         IPv4 Residual Deployment across IPv6-Service networks (4rd)
                     ISP-NAT's made optional
                   draft-despres-intarea-4rd-01

Abstract

   This document specifies an automatic tunneling mechanism for
   providing IPv4 connectivity service to end users over a service
   provider's IPv6 network. During the long transition period from IPv4
   to IPv6-only, a service provider's network will have to support IPv6,
   but will also have to maintain some IPv4 connectivity for a number of
   customers, for both outgoing and incoming connections, and for both
   exclusive 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 networks. In some others it can be used in parallel with NAT-
   based solutions such as DS-lite and/or NAT64/DNS4.

Table of Contents

1. Introduction

During the transition period from IPv4 to IPv6 Internet Service
Providers (ISP's), will deploy networks that are IPv6 only. Some of
them will do so while they still have to offer IPv4 connectivity.
The IPv4 service can be one or multiple IPv4 addresses per end-user,
or it can be an IPv4 address shared among multiple end-users.

In this document, Internet Service Provider 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
providing IPv4 connectivity across an IPv6 only infrastructure. As
such, it is the reverse of 6rd (IPv6 Rapid Deployment) whose purpose
is to rapidly introduce native IPv6 connectivity across an IPv4
network. 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.

The 4rd mechanism tunnels IPv4 over IPv6 using an algorithmic mapping
from IPv4 addresses or IPv4 addresses and ports to the IPv6 addresses
used as tunnel endpoints. Depending on ISP constraints and policies,
4rd can be used either standalone, with NAT44's in CE's but no NAT in
ISP networks, or can co-exist with other mechanisms in the network on
NAT's like DS-lite [I-D.ietf-softwire-dual-stack-lite] or NAT64/DNS64
[I-D.ietf-behave-v6v4-xlate-stateful] [I-D.ietf-behave-dns64].

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 BR's having the
                       same set of parameters. It offers to its 4rd-
                       capable CE's global IPv4 connectivity, both
                       outgoing and incoming, and with exclusive or
                       shared IPv4 addresses.

4rd Border Relay (BR): A 4rd-capable router managed by the service
                       provider at the edge of a 4rd domain. A BR has
                       an IPv6-enabled interface connected to the ISP
                       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
                       network and the 4rd domain. This node has an
                       IPv6 interface connected to the ISP 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 by other means
                       than 4rd itself, and used by 4rd to derive a CE
                       4rd prefix.

CE IPv6 address: In the context of 4rd, 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 4.3. Depending on its length, it is
                       an IPv4 prefix, an IPv4 address, or a shared
                       IPv4 address followed by a Port-set ID
                       (Section 4.3.2).

Port-set ID: In a CE 4rd prefix longer than 32 bits, bits
                       that follow the first 32. It algorithmically
                       identifies a set of ports exclusively assigned
                       to the CE. As specified in Section 4.3.3, the
                       set can comprise up to 4 disjoint port ranges.

   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: For a CE, the field that is common to its CE
                         IPv6 prefix and its CE 4rd prefix. In the
                         former, it follows the Domain IPv6 prefix. In
                         the latter, it follows the Domain 4rd prefix.

4. Protocol Specification

4.1. General Principles

   The principle of the 4rd protocol is that IPv4 packets, or in case of
   shared IPv4 addresses IPv4 datagrams, traverse a 4rd domain by means
   of automatic IPv4 in IPv6 tunnels. IPv6 addresses of destination
   tunnel endpoints are statelessly derived from IPv4 destinations,
   based on some mapping rule parameters, in such a way that tunnels
   between CE's follow direct IPv6 paths (i.e. without having to go via
   BR's). IPv4 destinations used for these mappings are either IPv4
   addresses alone or IPv4 addresses + ports depending on whether global
   addresses assigned to CE's are exclusive or shared.

