

Network Working Group  
Request for Comments: 5184  
Category: Experimental

F. Teraoka  
K. Gogo  
K. Mitsuya  
R. Shibui  
K. Mitani  
KEIO University  
May 2008

Unified Layer 2 (L2) Abstractions  
for Layer 3 (L3)-Driven Fast Handover

Status of This Memo

This memo defines an Experimental Protocol for the Internet community. It does not specify an Internet standard of any kind. Discussion and suggestions for improvement are requested. Distribution of this memo is unlimited.

IESG Note

This document is not an IETF Internet Standard. It represents the consensus of the MOBOPTS Research Group of the Internet Research Task Force. It may be considered for standardization by the IETF in the future.

Abstract

This document proposes unified Layer 2 (L2) abstractions for Layer 3 (L3)-driven fast handovers. For efficient network communication, it is vital for a protocol layer to know or utilize other layers' information, such as the form of L2 triggers. However, each protocol layer is basically designed independently. Since each protocol layer is also implemented independently in current operating systems, it is very hard to exchange control information between protocol layers. This document defines nine kinds of L2 abstractions in the form of "primitives" to achieve fast handovers in the network layer as a means of solving the problem. This mechanism is called "L3-driven fast handovers" because the network layer initiates L2 and L3 handovers by using the primitives. This document is a product of the IP Mobility Optimizations (MobOpts) Research Group.

## Table of Contents

1. Introduction .....	3
2. Terminology .....	3
3. Primitives for L2 Abstractions .....	4
4. Definitions of Primitives .....	6
4.1. L2-LinkStatus (Type 1) .....	6
4.2. L2-PoAList (Type 1) .....	6
4.3. L2-PoAFound (Type 2) .....	6
4.4. L2-PoALost (Type 2) .....	6
4.5. L2-LinkUp (Type 2) .....	7
4.6. L2-LinkDown (Type 2) .....	7
4.7. L2-LinkStatusChanged (Type 2) .....	7
4.8. L2-LinkConnect (Type 3) .....	7
4.9. L2-LinkDisconnect (Type 3) .....	8
5. Definitions of Static Parameters .....	8
5.1. Network Interface ID .....	8
6. Definitions of Dynamic Parameters .....	8
6.1. PoA (Point of Attachment) .....	8
6.2. Condition .....	8
6.3. PoA List .....	9
6.4. Enable/Disable .....	9
6.5. Ack/Error .....	9
7. Architectural Considerations .....	9
8. Security Considerations .....	13
9. Acknowledgements .....	14
10. References .....	14
10.1. Normative References .....	14
10.2. Informative References .....	14
Appendix A. Example Scenario .....	16
Appendix B. Example Operation for FMIPv6 .....	17
B.1. Example Operation-1 for FMIPv6 .....	18
B.2. Example Operation-2 for FMIPv6 .....	20
B.3. Experiment .....	21
Appendix C. Example Mapping between L2 Primitives and Primitives in IEEE 802.11 and IEEE 802.16e .....	22
Appendix D. Example Mapping of Primitives and IEEE 802.11 .....	24
D.1. L2-LinkStatus .....	24
D.2. L2-PoAList .....	24
D.3. L2-PoAFound .....	24
D.4. L2-PoALost .....	25
D.5. L2-LinkUp .....	25
D.6. L2-LinkDown .....	25
D.7. L2-LinkStatusChanged .....	25
D.8. L2-LinkConnect .....	26
D.9. L2-LinkDisconnect .....	26
Appendix E. Implementation and Evaluation of the Proposed Model .....	26

## 1. Introduction

Recent years have witnessed the rapid proliferation of wireless networks as well as mobile devices accessing them. Unlike wired network environments, wireless networks are characterized by dynamically changing radio conditions, connectivity, and available bandwidth. For efficient network communication, it is vital for a protocol layer to know or utilize other layers' control information. Mobile IPv4 [2] and Mobile IPv6 [3] have been standardized to support communication with mobile nodes. There are several proposals for seamless handovers in IPv6 networks, such as Fast Handovers for Mobile IPv6 (FMIPv6) [4] and Hierarchical Mobile IPv6 (HMIPv6) [5]. In FMIPv6, the network layer must know in advance the indication of a handover from the link layer to achieve seamless handovers. However, control information exchange between protocol layers is typically not available because each protocol layer is designed independently.

To solve the problem, this document defines nine kinds of L2 abstractions in the form of "primitives" to achieve fast handovers in the network layer. This mechanism is called "L3-driven fast handovers" because the network layer initiates L2 and L3 handovers by using the primitives.

IEEE 802.21 [6] also defines several services that make use of L2 information. For the sake of ease of implementation and deployment, the primitives defined in this document make use of only the information available in the mobile node, while IEEE 802.21 [6] introduces the information server in the network to provide the mobile node with network-related information, such as a global network map.

This document represents the consensus of the MobOpts Research Group. It has been reviewed by Research Group members active in the specific area of work.

## 2. Terminology

This document uses the following terms:

### L3-Driven Fast Handover

The handover mechanism that is initiated by the network layer on a mobile node. Since this mechanism allows handover preparation in L3 before the start of an L2 handover on the mobile node, it can reduce packet loss during a handover. The L3-driven fast handover mechanism requires L2 information as a trigger for a handover procedure.



For unified L2 abstractions for L3-driven fast handovers, three different usages of primitives are defined, as described below:

Type 1. To provide L2 information to upper layers immediately

This type of primitive is used to provide the L2 information to upper layers immediately. The Request and Confirm classes of primitives MUST be exchanged for the interaction. The Request primitive is for an acquisition request for the L2 information. The Confirm primitive is for the answer.

Type 2. To notify upper layers of L2 events asynchronously

This type of primitive is used to notify upper layers of L2 events asynchronously. The Request, Confirm, and Indication classes of primitive MUST be exchanged, and the Response class MAY be exchanged for the interaction. The Request and Confirm primitives are used just for registration. When an event occurs, the Indication primitive is asynchronously delivered to the upper layer.

Type 3. To control L2 actions from upper layers

This type of primitive is used to control L2 actions from upper layers. The Request and Confirm classes of primitives MUST be exchanged for the interaction. The Request primitive is a request for operation. Ack or Nack returns immediately as the Confirm primitive.

A protocol entity can register primitives anytime by exchanging the Request and Confirm messages that include the fields defined above. When the registered event occurs, the Indication and Response messages are exchanged as well.

