Institut Eurécom [1]
Department of Mobile Communications
2229, route des Crêtes
B.P. 193
06904 Sophia-Antipolis
FRANCE

Research Report RR-06-174

State of the art of Mobility Protocols

November 6, 2006

Thesis Advisor: Prof. Christian BONNET
PhD Student: Huu Nghia NGUYEN

Tel: (+33) 04.93.00.82.38
Fax: (+33) 04.93.00.26.27
Email : {Huu-Nghia.Nguyen,Christian.Bonnet}@eurecom.fr

[1] Institut Eurécom's research is partially supported by its industrial members: BMW, Bouygues Télécom, Cisco Systems, France Télécom , Hitachi Europe, SFR, Sharp, STMicroelectronics, Swisscom, Thales

Abstract

Beyond 3G (B3G) environments typically consist of multi-homed mobile terminals and wireless overlay networks in heterogeneous access technologies and aim at the Always Best Connected provision. In such B3G environments, the mobility feature and multi-homing feature are inseparable. Both multi-homing and mobility have to cope with the same problem of multiple IP addresses. However the former works on multiple simultaneous IP addresses and the later works on dynamic IP addresses.

In this report, we will go through the state of the art of host mobility management and the trend of the future host mobility management protocols including many works in progress of IETF which support mobility and/or multi-homing features (MMIP, HIP, mSCTP, MOBIKE, NETLMM). This work will later be used to propose the new B3G mobile internet architecture with a *Always Best Connected* provision.

Keywords: *Always Best Connected, Architecture, B3G, Multi-homing, Mobility, MIP, HIP, mSCTP, LIN6, HAWAII, CELLULAR IP, VNAT, ROAM, Migrate TCP, TCP Splice,*

Table of contents

Table of figures

Introduction

Host mobility can be implemented in different layers of the Internet architecture but this report focuses on layers three, three and a half, and four. In this document the term mobility will only refer to host mobility.

In practice, because of the diversity of mobility management protocols, it's really hard to propose a unique host mobility abstraction models. However, in 1] [2] authors showed that a mobility solution must resolve the compromise between the location and identity roles of IP address.

Traditionally, the classification of host mobility solution based on scopes (global mobility management or localized mobility management) or on layers (layer 2, 3, 3.5, 4...). Most of layer 3 or layer 3.5 solutions (MIP/HMIP/FMIP/MMIP, LIN6, VIP, VNAT, HIP, and even the new NETLMM) consider the host mobility an address translation problem. Some of them (HAWAII, Cellular IP, Multicast, ROAM) solve the mobility problem by modifying routing protocols or by constructing an overlay network.

For the address translation, the separation between the identifier and the locator can be done by answering the following questions:

- How and where to maintain the dynamic association between endpoints and locators? This may be perceived as a problem of database maintenance. The database may be maintained in a centralized fashion, wherein a single entity maintains the association and updates are sent to it by the mobile host or in a distributed fashion, wherein there are a number of entities that store the associations.
- Where to do the remapping between the endpoint and locator, in case of a change in association? By remapping, we mean associate a new locator with the endpoint. Some candidates are: the source, the "home" location of the host that has moved and any router (say, between the source and the destination) in the network.

Layer 4 solutions, however, try to solve the compromise by changing the behavior of the transport layer and allow the transport layer to learn the changes of IP addresses and keep the continuity of the on-going session with the help of a proxy (MSOCKS) or by updating the session state in an end-to-end manner (Migrate TCP and mSCTP).

1. Terminologies

We define the followings concepts for the rest of the document. Some definitions are based on [3]:

- ***Mobile Node (MN):*** a node that can change its point of attachment from one link to another, while still being reachable without breaking on-going sessions. (In a B3G environment, we should allow that the rate of change of location is even faster than the time it takes for the mobile routing protocols to take into account the mobile host's new location).

- ***Correspondent Node (CN):*** a peer node with which a mobile node is communicating. The correspondent node may be either mobile or stationary.
- ***Home Network:*** sub-network(s) for mobile hosts within each administrative domain. Stationary hosts always remain connected to their home network, while mobile hosts sometimes may not be found at their home networks.
- ***Foreign Network:*** any connected segment of an Internet, other than the home network of a mobile host, to which the mobile host is allowed to attach, is referred to as a *foreign network.*
- ***End point identifier (EID):*** each host has its own EID which can be used to address data packets; EIDs are mapped to current network attachment points by the routing infrastructure or by the end hosts. In general, the EID size is 32 or 128 bits which fits an IP address field in an IP packet.
- ***Locator:*** a routable IP address, assigned to a mobile node, used as a temporary address of the mobile node. Standard IP routing mechanisms will deliver packets destined for a mobile node's locator to its foreign network. Mobile nodes can have multiple locators when the terminal is multi-homed.

2. Overview of host mobility protocols

2.1. Abstraction primitives for layer 3 and 3.5

In this section, we present essential abstraction primitives for the layer 3/3.5 mobility deployment as following

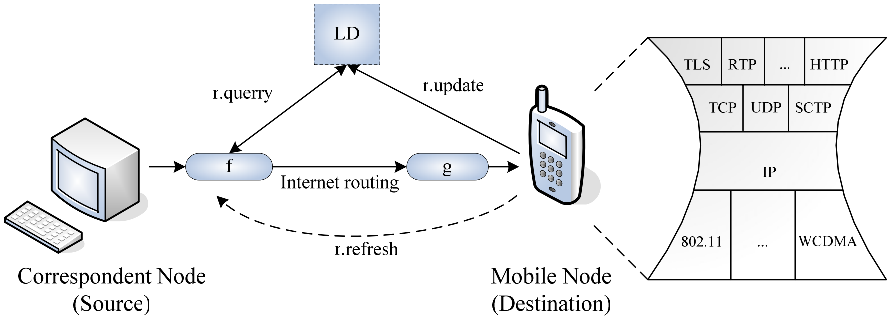


Figure 1. Abstraction model with f, g, LD, r primitives

From the deployment view point, a mobility solution using address translation can be represented by a 4-tuples *(f, g, LD, r)* where:

- Location Directory *LD* stores and provides the up-to-date information to find the location of the mobile node. It can be implemented as a routing table, as associations between the end point identifier and the locator of the mobile node.
- The function *f* is responsible for readdressing. It allows to resolve the locator from the end point identifier and to reconstruct the IP packet with the up-to-date routable locator. In practice, it can be implemented with encapsulation techniques (IP in IP, Minimal Encapsulation, Generic Routing Encapsulation, etc)
- The function *g* inverting the readdressing operation. It allows to resolve the end point identifier from the locator and to reconstruct the original IP packet.
- A set of registration messages *r* is used for the remote redirection. Those messages from the MN will allow the LD to maintain up-to-date values for the association.

If the routing protocol is modified to support mobility, we define the indirection function which can be considered as a combination of the *f* and *g* functions and the IP routing mechanism. In this case, the mobility solution then can be represented by a 3-tuples *(i, LD, r)*. The abstraction model then represents the idea of indirection philosophy.

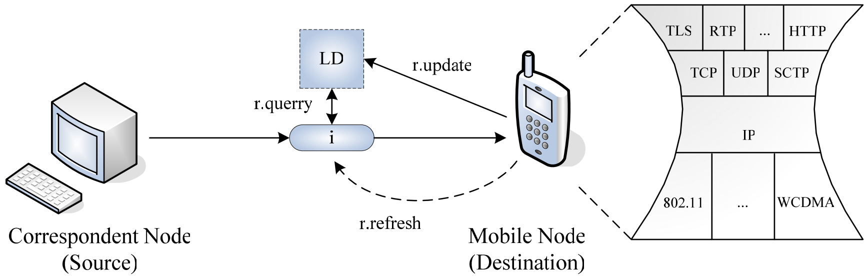


Figure 2. Abstraction model with i, LD, r primitives

Protocol		*i*		*LD*	Impact on Routing Infrastructure	Additional component in infrastructure	Impact on the CN
		f	*g*				
Routing Infra. (i, LD, r)	HAWAII	Routers		Routers	Yes	No	No
	Cellular IP	Routers		Routers	Yes	No	No
	ROAM/i3	i3 servers		i3 servers	No	Yes	May be
	Multicast	Routers		Routers	Yes	No	No
Address trans. (f, g, LD, r)	Mobile IP	HA	FA/MN	BC at HA	No	HA, FA	No
	LIN6	CN	MN	MA	No	MA	Yes
	VIP	CN	MN	DDNS	No	(DDNS)	Yes
	VNAT	CN	MN	CN	No	No	Yes
	HIP	CN	MN	DDNS	No	DDNS (require extensions)	Yes

2.2. Mobile IP

Mobile IP is the Internet Engineering Task Force (IETF) standard for supporting host mobility on the Internet. Mobile IP does not require a redesign of the IP routing infrastructure. The protocol offers transparent movement of a mobile node to transport and higher-level protocols and applications suitable for both homogenous and heterogeneous media [4]. MIPv4 is documented in RFC 3344: “IP Mobility Support in IPv4”. MIPv6 Relies on IPv6 and documented in the RFC 3375: “Mobility Support in IPv6” [3].

The basic principle of this approach is the use of a couple of addresses to identify the mobile node and manage its movements. Each mobile node is always identified by its Home Address (HoA), regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a Care-of Address (CoA), which provides information about the mobile node’s current location [5]. Correspondent node see only the host’s home address and have no indication that the host is mobile, or what its current network attachment point might be.

Mobile IP introduces Home Agents (HA) in the home network and Foreign Agents (FA) in foreign networks. The HA stores all MNs bindings in special table termed Binding Cache (BC) which plays the role of LD. It is used to locate the mobile at each moment. As for FA (only in MIPv4), it is a special router that manages the mobile node connected to the foreign link. When a correspondent send packets to the MN it uses its home address, located in the HA sub network, so this one will be able to intercept and encapsulate MN destination packets towards the suitable FA or access router, in an IPv4/IPv6 tunnel. The MN must inform its HA of this new address by the registration process, the HA can then create a binding between the MN home address and its new CoA. FA is not used in MIPv6, instead, it uses Collocated CoA and requires binding updates to be done using IPsec, so having IPsec is mandatory.

MIPv6 can use route optimization (RO mode) as well as tunneling (MIPv4 BT mode). The home agent is still needed and acts as a single point of failure, since it is needed for initialization of a new connection. Therefore any mobile host is not connectible if there is a problem connecting to the home network of the host.

Freeware implementation of Mobile IP can be found at www.monarch.cs.cmu.edu or www.cs.pdx.edu/research/SMN [6]

2.3. VIP

Virtual Internet Protocol (VIP) is a virtual IP layer that applies the principle of virtual addressing to Internet naming proposed by Teraoka, Yokote, and Tokoro in [7][8]. VIP extends the IP protocol to consider two distinct addresses: a virtual network address (VN) and a physical network address (PN). The virtual network address serves as an endpoint identifier, while the physical network address serves as a locator (a traditional IP address).

VIP's goal is to support mobility in a way that is incrementally deployable and that requires little installation or configuration effort. The problematic binding between locators and identifiers of host should be eliminated so that transport and higher layers could refer to end-points with an immutable identifier. [9]

The basic idea of VIP is simple; the VIP framework identifies a machine by its unique Fully-Qualified Domain Name (FQDN, e.g., example.acm.org) rather than their IP addresses (e.g., 199.222.69.43), and the VIP layer on each machine maintains a mapping from FQDN to the physical IP addresses of peer machines so that it can direct messages addressed to an FQDN to that machine's current location – its current physical IP address. As IP addresses change due to migration, VIP updates this FQDN ↔ IP mapping using secure dynamic DNS. Because FQDNs do not change, communication transparently continues across physical IP address changes. The data delivery is insured by IP-in-IP encapsulation.

The VIP address is integrated into the system by a VIP layer that resides immediately above the IP layer. Layers above VIP see and work with virtual IP addresses, which are merely backwards-compatible synonyms for FQDNs, and layers below VIP see the physical IP addresses required to route packets to their intended destination. A separate userlevel daemon maintains these FQDN↔VIP address and VIP address↔IP address mappings and updates the latter using dynamic DNS.

Unfortunately, current applications use IP address as the basis for communication and the FQDN merely as a means of obtaining it. We maintain backwards compatibility by virtualizing IP through a layer of indirection. Thus each FQDN is mapped to a 32-bit token, which we call a virtual IP address that in turn maps to the physical IP address. We refer to the former as the VIP address or virtual IP address and the latter as the IP address or physical IP address.

