

Internet Engineering Task Force
Internet-Draft
Intended status: Experimental
Expires: December 24, 2012

R. Despres, Ed.
RD-IPtech
B. Carpenter
Univ. of Auckland
D. Wing
Cisco
S. Jiang
Huawei Technologies Co., Ltd
June 22, 2012

**Native IPv6 Behind NAT44 CPEs (6a44)
draft-despres-6a44-02**

Abstract

In customer sites having IPv4-only CPEs, Teredo provides a last resort IPv6 connectivity [[RFC4380](#)] [[RFC5991](#)] [[RFC6081](#)]. However, because it is designed to work without involvement of Internet service providers, it has significant limitations (connectivity between IPv6 native addresses and Teredo addresses is uncertain; connectivity between Teredo addresses fails for some combinations of NAT types). 6a44 is a complementary solution that, being based on ISP cooperation, avoids these limitations. At the beginning of IPv6 addresses, it replaces the Teredo well-known prefix by network specific /48 prefixes assigned by local ISP's (an evolution similar to that from 6to4 to 6rd). The specification is complete enough for actual deployment, including with independently written codes.

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 December 24, 2012.

Copyright Notice

Copyright (c) 2012 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.

Table of Contents

1.	Introduction	3
2.	Requirements Language	4
3.	Definitions	4
4.	Design Goals, Requirements, and Model of Operation	6
4.1.	Hypotheses about NAT Behavior	6
4.2.	Native IPv6 Connectivity for unmanaged Hosts behind NAT44s	6
4.3.	Operational Requirements	7
4.4.	Model of Operation	8
5.	6a44 Addresses	11
6.	Specification of Clients and Relays	13
6.1.	Packet Formats	13
6.2.	IPv6 Packet Encapsulations	13
6.3.	6a44 Bubbles	13
6.4.	Maximum-Transmission-Unit Considerations	15
6.5.	6a44 Client Specification	16
6.5.1.	Tunnel Maintenance	16
6.5.2.	Client Transmission	18
6.5.3.	Client Reception	20
6.6.	6a44 Relay Specification	22
6.6.1.	Relay Reception in IPv6	22
6.6.2.	Relay Reception in IPv4	23
6.7.	Implementation of Automatic Sunset	26
7.	Security Considerations	26
8.	IANA considerations	30
9.	Acknowledgments	30
10.	References	31
10.1.	Normative References	31
10.2.	Informative References	31
	Authors' Addresses	33

1. Introduction

Although most customer premise equipments (CPE's) should soon be dual-stack capable, a large installed base of IPv4-only CPE's is likely to remain for several years. Their operation is based on IPv4 NAT's (NAT44's). Also, due to the IPv4 address shortage, more and more Internet service providers (ISP's), and more and more mobile operators, will assign private IPv4 addresses of [\[RFC1918\]](#) to their customers (the [\[NAT444\]](#) model). For a rapid and extensive use of IPv6 [\[RFC2460\]](#), there is therefore a need for IPv6 connectivity behind NAT44's, including those of the [\[NAT444\]](#) model.

At the moment, there are two tunneling techniques specified for IPv6 connectivity behind NAT44's:

- o Configured tunnels. They involve tunnel brokers with which users must register [\[RFC3053\]](#). Well-known examples include deployments of the Hexago tool, and the SixXs collaboration, which are suitable for IPv6 early trials. However, this approach is not adequate for mass deployment: it imposes that, even if two hosts are in the same customer site, IPv6 packets between them must transit via tunnel servers, which may be far away.
- o Automatic Teredo tunnels [\[RFC4380\]](#) [\[RFC5991\]](#). Teredo is specified as a last resort solution which, due to its objective to work without local ISP involvement, has the following limitations:
 - * Connectivity between IPv6 native addresses and Teredo addresses is uncertain. (As explained in [\[RFC4380\]](#) section 8.3, this connectivity depends on paths being available from all IPv6 native addresses to some Teredo Relays. ISP's lack sufficient motivations to ensure it).
 - * Between two Teredo addresses, IPv6 connectivity fails for some combinations of NAT44 types([\[RFC6081\]](#) section 3).
 - * According to [\[RFC4380\]](#) section 5.2, each Teredo host has to be configured with the IPv4 address of a Teredo server (a constraint that can however be avoided in some implementations).

6a44 is designed to avoid Teredo limitations where ISP's can participate to the solution. The approach for this is similar to that which permitted 6rd [\[RFC5569\]](#) [\[RFC5969\]](#) to avoid limitations of 6to4 [\[RFC3056\]](#) [\[RFC3068\]](#): at the beginning of IPv6 addresses, the Teredo well-known prefix is replaced by network specific prefixes assigned by local ISP's.

This document is organized as follows: terms used in the document are defined in [Section 3](#); design goals and model of operation are presented in [Section 4](#); [Section 5](#) describes the format of 6a44 IPv6 addresses; [Section 6](#) specifies in details behaviors of 6a44 clients and 6a44 relays; security and IANA considerations are respectively covered in [Section 7](#) and [Section 8](#).

The specification is expected to be complete enough for running codes to be independently written and the solution to be incrementally deployed and used. Its intended status is Experimental rather than Standard to reflect uncertainty as to which major Internet players may be willing to support it.

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. Definitions

The following definitions are used in this document:

MAJOR NEW DEFINITIONS

"6a44 ISP network": An IPv4-capable ISP network that supports at least one 6a44 relay. Additional conditions are that it assigns individual IPv4 addresses to its customer sites (global or private), that it supports ingress filtering [[RFC2827](#)], and that its path MTU's are at least 1308 octets.

"6a44 relay": A node that supports the 6a44 relay function defined in this document, and that has interfaces to an IPv6-capable upstream network and to an an IPv4-capable downstream network.

"6a44 client": A host that supports the 6a44 client function defined in this document, and has no other mean than 6a44 to have a IPv6 native address.

"6a44 tunnel": A tunnel established and maintained between a 6a44 client and 6a44 relays of its ISP network.

"6a44 bubble": A UDP/IPv4 packet sent from a 6a44 client to the 6a44-relay address, or conversely, and having a UDP payload that cannot be confused with an IPv6 packet. In the client to relay direction, it is a request for a response bubble. In the relay to

client direction, it conveys the up-to-date IPv6 prefix of the client.

SECONDARY NEW DEFINITIONS

(for reference, can be skipped by readers familiar with usual terminology)

"6a44 service": The service offered by a 6a44 ISP network to its 6a44 clients.

"6a44-client IPv6 address": The IPv6 address of a 6a44 client. It is composed of the client IPv6 prefix, received from a 6a44 relay, followed by the client local IPv4 address.

"6a44-client IPv6 prefix": For a 6a44 client, the IPv6 prefix (/96) composed of the IPv6 prefix of the local 6a44-network (/48) followed by the UDP/IPv4 mapped address of the client (32 + 16 bits).

"6a44-client UDP/IPv4 mapped address": For a 6a44 client, the external UDP/IPv4 address that, in the CPE NAT44 of the site, is that of its 6a44 tunnel.

"6a44-client UDP/IPv4 local address": For a 6a44 client, the combination of its local IPv4 address and the 6a44 port.

"6a44 port": The UDP port used for 6a44 (see [Section 8](#)).

"6a44-relay UDP/IPv4 address": The UDP/IPv4 address composed of the 6a44-relay anycast address and the 6a44 port.

"6a44-relay anycast address": The well-known IPv4 anycast address of 6a44 relays, to be reserved by IANA (see [Section 8](#)).

"6a44-network IPv6 prefix": An IPv6 /48 prefix assigned by an ISP to a 6a44 network.

USUAL DEFINITIONS

(for reference, can be skipped by readers familiar with usual terminology)

"Upstream direction": For a network border node, the direction toward the Internet core.

"Downstream direction": For a network border node, the direction toward end-user nodes (opposite to the upstream direction).

"IPv4 private address": An address that starts with one of the three prefixes of [\[RFC1918\]](#) (10/8, 172.16/12, or 192.168/16).

"IPv6 native address": An IPv6 global unicast address that starts with an aggregatable prefix assigned to an ISP.

"UDP/IPv4 address": The combination of an IPv4 address and a UDP port.

"UDP/IPv4 packet": A UDP datagram contained in an IPv4 packet.

"IPv6/UDP/IPv4 packet": An IPv6 packet contained in a UDP/IPv4 packet.

