Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2012

H. Yokota
KDDI Lab
F. J. Lin
Telcordia Technologies
A. Dutta
NIKSUN
C. Williams
MCSR Labs
V. Manral
IPInfusion Inc.

October 18, 2011

Managing Service Mobility for Virtualized Networks
draft-yokota-opsawg-virtnw-service-management-02.txt

Abstract

In virtualized networks, machines with processing and storage are virtualized resources on which network functional entities can be allocated and relocated dynamically. Such dynamic allocation and relocation of network entities imposes the challenge of service mobility, i.e., how to maintain not only coupling relations between those network entities but also sessions established between those entities in a virtualized environment. This document provides a reference model for managing service mobility in the virtual environment and defines the control protocol between the virtualized platform and the managing controller to realize service mobility. Such capability is especially longed for realizing more scalable and reliable telecom networks.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on April 20, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Requirements notation 4
2. Introduction 5
3. Overview and Terminology 6
4. Service Mobility Requirements 10
5. Protocol Operations 11
6. Message Format 15
6.1. Common header format 15
6.2. Control messages 17
6.2.1. REGISTRATION 17
6.2.2. KEEP-ALIVE 17
6.2.3. OPERATION 17
6.2.4. STATUS-UPDATE 17
6.2.5. DE-REGISTRATION 18
6.2.6. ACK 18
6.3. Operation commands 19
6.3.1. GET Operation 19
6.3.2. ADD Operation 19
6.3.3. DELETE Operation 20
6.3.4. MOVE Operation 20
6.3.5. COPY Operation 21
6.3.6. MOVE_SESSION Operation 22
6.4. Option values 22
6.4.1. IPv4 address 22
6.4.2. IPv6 address 23
6.4.3. Port number 23
6.4.4. Node ID 23
6.4.5. Function ID 23
6.4.6. Node information 23
6.4.7. Function information 24
6.4.8. Session information 24
6.4.9. Vendor-specific information 25
7. IANA Considerations 26
8. Security Considerations 27
9. Acknowledgments 28
10. References 29
10.1. Normative references 29
10.2. Informative references 29
Authors' Addresses 30

1. Requirements notation

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Introduction

There is a growing consensus that a significant number of services will be delivered using a virtualized network. These applications and services will be hosted in data centers located in various parts of the network. New services will be instantiated by deploying a service delivery system when needed such that the network and data center resources are allocated dynamically.

At the same time, telecom networks are moving toward all-IP based systems such as 3GPP SAE (System Architecture Evolution), IMS (IP Multimedia Subsystem) and SDP (Service Delivery Platform), whereby the transport and services are being handled by IP technology. These systems are composed of a number of components or functional entities closely coupled with each other, which makes the whole telecom network even more complex. Wide variety of services ranging from conventional voice, gaming to Web2.0 type of ones such SNS (Social Networking Service), are being introduced and the number of users also dynamically changes. In such environments, the telecom network needs to scale on demand with efficient use of infrastructure as well as to improve reliability and availability.

It is imperative that such ever-evolving new services be seen by the user as having no delay and can be accessed from anywhere with high reliability. If service components or functional entities in the telecom networks (e.g., call control function or policy control function) can be located and moved without any topological or geographical constraint, higher reliability can be achieved than conventional redundancy systems. By tapping into network virtualization technology, telecom network deployed in a virtualized environment (or virtualized telecom network), will be able to scale telecom services on demand, to improve reliability and availability and to efficiently use the infrastructure [IMSAA09]. One of the key features provided by the virtualized telecom network is service mobility. A general property of service mobility is to ensure service location is transparent to service user, that is, the on-going service must be continued in a transparent manner no matter where it is allocated and existing protocols or interfaces used by the service should not be affected.

This document provides a reference model for managing service mobility in a virtual environment and defines the control protocol between the virtualized platform (e.g., virtual machines) and the managing controller to enable service mobility.

3. Overview and Terminology

The components in the virtualized network include "Manager Node", "Execution Node" and optional "Information Server". The management role for the Execution Nodes can be realized in both a centralized way as well as a distributed (Peer-to-peer) way. The Execution Node is a physical or virtual machine on which target functions (software) are running. For example, in the context of IMS (IP Multimedia Subsystem), the CSCF (Call Session Control Function) and HSS (Home Subscriber Server) are candidates of such functions. Information servers include DHCP, DNS, etc. used for discovery and assignment of Execution Nodes for hosting these IMS functions (Proxy-CSCF at a UE's registration).

Execution Node:
: Logical entity that can execute functions. A well-known example is a virtual machine or VM.

