Internet-Draft
Expires: May 20, 2002

M. Py

November 21, 2001

Multi Homing Translation Protocol (MHTP)
draft-py-multi6-mhtp-01.txt

Status of this Memo

Copyright Notice

Abstract

This document describes a protocol for IPv6 Network Layer multihoming (MHTP) that does not affect the size of the routing table in the IPv6 DFZ (Default Free Zone) and does not use tunnels.

As a transition solution, and acknowledging the imperfections inherent to its design, MHTPËs main goal is to facilitate the initial deployment of IPv6 by providing a base for multihoming support.

MHTP is a Network Layer protocol, and the "Home" of MHTP is an IPv6 address. MHTP is a multi-address multi-homing protocol (MAMH).

MHTP provides fault tolerance, very good application compatibility, and simple configuration. It can be described as a semi-symmetric, end-to-end, NAT protocol.

Based on BGP4+ routing information, MHTP translates twice, leaving end-to-end traffic unchanged.

MHTP is a concept that has not yet been implemented. However, its building blocks are well known, which should facilitate a rapid development.

Table of contents

1. Conventions used in this document

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 RFC-2119 [1].

2. Introduction

Companies have been driven to multihome their IPv4 networks because of three important features that multihoming offers:
- Fault tolerance and continuous connectivity to the Internet by the means of several links (often from different providers)
- Better response times by being logically closer to the customer's network and avoiding bottlenecks, such as congested interconnects and Network Access Points, which also provides natural load balancing among different providers.
- Provider-independent addressing.

This document assumes that fault tolerance, better response times and PI addressing are the cornerstones of multihoming and proposes a protocol (MHTP) for IPv6 multihoming at the Network Layer.

3. Problem

The way Network Layer multihoming is achieved in IPv4 is by requesting a block of addresses (commonly called a PI block) independent of the providers' address space. The fault tolerance is achieved by multiple links advertising this block to multiple providers. Propagated in the DFZ by multiple sources, this achieves better response time (and natural load balancing) because the traffic will follow the path deemed as the best by the routing protocol (BGP4).

This approach has successfully provided IPv4 Network Layer multihoming and is in danger of succumbing to its own success. At the time of writing, the number of networks in the routing table of DFZ routers can reach 200,000 (some 120,000 public, plus one's own networks). Widespread issues linked to the size of the routing table have arisen, such as:

- Memory needed: A large number of installed routers have a 128MB DRAM limit, which will not contain the current growth of the DFZ's routing table for very long.
- Processing power and delay needed to handle updates: Despite optimized algorithms and hardware assistance, updating/indexing a table with 200,000 rows is no trivial task.
- Lookup speed / forwarding speed: The size of the routing table lookup is aggravated by the longest match rule.

At the time of the writing, the long-term evolution of the Internet is largely unknown. Whether or not we will ever have household appliances that are IPv6 enabled and multihomed remains to be seen, but the potential exists.

The multihoming mechanisms that are currently in place for IPv4 could easily be applied to IPv6, at the expense of scalability. If today one can purchase or build a router that handles 200,000 IPv4 routes at OC-192 speeds, the model that handles 500,000 IPv4 and 10,000,000 IPv6 networks at OC-768 speeds has not been manufactured yet.

4. Goals and non-goals

The goal of this document is to provide a mid-term design for an IPv6 Network Layer multihoming solution that is:
- Scalable beyond short-term needs.
- Easy to configure and administer.
- Transparent to the upper layers.

There is no goal at this time to provide an IPv4 solution. Although the protocol could be adapted to IPv4, the editor does not think that it can realistically implemented on today's Internet without a successful IPv6 implementation, and, by the time that is realized, solving the IPv4 problem might be a non-issue.

There is no goal to replace BGP4+. To the contrary, MHTP heavily relies on BGP4+ as the routing protocol.

5. Protocol design

The guidelines that have driven MHTP's design are as follows:

5.1 Use of NAT vs. Tunnels

Tunnels have been commonly discussed as a solution to the IPv6 multihoming problem. The editor's analysis is that tunnels that use the same protocol both as the payload and the encapsulation protocol are using tunneling mechanisms to achieve functions that are related to addressing matters. Tunneling IPv6 into IPv4 is fine; the tunneling mechanism is required to transport a protocol (IPv6) over a network (IPv4) that does not understand it. However, tunneling IPv6 into IPv6 for the purpose of solving what is indeed an addressing issue might not be the optimal solution. The editor is familiar with tunnels and has used them since the early ▒▒90s to transport IPX over IP on Novell servers. The use of tunnels for IPv6 multihoming purposes is indeed a way to hide the real IPv6 address to the network.

There are two potential problems with tunnels:
- The encapsulation process reduces the MTU.
- It is generally admitted that the migration to IPv6 will involve large amounts of tunneling IPv6 into IPv4. Adding a second layer of IPv6 into IPv6 tunneling, although feasible, might produce some implementation challenges, especially if encryption is to be used.

The editor thinks that a semi-symmetric, end-to-end NAT solution can be more efficient (because of the encapsulation overhead, or the lack thereof), more logical (because the multihoming problem is an addressing issue), and globally simpler than a tunneling solution.

5.2 Network Layer solution

Transport Layer solutions have been commonly discussed as a solution to the IPv6 multihoming problem.

There are three potential problems with Transport Layer multihoming solutions:
- They tend to be extensions of connection-oriented mechanisms (mostly TCP) that might not be optimal for connectionless protocols, such as UDP.
- They might not be completely in line with the OSI layer separation idea; a Transport Layer solution does not affect ICMP, (a layer 3 protocol). A multihoming solution without ICMP capabilities would be extremely difficult to troubleshoot.
- They might not be completely in line with broader ideas, such as end-to-end connectivity and the commonly admitted fact that routing is a layer 3 topic.

The editor's thinks that a semi-symmetric, end-to-end NAT solution can be easier to troubleshoot (because of separation between layers), and more logical (because it contains the multihoming problem within layer 3, where it belongs) than a Transport Layer based solution.

5.3 Centralized multihoming routing table

The one design element that has made possible today's IPv4 multihoming is the presence of each multihomed block in the DFZ's routing table. It is not issue-free, but it has delivered so far. The design proposed in this document does not intend to suppress the centralized routing table but rather tries to minimize the inconveniences that it causes.

5.4 Faster lookup with fixed-sized prefix

One of the aggravating factors of the size of the routing table is the longest math rule. This document proposes a fixed size prefix for multihoming purposes, which will allow faster routing table lookups by skipping the longest match rule process.

5.5 Separation of the routing tables

A widely recognized problem is the size of the DFZ's routing table, which causes issues both in terms of lookup speed and time required to process updates. These factors are aggravating the

fact that the routers that need to handle the DFZ's table are required to process extremely large numbers of packets per second and accommodate multiple, very high capacity circuits.
This document proposes a solution that dramatically reduces the size of the DFZ's table (for routers that need to process the bulk of the traffic) and dramatically reduces the traffic on routers that need to handle a large routing table by the following means:

5.5.1. Distinction between singlehomed and multihomed traffic: Two separate routing tables are to be kept: One for singlehomed traffic (the DFZ routing table) and the second one for multihomed networks (the MHTP routing table). This document formally distinguishes singlehomed and multihomed traffic. The main idea behind MHTP is to transform multihomed traffic into singlehomed traffic by the means of a semi-symmetric NAT process (described below).

5.5.2. Simplified routing on high-bandwidth routers: Since backbone routers would no longer need to handle multihomed traffic, the IPv6 DFZ could be summarized in the spirit that has guided its inception. To take the summarization to an unrealistically absurd level, the IPv6 backbone could be summarized at the /16 boundary, and the 6bone could be summarized at the /32 boundary without compromising the multihoming capabilities of MHTP.

5.5.3. Low load on routers containing the multihoming table: This document calls for a centralized routing table that would contain all multihomed prefixes. However, the routers ("MHTP rendezvous points") containing this table (the MHTP routing table that is likely to be bigger than the DFZ table itself) would not be required to process large amounts of traffic; only the very first packets of a session to a host on a multihomed network would hit these routers.