[4.](#) Design Goals, Requirements, and Model of Operation

[4.1.](#) Hypotheses about NAT Behavior

6a44 is designed to work with NAT44 behaviors identified in [section 3 of \[RFC6081\]](#). In particular, it has to work with endpoint-dependent mappings as well as with endpoint-independent mappings, including if there are dynamic changes from one mode to the other.

The only assumption is that, after a mapping has been established in the NAT44, it is maintained as long as it is re-used at least once, in each direction, every 30 seconds.

NOTE: 30 seconds is the value used for the same mapping-maintenance purpose in Teredo [\[RFC4380\]](#), and in SIP [\[RFC5626\]](#).

[4.2.](#) Native IPv6 Connectivity for unmanaged Hosts behind NAT44s

The objective remains that, as soon as possible, CPEs and ISPs support IPv6 native prefixes. 6a44 is therefore designed only as a temporary solution for hosts to obtain IPv6 native addresses in sites whose CPEs are not IPv6-capable yet.

As noted in [Section 1](#), IPv6 native addresses obtainable with

configured tunnels have important limitations. However, compared to 6a44 addresses, they have the advantage of remaining unchanged in case of NAT44 reset. 6a44 remains therefore the last resort solution for IPv6 native addresses in unmanaged hosts of IPv4-only-CPE sites, while configured tunnels may still be preferred for some managed hosts if reported limitations of configured tunnels are consciously found acceptable. Their scopes being different, the two solutions can usefully coexist.

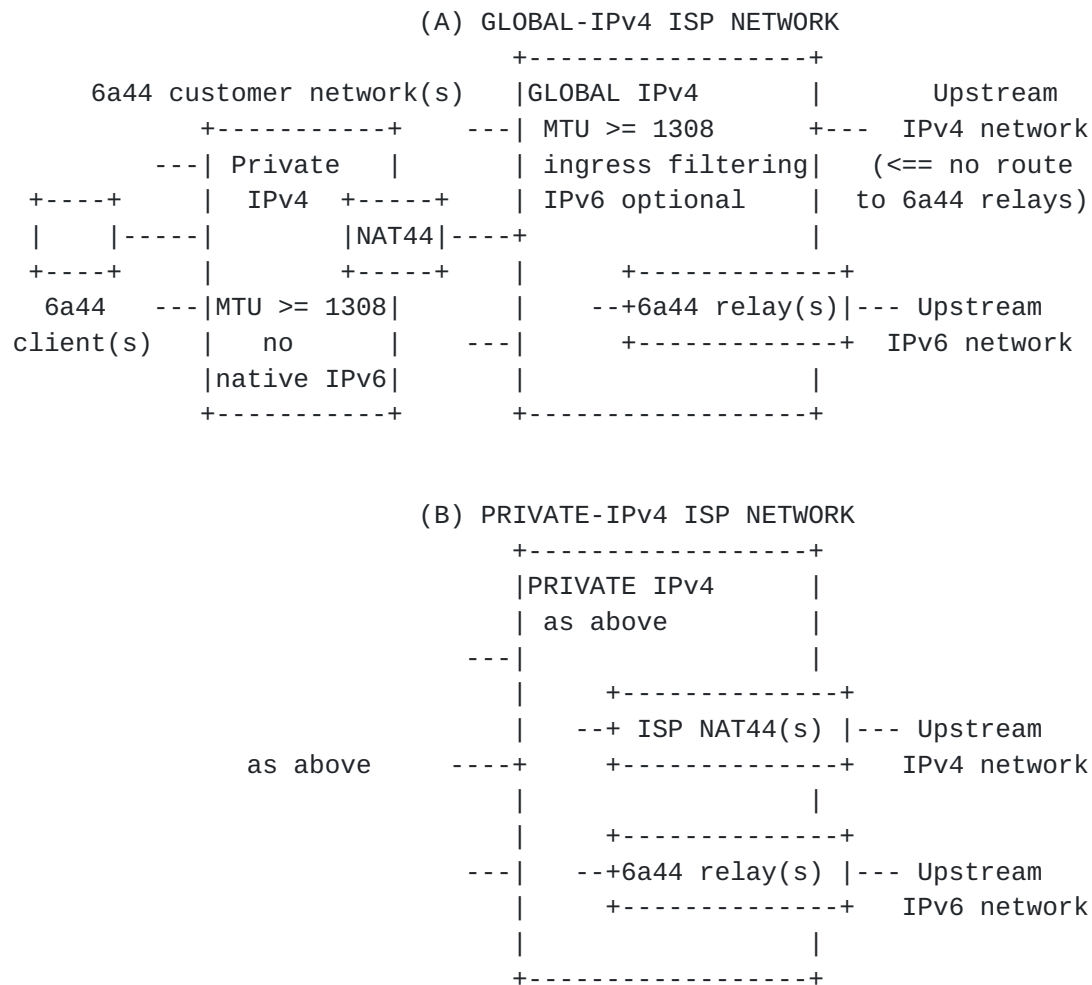
Note that Teredo remains a last resort solution for hosts to have IPv6 addresses where IPv6 native addresses cannot be available (and where Teredo limitations are consciously found acceptable).

4.3. Operational Requirements

Operational requirements of 6a44 include the following:

- "Robust IPv6 connectivity": A node having a 6a44 address must have paths across the Internet to and from all IPv6 native addresses that are not subject to voluntary firewall filtering.
- "Intra-site path efficiency": Packets exchanged between 6a44 clients that are behind the same CPE NAT44 must not have to traverse it. If these clients have IPv4 connectivity using their private IPv4 addresses, they must also have IPv6 connectivity using their 6a44 addresses.
- "Plug-and-play operation of 6a44 clients": In order to obtain a 6a44 address from its local ISP, a 6a44 client must need no parameter configuration.
- "Scalability of ISP functions": For the solution to be easily scalable, ISP-supported functions have to be completely stateless.
- "Anti-spoofing Protection": Where address anti-spoofing is ensured in IPv4 with ingress filtering of [\[RFC2827\]](#) [\[RFC3704\]](#), IPv6 addresses must benefit from the same degree of anti-spoofing protection.
- "Overall operational simplicity": As Antoine de Saint-Exupery said in [The Tool], "it seems that perfection is attained not when there is nothing more to add, but when there is nothing more to remove".
- "Incremental deployability": Hosts and ISP networks must be able to become 6a44 capable independently of each other. IPv6 must be operational where both are available, and there must be no perceptible effect where they are not both available.

