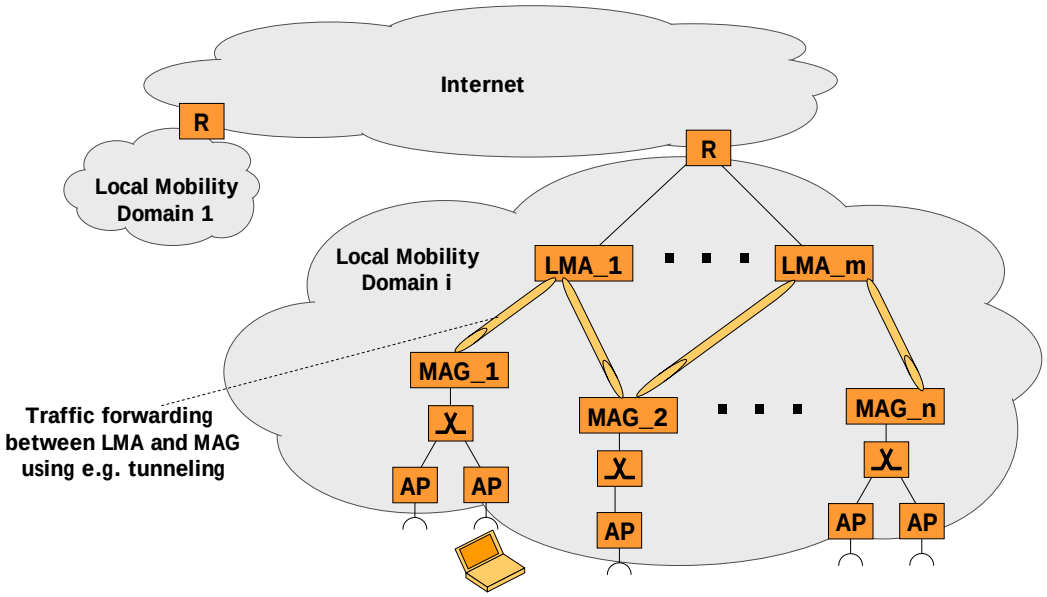

Figure 6-1: Overview of NetLMM

Figure 6-1 shows this principle of MAGs and LMAs supporting localized mobility management. For this purpose a tunnel is established between the MAGs and the LMAs. Via this tunnel data from the MN will be sent, that is, the data bridges thereby the localized mobility management domain from the MAG up to the LMA. In case a MN will roam to a new access router, the routing of the MN's IP address will be updated in order to be performed now via a new tunnel, the tunnel from the LMA to the new access router (MAG) the MN connected to.

NetLMM supports the split of identifier and locator. This is achieved by using within the NetLMM protocol identities for distinguishing the relevant nodes, such as MNs, MAGs and LMAs. These identities are often also referred to as handles. A single node distinguished by its respective identity could have multiple locators or addresses, which can be assigned to a single or multiple interfaces of the node. The significant benefit of the identifier and locator split is that NetLMM basically could run over protocols using different kind of addressing schemes, such as IPv4 or IPv6. However, for each of these address ranges a mechanism taking care if the mapping between identity and locator has to be available. In the very simple case one could use IP addresses also as identifier, that is, in this case identifier and locator would be identical and consequently no mapping would be required.

In NetLMM to each MN a complete address prefix will be assigned. This prefix could be obtained from the LMA, or via local means, such as DHCP or stateless address autoconfiguration. In the case where the MN obtains its address prefix locally, the LMA needs to be informed about this assignment. The link between the MN and the MAG is considered as a point-to-point link, that is, the MN can reach with its link-local address only the MAG, but no other nodes.

6.3 Protocol Procedures

The NetLMM protocol can be split into two phases:

- Setup and Node Association phase: this phase comprises of all signaling which is not related to a specific MN, such as the establishment of an association between MAG and LMA, the continuous monitoring of this association, as well as its removal.
- Mobility Management phase: this phase comprises of all signaling which is related to a specific MN, such as registration and deregistration of MNs, update of the MN's IP address configuration, support of MN handover between different MAGs, as well as of course the transport of data from MNs.

In the following the most important protocol procedures of the NetLMM protocol will be described in more detail in order to provide a better understanding of the protocol.

For example Figure 6-2 shows the procedure of the MAG association. From its list of available LMAs the MAG sends an Association Request to a LMA in order to establish an association between them, that is, to set up control and data plane. The LMA acknowledges this by sending back an Association Acknowledgement.

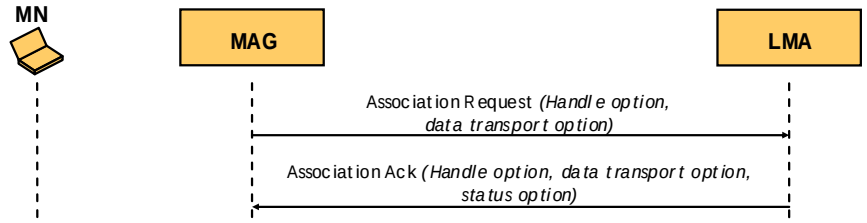

Figure 6-2: MAG Association

In order to assess the availability of the link between MAG and LMA, which is the basis for any MAG to LMA association, continuous Heartbeat messages are sent between the MAG and the LMA, which are answered by Heartbeat Acknowledgement messages. In the case where no acknowledgement message will be received after a certain time, the association between MAG and LMA will be removed. This message exchange is outlined in Figure 6-3.

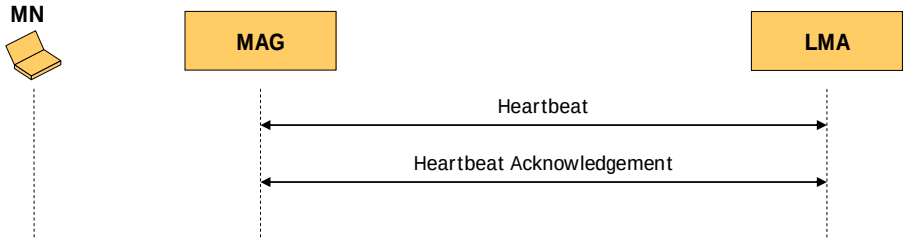

Figure 6-3: Link availability test

Both MAG and LMA can decide to remove the association between them, e.g. for reasons of upcoming maintenance. This is handled by a Disassociation Request/Acknowledgement exchange, which removes the data and control plane. Figure 6-4 illustrates this protocol procedure.

