

Internet Engineering Task Force  
INTERNET DRAFT

Fumio Teraoka  
Keio University/Sony CSL  
Masahiro Ishiyama  
Toshiba  
Mitsunobu Kunishi  
Keio University  
30 December 2003

## LIN6: A Solution to Multihoming and Mobility in IPv6

<[draft-teraoka-multi6-lin6-00.txt](#)>

### Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents, valid for a maximum of six months, and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

### Abstract

LIN6 is a protocol supporting multihoming and mobility in IPv6. LIN6 introduces the node id, not the interface id, for each node. Each node can be identified by its node id no matter where the node is connected and no matter how many interfaces the node has. In the IPv6 layer, 64-bit node id called LIN6 ID is used while 128-bit node-id called LIN6 generalized ID is used above the Transport layer. TCP connections and security associations can be preserved even if the node moves to another subnet or the node changes the using interface in a multihoming environment without modifying TCP or IPsec. In comparison with Mobile IPv6, LIN6 has several advantages in terms of header overhead and fault tolerance.

---

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

## 1. Introduction

This document describes the protocol specification of LIN6[IEICE01,WPMC01]. We propose LINA (Location Independent Network Architecture) to solve several problems such as multihoming and mobility by redesigning network architecture and addressing. LIN6 is an application of LINA to IPv6[RFC2460]. LIN6 supports mobility and multihoming with IPsec[RFC2401] in IPv6.

LIN6 has several advantages from viewpoint of multihoming support. Assume that Node-A starts TCP communication with Node-B, a multihoming node with two network interfaces, by using IPsec.

- o Node-A can recognize the identity of Node-B by the LIN6 address of Node-B no matter which network interface of Node-B is used.
- o The same security association (SA) can be used between Node-A and Node-B no matter which network interface of Node-B is used.
- o The TCP connection and the SA are still available even after the used network interface of Node-B is switched to another during communication.

From mobility support viewpoint, LIN6 has several advantages in comparison with Mobile IPv6[MIPv6] as follows:

- o LIN6 has no header overhead because it does not use any extension headers of IPv6 while Mobile IPv6 uses the Destination Options Header for the Home Address Option and the Routing Header for optimal routing.
- o LIN6 is more fault tolerant than Mobile IPv6. In Mobile IPv6, the Home Agent cannot be replicated to the subnet other than the home link of the mobile node. LIN6 introduces the Mapping Agent which can be replicated anywhere in the Internet.
- o LIN6 keeps end-to-end communication model, that is, LIN6 does not use any packet interceptor/forwarder such as the Home Agent of Mobile IPv6. There is no tunneling in LIN6.

## [2. Terminology](#)

This document uses the following terms.

### node:

The node is the general term to specify the equipment that understands IP in the Internet. The node includes hosts, mobile terminals, routers, and so on.

### LIN6 ID:

Teraoka

Expires 29 June 2004

[Page 2]

---

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

The LIN6 ID is assigned to the node and uniquely identifies the node in the Internet. It is 64 bits in length.

### LIN6 prefix:

The LIN6 prefix is the predefined constant value attached to the head of the LIN6 ID to construct the LIN6 generalized ID.

### LIN6 generalized ID:

The LIN6 generalized ID is the identifier of the node used in the transport layer and the upper layers. It is 128 bits in length. The higher 64 bits of the LIN6 generalized ID is the LIN6 prefix and the lower 64 bits is the LIN6 ID. The LIN6 generalized ID is assigned to the node, not to the network interface. Application programs use the LIN6 generalized ID to indicate the target node. TCP establishes the TCP connection between two LIN6 generalized IDs. Note that the LIN6 generalized ID does not appear in the IPv6 header on the link. The source and destination LIN6 generalized IDs are mapped to the LIN6 addresses in the IPv6 layer before the packet is passed to L2.

### network prefix:

The network prefix indicates the subnet to which the node is connected. It is attached to the head of the LIN6 ID to construct the LIN6 address.

### LIN6 address:

The LIN6 address is assigned to the network interface of the node. The higher 64 bits of the LIN6 address is the network prefix and the lower 64 bits is the LIN6 ID so that the LIN6 address specifies the identifier of the node as well as the point of

attachment to the Internet of the node. Note that the LIN6 address appears in the IPv6 header on the link and is not passed to the transport layer. The source and destination LIN6 addresses are mapped to the LIN6 generalized IDs in the IPv6 layer before the packet is passed to the transport layer.

stationary address:

The stationary address is a LIN6 address assigned to a stationary node that understands LIN6. In the lower half of the stationary address, the Universal/Local bit of EUI-64 is set to zero while that bit in the LIN6 address is set to one.

mapping:

The mapping is the relation between the LIN6 ID and the network prefix.

Mapping Agent:

The Mapping Agent (MA) is the function that maintains the mapping of the LIN6 node. Each LIN6 node is associated with one or more Mapping Agents. The AAAA record of the LIN6 node in the DNS consists of the network prefix of the MA and the LIN6 ID of the

LIN6 node. If the LIN6 node is associated with more than one MAs, it has several AAAA records in the DNS, each of which specifies the distinct MA. The MA also has packet forwarding function. If a non-LIN6 node sends a packet to a LIN6 node, the packet reaches the MA of the LIN6 node first because the AAAA record of the LIN6 node specifies the network prefix of the MA of the LIN6 node. When the MA receives such a packet, the MA forward the packet to the LIN6 node by IP-in-IP tunneling.

Mapping Cache:

The Mapping Cache is the cache of a mapping in the node.

MA WID:

MA WID is the well-known interface identifier which all Mapping Agents assign to their interfaces. All LIN6 nodes know this value.

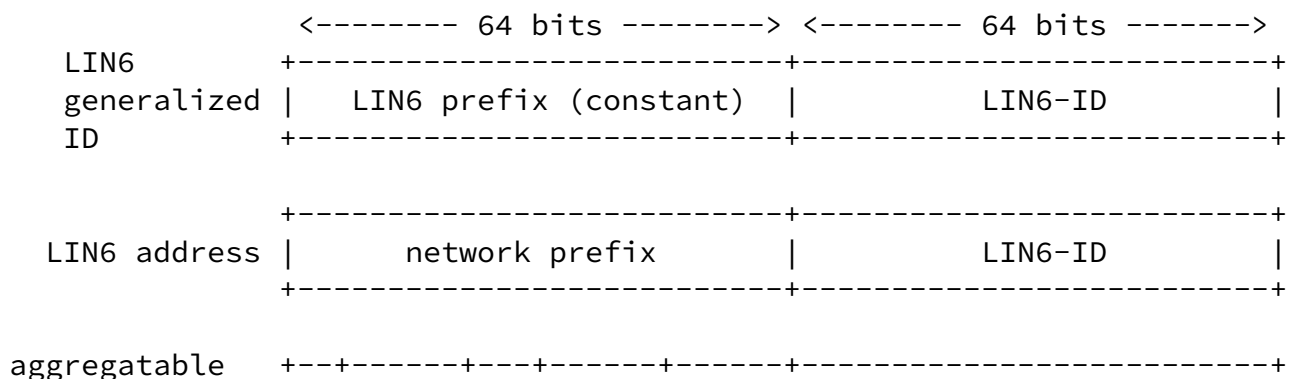
normal IPv6 address:

The aggregatable global unicast address.