The way to exchange a message between protocol entities is beyond the scope of this document. Any information-exchange method between layers, such as the work in [10], can be used.

The timing for sending an Indication primitive is also beyond the scope of this document. For example, a layer 2 event is generated when layer 2 status has been changed, and this depends upon how scanning algorithms, for example, are implemented.

#### 4. Definitions of Primitives

To obtain and exchange L2 information, the following primitives are defined. Appendix C shows example mapping between the L2 primitives and the primitives in IEEE 802.11 [7] and IEEE 802.16e [9].

##### 4.1. L2-LinkStatus (Type 1)

The L2-LinkStatus.request primitive is sent to the link layer when an upper layer requires the current information of a link. The L2-LinkStatus.request primitive contains the "Network Interface ID" parameter (see Section 5.1). In response, the L2-LinkStatus.confirm primitive returns. The L2-LinkStatus.confirm primitive contains three parameters: "Network Interface ID", "PoA", and "Condition". "PoA" and "Condition" indicate the current status of the link between the mobile node and a PoA.

##### 4.2. L2-PoAList (Type 1)

The L2-PoAList.request primitive is sent to the link layer when an upper layer requires a list of the candidate PoAs. The L2-PoAList.request primitive contains the "Network Interface ID" parameter. In response, the L2-PoAList.confirm primitive returns the "Network Interface ID" parameter and the "PoA List" parameter. The "PoA List" parameter is a list of the candidate PoAs.

##### 4.3. L2-PoAFound (Type 2)

The L2-PoAFound.indication primitive is asynchronously provided to an upper layer when new PoAs are detected. This primitive carries the "Network Interface ID" parameter and the "PoA List" parameter. The "PoA List" parameter contains information on new PoAs detected by the mobile node. In order to use this notification, the registration process using the L2-PoAFound.request primitive and the L2-PoAFound.confirm primitive is needed in advance. The L2-PoAFound.request primitive has two parameters: "Network Interface ID" and "Enable/Disable". The "Enable/Disable" parameter shows whether this notification function is turned on. When this registration succeeds, the L2-PoAFound.confirm primitive returns with the "Network Interface ID" parameter and the "Ack" parameter in response.

##### 4.4. L2-PoALost (Type 2)

The L2-PoALost.indication primitive is asynchronously provided to an upper layer when a PoA included in the list of candidate PoAs disappears. This primitive carries the "Network Interface ID" parameter and the "PoA List" parameter. The "PoA List" parameter

contains information on the PoAs that disappeared from the list of candidates. The registration process using the L2-PoALost.request primitive and the L2-PoALost.confirm primitive is similar to the L2-PoAFound primitive described above.

#### 4.5. L2-LinkUp (Type 2)

The L2-LinkUp.indication primitive is asynchronously provided to an upper layer when a new link is connected and IP packets can be transmitted through the new link. As described in RFC 4957 [12], what "link is connected" means depends on link types. For example, in case of the infrastructure mode in IEEE 802.11 [7] (WiFi), this primitive is provided when an association to an access point is established. This primitive carries the "Network Interface ID" parameter and the "PoA" parameter. The L2-LinkUp.request primitive contains the "Network Interface ID" parameter and the "Enable/Disable" parameter for registration. When the registration succeeds, the L2-LinkUp.confirm primitive with the "Network Interface ID" parameter and the "Ack" parameter returns.

#### 4.6. L2-LinkDown (Type 2)

The L2-LinkDown.indication primitive is asynchronously provided to an upper layer when an existing link is disconnected and IP packets cannot be transmitted through the link. The registration processing is the same as the L2-LinkUp primitive described above.

#### 4.7. L2-LinkStatusChanged (Type 2)

The L2-LinkStatusChanged.indication primitive is asynchronously provided to an upper layer when the status of a link has changed. This notification contains three parameters: "Network Interface ID", "PoA", and "Condition". The "PoA" parameter indicates the attachment point at which the link quality changed. In the registration processing, the L2-LinkStatusChanged.request primitive carries the "Network Interface ID" parameter, the "Enable/Disable" parameter, and the "Condition" parameter. "Condition" indicates the event type and the threshold for the Indication.

#### 4.8. L2-LinkConnect (Type 3)

The L2-LinkConnect.request primitive is sent to the link layer when an upper layer has to establish a new link to the specific "PoA". This primitive carries the "Network Interface ID" parameter and the "PoA" parameter. This operation begins after the link layer returns the L2-LinkConnect.confirm primitive with "Ack".

#### 4.9. L2-LinkDisconnect (Type 3)

The L2-LinkDisconnect.request primitive is sent to the link layer when an upper layer has to tear down an existing link to the specific "PoA". This primitive carries the "Network Interface ID" parameter and the "PoA" parameter. This operation begins after the link layer returns the L2-LinkDisconnect.confirm primitive with "Ack".

### 5. Definitions of Static Parameters

This section lists static parameters. Once the values of static parameters are configured, they basically remain unchanged during communication. The following parameters are transferred as a part of primitives.

#### 5.1. Network Interface ID

The "Network Interface ID" parameter uniquely identifies the network interface in the node. The syntax of the identifier is implementation-specific (e.g., name, index, or unique address assigned to each interface). This parameter also contains the network interface type that indicates the kind of technology of the network interface (e.g., IEEE 802.11a/b/g [7], Third Generation Partnership Project (3GPP), etc.). This parameter is required in all primitives.

### 6. Definitions of Dynamic Parameters

This section lists dynamic parameters. The values of dynamic parameters change frequently during communication. The following parameters are transferred as a part of primitives.

#### 6.1. PoA (Point of Attachment)

The "PoA" parameter uniquely identifies the PoA.

#### 6.2. Condition

The "Condition" parameter consists of the following sub-parameters: available bandwidth and link quality level. These sub-parameters are the abstracted information that indicates the current quality of service. The abstraction algorithms of sub-parameters depend on hardware devices and software implementation. The useful range of link quality is divided into five levels: EXCELLENT, GOOD, FAIR, BAD, and NONE, in descending order. The quality levels of an L2 device

are independent of those of other devices. However, making decisions based on these metrics is error prone and not guaranteed to result in an optimal choice of links. An example of the thresholds among the five levels in IEEE 802.11 [7] is described in Appendix E.

### 6.3. PoA List

The "PoA List" parameter consists of arbitrary couples of two sub-parameters: "PoA" and "Condition". This parameter shows a list of PoAs and their conditions.