To simplify reasoning about security, the system uses IPSec to encrypt and authenticate all VIP communication. Following our goal of minimal infrastructure, this security scheme uses a simple peer-to-peer "anonymous" key exchange protocol similar to the one used in SSH.

2.4. LIN6

LIN6 [10] is a host mobility protocol based on a new network architecture called LINA (Location Independent Network Architecture).

Based on the addressing architecture of IPv6, LIN6 conceptually divides the network address into the end point identifier and the locator. A 128bit-long IPv6 address is divided into two parts. The first half is called “locator” and the second half “identifier”. LIN6 uses the *Generalized identifier* and *ID-Embedded locator* concept of LINA communication model. The Mapping Agent (MA) is used to resolve the locator from the identifier

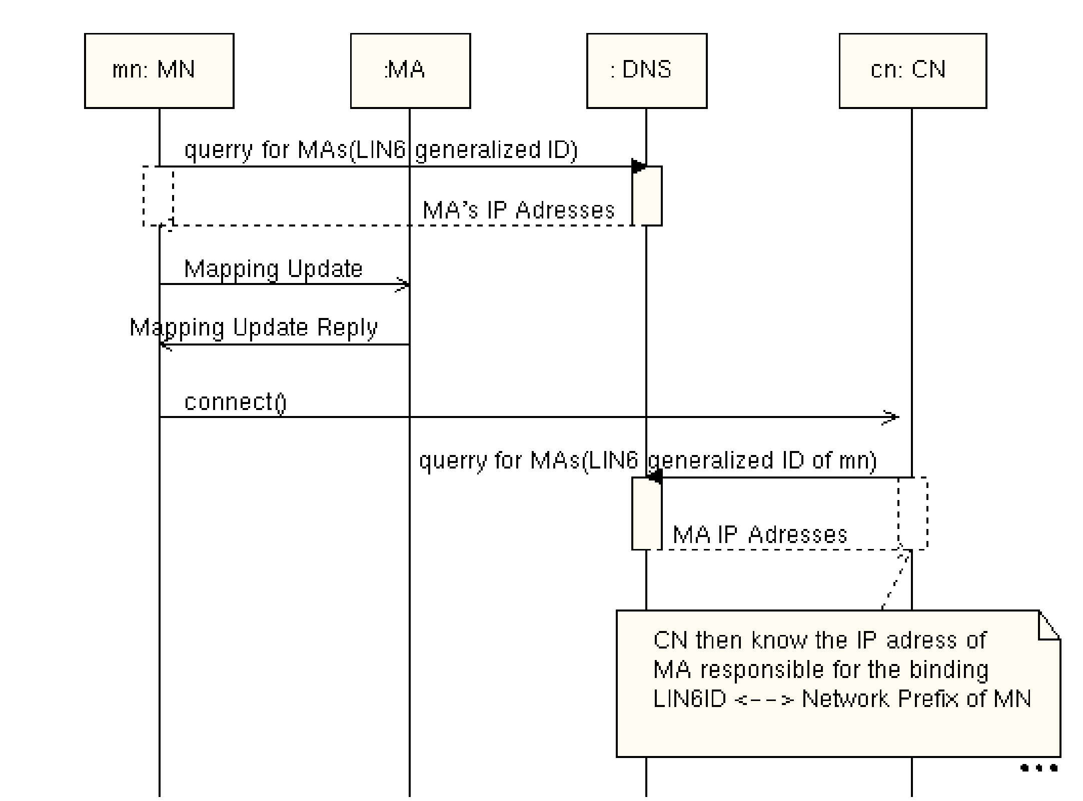

mn: MN
:MA
: DNS
cn: CN
querry for MAs(LIN6 generalized ID)
MA's IP Adresses
Mapping Update
Mapping Update Reply
connect()
querry for MAs(LIN6 generalized ID of mn)
MA IP Adresses
CN then know the IP adress of
MA responsible for the binding
LIN6ID <--> Network Prefix of MN
...

Figure 3. The mobile node finds its corresponding MA.

It supports two distinct handover mechanisms to accommodate various security needs. In the first mechanism, A Mapping Update operation is used to notify the MA and correspondents and IPsec is needed to protect against spoofing attack. In the second mechanism, A Mapping Refresh Request is sent to correspondent instead of Mapping Update. The CN then must query the MA. When received Destination Host Unreachable ICMP message, CN node must also query the MA.

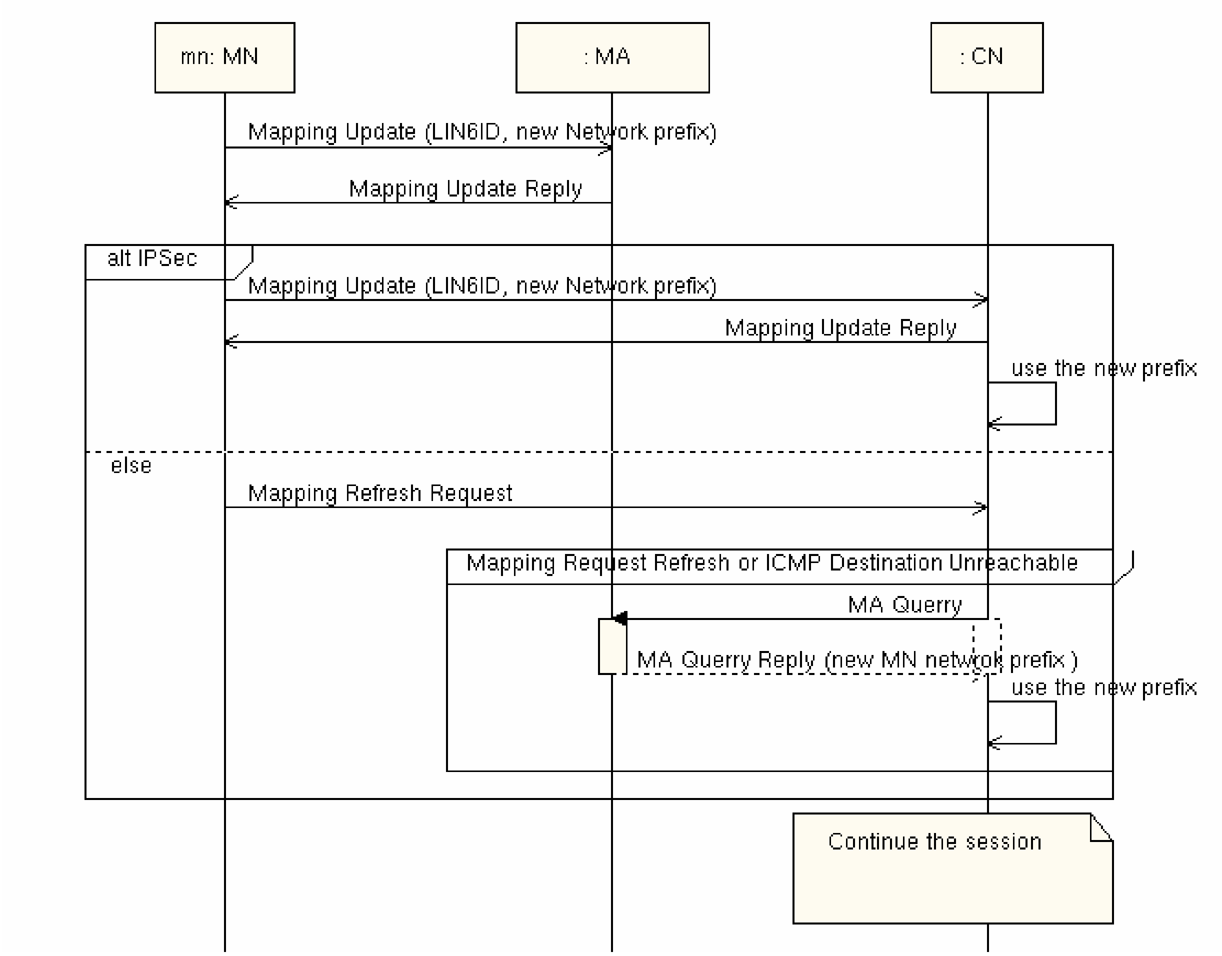

mn: MN
: MA
: CN
Mapping Update (LIN6ID, new Network prefix)
Mapping Update Reply
alt IPSec
Mapping Update (LIN6ID, new Network prefix)
Mapping Update Reply
use the new prefix
else
Mapping Refresh Request
Mapping Request Refresh or ICMP Destination Unreachable
MA Querry
MA Querry Reply (new MN netwrok prefix)
use the new prefix
Continue the session

Figure 4. Handover mechanism of LIN6

In the data plane, LIN6 uses the normal IPv6 header in which the LIN6 addresses are used in the source address field and the destination address field. Once the mapping is obtained from the Mapping Agent, LIN6 does not require additional processing by intermediate nodes for packet delivery and always guarantees end to end communication without using tunnels. In the control plane, it defines following messages:

- MA Query and MA Reply Messages
- Mapping Update, Mapping Reply Messages
- Mapping Refresh Message

However, the solution needs additional MA elements in the infrastructure. The detailed description of LIN6 can be found in http://doc.tm.uka.de/i-d/individual/teraoka/draft-teraoka-ipng-lin6.txt.gz .

2.5. HAWAII

The HAWAII [11] protocol is a micro-mobility protocol. It extends the concept of HA and FA in Mobile IP to Home Domain and Foreign Domain. HAWAII defaults to using

Mobile IP for macro-mobility. This combination of HAWAII for micro-mobility within a domain and Mobile IP for macro-mobility across domains provides for scalable and robust mobility across all levels.

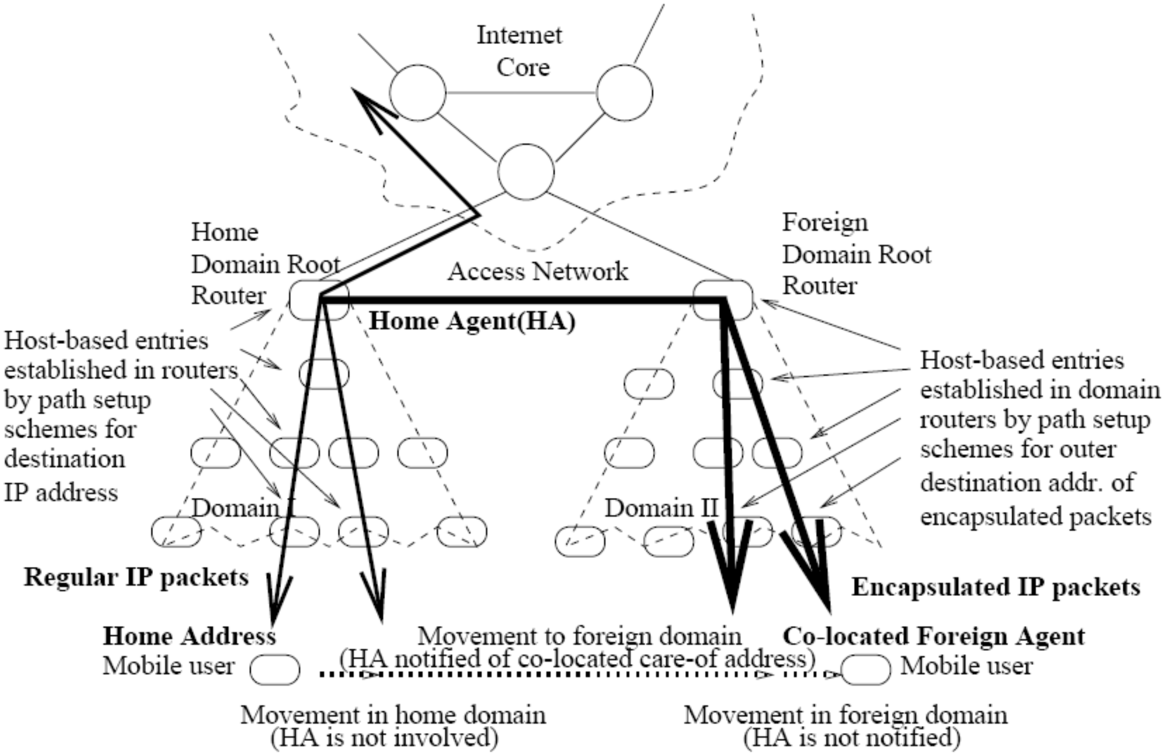

Internet Core
Home Domain Root Router
Access Network
Foreign Domain Root Router
Home Agent(HA)
Host-based entries established in routers by path setup schemes for destination IP address
Host-based entries established in domain routers by path setup schemes for outer destination addr. of encapsulated packets
Domain I
Domain II
Regular IP packets
Encapsulated IP packets
Home Address
Mobile user
Movement to foreign domain
(HA notified of co-located care-of address)
Co-located Foreign Agent
Mobile user
Movement in home domain
(HA is not involved)
Movement in foreign domain
(HA is not notified)

Figure 5. Hierarchy using domains with HAWAII

HAWAII uses specialized path setup schemes which install host-based forwarding entries in specific routers to support intra-domain micro-mobility. Mobile hosts retain their network address while moving within a domain. The HA and any corresponding hosts are unaware of the host's mobility within this domain. Routes to the mobile host are established by specialized path setup schemes that update the forwarding tables with host-based entries in selected routers in that domain. HAWAII path state is maintained in the routers as "soft state" by 3 types of messages for path setup: power-up, update and refresh. Whereas, it is reasonable to add host based route entries in wireless access networks, it is not scalable to add such routes in backbone networks;

A common approach for providing transparent mobility to correspondent hosts is to divide the network into hierarchies. HAWAII uses a similar strategy, segregating the network into a hierarchy of domains, loosely modeled on the autonomous system hierarchy used in the Internet.