### [3. Protocol Overview](#)

#### [3.1. Address](#)

LIN6 uses two types of network addresses: the LIN6 generalized ID and the LIN6 address. Figure 1 depicts their formats. The LIN6 generalized ID is 128 bits in length and is used in the transport layer and the upper layers. LIN6 generalized ID is the identifier of the node in the transport layer and the upper layers and does not change even if the node moves. The LIN6 address is also 128 bits in length and is used in the network layer. The LIN6 address specifies both the location and the identifier of the node. The network prefix part of the LIN6 address changes when the node moves to another subnet. The formats of the LIN6 generalized ID and the LIN6 address are the same as the format of IPv6 aggregatable global unicast address[RFC2374].



global unicast address	FP	TLA ID	res	NLA ID	SLA ID	Interface ID
	+	+	+	+	+	+

Figure 1: The LIN6 generalized ID and the LIN6 address

Both the LIN6 generalized ID and the LIN6 address consist of two fields: the network prefix and the LIN6 ID. Both fields are 64 bits in length. The LIN6 ID is the global unique identifier of the node. EUI-64[EUI64] will be used as LIN6-ID. The network prefix of the LIN6 address indicates the subnet to which the node is connected while that of the LIN6 generalized ID is the constant value and is called the LIN6 prefix. In other words, the LIN6 address indicates both the location and the identifier of the node while the LIN6 generalized only identifies the node. Thus, the LIN6 generalized ID is used in the transport layer and the upper layers to identify the node, and the LIN6 address is used in the network layer to indicate both the location and the identifier of the node. Note that the LIN6 ID and the LIN6 generalized ID are assigned per node while the LIN6 address is assigned per network interface. Also note that the normal IPv6 address, i.e., the aggregatable global unicast address, is assigned to the network interface of the node in addition to the LIN6 address.

### 3.2. Address Processing

Figure 2 shows the procedures of address processing. As mentioned above, the LIN6 generalized ID consists of the (constant) LIN6 prefix and the LIN6 ID. In packet transmission, the transport layer passes the LIN6 generalized ID of the destination node to the network layer. The network layer obtains the network prefix, i.e., the current location, of the destination node by some means (see [Section 3.3](#)). The network layer concatenates the obtained network prefix and the LIN6 ID contained in the LIN6 generalized ID to create the LIN6 address of the destination node. The transport layer passes its own LIN6 generalized ID to the network layer as the source address. In addition, the transport layer may pass the network prefix to specify the network interface from which the packet will be transmitted.

In packet reception, the source address field of the packet contains the LIN6 address of the source node. The network layer concatenates the

LIN6 prefix and the LIN6 ID contained in the LIN6 address of the source node to create the LIN6 generalized ID, and then the network layer notifies the transport layer of the packet reception with the LIN6 generalized ID of the source node. Thus, from the transport layer's viewpoint, communication is done between the two LIN6 generalized IDs. The network layer may pass the network prefixes of the source and the destination LIN6 addresses to the transport layer.

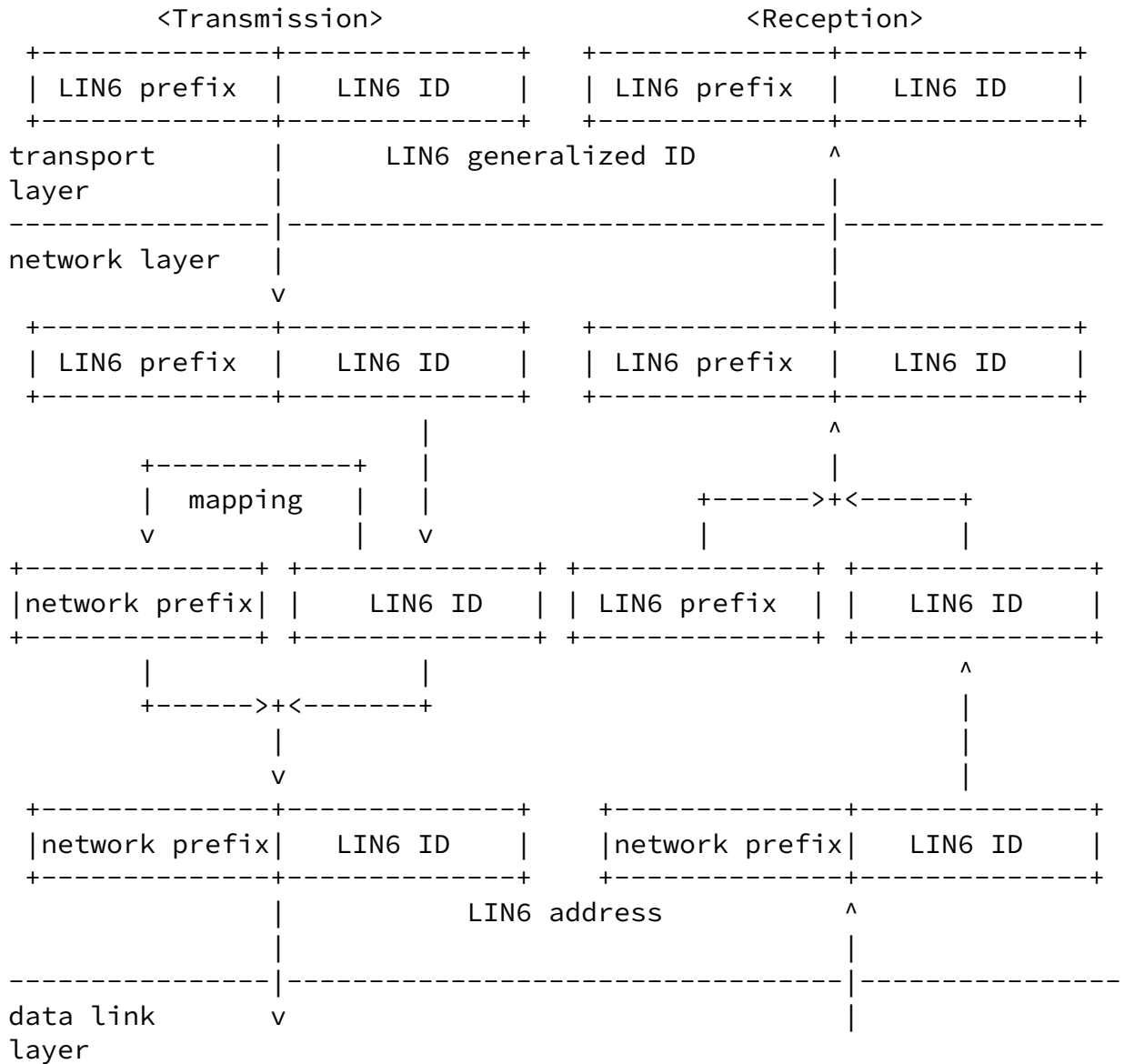


Figure 2: Address processing

### 3.3. Distinction between the LIN6 Address and the Normal IPv6 Address

As shown on the right side of Figure 2, the receiving node must distinguish the LIN6 address from the normal IPv6 address to decide whether address conversion must be done. From address format viewpoint, however, the LIN6 address is indistinguishable from the normal IPv6 address, i.e., LIN6 is fully compatible with IPv6. There are some methods to distinguish the two address types. As a temporary solution,

---

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

LIN6 employs a special value as a part of the LIN6 ID. To distinguish the LIN6 address, Keio University obtained the OUI value 0x00-0C-21 of EUI-64[EUI64]. According to the IPv6 addressing scheme, the Universal/Local bit of EUI-64 must be reversed in the global IPv6 address. Thus, if the upper 24 bits of the lower 64 bits of the IPv6 address is 0x02-0C-21, the IPv6 address is the LIN6 address.

