DHC Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 21, 2014

Q. Sun
Y. Cui
Tsinghua University
M. Siodelski
ISC
S. Krishnan
Ericsson
I. Farrer
Deutsche Telekom AG

October 18, 2013

DHCPv4 over DHCPv6 Transport
draft-ietf-dhc-dhcpv4-over-dhcpv6-02

Abstract

IPv4 connectivity is still needed as networks migrate towards IPv6. Users require IPv4 configuration even if the uplink to their service provider supports IPv6 only. This document describes a mechanism for obtaining IPv4 configuration information dynamically in IPv6 networks by carrying DHCPv4 messages over DHCPv6 transport. Two new DHCPv6 messages as well as a new DHCPv6 option are defined for the purpose of conveying DHCPv4 messages through IPv6 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 21, 2014.

Table of Contents

1. Introduction

As the migration towards IPv6 continues, IPv6-only networks will become more prevalent. At the same time, IPv4 connectivity will continue to be provided as a service over IPv6-only networks. In addition to providing IPv4 addresses for clients of this service, other IPv4 configuration parameters may also need to be provided

(e.g. addresses of IPv4-only services).

By conveying DHCPv4 messages over DHCPv6 transport, this document describes a mechanism for the dynamic provisioning of IPv4 addresses and other configuration parameters. The mechanism leverages existing

infrastructure for DHCPv4, e.g. failover, DNS updates, leasequery, etc. This mechanism is suitable for stateful allocation and management of IPv4 addresses (dynamic leasing) and other IPv4 configuration parameters across IPv6-only networks.

2. Requirements Language

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].

3. Terminology

This document makes use of the following terms:

DHCPv4-over-DHCPv6: A protocol described in this document, which is used to carry DHCPv4 messages encapsulated in DHCPv6 messages.

DHCP client: The 'DHCP client' in this document consists of both DHCPv4 and DHCPv6 client engines. The client is able to request IPv6 configuration information through DHCPv6, as well as to request IPv4 configuration information using DHCPv4-over-DHCPv6 transport.

4o6 Server: A DHCP server capable of processing DHCPv4 packets wrapped in the DHCPv6 option: BOOTP Message Option (defined below).

4. Architecture Overview

The architecture described in this document addresses a typical use case, where a DHCP client's uplink supports IPv6 only and the Service Provider's network supports IPv6 and limited IPv4 services. In this scenario, the client can only use the IPv6 network to access IPv4

services and so it must configure IPv4 services using IPv6 as the underlying transport protocol.

Although the purpose of this document is to address the problem of communication between DHCPv4 client and DHCPv4 server, the mechanism that it describes does not restrict the transported messages types only to DHCPv4. BOOTP messages can be transported using the same mechanism.

DHCP clients can be running on CPE devices, end hosts or any other device which supports the DHCP client function. At the time of writing, DHCP clients on CPE devices are easier to modify compared to

those implemented on end hosts. As a result, this document uses the CPE as an example for describing the mechanism. This does not preclude any end-host, or other device requiring IPv4 configuration, from implementing the mechanism in the future.

This mechanism works by carrying DHCPv4 messages encapsulated within DHCPv6 messages. Figure 1, below, illustrates one possible deployment architecture.

The DHCP client implements a new DHCPv6 message called Boot-request-v6, which contains a new option called BOOTP Message Option. The format of this option is described in Section 6.1.

The DHCPv6 packet can be transmitted either via Relay Agents or directly to the 4o6 Server. The server replies with a DHCPv6 response, which is a new DHCPv6 message called Boot-reply-v6. This message carries DHCPv4 response wrapped with the BOOTP Message Option.