2.6. CELLULAR IP

Cellular IP [12] [13] is a micro-mobility protocol that incorporates a number of important cellular system design principles and provides seamless mobility support in limited geographical areas, passive connectivity and paging. CIP is built on a foundation of IP forwarding, minimal signaling, and soft-state location management. It uses cache mechanisms: Routing Cache Maps and Paging Cache Map.

The learning feature of Ethernet switches is used for location management. So CIP can operate at layer two or three. It is similar to HAWAII in principle but differ from HAWAII in routing table update mechanism. There are no new explicit messages to maintain routing tables, it just observes the traffic. In case of no traffic, mobile host can explicitly send signalization messages (For example: an empty packet) to the gateway but not to the ancient as HAWAII. The gateway router broadcast beacons to allow CIP nodes to find the gateway. There are two types of handoff scheme: Hard handoff (break before make) and Semi-soft handoff. This solution is absolutely infrastructure based.

2.7. VNAT

By interposing a NAT between an end point and its network attachment point, NAT software can translate end point identifier into appropriate locator. This approach, termed Virtual Network Address Translation (VNAT), was proposed by Su and Nieh [14]

The VNAT architecture is based on the surprisingly simple idea of introducing a virtual address to identify a connection endpoint. In current IP networks, it is impossible to keep end-to-end transport connections alive when one or both connection endpoints move because physical network protocol endpoints are used by transport protocol to identify its connections. VNAT uses virtual addresses to break this tie between the transport protocol and network protocol by virtualizing the transport endpoint identification. Once the transport endpoint identification is made independent of network endpoint identification, the lifetime of a transport connection is no longer limited by changes in network endpoints.

The VNAT architecture can be decomposed into three components: Connection virtualization, Connection translation and Connection migration.

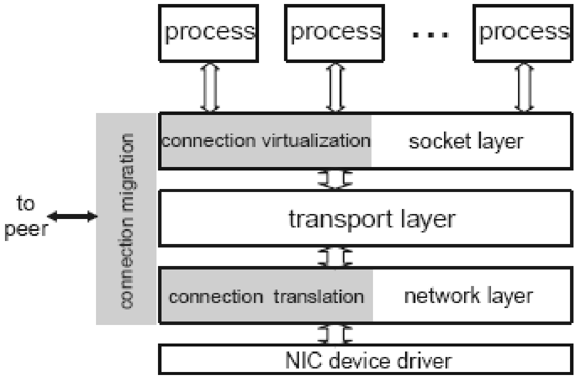


Figure 6. VNAT architecture preview

- The function of VNAT connection virtualization is to virtualize the end points used by the transport protocol to identify its end-to-end connections. With TCP/UDP, The EID is a virtual identification which is the combination of a network IP address and a transport port number.

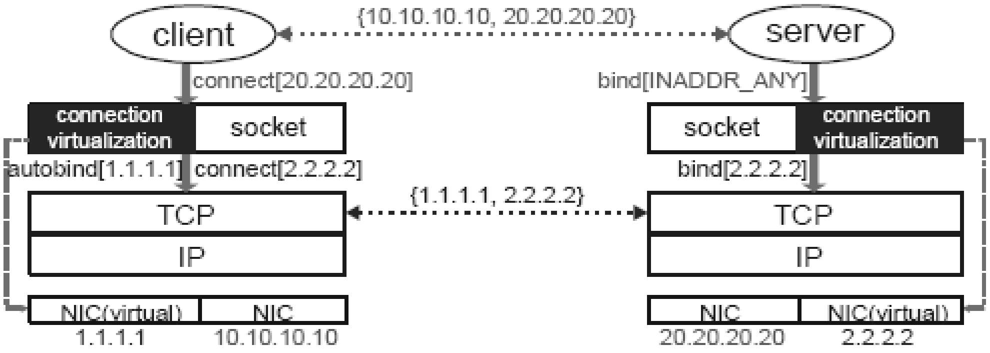


Figure 7. VNAT Connection virtualization

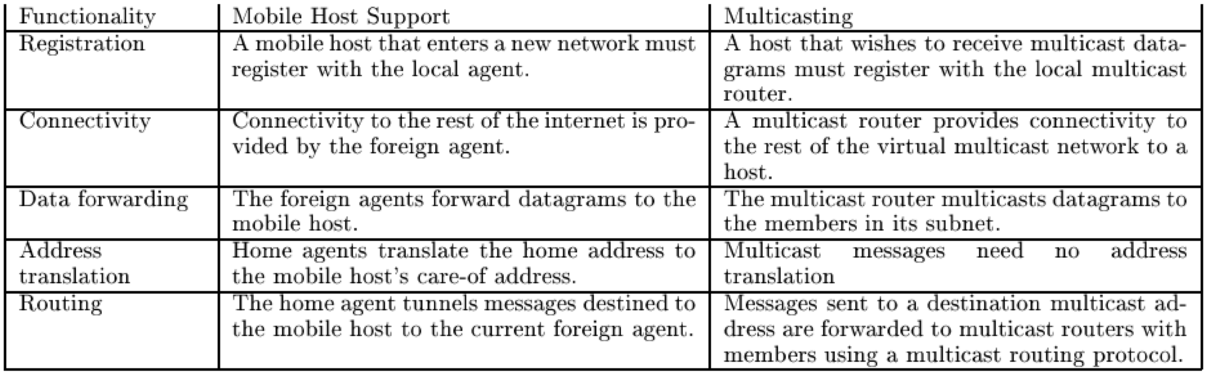
- VNAT connection translation makes it possible to communicate over virtual connections by translating a set of virtual addresses associated with virtual transport endpoints to and from a physical address associated with a physical network endpoint. VNAT connection virtualization creates the virtual addresses while VNAT connection translation maintains the proper association and mapping between the virtual addresses and the physical network addresses. VNAT connection translation is done using well-known Network Address Translation (NAT) technology.

- VNAT connection migration builds on VNAT connection virtualization and translation to provide the mechanisms necessary to actually move a connection from one machine to another.

VNAT can be incrementally deployed and operates entirely within communicating en systems without any reliance on third party services or proxies. Instead of using DNS to maintain the association between the endpoint identifier and the locator, it uses a “tracking” mechanism to preserve an end-to-end connection once it is established. However it doesn't mention about the situation when the connection is broken before the new state being updated and doesn't mention how the foreign address is obtained. Besides, it is required that both endpoint must support VNAT.

2.8. Multicast

Mysore and Bharghavan propose in [15] an approach to network-layer mobility that avoids the need for a home agent or a new protocol for binding updates entirely. They issue each mobile host a permanent Class D IP multicast address that serves as an end-point identifier. It places the burden of managing updates of end point bindings squarely on the routing infrastructure. The binding issue remains the same, however. The mobile node must send a binding update—it just takes the form of a multicast group join message. Similarly, the home agent functionality is replaced by whatever entity is in charge of multicast tree rendezvous.



Functionality	Mobile Host Support	Multicasting
Registration	A mobile host that enters a new network must register with the local agent.	A host that wishes to receive multicast datagrams must register with the local multicast router.
Connectivity	Connectivity to the rest of the internet is provided by the foreign agent.	A multicast router provides connectivity to the rest of the virtual multicast network to a host.
Data forwarding	The foreign agents forward datagrams to the mobile host.	The multicast router multicasts datagrams to the members in its subnet.
Address translation	Home agents translate the home address to the mobile host’s care-of address.	Multicast messages need no address translation
Routing	The home agent tunnels messages destined to the mobile host to the current foreign agent.	Messages sent to a destination multicast address are forwarded to multicast routers with members using a multicast routing protocol.

Figure 8. Comparison of Multicasting and Host Mobility Architectures

The multicast distribution tree for a host’s EIDs must be reconstructed each time a node moves, requiring an extremely agile and efficient tree-building protocol. It require a secure, robust, scalable, and efficient multicast infrastructure (not yet available in the Internet).

The starting premise of this work is that host mobility can be supported without making any special changes to an Internet multicasting architecture, though the state- of-the-art of IP multicasting is still inadequate for this purpose.

There are two key issues in this approach: (a) some routers will not forward packets that originate from hosts with addresses that do not correspond to the local network- such routers will drop packets from the mobile hosts, and (b) mobile hosts can- not addresses each other directly if the multicast routers use reverse path forwarding (RPF) (since RPF assumes that the source address is a unicast IP address in order for multicast routers to determine the shortest path to the source).

2.9. ROAM

ROAM (Robust Overlay Architecture for Mobility) [16] provides seamless mobility for Internet hosts. ROAM is built on top of the Internet Indirection Infrastructure (i3) [17] which offers a rendezvous-based communication abstraction. Instead of explicitly sending a packet to a destination, each packet is associated with an identifier; this identifier is then used by the receiver to obtain delivery of the packet. This level of indirection decouples the act of sending from the act of receiving, and allows to efficiently support a wide variety of fundamental communication services. The identifier defines an indirection point in i3, and is used by the receiver to obtain the packet.

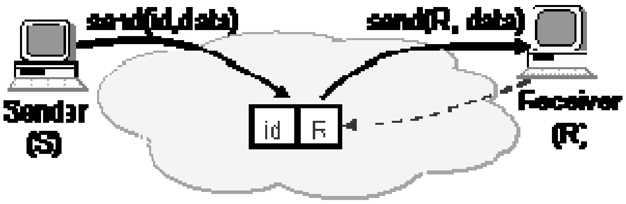


Figure 9. The indirection philosophy

To maintain this overlay network and to route packets in i3, the Chord lookup protocol is used. When a trigger (eid, locator) is inserted, it is stored at the i3 node responsible for the eid. When a packet is sent to id, it is routed by i3 to the node responsible for its eid; there it is matched against (any) triggers for that id and forwarded (using IP) to all hosts interested in packets sent to that identifier. Chord allows servers to leave and join dynamically, and it is highly robust against failures. End hosts must periodically refresh their triggers in i3. Hosts need only know one i3 node to use the i3 infrastructure. This can be done through a static configuration file, or by a DNS lookup assuming i3 is associated with a DNS domain name.

ROAM takes advantage of end-host ability to control the placement of indirection points in i3 to provide efficient routing, fast handoff, and preserve location privacy for mobile hosts. In addition, ROAM allows end hosts to move simultaneously, and is as robust as the underlying IP network to node failure. However it requires an Internet Indirection Infrastructure constructed as an overlay infrastructure over IP. The mobility solution is therefore absolutely an infrastructure based solution. For more information, please visite http://i3.cs.berkeley.edu/

2.10. Migrate TCP

Migrates TCP [18], proposed by Hari Balakrishnan and Alex Snoeren, introduces a new TCP option to support end-to-end connections between two applications. By adding a new option Migrate-Permitted in SYN segments which are exchanged at the initiation of a connection, a MN can later send a SYN packet as part of a previously established connection, rather than a request for a new connection. This Migrate option contains a token that identifies a previously established connection on the same destination. A drawback of Migrate TCP is that, it requires transport layer protocol changes which make it difficult to deploy.

2.11. TCP Splice

David A. Maltz and Pravin Bhagwat present in cite{MSOCKS:Maltz1998} an architecture called Transport Layer Mobility that allows mobile nodes to not only change their point of attachment to the Internet, but also to control which network interfaces are used for the different kinds of data leaving from and arriving at themobile node.

The transport layer mobility scheme uses split-connection proxy architecture and a new technique called TCP Splice that gives split-connection proxy systems the same end-to-end semantics as normal TCP connections.