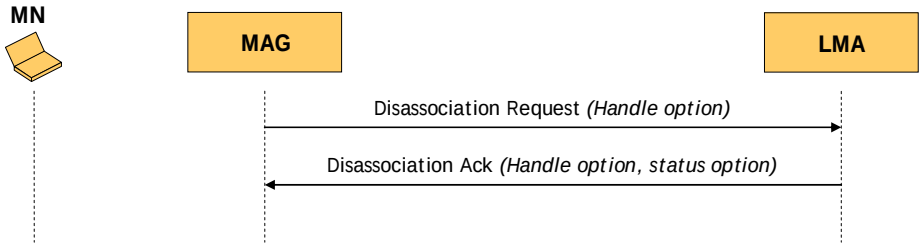

Figure 6-4: MAG disassociation

Within the Mobility Management phase one important task is to handle the network access of a MN as shown in Figure 6-5. When a MN attaches to the network, this will be indicated to the MAG via some kind of local API, more precisely, the MAG will receive a MN_Access_Network API event containing the MN ID. The ID of a MN could be specific to the technology used in the access network, for example it could be as simple as the MN's IPv6 or MAC address, or more sophisticated, the MN's SEND key. However, [NETLMM] provides no more specific information about the MN ID to be used and the local API. The MAG then sends a Location Registration to the LMA, including the IDs of the MN, the MAG and the LMA.

If the MN is not already registered at the LMA, the LMA generates a routing cache entry for it, and sends back a Location Registration Acknowledgement with the IDs of the MN, the MAG and the LMA. Additionally the LMA may decide to include in the acknowledgement an IP prefix served by the LMA. In this case the MAG needs to make use of this prefix within any kind of address allocation mechanisms it is providing. For example it needs to advertise this prefix in its Router Advertisements sent to the MN, allowing stateless address autoconfiguration for the MN, or act as DHCP relay making sure that appropriate addresses from this IP address prefix are assigned.

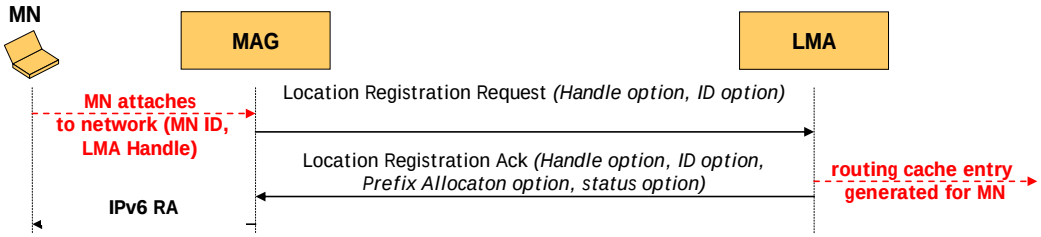

Figure 6-5: MN network access

After having registered with a MAG, the MN either configures its IP address via DHCP or stateless address auto-configuration. The MAG has to become aware of this IP address configuration, either by acting as DHCP relay itself, or monitoring the DAD process. Knowing the MN's configured IP addresses, the MAG informs the LMA about this by sending a MN Location Registration Request including the MN's current set of configured IP addresses in the NetLMM Prefix Delegation options. The LMA replies with a respective Location Registration Acknowledgement. If the registration was successful, the status option in the Location Registration Acknowledgement indicates success, and the routing cache entries for the MN on MAG and LMA are updated with the MN's current set of configured addresses. This procedure is shown in Figure 6-6. If the registration was not successful, the status option in the Location Registration Acknowledgement indicates the respective failure (e.g. maximum number of IP addresses allowed for a MN has been exceeded), and MAG and LMA do not update their respective routing cache entries.

An MN IP address notification procedure can also happen in case the MN configures an additional address at a later stage.

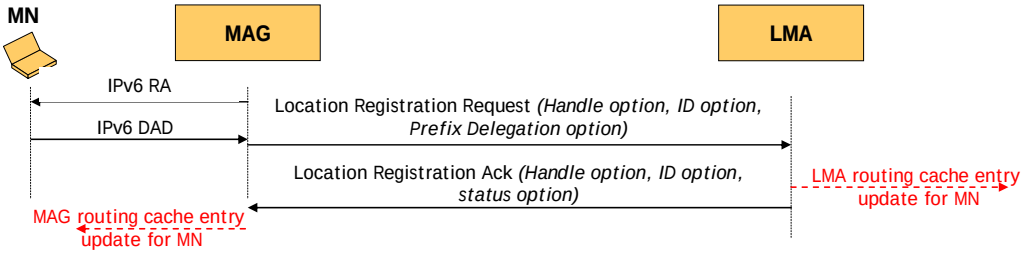

Figure 6-6: MN IP address notification

Figure 6-7 illustrates another important functionality in NetLMM, the handover between two MAGs. This is required in case the MN is moving, and attaches to a new access router. In this case the MAG sends as usual a Location Registration Request. Here the LMA already has a routing cache entry for the MN at a different MAG (previous MAG), consequently the LMA doesn't generate a new routing cache entry, but updates the existing one with the new MAG information. Finally the LMA sends back a Location Registration Acknowledgement to the MAG including a Prefix Allocation option to inform the MAG about which prefix it can use.

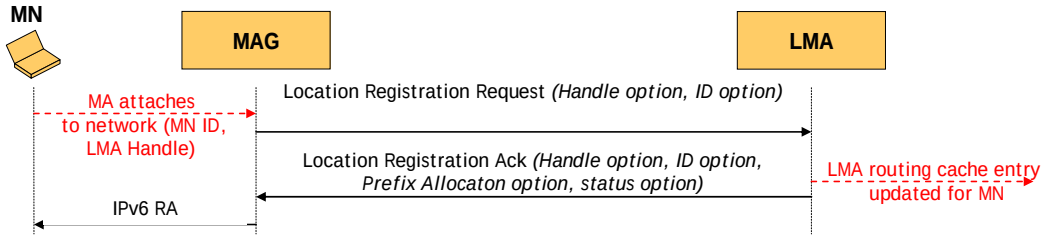

Figure 6-7: MAG to MAG handover

In certain cases it is no longer necessary for a MAG to maintain data like routing cache entries for a MN, e.g if the MN attaches to a new MAG. This situation will be detected by the LMA receiving Location Registration Requests for the MN from the new MAG. The LMA will then send a Location Deregistration message to the old MAG, triggering the old MAG to remove the MN from the MAG routing cache. This Location Deregistration can include a timer value in the Deregistration Timeout option, indicating how long the removal of the MN from the MAGs cache could be delayed. This delay of the deregistration process allows supporting make-before-break scenarios.

The respective message flow is shown in Figure 6-8.