Manager Node:
: Logical entity that manages functions running on the execution nodes.
[Editor's Note] It is for further study if the Manager of Managers (hierarchical structure for Manager Nodes) should be included in the architecture.

Information Server:
: Logical entity used for discovery of Manager Node or Execution Node via IETF standardized protocols (e.g., DNS server or DHCP server). This entity and the mechanisms of discovery are so far outside the scope of this document's specifications.

Function:
: Logical unit that provides a specific service such as "authentication" or "call control"

Session:
: Relationship between functions providing a specific service. Each function maintains a state related to the session.

Service mobility:
: Capability to move function(s) among Execution Nodes while maintaining the ongoing service.

Node ID:
: Uniquely identifies the execution node, which is provided and managed by the manager node.

Function ID:
Uniquely identifies the function, which is provided and managed by the manager node.

The reference model for service mobility is shown in Figure 1. The service user on the upper layer utilizes service units provided by the service provider via the service access point (SAP). By interacting with the management plane, the service unit may move its physical or topological location. This movement must be transparent to the service user meaning that the on-going service must be continued. Existing protocols and interfaces should not be affected by this capability.

```
+-----------------------------------+    +-------+
|                                   |    |       |
|            Service User           |    |       |
|                                   |    |       |
|             |       |             |    |Service|
+-------------|  SAP  |-------------+    |  Mgmt |
|             |       |             |    |       |
|+-------+ +-------+       +-------+|    |       |
||Service| |Service| <...> |Service||<==>|       |
|| Unit  | |  Unit |       |  Unit ||    |       |
|+-------+ +-------+       +-------+|    |       |
|          Service Provider         |    |       |
+-----------------------------------+    +-------+
```

Figure 1: Reference model

Service mobility in the virtual environment is depicted in Figure 2. The virtual environment provides the capability to move Execution Nodes among physical hardware. An Execution Node can run on one physical hardware unit (e.g., a server machine) or on multiple hardware units. How the virtual environment is provided is outside the scope of this document. Functions run on the Execution Nodes and one or more session(s) may be established with the corresponding status information. When functions and/or sessions move from one Execution Node to another, consistency in the status information must be maintained to continue the sessions.

```
                 ^            Session              ^
                 *       *******************      *
           +-----*------*-----+ +---------*-----*-----------+
           |  (Status)  *     | |      (Status)*            |
           |    +-*+   *      | |   +--+ +*-+ *   +--+      |
           |    |FN|*** <.........> |FN| |FN|*    |FN|      |
           |    +--+          | |   +--+ +--+     +--+      |
           | +-------------+  | |  +---------+ +---------+  |
           | |  Execution  |  | |  |Execution| |Execution|  |
           | |Node(e.g.,VM)|  | |  |  Node   | |  Node   |  |
           | +-------------+  | |  +---------+ +---------+  |
           +------------------+ +---------------------------+
                         ^            ^
                         |            |
                       Physical hardware

           FN=Function
           VM=Virtual Machine
```

Figure 2: Service mobility in virtual environment

The roles of the Manager Node and Execution Nodes and their relationship are depicted in Figure 3. The Manager Node communicates with the Execution Nodes to control the mobility of functions. The Manager Node can be deployed in a centralized or distributed fashion. Physical limitations (CPU, Memory, Bandwidth, etc.) may occur if the number of nodes managed by a single physical server is large. The physical entities comprising the Manager Node may distribute the management functions among themselves; however, that must be transparent to the execution nodes. The Execution Nodes communicate with each other to move functions between the nodes. An external information server or repository may be involved to support those nodes to discover other nodes. The main focus of this document is the control protocol between the Manager Node and execution nodes.

```
                    +-------------+        .............
                    |   Manager   |<......>:Information:
                    |     Node    |        :  Server   :
                    +-------------+        :...........:
                       /      \                   ^
                      /        \                  :
                     v          v                 :
                 +---------+   +---------+         :
                 |Execution|   |Execution|<........:
                 |  Node   |   |  Node   |
                 +---------+   +---------+
                     ^              ^
                     :..............:
```

Figure 3: Roles and relationship between components

4. Service Mobility Requirements

This section describes the functional requirements to realize service mobility in a virtualized environment. In the virtualized environment there are multiple Execution Nodes with multiple Functions on each Node, service sessions are set up by utilizing these Functions. A Manager Node is used to enable service mobility for the session in this virtualized environment where Execution Nodes and Functions can be dynamically added, deleted, re-allocated and migrated. The detailed specifications of how the virtualized network is provided and how the migration is executed are outside the scope of this document.