The goal of a TCP Splice is to make it appear to the endpoints of two separate TCP connections that those two connections are, in fact, one. From the point-of-view of the endpoints, it should appear that they are directly connected by a single TCP connection with all the end-to-end properties of a normal TCP connection. The insight behind TCP Splice is simple: data can be lost in split connection proxy schemes because the proxy acknowledges the receipt of data to the correspondent host before receiving an acknowledgment (ACK) from the mobile node. Data which is ACK'd to the server but lost in transmission to the mobile node or mired in the kernel socket buffer of a broken connection, is lost forever.

When the mobile node changes its point of attachment, it sends a RECONNECT request to the proxy which will then detach the connection from the old address and reattach to the new address. TCP-Slice is in charge of retransmitting datagram in order and assure the TCP connection semantic. The solution has a limited scalability and performance and only provides client mobility.

This solution also supports feature by allow us to control over which interface of the mobile node, the data will move. However, connecting and reconnecting two connections, as proposed doing, normally risks the loss of any data in flight while the reconnection happens, which would break the end-to-end semantics of the logical mobile-to-server communication session.

3. Works in progress of IETF

3.1. HIP

3.1.1. General

The basis of HIP [19] [20] [21] [22], proposed by R. Moskowitz and P. Nikander, is the separation of host identity from host location so that a network host could be referred independent of its current location. HIP introduces a new Host Identity layer (layer 3.5) between the IP layer (layer 3) and the upper layers. In HIP, The host identifier is the public key of a public-private key pair associated to the host. A Host Identity Tag (HIT) is a 128-bit hash of the host's public key. The interface to the transport layer uses Host Identity Tags in place of IP addresses (as an EID), while the interface to the Internet layer uses conventional IP addresses (as a locator). The purpose of HIP is to support trust between systems, enhance mobility, and greatly reduce the DoS attacks.

The protocol is documented mainly in following drafts (However, up to the moment of writing this report, there is still no RFCs for HIP):

- Architecture - draft-ietf-hip-arch-02.txt
- Protocol - draft-ietf-hip-base-02.txt
- DNS Extensions - draft-ietf-hip-dns-01.txt
- Rendezvous Extension - draft-ietf-hip-rvs-01.txt
- End-Host Mobility and Multi-Homing - draft-ietf-hip-mm-01.txt

The following public HIP implementations are known:

- HIP4BSD (http://www.hip4inter.net)-- FreeBSD kernel modifications and user-space keying daemon;
- HIPL (http://infrahip.hiit.fi)-- Linux kernel implementation;
- OpenHIP (http://www.openhip.org)-- Linux kernel modifications and user-space keying daemon, plus a fully user-space Windows XP implementation;
- pyHIP (http://www.sharemation.com/adm01bass/)-- Fully user-space implementation written in python (no longer maintained).

3.1.2. Architecture

HIP is similar to MIPv6 in the sense that the main goal for both of them is to make mobility transparent to the applications. In HIP, the hosts are identified with public keys, not IP addresses. A typical host identity is a public cryptographic key of an asymmetric key-pair. Each host will have at least one HI that can either be public or anonymous. It is important to understand that the end-point names based on Host Identities are slightly different from interface names; a Host Identity can be simultaneously reachable through several interfaces.

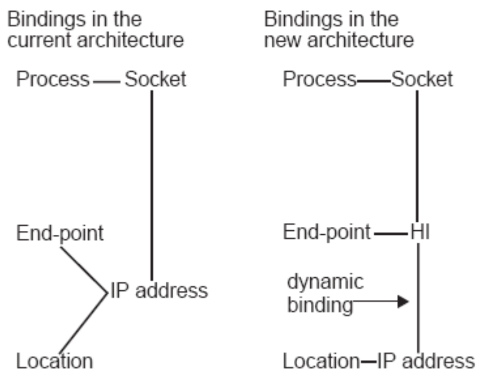


Figure 10. The difference between the bindings of the logical entities

It is possible that a single physical computer hosts several logical end-points. With HIP, each of these end-points would have a distinct Host Identity.

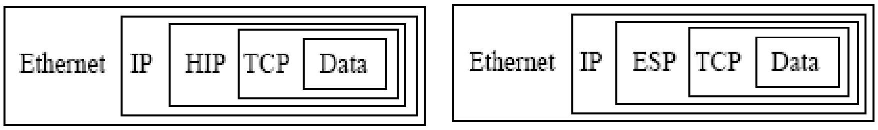
A HIP node stores in the DNS its Host Identity (HI, the public component of the node public-private key pair), Host Identity Tag (HIT, a truncated hash of its HI), and the Domain Name or IP addresses of its Rendezvous Servers (RVS) [21].

3.1.3. Protocol

By definition, the system initiating a HIP exchange is the Initiator, and the peer is the Responder. This distinction is forgotten once the base exchange completes, and either party can become the initiator in future communications.

3.1.3.1. Packet formats

HIP presents a new packet structure: The transport layer packet (e.g. TCP) must be enclosed with a HIP header, which contains the HIT. HIP could be carried out in every datagram throughout the connection but alternatively the HIP payload can be compressed into an ESP payload (in IPv6) after the HIP exchange. Thus, HIP packets are only needed to establish an authenticated connection. As mentioned above, the HIP protocol is used to authenticate the connection. In addition to authentication, the procedure establishes Security Associations for a secure connection with IPsec ESP.



Connection establishment

During connection

Figure 11. The HIP packet structure

All HIP packets start with a fixed header. A detail description of HIP header can be found in draft-ietf-hip-arch-02.txt

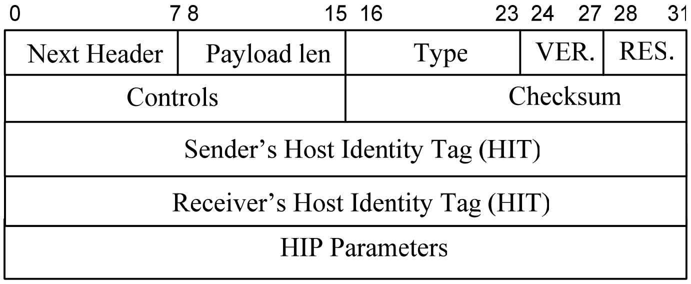


Figure 12. HIP header

3.1.3.2. HIP base exchange

HIP starts with one of the hosts looking up the HI and IP of the peer in the DNS: Upon query by an application for a FQDN → IP lookup, the resolver would then additionally perform an FQDN → HI lookup, and use it to construct the resulting HI →IP mapping. The host then sends an initial I1 message requesting a state to be established with the peer. Messages R1, I2 and R2 are exchanged successively in order to create an association.

The HIP base is protected with HIP Cookie Mechanism, Authenticated Diffie-Hellman protocol and HIP replay protection. The last three packets of the exchange, R1, I2, and R2, constitute a standard authenticated Diffie-Hellman key exchange for session key generation. During the Diffie-Hellman key exchange, a piece of keying material is generated. The HIP association keys are drawn from this keying material. If other cryptographic keys are needed, e.g., to be used with ESP, they are expected to be drawn from the same keying material.

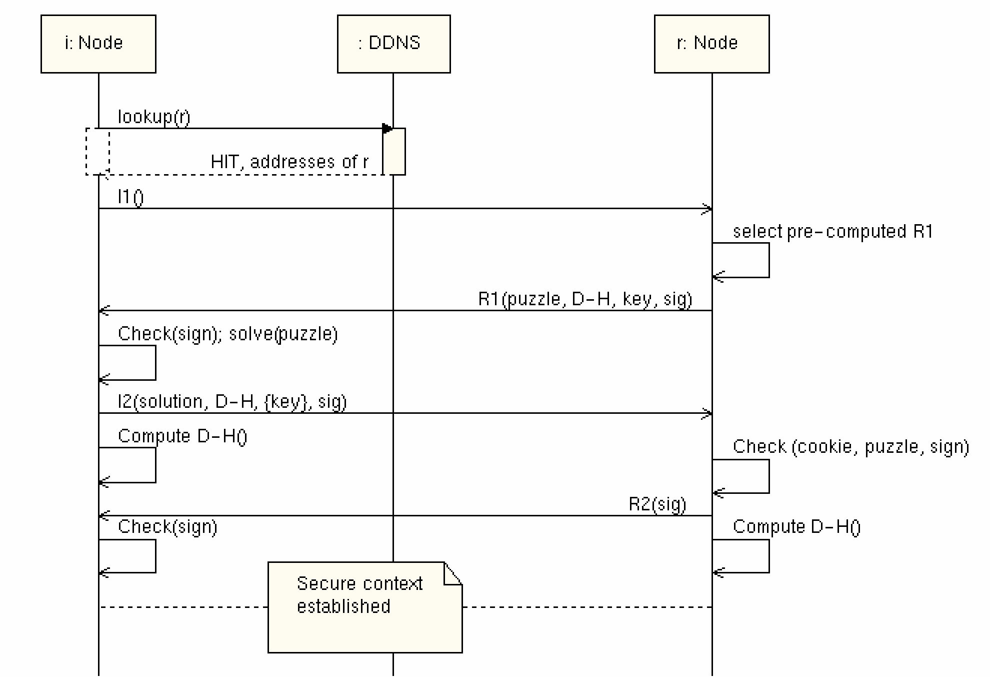


Figure 13. HIP Base Exchange

- **I1 packet** is sent by the initiator to see if the responder speaks HIP. The packet contains the HITs of the both parties.
- **R1 packet** is sent back as a reply by the responder. As the responder cannot yet trust the initiator, it initiates a three-way cookie exchange. Packet R1 holds the responders public Diffie-Hellman key, HI, and information about the supported ESP modes as well as a challenge. The impact of a DoS attack is minimized as the responder is the one giving the challenge.
- **I2 packet** contains the initiators public Diffie-Hellman key and a computed response to the challenge. The computation makes the DoS attack unprofitable for the initiator. The ESP options are also sent with the packet.
- **R2 packet** completes the handshake. The responder sends it if the initiators response to the challenge was correct. After the sending of the R2 packet, the ESP encrypted datagrams can be used to secure the whole connection.

3.1.3.3. Updating a HIP association

During the secured connection, mobility in HIP is quite straightforward. When one of the hosts changes its IP address, the new address needs to be updated with the peer. Only a simple signalling protocol (the HIP protocol discussed above) is needed to take care of the dynamic binding between the node's IP address and HI. When one of the communicating peers changes location, it simply sends a HIP readdress packet (indicates the following information: the new IP address, the SPI associated with new IP address, the address lifetime and whether the new address is a preferred address) through the secured ESP channel. This UPDATE packet is used for those and other similar purposes. The UPDATE mechanism has the following properties:

- UPDATE messages carry a monotonically increasing sequence number and are explicitly acknowledged by the peer. Lost UPDATEs or acknowledgments may be recovered via retransmission. Multiple
- UPDATE messages may be outstanding.
- UPDATE is protected by both HMAC and HIP_SIGNATURE parameters, since processing UPDATE signatures alone is a potential DoS attack against intermediate systems.

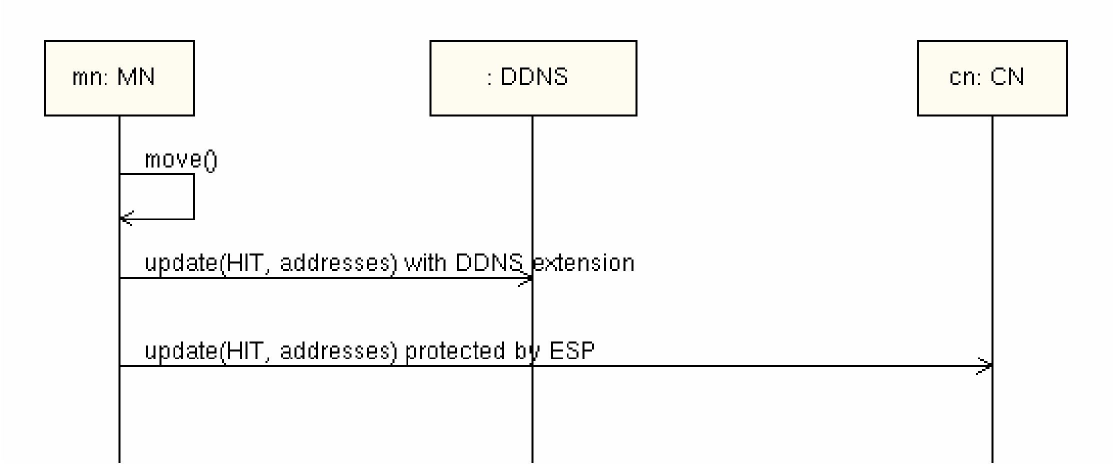

mn: MN
: DDNS
cn: CN
move()
update(HIT, addresses) with DDNS extension
update(HIT, addresses) protected by ESP

Figure 14. HIP Handover

3.1.3.4. Rendezvous mechanism

In order to start the HIP exchange, the initiator node has to know how to reach the mobile node. Although infrequently moving HIP nodes could use *Dynamic DNS* to update their reach-ability information in the DNS, an alternative to using DNS in this fashion is to use a piece of new static infrastructure to facilitate rendezvous between HIP nodes (Internet Indirection Infrastructure, for example). The mobile node keeps the rendezvous infrastructure continuously updated with its current IP address(es).