4.4. Model of Operation



6a44 APPLICABILITY SCENARIOS

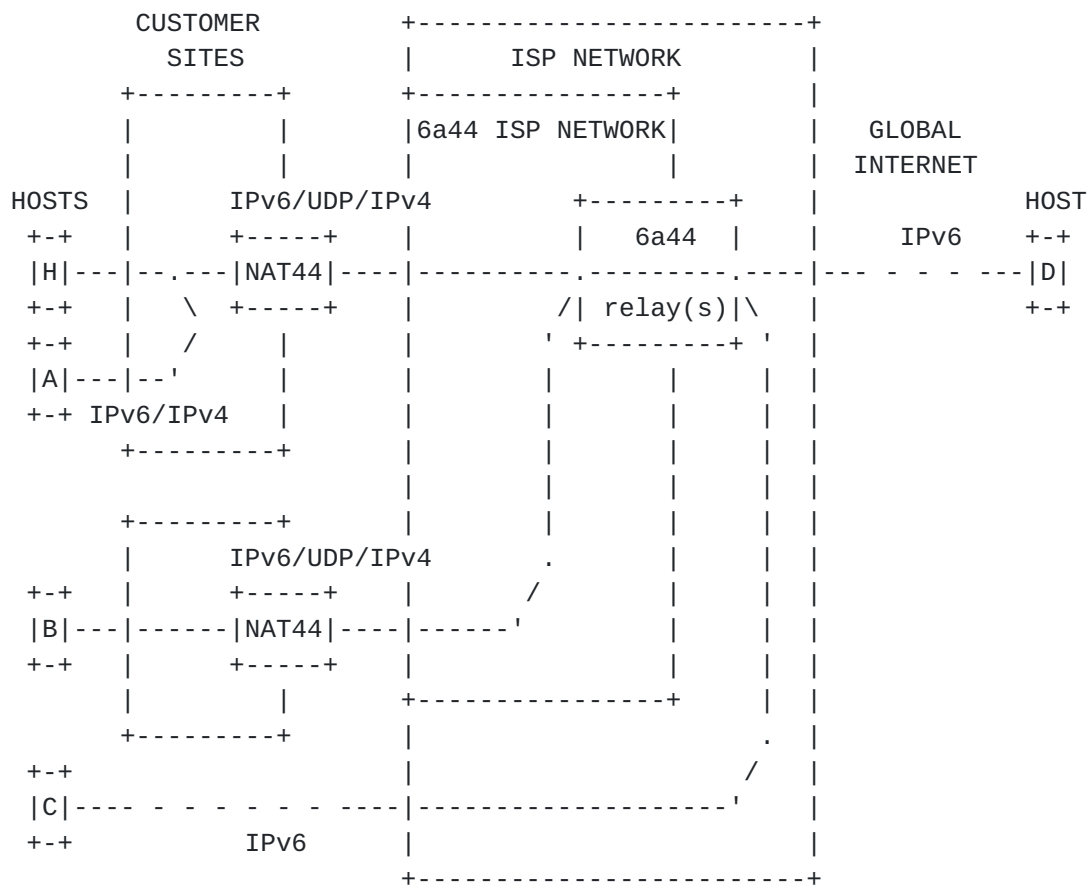
Figure 1

The operation of 6a44 involves two types of nodes: 6a44 clients and 6a44 relays. Figure 1 shows the two applicability scenarios:

- o In the first one, IPv4 addresses assigned to customer sites are global IPv4.
- o In the second one, they are private IPv4 addresses ([[NAT444](#)] model where ISPs operate one or several NAT44's, also called carrier-grade NATs, or CGN's).

In both configurations, the ISP network may also assign IPv6 prefixes to customer sites:

- o If customer sites are only assigned IPv4 addresses (IPv6 prefix available neither natively nor with any tunnel), 6a44 applies not only to sites whose CPE's are IPv4-only capable, but also to those whose CPE's are dual-stack capable.
- o If customer sites are assigned both IPv4 addresses and IPv6 prefixes, 6a44 only applies to sites whose CPE's are IPv4-only capable.



IPv6 PATHS H-A: A is a 6a44 client in the same site
 H-B: B is a 6a44 client in another site of the same ISP
 H-C: C is IPv6 of the same ISP, other than 6a44
 H-D: D is IPv6 of another ISP

IPv6 PATHS BETWEEN 6a44 HOSTS AND REMOTE HOSTS

Figure 2

Figure 2 illustrates paths of IPv6 packets in between a 6a44 client H and various possible locations of remote hosts (A in the same site, B in another 6a44 site of the same ISP, C in a non-6a44 IPv6 site of the same ISP, D in an IPv6 site of another ISP). Between 6a44 clients of a same site, IPv6 packets are encapsulated in IPv4 packets. Those Between 6a44 clients and 6a44 relays are encapsulated in UDP/IPv4 packets.

6a44 operates as follows (details in [Section 6](#)):

1. A 6a44 client starts operation by sending a 6a44 bubble to the 6a44-relay UDP/IPv4 address.
2. When a 6a44 relay receives a bubble from one of its 6a44 clients, it returns to this client a bubble containing the IPv6 prefix of this client.
3. When a 6a44 client receives a bubble from a 6a44 relay, it updates (or confirms) its 6a44 address. It is an update if the client has no IPv6 address yet or if, due to a CPE reset, this address has changed. After receiving a bubble, a client is ready to start, or to continue, IPv6 operation.
4. When a 6a44 client having a 6a44 address has an IPv6 packet to send whose destination IS in the same customer site, it encapsulates it in an IPv4 packet whose destination is found in the IPv6 destination address. It then sends the resulting IPv6/IPv4 packet.
5. When a 6a44 client receives a valid IPv6/IPv4 packet from a 6a44 client of the same site, it decapsulates the IPv6 packet and submits it to further IPv6 processing.
6. When a 6a44 client having a 6a44 address has an IPv6 packet to send whose destination IS NOT in the same the same customer site, it encapsulates the packet in a UDP/IPv4 packet whose destination is UDP/IPv4 address of 6a44 relays. It then sends the IPv6/UDP/IPv4 packet.
7. When a 6a44 relay receives via its IPv4 interface a valid IPv6/UDP/IPv4 packet whose destination IS one of its 6a44 clients, it forwards the contained IPv6 packet in a modified IPv6/UDP/IPv4 packet. The UDP/IPv4 destination of this packet is found in the IPv6 destination address.
8. When a 6a44 client receives a valid IPv6/UDP/IPv4 packet from a 6a44 relay, it decapsulates the IPv6 packet and submits it to further IPv6 processing.

9. When a 6a44 relay receives via its IPv4 interface a valid IPv6/UDP/IPv4 packet whose IPv6 destination IS NOT one of its 6a44 clients, it decapsulates the IPv6 packet and sends it via its IPv6 interface.
10. When a 6a44 relay receives via its IPv6 interface a valid IPv6 packet whose destination is one of its 6a44 clients, it encapsulates the packet in a UDP/IPv4 packet whose destination is the UDP/IPv4 address found in the IPv6 destination address. It then sends the resulting IPv6/UDP/IPv4 packet via its IPv4 interface.
11. To maintain the NAT44 mapping of its 6a44 tunnel, and to quickly detect the need to change its 6a44 address in case of NAT44 reset, a 6a44 client sends from time to time a bubble to the 6a44 relay address (see [Section 6.5.1](#)).
12. When a 6a44 relay receives via its IPv4 interface an IPv6/UDP/IPv4 packet whose IPv6 and UDP/IPv4 source addresses are not consistent, it discards the invalid packet, and returns a bubble to the UDP/IPv4 source address. (This permits the 6a44 client at this address to update its IPv6 address).

5. 6a44 Addresses

The 6a44 IPv6 address an ISP assigns to a host must contain all pieces of information needed to reach it from other IPv6 addresses. These pieces are, as illustrated in Figure 3:

- o the 6a44-network IPv6 prefix D (a /48 the ISP has assigned to its 6a44 relays);
- o the customer-site IPv4 address N (either global IPv4 or, if the ISP uses a [[NAT444](#)] model, private IPv4);
- o the mapped port Z of the 6a44 tunnel (i.e. the external port assigned by the NAT44 to the tunnel that the client maintains between its UDP/IPv4 local address A:W and the 6a44-relay UDP/IPv4 address B:W).
- o the client local IPv4 address A (i.e. the private IPv4 address assigned to the client in its customer site; it is needed for intra-site IPv6 connectivity).

NOTE: 6a44 addresses are not guaranteed to comply with the rule of [\[RFC4291\]](#) according to which bits 64-127 of aggregatable unicast addresses have to be the Modified-EUI-64 IID format. However, these bits of 6a44 addresses are interpreted only where 6a44 addresses are processed, i.e. in 6a44 relays and clients. No operational problem is therefore foreseen. Besides, because it is a purely transitional tool, it shouldn't prevent any "development of future technology that can take advantage of interface identifiers with universal scope" (the purpose of this format expressed in [\[RFC4291\]](#)).

6. Specification of Clients and Relays

6.1. Packet Formats

6.2. IPv6 Packet Encapsulations

For NAT44 traversal, an IPv6 packet transmitted from a 6a44 client to a 6a44 relay or conversely is encapsulated in a UDP/IP packet whose source and destinations addresses are those of the two endpoints (A:W and B:W in notations of Figure 3). The IPv4 packet is that of a complete datagram (its more-fragment bit is set to 0, its offset is set to 0, and its datagram identification may be set to 0). The UDP checksum is set to 0 (there is no need for an additional layer of checksum protection). The length of the IPv6 packet SHOULD NOT exceed 1280 octets (see [Section 6.4](#)).

```

Octets: |0           |20 |28           |68           |
        +-----+---+-----+-----+-----+//-----+
        |  IPv4   |UDP|   IPv6 header   | IPv6 payload |
        +-----+---+-----+-----+-----+//-----+

```

An IPv6 packet transmitted from a 6a44 client to another 6a44 client of the same site is encapsulated in an IPv4 packet whose source and destination addresses are the private-IPv4 addresses of the two hosts. The IPv4 packet is that of a complete datagram (its more-fragment bit is set to 0, its offset is set to 0, and its datagram identification may be set to 0). The size of the IPv6 packet SHOULD NOT exceed 1280 octets for off-link destinations, and MUST NOT exceed the link MTU minus 20 octets for on-link destinations (see [Section 6.4](#)).

```

Octets: |0           |20           |60           |
        +-----+-----+-----+//-----+
        |  IPv4   |   IPv6 header   | IPv6 payload |
        +-----+-----+-----+//-----+

```

6.3. 6a44 Bubbles

A Bubble is a UDP/IPv4 packet whose UDP payload comprises a "6a44-client IPv6 prefix" field and a "Bubble ID" field, and whose UDP checksum is set to 0. Having no UDP checksum protection in bubbles is a simplification that is acceptable because bubble contents are regularly updated and non-critical (a client accepting a corrupted IPv6 prefix never leads to any IPv6 packet being accepted by any wrong destination).