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

4.2. Mapping-Rule Parameters

   Both CE's and BR's have to know the BR IPv6 address of their domain
   as well as, for each mapping rule, the following parameters:

   o Domain IPv6 prefix

   o Domain 4rd prefix

   o IPv6-prefix length

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

4.3. Mapping Rules

4.3.1. From a CE IPv6 Prefix to a CE 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) -------->
```
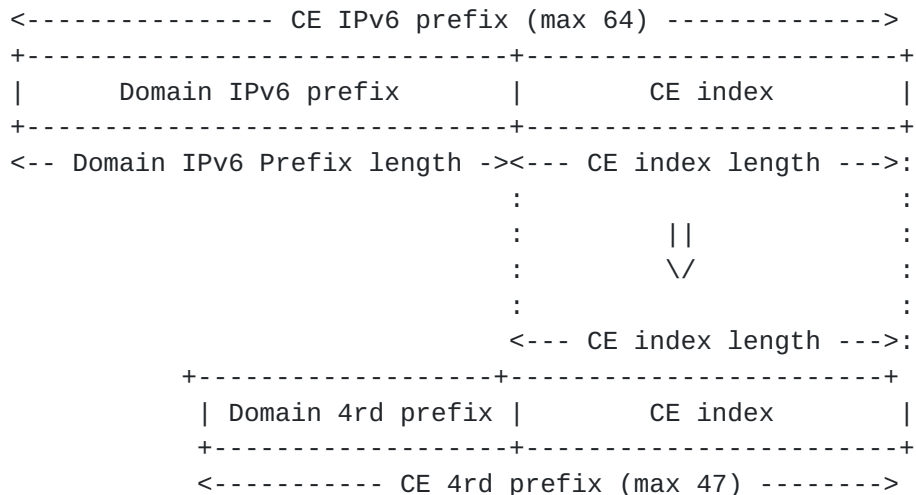
              Figure 1: From a CE IPv6 Prefix to a CE 4rd Prefix

   A CE derives its CE 4rd prefix from the IPv6 prefix it has been
   delegated on the IPv6 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).

4.3.2. **From a CE 4rd Prefix 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
   (Section 4.3.3).

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


                     <----- 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 -->
```
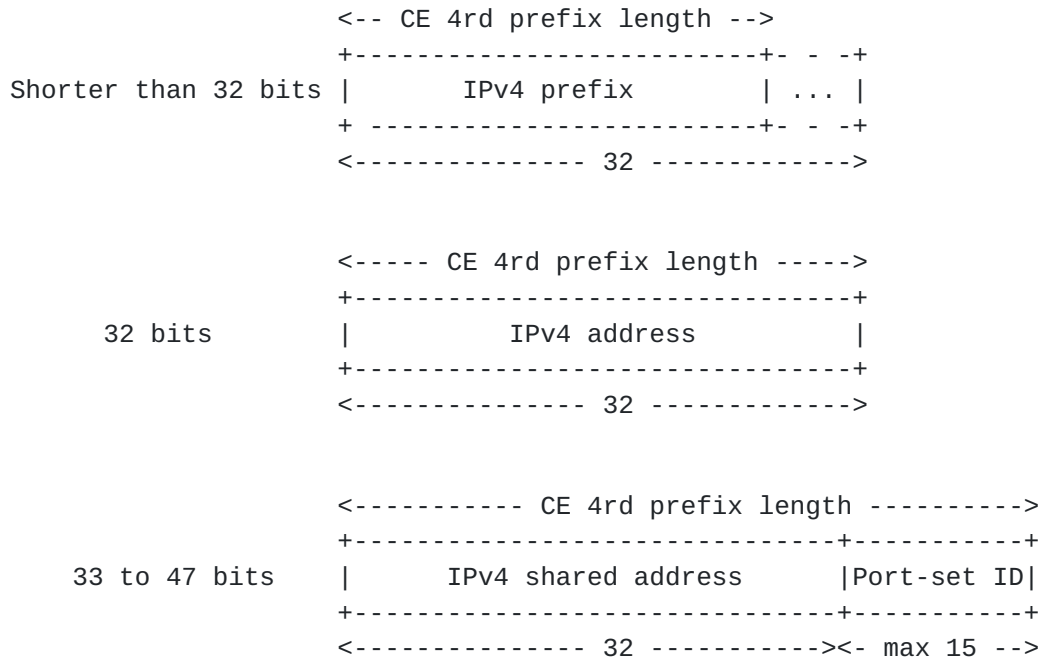
                   Figure 2: Variants of CE 4rd prefixes