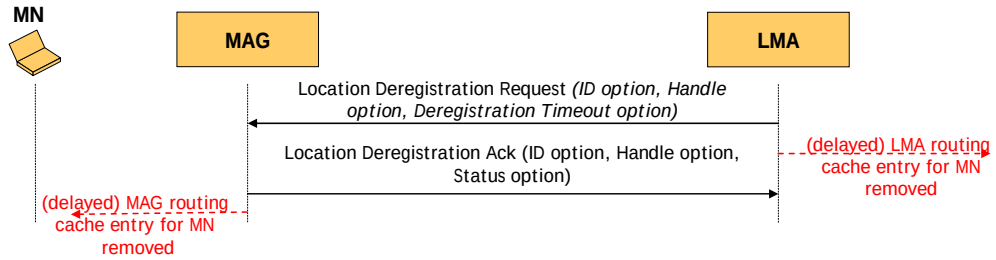

Figure 6-8: Resource revocation

Consequently, in case the MAG detects (e.g. via a local API) that a MN detaches from its network, it doesn't need to take any action, but only have to wait until it is informed by a Location Deregistration request message from the LMA as shown in Figure 6-8. After that the MAG removes the MN's entry from its routing cache. In case the MN doesn't attach to another MAG, and consequently the LMA cannot send a Location Deregistration message to the MAG, in real deployment each MAG needs some additional mechanisms to remove old MN states. [NETLMM] doesn't explicitly specify such mechanisms, however, one possibility could be using time-outs for the MAG state information, after which the respective state will automatically be removed on the MAG. The MAG can then include in its Location Registration Requests sent to the LMA the Registration Lifetime Option, which specifies the lifetime of the respective state information generated due to Location Registration Requests. This way time-outs could be synchronised between MAGs and LMAs.

A Location Deregistration could also be triggered administratively at the LMA, however, [NETLMM] doesn't specify more details about the nature of such triggers.

Finally one of the most important protocol procedures is the forwarding of data to a MN as outlined in Figure 6-9. The MAG acts as stated already above as first-hop / default router for the MN and therefore forwards and receives all the MN's packets. These packets are routed from the MAG to the LMA or vice versa. On the other side the LMA forwards / receives a MN's packets to / from outside the local mobility domain. MAG and LMA encapsulate MN's packets when forwarded between each other using tunneling mechanisms, such as IPv6-in-IPv6, GRE, or others.

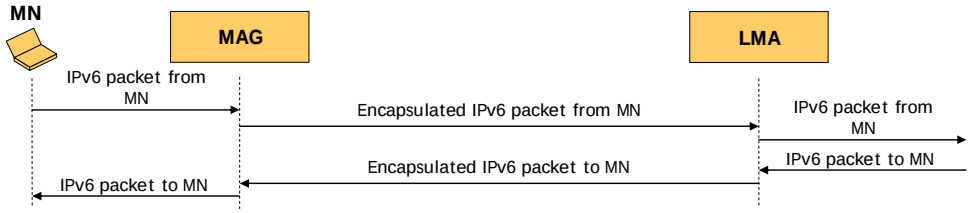

Figure 6-9: Data forwarding

6.4 NetLMM Assessment

Based on the evaluation criteria specified in Section 2 an assessment of NetLMM has been performed and is described below.

6.4.1 Functional assessment

- **Support for simultaneous movement of both endpoints**

  As long as both communication endpoints (MNs) move within their respective Localized Mobility Management Domain (LMMD), independently if they are located in the same or in different LMMDs, they will keep their assigned IP address. Consequently it is possible to support simultaneous movements of both communication endpoints and maintain session continuity. Only in case one of the communication endpoints roams into a new LMMD, its IP address will change, and NetLMM will no longer be able to provide session continuity.

- **Support for simultaneous use of multiple interfaces (multihoming)**

  It is possible to support host multihoming by NetLMM. A MN can configure multiple IPv6 prefixes assigned to a single interface, however, all of them have to be registered at the same LMA, that is, each MN only has a single LMA assigned each time.

In case these multiple IPv6 prefixes are assigned to different interfaces, the MN can also use NetLMM by attaching each of its interfaces to a different localized mobility domain. However, attaching more than one interface to a single localized mobility domain would cause problems, as the Location Registration of an interface will trigger the Location Deregistration of a previously registered interface, that is, only one interface will be able to keep registered at a single localized mobility domain.

Additionally LMAs and MAGs can have multiple IPv6 addresses themselves, that is, also they can be multi-homed.

- **Support for flexible placement of service elements**

  Since the mobility management of NetLMM is constrained within Localized Mobility Management Domains (LMMDs), NetLMM-aware service elements must be placed in a single LMMD. While the placement of LMAs is flexible within each LMMD, MAGs have to be placed as access router. Consequently the mobility anchor, that is the LMA, can be placed anywhere, therefore the relativity modifier for this criterion is **Any**.

- **Robustness level and failover support**

  Failure recovery mechanisms are mainly achieved by the deployment of redundant key components, i.e., multiple MAGs and multiple LMAs can be deployed in each single LMMD. In order to detect failures, Heartbeat messages are periodically sent between the LMA and the MAG. However, how to react in case of failures has not been specified in the current draft [NETLMM]. As precise failover mechanisms are not specified yet, the relativity modifier for this criterion is **None**.

- **Scalability**

  NetLMM does not target the global mobility management issue, but rather the provision of a localized mobility approach. However, the NetLMM domain (which contains MNs, LMAs, MAGs and in-between routing fabric) could spread over a very large geographical area. The number of MNs per LMMD that the NetLMM protocol can support is not limited as the level of scalability can be increased by increasing the number of MAGs and LMAs. Assuming that the LMA is not statically configured on the MAG, but dynamically discovered via an API on the MAG, the relativity modifier for this criterion is **High**.

6.4.2 Deployment assessment

- **Transparency to legacy applications**

  NetLMM is transparent to legacy applications because the NetLMM address of the MN will be kept constant in case the MN changes its point of attachment. In this way session continuity is provided as long as the MN roams within the LMMD.

- **Support for legacy hosts**

  The NetLMM protocol itself is transparent to MNs. Obviously, there are no specific requirements on the nodes outside the NetLMM domain. However, NetLMM assumes that the MN implements a number of other standard protocols. Examples are Neighbour Discovery for IPv6, DNA, SEND, CGA, etc. Therefore the relativity modifier for this criterion is **Both**.

- **Deployment effort**

  The NetLMM infrastructure requires a number of LMAs and MAGs. The number of them depends on the size of the domain and the requirements for robustness. There is no need to establish agreements (e.g., roaming agreements) between different LMMDs.

  Currently no bootstrapping mechanism is defined for NetLMM. Designing a bootstrapping approach could involve also service authorization, which can require interworking with a backend AAA infrastructure and configuring IPsec Security Associations (SAs) or IKE(v2) authentication credentials at the MAGs and LMAs.

  In summary many components need to be deployed for NetLMM, but their functionality has only limited complexity. Therefore the relativity modifier for this criterion is **Medium**.

- **Operational effort**

  The effort for monitoring NetLMM components grow linearly with the number of deployed entities in the NetLMM domain. Reconfiguration and manual re-keying effort is expected to be limited to the NetLMM specific components. Therefore the relativity modifier for this criterion is **Low**.