### 6.4. Enable/Disable

The "Enable/Disable" flag is used for turning event notification on/off. When an upper layer needs notifications, the Request primitive with "Enable" is sent to the link layer as registration. When an upper layer needs no more notifications, the Request primitive with "Disable" is sent.

### 6.5. Ack/Error

When an upper layer requests some notifications, the link layer receives and confirms this Request. If the Request is valid, the Confirm primitive with "Ack" is sent to the upper layer. If it is invalid, the Confirm with "Error" is sent to the upper layer.

## 7. Architectural Considerations

RFC 4907 [11] discusses the role and the issues of link indications within the Internet Architecture. This section discusses the architectural considerations mentioned in Section 2 of RFC 4907.

1. Proposals should avoid use of simplified link models in circumstances where they do not apply.

The information in each layer should be abstracted before it is sent to another layer. For example, in IEEE 802.11 [7], the Received Signal Strength Indicator (RSSI), the number of retransmissions, and the existence of association between the mobile node and the access point are used so that the link layer indications can adjust themselves to various environments or situations. The thresholds needed for some link indications are defined from empirical study.

In the conventional protocol-layering model, the Protocol Entity (PE) is defined as the entity that processes a specific protocol. Our proposal introduced the Abstract Entity (AE) to

achieve link independency of the link indications. An AE and a PE make a pair. An AE abstracts the PE-dependent information to the PE-independent information.

Figure 2 shows AEs and PEs using primitives.

2. Link indications should be clearly defined, so that it is understood when they are generated on different link layers.

To make the link information/indications clear, our proposal defines the 4 types of primitives: Request/Confirm and Indication/Response, as described in Section 3. The Request is used to obtain the information of another layer. The Confirm is the reply to the request and it includes the requested information. The Indication is generated when a particular event occurs. The Response is the reply to the indication.

In our proposal on IEEE 802.11b [7], L2-LinkUp is defined as the status in which an association to the Access Point (AP) is established, and L2-LinkDown is defined as the status in which an association to the AP is not established. L2-LinkStatusChanged is generated when the link quality goes below the predefined threshold. Since the Received Signal Strength Indicator (RSSI) and the number of retransmissions are used to abstract and evaluate the link quality, L2-LinkStatusChanged represents the link quality in both directions. It should use an average of the RSSI or the number of retransmissions damped for one second or more to cope with transient link conditions.

3. Proposals must demonstrate robustness against misleading indications.

Since RSSI changes significantly even when the mobile node stands still according to the measurements in our experiments, our proposal uses the RSSI, the number of retransmissions, and the existence of an association to calculate the link status, as described above. In our experiments, there were some "ping-pong" handovers between two APs. Such ping-pong handovers could be reduced by detecting the most suitable AP by L2-LinkStatus when L2-LinkStatusChanged is notified. The use of L2 indications is related to parameter thresholds that trigger handover. These thresholds vary based on the deployment scenario and, if not configured properly, could lead to misleading indications.

4. Upper layers should utilize a timely recovery step so as to limit the potential damage from link indications determined to be invalid after they have been acted on.

The proposed L3-driven handover described in Appendix E uses the L2-LinkStatusChanged indication as the trigger for starting handover. L2-LinkStatusChanged is indicated when the link quality goes below a specific threshold. This indication is not canceled even if the link quality goes up soon. As described above, L2-LinkStatus can be used to detect the most suitable AP. The IP layer can cancel a handover if it finds that the current AP is the most suitable one by using L2-LinkStatus when L2-LinkStatusChanged is notified.

5. Proposals must demonstrate that effective congestion control is maintained.

Since this mechanism is coupled to the IP layer, and not directly to the transport layer, the proposed mechanism does not directly affect congestion control.

6. Proposals must demonstrate the effectiveness of proposed optimizations.

In IPv6 mobility, the L3-driven handover mechanism using link indications can dramatically reduce gap time due to handover. The L3-driven handover mechanism needs the L2-LinkStatusChanged indication to predict disconnection. But L2-LinkStatusChanged is not trusted sometimes because it is difficult to abstract the link quality. Invalid L2-LinkStatusChanged may cause redundant handover. A handover mechanism using only L2-LinkUp/L2-LinkDown can also reduce gap time modestly. An example of an implementation and evaluation of the L3-driven handover mechanism is described in Appendix E.

7. Link indications should not be required by upper layers in order to maintain link independence.

Our proposal does not require any modifications to the transport and upper layers.

8. Proposals should avoid race conditions, which can occur where link indications are utilized directly by multiple layers of the stack.

Since our proposal defines the link indications only to the IP layer, race conditions between multiple layers never occur.

9. Proposals should avoid inconsistencies between link and routing layer metrics.

Our proposal does not deal with routing metrics.

10. Overhead reduction schemes must avoid compromising interoperability and introducing link-layer dependencies into the Internet and transport layers.

As described above, the link indications in our proposal are abstracted to the information independent of link types to reduce the gap time due to a handover, and the ordinary host can execute handover without using the link indications defined in our proposal.

11. Proposals advocating the transport of link indications beyond the local host need to carefully consider the layering, security, and transport implications. In general, implicit signals are preferred to explicit transport of link indications since they add no new packets in times of network distress, operate more reliably in the presence of middle boxes, such as NA(P)Ts (Network Address (Port) Translations), are more likely to be backward compatible, and are less likely to result in security vulnerabilities.

Our proposal does not define the exchange of link indications between nodes.