5.5.4. Restriction of the multihoming table distribution to a reasonably small number of MHTP rendezvous points.

5.6 Distribution of the translation load:

Extreme scalability will be achieved by sharing the processing power required by MHTP on each end router ("MHTP client", a router close to the user's workstation or server being accessed). The processing requirements of MHTP are likely to be comparable to those of NAT, so a router capable of handling n egress user NAT sessions would also be able to handle n egress MHTP sessions.

5.7 Use of BGP4+ to determine the best path:
MHTP is not a routing protocol and relies on BGP4+ for decisions regarding the selection of the optimal path. Implementation of MHTP should not change the way network administrators administer their BGP autonomous systems.

5.8 Stateful protocol:
There is no design guideline that calls for a stateful protocol. However, MHTP is clearly a stateful protocol for MHTP clients and MHTP rendezvous points. It is not connection-oriented since only one side keeps track of the current translations.

6. Protocol definition, description, and requirements

6.1 Terminology and descriptions of terms

6.1.1. Singlehomed address space: Any globally aggregatable IPv6 address EXCEPT those reserved for multihomed addresses. This address space has been allocated to a transit provider and is non-portable.

6.1.2. Multihomed address space: two blocks of globally aggregatable IPv6 addresses reserved for multihomed traffic. This document uses 2345::/16 and 3FFE:FFFF::/32 for explanatory purposes.

6.1.3. Singlehomed traffic:
Traffic whose destination IPv6 address is in the singlehomed address space.

6.1.4. Multihomed traffic:
Traffic whose destination IPv6 address is in the multihomed address space.

6.1.5. Site: an organization that receives IPv6 connectivity from two transit providers from two different TLAs (or 6bone pTLAs). If an organization that has been allocated a TLA (or a 6bone pTLA) wants to be allocated an MHTP prefix, they also need to have a block of singlehomed addresses from another TLA (pTLA) likely one of their direct competitors :-)

6.1.6 MHTP prefix size:
The fixed size /48 has been chosen because it is percept to be sufficient for most multihoming purposes.

6.1.7 MHTP prefix: a /48 block of multihomed addresses. MHTP prefixes are provider-independent and allocated to the end customer directly by the registry authority (or the 6bone). All connected MHTP prefixes are listed in the MHTP routing table.

6.1.8. MHTP translation block: a /48 block of singlehomed addresses. Recipients of MHTP prefixes must, for each MTHP prefix, reserve an MHTP translation block for EACH TLA and pTLA from which they have been allocated addresses.

6.1.9. MHTP translation table:
The dynamic table, in an MHTP client, that maps singlehomed addresses to multihomed addresses. The table contains /48 prefixes and does, in fact, map MHTP translation blocks to MHTP prefixes.
The MHTP translation table contains twelve columns: MHTP_prefix, MHTP preferred translation block and BGP metric, MHTP translation block #2, #3 and #4 and BGP metrics, (respectively MHTP_TB_1, MHTP_metric_1, MHTP_TB_2, MHTP_metric_2, MHTP_TB_3, MHTP_metric_3, MHTP_TB_4, MHTP_metric_4), MHTP_requests_sent, and two timers, MHTP_request_timer, MHTP_refresh_timer, and MHTP_key. The two timers are unsigned 16-bit integers, and key is a 64-bit unsigned integer.

6.1.10. MHTP request:
The request, sent from an MHTP client (or the client part of an MHTP endpoint) to an MHTP endpoint, contains a multihomed prefix that the client wants to translate into a singlehomed MHTP translation block. The address of the MHTP endpoint is unknown to the client when it sends the MHTP request. The MHTP request sent to the closest MHTP rendezvous point that will translate and forward it to the appropriate MHTP endpoint.

6.1.11. MHTP reply:
The reply, sent from an MHTP endpoint to an MHTP client upon request, contains the singlehomed MHTP translation block that is optimal for the requesting client.

6.1.12. MHTP client:
A router running MHTP in client mode. Customers of a certain size that have not been assigned an MHTP prefix would run in this mode.

The purpose of MHTP clients is to build and maintain the MHTP translation table (unique to each router). The MHTP client sends MHTP requests to MHTP endpoints and translates (by looking up the MHTP translation table built with MHTP replies) the destination IPv6 address, for egress traffic, from a multihomed address to a singlehomed address.

6.1.13. MHTP multihomed client:
An MHTP client that is part of the DFZ and that does NOT have a default route. The behavior of multihomed MHTP clients is the same as MHTP endpoints and will not be specifically addressed in this document.

6.1.14. MHTP endpoint:
A router running MHTP in both client and endpoint mode. Multihomed sites that have been allocated an MHTP prefix would run this mode.

MHTP endpoints have two purposes:

a) To translate back into multihomed traffic the singlehomed traffic sent from MHTP clients and MHTP rendezvous points. By a simple static subnet translation, the MHTP endpoint translates ingress singlehomed traffic into multihomed traffic. The endpoint translation process is stateless and looks up a very small table issued from static configuration. MHTP endpoints are likely to be found in data centers hosting multihomed server farms. The resources needed for MHTP endpoints are very low; basically, MHTP replaces the first 48 bits of the IPv6 address. This simple operation allows a single router to handle very large numbers of MHTP ingress packets without choking. Furthermore, the MHTP endpoint translation would be simple to implement in hardware and would enable IPv6 hardware-accelerated MHTP endpoints to handle ingress MHTP traffic at wire speed.

b) To provide MHTP clients with the information they need to

build their MHTP translation table by replying to MHTP requests.

Note that MHTP endpoints also need to be MHTP clients to handle the translation of egress traffic since they do not contain the full MHTP routing table. The MHTP client running on an MHTP endpoint is similar but not completely identical to an MHTP client only.

6.1.15. MHTP rendezvous point:
A router running MHTP in both client and rendezvous point mode. This is the mode that tier-1 transit providers (that have been allocated a TLA (or a 6bone pTLA)) would run.

MHTP rendezvous points have two purposes:
a) Translate and forward all MHTP requests from MHTP clients to appropriate MHTP endpoints.
b) Translate and proxy a controlled amount of multihomed traffic to appropriate MHTP endpoints.
As part of this translating / proxying, MHTP rendezvous points translate the destination IPv6 address of egress traffic to a singlehomed address like an MHTP client does except that the MHTP rendezvous point uses the MHTP routing table instead of the MHTP translation table.

6.1.16. MHTP routing table:
The MHTP routing table, present only in MHTP rendezvous points, contains all MHTP prefixes allocated. Technically, this is a BGP4+ routing table (MHTP rendezvous points are BGP4+ peers) with two extra characteristics: it contains only prefixes from the multihomed address space, and all the prefixes are of the same /48 size, which allows skipping the longest match rule and the use of optimized algorithms for lookups.

6.1.17. MHTP rendezvous point short-term table:
A dynamic table that contains pairs of source prefixes and destination multihomed MHTP prefixes. The MHTP rendezvous point will build and maintain this table in order to limit the number of non-MHTP multihomed packets that can be proxied for each pair.
This table contains five columns: SH_source_prefix, ASSOC_MHTP_prefix, PROXY_MHTP_packets, STATIC_maxpackets and

COUNT_unused. No timers are associated with entries in the short-term table. The table is dynamically built on-demand; the value of PROXY_MHTP_packets is reset every second. Entries are deleted when there has been no traffic for a specific entry for 30 seconds.
It is recommended that the MHTP_maxproxy is at least a 32-bit unsigned integer and COUNT_unused a signed byte.

6.1.18. Egress untranslated multihomed packet:
A multihomed packet that can be sent (by an MHTP client or an MHTP endpoint) to the best route to the multihomed address space (the closed MHTP rendezvous point) without a matching MHTP translation prefix in the MHTP translation table. The number of egress untranslated multihomed packets is strictly limited both at the MHTP client/endpoint and at the MHTP rendezvous point). The egress untranslated multihomed packet has some similarities with the TCP sliding window in the sense that it allows a certain amount of multihomed traffic to be sent to the MHTP rendezvous point (to be proxied, using a sub-optimal path) before waiting for an MHTP reply {"ships in the night") that will allow the client to transform the multihomed traffic into singlehomed traffic.
The egress untranslated multihomed packet is a double-edged sword: If the destination MHTP is valid and has valid MHTP translation block associations, it will reduce the latency of the first packet(s) of a yet unresolved MHTP translation. Otherwise, it will waste bandwidth. In either case, egress untranslated multihomed packets are a burden that needs to be carried by MHTP rendezvous points.