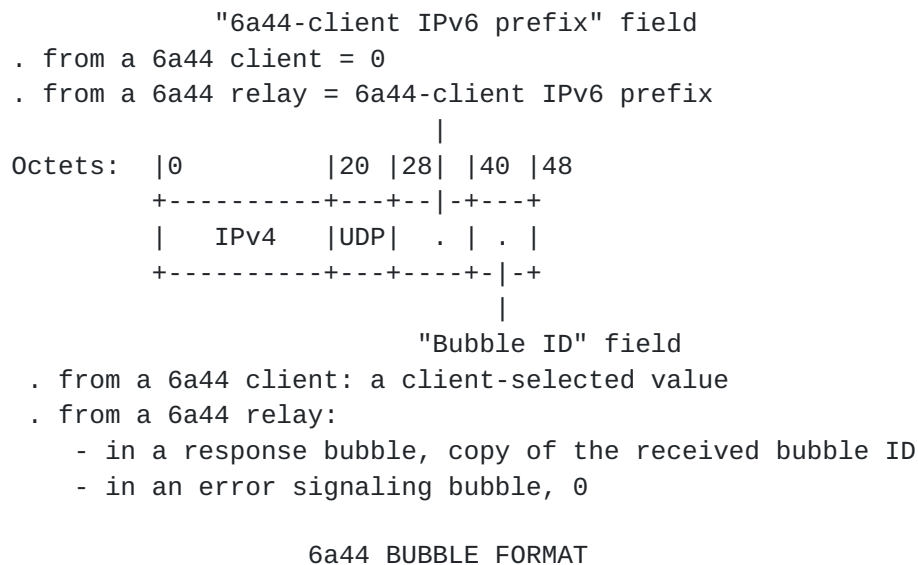


Figure 4

In a bubble from a 6a44 client to a 6a44 relay, the "6a44-client IPv6 prefix" field is only reserved space for the response. It is set to 0. In a bubble from a 6a44 relay to a 6a44 client, it contains the IPv6 prefix of the client, left justified.

In a bubble from a 6a44 client to a 6a44 relay, the "Bubble ID" field contains a randomly chosen value, renewed in circumstances defined in [Section 6.5.1](#). In a bubble from a 6a44 relay to a 6a44 client: if the bubble is a response to a bubble received from the client, the field contains the value found in the received bubble; if the bubble is a reaction to a received IPv6/UDP/IPv4 packet whose IPv6 and UDP/IPv4 sources are inconsistent (i.e. not conforming to R44-2 (3) of [Section 6.6.2](#)), the field is set to 0. The purpose of this field is a protection against 6a44-relay spoofing attacks (see [Section 7](#)).

In order to preserve forward compatibility with any extension of bubble formats, should one prove useful in the future, 6a44 clients and 6a44 relays MUST accept to receive bubbles whose UDP payloads lengths are longer than 20 octets (up to that of an IPv6-packet header since, as detailed in [Section 6.5.3](#) and [Section 6.6.2](#), bubbles are recognized by their lengths being shorter than that of tunneled IPv6 packets).

6.4. Maximum-Transmission-Unit Considerations

Reassembly of a fragmented IPv4 datagram necessitates to remember its identifier from reception of the first fragment to reception the last one, and necessitates a timeout protection against packet losses. If such an IP-layer stateful processing would be necessary for 6a44, it would make it more complex than needed, would introduce a vulnerability to denial of service attacks, and would impose that all fragments of a fragmented IPv4 datagram go to the same relay. This last point would be a constraint on how load balancing may be performed between multiple 6a44 relays, and would therefore be detrimental to scalability.

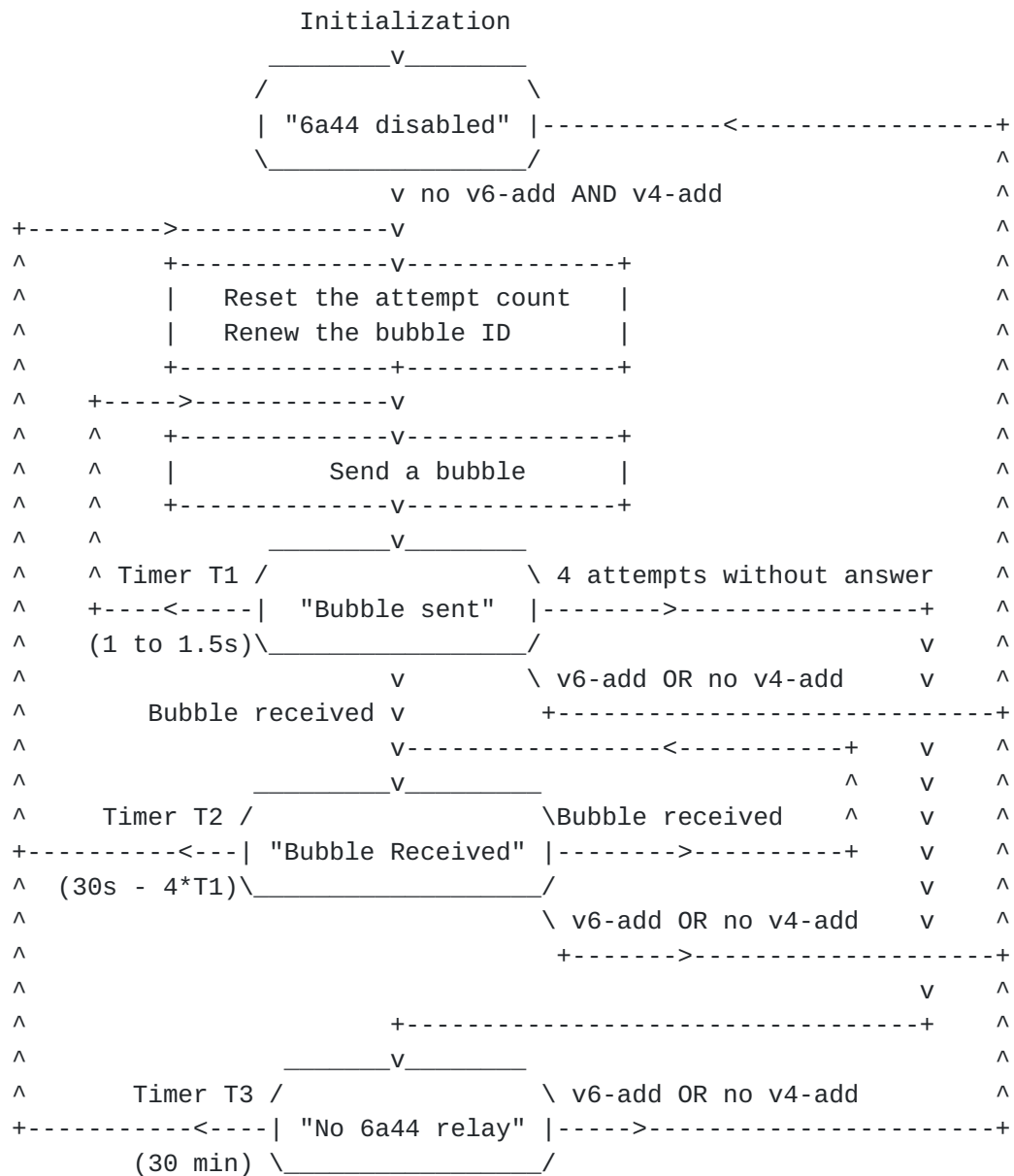
For 6a44 processing to remain completely stateless, IPv4 packets containing encapsulated IPv6 packets must never be fragmented (DF always set to 1). For this:

- o In customer sites, 6a44 clients **MUST** have IPv4 link MTU's that support encapsulated IPv6 packets of lengths up to 1280 octets, i.e., for IPv6/UDP/IPv4 packets that traverse the CPE, link MTU's of at least $1280+20+8=1308$ octets. (This condition is in general satisfied).
- o For the same reason, 6a44 ISP networks must have IPv4 path MTU's of at least 1308 octets. (This condition is in general satisfied).
- o 6a44 clients **SHOULD** limit the size of IPv6 packets they transmit to 1280 octets.
- o 6a44 relays **SHOULD** set their IPv6 MTU to 1280. (If a relay receives an IPv6 packets longer than this MTU via its IPv6 upstream interface, it **MUST** return ICMPv6 Packet Too Big message). Typical ISP networks have a path MTU's that would permit IPv6 MTU's of 6a44 to be longer than 1280, but taking 1280 octets is a precaution that guarantees against problems with customer sites that may have internal path MTU's smaller than those supported by their ISP networks.