#### [3.4.](#) Stationary Address

As mentioned above, LIN6 introduces the 64-bit LIN6 ID as the node identifier. The identity of a mobile node is guaranteed by the LIN6 ID even if the network prefix of the node changes. In case of a stationary node the entire 128-bit can be used as the node identifier because the network prefix does not usually change. Thus, LIN6 introduces another type of LIN6 address, the stationary address which is assigned to a stationary node.

The upper half of the stationary address is the normal network prefix. To distinguish the stationary address, the first three bytes of the lower half of the stationary address must be a special value such as the OUI value of Keio University in which the Universal/Local bit is set to zero. The remaining five bytes of the lower half of the stationary address can be generated randomly by the stationary node. Upon generating a stationary address, the Duplicate Address Detection procedure must be done.

By introducing the stationary address, it is not required to assign the LIN6 ID to stationary nodes. The LIN6 ID must be assigned only to mobile nodes. This reduces administrative workload for LIN6 ID assignment.

Upon transmission, if the destination address is a stationary address, this address is set to the destination address field of the IPv6 header without the address processing described in [Section 3.2](#). Upon reception, if the source address is a stationary address, this address is passed to the upper layer without the address processing.

#### [3.5.](#) Mapping Agent



The relation between the LIN6 ID and the network prefix is called mapping. LIN6 introduces the Mapping Agent (MA) to maintain the mapping of the LIN6 node. The Mapping Agent maintains the mapping of the LIN6 node and replies to queries about mapping. Each LIN6 node is associated with one or more Mapping Agents. The multihomed LIN6 node has multiple network interfaces, each of which has a network prefix. The mobile LIN6 node has one or more network prefixes which change when the mobile LIN6 node moves to another subnet. The LIN6 node registers its network prefixes with one of the MAs that maintain the mappings of the LIN6 node when the LIN6 node has booted or its network prefixes change.

Consistency among the databases on the Mapping Agents must be kept by some procedures. These procedures are beyond of the scope of this document.

Each Mapping Agent has one or more "virtual subnets". The Mapping Agent announces the routes of the virtual subnets it manages. The network prefixes of the virtual subnets are used as the upper 64-bits of the AAAA records of LIN6 nodes as described in the next section.

Each Mapping Agent assigns its network interface the predefined value called MA IFID as the interface identifier. The MA IFID is used as the lower 64-bits of the destination address when a LIN6 node sends a query packet to the Mapping Agent.

### [3.6.](#) DNS Entries

It can be assumed that the relation between the LIN6 node and its Mapping Agent is almost static in contrast to the mappings of the LIN6 node. LIN6 makes use of the Domain Name System (DNS) to maintain the relation between the mobile node and its Mapping Agents. The LIN6 node has one or more AAAA records in DNS. An AAAA records maps a FQDN to an IPv6 address which consists of the network prefix of the MA and the LIN6 ID of the LIN6 node. When a LIN6 node (sender node) sends a LIN6 packet to a LIN6 node (receiver node), the sender node obtains the network prefixes of the receiver node as follows:

The LIN6 node also has the PTR record. A PTR record maps a (reversed) LIN6 ID to a FQDN. When a LIN6 node (receiver node) that received a LIN6 packet returns a LIN6 packet to the sender node, the receiver node obtains the network prefix of the sender node as follows: 1) the

receiver node obtains the FQDN of the sender node from the PTR record of the sender node by specifying the LIN6 ID of the sender node; 2) the receiver node obtains the AAAA records of the sender node by specifying the FQDN of the sender node; 3) the receiver node learns the network prefixes of the sender node from the AAAA records.

Thus, no modifications to DNS are required.

### [3.7.](#) Communication Procedure

The LIN6 Communication procedure is shown in Figure 3. Assume that Node-A tries to send a packet to Node-B. For simplicity, Node-A and Node-B are associated with only a single Mapping Agent, respectively (MA-A and MA-B). The communication procedure is as follows:

1. At first, Node-A and Node-B register their network prefixes with MA-A and MA-B, respectively. The Authentication Header of IPv6 is used in registration.

2. The Node-A obtains the AAAA record of Node-B from the name server by indicating the domain name of Node-B. The obtained AAAA record consists of the network prefix of MA-B and the LIN6 ID of Node-B.
3. Node-A generates the LIN6 generalized ID of Node-B by concatenating the LIN6 prefix and the lower 64-bits of the AAAA record. Node-A also generates the IPv6 address of MA-B by concatenating the upper 64-bits of the AAAA record and the MA IFID.
4. Node-A obtains the network prefix of Node-B from MA-B by indicating the LIN6 ID of Node-B.
5. Node-A sends a packet to Node-B.
6. To avoid impersonation, Node-B must obtain the network prefix of Node-A from MA-A. Node-B obtains the FQDN of Node-A from the name server by indicating the LIN6 ID of Node-A, and then obtains the AAAA record of Node-A from the nameserver by indicating the FQDN of Node-A. Node-B generates the LIN6 generalized ID of Node-A and the IPv6 address of MA-A as described in step 3.

7. Node-B obtains the network prefix of Node-A from MA-A by indicating the LIN6 ID of Node-A.
8. Node-B sends a packet to Node-A.

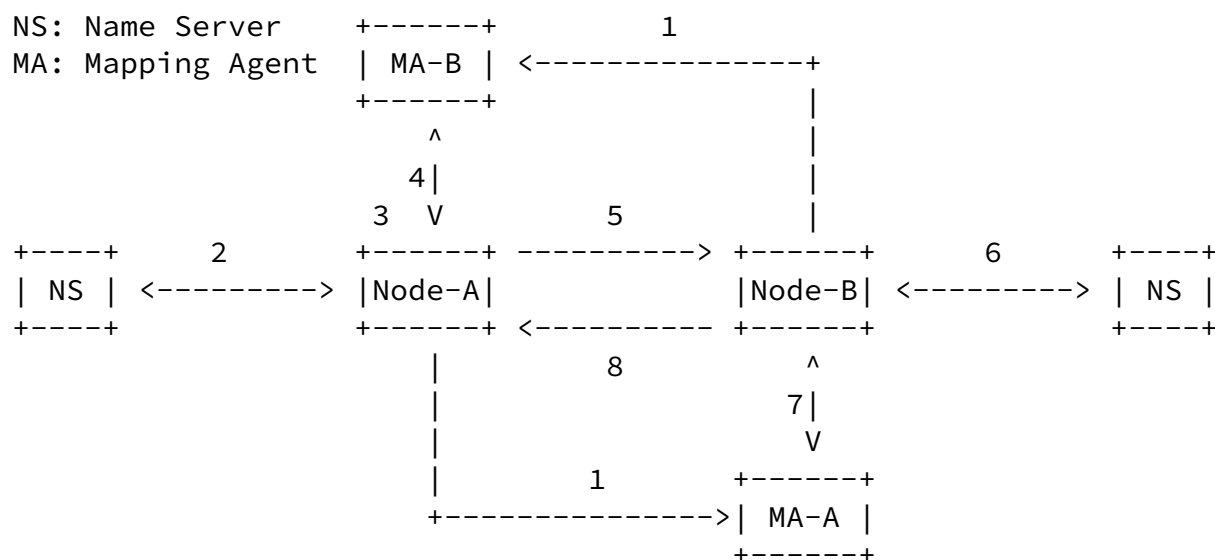


Figure 3: Communication procedures

### 3.8. IPsec Operation

In the sending node, IPsec is processed as follows. When the packet is passed from the transport layer, the source and the destination address fields in the IPv6 header contain LIN6 generalized IDs. The security

association is decided by using the destination LIN6 generalized ID, and then IPsec calculation is executed. After that, the source and the destination LIN6 generalized IDs are converted to LIN6 addresses.