6.1.19. The translation of an egress untranslated multihomed packet sent at the same time as an MHTP request by an MHTP client by the rendezvous point is called MHTP proxying.

6.1.20. The translation of an untranslated multihomed packet coming from a non-MHTP router by the rendezvous point is called non-MHTP proxying.

6.2 Protocol requirements and implementation

6.2.1. MHTP is a feature of routers. Implementing MHTP at the host level would greatly increase the load of both MHTP

endpoints and MHTP rendezvous points.
Successful deployment of MHTP requires that there is an MHTP client in the path of multihomed traffic, which probably means that the edge of each stub network is MHTP enabled. However, some configuration of MHTP rendezvous points will allow them to be used as MHTP proxies and enable prefixes that do not have MHTP-enabled routers to access multihomed networks.

6.2.2. Workstations or servers that are sending traffic to a multihomed address are, as defined in 6.1, sending multihomed traffic. The main idea behind MHTP is that multihomed traffic, with the exception of the very first packets in a session, is transformed into singlehomed traffic at a router close to the source (the MHTP client) and transformed back into multihomed traffic at the last router (the MHTP endpoint).

6.2.3. The requirements to qualify for an MHTP prefix are as follows:
For a backbone MHTP prefix (a /48 block in the 2345:: range): two or more separate physical connections from to two or more transit providers from two or more different TLAs.
For a 6bone MHTP prefix (a /48 block in the 3FFE:FFFF:: range): two or more tunnels from two or more 6bone pTLAs.

6.2.4. The address space allocation requirements are as follows: For each MHTP prefix, a site must reserve a prefix of the same size (/48) from EACH of the different TLAs (pTLAs) from which the site receives IPv6 connectivity, for the sole purpose of MHTP prefix translation. Thus, if a site is multihomed to three different TLAs (pTLAs), the total amount of IPv6 addresses to allocate is four times the size of an MHTP prefix: one time for the MHTP prefix itself (that is going to be the block of addresses being accessed by clients and resolved in DNS), and three times for each transit provider prefix translation block that is essentially wasted by the MHTP translation process. This space allocation problem, along with the fact that IPv4 multihomed customers would be very reluctant to discontinue using their IPv4 PI block, is why the editor thinks that MHTP is not a realistically deployable solution for IPv4.

6.2.5. MHTP clients do not require BGP4+. A static default route or any other routing mechanism is enough to configure a router as an MHTP client. MHTP clients, if BGP4+ enabled, receive only the two aggregates from the multihomed address space 2345::/16

and 3FFE:FFFF::/32. The best path to the multihomed address space is the path deemed as best by BGP4+ for the multihomed prefixes. MHTP clients must not advertise multihomed prefixed outside of their autonomous system.

6.2.6. Routers at the edge of stub networks should discard ingress multihomed traffic (they should also discard singlehomed traffic which destination address in not part of the addressing space they have been allocated).

6.2.7. Each site that has been allocated an MHTP prefix needs to have one or more MHTP endpoints. Technically, only one router is needed. However, having only one MHTP endpoint router would be counter-productive with why the customer wants to be multihomed. MHTP endpoints must not have a default route (they are part of the DFZ) and require a BGP4+ feed from all their transit providers. MHTP endpoints advertise their assigned multihomed prefixes to each of their transit providers' MHTP rendezvous points and receive only the two aggregates from the multihomed address space 2345::/16 and 3FFE:FFFF::/32.

6.2.8. Each TLA (pTLA) is required to have one or more MHTP rendezvous points. Each MHTP rendezvous point exchanges the full MHTP routing table with other MHTP rendezvous points, typically all MHTP rendezvous points within the same TLA (pTLA) and with at least one (preferably two or more) MHTP rendezvous points from other TLAs (pTLAs) directly connected (tunneled). MHTP rendezvous points peering with MHTP clients or endpoints must only advertise the two aggregates from the multihomed address space 2345::/16 and 3FFE:FFFF::/32 to these clients or endpoints.

6.2.9. MHTP endpoints and rendezvous points are BGP4+ routers. BGP requirements, such as full mesh of iBGP peers, use of route reflectors [6], and other BGP4+ topics, are fully applicable to all MHTP routers running BGP4+. BGP4+ configuration should not be affected by MHTP.

The interaction between MHTP and BGP4+ is three-fold:

a) MHTP endpoints will lookup their BGP4+ routing table in order to reply to MHTP translation requests from MHTP clients.

b) Implementation of BGP4+ on MHTP rendezvous points could be optimized to take advantage of the specifics of the MHTP routing table such as all prefixes being of the same length.

c) MHTP rendezvous points will lookup the MHTP routing table in order to translate the destination address of proxied traffic (up to the amount allowed) and MHTP requests.

6.2.10. The full MHTP routing table is present in MHTP rendezvous points only. In the same spirit that TLAs and pTLAs collaborate on BGP4+ peering for the DFZ routing table, they also need to collaborate on MHTP rendezvous point peering.

6.2.11. The MHTP routing table might be considered as the DFZ routing table for multihomed traffic.

6.2.12. Although technically possible, it is not recommended to configure a router as both an endpoint and a rendezvous point. This configuration would defeat the scalability feature of MHTP. Configuring a router both as an MHTP endpoint and rendezvous point requires that router to run two completely separate instances of BGP4+.

6.2.13. MHTP rendezvous points must not forward singlehomed traffic. They must not advertise any singlehomed routes and must discard ingress traffic with a singlehomed IPv6 destination address except their own. MHTP rendezvous points must not receive any BGP4+ routes from peers that are not MHTP endpoints or rendezvous points themselves.
The recommended connection of an MHTP rendezvous point is two direct, high-speed links to core routers. The recommended routing setup for MHTP rendezvous points is:

- o Full peering with other MHTP rendezvous points.
- o All other peers must be sent only the two aggregates from the multihomed address space 2345::/16 and 3FFE:FFFF::/32.
- o Accept only MHTP prefixes from MHTP endpoints.
- o Do not accept any BGP4+ routes other than those mentioned above.
- o Use routes from an IGP such as OSPF or EIGRP (not to be redistributed) or a static route to ::/0 pointing to the directly connected core router(s) to provide egress singlehomed connectivity.

6.2.14. When reaching an MHTP rendezvous point, multihomed traffic to an MHTP prefix that is not present in the MHTP routing table must be discarded and an ICMP unreachable sent to the originating router.

6.2.15. Peering from or to a multihomed address is STRICTLY PROHIBITED in any situation. This prohibition applies both to

BGP4+ peering and MHTP peering. It would likely result in a deadlock.

6.3 MHTP requests, replies, and other datagrams

6.3.1. MHTP requests are sent by MHTP clients and MHTP endpoints. What triggers the sending of an MHTP request is ingress multihomed traffic that does not have a matching entry in the MHTP translation table. The source address of the MHTP request is the same as the singlehomed host that sent the traffic that triggered the MHTP request, and the destination address is unchanged, which is the address of the destination multihomed host.

6.3.2. Multihomed to multihomed traffic:
It is STRICTLY PROHIBITED to send an MHTP request with a multihomed source address. When an ingress packet with both the source and destination IPv6 addresses being multihomed arrives in an MHTP endpoint or MHTP multihomed client, the MHTP multihomed client or MHTP endpoint client's behavior differs from the regular MHTP client because it will send not one MHTP request; instead it will send one MHTP request per interface configured with a singlehomed address MHTP translation block.

The source address of each of these MHTP requests must be translated to the singlehomed address that belongs to the interface the MHTP request is sent from.

