Internet Engineering Task Force R. Despres, Ed.
Internet-Draft RD-IPtech
Intended status: Standards Track S. Matsushima
Expires: September 8, 2011 SoftBank
T. Murakami
IP Infusion
O. Troan
Cisco
March 7, 2011

IPv4 Residual Deployment across IPv6-Service networks (4rd) ISP-NAT's made optional draft-despres-intarea-4rd-00

Abstract

This document specifies an automatic tunneling mechanism for providing IPv4 connectivity service to end users over a service provider's IPv6 network infrastructure. During the long transition period from IPv4-only to IPv6-only, a service provider's network infrastructure will have to deploy IPv6. But it will also have to maintain some IPv4 connectivity for a number of customers, for both outgoing and incoming connections, and for both customer-individual and shared IPv4 addresses. The 4rd solution (IPv4 Residual Deployment) is designed as a lightweight solution for this.

In some scenarios, 4rd can dispense ISPs from supporting any NAT in their infrastructures. In some others it can be used in parallel with NAT-based solutions such as DS-lite and/or NAT64/DNS4.

Copyright Notice

Table of Contents

1. Introduction

During the long transition period from only IPv4-only to IPv6-only, Internet-service providers (ISP's), will sooner or later deploy networks where routing is IPv6-only. Some of them will do so while they still have to offer residual IPv4 connectivity for the service to remain dual-stack. While this connectivity may be offered to privileged customers with in exclusive global addresses, more and more other customers will only have shared IPv4 addresses. In this document, "ISP" is used as a generic term. It includes DSL or Broadband service providers, mobile operators, and private operators of networks of any sizes.

4rd (IPv4 Residual Deployment) is a generic lightweight solution for the residual support of global IPv4 connectivity across IPv6-only routing networks. As such, it is the reverse of 6rd (IPv6 Rapid Deployment) whose purpose is to rapidly introduce native IPv6 connectivity across IPv4-only routing infrastructures. It applies the same principles of automatic tunneling, an stateless address mappings between IPv4 and IPv6.

On the tradeoff scale between efficiency of address sharing ratios and simplicity, 4rd is on the side of design and operational simplicity.

Depending on ISP constraints and policies, 4rd can be used either alone, with no NAT needed in ISP infrastructures, or in parallel with NAT based solutions such as DS-lite [I-D.ietf-softwire-dual-stack-lite] and NAT64/DNS64 [I-D.ietf-behave-v6v4-xlate-stateful] [I-D.ietf-behave-dns64].

At the time of writing, four ISP's in Japan have expressed interest for deploying 4rd (www.ietf.org/mail-archive/web/v6ops/current/msg05247).

This draft is still in an early stage. So far, it specifies details only for domains that have one mapping rule. In order to permit more flexible services, a next version is being worked on to deal with multiple mapping rules.

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

4rd domain (Domain): an IPv6 routing network operated by an ISP and comprising one or several 4rd relays operating with the same set of parameters. It offers to its 4rd-capable customers global IPv4 connectivity, both outgoing and incoming, and with exclusive or shared IPv4 addresses.

4rd Border Relay (BR): A 4rd-enabled router managed by the service provider at the edge of 4rd domain. A BR has an IPv6-enabled interface connected to the ISP routing network, and an IPv4 virtual interface acting as an endpoint for the automatic 4rd tunnel. This tunnel (IPv4 in IPv6) is between the BR and all CE's of the Domain.

4rd Customer Edge (CE): A node at the border between a customer infrastructure and the 4rd domain. This node has an IPv6 interface connected to the ISP routing network, and a virtual IPv4 interface acting as the end-point of the automatic 4rd tunnel. This tunnel (IPv4 in IPv6) is between the CE and all other CE's and all BR's of the Domain. It may be a host, a router, or both.

CE IPv6 prefix: The IPv6 prefix assigned to a CE independently from 4rd.