- The Execution Node MUST be able to report the available resources (e.g., CPU or memory usage) of its own to the Manager Node. The Execution Node SHOULD be able to obtain statistics on how much resources are consumed by each Function and to report them to the Manager Node. The Execution Node SHOULD be able to spontaneously report to the Manager Node according to the running condition of its own. The Manager Node MAY use such information for service mobility.

- Execution Node may migrate to another hardware unit during the operation. The IP address and/or data-link address to reach that Execution Node may change due to this migration. The Manager Node MUST be able to identify and keep track of the Execution Node wherever it moves. This requires a unique identifier for each Execution Node that is independent of the physical address for service mobility.

- A service may involve multiple Functions, which establish a session to interwork together for initiating and maintaining the service. Service mobility mandates the consistency of a session even if some of those Functions migrate to a different Execution Node, which could further migrate to a different hardware unit. This requires that each session and Function MUST be uniquely identified and manipulated by the Manager Node regardless of the hardware unit(s) that is/are providing resources.

- Security and reliability in communications between the Manager Node and Execution Node MUST be provided. The Manager Node MUST be able to control the admission of information exchange between Execution Nodes. In order to support these properties, existing architectures and mechanisms (e.g., AAA, IPsec, reliable transport protocols) SHOULD be taken into consideration for early adoption and deployment.

5. Protocol Operations

Figure 4 shows the signaling flow between the manager and execution nodes. When the Execution Node boots up, it registers itself with the manger node by sending the Registration message. If the IP address of the Manager Node is not known, the Information Server SHOULD be used for its discovery. The Manager Node SHOULD periodically check the status of each registered Execution Node by sending the Keep-Alive message [Editor's Note: the default interval time should be defined]. According to the situation on the execution nodes, the Manager Node sends various types of operation commands to the execution node. If the status of the Execution Node changes, it sends the Status-Update message to the manager node. If the Execution Node goes out of the scope of the management by the Manager Node or is going to be shut down, it sends the De-registration message to the manager node. The Error message is asynchronously sent by either node to notify the other peer when an erroneous event happens (e.g., when the Manager Node becomes unable to perform the management tasks). For each message, the recipient MUST send back the ACK message. These signaling messages are conveyed over a reliable transport mechanism. The Manager Node and Execution Node MUST support TCP and SHOULD support SCTP.

```
                        Execution                 Manager
                          Node                      Node
                            |                         |
                            |      Registration       |
                            |------------------------>|
                            |          Ack            |
                            |<------------------------|
                            |                         |
                            |       Keep-Alive        |
                            |<------------------------|
                            |          Ack            |
                            |------------------------>|
                            |                         |
                            |     <*> Operation       |
                            |<------------------------|
                            |          Ack            |
                            |------------------------>|
                            |                         |
                            |      Status-Update      |
                            |------------------------>|
                            |          Ack            |
                            |<------------------------|
                            |                         |
                            |      De-registration    |
                            |------------------------>|
                            |          Ack            |
                            |<------------------------|
                            |                         |
                            |         Error           |
                            |<------------------------>|
                            |          Ack            |
                            |<------------------------>|
                            |                         |
```

Figure 4: Signaling flow between the manager and execution nodes

In this document, the following control messages are defined:

Registration:
Used for the Execution Node to register itself with the manager node, by which the Execution Node goes under the control of the manager node.

Keep-Alive:
Used for the Manager Node to check the status of the managed execution node.

Operation:
Used for the Manager Node to indicate the Execution Node to do a specific action to the function or sessions on that execution node.

Status-Update:
Used for the Execution Node to notify the Manager Node on the updated status.

De-registration:
Used for the Execution Node to release the registration from the manager node. The Manager Node can also send this message to the Execution Node to force de-registration.

Ack:
Used to respond to the above messages for acknowledgment and returning requested information and/or status code (success or failure).

Error:
Asynchronous messaging mechanism that can be initiated by either the Manager Node or Execution Node to notify the peer of an error condition.

Further, the following operations are defined:

GET operation:
Used for the Manager Node to obtain specific information from the target execution node.

ADD operation:
Used to instruct the target Execution Node to run a new function.

DELETE operation:
Used to instruct the target Execution Node to terminate a running function.

MOVE operation:
Combination of ADD and DELETE operation, but the status information on the related function is also passed to a new execution node.

COPY operation:
Similar to ADD operation, but the status information on the related function is also passed to a new execution node.

MOVE_SESSION operation:

Used to instruct the target Execution Node to move sessions to another execution node.