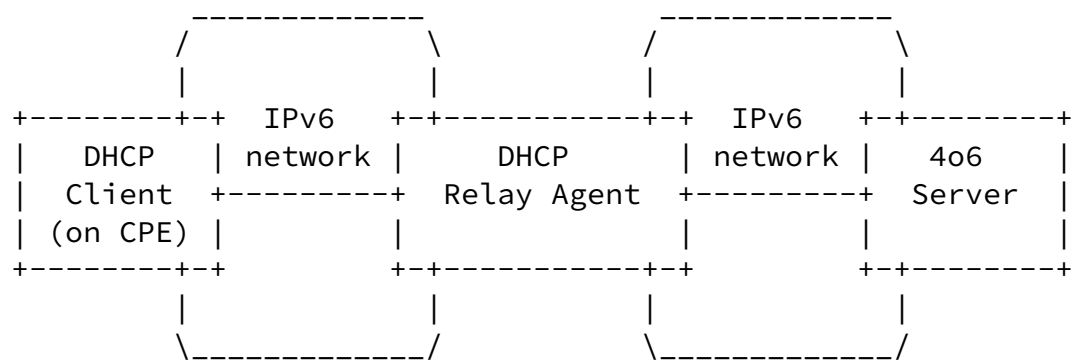


Figure 1: Architecture Overview

By default, the DHCPv4-over-DHCPv6 is disabled on the client. Before a client can use this protocol it MUST obtain the necessary IPv6 configuration. If the client is configured to use DHCPv6 to obtain its IPv6 configuration, the DHCPv6 server MAY include the DHCPv4 -over-DHCPv6 Enable Option in its Reply message to indicate that client SHOULD use the DHCPv4-over-DHCPv6 protocol to obtain additional configuration. The format of the DHCPv4-over-DHCPv6 Enable Option is described in Section 6.2.

Typically, a client communicates with the 4o6 Servers using well known All_DHCP_Relay_Agents_and_Servers multicast address. If a DHCPv6 server is configured to do so, it MAY send unicast addresses of the 4o6 Servers to the client during the client's configuration using DHCPv6. The unicast addresses are carried in the 4o6 Server Addresses Option encapsulated in the Reply message. The 4o6 Server

Addresses Option's format is defined in Section 6.3.

5. New DHCPv6 Messages

There are two new DHCPv6 messages defined in this document which carry DHCPv4 messages between a client and a server using DHCPv6 protocol: Boot-request-v6 and Boot-reply-v6. This section describes structures of these messages.

5.1. Message Types

The following new message types are defined in this document:

BOOTREQUESTV6 (TBD): Identifies a Boot-request-v6 message. A client sends this message to a server. The BOOTP Message Option carried by this message contains a BOOTREQUEST message that the client uses to request IPv4 configuration parameters from the server.

BOOTREPLYV6 (TBD): Identifies a Boot-reply-v6 message. A server sends this message to a client. It contains a

BOOTP Message Option carrying a BOOTREPLY message in response to a BOOTREQUEST received by the server in the BOOTP Message Option of the Boot-request-v6 message.

5.2. Message Formats

Both DHCPv6 messages defined in this document share the following format:

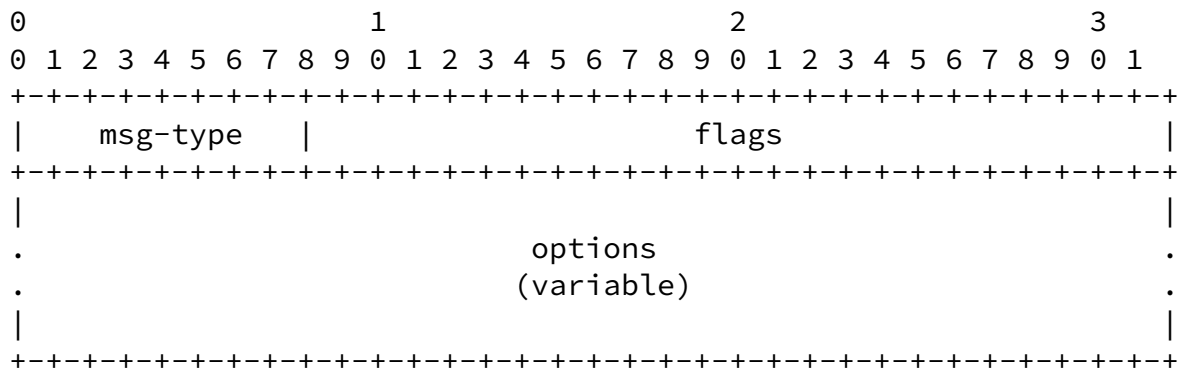


Figure 2: Architecture Overview

msg-type Identifies message type. It can be either BOOTREQUESTV6 (TBD) or BOOTREPLYV6 (TBD) which corresponds to the Boot-request-v6 or Boot-reply-v6

respectively.

flags Specifies flags which provide additional information required by the server to process a DHCPv4 message wrapped in Boot-request-v6 Message, or required by the client to process DHCPv4 message wrapped in Boot-reply-v6 Message.

options Options carried by the message and described in Section 6.