6.3.3. When an MHTP endpoint receives an MHTP request, it looks up its BGP4+ routing table to find out which interface is the best, as deemed by BGP4+, to send traffic back to the requesting client. The MHTP reply contains the MHTP translation block associated with the egress interface to send the reply back to the MHTP client, as well as up to three other MHTP translation blocks (from different interfaces) that match the MHTP address in the MHTP request, by order of their respective metrics. MHTP requests with a multihomed source address are discarded.

6.3.4. When an MHTP client or endpoint receives an MHTP reply, it authenticates it to verify that it was initiated from itself (explained in "Security considerations"), and then updates the entry of the translation table with the contents of the MHTP reply.

In an MHTP client, the BGP metric for each MHTP translation block is the same. MHTP clients, since they are not multihomed, are not able to use the load-balancing feature of MHTP and must not alter the order of the translation prefixes received in MHTP replies.

6.3.5. MHTP endpoints, since they send multiple MHTP requests, will receive multiple MHTP replies. MHTP endpoints must update the MHTP translation table by assigning the MHTP preferred translation and MHTP translation blocks #2, #3 and #4 in the order of their respective metrics. MHTP endpoints will optionally be able to load balance egress traffic to multihomed destinations.

6.3.6. No other IPv6 NAT
MHTP is a Network Address Translation mechanism. IPv6 to IPv6 NAT must be avoided at any cost for MHTP traffic (which includes both MHTP datagrams and singlehomed packets that are issued from an MHTP translation). IPv4 NAT of IPv6-encapsulated packets needs to be studied on a case-by-case basis and could be workable if end-to-end IPv6 connectivity is maintained. In other words, IPv6 encapsulated into IPv4 NATted traffic, if working, could accommodate MHTP traffic as well as regular IPv6 traffic. However, the editor thinks that no kind of NAT should be used in combination with MHTP regardless of whether it can be worked out.

6.3.7. MHTP client and endpoint timers and counters
Two timers are associated with each MHTP prefix in the MHTP translation table of MHTP clients and endpoints: MHTP_request_timer and MHTP_refresh_timer. Their purpose and relation to the various timeout values are described below.
The MHTP_request_timer starts when a new entry is added to the MHTP translation table (when a new entry is added, only the MHTP prefix field is populated).
The MHTP_request_timer is reset each time an MHTP keepalive is received for the matching entry in the MHTP translation table. For a description of MHTP keepalives, see "fault tolerance".

The MHTP_refresh_timer is reset each time it expires and triggers another sending of MHTP request(s).

6.3.8. MHTP_request_timeout

The MHTP_request_timeout is the value expressed in milliseconds of the life of an incomplete MHTP translation table entry (an entry in the translation table, triggered by an MHTP request, that never got an MHTP reply). The default value is 2,000 (2s). During the duration of MHTP_request_timeout, only MHTP_maxpackets_untranslated MHTP requests and egress untranslated multihomed packets can be sent for the matching MHTP prefix for the incomplete entry in the MHTP translation table. The MHTP client checks and updates the value of MHTP_requests_sent before sending an MHTP request/ MHTP untranslated packet. If MHTP_requests_sent reaches the value of MHTP_maxpackets_untranslated, no more MHTP requests/ MHTP untranslated packets can be sent until MHTP_request_timeout expires.
The MHTP_request_timeout also define the interval between MHTP keepalives when a valid MHTP table entry is present.

6.3.9. MHTP_refresh_timeout

The MHTP_refresh_timeout is the value, expressed in seconds that will trigger the re-sending of MHTP refresh request(s) for a given entry in the MHTP translation table. The default value is 120 (2 mn). MHTP refresh requests are identical to MHTP requests except that they are sent directly to the MHTP endpoint.

6.3.10. MHTP_maxpackets_untranslated

MHTP_maxpackets_untranslated is the number of MHTP requests and egress untranslated multihomed packets that can be sent for a given incomplete entry in the MHTP translation table. The default is 5.

6.3.11. MHTP rendezvous point timers and counters

The timer for the MHTP rendezvous point short-term table is fixed to one second.

6.3.12. MHTP_shorterm_blocksize_backbone

MHTP_shorterm_blocksize is the size of the block in the SH_source_prefix column for an auto-created entry related to the backbone multihomed address space (2345::/16). The default is /16. Acceptable values are 16 to 48. Note that one must be careful setting values such as /48 because it has the potential of creating a huge short-term table.

6.3.13. MHTP_shorterm_blocksize_6bone

MHTP_shorterm_blocksize is the size of the block in the SH_source_prefix column for an auto-created entry related to the 6bone multihomed address space (3FFE:FFFF:/32). The default is /32. Acceptable values are 16 to 48. Note that one must be careful setting values such as /48 because it has the potential of creating a huge short-term table.

6.3.14. MHTP_shorterm_flush

MHTP_shorterm_flush is the number of seconds that an unused entry in the short-term table will remain before it is flushed.

The COUNT_unused field is reset to MHTP_shorterm_flush each time the entry in the short-term table is matched and decremented every second. When it reaches zero, the entry is flushed.

6.3.15. MHTP_maxproxy is the number of non-MHTP multihomed packets that can be proxied, in one second, from and to a matching entry in the short-term table. The default is 100. Note that this value affects multihomed traffic only, NOT MHTP requests. The PROXY_MHTP_packets field is incremented each time the entry in the short-term table is matched and reset every second.

Since the multihomed packets that are to be MHTP-proxied are no different than the ones that are to be non-MHTP proxied, a compromise to allow MHTP-proxied packets is to count the number of MHTP requests against the number of proxied packets, by decrementing the PROXY_MHTP_packets field in the matching short-term table. See "compromises" for more details.

Setting this value to a very large number will effectively allow the rendezvous point to act as an MHTP client on behalf of other routers ("non-MHTP proxying") by allowing unconfigured MHTP clients or non-MHTP routers to send multihomed traffic that would be translated and sent to the appropriate MHTP endpoint. That would effectively transform the rendezvous point into an MHTP transparent proxy for all multihomed traffic. This approach could be used, with care, to facilitate the initial deployment of MHTP.

Changing the value of MHTP_maxproxy must be carefully thought through, however, because it will place a very high load on the MHTP rendezvous point. If one wants to enable proxying for a specific network, a static entry in the short-term table is preferred.

6.3.16. Static entries in the short-term table
Static entries can be configured in the short-term table in order to enable the rendezvous point to act as a proxy for specific networks or to deny any multihomed proxying for a specific network. Static entries are assigned the value 1 for

the COUNT_unused field, and never time out.

6.3.17. Short-term table lookup
Static entries in the short-term table are checked first in the order they were configured. When a match occurs, it is processed, and the lookup process stops.

- If STATIC_maxpackets is configured to 0, all multihomed proxying from / to the configured prefixes will be denied.
- If STATIC_maxpackets is configured to 1, MHTP_maxproxy will be used instead like in a dynamic entry.
- If STATIC_maxpackets is configured to 2, the PROXY_MHTP_packets will not be updated and will not be checked against MHTP_maxproxy.
- If STATIC_maxpackets is configured to any other number, the PROXY_MHTP_packets will be incremented and checked against STATIC_maxpackets

Static entries in the short-term table can have different prefix sizes than the size defined by MHTP_shorterm_blocksize. If ASSOC_MHTP_prefix is configured to zero, any multihomed traffic will match the entry.

Dynamic entries are checked next. Since dynamic entries are all of the same size, the order they are checked does not matter.

6.3.18. Examples of static short term table entries:

Note that PROXY_MHTP_packets is incremented/reset by the router itself and COUNT_unused is always 1 for a static entry

#	SH_source_prefix	ASSOC_MHTP_prefix	STATIC_maxpackets
1	::/0	0:0:0	0
2	2541:3672:/32	0:0:0	1
3	::/0	3FFE:FFFF:1234	2
4	::/0	0:0:0	2

#1: Deny any multihomed traffic proxying at all. This is not recommended since it will deny legitimate MHTP proxying as well.

#2: Allow non-MHTP multihomed traffic from 2541:3672:/32 to any multihomed destination to be proxied up to the limits of MHTP_maxproxy packets per second. This limits the proxying rate from that prefix. With default configuration, the proxying rate would have been MHTP_maxproxy packets per second per MHTP prefix.