3.1.4. End-host multi-homing

A system is considered multi-homed if it has more than one globally routable IP address at the same time. HIP links IP addresses together, when multiple IP addresses correspond to the same Host Identity, and if one address becomes unusable, or a more preferred address becomes available, existing transport associations can easily be moved to another address.

3.2. mobile Stream Control Transmission Protocol (mSCTP)

The stream control transmission protocol (STCP) [23] [4] [24] [25] is being standardized by the IEFT as a reliable transport protocol to transport SS7signaling messages over IP networks. One of the core features of SCTP is supporting multi-homing. It has ability for a single SCTP endpoint to support multiple IP addresses. To support multi-homing, SCTP endpoints exchange lists of addresses during initiation of the connection. This multi-homing feature enables SCTP to be used for Internet mobility support without any support of network routers or special agents.

Due to its attractive features such as multi-streaming and multi-homing which promise load balancing ability [23][26], SCTP has received much attention from the network community, in terms of both research and development. An SCTP patch to the ns-2 simulator has been contributed by a group at the University of Delaware (http://pel.cis.udel.edu). The patch provides the main SCTP features specified in RFC 2960, including multi-streaming, multi-homing, congestion control, and chunk bundling. This patch, which is still being developed, made it possible for various research groups to evaluate the performance of SCTP using ns-2. The LKSCTP project is an open source implementation under GNU General Public License (GPL) to provide an SCTP module in a Linux kernel (http://lksctp.sourceforge.net/) [24]

The ADDIP extension in mobile Stream Control Transmission Protocol (mSCTP) enables an mSCTP endpoint to add a new IP address or delete an unnecessary IP address, and also to change the primary IP address used for the association during an on-going session. When one of the events such as ADD, DELETE, and CHANGE occurs, the mSCTP endpoint will notify the corresponding event to the remote endpoint by sending an SCTP ASCONF (Address Configuration Change) chunk.

3.2.1. Architecture

A single message transmitted over an SCTP association from the originating host to the destination host will be sent using a single destination IP address chosen from the set of destination IP addresses available for that association. The paths used by the IP packets across the network might be different depending on the destination IP address. If a message fails to reach its destination, SCTP may retransmit the message using a different destination IP address.

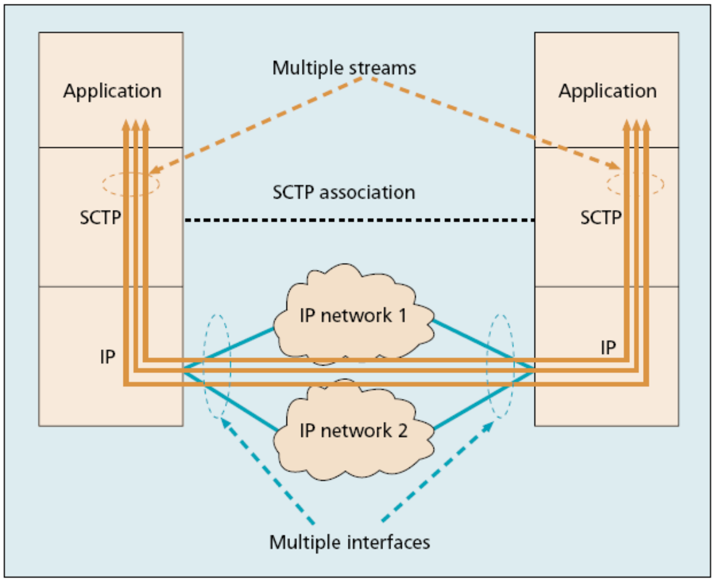


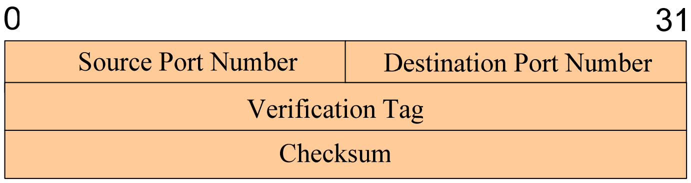
Figure 15. A schematic view of an SCTP association.

SCTP does not have any way to determine whether two paths share links and routers when traversing the network. The route of a path through a network can be static(example manual configuration) or dynamic(via routing protocols such as OSPF, BGP...). The route that a path takes through the network will change over time according to the routing protocols or routing decisions employed by the IP network layer.

3.2.2. Protocol

3.2.2.1. SCTP packet structure

An SCTP packet is composed of a 12 byte common header and chunks. In the header, a 32-bit checksum is used to detect transmission errors. SCTP packets with an invalid checksum are silently discarded. A randomly created 32 bit verification tag allows a receiver to verify that the SCTP packet belongs to the current association and not to an old one. The chunk on the other hand may contain either control information or user data. Chunks have variable length and there are currently 13 types of them in standard use.



0	31
Source Port Number	Destination Port Number
Verification Tag	
Checksum	

Figure 16. SCTP Common Header

Multiple chunks can be bundled into one SCTP packet up to the MTU size, except for the INIT, INIT ACK, and SHUTDOWN COMPLETE chunks. These chunks MUST NOT be bundled with any other chunk in a packet.

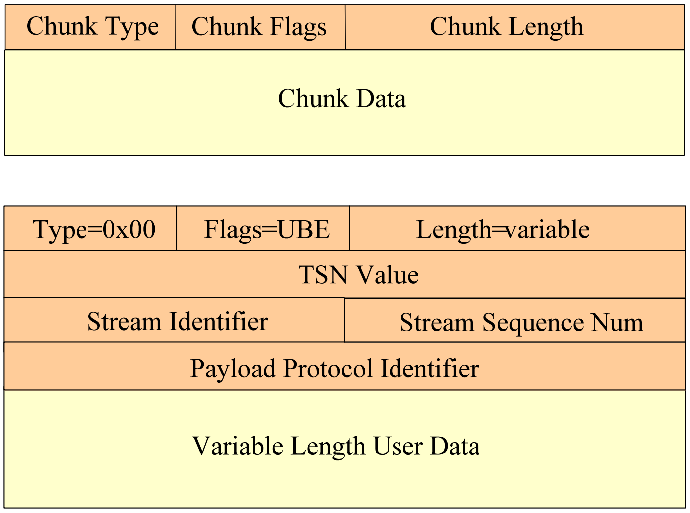


Figure 17. SCTP Chunk format & Data chunk format

The ADDIP extension defines two new chunk types that will be used to transfer the control information reliably. They are Address Configuration Change Chunk (ASCONF) and Address Configuration Acknowledgment (ASCONF-ACK).

It should be noted that the ASCONF Chunk format requires the receiver to report to the sender if it does not understand the ASCONF Chunk. This chunk is used to communicate to the remote endpoint one of the configuration change requests that MUST be acknowledged. The information carried in the ASCONF Chunk uses the form of a Type- Length-Value (TLV) in RFC2960 [27], for all variable parameters. This chunk MUST be sent in an authenticated way by using the mechanism defined in SCTP-AUTH [28]. If this chunk is received unauthenticated it MUST be silently discarded as described in SCTP-AUTH.

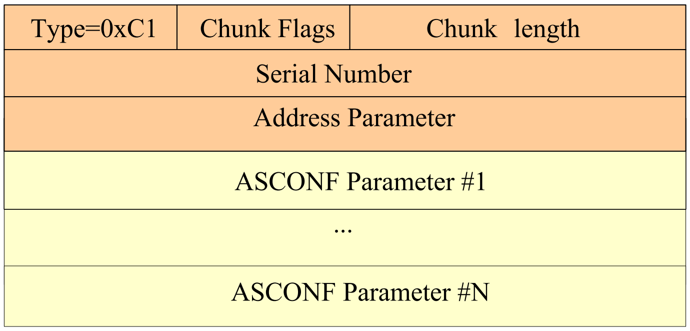
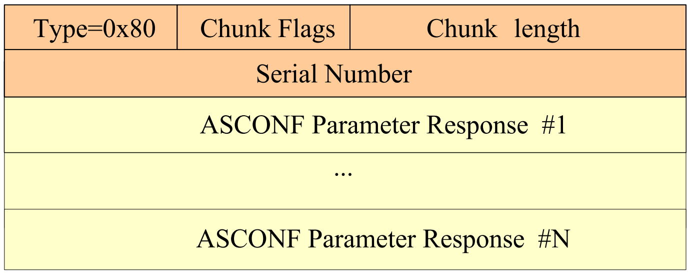


Figure 18. ASCONF Chunk format

Address Configuration Acknowledgment Chunk (ASCONF-ACK) is used by the receiver of an ASCONF Chunk to acknowledge the reception. It carries zero or more results for any ASCONF Parameters that were processed by the receiver.



Type=0x80	Chunk Flags	Chunk length
Serial Number		
ASCONF Parameter Response #1		
...		
ASCONF Parameter Response #N		

Figure 19. ASCONF-ACK Chunk format

The seven new parameters added follow the format defined in section 3.2.1 of RFC2960 [27]: Set Primary Address, Adaptation Layer Indication, Supported Extensions, Add IP Address, Delete IP Address, Error Cause Indication, Success Indication.

3.2.2.2. Association establishment

The association establishment in SCTP (therefore in mSCTP) uses the four-way handshake. The passive side is called a server and the other is a client. The handshake procedure is as follows: First, the server receives an INIT chunk. Using its data, the server generates a secure hash of these values and a secret key. These values along with a MAC are put into a COOKIE, and returned in an INIT-ACK chunk. The client using the received COOKIE assembles an COOKIE-ECHO chunk and returns it to the server. Finally, the server verifies with the MAC, that the COOKIE is the same as it sent, and replies with a COOKIE-ACK chunk. Now the association is established. When one of the communicating parties wants to end the association, it can be done in two ways: Either by graceful shutdown, ensuring that no data is lost, or hard termination (abort), not taking care of the peer. Unlike TCP, when either endpoint performs a shutdown, both of the endpoints stop accepting data.

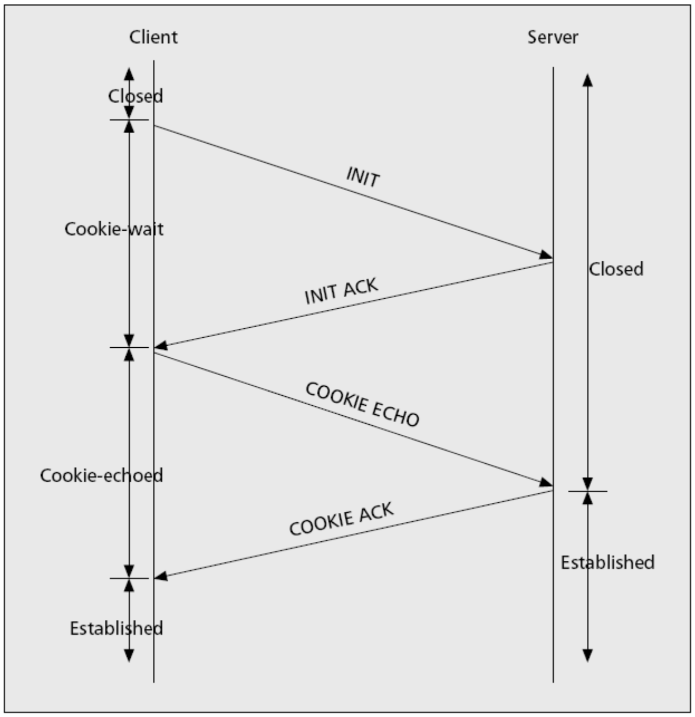


Figure 20. SCTP association setup message sequence.

During association startup, a list of transport addresses (i.e. IP address-port -pairs) is provided between the communicating entities. These addresses are used as the endpoints of different streams. SCTP regards each IP address of its peer as one "transmission path" towards this endpoint. The association spans transfers over all of the possible source/destination combinations. Also one of the addresses is selected as initial primary path, which may be changed later if needed.

3.2.2.3. Handover

The mSCTP handover needs to be triggered by the mobile node because only the mobile node knows the movement of itself and the signal strength from the old and new ARs.