6.5. 6a44 Client Specification

6.5.1. Tunnel Maintenance

For a 6a44-client IPv6 address to remain valid, the port mapping of the 6a44 tunnel MUST be maintained in the CPE NAT44.



TUNNEL MAINTENANCE ALGORITHM

Figure 5

For this, the 6a44 client SHOULD apply the equivalent of the following TM-x rules, illustrated in Figure 5:

- TM-1 At initialization, a timer value T1 is randomly chosen in the recommended range 1 to 1.5 seconds, and the "6a44 disabled" state is entered. (Randomness of this value is a precaution to avoid that, if many hosts happen to be re-initialized at the same time, the bubble traffic resulting from the following rules be synchronized.)

- TM-2 In the "6a44-disabled" state, if it appears that the the interface has no IPv6 native address BUT has a private IPv4 address, then: the Attempt count (a local variable) is set to 1; a new Bubble ID (another local variable) is randomly chosen (how much random is this new value is not critical, as explained in [Section 7](#)); a bubble is sent with this bubble ID; the "Bubble sent" state is entered with the timer set to T1.

- TM-3 In the "Bubble sent" state, if the timer expires AND the Attempt count is less than 4, then: the Attempt count is increased by 1; a new bubble is sent with the current Bubble ID; the "bubble sent" state is re-entered with the timer reset to T1.

- TM-4 In the "Bubble sent" state, if a bubble is received, then: the 6a44-client IPv6 address is set to the received 6a44-client IPv6 prefix followed by the host local IPv4 address; the "Bubble received" state is entered with the timer set to T2 whose recommended value is 30 seconds minus 4 times T1.

- TM-5 In the "Bubble sent" state, if timer T1 expires AND the Attempt count is equal to 4, then: the "No 6a44 relay" state is entered with the timer set to T3 whose recommended value is 30 minutes.

- TM-6 In the "Bubble sent" state, OR the "Bubble received" state, OR the "No 6a44 relay" state, if a IPv6 native address is obtained by some other mean, OR if the private IPv4 address of the host is no longer valid, then: the timer is disarmed; the "6a44 disabled" state is entered.

- TM-7 In the "Bubble received" state, if the timer T2 expires, then: the Attempt count is reset to 1; a new Bubble ID is randomly chosen; a bubble is sent with this bubble ID; the "Bubble sent" state is entered with the timer set to T1.

- TM-8 In the "Bubble received" state, if a bubble is received, then: the timer is reset to T2. (NOTE: Since a bubble is received by a 6a44 client either in response to a bubble it has sent or in reaction to a packet it has sent with inconsistent IPv6 and UDP/IPv4 source addresses, receiving a bubble is a sign that the tunnel mapping reported in the received bubble prefix has recently been used in BOTH directions, a condition required by some NAT44s to maintain their mappings).
- TM-9 In the "no 6a44 relay" state, if the timer expires, then: the Attempt count is reset to 1; a new Bubble ID is randomly chosen; a bubble is sent with this bubble ID; the "Bubble sent" state is entered with the timer set to T1.

6.5.2. Client Transmission

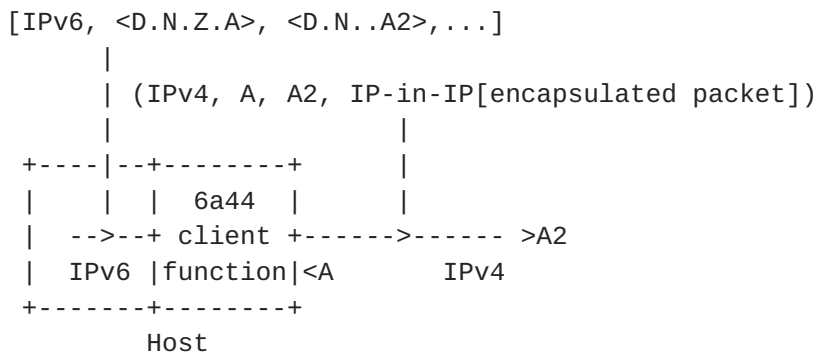
An 6a44 client transmits packets according to the following CT-x rules. In figures which illustrate these rules, symbols of [Section 5](#) are re-used; packets are represented as a succession of significant fields separated by commas, with sources preceding destinations as usual; != means different from.

CT-1 BUBBLE SENT BY A 6a44 CLIENT

```
(IPv4, A, B, UDP[::/96, <current Bubble ID>])
|
+-----+-----+
|         | 6a44 |         |
|         | client +----->----->B:W
|         |function|A:W<      UDP/IPv4
+-----+-----+
Host
```

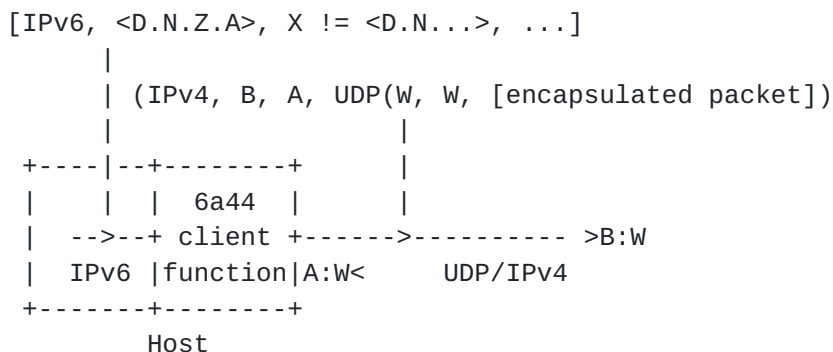
Bubbles are transmitted from time to time. Conditions of their transmission are specified in [Section 6.5.1](#), and their format is specified in [Section 6.3](#).

CT-2 IPv6/IPv4 PACKET SENT TO A HOST OF THE SAME SITE



If an IPv6 packet is submitted for transmission with ALL the following conditions satisfied, the 6a44 client MUST encapsulate the IPv6 packet in an IPv4 packet whose protocol is set to IP in IP (protocol = 41), and whose IPv4 destination is copied from the last 32 bits of the IPv6 destination: (1) the IPv6 source address is the 6a44-client IPv6 address; (2) the IPv6 destination is a 6a44 address of the same site (it has the same 80 bits as the 6a44-client IPv6 address); (3) either the IPv6 packet does not exceed 1280 octets, or it is longer but it does not exceed the IPv4 link MTU minus 20 octets and the IPv4 destination address starts with the IPv4 link prefix.

CT-3 IPv6/UDP/IPv4 PACKET TO A HOST OF ANOTHER SITE



If an IPv6 packet is submitted for transmission and ALL the following conditions are satisfied, the IPv6 packet MUST be encapsulated in a UDP/IPv4 packet whose destination is the 6a44-relay anycast address, and whose source and destination ports are both the 6a44 port: (1) the source address is the local 6a44-client IPv6 address; (2) The destination is not a 6a44 address of the same site (its first 80 bits differ from those of the 6a44-client IPv6 address); (3) The IPv6 packet does not exceed 1280 octets.

CT-4 IPv6 PACKET THAT DOESN'T CONCERN 6a44

If an IPv6 packet is submitted to the 6a44 client function for transmission with an IPv6 source address that is not the 6a44-client IPv6 address, the packet does not concern 6a44. It MUST be left for any other IPv6 transmission function that may apply (the source address can be a link-local address or a ULA of [\[RFC4193\]](#)).