#3: Allow unlimited proxying to the MHTP prefix 3FFE:FFFF:1234/48 regardless of the source.

#4: Allow unlimited proxying. Note that such an entry is preferred to setting the MHTP_maxproxy to a high value because it will save the rendezvous point the work of creating and checking the short-term table.

6.4 Compromises

The design of MHTP has required some compromises outlined below:

6.4.1. Sub-optimal path

The path of MHTP requests, of egress untranslated multihomed packets, and of non-MHTP proxied multihomed packets is not optimal. All of these will reach the closest MHTP rendezvous point where they will be translated or proxied and then sent the appropriate MHTP endpoint.
The number, location, bandwidth, and performance of MHTP rendezvous points can greatly affect multihomed performance.

6.4.2. Latency

The latency the very first MHTP packets destined to a yet unresolved MHTP translation block is higher than normal because these packets need to transit the MHTP rendezvous point.

6.4.3. Sub-optimal allocation of address space

For each MHTP prefix, if a given site is multihomed to n different TLAs / pTLAs, n MHTP translation blocks of the same size as the MHTP prefix are wasted by the MHTP process.

6.4.4. Proxying

An MHTP rendezvous point performs two types of proxying:

- MHTP-proxying, which allows the very first multihomed packets from an MHTP client to be sent before the MHTP reply that would allow that client to build the MHTP translation table arrives.
- non-MHTP proxying, which allows easier deployment of MHTP by enabling any host or router that is not MHTP enabled to access multihomed address.

Proxying in MHTP rendezvous points is not a stateful operation and does not distinguishes between the two types of proxying. The approximation made by decrementing the number of proxied packets for each MHTP request received is statistically correct but would not prevent a flood of non-MHTP multihomed packets from a given prefix to max out the proxying limits and therefore deny MHTP proxying.

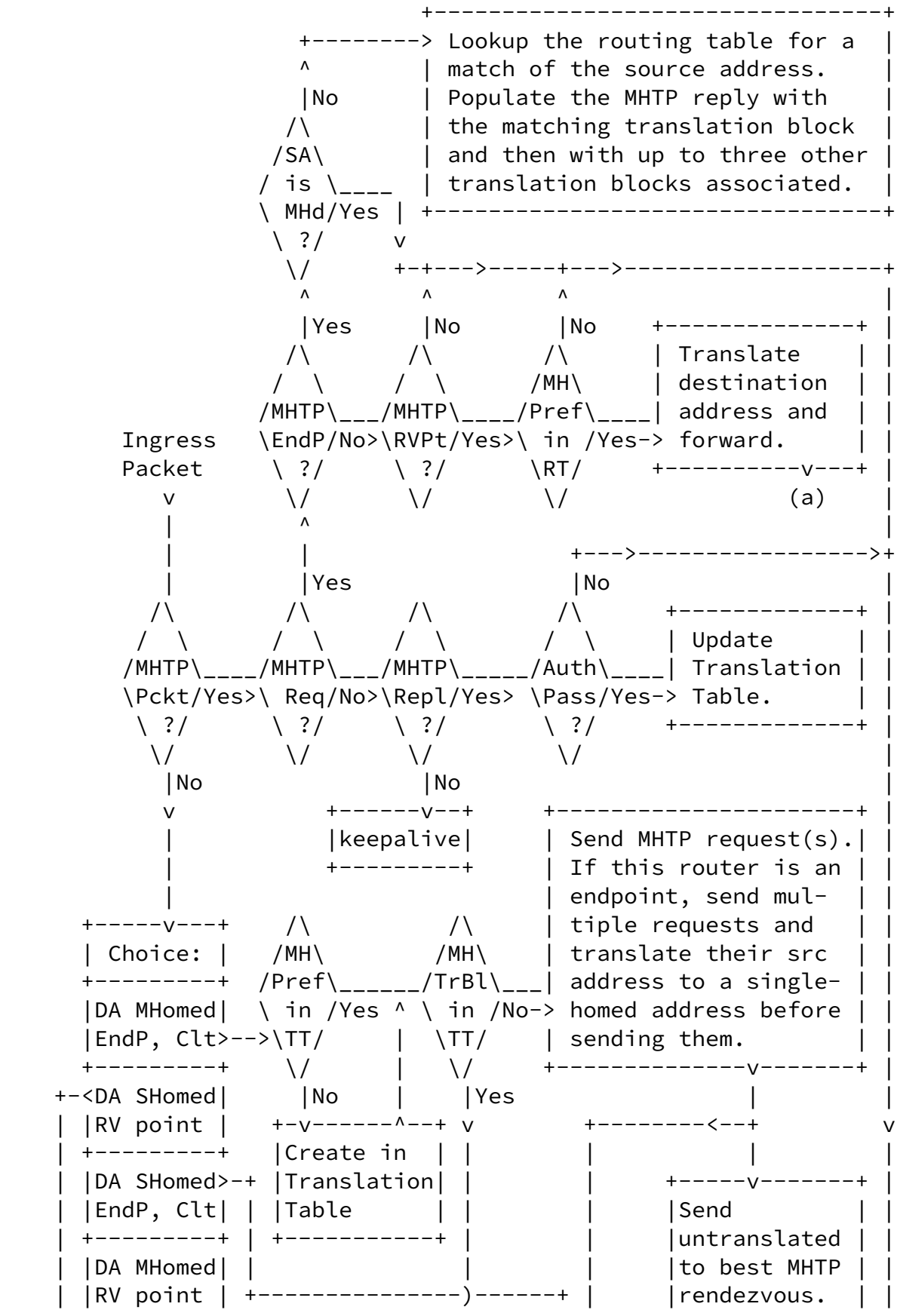
The editor thinks that:

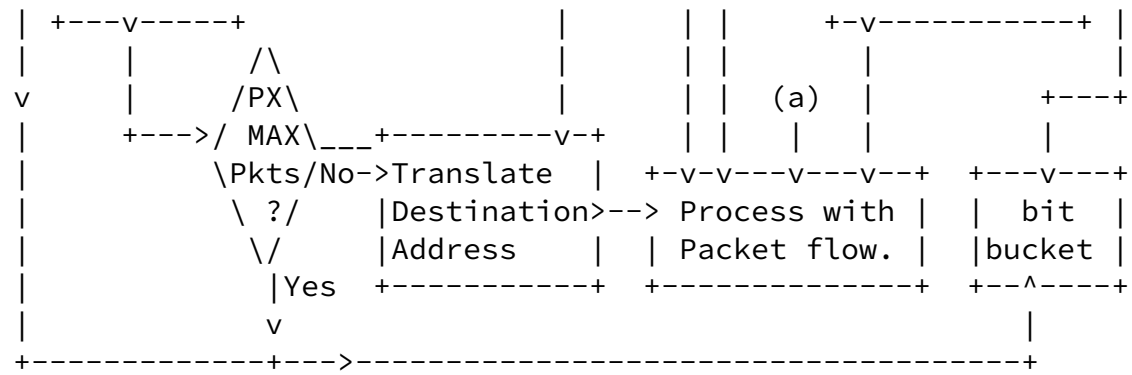
a) A MHTP endpoints MHTP rendezvous points only setup (all proxying, no clients) is not a viable solution (sub-optimal paths, high latency, does not scale on the rendezvous point side). MHTP rendezvous points have not been designed to be aggregators of the multihomed traffic.
b) Therefore, the amount of non-MHTP proxying must be kept under control, and the static entries in the rendezvous point short-term table will enable TLAs and pTLAs to fix deadline to their downstream to make their networks MHTP compliant.

6.4.5. Scalability

MHTP does not completely solve the scalability issue. It makes it better first by splitting the routing table into two independent parts and then by allowing the DFZ to be strongly summarized.