CE IPv6 address: In the context of 6rd, the IPv6 address used to reach a CE from other CE's and from BR's. A CE typically has another IPv6 address, assigned to it at its IPv6 interface without relationship with 6rd.

CE 4rd prefix: The 4rd prefix of the CE. It is derived from the CE IPv6 prefix by a mapping rule according to Section 4. Depending on its length, it is an IPv4 prefix, an IPv4 address, or a shared IPv4 address followed by a Port-set ID (Section 4.2).

Port-set ID: In a CE 4rd prefix longer than 32 bits, bits that follow the first 32. It identifies a set of ports exclusively assigned to the CE. As specified in Section 4.3, the set contains up to 4 port ranges, each range being defined by its port prefix.

Domain IPv6 prefix: An IPv6 prefix assigned by an ISP to a 4rd domain.

Domain 4rd prefix: A 4rd prefix assigned by an ISP to the 4rd domain. In typical operator applications, it is an IPv4 prefix. In a residential site in which an already shared IPv4 address has to be shared even more among several hosts, it may have more than 32 bits.

CE index: In a CE IPv6 prefix, all bits that follow the Domain IPv6 prefix. It is also, in a CE IPv4 prefix, all bits that follow the BR IPv4 prefix.

4. Mapping Rules

4.1. From an IPv6 Prefix to a 4rd Prefix

A 4rd mapping rule establishes a 1:1 mapping between CE IPv6 prefixes and CE 4rd prefixes.

```
    <---------------- CE IPv6 prefix (max 64) -------------->
    +--------------------------------+-----------------------+
    |      Domain IPv6 prefix        |        CE index       |
    +--------------------------------+-----------------------+
    <-- Domain IPv6 Prefix length -><--- CE index length --->:
                                    :                        :
                                    :         ||             :
                                    :         \/             :
                                    :                        :
                                    <--- CE index length --->:
               +-------------------+-----------------------+
                | Domain 4rd prefix |        CE index       |
                +-------------------+-----------------------+
                <----------- CE 4rd prefix (max 47) -------->
```

Figure 1: From an IPv6 Prefix to a 4rd Prefix

A CE derives its CE 4rd prefix from the IPv6 prefix it has been delegated on the IPv6-routing network, using for this parameters of the applicable mapping rule. If the domain has several mapping rules, that which applies is that whose Domain IPv6 prefix is at the beginning of the CE IPv6 prefix. As shown in Figure 1, the CE 4rd prefix is made of the Domain 4rd prefix followed by the CE index, where the CE index is the remainder of the CE IPv6 prefix after the

Domain IPv6 prefix (the length of the Domain IPv6 prefix is defined by the mapping rule).

4.2. From a 4rd Prefix longer than 32 bits to a Port-set ID

Depending on its length, a CE 4rd prefix is either an IPv4 prefix, a full IPv4 address, or a shared IPv4 address followed by a Port-set ID (Figure 2). If it includes a port set ID, this ID specifies which ports are assigned to the the CE for its exclusive use. This set, composed of up to 4 port ranges, is algorithmically derived from the Port-set ID (see Section 4.3).

```
                       <-- CE 4rd prefix length -->
                       +--------------------------+- - -+
    Shorter than 32 bits |        IPv4 prefix        | ... |
                       + -------------------------+- - -+


                       <----- CE 4rd prefix length ----->
                       +--------------------------------+
          32 bits      |           IPv4 address         |
                       +--------------------------------+
                       <--------------- 32 ------------->


                       <----------- CE 4rd prefix length ---------->
                       +-------------------------------+-----------+
        33 to 47 bits   |       IPv4 shared address      |Port-set ID|
                       +-------------------------------+-----------+
                       <--------------- 32 ----------><- max 15 -->
```

Figure 2: Variants of CE 4rd prefixes

4.3. From a Port-Set ID to a Port Set

Each value of a Port-set ID specifies which ports can be used by any protocol whose header format starts with source and destination ports (UDP, TCP, SCTP, etc.). Design constraint of the algorithm are the following:

"Fairness with respect to special-value ports"
No port-set must contain any port from 0 to 4095. (These ports, which have more value than others in OS's, are normally not used in dynamic port assignments to applications).