In the receiving node, IPsec is processed as follows. Upon packet reception, the source and the destination address fields of IPv6 header contain LIN6 addresses. First, these LIN6 addresses are converted to LIN6 generalized IDs. The security association is decided by using the destination LIN6 generalized ID, and then IPsec calculation is executed.

### [3.9. Multihoming Support](#)

Assume that Node-A and Node-B have multiple network prefixes, respectively. Let LIN6\_P be the LIN6 prefix, ID\_A be the LIN6 ID of Node-A, and ID\_B be the LIN6 ID of Node-B. Transport sessions are identified by the pair of {LIN6\_P+ID\_A, LIN6\_P+ID\_B} regardless of the number of network prefixes Node-A and Node-B have. That is, both a TCP connection and a UDP session are identified by the tuple {LIN6\_P+ID\_A, port\_A, LIN6\_P+ID\_B, port\_B}. In the network layer, LIN6\_P is converted to the actual network prefix. The transport communication can be preserved even if the actual network prefixes change during the communication.

When Node-A sends a LIN6 packet to Node-B, the transport layer of Node-A can choose the source and the destination network prefixes by indicating them explicitly to the network layer. Upon receiving a LIN6 packet on Node-B, the network layer passes the source and the destination prefixes to the transport layer in addition to the source and the destination LIN6 generalized IDs. Thus, the transport layer of Node-B can know the source and the destination network prefixes of the received packet.

Figure 7 shows an example of multihoming support in LIN6. In the figure, Node-B is a multihoming host which has two network interfaces. Node-A and Node-B have the LIN6 generalized ID, "LIN6\_P+ID\_A" and "LIN6\_P+ID\_B", respectively, where "LIN6\_P" is the LIN6 Prefix, and "ID\_A" and "ID\_B" are the identifiers of Node-A and Node-B, respectively. Node-A has the LIN6 address, "P\_A+ID\_A", where "P\_A" is the network prefix of Node-A. Node-B has two LIN6 addresses, "P\_B1+ID\_B" and "P\_B2+ID\_B", where "P\_B1" and "P\_B2" are the network prefixes of Node-B corresponding to the two network interfaces.

1. Node-A establishes a TCP connection with Node-B by using IPsec. This TCP connection is established between {LIN6\_P+ID\_A, port\_A} and {LIN6\_P+ID\_B, port\_B}. The SA of IPsec is established between "LIN6\_P+ID\_A" and "LIN6\_P+ID\_B". Assume that Node-A obtains "P\_B1" and "P\_B2" as the network prefixes of Node-B and selects "P\_B1". The packet in the TCP connection has "P\_A+ID\_A" and "P\_B1+ID\_B" as the source/destination addresses. The packets in the TCP connection traverse Path\_A in the figure.

2. Assume that Path\_A crashes due to some reason, and ICMP Unreach is

returned to Node-A.

3. Node-A knows that Path\_A is unavailable and selects "P\_B2" as the network prefix of Node-B. The packet in the TCP connection has "P\_A+ID\_A" and "P\_B2+ID\_B" as the source/destination addresses. The packets in the TCP connection traverse Path\_B in the figure. Although the LIN6 address of Node-B changes, the TCP connection and the SA are still available even after the path change because the TCP connection is established between {LIN6\_P+ID\_A, port\_A} and {LIN6\_P+ID\_B}, and the SA is established between "LIN6\_P+ID\_A" and "LIN6\_P+ID\_B".

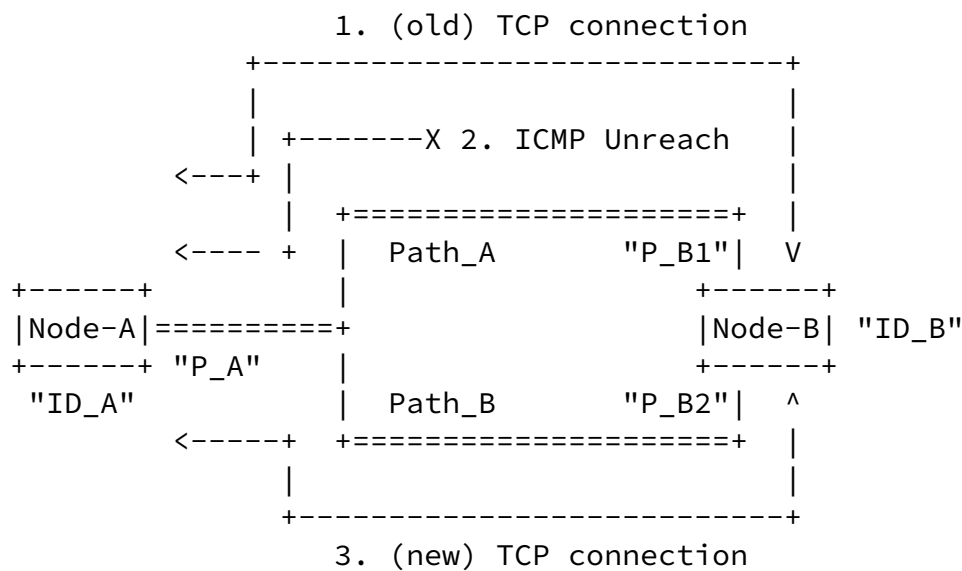


Figure 7: Multihoming Support

### [3.10.](#) Mobility Support

There are three types of procedures for location update to the correspondent node (CN) when the mobile LIN6 node (MN) moves to another subnet. Type 1 is for the LIN6 node that has a SA for the CN. Type 2 does not require a SA between the LIN6 node and the CN. Type 3 uses cookies that have been exchanged between the LIN6 node and the CN.

#### [3.10.1.](#) Mobility Support Type 1: Basic Procedure

Mobility support in LIN6 is shown in Figure 4. Assume that SAs are established between the mobile node (MN) and the correspondent node (CN), and between the MN and the MA. This mechanism relies on the SA to avoid impersonation.

1. The MN registers its current network prefix with the Mapping Agent (MA). The Authentication Header is used in this registration.

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

2. The CN obtains the LIN6 address and MA's address from the name server.
3. The CN obtains MN's network prefix from the MA.
4. The CN sends a packet to the MN. The source address of this packet is the normal IPv6 address because the CN is stationary in this case.
5. The MN returns a packet to the CN. The source address of the received packet is used as the destination address of the return packet because the source address of the received packet is the normal IPv6 address.
6. The MN moves to another subnet.
7. The MN registers the new network prefix with the MA and the CN. The Authentication Header is used in these registrations.
8. The CN sends a packet to the MN.