4.3.3. **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
```
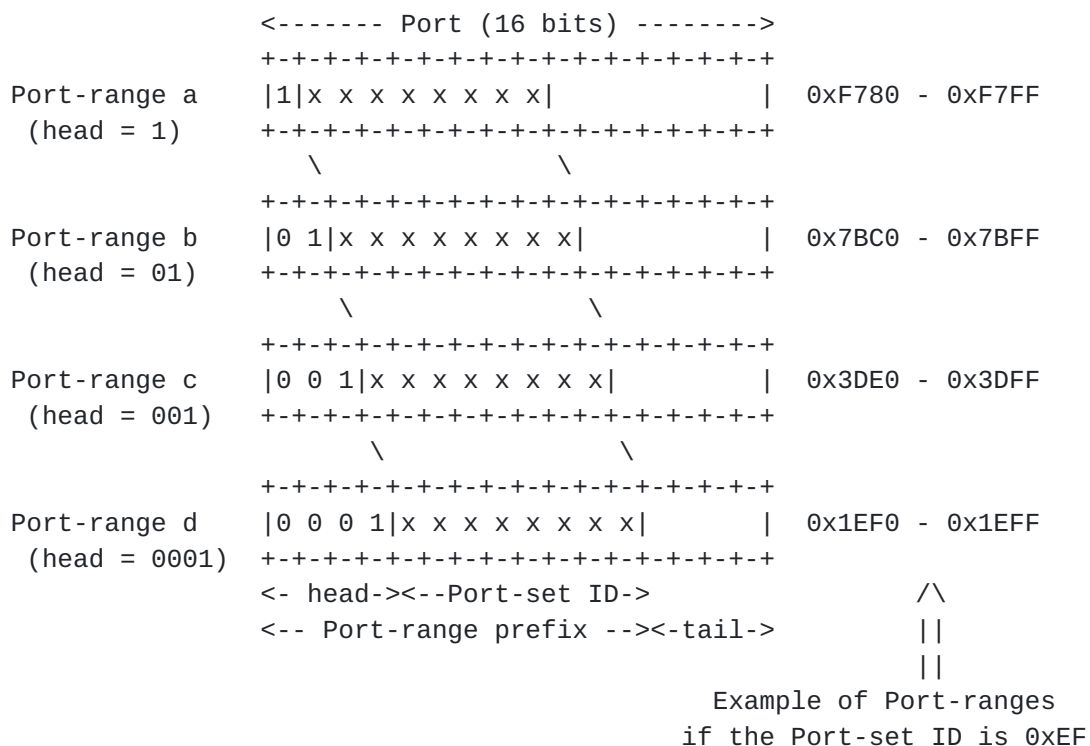
                Figure 3: From Port-set ID to Port ranges

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.

4.3.4. 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) --------------------->
```
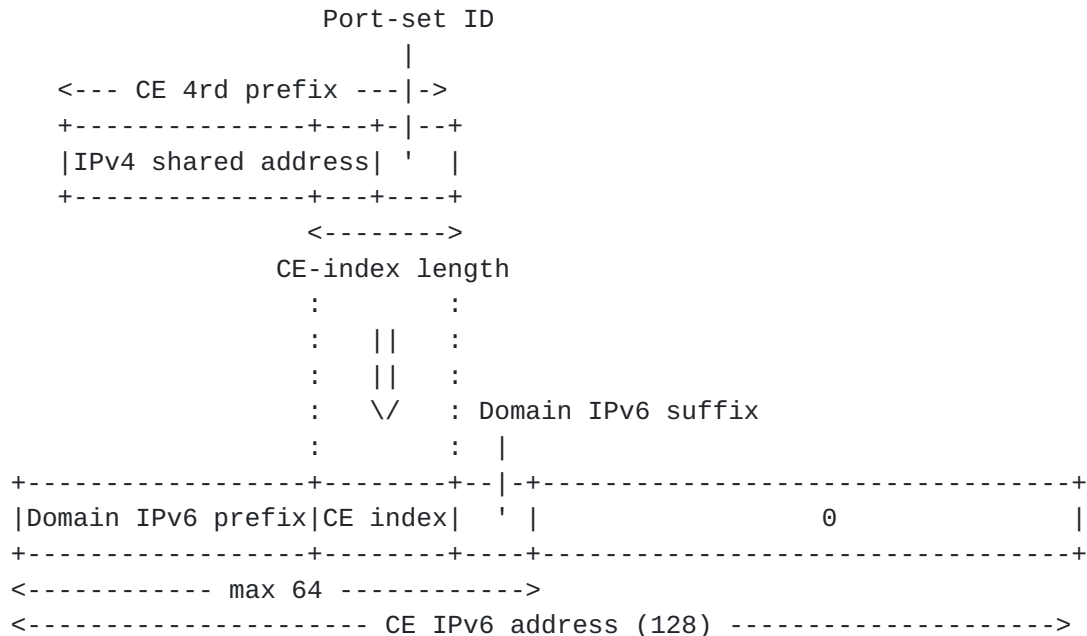
Figure 4: From 4rd Prefix to IPv6 address (shared IPv4 address case)