"Fairness with respect to the number of ports"
For a Port-set-ID's having the same length, all sets must have the same number of ports.

"Exhaustiveness"
For a any Port-set-ID length, the aggregate of port sets assigned for all values must include all ordinary-value ports (from 4,096 to 16,384).

If the Port-set ID has 1 to 12 bits, the set comprises 4 port ranges. As shown in Figure 3, each port range is defined by its port prefix, made of a range-specific "head" followed by the Port-set ID. Head values are in binary 1, 01, 001, and 0001. They are chosen to exclude ports 0-4095 and only them.

```
                <------- Port (16 bits) -------->
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Port-range a    |1|x x x x x x x x|             |  0xF780 - 0xF7FF
 (head = 1)     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                   \               \
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Port-range b    |0 1|x x x x x x x x|           |  0x7BC0 - 0x7BFF
 (head = 01)    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                     \               \
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Port-range c    |0 0 1|x x x x x x x x|         |  0x3DE0 - 0x3DFF
 (head = 001)   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                       \               \
                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
Port-range d    |0 0 0 1|x x x x x x x x|       |  0x1EF0 - 0x1EFF
 (head = 0001)  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                <- head-><--Port-set ID->                /\
                <-- Port-range prefix --><-tail->        ||
                                                         ||
                                             Example of Port-ranges
                                           if the Port-set ID is 0xEF
```

Figure 3: From Port-set ID to Port ranges

In the Port-set ID has 13 bits, only the 3 port ranges are assigned, having heads 1, 01, and 001. If it has 14 bits, only the 2 port ranges having heads 1 and 01 are assigned. If it has 15 bits, only the port range having head 1 is assigned. (In these three cases, the smallest port range has only one element).

Note: The port set assigned to a CE may be further subdivided by the CE among several functions such as the following: (1) an IPv4 NAPT

(possibly configurable to do port forwarding, and possibly doing dynamic port assignments to hosts with UPnP and/or NAT-PMP); (2) an API for applications in the CE that need dynamic port assignments; (3) a new 4rd BR which assigns to its CE's subsets of its own port set. How to chose among these functions and/or combine them is beyond the scope of this specification. Readers are referred to documents dealing with operational applicability in diverse environments, e.g. [draft-sun-intarea-4rd-applicability] prepared in parallel of this one.

4.4. From an IPv4 Address or IPv4 address + Port to a CE IPv6 address

```
                       Port-set ID
                            |
      <--- CE 4rd prefix ---|->
      +---------------+---+-|--+
      |IPv4 shared address| '  |
      +---------------+---+----+
                      <-------->
                    CE-index length
                      :        :
                      :   ||   :
                      :   ||   :
                      :   \/   : Domain IPv6 suffix
                      :        :  |
   +------------------+--------+--|-+-----------------------------------+
   |Domain IPv6 prefix|CE index|  ' |                  0                |
   +------------------+--------+----+-----------------------------------+
   <------------ max 64 ------------>
   <---------------------- CE IPv6 address (128) ---------------------->
```

Figure 4: From a CE 4rd Prefix to the CE IPv6 address

In order to find whether a CE IPv6 address can be derived from an IPv4 address, or an IPv6 address + a port, a mapping rule has to be found that matches the IPv4 information:

- o If a mapping rule has a length L of CE IPv4 prefixes which does not exceed 32 bits, there is a match if the IPv4 address starts with the Domain 4rd prefix. The CE 4rd prefix is then the first L bits of the IPv4 address.

- o If a mapping rule has a length L of CE IPv4 prefixes which exceeds 32 bits, the match can only be found with the IPv4 address and the port. For this, the port is examined to determine which port-range head it starts with: 1, 01,001, or 0001. The N bits that

follow this head are taken as Port-set ID, where N is the length of Port set ID of the mapping rule. The CE 4rd prefix is then made of the IPv4 address followed by the Port-set ID.