- **Need to deploy new security infrastructure**

  The NetLMM draft [NETLMM] has pointed out some potential threats, but it does not specify a new security infrastructure. However, it mentions as one possibility to address these threats the use of IPSec ESP for securing signaling traffic between MAGs and LMAs.

Furthermore NetLMM gives an overview of plausibility checks for NetLMM control messages. At the current level of specification the assumption is that the relativity modifier for this criterion is **Partial need**.

- **Maturity**

In its current stage, the technology is not ready for deployment. A version 2 draft of the protocol has been issued in October 2006, and probably will experience minor modifications and changes in future. The first prototyping work is currently ongoing. Therefore the relativity modifier for this criterion is **Low**.

6.4.3 Security assessment

- **DoS resistance**

In NELMM the flooding of LMAs and MAGs represent one possible DoS attack. To secure the communication between LMA and MAG, preconfigured IPsec SAs could be used. This would at least limit the number of possible DoS attacks. Therefore the relativity modifier for this criterion is **Medium**.

- **Support for location privacy**

The MN obtains its IP address either by using DHCP (using stateful address configuration), or by using stateless address autoconfiguration. While the MN roams within the LMMD, it will continue to keep this address, that is, location privacy is given in the LMMD. When the MN roams outside the LMMD, it needs to obtain a new address, and location privacy is no longer given. Consequently any correspondent node can trace the LMMD where the MN is located, but cannot figure out the exact location of the MN within the LMMD. Therefore support for location privacy can be improved building large LMMDs.

6.4.4 Performance assessment

- **Support for packet loss minimization**

NetLMM can support “make before break” by sending Location Deregistration with a Deregistration Timeout option, which allows simultaneous tunnels to be established between the LMA and more than one MAG during handover procedure. However, the possibility of seamless service provisioning depends on the possibility to do seamless handover on L2 (outside the scope of NetLMM). IP packets can be cached, but may be delayed. Under the assumption of exploiting the “make before break” capability of NetLMM the relativity modifier for this criterion is **High**.

- **Support for routing optimization**

  All traffic between the MN and its communication partner(s) must go through the LMA (in and out). Even if the LMA is not located necessarily on the shortest path between the MN and the communication partner, the number of additional hops should be very small since all entities involved in a single NetLMM domain. Furthermore in many scenarios routing optimization within the LMMD may not be needed at all. For example if the access network has a tree topology and the LMA is placed in the root of the tree, it is clear that routing optimization is not needed. Therefore, it's not necessary to perform routing optimization.

- **Support for signaling optimization**

  NetLMM is designed for minimizing signaling overhead via the air interface. NetLMM only exchanges signaling information between MAGs and LMAs, and therefore does not add any overhead on the air interface. Therefore the relativity modifier for this criterion is **Full**.

6.4.5 Additional properties

- **Handover delay**

  The preliminary simulation on NetLMM protocol has been done by TI before the draft [NETLMM] has been published. Through the simulation, it shows that the handover delay of NetLMM is nearly as low as that of HMIPv6 and better than that of MIPv6 (with or without local HA).

- **DNS update**

  As the NetLMM intends to provide local mobility solution, a DNS update is required each time the MN leaves the LMMD. However, when NetLMM is used in a combination with MIPv6, only the CoA will change and no DNS update for the HoA would be required. It is therefore better to regard NetLMM as a complementary support to MIPv6 rather than a contradictory solution.

- **Access authentication**

  NetLMM supports the integration of different network access technologies. That is, either specific L2 access control mechanisms or a common L3 access control mechanism such as PANA [PANA] can be used.

- **Split of locator and identifier**

  NetLMM supports a locator/identifier split. All NetLMM protocol messages use identifiers in order to identify specific nodes. This has the benefit of allowing the NetLMM protocol to run over different addressing schemes, such as IPv4 or IPv6. The NetLMM identifier needs to be resolved to a NetLMM locator for packet transmission. In the simplest case an IPv6 address is also used as identifier, so no resolution is required.

- **Open issues**

  The draft still has many open issues:

  1. More details need to be specified on how the routing cache of a MAG will be cleaned up from states of MNs which leave a MAG and do not sign up at a new MAG.

  2. A bootstrapping process needs to be designed, containing e.g.

     a. Mechanisms to have a MAG learning all available LMAs.

     b. Mechanisms to decide about with which available LMAs the MN should associate with.

     c. Mechanisms to decide about which LMA the MAG should choose for registration (required information for MN_Access_Network API).

  3. It needs to be specified, how the mapping between locator and identifier will be done.

6.5 Conclusions

NetLMM is a protocol that allows management of mobility in a localized domain. It has been consequently designed to fulfil a list of requirements concerning the management of local mobility in today's wireless networks, such as the transparency to mobile nodes, the efficient usage of the resources on the wireless link as well as the improvement of handover delay and signaling. Even if the version 2 draft of the protocol still has some open issues to be addressed in the future version, the NetLMM protocol design seems to be very promising in addressing the respective requirements. However, concerning the actual chances of deployment, NetLMM may face difficulties to compete with existing approaches used in standardization bodies like 3GPP and WiMax, less as it is less performable than alternatives, but more as these standardization bodies tend to stick to their established solutions for political and deployment reasons. A competitive approach to NetLMM in these areas is for example Proxy MIPv6.

From its nature NetLMM as localized mobility management protocol is basically suitable for being combined with Mobile IPv6 as a global mobility management protocol. One obvious approach for combining NetLMM and MIPv6 is using the IPv6 address which is assigned to a MN while roaming within the localized mobility domain as its Mobile IPv6 CoA. In this way the CoA remains the same as long as the MN stays within a certain localized mobility management domain, and consequently no MIPv6 handover happens. A MIPv6 handover, requiring an update of the Binding with the HA and potential CNs, only happens when the MN is roaming between different localized mobility management domains. Clearly further investigation about the detailed integration of NetLMM and MIPv6 is required, investigating, amongst other items, handovers during local and global movement, as well as the bootstrapping and service authorization phase.

7. PROXY MOBILE IPV6 (PMIPV6)

7.1 PMIPv6 Overview

PMIPv6 [PMIP6] is one of network-based mobility management protocols which can avoid tunneling overhead over the air as well as hosts' involvement in mobility management. PMIPv6 focuses on extending MIPv6 [RFC3775] to achieve mobility due to two main reasons. The first reason is that MIPv6 is a very mature mobility protocol for IPv6. There have been many implementations and inter-operability events where MIPv6 has been tested. PMIPv6 may re-use these mature mechanisms as much as possible to solve the real deployment problem. Second, PMIPv6 allows re-using the MIPv6-aware home agents to provide mobility to hosts without using any additional mobility management protocol.