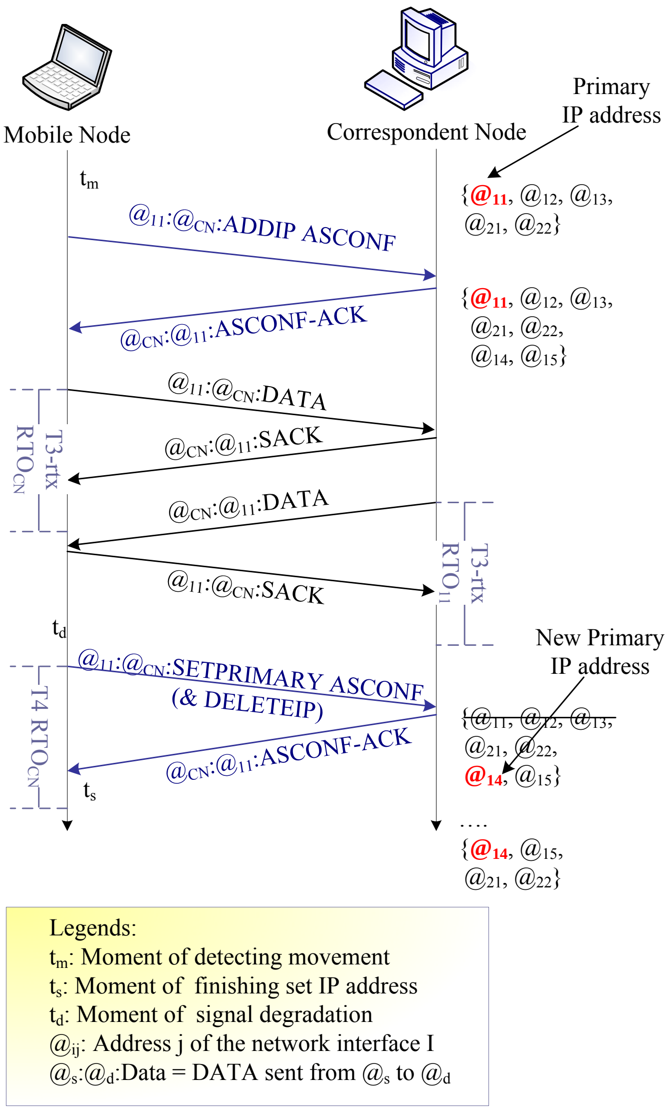


Figure 21.A mSCTP handover scenario

A mSCTP handover scenario will have the following steps:

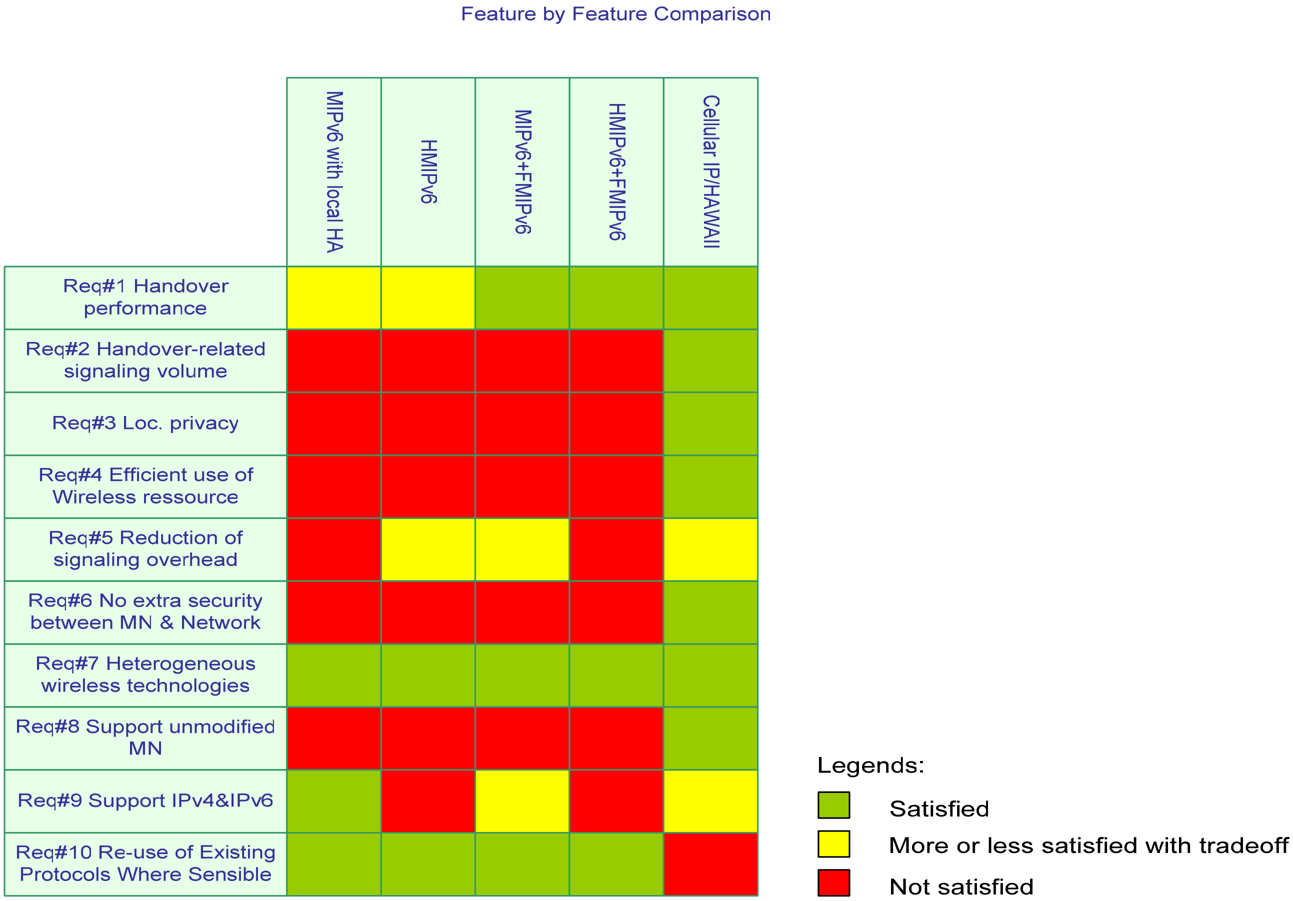
- Obtaining an IP address for a new location
- Adding the new IP address to the SCTP association
- Changing the Primary IP address
- Deleting the old IP address from the SCTP association

3.3. NetLMM

A Network-based Localized Mobility Management (netlmm) BOF was established at the 63rd IETF(Paris) in August 2005. Localized Mobility Management is a generic term for protocols dealing with IP mobility management confined within the access network. The

Localized mobility management signaling is not routed outside the access network, although a handover may trigger Global Mobility Management signaling. Localized mobility management protocols exploit the locality of movement by confining movement related changes to the access network. The LMM addresses at the 3 following problems: Update latency, Signaling overhead and Location privacy.

The document [29] develops more detailed requirements for a localized mobility management protocol (There are 10 requirements at the moment of writing this report). The analysis reveals that none of the existing protocol can satisfy all the requirement of Localized Mobility Management. IETF therefore recommended a network-based approach to localized mobility management called NetLMM [29][30][31][32].



Feature by Feature Comparison

	MIPv6 with local HA	HMIPv6	MIPv6+FMIPv6	HMIPv6+FMIPv6	Cellular IP/HAWAII
Req#1 Handover performance	🟨	🟨	🟩	🟩	🟩
Req#2 Handover-related signaling volume	🟥	🟥	🟥	🟥	🟩
Req#3 Loc. privacy	🟥	🟥	🟥	🟥	🟩
Req#4 Efficient use of Wireless ressource	🟥	🟥	🟥	🟥	🟩
Req#5 Reduction of signaling overhead	🟥	🟨	🟨	🟥	🟨
Req#6 No extra security between MN & Network	🟥	🟥	🟥	🟥	🟩
Req#7 Heterogeneous wireless technologies	🟩	🟩	🟩	🟩	🟩
Req#8 Support unmodified MN	🟥	🟥	🟥	🟥	🟩
Req#9 Support IPv4&IPv6	🟩	🟥	🟨	🟥	🟨
Req#10 Re-use of Existing Protocols Where Sensible	🟩	🟩	🟩	🟩	🟥

Legends:

🟩 Satisfied

🟨 More or less satisfied with tradeoff

🟥 Not satisfied

Figure 22. Feature by Feature comparison of different LMM solutions

The analysis is based on some personal estimation and the document [29] which is just a qualitative estimation. There are only 3 possible value for each feature {Not satisfied, Partial, Satisfied}.

The NetLMM (edge-based LMM) protocol is principally based on an assumption of unmodified MN. It comprises two parties. The first part defines the interface between the MN and the AR and the second part defines the interface between the AR and the LMA. The interface between the MN and the AR can be realized with DNA, NDP and SEND for Stateless address auconfiguration or with the help of DHCP for Stateful address configuration. The interface between the AR and the LMA are still evolving. The simplest version is EMP (Edge Mobility Protocol). Recently a new version for this interface is proposed by Giaretta in the draft-giaretta-netlmm-dt-protocol-00.txt [31].

The NetLMM addressing mechanism is CGA (Cryptographically Generated Address) which provides a mean to secure the mobility. The NetLMM protocol only concentrates on the control plane between entities in the NetLMM domain and not the data plane. There is something missing: The data plane is supposed to use a tunneling mechanism (IP in IP, GRE, MPLS). The interaction between the LMM and GMM in the control plane is still not defined. It supports a one-to-many relation between the MNID and the locators. However, it doesn't mention about the use of simultaneous locators.

3.3.1. Architecture

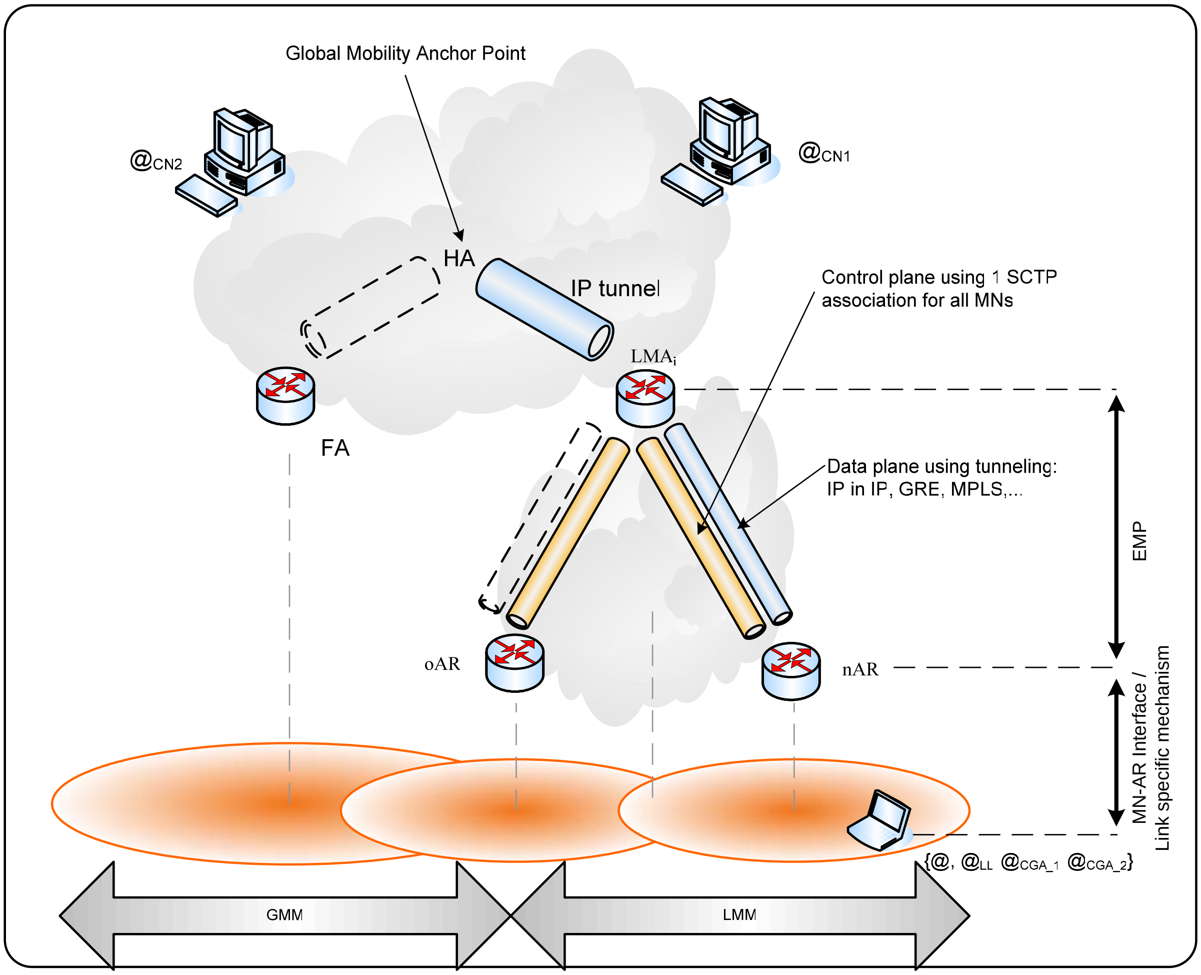


Figure 23. Architecture for netLMM solution using EMP

3.3.1.1. MN-AR Interface

The MN-AR NetLMM interface is used between a MN node and an AR of a NetLMM domain. In the absence of link-layer specific mechanism, it allows the AR to detect the network attachment of a MN and update routing at the LMA so that the MN stays reachable when it roams across the NetLMM domain. The draft draft-ietf-netlmm-mn-ar-if [32] specifies such an IP layer interface between mobile nodes (MN) and access routers (AR) of a network-based localized mobility. It is required that no NetLMM specific software support is present on MNs. The IP layer MN-AR interface described in this document fulfills these requirements by using the SEND public key as the MN identifier, while being solely based on standard track IPv6 protocols (DNA and SEND) implemented by non-NetLMM MNs.