If a match has been found, the CE IPv6 prefix is then made of the Domain IPv6 prefix followed by bits of the CE 4rd prefix that follow the Domain 4rd prefix, followed by the Domain IPv6 prefix of the mapping rule if there is one, and followed by 0's up to 128 bits to satisfy [RFC4291]. Figure 4 illustrates this process in the case of a shared IPv4 address.

4.5. MTU Considerations

The maximum size of IPv4 packets that are forwarded across a 4rd domain must be the path MTU of the domain minus the 40 octets (the length of the encapsulation header). (Otherwise, due to the impossibility of intermediate IPv6 nodes to fragment packets, they might be discarded during domain traversal.) Absent more specific indication, this path MTU has to be set to 1280, the minimum size that all IPv6 networks must support.

- In domains where IPv4 addresses are not shared, IPv6 destinations are derived from IPv4 addresses alone. Thus, each IPv4 packet can be encapsulated and decapsulated independently of each other. 4rd processing is completely stateless.

- On the other hand, in domains where IPv4 addresses are shared, BR's and CE's can have to encapsulate IPv4 packets whose IPv6 destinations depend on destination ports. Precautions are needed, due to the fact that the destination port of a fragmented datagram is available only in its first fragment. A sufficient precaution consists in reassembling each datagram received in multiple packets, and to treat it as though it would have been received in single packet. This function is such that 4rd is in this case stateful at the IP layer. (This is common with DS-lite and NAT64/DNS64 which, in addition, are stateful at the transport layer.)

  In the encapsulating direction, this permits to address all pieces of a received datagram to the right IPv6 destination. In the decapsulating direction, this permits to ensure that all pieces of received datagram do come from the right CE.

5. 4rd Configuration

Both CE's and BR's have to know parameters of their 4rd domain. These include the BR IPv6 address plus, for each mapping rule, the following:

o Domain IPv6 prefix

o CE-index length
   or CE-IPv6-prefix length (derivable from one another)

o Domain IPv6 suffix (optional - default ::/0)

o Domain 4rd prefix

A CE can acquire 4rd parameters of its 4rd domain in various ways.

o The simplest one, which requires no protocol, can be used if the CE software is provided by the ISP and is downloadable. 4rd parameters can be included in update packages. (This case is similar to that which permitted the first 6rd deployment [RFC5569].)

o Another way is based on DHCPv6 [RFC3315]. As such, it is similar to that of 6rd in [RFC5969] where CE parameters are acquired in IPv4 DHCP. How to format these parameters in a DHCPv6 option, or for other ways to advertise them to CE's has still to be specified. The design may be influenced by the following facts: (1) if the length of the Domain IPv6 prefix is known, the length of the CE index or that of the CE IPv6 prefix can be derived from one another; (2) if a single mapping rule applies to all CE's, each CE can derive the Domain IPv6 prefix from its CE IPv6 prefix by just knowing the length of this Domain IPv6 prefix; (3) sateless DHCPv6 servers are simpler to manage than stateful.

o Other methods of parameter acquisition are for further study.

6. BR and CE behaviors

BR's and CE's MAY have the behaviors specified in the following sections. Different behaviors are permitted, but MUST be equivalent as far as exchanged packets are concerned.

6.1. Encapsulation and IPv6 Fragmentations

For 4rd domain traversal, IPv4 packets are encapsulated in IPv6 packets whose Next header is set to 4 (i.e. IPv4). If fragmentation of IPv6 packets is needed (see Section 4.5), it is performed according to [RFC2460], as shown in Figure 5. All IPv6 packets except the last one have the same length, namely the IPv6 path MTU of the Domain.