These control messages are conveyed over a reliable transport mechanism. The Manager Node and Execution Node MUST support TCP and SHOULD support SCTP.

6. Message Format

6.1. Common header format

Figure 5 shows the common header format for all messages.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Type     |   Sub-Type    |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      Code     |                Sequence Number                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                            Options                            :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                              ...                              :
|                                                               |
```

Figure 5: Common header format

Type

Control message type. In this document, the following operations are defined: Registration (1)/Keep-alive (2)/Operation (3)/Status-Update (4)/De-registration (5)/Ack (6).

Sub-Type

If the Type value is 3 (Operation) or 6 (Ack), the Sub-type represents the operation command. Otherwise, this value MUST be zero and ignored by the receiver.

Length

16-bit unsigned integer indicating the length of the whole message in octets, excluding the type and length fields.

Code

An 8-bit unsigned integer used for containing the status code in the Ack message. For the other message, this field MUST be filled with zero and ignored by the receiver.

Sequence Number

A 24-bit unsigned integer used by the receiving node to sequence the operation command and by the sending node to match a returned Acknowledgement with this operation command.

Options

Variable-length field that contains a command message and/or one or more option(s).

Figure 6 shows the TLV-encoded option format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OP-Type   |  Sub-OP-Type  |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
:                             Value                             :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6: Option format

OP-Type

8-bit unsigned integer indicating the type of the option value.

Sub-OP-Type

8-bit unsigned integer indicating the sub-type of the option value. MUST be set to zero if not defined.

Length

16-bit unsigned integer indicating the length of the option in octets, excluding the OP-Type, Sub-OP-Type and Length fields.

Value

value for the specified option type is contained.

6.2. Control messages

6.2.1. REGISTRATION

Type: 1

Mandatory options:

- o Node ID

Additional options:

- o Node Information

6.2.2. KEEP-ALIVE

Type: 2

Mandatory options:

- o Node ID

Additional options:

- o Node Information
- o Function ID
- o Function Information

6.2.3. OPERATION

Type: 3

Sub-Type: Requested operation (See Section 6.3)

Mandatory options:

- o Node ID

6.2.4. STATUS-UPDATE

Type: 4

Mandatory options:

- o Node ID

Additional options:

- o Node Information
- o Function ID
- o Function Information

6.2.5. DE-REGISTRATION

Type: 5

Mandatory options:

- o Node ID

6.2.6. ACK

Type: 6

Sequence number: MUST be copied from the corresponding control message.

Sub-Type: MUST be copied from the corresponding control message.

Code: Result code for the requested operation. In this document, the following code values are defined:

0: Successful

128: Failed

129: Insufficient resources

130: Administratively prohibited

131: Status unknown

Mandatory options:

- o Node ID

The Vendor-specific information option can be used to convey more detailed information, for example, to notify the sender about the reason for the failure.

6.3. Operation commands

6.3.1. GET Operation

Type: 3

Sub-Type: 1

Mandatory options:

- o Node ID
- o Function ID
- o List of options (with zero values in the value field)

Expected options in ACK:

- o Node ID
- o Function ID
- o List of options (with non-zero values in the value field)
- o Result Code
- o Running Status

6.3.2. ADD Operation

Type: 3

Sub-Type: 2

Mandatory options:

- o Node ID
- o Function ID

Some function takes time to boot up, thus when it successfully booted up, the Execution Node sends Status-update message to the manager node.

Expected options in ACK:

- o Node ID
- o Function ID
- o Result Code
- o Running Status

6.3.3. DELETE Operation

Type: 3

Sub-Type: 3

Mandatory options:

- o Node ID
- o Function ID

Expected options in ACK:

- o Node ID
- o Function ID
- o Result Code
- o Running Status

Some function takes time to terminate, thus when termination is completed, the Execution Node sends Status update message to the manager node.

6.3.4. MOVE Operation

Type: 3

Sub-Type: 4

Mandatory options:

- o Source Node ID
- o Destination Node ID

- o Function ID

Expected options in ACK:

- o Source Node ID
- o Destination Node ID
- o Function ID
- o Result Code
- o Running Status