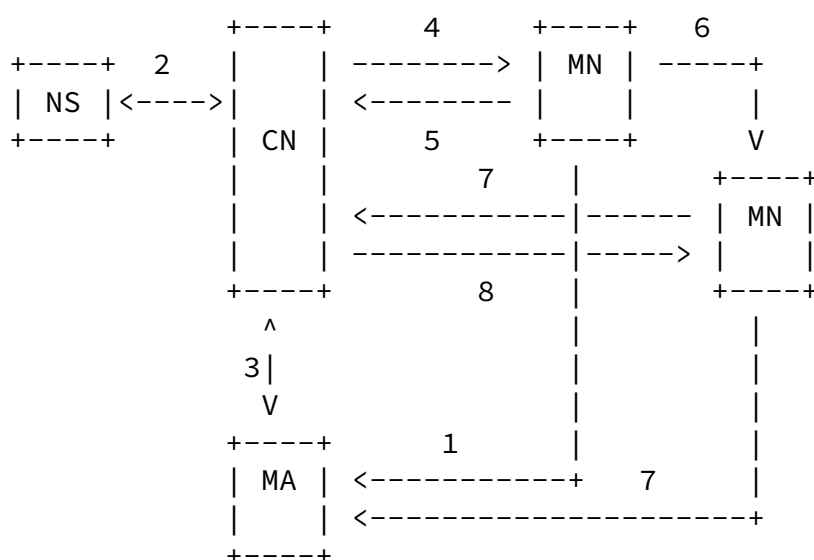


Figure 4: Mobility Support

### [3.10.2.](#) Mobility Support Type 2: Mapping Refresh Message

Figure 5 shows mobility support in which there is no SA between the MN and the CN. In terms of avoidance of impersonation, this mechanism is as secure as the current internet where every node trusts the DNS. This mechanism relies on the DNS and the MA. Steps 1 to 6 are the same as Figure 4.

7. The MN registers the new network prefix with the MA. The Authentication Header is used in this registration.

8. The MN sends the Mapping Refresh Message to CN to notify the CN of the event that the MN has moved.
9. The CN obtains the new network prefix of the MN from the MA.
10. The CN sends a packet to the MN.

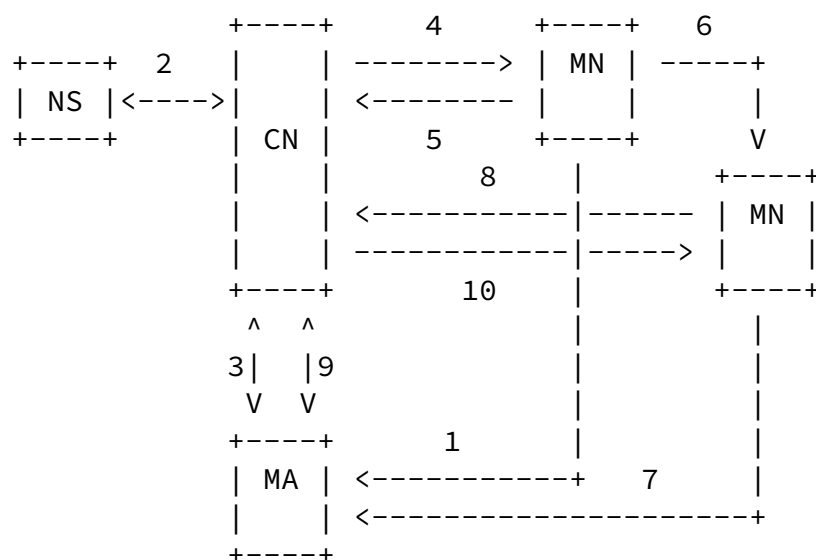


Figure 5: Mobility Support (no SA between MN and CN)

### [3.10.3. Mobility Support Type 3: Cookie](#)

Figure 6 shows mobility handling mechanism that uses a cookie for authentication. This mechanism is effective in the environment in which there is no possibility of eavesdropping. For example, messages are

encrypted on wireless links and wired links are well managed to avoid eavesdropping. Steps 1 and 2 are the same as Figure 4.

3. The CN receives MN's network prefix from the MA. At the same time, the CN notifies the MA of CN's cookie.
4. The MA notifies the MN of CN's cookie. This notification uses IPsec Authentication Header.
5. The CN sends a packet to the MN.
6. The MN returns a packet to the CN.
7. The MN moves to another subnet.
8. The MN registers the new network prefix with the CN. This registration includes CN's cookie received from the MA. The CN can trust this registration if the cookie sent to the MA and the cookie received from the MN are the same.

9. The MN registers the new network prefix with the MA. The Authentication Header is used in this registration.
10. The CN sends a packet to the MN.

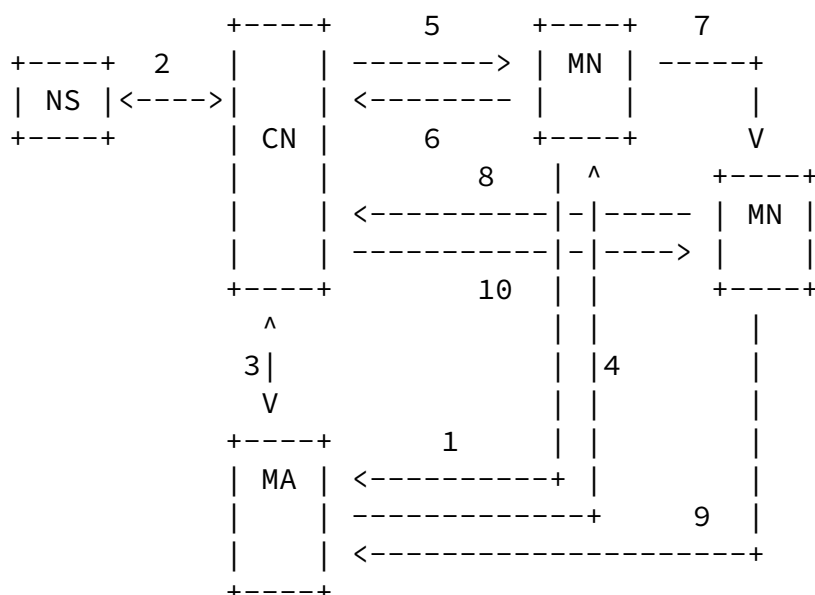




Figure 6: Mobility Support (simplified authentication)

### 3.11. Communication with non-LIN6 nodes

When a non-LIN6 node sends a packet to a LIN6 node, the destination address of the packet consists of the network prefix of the Mapping Agent of the LIN6 node and the LIN6 ID of the LIN6 node because this tuple is registered as the AAAA record of the LIN6 node in DNS. The Mapping Agent intercepts this packet and forwards it to the LIN6 node by IP-in-IP tunneling. Upon receiving this packet, the LIN6 node knows that the source host does not implement LIN6. The LIN6 node sends a return packet to the Mapping Agent by IP-in-IP tunneling, and then the Mapping Agent forwards the packet to the non-LIN6 node after decapsulation.