PMIPv6 introduces a new entity, Proxy Mobile Agent (PMA), which acts as a relay node between the HA and the mobile node. It also conceals the roaming information to the mobile node by emulating its home link properties. On the one side, PMA performs the mobility signaling on behalf of the mobile node. By establishing a tunneling between the home agent and the PMA, any packet sent by the mobile node will be simply routed to the home agent over the tunnel. When the PMA receives packets from the tunnel, it will forward them on the access link. On the other side, the PMA makes the mobile node believe that it is always at its home link.

7.2 PMIPv6 Procedure

Totally, the PMIPv6 protocol consists of five phases:

- **Access Authentication**

  This procedure ensures a valid MN is accessing into the network, as shown in Figure 7-1. Once a MN enters the PMIPv6-aware network and performs the access authentication, a network home address/prefix must be specifically assigned for it. Meanwhile, MN firstly presents its identifier, such as NAI, to the network for the purpose of access authentication. Through a successful authentication by the policy server (e.g., AAA server), the PMA can retrieve the MN's profile which is associated with MN’s current identifier to identify the essential parameters for providing mobility services. For example, MN’s profile may contain the MN’s home network prefix, permitted address configuration modes and roaming policy. Moreover, the PMA emulates the home link properties based on the information from the profile.

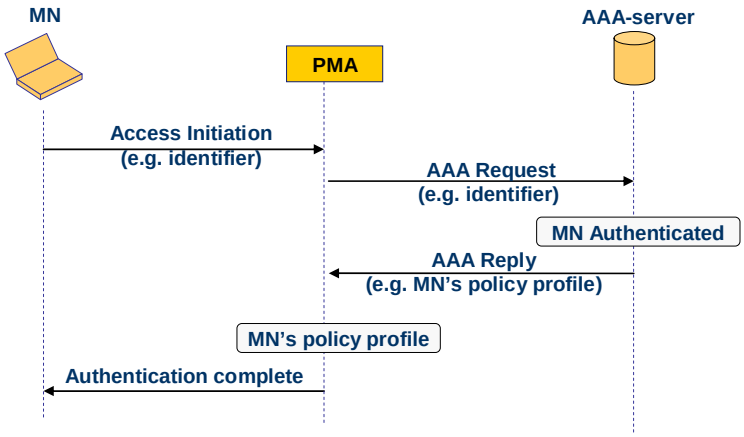

Figure 7-1: Access Authentication

- **Binding Update (BU)**

This is the second phase as depicted in Figure 7-2, in which the PMA needs to register the MN's new point of attachment with its home agent by sending a Proxy Binding Update message. The contents of the message include the MN's NAI option, alternate CoA option (optionally) and a NAI identifier of the PMA that is sending this request. The message will be then recognized through the MN's identifier. After receiving such a PBU message, the HA will send an AAA Query to the AAA server for obtaining MN's policy profile. Typically, the policy information is configured in a policy store, such as AAA. If the query can be validated, the AAA server will reply with MN-related policy information to the HA.

Accordingly, the home agent will create a binding cache entry, a tunnel towards the PMA, and as well as a route for the MN's home prefix. Upon receiving the Binding Acknowledgment from the home agent, the PMA will create a reverse route over the tunnel to the home agent. All traffic from the MN that the PMA receives in the role of a default router will be routed to the HA over the established tunnel.

To protect the PMIPv6 signaling transmission, the PMA and the HA use Encapsulating Security Payload (ESP) header and non-NULL payload authentication algorithm to provide data origin authentication, data integrity and optional anti-replay protection.

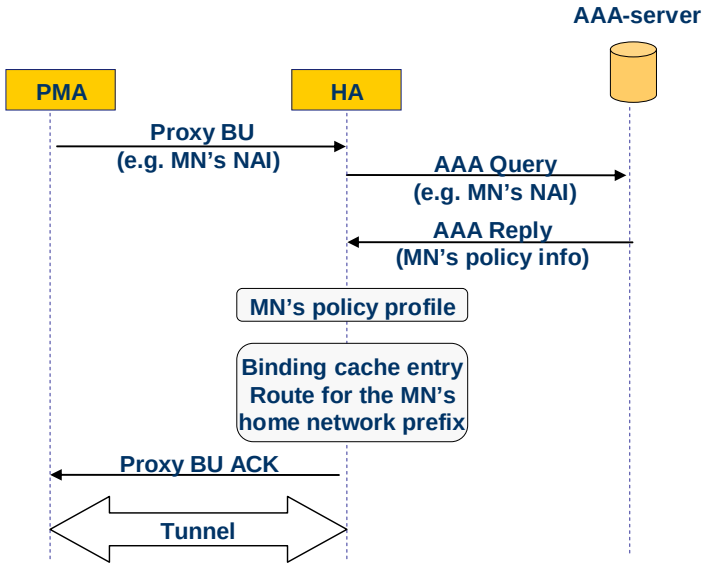

Figure 7-2: Binding Update

- **Home Link Emulation**

  The third phase will be the home link emulation, where the PMA emulates the MN's home interface on the access interface. Based on the MN's policy profile, the PMA obtains the information which contains the MN's home prefix and other attributes defined for the MN's home link. Then, it sends *Router Advertisement* message with the MN's home link related information. Further, the PMA will act as a default router for the MN.

- **MN's address configuration**

  Figure 7-3 describes the whole procedure of MN's address configuration. Based on the flags specified in the *Router Advertisement (RA)*, the MN will use either stateful or stateless configuration methods for its interface configuration. In the *RA* message, only the current link local address will be different from the one which MN received from the previous router, which makes the MN believe that there is a new default router on the home link. Then, the MN will obtain a valid home address for configuring its interface. Further, in the current specification, the DAD procedure is not necessary since all messages are sent and seen only by the MN and the serving PMA, and the network prefix is specifically allocated to the MN.

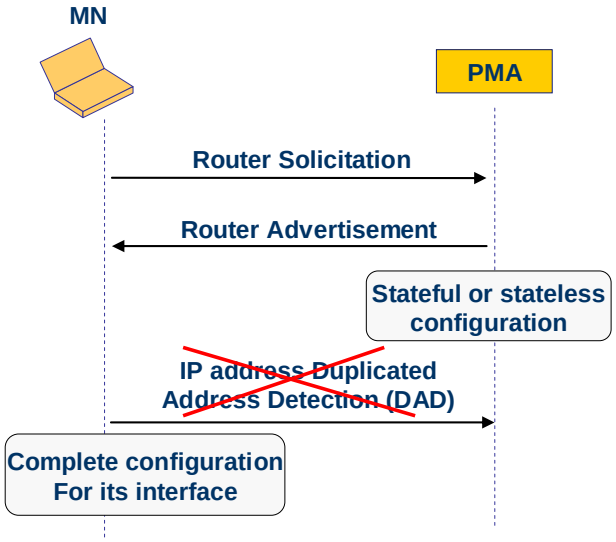

Figure 7-3: Address Configuration

- **Packet Routing**

  For the packet routing, the home agent will route all received packets over the established tunnel (shown in Figure 7-2) to the PMA. The PMA will forward them on the access link. Certainly, the PMA will forward all the packets from the MN over the tunnel to the home agent and then they will be routed to the destination.