5.3. Boot-request-v6 Message Flags

The "flags" field of the Boot-request-v6 is used to carry additional information which may be used by the server to process the

encapsulated DHCPv4 message. Currently only one bit of this field is used. Remaining bits are reserved for the future use. Currently the "flags" field has the following format:

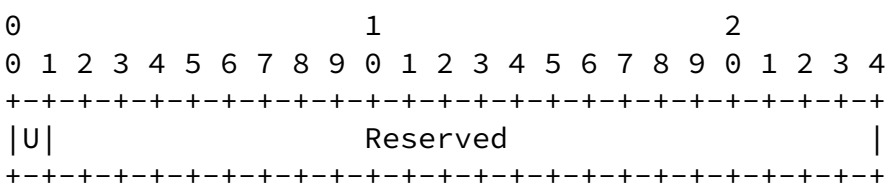


Figure 3: Boot-request-v6 flags format

U Unicast Flag. If it is set to 1, it indicates that the DHCPv4 message encapsulated with the Boot-request-v6 message would be sent to a unicast address if it was sent using IPv4. If this flag is set to 0 it indicates that the DHCPv4 message would be sent to broadcast address if it was sent using IPv4.

Reserved Bits reserved for future use. A client which doesn't implement future extensions using these bits MUST set them to 0.

5.4. Boot-reply-v6 Message Flags

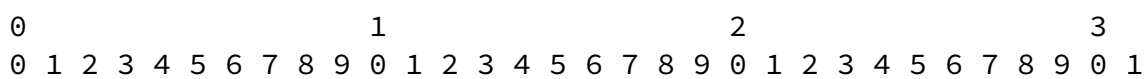
This document introduces no flags to be carried in the "flags" field of the Boot-reply-v6 message. They are all reserved for the future use. Server MUST set all bits of this field to 0.

6. DHCPv6 Options

6.1. BOOTP Message Option Format

The BOOTP Message option carries a BOOTP message that is sent by the client or the server. Such BOOTP messages exclude any IP or UDP headers.

The format of the BOOTP Message Option is:



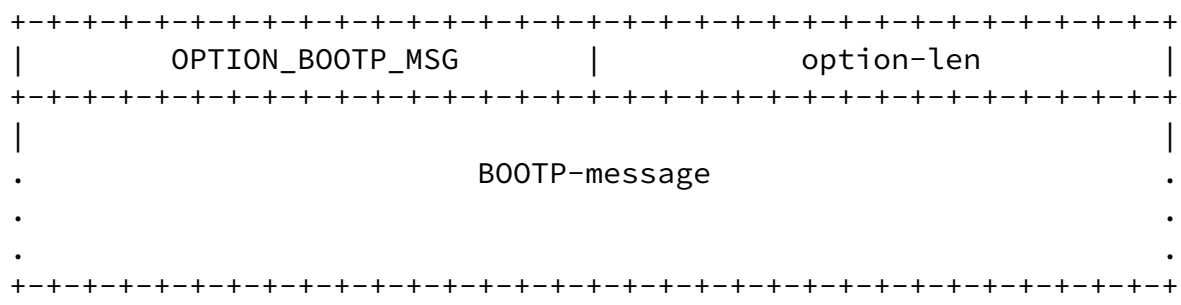


Figure 4: BOOTP Message Option Format

option-code OPTION_BOOTP_MSG (TBD)

option-len Length of BOOTP message

BOOTP-message The BOOTP message sent by the client or the server. In a Boot-request-v6 message it contains a BOOTREQUEST message sent by a client. In a Boot-reply-v6 message it contains a BOOTREPLY message sent by a server in response to a client.