Received encapsulating packets are reassembled before being processed

according to Section 6.2.4 and Section 6.2.4.

```
                              +-------------------//---------------+
                              |             IPv4 packet            |
                              +-------------------//---------------+
                              :                                    :
                              :                                    :
                              +-----//----+-----//----+--//--+--//--+
                              |   frag 1  |   frag 2  |      |frag n|
               IPv6           +-----//----+----//-----+--//--+------+
     fragmentation extension  :           :           :      :      :
                            \ :           :           :      :      :
                 |0          \|48         :           :      :      :
                 +----------++-----//----+            :      :      :
                 |   IPv6   ||   frag 1  |            :      :      :
                 +----------++-----//----+            :      :      :
                 <---- IPv6 path MTU --->:            :      :      :
                                          :           :      :      :
                              +----------++-----//----+      :      :
                              |   IPv6   ||   frag 1  |      :      :
                              +----------++-----//----+      :      :
                              <---- IPv6 path MTU --->       :      :
                                                         ... :      :
                                                             :      :
                                                 +----------++--//--+
                                                 |   IPv6   ||frag n|
                                                 +----------++------+
```

Figure 5: IPv4 packet fragmentation for Domain traversal

6.2. Domains having only One Mapping rule

6.2.1. BR reception of an IPv4 packet

Step 1 When a BR receives an IPv4 packet at its IPv4 interface, its process depends on whether the length of CE 4rd prefix exceeds 32 bits. If not, the BR proceeds to the next step. If yes, there are three cases: (1) if the packet is a single-packet datagram, the BR proceeds to the next step; (2) if it is a fragment of an datagram that is still incomplete, the packet is kept for later reassembly of a complete IPv4 datagram (with a timeout protection as usual for this function); (3) if it is the last fragment that was missing to assemble a complete datagram, the BR proceeds to the next

step with the datagram considered as received in a single packet.

Step 2 The BR first checks that no CE IPv6 address can be derived per Section 4.4) from the IPv4 source. If this is the case, it derives from the IPv4 destination the IPv6 address of the destination CE from per Section 4.4). The CE then transmits the packet via its IPv6 interface, duly fragmented in IPv6 if necessary.

6.2.2. BR reception of an IPv4/IPv6 packet

Step 1 When a BR receives an IPv6 packet at its IPv6 interface, with the BR IPv6 address as destination, its process depends on whether the length of CE 4rd prefix exceeds 32 bits. If not, or if the IPv4 packet is a complete datagram, the BR proceeds to the next step. If yes, there are two cases: (1) if the packet contains the last fragment that was missing to assemble a complete IPv4 datagram, the BR proceeds to the next step with the reassembled datagram as if it had been received in a single IPv4 packet; (2) otherwise, the packet is kept for later reassembly of a complete IPv4 datagram (with a timeout protection as usual for this function).

Step 2 The BR checks that the IPv6 source address of the packet is not the BR IPv6 address, and is that which is derived per Section 4.4 from the IPv4 destination. If this is the case, the BR transmits the IPv4 packet via its IPv4 interface, duly fragmented in IPv4 if necessary for the link MTU.

6.2.3. CE reception of an IPv4 packet

Step 1 When a CE receives an IPv4 packet at its IPv4 interface, its process depends on whether the length of CE 4rd prefix exceeds 32 bits. If not, the BR proceeds to the next step. If yes, there are three cases: (1) if the packet is a complete IPv4 datagram, the CE proceeds to the next step; (2) if it is a fragment of datagram that is still incomplete, the packet is kept for later reassembly with other fragments (with a timeout protection as usual for this function); (3) if it is the last fragment that was missing to assemble a complete datagram, the CE proceeds to the next step with the datagram considered as received in a single packet.

Step 2 The CE tries to derive an IPv6 address from the IPv4 destination per Section 4.4. In case of success, this address taken as IPv6 destination of the encapsulating packet. Otherwise, it is the BR IPv6 address that is taken.

The CE then transmits the packet via its IPv6 interface, duly fragmented in IPv6 if necessary.