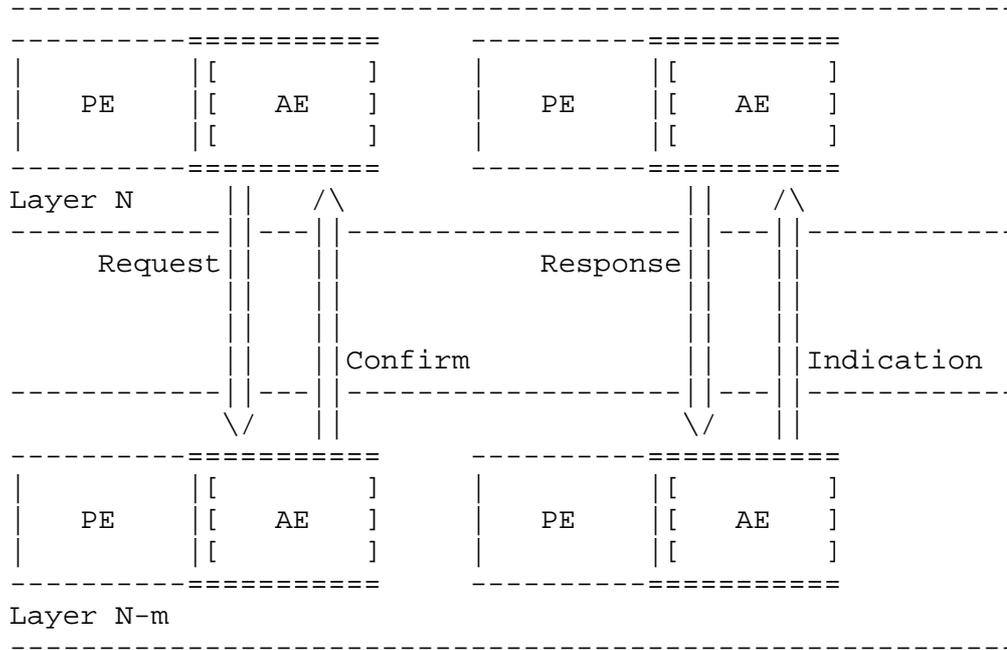


Figure 2: AE and PE with Primitives

## 8. Security Considerations

RFC 4907 [11] discusses the role and issues of link indications within the Internet Architecture. This section discusses the security considerations mentioned in Section 4 of RFC 4907.

### 1. Spoofing

The proposed primitives suffer from spoofed link-layer control frames. For example, if a malicious access point is set up and spoofed beacon frames are transmitted, L2-PoAFound.indication is generated in the mobile node. As a result, the mobile node may establish an association with the malicious access point by an L2-LinkConnect.request.

### 2. Indication validation

Transportation of the link indications between nodes is not assumed; hence, this consideration is beyond the scope of our proposal.

### 3. Denial of service

Since this proposal does not change link-layer protocols, no more insecurity is added to a particular link-layer protocol. However, the proposed primitives suffer from denial-of-service attacks by spoofed link-layer frames. For example, L2-PoAFound.indication and L2-PoALost.indication may frequently be generated alternately if a malicious access point frequently transmits control frames that indicate strong RSSI and weak RSSI alternately.

### 9. Acknowledgements

The authors gratefully acknowledge the contributions of Jukka Manner, Christian Vogt, and John Levine for their review.

### 10. References

#### 10.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

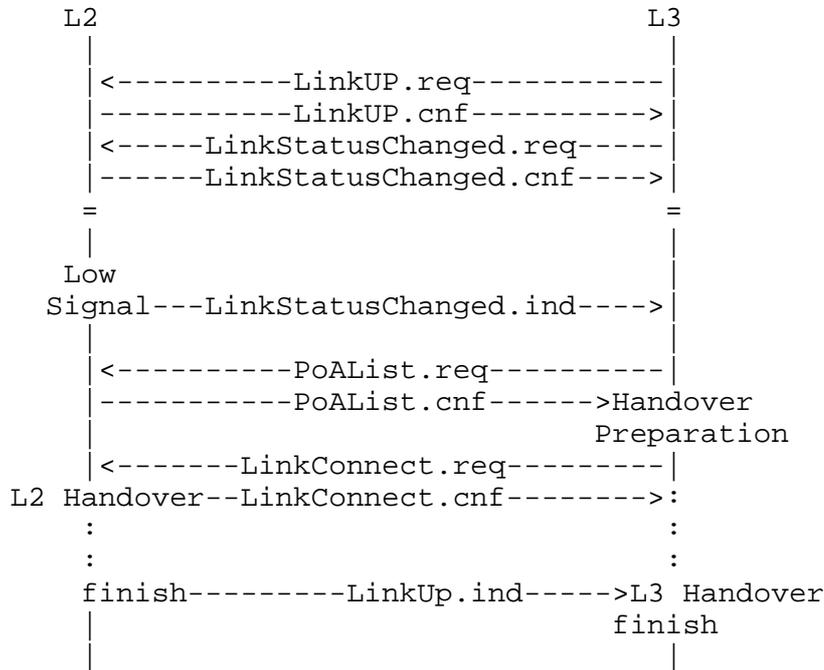
#### 10.2. Informative References

- [2] Perkins, C., Ed., "IP Mobility Support for IPv4", RFC 3344, August 2002.
- [3] Johnson, D., Perkins, C., and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
- [4] Koodli, R., Ed., "Fast Handovers for Mobile IPv6", RFC 4068, July 2005.
- [5] Soliman, H., Castelluccia, C., El Malki, K., and L. Bellier, "Hierarchical Mobile IPv6 Mobility Management (HMIPv6)", RFC 4140, August 2005.
- [6] "Draft IEEE Standard for Local and Metropolitan Area Networks: Media Independent Handover Services", IEEE P802.21/D02.00, September 2006.
- [7] IEEE, "802.11-2007 IEEE Standard for LAN/MAN - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 2007.

- [8] IEEE, "802.3, 2000 EDITION ISO/IEC 8802-3:2000 (E) Information Technology - LAN/MAN - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications", 2000.
- [9] IEEE, "802.16e-2005 & 802.16/COR1 Part 16: Amendment for Physical & Medium Access Control Layers for Combined Fixed & Mobile Operation", 2006.
- [10] Gogo, K., Shibu, R., and F. Teraoka, "An L3-Driven Fast Handover Mechanism in IPv6 Mobility", In Proc. of International Symposium on Applications and the Internet (SAINT2006) Workshop in IPv6, February 2006.
- [11] Aboba, B., Ed., "Architectural Implications of Link Indications", RFC 4907, June 2007.
- [12] Krishnan, S., Ed., Montavont, N., Njedjou, E., Veerepalli, S., and A. Yegin, Ed., "Link-Layer Event Notifications for Detecting Network Attachments", RFC 4957, August 2007.
- [13] Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H., and F. Teraoka, "LINA: A New Approach to Mobility Support in Wide Area Networks", IEICE Transactions on Communication vol. E84-B, no. 8, pp. 2076-2086, August 2001.

## Appendix A. Example Scenario

For example, the picture below shows L3-driven fast handover mechanism using the L2 triggers on a mobile node (MN).



L2: Link Layer on MN  
L3: Network Layer on MN  
req: Request  
cnf: Confirm  
ind: Indication

Figure 3: L3-Driven Fast Handover Mechanism

First, L3 issues LinkUp.request to receive LinkUp.indication when the link becomes available. L3 also issues LinkStatusChanged.request to receive LinkStatusChanged.indication when the link quality goes below the threshold.