The source Node ID and destination Node ID MUST be listed in this order.

6.3.5. COPY Operation

Type: 3

Sub-Type: 5

Mandatory options:

- o Source Node ID
- o Destination Node ID
- o Function ID

Expected options in ACK:

- o Source Node ID
- o Destination Node ID
- o Function ID
- o Result Code
- o Running Status

The source Node ID and destination Node ID MUST be listed in this order.

6.3.6. MOVE_SESSION Operation

Type: 3

Sub-Type: 6

Mandatory options:

- o Source Node ID
- o Destination Node ID
- o Session information

Additional options:

- o Function ID

Expected options in ACK:

- o Function ID, if specified in the corresponding operation command.
- o Result Code
- o Running Status

The source Node ID and destination Node ID MUST be listed in this order. When Function ID is specified, all the sessions related to that Function ID are the target for movement.

6.4. Option values

In this document, the following attributes are defined. Sub-OP-type MUST be set zero if not defined.

6.4.1. IPv4 address

OP-Type: 1

Length: 4

Value: Unsigned integer. IPv4 address (e.g., for the target execution node)

6.4.2. IPv6 address

OP-Type: 2

Length: 16

value: Unsigned integer. IPv6 address (e.g., for the target execution node)

6.4.3. Port number

OP-Type: 3

Length: 2

value: Unsigned integer. Port number (e.g., for the target session)

6.4.4. Node ID

OP-Type: 4

Length: 4

Value: Unsigned integer. Node ID for the target Execution Node

6.4.5. Function ID

OP-Type: 5

Length: 4

Value: Unsigned integer. Function ID for the target function

6.4.6. Node information

OP-Type: 6

Length: variable

Value: Octet string. Node information specified by the Sub-OP-Type. In this document, the following types are defined:

Sub-OP-Type
- 1: Processing capacity (GHz)
- 2: Current load (number of waiting processes)
- 3: Average load (number of waiting processes)
- 10: Total memory size (byte)
- 11: Available memory size (byte)

20: Total storage size (byte)
21: Available storage size (byte)
30: Total bandwidth (bps)
31: Current used bandwidth (bps)
32: Average used bandwidth (bps)
40: Boot time (NTP timestamp format)

6.4.7. Function information

OP-Type: 7

Length: variable

Value: Octet string. Used to describe the type or status of the function. In this document, the following values are defined:

Sub-OP-Type
1: Function name (e.g., "P-CSCF" or "HSS")
2: Current load (number of waiting processes)
3: Average load (number of waiting processes)
10: Required memory size (byte)
11: Available memory size (byte)
20: Required storage size (byte)
21: Available storage size (byte)
31: Current used bandwidth (bps)
32: Average used bandwidth (bps)
40: Boot time (NTP timestamp format)
50: Running status ("Starting", "Running", "Terminating" or "Unknown")

6.4.8. Session information

OP-Type: 8

Length: variable

Value: Octet string to represent one or group of session(s). This value MUST be understood by both the manager and execution nodes. In this document, the following values are defined:

Sub-OP-Type
1: SIP URI (e.g., sip:alice@example.com)
2: Contact address (IPv4 or v6 address)
3: the ratio to the whole sessions (e.g., 0.3)
4: the number of sessions (e.g., 1000)

6.4.9. Vendor-specific information

OP-Type: 9

Length: variable

Value: Octet string to convey vendor-specific information. This value MUST be understood by both the Manager and Execution Nodes. In this document, the following value is defined:

Sub-OP-Type
1: Information message (e.g., "Protocol error")

7. IANA Considerations

This specification requests one TCP and SCTP port number for exchanging the defined control messages.

8. Security Considerations

It is assumed that necessary level of security between the Manager Node and Execution Nodes is provided by other means and a secure connection between them is not mandated in this document.

9. Acknowledgments

The authors would like to thank Christian Makaya for his valuable comments to and reviews of this document.

10. References

10.1. Normative references

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10.2. Informative references

[IMSAA09] Dutta, A., Makaya, C., Das, S., Chee, D., Lin, F., Komorita, S., Chiba, T., Yokota, H., and H. Schulzrinne, "Self Organizing IP Multimedia Subsystem", IEEE IMSAA, December 2009.

Authors' Addresses

Hidetoshi Yokota
KDDI Lab
2-1-15 Ohara, Fujimino
Saitama, 356-8502
JP

Email: yokota@kddilabs.jp

Fuchun J. Lin
Telcordia Technologies
One Telcordia Drive
Piscataway, NJ 08854-4157
US

Email: fjlin@research.telcordia.com

Ashutosh Dutta
NIKSUN
100 Nassau Park Boulevard, Princeton, NJ
08540
US

Email: ashutosh.dutta@ieee.org

Carl Williams
MCSR Labs
Palo Alto, CA
94306
US

Email: carlw@mcsr-labs.org

Vishwas Manral
IPInfusion Inc.
1188 E. Arques Ave.
Sunnyvale, CA 94085
US

Phone: 408-400-1900
Email: vishwas@ipinfusion.com