6.2. DHCPv4-over-DHCPv6 Enable Option Format

The DHCPv4-over-DHCPv6 Enable Option indicates that the client SHOULD enable the DHCPv4-over-DHCPv6 function.

The format of the DHCPv4-over-DHCPv6 Enable Option is:

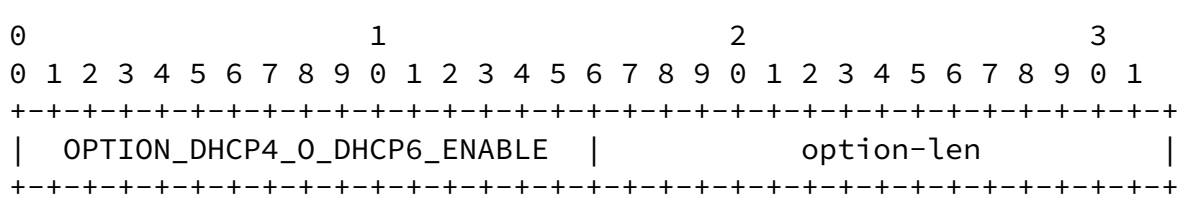


Figure 5: DHCPv4-over-DHCPv6 Enable Option Format

option-code OPTION_DHCP4_O_DHCP6_ENABLE (TBD)

option-len 0

6.3. 4o6 Servers Address Option Format

The 4o6 Servers Address Option carries unicast IPv6 addresses of the 4o6 Servers.

The format of the 4o6 Servers Address Option is:

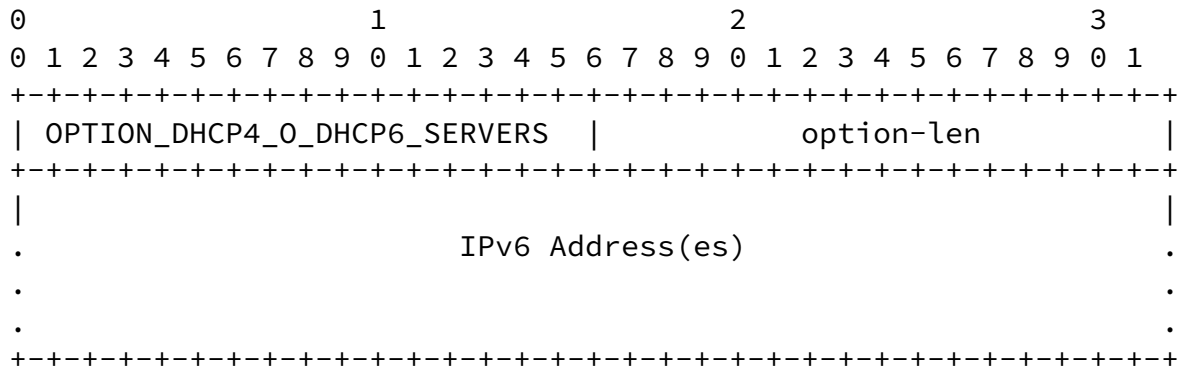


Figure 6: 4o6 Servers Address Option Format

option-code OPTION_DHCP4_O_DHCP6_SERVERS (TBD)

option-len Length of the IPv6 address(es), i.e. integer times of 16.

IPv6 Address The IPv6 address(es) of the 4o6 Server(s).

7. Use of the Boot-request-v6 Unicast Flag

A DHCPv4 client conforming to the [RFC2131] may send its DHCPREQUEST message to either broadcast or unicast address depending on its state. For example, the client in the RENEWING state will use a unicast address to contact a server and renew its lease. The client in the REBINDING state MUST use a broadcast address. If there is a relay agent in the middle, a client in the RENEWING state may send a DHCPREQUEST message to the unicast address of the relay agent. In such case the server can't find out whether client sent a message to a unicast or broadcast address and thus it can't determine the client's state. [RFC5010] introduced the "Flags Suboption" which relay agents add to relayed messages to indicate whether broadcast or unicast was used by the client.