7.3 PMIPv6 Operation

7.3.1 Home agent operation

Regarding with the home agent's operations in the PMIPv6, it has four key differences from its operation in the base protocol IPv6 [RFC2460]:

- Management of a binding cache entry in order to properly route the packets to the MN because the MN is not anchored on any physical interface on the HA.
- Management of the specific correspondence between the MN's home address and its home network prefix.
- Management of the route entry to specify whether the MN's home prefix is reachable or not.
- Management of the binding updates sharing with the same Care-of-Address and possibly the same tunnel in the case that multiple MNs are currently visiting the same PMA.

7.3.2 Proxy mobile agent operation

With respect to the Proxy Mobile Agent, it performs the following three functionalities.

- Management of serving MNs by maintaining a Visitor List.
- Emulation of the MN's home network on the access link.
- Updating the current location of the MN to the home agent.
- Setting up the tunnel to the home agent and maintaining such a tunnel.
- Establishment of data path for enabling the MN to use its home address for communication.

The bi-directional tunnel between the home agent and the PMA is configured as follows:

- Tunnel source address is either home agent's address or the PMA's address.
- Tunnel destination address is either the PMA's address or the home agent's address.
- Tunnel encapsulation mode is IPv6 in IPv6.

7.3.3 Mobile node operation

The operations of MN can be classified into two aspects, which are bootstrapping and roaming:

- Bootstrapping
    - Presenting the identifier to the network for the access authentication.
    - Sending Router Solicitation message to the PMA for address configuration.
- Roaming
    - Detection of its home network prefix.
    - Disable Duplicate Address Detection (DAD).
    - Update the default-router list.

7.4 PMIPv6 Assessment

Based on the evaluation criteria specified in Section 2, an assessment of PMIPv6 has been performed and is described in the following.

7.4.1 Functional assessment

- **Support for simultaneous movement of both endpoints**

  PMIPv6 uses Proxy Mobile Agent (PMA) to send Proxy Binding Updates. Each PMA updates the current point of attachment to the home agent and hence support of simultaneous movement of end hosts is guaranteed.

- **Support for simultaneous use of multiple interfaces (multihoming)**

  PMIPv6 alone cannot support end host multi-homing. However, it is possible that a MN configures multiple home network prefixes and all of them have to be registered at the same HA. Besides, each MN only has single serving PMA assigned each time.

- **Support for flexible placement of service elements**

  The mobility anchor in PMIPv6 is the HA. PMIPv6-aware service elements (i.e. PMAs) must be placed in each PMIP domain since they have to emulate the home link properties to make the MN believe that it is at its home link. While the placement of PMAs for each PMIP domain is flexible, therefore the relativity modifiers concerning this criterion will be **Any**.

- **Robustness level and failover support**

  Failure recovery mechanisms could be achieved by the deployment of redundant PMAs in each network. In addition, the proxy binding update and associated acknowledgement messages between proxy mobile agent and home agent will ensure the robustness. Thus, the relativity modifier for this criterion is **None** since the failure recovery mechanisms have not been specified in the current specification of PMIPv6.

- **Scalability**

  Similar to MIPv6, PMIPv6 makes use of IPv6 addresses which accommodate the increasing number of mobile nodes and ensures that each mobile node has at least one IP address, and as well as specified home network prefixes. However, in current proposal PMA is statically configured within each PMIP domain (and no fine grained load sharing across available PMAs is possible). That is, the scalability level is **Medium**.

7.4.2 Deployment assessment

- **Transparency to legacy applications**

  PMIPv6 is transparent to legacy applications because once the address configuration is complete, the mobile node will always able to use the IPv6 address anywhere within that managed network where proxy mobile agents are deployed.

- **Support for legacy hosts**

  PMIPv6 certainly supports the legacy IPv6 hosts. Note the PMIPv6 alone does not support communication with legacy IPv4 nodes. Therefore, the relativity modifier for this aspect is **Both**.

- **Deployment efforts**

  The PMIPv6 infrastructure introduces a new functional entity, PMA, which is responsible for performing the Proxy MIPv6 signaling on behalf of the MN. The number of them may depend on the size of the domain and the requirements for robustness. There is no need to establish agreements (e.g., roaming agreements) between different PMIP domains. In summary the relativity modifier for this criterion is **Medium**.

- **Operational efforts**

  In order to make PMIPv6 operational, one entity called proxy mobile agent (PMA) is mandatory except for the operational requirement from MIPv6. The relativity modifier for this criterion is **Low**.

- **Need to deploy new security infrastructure**

  PMIPv6 introduces a new functional entity, PMA to perform the Mobile IPv6 signaling on behalf of the mobile node. Besides, it requires the valid home address to tunnel packets and therefore it might be necessary to have the security infrastructure. Based on the current specification, the relativity modifier for this criterion is **Partial Need**.

- **Maturity**

  PMIPv6 is mainly based on MIPv6 which can be regarded as a mature concerning technology. In current stage, however, the PMIPv6 has not been deployed. So its relativity modifier for this criterion is **Low**.

7.4.3 Security assessment

- **DoS resistance**

  In PMIPv6, many attacks can be prevented by authenticating proxy binding updates. Besides, both the PMA and the HA must support or should use the Encapsulating Security Payload (ESP) header in transport mode and must use a non-NULL payload authentication, data integrity and operational anti-relay protection. Therefore, at the current level of specification of PMIPv6 relativity modifier for this criterion is **Medium**.

- **Support for location privacy**

  The MN obtains its IP address either by using DHCP (using stateful address configuration), or by using stateless address configuration. While the MN roams within the managed network, it will continue to keep this address, that is, location privacy is given in the PMIPv6-aware network.

7.4.4 Performance assessment

- **Support for packet loss minimization**

  In PMIPv6, proxy mobile agent prevents the mobile node from knowing the changes of its location. PMA is responsible for updating bindings to the HA instead of the MN. Therefore, this aspect of PMIPv6 should be **Low**.

- **Support for routing optimization**

  All traffic between the MN and its communication partner(s) must go through the PMA (in and out). If there are hierarchical PMAs deployed in the same network, it might have the necessity to perform routing optimization. Otherwise, it would be not useful.

- **Support for signaling optimization**

  PMIPv6 is designed for minimizing signaling overhead over the air. PMIPv6 only exchanges signaling information between HAs and PMAs, and therefore does not add any overhead on the air interface. Moreover, PMIPv6 allows the mobile node to move within the managed network without changing its home address. The relativity modifier for this aspect is **Full**.

7.4.5 Additional properties

Current PMIPv6 [PMIP6] is still under working process.