In the beginning of the L3-driven handover procedure, L2 detects that the radio signal strength is going down. Then, L2 sends L2-LinkStatusChanged.indication to L3. L3 prepares for handover (e.g., Care-of Address (CoA) generation, Duplicate Address Detection (DAD), Neighbor Discovery (ND) cache creation, and routing table setting) and sends L2-PoAList.request to L2 if the list of access points is needed.

If L3 decides to perform handover according to some rules, L3 sends L2-LinkConnect.request with some parameters about candidate access points to request L2 handover. L2 handover begins after L2 sends L2-LinkConnect.confirm to L3. When the L2 handover finishes, L2 sends L2-LinkUp.indication to notify L3. Finally, L3 performs handover (e.g., sending a Binding Update (BU)).

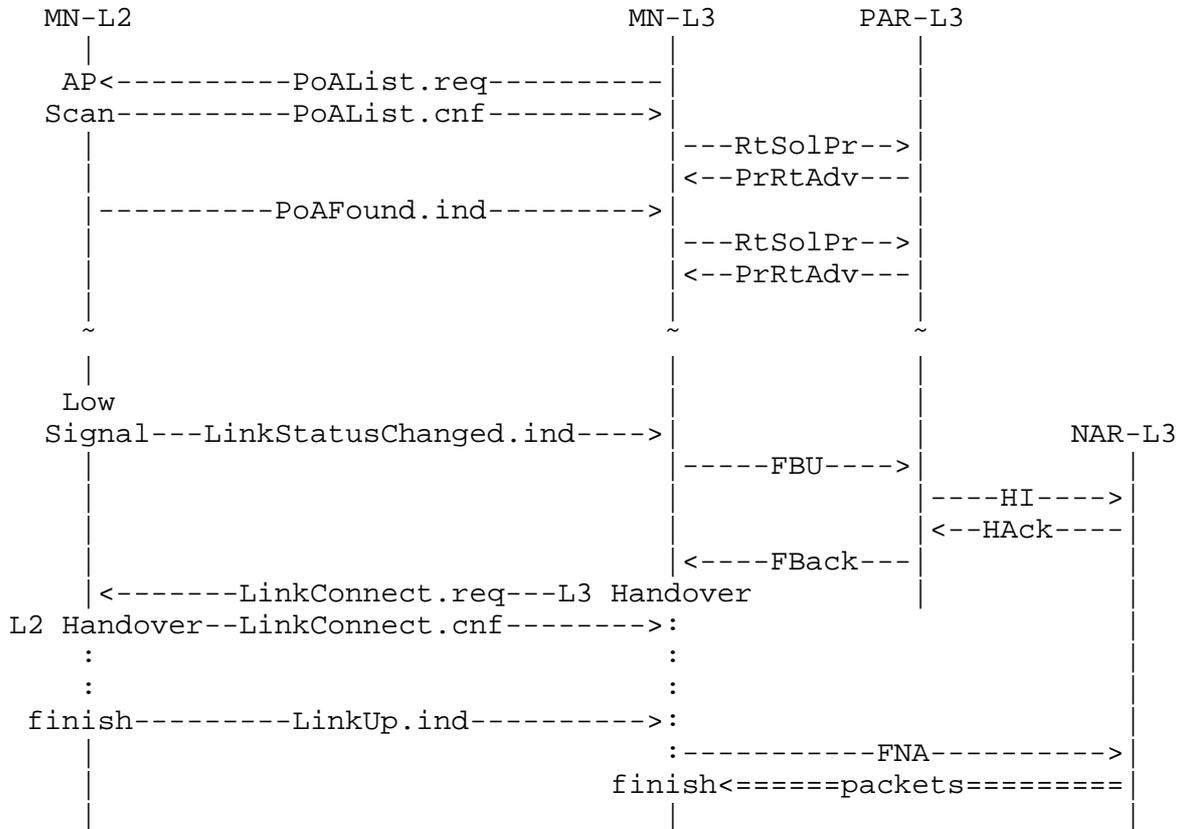
One of the important features of L3-driven fast handover using primitives is that L3 handover preparation can be done during communication. So, it can reduce disruption time during handover.

#### Appendix B. Example Operation for FMIPv6

There are two scenarios of L3-driven fast handover for FMIPv6. Scenario 2 is different from scenario 1 for the timing of sending some messages.

B.1. Example Operation-1 for FMIPv6

Figure 4 shows the predictive mode of FMIPv6 operation with an L3-driven link-switching mechanism.



- MN-L2 : Link Layer on Mobile Node
- MN-L3 : Network Layer on Mobile Node
- PAR-L3 : Network Layer on Previous Access Router
- NAR-L3 : Network Layer on New Access Router
- req : Request
- cnf : Confirm
- ind : Indication
- RtSolPr : Router Solicitation for Proxy
- PrRtAdv : Proxy Router Advertisement
- FBU : Fast Binding Update
- FBack : Fast Binding Acknowledgment
- FNA : Fast Neighbor Advertisement
- HI : Handover Initiate
- HAcK : Handover Acknowledge