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 make a complete IPv6 address [RFC4291]. Figure 4
   illustrates this process in the case of a shared IPv4 address.

4.4. Encapsulation and Fragmentation Considerations

   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, it is performed according to [RFC2460],
   and as illustrated in Figure 5. Absent more specific information,
   the path MTU of a 4rd Domain has to be set to 1280 [RFC2460].

```
                          +--------------------//---------------+
                          |             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|
                              +----------++------+
```
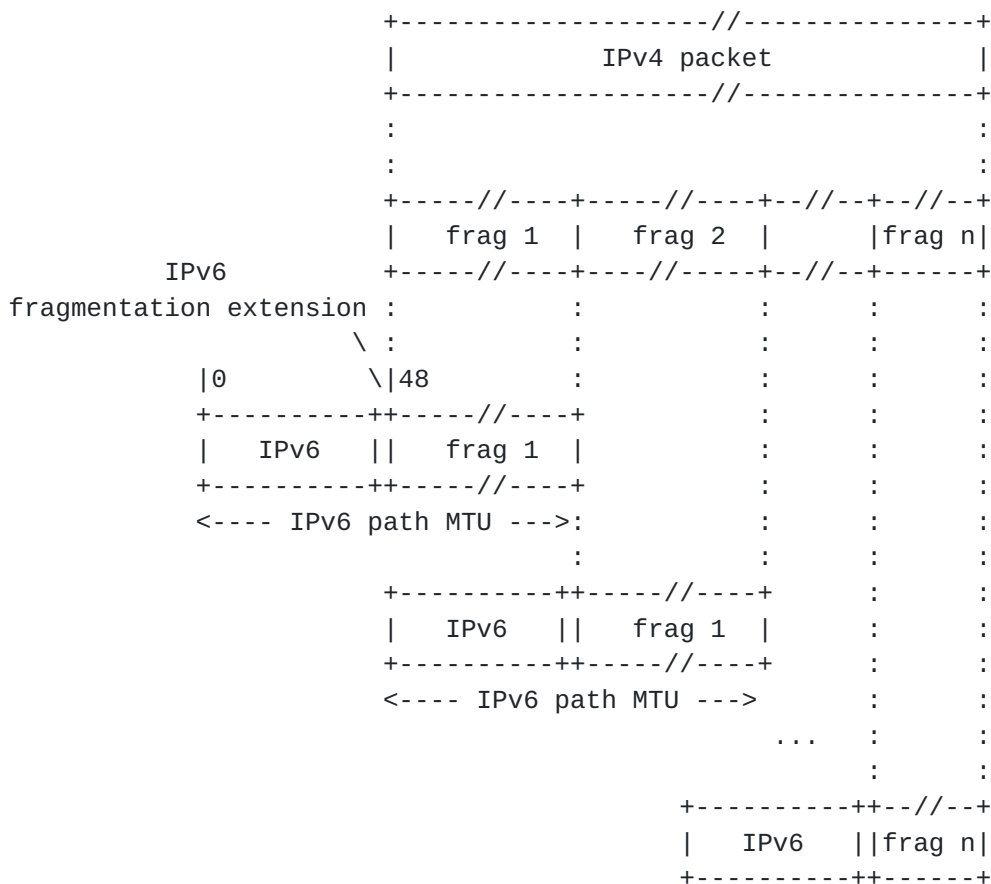
      Figure 5: Fragmentation of long IPv4 packets for Domain Traversal