6.5.1 MHTP Flowchart





6.5.2 Flowchart abbreviations

SA is MHd?: Is the IPv6 source address multihomed?
MHTP EndP?: Is this router an MHTP endpoint?
MHTP RVPt?: Is this router an MHTP rendezvous point?
MH Pref in RT: Is the MHTP prefix present in the MHTP routing table?
MHTP Pckt?: Is this packet an MHTP datagram?
MHTP Req?: Is this datagram an MHTP request?
MHTP Repl?: Is this datagram an MHTP reply?
Auth Pass?: Does this datagram pass a valid security check?
MH Pref in TT?: Is the MHTP prefix in the MHTP translation table?
MH TrBl in TT?: Is there at least one translation block in the corresponding MHTP prefix entry in the translation table?
PX MAX Pkts?: Has the maximum number of proxied packets for the first source prefix/destination prefix match in the short-term table been reached?

DA Mhomed, EndP,Clt: The destination address is multihomed and this router is an MHTP client or an MHTP endpoint.
DA Mhomed, RV point: The destination address is multihomed and this router is an MHTP rendezvous point.
DA Shomed, EndP,Clt: The destination address is singlehomed and this router is an MHTP client or an MHTP endpoint.
DA Shomed, EndP,Clt: The destination address is singlehomed and this router is an MHTP rendezvous point.

6.5.3. Flowchart notes

The flowchart is a logical, high-level conceptual model of the way MHTP-related data flows inside an MHTP-enabled router. It has not been designed to closely map the actual configuration tasks. For example, the flowchart discards singlehomed traffic that hits a rendezvous point. In reality, no check is to be performed to decide if the router is a rendezvous point; an ingress traffic filter to discard singlehomed traffic configured only on rendezvous points would be simpler.

7. Fault tolerance

The purpose of MHTP is to translate multihomed traffic (to an MHTP prefix) to singlehomed traffic (to an MHTP translation block). If the preferred translation block (MHTP_TB_1) associated with the MHTP prefix becomes unavailable, MHTP should be able to recover preferably fast enough for upper layer connections not to timeout.

MHTP clients and endpoints send at periodic intervals (defined by MHTP_request_timeout) a keepalive datagram to each MHTP_TB_1. If the keepalive fails to return, traffic is immediately failed over to the translation block defined in MHTP_TB_2. If the keepalive fails to return three times in a row, a new MHTP request (that can be multiple in the case of an MHTP endpoint) is sent, and traffic to

the affected MHTP prefix is still being sent to MHTP_TB_2 in the meantime.
Keepalives are a reasonable waste of bandwidth. They are sent only when there is other traffic to a specific MHTP prefix.

MHTP multihomed clients and MHTP endpoints also check the validity of the route (NLRI) of each MHTP_TB_1 at the interval defined by MHTP_request_timeout and immediately (without waiting for the keepalive that they cannot send if there is no route) fail over the traffic to MHTP_TB_2 and send a new MHTP request (that can be multiple in the case of an MHTP endpoint) if no valid route is found.

To insure the validity and availability of MHTP_TB_2 when needed, MHTP clients and endpoints send at periodic intervals (defined by MHTP_refresh_timeout) an MHTP REFRESH request (that can be multiple

in the case of an MHTP endpoint) to each MHTP_TB_2. MHTP refresh requests are identical to MHTP requests except that they are sent to an MHTP translation block directly and do not transit the MHTP rendezvous point.

8. Load balancing

MHTP can use two different types of load balancing:
- Network load balancing, which is available to any network traffic. Since traffic translated by MHTP is no different than any other traffic, the only requirement of network load balancing is making sure that the MHTP translation process occurs before the load balancing process.
- MHTP load balancing. This future enhancement of MHTP leverages the fact that the MHTP translation table knows up to four MHTP translation blocks for each MHTP prefix will be described in a later revision of this document.

9. Application compatibility

End-to-end traffic is unaware of MHTP. The MHTP client translates the destination multihomed into a singlehomed address, and the MHTP endpoint translates it back to the same multihomed address that was sent by the originating end device. There is no way for an end device to know that the multihomed traffic it sends or receives has been or will be translated twice.
Therefore, MHTP is transparent to the upper layers and should not require any modifications of applications or network components at the Transport level and above.

The only two situations that have been identified so far as requiring special handling of MHTP are a) a firewall performing stateful packet inspection and/or dynamic stateful access filtering in case of asymmetric traffic and b) ICMP unreachable or related messages that would need to be reversed-natted to reach the original host.

10. Security considerations

By modifying the MHTP translation table, MHTP reply datagrams can alter the destination of traffic. With such potential for abuse, MHTP clients and endpoints must not process any MHTP reply datagram that is not a reply from a request they sent.

Each MHTP request, refresh request, and keepalive request contains a unique 64-bit random number, MHTP_key. The algorithm used to generate the key is left to each vendor as long as the key is unique and the sequence unpredictable even when the router boots.

BGP4+ peering between MHTP rendezvous points and other routers might bring security issues. These issues are not specific to MHTP and should be addressed the same way they are addressed for regular BGP peering.

11. IANA Considerations

Since there is no MHTP running code at the time of the writing, this document IMAGINES that the IANA has reserved the following:

- UDP Port number 7777/UDP.
- 2345::/16 Main MHTP block, /48 blocks be allocated by various registry authorities.
- 3FFE:FFFF::/32 6bone MHTP block, /48 blocks to be allocated by the 6bone.