Figure 4: L3-Driven Fast Handover Mechanism with FMIPv6 Scenario 1

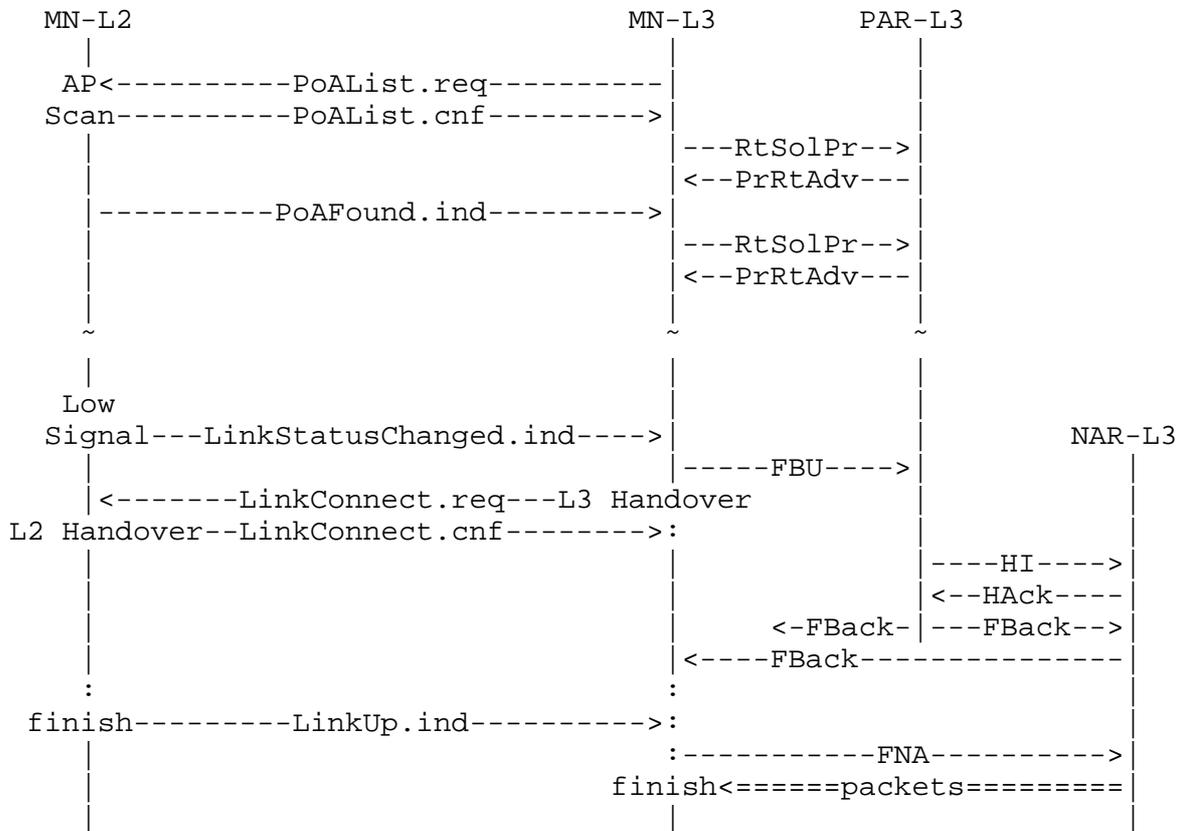
When MN establishes link connectivity to PAR, it performs router discovery. MN sends an RtSolPr message to PAR to resolve the access point identifiers to the subnet router information. To send RtSolPr, MN discovers one or more access points by sending L2-PoAList.request to the link layer. As a response to L2-PoAList.request, L2-PoAList.confirm returns with "PoA List" parameter that contains identifiers and conditions of nearby access points. After initial AP discovery, L2-PoAFound.indication with "PoA List" is also sent from the link layer when one or more access points are discovered.

When the link layer of MN detects that radio signal strength is dropping, it sends L2-LinkStatusChanged.indication to the network layer. Then, MN sends the FBU message to PAR as the beginning of the L3 handover procedure. The NCoA required for the FBU message is determined according to the MN's policy database and the received PrRtAdv message. As a response to the FBU message, MN receives the FBack message and then immediately switches its link by L2-LinkConnect.request with the specific "PoA" parameter. The handover policy of the MN is outside the scope of this document.

After L2 handover, the link layer of the MN sends L2-LinkUp.indication to the network layer. MN immediately sends the FNA message to the New Access Router (NAR). The NAR will send tunneled and buffered packets to MN.

B.2. Example Operation-2 for FMIPv6

Figure 5 shows the predictive mode of FMIPv6 operation with an L3-driven link-switching mechanism.



- MN-L2 : Link Layer on Mobile Node
- MN-L3 : Network Layer on Mobile Node
- PAR-L3 : Network Layer on Previous Access Router
- NAR-L3 : Network Layer on New Access Router
- req : Request
- cnf : Confirm
- ind : Indication
- RtSolPr : Router Solicitation for Proxy
- PrRtAdv : Proxy Router Advertisement
- FBU : Fast Binding Update
- FBack : Fast Binding Acknowledgment
- FNA : Fast Neighbor Advertisement
- HI : Handover Initiate
- HACK : Handover Acknowledge

Figure 5: L3-Driven Fast Handover Mechanism with FMIPv6 Scenario 2

Unlike scenario 1, MN in scenario 2 sends LinkConnect.req from the network layer to the link layer after MN sends the FBU message. As PAR sends the FBack messages not only to PAR's subnet but also to NAR's subnet, MN can get the FBack message sent by PAR through NAR, and then it moves to NAR.

### B.3. Experiment

We implemented FMIPv6 and the proposed L2 primitives on FreeBSD-5.4. Figure 6 shows our test network. MN is connected to access routers by using IEEE802.11a wireless LAN. MN moves from PAR to NAR.

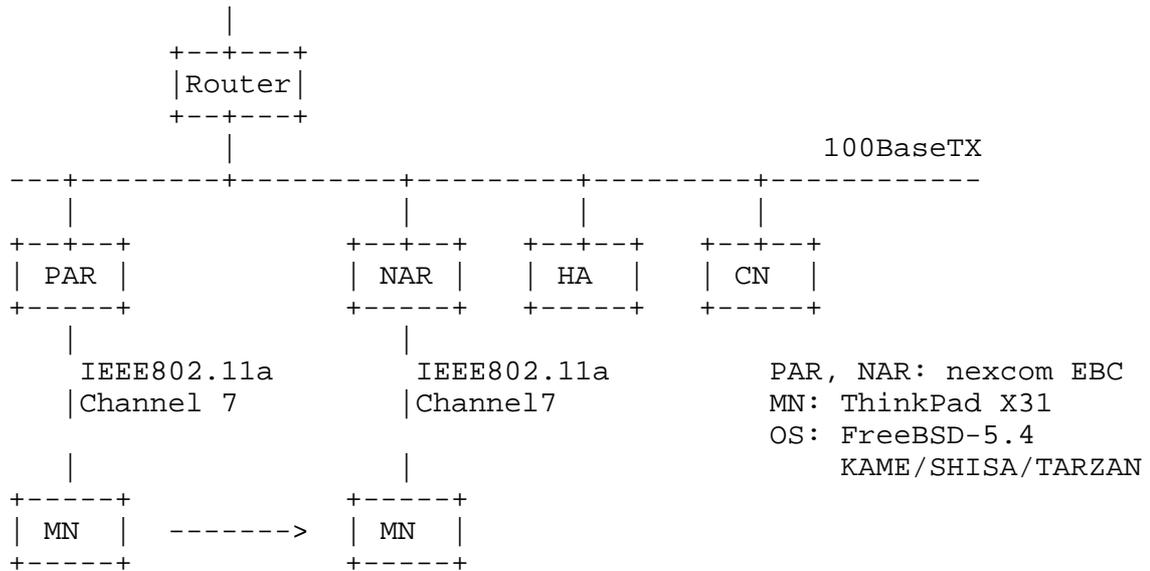


Figure 6: Test Network

Scenario 1 was used in this experiment since it was more stable than scenario 2. Upon receiving L2-LinkStatusChanged.indication, L3 of MN sent the FBU message and then received the FBack message. It took 20ms from the transmission of the FBU message to the reception of the FBack message. After receiving the FBack message, L3 of MN issued L2-LinkConnect.request to L2. When L2 handover finished, L3 received L2-LinkUp.indication from L2. It took 1ms for an L2 handover. Next, MN sent the FNA message to NAR. Upon receiving the FNA message, NAR started forwarding packets to NM. ICMP echo request packets were sent at 10ms intervals. It was observed that zero or one ICMP echo reply packet was lost during a handover.