6.5.3. Client Reception

Upon reception of an IPv4 packet, a 6a44 client applies the following CR-x rules:

CR-1 BUBBLE RECEIVED FROM A 6a44 RELAY

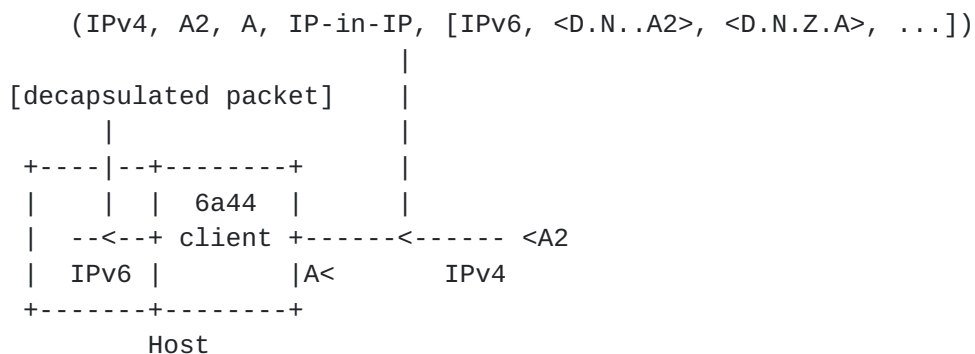
```

      (IPv4, B, A, UDP(w, w, [<D.N.Z>, <current bubble ID>]))
      |
+-----+-----+
|         | 6a44 |         |
|         | client +-----<-----<B:W
|         |         |A:W<      UDP/IPv4
+-----+-----+
      Host
      (updates D.N.Z)

```

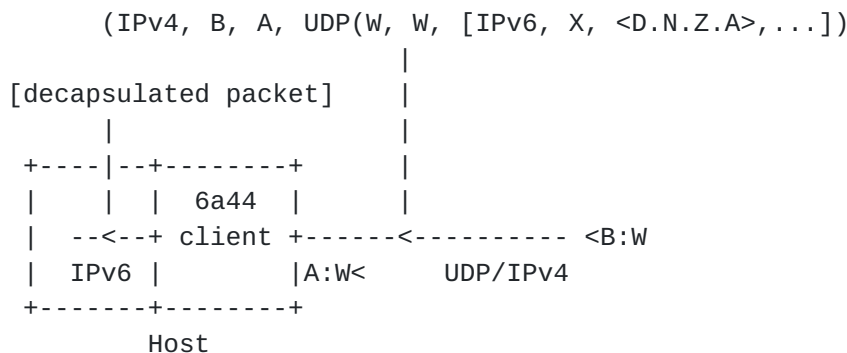
If ALL the the following conditions are satisfied (i.e. the packet is a 6a44 bubble from a 6a44 relay), the 6a44-client IPv6 address MUST be updated using the received IPv6 prefix D.N.Z: (1) the IPv4 packet contains a complete UDP datagram (protocol = 17, offset = 0, more-fragment bit = 0); (2) Both ports of the UDP datagram are the 6a44 port, and the payload length is enough to contain a 6a44-client IPv6 prefix and a Bubble ID but shorter than an IPv6-packet header(protocol = 17, UDP payload length = at least 20 octets and less than 40 octets; the received Bubble ID matches the current value of the Bubble-ID local variable.

CR-2 IPv6/IPv4 PACKET FROM A HOST OF THE SAME SITE



If ALL the following conditions are satisfied (i.e. the packet comes from a 6a44 client of the same site), the 6a44 client MUST decapsulate the inner packet and treat it as a received IPv6 packet: (1) the IPv4 packet contains a complete UDP datagram (protocol = 17, offset = 0, more-fragment bit = 0); (2) both ports of the UDP datagram are the 6a44 port, and the UDP payload is an IPv6 packet (UDP length of at least 40 octets, version = 6); (3) the IPv6 source address is one of the same site (the first 80 bits match those of the 6a44-client IPv6 address; (4) its last 32 bits are equal to the IPv4 source address; (5) the IPv6 destination address is the 6a44-client IPv6 address.

CR-3 IPv6/UDP/IPv4 PACKET FROM A HOST OF ANOTHER SITE



If ALL the following conditions are satisfied (i.e. the packet has been relayed by a 6a44 relay), the 6a44 client MUST decapsulate the inner packet and treat it as a received IPv6 packet: (1) the IPv4 packet contains a complete UDP datagram (protocol = 17, offset = 0, more-fragment bit = 0); (2) the UDP payload is an IPv6 packet (length of at least 40 octets, version = 6); (3) the UDP/IPv4 source address is the 6a44-relay UDP/IPv4 address; (4) the IPv6 destination address is the 6a44-client IPv6 address.

CR-4 RECEIVED ICMPv4 ERROR MESSAGE CONCERNING A 6a44 PACKET

If the 6a44 client receives an IPv4 error message of [[RFC0792](#)] that concerns a discarded 6a44 packet (i.e. if the copied header of the discarded packet is that of a transmitted packet according to CT-2 or CT-3), it SHOULD translate it into an ICMPv6 error message of [[RFC4443](#)], and then treat it as a received IPv6 packet. Translation of Type and Code conversions between IPv4 and IPv6 are described in [[RFC6145](#)] ([Section 4.2](#). - IPv4 error messages).

CR-5 RECEIVED IPv4 PACKET OTHER THAN 6a44

If ANY of the following conditions is verified, the received IPv4 packet does not concern 6a44 and MUST therefore be left for any other IPv4 reception function that may apply: (1) the IPv4 payload is neither UDP nor IPv6 (protocol = neither 17 nor 41), or protocol = 41 and IP version in the payload is not = 6); (2) the IPv4 packet is an IP-datagram fragment other than the first one (offset > 0); (3) the IPv4 packet contains the first or unique fragment of a UDP datagram (protocol = 17, offset = 0), with neither port equal to the 6a44 port.

6.6. 6a44 Relay Specification**6.6.1. Relay Reception in IPv6**

Upon reception of a packet via its IPv6 interface with a destination address starting with the 6a44-network IPv6 prefix, a 6a44 relay MUST apply the following RR6-x rules:

RR6-1 VALID IPv6 PACKET FROM OUTSIDE THE 6a44 ISP NETWORK

```
[IPv6, (X != <D...> AND != <Teredo(IPv4=B)>), <D.<N != B>.Z...>, ...]
|
(IPv4, B, N, UDP(W, Z, |
[encapsulated packet])) |
|
| +-----+ |
| >B:W | 6a44 |D/48< |
N:Z< ---<-----| relay |-----<----- D.N.Z...<
IPv4 | | IPv6
+-----+
```

If ALL the following conditions are satisfied, the IPv6 packet MUST be encapsulated in an UDP/IPv4 packet whose UDP/IPv4 destination is copied from bits 48 to 95 of the IPv6

destination address: (1) the IPv6 source address is not that of a 6a44 client of the ISP (it does not start with the 6a44-network IPv6 prefix); (2) the IPv6 source address is not a Teredo address whose embedded UDP/IPv4 address is the 6a44-relay anycast address; (3) the customer-site IPv4 address embedded in the 6a44 destination address is not the 6a44-relay anycast address; (4) the packet has at most 1280 octets.

RR6-2 INVALID IPv6 PACKET FROM OUTSIDE THE 6a44 ISP NETWORK

If ANY of the following conditions is satisfied, the IPv6 packet MUST be discarded : (1) the packet has more than 1280 octets (in this case, an ICMP Packet Too Big error message MUST be returned to the source); (2) the customer-site IPv4 address embedded in the IPv6 destination address is the 6a44-relay anycast address; (3) the IPv6 source address is a Teredo address whose embedded IPv4 address is the 6a44-relay anycast address.