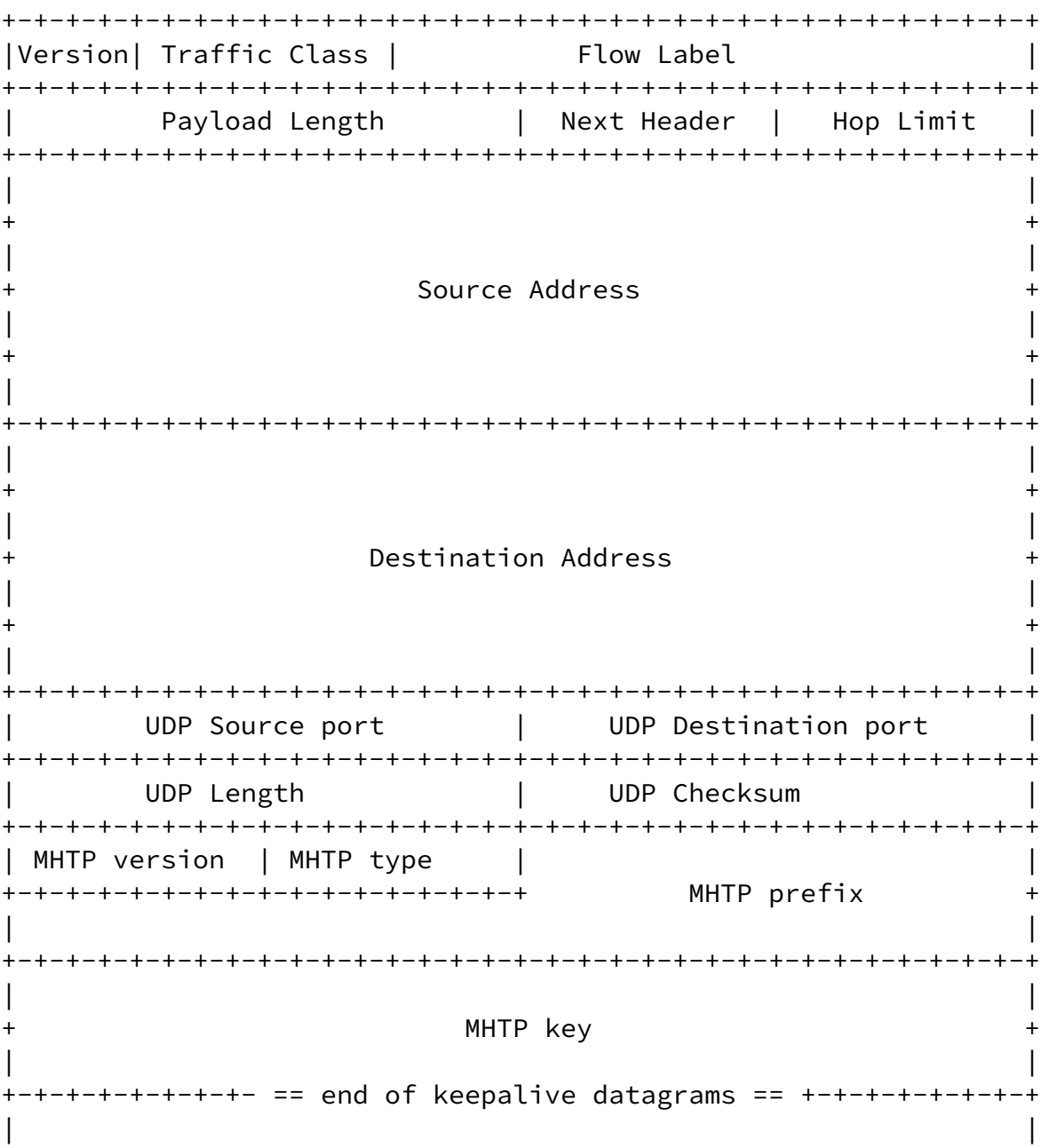
12. Registry considerations

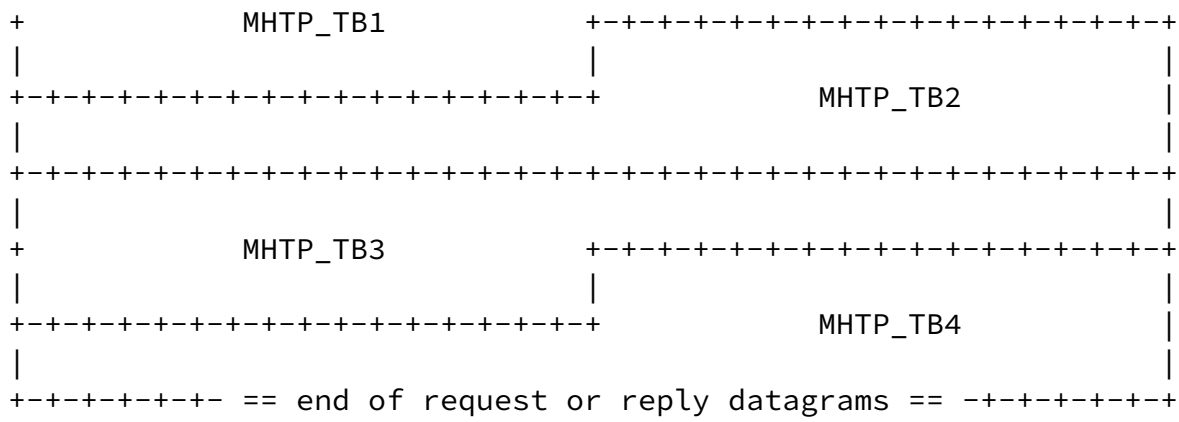
This document does not intend to define any policy about IPv6 address allocation. The editor thinks that MHTP block allocation policy should be a separate document. The following addresses, besides being fictitious, merely provides a possible, not even suggested, allocation scheme.

```
2345:0::/32 Reserved
2345:1::/32 ARIN
   2345:1:1::/48 American company
2345:2::/32 RIPE
   2345:2:1::/48 European company
2345:3::/32 APCNIC
   2345:3:1::/48 Asian company
3FFE:FFFF::/32 6bone
   3FFE:FFFF::0:/48 M. Py
   3FFE:FFFF::1:/48 The first to provide MHTP running code
   3FFE:FFFF::2:/48 The second to provide MHTP running code
```

13. Datagram structure

13.1 MHTP datagram structure. All MHTP datagrams begin with the same structure and are carried over UDP port 7777.





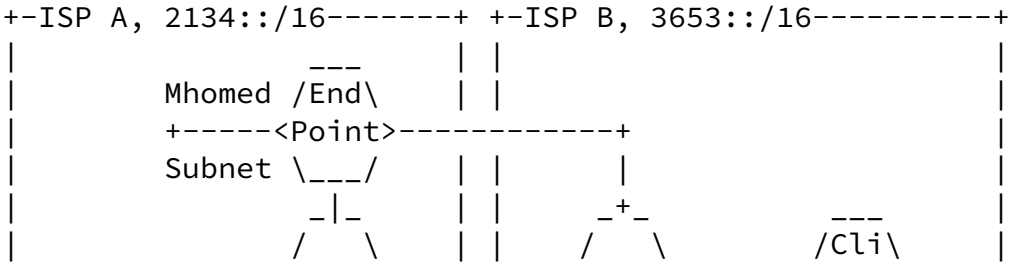
Version	4-bit Internet Protocol version number = 6.
Payload Length	48 for requests and replies, 24 for keepalive requests and replies
Next Header	8-bit unsigned integer. 17: UDP
UDP destination port	7777: MHTP
UDP length	40 for requests and replies, 16 for keepalive requests and replies.
MHTP version	1
MHTP type	1 = request 2 = reply 3 = keepalive request 4 = keepalive reply

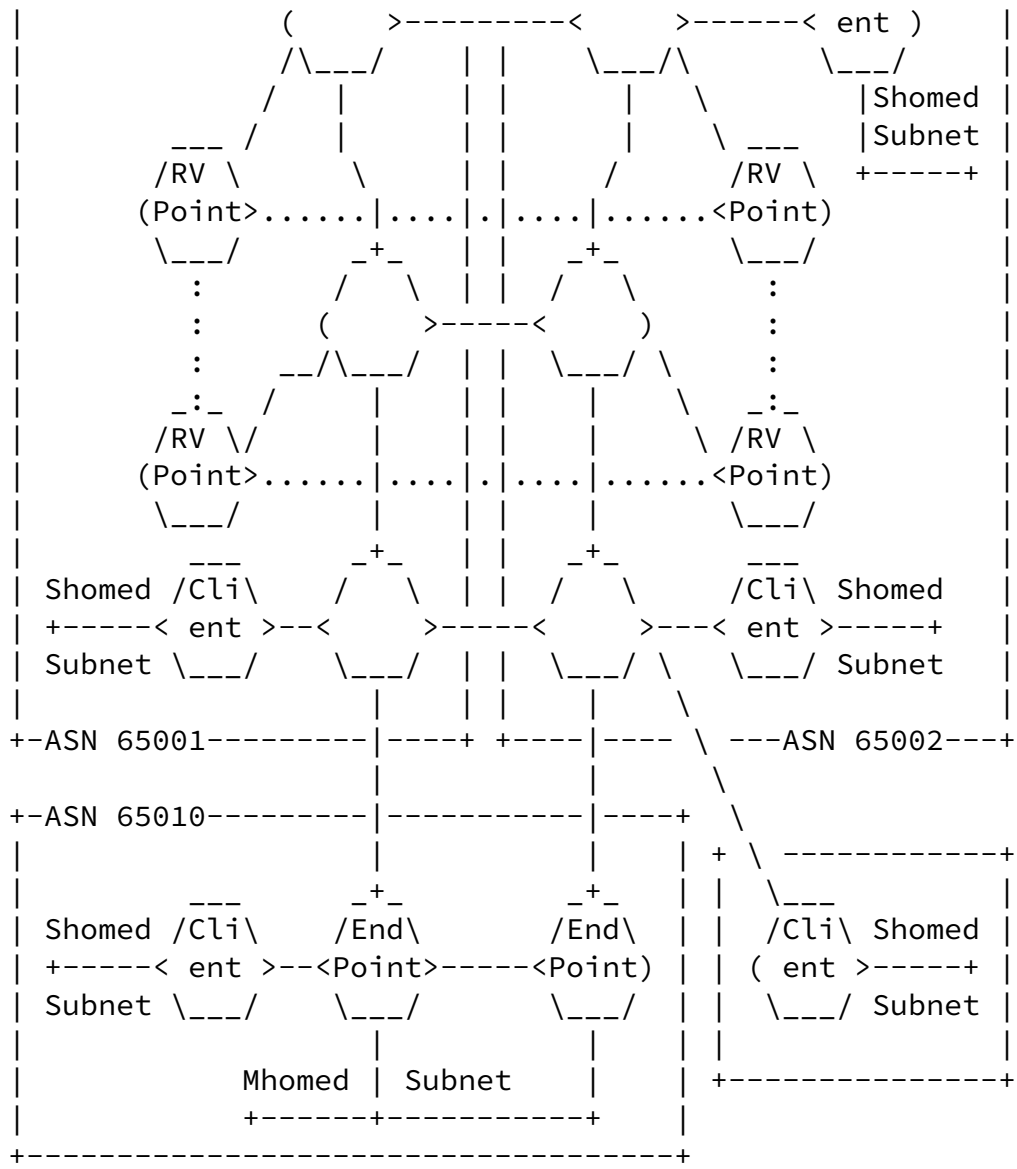
13.2 Packets or datagrams that have been translated by MHTP are no different than the original packet or datagram except the destination address. In fact, packets or datagrams that have been translated twice (to singlehomed by the MHTP client and back to

multihomed by the MHTP endpoint) should be identical to the original.