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.) At Domain entrance, this ensures that all pieces of all received IPv4 datagrams go to the right IPv6 destinations.

Another peculiarity of shared IPv4 addresses is that, without precaution, a destination could simultaneously receive from different sources fragmented datagrams that have the same Datagram ID (the Identification field of [RFC0791]). This would disturb the reassembly process. To eliminate this risk, BR's and CE's SHOULD, in datagrams they receive from shared-IPv4-address CE's, replace received Datagram ID's by new ones. New values SHOULD be generated as though these datagrams would have been created locally (and with due respect of [RFC0791]). Note that replacing a Datagram ID in an IPv4 header implies an update of its Header-checksum field, by adding to it the one's complement difference between the old and the new values.

4.5. BR and CE behaviors

4.5.1. Domains having only One Mapping rule

(a) BR reception of an IPv4 packet

   Step 1 If the length of CE 4rd prefixes does not exceed 32 bits,
           the BR proceeds to step 2. Otherwise, and unless the
           packet contains a complete IPv4 datagram, IPv4 datagram
           reassembly is performed. If a complete datagram is
           available, the BR proceeds to step 2 as though the
           datagram had been received in a single packet.

   Step 2 The BR checks that the IPv4 source doesn't start with the
           Domain 4rd prefix, and that a CE IPv6 address is
           successfully derived from the IPv4 destination. In case
           of success, the packet is encapsulated and forwarded to
           this CE IPv6 address via the IPv6 interface.

   (b) BR reception of an IPv6 packet

           The BR checks that a CE IPv6 address is successfully
           derived from the source of the IPv4 encapsulated packet,
           and that the source address of the encapsulating packet is
           equal to it. In case of success: (1) if the length of CE
           4rd prefixes exceeds 32 bits, the Datagram ID of the
           packet is replaced by a locally generated one; (2) the
           IPv4 packet is forwarded via the IPv4 interface.

   (c) CE reception of an IPv4 packet

     Step 1 If the CE 4rd prefix of the CE does not exceed 32 bits and
             the IPv4 destination address starts with the Domain 4rd
             prefix, the CE proceeds to step 2. Otherwise, and unless
             the packet contains a complete IPv4 datagram, IPv4
             datagram reassembly is performed. If a complete datagram
             is available, the BR proceeds to step 2 as though the
             datagram had been received in a single packet.

     Step 2 The CE tries tries to derive a CE IPv6 address from the
             IPv4 destination. It then encapsulates the IPv4 packet
             into an IPv6 packet whose destination is this CE IPv6
             address, if one is obtained, or the BR IPv6 address
             otherwise.

   (d) CE reception of an IPv6 packet (reassembled if applicable)

           The CE checks that a CE IPv6 address is successfully
           derived from the source of the IPv4 encapsulated packet,
           AND that it is equal to the source address of the
           encapsulating packet. In case of success: (1) if the
           length of CE 4rd prefixes exceeds 32 bits, the Datagram ID
           of the packet is replaced by a locally generated one; (2)
           the IPv4 packet is forwarded via the IPv4 interface.

4.5.2. Domains having Multiple Mapping Rules

   Some ISP will want to use 4rd in networks having several Domain 4rd
   prefixes, an/or several Domain IPv6 prefixes, and/or assigning CE 4rd
   prefixes of different lengths. For this several mapping rules are
   needed.