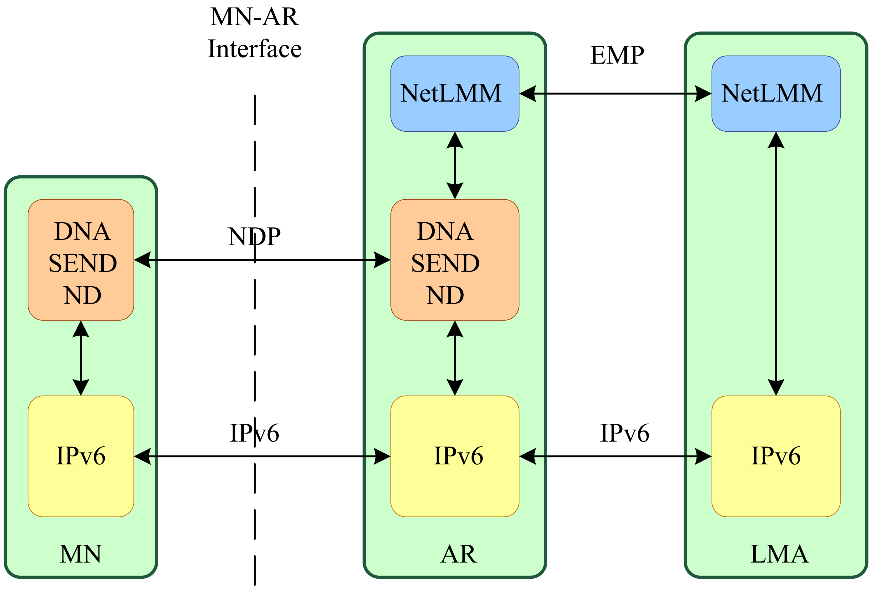


Figure 24. Protocol stack for netLMM solution

The interface MN-AR supports the following scenarios

- MN powers on in a NetLMM domain
- First attachment of MN moving into a NetLMM domain
- MN handovers in a NetLMM-domain
- MN configuring additional CGAs
- MN configuring CGA that is in use by another MN in the NETLMM domain
- MN un-configures CGAs, powers off, crash or leave the domain

3.3.1.2. AR-LMA Interface (EMP)

The interface between LMA and the AR can be EMP. EMP only defines the control plane. The data plane is supposed to use any available tunneling method specified in the HELLO message.

EMP uses a MN identifier, referred to as a MNID in this document, to manage tunnel information or forwarding entries at the LMA or AR. The MNID must be unique and unchanging in the LMM domain, and is used to associate the MN with its related information. Some examples of MNIDs are a Network Access Identifier, a Mobile IP Home Address, and a link dependent identifier. In the case of the 802.11 binding, the ID will be simply the 802.11 MAC address. The AR must be able to set the MNID in all EMP messages it sends. If the link-layer technology is unable to provide such functionality, the AR must keep some state on the MNID.

The EMP signaling is sent using SCTP association between the LMA and the AR. The association is established when the AR powers up and is used for all MNs. The message structure follows the TLV format like other SCTP messages. EMP defines 4 message:

Name	**Meanings**
Hello	HELLO messages are exchanged between an AR and the LMA during AR startup.
Query	When an AR detects that a MN has joined its link, it sends a QUERY containing the MNs ID to the LMA. The LMA responds with an UPDATE REPLY containing the MN's ID and all global addresses belonging to the MN, if any are known.
Update	Either an AR or the LMA can send an UPDATE. When sent from an AR to the LMA with the code set to 0, the message contains the MN ID and a new IP Address for verification, and the AR expects a reply.
Reply	REPLY messages are sent from the LMA to the AR in response to an UPDATE or a QUERY. Each REPLY message always contains a MNID. If the REPLY is sent in response to an UPDATE, the address is the same address that was in the UPDATE, and conveys status information to the AR. If the REPLY is sent in response to a QUERY, the reply contains all known IP addresses belonging to the MN.

EMP must handle three basic scenarios:

1. A MN powers-on in the LMMD.
2. A MN moves to a new AR in the same LMMD
3. A MN crashes, powers-off, leaves the coverage area, or moves to a different LMMD

3.3.2. Protocol

3.3.2.1. First attachment of MN to the LMM domain

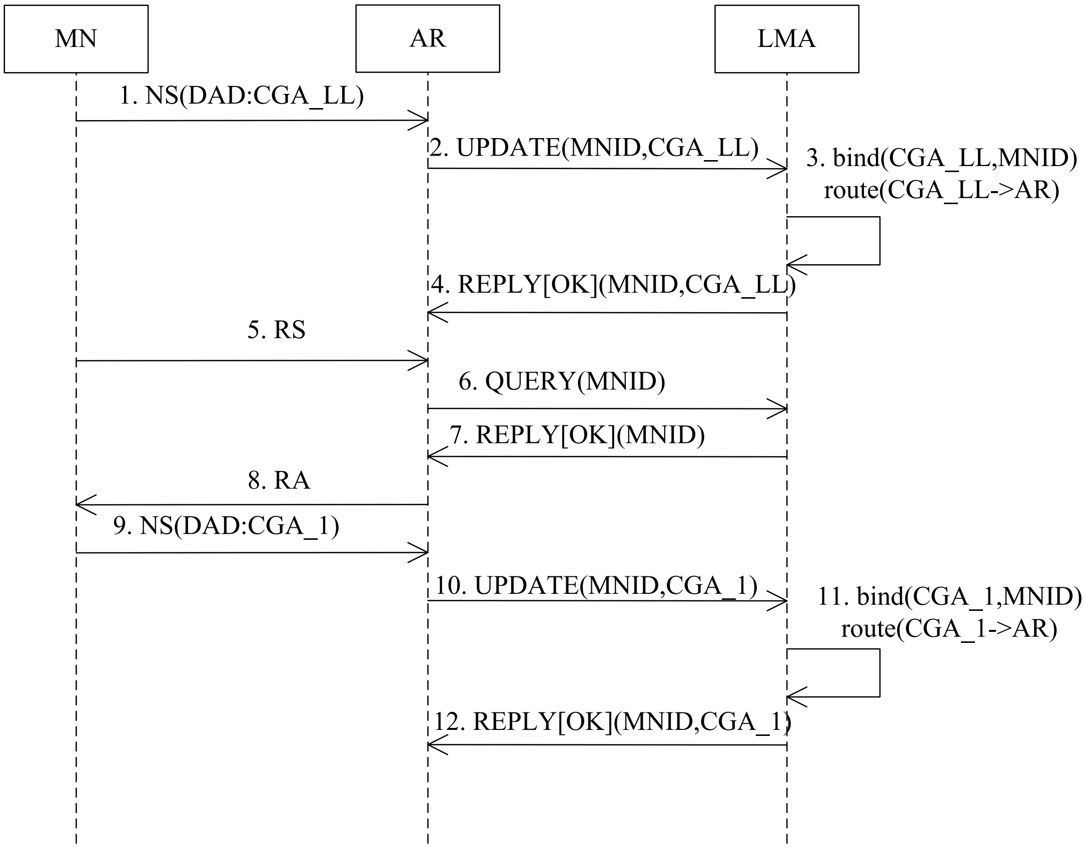


Figure 25: MN powers on and configures a Link-Local and 1 Global Unicast CGAs

The Idea is similar to the idea of HIP in the sense of using the SEND public key as an identifier MNID. When a MN powers on for the first time, it will generate a link local address based on its public key (CGA_LL) as per RFC3972 [33]:

1. The MN performs DAD on the address as per RFC2462 [34]. The DAD-NS message generated will contain the public key in the CGA option as defined by SEND [35].
2. Upon reception of this NS message, the access router AR SHOULD generate a UPDATE to the LMA with the public key as the MNID along with CGA_LL.
3. The LMA SHOULD bind the CGA_LL to the MNID and establish a route binding for the CGA_LL to the access router AR1.
4. The LMA acknowledges the receipt of the UPDATE message.
5. While waiting for the completion of DAD, the MN may generate RS message as per

RFC2461 [36] with the unspecified address as the source address. Such an RS message will not contain a CGA option.

6. When the AR detects that a MN has connected to its link (i.e. by receipt of a RS), in order to recognize if the MN is powered or is moving, the AR queries the LMA for information about the MN.
7. Because this is the first attachment, the LMA has no information for the MN, so it replies with a message empty except for the MNID.
8. The access router will respond with a **multicast RA** as per RFC2461 [36]. With the prefix information received in the RA message,
9. the MN will cryptographically generate one or more global addresses (CGA_*). For each of these addresses, the MN will perform DAD as the IID (???) is likely to be different for each of these cryptographically generated addresses. In this example, we assume that there is a global address CGA_1
10. For every DAD-NS received from the MN, the access router AR1 will generate a UPDATE message to the LMA establishing binding in the LMA.
11. The LMA SHOULD bind the CGA_1 to the MNID and establish a route binding for the CGA_1 to the access router AR1.
12. The LMA acknowledges the receipt of the UPDATE message.

3.3.2.2. Moving to a new link in the LMM domain

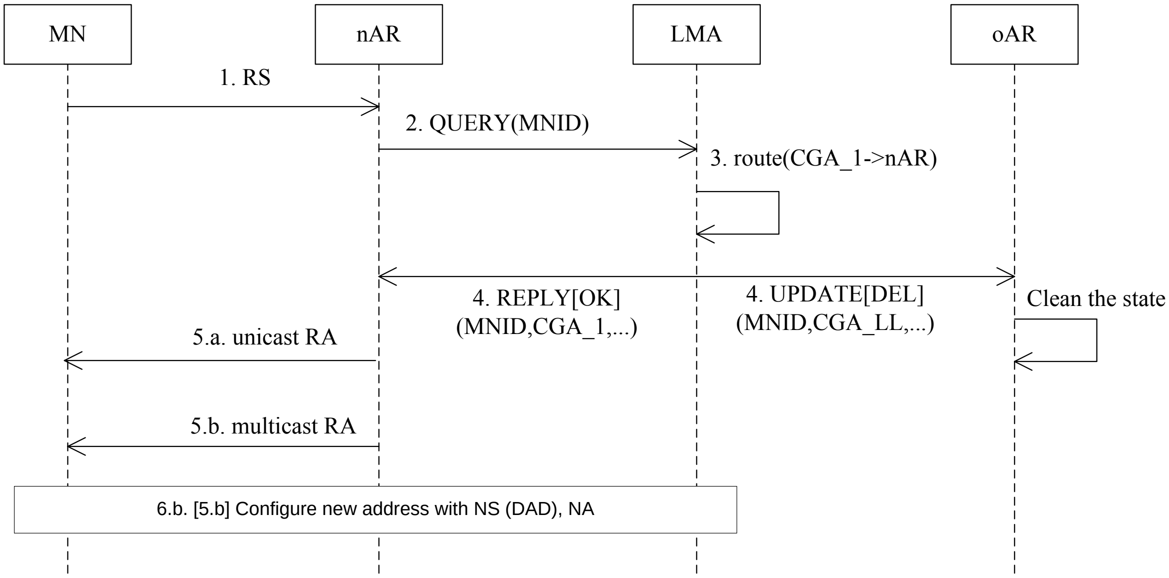


Figure 26: MN getting handover hint

A MN can configure a new address at any time, however it is most likely to do so when it enters a new LMMD. When the MN moves within the NETLMM domain:

1. It will send a RS message with the source address as its link-local address as specified by [37].
2. The new Access Router again can use the public key in CGA option to infer the MNID and sends a QUERY to the LMA. Because the MN has registered to the LMA before and is moving to a new AR, the LMA has an entry for the MN,
3. It also deduces that the MN has moved to a new AR in its LMMD, so it switches the MN's traffic to the tunnel to the new AR,
4. The LMA sends the new AR the MN's IP addresses so the new AR can update its forwarding state (Figure 2) and informs the old AR so that it can clean up state.
5. The new AR responds a message RA to the MN
   a. If the new access router chooses to respond with a unicast RA, all required steps are done.
   b. The new access router can choose to respond with a multicast RA
6. If 5.b happens, the MN will send a NS to learn about the new access router and confirm the reachability.