## 4. Packet Formats

#### 4.1. Data Packet

LIN6 uses the normal IPv6 header in which the LIN6 addresses are used in the source address field and the destination address field. Figure 8 shows the format of the normal IPv6 header.

version	Traffic Class	Flow Label		
Payload Length		Next Header	Hop Limit	
Source Address (LIN6 Address)				

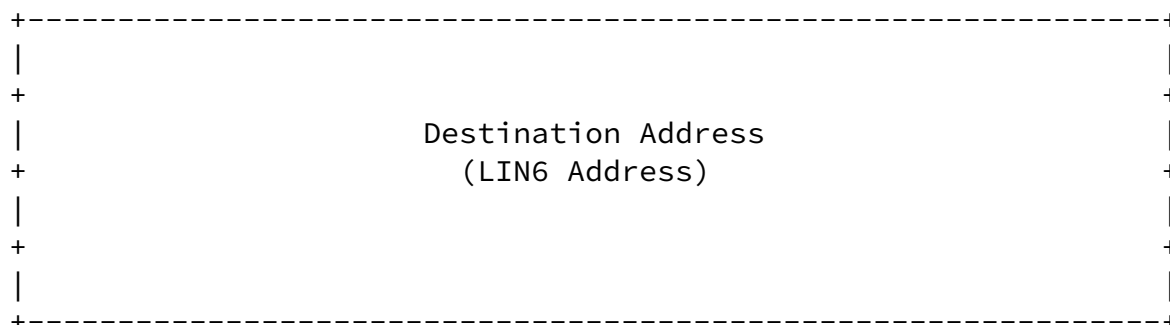
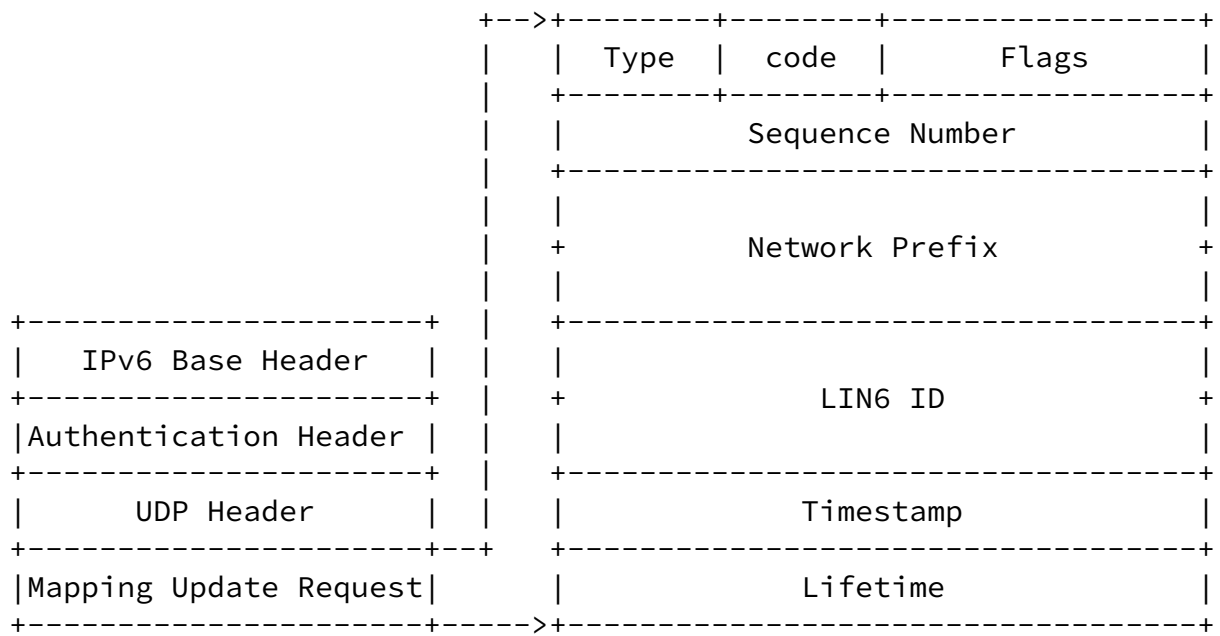


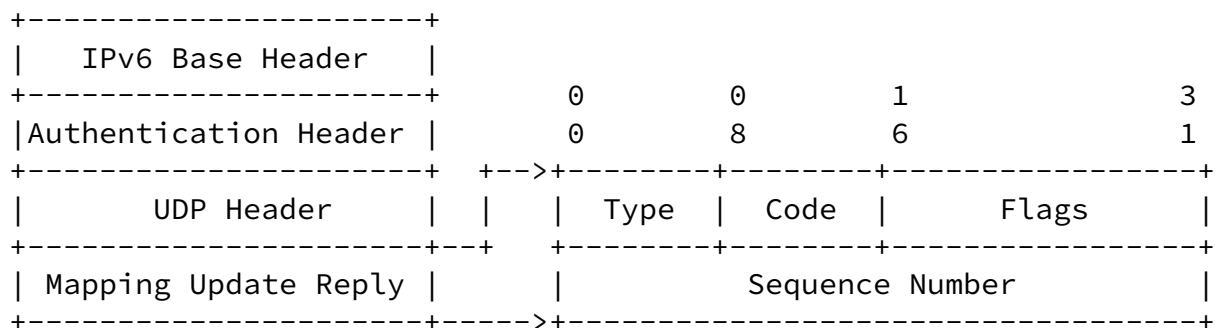
Figure 8: IPv6 (LIN6) header format

#### [4.2.](#) Mapping Update and Reply Messages

When a mobile node moves to another subnet, i.e., when the network prefix of the mobile node changes, the mobile node sends the Mapping Update Message to the Mapping Agent and the correspondent nodes. Upon receiving the Mapping Update Message, the Mapping Agent or the correspondent node returns the Mapping Reply Message to the mobile node. The Mapping Update and Reply Messages are UDP packets. The Authentication Header of IPv6 must be included in the Mapping Update Message to avoid illegal mapping update. Figure 9 shows the formats of the Mapping Update and Reply Messages.



(a) Mapping Update Request Message



(b) Mapping Update Reply Message

Figure 9 Mapping Update Request/Reply formats

Source Address: the LIN6 address of the source node.

Destination Address: the LIN6 address of the destination node.

Source Port: TBD.

Destination Port: TBD.

Type:

0x01: update request

0x02: update reply

Code:

---

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

0x00: succeeded  
0x01: authentication failed  
0x02: ...

Flags: TBD

Sequence Number:

the source node of the Mapping Update Request Message assigns this field a sequence number. This value is copied to this field of the Mapping Update Reply Message.

LIN6 ID: the LIN6 ID of the source node.

Network Prefix: the current network prefix of the source node.

Timestamp: the current time.

Lifetime: the period of time in which this mapping is valid.

#### [4.3.](#) MA Query and Reply Messages

When a node tries to send a packet to a mobile node, the node sends the MA Query Message to the Mapping Agent to obtain the current network prefix of the mobile node. When the Mapping Agent receives the MA Query Message, it returns the MA Reply Message to the node to notify the network prefix of the mobile node. Figure 10 shows the format of the MA Query and Reply Messages.