[6.6.2.](#) Relay Reception in IPv4

Upon reception via its IPv4 downstream interface of an IPv4 packet that contains a complete IP datagram (fragment offset = 0 and more-fragment bit = 0), and that contain a UDP datagram whose UDP/IPv4 destination is the 6a44-relay UDP/IPv4 address, a 6a44 relay MUST apply the following rules:

RR4-1 BUBBLE FROM 6a44 CLIENT

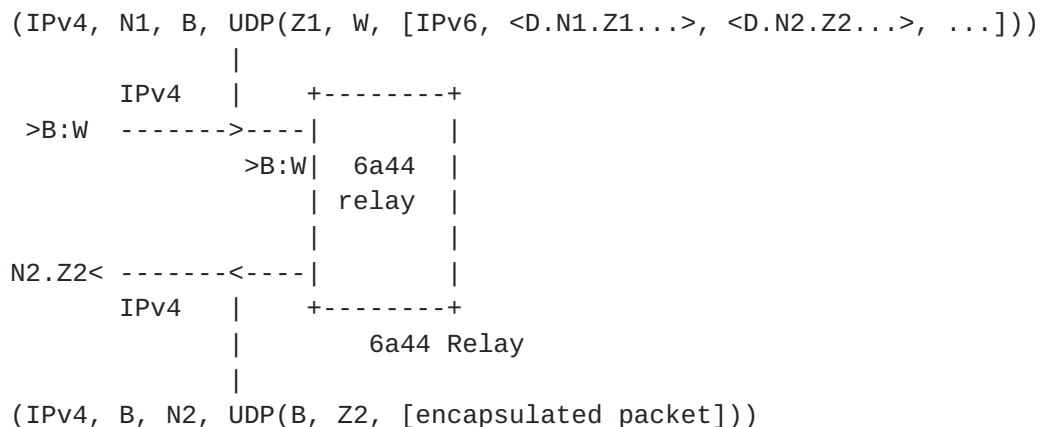
```

      (IPv4, N, B, UDP(Z, W, [::/96, bubble ID]))
          |
      IPv4  |  +-----+
    >B:W  ----->----|  |
          >B:W|  6a44  |
          |  relay  |
    N:Z<  -----<----|  |
      IPv4  |  +-----+
          |
          |
      (IPv4, B, N, UDP(B, W, [<D.N.Z>, bubble ID]))

```

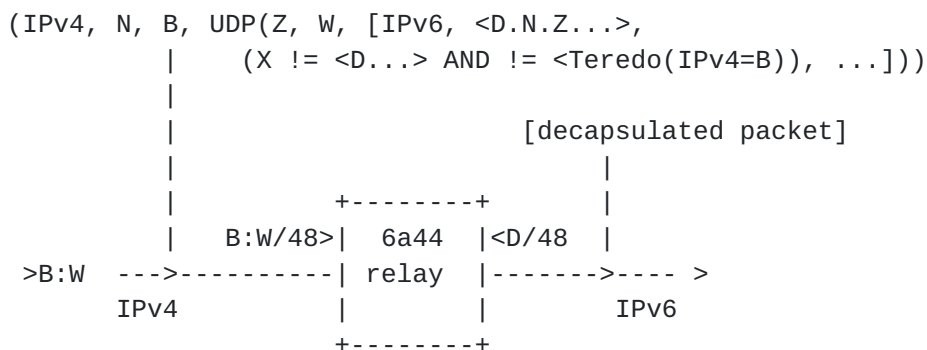
If the following condition is satisfied, the 6a 44 relay MUST return to the source a bubble derived from the received one by permuting its UDP/IPv4 source and destination, and by putting in its 6a44-client-IPv6-prefix field the received UDP/IPv4 source address: the UDP payload is a bubble, i.e has at least 20 octets and less than 40 octets

RR4-2 IPv6 PACKET FROM A 6a44 CLIENT TO ANOTHER 6a44 CLIENT



If ALL the following conditions are satisfied, the 6a44 relay MUST return back via its downstream IPv4 interface an IPv6/UDP/IPv4 packet containing the same encapsulated packet, having its UDP/IPv4 destination set to the UDP/IPv4 address found in the 6a44 destination address, and having its UDP/IPv4 source set to the 6a44-relay UDP/IPv4 address: (1) the IPv4 packet contains a complete UDP datagram (protocol = 17, offset = 0, more-fragment bit = 0); (2) the UDP payload is an IPv6 packet (length of at least 40 octets, version = 6); (3) the IPv6 source address starts with the 6a44-network IPv6 prefix followed by the UDP/IPv4 source address of the received packet; (3) the IPv6 destination address starts with the 6a44-network IPv6 prefix.

RR4-3 IPv6 PACKET FROM A 6a44 CLIENT TO A NON-6a44-CLIENT



If ALL the following conditions are satisfied, the 6a44 relay MUST decapsulate the IPv6 packet and forward it via the IPv6 interface: (1) the IPv4 packet contains a complete UDP datagram (protocol = 17, offset = 0, more-fragment bit = 0); (2) the UDP payload is an IPv6 packet (length of at least 40 octets, version = 6); (3) the IPv6 source address starts with the 6a44-network IPv6 prefix, followed by the UDP/IPv4 source address of the received packet; (4) the IPv6 destination

address does not start with the 6a44-network IPv6 prefix and is not a Teredo address whose embedded IPv4 address is the 6a44-relay anycast address.

RR4-4 RECEIVED ICMPv4 ERROR MESSAGE CONCERNING A 6a44 PACKET

If the 6a44 relay receives an IPv4 error message of [[RFC0792](#)] that concerns a discarded 6a44 packet (i.e. if the copied header of the discarded packet is that of a transmitted packet according to RR6-1 or RR4-2), it SHOULD translate it into an ICMPv6 error message of [[RFC4443](#)], and then treat it as a received IPv6 packet. Translation of Type and Code conversions between IPv4 and IPv6 are described in [[RFC6145](#)] ([Section 4.2](#). - IPv4 error messages).

RR4-5 INVALID IPv6/UDP/IPv4 PACKET

If ANY other case, the 6a44 relay MUST discard the packet.

[6.7](#). Implementation of Automatic Sunset

6a44 is designed as an interim transition mechanism, not to be used any longer than strictly necessary. Its sole purpose is to accelerate availability of IPv6 native addresses where, for any reason, CPE's cannot quickly be replaced, or where, for any reason, ISP networks cannot quickly support dual-stack routing or 6rd.

A 6a44-capable ISP can first have an increase of its 6a44 traffic, as more and more hosts behind IPv4-only CPEs support the 6a44 client function. But it should later have a decrease of this traffic as more and more CPE's operate in dual stack.

When this traffic becomes sufficiently negligible, it may, after due prior notice, discontinue 6a44-relay operation. This terminates its sunset procedure.

In a host that obtains a IPv6 native address by some other mean than 6a44, the effect of having the 6a44 function in its protocol stack is inexistent. OS providers may therefore keep this function in their code for many years. When it becomes clear that the number of users of this unction has become negligible they can delete it from later releases. This terminates their sunset procedure.

[7](#). Security Considerations

Incoming reachability:

Hosts that acquire 6a44 addresses become reachable from the Internet in IPv6 while they remain unreachable in IPv4 at their private IPv4 addresses.

For ordinary use, this should not introduce a perceptible new security risk for two reasons: (1) hosts can, without IPv6, use NAT44 hole-punching techniques such as ICE of [[RFC5245](#)] to receive incoming connections; (2) modern operating systems that support IPv6 have by default their own protections against incoming connections.

If nevertheless 6a44 reachability across an ordinary NAT44 has to be barred, this can be done by configuring its port-forwarding function with the 6a44 port bound to any internal address that is not assigned to any host. Thus, no bubble from a 6a44 relay can reach any 6a44-capable host, and this is sufficient to prevent hosts from using 6a44.

For more sophisticated uses with managed firewalls, default configuration are in general such that packets that are not explicitly authorized are discarded. Thus, 6a44 can be used only if the 6a44 port is consciously opened to incoming traffic.

Subscriber authentication:

Any authentication that applies to an IPv4 address extends its effect to 6a44 addresses that are derived from it.

Host-address spoofing:

With ingress filtering required in 6a44 ISP networks, and with address checks of [Section 6](#), no new IPv6 address-spoofing vulnerability is introduced by 6a44.

Address-and-port scanning:

To mitigate the (limited) risk of a malicious user trying to scan address-and-port IPv4 couples to reach a host, Teredo addresses contain 12 random bits [[RFC5991](#)]. 6a44 addresses have no random bits but contain local IPv4 addresses of clients. Since possible values of these addresses are not deterministically known from outside customer sites, and are in ranges that can be configured in typical NAT44s, some protection against address and port scanning is thus achieved. This protection may be less effective than that achieved with random bits, but is in any case better for 6a44 IPv6 addresses than for IPv4 addresses alone.