6.2.4. CE reception of an IPv4/IPv6 packet

Step 1 When a CE receives an IPv6 packet at its IPv6 interface, with the BR IPv6 address as destination, its process depends on whether the length of CE 4rd prefix exceeds 32 bits. If not, or if the IPv4 packet is a complete datagram, the BR proceeds to the next step. If yes, there are two cases: (1) if the packet contains the last fragment that was missing to assemble a complete IPv4 datagram, the CE proceeds to the next step with the reassembled datagram as if it had been received in a single IPv4 packet; (2) otherwise, the packet is kept for later reassembly of a complete IPv4 datagram (with a timeout protection as usual for this function).

Step 2 The CE tries to derive an IPv6 address from the IPv4 source per Section 4.4. If this succeeds, it checks that the obtained address is the same as the IPv6 source address. Otherwise, it checks that the IPv6 source address is the BR IPv6 address. In case of success, the CE transmits the IPv4 packet via its IPv4 interface, duly fragmented in IPv4 if necessary for the link MTU.

6.3. Domains having Multiple Mapping Rules (TBD)

7. Security considerations

Spoofing attacks

With consistency checks between IPv4 and IPv6 sources that are performed on IPv4/IPv6 packets received by BR's and CE's (Section 6), 4rd does not introduce any opportunity for spoofing attack that would not pre-exist in IPv6.

Denial-of-service attacks

In 4rd domains where IPv4 addresses are shared, the fact that IPv4 datagram reassembly may be necessary introduces an opportunity for DOS attacks (see Section 4.5). This is inherent to address sharing, and is common with other address sharing approaches such as DS-lite and NAT64/DNS64.

The best protection against such attacks is to accelerate IPv6 enablement in both clients and servers so that, where 4rd is supported, it is less and less used.

Routing-loop attacks

Routing-loop attacks that may exist in some automatic-tunneling scenarios are documented in [I-D.ietf-v6ops-tunnel-loops]. They cannot exist with 4rd because each BRs checks that the IPv6 source address of an IPv4/IPv6 packet it receives is not the 4rd-relay IPv6 address (Section 6.2.2).

8. IANA Considerations

IANA is requested to assign a DHCPv6 option number for 4rd (Section 5).

9. Acknowledgments

The authors wish to thank Mark Townsley for his active encouragements to pursue the 4rd approach since it was first introduced in [I-D.despres-softwire-sam]. Olivier Vautrin, who independently proposed a similar approach with the same acronym deserves special recognition. Particular gratitude is due to decision makers of the Japan ISP's that have announced actual 4rd deployment projects (see Section 1).

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.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[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.

[RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, February 2006.

10.2. Informative References

[I-D.despres-softwire-sam]
Despres, R., "Stateless Address Mapping (SAM) - a Simplified Mesh-Softwire Model", draft-despres-softwire-sam-01 (work in progress), July 2010.

[I-D.ietf-behave-dns64]
Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, "DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-dns64-11 (work in progress), October 2010.

[I-D.ietf-behave-v6v4-xlate-stateful]
Bagnulo, M., Matthews, P., and I. Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", draft-ietf-behave-v6v4-xlate-stateful-12 (work in progress), July 2010.

[I-D.ietf-softwire-dual-stack-lite]
Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", draft-ietf-softwire-dual-stack-lite-07 (work in progress), March 2011.

[I-D.ietf-v6ops-tunnel-loops]
Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations", draft-ietf-v6ops-tunnel-loops-03 (work in progress), February 2011.

[RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", RFC 5569, January 2010.

[RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", RFC 5969, August 2010.

Authors' Addresses

Remi Despres (editor)
RD-IPtech
3 rue du President Wilson
Levallois,
France

Email: remi.despres@free.fr

Satoru Matsushima
SoftBank
1-9-1 Higashi-Shinbashi, Munato-ku
Tokyo
Japan

Email: satoru.matsushima@tm.softbank.co.jp

Tetsuya Murakami
IP Infusion
1188 East Arques Avenue
Sunnyvale
USA

Email: tetsuya@ipinfusion.com

Ole Troan
Cisco
Bergen, Norway
France

Email: ot@cisco.com