14. Topology

14.1 Sample topology diagram





14.2 Topology notes

14.2.1 The number of MHTP clients must be strictly controlled. For singlehomed customers, there are no advantages of having MHTP clients all over their network.

14.2.2 Customers connected with a single physical link must not have any MHTP clients and must send untranslated traffic to their transit provider.

14.2.3 Tier-2 transit providers are required encouraged to provide MHTP client services to their customers connected with a single physical link and are encouraged to provide MHTP client services to all of their customers using multihomed MHTP clients.

15. Statement of direction

- The editor thinks that there is no good viable long-term IPv6 multihoming solution at the time of writing. MHTP does not pretend to be one either.
- There is some danger in people requesting TLAs for the sole purpose of being multihomed.
- To some extent, deployment of IPv6 has been or will be delayed by the lack of a solid IPv6 multihoming solution.

The motivations behind the design of MHTP are:

- A good waiting solution: Between the increased scalability provided by MHTP and the natural increase in router processing power and memory, will keep the IPv6 DFZ cleanly summarized until the perfect multihoming solution is invented.

- A proven mechanism: Uses BGP4 and a centralized routing table, which is the only proven mechanism. MHTP is an evolution in the tradition of incremental changes rather than a revolution.

- A foundation for other MAMH IPv6 solutions.

- A consensus builder: A middle solution, between multihoming the same way it is done in IPv4 and solutions that would be consider too radical by many.

16. Revision History

November 21, 2001
- Submitted as http://search.ietf.org/internet-drafts/draft-py-multi6-mhtp-01.txt.
- Editorial changes.
- Changes regarding provider-independent motivations.
- Added: 18. Compliance with the requirements and 19. Full Copyright Statement.
- Moved 18. References to 20. and 19. Editor's address to 21.

August 20,2001 version -01b
- Not submitted. Text available as: http://arneill-py.sacramento.ca.us/draft-py-multi6-mhtp-01.txt
- Minor editorial changes.
- Added: 16. Revision history and 17. Acknowledgements.
- Moved: 16. References and 17. Editor's address to: 18. References and 19. Editor's address

August 14,2001 version -01a
- Not submitted. Text available as: http://arneill-py.sacramento.ca.us/draft-py-multi6-mhtp-01.txt
- Minor editorial changes.
- Added: 14. Topology and 15. Statement of direction.
- Moved: 14. References and 15. Editor's address to: 16. References and 17. Editor's address

August 6,2001 version -00
- Original submission to ther IETF as: http://search.ietf.org/internet-drafts/draft-py-multi6-mhtp-00.txt

17. Acknowledgements

- Pekka Savola for reviewing the draft and bringing up ICMP issues.

18. Compliance with the requirements

This chapter details the compliance of this document with [11. B. Black, V. Gill, J. Abley, "Requirements for IP Multihoming Architectures", work in progress, http://www.ietf.org/internet-drafts/draft-ietf-multi6-multihoming-requirements-02.txt, November 2001.]

Chapter numbers are from the document mentioned above.

3.1.1 Redundancy

MHTP does not present significant changes in terms of redundancy

with the currently implemented IPv4 solution.

3.1.2 Load sharing

MHTP does not present significant changes in terms of load sharing with the currently implemented IPv4 solution.

3.1.3 Performance

MHTP does not present significant changes in terms of performance with the currently implemented IPv4 solution.

3.1.4 Policy

MHTP is simply added to the list of modules that look at policy (i.e. a local config table) as part of the routing process. That would be "process with packet flow" in 6.5.1 [MHTP Flowchart].

3.1.5 Simplicity

MHTP will be slightly more complex to implement than the current IPv4 solution because there will be a few more routers (the MHTP rendezvous points) to configure. This is by far balanced by the fact that MHTP endpoints will not require a full MHTP table and

will be simpler to configure. Overall, MHTP is not substantially more complex than current multihoming practices.

3.1.6 Transport-layer Survivability

MHTP provides a significant improvement of transport-layer survivability by the use of keepalives that are sent by the MHTP router, a lot closer to the host than the current solution.

3.2.1 Scalability

- Scalability on most routers is improved by two orders of magnitude (100 times). The size of the routing table will be divided by 100. With a 120,000+ table at the time of writing, it is reasonable to assume that the same table, if summarized correctly, would be 1,200 or smaller, which is two orders of magnitude. MHTP makes possible a "8K routing table", that would

be the maximum size when all 8,192 possible TLAs have been allocated.

- Scalability on MHTP rendezvous points (the weak point of MHTP) is still improved by two orders of magnitude:

  a) The size of the MHTP table shall be a tenth of the existing public routing table. This number (12,000) is the number of allocated ASNs at the time of writing, which dictates the number of multihomed sites [11].

  b) MHTP rendezvous point do not process as much traffic (they process only the very first packets of a given session). Combined with the fact that the MHTP routing table has a fixed size, it is reasonable to assume that an MHTP router could service ten times more prefixes than a regular router.

MHTP will provide a solution that is two orders of magnitude, or about 100 times, more scalable than the current solution. That is 1,000,000 multihomed IPv6 sites with currently available hardware.

3.2.2 Impact on Routers

- Changes to the routers require an MHTP implementation. The main component of MHTP (BGP) already exists and requires only optimization (which could be delayed to accelerate deployment). The other components (NAT/lookup) are based on well-known technologies and should not require un-reasonable development efforts.

All routers need not to be changed, only clients, endpoints and rendezvous points. End customers and tier-2 transit providers that are not multihomed do not require MHTP-capable routers.

Long-term scalability will require tier-2 transit providers to configure MHTP; however the minimum requirements for MHTP initial deployment are two MHTP-capable routers for each TLA or pTLA.

Not only MHTP does not prevent single-homed operations, but it does provide access to compliant multi-homed networks from

unmodified single-homed networks.

3.2.3 Impact on Hosts

MHTP does not require any host modifications.

3.2.4 Interaction between Hosts and the Routing system

MHTP does not require any specific host-to-router communications.

3.2.5 Operations and management

As every new protocol, MHTP will require monitoring commands and SNMP OIDs, among other things. At the current stage of development, there is no compelling reason to think that reasonable monitoring tools would not be developed as part of a vendorËs implementation.

3.2.6 Cooperation between Transit Providers

MHTP does not require site-specific cooperation between transit providers.

4. Security

Refer to 10. Security Requirements

19. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

20. References

1. Bradner, S, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, Harvard University, March 1997.

2. Deering, S. and R. Hinden, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

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

4. Carpenter, B., "Architectural Principles of the Internet", RFC 1958, June 1996.

5. Egevang, K. and Francis, P., "The IP Network Address Translator (NAT)", RFC 1631, May 1994.

6. T. Bates, R. Chandra, E. Chen, "BGP Route Reflection - An Alternative to Full Mesh IBGP", RFC2796, April 2000.

7. P. Marques, F. Dupont, "Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing", RFC 2545, March 1999.

8. A. Heffernan, "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

9. Y. Rekhter, T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

10. M. Blanchet, "IPv6 Address Space Reserved for Documentation", work in progress, http://search.ietf.org/internet-drafts/draft-blanchet-ngtrans-exampleaddr-01.txt, July 2001.

11. B. Black, V. Gill, J. Abley, "Requirements for IP Multihoming Architectures", work in progress, http://www.ietf.org/internet-drafts/draft-ietf-multi6-multihoming-requirements-02.txt, November 2001.

21. Editor's address

arn-py@arneill-py.sacramento.ca.us
or mpy@ieee.org