Denial-of-service:

Provided 6a44 relays are provisioned with enough processing power, which is facilitated by their being completely stateless, 6a44 introduces no denial of service vulnerabilities of its own.

Routing-loops:

A risk of routing-loop attacks has been identified in [\[draft-ietf-v6ops-tunnel-loops\]](#). Without precaution, it applies to some combinations of automatic-tunnel mechanisms such as 6to4, ISATAP, 6rd and Teredo. This risk does not exist with 6a44 for the following reasons:

1. When an packet enters a 6a44 relay via its IPv6 interface:

- + An IPv6/UDP/IPv4 packet cannot be sent to another 6a44 relay because its IPv4 destination would have to be 6a44-relay IPv4 address. This is prevented by rule RR6-1 of [Section 6.6.1](#).
- + If an IPv6/UDP/IPv4 packet is sent to the address of a 6to4 relay, 6rd relay, or ISATAP relay, it will be discarded there because these relays don't accept UDP/IPv4 packets.
- + If an IPv6/UDP/IPv4 packet is sent to a Teredo relay, it will be discarded there because: (1) Teredo relays check that the IPv4 addresses that is embedded in the IPv6 source address of a received IPv6/IPv4 packet does match the IPv4 source address of the encapsulating packet ([section 5.4.2 of \[RFC4380\]](#)); (2) encapsulating packets sent by 6a44 relays have the 6a44-relay anycast address as IPv4 source address; (3) a 6a44 relay forwards a received IPv6 packet as an IPv6/UDP/IPv4 packets only if its IPv6 source address is not a Teredo address whose embedded IPv4 address is the 6a44-relay IPv4 address.

2. When a packet enters a 6a44 relay via its IPv4 interface:

- + The received packet cannot come from another 6a44 relay (as just explained, 6rd relays do not send IPv6/UDP/IPv4 packets to other 6a44-relays).
- + If the IPv4 packet comes a 6to4 relay, a 6rd relay, or an ISATAP relay, its IPv6 encapsulated packet cannot be forwarded (the received packet is IPv6/IPv4 instead of being IPv6/UDP/IPv4, as required by rules RR4-2 and RR4-3 of [Section 6.6.2](#)).

- + If the received packet is an IPv6/UDP/IPv4 packet coming from a Teredo relay, this packet cannot have been sent to the Teredo relay by a 6a44 relay ((1) in order to reach the 6a44 relay, the IPv6 destination of the IPv6 encapsulated packet must be a Teredo address whose embedded IPv4 address is the 6a44-relay anycast address ([section 5.4.1 of \[RFC4380\]](#)); (2) a 6a44 relay does not forward via its IPv6 interface an IPv6 packet whose destination is a Teredo address whose embedded IPv4 address is the 6a44-relay anycast address (rule RR4-3 of [Section 6.6.2](#)).

6a44-relay spoofing:

In a 6a44 network, no node can spoof a 6a44 relay because ingress filtering prevents any 6a44-relay anycast address to be spoofed.

In a network that does not support ingress filtering (and therefore is not a 6a44 network):

- * 6a44 packets sent by 6a44-capable hosts are discarded in the IPv4 backbone because their IPv4 destination, the 6a44-relay anycast address, does not start with any ISP assigned prefix.
- * If an attacker tries to send to a 6a44-capable host a faked relay-to-client bubble, the probability that it would be accepted by its destination is negligible. It would require that all the following conditions be simultaneously satisfied:
 - + The UDP/IPv4 destination set by the attacker must reach a NAT44 node in which it is the external mapping of a 6a44 tunnel established by a 6a44-capable host.
 - + This host must be in the "Bubble sent" state, the only one in which it listens to bubbles when its ISP is not 6a44 capable. This state is taken only for a few seconds every 30 minutes (rule TM-5 of [Section 6.5.1](#)).
 - + This host accepts the bubble only if its bubble ID has the right value, an extremely unlikely possibility with a 64-bits randomly chosen Bubble ID (see [Section 6.5.1](#)).
- * If a 6a44-capable host, despite this being very unlikely, accepts a faked bubble, the effect is that it wrongly believes, for about 30 seconds, that it has an assigned public IPv6 address. All IPv6 packets it then sends with this address as source cannot be accepted by any destination (no relay will forward them, and no host of the same site will accept them). The consequence would therefore not impair security.

8. IANA considerations

IANA is solicited to assign:

1. 192.88.99.2 as the 6a44 IPv4 anycast address;
2. a registered UDP port as the 6a44 well known port. Proposed value is the currently unused 1027.

The choice of 192.88.99.2 as 6a44 IPv4 anycast address doesn't conflict with any existing IETF specification because:

- o It starts with the 6to4 prefix 192.88.99.0/24 [[RFC3068](#)].
- o It differs from the only currently assigned address that starts with this prefix (the anycast address of 6to4 relays 192.88.99.1 of [[RFC3068](#)]).

This choice is made to permit implementations of 6a44 relays both in physical nodes that are independent from any 6to4 relay or, if found more optimum, in nodes in which 6to4 relays and 6a44 relays are collocated.

9. Acknowledgments

This specification, whose origin is a convergence effort based on two independent proposals, [6rd+] and [[SAMPLE](#)], has benefited from various suggestions. Comments have been received during this process, in particular from Dave Thaler, Fred Templin, Ole Troan, Olivier Vautrin, Pascal Thubert, Washam Fan, and Yu Lee. Authors wish to thank them, and all others, for their useful contributions. Special recognition is due to Dave Thaler whose detailed review led to a few useful modifications.

10. References

10.1. Normative References

- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, [RFC 792](#), 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.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), February 2006.

10.2. Informative References

- [6rd+] Despres, R., "Rapid Deployment of Native IPv6 Behind IPv4 NATs (6rd+) - [draft-despres-softwire-6rdplus-00](#)", July 2010.
- [NAT444] Yamaguchi, J., Shirasaki, Y., Miyakawa, S., Nakagawa, A., and H. Ashida, "NAT444 addressing models - [draft-shirasaki-nat444-isp-shared-addr-03](#) - Work in progress", March 2010.
- [RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", [BCP 5](#), [RFC 1918](#), February 1996.
- [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.
- [RFC3053] Durand, A., Fasano, P., Guardini, I., and D. Lento, "IPv6 Tunnel Broker", [RFC 3053](#), January 2001.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", [RFC 3056](#), February 2001.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", [RFC 3068](#), June 2001.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", [BCP 84](#), [RFC 3704](#), March 2004.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", [RFC 4193](#), October 2005.

- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", [RFC 4380](#), February 2006.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", [RFC 4443](#), March 2006.
- [RFC5245] Rosenberg, J., "Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols", [RFC 5245](#), April 2010.
- [RFC5569] Despres, R., "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd)", [RFC 5569](#), January 2010.
- [RFC5626] Jennings, C., Mahy, R., and F. Audet, "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)", [RFC 5626](#), October 2009.
- [RFC5969] Townsley, W. and O. Troan, "IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) -- Protocol Specification", [RFC 5969](#), August 2010.
- [RFC5991] Thaler, D., Krishnan, S., and J. Hoagland, "Teredo Security Updates", [RFC 5991](#), September 2010.
- [RFC6081] Thaler, D., "Teredo Extensions", [RFC 6081](#), January 2011.
- [RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", [RFC 6145](#), April 2011.
- [SAMPLE] Carpenter, B. and S. Jiang, "Legacy NAT Traversal for IPv6: Simple Address Mapping for Premises - Legacy Equipment (SAMPLE) - [draft-carpenter-softwire-sample-00](#)", July 2010.
- [The Tool] Saint-Exupery, A de., "Wind, sand and Stars, Chap. III - The tool", 1939.
- [[draft-ietf-v6ops-tunnel-loops](#)] Nakibly, G. and F. Templin, "Routing Loop Attack using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations (Work in progress)".

Authors' Addresses

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

Email: despres.remi@laposte.net

Brian Carpenter
University of Auckland
Department of Computer Science
PB 92019
Auckland, 1142
New Zealand

Email: brian.e.carpenter@gmail.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com

Sheng Jiang
Huawei Technologies Co., Ltd
Q14, Huawei Campus - No.156 Beiqing Road
Hai-Dian District, Beijing 100095,
P.R. China

Email: shengjiang@huawei.com