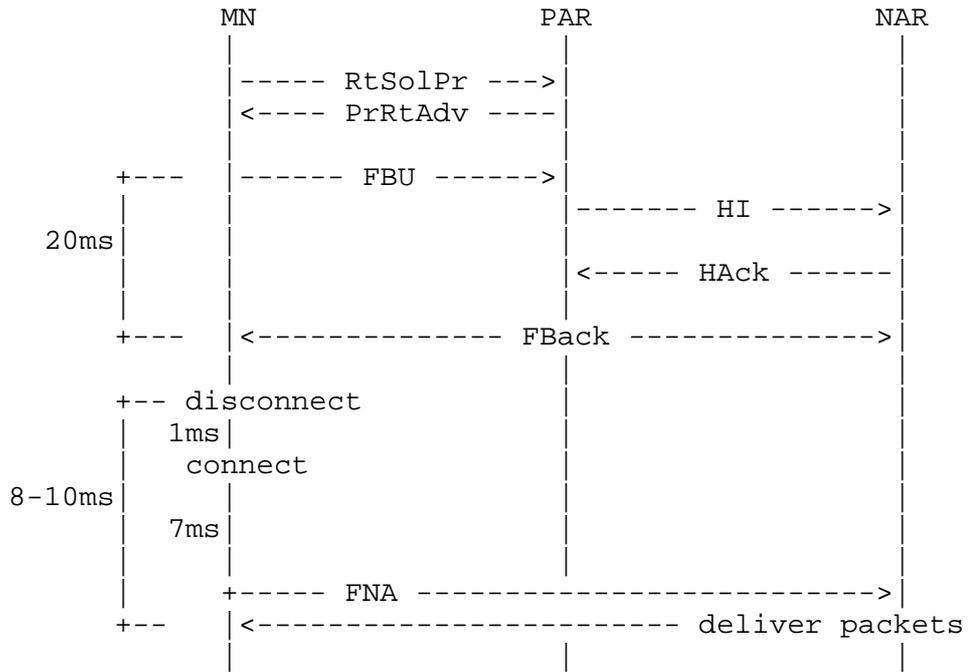


Figure 7: Measured Results

Appendix C. Example Mapping between L2 Primitives and the Primitives in IEEE 802.11 and IEEE 802.16e

This section shows example mapping between the L2 primitives and the primitives in IEEE 802.11 [7] and IEEE 802.16e [9] in Table 1.

L2 Primitive	IEEE802.11	IEEE802.16e
L2-LinkStatus	PMD_RSSI	Available
	PMD_RATE	
L2-PoAList	MLME-SCAN	M_ScanScheduling
		M_Scanning
L2-PoAFound	MLME-SCAN	M_Neighbor
		M_Scanning
L2-PoALost	MLME-SCAN	M_Neighbor
		M_Scanning
L2-LinkUp	MLME-ASSOCIATE	M_Registration
	MLME-AUTHENTICATE	
L2-LinkDown	MLME-DEASSOCIATE	M_Ranging
	MLME-DISAUTHENTICATE	
L2-StatusChanged	PMD_RSSI	M_Ranging
		M_ScanReport
		M_Scanning
L2-LinkConnect	MLME-JOIN	M_MACHandover
	MLME-ASSOCIATE	M_HOIND
	MLME-REASSOCIATE	
	MLME-AUTHENTICATE	
L2-LinkDisconnect	MLME-DISASSOCIATE	M_Management
	MLME-DEASSOCIATE	(Deregistration)

Table 1: Mapping between L2 Primitives and the Primitives in IEEE 802.11 and IEEE 802.16e

## Appendix D. Example Mapping of Primitives and IEEE 802.11

This section shows examples of the mapping between primitives and IEEE 802.11 [7] parameters.

### D.1. L2-LinkStatus

Most parameters of L2-LinkStatus are related to the configuration of network-interface hardware. The operating system and the device-driver module can easily collect such information. However, to create the "Condition" parameter, the MN should maintain statistics and parameters related to the current wireless environment.

There are two sub-parameters of the "Condition" parameter: available bandwidth and link quality level. The available bandwidth of the current PoA can be obtained by maintaining the transmission rate indication and the statistics of frame transmission every time a frame is sent. A link quality level can be updated by maintaining the following parameters and statistics every time a frame is received: Received Signal Strength Indicator (RSSI), transmission/reception rate indication, transmit/receive latency, bit-error rate, frame-error rate, and noise level. Link quality level is divided into five levels: EXCELLENT, GOOD, FAIR, BAD, and NONE, in descending order. Some parameters can be managed by setting thresholds from software. When the parameters cross the threshold, an interrupt is generated for the software.

### D.2. L2-PoAList

In IEEE 802.11 networks, L2-PoAList returns the detected APs whose quality level exceeds the specified threshold for PoA candidates (by the "PoA List" parameter). Therefore, an MN should always maintain the database of available access points according to reception of beacon frame, probe response frame, and all frames. This AP database is managed in consideration of time, number of frames, and signal strength. There are some kinds of network-interface hardware that can notify events to operating system only when the desired event occurs by setting a threshold from software. Moreover, MN can transmit the probe request frame for access point discovery, if needed.

### D.3. L2-PoAFound

In IEEE 802.11 networks, L2-PoAFound is notified when new PoAs whose link quality level exceeds the specified threshold are detected by listening beacons or scanning APs. If the received frame (mainly the

beacon or the probe response) is not in the AP database described in Appendix D.2, then the link layer creates L2-PoAFound.indication.

For example, if the threshold of link quality level specified by L2-PoAFound.request is GOOD, L2-PoAFound.indication is created and sent to the upper layer when PoA's link quality becomes better than GOOD.