The DHCPv4-over-DHCPv6 protocol uses IPv6 to deliver DHCPv4 messages to the server. There is no relation between the outer IPv6 address and the inner DHCPv4 message. So the server is not able to know

whether the DHCPv4 messages should have been sent using broadcast or unicast in IPv4 by checking the IPv6 address. This is similar to the case [RFC5010] handled.

In order to allow the server to determine the client's state, the "Unicast" flag is carried in the Boot-request-v6 message. Client MUST set this flag to 1 when the DHCPv4 message would have been sent to the unicast address if using DHCPv4 over IPv4. This flag MUST be set to 0 if the DHCPv4 client would have sent the message to the broadcast address in IPv4. The choice whether a given message should be sent to a broadcast or unicast address MUST be made based on the [RFC2131] and its extensions.

8. Client Behavior

The DHCP client by default doesn't use DHCPv4-over-DHCPv6 protocol to obtain its DHCPv4 configuration. Client MUST obtain its IPv6 configuration before it MAY use DHCPv4-over-DHCPv6 to obtain DHCPv4 configuration. If IPv6 configuration is obtained using DHCPv6 as described in [RFC3315], client SHOULD request the DHCPv4-over-DHCPv6 Enable Option and the 4o6 Server Addresses Option in the Option Request Option (ORO) to check if it SHOULD use DHCPv4-over-DHCPv6.

The DHCPv6 server MAY include these options in the Reply message sent to the client. The client determines how to launch the DHCPv4-over-DHCPv6 function based on the presence / absence of these two options:

- If the client doesn't receive the DHCPv4-over-DHCPv6 Enable Option, it SHOULD NOT enable the DHCPv4 over DHCPv6 function.

- If the client receives the DHCPv4-over-DHCPv6 Enable Option but no 4o6 Servers Address Option, it SHOULD enable the DHCPv4-over-DHCPv6 function, but use IPv6 All_DHCP_Relay_Agents_and_Servers multicast address to communicate with the servers or relays as described above.

- If the client receives both options, it SHOULD enable the DHCPv4-over-DHCPv6 function, and send requests to all unicast addresses conveyed by the 4o6 Server Addresses Option.

If the client is instructed by the DHCPv6 server to use DHCPv4-over-DHCPv6 function it SHOULD generate a DHCPv4 message to obtain configuration from the 4o6 Server. This message is stored verbatim in the BOOTP Message Option carried by the Boot-request-v6 message. The client MUST put exactly one BOOTP Message Option into a single Boot-request-v6 message.

A client MUST set the Unicast flag as specified in Section 7.

If the client has not received a 4o6 Server Addresses Option from the DHCPv6 server, it transmits the Boot-request-v6 message as specified in Section 13 of [RFC3315]. If the client received this option, it MUST send Boot-request-v6 message to all unicast addresses listed in the received option.

When a client receives a Boot-reply-v6 message, it MUST look for the BOOTP Message Option within this message. If this option is not found, the Boot-reply-v6 message is discarded. If the BOOTP Message Option is found, the client extracts the DHCPv4 message it contains and processes it as described in section 4.4 of [RFC2131].

DHCP clients are responsible for the retransmission of messages. When requesting IPv4 configuration, the client SHOULD follow the normal DHCPv4 retransmission requirements and strategy as specified in section 4.1 of [RFC2131]. As a result there are no explicit transmission parameters associated with a Boot-request-v6 message.

As the DHCPv4 and DHCPv6 clients are running on the same host, the client MUST implement [RFC4361] to ensure that the device correctly identifies itself.

9. Relay Agent Behavior

When a DHCPv6 relay agent receives a Boot-request-v6 message, it MUST handle the message as described in section 4 of [I-D.ietf-dhc-dhcpv6-unknown-msg].

A DHCPv6 relay agent MUST implement the Relay behaviour described in section 20.1.1 of [RFC3315].

Additionally, the DHCPv6 relay agent MAY allow the configuration of dedicated DHCPv4-over-DHCPv6 specific destination addresses, differing from the addresses of the DHCPv6 only server(s). To implement this function, the relay checks the received DHCPv6 message type and forwards according to the following logic:

1. If the message type is Boot-request-v6, then the DHCPv6 request is relayed to the configured DHCPv4 aware 4o6 Server's

address(es).