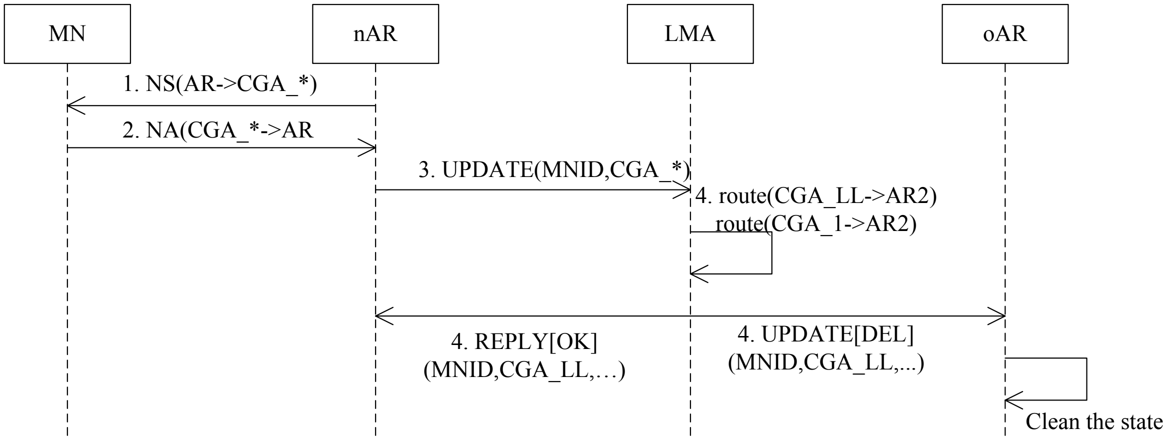


Figure 27: AR getting handover hint of MN whose IP address is known

Instead of the MN receiving the hint, in scenarios were the new access router receives the hint with the IP address of the handing over MN,

1. the AR can send a NS to that IP address.
2. The NA message received in response will contain the public key of the MN
3. with the received MNID, the AR can send update message to the LMA.

4. The LMA sends REPLY to nAR and send UPDATE[DEL] to oAR to clean up the state as in the previous scenario.

3.4. MOBIKE

MOBIKE [38][39], an extension of IKEv2, stands for IKEv2 Mobility and Multihoming Protocol and is designed to allow remote VPN users move (the addresses need to be changed) without re-establishing new IKE and IPsec Security Associations (SAs). The MOBIKE protocol provides a mechanism for updating the IP addresses of existing IKE and IPsec SAs.

The main scenario for MOBIKE is enabling a remote access VPN user to move from one address to another without re-establishing all security associations with the VPN gateway. MOBIKE updates only the outer (tunnel header) addresses of IPsec SAs, and the addresses and other traffic selectors used inside the tunnel stay unchanged. Thus, mobility can be (mostly) invisible to applications and their connections using the VPN.

MOBIKE also supports more complex scenarios where the VPN gateway also has several network interfaces: these interfaces could be connected to different networks or ISPs, they may be a mix of IPv4 and IPv6 addresses, and the addresses may change over time.

However MOBIKE supports the mobility of only one endpoint, it is best suited for situations where the address of at least one endpoint is relatively stable, and use DNS as a rendez-vous mechanism. Besides, MOBIKE doesn't describe the load balancing possibility. Only one pair of addresses is used for a SA at a moment.

Conclusion

The state of the art during the first 9 months has shown many works in progress of IETF which support mobility and/or multi-homing features (MMIP, HIP, mSCTP, MOBIKE, NETLMM). Because the goals for each IETF protocol are different, some features exist in only one protocol while other features are covered by other protocols, the solution for multihoming mobility internet must be a combination of those protocols. Though those protocols are not all completed, it allows us to construct the first simple architecture as a first proof of concept.

In our future work, we will try to combine different features of different protocols for different use case scenarios and carry out the measurement to have a quantitative estimation as an input for the new B3G architecture design process.

References

[1] R. Ramanathan, "Mobility support for nimrod : Challenges and solution approaches," RFC2103, February 1997.

[2] P. Bhagwat, S. K. Tripathi, and C. Perkins, "Network layer mobility: an architecture and survey," 1995. [Online]. Available: citeseer.ist.psu.edu/article/bhagwat96network.html

[3] C. E. Perkins and D. B. Johnson, "Mobility support in ipv6," RFC3775, June 2004.

[4] M. R. Helsinki, "Which layer for mobility? - comparing mobile ipv6, hip and sctp." [Online]. Available: citeseer.ist.psu.edu/711917.html

[5] C. Perkins, "Ip mobility support for ipv4," IETF RFC 3344, August 2002. [Online]. Available: http://www.faqs.org/rfcs/rfc3344.html

[6] C. E. Perkins, S. R. Alpert, and B.Woolf, Mobile IP; Design Principles and Practices, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., May 1998.

[7] F. Teraoka, K. Uehara, H. Sunahara, and J. Murai, "VIP: A protocol providing host mobility," Communications of the ACM, vol. 37, no. 8, pp. 67–75, 1994.

[8] P. Yalagandula, A. Garg, M. Dahlin, L. Alvisi, and H. Vin, "Transparent mobility with minimal infrastructure," Austin, TX, USA, 2001.

[9] E. Vehmersalo, "Mobility and multi-homing solutions with host identity layer - introduction and comparison." [Online]. Available: citeseer. ist.psu.edu/vehmersalo04mobility.html

[10] M. Ishiyama, M. Kunishi, and F. Teraoka, "An analysis of mobility handling in lin6," in Proceedings of the Fourth Inernational Sympsium on Wireless Personal Multimedia Communications, September 2001.

[11] R. Ramjee, T. F. L. Porta, S. Thuel, K. Varadhan, and S. Y. Wang, "HAWAII: A domain-based approach for supporting mobility in wide-area wireless networks," IEEE/ACM Transactions on Networking, vol. 6, pp. 283–292, June 2002. [Online]. Available: http://www.belllabs. com/user/ramjee/papers/papers.html

[12] A. Campbell, J. Gomez, S. Kim, A. Valko, C. Wan, and Z. Turanyi, "Design, implementation, and evaluation of cellular ip," IEEE Personal Commun. Mag, vol. 7, 2000. [Online]. Available: citeseer.csail.mit.edu/article/campbell00design.html

[13] Z. D. Shelby, D. Gatzounas, A. T. Campbell, and C.-Y. Wan, "Cellular ipv6," draft-shelby-seamoby-cellularipv6-00.txt, November 2000.

[14] G. Su and J. Nieh, "Mobile communication with virtualnetwork address translation,"

Department of Computer Science, Columbia University, Tech. Rep., February 2002.

[15] J. Mysore and V. Bharghavan, “A new multicasting-based architecture for internet host mobility,” in Mobile Computing and Networking, 1997, pp. 161–172. [Online]. Available: citeseer.ist.psu.edu/mysore97new.html

[16] S. Zhuang, K. Lai, I. Stoica, R. Katz, and S. Shenker, “Host mobility using an internet indirection infrastructure,” 2002. [Online]. Available: citeseer.ist.psu.edu/zhuang02host.html

[17] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, “Internet indirection infrastructure,” in SIGCOMM ’02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications. New York, NY, USA: ACM Press, 2002, pp. 73–86.

[18] A. C. Snoeren and H. Balakrishnan, “Tcp connection migration,” Internet Draft, draft-snoeren-tcp-migrate-00.txt (work in progress), November 2000.

[19] R. Moskowitz and P. Nikander, “Host identity protocol architecture,” draftietf-hip-arch-02, January 2004. [Online]. Available: http://homebase.httconsult. com/Docs/draft-ietf-hip-arch-02.txt

[20] R. Moskowitz, P. Nikander, P. J. (editor), and T. Henderson, “Host identity protocol,” draft-ietf-hip-base-02, February 2005. [Online]. Available: http://homebase.htt-consult.com/Docs/draft-ietf-hip-base-02.txt

[21] P. Nikander and J. Laganier, “Host identity protocol (hip) domain name system (dns) extensions,” draft-ietf-hip-dns-01.txt, Febuary 2005, work in progress.

[22] P. Nikander, J. Ylitalo, and J. Wall, “Integrating security, mobility, and multi-homing in a hip way,” in Proceedings of Network and Distributed Systems Security Symposium (NDSS’03), San Diego, CA, February 6-7 2003, pp. 87–99. [Online]. Available: http://www.tml.tkk.fi/ pnr/publications/NDSS03-Nikander-et-al.pdf

[23] A. A. E. Al, T. N. Saadawi, and M. J. Lee, “A transport layer load sharing mechanism for mobile wireless hosts.” in PerCom Workshops, 2004, pp. 87–91.

[24] S. J. Koh, M. J. Chang, and M. Lee;, “msctp for soft handover in transport layer,” Communications Letters, IEEE, vol. 8, pp. 189 – 191, March 2004.

[25] S.-J. Koh and S.-W. Kim, “msctp for vertical handover between heterogeneous networks,” Web and Communication Technologies and Internet Related Social Issues HSI 2005, 2005.

[26] R. Fracchia, C. Casetti, C.-F. Chiasserini, and M. Meo, “A wise extension of sctp for wireless networks,” IEEE International Conference on Communications IEEE Cat, 2005.

[27] M. Kalla, K. Morneault, V. Paxson, I. Rytina, H. J. Schwarzbauer, C. Sharp, R. Stewart, T. Taylor, Q. Xie, , and L. Zhang, “Stream control transmission protocol. internet engineering task force,” RFC2960, http://www.ietf.org/rfc/rfc2960.txt, October 2000.

[28] P. L. M. Tuexen, R. Stewart and E. Rescorla, “Authenticated chunks for stream control transmission protocol (sctp),” Internet draft, draft-ietftsvwg-sctp-auth-04.txt (work in progress), September 2006.

[29] J. Kempf, “Goals for network-based localized mobility management (netlmm),” Internet draft, draft-ietf-netlmm-nohost-req-04.txt (work in progress), August 2006.

[30] J. Kempf, “Problem statement for network-based localized mobility management,”draft-ietf-netlmm-nohost-ps-04.txt, June 2006.

[31] G. Giaretta, K. Leung, M. Liebsch, P. Roberts, K. Nishida, H. Yokota, M. Parthasarathy, and H. Levkowetz, “Netlmm protocol,” Internet draft, draft-giaretta-netlmm-dt-protocol-00.txt (work in progress), June 2006.

[32] J. Laganier, S. Narayanan, and F. Templin, “Network-based localized mobility management interface between mobile node and access router,” Internet draft, draft-ietf-netlmm-mn-ar-if-01.txt (work in progress), June 2006.

[33] T. Aura, “Cryptographically generated addresses (cga),” RFC3972, March 2005.

[34] B. S. Thomson and T. Narten, “Ipv6 stateless address autoconfiguration,”RFC2462, December 1998.

[35] J. Arkko, J. Kempf, B. Zill, and P. Nikander, “Secure neighbor discovery (send),” RFC3971, March 2005.

[36] T. Narten, E. Nordmark, and W. Simpson, “Neighbor discovery for ip version 6 (ipv6),” RFC2461, December 1998.

[37] S. Narayanan, “Detecting network attachment in ipv6 networks (dnav6),”Internet draft, draft-pentland-dna-protocol-01 (work in progress), July 2005.

[38] P. Eronen, “Ikev2 mobility and multi-homing protocol (mobike),” RFC455, June 2006.

[39] T. Kivinen and H. Tschofenig, “Design of the ikev2 mobility and multi-homing (mobike) protocol,” RFC4621, August 2006.