7.5 Conclusions

In this section we've firstly given a brief overview about Proxy Mobile IPv6. Then, five basic procedures, namely, Access Authentication, Binding Update, Home Link Emulation, MN's Address Configuration, and Packet Routing have been introduced. In terms of involved three entities in PMIPv6, in section 7.3 their operations have been presented as well. Lastly, according to the evaluation criteria mentioned in Section 2, the initial assessments of PMIPv6 are depicted.

Actually, the design of PMIPv6 follows the same architectural principal as NetLMM, which handles mobility in a localized domain with minimal terminal involvement. PMIPv6 focuses on maintaining a stable "home" address, whereas NetLMM proposes to maintain a stable "regional" addresses. For PMIPv6, PMA performs the necessary mobility signaling with the home agent (HA) on behalf of the MN. In this case, the MN's movement will be updated at the HA but MN itself always believes that it is on its home network. Differently, NetLMM hides the MN's movement from the HA if it moves within the domain, and maintains the reachability for the MN.

In summary, PMIPv6 is actually an extension to the existing MIPv6, which intends to provide seamless mobility within a locality. As a lot of implementations of MIPv6 have been tested, PMIPv6 likely has no collision with the realistic deployment issue. Currently, PMIPv6 is still under development so that several other properties might be specified in the near future.

8. SUMMARY TABLE

Table 8-1 shows a summary of all the technology assessments, allowing the reader to easily compare the different mobility technologies evaluated in this deliverable.

Table 8-1. Evaluation summary table

Functional Criteria						
Criterion	**MIPv6**	**HIP**	***i3***	**SHIM6**	**NetLMM**	**PMIPv6**
Support for simultaneous movement of both endpoints	✓	✓	✓	✗	✓	✓
Support for simultaneous use of multiple interfaces (multihoming)	✗	✓	✓	✓	✓	✗
Support for flexible placement of service elements	Home only	Any	Any	Any	Any	Any
Robustness level and failover support	None	Partial	Full	Full	None	None
Scalability	High	High	Medium	High	High	Medium
Deployment Criteria						
Criterion	**MIPv6**	**HIP**	***i3***	**SHIM6**	**NetLMM**	**PMIPv6**
Transparency to legacy applications	✓	✗	✓	✓	✓	✓
Support for legacy hosts	CN Only	None	None	None	Both	Both
Deployment Efforts	Medium	High	High	Medium	Medium	Medium
Operational Efforts	Medium	Medium	Medium	Low	Low	Low

Need to deploy new security infrastructure	No need	Partial need	Partial need	No need	Partial need	Partial need
Maturity	High	Medium	Low	Low	Low	Low
Security Criteria						
Criterion	**MIPv6**	**HIP**	***i3***	**SHIM6**	**NetLMM**	**PMIPv6**
DoS resistance	Medium	High	High	High	Medium	Medium
Support for location privacy	✓	✗	✓	✗	✓	✓
Performance Criteria						
Criterion	**MIPv6**	**HIP**	***i3***	**SHIM6**	**NetLMM**	**PMIPv6**
Support for packet loss minimization	Low	Low	Low	High	High	Low
Support for routing optimizations	✓	✓	✓	✓	✗	✗
Support for signaling optimizations	None	None	Full	None	Full	Full

9. CONCLUSIONS

This deliverable has provided a state of the art analysis of the different Mobile IPv6 alternatives currently under study within different standardization forums, as well as experimental approaches published in the scientific literature. The analysis of each technology included an overview of the technology, detailing the most important aspects and the possible mobility-related issues. After this description, an assessment of the technology has been done in order to provide an in-depth evaluation of the mobility solution. In order to perform this assessment, a set of evaluation criteria for mobility systems has been developed. All the technologies have been assessed against those criteria and the results have been presented in a summary table for easy comparison.

After this analysis and the comparison between of all these mobility solutions, we have identified those that will have more chances to be deployed in the future. One thing that becomes clear after this analysis is that the studied technologies can be classified into two different groups. On the one hand, we have technologies that are disruptive in the sense that they are, in essence, a substitute for Mobile IPv6. This is the case of all the indirection technologies (i.e. HIP, I3 and related technologies). Future deployment of these technologies is a quite difficult task, as Mobile IPv6 is a standardized solution that won't likely be replaced by a different approach in a long time. While they are undoubtedly interesting technologies that in some cases have the potential for a great improvements over Mobile IPv6, there may be little benefit in continuing work on them within ENABLE, as the chances of mid-term or even long-term deployment of these solutions are very low.

On the other hand, we find that other mobility solutions studied in this document are conceived as a complement to Mobile IPv6, basically supporting some optimization aspects. These technologies are less disruptive and have a greater chance to be deployed. They also offer significant improvements over the existing Mobile IPv6 standard, so they are ideal candidates for further research within ENABLE. In particular, NETLMM, PMIPv6 (due to its recent adoption by WiMAX) and Shim6 are the technologies that have been chosen to be studied and improved in the second year of the project.

10. REFERENCES

[CHORD] Stoica, I., Morris, R., Karger, D., Kasshoek, M. F., Balakrishnan, H. “Chord: A scalable peer-to-peer lookup service for internet applications”. In Proc. ACM SIGCOMM’01 (San Diego, 2001), pp. 149–160.

[ENA-D1.2] Project IST-ENABLE Deliverable D1.2, “Solutions for Mobile IPv6 bootstrapping and load sharing across Home Agents”, January 2007

[ESP] S. Kent, " IP Encapsulating Security Payload (ESP)", draft-ietf-ipsec-esp-v3-10.txt (work in progress), March 2005

[FARA] D. Clark, R. Braden, A. Falk, and V. Pingali. “FARA: Reorganizing the Addressing Architecture”. In Proc. ACM SIGCOMM’03, Aug. 2003. ACM SIGCOMM 2003 Workshops, August 25-27, Karlsruhe, Germany.