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

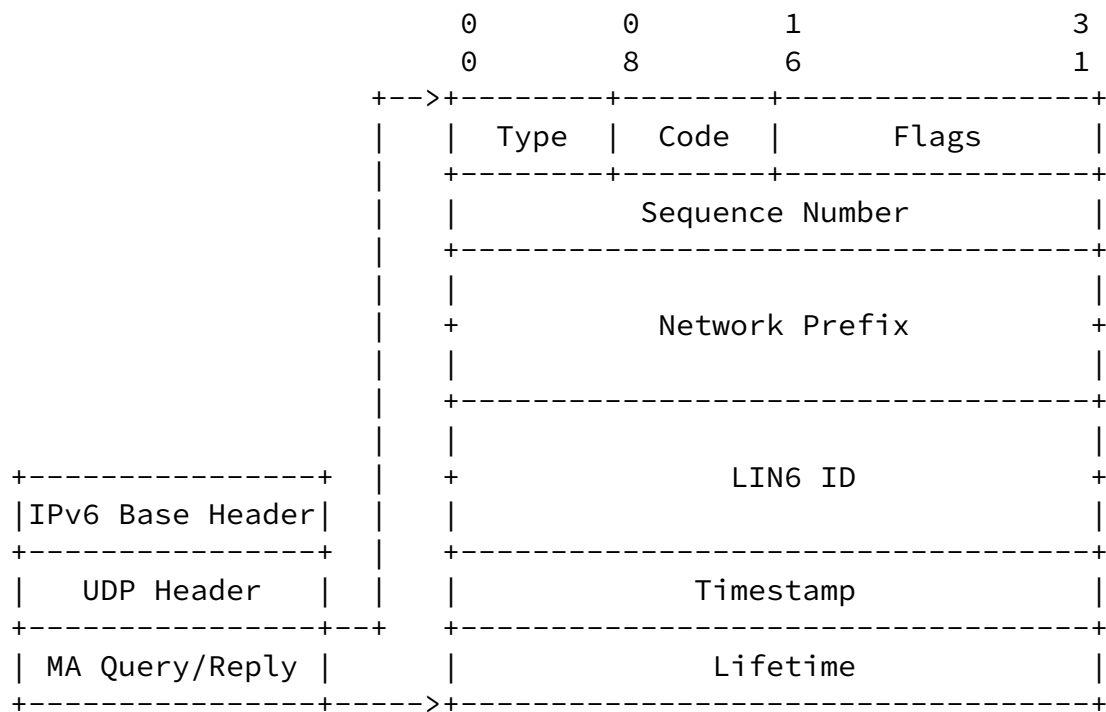


Figure10 MA Query/Reply Message format

Source Address: the LIN6 address of the source node.

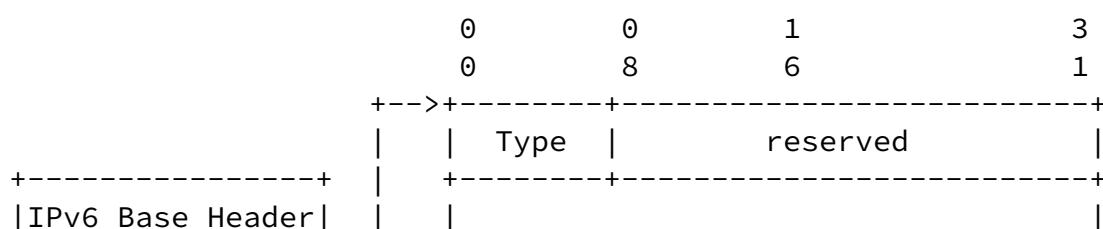
Destination Address: the LIN6 address of the destination node.

Source Port: TBD.

Destination Port: TBD.

Type:

0x01: query



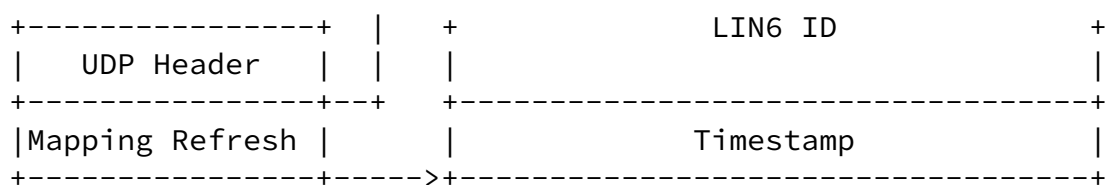


Figure11 Mapping Refresh Message format

Source Address: the LIN6 address of the source node.

Destination Address: the LIN6 address of the destination node.

Source Port: TBD.

Destination Port: TBD.

Type: TBD

LIN6 ID: the LIN6 ID of the target node.

Timestamp: the timestamp of this mapping.

## [5.](#) Processing on the Mobile Node

### [5.1.](#) Bootstrap

When the mobile node is powered on, it obtains the network prefix of the subnet to which it is connected by sending the Router Solicitation Message[RFC2461] and receiving the Router Advertisement Message. Next, the mobile node sends a DNS query packet to obtain the address of the Mapping Agent that maintains the mapping of the mobile node. Next, the mobile node establishes a security association of IPsec[RFC2401] with the Mapping Agent. Next, the mobile node sends the Mapping Update Request Message to the Mapping Agent to register the current network prefix and receives the Mapping Update Reply Message.

## [5.2.](#) Processing on Movement

The mobile node detects the change of the point of attachment to the Internet by some mechanisms, for example, 1) interrupt by hardware, 2) upcall from the link layer, and 3) router advertisement message. When the mobile node detects a location change, first, it sends the Router Solicitation Message and receives the Router Advertisement Message to obtain the network prefix of the subnet to which the mobile node is connected. Next, the mobile node sends the Mapping Update Request Message to the Mapping Agent to notify the current network prefix. The mobile node also sends the Mapping Update Request Message to the correspondent nodes with which it has a security association. The Mapping Update Request Message must include the Authentication Header. The mobile node sends the Mapping Refresh Message to the correspondent nodes with which it has no security association.

## [6.](#) Processing on Mapping Agent

Upon receiving the Mapping Update Request Message from the mobile node, first, the Mapping Agent makes it sure that the Authentication Header is correct. If authentication fails, the Mapping Agent returns the Mapping Update Reply Message with the error code Authentication Failed. If authentication succeeds, the Mapping Agent updates the mapping of the mobile node and returns the Mapping Update Reply Message to the mobile node.

If the mobile node is associated with two or more Mapping Agents, consistency among the databases on the Mapping Agents must be kept by some procedures. These procedures are beyond of the scope of this document.

## [7.](#) Packet Transmission and Reception

### [7.1.](#) Packet Transmission



When the network layer receives a packet transmission request from the transport layer, the network layer makes sure that the destination address passed from the TCP/UDP is a LIN6 generalized ID or a normal IPv6 address by checking the upper 64 bits of the destination address. If the destination address is a normal IPv6 address, the network layer executes the normal IPv6 transmission procedure. If the destination address is a LIN6 generalized ID, the network layer executes the LIN6 procedure described below.

The network layer extracts the LIN6 ID from the LIN6 generalized ID and searches the Mapping Cache for the network prefix by using the LIN6 ID as the key. If the network prefix is found, the network layer concatenates the network prefix and the LIN6 ID to create the LIN6 address of the destination node. After that, the network layer executes the normal IPv6 transmission procedure.

If the network prefix of the destination node is not found in the Mapping Cache, the node keeps the packet waiting for transmission, and then sends the MA Query Message to the Mapping Agent to obtain the network prefix. If the source node does not know the address of the Mapping Agent, it obtains the Mapping Agent's address from the name server by indicating the LIN6 ID of the destination address. Upon receiving the MA Reply Message, the node creates the LIN6 address of the destination node, and then executes the normal IPv6 transmission procedure.

## 7.2. Packet Reception

When the network layer receives a packet from the link layer, first the network layer makes sure that the source address of the IPv6 header is a LIN address or a normal IPv6 address. Refer to the next subsection about how to distinguish between the LIN6 address and the normal IPv6 address. If the source address is the normal IPv6 address, the network layer executes the normal IPv6 reception procedure.