#### D.4. L2-PoALost

In IEEE 802.11 networks, L2-PoALost is notified when an AP included in the list of candidate APs is lost by listening beacons or scanning APs. If the entry in the AP database described in Appendix D.2 expires, or link quality level goes under the threshold, then the link layer creates L2-PoALost.indication. To calculate the link quality level, the signal strength of the received frame (mainly the beacon or the probe response) can be used. For example, if the threshold of the link quality specified by L2-PoALost is BAD, L2-PoALost.indication is created and sent to the upper layer when PoA's link quality becomes worse than BAD.

#### D.5. L2-LinkUp

In IEEE 802.11 networks, L2-LinkUp is notified when association or reassociation event occurs. When such an event occurs, MN receives the association response frame or the reassociation response frame. Immediately after receiving it, the link layer creates L2-LinkUp.indication.

#### D.6. L2-LinkDown

In IEEE 802.11 networks, L2-LinkDown is notified when a disassociation event occurs or when no beacon is received during a certain time. When such an event occurs, MN sends the disassociation frame to AP, or the entry of the specific AP is deleted from the AP database described in Appendix D.2. By detecting such events, the link layer creates an L2-LinkDown.indication.

#### D.7. L2-LinkStatusChanged

In IEEE 802.11 networks, L2-LinkStatusChanged is notified when the radio signal strength of the connected AP drops below the specified threshold.

#### D.8. L2-LinkConnect

In IEEE 802.11 networks, each AP is identified by the BSSID and the service set of several APs is identified by the SSID. When L2-LinkConnect is used to connect a specific AP or a service set, the link layer should set the Basic Service Set Identifier (BSSID) or the Service Set Identifier (SSID). Also, the channel should be set appropriately at the same time by searching the database described in Appendix D.2. To connect to AP, MN should be authenticated by AP. MN sends the authentication message to AP, and then MN sends the association message to associate with AP.

#### D.9. L2-LinkDisconnect

In IEEE 802.11 networks, MN sends the disassociation message to AP for disconnection. When L2-LinkDisconnect is used for disconnection from the current AP, the link layer should send the disassociation message to the appropriate AP, and stop data transmission.

### Appendix E. Implementation and Evaluation of the Proposed Model

This section describes an implementation of the proposed link indication architecture and its evaluation.

An IEEE 802.11a wireless LAN device driver was modified to provide abstract link-layer information in the form of primitives defined in Section 4. The modified driver has two AP lists. One contains the device-dependent information such as RSSI, retransmission count, various AP parameters, and various statistics. The device-dependent information, except for the AP parameters, is updated whenever the device receives a frame. If the received frame is the management frame, the AP parameters are also updated according to the parameters in the frame.

Another AP list contains the abstract information. The abstract information is updated periodically by using the device-dependent information. In the abstraction processing, for example, RSSI or the retransmission count is converted to the common indicator "link quality". In our outdoor testbed described below, the thresholds of the RSSI value between the link quality levels were defined as follows:

- o EXCELLENT -- 34 -- GOOD
- o GOOD -- 27 -- FAIR
- o FAIR -- 22 -- BAD

- o BAD -- 15 -- NONE

L2-PoAList and L2-LinkStatus were implemented by using only the abstract information. Thus, the upper layers can use information of the current AP and the adjacent APs without depending on the devices. L2-PoAFound or L2-PoALost is notified if the link quality rises or falls beyond the thresholds when the abstract information is updated. If the link quality of the current AP goes below the specific threshold, L2-LinkStatusChanged is notified. L2-LinkUp or L2-LinkDown is notified when an association with an AP is established or destroyed. To realize L2-LinkConnect and L2-LinkDisconnect, functions to connect or disconnect to the specified AP were implemented. In these functions, since only abstract information is needed to specify the AP, other layers need not know the device-dependent information.

In our outdoor testbed, there are eight Access Routers (ARs) located at 80-100m intervals. AP is collocated at AR. IEEE 802.11a was used as the link layer. The same wireless channel was used at all APs. Thus, there are eight wireless IPv6 subnets in the testbed. The mobile node was in a car moving at a speed of 30-40 km/h. When link quality of the current AP goes down, the mobile node executes L3-driven handover, described in Appendix A. An application called Digital Video Transport System (DVTS) was running on the mobile node and sent digital video data at a data rate of about 15Mbps through the wireless IPv6 subnet to the correspondent node, which replayed received digital video data. In this environment, the L2 handover required less than 1 msec, and the mobility protocol Location Independent Networking in IPv6 (LIN6) [13] required a few msec for location registration. Thus, the total gap time due to the handover was 3-4 msec. In most cases, there was no effect on the replayed pictures due to handover.

#### Authors' Addresses

Fumio Teraoka  
Faculty of Science and Technology, KEIO University  
3-14-1 Hiyoshi, Kohoku-ku  
Yokohama, Kanagawa 223-8522  
Japan

Phone: +81-45-566-1425  
EMail: [tera@ics.keio.ac.jp](mailto:tera@ics.keio.ac.jp)  
URI: <http://www.tera.ics.keio.ac.jp/>

Kazutaka Gogo  
Graduate School of Science and Technology, KEIO University  
3-14-1 Hiyoshi, Kohoku-ku  
Yokohama, Kanagawa 223-8522  
Japan

Phone: +81-45-566-1425  
EMail: gogo@tera.ics.keio.ac.jp  
URI: <http://www.tera.ics.keio.ac.jp/>

Koshiro Mitsuya  
Jun Murai Lab, Shonan Fujisawa Campus, KEIO University  
5322 Endo  
Fujisawa, Kanagawa 252-8520  
Japan

Phone: +81-466-49-1100  
EMail: mitsuya@sfc.wide.ad.jp

Rie Shibui  
Graduate School of Science and Technology, KEIO University  
3-14-1 Hiyoshi, Kohoku-ku  
Yokohama, Kanagawa 223-8522  
Japan

Phone: +81-45-566-1425  
EMail: shibrie@tera.ics.keio.ac.jp  
URI: <http://www.tera.ics.keio.ac.jp/>

Koki Mitani  
Graduate School of Science and Technology, KEIO University  
3-14-1 Hiyoshi, Kohoku-ku  
Yokohama, Kanagawa 223-8522  
Japan

Phone: +81-45-566-1425  
EMail: koki@tera.ics.keio.ac.jp  
URI: <http://www.tera.ics.keio.ac.jp/>

## Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78 and at <http://www.rfc-editor.org/copyright.html>, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