A first possibility consists in establishing several 4rd domains,
each on having a single mapping rule. In this case, paths between
CE's belonging to different 4rd domains go from one domain to the
other in IPv4, and cross two BR's.

A second possibility permits direct IPv6 paths between CE's by
supporting several mapping rules in a single domain, as described in
this section. At time of writing, whether this will be in the 4rd
specification a MAY, a SHOULD, or a MUST, remains an open question.

(a) BR reception of an IPv4 packet

   Step 1 If a mapping rule whose length of CE 4rd prefixes does not
           exceed 32 bits applies to the IPv4 destination, the BR
           proceeds to step 2. Otherwise, and unless the packet
           contains a complete IPv4 datagram, IPv4 datagram
           reassembly is performed. If a complete datagram is
           available, the BR then proceeds to step 2 as though the
           datagram had been received in a single packet.

   Step 2 The BR checks that the IPv4 source doesn't start with the
           Domain 4rd prefix of any rule. In case of success, the
           packet is encapsulated and forwarded to this CE IPv6
           address via the IPv6 interface.

(b) BR reception of an IPv6 packet (reassembled if applicable)

           The BR checks that a CE IPv6 address is successfully
           derived from the source of the IPv4 encapsulated packet,
           and that the source address of the encapsulating packet is
           equal to it. In case of success, the BR tries to derive a
           CE IPv6 address from the destination of the encapsulated
           packet. In case of success: (1) if the source CE 4rd
           prefix exceeds 32 bits, the Datagram ID of the packet is
           replaced by a locally generated one; (2) the encapsulating
           packet is retransmitted via the IPv6 interface with this
           CE IPv6 address as destination (and the BR IPv6 address as
           source address); in case of failure, the IPv4 packet is
           decapsulated and forwarded via the IPv4 interface.

(c) CE reception of an IPv4 packet

   Step 1 If the CE 4rd prefix of the CE does not exceed 32 bits,
           and a mapping rule whose length of CE 4rd prefixes does
           not exceed 32 bits applies to the IPv4 destination, the CE
           proceeds to step 2. Otherwise, and unless the packet
           contains a complete IPv4 datagram, IPv4 datagram
           reassembly is performed. If a complete datagram is

available, the BR then proceeds to step 2 as though the
datagram had been received in a single packet.

   Step 2   The CE tries to derive a CE IPv6 address from the IPv4
            destination. It then encapsulates the IPv4 packet into an
            IPv6 packet whose destination is this CE IPv6 address, if
            one is obtained, or the BR IPv6 address otherwise.

   (d) CE reception of an IPv6 packet (reassembled if applicable)
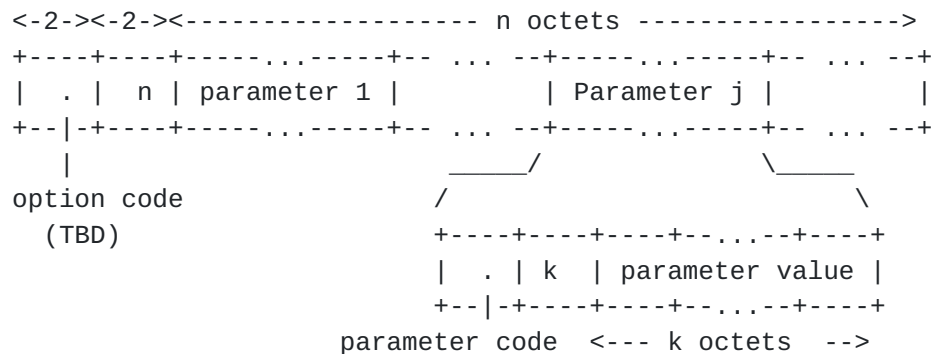
            The CE checks that a CE IPv6 address is successfully
            derived from the source of the IPv4 encapsulated packet,
            and that it is equal to the source address of the
            encapsulating packet. In case of success: (1) if the
            source CE 4rd prefix exceeds 32 bits, the Datagram ID of
            the packet is replaced by a locally generated one; (2) the
            IPv4 packet is decapsulated and forwarded via the IPv4
            interface.