If the source address is the LIN6 address, the network layer removes the network prefix part of the LIN6 address, and then attaches the LIN6 prefix to create the LIN6 generalized ID of the source node. After that, the network layer executes the normal IPv6 reception procedure.

## 7.3. Mapping Refresh Message Reception

When a node receives the Mapping Refresh Message, it sends the MA Query Message to the Mapping Agent of the node sending the Mapping Refresh Message. If the node does not know the address of the Mapping Agent, it

---

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

obtains the Mapping Agent's address from the name server by indicating the LIN6 ID of the node sending the Mapping Refresh Message.

## [8.](#) Considerations on [RFC 3582](#)

This section evaluates the characteristics of LIN6 based on [RFC 3582](#)[\[RFC3582\]](#).

### [8.1.](#) Redundancy (3.1.1)

LIN6 supports redundancy. As described above, the transport layer can know all the prefixes that the multihomed site has. Even if some failures (physical failure, logical link failure, routing protocol failure, transit provider failure, or exchange failure) occur, it is possible for the transport layer to communicate with the multihomed site by trying every prefix.

### [8.2.](#) Load Sharing (3.1.2)

LIN6 can support load sharing. In case of packet transmission, the transport layer can specify the source and the destination network prefixes to the network layer. In case of packet reception, the network layer can pass the source and the destination network prefixes to the transport layer. Thus, the transport layer can use multiple routes (transit providers) simultaneously.

### [8.3.](#) Performance (3.1.3)

LIN6 can distribute inbound traffic to protect itself from performance difficulties directly between the site's transit providers. As described in the previous section, the transport layer in the receiving node can know the source network prefix. The receiving node can return packets to the network prefix from which the sender node sent packets.

### [8.4.](#) Policy (3.1.4)

LIN6 obeys policy control of the upper layers. The transport layer can specify the source and the destination network prefixes according to its policy.

### [8.5.](#) Simplicity (3.1.5)

LIN6 requires implementation the LIN6 protocol on hosts and establishment of Mapping Agents. Router modifications are not necessary.

#### [8.6.](#) Transport-Layer Survivability (3.1.6)

As described in Sec. 3.9, LIN6 supports survivability of transport sessions after re-homing.

#### [8.7.](#) Impact on DNS (3.1.7)

LIN6 does not require any additional records in DNS.

#### [8.8.](#) Packet Filtering (3.1.8)

LIN6 does not preclude filtering packets with inappropriate source IP address because LIN6 can use topologically-correct address as the source address if the source LIN6 node can choose the correct exit router.

#### [8.9.](#) Scalability (3.2.1)

LIN6 does not impose any requirements on the routing system. Although each Mapping Agent (MA) announces network prefixes it manages, the number of routes on the Internet backbone does not increase because the prefixes a MA announces can be aggregated into a single route on the backbone.

#### [8.10.](#) Impact on Routers (3.2.2)

LIN6 does not require any modifications to routers.

#### [8.11.](#) Impact on Hosts (3.2.3)

The LIN6 protocol must be implemented on each end node to fully benefit from site-multihoming. A non-LIN6 host can communicate with a LIN6 node even though the route is redundant.

#### [8.12.](#) Interaction between Hosts and the Routing System (3.2.4)

There is no interaction specific to LIN6 between site's hosts and its routing system.

### [8.13.](#) Operations and Management (3.2.5)

It is assumed that a multihomed side has its own Mapping Agents. In addition, secondary Mapping Agents should be established in transit providers. It is also assumed that a multihomed site manages its DNS with which AAAA records and PTR records of hosts in the site are registered. Thus, staff of a multihomed site can configure the site's multihoming system.

### [8.14.](#) Cooperation between Transit Providers (3.2.6)

LIN6 does not require cooperation between transit providers.

### [8.15.](#) Multiple Solutions (3.2.7)

We have no idea in this section.

Teraoka

Expires 29 June 2004

[Page 23]

---

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

### [8.16.](#) Security Considerations (4)

LIN6 depends on DNS and the Mapping Agents. Location registration with the Mapping Agents is authenticated by IPsec or exchanged cookies. Thus, the security level is almost same as Mobile IPv6 which depends on the Home Agents.

## [9.](#) Intellectual Property Right

There are mechanisms included in the following pending patent applications that have been FILED to the Japanese Patent office.

- Japanese application No.2000-005560 Sony
- Japanese application No.2000-036693 Toshiba

These include the mapping agent and address processing mechanisms. It must be stressed that they have only been filed, and have not been granted. They have been filed separately by Sony and Toshiba.

We really intend to contribute to development of the Internet community and to keep a good relationship with IETF. And our primary concern is

to promote our technology so that others may feel it is useful and worthwhile in the spirit of the IETF.

If indeed the mechanisms are granted as patents, the patents will be licensed under reasonable and non-discriminatory conditions to any person who wants to implement the mechanisms.

#### Author's Address

- o Fumio Teraoka  
Keio University  
3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan.  
Phone: +81-45-566-1425  
and  
Sony Computer Science Laboratories, Inc.  
3-14-13 Higashigotanda, Shinagawa-ku, Tokyo 141-0022, Japan.  
Phone: +81-3-5448-4380  
Email: [tera@ics.keio.ac.jp](mailto:tera@ics.keio.ac.jp)
- o Masahiro Ishiyama  
R&D Center, Toshiba.  
1 Komukai Toshiba-Cho, Saiwai-Ku, Kawasaki, Kanagawa 212-8582, Japan.  
Phone: +81-44-549-2238  
Email: [masahiro@isl.rdc.toshiba.co.jp](mailto:masahiro@isl.rdc.toshiba.co.jp)
- o Mitsunobu Kunishi  
Keio University

Teraoka

Expires 29 June 2004

[Page 24]

---

[draft-teraoka-multi6-lin6-00.txt](#)

30 December 2003

3-14-1 Hiyoshi, Kohoku-ku, Yokohama, Kanagawa 223-8522, Japan.  
Phone: +81-45-566-1425  
Email: [kunishi@tokoro-lab.org](mailto:kunishi@tokoro-lab.org)

#### References

- [IEICE01] M. Ishiyama, M. Kunishi, K. Uehara, H. Esaki, and F. Teraoka.  
LINA: A New Approach to Mobility Support in Wide Area Networks,  
IEICE Transactions on Communication, Vol. E84-B, No. 8, August  
2001.
- [WPMC01] M. Ishiyama, M. Kunishi, K. Uehara, and F. Teraoka. An Analysis

of Mobility Handling in LIN6. In Proc. of 4th International Symposium on Wireless Personal Multimedia Communications, September 2001.

- [RFC2460] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. [RFC 2460](#), December 1998.
- [MIPv6] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. Internet Draft [draft-ietf-mobileip-ipv6-24.txt](#), June 2003.
- [RFC2401] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. [RFC 2401](#), November 1998.
- [RFC2374] R. Hinden, M. O'Dell, and S. Deering. An IPv6 Aggregatable Global Unicast Address Format. [RFC 2374](#), July 1998
- [EUI64] IEEE. Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority, <http://standards.ieee.org/regauth/oui/tutorials/EUI64.html>, 1997.
- [RFC3582] J. Abley, B. Black, and V. Gill. Goals for IPv6 Site-Multihoming Architectures. [RFC 3582](#), August 2003.