2. For any other DHCPv6 message type, forward according to section 20 of [RFC3315].

The above logic only allows for separate relay destinations configured on the relay agent closest to the client (single relay hop). Multiple relaying hops are not considered in the case of separate relay destinations.

10. 4o6 Server Behavior

When the server receives a Boot-request-v6 message from a client, it searches for a BOOTP Message Option. If this option is missing, the server discards the packet. The server MAY notify an administrator about the receipt of a malformed packet. The mechanism for this notification is out of scope for this document

If the server finds a valid BOOTP Message Option, it extracts the original DHCPv4 message sent by the client. This message is passed to the DHCPv4 server engine, which generates a response to the client as specified in [RFC2131]. This engine can be implemented as a built-in DHCPv4 server function of the 4o6 Server, or it can be a separate DHCPv4 server instance. Discussion regarding communication between the 4o6 Server and a DHCPv4 server engine is out of scope for this document.

When appropriate DHCPv4 response is generated, 4o6 Server places it in the payload of a BOOTP Message Option, which it puts into the Boot-reply-v6 message.

If the Boot-request-v6 message was received directly by the server, the Boot-reply-v6 message MUST be unicast from the interface on which the original message was received.

If the Boot-request-v6 message was received in a Relay-forward message, the server creates a Relay-reply message with the Boot-reply-v6 message in the payload of a Relay Message Option, and responds as described in section 20.3 of [RFC3315].

11. Security Considerations

In this specification, DHCPv4 messages are encapsulated in the newly defined option and messages. This is similar to the handling of the current relay agent messages. In order to bypass firewalls or network authentication gateways, a malicious attacker may leverage this feature to convey other messages using DHCPv6, i.e. use DHCPv6 as a form of encapsulation. However, the potential risk from this is not seen to be greater than that with current DHCPv4 and DHCPv6 practice.

12. IANA Considerations

IANA is requested to allocate three DHCPv6 option codes for use by OPTION_BOOTP_MSG, OPTION_DHCP4_O_DHCP6_ENABLE and OPTION_DHCP4_O_DHCP6_SERVERS, and two DHCPv6 message type codes for the BOOTREQUESTV6 and BOOTREPLYV6.

13. Contributors List

Many thanks to Ted Lemon, Bernie Volz, Tomek Mrugalski, Yuchi Chen and Cong Liu, for their great contributions to the draft.

14. References

14.1. Normative References

[I-D.ietf-dhc-dhcpv6-unknown-msg]
Cui, Y., Sun, Q., and T. Lemon, "Handling Unknown DHCPv6 Messages", draft-ietf-dhc-dhcpv6-unknown-msg-02 (work in progress), September 2013.

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

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC4361] Lemon, T. and B. Sommerfeld, "Node-specific Client

Identifiers for Dynamic Host Configuration Protocol Version Four (DHCPv4)", RFC 4361, February 2006.

14.2. Informative References

[I-D.ietf-dhc-dhcpv4-over-ipv6]
Cui, Y., Wu, P., Wu, J., and T. Lemon, "DHCPv4 over IPv6 Transport", draft-ietf-dhc-dhcpv4-over-ipv6-07 (work in progress), September 2013.

[RFC5010] Kinnear, K., Normoyle, M., and M. Stapp, "The Dynamic Host Configuration Protocol Version 4 (DHCPv4) Relay Agent Flags Suboption", RFC 5010, September 2007.

Authors' Addresses

Qi Sun
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6278-5822
Email: sunqi@csnet1.cs.tsinghua.edu.cn

Yong Cui
Tsinghua University
Department of Computer Science, Tsinghua University
Beijing 100084
P.R.China

Phone: +86-10-6260-3059
Email: yong@csnet1.cs.tsinghua.edu.cn

Marcin Siodelski
950 Charter Street
Redwood City, CA 94063
USA

Phone: +1 650 423 1431
Email: msiodelski@gmail.com

Suresh Krishnan
Ericsson

Email: suresh.krishnan@ericsson.com

Ian Farrer
Deutsche Telekom AG
GTN-FM4,Landgrabenweg 151
Bonn, NRW 53227
Germany

Email: ian.farrer@telekom.de