[HBA] M. Bagnulo, “Hash Based Addresses (HBA),” Internet Draft (work in progress), IETF, Oct. 2005. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-shim6-hba-01.txt

[HIP AF] Al-Shraideh, F., “Host Identity Protocol**”,** Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, 2006. ICN/ICONS/MCL 2006. International Conference on, 23-29 April 2006 Page(s):203 – 203

[HIP Arch] R. Moskowitz, P. Nikander, “Host Identity Protocol Architecture“, RFC 4423, MMay 2006

[HIP Base] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, “Host Identity Protocol“, draft-ietf-hip-base-06 (work in progress), June 2006

[HIP DNS] P. Nikander, J. Laganier, “Host Identity Protocol (HIP) Domain Name System ((DNS) Extension”, draft-ietf-hip-dns-06 (work in progress), February 2006

[HIP DT] D. Thaler, “A Comparison of Mobility-Related Protocols”, < draft-thaler-mobility-comparison-01.txt

[HIP DXD] Deguang Le, Xiaoming Fu, Dieter Hogrefe, “A Review of Mobility Support Paradigms for the Internet”, IEEE Communications Surveys and Tutorials, Volume 8, No. 1, First Quarter, IEEE, ISSN 1553-877X, 2006

[HIP ESP] P. Jokela, R. Moskowitz, P. Nikander, "Using ESP transport format with HIP", draft-ietf-hip-esp-03 (work in progress), June 2006

[HIP HTAJJ] Henderson, T.R., Ahrenholz, J.M., Kim, J.H., "Experience with the host identity protocol for secure host mobility and multihoming", Wireless Communications and Networking, 2003, WCNC 2003. 2003 IEEE, Volume 3, 16-20 March 2003 PPages (s): 2120 – 2125 vol.3

[HIP HTR] Henderson, T.R., „Host mobility for IP networks: a comparison"**,** Network, IEEE, Volume 17, Issue 6, Nov.-Dec. 2003 Page(s):18 – 26

[HIP JRJ] Jokela, P.; Rinta-aho, T.; Jokikyyny, T.; Wall, J.; Kuparinen, M.; Mahkonen, H.; Melen, J.; Kauppinen, T.; Korhonen, J.; "Handover performance with HIP and MIPv6**",** Wireless Communication Systems, 2004. 1st International Symposium on, 20-22 Sept. 2004 Page(s):324 – 328

[HIP Mika] Mika Ratola, "Which Layer for Mobility? – Comparing Mobile IPv6, HIP and SCTP"

[HIP MM] T. Henderson, "End-Host Mobility and Multi-homing with the Host Identity Protocol ", draft-ietf-hip-mm-04 (work in progress), June 2006

[HIP Reg] J. Laganier, T. Koponen, L. Eggert, "Host Identity Protocol (HIP) Registration Extension" draft-ietf-hip-registration-02 (work in progress), June 2006

[HIP RVS] J. Laganier, L. Eggert, "Host Identity Protocol (HIP) Rendezvous Extension", draft-ietf-hip-rvs-05 (work in progress), June 2006

[HIP4Internet] "HIP for inter.net Project", http://hip4inter.net/

[I3] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana. "Internet Indirection Infrastructure". In Proc. ACM SIGCOMM 2002, Aug. 2002. Pittsburgh, PA, USA.

[InfraHIP] "Infrastructure for HIP (InfraHIP)", http://infrahip.hiit.fi/

[IPV6MHS] M. Bagnulo, "Updating RFC 3484 for multihoming support," draftbagnulo-ipv6-rfc3484-update-00.txt (work in progress), Dec. 2005.

[LOCSPLIT] B. Aboba, "IAB Considerations for the Split of Identifiers and Locators," draft-iab-id-locsplit-00.txt (work in progress), IAB, Mar. 2004.

[MIP6LPPS] R. Koodli. "IP Address Location Privacy and Mobile IPv6: Problem Statement", draft-ietf-mip6-location-privacy-ps-04.txt, October 2006.

[NETLMM] G. Giaretta et. al. (NetLMM Design Team), “The NetLMM Protocol”, draft-giaretta-netlmm-dt-protocol-02.txt (work in progress), IETF, October 2006.

[NLMMGL] J. Kempf, “Goals for Network-based Localized Mobility Management (NetLMM)”, draft-ietf-netlmm-nohost-req-05.txt (work in progress), IETF, October 2006

[NLMMNH] J. Kempf, “Problem Statement for Network-based Localized Mobility Management”, draft-ietf-netlmm-nohost-ps-05.txt (work in progress), IETF, September 2006.

[OCALA] Overlay Convergence Architecture for Legacy Applications. http://ocala.cs.berkeley.edu/

[OpenHIP] OpenHIP project, http://www.openhip.org/

[PANA] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H. and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", draft-ietf-pana-pana-07 (work in progress), December 2004.

[PMIP6] S. Gundavelli, K. Leung, and V: Devarapalli, “Proxy Mobile Ipv6”, internet draft, working in process, draft-sgundave-mip6-proxymip6-00, Oct. 2006.

[RFC2136] P. Vixie, S. Thomson, and Y. Rekhter, “Dynamic Updates in the Domain Name System (DNS UPDATE),” RFC 2136, IETF, Apr. 1997. [Online]. Available: http://www.ietf.org/rfc/rfc2136.txt

[RFC2401] S. Kent and R. Atkinson, “Security Architecture for the Internet Protocol,” RFC 2401, IETF, Nov. 1998. [Online]. Available: http://www.ietf.org/rfc/2401.txt

[RFC2402] S. Kenta and R. Atkinson, “IP Authentication Header,” RFC 2402, IETF, Nov. 1998. [Online]. Available: http://www.ietf.org/rfc/2402.txt

[RFC2406] S. Kent and R. Atkinson, “IP Encapsulating Security Payload (ESP),” RFC 2406, IETF, Nov. 1998. [Online]. Available: http://www.ietf.org/rfc/2406.txt

[RFC2460] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) Specification”, RFC 2460, December 1998.

[RFC2462] S. Thomson and T. Narten, “IPv6 Stateless Address Autoconfiguration,” RFC 2462, IETF, Dec. 1998. [Online]. Available: http://www.ietf.org/rfc/rfc2462.txt

[RFC3315] R. Droms, J. Bound, and B. Volz, “IPv6 Dynamic Host Configuration Protocol for IPv6 (DHCPv6),” RFC 3315, IETF, July 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3315.txt

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, February 2003

[RFC3775] D. Johnson, C. Perkins and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.

[RFC3776] J. Arkko, V. Devarapalli and F. Dupont, “Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents”, RFC 3776, June 2004.

[RFC3972] T. Aura, “Cryptographically Generated Addresses (CGA),” RFC 3972, IETF, Mar. 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3972.txt

[ROAM] Sehlly Q. Zhuang, Kevin Lai, Ion Stoica, Randy H. Katz, and Scott Shenker. “Host mobility using an internet indirection infrastructure”. In First Internet Conference on Mobile Systems, Applications, and Services (ACM/USENIX Mobisys), May 2003.

[SHIM6] E. Nordmark and M. Bagnulo, “Level 3 multihoming shim protocol,” Internet Draft (work in progress), IETF, May 2006. [Online]. Available: http://www.ietf.org/internet-drafts/draft-ietf-shim6-proto-05.txt

[SHIM6FL] J. Arkko and I. Beijnum, “Failure Detection and Locator Pair Exploration Protocol for IPv6 Multihoming,” draft-ietf-shim6-failuredetection-03.txt (work in progress), Dec. 2005.

[SI3] D. Adkins, K. Lakshminarayanan, A. Perrig, and I. Stoica. “Towards a more functional and secure network infrastructure”. Technical Report UCB/CSD-03-1242, Computer Science Division (EECS), University of California, Berkeley, 2003