   NOTE: With consistency check made between encapsulated and
   encapsulating sources in BR's and CE's when they received tunneled
   packets, no CE can forward an invalid IPv4 source address, or address
   plus port, and have it forwarded at by the egress BR or CE. Yet, if
   before tunneling a packet, a CE makes an additional check that the
   IPv4 source is consistent with the CE IPv6 address, it can discard
   invalid packets earlier than by leaving it to the egress BR or CE.
   At time of writing, whether this test can remain a MAY, or might
   require a SHOULD or a MUST remains an open question.

5. 4rd Configuration

   A CE can acquire 4rd parameters of its 4rd domain in various ways:
   manual configuration by an administrator, software download by the
   ISP, a new DHCPv6 option, etc. This document describes how to
   configure the necessary parameters via a single DHCPv6 option. A CE
   that allows IPv6 configuration by DHCPv6 SHOULD implement this
   option. Other configuration and management methods, MAY use the
   format described by this option for consistency and convenience of
   implementation on CEs that support multiple configuration methods.

   The format of Figure 6 is proposed for the DCHPv6 option. It is
   chosen to permit multiple mapping rules:

```
     <-2-><-2-><------------------ n octets ---------------->
     +----+----+-----..------+-- ... --+-----..-----+-- ... --+
     | . |  n | parameter 1 |         | Parameter j |        |
     +--|-+----+-----..------+-- ... --+-----..-----+-- ... --+
        |                       _____/              \_____
     option code              /                           \
        (TBD)             +----+----+----+--..--+----+
                          | . |  k  | parameter value |
                          +--|-+----+----+--..--+----+
                    parameter code  <--- k octets  -->


      PARAMETER-CODES (in Hexadecimal)
     0x10 :  BR IPv6 address
     0x11 :  Length of CE-IPv6-prefixes
     0x2m :  Domain IPv6 prefix, with m useful bits in last octet
     0x3m :  Domain 4rd prefix, with m useful bits in last octet
     0x4m :  Domain IPv6 suffix, with m useful bits in last octet
```

Figure 6: 4rd DHCPv6 option

In the parameter list the BR IPv6 address is first, followed by
parameters of each rule. For each rule, the order is <Domain IPv6
prefix, Domain IPv4 prefix, Length of CE IPv6 prefixes, Domain IPv6
suffix (optional)>.

6. 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 4.5), 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 (Section 4.4). 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 a received IPv6 packet is a CE address (Section 4.5.1
   (b) and Section 4.5.2 (b) />).

Attacks facilitated by restricted port sets

   From hosts that are not subject to ingress filtering of [RFC2827],
   some attacks are possible by intervening with faked packets during
   ongoing transport connections ([RFC4953], [RFC5961], [RFC6056].
   These attacks, that have mitigations of their own are easier with
   hosts that only use restricted port sets (they depend on guessing
   which ports are currently used by target hosts). To avoid using
   restricted port sets, the easiest approach consists in increasing
   the proportion of connections that are IPv6, i.e. using
   unrestricted port sets.

7. IANA Considerations

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

8. 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]. Questions raised by Wojciech Dec have
   been useful to clarify explanations. 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 (www.ietf.org/mail-archive/web/v6ops/current/
   msg05247).

9. References

9.1. Normative References

   [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791,
              September 1981.

   [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.

9.2. 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.

[RFC2827]   Ferguson, P. and D. Senie, "Network Ingress Filtering:
            Defeating Denial of Service Attacks which employ IP Source

                 Address Spoofing", BCP 38, RFC 2827, May 2000.

   [RFC4953] Touch, J., "Defending TCP Against Spoofing Attacks",
              RFC 4953, July 2007.

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

   [RFC5961] Ramaiah, A., Stewart, R., and M. Dalal, "Improving TCP's
              Robustness to Blind In-Window Attacks", RFC 5961,
              August 2010.

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

   [RFC6056] Larsen, M. and F. Gont, "Recommendations for Transport-
              Protocol Port Randomization", BCP 156, RFC 6056,
              January 2011.

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